

DISEÑO INSTRUCCIONAL BASADO EN COMPETENCIAS MEDIADO POR
TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN (TIC's) PARA LA
ASIGNATURA INGENIERÍA DEL SOFTWARE DEL PROGRAMA ACADÉMICO
DE INGENIERÍA DE SISTEMAS E INFORMÁTICA.

TRABAJO DE GRADO PARA OPTAR AL TÍTULO DE
INGENIERO DE SISTEMAS

NADIA ALEXANDRA CARREÑO AVELLA
PAOLA CAROLINA ESPINOSA RODRÍGUEZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECAÑICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2007

DISEÑO INSTRUCCIONAL BASADO EN COMPETENCIAS MEDIADO POR
TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN (TIC's) PARA LA
ASIGNATURA INGENIERÍA DEL SOFTWARE DEL PROGRAMA ACADÉMICO
DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

NADIA ALEXANDRA CARREÑO AVELLA
PAOLA CAROLINA ESPINOSA RODRÍGUEZ

TRABAJO DE GRADO PARA OPTAR AL TÍTULO DE
INGENIERO DE SISTEMAS

DIRECTOR

ING. LEONARDO AUGUSTO CAMARGO CORZO

CODIRECTORES

ING. FERNANDO ROJAS MORALES

ING. SERGIO EDUARDO GÓMEZ ARDILA

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2007

AGRADECIMIENTOS

Queremos expresar nuestros agradecimientos:

Al profesor Leonardo Augusto Camargo Corzo, por su dirección, orientación, sugerencias y estímulo, por estar pendiente de los detalles y por haber dedicado parte de su tiempo a escucharnos.

A la Dra. Clara Inés Peña de Carrillo, por disponer los recursos y escenarios necesarios para que este proyecto pudiera ser realizado.

Al profesor Fernando Rojas Morales por sus valiosos aportes e ideas.

Al ingeniero Sergio Eduardo Gómez, por su paciencia, disposición y constante participación en el asesoramiento de este trabajo.

A los ingenieros vinculados de una u otra forma con el desarrollo del proyecto a través del CENTIC.

A todos, muchas gracias...

DEDICATORIA

*Ante todo gracias a papito Dios
por corresponder a mi amor
y darme mas de lo que le pido.
A ti padre por ser mi fuerza,
mi motor e inspiración
para cada logro de mi vida, te amo.
A ti madre porque con tu cariño
y apoyo me has ayudado y
enseñado a armar el
rompecabezas de la vida.
A las lucecitas que iluminan mi vida
mis hermanos y mi sobrina
por los juegos, conversaciones y
momentos vividos,
y a ti amor por llegar y
hacer más perfecto aquello en lo que creo...
Sencillamente ustedes son la base de mi vida.*

*A mis profesores, compañeros de trabajo,
amigos y todos aquellos que hicieron posible
la elaboración de este trabajo.*

Nadia Alexandra

DEDICATORIA

A Dios, por todas las bendiciones.

*A mi papá por ese ejemplo de vida,
por ser la fuerza y la luz
en el camino que he recorrido hasta hoy.
A mi mamá por el apoyo, la comprensión,
por la voz de aliento, y por todo el amor.
Son ustedes dos, el mejor regalo de Dios.*

*A mi hermana, mi sobrina,
a ese hombre excepcional.
A mi familia, amigos y compañeros,
y a todos aquellos que de una u otra manera
estuvieron a mi lado.*

Paola Carolina

TABLA DE CONTENIDO

INTRODUCCIÓN	18
1. DESCRIPCIÓN DEL PROYECTO	19
1.1 PLANTEAMIENTO DEL PROBLEMA	19
1.2 JUSTIFICACIÓN	21
1.3 OBJETIVOS	22
1.3.1 Objetivo General	22
1.3.2 Objetivos Específicos	22
1.4 VIABILIDAD DEL PROYECTO	23
1.5 IMPACTO	23
2. MARCO TEÓRICO	25
2.1 DISEÑO INSTRUCCIONAL	27
2.1.1 Formación superior basada en competencias	28
2.1.2 Teorías del aprendizaje	30
2.1.3 Metodologías de diseño instruccional para programas de formación	68
2.2 DISEÑO DE MATERIALES	80
2.2.1 Objeto de aprendizaje	80
2.2.2 Aproa	84
2.2.3 Estándar SCORM	86
2.2.4 Ingeniería del software	94
2.2.5 Lenguaje unificado de modelado (UML)	112
2.2.6 JavaScript	116
2.2.7 Macromedia flash 8 profesional	116
3. SEGUNDA FASE. DISEÑO INSTRUCCIONAL	118
3.1 ETAPAS PRELIMINARES AL DISEÑO INSTRUCCIONAL	119
3.2 ETAPAS DEL DISEÑO INSTRUCCIONAL	120
3.2.1 Diagrama secuencial de actividades de aprendizaje	121
3.2.2 Establecimiento de los saberes y haceres	128
3.2.3 Establecimiento de la relación propósitos-contenidos	130
3.2.4 Establecimiento de la estructuración modular	132
3.2.5 Establecimiento de la planeación curricular	137
4. TERCERA FASE: DISEÑO Y PRODUCCIÓN DE UN OBJETO DE APRENDIZAJE	142
4.1 CONCEPTO Y CARACTERÍSTICAS DE UN OBJETO DE APRENDIZAJE	142
4.1.1 Sobre el nombre del objeto de aprendizaje	144
4.1.2 Sobre el objetivo del objeto de aprendizaje	144
4.1.3 Sobre el contenido del objeto de aprendizaje	145

4.1.4	Sobre la aplicación del objeto de aprendizaje	145
4.1.5	Sobre la evaluación del objeto de aprendizaje	145
4.1.6	Sobre los vínculos de profundización del contenido	146
4.1.7	Sobre la declaración de autoría del contenido	146
4.1.8	Proceso de generación de un objeto de aprendizaje	146
4.2	ETAPA DE INICIO	146
4.3	ETAPA DE ELABORACIÓN	148
4.3.1	Casos de uso – documento soporte	148
4.3.2	Casos de uso – audio	152
4.3.3	Casos de uso – animación	157
4.3.4	Casos de uso – gráficos y tablas	162
4.3.5	Casos de uso – aplicativo	166
4.3.6	Diagramas de secuencia – documento soporte	171
4.3.7	Diagramas de secuencia – audio	173
4.3.8	Diagramas de secuencia – animación	174
4.3.9	Diagramas de secuencia – aplicativo	176
4.3.10	Diagramas de secuencia – gráficos y tablas	177
4.4	ETAPA DE CONSTRUCCIÓN	179
4.4.1	Descripción del objeto de aprendizaje sobre la plantilla	180
4.4.2	Secuenciación de objetos de aprendizaje	194
4.4.3	Ficha de catalogación de objeto de aprendizaje	195
5.	ESTRUCTURACIÓN Y PUESTA EN FUNCIONAMIENTO DEL PORTAL WEB DEL PROFESOR	220
5.1	LINK INICIO	220
5.2	LINK CURRÍCULO	221
5.3	LINK INVESTIGACIÓN	221
5.4	LINK EXTENSIÓN	221
5.5	LINK ADMINISTRACIÓN	221
5.6	LINK DOCENCIA	222
5.7	LINK ENLACES DE INTERÉS	222
6.	ELABORACIÓN DE LA EVALUACIÓN DEL OBJETO DE APRENDIZAJE	223
7.	EMPAQUETAMIENTO DEL OBJETO DE APRENDIZAJE	229
7.1	EJEMPLO PRÁCTICO DEL PROCESO DE EMPAQUETAMIENTO	229
	CONCLUSIONES	237
	RECOMENDACIONES	238
	BIBLIOGRAFÍA	239

LISTA DE FIGURAS

Figura 1.	Ingeniería instruccional	25
Figura 2.	Base para la metodología de MISA. Metodologías de diseño del sistema	26
Figura 3.	Factores que intervienen en el proceso de aprendizaje	29
Figura 4.	Componentes de una solución e-learning	38
Figura 5.	Tipos de estándares del e-learning	39
Figura 6.	Interacción de componentes del proceso institucional	83
Figura 7.	Organigrama del Sistema de APROA	86
Figura 8.	Visión general de las especificaciones de SCORM	90
Figura 9.	Estructura de un árbol de actividad	92
Figura 10.	Componentes funcionales de SCORM	93
Figura 11.	Entidades del software	105
Figura 12.	Recursos de desarrollo de software	106
Figura 13.	Determinantes de la calidad del software y de la efectividad de Organización	108
Figura 14.	Proceso de recopilación de métricas del software	112
Figura 15.	Elementos de un diagrama de Casos de Uso	114
Figura 16.	Elementos de un diagrama de secuencia	115
Figura 17.	Fase 2 - Diseño Instruccional	118
Figura 18.	Etapas de la propuesta metodológica del diseño curricular	120
Figura 19.	Convenciones para el diagrama secuencial de actividades DSA ²	121
Figura 20.	Esquemmatización de los elementos del DSA ²	123
Figura 21.	Relación de desagregación	123
Figura 22.	Relación de secuencialidad	124
Figura 23.	Relación de paralelismo	124
Figura 24.	Relación de dependencia	125
Figura 25.	Relación de preconcepto	125
Figura 26.	Relación causa-consecuencia	125
Figura 27.	Relación de transversalidad	126
Figura 28.	DSA ² asignatura ingeniería del software	127
Figura 29.	Visión tridimensional de la relación. Saber, Hacer y ser	128
Figura 30.	Tabla de saberes	130
Figura 31.	Relación propósitos-contenidos	132
Figura 32.	Actividades de enseñanza-aprendizaje	134
Figura 33.	Agrupación en unidades de aprendizaje	135
Figura 34.	Agrupación en módulos de formación y estructura modular	136
Figura 35.	Planeación curricular primera parte	137
Figura 36.	Elementos planeación curricular	139
Figura 37.	Fase 3 - Diseño y Producción de un Objeto de Aprendizaje	142
Figura 38.	Estructura de un objeto de aprendizaje	143
Figura 39.	Objetos temáticos y específicos derivados de un objeto global	144
Figura 40.	Diagrama de Actores y Sistema – Documento soporte	148
Figura 41.	Diagrama de Casos de Uso – Documento soporte	149
Figura 42.	Diagrama de Casos de Uso para Actores y Sistema – Documento	149

	soporte	
Figura 43.	Diagrama de Casos de Uso General – Documento soporte	152
Figura 44.	Diagrama de Actores y Sistema – Audio	153
Figura 45.	Diagrama de Casos de Uso – Audio	153
Figura 46.	Diagrama de Casos de Uso para Actores y Sistema – Audio	154
Figura 47.	Diagrama de Casos de Uso General – Audio	157
Figura 48.	Diagrama de Actores y Sistema – Animación	158
Figura 49.	Diagrama de Casos de Uso – Animación	158
Figura 50.	Diagrama de Casos de Uso para Actores y Sistema – Animación	159
Figura 51.	Diagrama de Casos de Uso General – Animación	161
Figura 52.	Diagrama de Actores y Sistema – Gráficos y tablas	162
Figura 53.	Diagrama de Casos de Uso – Gráficos y tablas	162
Figura 54.	Diagrama de Casos de Uso para Actores y Sistema – Gráficos y tablas	163
Figura 55.	Diagrama de Casos de Uso General – Gráfico o tabla	166
Figura 56.	Diagrama de Actores y Sistema – Aplicativo	167
Figura 57.	Diagrama de Casos de Uso – Aplicativo	167
Figura 58.	Diagrama de Casos de Uso para Actores y Sistema – Aplicativo	168
Figura 59.	Diagrama de Casos de Uso General – Aplicativo	171
Figura 60.	Diagrama de secuencia – Documento soporte – Estudiante	171
Figura 61.	Diagrama de secuencia – Documento soporte – Profesor	172
Figura 62.	Diagrama de secuencia – Documento soporte- Desarrollador	172
Figura 63.	Diagrama de secuencia – Audio – Estudiante	173
Figura 64.	Diagrama de secuencia – Audio – Profesor	173
Figura 65.	Diagrama de secuencia – Audio – Desarrollador	174
Figura 66.	Diagrama de secuencia – Animación – Estudiante	174
Figura 67.	Diagrama de secuencia –Animación – Profesor	175
Figura 68.	Diagrama de secuencia – Animación – Desarrollador	175
Figura 69.	Diagrama de secuencia – Aplicativo – Estudiante	176
Figura 70.	Diagrama de secuencia – Aplicativo – Profesor	176
Figura 71.	Diagrama de secuencia – Aplicativo – Desarrollador	177
Figura 72.	Diagrama de secuencia – Gráfico y/o tabla – Estudiante	177
Figura 73.	Diagrama de secuencia- Gráfico y/o tabla – Profesor	178
Figura 74.	Diagrama de secuencia – Gráfico y/o tabla – Desarrollador	178
Figura 75.	Tabla de contenidos del Objeto de Aprendizaje	180
Figura 76.	Ventana principal de la plantilla para la visualización del objeto de aprendizaje sobre la plataforma educativa institucional e-escen@riUIS	181
Figura 77.	Núcleo de conocimiento - Clasificación de métricas	182
Figura 78.	Botones activos dentro de la plantilla	182
Figura 79.	Documento de soporte - Clasificación de métricas	183
Figura 80.	Archivos de audio - Clasificación de métricas	184
Figura 81.	Archivos de video - Clasificación de métricas - escena 1	185
Figura 82.	Archivos de video - Clasificación de métricas - escena 2	186
Figura 83.	Archivos de video - Clasificación de métricas - escena 3	186
Figura 84.	Grafico nº 1 - Clasificación de métricas	187
Figura 85.	Grafico nº 2 - Clasificación de métricas	187
Figura 86.	Grafico nº 3 - Clasificación de métricas	188
Figura 87.	Grafico nº 4 - Clasificación de métricas	188

Figura 88.	Grafico nº 5 - Clasificación de métricas	189
Figura 89.	Enunciado del aplicativo - Clasificación de métricas	189
Figura 90.	Ayuda aplicativo - Clasificación de métricas	190
Figura 91.	Aplicativo - Clasificación de métricas - opción 1	190
Figura 92.	Aplicativo - Clasificación de métricas - opción 2	191
Figura 93.	Aplicativo - Clasificación de métricas - opción 3	191
Figura 94.	Objetivos - Clasificación de métricas	192
Figura 95.	DSA ² enfocando las actividades de métricas del software	193
Figura 96.	DSA ² resaltando la temática “Clasificación de métricas”	193
Figura 97.	Ficha de créditos	194
Figura 98.	Esquema de secuenciación de objetos de aprendizaje	194
Figura 99.	Núcleo de conocimiento – Generalidades	195
Figura 100.	Animación escena 1. Definiciones generales	196
Figura 101.	Animación escena 2. Definiciones generales	196
Figura 102.	Animación escena 3. Definiciones generales	197
Figura 103.	Animación escena 4. Definiciones generales	197
Figura 104.	Animación escena 1. Historia de las métricas	198
Figura 105.	Animación escena 2. Historia de las métricas	198
Figura 106.	Información de Soporte – Generalidades	199
Figura 107.	Grafico nº 1 – Generalidades	200
Figura 108.	Grafico nº 2 – Generalidades	200
Figura 109.	Grafico nº 3 – Generalidades	200
Figura 110.	Archivo de audio – Generalidades	201
Figura 111.	Objetivos – Generalidades	202
Figura 112.	Énfasis en la temática Definiciones generales	202
Figura 113.	Ficha Créditos de desarrollo	202
Figura 114.	Núcleo de conocimiento - Entidades a medir	207
Figura 115.	Animación escena 1. Entidades a medir	208
Figura 116.	Animación escena 2. Entidades a medir	208
Figura 117.	Información de Soporte - Entidades a medir	209
Figura 118.	Grafica nº 1 - Entidades a medir	210
Figura 119.	Grafica nº 2 - Entidades a medir	210
Figura 120.	Grafica nº 3 - Entidades a medir	210
Figura 121.	Archivo de audio - Entidades a medir	211
Figura 122.	Objetivos - Entidades a medir	212
Figura 123.	Énfasis en la temática Entidades a medir	212
Figura 124.	Ficha créditos de desarrollo	212
Figura 125.	Núcleo de conocimiento - Métricas en proyectos de software	213
Figura 126.	Animación escena 1. Métricas en proyectos software	214
Figura 127.	Animación escena 2. Métricas en proyectos software	214
Figura 128.	Animación escena 3. Métricas en proyectos de software	214
Figura 129.	Información de Soporte - Métricas en proyectos software	215
Figura 130.	Grafica nº 1 - Métricas en proyectos software	216
Figura 131.	Grafica nº 2 - Métricas en proyectos software	216
Figura 132.	Grafica nº 3 - Métricas en proyectos software	217
Figura 133.	Grafica nº 4 - Métricas en proyectos software	217
Figura 134.	Grafica nº 5 - Métricas en proyectos software	217
Figura 135.	Archivos de audio - Métricas en proyectos software	218
Figura 136.	Objetivos - Métricas en proyectos software	219

Figura 137.	Énfasis en la temática Métricas en proyectos software	219
Figura 138.	Ficha créditos de desarrollo	219
Figura 139.	Aspecto general del portal del profesor	220
Figura 140.	Link currículum	221
Figura 141.	Link investigación	221
Figura 142.	Link extensión	221
Figura 143.	Link administración	221
Figura 144.	Portal profesor link enlaces de interés	222
Figura 145.	Gestor de ejercicios de la plataforma <i>e-escen@riUIS</i>	223
Figura 146.	Modelo generalizado del editor ReLOAD	229
Figura 147.	Empaquetamiento en SCORM con RELOAD2004, paso 1	230
Figura 148.	Empaquetamiento en SCORM con RELOAD2004, paso 2	230
Figura 149.	Empaquetamiento en SCORM con RELOAD2004, paso 2	231
Figura 150.	Empaquetamiento en SCORM con RELOAD2004, paso 3	231
Figura 151.	Empaquetamiento en SCORM con RELOAD2004, paso 4	232
Figura 152.	Empaquetamiento en SCORM con RELOAD2004, paso 5	233
Figura 153.	Empaquetamiento en SCORM con RELOAD2004, paso 6	234
Figura 154.	Empaquetamiento en SCORM con RELOAD2004, paso 7	235
Figura 155.	Empaquetamiento en SCORM con RELOAD2004	235
Figura 156.	Empaquetamiento en SCORM con RELOAD2004, paso 8	236
Figura 157.	Visualización del fichero en .zip	236

LISTA DE TABLAS

Tabla 1.	Modelos de estilos de aprendizaje más conocidos	40
Tabla 2.	Dicotomías de los cuatro niveles de estilos de aprendizaje del Modelo FLSM	41
Tabla 3.	Cuestionario planteado por el modelo de aprendizaje de Felder y Silverman	41
Tabla 4.	Componentes de un curso hipermedia para los objetos de aprendizaje de una unidad docente	44
Tabla 5.	Clasificación de métodos de enseñanza de Pienkevich y González	46
Tabla 6.	Formas del trabajo autónomo	47
Tabla 7.	Diferencias entre el Aprendizaje Individualista y el Cooperativo	55
Tabla 8.	Técnicas de enseñanza-aprendizaje	60
Tabla 9.	Características y recomendaciones del análisis funcional para su aplicación en procesos de formación académica	79
Tabla 10.	Datos orientados al tamaño del software	100
Tabla 11.	Tamaño de diseño y desarrollo de un sistema de cómputo	101
Tabla 12.	Preguntas de Fi	102
Tabla 13.	Nivel de información procesando funciones	103
Tabla 14.	Estructura de atributos	105
Tabla 15.	Aspectos de calidad de software	109
Tabla 16.	Estrategias y técnicas de enseñanza-aprendizaje	138
Tabla 17.	Técnicas e instrumentos de evaluación	140

ANEXOS

ANEXO A.	Test de estilos de aprendizaje de FELDER Y SILVERMAN	247
ANEXO B.	Taxonomía de Bloom (verbos para enunciar saberes)	254
ANEXO C.	Documentos relacionados con la realización del diseño instruccional	262

RESUMEN

TITULO

DISEÑO INSTRUCCIONAL BASADO EN COMPETENCIAS MEDIADO POR TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN (TIC's) PARA LA ASIGNATURA INGENIERÍA DEL SOFTWARE DEL PROGRAMA ACADÉMICO DE INGENIERÍA DE SISTEMAS E INFORMÁTICA.*

AUTORES

Nadia Alexandra Carreño Avella
Paola Carolina Espinosa Rodríguez**

PALABRAS CLAVE

Diseño Instruccional, Estilos de Aprendizaje, Competencias Educativa, Tecnologías de Información y Comunicación, e-Learning, Objeto de Aprendizaje, Ingeniería del Software, Métricas del Software.

DESCRIPCIÓN

Los cambios que plantean los avances tecnológicos generan un reto cada vez más grande para el sistema educativo, a la vez que implican un mayor distanciamiento entre la educación que se imparte y la realidad laboral de los egresados. La experiencia del país muestra que los jóvenes necesitan mejores herramientas conceptuales y metodológicas que les permitan desempeñarse con éxito en su quehacer laboral y una propuesta educativa que los prepare para enfrentar con seguridad el desafío y la responsabilidad de ser productivos para sí mismos y para quienes los rodean. Incluir entonces la formación de competencias en los estudiantes constituye uno de los elementos básicos para mejorar la calidad de la educación.

Con las TIC's (tecnologías de información y comunicación) aparecen ambientes de aprendizaje que vienen a complementar las aulas tradicionales y generar nuevas oportunidades para acceder al conocimiento. La incorporación de estas nuevas tecnologías de información y comunicación en las Instituciones de Educación Superior implica la transformación permanente de la estructura organizacional y académica de las mismas y es por ello que la universidad colombiana debe hoy superar los grandes desafíos que le plantean el uso de las TIC's como un recurso, pero también como un reto que impone la cualificación de los estudiantes.

Este proyecto contempla la realización del Diseño Instruccional para asignatura *Ingeniería del Software* siguiendo la metodología de un modelo de formación basado en competencias mediado por Tecnologías de Información y Comunicación (TIC's), que permita el aprendizaje significativo y personalizado (considerando Estilos de Aprendizaje) y la construcción de un objeto de aprendizaje abierto e interoperable, siguiendo los estándares de *e-learning*, para la temática de Métricas del Software de la asignatura en cuestión. Este proyecto pretende propiciar ambientes, mediante la implementación de materiales en línea soportados por una estructura pedagógica, que apoyen un proceso de enseñanza-aprendizaje integral.

* Proyecto de Grado

** Facultad de Ingenierías Físico Mecánicas, Escuela de Ingeniería de Sistemas e Informática
Director del Proyecto Leonardo Augusto Camargo Corzo
Codirectores del Proyecto Fernando Rojas Morales
Sergio Eduardo Gómez

SUMMARY

TITLE

INSTRUCTIONAL DESIGN BASED ON COMPETITIONS MEDIATED BY INFORMATION AND COMMUNICATION TECHNOLOGIES (TIC's), FOR THE SOFTWARE ENGINEERING SUBJECT OF THE SYSTEMS AND INFORMATICS ENGINEERING ACADEMIC PROGRAM.*

AUTHORS

Nadia Alexandra Carreño Avella
Paola Carolina Espinosa Rodríguez**

KEY WORDS

Instructional Design, Learning Styles, Competitions Educative, Styles of Learning, Information and Communication Technologies, e-Learning, Learning Object, Software Engineering, Software Metrics.

ABSTRACT

The changes that raise the technological advances more and more generate a great challenge for the educative system, simultaneously that implies a greater gap between the education that is distributed and the reality work of the withdrawn ones. The experience of the country shows that young people need better conceptual tools and methodology that they can make possible for them to evolve successfully in their work task and an educative proposal that prepares them to face the challenge and the responsibility of being productive for themselves and for those who surround them. To include the formation of competitions in the students constitutes one of the basic elements to improve the quality of the education.

With the TIC's (technologies of information and communication) they appear atmospheres of learning that come to complement the traditional classrooms and generate new opportunities to accede the knowledge. The incorporation of these new technologies of information and communication in the Institutions of Superior Education implies the permanent transformation of organizational and academic structure of the same ones and for that reason the Colombian university must surpass the great challenges that raise the use of the TIC's like a resource, but also like a challenge that imposes the qualification of the students.

This project contemplates to the accomplishment of Design Instructional for subject Software Engineering following the methodology of a model of formation based on competitions half-full by Information and Communication Technologies (TIC's), which it allows to the significant and customized learning (considering Styles of Learning) and the construction of an object to open and interoperable learning, following the standards of e-learning, for the thematic Software Metrics of the subject at issue. This project tries to cause atmospheres, by means of the implementation of materials in line supported by a pedagogical structure that support an integral education-learning.

* Grade Project

** Faculty of Physical Mechanical Engineering, Systems and Computer Science Engineering School
Project Director Leonardo Augusto Camargo Corzo
Project co-director Fernando Rojas Morales
Sergio Eduardo Gómez

INTRODUCCIÓN

El fin último del proceso pedagógico es el estudiante, de quien se debe alcanzar su madurez como persona, su desarrollo cognitivo, afectivo, psicomotor y actitudinal; es decir: su desarrollo integral

Los cambios que plantean los avances técnicos y tecnológicos generan un reto cada vez más grande para el sistema educativo, a la vez que implican un mayor distanciamiento entre la educación que se imparte y la realidad laboral de los egresados.

También debe tenerse en cuenta que frente a los efectos socioeconómicos de la globalización de los mercados se requiere de procesos de enseñanza aprendizaje que le aporten a la fuerza laboral profesionales versátiles y polivalentes que sepan identificar las oportunidades de negocios, con capacidad de asociación y que puedan generar nuevas tendencias productivas dentro de los esquemas que la economía mundial plantea.

Adoptar un plan de articulación de la educación con el Mundo Productivo basado en el desarrollo de las Competencias, no es otra cosa que proyectar una solución a los vacíos y limitaciones que los empresarios observan en los jóvenes egresados que aspiran conseguir empleo. En virtud de ello, cuando se habla de mejorar la Calidad de la Educación, no cabe duda que se trata de cualificar la eficiencia de los egresados en cuanto a su desenvolvimiento en la actividad productiva.

Desarrollar competencias para la vida implica la formación de competencias laborales, es decir, competencias asociadas a la productividad y la competitividad. La experiencia del país muestra que los jóvenes necesitan mejores herramientas conceptuales y metodológicas que les posibiliten desempeñarse con éxito en su quehacer laboral y una propuesta educativa que los prepare para enfrentar con seguridad el desafío y la responsabilidad de ser productivos para sí mismos y para quienes los rodean. La responsabilidad de crear condiciones para un país mejor, más equitativo y más justo, más lleno de oportunidades para todos. Esta necesidad se acrecienta en la actualidad cuando los cambios sociales, económicos, culturales y tecnológicos plantean cada día nuevas exigencias al mundo productivo.

Incluir entonces la formación de competencias en los estudiantes constituye uno de los elementos básicos para mejorar la calidad de la educación; por tanto se requiere de un enfoque que dé paso a una educación más integradora, que articule la teoría y la práctica, y garantice aprendizajes aplicables a la vida productiva cotidiana.

El estudiante competente es aquel que posee un conocimiento y sabe utilizarlo. Tener una competencia es usar el conocimiento para aplicarlo a la solución de situaciones nuevas o imprevistas, fuera del aula, en contextos diferentes, y para desempeñarse de manera eficiente en la vida personal, intelectual, social y laboral.

Las competencias que se deberán transmitir son aquellas que les permitan a los estudiantes desarrollar la capacidad de avanzar en la construcción del conocimiento, las

que propicien su cualificación y superación permanentes. Competencia para crear o recrear el conocimiento a partir de lo adquirido y competencia para argumentar y justificar lo creado y lograr su validación social.

Con las TIC`s (tecnologías de información y comunicación) aparecen nuevos ambientes de aprendizaje que no pretenden sustituir a las aulas tradicionales, pero que vienen a complementarlas y a generar nuevas oportunidades para acceder al conocimiento.

La incorporación de estas nuevas tecnologías de información y comunicación en las Instituciones de Educación Superior implica la transformación permanente de la estructura organizacional y académica de las mismas y es por ello que debemos ser conscientes y consecuentes con que la universidad colombiana debe hoy superar los grandes desafíos que le plantean el uso de las TIC`s como un recurso, pero también como un reto que impone la cualificación de los estudiantes.

En la Universidad Industrial de Santander, mediante el proyecto ProSPETICuis, se definió una **política** y se diseñaron **estrategias** tendientes a asegurar condiciones que permitan, mediante sistemas de aprendizaje en línea, ofrecer experiencias orientadas a la apropiación del conocimiento con elevados estándares de calidad. Todo esto a través del desarrollo de diseños instruccionales y objetos de aprendizaje que se sustentan en un modelo de formación basado por competencias. En ese modelo se soportará el proceso de enseñanza-aprendizaje en las asignaturas correspondientes a los diferentes planes de estudio de la UIS.

Este proyecto de grado se fundamenta en la creación de un diseño instruccional basado en competencias, mediado por tecnologías de información y comunicación (TIC`s) para la asignatura Ingeniería del Software del programa académico de Ingeniería de Sistemas e Informática.

El documento contiene siete capítulos donde se referencian los conceptos que fundamentan la propuesta del proyecto y las actividades desarrolladas para el cumplimiento de los objetivos planteados en el mismo.

1. DESCRIPCIÓN DEL PROYECTO

1.1. PLANTEAMIENTO DEL PROBLEMA

La problemática fundamental del actual modelo educativo es que el proceso de aprendizaje es lento, escaso y estático. Lento, debido a que el alumno se limita a recibir el conocimiento en la medida en que el maestro lo trasmite, circunstancia que conlleva a que no sea el estudiante quien protagonice su propio camino de aprendizaje; escaso, porque no se dispone de las herramientas necesarias para proporcionar un ambiente de aprendizaje propicio al estudiante y las pocas que dan soporte a la enseñanza actualmente, no están al alcance de todos; y estático en la medida en que el alumno adquiere la estructura de conocimiento que le proporciona el docente pero pocas veces alguno de los dos la amplía, siendo la misma a través de los años. Igualmente, la escuela, como institución física es el único espacio de aprendizaje.

Para enfrentar los desafíos que plantea la sociedad del conocimiento, es necesario adoptar un nuevo enfoque de enseñanza, en el cual el docente sea un guía que contribuya a satisfacer las expectativas de sus alumnos, detectando sus debilidades y desarrollando sus fortalezas. Por su parte, la institución educativa debe facilitar a los estudiantes el acceso a los recursos que ofrecen las Tecnologías de Información y Comunicación (TIC's).

La llegada de las TIC's a las instituciones implica nuevas concepciones del proceso de enseñanza-aprendizaje. El énfasis se traslada desde la enseñanza hacia el aprendizaje estableciéndose nuevos roles y responsabilidades para los alumnos y profesores. El alumno se transforma en un participante activo y constructor de su propio aprendizaje y el profesor asume el rol de guía y facilitador de este proceso, lo cual varía su forma de interactuar con sus alumnos y, a su vez, la forma de planificar y de diseñar el ambiente de aprendizaje. Se trata de un modelo basado en el aprendizaje a través de la investigación y la construcción del conocimiento en forma cooperativa, para que los alumnos puedan profundizar en los contenidos que deben adquirir.

El cambio fundamental en el modelo tradicional comprende los siguientes pasos:

1. La transición del rol docente de trasmisor de conocimientos a facilitador de conocimientos.
2. La reorganización del conocimiento preorganizado en conocimiento construido.
3. Incremento por parte de los alumnos del acceso a las TIC's.

En síntesis, el nuevo modelo de enseñanza aprendizaje implica el mejoramiento de la capacidad de análisis crítico de los alumnos, del pensamiento independiente, del trabajo en grupo y la creatividad para la resolución de problemas, preparando así a los educandos para un mundo en el cual la creatividad y la capacidad de adaptarse a nuevas situaciones se convertirán en activos indispensables.

La meta de la educación de cualquier sociedad democrática y moderna debe ser la orientación hacia individuos autónomos, capaces de adquirir información por su cuenta, capaces de juzgar la validez de dicha información y hacer, a partir de ella, inferencias lógicas y coherentes. La educación, entonces, está dirigida a hacer independientes a los estudiantes. Parte de lo que hace a los estudiantes independientes es la información, sin embargo, si existiera algún conflicto entre la adquisición de información y la habilidad intelectual de cómo adquirirla, esto último es, sin duda, lo más importante y lo que hay que privilegiar desde la docencia.¹

La Universidad Industrial de Santander en su deseo de integrar las Tecnologías de la Información y Comunicación en los procesos de aprendizaje centrados en el estudiante, mediante el desarrollo de modelos pedagógicos apropiados, con el fin de fortalecer la cultura del aprendizaje permanente, diversificar estrategias pedagógicas, implementar sistemas evaluativos eficaces e interactuar en la sociedad global de conocimiento, ha desarrollado el PROYECTO “SOPORTE AL PROCESO EDUCATIVO UIS MEDIANTE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN – ProSPETIC”, con el cual da inicio a la transformación de la educación en nuestra universidad.

Es por esto que nace el interés de participar en el desarrollo de estos nuevos escenarios de aprendizaje con una propuesta instruccional para la asignatura de Ingeniería del Software basada en un modelo de formación por competencias y apoyada en las Tecnologías de Información y Comunicación, e impulsando desde la perspectiva de la Ingeniería, la generación de herramientas adaptativas e interoperables que propicien el mejoramiento de la calidad en el proceso de enseñanza-aprendizaje y el aprovechamiento de los adelantos tecnológicos para agregar valor a los procesos académicos.

1.2. JUSTIFICACIÓN

En el transcurso de los últimos años las herramientas educativas han evolucionado, y este cambio permite y facilita los recursos para obtener un aprendizaje personalizado y significativo en campos de estudio con diversa complejidad.

¹ BECCO, GUILLERMO R. Vigotsky y Teorías sobre el Aprendizaje. [En línea]. Disponible en: [<http://www.ciudad.com.ar/>]

La Universidad Industrial de Santander por ende, a partir de las Tecnologías de Información y Comunicación y el proyecto institucional ProSPETIC, dará la base a los procesos de enseñanza/aprendizaje en línea, de las asignaturas de sus diferentes programas académicos.

Esta propuesta pretende la creación del plan instruccional basado en competencias de la asignatura Ingeniería del Software del programa académico de la Escuela de Ingeniería de Sistemas e Informática, y la implementación de uno de los objetos de aprendizaje allí planteados.

Además, este proyecto se desarrolla paralelamente con los planes y condiciones de calidad para el desarrollo de la Educación Superior Colombiana que hacen referencia a competencias y conocimientos que se consideran esenciales en la formación de cada uno de los profesionales o egresados. En segunda medida concuerda en absoluto con el proyecto educativo de la Universidad Industrial de Santander, que en su modelo Institucional – Acuerdo No. 015 del 2000 – pretende modificar sus políticas, con el fin de lograr “la reforma de sus programas académicos de tal forma que los planes de las asignaturas constituyan un currículo de formación integral, y el desarrollo de nuevas metodologías pedagógicas, que vayan en pro de sus principios orientadores como lo son la formación integral y la vigencia social de los saberes, actitudes y prácticas construidas en el estudiantado”.

1.3. OBJETIVOS

1.3.1. Objetivo General:

Realizar el diseño instruccional de la asignatura Ingeniería del Software siguiendo la metodología de un modelo de formación basado en competencias mediado por Tecnologías de Información y Comunicación, que permita el aprendizaje significativo y personalizado (considerando estilos de aprendizaje); y construir un objeto de aprendizaje abierto e interoperable siguiendo los estándares de e-learning, relacionado con una temática de la asignatura.

1.3.2. Objetivos Específicos:

- Crear el diseño instruccional de la asignatura INGENIERÍA DEL SOFTWARE, aplicando principios basados en la propuesta metodológica para el diseño curricular bajo la visión de competencias y mediado por Tecnologías de Información y Comunicación (TIC's).
- Diseñar y desarrollar un objeto de aprendizaje, basado en lineamientos del estándar SCORM² tomando como base las estrategias pedagógicas

² SCORM (*Sharable Content Object Reference Model*): Estándares técnicos interrelacionados para desarrollar enseñanza de contenidos vía WEB.

planteadas en el diseño instruccional, fundamentado en la temática de *Métricas* de la asignatura INGENIERÍA DEL SOFTWARE.

- Implementar dicho objeto de aprendizaje, utilizando el portal del profesor como medio de enlace a la Biblioteca Digital de recursos didácticos de la UIS, para su inmediata exploración como material de soporte en la enseñanza-aprendizaje de la asignatura INGENIERÍA DEL SOFTWARE.
- Desarrollar el Portal Web del profesor encargado de la asignatura Ingeniería del Software, utilizando las plantillas estándar creadas para este propósito, con el fin de facilitar el acceso a contenidos de la asignatura INGENIERÍA DEL SOFTWARE por parte de los estudiantes.

1.4. VIABILIDAD DEL PROYECTO

Este proyecto se soporta en el compromiso que la Universidad Industrial de Santander ha adquirido con la comunidad académica a través del PROYECTO “SOPORTE AL PROCESO EDUCATIVO UIS MEDIANTE TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN – ProSPETIC” ofreciendo así, la disponibilidad institucional de una infraestructura física, informática y telemática tecnológicamente vigente y de capacidad adecuada para soportar los procesos formativos y, la plataforma en desarrollo *e-escen@riUIS* de la división de servicios de información de la UIS, basada en el concepto de plataforma hipermedia, adaptativa, multiagente, multilinguaje y de tecnología abierta e interoperable. Se hará uso de licencias GNU (Licencias Públicas Generales) en la aplicación del estándar SCORM (Modelo de Referencia de Objetos de Contenidos Intercambiables), que permitirá que el objeto creado sea abierto e interoperable. Además, aprovecha la logística y recursos propios de la universidad y la experiencia de sus docentes como medio facilitador de nuevas propuestas tendientes a ampliar la gama de recursos educativos de la Universidad Industrial de Santander.

1.5. IMPACTO

- El docente contará con una ayuda práctica para sus alumnos, que facilitará su labor educativa y la difusión de contenidos de la asignatura Ingeniería del Software a través de su respectivo portal.
- Los alumnos, mediante el Objeto de Aprendizaje desarrollado en la temática de *Métricas*, podrán reforzar (interactuar) y evaluar conocimientos adquiridos.
- Creación de la cultura del trabajo en red.
- Consulta de las bibliotecas y revistas digitales adquiridas por la Universidad.
- Desarrollo de contenidos digitales propios ofreciendo el soporte pedagógico, de diseño gráfico e informático necesario para lograr en el estudiante un grado de interactividad significativo, adaptación al contexto local en aspectos técnicos, lingüísticos, sociales y económicos, y suministro de información de forma clara, precisa y concisa.
- Flexibilidad en el proceso formativo.

- Mejoramiento de la práctica de la docencia (nuevas estrategias, nuevas herramientas), facilitando la divulgación de contenidos y material de trabajo.
- Creación de disciplinas de autoaprendizaje por parte del estudiante mediante el acceso a los contenidos en cualquier momento y lugar.
- Desarrollo de la concepción educativa desde un ambiente en línea utilizando la Internet como soporte al proceso de enseñanza-aprendizaje.

2. MARCO TEÓRICO

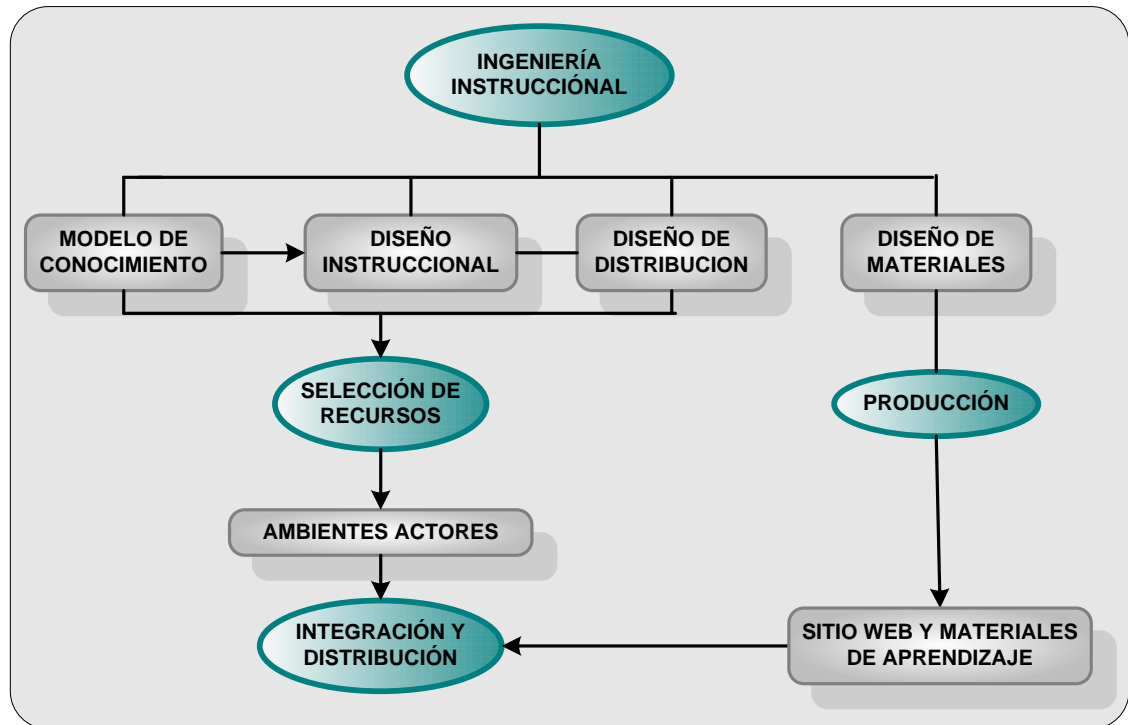


Figura 1. Ingeniería instruccional³

La Ingeniería Instruccional se define como un método que apoya el análisis, el diseño y la entrega de la planificación de un sistema de aprendizaje, integrando los conceptos, procesos y principios del diseño instruccional, la ingeniería del software y la ingeniería de conocimiento”.⁴

Este método en su planificación esta conformado por el modelo de conocimiento, diseño instruccional, diseño de distribución y diseño de materiales.

La ingeniería instruccional aprovecha los aportes de la ingeniería del conocimiento para modelar el proceso de conocimiento; el diseño instruccional para articular la información pertinente de una asignatura con base en objetivos de aprendizaje y la ingeniería de sistemas de información para ofrecer el soporte tecnológico en la implementación de estrategias para el aprendizaje adaptativo y personalizado con base en estándares de e-learning.

³ Adaptación hecha por el laboratorio I+D CENTIC UIS de las apreciaciones Gilbert Paquette.

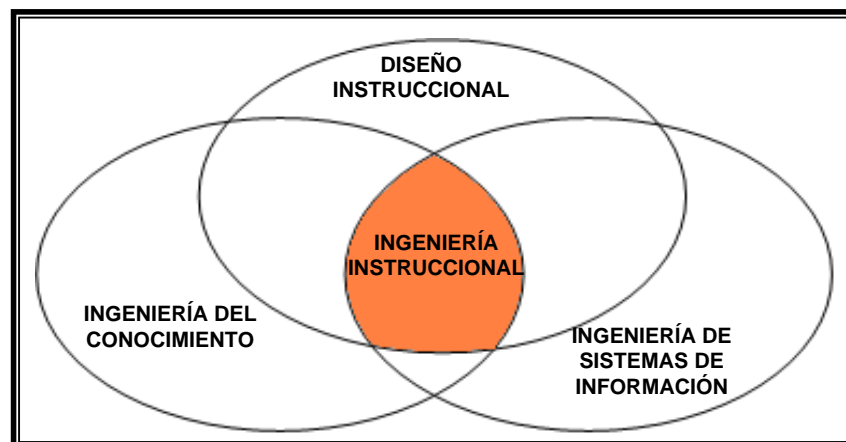
⁴ PAQUETTE, GILBERT. Educational Modeling Languages, From an Instructional Engineering Perspective. [En línea]. Disponible en: [www.liceftel.uq.quebec.ca/gp/docs/Article%20EML-MISAedited.doc]

Desde la perspectiva del modelo del Conocimiento, cobra relevancia, en el proceso de desarrollo de un sistema de aprendizaje distribuido⁵, la identificación y estructuración del conocimiento, representado mediante un lenguaje gráfico. Este lenguaje permite estructurar el conocimiento en las fases del diseño instruccional, que junto con el modelo de distribución permiten hacer una selección de recursos necesarios dependiendo de los ambientes y actores presentes dentro del sistema.

Por otra parte en el diseño de materiales están presentes los medios que integran distintos tipos de programas, elementos multimediales digitalizados y servicios de comunicación, que fortalezcan la capacidad del estudiante de forma integral y que permitan encontrar la solución a problemas, creando así materiales de aprendizaje preparados para la distribución a los actores ya mencionados.

MISA (*método de ingeniería de sistemas de aprendizaje*) es una metodología de diseño de unidades didácticas en la que se propone un meta-modelo propio. Por su parte el Modelador por Objetos Tipo MOT proporciona una representación gráfica de los elementos de dicho meta-modelo. El objetivo final de esta metodología es diseñar sistemas educativos basados en las TIC's. Este método es el resultado principal de una nueva concepción pedagógica fundamentada en las ciencias cognitivas que se denominaron como "Ingeniería Didáctica" (*Instructional Engineering*) desarrollada en la Télé-Université de Canadá.

La metodología MISA fue desarrollada hace más de 10 años por el LICEF, el lenguaje gráfico MOT y su software de modelización y su versión más avanzada MOT+.



**Figura 2. Base para la metodología de MISA.
Metodologías de diseño del sistema⁶**

⁵ Aprendizaje distribuido significa educación virtual, aprendizaje en red, todos los términos que significan estudiar de una manera no tradicional, no necesariamente se está en un salón de clase con un profesor al frente.

⁶ Adaptación hecha por las autoras del proyecto a partir de: PAQUETTE, GILBERT. TeleLearning Models Impact on Instructional Technology in the 21st Century. [En línea]. Disponible en: [www.licef.teluq.uqam.ca/gp/ppt/aera.pps]

2.1. DISEÑO INSTRUCCIONAL

Se define Diseño Instruccional (Instructional Design) como el proceso que genera especificaciones instruccionales por medio del uso de teorías instruccionales y teorías de aprendizaje para asegurar que se alcancen los objetivos planteados.

En el diseño instruccional se hace un completo análisis de las necesidades y metas educativas a cumplir y posteriormente se diseña e implementa un mecanismo que permita alcanzar esos objetivos. Así, este proceso involucra el desarrollo de materiales y actividades instruccionales, y luego las pruebas y evaluaciones de las actividades del alumno.

El Diseño Instruccional es un proceso que viene siendo formalmente aplicado desde los años 60's (la expresión fue creada en 1962 por Robert Glaser) y desde esas fechas se han planteado varios modelos para aplicar el Diseño Instruccional. Algunos de los modelos más conocidos son ADDIE (Analysis, Design, Development, Implementation, Evaluation) el cual es considerado como uno de los más genéricos.⁷

El aprendizaje y las teorías que tratan los procesos de adquisición de conocimiento han tenido durante este último siglo un enorme desarrollo debido fundamentalmente a los avances de la psicología y de las teorías instruccionales, que han tratado de sistematizar los mecanismos asociados a los procesos mentales que hacen posible el aprendizaje.

El propósito de las teorías del aprendizaje es el de comprender e identificar estos procesos y, a partir de ellos, tratar de describir métodos para que la instrucción sea más efectiva. Es en este último aspecto en el que principalmente se basa el diseño instruccional, que se fundamenta en identificar cuáles son los *métodos* que deben ser utilizados en el diseño del proceso de instrucción, y también en determinar en qué *situaciones* estos métodos deben ser usados.⁸

Las teorías del aprendizaje son indispensables en el desarrollo del diseño instruccional, ya que éstas proporcionan los fundamentos psicológicos y filosóficos que enriquecen los modelos de diseño. El aporte determinante para el diseño instruccional según el Conductismo es que su aplicación facilita el dominio del contenido, iguala el aprendizaje con los cambios en la conducta observable y los alumnos serán capaces de reconocer y aplicar reglas, hechos y operaciones.

⁷ Definición tomada de [<http://go2learn.cl/id.html>]

⁸ FERNÁNDEZ A., RAÚL R. Aprendizaje con nuevas tecnologías paradigma emergente. ¿Nuevas modalidades de aprendizaje? *EduTec. Revista Electrónica de Tecnología Educativa*. Núm. 20/ Enero 06. [En línea]. Disponible en: [<http://edutec.rediris.es/Revelec2/revelec20/raul20.htm>]

Según los Cognitivistas, esta teoría es útil para la enseñanza de estrategias en la solución de problemas en donde se aplican hechos, reglas, a situaciones poco familiares. Su énfasis está en la conceptualización de los procesos del aprendizaje del estudiante y de cómo la información es recibida, organizada, almacenada y localizada. Son determinantes en la participación activa del estudiante en el proceso de aprendizaje y desarrollan ambientes estimulantes que permiten a los estudiantes hacer conexiones con el material previamente adquirido.

Por otra parte el Constructivismo trata problemas poco definidos a través de la reflexión en acción, equiparando el aprendizaje con la creación de significados a partir de experiencias. Esta teoría está situada en la creación de herramientas que reflejan la sabiduría de la cultura en la cual se utilizan, así como los deseos y experiencias de los individuos.

Cada teoría aporta aspectos importantes en la construcción de objetivos, contenidos, estrategias y evaluación de los aprendizajes.

2.1.1. FORMACIÓN SUPERIOR BASADA EN COMPETENCIAS

La formación por competencias tiene su origen en las aportaciones del mundo del trabajo y ha llegado a la educación debido a que en el sistema actual de formación se imparten muchos conocimientos que por no haber sido adquiridos de manera científica – construyendo el saber con base en la experiencia- se convierten en conocimientos inútiles, ante la incapacidad del poseedor de aplicarlos a su actividad profesional practica.

En la formación de profesionales es necesario realizar cambios metodológicos, didácticos y actitudinales en donde el docente deje de transmitir conocimientos definidos y enmarcados dentro de ese tradicionalismo de la verdad que se transmite, para que trace estrategias que permitan a los estudiantes llegar al conocimiento como resultado de la experiencia práctica aplicada, fruto de la investigación y el razonamiento, pero en todo caso distante del ejercicio memorístico de la educación tradicional.

Sólo así logrará el sistema aportar al país profesionales realmente capaces, críticos y creativos que estén en posibilidad de generarle desarrollo del país con elementos reales de competitividad y eficiencia.

❖ Concepto de competencia

La competencia es un saber hacer con conciencia. El concepto de competencia otorga un significado de unidad e implica que los elementos del conocimiento tienen sentido sólo en función del conjunto. En efecto, aunque se pueden fragmentar sus componentes, estos por separado no constituyen la competencia:

ser competente implica el dominio de la totalidad de elementos y no sólo de alguna(s) de las partes.

En otras palabras, se entiende por Competencia, una construcción social compuesta de aprendizajes significativos en donde se combinan atributos tales como conocimientos, actitudes, valores y habilidades, con las tareas que se tienen que desempeñar en determinadas situaciones.

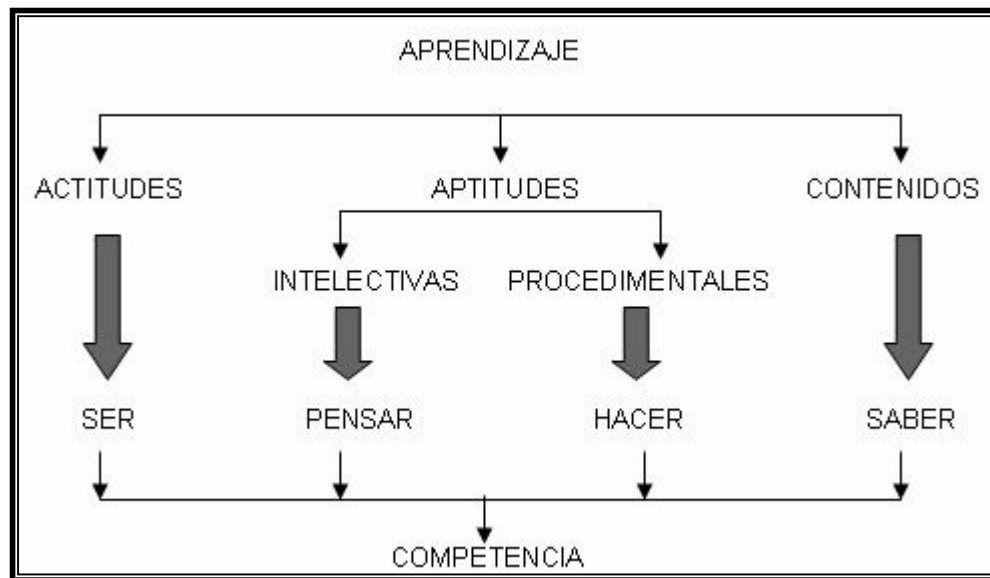


Figura 3. Factores que intervienen en el proceso de aprendizaje⁹

Desde la perspectiva de la educación por competencias, lo importante no es la posesión de determinados conocimientos, sino el uso que se haga de ellos. Este criterio obliga a las instituciones educativas a replantear lo que comúnmente han considerado como formación. Bajo esta óptica, para determinar si un individuo es competente o no lo es, deben tenerse en cuenta las condiciones reales en las que el desempeño tiene sentido, en lugar del cumplimiento formal de una serie de objetivos de aprendizaje que en ocasiones no tienen relación con el contexto.

Con una formación desde el marco de las competencias profesionales integradas se espera promover una preparación más realista, que retome las necesidades de la sociedad (experiencia social, práctica profesional y desarrollo disciplinar). Sin embargo, tales necesidades, así como los contextos que enfrentará el futuro profesional, se encuentran en permanente cambio, situación que requiere que los estudiantes se preparen no sólo para combinar momentos de trabajo con momentos educativos, sino también para ser capaces de transitar por ellos.

⁹ SALAS Z. WALTER A. "Formación Por Competencias en Educación Superior. Una aproximación conceptual a propósito del caso colombiano". *Revista Iberoamericana de Educación* (ISSN: 1681-5653)

Paralelamente, el aprendizaje basado en competencias se apoya en las Tecnologías de información y comunicación TIC's para reforzar la eficacia del aprendizaje en individuos y grupos. La relación TIC's-competencias integrará los aspectos tecnológicos, cognitivos, pedagógicos y organizativos, conduciendo hacia sistemas interactivos y plataformas abiertas que hagan posible el aprendizaje distribuido, el aprendizaje a partir de la experiencia, el aprendizaje en colaboración y el aprendizaje personalizado.

2.1.2. TEORÍAS DEL APRENDIZAJE

La manera como se ha tratado de comprender, predecir, y controlar el comportamiento humano y de explicar cómo los estudiantes acceden al conocimiento, a la adquisición de destrezas y habilidades, al razonamiento y a la apropiación y comprensión de conceptos, se ha basado en las premisas de las diferentes teorías del aprendizaje.

Las tres grandes teorías de aprendizaje utilizadas más a menudo en la creación de entornos educativos son conductismo, cognitivismo y constructivismo. Las tres pretenden entender y definir cómo aprenden las personas.

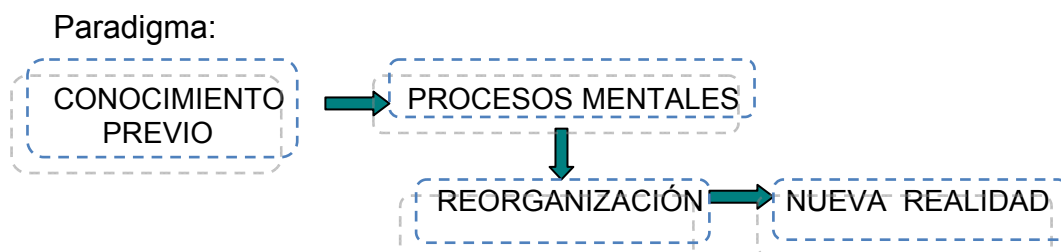
❖ Conductismo

Pretende definir las condiciones situacionales a las cuales se enfrenta una persona, y registrar rigurosamente las respuestas correspondientes. Renuncia a tomar en cuenta los procesos intermedios que no sean inmediatamente accesibles a la observación.



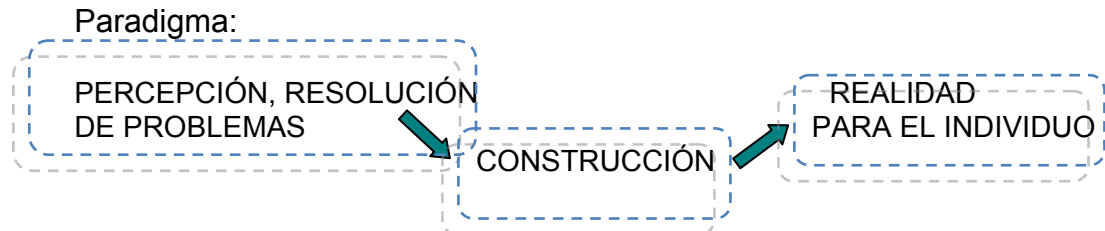
❖ Cognitivismo

El ser humano es un procesador de información capaz de realizar procesos cognitivos complejos, es decir, es el ente activo, interactivo, adaptativo y con dominio del ambiente, capaz de considerar e incorporar información a su estructura cognitiva a través del establecimiento de relaciones significativas.



❖ **Constructivismo**

Todo conocimiento nuevo que el sujeto adquiere es producto de un proceso constructivo si se analiza en el interior del individuo. Es decir, se apoya en conocimientos anteriores y supone una actividad por parte de quien lo adquiere.



Estas teorías, sin embargo, fueron desarrolladas en una época en la que las tecnologías no habían impactado el aprendizaje.

En la sociedad de la información, el conocimiento sobre el aprendizaje adquiere renovada importancia. Ha surgido por lo tanto la necesidad de un cambio a nuevas teorías del aprendizaje, ya que estas deben responder a las necesidades y exigencias actuales de la sociedad y, a que el proceso educativo se encuentre ligado al escenario en el cual se lleva a cabo.

Hoy el reto frente a la promoción del aprendizaje es más impreciso pero a la vez importante. Es por esto que los procesos de modernización de los sistemas de formación ven en el movimiento de las competencias un referente muy válido para optimizar los insumos del diseño curricular y organizar el proceso enseñanza-aprendizaje en torno a la construcción de capacidades para llegar a ser competente.

❖ **HISTORIA DE LAS TEORÍAS DE APRENDIZAJE EN EL DISEÑO INSTRUCCIONAL¹⁰**

➤ **El Conductismo en el Diseño Instruccional**

El conductismo tenía impacto en la educación hasta mediados de los 60 donde empezó a perder popularidad entre los psicólogos, entonces¹¹ identifica seis áreas de impacto en la educación moderna: el movimiento de objetivos conductistas; la fase de la máquina de enseñanza; el movimiento de la instrucción programada; la

¹⁰ AGUIRRE., RODOLFO P. Programa de Educación Continua, Abierta y a Distancia de la Universidad de Sonora. [En línea] Disponible en: [www.educadis.uson.mx]. Mtra. ANAYA M., MARCELA I. Centro de Tecnología Educativa, Secretaría de Educación y Cultura del Estado de Sonora. [En línea]. Disponible en [www.cete-sonora.gob.mx/]

¹¹ SAETTLER, PAÚL. Historia de la Tecnología Educativa Americana, 1990. En: [MERGEL. 1998]

aproximación de la instrucción individualizada; el aprendizaje asistido por computadora y la aproximación de sistemas para la instrucción.

- **El Movimiento de Objetivos Conductistas**

El objetivo del estudio en términos específicos es cuantificable. Por ejemplo al haber visto el tema completo de una unidad el estudiante está en condiciones de responder acertadamente un 90% de las preguntas relacionadas con dicho tema.

Para el desarrollo de dicho objetivo una temática debe ser segmentada en fracciones o conceptos específicos y cuantificables.

- **Las Máquinas de Enseñanza**

B.F. Skinner fue quien se encargó de asociar las máquinas a la educación, a este se le suman una serie de contribuciones como:

- Pressey: Introduce la máquina de opción múltiple en una reunión de la Asociación Americana de Psicología en 1925.
- Peterson: Un estudiante de postgrado de Pressey quien desarrolló las “hojas de quimos” en las cuales el aprendiz verifica sus respuestas con una “plantilla perforada”.
- W.W.II: Dispositivo llamado “verificación de fase”, construido en la década de los 40 y 50, pensado y probado en habilidades para desarmar y armar equipos.
- Crowder: Diseñó un estilo de rama de programación para la fuerza aérea de los EE.UU. en 1950 para la capacitación en la detección de fallas en equipo electrónico.
- Skinner: Basado en las condiciones operatorias, la máquina de enseñanza de Skinner usada para que los aprendices completaran o contestaran cuestionarios y después recibieran la retroalimentación con las respuestas correctas. Skinner demostró su máquina en 1954.

En un comienzo la instrucción programada se enfocaba en hacer aparatos para explicar los contenidos de temáticas específicas pero después se basó en el estudio del aprendizaje y la formulación de bases para la teoría del aprendizaje.

- **La Instrucción Individualizada**

Sus principios se remontan a 1900 y resurge en la década de los 60 con el Plan Keller en el que la educación se enfoca individualmente dependiendo de las necesidades de cada estudiante.

De igual forma se desarrollaron otros como la Instrucción Prescrita Individualmente (IPI) (1964) y el Programa de Aprendizaje de Acuerdo con las Necesidades (PLAN) (1967).

Todas estas metodologías o planes se basaban en la educación fundamentada en las conductas.

- **La Aproximación de Sistemas**

Su desarrollo surge en las décadas de los 50 y 60 basada en laboratorios de lenguaje, consiste en la instrucción programada para el uso de la computadora y multimedia. Dichos sistemas parecen diagramas de flujo en los que el diseñador se mueve en el desarrollo de la instrucción.

Se aplicó al mundo de los negocios y las fuerzas militares; incluía metas y un constante análisis y evaluación de los resultados obtenidos durante su desarrollo.

- **El Cognoscitismo en el Diseño Instruccional**

Aunque surgió a principios de los 50, adquiere importancia en la teoría del aprendizaje a finales de los 70 cuando empieza a tener influencia sobre el diseño instruccional. Difiere de la técnica conductista porque ya no se fija en las conductas externas sino en los procesos mentales y en cómo estos se pueden aprovechar para promover aprendizajes efectivos¹².

Los diseños de modelos que se habían desarrollado para el conductismo tradicional no se desecharon, sino que se enriquecieron con el “*análisis de actividades*” y el “*análisis del aprendiz*”. Los nuevos modelos incluyen componentes de procesos de aprendizaje como codificación y representación de conocimientos, almacenamiento y recuperación de información así como, incorporación e integración de los nuevos conocimientos con los conocimientos previos.

La ciencia cognoscitivista influyó en el diseño instruccional dando paso al uso de organizadores avanzados, dispositivos nemónicos, metafóricos, segmentados en partes con significado y la organización cuidadosa del material instruccional de lo simple a lo complejo.

- **El Cognoscitismo y la Instrucción Basada en la Computadora**

El proceso que la computadora lleva a cabo en el cual se captura, se almacena y se recupera la información, es un proceso similar al que los investigadores

¹² MERGEL, BRENDA. Diseño Instruccional y Teoría del Aprendizaje. Universidad de Saskatchewan Canadá. [En línea] Disponible en: [http://www.educadis.uson.mx/Educa_subpaginas/Biblio-virt-art%EDculos-DI.htm]

cognitivos conciben como el proceso de información en los humanos. Esto hace pensar en los computadores como máquinas con inteligencia artificial.

En este caso se llama inteligencia artificial a la capacidad que tienen las computadoras de dar respuestas acertadas a los estudiantes mediante la consulta en una base de datos.

➤ **El Constructivismo en el Diseño Instruccional**

El cambio que se dio entre las dos anteriores teorías que eran muy similares a la constructivista fue mucho más drástico. Tanto el conductismo como el cognoscitismo son de naturaleza objetiva, ambos se soportan en el estudio de tareas y la medición del logro de estas. Por el contrario, el constructivismo origina un aprendizaje más abierto, en el que sus métodos y resultados no son tan fáciles de medir y podrían ser diferentes entre cada estudiante.

El conductismo y el constructivismo tienen perspectivas teóricas distintas, pero el cognoscitismo tiene similitudes con el constructivismo. Un ejemplo de su compatibilidad es el hecho de que comparten la analogía de comparar el procesamiento mental de la información con el de las computadoras.

“... los modelos de procesamiento de la información han adoptado el modelo de computadora de la mente como un procesador de información. El constructivismo agrega que este procesador de información debe verse justamente como un sorteador de datos, pero manejando su flexibilidad durante el aprendizaje haciendo hipótesis, probando las interpretaciones tentativas, etc.”¹³.

❖ **TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN (TIC's) ORIENTADAS A LA EDUCACIÓN**

Las TIC's han llegado a ser uno de los pilares básicos de la sociedad y hoy es necesario proporcionar al ciudadano una educación que tenga en cuenta esta realidad.

Las nuevas tecnologías de la Información y Comunicación son aquellas herramientas computacionales e informáticas que procesan, almacenan, sintetizan, recuperan y presentan información representada de la más variada forma.

Para todo tipo de aplicaciones educativas, las TIC's son medios y no fines. Es decir, son herramientas y materiales de construcción que facilitan el aprendizaje,

¹³ PERKINS, D. Technology meets constructivism: Do they make a marriage? Educational Technology, 1991. En [SCHWIER. 1998] en [MERGEL. 1998]

el desarrollo de habilidades y distintas formas de aprender, estilos y ritmos de los aprendices. Del mismo modo, la tecnología es utilizada tanto para acercar al aprendiz al mundo, como el mundo al aprendiz.

Las tecnologías de la información y la comunicación tienen varios aspectos para resaltar principalmente si se refiere a las TIC's encaminadas a la pedagogía. En el proceso de enseñanza aprendizaje, las TIC's son una herramienta para la conceptualización de los contenidos que hacen parte de currículos basados por competencias.

Además es importante señalar el impresionante cambio en la calidad de la enseñanza y el aprendizaje al implementar los diferentes tipos de TIC's como las plataformas de enseñanza aprendizaje y el software que se utilizan en las aulas inteligentes todo eso con el servicio de la multimedia.

El uso de las TIC's apunta a los objetivos siguientes para los estudiantes con el fin que:

- Dominen el manejo técnico de cada tecnología: conocimiento práctico del hardware y software que caracteriza a cada medio.
- Posean un conjunto de conocimientos, procedimientos y habilidades específicos que les permitan buscar, seleccionar, analizar, comprender y recrear la enorme y veloz cantidad de información a la que se accede a través de las TIC's.
- Desarrollen valores reflexivos y actitudes realistas y críticas hacia la tecnología que les permitan reconocer sus falencias y/o aportes en pro de la educación.
- Reconozcan medios y mediaciones tecnológicas en su vida cotidiana no sólo como recursos de ocio recreativo o de uso obligatorio, sino también como entornos para expresión y comunicación con otras personas y grupos.

❖ ESTÁNDARES DE E-LEARNING¹⁴

La constante capacitación es un requerimiento de las sociedades actuales. Con todas las presiones del día a día, utilizar Internet para capacitarnos no es una moda sino una herramienta que nos facilitará esta constante actualización.

Uno de los grandes problemas aún sin resolver de las nuevas tecnologías de la información y la comunicación aplicadas a la educación es la falta de una metodología común que garantice los objetivos de accesibilidad, interoperabilidad, durabilidad y reutilización de los materiales didácticos basados en Web.

¹⁴ Centro de Tecnologías de Información de Intec. *Estándares e-Learning: Estado del Arte*.
<http://empresas.sence.cl.>

En las actuales soluciones e-learning, generalmente los contenidos preparados para un sistema no pueden ser fácilmente transferidos a otro. Los estándares e-learning son el vehículo a través del cual será posible dotar de flexibilidad a las soluciones e-learning, tanto en contenido como en infraestructura. Ellos han abierto una puerta hacia una manera más coherente de empaquetar los recursos y contenidos, tanto para los estudiantes como para los desarrolladores.

Esta convergencia de tecnologías e-learning es muy importante para los consumidores de estas tecnologías, debido a que los productos que se adhieran a estos estándares no quedarán obsoletos a corto plazo, protegiendo así las inversiones realizadas en ellos.

Además, estándares comunes para asuntos tales como metadata de contenidos, empaquetamiento de contenidos, secuencia de contenidos, interoperabilidad de preguntas y test, perfil de alumnos, interacción en tiempo de ejecución, etc., son requisitos indispensables para el éxito de la economía del conocimiento y para el futuro del e-learning.

➤ **Elementos del e-learning**

E-learning no se trata solamente de tomar un curso y colocarlo en un computador. Se trata de una combinación de recursos, interactividad, soporte y actividades de aprendizaje estructuradas.

Por lo tanto, podemos definir e-learning como: *“aquella actividad que utiliza de manera integrada y pertinente computadores y redes de comunicación, en la formación de un ambiente propicio para la construcción de la experiencia de aprendizaje”*¹⁵

Esta modalidad puede ser efectuada en forma sincrónica o asincrónica, entendiendo por tales:

Sincrónica: Los alumnos se conectan en tiempo real con el facilitador y/o profesor, por tanto el proceso de aprendizaje sólo queda diferido en el espacio.

Asincrónica: Los alumnos se conectan, de acuerdo con sus tiempos, dentro de los parámetros que imponga el curso. En este caso el material con que se trabaja asume mayor importancia al igual que las tutorías, por tanto el proceso de aprendizaje queda diferido en el tiempo y en el espacio.

A continuación se describen los principales elementos del e-learning.

¹⁵ Definición basada en el trabajo desarrollado por <http://empresas.sence.cl> con un grupo de representantes de la industria chilena de e-learning.

♦ **Learning Management System o LMS**

Es el núcleo alrededor del cual giran los demás elementos. Básicamente se trata de un software para servidores de Internet/Intranet que se ocupa de:

- Gestionar los usuarios: inscripción, control de sus aprendizajes e historial, generación de informes, etc.
- Gestionar y lanzar los cursos, realizando un registro de la actividad del usuario: tanto los resultados de los test y evaluaciones que realice, como de los tiempos y accesos al material formativo.
- Gestionar los servicios de comunicación que son el apoyo al material online, foros de discusión, charlas, videoconferencia; programarlos y ofrecerlos conforme sean necesarios.

El panorama actual de los LMS está caracterizado por su gran dispersión, ya que todavía no hay entre ellas ningún liderazgo claro, comparable al existente en otras áreas de software, como por ejemplo en los programas de ofimática: procesadores de texto, hojas de cálculo, etc.

♦ **Courseware o Contenidos**

Los contenidos para e-learning pueden estar en diversos formatos, en función de su adecuación a la materia tratada. El más habitual es el WBT (Web Based Training), cursos online con elementos multimedia e interactivos que permiten que el usuario avance por el contenido evaluando lo que aprende.

Sin embargo, en otros casos puede tratarse de una sesión de “aula virtual”, basada en videoconferencia y apoyada con una presentación en forma de diapositivas tipo PowerPoint, o bien en explicaciones en una “pizarra virtual”.

En este tipo de sesiones los usuarios interactúan con el docente, dado que son actividades sincrónicas en tiempo real. Lo habitual es que se complementen con materiales online tipo WBT o documentación adicional que puede ser descargada e impresa.

Otras veces el contenido no se presta a su presentación multimedia, por lo que se opta por materiales en forma de documentos que pueden ser descargados, complementados con actividades online tales como foros de discusión o charlas con los tutores.

♦ **Sistemas de comunicación sincrónica y asincrónica**

Un sistema sincrónico es aquel que ofrece comunicación en tiempo real entre los estudiantes o con los tutores. Por ejemplo, las charlas o la videoconferencia.

Los sistemas asincrónicos no ofrecen comunicación en tiempo real, pero por el contrario ofrecen como ventaja que las discusiones y aportes de los participantes quedan registrados y el usuario puede estudiarlos con detenimiento antes de ofrecer su aporte o respuesta.

La diferencia fundamental entre el e-learning y la enseñanza tradicional a distancia está en esa combinación de los tres factores, en proporción variable en función de la materia a tratar:

Seguimiento + contenido + comunicación.

Esquemáticamente, los distintos componentes de una solución e-learning se pueden ver de la siguiente manera:

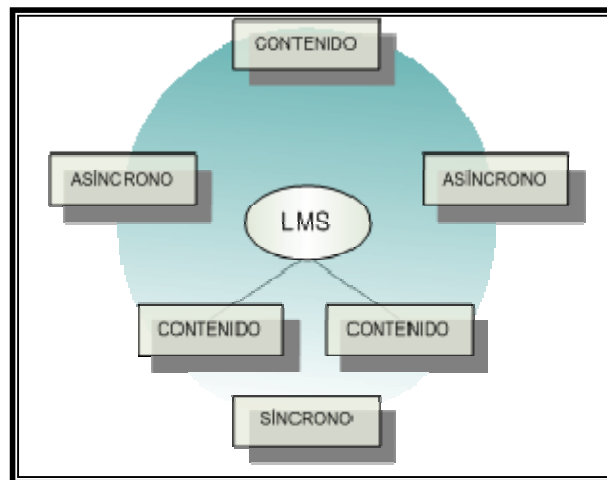


Figura 4. Componentes de una solución e-learning¹⁶

➤ **Sobre la estandarización**

En el mercado existen tanto LMS como Courseware de muchos fabricantes distintos. Por ello se hace necesaria una normativa que permita compatibilizar los distintos sistemas y cursos a fin de lograr dos objetivos:

- Que un curso de cualquier fabricante pueda ser cargado en cualquier LMS de otro fabricante, evitando restricciones de uso y ampliando el cubrimiento de los contenidos de cada curso.
- Que los resultados de la actividad de los usuarios en el curso puedan ser registrados por el LMS, logrando de esta manera obtener información relevante sobre el funcionamiento del curso y el desarrollo educativo del usuario.

¹⁶ Adaptación hecha por las autoras del proyecto a partir de: [<http://empresas.sence.cl/documentos/elearning>].

Como se puede ver en la siguiente figura, los distintos estándares que se desarrollan hoy en día para la industria del e-learning se pueden clasificar en los siguientes tipos:

1. Sobre el Contenido o Curso: Estructuras de los contenidos, empaquetamiento de contenidos, seguimiento de los resultados.

2. Sobre el Alumno: Almacenamiento e intercambio de información del alumno, competencias (habilidades) del alumno, privacidad y seguridad.

3. Sobre la interoperabilidad: Integración de componentes del LMS, interoperabilidad entre múltiples LMS.

Al hablar sobre un estándar e-learning, se hace referencia a un conjunto de reglas en común que especifican cómo los fabricantes pueden construir cursos on-line y las plataformas sobre las cuales son impartidos estos cursos de tal manera que puedan interactuar unas con otras.

Estas reglas además, definen un modelo de empaquetamiento estándar para los contenidos. Los contenidos pueden ser empaquetados como “objetos de aprendizaje” (*learning objects* o LO), de tal forma que permitan a los desarrolladores crear contenidos que puedan ser fácilmente reutilizados e integrados en distintos cursos.

Finalmente, los estándares permiten crear tecnologías de aprendizaje más poderosas, y “personalizar” el aprendizaje basándose en las necesidades individuales de los alumnos.

Básicamente, lo que se persigue con la aplicación de un estándar para el e-learning es lo siguiente:

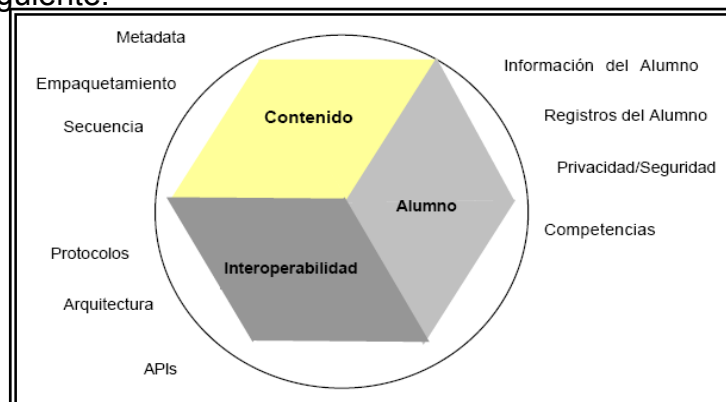


Figura 5. Tipos de estándares del e-learning ¹⁷

¹⁷ Tomado de: [\[http://empresas.sence.cl/documentos/elearning\]](http://empresas.sence.cl/documentos/elearning)

- *Durabilidad*: Que la tecnología desarrollada con el estándar evite la obsolescencia de los cursos.
- *Interoperabilidad*: Que se pueda intercambiar información a través de una amplia variedad de LMS.
- *Accesibilidad*: Que se permita un seguimiento del comportamiento de los alumnos.
- *Reusabilidad*: Que los distintos cursos y objetos de aprendizaje puedan ser reutilizados con diferentes herramientas y en distintas plataformas.

❖ ESTILOS DE APRENDIZAJE DE FELDER Y SILVERMAN (FSLSM)

El término ‘estilo de aprendizaje’ se refiere al hecho de que cada persona utiliza su propio método o estrategias a la hora de aprender. Aunque las estrategias varían según lo que se quiera aprender, cada uno tiende a desarrollar ciertas preferencias o tendencias globales, tendencias que definen un estilo de aprendizaje. Se habla de una tendencia general, puesto que, por ejemplo, alguien que casi siempre es auditivo puede en ciertos casos utilizar estrategias visuales.

Cada persona aprende de manera distinta a las demás: utiliza diferentes estrategias, aprende con diferentes velocidades e incluso con mayor o menor eficacia aún cuando tengan las mismas motivaciones, el mismo nivel de instrucción, la misma edad o estén estudiando el mismo tema.

Sin embargo, más allá de esto, es importante no utilizar los estilos de aprendizaje como una herramienta para clasificar a los alumnos en categorías cerradas, ya que la manera de aprender evoluciona y cambia constantemente.

Los distintos modelos y teorías existentes sobre estilos de aprendizaje ofrecen un marco conceptual que permite entender los comportamientos diarios en el aula, cómo se relacionan con la forma en que están aprendiendo los alumnos y el tipo de acción que puede resultar más eficaz en un momento dado.

Tabla 1. Modelos de estilos de aprendizaje más conocidos.

Según el hemisferio cerebral	Lógico Holístico
Según el cuadrante cerebral (Herrmann)	Cortical izquierdo Limbico izquierdo Limbico derecho Cortical derecho
Según el sistema de representación (PNL)	Visual Auditivo Kinéstico
Según el modo de procesar la información (Kolb)	Activo Reflexivo Pragmático Teórico

Según la categoría bipolar (Felder y Silverman)	Activo/Reflexivo Sensitivo/inductivo Visual/Verbal Secuencial/Global
Según el tipo de inteligencia (Gardner)	Lógico-matemático lingüístico-verbal Corporal-kinéstico Espacial Musical Interpersonal Intrapersonal Naturalista

El modelo Felder y Silverman ha sido el resultado final de un trabajo de investigación de muchos años. Fue diseñado con dimensiones dicotómicas que pueden ser particularmente importantes si se aplican al campo de las Ciencias de la Educación y al aprendizaje asistido por computador. En la siguiente tabla se pueden observar tales dimensiones:

Tabla 2. Dicotomías de los cuatro niveles de estilos de aprendizaje del modelo FLSM

DICOTOMÍA	
Activo	Reflexivo
Sensitivo	Intuitivo
Visual	Verbal
Secuencial	Global

A partir de las cuatro dicotomías, el modelo FLSM clasifica los estilos de aprendizaje con base en las respuestas que se puedan obtener a las siguientes preguntas:

Tabla 3. Cuestionario planteado por el modelo de aprendizaje de Felder y Silverman.

PREGUNTA	DIMENSIÓN DEL APRENDIZAJE Y ESTILOS	DESCRIPCIÓN DE LOS ESTILOS
¿Qué tipo de información perciben preferentemente los estudiantes?	Dimensión relativa al tipo de información: sensitivos-intuitivos	Básicamente, los estudiantes perciben dos tipos de información: información externa o sensitiva a la vista, al oído o a las sensaciones física e información interna o intuitiva a través de memorias, ideas, lecturas, etc.
¿A través de qué modalidad sensorial es más efectivamente percibida la información cognitiva?	Dimensión relativa al tipo de estímulos referenciales: visuales-verbales	Con respecto a la información externa, los estudiantes básicamente la reciben en formatos visuales mediante cuadros, diagramas, gráficos,

		demostraciones, etc. o en formatos verbales mediante sonidos, expresión oral y escrita, fórmulas, símbolos, etc.
¿Cómo progresa el estudiante en su aprendizaje?	Dimensión relativa a la forma de procesar y comprensión de la información: secuenciales-globales	El progreso de los estudiantes sobre el aprendizaje implica un procedimiento secuencial que necesita progresión lógica de pasos incrementales pequeños o entendimiento global que requiere de una visión integral.
¿Cómo prefiere el estudiante procesar la información?	Dimensión relativa a la forma de trabajar con la información: activos-reflexivos	La información se puede procesar mediante tareas activas a través de compromisos en actividades físicas o discusiones o a través de la reflexión o introspección.

De acuerdo con esta información, los estudiantes se clasifican en¹⁸:

1. ACTIVOS Y REFLEXIVOS

Los estudiantes activos, para conservar y entender la información lo mejor posible, tienden a hacer algo activo con ella. Discuten, o aplican explicando a otros lo que han aprendido. Los reflexivos, por su parte, prefieren pensar y analizar la información primero.

Sentarse en una conferencia sin poder hacer ninguna actividad con la información y solo tomar nota es difícil para el activo como para el reflexivo, pero particularmente más para el primero. Todas las personas son activas a veces y reflexivas a veces. Su preferencia por una categoría o la otra puede ser fuerte, moderada o leve. Un equilibrio entre los dos es deseable. Si se actúa siempre antes de reflexionar es posible pasar cosas por alto, mientras que si se pasa demasiado tiempo reflexionando puede que nunca se logre alguna acción concreta con base en el conocimiento.

Según Felder el estudiante activo puede ayudarse involucrándose en grupos de estudio en donde los miembros retomen cuantas veces sea necesario un asunto y lo expliquen unos a otros. También es importante que plantee hipótesis sobre las respuestas que daría para determinada pregunta. El estudiante activo conservará mejor la información siempre que encuentre la forma de hacer algo con ella.

Por otra parte el estudiante reflexivo puede ayudarse repasando periódicamente lo que lee y planteándose preguntas o usos posibles sobre esa información. Para este tipo de estudiante puede ser provechoso escribir resúmenes cortos sobre

¹⁸ Adaptado por las autoras del proyecto a partir de:
[<http://www4.ncsu.edu/unity/lockers/users/ff/felder/public/ILSdir/styles.htm>]

lecturas o notas de la clase con sus propias palabras. Hacer esto le puede tomar tiempo adicional, pero le permitirá conservar el material con mas eficiencia.

2. SENSITIVOS E INTUITIVOS.

Los estudiantes sensitivos tienden a ser pacientes con los detalles y buenos en memorizar hechos y en hacer trabajo de laboratorio (práctico). Los intuitivos pueden ser mejores en captar nuevos conceptos y a menudo se desenvuelven mejor que los sensitivos en el análisis que implique abstracciones y formulaciones matemáticas.

Los sensitivos procuran ser más prácticos y cuidadosos, los intuitivos se inclinan a trabajar más rápidamente y a ser más innovadores que los sensitivos. A estos últimos no les llaman la atención los cursos que no tienen ninguna conexión evidente con el mundo real.

Todas las personas pueden ser sensitivas a veces e intuitivas a veces. Su preferencia por una categoría o la otra puede ser fuerte, moderada o leve. Para ser eficaz en la resolución de problemas, un estudiante necesitará poder funcionar de ambas maneras. Si se acentúa la intuición demasiado, es posible pasar por alto detalles importantes o recurrir a equivocaciones en cálculos manuales por un descuido; si se acentúa demasiado la sensibilidad, se puede confiar demasiado en la memoria y no concentrarse bastante en entender.

Los sensitivos recuerdan y entienden la información mejor cuando pueden ver como esta conectada con el mundo verdadero. Si un estudiante sensitivo asiste a una clase donde el material, en su mayoría, es abstracto y teórico, puede tener dificultades, es necesario proporcionarle ejemplos específicos de conceptos y de procedimientos, y permitirle descubrir cómo los conceptos se aplican en la práctica. Un estudiante intuitivo que esta en una clase donde se trabaja memorísticamente y con muchas formulas puede tener problemas de aburrimiento, así que es importante ofrecer a este tipo de estudiante interpretaciones de las teorías ligadas a los hechos.

3. VISUALES Y VERBALES

Los estudiantes visuales recuerdan lo mejor posible aquello que ven, los cuadros, los diagramas, los organigramas, las líneas del tiempo, las películas, y las demostraciones. Los estudiantes verbales captan más de las palabras escritas y de las explicaciones habladas. Cada uno aprende más cuando la información se presenta visualmente y verbalmente.

En la universidad generalmente la información visual se presenta en menor cantidad que la verbal; los estudiantes escuchan principalmente las conferencias y

leen el material escrito en las pizarras y en libros de textos y folletos. Desafortunadamente, la mayoría de los estudiantes son principalmente visuales.

Los buenos estudiantes son capaces de procesar la información presentada visualmente o verbalmente.

Los estudiantes visuales deben apoyarse en diagramas, bosquejos, diagramas esquemáticos, fotografías, organigramas, o algún otro material. Los estudiantes verbales podrían escribir resúmenes o el material gráfico en sus propias palabras.

4. SECUENCIALES Y GLOBALES

Los estudiantes secuenciales tienden a entender en pasos lineales, con cada paso siguiendo lógicamente el anterior. Los estudiantes globales tienden a aprender a grandes saltos, con material elegido casi aleatoriamente sin ver conexiones, y después repentinamente "conseguirlas."

Los estudiantes secuenciales no pueden entender completamente el material pero pueden sin embargo hacer algo con él puesto que los pedazos que han absorbido están conectados lógicamente. Los estudiantes fuertemente globales que carecen de buenas capacidades de pensamiento secuencial, por otra parte, pueden tener dificultades serias hasta que tienen una visión general. Incluso después de tenerla, pueden no comprender los detalles del tema, mientras que los estudiantes secuenciales pueden saber mucho sobre aspectos específicos de un tema pero pueden tener dificultades al querer entender el todo del mismo.

Si se observa lo expuesto en la Tabla 4D, las herramientas de navegación presentadas son idóneas para casi todos los estilos de aprendizaje o se pueden adaptar para estudiantes globales, secuenciales o reflexivos. La idea principal de realizar esta clasificación de elementos es poder presentar los contenidos y el entorno de aprendizaje que más se acerque a la primera aproximación del estilo de aprendizaje del estudiante, obtenida mediante la aplicación del cuestionario del modelo FLSM (ver anexo). Posteriormente, este perfil se refina mediante la misma interacción del estudiante con los materiales didácticos.

Tabla 4. Componentes de un curso hipermedia para los objetos de aprendizaje de una unidad docente

A. Estrategia Instruccional

	Objetivos	Casos de estudio	Lecturas	Núcleos del conocimiento	Mapas conceptuales	Síntesis
Global	√					√
Secuencial					√	
Verbal	√		√		√	

Visual		√			√	√
Activo				√		
Reflexivo	√	√	√		√	
Sensitivo		√			√	
Intuitivo	√				√	

B. Materiales instruccionales complementarios y elementos de interactividad y de evaluación.

	Ejemplos	Animaciones	Simulaciones	Gráfico interactivo	Glosarios	Ejercicios de auto evaluación	Ejercicios de respuesta abierta
Global	√			√	√	√	√
Secuencial	√	√	√	√	√	√	√
Verbal	√				√	√	√
Visual	√	√	√	√		√	
Activo	√		√			√	√
Reflexivo	√	√	√	√	√	√	√
Sensitivo			√	√			√
Intuitivo	√	√	√	√	√	√	√

C. Formato del material.

	Diapositivas		Media clips			Texto lineal
	Texto	Multimedia	Gráficos	Video digital	Audio	
Global			√	√		
Secuencial	√	√		√	√	√
Verbal	√				√	√
Visual		√	√	√		
Activo						√
Reflexivo		√	√	√		√
Sensitivo		√	√	√	√	√
Intuitivo	√	√	√	√	√	√

D. Herramientas de navegación.

	Puntuales			Estructurales			Para el trabajo colaborativo	
	Flechas (avanzar y retroceder)	Impresiones	Ayuda en línea	Mapas de visión general	Filtros	Chat	Forum	Correo electrónico
Global				√	√	√	√	√
Secuencial	√	√	√			√	√	√
Verbal	√	√	√	√	√	√	√	√
Visual	√	√	√	√	√	√	√	√
Activo	√	√		√	√	√	√	√
Reflexivo	√	√	√	√	√			√
Sensitivo	√	√	√	√	√	√	√	√
Intuitivo	√	√	√	√	√	√	√	√

❖ ESTRATEGIAS DE ENSEÑANZA APRENDIZAJE E INNOVACIÓN EDUCATIVA

Si se trata de dar respuesta a la pregunta de cómo enseñar, el primer término que sale a relucir es el de método, entendiéndose por método la forma de dar o realizar una orden¹⁹.

Para el caso de la enseñanza el método es el camino a seguir sobre la estructura curricular diseñada para lograr motivación en el estudiante.

Basándose en los modelos pedagógicos, las teorías de aprendizaje y las perspectivas curriculares se pueden determinar una serie de proposiciones metodológicas que se pueden catalogar en dos grupos.

El primero define métodos lógicos o del conocimiento y métodos pedagógicos o tradicionales resumidos en la siguiente tabla.²⁰

Tabla 5. Clasificación de métodos de enseñanza de Pienkevich y González

Métodos lógicos	(Permiten la obtención o producción del conocimiento)
Método inductivo	Lo que se estudia se presenta por medio de casos particulares, hasta llegar al principio general que lo rige. Se basa en la experiencia, la observación y los hechos que suceden. Consiste en inferir proposiciones particulares de premisas generales o universales.
Método deductivo	El docente presenta conceptos, principios o definiciones de las cuales se extraen conclusiones y consecuencias. Emplea procedimientos como la aplicación, la comprobación y la demostración.
Método analítico	Estudia los fenómenos y los hechos separando sus elementos constitutivos, determinando su importancia, relación entre ellos, su organización y funcionamiento. Le atañen procedimientos como la división y la clasificación.
Método sintético	Complementario al método analítico; reúne las partes separadas para verlas como un todo. Exige al estudiante combinar los elementos en una estructura organizada. Presenta como procedimientos el resumen, la sinopsis,

¹⁹ CATALANO, ANA; AVOLIO, SUSANA; SLADOGNA, MÓNICA. Diseño Curricular Basado en Normas de Competencia Laboral: Conceptos y Orientaciones Metodológicas. 1º ed. - Buenos Aires: Banco Interamericano de Desarrollo, 2004. [En línea]. Disponible en: [\[http://www.cinterfor.org.uy/public/spanish/region/ampro/cinterfor/publ/dis_curr/pdf/dis_curr.pdf\]](http://www.cinterfor.org.uy/public/spanish/region/ampro/cinterfor/publ/dis_curr/pdf/dis_curr.pdf)

²⁰ HÉRNANDEZ POU, PRISCILLA. Psicología Educativa y Método de Enseñanza. Santo Domingo, República Dominicana, 2004. [En línea]. Disponible en: <http://www.monografias.com/trabajos5/psicoedu/psicoedu.shtml>

Métodos pedagógicos	la recapitulación, el esquema, el diagrama, la definición. El estudiante recibe como dogma todo lo que el maestro o el libro de texto le transmiten.
----------------------------	---

El segundo centrado en el contexto universitario establece tres grandes métodos:

- **Método magistral:** El profesor mediante una exposición sistemática y sintetizada utilizando la metodología de lección magistral expresa todos los contenidos a sus estudiantes valiéndose de herramientas como los gestos y las preguntas.
- **Trabajo autónomo de los estudiantes:** Aplicada a aprendizajes de tipo práctico y funcional, este método permite que el estudiante aprenda a su propio ritmo y ajustándose a las circunstancias particulares a la que éste se encuentre sometido. Se han desarrollado muchas formas combinadas de esta metodología con otras, como la lección magistral o el trabajo en seminarios. Algunas de estas formas están presentadas en la siguiente tabla.²¹

Tabla 6. Formas del trabajo autónomo

Contratos de aprendizaje	Se establecen las condiciones del “contrato” de trabajo entre profesores y estudiantes, dejando claro los contenidos de aprendizaje como las tareas a realizar (incluyendo en algunas ocasiones, nivel de exigencia y la calificación que se desea obtener por el estudiante). El proceso se desarrolla bajo una supervisión “contratada” de los profesores.
Guías de aprendizaje	Los docentes preparan guías que posean información suficiente (suele ser abundante) sobre el proceso de estudio de la disciplina: contenidos, sugerencias, materiales, dificultades que se pueden encontrar y posibles soluciones, actividades para realizar, autoevaluaciones, etc. Las primeras sesiones se realizan de forma presencial y después el estudiante sigue con el trabajo autónomo bajo la información de la guía.
Semi - presenciales	Se combinan las sesiones presenciales con el trabajo independiente del estudiante. Las sesiones presenciales tienen como objetivo mostrar los conceptos relevantes de la disciplina o asignatura, la metodología de estudio, supervisar y reforzar el proceso y sintetizar los puntos clave.

²¹ ZABALZA, MIGUEL A. Competencias Docentes del Profesorado Universitario. Calidad y Desarrollo Profesional. Madrid: Nancea S.A. Editores, 2003. En: [ESTRADA. 2005]

A distancia

Es el aprendizaje autónomo por antonomasia. Los contenidos y las orientaciones a su respecto se proveen mediante materiales impresos o digitales (pueden ser en plataformas en línea). El contacto con el docente se realiza vía telefónica, correo electrónico o foros en línea.

- **Trabajo en grupo:** Tiene una gran ventaja y es la retroalimentación de conocimientos y experiencias de los integrantes del grupo. Su principal virtud esta en el aprendizaje actitudinal. Tiene un adecuado funcionamiento cuando se necesita que exista intercambio de conocimientos entre profesor y estudiantes dando un ambiente propicio para el debate.

➤ Estrategias Metodológicas

Hacen referencia a los medios que se utilizan para llegar al pleno desarrollo de un asunto. Las estrategias se pueden definir como planes de acción, bajo ciertas normas determinadas, que se enfocan hacia un resultado de enseñanza y/o aprendizaje.

Se define a continuación el camino a seguir (estrategias que enmarcan) para el desarrollo del diseño curricular. Se tendrán en cuenta las estrategias metodológicas más significativas para dicho desarrollo.

➤ Aprendizaje Individual

El concepto de aprendizaje individual permanente o aprendizaje autodirigido ha existido desde la antigüedad. Sin embargo, las primeras referencias escritas se encuentran en los documentos publicados por Craik (1840) sobre los esfuerzos de auto-educación de varias personalidades de los Estados Unidos, y en un libro titulado Self - help, publicado por Smiles en 1859²².

El tema del aprendizaje autodirigido ha sido una de las áreas de investigación más importantes en el campo del aprendizaje de los adultos durante las últimas cuatro décadas. Al respecto es importante destacar los siguientes trabajos, que han servido de base para múltiples investigaciones²³:

²² VALENZUELA G, Jaime Ricardo, Los Tres "Autos" del aprendizaje: Aprendizaje estratégico en educación a distancia, 2000. En [ESTRADA. 2005]

²³ BAHAMÓN, José Hernando. El aprendizaje individual permanente: ¿cómo lograr el desarrollo de esta capacidad de los estudiantes?, Centro de Recursos Para el Aprendizaje, 2003. [En línea]. Disponible en: [http://www.eduteka.org/pdfdir/cartilla_aprendizaje.pdf]

- Las investigaciones de Houle (1961), en la Universidad de Chicago, que clasificó a los estudiantes según las razones por las cuales participaban en su aprendizaje en tres categorías:
 - 1) Estudiantes orientados a las metas.
 - 2) Estudiantes orientados a las actividades.
 - 3) Estudiantes orientados al aprendizaje como un fin en sí mismo.

- El trabajo de Knowles, en 1975, quien popularizó el término Androgogía como referencia a los procesos de instrucción de adultos, y quien en su trabajo sobre aprendizaje autodirigido enunció los siguientes supuestos, que han servido de base a muchas de las investigaciones actuales:
 - 1) En el aprendizaje autodirigido se parte del supuesto de que los humanos crecen en capacidad y necesitan ser auto-aprendices.
 - 2) Las experiencias de los aprendices son una fuente importante para el aprendizaje.
 - 3) El aprendizaje individual es requerido para lograr una evolución en las tareas a lo largo de la vida.
 - 4) La orientación natural de los adultos es hacia el aprendizaje centrado en las tareas o los problemas.
 - 5) El aprendizaje auto-controlado es motivado por varios incentivos internos, tales como la necesidad de auto-estima, la curiosidad, el deseo de logro y la satisfacción de concluir una tarea.

- El desarrollo de un instrumento, para medir la capacidad de aprendizaje autodirigido, elaborado por Guglielmino (1977) y conocido como “Self directed learning Readiness Scale (SDLRS)”.

A pesar de todas estas investigaciones realizadas en los últimos años no hay todavía una teoría completa y coherente que haya sido adoptada por los investigadores y teóricos del campo de la educación de adultos; prueba de ello son las diversas definiciones que existen sobre este concepto.

Como estas definiciones son generalizaciones no aceptadas universalmente, es importante revisar algunos de los diferentes consensos adoptados por la mayoría de los investigadores sobre el tema:

- Los aprendices pueden ser gradualmente “empoderados” para aceptar mayor responsabilidad por las diferentes decisiones asociadas con los propósitos y el esfuerzo del aprendizaje.
- El aprendizaje autodirigido puede ser considerado como una característica que existe en algún grado en cada persona y situación de aprendizaje.

- El aprendizaje autodirigido no necesariamente significa que todo el aprendizaje deberá ocurrir de manera individual y en total aislamiento de los demás.
- Los auto-aprendices parecen ser capaces de transferir aprendizajes, en términos de conocimientos y habilidades de estudio, de una situación a otra.
- El estudio autodirigido involucra un conjunto de actividades y recursos tales como las auto-guías de lectura, la participación en grupos de estudio, las actividades de reflexión y la escritura de ensayos.
- Algunos de los roles efectivos de los profesores en el aprendizaje autodirigido son: facilitador de recursos; gestor del diálogo entre el estudiante y el profesor y entre los estudiantes; evaluador de los resultados obtenidos; y promotor del pensamiento crítico.
- Algunas opciones válidas para el logro del desarrollo de la capacidad de aprendizaje autodirigido, que pueden ser empleadas por las instituciones de educación superior, son: las opciones de estudio individualizado, la oferta de cursos no tradicionales, y los programas innovadores.

Otro aspecto fundamental del aprendizaje autodirigido, que es importante destacar, y que ha sido presentado por algunos autores, es la dicotomía del término, pues éste puede ser entendido al menos de dos formas: como una metodología para el aprendizaje o como una meta formulada en términos de la capacidad que los estudiantes deben alcanzar como resultado de su proceso de formación.

Otra aproximación interesante al concepto de aprendizaje autodirigido es el modelo presentado²⁴, basado completamente en la relación existente entre el aprendiz y la institución en función del control de las decisiones sobre los objetivos y los medios de aprendizaje.

De acuerdo con este modelo, el aprendizaje autodirigido ocurre cuando el aprendiz controla tanto los objetivos como los medios de aprendizaje; esta definición constituye una posición extrema del concepto, empleada por algunos autores para argumentar que solamente es posible lograr algunos grados de auto-dirección por parte de los alumnos, por cuanto la necesidad de mantener los estándares institucionales no permite que los aprendices establezcan los objetivos de formación, ni controlen los medios de aprendizaje.

²⁴ MOCKER, D. W., SPEAR, G. E. Lifelong Learning: Formal, Nonformal, Informal, and Self-Directed. The National Center for Research in Vocational Education, The Ohio State University, 1982. En: [LOWRY. 2003]

➤ El Método de Casos

El método de casos enseña basado en casos, instala al participante dentro de una situación real y le da la oportunidad de dramatizar sus propios enfoques y sus decisiones, lo cual lo prepara para la acción²⁵.

El propósito de un método de casos es dar a los estudiantes la oportunidad de adquirir un entendimiento generalizado de los problemas que pueden encontrar y de ayudarles a desarrollar habilidades y destrezas para su solución, de una forma sistemática, que conduzca a soluciones viables.

Un método de casos se puede definir como la descripción narrativa que hace un grupo de observadores de una determinada situación de la vida real, incidente o suceso, que envuelva una o más decisiones. Este debe contener además del hecho o problema, la información básica apropiada que conduzca a la decisión o decisiones que conlleven a una solución, o varias opciones²⁶.

En este método la palabra (tomada como la descripción de los hechos), es fundamental para la elaboración narrativa de los sucesos, y debe tenerse en cuenta que el narrador no debe hacer ninguna interpretación, ya que esta puede cambiar la percepción del receptor. Las sesiones pueden durar entre 20 minutos y dos horas, pueden ser organizados grupos de personas para que realicen una socialización del caso antes de la reunión general, a fin de compartir las ideas, detectar líderes, detectar estudiantes tímidos, etc. El objetivo de esta confrontación en grupos pequeños, es el de encontrar puntos de vista antagónicos.

La utilidad del método de casos es aproximar al individuo a las condiciones de la vida real, para prepararlo desarrollando talentos latentes de visión, autoridad, comunicación y liderazgo, que lo capacite para la confrontación civilizada, la comunicación ágil y efectiva, el procesamiento de la información racional y objetiva y la toma de decisiones en condiciones de incertidumbre.

El método de casos desarrolla en los participantes un amplio conocimiento de la naturaleza humana y de su psicología que es la materia prima del individuo, con la característica de desarrollar gradualmente la capacidad de comunicarse. De igual manera el sistema de casos puede contribuir grandemente al desarrollo de las habilidades del ser humano en la medida en que vincula al participante con

²⁵ ESTRADA, Lilia. Elaboración y Documentación de una Propuesta de Diseño Curricular bajo la Visión de Competencias para la Asignatura Mediciones Eléctricas y Estudio de su Implementación en una Plataforma e-learning, Tesis de grado. Universidad Industrial de Santander, 2005

²⁶ SERNA, José María. El Método de Casos: Reflexiones sobre el cambio en la Metodología de la Enseñanza del Derecho en México, 2003. [En línea]. Disponible en: <http://www.bibliojuridica.org/libros/4/1793/9.pdf>

hechos reales y le permiten desarrollar su propio análisis y adoptar una solución que considere adecuada. En este sentido los casos presentan las siguientes características:

- Permiten la aplicación de conceptos teóricos y técnicos probados en la vida real.
- Ayudan al participante a desarrollar habilidades tanto para resolver problemas, como para tomar decisiones.
- Requieren de una participación activa y favorecen al desarrollo de las facilidades de comunicación.
- Pueden replicar situaciones de crítica, de riesgo o incertidumbre, que son propios de la vida real.
- Contribuyen a dar un enfoque práctico y pragmático a situaciones diversas y variadas.
- Descargan en el participante la responsabilidad de su propio aprendizaje, y lo motivan a permanecer informado y activo en su profesión.

De lo anterior se desprende que el Método de Casos puede ser altamente formativo pero requiere una gran disciplina y una enorme preparación por parte del profesor en todo sentido.

• Componentes de Método de Casos

- *El Alumno:* El cual debe ser participante, tiene un bagaje único de sentimientos, experiencias percepciones tradiciones y valores que lo llevan a interpretar las cosas de una manera única, a dar valor a una cosa, a desestimar otra. Cada individuo es único y es posible que en unas situaciones represente la uniformidad y en otras el contraste, esta diversidad es la que hace que este método se enriquezca y se convierta en un proceso activo.
- *El Caso:* El fin primordial de este es servir como base de la discusión; no se trata de un mecanismo para difundir reglas o principios.
- *El Profesor:* El cual tiene su propio sistema de referencia, basado en su propia formación, su experiencia particular y es afectado por el entorno económico, social y cultural. Suponer que el profesor sabe más y mejor que nadie todo lo relacionado con un caso determinado equivaldría a optar por el principio de subordinación que da al traste con la interdependencia que es la tendencia actual. Éste debe jugar un papel dinamizador y brindar los siguientes aspectos:
 - Proporcionar instrumentos y servicios requeridos para la discusión.
 - Mantener el orden del procedimiento.
 - Orientar la discusión para evitar posiciones simplistas.
 - Motivar la participación y estimular planteamiento de tesis novedosas.
 - Correlacionar los aportes individuales.

- Mantener el ritmo de la discusión de tal modo que permita el adecuado uso del tiempo y la comprensión de los asistentes.
 - Actuar como preceptor y consejero para mantener la discusión dentro de un ambiente ideal.
 - Mantener el interés de los participantes en el tema en curso aportando autoridad, dirección, e incluso humor para procurar un ambiente cálido dinámico y agradable.
- *La Asignatura:* Proporciona los conceptos, temas, métodos y herramientas cuya validez y vigencia serán sometidos a prueba en la discusión del caso.[SERNA. 2003]

➤ **El Aprendizaje Basado en Problemas**

El Aprendizaje Basado en Problemas (PBL-en inglés) es una alternativa interesante al aprendizaje en el aula tradicional.

Con el PBL, el profesor presenta un problema, sin clase o tarea o ejercicios. Dado que no es impartido, el aprendizaje se activa en el sentido que el estudiante descubre y trabaja con el contenido que el mismo determina necesario para resolver el problema.

En el PBL, el profesor actúa como facilitador y mentor, más que como una fuente de “soluciones”.

El Aprendizaje Basado en Problemas ²⁷ brinda oportunidades para:

- Evaluar e intentar lo que el estudiante conoce.
- Descubrir lo que el estudiante necesita aprender.
- Desarrollar sus habilidades interpersonales para lograr un desempeño más alto en equipos.
- Mejorar las habilidades de comunicación.
- Establecer y defender posiciones con evidencia y argumento sólido.
- Volverse más flexible en el procesamiento de información y enfrentar obligaciones.
- Practicar habilidades que necesite para su educación.

Empleado como estrategia y/o técnica, el aprendizaje basado en problemas consiste en el análisis y resolución de un problema seleccionado o diseñado para el cumplimiento de ciertos propósitos educativos por parte de los estudiantes, quienes se han organizado en pequeños grupos. En el proceso los estudiantes

²⁷ LANDSBERGER, Joe. Aprendizaje Basado en Problemas, 2005. [En línea]. Disponible en: [<http://www.studygs.net/espanol/pbl.htm>]

identifican las temáticas y necesidades de aprendizaje, se busca la información pertinente y finalmente se retoma el problema para darle solución²⁸.

➤ **Aprendizaje Basado en Proyectos (ABP)**

El aprendizaje basado en proyectos (ABP) es hoy en día una herramienta útil para muchos educadores, especialmente ambientales, sobre todo ahora que se ha enriquecido con los recursos de las TIC's, convirtiéndose en un poderoso vehículo para la promoción del aprendizaje significativo y el apropiamiento y uso efectivo de las tecnologías.

El aprendizaje basado en proyectos se enfoca en un problema que hay que solucionar o en una tarea que se debe llevar a cabo. La idea fundamental en la solución de problemas o la realización de tareas es la de que estas se construyen sobre el trabajo que se va a generar a partir del problema o tarea planteados.

Cuando se enfrenta un problema o tarea que constituye un desafío, se debe utilizar el conocimiento, las habilidades, y las ayudas que otras personas han desarrollado, así como el conocimiento propio, habilidades y la experiencia que se tiene. Por ejemplo, para iniciar un proyecto es recomendable trabajar básicamente en tres niveles:

- Herramientas que enriquecen y potencian la búsqueda de información, como los computadores, las bibliotecas tradicionales, bibliotecas digitales, organizaciones gubernamentales o no gubernamentales, entre otros.
- Herramientas que amplían las capacidades físicas, tales como el computador, el teléfono y la Red.
- El sistema de educación formal e informal, que proporciona elementos que ayudan a los miembros del grupo a construir y mantener sus capacidades físicas y mentales.

Uno de los principales objetivos que tiene la educación, es el facilitar experiencias, donde los estudiantes aprendan a reconocer, reflexionar, y proponer iniciativas para resolver problemas y tareas complejas. Los estudiantes necesitan recibir orientaciones muy claras, para poder iniciar y funcionar adecuadamente en el desarrollo del proyecto. Las características fundamentales del aprendizaje basado en proyectos son:

- Se basa en el aprender haciendo y el aprender a aprender, y busca preparar al estudiante para la vida profesional.

²⁸ GARZA, Eugenio. Las Técnicas Didácticas en el Modelo Educativo del Tec. de Monterrey. Col. Tecnológico, Monterrey, 2000. [En línea]. Disponible en: [<http://tecnologiaedu.us.es/bibliovir/pdf/309.pdf>]

- Induce a la exploración, la búsqueda y la selección de información, la generación y construcción de soluciones, el trabajo en grupo y la interdisciplinariedad.
- Permite diferentes niveles de complejidad del proyecto dependiendo de si el enfoque de desarrollo se establece sobre un tópico o mezcla varias disciplinas, lo que a su vez se refleja en un menor o mayor tiempo de ejecución respectivamente.
- El diseño y control del proyecto, es decir la selección de tópicos y los resultados esperados, puede estar completamente en manos del docente o del grupo de estudiantes o determinarse mediante mutuo acuerdo.

➤ **Aprendizaje Cooperativo y Proceso de Enseñanza**

Son de gran importancia las interacciones que establece el alumno con las personas que lo rodean, por lo cual se debe de tomar en cuenta la influencia educativa que ejerce en el alumno el aprendizaje cooperativo.

Según el estudio de varios y destacados psicólogos, se puede analizar el hecho de que los aprendizajes ocurren primero en un plano inter-psicológico (mediado por la influencia de los otros), y en segundo plano a nivel intra-psicológico, una vez que los aprendizajes han sido interiorizados debido al andamiaje que ejercen en el aprendiz aquellos individuos "expertos" que lo han apoyado a asumir gradualmente el control de sus actuaciones.

La enseñanza debe ser individualizada en el sentido de permitir a cada alumno trabajar con independencia y a su propio ritmo, pero al mismo tiempo es importante promover la colaboración y el trabajo grupal. En estudios realizados se ha comprobado que los estudiantes aprenden más, les agrada más la escuela, establecen mejores relaciones con los demás, aumentan su autoestima y aprenden habilidades sociales más efectivas cuando trabajan en grupos cooperativos que al hacerlo de manera individualista y competitiva.

Es aquí donde el aprendizaje cooperativo se compara (Tabla 7) con el aprendizaje individual para evaluar así los alcances y falencias de cada uno de ellos:

Tabla 7. Diferencias entre el Aprendizaje Individualista y el Cooperativo

APRENDIZAJE INDIVIDUALISTA	APRENDIZAJE COOPERATIVO
No existe relación entre los objetivos que persigue cada uno de los alumnos, las metas son independientes entre si.	Se establecen metas que son benéficas para si mismo y para los demás miembros del equipo.
El alumno percibe que el conseguir sus objetivos depende de su propia capacidad y esfuerzo, de la suerte y de la dificultad de la tarea.	El equipo debe trabajar junto hasta que todos los miembros del grupo hayan entendido y completado la actividad con éxito.
Existe una motivación extrínseca, con metas	Se busca maximizar el aprendizaje individual

orientadas a obtener valoración social y recompensas externas.

Los alumnos pueden desarrollar una percepción pesimista de sus capacidades de inteligencia.

Se evalúan a los estudiantes en pruebas basadas en los criterios, y cada uno de ellos trabaja en sus materias o textos ignorando a los demás.

La comunicación en clases con los compañeros es desestimada y muchas veces castigada.

Se convierte en un sistema competitivo y autoritario, produciendo una estratificación social en el aula.

pero al mismo tiempo el aprendizaje de los otros.

Los fracasos son tomados como fallas del grupo, y no como limitaciones personales en las capacidades de un estudiante.

Se evalúa el rendimiento académico de los participantes así como las relaciones afectivas que se establecen entre los integrantes.

Se basa en la comunicación y en las relaciones. Respeto hacia las opiniones de los demás.

Es un sistema que valora aspectos como la socialización, la adquisición de competencias sociales, el control de los impulsos agresivos, la relatividad de los puntos de vista, el incremento de las aspiraciones y el rendimiento escolar.

El aprendizaje cooperativo, de manera general se relaciona y facilita con los siguientes procesos:

- 1) Procesos Cognitivos:
 - a) Colaboración entre iguales.
 - b) Regulación a través del lenguaje.
 - c) Manejo de controversias.

- 2) Procesos Motivacionales:
 - a) Atribuciones.
 - b) Metas.

- 3) Procesos Afectivo Relacionales:
 - a) Pertenencia al grupo.
 - b) Autoestima.
 - c) Sentido.

El docente puede emplear el aprendizaje cooperativo en el aula para promover en sus estudiantes el hecho de que se sientan involucrados en las relaciones con sus compañeros (preocupación y apoyo), la capacidad de influir en las personas con las que están relacionados e involucrados y el disfrute de manera global del aprendizaje.

Este tipo de aprendizaje se caracteriza principalmente por:

- *Elevado grado de Igualdad:* debe existir un grado de simetría en los roles que desempeñan los participantes en una actividad grupal.

- *Grado de Mutualidad Variable:* Mutualidad es el grado de conexión, profundidad y bidireccionalidad de las transacciones comunicativas. Los

más altos niveles de mutualidad se darán cuando se promueva la planificación y la discusión en conjunto, se favorezca el intercambio de roles y se delimite la división del trabajo entre los miembros.

El aprendizaje cooperativo presenta una serie de componentes entre los que se puede encontrar:

- *Interdependencia Positiva:* Ocurre cuando los estudiantes pueden percibir un vínculo con el grupo de forma tal que no pueden lograr el éxito sin ellos y viceversa. Deben coordinar los esfuerzos con los compañeros para poder completar una tarea, compartiendo recursos, proporcionándose apoyo mutuo y celebrando juntos sus éxitos.
- *Interacción Promocional Cara a Cara:* Más que una estrella se necesita gente talentosa que no pueda hacer una actividad sola. La interacción cara a cara es muy importante ya que existe un conjunto de actividades cognitivas y dinámicas interpersonales que sólo ocurren cuando los estudiantes interactúan entre sí en relación a los materiales y actividades.
- *Valoración Personal o Responsabilidad Personal:* Se requiere la existencia de una evaluación del avance personal, la cual va haciendo tanto el individuo como el grupo. De esta manera el grupo puede conocer quien necesita más apoyo para completar las actividades, y evitar que unos descansen con el trabajo de los demás. Para asegurar que cada individuo sea valorado convenientemente se requiere:
 - Evaluar cuanto del esfuerzo que realiza cada miembro contribuye al trabajo de grupo.
 - Proporcionar retroalimentación a nivel individual así como grupal.
 - Auxiliar a los grupos a evitar esfuerzos redundantes por parte de sus miembros.
 - Asegurar que cada miembro sea responsable del resultado final.

Así mismo, el aprendizaje cooperativo se basa en una serie de objetivos que son:

- 1) En primer lugar los objetivos deben ser referentes a los aprendizajes esperados en relación con el contenido curricular. Se debe considerar el nivel conceptual y la motivación de los alumnos, los conocimientos previos y el propio significado de los materiales.
- 2) Los objetivos para el desarrollo de las habilidades de colaboración, donde deberá decidirse que tipo de habilidades de cooperación se enfatizarán.

Es recomendable que la conformación de los grupos contenga un máximo de 6 personas por equipos de trabajo. El rango puede variar de dos a seis. Los grupos

de trabajo también deben ser heterogéneos, colocando alumnos de nivel alto, medio y bajo (en cuanto al rendimiento académico u otro tipo de habilidades) dentro del mismo grupo. Otra recomendación es que los grupos de trabajo se acomoden en forma de círculos.

Las **estrategias de enseñanza** son las acciones planificadas por el agente de enseñanza o del docente, que tienen como fin promover el aprendizaje de los contenidos estructurados y el cumplimiento de los propósitos construidos.

Las **estrategias de aprendizaje** son una serie de conocimientos y pasos que el estudiante aprende y pone en práctica con el fin de solucionar sus problemas y necesidades de tipo académico. Dichas técnicas o estrategias no son solo asunto de los estudiantes, también son parte de las obligaciones del docente con el fin de lograr un adecuado y satisfactorio proceso de aprendizaje en sus estudiantes. Estas estrategias se dividen en dos grupos mencionados a continuación.

❖ ESTRATEGIAS GENERALES EN EL ÁMBITO EDUCATIVO

Se han identificado cinco tipos de estrategias generales en el ámbito educativo. Las tres primeras ayudan al alumno a elaborar y organizar los contenidos para que resulte más fácil el aprendizaje (procesar la información), la cuarta está destinada a controlar la actividad mental del alumno para dirigir el aprendizaje y, por último, la quinta está de apoyo al aprendizaje para que éste se produzca en las mejores condiciones posibles.²⁹

- **Estrategias de ensayo:** Son aquellas que implican la repetición activa de los contenidos (diciendo, escribiendo), o centrarse en partes claves de él. Son ejemplos: Repetir términos en voz alta, reglas nemotécnicas, copiar el material objeto de aprendizaje, tomar notas literales, el subrayado.
- **Estrategias de elaboración:** Implican hacer conexiones entre lo nuevo y lo familiar. Por ejemplo: Parafrasear, resumir, crear analogías, tomar notas no literales, responder preguntas (las incluidas en el texto o las que pueda formularse el alumno), describir como se relaciona la información nueva con el conocimiento existente.
- **Estrategias de organización:** Agrupan la información para que sea más fácil recordarla. Implican imponer estructura al contenido de aprendizaje, dividiéndolo en partes e identificando relaciones y jerarquías. Incluyen

²⁹ DÍAZ BARRIGA, Frida; HERNÁNDEZ ROJAS, Gerardo. Estrategia Docentes para un Aprendizaje Significativo: Una Interpretación constructivista. México: McGRAW-HILL Interamericana, 1998. [En línea]. Disponible en: http://www.uv.mx/cpp/doc/DC_Diaz_Barriga_Estrategias_de_ensenanza...doc

ejemplos como: Resumir un texto, esquema, subrayado, cuadro sinóptico, red semántica, mapa conceptual, árbol ordenado.

- ***Estrategias de control de la comprensión:*** Estas son las estrategias ligadas a la metacognición. Implican permanecer consciente de lo que se está tratando de lograr, seguir la pista de las estrategias que se usan y del éxito logrado con ellas y adaptar la conducta en concordancia. Si se utilizara la metáfora de comparar la mente con un ordenador, estas estrategias actuarían como un procesador central de ordenador. Son un sistema supervisor de la acción y el pensamiento del alumno, y se caracterizan por un alto nivel de conciencia y control voluntario. Entre las estrategias metacognitivas están: la planificación, la regulación y la evaluación.
 - *Estrategias de planificación:* Son aquellas mediante las cuales los alumnos dirigen y controlan su conducta. Son, por tanto, anteriores a que los alumnos realicen ninguna acción. Se llevan a cabo actividades como:
 - Establecer el objetivo y la meta de aprendizaje.
 - Seleccionar los conocimientos previos que son necesarios para llevarla a cabo.
 - Descomponer la tarea en pasos sucesivos.
 - Programar un calendario de ejecución.
 - Prever el tiempo que se necesita para realizar esa tarea, los recursos que se necesitan, el esfuerzo necesario.
 - Seleccionar la estrategia a seguir.
 - *Estrategias de regulación, dirección y supervisión:* Se utilizan durante la ejecución de la tarea. Indican la capacidad que el alumno tiene para seguir el plan trazado y comprobar su eficacia. Se realizan actividades como:
 - Formularles preguntas.
 - Seguir el plan trazado.
 - Ajustar el tiempo y el esfuerzo requerido por la tarea.
 - Modificar y buscar estrategias alternativas en el caso de que las seleccionadas anteriormente no sean eficaces.
 - *Estrategias de evaluación:* Son las encargadas de verificar el proceso de aprendizaje. Se llevan a cabo durante y al final del proceso. Se realizan actividades como:
 - Revisar los pasos dados.
 - Valorar si se han conseguido o no los objetivos propuestos.

- Evaluar la calidad de los resultados finales.
 - Decidir cuando concluir el proceso emprendido, cuando hacer pausas, la duración de las pausas, etc.
- **Estrategias de apoyo o afectivas:** Estas estrategias, no se dirigen directamente al aprendizaje de los contenidos. La misión fundamental de estas estrategias es fortalecer la eficacia del aprendizaje mejorando las condiciones en las que se produce. Incluyen: establecer y mantener la motivación, enfocar la atención, mantener la concentración, manejar la ansiedad, manejar el tiempo de manera efectiva, etc.

Por ultimo cabe señalar, que algunos autores relacionan las estrategias de aprendizaje con un tipo determinado de aprendizaje. Para estos autores cada tipo de aprendizaje (por asociación/por reestructuración) estaría vinculado a una serie de estrategias que le son propias.

- El aprendizaje asociativo: ESTRATEGIAS DE ENSAYO.
- El aprendizaje por reestructuración: ESTRATEGIAS DE ELABORACIÓN, O DE ORGANIZACIÓN.

➤ **Técnicas Metodológicas**

Son técnicas que indican la secuencia que se debe llevar para poder hacer fácil el proceso de enseñanza aprendizaje. Se sustentan en las estrategias, extrayendo de ellas el sentido y dirección en el proceso de enseñanza-aprendizaje y son los recursos que permiten aplicar los lineamientos de la estrategia³⁰. A continuación (Tabla 8) se realiza una descripción de algunas técnicas de enseñanza-aprendizaje:

Tabla 8. Técnicas de enseñanza-aprendizaje.

TÉCNICA DE ENSEÑANZA APRENDIZAJE	DESCRIPCIÓN
Foro	El grupo en su totalidad discute informalmente un tema, hecho o problema conducido por un coordinador.
Simposio	Equipo de expertos que desarrolla diferentes aspectos de un tema o problema en forma sucesiva ante el grupo.
Mesa Redonda	Equipo de expertos que sostienen puntos de vista divergentes o contradictorios sobre un mismo tema, expone ante el grupo en forma sorteada.
Panel	Equipo de expertos que discute un tema en forma de diálogo o conversación ante el grupo.
Seminario	Grupo que investiga o estudia intensamente un tema

³⁰ ARMAS, Sara. Técnicas Grupales de Enseñanza-Aprendizaje. Universidad de San Carlos de Guatemala, 1997. [En línea]. Disponible en: <http://medicina.usac.edu.gt/pfd/semanas/semana8/tecnicas.pdf>

Conferencia/ Exposición Oral/Expositiva	en sesiones planificada, recurriendo a fuentes originales de información. Presentación oral de un tema planificado por un grupo, al resto del gran grupo.
Discusión de grupo/ Discusión Colectiva	Desarrollo de un tema por medio de opiniones, preguntas ó información que aporta el grupo.
Cuchicheo/Phillips 2-2	Parejas o grupos de 2 personas que tratan un tema en voz baja.
Corrillos/Bina	Grupos de 6 personas que durante 6 minutos trabajan en un tema.
Phillips 6-6/Phillips 5-5	Grupo pequeño o grande con la tarea de encontrar posibles soluciones a problemas planteados.
Taller de trabajo	Enjuiciamiento de un tema ante un tribunal, incluye la Representación de un juez, secretario, fiscal, defensa testigos, jurado y público.
Juicio Educativo	Dos grupos de participantes que presentan puntos de vista en pro o en contra de determinada posición, argumentando o refutando al grupo contrario.
Debate	

❖ EVALUACIÓN

Es la última etapa en la parte del diseño curricular pero no por esto es de menor importancia, ya que gracias al proceso de evaluación podemos medir la eficacia de las técnicas aplicadas al proceso de enseñanza-aprendizaje.

Las funciones de la evaluación son el diagnóstico y la reestructuración. Diagnóstico por que permite evaluar todos los aspectos del proceso educativo (aprendizaje, enseñanza, acción docente, estudiantil, administrativa, contexto físico, social, educativo, el currículo, la institución, etc.) y restauración porque permite la elaboración de estrategias para la mejora de los procesos educativos. En el proceso de evaluación, existen varias fases a desarrollar:

- *Demarcación de la situación:* delimitar el objeto, nivel de referencia, asunto, hecho circunstancia que se va a evaluar.
- *Criterios de evaluación:* deben establecerse ciertos criterios o procedimientos que sirvan de factor de medición para emitir un juicio, en el caso del currículo los criterios provienen de los propósitos esencialmente, pero también se incluyen las intenciones de los otros elementos del currículo.
- *Recolección de la información:* los datos e informaciones son los que alimentan la evaluación y se obtienen por medio de procedimientos o técnicas libres o estandarizados, en esta recolección debe haber un compromiso que permita extraer la información suficiente para una evaluación efectiva.

- *Valoración de la información:* comparando los datos con los criterios establecidos se realizan juicios sobre el valor y pertinencia de estos, es decir se diagnostica alrededor de la situación evaluada.
- *Toma de decisión:* lo más importante de la evaluación es la posibilidad de realimentar, mejorar y reestructurar a través del diagnóstico realizado, para ello se debe generar e implementar las decisiones que sean necesarias.

➤ **Técnicas de Evaluación**

Para poder asignar un valor a la información recolectada con el fin de determinar la veracidad de la misma y así poder evaluar los conocimientos obtenidos en el proceso de aprendizaje es necesario valerse de una serie de técnicas que permitan realizar dicho proceso de una forma imparcial y objetiva. A continuación se mencionan algunas de las más importantes técnicas de evaluación, planteadas por [DEPRESBITERIS. 2002.]

○ **Observación**

Hay dos formas de observación: sistemática y asistemática.

- *La observación sistemática* es aquella en que el observador tiene objetivos previamente definidos y como consecuencia, sabe cuáles son los aspectos que evaluará.
- *La observación asistemática* es aquella que se refiere a las experiencias casuales, de las que el observador registrará el mayor número posible de informaciones, sin correlacionarlas previamente con objetivos claros y definidos.

○ **Pruebas Prácticas**

Las pruebas prácticas exigen que haya una observación sistemática. Los pasos sugeridos para la elaboración de una prueba práctica son:

- Definir la competencia que será evaluada.
- Seleccionar una o más tareas que permitan la manifestación de la competencia.
- Elaborar la prueba.
- Validar la prueba con un alumno.
- Aplicar la prueba.

○ **La Entrevista**

La entrevista es una técnica que propicia la recolección de datos de naturaleza cuantitativa y cualitativa. Puede ser individual o en grupo.

La gran ventaja de la entrevista es que ella permite la captación inmediata y continua de la información deseada. Permite también, profundizar en algunos aspectos que fueron observados de manera superficial. El instrumento que acompaña la técnica de la entrevista es el guión de preguntas.

Para la recolección de datos cuantitativos, el guión de la entrevista es más cerrado; en la recolección de datos cualitativos, el guión puede tener una estructura básica de preguntas que serán enriquecidas a medida que se desea profundizar determinados aspectos.

- **Test y Pruebas**

Testear quiere decir verificar alguna cosa por medio de situaciones previamente organizadas llamadas test. Hay varios tipos de test: de aptitudes, de actitudes, de maduración, de personalidad, de rendimiento escolar (en inglés achievement test).

Los test usados para la verificación del aprendizaje de los alumnos son conocidos también como test de rendimiento escolar, test de aprovechamiento, test de conocimiento, test de escolaridad. Ellos son pensados generalmente, como una muestra de indicadores del conocimiento de un alumno recolectados en un punto determinado en el tiempo.

- **Ítem de selección múltiple:** Es un enunciado, representado por una frase o por una pregunta, seguida de alternativas, entre las cuales se debe escoger una. Las alternativas pueden ser representadas por palabras, frases o dibujos. Algunas consideraciones básicas al elaborar un ítem de selección múltiple:
 - El ítem debe medir un aspecto importante del aprendizaje.
 - El enunciado debe ser formulado de manera clara y directa.
 - Se debe evitar alternativas con las opciones “ninguna respuesta anterior”, “todas las respuestas anteriores”. Se debe escoger alternativas, de modo que una de ellas sea la verdadera, las otras alternativas deben ser plausibles y atractivos y nunca totalmente absurdas.
 - Las alternativas deben ser gramaticalmente coherentes con el enunciado.
 - Se debe evitar indicios que permitan al alumno adivinar cuál es la respuesta correcta.
 - Siempre que sea posible, se debe evitar la formulación de los enunciados con frases negativas. Si esto ocurre, se debe señalar las palabras negativas, escribiéndolas con negritas o en letra grande.
 - Siempre que sea posible se deben colocar cinco alternativas.
 - Colocar, en el enunciado, la parte de la redacción que se repite en las alternativas.

- **Ítem de correspondencia o asociación de columnas:** Este ítem presenta dos columnas para que sean relacionadas. Una columna presenta las preguntas y la otra, las respuestas. Los cuidados para la elaboración de estos ítems son:
 - Dar al alumno una orientación clara de cómo debe hacer la correspondencia.
 - Cuidar para que la lista de preguntas, así como también el contenido de la lista de respuestas tenga un contenido homogéneo.
 - Usar más respuestas que preguntas.
 - Colocar las columnas de preguntas y respuestas lado a lado, de manera que el alumno pueda economizar tiempo en la respuesta.
 - No se puede colocar más que una respuesta correcta para cada pregunta.

- **Ítem de falso-verdadero/correcto-equivocado:** Este ítem presenta afirmaciones que el alumno señala como falsas o verdaderas. Precauciones en la formulación de estos ítems:
 - Elaborar las afirmaciones de modo que ellas sean completamente correctas o incorrectas, evitando provocar dudas en el alumno.
 - Evitar el uso de palabras tales como: “ninguna”, “todo”, “nunca”, “a veces”, “generalmente”.
 - Evitar el uso de palabras negativas.

- **Ítem de completar:** En este ítem el alumno debe escribir en los espacios vacíos, completando la frase presentada. Las principales precauciones al elaborar un ítem de completar son:
 - Formular el ítem de manera que el alumno no tenga dudas sobre aquello que debe hacer.
 - Colocar el espacio vacío, siempre que sea posible al final de la frase que debe ser completada. Esto evitará que el alumno tenga que volver al inicio de la frase para escribir lo que le es solicitado.
 - Colocar un máximo de dos espacios vacíos para completar. Un ítem con muchos espacios es más un rompecabezas que un ítem de prueba.
 - Evitar pistas, tales como colocar la primera letra de la palabra o dividir los espacios en el número de sílabas de la palabra.
 - Subrayar los espacios de modo que todos tengan el mismo tamaño, evitando así, que el alumno pueda adivinar.
 - En los casos de espacios que se refieren a medidas, es necesario indicar el grado de precisión esperado y las unidades de medidas, en las que deben ser expresadas.

- **Ítem de respuesta corta:** Es un ítem en el cual el alumno debe escribir una respuesta a una pregunta o a una orden formulada. Los cuidados para la elaboración de este ítem son:
 - Formular las preguntas de manera que los alumnos sepan lo que se les pide.
 - Si el alumno debe citar nombres, se debe especificar cuántos son.
- **Ítems disertativos:** Es aquel que solicita del alumno una respuesta elaborada, esto es, una respuesta en la cual debe recurrir a raciocinios más complejos, tales como: establecer relaciones, analizar ventajas y desventajas, justificar la razón de lo que se escogió y evaluar las mejores alternativas para los problemas planteados. Los cuidados para la elaboración de ítems disertativos son:
 - Escribir el ítem de manera que el alumno no tenga dudas de lo que le es solicitado.
 - Estructurar el ítem de manera que el alumno sepa el nivel del raciocinio deseado.
 - Evitar expresiones ambiguas como: “¿Qué piensa usted de este tema?”, “Escriba todo lo que sepa de este tema”.
 - Elaborar, paralelamente al ítem, la respuesta deseada, para que se pueda percibir lo que realmente se desea del alumno.
- **Seguimiento de Actividades**

Su objetivo es permitir el rastreo de tareas inmersas a actividades generales, que a su vez pueden servir de apoyo a otras técnicas como la observación, la entrevista, el proyecto o las prácticas de laboratorio.

Esta técnica utiliza varios instrumentos entre los que se encuentran: las bitácoras, los registros de actividades, las autoevaluaciones, el anecdotario y las coevaluaciones.

- **Actividades Complementarias**

Es considerada más que una técnica una categoría donde se agrupan instrumentos que permiten reafirmar, contrastar o validar los juicios emitidos bajo las técnicas de evaluación u otros instrumentos utilizados, es por ello que se les nomina como actividades complementarias.

Algunos de los instrumentos que conforman este grupo son: relatorías, resumen, ejercicios, talleres, etc.

➤ Instrumentos de Evaluación

Los medios empleados para la recolección de datos o informaciones en el proceso de evaluación se conocen como los instrumentos, convirtiéndose en el soporte teórico de la evaluación y el soporte empírico del proceso de evaluación. Los instrumentos de evaluación pueden asociarse a diferentes técnicas de evaluación, y entre estos se encuentran:

- **Prueba Operatoria**

La prueba operatoria es un instrumento de evaluación que tiene la finalidad de verificar la habilidad del alumno para operar con los contenidos aprendidos. La palabra operatoria viene de operación que significa acción elaborada y compleja, como por ejemplo, analizar, clasificar, comparar, criticar, generalizar y plantear hipótesis.

La prueba operatoria tiene el mérito de haber terminado con las maneras clásicas de evaluar sobre todo con relación a los conceptos de correcto e incorrecto. La intención de esa prueba es orientar paso a paso el aprendizaje del alumno, dejando claros los objetivos de las preguntas, las cuales no son planteadas de manera aislada o fragmentada. Las preguntas están formuladas de manera que propician en el alumno el abandono de la simple memorización de las informaciones a cambio de establecer relaciones con hechos, fenómenos e ideas para hacerle percibir que nada ocurre aisladamente.

- **Análisis y Solución de Casos**

Otra técnica interesante es el del análisis y solución de casos. Los casos son desencadenadores de un proceso de pensar, estimuladores de la duda, del levantamiento de hipótesis, de la comprobación de las mismas, de la inferencia y del pensamiento divergente. Los usos de tareas auténticas derivadas de casos reales son esencialmente significativos por ser verdaderos, lo que hace que para las personas tengan una mayor credibilidad y significado.

Para utilizar la evaluación empleando casos es necesario que se produzca un ambiente constructivista de aprendizaje. En este ambiente, son consideradas las múltiples perspectivas, las diversas interpretaciones de la realidad y la construcción del conocimiento con base en experiencias significativas. Este método contribuye al desarrollo intelectual del alumno en la medida en que aprender no es copiar o reproducir punto por punto la realidad. Según la concepción constructivista, se aprende cuando se es capaz de elaborar una representación de un objeto de la realidad o contenido que se intenta aprender.

Esa aproximación no es vacía, no parte de la nada, pues se basa en la experiencia, en el interés y en los conocimientos previos que puedan dar cuenta de la novedad. Por otro lado, construir no es inventar. Se debe, por ejemplo, motivar al alumno a usar la ortografía de manera creativa y poco convencional, pero esa creación debe estar culturalmente ligada a lo establecido, de forma que la comunicación y la comprensión ocurran.

- **Mapas Conceptuales**

El mapa conceptual es una técnica de evaluación propia del enfoque constructivista en el cual, el aprendizaje se expresa como un proceso fundamentalmente interno³¹. Los criterios de evaluación, por lo tanto, no pueden limitarse solamente a los comportamientos observables.

La finalidad principal de un mapa conceptual es analizar los procesos de pensamiento de los alumnos. Los mapas son indicadores del grado de diferenciación que una persona establece entre los conceptos.

Por ese tipo de mapa, se puede ver claramente si un estudiante ha conseguido comprender las relaciones conceptuales y si ha captado los significados básicos enseñados. Es un modo de conseguir que los alumnos piensen en relaciones que antes no habían observado.

Con los mapas conceptuales se puede verificar, por ejemplo, si un alumno es capaz de distinguir entre conceptos generales y específicos sobre un tema ya que la ordenación jerárquica entregada a través del mapa, representa el conjunto de relaciones entre un concepto y sus conceptos subordinados.

El mapa parte del principio de que hay una evolución en el aprendizaje, cuando el que aprende reconoce nuevas relaciones o vínculos conceptuales entre conjuntos de conceptos o proposiciones. En este sentido, el aprendizaje se torna significativo.

- **Portafolio**

El portafolio es un instrumento que permite la compilación de todos los trabajos realizados por los estudiantes durante un curso o disciplina. En él pueden ser agrupados datos de visitas técnicas, resúmenes de textos, proyectos, informes, anotaciones diversas. El portafolio incluye también, las pruebas y las autoevaluaciones de los alumnos. La finalidad de este instrumento es auxiliar al

³¹ ONTORIA, A. et al. Mapas conceptuales - una técnica para aprender. 7ed, Madrid, Nancea, 1997. Referenciado en los resúmenes de la Biblioteca de Documentos, Chilecalifica. [En línea]. Disponible en: <http://www.chilecalifica.cl/califica/showCuerpo.do?cuerpo=309&barra=1&pagina=bajada-cuerpo-documento.jsp>

estudiante a desarrollar la capacidad de evaluar su propio trabajo, reflexionando sobre él, mejorando su producto. El portafolio le ofrece al profesor la oportunidad de obtener referencias de la clase como un todo, a partir de los análisis individuales, con foco en la evolución de los alumnos a lo largo del proceso de la enseñanza y del aprendizaje.

Los portafolios no involucran sólo la compilación de trabajos. Como instrumento de motivación del razonamiento reflexivo, propician oportunidades para documentar, registrar y estructurar los procedimientos y el propio aprendizaje. Es por esa reflexión que el estudiante puede, con ayuda del profesor, verificar lo que necesita mejorar en sus desempeños. Por otro lado, el portafolio permite al profesor conocer mejor a su alumno, sus ideas, sus expectativas, su concepción de mundo.

- **Proyectos**

El proyecto es un instrumento útil para evaluar el aprendizaje de los alumnos, a la vez que permite verificar las capacidades de:

- Representar objetivos que deben ser alcanzados.
- Caracterizar propiedades de lo que será trabajado.
- Anticipar resultados intermedios y finales.
- Escoger estrategias más adecuadas para la solución de un problema.
- Ejecutar las acciones para alcanzar procesos y resultados específicos.
- Evaluar condiciones para la solución del problema.
- Seguir criterios preestablecidos.

El proyecto puede ser propuesto individualmente y/o en equipo. En los proyectos en equipo, además de las capacidades ya descritas, se puede verificar, por ejemplo, la presencia de algunas actitudes tales como: respeto, capacidad de oír, tomar decisiones en conjunto y solidaridad.

2.1.3. METODOLOGÍAS DE DISEÑO INSTRUCCIONAL PARA PROGRAMAS DE FORMACIÓN POR COMPETENCIAS

❖ MODELOS DE DISEÑO INSTRUCCIONAL

Los modelos instruccionales son guías o estrategias basadas en enfoques de aprendizaje, que los instructores utilizan en el proceso de enseñanza y aprendizaje. Constituyen el armazón procesal sobre el cual se produce la instrucción de forma sistemática y fundamentada en teorías del aprendizaje. Incorporan los elementos fundamentales del proceso de diseño instruccional, que

incluye el análisis de los participantes, la ratificación de metas y objetivos, el diseño e implantación de estrategias y la evaluación³².

Los modelos instruccionales están prescritos por los siguientes componentes:

- El uso de la práctica.
- El uso de múltiples ejemplos.
- Variación en la retroalimentación.
- Estrategias secuenciales.
- El uso de organizadores avanzados.

Los modelos de Diseño Instruccional proveen los procedimientos para la producción de instrucciones y ayudan al fortalecimiento de la instrucción. Estos modelos incorporan elementos fundamentales en el proceso de diseño instruccional, incluyendo el análisis de las metas y objetivos. Los modelos pueden ser usados en diferentes contextos. Un modelo puede ser usado para un curso completo de instrucciones o se pueden combinar múltiples conceptos.

➤ Modelos Curriculares

La teoría curricular, como conjunto de principios teóricos de carácter político, filosófico, psicológico y pedagógico que guían y dan lugar a diferentes proyectos curriculares experimenta un impetuoso desarrollo a partir de la década de los 60.

Se producen numerosos trabajos y surgen los “*teóricos del currículo*” que desde diferentes presupuestos teóricos y metodológicos abordan el campo de lo curricular³³. El propósito principal de esta teoría del currículo es fundamentar la práctica por medio de un cuerpo de ideas coherentes y sistemáticas que permitan dar significado a los fenómenos y problemas curriculares así como guiar en la toma de decisiones de las acciones más apropiadas y justificadas.

El diseño curricular, como proceso de elaboración de un currículo de estudio debe estar fundamentado en determinados presupuestos teóricos y metodológicos, sin embargo, en la actualidad se observa un desfase entre el nivel de elaboración conceptual desarrollado en el campo teórico del currículo y las características de los modelos de diseño curricular propuestos.

A continuación se definen las principales aproximaciones metodológicas del diseño curricular vinculadas con distintos enfoques teórico-conceptuales del currículo. Se exponen propuestas metodológicas que se sustentan en

³² HURST, Geoconda. Módulo Taller de Metodología de Diseño Curricular. Programa de Capacitación Modular (PCM) para Docentes de Formación Profesional de la Enseñanza Media de Honduras, 2005.

³³ RUÍZ, José María. Teoría del curriculum: Diseño y desarrollo curricular. Editorial Universitat. S.A. España, 1996. En [SANZ. 2004]

concepciones tecnológicas y otras que pueden considerarse como metodologías cuyo enfoque tiene un carácter sociopolítico o crítico.

➤ **Modelos Curriculares Tecnológicos**³⁴

El enfoque que ha caracterizado y determinado la mayoría de los modelos funcionales de diseño curricular que han surgido desde principios del siglo XX hasta nuestros días ha sido el tecnológico.

✓ Propuesta de Ralph Tyler

Entre estos modelos se encuentra el modelo de Tyler, el cual se sustenta en la tecnología de la educación entendida como el desarrollo de un conjunto de técnicas sistemáticas, junto con conocimientos prácticos, para el diseño, la validación y la operación de las escuelas como sistemas educativos.³⁵

Las 4 preguntas básicas que se utilizan como guía para elaborar un currículo basado en el modelo de Tyler son:

1. ¿Qué fines desea alcanzar la escuela?
2. De todas las experiencias educativas que pueden brindarse. ¿Cuáles ofrecen mayores posibilidades de alcanzar estos fines?
3. ¿Cómo se puede organizar de manera eficaz esas experiencias?
4. ¿Cómo podemos comprobar si se han alcanzado los objetivos propuestos?

De las anteriores preguntas se desprende la importancia de la delimitación de metas y objetivos educativos. Las bases referenciales para establecer estos objetivos deben surgir del análisis de investigaciones sobre los alumnos y sus necesidades, la sociedad, el análisis de tareas y los procesos culturales, y sobre la función y el desarrollo de los contenidos. A los resultados de estas investigaciones se les denomina fuentes

Este modelo incluye los siguientes pasos:

- Selección y organización de objetivos a partir de las propuestas provenientes de las distintas fuentes y filtradas por el análisis filosófico y psicológico.
- Definición de objetivos conductuales.
- Selección y organización de actividades de aprendizaje.
- Evaluación de experiencias.

³⁴ Fuente [http://www.wikilearning.com/tyler_e_hilda_taba_modelo_racional_normativo_i-wkccp-10358-11.htm y www.cepes.uh.cu/bibliomaestria/teoriaydisenocurricular/MODELOSDEOBJETIVOS.doc]

³⁵ DÍAZ BARRIGA, Ángel. Didáctica y Currículum, Buenos Aires, Paidós, 1997. En [SANZ. 2004]

✓ Propuesta de Hilda Taba

La propuesta de H. Taba constituye una continuidad y un avance en relación a la de Tyler. La misma, parte de la investigación de las demandas y requisitos de la cultura y la sociedad que permitirán determinar los principales objetivos de la educación, los contenidos y las actividades de aprendizaje.

Taba introduce en el campo del diseño curricular la noción de diagnóstico de necesidades sociales como sustento principal de una propuesta curricular. De esta manera especifica el vínculo entre institución educativa y sociedad.

En su modelo diferencia dos planos: el primero vinculado con las bases para la elaboración del currículo y el segundo con los elementos o fases a considerar en la elaboración y desarrollo curricular.

En el primer plano plantea la fundamentación científica del currículum a partir de las aportaciones de las disciplinas básicas sobre:

- Las funciones de la escuela en la sociedad y la cultura.
- Los procesos de desarrollo y aprendizaje en el alumno.
- La naturaleza del conocimiento.

El segundo plano está referido a objetivos, contenidos y experiencias de aprendizaje así como a los sistemas de evaluación. La ordenación de todos estos elementos se traduce en las siguientes fases:

- Diagnóstico de necesidades.
- Formulación de objetivos.
- Selección del contenido.
- Organización del contenido.
- Selección de actividades de aprendizaje.
- Organización de actividades de aprendizaje.
- Sistema de evaluación.

Es conveniente precisar que la noción de diagnóstico de necesidades introducida por Taba, con el tiempo dio lugar a un reduccionismo en la tarea de elaboración de los diseños curriculares debido a que se ha considerado cualquier investigación aislada, fragmentada y no el análisis de un conjunto de investigaciones para fundamentar el currículum como lo concibió Taba.

Estas propuestas de Tyler y Taba intentan desde posiciones conductistas trascender los límites de los aspectos técnicos de la formulación de objetivos y dar una mayor apertura (con orientación funcionalista) a una serie de elementos que consideran determinantes en la elaboración de los programas.

Estas propuestas curriculares clásicas hicieron aportes importantes a la práctica del currículum como el tratar de lograr una mayor coherencia e integración de los elementos curriculares, el buscar una fundamentación teórica que permita superar las prácticas empíricas y poco sistemáticas del trabajo curricular.

Sin embargo su visión reduccionista del fenómeno educativo no propicia un análisis integral que tome en cuenta todos los elementos que lo determinan y las interrelaciones existentes entre los mismos.

A partir de estos proyectos curriculares clásicos se desarrollan otras propuestas, que aunque difieren en algunas de sus concepciones, tienen en común el vertebrarse en torno a objetivos conductuales. Entre los representantes de este modelo centrado en los objetivos se destacan las figuras de M. Jhonson, R. Mager, Popham, Baker, B. S. Bloom entre otros.

El esquema básico de estos modelos tecnológicos puede presentarse en los siguientes pasos³⁶:

1. Definición de objetivos de institución en términos de comportamiento.
2. Identificación de contenidos apropiados para conseguir los fines propuestos.
3. Identificación de materiales-medios de instrucciones específicas.
4. Desarrollo de actividades de instrucción.
5. Comprobación (medición del logro de los objetivos pretendidos o resultados).

✓ Propuesta de Bloom y Mager.

En esta propuesta se privilegia la técnica de objetivos cognitivo – conductuales en la elaboración de programas educativos. Los objetivos se redactan en términos referidos al estudiante, estableciendo las condiciones y criterios que se considera muestran un cumplimiento aceptable de la conducta deseada.

La planificación de los cursos es cerrada generando rigidez y esquematismo, convirtiéndose en uno de los puntos críticos, junto con las conductas memorísticas y la superficialidad de los conocimientos impartidos.

Es innegable que toda actividad humana se orienta hacia la consecución de determinadas finalidades u objetivos que desean alcanzarse, pero estos han de interpretarse como guías orientadoras del proceso por lo que deben ser flexibles, ajustables, adecuados al contexto.

³⁶ ÁLVAREZ, J.M. Dos perspectivas contrapuestas sobre el currículum y su desarrollo. Revista de Educación No. 282, enero-abril, 1987. En: [SANZ. 2004]

Igualmente la evaluación de los resultados curriculares no es posible “medirlos” únicamente por el solo logro de los objetivos previamente concebidos en término de conductas externas observables sino que estos resultados son producto de complejos y ricos procesos donde participan también seres humanos lo que hace difícil su correspondencia lineal con lo previsto inicialmente.

➤ **Modelos Sociopolíticos ó Críticos**

Con este nombre se agrupan un conjunto de propuestas de diversos orígenes que tienen como denominador común el rechazo a los modelos tecnológicos del currículum y el enfatizar los aspectos sociales, políticos e ideológicos que están presentes en todo proyecto curricular. Se caracterizan por enfatizar los vínculos existentes entre institución educativa y desarrollo social, por ser altamente flexibles, contextualizados e incorporar en sus diseños curriculares, en mayor o menor medida, elementos de interdisciplinariedad y globalización.

➤ **Sistema Modular**

Sus raíces pueden rastrearse en la combinación de diversas influencias teóricas como la escuela nueva, pragmatismo, antiautoritarismo, psicoanálisis, tecnología educativa y psicología cognitiva así como las características de las condiciones concretas existentes en la época de su surgimiento y de las diversas tendencias políticas que en ella se manifestaban.

Surgió como una alternativa de enseñanza que rompe con el paradigma clásico de organización del conocimiento por disciplinas, nace en 1974 el Sistema Modular en la Universidad Autónoma Metropolitana-Unidad Xochimilco en México ³⁷

En el plano pedagógico retoma los planteamientos de la escuela nueva que concibe al alumno como el centro del proceso de enseñanza-aprendizaje y lo sitúa en una posición activa frente al aprendizaje. El conocimiento se organiza de forma globalizada y estrechamente vinculado con la realidad siendo el papel del docente el de facilitador del aprendizaje de los mismos.

Sus bases psicológicas podemos hallarlas en la teoría de J. Piaget donde se destaca la importancia de la actividad del sujeto cognoscente con relación al objeto de conocimiento y como se van transformando las estructuras mentales del sujeto en su interacción con el objeto.

Las principales características del Sistema Modular Xochimilco son:

- Vinculación de la educación con los problemas de la realidad, posibilitando a la universidad una mejor formación de los futuros profesionales que les

³⁷ ARBESÚ, M. El sistema modular Xochimilco. En “El Sistema Modular en la Unidad Xochimilco de la Universidad Autónoma Metropolitana” México, 1996. En [SANZ. 2004]

permita enfrentar y solucionar exitosamente los problemas que se le presentan en su vida laboral y social.

- Integración de la docencia, la investigación y el servicio, las tres tareas sustantivas de la universidad.
- Organización global del proceso de enseñanza-aprendizaje, por medio de módulos en torno a problemas de la realidad cuyo estudio se aborda de manera interdisciplinaria.
- Participación activa de los estudiantes en su formación propiciando el desarrollo de la responsabilidad individual ante su propio proceso de aprendizaje.
- Concepción de la función del profesor universitario como guía, orientador y organizador del proceso pedagógico.
- Utilización de métodos activos de enseñanza fundamentalmente el trabajo en equipos.

A partir de estas características se organiza el Sistema Modular centrado en la definición de las prácticas profesionales emergentes y dominantes de cada profesión como elemento articulador del diseño curricular.

El plan de estudios se conforma por un conjunto de módulos estructurados a partir de un objeto de transformación que se aborda a través de un proceso investigativo y de trabajo grupal.

El módulo constituye una estructura de enseñanza-aprendizaje compuesta por varias unidades interdisciplinarias. Cada una de estas unidades tiene un objetivo de proceso que debe alcanzarse realizando diferentes actividades durante el periodo de duración de cada módulo.

El objeto de transformación, noción central en este modelo, constituye un problema de la realidad que es considerado por la universidad como pertinente para ser incluido en el proceso de enseñanza-aprendizaje. Durante el desarrollo de cada módulo, alumnos y docentes desarrollan una investigación científica en torno al objeto de transformación (problema) que se pretende estudiar el cual es abordado de forma interdisciplinaria.

El plan de estudio de cada carrera por lo general está formado por módulos que se distribuyen a lo largo de la carrera en troncos a partir del criterio de que los iniciales se relacionen con un mayor número de disciplinas y carreras y los terminales sean más específicos. Cada módulo tiene determinados objetivos particulares y el conjunto de ellos se encamina a lograr los objetivos generales de la carrera.

Un análisis crítico del Sistema Modular, lleva a afirmar que el mismo constituye una innovación educativa importante a nivel de la educación superior que ha

permanecido en el tiempo y ha posibilitado la formación de muy buenos profesionales. Sin embargo, como señalan sus propios protagonistas, es necesario continuar desarrollando sus bases teóricas y metodológicas de manera que se pueda identificar y superar las fallas que presenta.

El romper con la estructura lógica de la ciencia al organizar todos los contenidos y asimilarlos de forma interdisciplinaria necesita ser analizado más a fondo, pues hay conocimientos que son básicos para la formación del profesional, que tienen su propia lógica, y que no pueden incorporarse en un problema concreto de investigación y en la organización del módulo.

A su vez la organización global del conocimiento en módulos, si bien tiene la gran ventaja de vincular al estudiante con los problemas reales del contexto en que vive, tiene el peligro de llevar a un pragmatismo extremo si no se hace una adecuada selección de estos problemas que puedan garantizar la formación científica del profesional.

Otro reto importante del Sistema Modular está en la formación científica y pedagógica que requieren sus profesores para poder desarrollar una enseñanza interdisciplinaria a través de un proceso de investigación científica y donde el rol del profesor es el de orientador del proceso de aprendizaje de sus estudiantes que implica nuevas formas de relación del trabajo que posibiliten el desarrollo de las potencialidades de cada uno de sus alumnos.

➤ **Modelo de Investigación en la Acción**

El término “investigación en la acción” propuesto por K. Lewin a finales de la década de los 30’ ha tenido una amplia aceptación en el campo de las investigaciones sociales y educativas. La Deaking University de Victoria en Australia lo define como “término utilizado para denominar a un conjunto de actividades del desarrollo profesional, de los proyectos de mejora escolar y de la práctica y planificación educativa. Éstas tienen en común la utilización de estrategias de planificación de la acción, llevadas a la práctica y sometidas a observación, reflexión y cambio. Los participantes de esta acción están plenamente integrados e implicados en todas las actividades “. ³⁸

Nació como otra alternativa de superar las concepciones teórico-metodológicas imperantes en los enfoques curriculares tecnológicos surgiendo el modelo de investigación en la acción.

Las características de este tipo de investigación son:

³⁸ SÁEZ, M., ELLIOT, J. La investigación en Acción en España: Un Proceso que Empieza. Revista de Educación. Madrid, 1988. En [SANZ. 2004]

- El problema nace de la comunidad que lo define, analiza y resuelve.
- Su fin último es la transformación de la realidad social y el mejoramiento de la vida de los involucrados.
- Exige la participación plena de la comunidad durante toda la investigación.
- El investigador es un participante comprometido que aprende durante la investigación.

Se reconoce a L. Stenhouse como uno de sus principales representantes quien introduce el término en la teoría curricular como Modelo de Investigación en la Acción. El currículo en esta concepción es considerado como un proyecto en ejecución que se verifica en la acción del aula. Intenta superar el divorcio existente entre quienes programan (diseñadores, directivos), los que desarrollan el programa (profesores) y los destinatarios (alumnos), siendo los profesores, como profesionales de la enseñanza, los que elaboran el plan y lo llevan a la práctica de una manera flexible, sujeta a replanteamiento y ajuste.

Otra peculiaridad de este modelo curricular es la no existencia de distancias entre los momentos de elaboración, desarrollo curricular y evaluación, ya que es en su propio proceso de desarrollo que se va construyendo y reconstruyendo el currículum a partir de las reflexiones que surjan de su propia puesta en práctica.

Esto conlleva la necesidad de una sólida formación científica y psicopedagógica del profesor que posibilite realizar los ajustes necesarios a los problemas que se presentan en su desarrollo lo cual plantea el rescate de la profesionalidad del docente, de la preparación que éste debe tener en las distintas esferas que exige el desarrollo de su rol, del desarrollo de una enseñanza que sea fruto de la reflexión y la investigación en el aula, que posibiliten una participación activa del estudiante ante el objeto de conocimiento. Por lo que el profesor debe organizar y problematizar los contenidos de enseñanza, propiciar la reflexión, la actitud crítica ante los problemas, la libertad de expresión de diferentes puntos de vista así como estimular la investigación científica de los estudiantes.

Este proceso de enseñanza aprendizaje se conforma tomando en cuenta los pasos básicos de la investigación en la acción de manera que se estructure a partir de situaciones de interés para los participantes. Entre ellas destaca:

- La formulación de problemas por los propios estudiantes con la participación del profesor.
- La búsqueda de soluciones.
- La prueba de soluciones.

La participación conjunta de estudiantes y profesores posibilita el establecimiento de relaciones horizontales entre los mismos orientadas a la solución de problemas comunes.

Una peculiaridad de este modelo es el papel que se le otorga a la figura del profesor como profesional. Este posee autonomía suficiente para tomar las decisiones que considere oportunas e intervenir en el proceso de enseñanza-aprendizaje.

Este modelo ha tenido un impacto restringido en la práctica educativa actual, encontrándose pocas experiencias concretas de su aplicación. Es innegable sus ideas valiosas como son la utilización de la investigación científica como método para el organizar el proceso de aprendizaje de los estudiantes, el papel activo, reflexivo y responsable de estos en la asimilación de los contenidos, el propiciar una posición de compromiso del estudiante ante su realidad profesional y social, así como el énfasis en garantizar el desarrollo profesional del docente.

Sin embargo es preocupante la ausencia de un momento inicial de planificación teórica, de reflexión colectiva hacia donde orientar la formación del profesional y cómo hacerlo y que estas decisiones durante el desarrollo curricular puedan llevar a una enseñanza muy empírica y excesivamente contextualizada que no posibilite la formación básica y amplia que requiere el profesional.

Así mismo la necesidad de una sólida formación científica y pedagógica del profesor y de la responsabilidad que éste debe de asumir ante el aprendizaje de sus estudiantes constituye también en este modelo un reto complejo y difícil de lograr.

❖ PROPUESTA METODOLÓGICA DE DISEÑO INSTRUCCIONAL

➤ EL ANÁLISIS FUNCIONAL

El análisis funcional es una metodología de investigación que permite reconstruir - luego de desarrollar una serie de etapas - las competencias que debe reunir un/a trabajador/a para desempeñarse competentemente en un ámbito de trabajo determinado.³⁹

El análisis funcional es una metodología que busca la identificación o reconstrucción de competencias que un trabajador debe reunir para desempeñarse en un oficio específico, ahora esto se puede aplicar en el ámbito académico, buscando e identificando las competencias que debe reunir un estudiante en su proceso de formación para desempeñarse con éxito en campos específicos que se relacionan con su profesión.

³⁹ CATALANO, ANA M. AVOLIO DE COLS, Susana. SLADOGNA, MONICA G. Diseño curricular basado en normas de competencia laboral: conceptos y orientaciones metodológicas. 1º. ed. - Buenos Aires: Banco Interamericano de Desarrollo, 2004. 41.

El análisis funcional sigue una serie de principios y fases para cumplir el objetivo deseado el cual es, en el ámbito académico la creación de un diseño instruccional bajo la visión de competencias. Los principios y fases que se aplican en esta metodología se explicarán a lo largo de este capítulo.

➤ **PRINCIPIOS DE APLICACIÓN DEL ANÁLISIS FUNCIONAL**

Los principios rectores para la aplicación de la metodología del análisis funcional se concentran en tres sentencias específicas:

- **Ir de lo general a lo particular:**

El punto de arranque es el contexto de la asignatura (lo general) enmarcado por los contenidos temáticos básicos, genéricos y específicos, seleccionados a través del análisis de los contenidos presentes en literatura académica, empresarial e institucional concerniente, combinado a su vez con la experiencia y conocimientos de los expertos docentes, expertos pedagogos y expertos en la metodología de la planeación del diseño instruccional que acompañen el proceso. Este principio permite delimitar el área de estudio que se pretende abarcar con la asignatura junto con primera selección y estructuración los contenidos.

- **Identificar acciones delimitadas (discretas) manteniendo la separación de los contextos específicos:**

La desagregación de los contenidos generales debe ser única; poseer un inicio y un fin en su descripción, definiendo un propósito y un alcance preciso; además deben estar en consonancia con el área de estudio abarcada por la asignatura y por el programa de formación general. En la propuesta metodológica los contenidos desagregados se clasifican en tres tipos: “Contenidos Conceptuales (saber)”, “Contenidos Procedimentales (saber hacer)” y “Contenidos Actitudinales (saber ser)”, que corresponden a competencias evidenciables en el estudiante. Este principio metodológico se evidencia en la estructura gramatical de los contenidos desagregados que consta de: Verbo, Objeto y Condición, en el estricto orden en que se enuncian.

- **Mantener una relación causa-consecuencia:**

Este principio permite que los contenidos obtenidos de la desagregación sean realmente la suma de partes que den como resultado el contenido y/o propósito origen, o dicho de otra forma, el todo este realmente sustentado en los componentes que la conforman, además que tiene la utilidad de proveer la visión de correlación que debe establecerse entre las partes.

➤ **CARACTERÍSTICAS Y RECOMENDACIONES PARA LA APLICACIÓN DEL ANÁLISIS FUNCIONAL**

En la tabla 1 se presentan en compendio las características fundamentales junto con las recomendaciones propias del análisis funcional que corresponden al desarrollo y aplicación de la metodología para el diseño instruccional.

Tabla 9. Características y recomendaciones del análisis funcional para su aplicación en procesos de formación académica.⁴⁰

De lo general a lo particular	Partir de los contenidos generales	Delimitar mediante el análisis y establecimiento de los contenidos el área de estudio de la asignatura.
	Mantener la relación causa – consecuencia	Los contenidos desglosados y clasificados en conceptuales, procedimentales y actitudinales deben en conjunto proveer las herramientas para el cumplimiento de los propósitos y actividades de la asignatura.
	Desglosar hasta lograr los contenidos de realización individual	El proceso de desglose o desagregación del contenido concluye cuando se identifican y enuncian competencias que puedan ser ejecutadas por un individuo y/o estudiante.
Enunciar contenidos discretos	Cada contenido tiene un comienzo y un fin, incluyendo en su descripción un alcance preciso	El enunciado del contenido permite delimitar el comienzo y final de la acción de dicho contenido y el resultado que pretende, proveyendo así las bases de las evidencias a recolectar para corroborar el aprendizaje.
	Los contenidos generales y/o desglosados aparecen solo una vez.	Los desgloses deben ser excluyentes entre sí. Si en el proceso de desagregación se repite algún contenido es necesario analizar si no corresponde realmente a un contenido más general de lo que se planteó inicialmente.
	Describir las acciones de aprendizaje del estudiante	En la identificación de los saberes deben establecerse las acciones de aprendizaje del estudiante que permitan la adquisición de las concepciones de la asignatura y la evaluación posterior de dichas acciones.
Utilizar una estructura gramatical uniforme	Los saberes y/o contenidos se enuncian bajo la estructura Verbo + Objeto + Condición	La normalización de la redacción permite mantener la consistencia en los enunciados y facilita la asociación y agrupamiento de los saberes y contenidos a lo largo del diseño instruccional.
	El verbo debe ser “activo”, con enfoque en la evaluación del estudiante	En lo posible debe usarse un solo verbo. El verbo es una acción real, medible y evaluable en términos de los resultados de aprendizaje que se buscan en el estudiante.
	El objeto es aquello sobre lo cual ocurre la acción de aprendizaje	El objeto especifica el contenido sobre el que se realizará el enfoque del verbo.
	La condición debe ser evaluable y	La condición debe estar directamente

⁴⁰ Fuente: Adaptación de los autores en base a GIRALDO P., Wilson.

	debe evitar el uso de calificativos y condiciones irreales	relacionada con el objeto, expresando parámetros o criterios contra los cuales se pueda comparar el resultado del aprendizaje. La condición define el alcance, la restricción y los límites para evaluar el aprendizaje del contenido. Se debe evitar incluir en la condición calificativos como: “adecuado”, “correcto”, “óptimo”, “completo”, “preciso”, etc., porque dificultan una evaluación objetiva.
Evitar el análisis excesivo de una palabra o frase	Tener dificultades en el manejo del lenguaje es una situación general en el desarrollo del análisis funcional. Evitar la discusión exhaustiva en palabras determinadas permite un mejor desarrollo metodológico.	
Evitar las discusiones pedagógicas y políticas	En la aplicación de la metodología es frecuente que se planteen discusiones sobre aspectos de diferentes índoles y que conciernen o tocan el proceso educativo. Es importante escuchar estas inquietudes y tenerlas en cuenta si lo ameritan, pero no debe dedicarse tiempo a discutir las sin sentido, ya que pueden alejar al equipo de desarrollo del camino metodológico.	

2.2. DISEÑO DE MATERIALES

2.2.1. Objeto de aprendizaje

Se entiende por objeto de aprendizaje cualquier recurso digital que pueda ser reutilizado para darle soporte a la educación. El término “Objeto de aprendizaje” generalmente se aplica a materiales educativos diseñados y creados en pequeñas unidades con el propósito de maximizar el número de situaciones de aprendizaje en las cuales pueden ser utilizados⁴¹. Estos son soportados en computadora con el objetivo de generar conocimientos, habilidades y actitudes en función de las necesidades del alumno”.

Un objeto de aprendizaje es una entidad digital con diseño instruccional que puede ser utilizado, reutilizado o referenciado durante el aprendizaje en línea con el objetivo de generar conocimientos, habilidades y actitudes en función de las necesidades del alumno.⁴²

Estandarizar y reutilizar los contenidos educativos es el propósito fundamental de la educación apoyada con tecnología. Desde este punto de vista las primeras definiciones de Objetos de Aprendizaje son bastantes amplias y se refieren a: cualquier recurso que pueda apoyar el proceso de aprendizaje mediado por alguna tecnología.

En este contexto a medida que las metodologías se fueron depurando y en la medida que tecnologías como la Internet empiezan a posibilitar el intercambio de

⁴¹ A. KOVALCHICK & K. DAWSON (Eds) *Educational Technology. An Enciclopedia*. Santa Bárbara: ABC-CLIO

⁴² PEÑA, CLARA INES. Metodología para la Ingeniería Instruccional aplicada al e-escen@riUIS.

información, surge la necesidad de precisar y depurar maneras estándares. Este esfuerzo ha permitido que los proveedores de diferentes tecnologías de e-learning (Learning Management System) vean en la estandarización la posibilidad de reutilizar contenidos para dar soporte a cursos sobre sus plataformas.

➤ **Características de un OA:**

- Interoperatividad: Capacidad de integración.
- Reusabilidad: Capacidad para combinarse dentro de nuevos cursos.
- Escalabilidad: Permite integración con estructuras mas complejas.
- Generatividad: Capacidad que permite generar otros objetos derivados de él.
- Gestión: Información concreta y correcta sobre contenido y posibilidades que ofrece.
- Interactividad: Capacidad de generar actividad y comunicación entre sujetos involucrados.
- Accesibilidad: Facilidad de acceso a contenidos apropiados en tiempos apropiados.
- Durabilidad: Vigencia de la información de los objetos, a fin de eliminar obsolescencia.
- Adaptabilidad: Característica de acoplarse a las necesidades de aprendizaje de cada individuo.
- Autocontención conceptual: Capacidad para auto explicarse y posibilitar experiencias de aprendizaje integral.

➤ **Funciones de un OA:**

- Favorecer la generación, integración y reutilización de contenidos temáticos.
- Estimular el estudio autogestivo.
- Promover el trabajo colaborativo.
- Posibilitar el acceso remoto a la información y contenidos de aprendizaje.
- Posibilitar la integración de diferentes elementos multimedia a través de una interfaz gráfica.
- Contribuir a la actualización permanente de profesores y alumnos.
- Estructuración de la información en formato hipertextual.
- Facilitar la interacción de diferentes niveles de usuarios. (Administrador, diseñador, alumno)

➤ **Componentes de un OA:**

- **Unidad de información:** Contenidos multimedia individuales (texto, imágenes, audio, etc.) en la que se tiene la posibilidad de generar contenido textual mediante el acceso a editores de texto.
- **Unidad de contenido:** Define la ubicación en la que se encuentran albergados los contenidos, facilitando la generación de plantillas.

- **Unidad didáctica:** Abarca cada uno de los elementos que permiten generar planteamientos de aprendizaje significativo, determinar criterios de evaluación, contenidos, recursos y actividades de enseñanza- aprendizaje.

Los objetos para ser incluidos además de las características señaladas, deben contener:

- Recomendaciones para conducir y orientar el trabajo
- Motivar el autoaprendizaje
- Información eficaz
- Desarrollar el análisis y la reflexión
- Mecanismos para aclaración de dudas
- Herramientas para la transferencia y aplicación de lo aprendido
- Herramientas para diálogo simulado
- Mecanismos de control y evaluación

➤ **Beneficios de un OA:**

Muchos son los beneficios que un Objeto de Aprendizaje aporta a la educación, siendo el más importante evitar "*reconstruir la rueda*" una y otra vez. Entre los beneficios podemos señalar:

- Para el usuario del Objeto de Aprendizaje:
 - Utilizar material educativo en forma independiente.
 - Contar con material educativo que puede organizar a sus necesidades.
 - Le da la opción de elegir cuál objeto de aprendizaje usar para su problema.
 - Le permite acceder a material educativo más didáctico y/o de prestigiosas instituciones educativas.
- Para el creador de Objetos de Aprendizaje:
 - Si ya existe el OA, le permite orientar su esfuerzo en mejorar el actual o bien crear un OA de otro tipo.
 - Le permite orientar su esfuerzo en la construcción de OA sobre temas puntuales. Le evita desarrollar o abarcar temas no relacionados con su área.
 - Le entrega derechos de uso claros (copyright).
 - Le permite distribuir o compartir su material con mayor facilidad.
- Para las plataformas de e-learning
 - Tienen acceso a material reutilizable.
 - Les permite acomodar los OA en función del tipo de curso que deseen ofrecer.
 - Les permite ahorrar esfuerzo (y dinero) en la construcción del material educativo.

➤ Estándares de uso de la tecnología

- Utilizan teclados y otros mecanismos comunes de entrada y salida en forma efectiva y eficiente.
- Utiliza el software y recursos en línea adecuados identificando ventajas y desventajas para consultar la información.
- Exhiben comportamientos legales y éticos cuando utilizan información-tecnología y discuten las consecuencias de usarlo inadecuadamente. Discuten sobre conceptos básicos relacionados con el uso responsable de la tecnología y la información, y describen las consecuencias personales de usos no adecuados.
- Utilizan herramientas de uso general y periféricos para apoyar la productividad personal, remediar déficit de habilidades y facilitar el aprendizaje en todo el plan de estudios.
- Utilizan herramientas tecnológicas (por ejemplo: presentaciones multimedia, herramientas de Internet, cámaras digitales, scanner o digitalizadores) para actividades individuales y colaborativas de escritura, comunicación y publicación que les permita crear productos de aprendizaje para audiencias internas y externas al salón de clases.
- Diseñan, desarrollan, publican y presentan productos (páginas Web, cintas de video) utilizando recursos tecnológicos que demuestran y comunican conceptos del currículum a público de adentro y fuera del salón.
- Utilizan las telecomunicaciones y otros recursos en línea (correo electrónico, discusiones en línea, ambientes del Web) para participar en actividades colaborativas de resolución de problemas con el propósito de dar soluciones o crear productos para público de adentro y fuera del salón.
- Utilizan recursos tecnológicos (calculadoras, instrumentos científicos, videos educativos, software educativo) para resolver problemas, aprender en forma autodirigida y realizar actividades de ampliación de aprendizaje.
- Determinan cuándo es útil la tecnología y seleccionan las herramientas apropiadas y los recursos tecnológicos necesarios para resolver una gran variedad de tareas y problemas.
- Evalúan la exactitud, relevancia, comprensión, conveniencia y sesgos de las fuentes de información electrónicas.

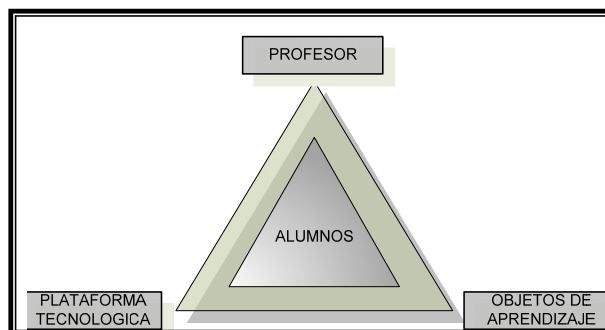


Figura 6. Interacción de componentes del proceso institucional

2.2.2. Aproa

Para el proyecto Aproa⁴³, se define como Objeto de Aprendizaje a: *"la mínima estructura independiente que contiene un objetivo, una actividad de aprendizaje, un metadato y un mecanismo de evaluación, el cual puede ser desarrollado con tecnologías de información y comunicación (TIC) con el fin de posibilitar su reutilización, interoperabilidad, accesibilidad y duración en el tiempo."* .

APROA es una iniciativa en el ámbito de la educación que propicia la adopción de tecnologías de Objetos de Aprendizaje, con el fin de crear una comunidad de desarrolladores y usuarios de objetos que por la vía de la colaboración y el intercambio de experiencias en el diseño de objetos, puedan sentar las bases de un programa de formación continua extensible a otras latitudes.

Aproa es un proyecto que nació en la universidad de Chile, buscando proporcionar herramientas que permitieran hacer más eficiente el proceso de desarrollo de aplicaciones de Tecnologías de Información y Comunicación (TIC) para la educación continua.

❖ **Objetivos del proyecto APROA**

➤ **Objetivo general**

Potenciar a las entidades interesadas en desarrollar aplicaciones TIC en la generación de productos y servicios en temáticas científicas, tecnológicas y productivas dirigidos a los distintos niveles de la educación técnica y superior del país, esto mediante la implementación de un sistema cooperativo, que facilite la relación entre una red de universidades y las empresas para el desarrollo de aplicaciones sobre la base del método de enseñanza por objetos de aprendizaje (OA).

➤ **Objetivos específicos**

- Generar una red de cooperación eficiente, entre los desarrolladores de contenidos para la educación técnica y superior, y las empresas desarrolladoras de aplicaciones TIC, e indirectamente de estos con los usuarios finales.
- Incorporar y fortalecer la utilización del método de enseñanza por OA en la educación a distancia, que posibilite la generación de aplicaciones TIC reutilizables e interoperables en las diversas áreas y niveles de la educación continua.

⁴³ APROA(aprendiendo con objetos de aprendizaje): *Metodología de desarrollo para diseño y producción de objetos de aprendizaje*

- Situar el desarrollo de aplicaciones TIC en un contexto de educación continua que haga más atractivo a los usuarios sus esfuerzos educativos por este medio.
- Estimular el crecimiento de un mercado, con amplitud hacia el extranjero, para las empresas desarrolladoras de aplicaciones TIC en distintos niveles y áreas de la educación continua.
- Nuclear y organizar el trabajo de los desarrolladores de contenido y de los desarrolladores de aplicaciones TIC a través de una plataforma basada en XML de amplio acceso y compatibilidad por parte de las instituciones asociadas.
- Iniciar una activa cooperación con países iberoamericanos en el desarrollo e integración de contenidos mediante OA (basados en XML), con quienes podrían articularse las plataformas locales para el intercambio de aplicaciones desarrolladas sobre la base de OA en la educación a distancia.

➤ **PARTICIPANTES**

- Universidad Arturo Prat (Chile)
- Universidad Tecnológica de Puebla (México)
- Centro de Agricultura y Medio Ambiente - AGRIMED - (Chile)
- Sun Microsystems Chile SA
- Newtonberg Publicaciones Digitales Ltda. (Chile)
- Universidad Nacional del Altiplano (Perú)
- Instituto Interamericano de Coop. p/ Agricultura - IICA
- Instituto Nacional de Astrofísica (México)
- Centro de Investigación Tecnol. de Información (México)
- Universidad Autónoma de Aguascalientes (México)
- Universidad Complutense de Madrid (España)
- Corporación Red Universitaria Nacional - REUNA (Chile)
- Universidad de Tarapacá (Chile)

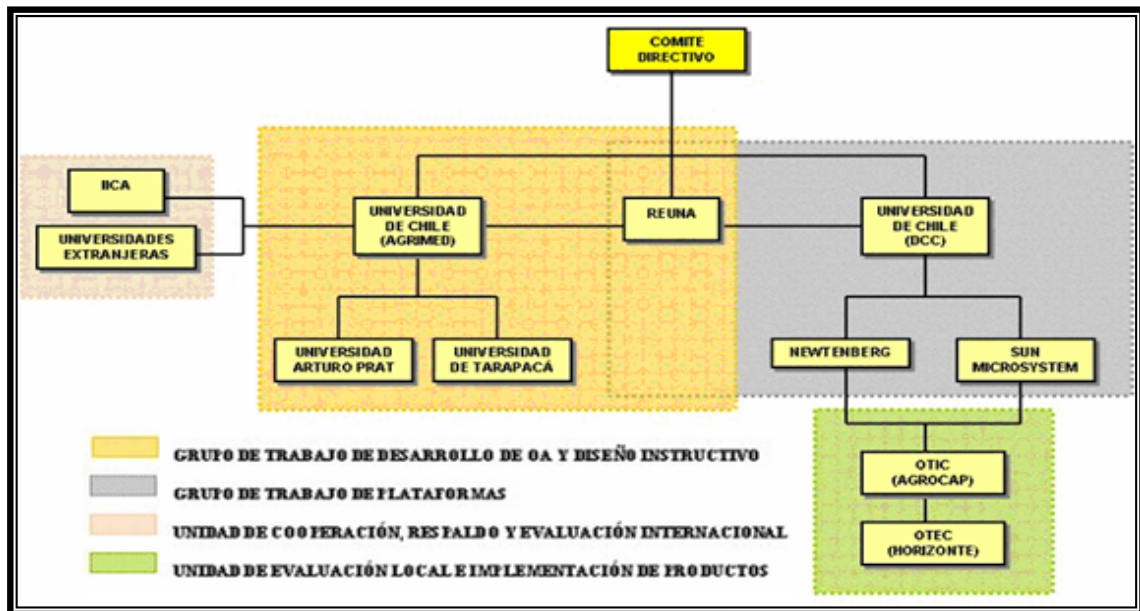


Figura 7. Organigrama del Sistema de APROA ⁴⁴

El proceso de estandarización para los objetos de aprendizaje ha llevado a la comunidad de desarrolladores a plegarse a uno de los primeros intentos de estandarización desarrollado en USA y ligado a lo que ahora se conoce como ADLScorm.

2.2.3. Estándar SCORM

SCORM es la sigla de Sharable Content Object Reference Model. Es un conjunto de especificaciones para desarrollo, empaquetamiento y distribución de material educativo en cualquier momento y en cualquier lugar. El estándar SCORM asegura que este material es: Reutilizable, Accesible, Interoperable y Durable.

Sin una especificación para empaquetamiento de los cursos on-line, las plataformas de e-Learning organizan sus bases de datos de maneras muy diversas. Aunque su formato de distribución de los contenidos sea Web, el manejo de los cursos entre distintas plataformas es una tarea complicada.

❖ Accesibilidad

Es la capacidad para localizar y acceder a componentes de aprendizaje situados en un punto remoto y suministrarlos a otros puntos.

El SCORM resuelve el requisito de accesibilidad proporcionando una manera

⁴⁴ Fuente [www.aproa.cñ/1116/propertyvalue-5538.html]

estándar de empaquetamiento de contenidos como objetos reutilizables y auto contenidos.

A cada objeto le está asociado un conjunto de meta-datos también estandarizados que describen en detalle los contenidos facilitando las búsquedas.

ADL está desarrollando una base de datos (repository) publicada en Internet, que contiene materiales de aprendizaje que se pueden fácilmente buscar, acceder y utilizar en la construcción de cursos.

De momento estos depósitos de información (repository) existen de forma experimental, pero el ADL-SCORM prevé su disponibilidad en un futuro.

De modo similar a los grandes buscadores de Internet (google, yahoo, etc.), se van a desarrollar nuevos motores de búsqueda para los materiales de aprendizaje.

➤ **Interoperabilidad**

Es la habilidad de poder utilizar en distintas plataformas componentes educativos creados con diferentes herramientas y desde cualquier ubicación.

SCORM responde a este requerimiento estandarizando la manera de comunicar entre los LMS y los objetos de contenido, especificando cómo se debe agregar y empaquetar el contenido. Crea un modo común de inicialización y finalización de lanzamiento en ejecución de los contenidos (a través de unos comandos especiales asociados) y también un vocabulario único para el intercambio de datos entre los objetos y el LMS (por ejemplo entre un test de evaluación de una lección y los resultados que tiene que monitorizar el LMS).

➤ **Durabilidad**

Es la capacidad de un componente educativo de hacer frente a los cambios tecnológicos sin un rediseño, re-configuración o sobrescribir el código.

SCORM responde por la estandarización de las comunicaciones entre los LMS y los contenidos y especificando los detalles críticos acerca de la agregación y empaquetamiento de los contenidos.

➤ **Re-usabilidad**

Es la flexibilidad de incorporar componentes educativos en múltiples aplicaciones y contextos.

SCORM responde proporcionando un modo uniforme de empaquetamiento de contenidos, especificaciones para los metadatos que aportarán información detallada sobre contenidos facilitando su localización.

Una manera estándar de comunicación entre los contenidos y los LMS que permite la compatibilidad entre cursos producidos con distintas herramientas y plataformas de distintos autores.

❖ **Orígenes de SCORM**

SCORM es un producto de la iniciativa del gobierno EE.UU. llamada ADL (Advanced Distributed Learning), del noviembre de 1997. Esta iniciativa lanzada por el Departamento de Defensa viene a responder a la necesidad de crear accesos a los materiales educativos de alta calidad y alto grado de personalización.

• **Visión de ADL**

Es la visión de ADL una economía de aprendizaje basada en redes de comunicación que puedan proporcionar productos de aprendizaje personalizados donde y cuando se necesiten.

Estas redes serán cada vez más transparentes para los usuarios y más fáciles de utilizar. El elemento clave del desarrollo de este tipo de economía son unos depósitos de información en la red (repository), fácil de acceder y buscar.

• **Objetivos de ADL**

- Proporcionar acceso a la enseñanza y materiales educativos de alta calidad y alto grado de personalización
- Accesibilidad de los materiales educativos
- Acelerar el desarrollo del software educativo
- Crear un mercado sólido para los productos educativos.

• **La estrategia de ADL**

- Facilitar el desarrollo de estándares comunes para las tecnologías educativas.
- Hacer uso de las tecnologías emergentes basadas en red.
- Promover la colaboración basada en las necesidades comunes.
- Trabajar con la industria para influir en el desarrollo de los productos comerciales.

❖ Tecnologías concurrentes en SCORM

Durante muchos años, varias organizaciones estuvieron trabajando distintos aspectos del e-learning, desarrollando especificaciones para unir esfuerzos y apoyar la evolución de los productos y materiales educativos:

- AICC (Aviation Industry CBT Committee)
- ARIADNE (Alliance of Remote Instructional Authoring & Distribution Networks for Europe)
- IMS Global Learning Consortium
- IEEE/LTSC
- A través de IEEE, DCMI (Dublin Core Metadata Initiative)

❖ La Organización del SCORM

Naturalmente que Scorm ha pasado por una etapa de maduración. Como resultado de lo anterior, a la fecha han sido publicadas cuatro versiones⁴⁵:

1. **Scorm versión 1.0:** Disponible a partir de enero del 2000. Presenta ejemplos de implementación y el test de conformidad.
2. **Scorm versión 1.1:** Sale un año más tarde. Toca la fase de aplicación y realiza correcciones.
3. **Scorm versión 1.2:** Presenta las especificaciones de almacenamiento (Content Packaging) y realiza mejoras en la Meta data.
4. **Scorm 2004:** Introduce secuenciación y navegación.

Esta última versión se compone de cuatro libros:

1. Overview: Historia de la iniciativa ADL y Scorm.
2. Content Aggregation Model: Especifica como debe ser empaquetado el contenido para ser importado por el LMS. Describe las componentes para su búsqueda y descubrimiento.
3. Sequencing and Navigation: Describe como un contenido Scorm puede ser secuenciado.
4. Run-Time Environment: Especifica como el contenido (OA) debe comportarse una vez que éste ha sido activado por el LMS. Describe el protocolo de comunicación entre el LMS y OA.

⁴⁵ Fuente [\[www.aproa.cl\]](http://www.aproa.cl)

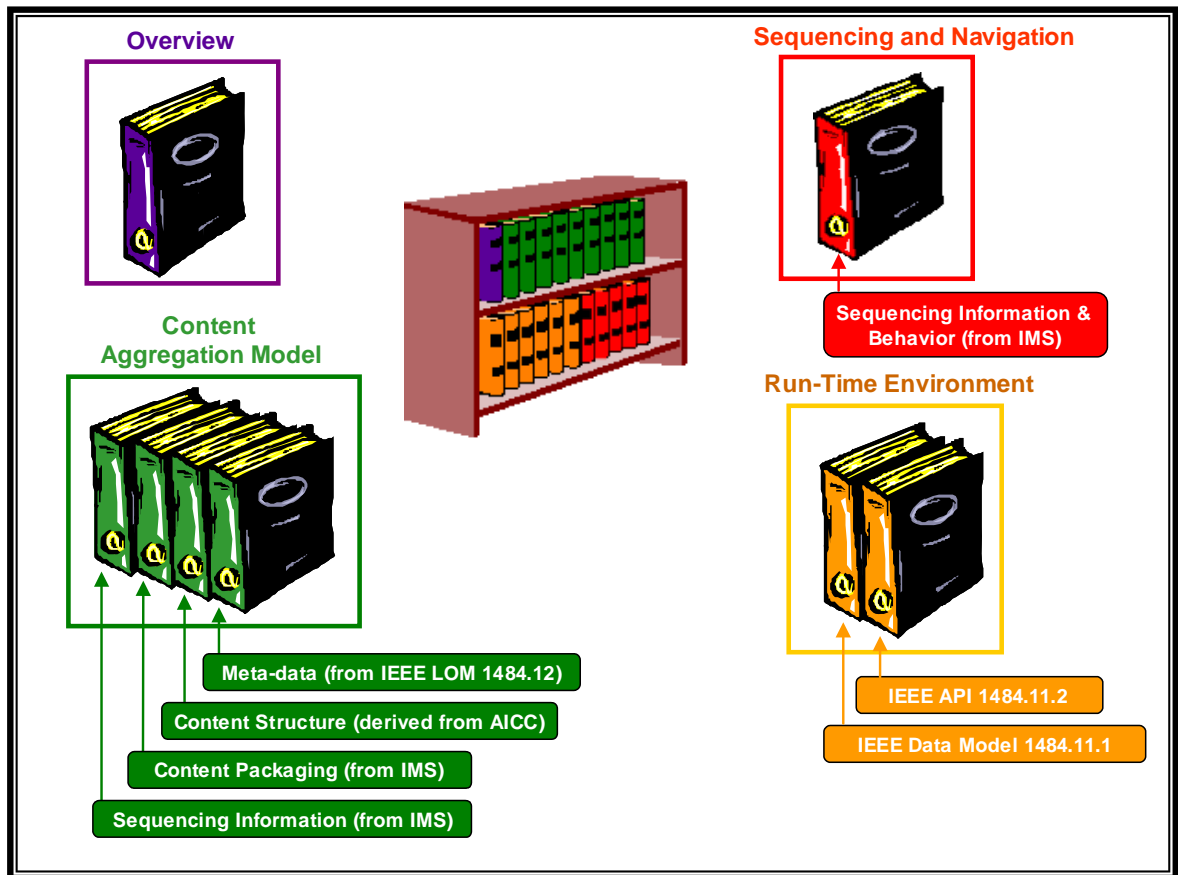


Figura 8. Visión general de las especificaciones de SCORM⁴⁶

SCORM es una colección de especificaciones y estándares que quedan recogidos en varios libros técnicos. La mayoría de las especificaciones se han tomado de otras organizaciones como IMS, AICC, ARIADNE e IEEE.⁴⁷

- **SCORM 2004 Overview**

Este libro cubre la historia y los objetivos de ADL, proporcionando información a un alto nivel conceptual sobre SCORM y las especificaciones de las que parte. En él se introduce la terminología de SCORM y de los elementos que componen su propuesta. También describe brevemente las áreas de los otros tres libros (CAM, RTE y SN), mostrando cómo se relacionan una con otras.

⁴⁶ Advanced Distributed Learning (ADL), Sharable Content Object Reference Model (SCORM®) 2004 2nd Edition Overview, 2004.

⁴⁷ Miguel Rebollo Pedruelo, El estándar SCORM para EaD, Tesina del Máster en Enseñanza y Aprendizaje abiertos y a Distancia Universidad Nacional de Educación a Distancia Diciembre 2004

• SCORM 2004 Content Aggregation Model

Contiene una guía para identificar y agregar recursos dentro de un contenido de aprendizaje estructurado. Este libro describe una nomenclatura para el contenido de aprendizaje, denominado *SCORM Content Packaging*, basado en las especificaciones de LOM de la IEEE y en el *IMS Learning Resource Meta-data Information Model*. El objetivo del modelo de agregación de contenidos de SCORM es proporcionar un medio común para construir contenidos educativos desde diversas fuentes compatibles y reutilizables.

Define cómo un contenido educativo puede ser identificado, descrito y agregado dentro de un curso o una parte de un curso, y cómo puede ser compartido por varias plataformas y repositorios. Un paquete de contenidos agrupa una serie de objetos cuya organización se describe en un *manifiesto*. Un paquete puede representar un curso, una lección, un módulo o una colección de objetos que no se asimila con ningún nivel concreto. El manifiesto es un fichero XML de nombre "imsmanifest.xml". Además, el paquete puede incorporar información para proporcionar instrucciones a la plataforma sobre cómo manejar su contenido. Algunos de estos elementos se emplean luego en el modelo SCORM RTE.

• SCORM 2004 Run-Time Environment

Incluye una guía para lanzar contenidos y hacerle un seguimiento en un ambiente basado en Web. Este libro es derivado del *CMI001 Guidelines for Interoperability* del AICC. Como hemos comentado, un requerimiento de SCORM es que el contenido educativo sea interoperativo a través de múltiples plataformas, sin tener en cuenta las herramientas que se usen para crear o usar los contenidos. Para que esto sea posible, debe existir un método común para lanzar un contenido, un método común para que los contenidos se comuniquen con las plataformas y elementos de datos predefinidos que sean intercambiables entre las plataformas y el contenido durante su ejecución.

Los tres componentes del entorno de ejecución de SCORM son:

Lanzador: Es el mecanismo que define el método común para que las plataformas lancen un SCO basado en Web. Este mecanismo define los procedimientos y las responsabilidades para el establecimiento de la comunicación entre el contenido a mostrar y el LMS. El protocolo de comunicación está estandarizado a través del uso común de la API.

API (*Application Program Interface*). Proporciona un conjunto de funciones predefinidas para que la plataforma pueda comunicarse y controlar a los SCO que lanza. EL objeto queda enlazado a la plataforma, enlace que se rompe cuando ya no se necesita el objeto. Las funciones también permiten que los objetos lean y

escriban información en la plataforma y comprobar los errores que se produzcan durante el proceso.

Modelo de Datos: Está formado por una lista estandarizada de elementos (un vocabulario) que se emplea para intercambiar información. Por ejemplo, la puntuación que un estudiante ha obtenido al realizar un test contenido en un SCO.

- **SCORM 2004 Sequence Information and Navigation**

El cuarto libro, que surge con la última versión de SCORM (SCORM 2004, que correspondería con la versión 1.3 siguiendo la nomenclatura inicial), describe cómo debe producirse la secuenciación de los contenidos almacenados en los SCO a través de una serie de eventos de navegación que pueden ser provocados por el estudiante o a iniciativa del propio sistema. Las posibles ramificaciones de los contenidos y los flujos que describen los posibles recorridos se establecen habitualmente durante el diseño.

En este libro se define un método para representar el comportamiento de una actividad de aprendizaje, que queda recogido en el SCORM SN Model. La plataforma debe incluir las funciones necesarias para seguir la secuencia indicada en el SCO en tiempo de ejecución. La estructura que se emplea para definir las ramificaciones y los recorridos por una actividad de aprendizaje es un Árbol de Actividad:

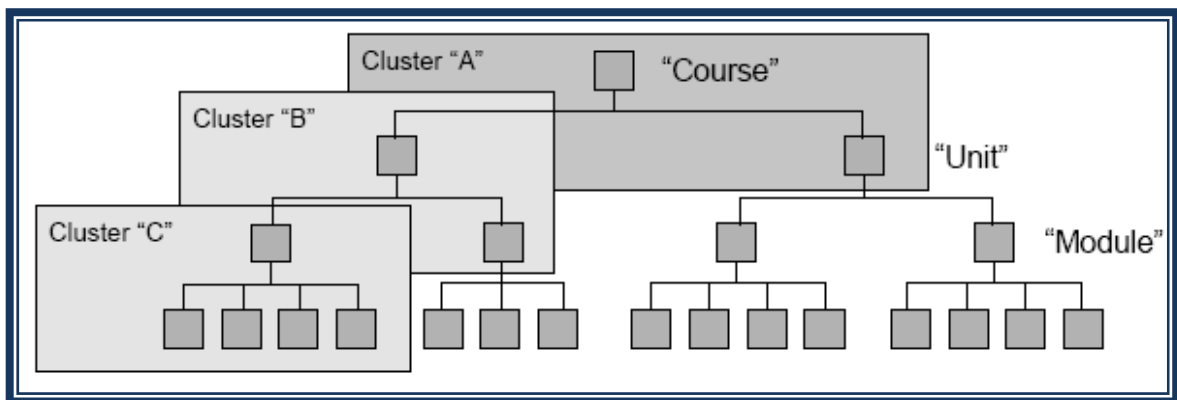


Figura 9. Estructura de un árbol de actividad ⁴⁸

Es una estructura conceptual que muestra las actividades que la plataforma de formación ha gestionado para cada usuario (determina el itinerario formativo seguido en la plataforma).

⁴⁸ Miguel Rebollo Pedruelo, El estándar SCORM para EaD, Tesina del Máster en Enseñanza y Aprendizaje abiertos y a Distancia Universidad Nacional de Educación a Distancia Diciembre 2004

SCORM también divide el mundo de la tecnología e-learning en componentes funcionales. SCORM representa el conjunto de especificaciones que permiten desarrollar, empaquetar y entregar materiales educativos de alta calidad en el lugar y momento en que sea necesario. Los principales componentes son: *Learning Management System*.

(LMS) Y *Sharable Content Objects (SCO)*. SCO se refiere a objetos de aprendizaje reusables y estandarizados. Otros componentes en el modelo SCORM son herramientas que crean los SCOs y los ensamblan en unidades de aprendizaje más grandes (un curso por ejemplo).

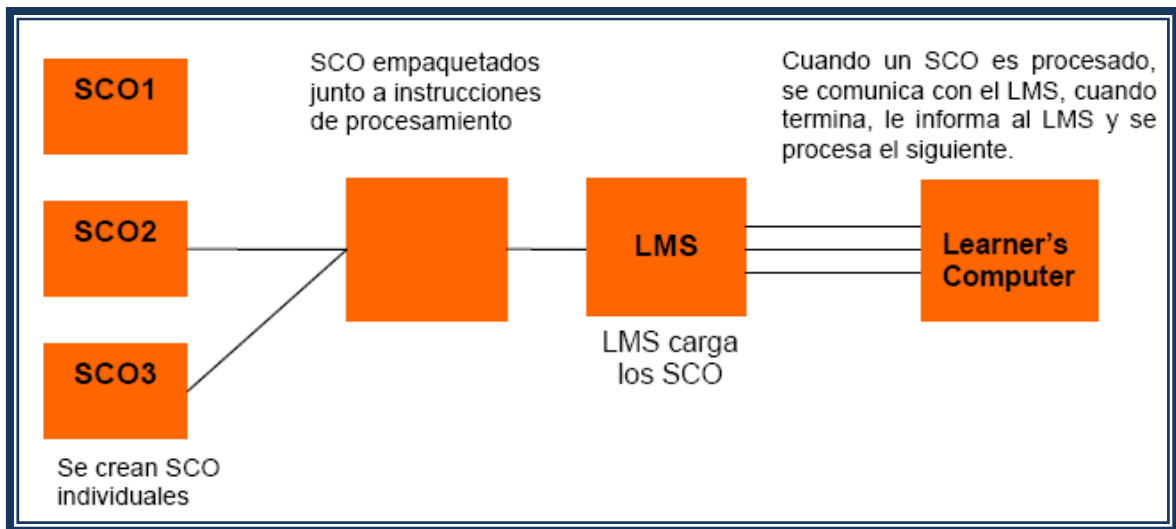


Figura 10. Componentes funcionales de SCORM⁴⁹

❖ COMPONENTES DE LA ESPECIFICACIÓN SCORM

La especificación SCORM cuenta con tres componentes:

1. El "**modelo de agregación de contenidos**", que asegura métodos coherentes en materia de almacenamiento, de identificación, de condicionamiento de intercambios y de recuperación de contenidos.

El modelo de agregación de contenidos puede descomponerse en varias funcionalidades.

La primera es la definición de « Learning Object Metadata » (LOM). Estos metadatos, utilizados dentro de los estándares de IEEE, de Ariadne y de IMS,

⁴⁹ Tomado de: <http://empresas.sence.cl/documentos/elearning>

permiten la definición de un diccionario de términos describiendo el contenido del objeto de aprendizaje.

La segunda especificación une los metadatos y el/los archivo(s) XML, reutilizándose de IMS. Define como codificar los archivos XML a fin de que sean legibles por la máquina.

La última especificación trata del empaquetado. Define cómo empaquetar el conjunto de una colección de objetos de aprendizaje, sus metadatos, y las informaciones sobre la manera en que el contenido debe ser leído para el usuario. En la práctica, se trata de crear un archivo zip que contiene todos los ficheros apropiados, así como un fichero manifest.XML, definiendo los contenidos de los diferentes ficheros y las relaciones entre ellos.

2. El "entorno de ejecución" describe las exigencias sobre el sistema de gestión del aprendizaje (SGA) que éste debe implementar para que pueda gestionar el entorno de ejecución con el contenido SCORM.

Se necesita una comunicación entre el objeto pedagógico (más particularmente, el estudiante) y el sistema de aprendizaje (Learning Management System). Por ello, ADL ha trabajado en colaboración con AICC para establecer un envío estandarizado de la información entre los dos sentidos, y compatible con las tecnologías Internet. Se ha definido una API (Application Program Interface) en Javascript, que suministra una manera estándar de comunicar con un LMS, independientemente de la herramienta utilizada para desarrollar el contenido.

3. El "modelo de secuenciamiento y de navegación" permite una presentación dinámica del contenido. Describe como el sistema interpreta las reglas de secuenciamiento introducidas por un desarrollador de contenidos, así como los eventos de navegación lanzados por el estudiante o por el sistema. Esta especificación describe el orden de la presentación de los contenidos según la navegación hecha por el usuario. Con este propósito se definen los llamados árboles de actividades, que definen las posibles ordenaciones según las acciones efectuadas por el usuario.

2.2.4. INGENIERÍA DEL SOFTWARE

La **Ingeniería de software** es la rama encargada, a través del estudio y establecimiento de principios, métodos y herramientas, del desarrollo y mantenimiento de aplicaciones software de calidad.

El término Ingeniería se define en el DRAE⁵⁰ como:

⁵⁰ DRAE, Diccionario de la Real Academia Española de la Lengua.

1. Conjunto de conocimientos y técnicas que permiten aplicar el saber científico a la utilización de la materia y de las fuentes de energía. 2. Profesión y ejercicio del Ingeniero" y el término Ingeniero se define como: persona que profesa o ejerce la Ingeniería. De igual modo la Real Academia de Ciencias Exactas, Físicas y Naturales de España define el término Ingeniería como: " Un conjunto de conocimientos y técnicas cuya aplicación permite la utilización racional de los materiales y de los recursos naturales, mediante invenciones, construcciones u otras realizaciones provechosas para el hombre".⁵¹

Sin embargo ni el DRAE ni la Real Academia Española de Ciencias han incluido todavía el término en sus últimas ediciones; por esta razón citaremos definiciones de otras fuentes:

- Ingeniería del Software es el estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas de software.⁵²
- Ingeniería del Software es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación asociada requerida para desarrollar, operar (funcionar) y mantenerlos. Se conoce también como desarrollo de software o producción de software.⁵³
- Ingeniería del Software trata del establecimiento de los principios y métodos de la Ingeniería a fin de obtener software de modo rentable que sea fiable y trabaje en máquinas reales.⁵⁴
- La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación (funcionamiento) y mantenimiento del software; es decir, la aplicación de ingeniería al software.⁵⁵

La necesidad de establecer técnicas que permitieran reducir los costos y mejorar la calidad de aplicaciones computacionales ha sido uno de los objetivos más perseguidos de la informática. Desde sus inicios, la creación de un producto software se había convertido en una tarea complicada, por esto fue indispensable introducir una serie de herramientas y procedimientos que apoyaran la labor de creación de nuevo software y, la comprensión y el manejo del mismo. Estos fueron los inicios de la Ingeniería del Software. Con el paso del tiempo, la evolución de estos métodos nos han llevado a reconocer la Ingeniería del Software como una verdadera disciplina, derivada de una investigación seria y de un estudio minucioso.

⁵¹ PRESSMAN, Roger S.: *Ingeniería del Software un enfoque práctico*. Mc Graw Hill, Quinta Edición.

⁵² ZELKOVIT, M. V., Shaw, A. C. y GANNON, J. D.: *Principles of Software Engineering and Design*. Prentice Hall, 1979.

⁵³ BOEHM, B. W.: <<Software Engineering>>, *IEEE Transactions on Computers*, C-25, núm. 12.

⁵⁴ BAUER, F. L.: <<Software Engineering>>, *Information Processing, 71*, North Holland Publishing Co., Amsterdam, 1972.

⁵⁵ IEEE: *Standards Collection: Software Engineering*, IEEE Standard 610.12-1990.

Utilizar los principios de la Ingeniería del Software, esto es, la unión de un proceso y unos métodos de desarrollo, con el apoyo de herramientas de soporte a lo largo de todo el ciclo de vida del software, es la única forma de afrontar de forma racional los, cada día más complejos, proyectos software que demanda la sociedad, sin volver a caer en los errores de un proceso de desarrollo inmaduro que da como resultado una base de software que no se puede mantener.

❖ MÉTRICAS

El proceso de planificación del desarrollo de cualquier sistema debe hacerse partiendo de una estimación del trabajo a realizar. Sólo a partir de ello es factible conocer los recursos necesarios y el tiempo necesario para su realización.

Una **métrica** es una medida efectuada sobre algún aspecto del sistema en desarrollo o del proceso empleado que permite, previa comparación con unos valores (medidas) de referencia, obtener conclusiones sobre el aspecto medido con el fin de adoptar las decisiones necesarias. Con esta definición, la aplicación de una métrica no es un objetivo en sí mismo sino un medio para controlar el desarrollo de un sistema de software.

Hay cuatro (4) razones para medir los procesos del software, los productos y los recursos:

- **Caracterizar:**
Comprender el proceso, producto y entorno actual.
Proporcionar una línea base para futuras evaluaciones.
- **Evaluar:**
Determinar el estado, de forma que los proyectos y procesos puedan ser evaluados.
Evaluar el alcance de los objetivos de calidad.
Evaluar el impacto de la tecnología sobre los procesos y productos.
- **Predecir:**
Comprender la relación entre procesos y productos.
Establecer objetivos alcanzables de calidad, coste y planificación.
- **Mejorar:**
Identificar causas raíz y oportunidades para mejorar.
Seguir los cambios en las realizaciones y comparar con la línea base.
Comunicar razones para mejorar.

Las mediciones del mundo físico pueden englobarse en dos categorías: medidas directas y medidas indirectas.

Medidas Directas. En el proceso de ingeniería se encuentran el costo, y el esfuerzo aplicado, las líneas de código producidas, velocidad de ejecución, el tamaño de memoria y los defectos observados en un determinado periodo de tiempo.

Medidas Indirectas. Se encuentra la funcionalidad, calidad, complejidad, eficiencia, fiabilidad, facilidad de mantenimiento, etc.

➤ Definiciones

- Medida. Proporciona una indicación cuantitativa de extensión, cantidad, dimensiones, capacidad y tamaño de algunos atributos de un proceso o producto. Pueden ser *directas*, por ejemplo: número de líneas de código, número de errores encontrados, etc., o pueden ser *indirectas*, por ejemplo: funcionalidad, calidad, complejidad, etc.
- Medición. Acto de determinar una medida.
- Métrica. Es una medida cuantitativa del grado en que un sistema o proceso posee un atributo dado. Por lo general relaciona una o más medidas, por ejemplo: número de errores encontrados por cada mil líneas de código.
- Indicador. Es una métrica o combinación de métricas que proporcionan una visión del proceso, del proyecto o del software en sí, y poder hacer ajustes para que las cosas mejoren.

➤ Historia

A finales de los años '70, Allan J. Albrecht [ALBR79] [ALBR83] desarrolló una unidad de medida capaz de determinar la funcionalidad de los sistemas software. A esta métrica la llamó "*Puntos de Funcionalidad*" o, más sencillamente, "*Puntos Función*". La teoría de Albrecht se basaba en estudiar, por separado, cinco componentes o características principales del sistema software: las entradas, las salidas, las consultas o peticiones interactivas (cuando el usuario hace una petición al sistema y éste devuelve una respuesta), los ficheros lógicos internos (ficheros maestros) y los ficheros lógicos externos (interfaces con otras aplicaciones).

Pero el conteo de Puntos Función no se limitaba sólo a contar el número de componentes del sistema, sino que era preciso, además, aplicar ciertos valores representativos de la complejidad de cada elemento. Así, tras muchos ensayos, Albrecht obtuvo empíricamente los factores de peso que debían aplicarse a cada uno de los cinco componentes. Estos factores variaban en función de la complejidad de cada componente, siendo esto indicativo de la mayor o menor dificultad que conllevaba su implementación.

En Octubre de 1979, en Monterrey (California), Albrecht presentó por vez primera los resultados obtenidos en sus estudios [ALBR79], proponiendo el concepto de *Puntos de Funcionalidad* como una nueva métrica de las aplicaciones software.

En 1986 nació la IFPUG (Agrupación Internacional de Usuarios de Puntos Función), cuyos objetivos eran, entre otros, dar soporte a esta nueva técnica y promocionar su uso. En 1987, el gobierno británico adoptó la técnica propuesta

por Albrecht como el estándar a utilizar para medir la productividad de las aplicaciones software que se desarrollaban. Más tarde, en 1990, la IFPUG publicó la versión 3.0 del compendio de reglas y criterios para el conteo de Puntos Función: el CPM (*Counting Practices Manual*).

Desde Abril de 1995 está en vigor la versión 4.0 de dicho manual.

Actualmente, la Organización Internacional para la Estandarización (ISO), junto con las más importantes asociaciones de usuarios de Puntos Función, están trabajando en la elaboración de lo que será la norma ISO-14143 sobre Medida del Tamaño Funcional de aplicaciones software.

❖ MÉTRICAS DEL SOFTWARE

Se pueden definir las métricas de software como “La aplicación continua de mediciones basadas en técnicas para el proceso de desarrollo del software y sus productos para suministrar información relevante a tiempo, así el desarrollador junto con el empleo de estas técnicas mejorará el proceso y sus productos”. Las métricas de software proveen la información necesaria para la toma de decisiones técnicas.

Las métricas son la maduración de una disciplina, que, según Pressman⁵⁶ van a ayudar a la evaluación de los modelos de análisis y de diseño, en donde proporcionarán una indicación de la complejidad de diseños procedimentales y de código fuente, y ayudarán en el diseño de pruebas más efectivas; Es por eso que propone un proceso de medición, el cual se puede caracterizar por cinco actividades:

- 1) *Formulación*: La obtención de medidas y métricas del software apropiadas para la representación de software en cuestión.
- 2) *Colección*: El mecanismo empleado para acumular datos necesarios para obtener las métricas formuladas.
- 3) *Análisis*: El cálculo de las métricas y la aplicación de herramientas matemáticas.
- 4) *Interpretación*: La evaluación de los resultados de las métricas en un esfuerzo por conseguir una visión interna de la calidad de la representación.
- 5) *Realimentación*: Recomendaciones obtenidas de la interpretación de métricas técnicas transmitidas al equipo de software.

Las métricas de software incluyen otras varias actividades, tales como:

- Estimación de costo y el esfuerzo
- Medición de la productividad
- Acumulación de datos

⁵⁶ PRESSMAN, Roger S. *Ingeniería del Software un enfoque practico*. Mc Graw Hill, Quinta Edición

- Realización de modelos y mediciones de la calidad
- Elaboración de modelos de seguridad
- Evaluación y modelos de desempeño
- Valoración de las capacidades y de la madurez
- Administración por métricas
- Evaluación del método y herramientas

➤ CLASIFICACIÓN DE MÉTRICAS

- ❖ **Métricas técnicas.** Se centran en las características de software por ejemplo: la complejidad lógica, el grado de modularidad. Mide la estructura del sistema, el cómo esta hecho.
 - ❖ **Métricas de calidad.** Proporcionan una indicación de cómo se ajusta el software a los requisitos implícitos y explícitos del cliente. Es decir cómo se va a medir para que el sistema se adapte a los requisitos que pide el cliente.
 - ❖ **Métricas de productividad.** Se centran en el rendimiento del proceso de la ingeniería del software. Es decir, qué tan productivo va a ser el software que se va a diseñar.
 - ❖ **Métricas orientadas a la persona.** Aportan información sobre la forma en que la gente desarrolla software.
 - ❖ **Métricas orientadas al tamaño.** Miden de forma directa el software y el proceso por el cual se desarrolla.
 - ❖ **Métricas orientadas a la función.** Se centran en la *funcionalidad* o *utilidad* del programa, y que se estudiará con detalle en los siguientes apartados.
- ❖ **Métricas Orientadas al Tamaño⁵⁷**

Las métricas del software orientadas al tamaño provienen de la normalización de las medidas de calidad y/o productividad considerando el “tamaño” del software que se haya producido. Si una organización de software mantiene registros sencillos, se puede crear una tabla de datos orientados al tamaño, como la que muestra la Tabla 10, que lista cada proyecto de desarrollo de software y las medidas correspondientes de cada proyecto. Debe tenerse en cuenta que el esfuerzo y el costo registrados en la tabla incluyen todas las actividades de ingeniería del software (análisis, diseño, codificación y prueba) y no sólo la codificación. Otra información sobre el proyecto indica el número de páginas de documentación que se desarrollaron, la cantidad de errores registrados antes de que el software se entregara al cliente y los errores encontrados después de entregárselo al cliente dentro del primer año de utilización. También el número de personas que trabajaron en el desarrollo del proyecto.

⁵⁷ PRESSMAN, Roger S.: *Ingeniería del Software un enfoque practico*. Mc Graw Hill, Quinta Edición

Tabla 10. Datos orientados al tamaño del software.

Proyecto	LDC	Esfuerzo	\$	Pág. doc	Errores	Defectos	Personas
Alfa	12,100	24	168	365	134	29	3
Beta	27,200	62	440	1224	321	86	5
Gamma	20,200	43	314	1050	256	64	6

Para desarrollar métricas que se puedan comparar entre distintos proyectos, se seleccionan las líneas de código (LDC) como valor de normalización. Con los rudimentarios datos contenidos en la tabla 10, se puede desarrollar para cada proyecto un conjunto de métricas simples orientadas al tamaño, tales como:

- errores por KLDC (miles de líneas de código)
- defectos por KLDC
- \$ por LDC
- páginas de documentación por KLDC

Además, se pueden calcular otras métricas interesantes:

- errores por persona-mes
- LDC por persona-mes
- \$ por página de documentación

Las métricas orientadas al tamaño no están aceptadas universalmente como el mejor modo de medir el proceso de desarrollo del software. La mayor parte de la discusión gira en torno al uso de las líneas de código (LDC) como medida clave.

Los defensores de la medida LDC afirman que las LDC es un “artificio” que se puede calcular fácilmente para todos los proyectos de desarrollo de software, que muchos modelos de estimación del software existente utilizan LDC o KLDC como clave de entrada. En el lado opuesto los ofensores defienden que las medidas en LDC son dependientes del lenguaje de programación, que perjudican a los programas más cortos, pero bien diseñados, que no pueden acomodar fácilmente lenguajes procedimentales, y que su uso en estimación requiere un nivel de detalle que puede resultar difícil de alcanzar (es decir, el planificador debe estimar las LDC a producirse mucho antes de que se complete el análisis y el diseño).

❖ **Métricas Orientadas a la Función**

Utilizan una medida de la funcionalidad entregada por la aplicación como un valor de normalización. Ya que la “funcionalidad” no se puede medir directamente, se debe derivar indirectamente mediante otras medidas directas. Las métricas orientadas a la función fueron propuestas por primera vez por Albrecht, que sugirió una medida llamada “*Punto de función*”. Los puntos de función se derivan con una

relación empírica según las medidas contables (directas) del dominio de información del software y las evaluaciones de la complejidad de software.⁵⁸

El tamaño de la tarea de diseño y desarrollo de un sistema de cómputo es determinado por el producto de tres factores mostrados en la Tabla 11.

Tabla 11. Tamaño de diseño y desarrollo de un sistema de cómputo.

← Tamaño intrínseco del proyecto (Para estudios de productividad) →

Tamaño de información procesada	Factor técnico de complejidad	Factores de entorno (medio)
-Entradas - Salidas - etc.	-Batch y en línea - Fácil uso - etc.	Proyectos administrativos/riesgos. Habilidades de las personas, etc. Métodos, herramientas, lenguajes.

← Tamaño total del Proyecto (Para estimar necesidades) →

- *El tamaño de información procesada:* ésta es una medida de la información procesada y proporcionada por el sistema.
- *Factor técnico de complejidad:* en éste se toma en cuenta la medida de varias técnicas y otros factores implicados en el desarrollo y en el implemento de la información procesada requerida.
- *Factores de entorno (o medio):* éste es el grupo de factores que surge del entorno del proyecto típicamente valorado en proyectos con riesgo de medidas. Incluye habilidades, experiencia y motivaciones del personal involucrado y de los métodos, lenguajes y herramientas usadas por el equipo del proyecto.

Los primeros dos de estos tres factores son intrínsecos al tamaño del sistema en el sentido que éstos resultan directamente de los requerimientos del sistema que serán entregados al usuario.

El método de Punto de Función ha ganado aceptación para la evaluación del tamaño del sistema como un componente de la medida de productividad. Cuando están disponibles datos históricos de productividad este método puede también utilizarse como una ayuda a estimar horas-persona. Para estimar propósitos, el tercer grupo de factores del entorno tiene que ser tenido en cuenta también.

El método de Punto de Función de Allan Albrecht consiste en componentes de un sistema que se clasifican en cinco tipos: *entradas externas* (o lógicas), *salidas*, *preguntas*, *interfaces externas a otros sistemas*, y los *archivos lógicos internos*. Dependiendo del número de elementos de datos estos se denominan como “simple”, “promedio” o “complejo”. Cada componente es el número dado de puntos

⁵⁸ PRESSMAN, Roger S.: *Ingeniería del Software un enfoque practico*. Mc Graw Hill, Quinta Edición

dependiendo en tipo y complejidad (Tabla 4) y la suma para todos los componentes es expresada en "Puntos funcionales sin ajustar".

Los factores técnicos de complejidad se determinan, estimando el grado de influencia de algunos componentes "características generales de aplicación" (Tabla 3). El grado de influencia en la escala recorre de cero (no presente o no influenciada) hasta 5 (influencia fuerte). La suma de las 14 características (mostradas en la Tabla 3), que es el *Grado Total de Influencia (Fi)*, se convierte al *Factor Técnico de Complejidad (TCF)* calculándose:

$$\text{TCF} = 0.65 + 0.01 * _Fi \quad (1)$$

El valor de Fi, donde los valores de ajuste de complejidad *i* va de 1 a 14 según las respuestas de las preguntas de Tabla 12:

Tabla 12. Preguntas de Fi

Fi	Preguntas
C1	¿Requiere el sistema copias de seguridad y de recuperación fiables?
C2	¿Se requiere de comunicación de datos?
C3	¿Existen funciones de procesamiento distribuido?
C4	¿Es crítico el rendimiento?
C5	¿Se ejecutara el sistema en un entorno operativo existente y fuertemente utilizado?
C6	¿Requiere el sistema entrada de datos interactiva?
C7	¿Requiere la entrada de datos interactiva que las transacciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?
C8	¿Se actualizan los archivos maestros de forma interactiva?
C9	¿Son complejas las entradas, salidas, archivos o peticiones?
C10	¿Es complejo el procesamiento interno?
C11	¿Se ha diseñado el código para ser reutilizable?
C12	¿Están incluidas en el diseño la conversión y la instalación?
C13	¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?
C14	¿Se ha diseñado la aplicación para facilitar los cambios y para hacer fácilmente utilizada por el usuario?

❖ **Valores de Fi**

- No presente o no influencia = 0
- Influencia insignificante o incidental = 1
- Influencia moderada = 2
- Influencia promedio o medio = 3
- Influencia significativa = 4
- Influencia esencial o fuerte, a través de = 5

Los valores constantes de la ecuación anterior (1) y los pesos que se aplican a las cuentas de los dominios de información se determinan empíricamente. El tamaño intrínseco relativo del sistema en Puntos Funcionales (**FP's**) se calcula con ayuda de la Tabla 13, utilizando la siguiente fórmula:

$$FP's = UFP's * TCF$$

(2)

Tabla 13. Nivel de información procesando funciones.

Descripción	Simple	Promedio	Complejo	Total
Entrada externas	*3=	*4=	*6=	-
Salidas externas	*4=	*5=	*7=	-
Archivos internos lógicos	*7=	*10=	*15=	-
Archivos de interfaz externa	*5=	*7=	*10=	-
Indagación externa	*3=	*4=	*6=	-

Se puede notar que los Puntos Funcionales son por lo tanto números dimensionales en una escala arbitraria.

Los valores de la información mostrados en la Tabla 11 se definen a continuación:

- *Entradas Externas* (o número de entradas de usuario). Se suma cada entrada dada por el usuario, donde nos proporcione distintos datos orientados a la aplicación. Estas se diferencian de las peticiones.
- *Salidas Externas* (o número de salidas de usuario). Se suma cada salida que le proporcionará al usuario información orientada a la aplicación (informes, pantallas, mensajes de error, etc.). Los elementos de datos particulares de un informe no se cuentan en forma separada.
- *Archivos Internos Lógicos o Número de archivos*. Se suma cada archivo maestro lógico (grupo lógico de datos que sean parte de una base de datos o un archivo independiente).
- *Archivos de Interfaz Externa o Número de interfaces externas*. Se suman todas las interfaces legibles por la máquina (archivos de datos de cinta o discos, etc.) que se utilizan para transmitir información a otro sistema.
- *Indagaciones externas o Número de peticiones de usuario*. La petición es una entrada dada que nos va a producir una respuesta inmediata del software en forma de salida. Las peticiones se cuentan por separado.
- *Total de Puntos funcionales sin ajustar o Cuenta-Total*. Es la suma de todas las entradas obtenidas de la Tabla 11.

Cuando se calculan los puntos de función, éstos se utilizan de forma análoga a las LDC (Líneas de Código) para normalizar medidas de productividad, calidad y otros ámbitos de software, como por ejemplo:

- Errores por Puntos de Función.
- Defectos por Puntos de Función.
- Costo (dinero) por Puntos de Función.
- Páginas de documentación por Puntos de Función.
- Puntos de Función por persona-mes.

Las razones de Albrecht para proponer los Puntos Funcionales como medidas de tamaño de un sistema son:

- Estas medidas aíslan el tamaño intrínseco del sistema de los factores del medio, facilitando el estudio de factores que influyen en la producción.
- Estas medidas están basadas en las observaciones de los usuarios externos del sistema, y es tecnología independiente.
- Estas medidas pueden determinarse al inicio del ciclo de desarrollo lo que permite utilizar los Puntos Funcionales en la estimación de procesos.
- Los Puntos Funcionales pueden ser entendidos y evaluados por usuarios que no son técnicos.

❖ ENTIDADES A MEDIR

Cualquier cosa que queramos medir o predecir en un software es un *atributo* de cualquier *entidad* de un producto, proceso o recurso asociado a éste.

Cada entidad de software tiene varios atributos que pueden ser medidos. Es por eso que se necesita hacer una distinción entre atributos que son internos o externos:

➤ Atributos Internos y Atributos Externos

Los *atributos internos* de un producto, proceso o recurso son aquellos que podemos medir puramente en términos del producto, proceso o recurso del mismo. Pueden ser medidos directamente. Por ejemplo: la longitud de un programa o el tiempo transcurrido en la obtención de cualquier documento de software.

Los *atributos externos* de un producto, proceso o recurso son aquellos que solamente pueden ser medidos con respecto a cómo el producto, proceso o recurso se relacionan con su ambiente.

Estos tienden a ser los que el administrador y el usuario del software comúnmente gustan de medir y predecir. Por ejemplo el administrador de software le gustaría saber el costo de eficacia de algunos procesos o de la productividad de su personal, mientras los usuarios les gustaría saber la usabilidad, fiabilidad, o portabilidad de un sistema que ellos observan para comprar.

Desgraciadamente los atributos externos son los más difíciles de medir, porque estos no pueden ser medidos directamente. En la Tabla 14 se describe la estructura, y se dan ejemplos de varios tipos de atributos.

Tabla14. Estructura de atributos.⁵⁹

Entidades		Atributos	
Recursos		Internos	Externos
Personal		Edad, precio,...	Productividad, experiencia, inteligencia,...
Equipo		Tamaño, estructuras,...	Productividad, calidad,...
Software		Precio, tamaño,...	Usabilidad, seguridad,...
Hardware		Precio, velocidad, tamaño de memoria,...	Seguridad,...
Oficinas		Tamaño, temperatura, luz,...	Confort, calidad,...
...		...	
Procesos			
Construcción de especificaciones	de	Tiempo, esfuerzo, número de cambios,...	Calidad, costo, estabilidad, ...
Diseño detallado		Tiempo, esfuerzo,...	Costo, costo efectivo, ...
Pruebas		Número de errores de código encontrado, tiempo,...	Estabilidad, costo, costo efectivo,...
...		...	
Productos			
Especificaciones		Tamaño, usabilidad, modularidad, funcionalidad,...	Comprensibilidad, mantenimiento, ...
Diseños		Acoplamiento, modularidad, usabilidad, funcionalidad,...	Calidad, complejidad, mantenimiento,...
Código		Funcionalidad, complejidad algorítmica, control de flujo, ...	Seguridad, usabilidad, mantenimiento, ...
Datos de prueba		Tamaño, nivel de protección,...	Calidad, ...
...		...	

➤ **Recursos, Procesos y Productos**

La primera obligación en cualquier actividad de medición de software es identificar las entidades y atributos de interés que se desea medir. Sabiendo de antemano que una *entidad* es un objeto o un evento y un *atributo* son las características o propiedades del software a medir. En el software hay tres clases de entidades cuyos atributos se miden, y se muestran en la figura 11.

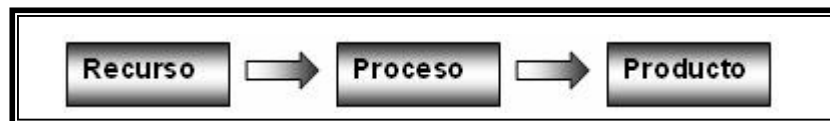


Figura 11. Entidades del software.

⁵⁹ PRESSMAN, Roger S.: *Ingeniería del Software un enfoque practico*. Mc Graw Hill, Quinta Edición

➤ Recursos

Los recursos son los diversos puntos de partida a considerar para la producción de software. La estimación de los recursos requeridos para acometer el esfuerzo de desarrollo de software se muestra en la figura 12 en forma de pirámide. En la base de la pirámide de recursos se encuentra el *entorno de desarrollo* – hardware y software que proporcionan la infraestructura de soporte al esfuerzo de desarrollo. En un nivel más alto se encuentran los *componentes de software reutilizables* – que son bloques de software que pueden reducir drásticamente los costos de desarrollo y acelerar la entrega. En la parte más alta de la pirámide está el *recurso primario*: las personas.

Cada recurso queda especificado mediante cuatro características: descripción del recurso, informe de disponibilidad, fecha cronológica en la que se requiere el recurso, tiempo durante el cual será aplicado el recurso. Las dos últimas características pueden verse como una *ventana temporal*. La disponibilidad del recurso para una ventana específica tiene que establecerse lo más pronto posible.

Recursos Humanos: El encargado de la planificación empezará elevando el ámbito y seleccionando las habilidades técnicas requeridas para llevar a cabo el desarrollo. Dentro de la selección se deberá especificar la posición que ocupará dentro de la organización (ingeniero de software, administrador,...) y la especialidad (telecomunicaciones, bases de datos, microprocesadores, administrador de proyectos o de grupo). Para proyectos relativamente pequeños (una persona-año o mes o menos) una sola persona puede llevar a cabo todos los pasos de ingeniería del software, consultado con especialistas siempre que requiera. El número de personas requerido para un proyecto de software sólo puede ser determinado después de hacer una estimación del esfuerzo de desarrollo (personas-mes, personas-año).



Figura 12. Recursos de desarrollo de software.

➤ Procesos

Las entidades de procesos de software incluyen actividades relacionadas con el software y eventos que usualmente son asociados con un factor de tiempo. Las métricas del proceso de software se utilizan para propósitos estratégicos. Por ejemplo: actividades definidas, como el desarrollo de un sistema de software desde los requerimientos hasta la liberación al usuario o la inspección de una parte del código. Las métricas de proceso también se extraen midiendo las características de tareas específicas de la ingeniería de software y obteniendo como resultados medidas de errores detectados antes de la entrega del software, defectos detectados e informados por los usuarios finales, productos de trabajo entregados, el esfuerzo humano y tiempo consumido, ajuste con la planificación y otras medidas.

Los indicadores de procesos permiten a una organización de ingeniería del software tener una visión profunda de la eficacia de un proceso ya existente; por ejemplo: el paradigma, las tareas de ingeniería del software, los productos del trabajo e hitos. También permiten que los administradores evalúen lo que funciona y lo que no. Las métricas de proceso se recopilan de todos los proyectos y durante un largo periodo de tiempo. El objetivo es proporcionar indicadores que lleven a mejoras de los procesos de software a largo plazo.

Según Pressman⁶⁰ los indicadores de proyecto permiten al administrador de software:

- Evaluar el estado del proyecto en curso.
- Seguir la pista de los riesgos potenciales.
- Detectar las áreas de problemas antes de que se conviertan en “críticas”
- Ajustar el flujo y las tareas de trabajo.
- Evaluar la habilidad del equipo del proyecto en controlar la calidad de los productos de trabajo de la ingeniería del software.

La única forma racional de mejorar cualquier proceso es medir atributos del proceso, desarrollando un juego de métricas para proporcionar indicadores que conducirán a una estrategia de mejora.

En la figura 13, se muestra el *proceso*, en el centro de un triángulo que es conectado por tres factores con una gran influencia en la calidad del software y en el rendimiento como organización donde la complejidad del *producto* puede tener un impacto sustancial sobre la calidad y el rendimiento del equipo, la *tecnología* (p. Ej. métodos de la ingeniería del software) que sustenta el proceso, también tiene su impacto.

⁶⁰ PRESSMAN, Roger S.: *Ingeniería del Software un enfoque practico*. Mc Graw Hill, Quinta Edición

Están dentro del círculo de condiciones entornos que incluyen “Entornos de Desarrollo” (p. Ej. Herramientas CASE), “Condiciones del Negocio” (p. Ej. fechas, tope, reglas de empresa), y “Características del Cliente” (p. Ej.: facilidad de comunicación”.

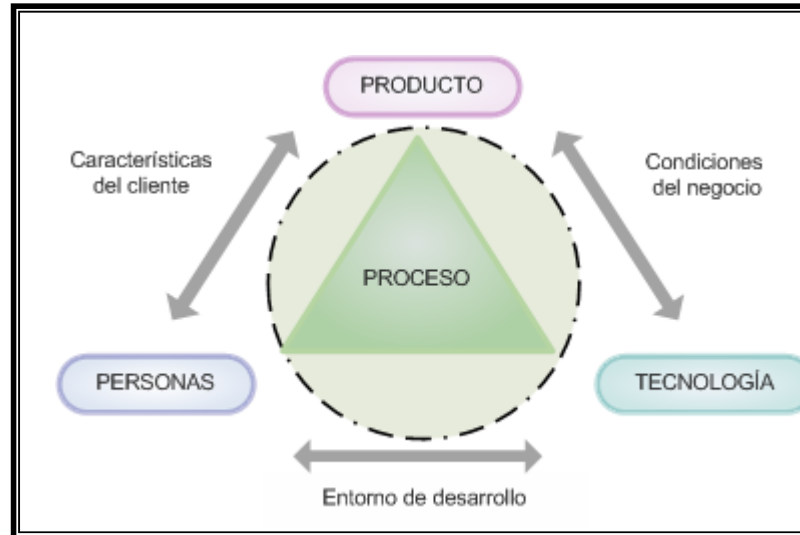


Figura 13. Determinantes de la calidad del software y de la efectividad de organización. Adaptación de [Pressman 02]

Las métricas de proceso del software pueden proporcionar grandes resultados a medida que una empresa trabaja para aumentar su nivel de madurez del proceso.

También clarifican el estatus de los proyectos y su aprobación con el plan de desarrollo ya que crear un buen proceso (buenas técnicas, administración, horarios, etc.), se reflejará en la calidad de desarrollo del software. Aunque como en todas las métricas éstas pueden ser mal utilizadas originando así un mayor número de problemas, se sugiere que los administradores a medida que instituyen un programa de métricas de proceso tengan en cuenta las siguientes consideraciones:

- Utilizar el sentido común y una sensibilidad organizativa al interpretar datos de métricas.
- Proporcionar una realimentación regular a particulares y equipos que hayan trabajado en la recopilación de medidas y métricas.
- No utilizar métricas para evaluar a particulares. Trabajar con profesionales y equipos para establecer objetivos claros y métricas que se vayan a utilizar para alcanzarlos.
- No utilizar nunca métricas que amenacen a particulares o equipos.

- Los datos de métricas que indican un área de problemas no se deberían considerar “negativos”. Estos datos son meramente un indicador de mejora del proceso.
- No debe haber una sola métrica excluyendo otras métricas importantes.

➤ Productos

Los productos del software son las salidas del proceso de producción del software. Éstas incluyen todos los artefactos entregados o documentos que son productos durante el ciclo de vida del software. Las métricas de los productos usualmente cuantifican algunos aspectos de calidad relacionados a la Tabla 15.

Tabla 15. Aspectos de calidad de software.

Lista de Requerimientos de Calidad	
Las capacidades de los productos	El grado de la funcionalidad
Usabilidad	Fiabilidad
Desempeño (eficiencia)	Seguridad
Factores humanos	Portabilidad
La extensión de la solución	Futuro/expectativas de tiempo de vida
Compatibilidad	Mantenibilidad:
Adaptabilidad	Correctivo
Recursos	Preventivo
Limitaciones personales	Instalación
Limitaciones presupuestales	Horario
Beneficios:	Limitaciones físicas
Tangibles	Documentación
Intangibles	Limitaciones organizacionales

No siempre será posible medir éstos directamente, especialmente cuando las métricas son *predecibles*, por ejemplo, cuando el software no está en desarrollo y la métrica es usada para predecir su calidad operacional; entonces los subfactores y atributos deliberados son medidas que correlacionan dando como resultado una métrica indirecta.

❖ Métricas en proyectos software.

Las medidas del proyecto de software son tácticas. Esto es, las métricas de proyectos y los indicadores derivados de ellos los utilizan un administrador de proyectos y un equipo de software para adaptar el flujo de trabajo del proyecto y las actividades técnicas.

La primera aplicación de métricas de proyectos en la mayoría de los proyectos de software ocurre durante la estimación. Las métricas recopiladas de proyectos anteriores se utilizan como una base desde la que se realizan las estimaciones del

esfuerzo y del tiempo para el actual trabajo de software. A medida que avanza un proyecto, las medidas del esfuerzo y del tiempo consumido se comparan con las estimaciones originales (y la planificación del proyecto). El administrador de proyectos utiliza estos datos para supervisar y controlar el avance.

A medida que comienza el trabajo técnico, otras medidas de proyectos comienzan a tener significado. Se miden los índices de producción representados mediante páginas de documentación, las horas de revisión, los puntos de función y las líneas fuente entregadas. Además, se sigue la pista de los errores detectados durante todas las tareas de ingeniería del software. Cuando va evolucionando el software desde la especificación al diseño, se recopilan las métricas técnicas para evaluar la calidad del diseño y para proporcionar indicadores que influirán en el enfoque tomado para la generación de código, módulos y pruebas de integración.

La utilización de métricas para el proyecto tiene dos aspectos fundamentales: En primer lugar, estas métricas se utilizan para minimizar la planificación de desarrollo guiando los ajustes necesarios que eviten retrasos y atenúen problemas y riesgos potenciales. En segundo lugar, las métricas para el proyecto se utilizan para evaluar la calidad de los productos en el momento actual y cuando sea necesario, modificar el enfoque técnico del mejoramiento de la calidad.

A medida que mejora la calidad, los errores se minimizan y el número de defectos disminuye, la cantidad de trabajo que ha de rehacerse también se reduce. Esto lleva a una reducción del costo global del proyecto.

Otro modelo de métricas del proyecto de software sugiere que todos los proyectos deberían medir:

- *Entradas.* La dimensión de los recursos (p. ej. personas, medio ambiente) que se requieren para realizar el trabajo.
- *Salidas.* Medidas de las entregas o productos creados durante el proceso de ingeniería del software.
- *Resultado.* Medidas que indican la efectividad de las entregas

En realidad, este modelo se puede aplicar tanto al proceso como al proyecto. En el contexto del proyecto, el modelo se puede aplicar recursivamente a medida que aparece cada actividad estructural. Por consiguiente, las salidas de una actividad se convierten en las entradas de la siguiente. Las métricas de resultados se pueden utilizar para proporcionar una indicación de la utilidad de los productos cuando fluyen de una actividad (o tarea) a la siguiente.

➤ **Integración de las métricas dentro del proceso de software**

La mayoría de los desarrolladores de software todavía no miden, y por desgracia, la mayoría no desean ni comenzar. Como se ha señalado, el problema es cultural. En un intento por recopilar medidas en donde no se haya recopilado nada

anteriormente, a menudo se opone resistencia: “¿Por qué necesitamos hacer esto?”, se pregunta un administrador de proyectos fatigado. “No entiendo por qué”, se queja un profesional saturado de trabajo. ¿Por qué es tan importante medir el proceso de ingeniería del software y el producto (software) que produce? La respuesta es relativamente obvia. Si no se mide, no hay una forma real de determinar si se está mejorando, y si no se está mejorando, se está perdido. La medición es una de las “medicaciones” que pueden ayudar a curar el “mal del software”. Ésta proporciona beneficios al nivel de proyecto, estratégico y técnico.

La administración de alto nivel puede establecer objetivos significativos de mejora del proceso de ingeniería del software solicitando y evaluando las medidas de productividad y de calidad.

Se ha señalado que el software es un aspecto de administración estratégico para muchas compañías. Si el proceso por medio del cual se desarrolla puede mejorarse, entonces puede producirse un impacto directo en lo sustancial. Pero para establecer objetivos de mejora, se debe comprender el estado actual de desarrollo del software.

Los rigores del trabajo diario de un proyecto de software no dejan mucho tiempo para pensar en estrategias. Los administradores del proyecto de software están más preocupados por aspectos mundanos (aunque igualmente importantes): desarrollo de estimaciones significativas del proyecto, producción de sistemas de alta calidad, y terminar el producto a tiempo.

Mediante el uso de la medición para establecer una línea base del proyecto, cada uno de estos asuntos se hace más fácil de manejar. Ya se ha apuntado que la línea base sirve como un lineamiento para la estimación. Además, la recopilación de métricas de calidad permite a una organización “sintonizar” su proceso de ingeniería del software para eliminar las causas “poco vitales” de los defectos, que tienen el mayor impacto en el desarrollo del software.

Técnicamente las métricas del software, cuando se aplican al producto, suministran beneficios inmediatos. Cuando se ha terminado el diseño del software, la mayoría de los que desarrollan pueden estar ansiosos por obtener respuestas a preguntas como:

- ¿Qué requisitos del usuario son más susceptibles de cambio?
- ¿Qué módulos del sistema son más propensos a error?
- ¿Cómo se debe planificar la prueba para cada módulo?
- ¿Cuántos errores (de tipos concretos) se pueden esperar cuando comience la prueba?

Se pueden encontrar respuestas a esas preguntas si se han recopilado métricas y se han usado como guía técnica.

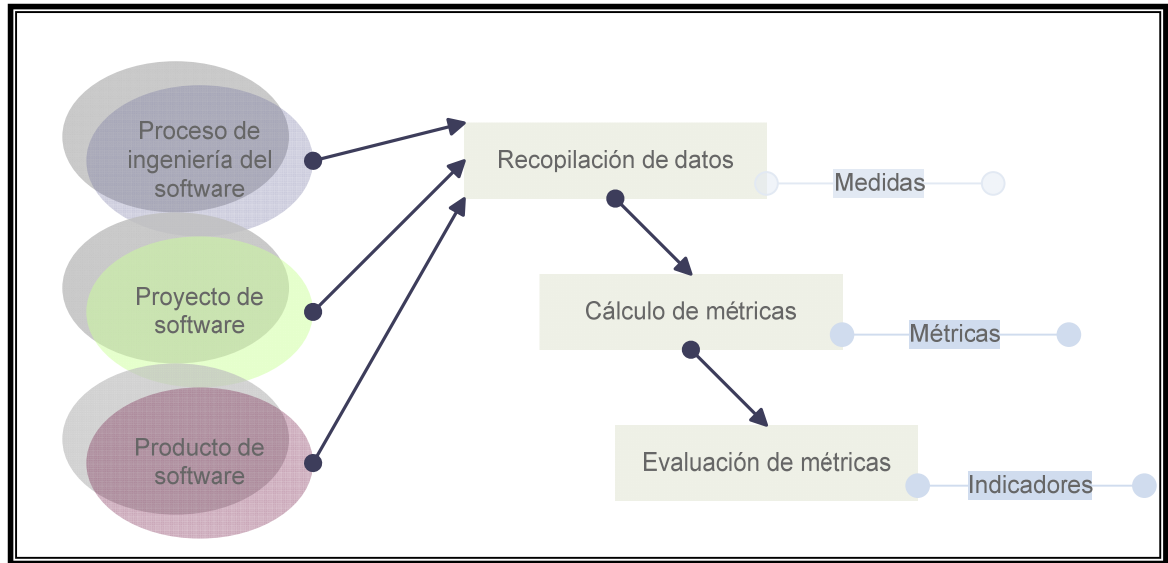


Figura 14. Proceso de recopilación de métricas del software.

2.2.5. Lenguaje unificado de modelado (UML)

El Lenguaje de Modelado Unificado (UML) es la sucesión de una serie de métodos de análisis y diseños orientados a objetos que aparecen a finales de los 80's y principios de los 90s. Directamente unifica los métodos de Booch, Rumbaugh (OMT), y Jacobson, y algo más.

UML es llamado un lenguaje de modelado, no un método. El lenguaje de modelado es la notación (principalmente gráfica) que usan los métodos para expresar un diseño. El proceso indica los pasos que se deben seguir para llegar a un diseño.⁶¹

Esta herramienta permite captar información sobre la estructura estática y el comportamiento dinámico de un sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo.

Según su definición, UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software, desde una perspectiva orientada a objetos:

Es un lenguaje, porque ofrece una serie de reglas y vocabulario para la representación conceptual y operativa de un sistema.

⁶¹ Fuente: <http://www.ultrasist.com.mx/tecnologias/uml.htm>

Es un lenguaje para visualizar, porque permite a los desarrolladores representar gráficamente, el proceso de modelado aplicado a sus proyectos software. UML combina unos símbolos gráficos con una semántica bien definida, lo que permite elaborar modelos totalmente comprensibles.

Es un lenguaje para especificar, Cubre la especificación de todas las decisiones de análisis, diseño e implementación que deben realizarse al desarrollar y desplegar un sistema con gran cantidad de software.

Es un lenguaje para construir, aunque UML no es un lenguaje de programación, tiene una relación directa con las plataformas de desarrollo como Java, C++, Visual Basic, inclusive, con bases de dato relacional y/o orientado a objetos. Es posible la generación de código a partir de un modelo en UML.

Es un lenguaje para documentar, porque documenta toda clase de artefactos que produce una organización software además de código ejecutable. Estos artefactos incluyen: requisitos, arquitectura, diseño, código fuente, planificación de proyectos, pruebas, prototipos y versiones.

Objetivos del UML⁶²

- UML es un lenguaje de modelado de propósito general que pueden usar todos los modeladores. No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática.
- UML no pretende ser un método de desarrollo completo. No incluye un proceso de desarrollo paso a paso. UML incluye todos los conceptos que se consideran necesarios para utilizar un proceso moderno iterativo, basado en construir una sólida arquitectura para resolver requisitos dirigidos por casos de uso.
- Ser tan simple como sea posible pero manteniendo la capacidad de modelar toda la gama de sistemas que se necesita construir. UML necesita ser lo suficientemente expresivo para manejar todos los conceptos que se originan en un sistema moderno, tales como la concurrencia y distribución, así como también los mecanismos de la ingeniería de software, como son la encapsulación y componentes.
- Debe ser un lenguaje universal, como cualquier lenguaje de propósito general.
- Imponer un estándar mundial.

UML se compone de diferentes diagramas, entre ellos están: diagrama de casos de uso, diagrama de clases, diagrama de objetos, diagrama de secuencia, diagrama de colaboración, diagrama de estados, diagrama de actividades, diagrama de componentes, diagrama de despliegue y algunos menos usados.

⁶² Fuente: <http://www.creangel.com/uml/intro.php>

Para modelar un proyecto, hay que centrarse en los detalles relevantes mientras se ignoran los demás, por lo cual no es necesario utilizar todos los diagramas UML. Varios diagramas aportan diferentes vistas de un sistema los cuales nos ayudan a comprenderlo desde varios frentes. Así, para este proyecto se emplearon 2 diagramas: de Casos de Uso y de Secuencia.

❖ DIAGRAMAS DE CASOS DE USO

Este diagrama representa la funcionalidad completa de un sistema mostrando su interacción con los agentes externos. Dicha representación se hace a través de las relaciones entre los actores (agentes externos) y los casos de uso (acciones) dentro del sistema.

Los diagramas de Casos de Uso definen conjuntos de funcionalidades afines que el sistema debe cumplir para satisfacer todos los requerimientos que tiene a su cargo. Esos conjuntos de funcionalidades son representados por los casos de uso.

- Los Casos de Uso describen bajo la forma de acciones y reacciones el comportamiento de un sistema desde el punto de vista del usuario.
- Permiten definir los límites del sistema y las relaciones entre el sistema y el entorno.
- Los Casos de Uso son descripciones de la funcionalidad del sistema independientes de la implementación.
- Están basados en el lenguaje natural, es decir, es accesible por los usuarios.

Un diagrama de casos de uso se compone de:

1. **Casos de uso.** Los cuales se representan por una elipse y una frase corta que describe el caso de uso.
2. **Actores.** Se representan por una figura de palillos con una silueta humana y la cual corresponde al usuario del sistema. Estos usuarios pueden ser personas, máquinas u otros sistemas software.
3. **Relaciones.** Se representan por flechas las cuales indican las relaciones que existen entre casos de uso.

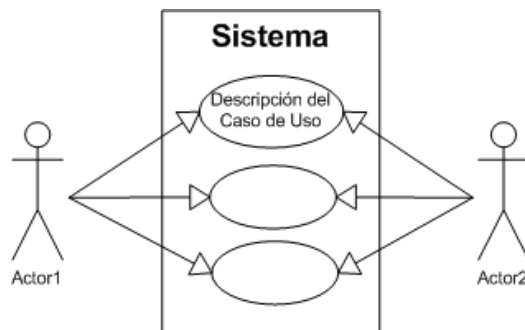


Figura15. Elementos de un diagrama de Casos de Uso

Parámetros para la construcción de un caso de uso:

Un caso de uso debe ser simple, inteligible, claro y conciso. Generalmente hay pocos actores asociados a cada Caso de Uso. Preguntas clave:

- cuáles son las tareas del actor?
- qué información crea, guarda, modifica, destruye o lee el actor?
- debe el actor notificar al sistema los cambios externos?
- debe el sistema informar al actor de los cambios internos?

❖ DIAGRAMA DE SECUENCIA

Representan un conjunto de elementos de un sistema que interactúan entre ellos organizados en secuencias de tiempo. Los diagramas de secuencia presentan gráficamente los objetos que toman parte en un escenario particular y muestra sus interacciones a través del tiempo. Un diagrama de secuencia se modela para cada caso de uso.

Un diálogo de secuencia posee dos dimensiones: la vertical representa el tiempo, la horizontal representa los objetos que participan en la interacción. En general, el tiempo avanza hacia abajo dentro de la página (se pueden invertir los ejes si se desea). Con frecuencia sólo son importantes las secuencias de mensajes pero en aplicaciones de tiempo real el eje temporal puede ser una métrica. La ordenación horizontal de los objetos no tiene ningún significado.⁶³

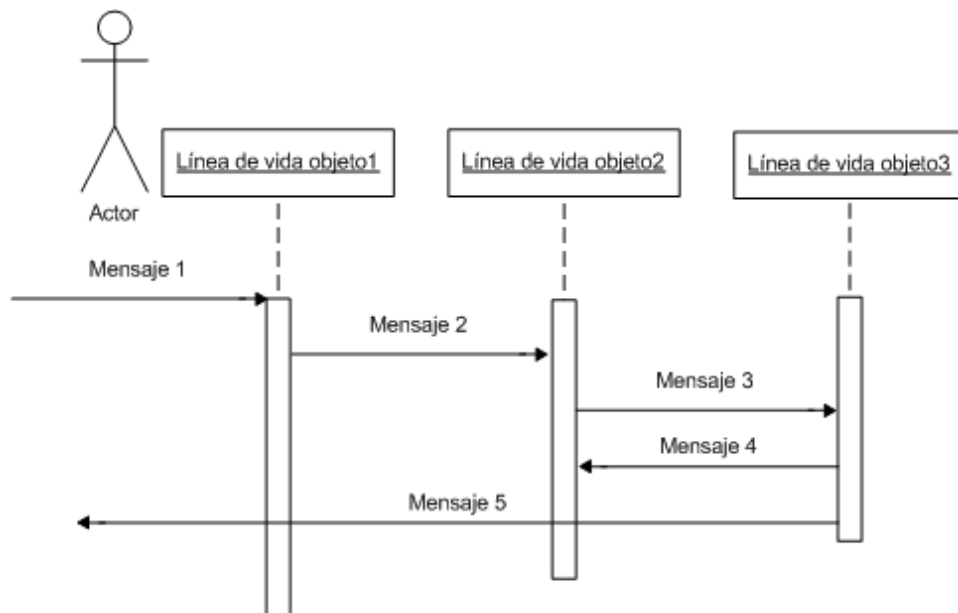


Figura16. Elementos de un diagrama de secuencia

⁶³ Fuente: <http://www.creangel.com/uml/secuencia.php>

2.2.6. Javascript

Javascript es un lenguaje sencillo que puede permitir a un programador mejorar sus páginas y la potencia de sus proyectos en la Web complementándose con código HTML. Incluso las personas que no tengan una experiencia previa en la programación podrán aprender este lenguaje con facilidad y utilizarlo en toda su potencia con sólo un tutorial.

Javascript es un tipo de lenguaje de programación que se interpreta y ejecuta por parte del navegador. Para ello, las páginas Web incluyen lo que se llama scripts, las instrucciones que el ordenador cliente ejecutará a través del programa navegador.

Lo más habitual es que el código o script esté dentro de la propia página Web, aunque también puede estar vinculado a ésta y estar contenido en un fichero con extensión *.js, así se evita tener que modificar todas las páginas si se quiere cambiar el script: bastará con modificar el fichero al cual todas estas están enlazadas.

Javascript permite ejecutar instrucciones como respuesta a las acciones del usuario, con esto se pueden crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo.

Los programas escritos en JavaScript se ejecutan en el navegador del cliente, sin necesidad de que intervenga el servidor. De esta forma, una sola transacción basta para cargar la página en la que se encuentra tanto el formulario, para los datos de entrada, como el programa en JavaScript que proporciona los resultados.

Aunque comparte muchas de las características y de las estructuras del lenguaje Java, fue desarrollado independientemente. El lenguaje Javascript puede interactuar con el código HTML, permitiendo a los programadores Web utilizar contenido dinámico. Por ejemplo, hace fácil responder a los acontecimientos iniciados por usuarios (como introducción de datos en formularios) sin tener que utilizar CGI.

Este lenguaje no requiere ser compilado, es decir, es un lenguaje interpretado. Una ventaja que tiene este lenguaje es que todos los navegadores interpretan el código JavaScript integrado dentro de las páginas Web.

2.2.7. Macromedia flash 8 profesional

Flash es una tecnología desarrollada originalmente por Macromedia (ahora hace parte de adobe), comúnmente utilizada en la Web ya que permite crear

animaciones de poco peso, es decir, que tardan poco tiempo en ser cargadas por el navegador.

Esta herramienta optimiza el tamaño de los archivos contenidos en las animaciones, como imágenes y sonidos, almacenándolos de manera inteligente por medio de bibliotecas. También permite cargar la animación al mismo tiempo que ésta se muestra en el navegador (técnica denominada *streaming*), con lo que se logra aportar elementos visuales que dan vida a una Web con la ventaja de no tener que esperar tiempos prolongados para que la página cargue.

Además, Flash introduce en su entorno la posibilidad de que el usuario interactúe con las aplicaciones. Esto se logra por medio de un lenguaje de programación llamado Action Script. Este lenguaje tiene claras influencias del Javascript y permite, entre otras muchas cosas, gestionar el relleno de formularios, ejecutar distintas partes de una animación en función de eventos producidos por el usuario, saltar a otras páginas, etc.

Algunas características de flash son:

- Las películas hechas con Flash son multiplataforma, se ven de la misma forma independientemente del navegador (Explorer, Netscape, u otros) y sistema (Windows, Macintosh, Linux u otros) que utilice el usuario.
- Las películas Flash se descargan rápidamente (gracias a la tecnología de flujo o *streaming*), en el ordenador del usuario, ya que, aunque se incorporen elementos complejos, conservan unos tamaños relativamente modestos.
- La creación de animaciones resulta sencilla. Objetos que cambian de posición, de color, de forma o, incluso, que se convierten en otros objetos, también se pueden incorporar fácilmente elementos multimedia como sonido y vídeo.
- Flash dispone de un sistema de bibliotecas que permite reutilizar los elementos de una película dentro de la misma película o en otra, aprovechando al máximo el trabajo del diseñador.

3. SEGUNDA FASE. Diseño Instruccional.⁶⁴



Figura 17. Fase 2 - Diseño Instruccional

A partir de las concepciones de lo que actualmente se maneja para el diseño instruccional y del programa de las competencias en el contexto educativo, se presentan a continuación los pasos generales que permiten hacer una mejor gestión del conocimiento dentro del proyecto PROSPETICuis con base en la metodología del análisis funcional.

REFERENTE METODOLÓGICO⁶⁵

El referente metodológico utilizado y adaptado para el desarrollo e implementación de diseños curriculares bajo la visión de competencias es el análisis funcional. A continuación se establece una base de fundamentos de la propuesta metodológica que serán tenidos en cuenta durante la construcción del proceso.

Principios de aplicación del análisis funcional:

- **Aplicar de lo general a lo particular:** el punto de arranque es el contexto de la asignatura (lo general) enmarcado por los contenidos temáticos básicos, genéricos y específicos, seleccionados a través del análisis de los contenidos presentes en literatura académica e institucional concerniente, combinado a su vez con la experiencia y conocimientos de los expertos docentes, expertos pedagogos y expertos en la metodología de la planeación del diseño curricular que acompañen el proceso. Este principio permite delimitar el área de estudio

⁶⁴ La primera fase se refiere a la definición del Proyecto de Grado.

⁶⁵ RAMÍREZ P., Dorys Consuelo - VERJEL A., Dania Rubiela. *Diseño y elaboración de la estructura curricular para la asignatura tratamiento de señales bajo una visión de competencias y estudio de adaptación a una plataforma e-learning*. Trabajo de grado (Ingeniera Electrónica) Universidad Industrial de Santander

que se pretende abarcar con la asignatura junto con la estructuración de los contenidos.

- **Identificar acciones delimitadas separándolas de un contexto específico:** la desagregación de los contenidos generales debe ser única; poseer un inicio y un fin en su descripción, definiendo un propósito y un alcance preciso; además deben estar en consonancia con el área de estudio abarcada por la asignatura y por el programa de formación general. En la propuesta metodológica los contenidos desagregados se clasifican en tres tipos: “Contenidos Conceptuales (saber)”, “Contenidos Procedimentales (saber hacer)” y “Contenidos Actitudinales (saber ser)”, que corresponden a competencias evidenciables en el estudiante. Este principio metodológico se evidencia en la estructura gramatical de los contenidos desagregados que consta de: Verbo, Objeto y Condición, en el estricto orden en que se enuncian.
- **Mantener una relación causa-consecuencia:** este principio permite que los contenidos obtenidos de la desagregación sean realmente la suma de partes que den como resultado el contenido y/o propósito origen, o dicho de otra forma, el todo este realmente sustentado en los componentes que la conforman, además que tiene la utilidad de proveer la visión de correlación que debe establecerse entre las partes.

3.1. ETAPAS PRELIMINARES AL DISEÑO INSTRUCCIONAL

a. Conformar el equipo de trabajo

El equipo de trabajo de la propuesta está compuesto por un metodólogo que posee los conocimientos en la identificación de las competencias y la metodología el análisis funcional y que se encuentra relacionado con el proceso educativo, un coordinador tecnológico que realiza el acompañamiento en la elaboración del objeto de aprendizaje y soporta el área tecnológica del proyecto, expertos docentes con conocimientos de la asignatura y encargados de proveer el manejo de los elementos del currículo y los desarrolladores concedores del análisis funcional y del área de la asignatura que sirven de medio para enriquecer y sustentar documentalmente la propuesta.

Para este proyecto específicamente el equipo de trabajo se conformó de la siguiente manera:

- Metodólogo: Ingeniero Oscar Morantes.
- Coordinador tecnológico: Ingeniero Sergio Gómez Ardila.
- Expertos temáticos: Ingeniero Leonardo Camargo Corzo.
Ingeniero Fernando Rojas Morales.
- Desarrolladores: Nadia Alexandra Carreño.
Paola Carolina Espinosa.

b. Conocer la misión profesional UIS y la de la carrera profesional en cuestión.

- **Misión profesional**

El profesional UIS es una persona de alta calidad ética, política y profesional que genera y adecua conocimientos; conservando y reinterpretando su cultura y participa de forma activa liderando procesos de cambio por el progreso y mejor calidad de vida de la comunidad; mediante el trabajo interdisciplinario y de alta relación con el mundo externo.

- **Objetivo de la Escuela de Ingeniería de Sistemas e Informática**

El ingeniero de sistemas UIS es una persona autónoma, creativa, que actúa según principios éticos universalmente aceptados, de alta calidad ciudadana y comprometidos con el desarrollo regional y nacional; y la construcción, innovación y mejoramiento del conocimiento, que dispone de la fundamentación teórica, tecnológica e instrumental para administrar y tratar los sistemas de información, las comunicaciones y la automatización industrial.

c. Definir el objetivo de la asignatura dentro de la misión universitaria y profesional que encierre los conocimientos y habilidades que se espera dentro del proceso de formación.

El Objetivo de la asignatura Ingeniería del Software del programa académico de la Escuela de Ingeniería de Sistemas e Informática es “Implementar las metodologías para el desarrollo de software buscando obtener productos de calidad, haciendo más eficientes los procesos de planteamiento y dirección de proyectos”.

ETAPAS DEL DISEÑO INSTRUCCIONAL

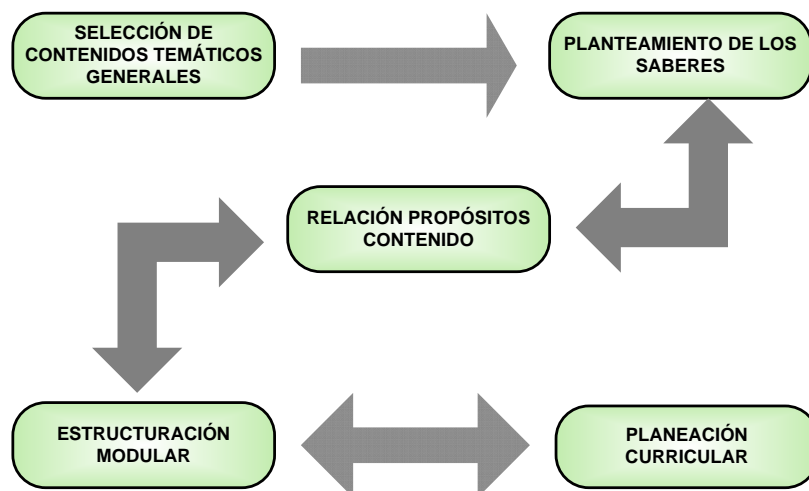


Figura 18. Etapas de la propuesta metodológica del diseño curricular.65

3.2.1. Diagrama secuencial de actividades de aprendizaje

- a. **Realizar el ordenamiento lógico y secuencial de los contenidos de la asignatura de acuerdo al programa académico aprobado por la universidad y recopilando información con base en planes académicos de otras universidades.**

Después de establecer el objetivo que persigue la asignatura se determinan todas aquellas actividades de conocimiento que requiere el estudiante con anterioridad para poder ver el contenido de la asignatura, es decir, se establecen las asignaturas que el estudiante debe cursar previamente para poder cursar la asignatura en cuestión.

Con base en este estudio el docente determina qué conocimientos de aquellas asignaturas previas son necesarios para dar inicio a la asignatura, o si por el contrario no se requieren conceptos anteriores. Con los conocimientos y experiencia del docente se estructura y determina el contenido de la asignatura, si fuese necesario complementar dicho contenido, se puede utilizar como referente recursos bibliográficos y programas académicos análogos.

- b. **Identificar la relación causa consecuencia de actividades relacionadas con el objetivo de aprendizaje trazado para determinar la forma de cómo alcanzar el mismo y el para qué, contextualizando el proceso de enseñanza aprendizaje en la metodología del análisis funcional.**

Teniendo en cuenta los temas relacionados se procede a cuestionar al docente sobre los temas que pueden englobar las actividades de aprendizaje que conformen una macro estructura de la asignatura y que sirvan como base para partir de lo general a lo particular, es decir, puedan desglosarse hasta llegar a temas que no puedan desagregarse mas. El objetivo es observar la desagregación de los temas y su interacción con otros la cual vendrá descrita por las siguientes convenciones:

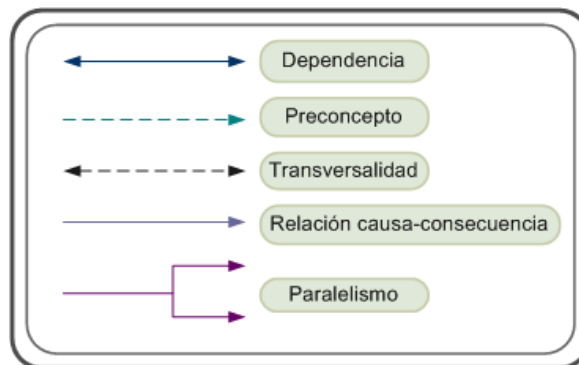


Figura 19. Convenciones para el diagrama secuencial de actividades DSA²

- ❖ **Desagregación:** Esta relación se sigue de lo general a lo particular y se representa en el diagrama a través de caminos de un contenido temático hacia otro u otros.
 - ❖ **Secuencialidad:** Representa el desarrollo lógico en la desagregación de los contenidos temáticos de la asignatura en sentido vertical, definiendo así la sucesión al abarcar los contenidos.
 - ❖ **Paralelismo:** Se refiere a los contenidos temáticos que son posibles de abordar en cualquier orden, es decir, sin tener en cuenta el orden cronológico, ya que tienen la misma jerarquía dentro del proceso de aprendizaje. Esta representado por dos o mas contenidos dentro de un mismo nivel de desagregación.
 - ❖ **Dependencia:** Establece la necesidad mutua de los conceptos, proporciona la contextualización de dos temas en el proceso de aprendizaje de la asignatura. Está representado en el diagrama mediante flechas de doble vía.
 - ❖ **Preconcepto:** Se muestra cuando existe información necesaria aunque no suficiente para abordar el tema destino por lo tanto se requiere información agregada que permita el proceso de enseñanza-aprendizaje, o en otras palabras es la articulación de dos contenidos temáticos que se relacionan pero no de forma evidente. Esta representada por flechas verticales discontinuas.
 - ❖ **Causa-consecuencia:** Simboliza que existe información necesaria y suficiente entre el tema origen y el tema de destino involucrados en el proceso de aprendizaje, es decir el contenido al inicio de la flecha es causa para el que se encuentra al final, por lo cual debe abarcarse primero el contenido origen y luego el destino. Esta representada mediante flechas horizontales que van de un contenido a otro u otros.
 - ❖ **Transversalidad:** Se presenta cuando algunos contenidos son referencia y complemento de dos o más al tiempo, este contenido se requiere para múltiples temas en diferentes espacios de tiempo y contextos para el proceso de aprendizaje (se pretende evitar la redundancia de temas dentro de la asignatura). De esta forma se puede abordar dicho contenido en varias ocasiones, proporcionándole el contexto asociado al contenido al que complementa o que lo necesita como referencia.
- c. Obtención del diagrama secuencial de actividades DSA² que exprese de forma articulada las actividades de aprendizaje para satisfacer el objetivo planteado para el aprendizaje.**

El DSA² representa gráficamente el entorno temático de la asignatura y permite definir los temas mediante actividades de aprendizaje.

También permite el desglose de las actividades de aprendizaje mediante mapas conceptuales para establecer la orientación de la navegación a seguir sobre los objetos de aprendizaje.

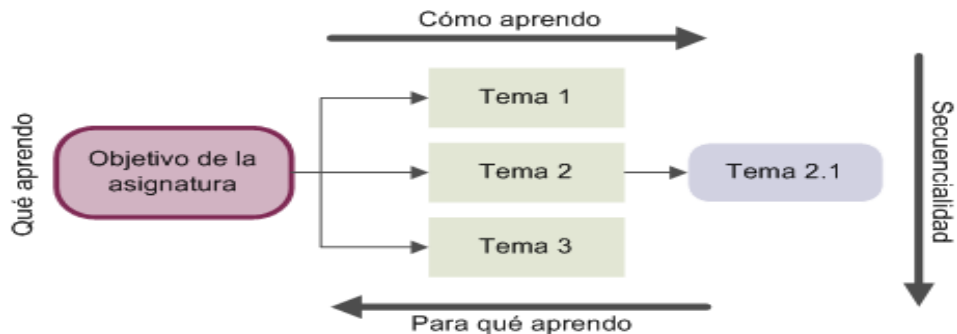


Figura 20. Esquematización de los elementos del DSA²

- ❖ **Desagregación:** En la figura 21 se puede observar qué temas se desagregan como actividades de aprendizaje de una temática general, es decir, que si el estudiante realiza las cuatro actividades de aprendizaje se puede esperar que esté en la capacidad de definir y plantear métricas para proyectos de producción de software.



Figura 21. Relación de desagregación.

- ❖ **Secuencialidad:** En la figura 22 se reflejan 3 contenidos generales expresados en actividades de aprendizaje que muestran una relación de secuencialidad, es

decir, que para definir los riesgos en proyectos de producción de software es necesario conocer los procesos de gestión de proyectos de producción de software y, para gestionar proyectos de producción de software, el estudiante debe tener la capacidad de, como primera medida, planear proyectos de producción de software.

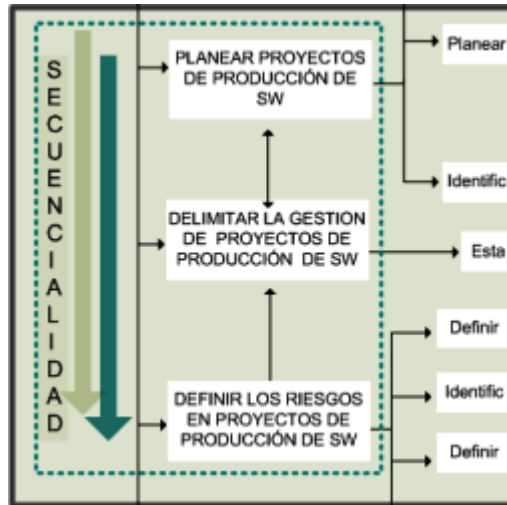


Figura 22. Relación de secuencialidad

- ❖ **Paralelismo:** En la figura 23 se muestra una relación de paralelismo, es decir, que cualquiera de los dos contenidos temáticos se puede abordar en cualquier orden y no se ve afectado de ninguna manera el proceso de aprendizaje.



Figura 23. Relación de paralelismo

- ❖ **Dependencia:** En la figura 24 se observa una relación de dependencia, esto significa que los temas se complementan entre si y que cada uno no puede ser visto en un contexto diferente al del otro.



Figura 24. Relación de dependencia

- ❖ **Preconcepto:** En la figura 25 se muestra que para poder realizar las actividades de aprendizaje relacionadas con los métodos y herramientas para el desarrollo del software es necesario que el estudiante ya haya adquirido anteriormente (tenga los preconceptos) los conocimientos sobre programación orientada a objetos (P.O.O) y programación procedimental (P.P).



Figura 25. Relación de preconcepto

- ❖ **Causa-consecuencia:** En la figura 26 se muestra la relación causa consecuencia que en este caso significa que después de haber visto la clasificación de riesgos es necesario ver como se valoran los mismos y como se supervisan para completar el proceso de aprendizaje.



Figura 26. Relación causa-consecuencia

- ❖ **Transversalidad:** En la figura 27 se muestra que los conceptos de recursos tecnológicos se requieren para múltiples temas como gestión del producto y del proceso.

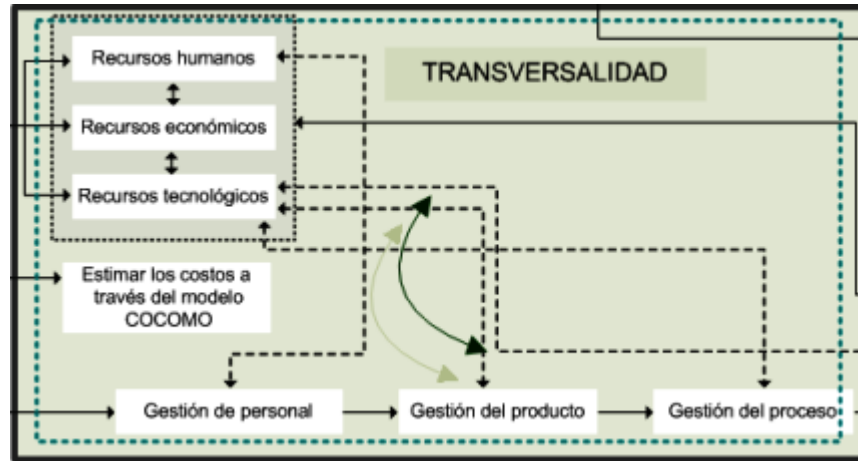


Figura 27. Relación de transversalidad

Diagrama secuencial de actividades, DSA².

El DSA² desarrollado con base en los contenidos recopilados para la asignatura es el resultado del análisis y selección de los mismos.

Las características del DSA² en su desarrollo e interpretación son:

- Mantener la relación causa consecuencia
- Mostrar las actividades de aprendizaje que conforman la asignatura
- Mostrar las relaciones descritas entre las actividades de aprendizaje.

3.2.2. Establecimiento de los saberes y haceres

Continuando con la propuesta metodológica, el siguiente paso es la desagregación de los contenidos generales presentes en el diagrama secuencial en saberes: saber, saber hacer y saber ser que a su vez corresponden a los contenidos conceptuales, procedimentales y actitudinales; este paso queda plasmado en la tabla de saberes.

A continuación se presenta gráficamente la relación entre el saber, hacer y el ser, esto quiere decir que tanto un ítem como el otro se relacionan estrechamente.

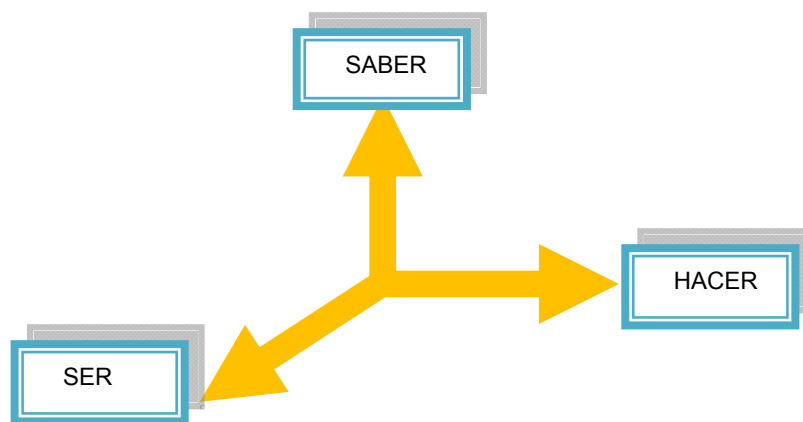


Figura 29. Visión tridimensional de la relación. Saber, Hacer y ser.

a. Identificación de las actividades de formación asociadas con los saberes - haceres del proceso de aprendizaje que describan la enseñanza aprendizaje esperados con desempeño competente del individuo

Siguiendo con la propuesta se hace la desagregación de actividades de aprendizaje principales presentes en el DSA² mediante saberes y haceres.

Los saberes son acciones puntuales de aprendizaje que se esperan desarrollar en el estudiante, en el diseño instruccional y son de tres tipos:

- ✍ “**el saber**” que se refiere a hechos, teorías y principios del conocimiento.
- ✍ “**el saber hacer**” que relaciona los procedimientos, técnicas, métodos, habilidades y destrezas que son necesarias desarrollar en el estudiante.
- ✍ **El saber ser**, que concierne a las actitudes y valores en el comportamiento del estudiante en su proceso de enseñanza – aprendizaje, el saber ser se aborda en este proyecto según lo concebido en el modelo de estilos de

aprendizaje de Felder y Silverman por esto no se establecerá en el contenido de las tablas que agruparan los saberes y haceres.

La estructura gramatical que se empleó para los contenidos desagregados consta de:

Verbo + Objeto + Condición

El saber y el saber hacer se establecen para cada uno de los contenidos generales del diagrama secuencial. En la elección de los verbos se empleo una gran variedad de verbos existentes en la Taxonomía de Bloom⁶⁶ quien define seis niveles de conocimiento. Adicionalmente se estableció la relación entre saberes y haceres.

En la *Figura 30*. Se publica un fragmento de la tabla de saberes final que se desarrolló para la asignatura “Ingeniería del software”, en ella se presenta la relación de los saberes y haceres mediante la conveniencia de asignar al final de cada hacer, entre paréntesis, la numeración dada a los saberes que se relacionan con él.

Mediante la realización de la tabla de saberes se busca:

Propósitos

- Clasificar los saberes en *saber* y *saber hacer*.
- Identificar las competencias individuales a desarrollar en la asignatura.

Resultados

- La tabla muestra en forma ordenada la clasificación de los *saberes*.
- Los *saberes* describen las acciones específicas del proceso de enseñanza-aprendizaje que se desarrollará en el estudiante, y son la guía para el docente en cuanto a las directrices de los resultados a desarrollar en los aprendices.
- Los *saberes* se relacionan verticalmente de forma secuencial, y en algunos casos de manera jerárquica, manteniendo siempre la relación causa-consecuencia de forma horizontal.

⁶⁶ Ver Anexo. Taxonomía de Objetivos de Bloom

TABLA DE SABERES ASIGNATURA: Ingeniería del Software		Versión Final	ENCABEZADO
			10
DEFINIR Y PLANTEAR MÉTRICAS PARA PROYECTOS DE PRODUCCIÓN DE SOFTWARE			
SABER		CAUSA - CONSECUENCIA	HACER
SECUENCIALIDAD	1. Definir métricas, medidas e indicadores de forma general. 2. Establecer las Métricas del software. 3. Conocer las cinco (5) actividades (Formulación, Colección, Análisis, Interpretación y Realimentación) en un proceso de medición del software 4. Definir las Métricas orientadas al tamaño. 5. Definir las Métricas orientadas a la función. 6. Describir las ventajas y desventajas de las Métricas orientadas al tamaño y orientadas a la función. 7. Conocer el concepto de atributo dentro del contexto de las métricas de software.	a. Diferenciar los conceptos de métricas, medidas e indicadores de forma general. [1] b. Interpretar el concepto de métricas en el contexto de software. [2] c. Caracterizar cada una de las cinco (5) actividades en el proceso de medición del software. [3] d. Implementar métricas orientadas al tamaño. [4, 6] e. Implementar métricas orientadas a la función. [5, 6] f. Estimar los beneficios de aplicar tanto las métricas orientadas al tamaño como las orientadas a la función. [4, 5, 6] g. Clasificar cada uno de los atributos en	
	VERBO	CONDICIÓN	OBJETO

Figura 30. Tabla de saberes

3.2.3. Establecimiento de la relación propósitos-contenidos

- a. Establecimiento de la relación propósitos-actividades de formación, en este punto se describe los valores agregados por parte del proceso de enseñanza-aprendizaje sobre el estudiante.

La finalidad de esta etapa es construir los propósitos de la asignatura para los diferentes contenidos temáticos presentes en el diagrama secuencial de actividades que se clasifican y resumen en la tabla de saberes, estableciendo así las finalidades completas de la asignatura, para esto es necesario agrupar los

saberes en conjuntos que sean de correspondencia con contenidos temáticos producto de la división de cada uno de los contenidos generales del diagrama.

Para cada relación entre el conjunto de saberes y contenidos temáticos se estructura un propósito que defina el “para qué” del aprendizaje de dichos contenidos y saberes, adquiriendo al final del proceso los propósitos correspondientemente relacionados en causa-consecuencia con sus saberes, haceres y contenidos temáticos asociados.

El principio básico para la obtención de la relación propósitos-contenidos es que la conjugación de los saberes y haceres asociados a cada propósito permita su realización y cumplimiento en toda la extensión que se define en él.

Mediante la elaboración de esta etapa se busca:

Propósitos

- Enunciar los propósitos de la asignatura.
- Relacionar los propósitos con los saberes para su cumplimiento.
- Analizar y plasmar las relaciones de causa-consecuencia entre propósitos y saberes, y a su vez, entre saber y saber hacer.
- Demostrar la secuencialidad de los propósitos y los saberes de la asignatura.

Resultados

- Determinar las políticas de aprendizaje que se desean lograr.
- Siguen la estructura gramatical uniforme dada por los principios metodológicos, junto con la recomendación de utilizar verbos activos.
- Engloban los saberes asociados para su cumplimiento.

La *Figura 31*. Expone un fragmento de la relación propósitos-contenidos de la asignatura Ingeniería del Software, donde se muestra la secuencialidad de los propósitos, los contenidos temáticos y los saberes en forma vertical, y en forma horizontal se interpreta la relación causa-consecuencia de propósitos a contenidos, de contenidos a saberes, de saber a saber hacer e igualmente en sentido inverso la relación consecuencia-causa.

RELACION PROPOSITOS CONTENIDOS		Versión Final	Página 10
DEFINIR Y PLANTEAR MÉTRICAS PARA PROYECTOS DE PRODUCCIÓN DE SOFTWARE			
PROPOSITOS	CONTENIDOS TEMATICOS	SABER	HACER
Definir y analizar los conceptos básicos de métricas	<ul style="list-style-type: none"> ➤ Definición medidas, métricas indicadores. ➤ Métricas aplicadas software 	<ol style="list-style-type: none"> 1. Definir métricas, medidas e indicadores de forma general. 2. Establecer las Métricas del software. 3. Conocer las cinco (5) actividades (Formulación, Colección, Análisis, Interpretación y Realimentación) en un proceso de medición del software. 	<ol style="list-style-type: none"> a. Diferenciar los conceptos de métricas, medidas e indicadores de forma general. [1] b. Interpretar el concepto de métricas en el contexto de software. [2] c. Caracterizar cada una de las cinco (5) actividades en el proceso de medición. [3]
Conocer los diferentes enfoques para aplicar métricas a proyectos de producción de software	<ul style="list-style-type: none"> ➤ Mediciones software: <ul style="list-style-type: none"> ▪ Métricas orientadas tamaño ▪ Métricas orientadas a la función 	<ol style="list-style-type: none"> 1. Definir las Métricas orientadas al tamaño. 2. Definir las Métricas orientadas a la función. 3. Describir las ventajas y desventajas de las métricas orientadas a la función. 	<ol style="list-style-type: none"> a. Implementar métricas orientadas al tamaño. [1, 3] b. Implementar métricas orientadas a la función. [2, 3] c. Estimar los beneficios de aplicar tanto las métricas orientadas al tamaño como las orientadas a la función. [1, 2, 3]

Figura 31. Relación propósitos-contenidos

3.2.4. Establecimiento de la estructuración modular

En esta etapa se realizan una serie de agrupaciones que permiten estructurar la asignatura teniendo en cuenta los principios metodológicos: partir de lo general a lo particular, identificar acciones delimitadas y mantener la relación de causa – consecuencia entre los diferentes desgloses en los siguientes bloques de procesos de enseñanza-aprendizaje:

- actividades de enseñanza-aprendizaje
- unidades de aprendizaje
- módulos de formación

✓ **Las actividades de enseñanza-aprendizaje:** partiendo de estas se componen las unidades de aprendizaje que dan iniciación a la estructura modular de la asignatura, convirtiéndose así en parte fundamental del diseño curricular.

Las actividades de enseñanza-aprendizaje conciernen a las actividades que el estudiante debe estar en capacidad de desarrollar de manera individual durante su proceso de formación en la asignatura.

La captura de estas actividades se realiza partiendo del cuadro de relación propósitos-contenidos establecidos, y apoyándose en la tabla de saberes, se agrupan los contenidos conceptuales, procedimentales y actitudinales (y en consecuencia los propósitos) por tipo de afinidad. Una vez determinada la afinidad más adecuada se realiza la agrupación de los propósitos evaluando siempre que se pueda identificar de forma clara una relación causa-consecuencia entre los mismos.

Como se mencionó, éstas parten de la agrupación de los propósitos ya elaborados, es por esto que el cumplimiento de dichas actividades es consecuente con el cumplimiento de uno o varios propósitos de la asignatura.

Las actividades de enseñanza-aprendizaje se pueden clasificar de la siguiente forma:

- **Actividades básicas:** Son las actividades pedagógicas mínimas para la estructuración de los conocimientos, destrezas, habilidades y valores.
- **Actividades genéricas:** Representan el hacer, el saber y el ser para cumplir con los requerimientos de formación.
- **Actividades específicas:** Corresponden al agrupamiento de actividades particulares que complementan los contenidos de la unidad de aprendizaje.

Los principios de análisis funcional presentes para la identificación de actividades de enseñanza-aprendizaje son presentar una estructura gramatical uniforme, mantener una relación causa-consecuencia, utilizar verbos activos (reales, medibles, y evaluables)

En la *Figura 32*. Se observa una actividad identificada para la asignatura Ingeniería del Software en la cual con la finalidad de no perder de vista los referentes, se mantienen presentes los propósitos, contenidos y saberes asociados a cada actividad.

ACTIVIDADES DE ENSEÑANZA APRENDIZAJE		VERSIÓN FINAL		PAGINA 10
DEFINIR Y PLANTEAR METRICAS PARA PROYECTOS DE PRODUCCION DE SOFTWARE				
PROPOSITO	CONTEHIDO TEMÁTICO	SABER	HACER	ACTIVIDADES
Definir y analizar los conceptos básicos de métricas	<ul style="list-style-type: none"> ➤ Definición de medidas, métricas e indicadores. ➤ Métricas aplicadas al software. 	<ol style="list-style-type: none"> 1. Definir métricas, medidas e indicadores de forma general. 2. Establecer las Métricas del software. 3. Conocer las cinco (5) actividades (Selección, Colección, Análisis, Interpretación y Realimentación) en un proceso de medición del software. 	<ol style="list-style-type: none"> a. Diferenciar los conceptos de métricas, medidas e indicadores de forma general. [1] b. Interpretar el concepto de métricas en el contexto de software. [2] c. Caracterizar cada una de las cinco (5) actividades del proceso de medición. [3] 	Describir el concepto de métricas y analizar su utilización en el proceso de medición.
Conocer los diferentes enfoques para aplicar métricas a proyectos de producción de software	<ul style="list-style-type: none"> ➤ Mediciones del software: <ul style="list-style-type: none"> ▪ Métricas orientadas al tamaño ▪ Métricas orientadas a la función 	<ol style="list-style-type: none"> 1. Definir las Métricas orientadas al tamaño y las Métricas orientadas a la función. 2. Describir las ventajas y desventajas de las Métricas orientadas al tamaño y orientadas a la función. 3. Describir las ventajas y desventajas de las Métricas orientadas al tamaño y orientadas a la función. 	<ol style="list-style-type: none"> a. Implementar métricas orientadas al tamaño. [1, 3] b. Implementar métricas orientadas a la función. [2, 3] c. Estimar los beneficios de aplicar tanto las métricas orientadas al tamaño como las orientadas a la función. [1, 2, 3] 	

Figura 32. Actividades de enseñanza-aprendizaje

La segunda agrupación se realiza sobre las actividades de enseñanza-aprendizaje realizadas y se denominan unidades de aprendizaje

✓ **Identificación de Unidades de Aprendizaje:** Para este proceso se toman como referente principal las actividades de enseñanza-aprendizaje identificadas para la asignatura. Estas actividades se asocian por afinidad para conformar las unidades de aprendizaje las cuales constituyen un elemento de mayor nivel en la estructura modular. Con la conformación de las unidades de aprendizaje se empieza a evidenciar la flexibilidad que ofrece el diseño curricular propuesto, ya que con base en los elementos básicos (actividades de enseñanza-aprendizaje)

se puede estructurar de forma diferente las unidades de aprendizaje de acuerdo a la orientación que se desee dar al desarrollo del proceso de formación.

Los principios de análisis funcional presentes para la identificación de Unidades de aprendizaje son presentar una estructura gramatical uniforme, mantener una relación causa-consecuencia, utilizar verbos activos (reales, medibles, y evaluables)

En la *Figura 33* Se muestra un agrupamiento de actividades en unidades de aprendizaje.

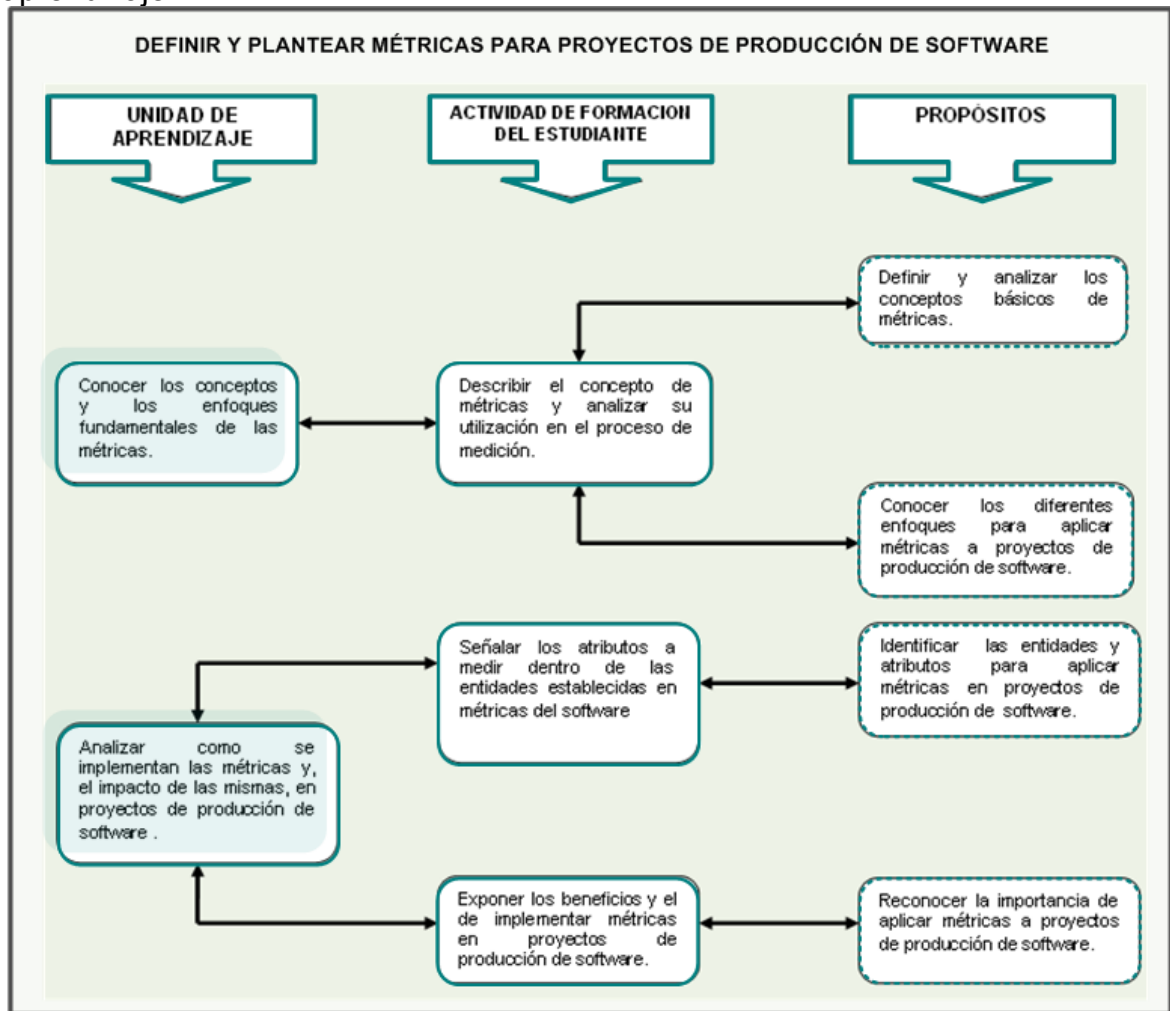


Figura 33. Agrupación en unidades de aprendizaje

La tercera agrupación y el mayor nivel que se alcanza es el módulo de formación formado por múltiples unidades de aprendizaje afines entre sí.

✓ **Los módulos de formación** son los elementos particulares del diseño curricular de la formación basada en competencias, un módulo es una unidad autónoma con sentido propio que, al mismo tiempo, se articula con los distintos módulos que integran la estructura curricular.

Se basa en una concepción del proceso de enseñanza-aprendizaje válida y coherente con el desarrollo de competencias. El aprendizaje se entiende como un proceso de adquisición de significados que tiende a la permanente vinculación entre los contenidos de la formación y todas sus dimensiones: integra habilidades, destrezas, conocimientos y actitudes.

En la conformación de los módulos de formación se emplea como principios del análisis funcional mantener una relación causa-consecuencia y mantener una secuencia lógica.

Con la identificación de los módulos concluye la etapa de estructuración modular, teniendo como efecto un conjunto de cuatro niveles básicos de desagregación como se observa en la Figura 34.

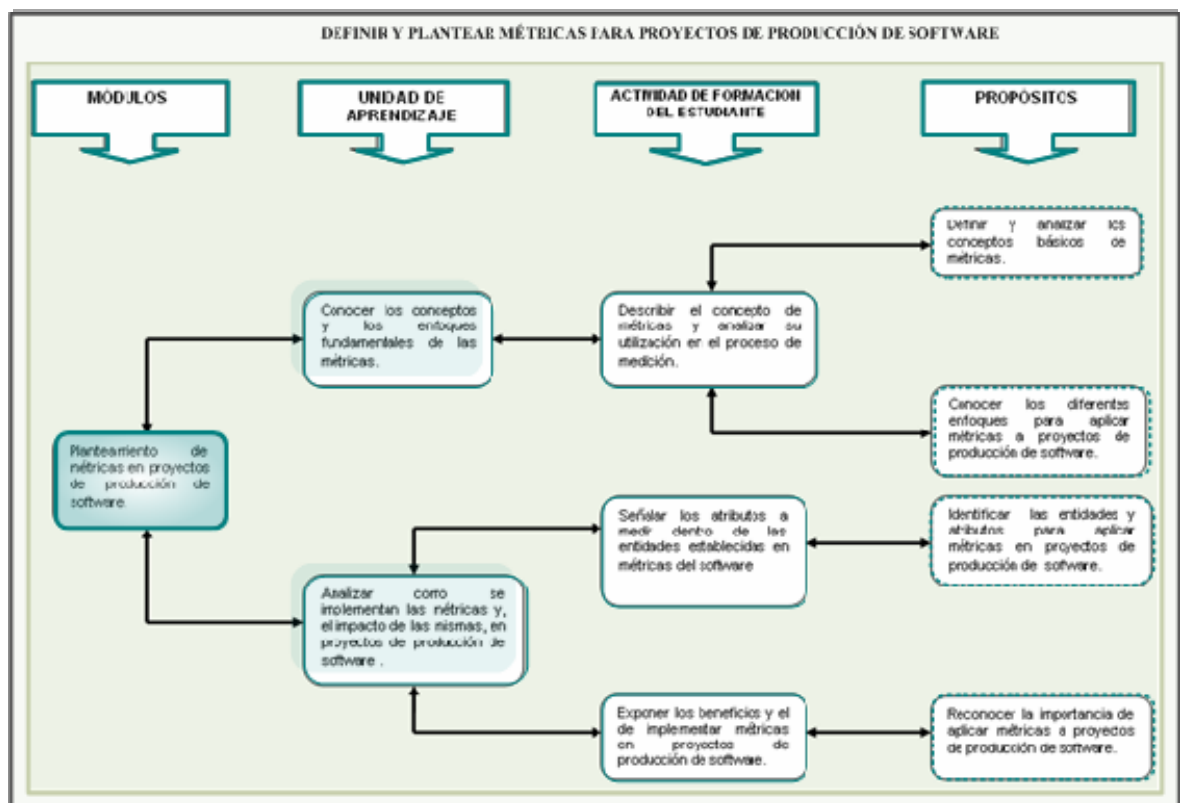


Figura 34. Agrupación en módulos de formación y estructura modular

La estructura modular de la asignatura se interpreta de izquierda a derecha como las acciones a realizar para el cumplimiento del nivel anterior y de derecha a izquierda provee la finalidad por la que realizamos las diferentes acciones en cada nivel.

Horizontalmente la estructuración modular muestra el principio de causa-consecuencia y verticalmente la secuencialidad de la asignatura, así se conservan los principios metodológicos a través de todas las etapas de aplicación y a su vez a través de todos los productos desarrollados para cada una de ellas.

3.2.5. Establecimiento de la planeación curricular

a. Describir mediante la planeación curricular las actividades de formación desarrolladas con base en las estrategias.

Esta es la última etapa de la propuesta metodológica, ofrece una visión ampliada y a la vez detallada de la asignatura. Los elementos que integran la planeación permiten dar respuesta a preguntas acerca de cómo enseñar, con qué, cuando y como evaluar. La planeación contiene los criterios, contenidos, métodos y medios didácticos, las evidencias de aprendizaje y las técnicas e instrumentos de evaluación que darán soporte a toda la propuesta metodológica y que permitirán construir las acciones tangibles y concretas para el desarrollo de la misma.

En la figura 35 se observa la primera parte del formato de la planeación curricular y se describen a continuación cada uno de los elementos que la componen:

DEFINIR Y PLANTEAR MÉTRICAS PARA PROYECTOS DE PRODUCCIÓN DE SOFTWARE		PLANEACIÓN CURRICULAR		Versión Final
MÓDULO DE FORMACIÓN		<i>Planteamiento de métricas en proyectos de producción de software.</i>		
UNIDAD DE APRENDIZAJE		<i>Conocer los conceptos y los enfoques fundamentales de las métricas.</i>		
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE		<i>Describir el concepto de métricas y analizar su utilización en el proceso de medición en proyectos de producción de software.</i>		
DURACIÓN DE LA ACTIVIDAD				
CRITERIOS	CONTENIDO		METODOLOGÍA	
	CONCEPTUALES (SABER)	PROCEDIMENTALES (HACER)	Métodos didácticos	Medios didácticos
Conocer los diferentes enfoques para aplicar métricas a proyectos de producción de software	A. Definir las Métricas orientadas al tamaño.	D. Implementar métricas orientadas al tamaño. (A, C)	a. Análisis e interpretación de lectura	Textos: • Contenido en formato ... Gráficos: • Diagrama represente... que Animaciones: • Animación que ilustre ... Simulador: • Simulador que permite realizar el cálculo de métricas orientadas al tamaño y orientadas a la función.
	B. Definir las Métricas orientadas a la función.	E. Implementar métricas orientadas a la función. (B, C)	b. Resumen	
	C. Describir las ventajas y desventajas de las Métricas orientadas al tamaño y orientadas a la función.	F. Estimar los beneficios de aplicar tanto las métricas orientadas al tamaño como las orientadas a la función. (A, B, C)	c. Taller de ejercicios d. Simulaciones e. Consulta	

Figura 35. Planeación curricular primera parte.

- El encabezado que contiene el módulo de formación, la unidad de aprendizaje y la actividad de enseñanza-aprendizaje a la que hace referencia la planeación.
- Los criterios relacionados con la actividad que se deducen de los propósitos establecidos para un grupo de saberes y haceres asociados debido a su afinidad temática como se vio en la relación propósitos-contenidos. Estos criterios definen las metas de la actividad.
- Los contenidos que son de dos tipos, conceptuales (saberes) y procedimentales (haceres) y que se encuentran debidamente relacionados entre si.
- Los métodos y medios didácticos comprenden la metodología a seguir para desarrollar los contenidos de la asignatura. Esta metodología esta totalmente acorde con el objetivo propuesto que se encuentra enfocado en lograr el aprendizaje significativo por parte del estudiante, es por esto que tanto los métodos y los medios han sido establecidos para soportar el modelo de estilos de aprendizaje de Felder y Silverman.

En esta parte de la metodología el docente puede, según su criterio, establecer algunos métodos o medios adicionales a los determinados si con el tiempo lo creyera necesario. Los métodos acá mencionados han sido tomados de la siguiente tabla.

Tabla 16. Estrategias y técnicas de enseñanza-aprendizaje⁶⁷

ESTRATEGIA	TÉCNICA
<i>Aprendizaje interactivo</i>	<ul style="list-style-type: none"> • Presentación participativa • Exposición • Conferencia por un experto • Entrevista • Panel • Debate • Formulación de preguntas • Seminario
<i>Aprendizaje individual</i>	<ul style="list-style-type: none"> • Consulta • Reporte • Elaboración de ensayo • Tareas individuales • Resumen • Laberintos de acción • Análisis e interpretación de lectura • Análisis y resolución de problemas

⁶⁷ Recopilación realizada por Dorys Consuelo Ramírez Prada, Lilia Yarley Estrada Díaz y Dania Rubiela Verjel Arenas.

ESTRATEGIA	TÉCNICA	
<i>Aprendizaje Colaborativo</i>	<ul style="list-style-type: none"> • Consulta • Resumen • Análisis e interpretación de lectura • Análisis y resolución de problemas • Taller de ejercicios • Exposición • Técnica del rompecabezas 	<ul style="list-style-type: none"> • Investigación • Proyecto • Panel • Debate • Seminario • Concurso • Juego de roles • Lluvia de ideas • Tutorial
<i>Aprendizaje por descubrimiento</i>	<ul style="list-style-type: none"> • Práctica de laboratorio • Proyecto • Investigaciones 	
<i>Aprendizaje basado en problemas</i>	<ul style="list-style-type: none"> • Análisis de ejercicios • Resolución y análisis de ejercicios • Solución de casos 	<ul style="list-style-type: none"> • Análisis y resolución de problemas • Simulaciones
<i>Aprendizaje significativo</i>	<ul style="list-style-type: none"> • Analogía • Resumen • Organizador previo • Ilustraciones • Mapas conceptuales 	<ul style="list-style-type: none"> • Redes semánticas • Mapa mental • Diagramas • Lluvia de ideas • Formulación de preguntas

En la figura 36 se puede observar la segunda parte del formato de la planeación curricular y se describen a continuación cada uno de los elementos que la componen:

EVIDENCIAS DE APRENDIZAJE	EVALUACIÓN	
CONOCIMIENTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Conoce los enfoques para aplicar métricas en un proyecto de producción de software (A, B, D, E)	<ol style="list-style-type: none"> 1. Mesa redonda 2. Prueba o examen 3. Actividades complementarias 	<ol style="list-style-type: none"> a. Cuestionario informal (1) b. Test (2) c. Resumen (1, 3)
Comprende el concepto de métricas orientadas al tamaño (A, D)	<ol style="list-style-type: none"> 1. Exposición 2. Debate 	<ol style="list-style-type: none"> a. Resumen (1, 2, 3) b. Cuestionario informal (3)
DESEMPEÑO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Describe las métricas orientadas al tamaño (A, D)	<ol style="list-style-type: none"> 1. Exposición 2. Prueba o examen 	<ol style="list-style-type: none"> a. Preguntas informales (1) b. Test (2)
PRODUCTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Calcula métricas orientadas al tamaño (A, D)	<ol style="list-style-type: none"> 1. Prueba o Examen. 2. Actividades Complementarias. 3. Exposición. 	<ol style="list-style-type: none"> a. Cuestionario (1) b. Taller de Problemas (1, 2) c. Ejercicios (1, 2) d. Exposición (3)
Calcula métricas orientadas a la función (B, E)		

Figura 36. Elementos planeación curricular

- Evidencias de aprendizaje, que pueden ser de tres tipos:

Evidencias de conocimiento que establecen los condiciones de conocimiento y comprensión necesarios para el cumplimiento del criterio de la actividad en cuestión.

Evidencias de desempeño que corresponden a los procedimientos y habilidades que debe desarrollar el estudiante con el fin de fortalecer el proceso de aprendizaje.

Evidencias de producto que fusionando las condiciones de conocimiento y comprensión con los procesos y habilidades permiten obtener un resultado tangible de un proceso como evidencia de una acción realizada.

Se espera que las evidencias de aprendizaje se complementen entre si en lo posible. Es por esto que se establecieron evidencias de aprendizaje de por lo menos dos tipos para cada criterio en el desarrollo de esta propuesta.

- Técnicas e instrumentos de evaluación, relacionados mutuamente según la afinidad que existe entre unos instrumentos y las características de unas técnicas más otros.

Con las técnicas e instrumentos de evaluación se busca establecer estrategias y herramientas que permitan demostrar que el aprendizaje significativo ha sucedido en el estudiante.

Se debe tener en cuenta el criterio(meta) que se desea el estudiante alcance, los contenidos, los métodos y medios didácticos utilizados para presentar el contenido y las evidencias que se desean demostrar para elegir la técnica y el instrumento de evaluación adecuados.

Las técnicas e instrumentos planteados en esta etapa de la propuesta han sido extraídos de la siguiente tabla:

Tabla 17. Técnicas e instrumentos de evaluación⁶⁸

TÉCNICA	INSTRUMENTOS
Observación	<ul style="list-style-type: none"> • Lista de verificación • Ficha de observación
Entrevista	<ul style="list-style-type: none"> • Cuestionario formal • Cuestionario informal
Debate	<ul style="list-style-type: none"> • Anecdótico • Resumen • Toma de notas
Mesa Redonda	<ul style="list-style-type: none"> • Anecdótico • Toma de notas • Resumen • Cuestionario informal
Exposición	<ul style="list-style-type: none"> • Lista de verificación • Informe • Anecdótico • Toma de notas • Resumen • Relatoría • Preguntas informales

⁶⁸ Recopilación realizada por Dorys Consuelo Ramírez Prada, Lilia Yarley Estrada Díaz y Dania Rubiela Verjel Arenas.

TÉCNICA	INSTRUMENTOS
<i>Ensayo</i>	<ul style="list-style-type: none"> • Ensayo • Lista de verificación
<i>Prueba o examen</i>	<ul style="list-style-type: none"> • Cuestionario • Taller de problemas • Ejercicios • Test
<i>Mapa conceptual</i>	<ul style="list-style-type: none"> • Mapa conceptual
<i>Diagramas de información</i>	<ul style="list-style-type: none"> • Mapa mental • Cuadro sinóptico • Esquema • Redes semánticas • Algoritmo • Panel de información • Tablas
<i>Proyectos</i>	<ul style="list-style-type: none"> • Informe • Productos asociados • Portafolio
<i>Actividades Complementarias</i>	<ul style="list-style-type: none"> • Relatorías • Resumen • Ejercicios • Taller de problemas • Visitas técnicas • Portafolio
<i>Seguimiento de Actividades</i>	<ul style="list-style-type: none"> • Encuestas • Bitácoras • Registro de actividades • Anecdótico • Auto evaluación • Coevaluación
<i>Práctica de laboratorio</i>	<ul style="list-style-type: none"> • Informe • Lista de chequeo • Cuestionario • Algoritmo • Anecdótico

4. TERCERA FASE: Diseño y Producción de un Objeto de Aprendizaje

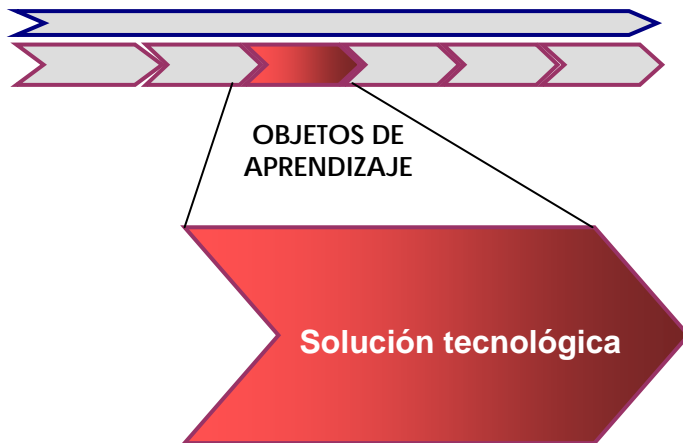


Figura 37. Fase 3 - Diseño y Producción de un Objeto de Aprendizaje

En concordancia con la visión del equipo de profesionales pertenecientes al Proyecto FONDEF “**Aprendiendo con Repositorios de Objetos de aprendizaje**”, APROA, este capítulo expone la metodología utilizada en la generación y encapsulamiento del objeto de aprendizaje.⁶⁹

4.1. CONCEPTO Y CARACTERÍSTICAS DE UN OBJETO DE APRENDIZAJE

Según el Proyecto “Aprendiendo con Objetos de Aprendizaje”, se puede definir este objeto de aprendizaje, OA, como **una composición digital basada en un objetivo de enseñanza que posee un contenido, una aplicación, una evaluación, algunos vínculos de profundización del contenido y un metadato.**

De igual forma, este objeto de aprendizaje posee algunas características que garantizan su eficiencia como tal, estas son:

- a. Es **autocontenido**, es decir, tiene la capacidad de dar cumplimiento al objetivo propuesto. Solamente incorpora vínculos hacia documentos digitales que profundizan y/o complementan algunos conceptos del contenido.
- b. Es **interoperable**, es decir, cuenta con una estructura basada en un lenguaje de programación XML, y con un estándar internacional de interoperabilidad (SCORM para efectos del proyecto), que garantiza su utilización en plataformas con distintos ambientes de programación.

⁶⁹ Proyecto FONDEF “Aprendiendo con Repositorios de Objetos de Aprendizaje”, APROA. *Manual de buenas practicas para el desarrollo de objetos de aprendizaje.* Versión 1.

- c. Es **reutilizable**, es decir, debido a que pretende dar cumplimiento a un objetivo específico, puede ser utilizado por diversos educadores bajo distintos contextos de enseñanza.
- d. Es **durable y actualizable** en el tiempo, es decir, esta respaldado por una estructura (Repositorio) que permite, en todo momento, incorporar nuevos contenidos y/o modificaciones a los existentes. De esta forma se evita la obsolescencia.
- e. Es de **fácil acceso y manejo** para los alumnos, es decir, la misma estructura de respaldo facilita a los alumnos el acceso al objeto así como el manejo de éste en el aprendizaje.
- f. Es **secuenciable** con otros objetos, es decir, la estructura de respaldo posibilita la secuenciación del objeto con otros bajo un mismo contexto de enseñanza.
- g. Es **breve y sintetizado**, es decir, alcanza el objetivo propuesto mediante la utilización de los recursos (textos, imágenes, diagramas, figuras, videos, animaciones, otros) mínimos necesarios, sin extremar en la saturación de recursos y en la carencia de los mismos.
- h. Incorpora la **fuentes de los diversos recursos** de autoría utilizados en el contenido de enseñanza, de esta forma se asegura que el objeto cumpla con las leyes de derecho de autor existentes.



Figura 38. Estructura de un objeto de aprendizaje⁷⁰

⁷⁰ Proyecto FONDEF “Aprendiendo con Repositorios de Objetos de Aprendizaje”, APROA. *Manual de buenas practicas para el desarrollo de objetos de aprendizaje*. Versión 1.

4.1.1. Sobre el nombre del objeto de aprendizaje

El nombre del objeto de aprendizaje representa de forma clara y simple el contenido tratado, evitando la ambigüedad en la idea. El nombre describe completamente el tema al que hace referencia, “**Objeto de Aprendizaje para Métricas del Software**”.

4.1.2. Sobre el objetivo del objeto de aprendizaje

Según el nivel de globalidad del objetivo propuesto por un objeto de aprendizaje, es posible diferenciar tres tipos de objetos:

- Objeto de aprendizaje **global** (OAg), aquel que presenta un objetivo general, que puede ser la base para el desarrollo de objetos con objetivos más específicos.
- Objeto de aprendizaje **temático** (OAt), aquel que presenta un objetivo orientado a un tema específico, que puede permitir el desarrollo de objetos aún más específicos.
- Objeto de aprendizaje **específico** (OAe), aquel que presenta un objetivo orientado a un aspecto específico de un tema, siendo el escalafón más alto en especificidad de objetivos.

Este objeto de aprendizaje se ubica en un **objeto de aprendizaje de tipo temático (OAt)**, ya que fue diseñado para un tema específico de la Ingeniería del Software, y permite el desarrollo de objetos aun más específicos ya que se puede profundizar mas en cada una de las áreas comprendidas en las Métricas del Software.

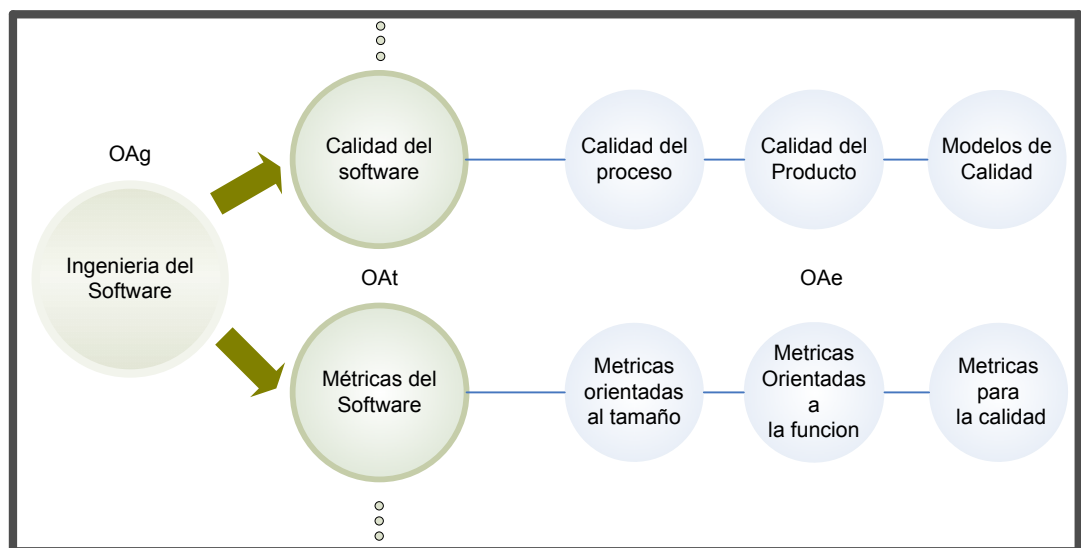


Figura 39. Objetos temáticos y específicos derivados de un objeto global

4.1.3. Sobre el contenido del objeto de aprendizaje

Para cumplir el objetivo planteado, se hizo uso de diversos recursos digitales, tales como textos, imágenes, diagramas, gráficos, figuras, videos, narración, animaciones y simulaciones, los cuales se encuentran organizados metodológicamente asegurando así un óptimo aprendizaje por parte del alumno junto con la capacidad de síntesis del objeto.

Para llevar a cabo el desarrollo del contenido del objeto, se implementaron **plantillas** que permitieron facilitar el diseño del mismo, economizando tiempo y recursos en la generación de objetos, y facilitando la secuenciación de estos bajo un mismo contexto de enseñanza. El uso de plantillas no solamente favorece el trabajo de diseño del objeto, sino también el proceso de comprensión del contenido por parte de los mismos alumnos, quienes dispondrán de objetos con un formato estándar.

4.1.4. Sobre la aplicación del objeto de aprendizaje

Este objeto de aprendizaje tiene la capacidad de cerrar el proceso de enseñanza de un objetivo por si solo, a través de la incorporación de una aplicación ó experiencia que permite al alumno aplicar el conocimiento aprendido, ya sea en ambientes reales (aula de clase) o bajo ambientes simulados (aula virtual).

La finalidad del objeto es guiar al alumno en los pasos de la actividad que desempeñara en la práctica, contando con el acompañamiento participativo de un docente que vigile el alcance del objetivo planteado.

Si el tema tratado en un objeto no permite el desarrollo de una aplicación, bastará con incorporar la experiencia del profesor en la materia, la que podrá explicarse mediante un estudio de caso real o simulado.

4.1.5. Sobre la evaluación del objeto de aprendizaje

Finalmente, el objeto cierra su ciclo de enseñanza con una evaluación, que guía al alumno en las preguntas facilitando el trabajo autónomo.

La plataforma **e-escen@riUIS** cuenta con diversos métodos de evaluación, tales como preguntas de alternativas, desarrollo de términos pareados, completado de oraciones, desarrollo de cálculos matemáticos, la plataforma brinda soporte en cuanto a herramientas de evaluación al objeto de aprendizaje siendo la labor del profesor escoger aquellas que le permitan asegurar una correcta evaluación del contenido aprendido por el alumno.

4.1.6. Sobre los vínculos de profundización del contenido

Al igual que todos los objetos de aprendizaje diseñados bajo esta metodología, incorpora vínculos a información digital y referencias bibliográficas que permitirán profundizar y complementar el contenido del objeto.

4.1.7. Sobre la declaración de autoría del contenido

El contenido presentado por el objeto de aprendizaje declara la autoría del o los profesores que participaron en la generación del objeto. De igual manera, se citan las fuentes de los textos, imágenes, gráficos, videos, o cualquier otro recurso incorporado que no haya sido preparado por el docente.

4.1.8. Proceso de generación de un objeto de aprendizaje

En este se integran en forma organizada todos los componentes que conforman el objeto de aprendizaje y el proceso que conlleva su diseño e implementación.

Los materiales didácticos deben permitir:

- Aprender a aprender
- Construir aprendizajes
- Establecer relaciones entre conocimientos
- Facilitar la autoevaluación y el control del proceso de aprendizaje
- Involucrar y motivar al estudiante.

4.2. ETAPA DE INICIO

Para empezar se realiza la especificación de requisitos, ya que es un requisito fundamental para cumplir con los objetivos que definen el esquema lógico de las funcionalidades del sistema. Asimismo se lleva a cabo la descripción del producto y su finalidad.

Definición de requerimientos en términos de funcionalidad

Esta definición se realiza por la necesidad de especificar los requerimientos del producto, recopilando esto con el equipo de trabajo, con base en las especificaciones de los expertos docentes, obteniendo de esta manera los siguientes requerimientos:

- Descripción de la temática métricas del software, especificando los conceptos básicos necesarios para su comprensión, las clases de métricas

existentes, entidades a medir dentro de un proyecto de software y la implicación de las métricas dentro de dichos proyectos.

- Cada una de las temáticas correspondientes a métricas del software tendrá material de apoyo necesario para el aprendizaje de estas, este material esta conformado por diferentes tipos de ayudas como documentos de soporte en formato pdf, videos, audio y gráficos.
- Selección del material de apoyo según el tipo de ayuda(s) definidas para soportar el aprendizaje de cada temática.
- Implementación de las ayudas de video mediante el programa de Macromedia Flash 8, que permite un ambiente más amigable al estudiante al momento del aprendizaje.
- Creación de un aplicativo en el lenguaje de programación Java Script para la temática clasificación de las métricas del software, la cual permite al estudiante aplicar los conceptos adquiridos y también observar los pasos necesarios para el cálculo de distintos tipos de métricas del software.

Definición del alcance del objeto de aprendizaje realizado para la temática “métricas del software”

El objeto de aprendizaje realizado para la asignatura “Ingeniería del Software”, en la temática “Métricas del software”, del programa académico de Ingeniería de sistemas e informática de la UIS, beneficiará a los estudiantes como a los profesores de dicha asignatura, permitiendo el logro de aprendizajes significativos y personalizados según la metodología basada en competencias, haciendo uso de las nuevas tecnologías.

El docente contará con una ayuda práctica para sus alumnos haciendo su labor educativa más versátil y amena, logrando cumplir con los propósitos planteados en el diseño instruccional, utilizando las estrategias definidas en la planeación curricular del mismo. Así el docente puede reforzar los niveles de conocimiento que los alumnos poseen en forma personalizada.

Con relación al estudiante o usuario final de este medio didáctico de educación, se espera que éste, en forma mas variable, pueda acoplarse con las ideas que el docente desea proporcionarle, mostrando así un mejor aprendizaje y avances significativos en su rendimiento académico, ya que en el desarrollo de este proyecto se tuvieron en cuenta estrategias donde el estudiante puede aprender a su estilo y a su ritmo, puntos que en este proyecto se rescatan y aplican ya que en la educación tradicional no se implementan.

Además este medio didáctico en línea entre sus alcances permite al estudiante en un entorno de navegación cómodo realizar seguimiento a las actividades de aprendizaje llevadas a cabo.

4.3. ETAPA DE ELABORACIÓN

En esta etapa se presenta la arquitectura del sistema y los diagramas de casos de uso de uso.

LENGUAJE DE MODELADO UNIFICADO (DIAGRAMAS DE CASOS DE USO)

DIAGRAMAS DE CASOS DE USO:

En el proceso de desarrollo del Objeto de Aprendizaje se realizaron los siguientes diagramas de casos de uso para representar la funcionalidad completa del sistema, mostrando su interacción con los agentes externos.

4.3.1. Casos de uso – documento soporte

❖ ELEMENTOS:

➤ Actores

- Estudiante (Emplea el sistema para obtener información).
- Profesor (Se apoya en el sistema para complementar los contenidos de su clase y recibe información del mismo).
- Desarrollador (Recibe información para diseñar y crear el sistema).

➤ Sistemas

- Plataforma (Integra el sistema a su ambiente).

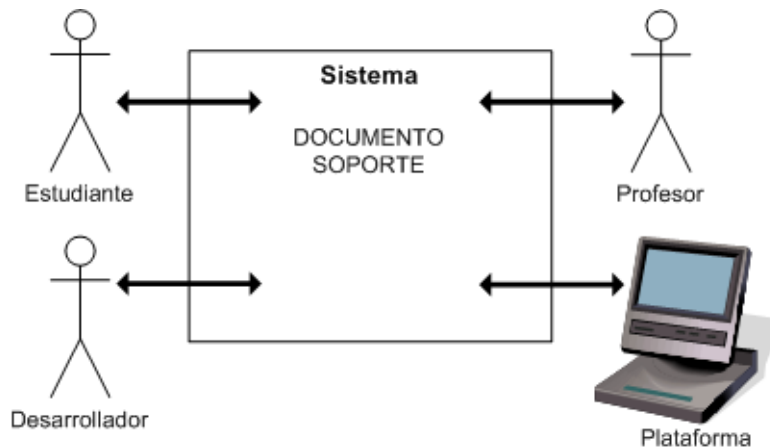


Figura 40. Diagrama de Actores y Sistema - Documento soporte

❖ **CASOS DE USO**

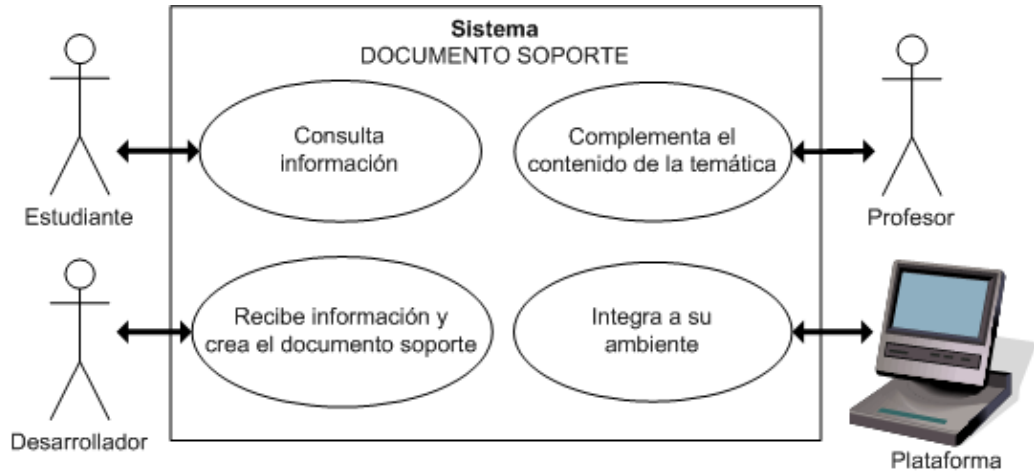


Figura 41. Diagrama de Casos de Uso – Documento soporte

❖ **DIAGRAMAS DE CASOS DE USO**

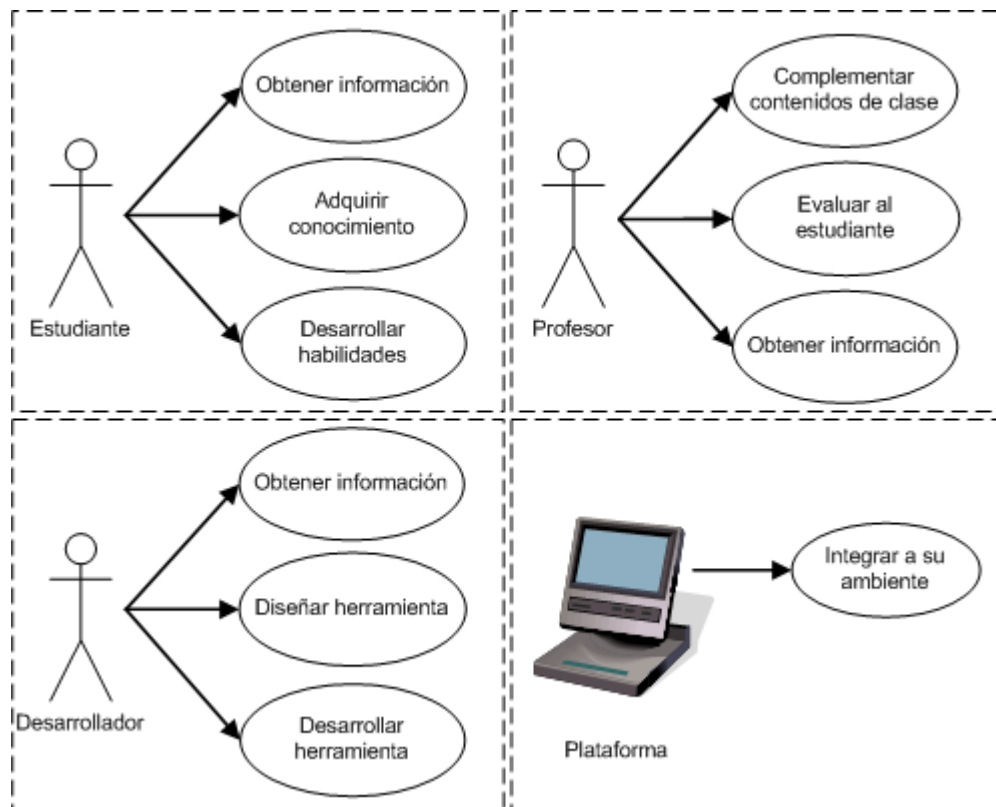


Figura 42. Diagrama de Casos de Uso para Actores y Sistema – Documento soporte

❖ DESCRIPCIÓN DE LOS CASOS DE USO

Nombre:	Consulta información
Descripción: Permite consultar la información que contiene el documento soporte sobre la temática en cuestión.	
Actores: Estudiante	
Precondiciones: El Estudiante debe haber ingresado a la plataforma que aloja el sistema.	
Flujo Normal: <ol style="list-style-type: none"> 1. El estudiante ingresa a la plataforma. 2. El estudiante se ubica en la plantilla de la asignatura Ingeniería del software. 3. El estudiante escoge la temática a la que pertenece el documento soporte. 4. El estudiante abre el documento soporte. 5. El estudiante recibe información sobre la temática a través de la lectura del documento soporte. 	
Poscondiciones: El actor analiza la información obtenida del documento soporte.	

Nombre:	Complementa el contenido de la temática
Descripción: Permite complementar el contenido de la temática expuesto en el aula.	
Actores: Profesor	
Precondiciones: El profesor debe tener un objeto de aprendizaje relacionado con su asignatura en la plataforma y debe preparar la información que va a suministrar a los estudiantes.	
Flujo Normal: <ol style="list-style-type: none"> 1. El estudiante ingresa a la plataforma por indicación del profesor. 2. El profesor guía al estudiante sobre cómo ingresar a la plantilla de la asignatura Ingeniería del software y a la temática que se está tratando. 3. El profesor explica al estudiante cómo acceder a la opción documento soporte de la temática en cuestión. 4. El profesor explica cómo se ha estructurado la información en la herramienta (documento soporte). 5. El profesor complementa la explicación de la temática una vez el estudiante haya leído el documento soporte. 	

Poscondiciones: El profesor evalúa la comprensión de la información obtenida del documento soporte por parte de los estudiantes.	
Nombre:	Recibe información y crea el documento soporte
Descripción: Permite desarrollar herramientas que dan soporte al proceso de enseñanza-aprendizaje.	
Actores: Desarrollador	
Precondiciones: El desarrollador debe tener los conocimientos previos necesarios sobre la temática a tratar en la herramienta y sobre los recursos necesarios para construirla.	
Flujo Normal: <ol style="list-style-type: none"> 1. El desarrollador adquiere y organiza la información que contendrá el documento soporte. 2. El desarrollador crea el documento soporte. 3. El desarrollador implanta la herramienta (documento soporte) en la plataforma. 4. El desarrollador se cerciora del correcto funcionamiento de la herramienta. 	
Poscondiciones: El desarrollador evalúa la comprensión del estudiante frente al documento soporte con el fin de realizar correcciones o posibles mejoras.	

Nombre:	Integra a su ambiente
Descripción: Permite acceder a los objetos de aprendizaje y proporciona el entorno de interacción de dichos objetos con sus usuarios.	
Actores: Plataforma	
Precondiciones: Todos los recursos necesarios para acceder a la plataforma deben estar disponibles.	
Flujo Normal: <ol style="list-style-type: none"> 1. Se carga la plataforma desde un Terminal y se accede a sus contenidos. 2. La plataforma adapta las plantillas a su entorno. 3. La plataforma carga los componentes de la plantilla. 4. La plataforma permite al usuario el acceso al documento soporte. 	

Poscondiciones:

La plataforma guarda un registro del ingreso a la misma y de las actividades que se realizaron por parte del usuario en ella.

❖ **DIAGRAMA DE CASOS DE USO GENERAL**

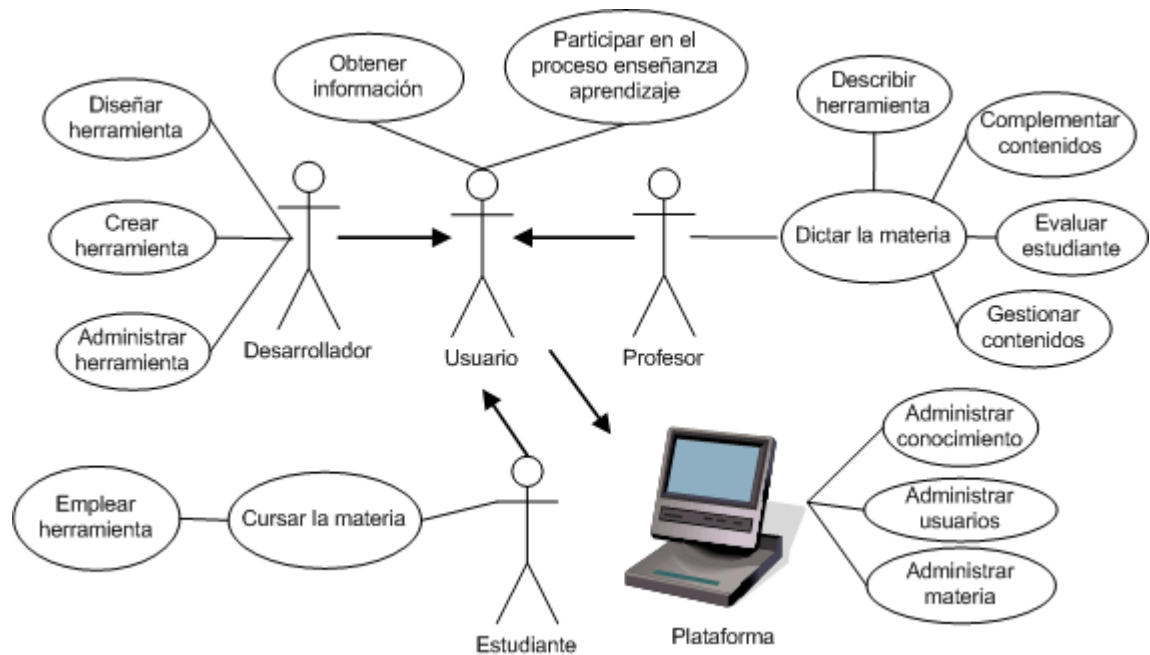


Figura 43. Diagrama de Casos de Uso General – Documento soporte

4.3.2. Casos de uso - audio

❖ **ELEMENTOS:**

➤ **Actores**

- Estudiante (Emplea el sistema para obtener información).
- Profesor (Se apoya en el sistema para complementar los contenidos de su clase y recibe información del mismo).
- Desarrollador (Recibe información para diseñar y crear el sistema).

➤ **Sistemas**

- Plataforma (Integra el sistema a su ambiente).

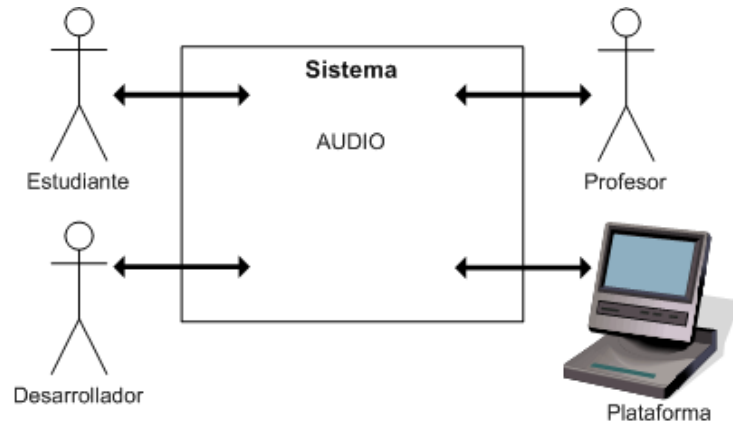


Figura 44. Diagrama de Actores y Sistema – Audio

❖ **CASOS DE USO**

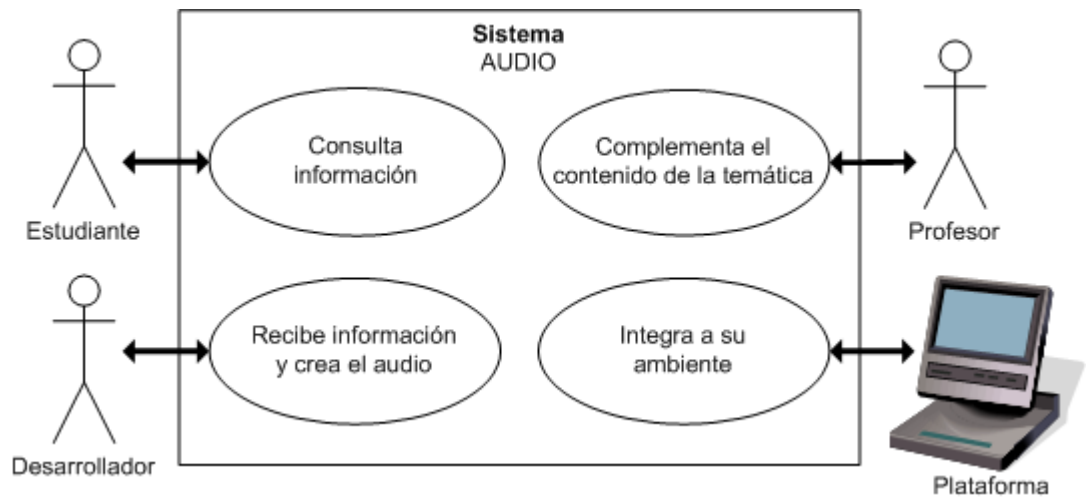


Figura 45. Diagrama de Casos de Uso - Audio

❖ **DIAGRAMAS DE CASOS DE USO**

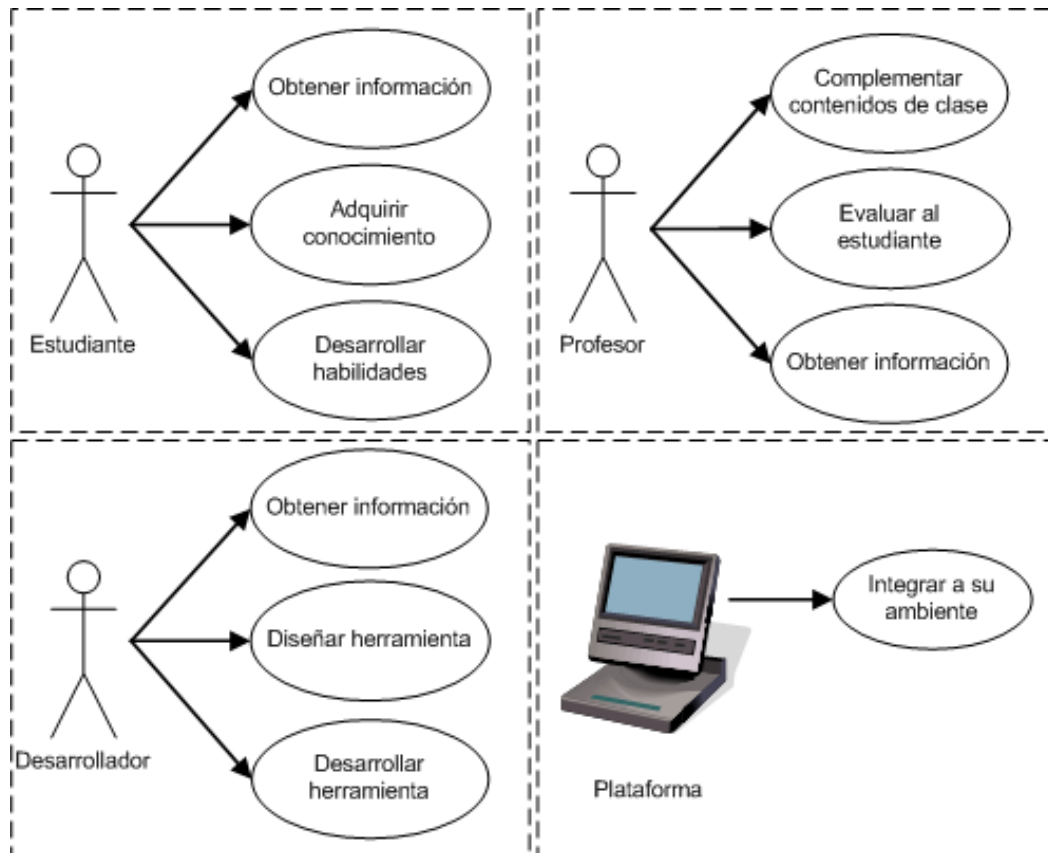


Figura 46. Diagrama de Casos de Uso para Actores y Sistema – Audio

❖ **DESCRIPCIÓN DE LOS CASOS DE USO**

Nombre:	Consulta información
Descripción:	Permite acceder a la información que contiene el audio sobre la temática en cuestión.
Actores:	Estudiante
Precondiciones:	El Estudiante debe haber ingresado a la plataforma que aloja el sistema.
Flujo Normal:	1. El estudiante ingresa a la plataforma.

<ol style="list-style-type: none"> 2. El estudiante se ubica en la plantilla de la asignatura Ingeniería del software. 3. El estudiante escoge la temática a la que pertenece el archivo de audio. 4. El estudiante ejecuta el archivo de audio. 5. El estudiante recibe información sobre la temática a través del audio.
Poscondiciones: El actor analiza la información obtenida del audio.

Nombre:	Complementa el contenido de la temática
Descripción: Permite complementar el contenido de la temática expuesto en el aula.	
Actores: Profesor	
Precondiciones: El profesor debe tener un objeto de aprendizaje relacionado con su asignatura en la plataforma y debe preparar la información que va a suministrar a los estudiantes.	
Flujo Normal: <ol style="list-style-type: none"> 1. El estudiante ingresa a la plataforma por indicación del profesor. 2. El profesor guía al estudiante sobre cómo ingresar a la plantilla de la asignatura Ingeniería del software y a la temática que se está tratando. 3. El profesor explica al estudiante cómo acceder al audio de la temática en cuestión. 4. El profesor enseña el uso y manejo de la herramienta audio. 5. El profesor complementa la explicación de la temática una vez se haya utilizado la herramienta. 	
Poscondiciones: El profesor evalúa la comprensión de la información obtenida del audio por parte de los estudiantes.	

Nombre:	Recibe información y crea el audio
Descripción: Permite desarrollar herramientas que dan soporte al proceso de enseñanza-aprendizaje.	
Actores: Desarrollador	
Precondiciones: El desarrollador debe tener los conocimientos previos necesarios sobre la	

temática a tratar en la herramienta y sobre los recursos necesarios para construirla.
Flujo Normal: <ol style="list-style-type: none"> 1. El desarrollador adquiere y organiza la información que contendrá el archivo de audio. 2. El desarrollador diseña el audio. 3. El desarrollador crea el audio. 4. El desarrollador implanta la herramienta (archivo de audio) en la plataforma. 5. El desarrollador se cerciora del correcto funcionamiento de la herramienta.
Poscondiciones: El desarrollador evalúa la actitud del estudiante frente al archivo de audio con el fin de realizar correcciones o posibles mejoras.

Nombre:	Integra a su ambiente
Descripción:	Permite acceder a los objetos de aprendizaje y proporciona el entorno de interacción de dichos objetos con sus usuarios.
Actores:	Plataforma
Precondiciones:	Todos los recursos necesarios para acceder a la plataforma deben estar disponibles.
Flujo Normal:	<ol style="list-style-type: none"> 1. Se carga la plataforma desde un Terminal y se accede a sus contenidos. 2. La plataforma adapta las plantillas a su entorno. 3. La plataforma carga los componentes de la plantilla. 4. La plataforma permite al usuario el acceso al archivo de audio.
Poscondiciones:	La plataforma guarda un registro del ingreso a la misma y de las actividades que se realizaron por parte del usuario en ella.

❖ DIAGRAMA DE CASOS DE USO GENERAL

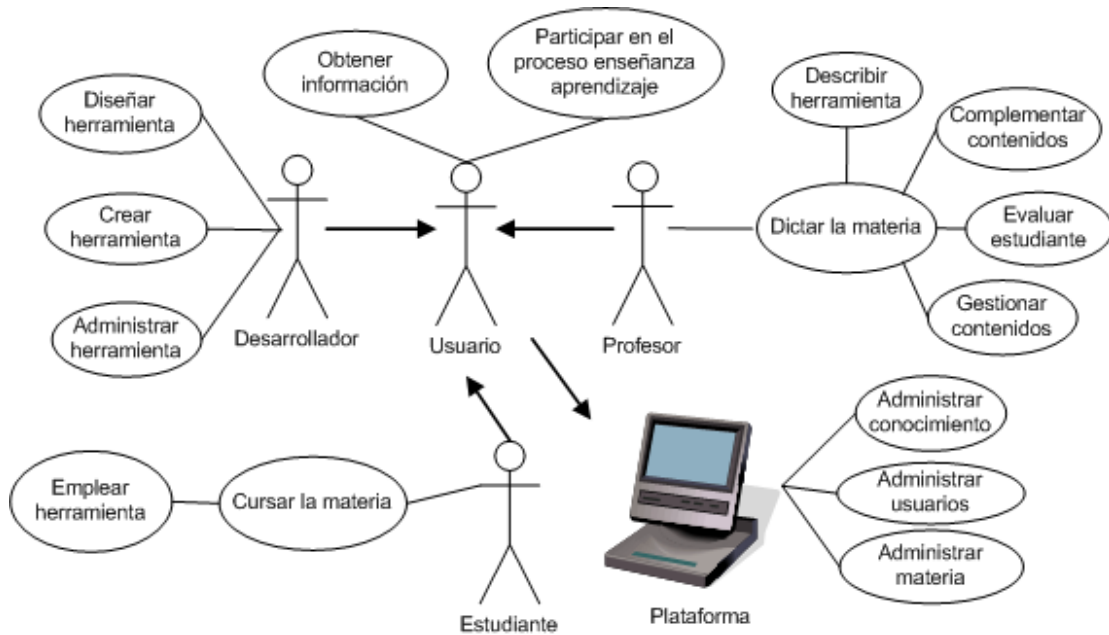


Figura 47. Diagrama de Casos de Uso General – Audio

4.3.3. Casos de uso - animación

❖ ELEMENTOS:

➤ Actores

- Estudiante (Emplea el sistema para obtener información).
- Profesor (Se apoya en el sistema para complementar los contenidos de su clase y recibe información del mismo).
- Desarrollador (Recibe información para diseñar y crear el sistema).

➤ Sistemas

- Plataforma (Integra el sistema a su ambiente).

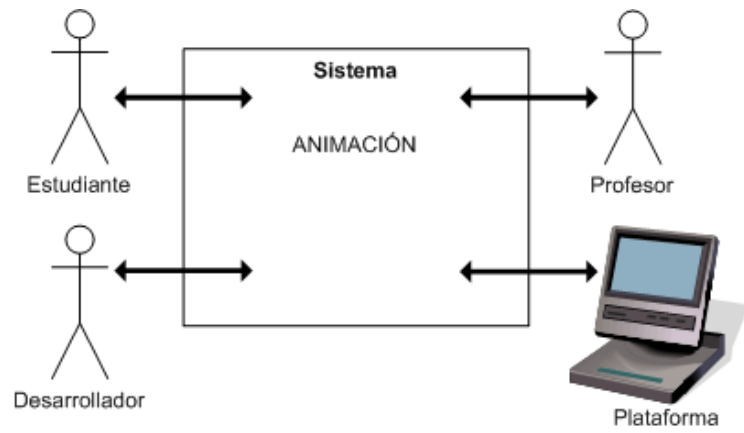


Figura 48. Diagrama de Actores y Sistema - Animación

❖ **CASOS DE USO**

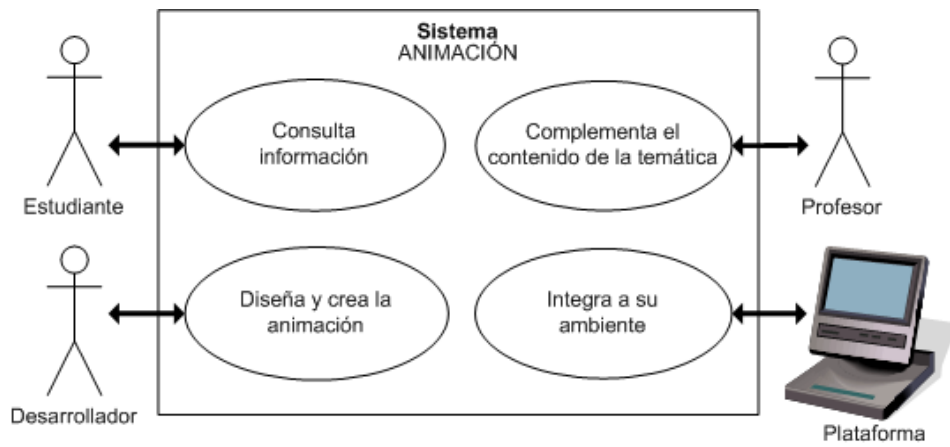


Figura 49. Diagrama de Casos de Uso - Animación

❖ **DIAGRAMAS DE CASOS DE USO**

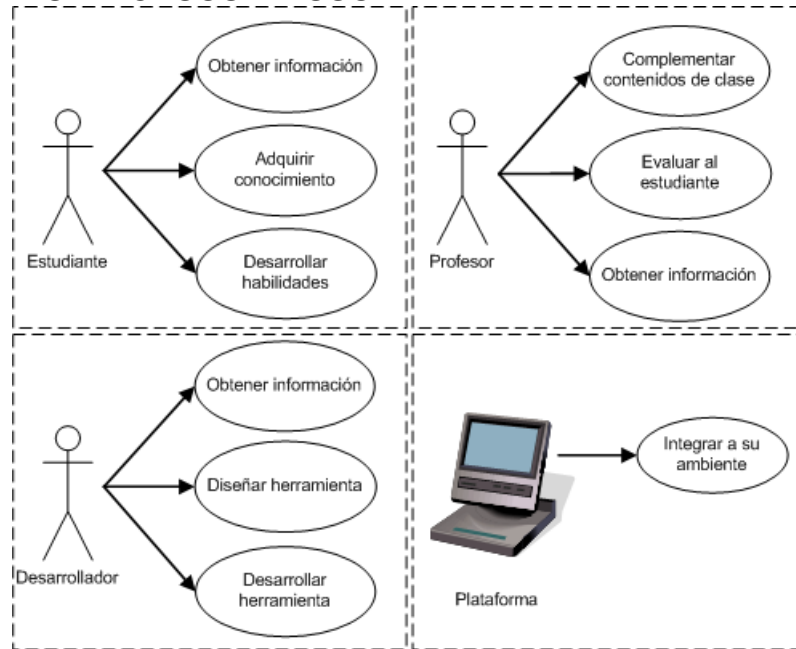


Figura 50. Diagrama de Casos de Uso para Actores y Sistema – Animación

❖ **DESCRIPCIÓN DE LOS CASOS DE USO**

Nombre:	Consulta información
Descripción:	Permite consultar la información que contiene la animación sobre la temática en cuestión.
Actores:	Estudiante
Precondiciones:	El Estudiante debe haber ingresado a la plataforma que aloja el sistema.
Flujo Normal:	<ol style="list-style-type: none"> 1. El estudiante ingresa a la plataforma. 2. El estudiante se ubica en la plantilla de la asignatura Ingeniería del software. 3. El estudiante escoge la temática a la que pertenece la animación. 4. El estudiante ejecuta la animación. 5. El estudiante recibe información sobre la temática a través de la animación.
Poscondiciones:	El actor analiza la información obtenida de la animación.

Nombre:	Complementa el contenido de la temática
Descripción: Permite complementar el contenido de la temática expuesto en el aula.	
Actores: Profesor	
Precondiciones: El profesor debe tener un objeto de aprendizaje relacionado con su asignatura en la plataforma y debe preparar la información que va a suministrar a los estudiantes.	
Flujo Normal: <ol style="list-style-type: none"> 1. El estudiante ingresa a la plataforma por indicación del profesor. 2. El profesor guía al estudiante sobre cómo ingresar a la plantilla de la asignatura Ingeniería del software y a la temática que se está tratando. 3. El profesor explica al estudiante cómo acceder a la animación de la temática en cuestión. 4. El profesor enseña el uso y manejo de la herramienta animación 5. El profesor complementa la explicación de la temática una vez se haya utilizado la herramienta. 	
Poscondiciones: El profesor evalúa la comprensión de la información obtenida de la animación por parte de los estudiantes.	

Nombre:	Diseña y crea la animación
Descripción: Permite desarrollar herramientas que dan soporte al proceso de enseñanza-aprendizaje.	
Actores: Desarrollador	
Precondiciones: El desarrollador debe tener los conocimientos previos necesarios sobre la temática a tratar en la herramienta y sobre los recursos necesarios para construirla.	
Flujo Normal: <ol style="list-style-type: none"> 1. El desarrollador adquiere y organiza la información que contendrá la animación. 2. El desarrollador diseña la animación. 3. El desarrollador crea la animación. 4. El desarrollador implanta la animación en la plataforma. 5. El desarrollador se cerciora del correcto funcionamiento de la animación. 	
Poscondiciones:	

El desarrollador evalúa la actitud del estudiante frente a la animación con el fin de realizar correcciones o posibles mejoras.

Nombre:	Integra a su ambiente
Descripción:	Permite acceder a los objetos de aprendizaje y proporciona el entorno de interacción de dichos objetos con sus usuarios.
Actores:	Plataforma
Precondiciones:	Todos los recursos necesarios para acceder a la plataforma deben estar disponibles.
Flujo Normal:	<ol style="list-style-type: none"> 1. Se carga la plataforma desde un Terminal y se accede a sus contenidos. 2. La plataforma adapta las plantillas a su entorno. 3. La plataforma carga los componentes de la plantilla. 4. La plataforma permite al usuario el acceso a la animación.
Poscondiciones:	La plataforma guarda un registro del ingreso a la misma y de las actividades que se realizaron por parte del usuario en ella.

❖ DIAGRAMA DE CASOS DE USO GENERAL

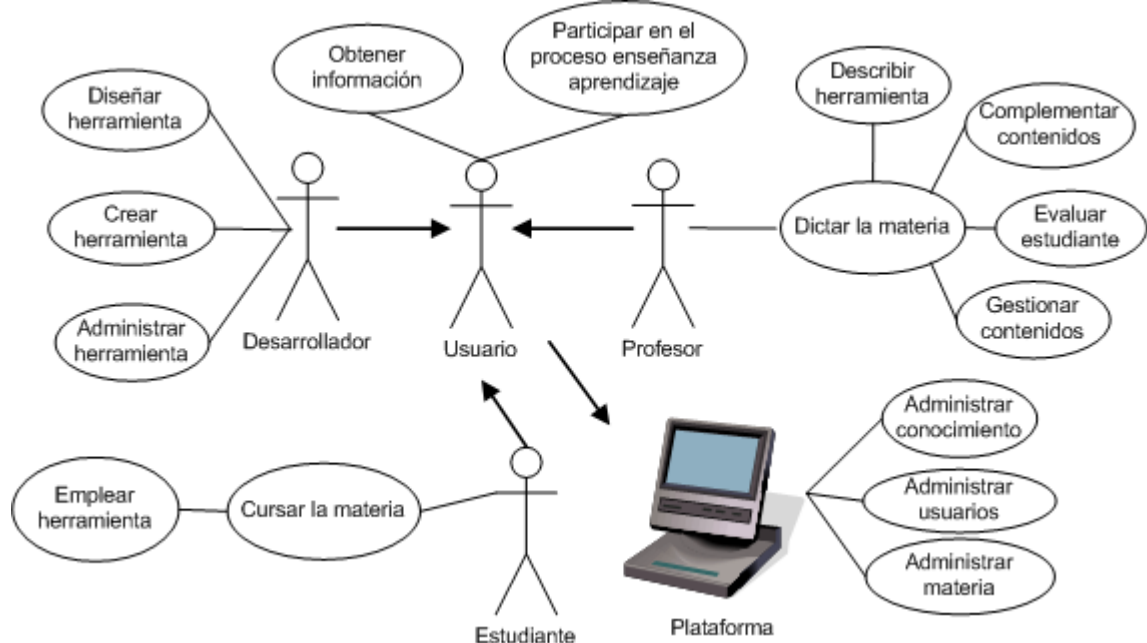


Figura 51. Diagrama de Casos de Uso General – Animación

4.3.4. Casos de uso – gráficos y tablas

❖ ELEMENTOS:

➤ Actores

- Estudiante (Emplea el sistema para obtener información).
- Profesor (Se apoya en el sistema para complementar los contenidos de su clase y recibe información del mismo).
- Desarrollador (Recibe información para diseñar y crear el sistema).

➤ Sistemas

- Plataforma (Integra el sistema a su ambiente).

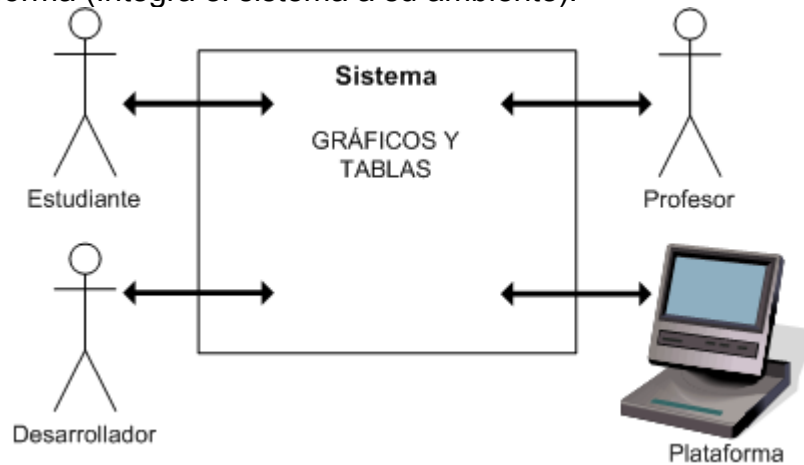


Figura 52. Diagrama de Actores y Sistema – Gráficos y tablas

❖ CASOS DE USO

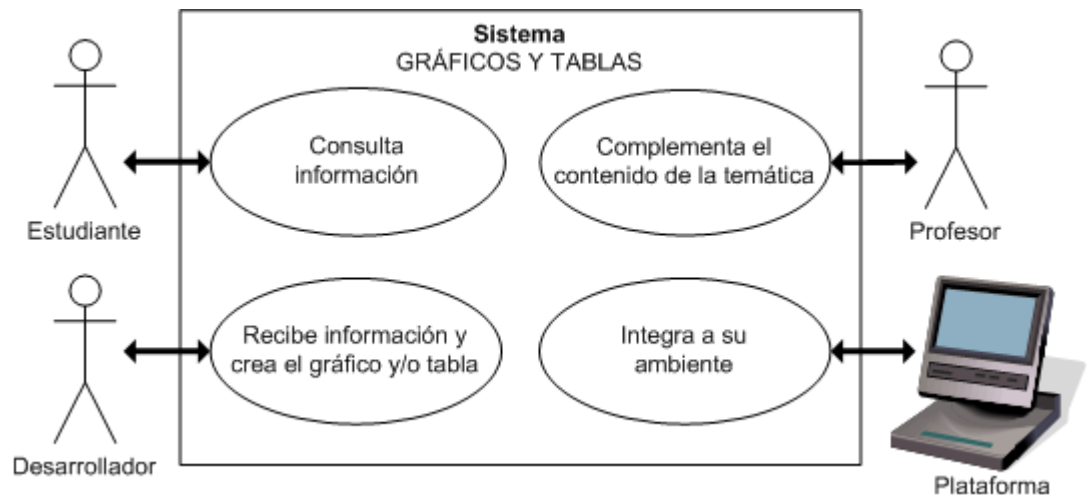


Figura 53. Diagrama de Casos de Uso – Gráficos y tablas

❖ **DIAGRAMAS DE CASOS DE USO**

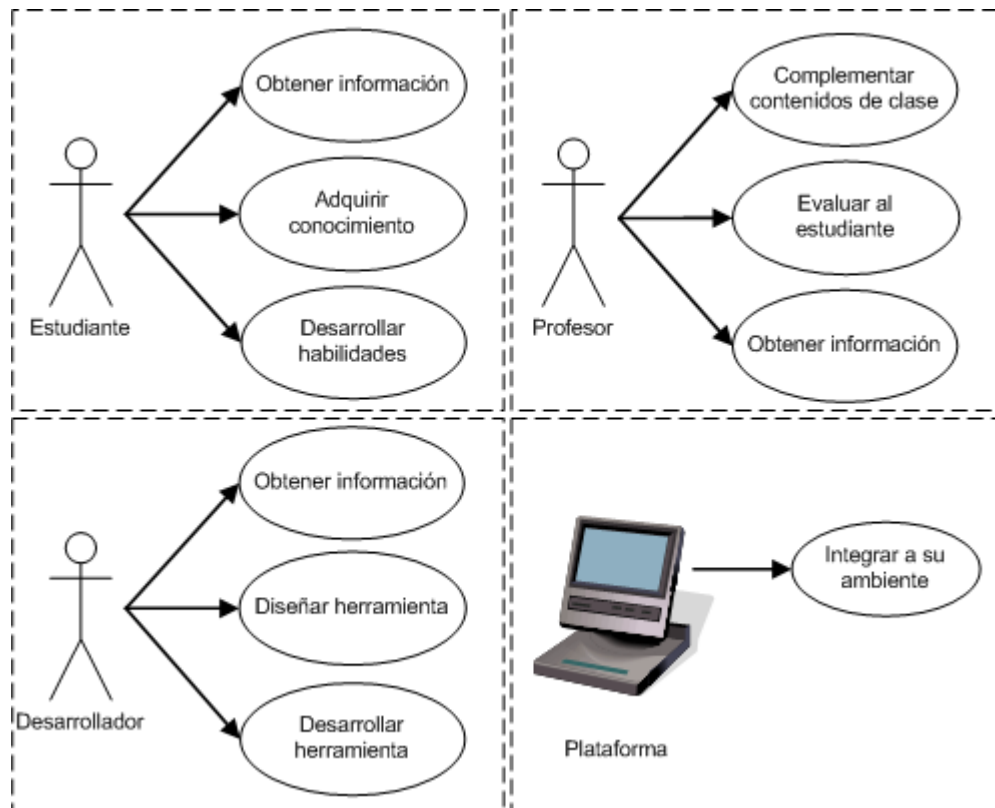


Figura 54. Diagrama de Casos de Uso para Actores y Sistema – Gráficos y tablas

❖ **DESCRIPCIÓN DE LOS CASOS DE USO**

Nombre:	Consulta información
Descripción:	Permite acceder a la información que contiene el gráfico y/o tabla sobre la temática en cuestión.
Actores:	Estudiante
Precondiciones:	El Estudiante debe haber ingresado a la plataforma que aloja el sistema.
Flujo Normal:	1. El estudiante ingresa a la plataforma.

<ol style="list-style-type: none"> 2. El estudiante se ubica en la plantilla de la asignatura Ingeniería del software. 3. El estudiante escoge la temática a la que pertenece el gráfico y/o tabla. 4. El estudiante accede al gráfico y/o tabla. 5. El estudiante recibe información sobre la temática a través del gráfico y/o tabla.
<p>Poscondiciones: El actor analiza la información obtenida del gráfico y/o tabla.</p>

Nombre:	Complementa el contenido de la temática
Descripción: Permite complementar el contenido de la temática expuesto en el aula.	
Actores: Profesor	
Precondiciones: El profesor debe tener un objeto de aprendizaje relacionado con su asignatura en la plataforma y debe preparar la información que va a suministrar a los estudiantes.	
Flujo Normal: <ol style="list-style-type: none"> 1. El estudiante ingresa a la plataforma por indicación del profesor. 2. El profesor guía al estudiante sobre cómo ingresar a la plantilla de la asignatura Ingeniería del software y a la temática que se está tratando. 3. El profesor explica al estudiante cómo acceder al gráfico y/o tabla de la temática en cuestión. 4. El profesor enseña el uso y manejo de la herramienta (gráficos y tablas). 5. El profesor complementa la explicación de la temática una vez se haya utilizado la herramienta. 	
Poscondiciones: El profesor evalúa la comprensión de la información obtenida del gráfico y/o tabla por parte de los estudiantes.	

Nombre:	Recibe información y crea el gráfico y/o tabla
Descripción: Permite desarrollar herramientas que dan soporte al proceso de enseñanza-aprendizaje.	
Actores: Desarrollador	

<p>Precondiciones: El desarrollador debe tener los conocimientos previos necesarios sobre la temática a tratar en la herramienta y sobre los recursos necesarios para construirla.</p>
<p>Flujo Normal:</p> <ol style="list-style-type: none"> 1. El desarrollador adquiere y organiza la información que contendrá el gráfico y/o tabla. 2. El desarrollador diseña el gráfico y/o tabla. 3. El desarrollador crea el gráfico y/o tabla. 4. El desarrollador implanta la herramienta (gráfico y/o tabla) en la plataforma. 5. El desarrollador se cerciora del correcto funcionamiento de la herramienta.
<p>Poscondiciones: El desarrollador evalúa la actitud del estudiante frente al gráfico y/o tabla con el fin de realizar correcciones o posibles mejoras.</p>

Nombre:	Integra a su ambiente
Descripción:	Permite acceder a los objetos de aprendizaje y proporciona el entorno de interacción de dichos objetos con sus usuarios.
Actores:	Plataforma
Precondiciones:	Todos los recursos necesarios para acceder a la plataforma deben estar disponibles.
Flujo Normal:	<ol style="list-style-type: none"> 1. Se carga la plataforma desde un Terminal y se accede a sus contenidos. 2. La plataforma adapta las plantillas a su entorno. 3. La plataforma carga los componentes de la plantilla. 4. La plataforma permite al usuario el acceso al gráfico y/o tabla.
Poscondiciones:	La plataforma guarda un registro del ingreso a la misma y de las actividades que se realizaron por parte del usuario en ella.

❖ **DIAGRAMA DE CASOS DE USO GENERAL**

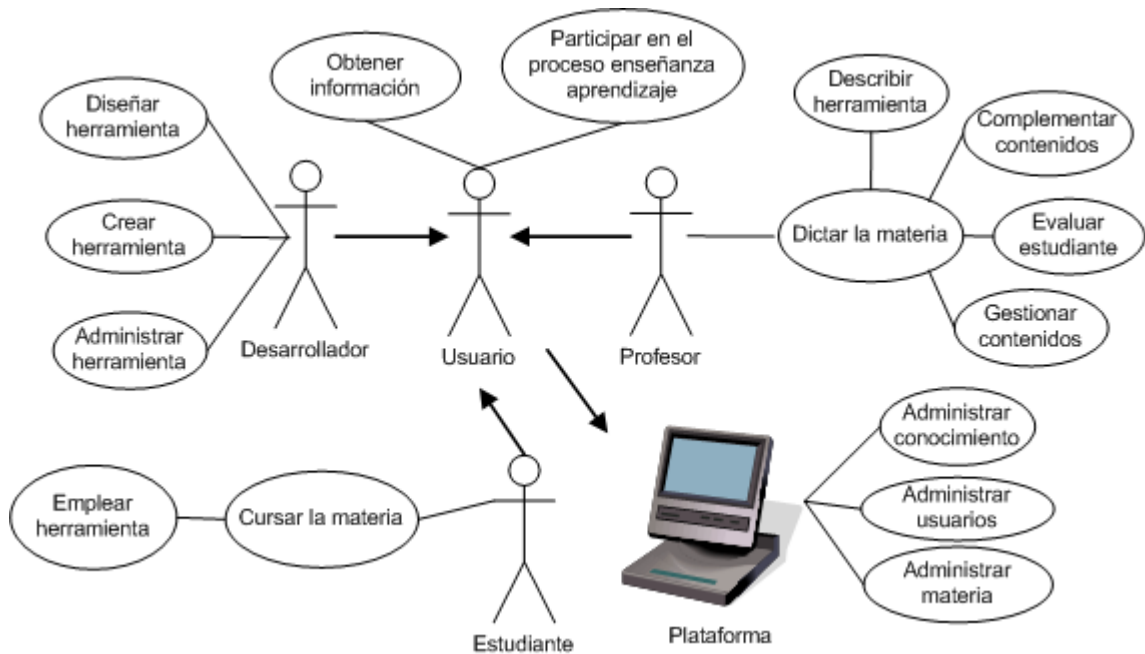


Figura 55. Diagrama de Casos de Uso General – Gráfico o tabla

4.3.5. Casos de uso – aplicativo

❖ **ELEMENTOS:**

➤ **Actores**

- Estudiante (Emplea el sistema para obtener información).
- Profesor (Se apoya en el sistema para complementar los contenidos de su clase y recibe información del mismo).
- Desarrollador (Recibe información para diseñar y crear el sistema).

➤ **Sistemas**

- Plataforma (Integra el sistema a su ambiente).

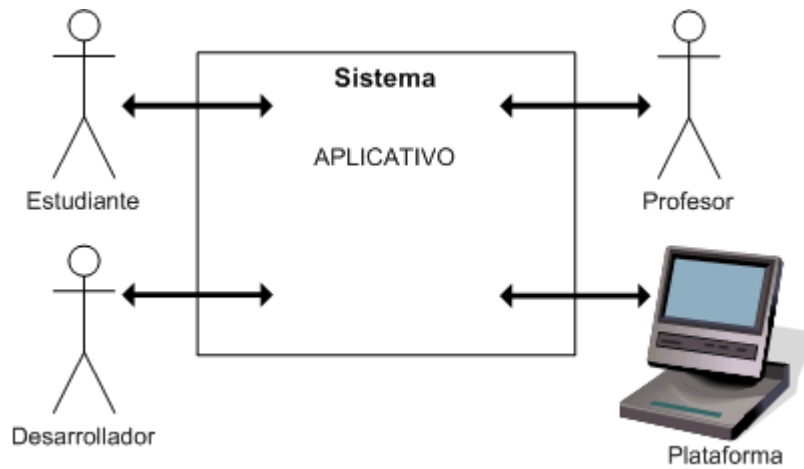


Figura 56. Diagrama de Actores y Sistema – Aplicativo

❖ CASOS DE USO

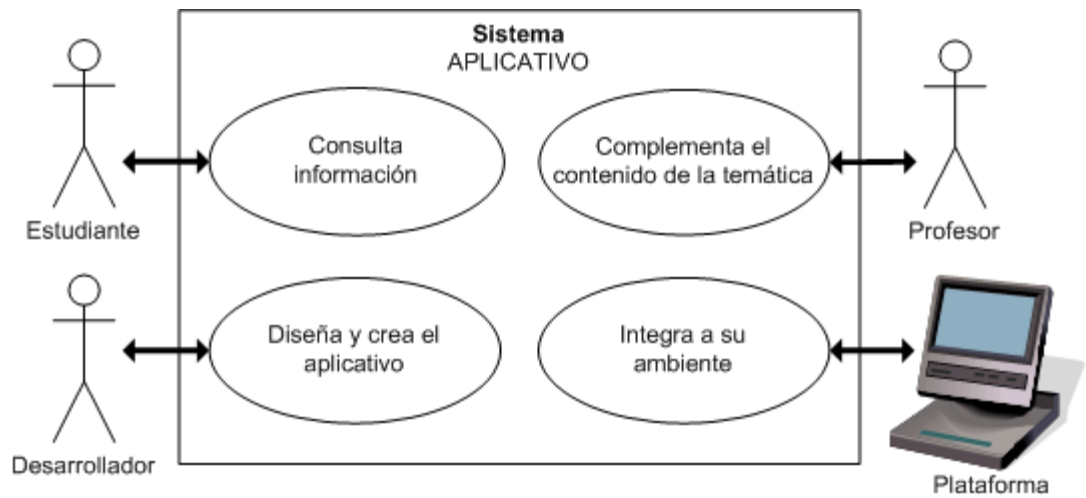


Figura 57. Diagrama de Casos de Uso – Aplicativo

❖ **DIAGRAMAS DE CASOS DE USO**

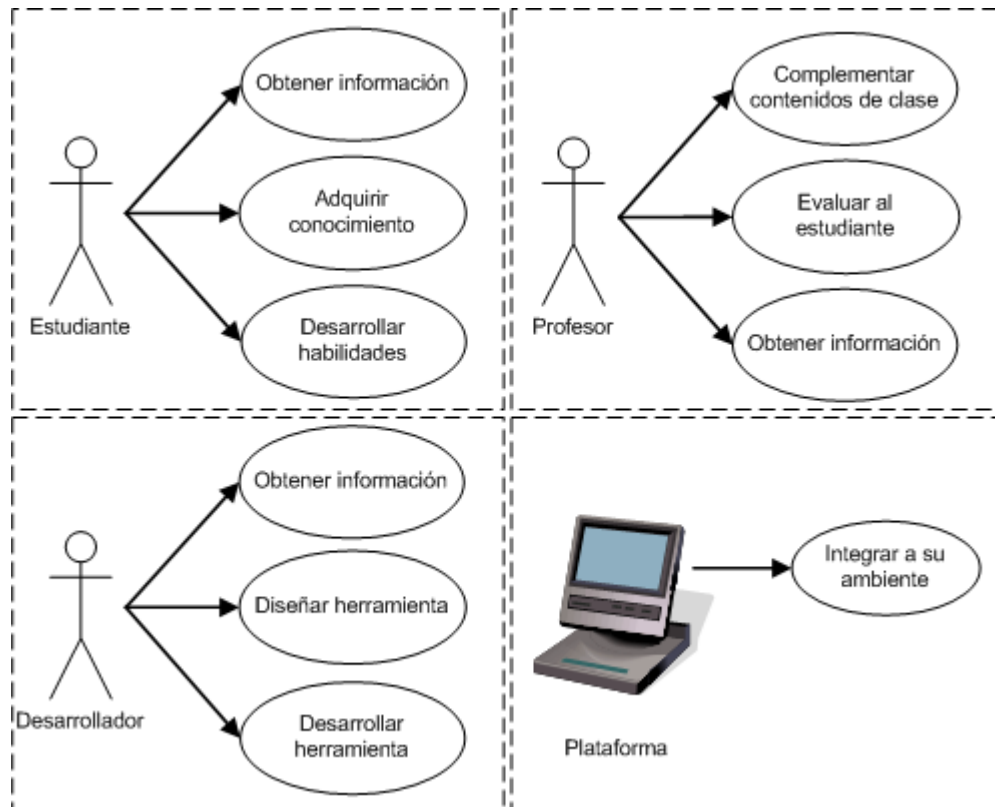


Figura 58. Diagrama de Casos de Uso para Actores y Sistema – Aplicativo

❖ **DESCRIPCIÓN DE LOS CASOS DE USO**

Nombre:	Consulta información
Descripción:	Permite interactuar con el aplicativo relacionado con la temática clasificación de métricas.
Actores:	Estudiante
Precondiciones:	El Estudiante debe haber ingresado a la plataforma que aloja el sistema.
Flujo Normal:	<ol style="list-style-type: none"> 1. El estudiante ingresa a la plataforma. 2. El estudiante se ubica en la plantilla de la asignatura Ingeniería del software. 3. El estudiante escoge la temática clasificación de métricas (a la cual pertenece el aplicativo para calcular métricas).

<ol style="list-style-type: none"> 4. El estudiante accede al aplicativo. 5. El estudiante analiza el enunciado y accede a la ayuda para utilizar el aplicativo si es necesario. 6. El estudiante selecciona el tipo de cálculo que desea realizar. 7. El estudiante ingresa los datos necesarios. 8. El estudiante ejecuta el aplicativo 9. El estudiante obtiene el resultado del proceso de ejecutar el aplicativo.
<p>Poscondiciones: El actor analiza la información obtenida de los resultados del aplicativo y saca sus propias conclusiones.</p>

Nombre:	Complementa el contenido de la temática
Descripción:	Permite complementar el contenido de la temática expuesto en el aula.
Actores:	Profesor
Precondiciones:	El profesor debe tener un objeto de aprendizaje relacionado con su asignatura en la plataforma y debe preparar la información que va a suministrar a los estudiantes.
Flujo Normal:	<ol style="list-style-type: none"> 1. El profesor realiza una explicación sobre las métricas orientadas al tamaño y a la función en el aula de clase. 2. El estudiante ingresa a la plataforma por indicación del profesor. 3. El profesor guía al estudiante sobre cómo ingresar a la plantilla de la asignatura Ingeniería del software y a la temática que se está tratando. 4. El profesor explica al estudiante cómo acceder al aplicativo para calcular métricas. 5. El profesor enseña el uso y manejo del aplicativo. 6. El profesor complementa la explicación de la temática una vez se haya utilizado la herramienta.
Poscondiciones:	El profesor evalúa la comprensión de la información obtenida del aplicativo por parte de los estudiantes.

Nombre:	Recibe información y crea el gráfico y/o tabla
Descripción:	Permite desarrollar herramientas que dan soporte al proceso de enseñanza-

aprendizaje.
Actores: Desarrollador
Precondiciones: El desarrollador debe tener los conocimientos previos necesarios sobre la temática a tratar en la herramienta y sobre los recursos necesarios para construirla.
Flujo Normal: <ol style="list-style-type: none"> 1. El desarrollador adquiere y organiza la información y operaciones que contendrá el aplicativo. 2. El desarrollador diseña el aplicativo. 3. El desarrollador crea el aplicativo. 4. El desarrollador implanta la herramienta (aplicativo) en la plataforma. 5. El desarrollador se cerciora del correcto funcionamiento de la herramienta.
Poscondiciones: El desarrollador evalúa la actitud del estudiante frente al aplicativo con el fin de realizar correcciones o posibles mejoras.

Nombre:	Integra a su ambiente
Descripción: Permite acceder a los objetos de aprendizaje y proporciona el entorno de interacción de dichos objetos con sus usuarios.	
Actores: Plataforma	
Precondiciones: Todos los recursos necesarios para acceder a la plataforma deben estar disponibles.	
Flujo Normal: <ol style="list-style-type: none"> 1. Se carga la plataforma desde un Terminal y se accede a sus contenidos. 2. La plataforma adapta las plantillas a su entorno. 3. La plataforma carga los componentes de la plantilla. 4. La plataforma permite al usuario el acceso al aplicativo. 	
Poscondiciones: La plataforma guarda un registro del ingreso a la misma y de las actividades que se realizaron por parte del usuario en ella.	

❖ DIAGRAMA DE CASOS DE USO GENERAL

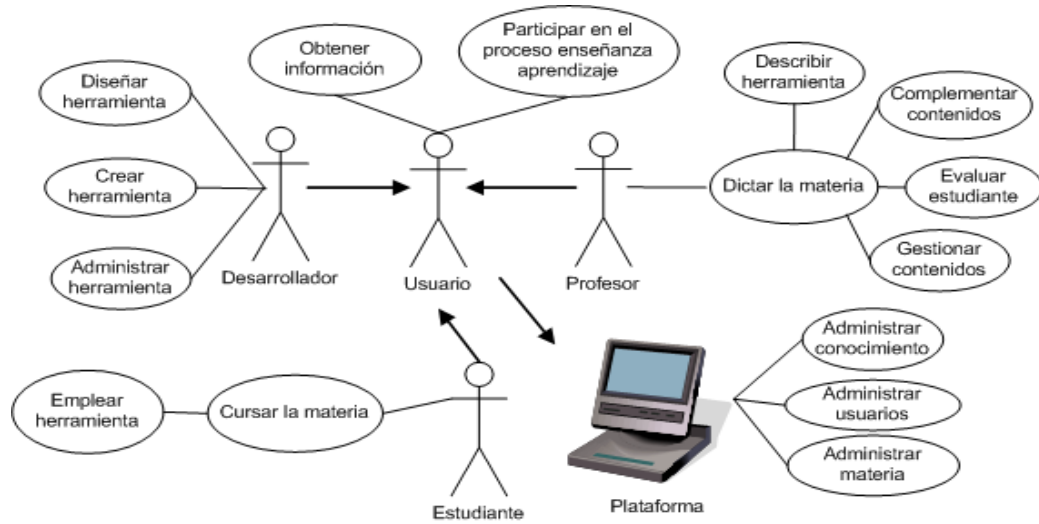


Figura 59. Diagrama de Casos de Uso General – Aplicativo

LENGUAJE DE MODELADO UNIFICADO (DIAGRAMAS DE SECUENCIA)

DIAGRAMAS DE SECUENCIA:

En el proceso de desarrollo del Objeto de Aprendizaje se realizaron los siguientes diagramas de secuencia para representar las interacciones entre los objetos organizadas en una secuencia temporal.

4.3.6. Diagramas de secuencia – Documento soporte

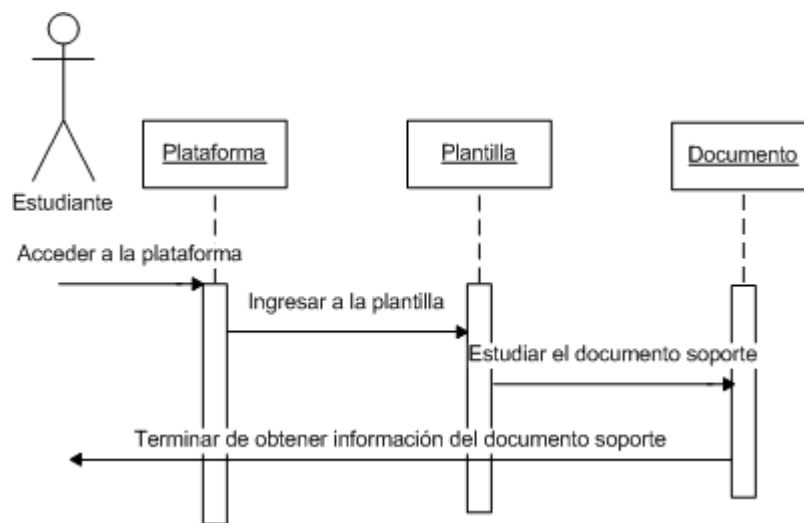


Figura 60. Diagrama de secuencia – Documento soporte - Estudiante

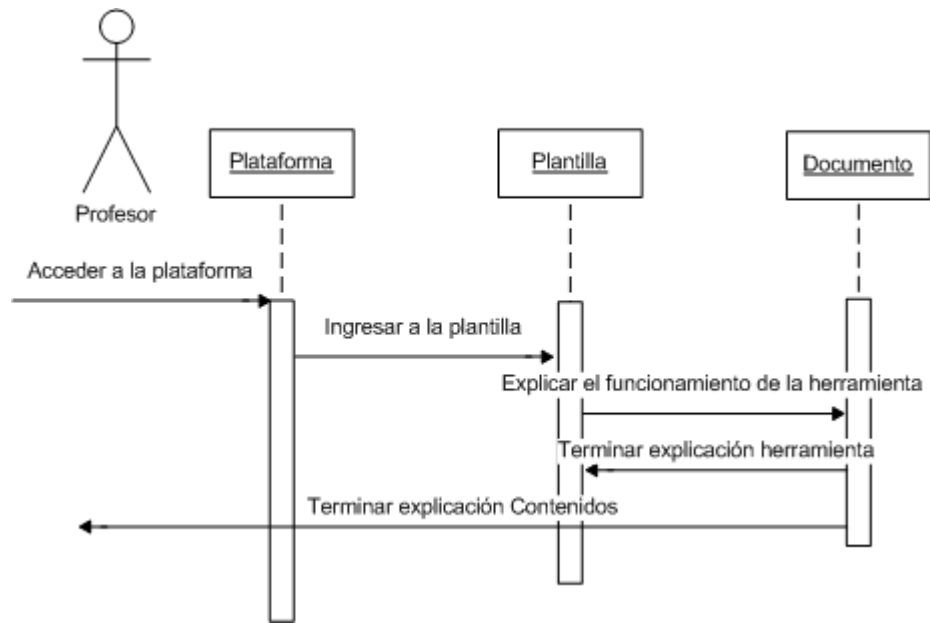


Figura 61. Diagrama de secuencia – Documento soporte - Profesor

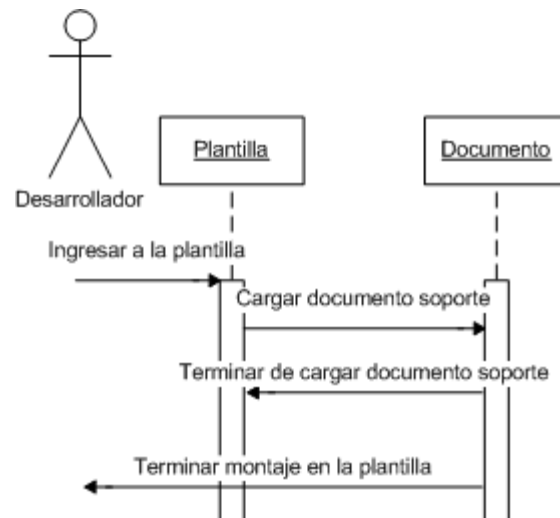


Figura 62. Diagrama de secuencia – Documento soporte- Desarrollador

4.3.7. Diagramas de secuencia – Audio

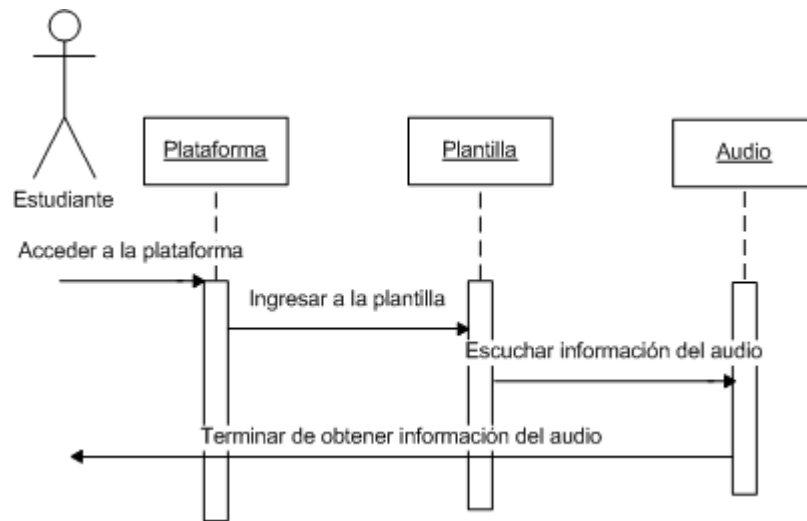


Figura 63. Diagrama de secuencia – Audio – Estudiante

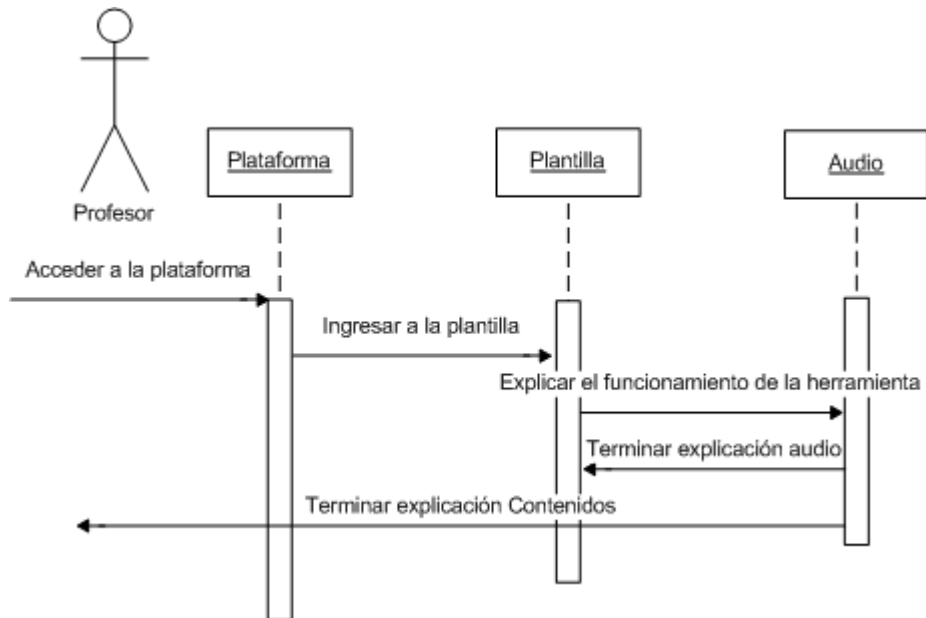


Figura 64. Diagrama de secuencia – Audio – Profesor

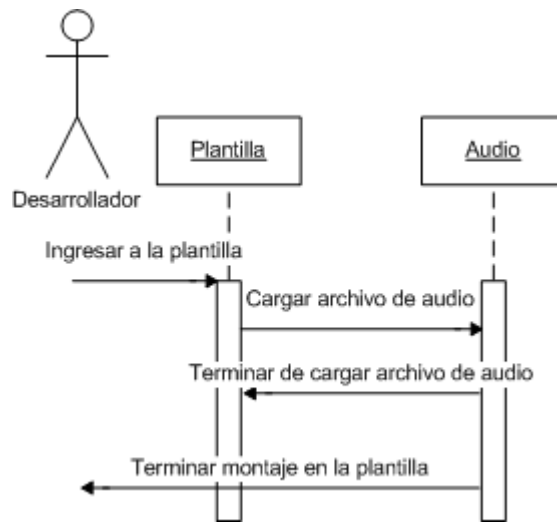


Figura 65. Diagrama de secuencia – Audio – Desarrollador

4.3.8. Diagramas de secuencia – Animación

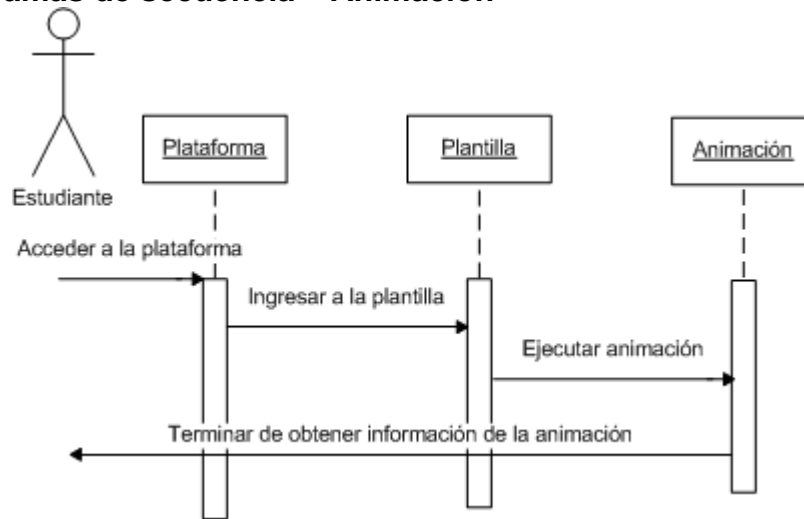


Figura 66. Diagrama de secuencia – Animación – Estudiante

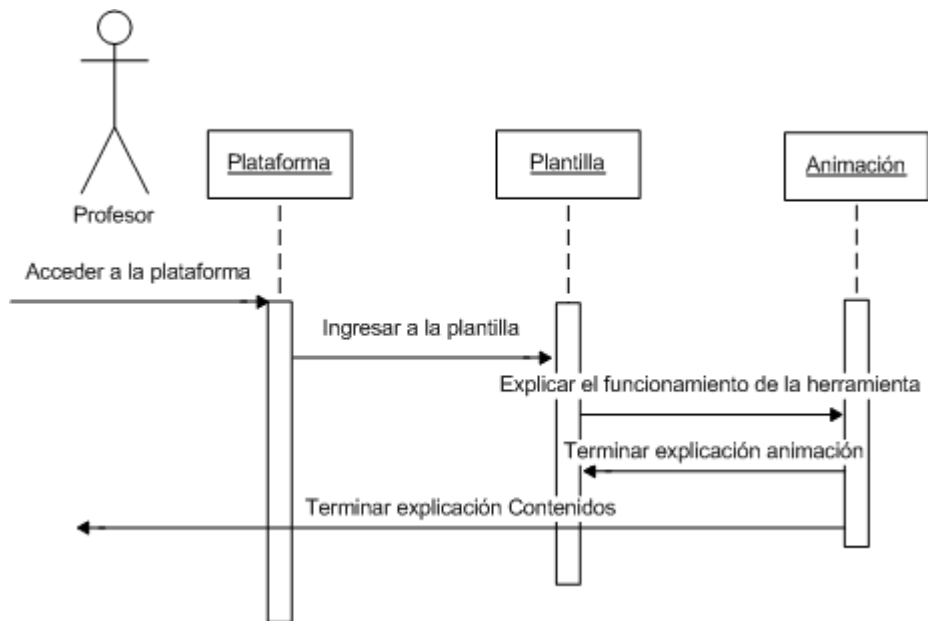


Figura 67. Diagrama de secuencia –Animación – Profesor

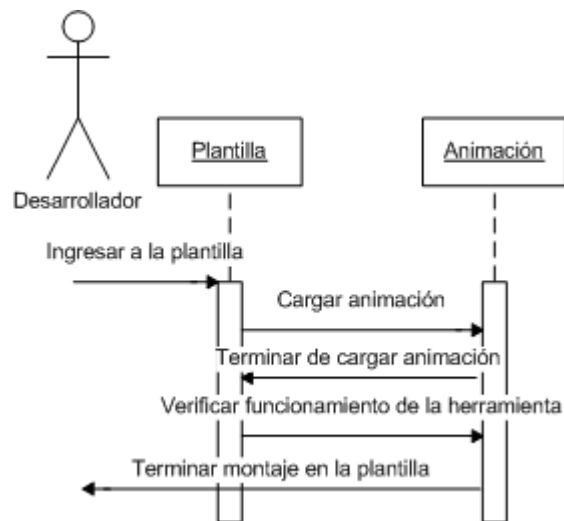


Figura 68. Diagrama de secuencia – Animación – Desarrollador

4.3.9. Diagramas de secuencia – Aplicativo

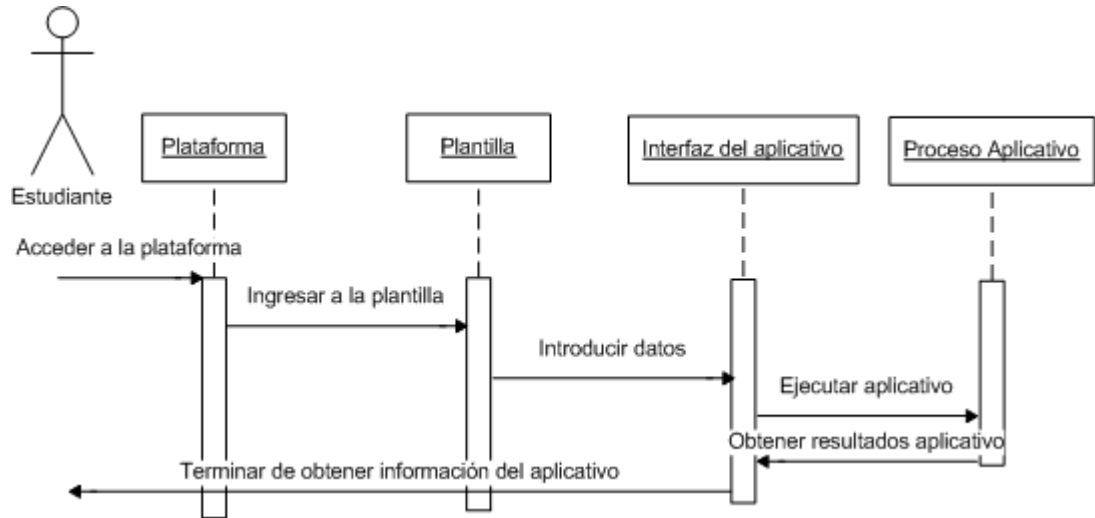


Figura 69. Diagrama de secuencia – Aplicativo – Estudiante

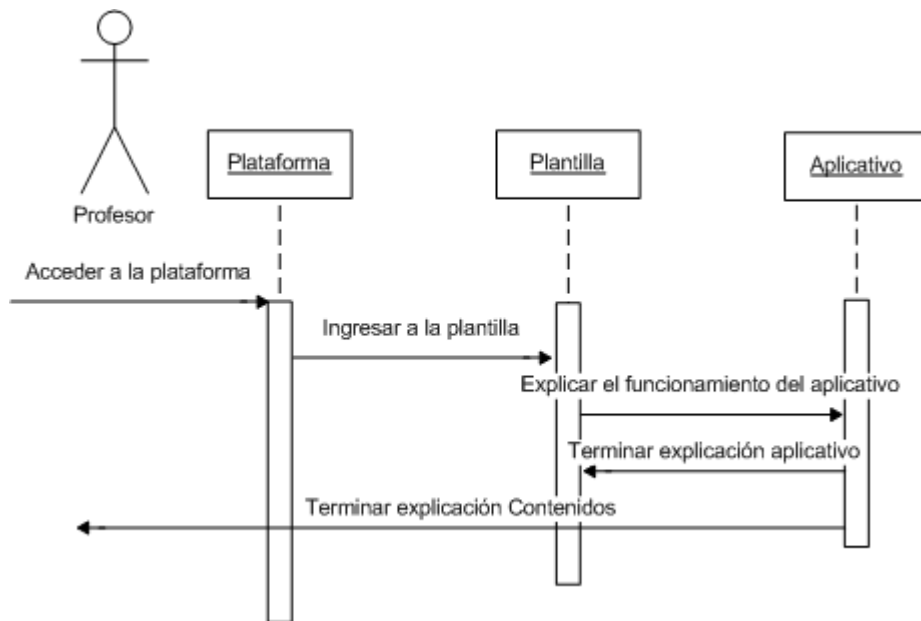


Figura 70. Diagrama de secuencia – Aplicativo – Profesor

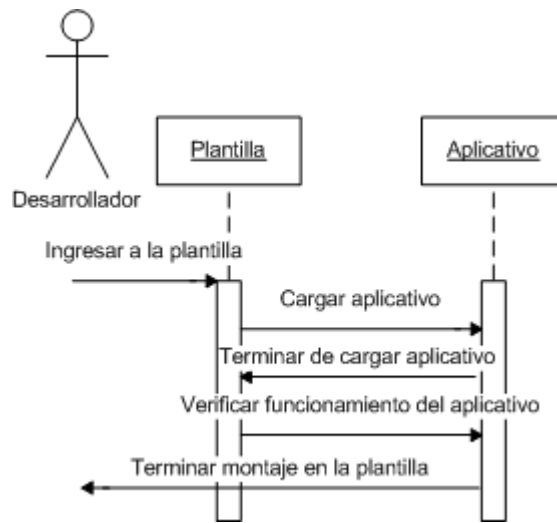


Figura 71. Diagrama de secuencia – Aplicativo – Desarrollador

4.3.10. Diagramas de secuencia – Gráficos y tablas

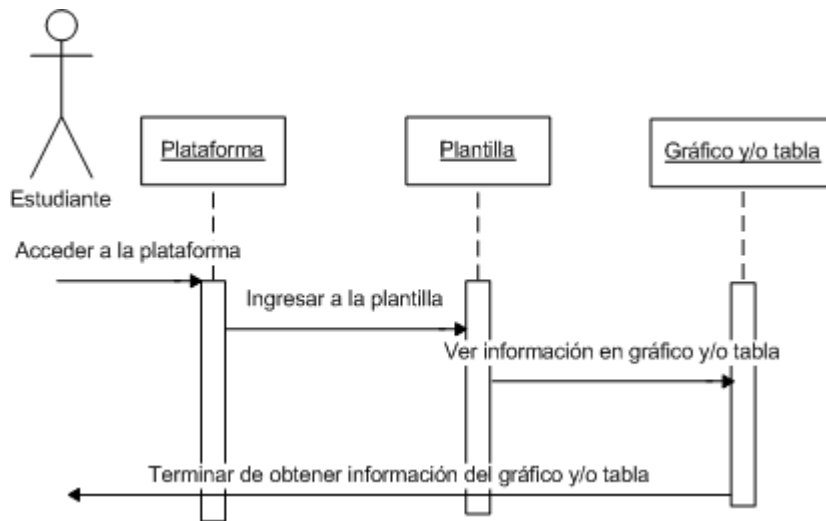


Figura 72. Diagrama de secuencia – Gráfico y/o tabla – Estudiante

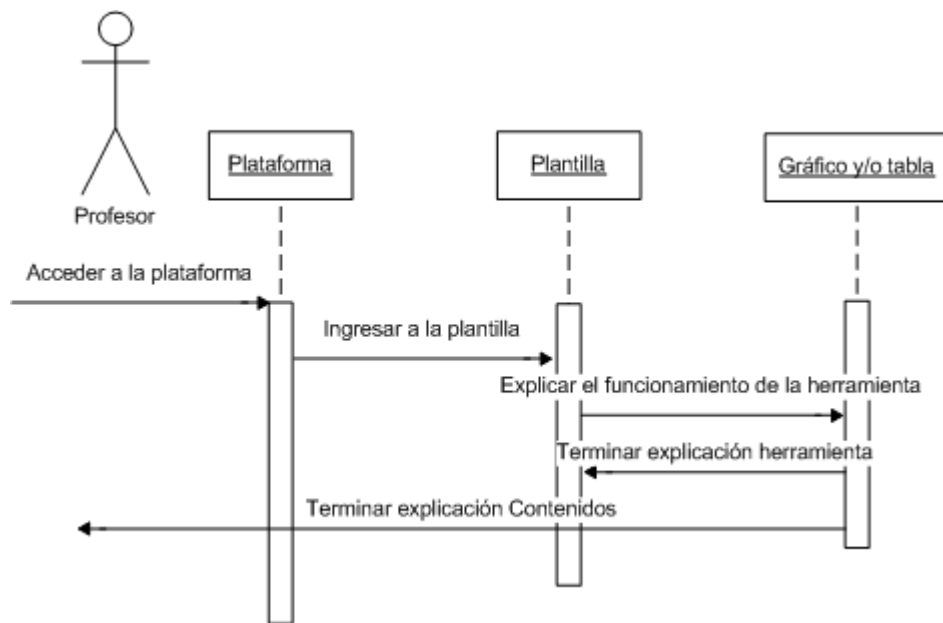


Figura 73. Diagrama de secuencia- Gráfico y/o tabla – Profesor

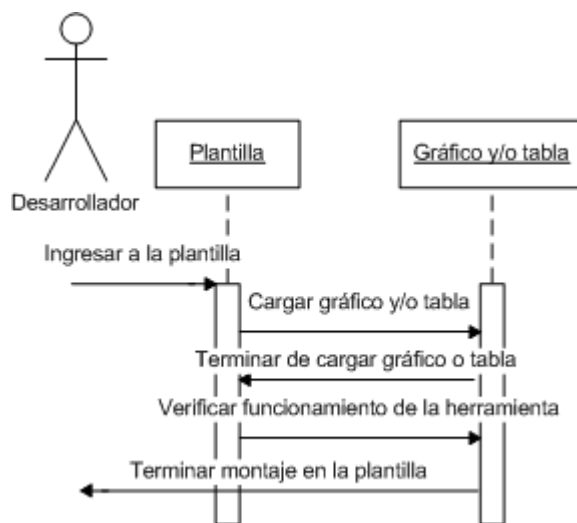


Figura 74. Diagrama de secuencia – Gráfico y/o tabla – Desarrollador

4.4. ETAPA DE CONSTRUCCIÓN

Esta fase aborda el desarrollo y la integración de los componentes y características que conforman el objeto de aprendizaje, el resultado de esta etapa es el producto ya desarrollado, preparado para ser puesto en manos de los estudiantes que son los usuarios finales del sistema. A continuación se presenta el desarrollo del objeto de aprendizaje correspondiente a la temática “métricas del software”.

Temáticas que conforman el objeto de aprendizaje de métricas del software

Los temas que se implementaron en el objeto de aprendizaje sobre las métricas del software son extraídos según las actividades plasmadas en el diagrama secuencial de actividades de la asignatura ingeniería del software, y se descomponen en las siguientes:

- Generalidades de métricas del software
- Clasificación de las métricas del software
- Entidades a medir del software
- Métricas en proyectos de software

Para llevar a cabo el desarrollo del contenido del objeto, se hace necesaria la implementación de una plantilla desde la cual se pueda acceder a diversos recursos digitales, tales como textos, imágenes, diagramas, videos, narraciones, animaciones.

Con relación a la presentación del objeto como tal y para efectos de implementación en la plataforma educativa institucional **e-escen@riuis**, se contó con la plantilla realizada por el equipo del ProSPETIC, la cual no solamente favorece el trabajo de diseño del objeto, sino también el proceso de secuenciación y comprensión del contenido por parte de los alumnos.

Para permitir el acceso a cada uno de los contenidos del objeto de aprendizaje, se cuenta con la ventana de contenidos (ver figura), allí estos se encuentran organizados de manera tal que se identifique el tema general, y cada uno de los subtemas o temas específicos que lo complementan. (En la tabla de contenidos se presentan inactivas las temáticas que no se han desarrollado aún dentro del proyecto).

4.4.1. Descripción del objeto de aprendizaje sobre la plantilla



Figura 75. Tabla de contenidos del Objeto de Aprendizaje

Después de seleccionar el contenido a consultar, en la ventana principal de la plantilla (ver figura 76) se presenta la información o núcleo del conocimiento relacionada con la temática seleccionada, en la parte derecha de esta ventana se encuentran los botones de acceso a otros recursos didácticos que dan soporte al contenido temático, dentro de estos encontramos información de soporte en documentos con formato PDF, archivos de audio, videos, gráficos, aplicativos e información complementaria, también herramientas como calculadora, iconos de navegación (anterior, siguiente, inicio), imprimir, agente inteligente, libreta de notas, ejercicios, glosario, pizarra y descanso⁷¹.

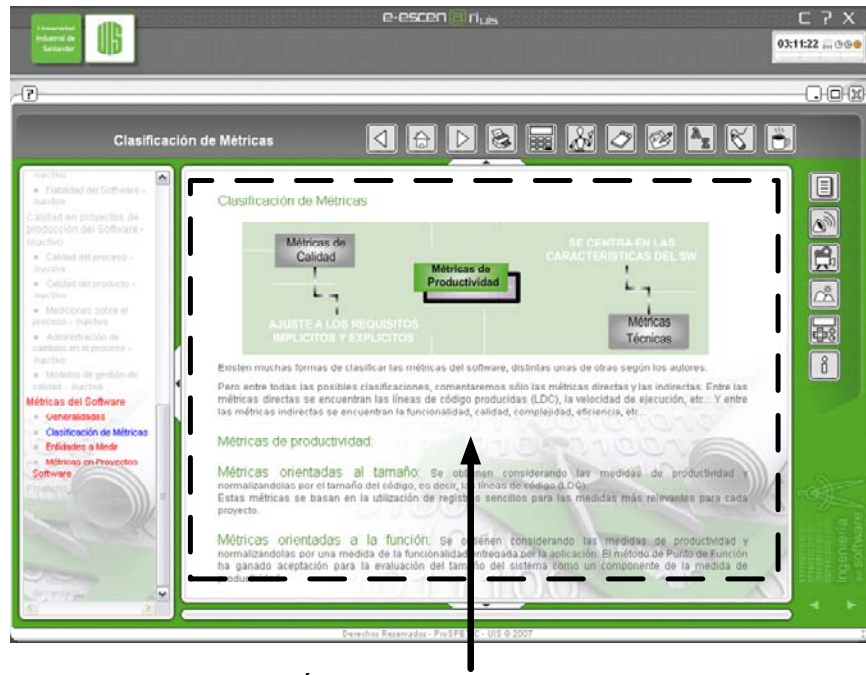
⁷¹ Las herramientas se encuentran en la plantilla pero aún no han sido activadas.

The screenshot shows a web browser window titled 'Clasificación de Métricas' from the 'e-escen@riUIS' platform. The browser's address bar shows the URL 'http://www.riuis.edu.co/escen'. The page content includes a navigation menu on the left with categories like 'Inactivo', 'Calidad en proyectos de producción del Software', and 'Métricas del Software'. The main content area features a diagram titled 'Clasificación de Métricas' with three boxes: 'Métricas de Calidad' (linked to 'AJUSTE A LOS REQUISITOS IMPLICITOS Y EXPLICITOS'), 'Métricas de Productividad' (highlighted with a red box), and 'Métricas Técnicas' (linked to 'SE CENTRA EN LAS CARACTERISTICAS DEL SW'). Below the diagram, there is explanatory text about different classification methods and a section on 'Métricas de productividad' with sub-sections for 'Métricas orientadas al tamaño' and 'Métricas orientadas a la función'. The footer of the page reads 'Derechos Reservados - ProSPETIC - UIS © 2007'.

Figura 76. Ventana principal de la plantilla para la visualización del objeto de aprendizaje sobre la plataforma educativa institucional e-escen@riUIS

Núcleo del Conocimiento

Los núcleos de conocimiento son los contenidos de una acción formativa presentados de forma sintética y desarrollados a partir de las ideas básicas que el estudiante debe aprender. Estos contenidos sintetizados remiten al estudiante a las fuentes documentales y a las actividades que le permiten construir su aprendizaje.



NÚCLEO DE CONOCIMIENTO

Figura 77. Núcleo de conocimiento - Clasificación de métricas

A continuación se expone las imágenes de cada botón activo dentro de la plantilla y su funcionalidad:

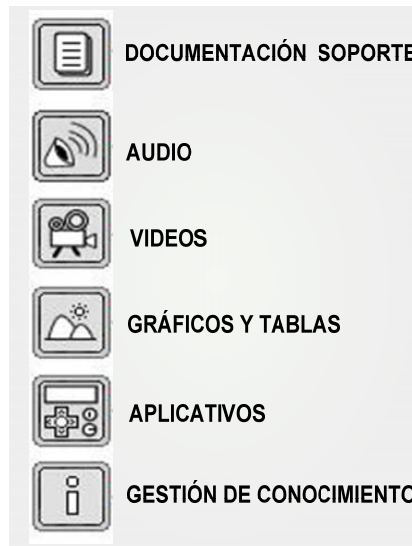


Figura 78. Botones activos dentro de la plantilla

- **Botón de documentación de Soporte**

Mediante este botón se puede acceder al material de soporte que se encuentra en formato PDF. El documento PDF se desplegará en la sección correspondiente al núcleo de conocimiento para así evitar distracciones en el proceso de aprendizaje del usuario.

La utilización de texto se justifica por ser el lenguaje verbal una de las herramientas fundamentales del razonamiento, la cognición y la abstracción; porque el texto disminuye la ambigüedad de los mensajes y la divergencia en las interpretaciones.

En las lecturas se desarrollan los diferentes contenidos del programa que ayudarán al estudiante a conseguir los objetivos de la acción formativa.

En la figura 79 se puede apreciar la forma de presentación de documentos a través de la plantilla.

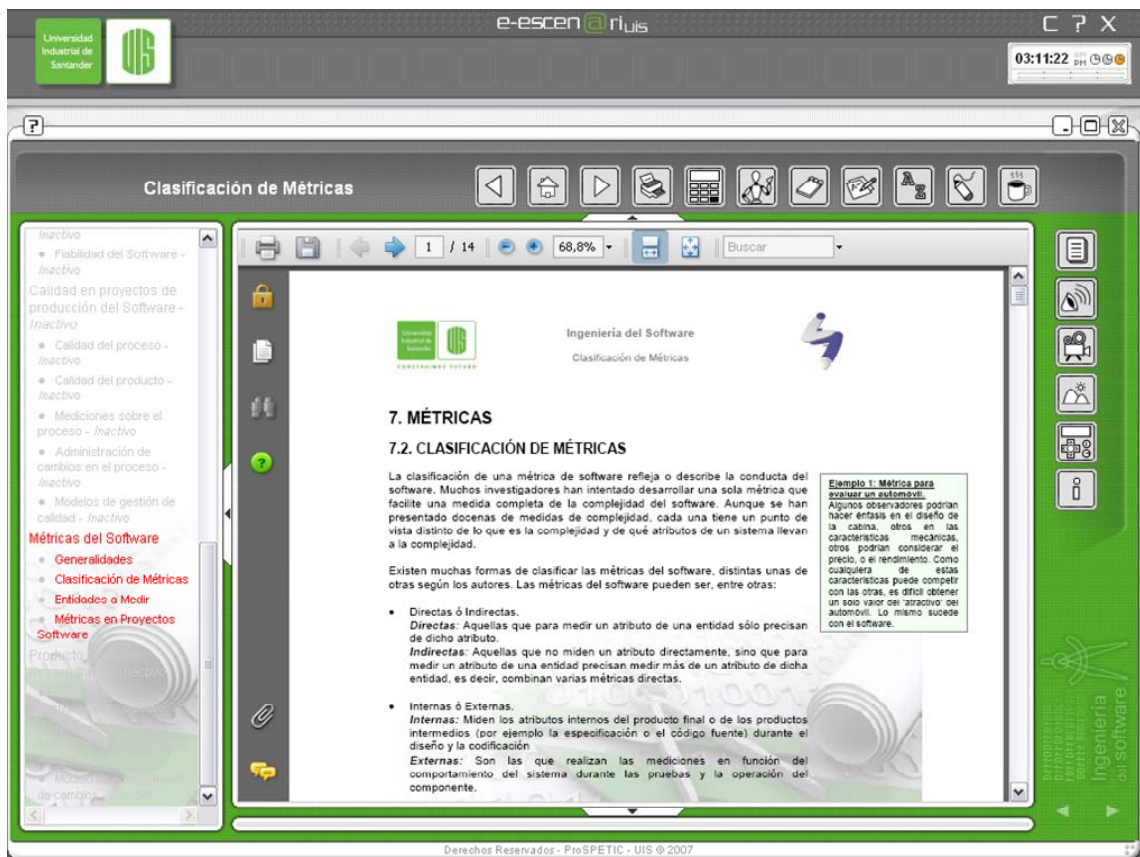


Figura 79. Documento de soporte - Clasificación de métricas

- **Botón de Archivos de Audio**

Mediante el botón de audio se puede acceder al material que se tenga como refuerzo para el aprendizaje de esta temática específica.

Se utilizan para reforzar unos espacios determinados del material didáctico multimedia o para generar ambientes psicológicos específicos. Se consigue por medio de voz, diálogos, música, efectos sonoros, grabaciones, etc.

El audio en aplicaciones multimedia permite:

- Generar un hilo de continuidad en la narrativa de la aplicación.
- Humanizar la relación usuario-máquina.
- Captar la atención del estudiante y motivar sus acciones.
- Desarrollar procesos de identificación y participación en el usuario.
- Reforzar la interacción en la navegación.

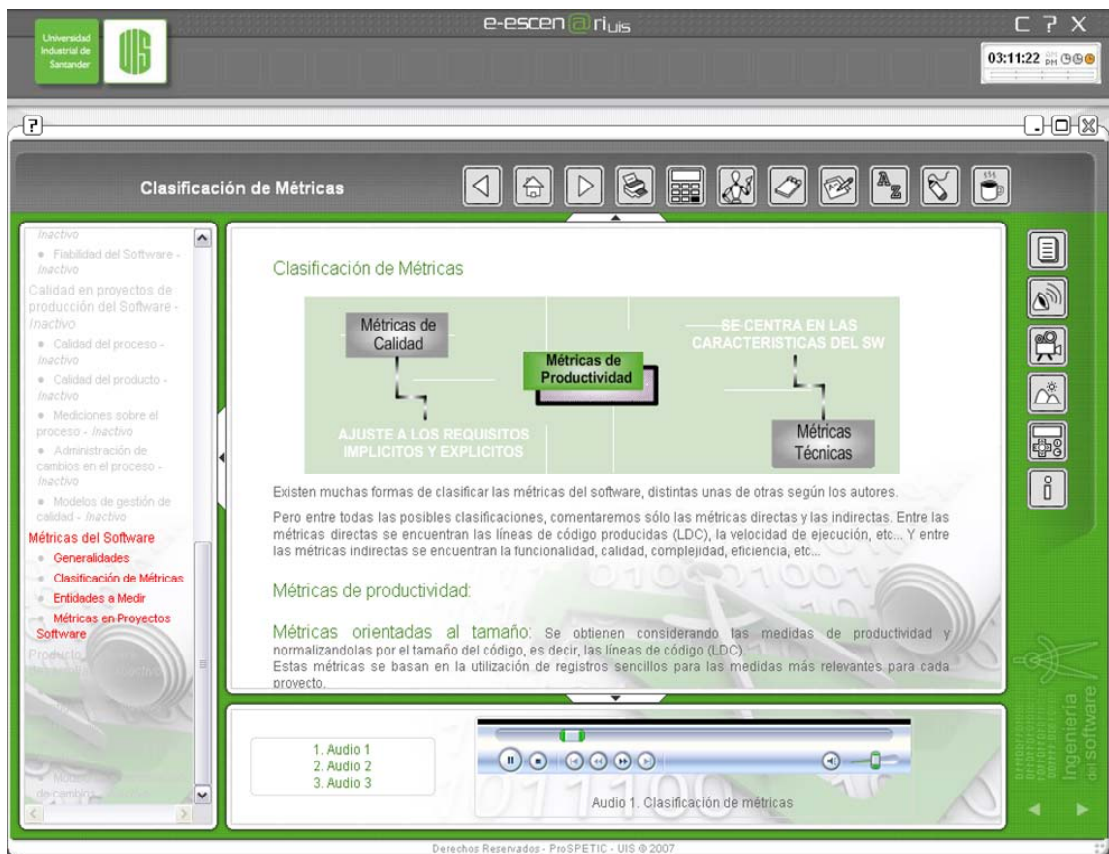


Figura 80. Archivos de audio - Clasificación de métricas

- **Botón de Archivos de Video**

Con este botón se puede acceder a los videos o animaciones que permitan el afianzamiento de los contenidos temáticos.

Resuelve la dificultad de poder “verbalizar” los contenidos que incluyen cierta complejidad para ser explicados con otros medios.

Tiene la ventaja de que aumenta la sensación de realismo y se aprovecha de la cultura audiovisual.

Es importante que el usuario pueda interactuar a través de los comandos de control (para avanzar, retroceder, detener o volver a revisar cierta secuencia).



Figura 81. Archivos de video - Clasificación de métricas - escena 1

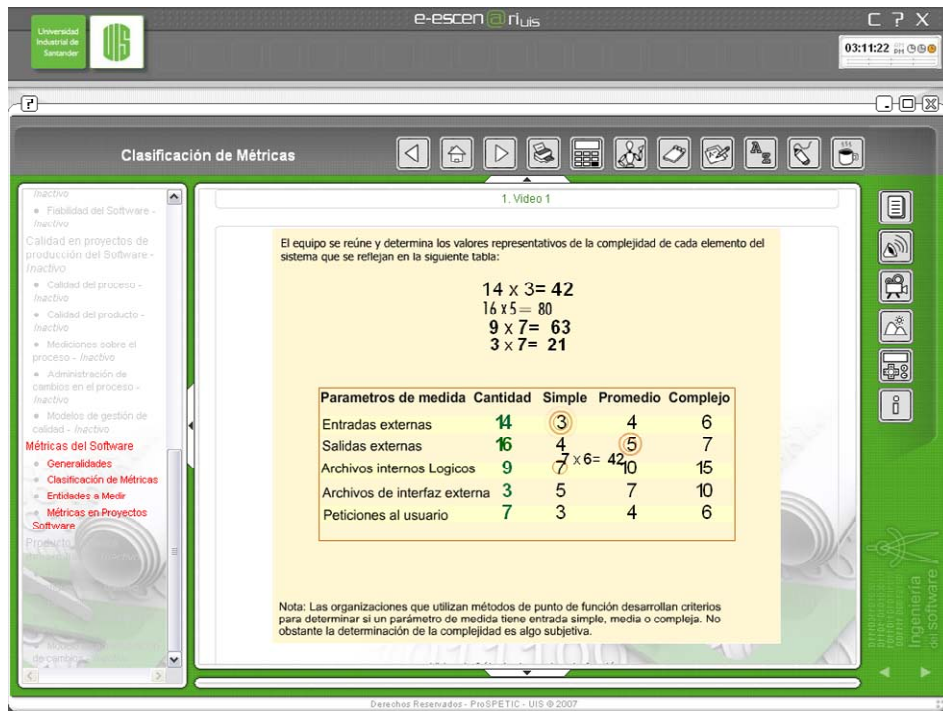


Figura 82. Archivos de video - Clasificación de métricas - escena 2

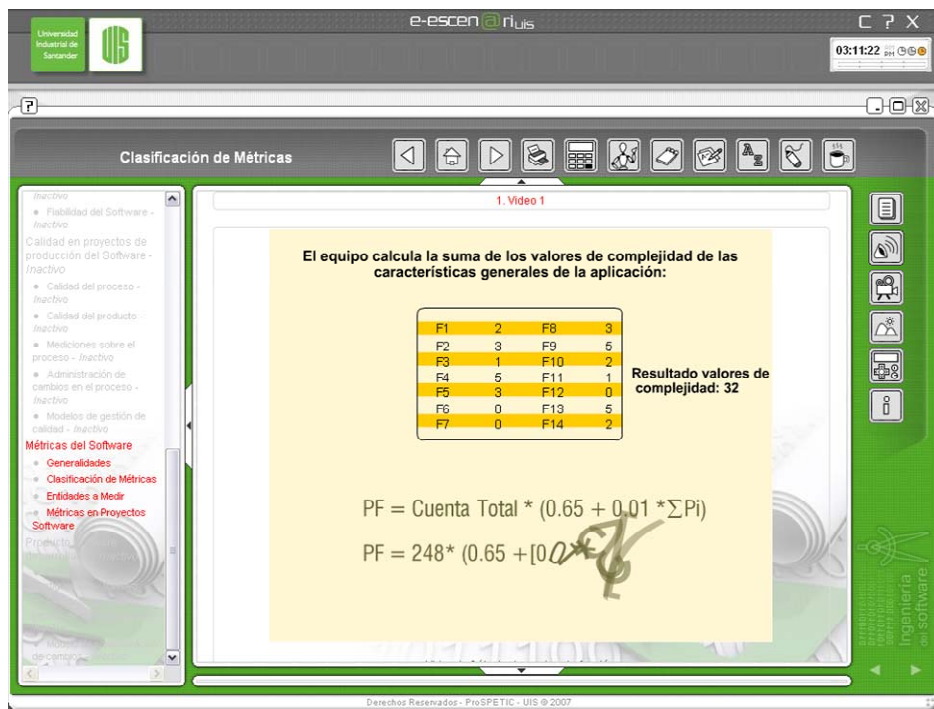


Figura 83. Archivos de video - Clasificación de métricas - escena 3

- **Botón de Gráficos**

Con este botón se accede a los gráficos que permitan el afianzamiento de los contenidos temáticos.

Son todas las imágenes fijas que se incorporan al material y que sirven para enriquecerlo provocando impacto, presentando de manera rápida y concisa información compleja, complementando informaciones, reforzando contextos, etc. Su desventaja es que en poco espacio dan mucha información y pueden admitir múltiples interpretaciones debido a su carácter ambiguo.

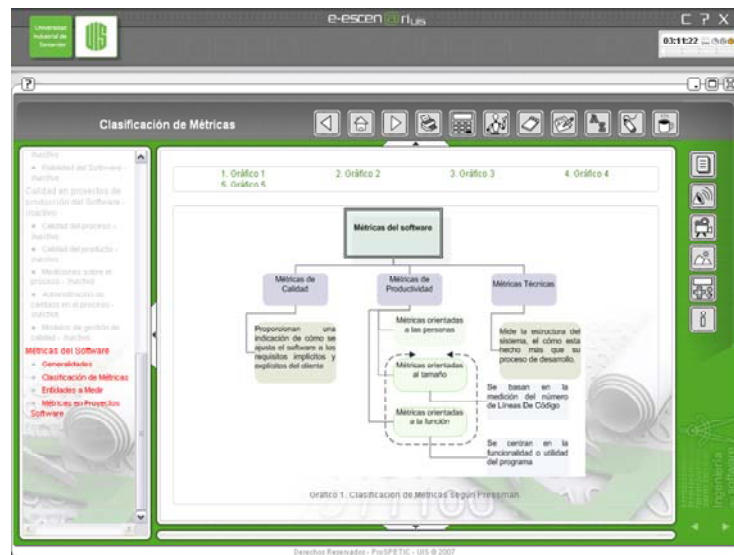


Figura 84. Gráfico nº 1 - Clasificación de métricas



Figura 85. Gráfico nº 2 - Clasificación de métricas



Figura 86. Gráfico nº 3 - Clasificación de métricas

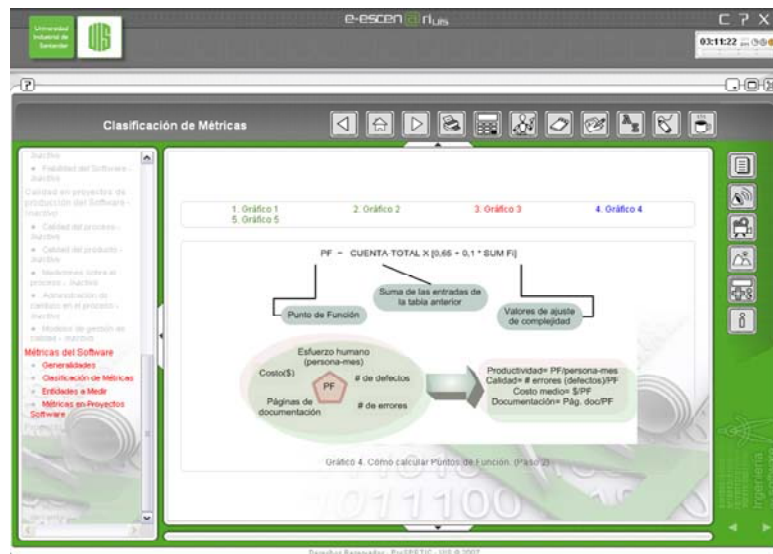


Figura 87. Gráfico nº 4 - Clasificación de métricas

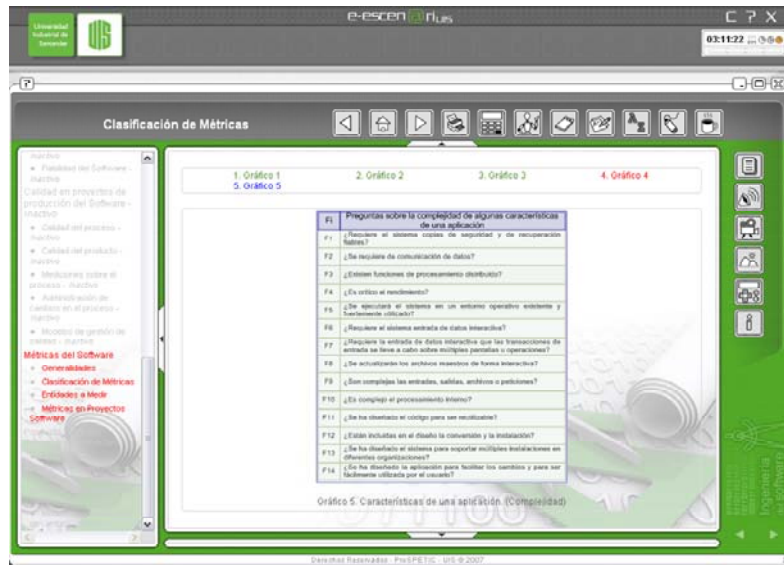


Figura 88. Gráfico nº 5 - Clasificación de métricas

- Botón de Aplicativos

Este link permite ingresar a aplicativos que permitan comunicarse en forma interactiva con un caso u objeto de estudio. Reproduce el comportamiento de un sistema en determinadas condiciones, aplicado generalmente para el entrenamiento. También se presenta el enunciado de dicho caso u objeto de estudio, para que el estudiante pueda analizar las pautas a seguir; y se expone una ayuda que explica paso a paso cómo emplear este aplicativo.

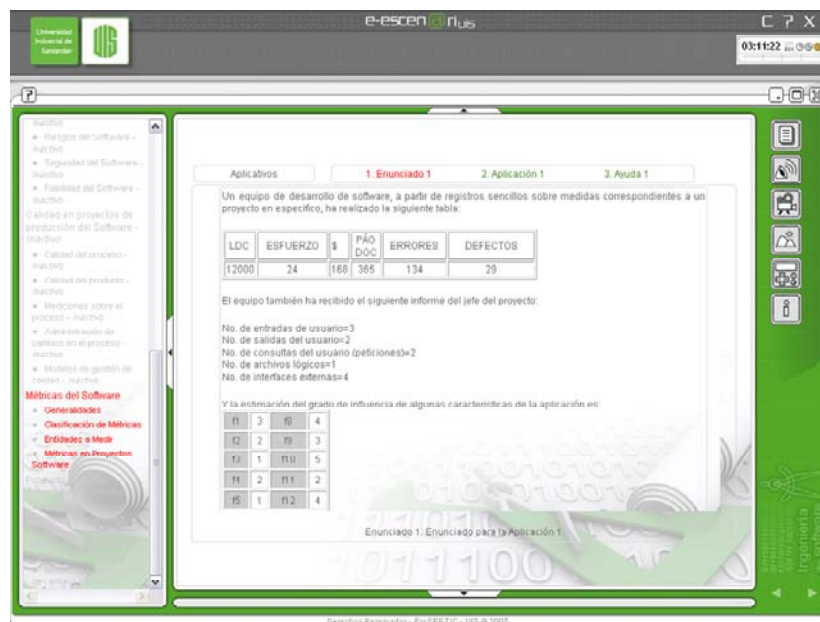


Figura 89. Enunciado del aplicativo - Clasificación de métricas



Figura 90. Ayuda aplicativo - Clasificación de métricas

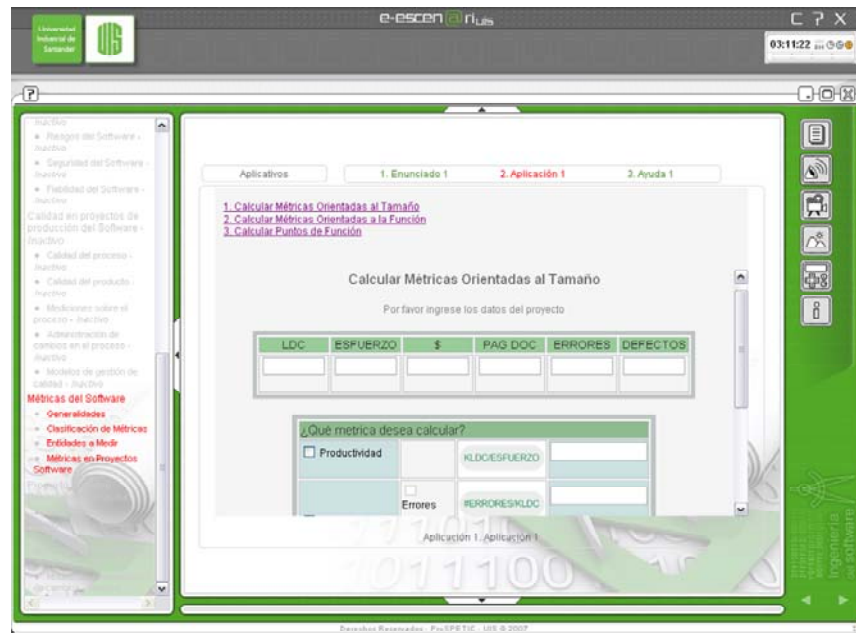


Figura 91. Aplicativo - Clasificación de métricas - opción 1

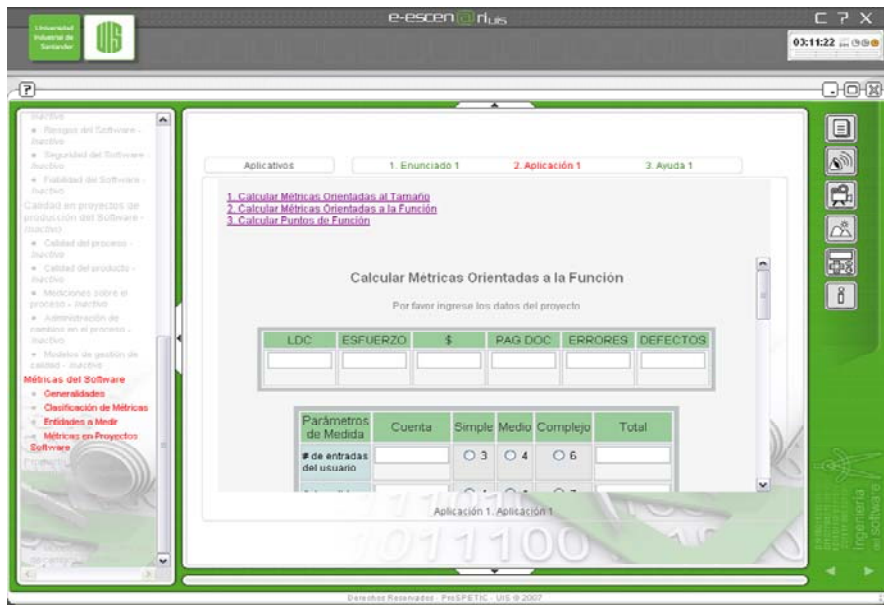


Figura 92. Aplicativo - Clasificación de métricas - opción 2



Figura 93. Aplicativo - Clasificación de métricas - opción 3

- **Botón de gestión de conocimiento**

El contenido de este botón expone la información adicional necesaria para el soporte a la temática planteada.

Se presenta una ficha de créditos que esta conformada por las persona responsables del desarrollo del proyecto, al igual que se explica su papel dentro de éste.

Se muestran las tablas de saber y saber hacer creadas en el diseño instruccional para que el estudiante sepa los conocimientos y destrezas que debe adquirir.

También incluye el DSA² de la asignatura y por medio de una animación se ubica al estudiante en el lugar de la actividad que se encuentra desarrollando.

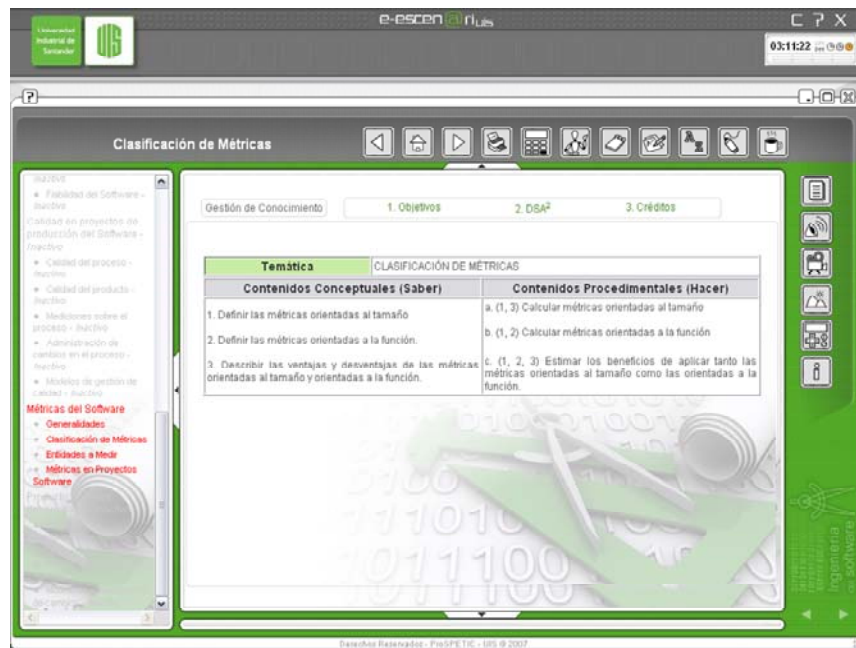


Figura 94. Objetivos - Clasificación de métricas

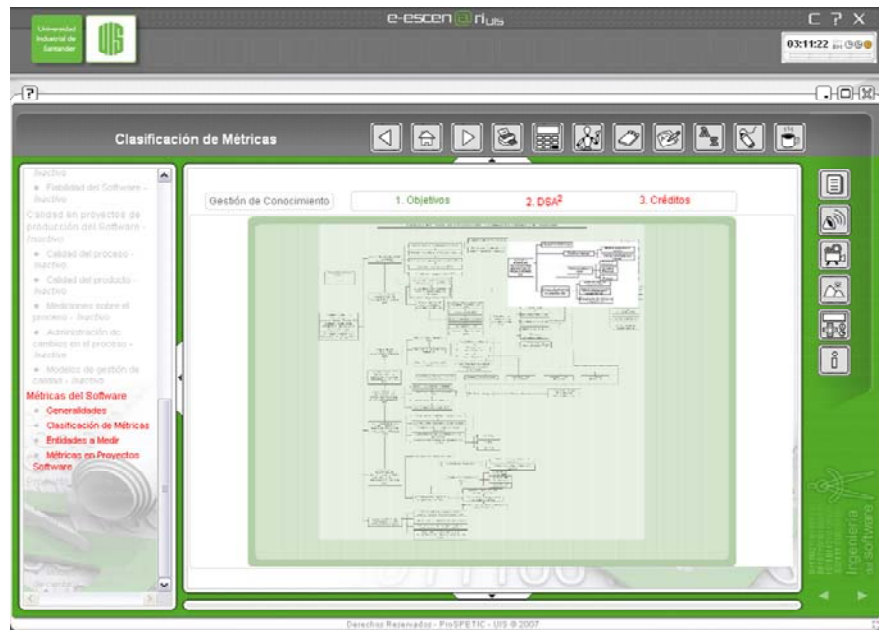


Figura 95. DSA² enfocando las actividades de métricas del software

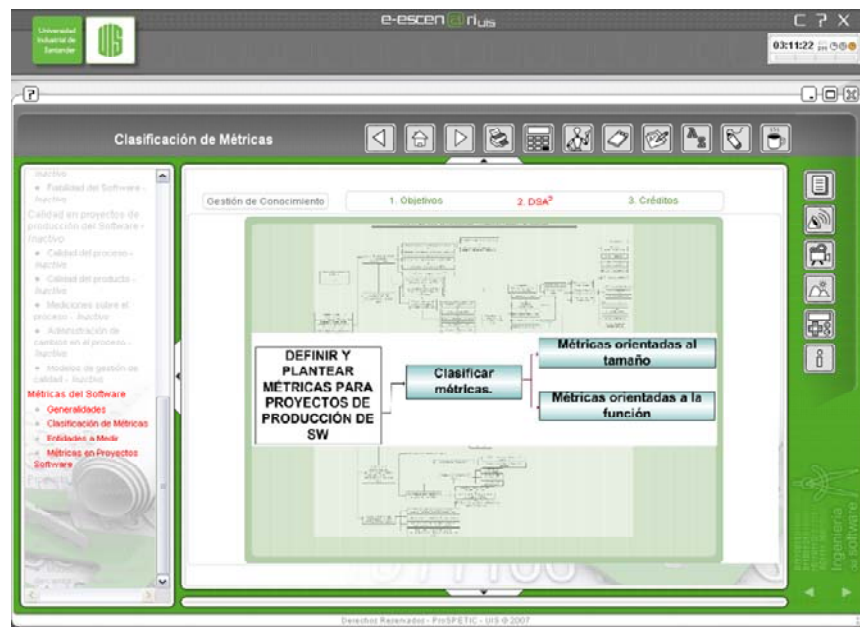


Figura 96. DSA² resaltando la temática “Clasificación de métricas”



Figura 97. Ficha de créditos

4.4.2. Secuenciación de objetos de aprendizaje

Los objetos de aprendizaje generados bajo un mismo contexto de enseñanza, deben ser secuenciados metodológicamente (Diseño Instruccionista) para dar origen a lecciones, y éstas, de igual forma, deben ser secuenciadas para dar origen a cursos que permitan al alumno alcanzar los objetivos de aprendizaje propuestos.

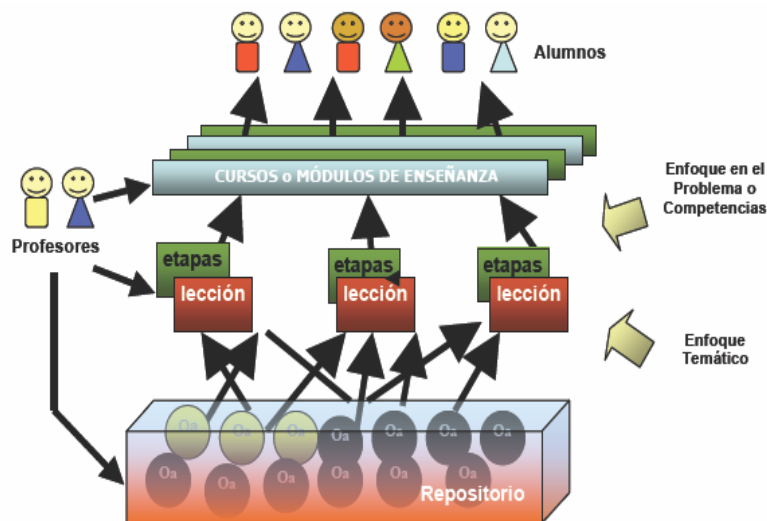


Figura 98. Esquema de secuenciación de objetos de aprendizaje ⁷²

⁷² DISEÑO INSTRUCCIONAL MEDIADO POR TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN (TICs), COMO ESTRATEGIA DE FORMACIÓN BASADA EN COMPETENCIAS; PARA LA ASIGNATURA ESTADÍSTICA I DEL PROGRAMA ACADÉMICO DE INGENIERÍA DE SISTEMAS E INFORMÁTICA. Autores EDWIN HUMBERTO GÓMEZ JIMÉNEZ, SAMUEL RICARDO HERNANDEZ GARCÍA

4.4.3. Ficha de Catalogación de O.A.⁷³

TEMA: GENERALIDADES DE MÉTRICAS DEL SOFTWARE



Figura 99. Núcleo de conocimiento - Generalidades

Título: Animación sobre Generalidades de las métricas y métricas del software.

- **Objetivos Específicos:**
 - Conocer los conceptos necesarios para que el estudiante entre en la relación con el tema.
 - Ilustrar los conceptos de métricas y métricas del software.
 - Representar gráficamente el proceso de calidad del software.
 - Mostrar ejemplos que clarifiquen el concepto de métrica.
 - Exponer gráficamente el proceso de medición.
 - Interpretar el concepto de métricas en el contexto de software.

⁷³ Principios Metodológicos de la Educación Virtual Dra., Clara Inés Peña de Carrillo
<http://gavilan.uis.edu.co/~clarenes>

- **Núcleo de Conocimiento:**

Para la comprensión del tema de las métricas del software es necesario previamente repasar algunos conceptos importantes como: medida, medición, unidad, escala, función de medida. También se muestra el proceso de medición mediante un diagrama que permite ver su secuenciamiento.

Se presenta un ejemplo que aclara la diferencia cuando una medida es una métrica y cuando no lo es.

- **Tiempo de Dedicación:**

Para lograr que el usuario adquiriera la competencia de las definiciones generales de métricas del software empleando el material multimedia que se encuentra en el Objeto de Aprendizaje se estima que debe interactuar por un tiempo aproximado de 15 minutos.

- **Bibliografía y Documentación de Interés:**

- [PRE 97] PRESSMAN, Roger S. "Ingeniería del Software: Un enfoque práctico". 5ª Ed. Madrid, McGraw Hill, 1997.
- [RAG 95] RAGLAND, B., "Measure, Metric or Indicator: What's the Difference?" Crosstalk, vol. 8, n. ° 3, Marzo 1995, 29-30 p.
- [ROC 94] ROCHE, J. M. "Software Metrics and Measurement Principles". Software Engineering Notes, ACM, vol. 19, n° 1, Enero 1994, 5-9 p.



Figura 100. Animación escena 1
Definiciones generales

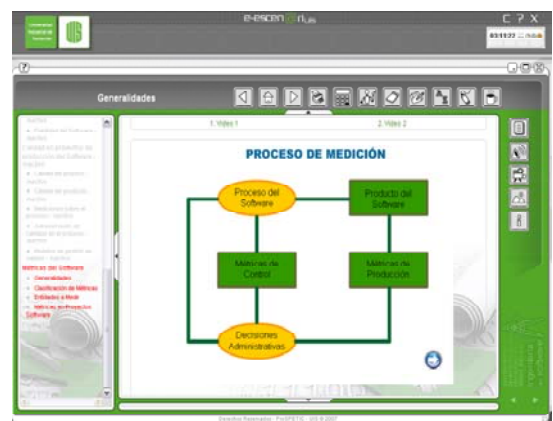
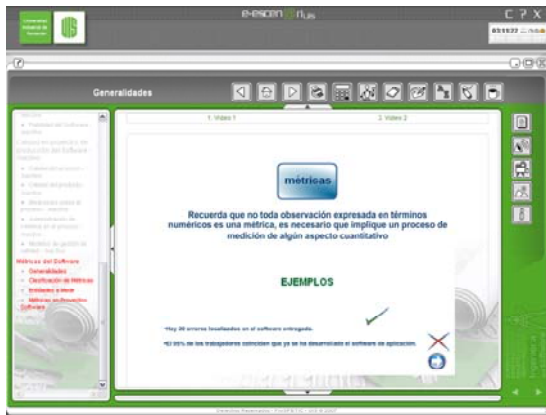


Figura 101. Animación escena 2
Definiciones generales



**Figura 102. Animación escena3
Definiciones generales**



**Figura 103. Animación escena 4
Definiciones generales**

Título: Animación de la historia de las métricas del software.

- **Objetivos Específicos:**

- Describir cómo, cuando y quien dio origen a las métricas del software.
- Ilustrar gráficamente como surgieron las métricas del software.

- **Núcleo de Conocimiento:**

A finales de los años '70, Allan J. Albrecht [ALB 79] [ALB 83] desarrolló una unidad de medida capaz de determinar la funcionalidad de los sistemas software. A esta métrica la llamó "Puntos de Funcionalidad" o, más sencillamente, "Puntos Función". La teoría de Albrecht se basaba en estudiar, por separado, cinco componentes o características principales del sistema software.

- **Tiempo de Dedicación:**

Para lograr que el usuario adquiriera la competencia del tema historia de las métricas del software empleando el material multimedia que se encuentra en el Objeto de Aprendizaje se estima que debe interactuar por un tiempo aproximado de 10 minutos.

- **Bibliografía y Documentación de Interés:**

- http://es.wikipedia.org/wiki/M%C3%A9trica_de_punto_funcion%C3%B3n



**Figura 104. Animación escena 1
Historia de las métricas**

**Figura 105. Animación escena 2
Historia de las métricas**

Título: Documentación Soporte de Generalidades de métricas del software.

▪ **Objetivos Específicos:**

- Clarificar los conceptos fundamentales sobre medición, medidas, métricas y todos sus componentes.
- Exponer los principios básicos de la medición caracterizando sus 5 actividades.
- Presentar el origen de las métricas del software.
- Describir las características fundamentales de las métricas del software.

▪ **Núcleo de Conocimiento:**

Los temas citados en el núcleo de conocimiento de la animación del tema definiciones generales se exponen con más detalle en este documento en el que además se incluye la historia de las métricas, principios de medición y características de estas.

▪ **Tiempo de Dedicación:**

Para la consulta de dicho material junto con algunas de sus referencias bibliográficas se debe emplear un tiempo aproximado de 25 min.

▪ **Bibliografía y Documentación de Interés:**

- [AEMES] VERDÚN, José Carrillo. Asociación de Doctores, Licenciados e Ingenieros en Informática Madrid, 26 de Noviembre de 2003.
- [ALB 79] ALBRECHT, Allan J. "Measuring Applications Development Productivity". Proceedings of IBM Application Development Symposium, Monterrey, CA, 1979, 83-92 p.

- [ALB 83] ALBRECHT, Allan J., y J. E. GAFFNEY. “Software Function, Source Lines of Code and Development Effort Prediction: A Software Science Validation”. IEEE Trans. Software Engineering, Noviembre 1983, 639-648 p.
- [PRE 97] PRESSMAN, Roger S. “Ingeniería del Software: Un enfoque práctico”. 5ª Ed. Madrid, McGraw Hill, 1997.

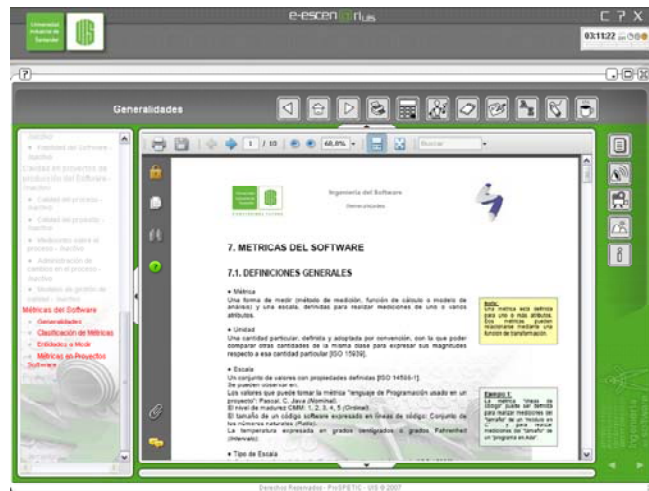


Figura 106. Información de Soporte - Generalidades

Título: Soporte Gráfico y tablas de la temática Generalidades de las métricas del software.

▪ **Objetivos Específicos:**

- Presentar los beneficios que trae la aplicación de medidas en un proyecto de producción de software.
- Mostrar a través de diagramas los procesos de las medidas directas e indirectas.

▪ **Núcleo de Conocimiento:**

Las mediciones del mundo físico se pueden categorizar de dos maneras; medidas directas (ej. La longitud de un tornillo) y medidas indirectas (ej. La cantidad de tornillos producidos medidos contando la cantidad de artículos defectuosos). Cuando recogemos la información cuantitativa que nos ayuda a identificar obstáculos, problemas de raíz, etc., medimos para mejorar.

- **Tiempo de Dedicación:**

El tiempo empleado para el análisis y comprensión del material gráfico y tablas de la temática definiciones generales es de 20 minutos.

- **Bibliografía y Documentación de Interés:**

- <http://www.di.uniovi.es/~cueva/asignaturas/doctorado/2004/MetricasUsabilidad.pdf>
- [PRE 97] PRESSMAN, Roger S. "Ingeniería del Software: Un enfoque práctico". 5ª Ed. Madrid, McGraw Hill, 1997.

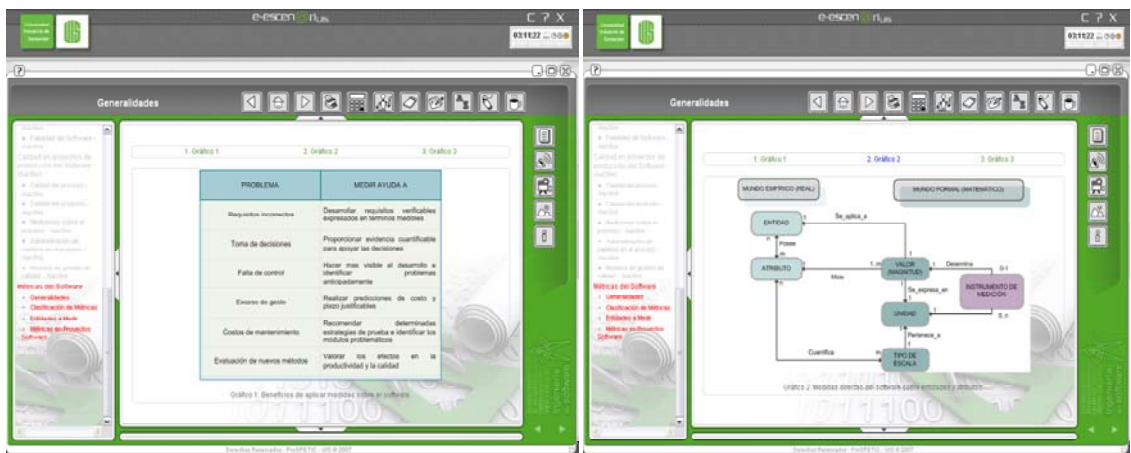


Figura 107. Grafico nº 1 Generalidades

Figura 108. Grafico nº 2 Generalidades

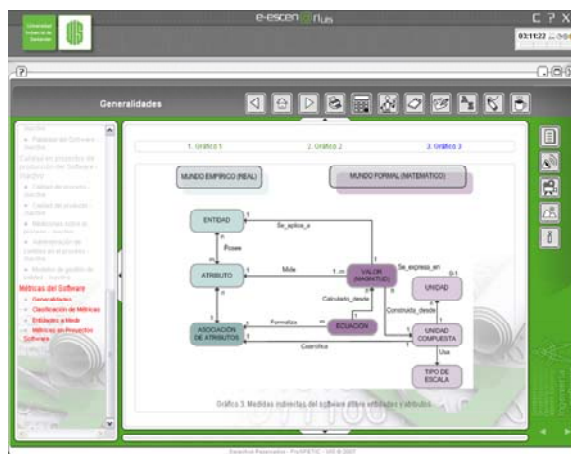


Figura 109. Grafico nº 3 - Generalidades

Título: Soporte de Audio para la temática de Generalidades de métricas del software

▪ Objetivos Específicos:

- Dar soporte y complementar los temas expuestos en el núcleo de conocimiento sobre definiciones generales de las métricas del software.

▪ Núcleo de Conocimiento:

Las métricas del software se refieren a un amplio elenco de mediciones sobre el mismo, estas son reunidas, analizadas y evaluadas por los ingenieros del software. Es importante medir porque si no se hace solo se puede juzgar basado en una evaluación subjetiva.

▪ Tiempo de Dedicación:

El tiempo empleado para el análisis y comprensión de los archivos de audio de la temática “Generalidades de métricas” es de 10 minutos.

▪ Bibliografía y Documentación de Interés:

- [PRE 97] PRESSMAN, Roger S. “Ingeniería del Software: Un enfoque práctico”. 5ª Ed. Madrid, McGraw Hill, 1997.



Figura 110. Archivo de audio - Generalidades

Título: Gestión información para la temática de Generalidades de métricas de software

▪ **Objetivos Específicos:**

- Mostrar la ficha de créditos que está conformada por las persona responsables del desarrollo del proyecto.
- Presentar las tablas de saber y saber hacer creadas en el diseño instruccional para que el estudiante sepa los conocimientos y destrezas que debe adquirir.
- Presentar el DSA² de la asignatura y ubicar al estudiante dentro del diagrama en la temática Generalidades de las métricas del software.

▪ **Tiempo de Dedicación:**

El tiempo empleado para visualizar y analizar la información de gestión es de 15 minutos.



Figura 111. Objetivos – Generalidades



Figura 112. Énfasis en la temática Generalidades



Figura 113. Ficha Créditos de desarrollo

Título: Animación de Clasificación de Métricas⁷⁴

- **Objetivos Específicos:**
 - Ejemplificar paso a paso un problema con métricas orientadas a la función para el cálculo de puntos de función.
 - Presentar los componentes de un sistema con métricas orientadas a la función.
 - Mostrar el procedimiento necesario para calcular puntos de función con base en estas métricas.

- **Núcleo de Conocimiento:**

Los puntos de función se calculan asociando un valor de complejidad a los datos recopilados para los elementos del sistema y realizando la estimación del grado de influencia de algunas características generales de la aplicación.

- **Tiempo de Dedicación:**

Para lograr que el usuario adquiera la competencia del tema “clasificación de métricas” empleando el material multimedia que se encuentra en el Objeto de Aprendizaje se estima que debe interactuar por un tiempo aproximado de 25 minutos.

- **Bibliografía y Documentación de Interés:**
 - [ALB 79] ALBRECHT, Allan J. “Measuring Applications Development Productivity”. Proceedings of IBM Application Development Symposium, Monterrey, CA, 1979, 83-92 p.
 - [ALB 83] ALBRECHT, Allan J., y J. E. GAFFNEY. “Software Function, Source Lines of Code and Development Effort Prediction: A Software Science Validation”. IEEE Trans. Software Engineering, Noviembre 1983, 639-648 p.
 - [PRE 02] PRESSMAN, Roger S. “Ingeniería del Software, un enfoque práctico”. 5ª ED. Madrid, Editorial McGraw Hill, 2002. 59 p. ISBN 0-07-709677-0.

Título: Documentación Soporte Clasificación de métricas del software.

- **Objetivos Específicos:**
 - Explicar los tipos de métricas según el enfoque propuesto por Pressman.
 - Profundizar en las métricas orientadas al tamaño y su proceso de cálculo.

⁷⁴ Las figuras correspondientes al tema “clasificación de métricas” se introdujeron con anterioridad en la explicación de la plantilla

- Detallar conceptos necesarios para calcular métricas orientadas a la función y puntos de función.
 - Establecer la relación entre puntos de función y líneas de código.
 - Evidenciar las ventajas y desventajas de la utilización de estos dos tipos de métricas.
- **Núcleo de Conocimiento:**
La clasificación de una métrica de software refleja o describe la conducta del software. Muchos investigadores han intentado desarrollar una sola métrica que facilite una medida completa de la complejidad del software.

Aunque se han presentado docenas de medidas de complejidad, cada una tiene un punto de vista distinto de lo que es la complejidad y de qué atributos de un sistema llevan a la complejidad.

- **Tiempo de Dedicación:**
Para la consulta de dicho material junto con algunas de sus referencias bibliográficas se recomienda emplear un tiempo aproximado de 40 minutos.
- **Bibliografía y Documentación de Interés:**
 - [ALB 83] ALBRECHT, Allan J., y J. E. GAFFNEY. "Software Function, Source Lines of Code and Development Effort Prediction: A Software Science Validation". IEEE Trans. Software Engineering, Noviembre 1983, 639-648 p.
 - [IFP 94] "Function Point Counting Practices Manual", Release 4.0, International Function Point Users Group (IFPUG), 1994.
 - [JON 98] JONES, Casper. "Estimating Software Cost". Editorial McGraw Hill, 1998.
 - [GRA 94] GRADY, R., B. "Succesfully Applying Software Metrics", Computer, vol. 27, nº9, Septiembre 1994, 18-25 p.
 - www.ifpug.com

Título: Soporte Gráfico a la temática Clasificación de métricas del software.

- **Objetivos Específicos:**
 - Mostrar visualmente la clasificación de las métricas según Pressman.
 - Detallar los pasos y datos necesarios para calcular métricas orientadas al tamaño.
 - Especificar los pasos y datos necesarios para calcular métricas orientadas a la función.
 - Exponer las características de la aplicación sobre las cuales se calcula el factor de complejidad.

- **Núcleo de Conocimiento:**
Pressman clasifica el campo de las métricas en métricas técnicas, métricas de calidad y métricas de productividad. Estas últimas se conforman de las métricas orientadas al tamaño y orientadas a la función.
- **Tiempo de Dedicación:**
El tiempo empleado para el análisis y comprensión del material gráfico y tablas requiere aproximadamente de 20 minutos.
- **Bibliografía y Documentación de Interés:**
 - [PRE 02] PRESSMAN, Roger S. "Ingeniería del Software: Un enfoque práctico". 5ª Ed. Madrid, McGraw Hill, 2002.

Título: Soporte de Audio para la temática de Clasificación de las métricas del software

- **Objetivos Específicos:**
 - Dar soporte y complementar los temas expuestos en el núcleo de conocimiento.
- **Núcleo de Conocimiento:**
La clasificación de una métrica del software refleja o describe la conducta del mismo. Las métricas orientadas al tamaño se basan en la medición del número de LDC para proporcionar medidas directas del software y las métricas orientadas a la función se centran en la funcionalidad o utilidad del programa.
- **Tiempo de Dedicación:**
El tiempo empleado para el análisis y comprensión de los archivos de audio es de aproximadamente 10 minutos.
- **Bibliografía y Documentación de Interés:**
 - [PRE 97] PRESSMAN, Roger S. "Ingeniería del Software: Un enfoque práctico". 4ª Ed. Madrid, McGraw Hill, 1997.

Título: Aplicativo del cálculo de diferentes tipos de métricas del software⁷⁵

- **Objetivos Específicos:**
 - Sintetizar el proceso del cálculo de métricas orientadas a la función, al tamaño y/o puntos de función.

⁷⁵ todas las gráficas correspondientes al tema "clasificación de métricas" se introdujeron con anterioridad en la explicación de la plantilla

- Especificar los elementos necesarios para el cálculo de distintas clases de métricas.
 - Establecer el tipo de relación entre los datos necesarios para calcular los diferentes tipos de métricas.
 - Determinar las condiciones bajo las cuales se calcula un cierto tipo de métrica.
- **Núcleo de Conocimiento:**
Las métricas del software orientadas al tamaño provienen de la normalización de medidas que tienen en consideración el tamaño del software que se hay a producido. Si una organización mantiene registros sencillos, podrá crear una tabla de datos orientados al tamaño. Las métricas del software orientadas a la función utilizan una medida de la funcionalidad entregada por la aplicación como valor de la normalización a través de los puntos de función, una vez calculados se pueden obtener medidas de productividad, calidad y otros atributos del software.
 - **Tiempo de Dedicación:**
Para lograr que el usuario adquiera la competencia del tema: cálculo de diferentes tipos de métricas del software, empleando el material multimedia que se encuentra en el Objeto de Aprendizaje se estima que debe interactuar por un tiempo aproximado de 45 minutos.
 - **Bibliografía y Documentación de Interés:**
 - [PRE 97] PRESSMAN, Roger S. "Ingeniería del Software: Un enfoque práctico". 5ª Ed. Madrid, McGraw Hill, 1997.
 - [ALB 83] ALBRECHT, Allan J., y J. E. GAFFNEY. "Software Function, Source Lines of Code and Development Effort Prediction: A Software Science Validation". IEEE Trans. Software Engineering, Noviembre 1983, 639-648 p.
 - www.ifpug.org

Título: Gestión de Conocimiento para la temática de Clasificación de métricas del software

- **Objetivos Específicos:**
 - Mostrar la ficha de créditos que está conformada por las persona responsables del desarrollo del proyecto.
 - Exponer las tablas de saber y saber hacer creadas en el diseño instruccional para que el estudiante sepa los conocimientos y destrezas que debe adquirir.
 - Presentar el DSA² de la asignatura y ubicar al estudiante dentro del diagrama en la temática clasificación de métricas.

- **Tiempo de Dedicación:**
El tiempo empleado para visualizar y analizar la información de gestión es de 15 minutos.

TEMA: ENTIDADES A MEDIR



Figura 114. Núcleo de conocimiento - Entidades a medir

Título: Animación de Entidades y atributos del software

- **Objetivos Específicos:**
 - Citar ejemplos de atributos y ubicarlos dentro de la entidad a la que pertenecen.
 - Nombrar las clases de entidades y atributos presentes en un proyecto de producción de software.
 - Establecer las relaciones presentes entre entidades (recurso, proceso y producto) y atributos externos e internos.

Núcleo de Conocimiento:

Cualquier cosa que queramos medir o predecir en un software es un atributo de cualquier entidad de un producto, proceso o recurso asociado a éste. Cada entidad de software tiene varios atributos que pueden ser medidos. Es por eso que se necesita hacer una distinción entre atributos que son internos o externos.

- **Tiempo de Dedicación:**
Para lograr que el usuario adquiera la competencia del tema entidades a medir empleando el material multimedia que se encuentra en el Objeto de

Aprendizaje se estima que debe interactuar por un tiempo aproximado de 25 minutos.

▪ **Bibliografía y Documentación de Interés:**

- www.sei.cmu.edu
- [FEN 91] FENTON, N. "Software Metrics". Chapman & Hall. 1991.
- [HUM 95] HUMPHREY, W. "A Discipline for Software Engineering". Addison-Wesley, 1995.



Figura 115. Animación escena 1.
Entidades a medir



Figura 116. Animación escena 2.
Entidades a medir⁷⁶

Título: Información de Soporte para la temática Entidades a medir del software

▪ **Objetivos Específicos:**

- Exponer las relaciones existentes entre atributos y entidades a medir.
- Conocer el concepto de atributo dentro del contexto de las métricas del software.
- Distinguir las entidades que son objeto de medición en proyectos de producción de software.

▪ **Núcleo de Conocimiento**

La primera obligación en cualquier actividad de medición de software es identificar las entidades y atributos de interés que se desea medir. Sabiendo de antemano que una entidad es un objeto o un evento y un atributo son las características o propiedades del software a medir.

▪ **Bibliografía y Documentación de Interés:**

- [FEN 91] FENTON, N. "Software Metrics". Chapman & Hall. 1991.

⁷⁶ Las escenas siguientes no se introdujeron ya que tienen el mismo diseño de la escena 2.

- [PAU 94] PAULISH, D., y A CARLETON, “Case Studies of Software Process Improvement Measurement”, Computer, vol. 27, n. ° 9, Septiembre 1994, 50-57.
- [PRE 02] PRESSMAN, Roger S. “Ingeniería del Software: Un enfoque práctico”. 5ª Ed. Madrid, McGraw Hill, 2002.



Figura 117. Información de Soporte – Entidades a medir

Título: Soporte Gráfico a la temática Entidades a medir del software.

- **Objetivos Específicos:**
 - Describir cada una de las entidades a medir en proyectos de producción de software.
 - Establecer los recursos que se deben estimar en un proyecto.
 - Listar los requerimientos de medición necesarios para la calidad de un proyecto.
 - Presentar la entidad proceso como una relación directa entre las personas, el producto y la tecnología.
- **Núcleo de Conocimiento:**

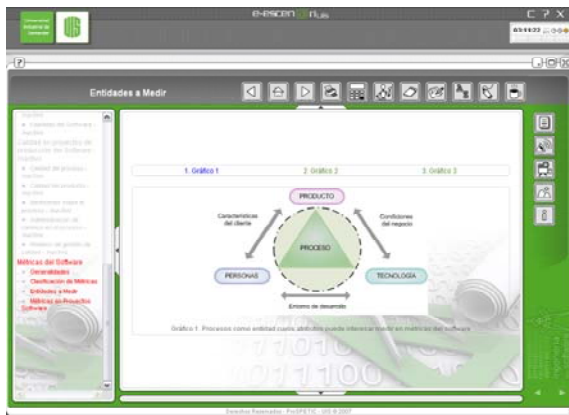
Los atributos externos de un producto, proceso o recurso son aquellos que solamente pueden ser medidos con respecto a cómo el producto, proceso o recurso se relacionan con su ambiente. Estos tienden a ser los que el administrador y el usuario del software comúnmente gustan de medir y predecir.

- **Tiempo de Dedicación:**

El tiempo empleado para el análisis y comprensión del material gráfico es aproximadamente de 20 minutos.

- **Bibliografía y Documentación de Interés:**

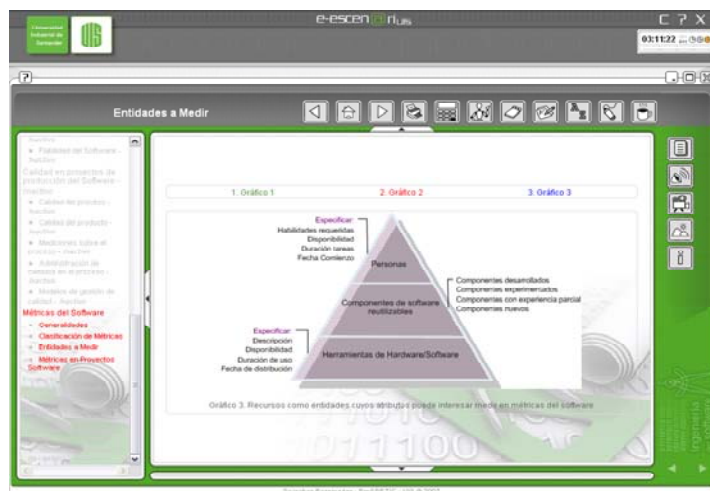
- [FEN 91] FENTON, N. "Software Metrics". Chapman & Hall. 1991.
- [PAU 94] PAULISH, D., y A CARLETON, "Case Studies of Software Process Improvement Measurement", Computer, vol. 27, n. ° 9, Septiembre 1994, 50-57.
- [PRE 02] PRESSMAN, Roger S. "Ingeniería del Software: Un enfoque práctico". 5ª Ed. Madrid, McGraw Hill, 2002.



**Figura 118. Gráfica nº 1
Entidades a medir**



**Figura 119. Gráfica nº 2
Entidades a medir**



**Figura 120. Gráfica nº 3
Entidades a medir**

Título: Soporte de Audio para la temática de Entidades a medir del software

- **Objetivos Específicos:**
 - Complementar y dar soporte a la temática relacionada con entidades a medir del software.

- **Núcleo de Conocimiento:**

La estimación de los recursos requeridos para acometer el esfuerzo de desarrollo de software esta compuesta por personas, componentes del software reutilizables y herramientas hardware/software. Las entidades de procesos de software incluyen actividades relacionadas con el software y eventos que usualmente son asociados con un factor de tiempo. Los productos del software son las salidas del proceso de producción del software, éstas incluyen todos los artefactos entregados o documentos que son productos durante el ciclo de vida del software.

- **Tiempo de Dedicación:**

El tiempo empleado para el análisis y comprensión de los archivos de audio es aproximadamente de 10 minutos.

- **Bibliografía y Documentación de Interés:**
 - [FEN 91] FENTON, N. "Software Metrics". Chapman & Hall. 1991.
 - [PAU 94] PAULISH, D., y A CARLETON, "Case Studies of Software Process Improvement Measurement", Computer, vol. 27, n. ° 9, Septiembre 1994, 50-57.
 - [PRE 02] PRESSMAN, Roger S. "Ingeniería del Software: Un enfoque práctico". 5ª Ed. Madrid, McGraw Hill, 2002.

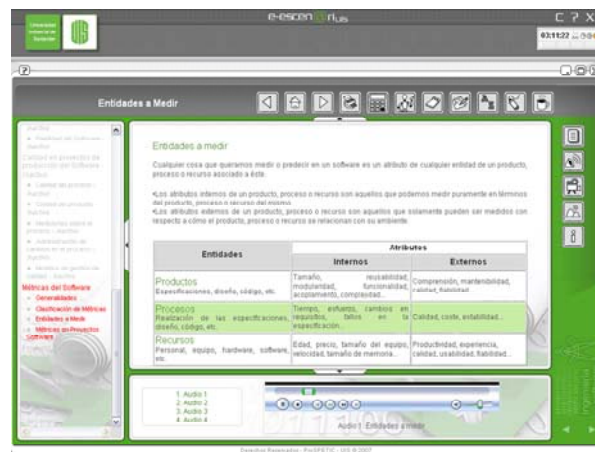


Figura 121. Archivo de audio
Entidades a medir

Título: Gestión de conocimiento para la temática Entidades a medir del software

- **Objetivos Específicos:**
 - Mostrar la ficha de créditos que está conformada por las persona responsables del desarrollo del proyecto.
 - Exponer las tablas de saber y saber hacer creadas en el diseño instruccional para que el estudiante sepa los conocimientos y destrezas que debe adquirir.
 - Presentar el DSA² de la asignatura y ubicar al estudiante dentro del diagrama en la temática “entidades a medir del software”.

- **Tiempo de Dedicación:**
El tiempo empleado para visualizar y analizar la información de gestión es de 15 minutos.



Figura 122. Objetivos Entidades a medir

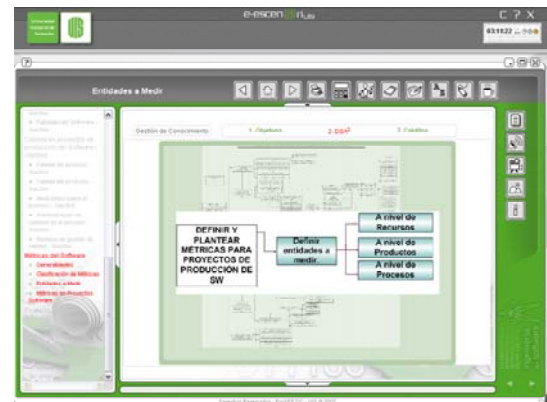


Figura 123. Énfasis en la temática Entidades a medir



Figura 124. Ficha créditos de desarrollo

TEMA: MÉTRICAS EN PROYECTOS SOFTWARE

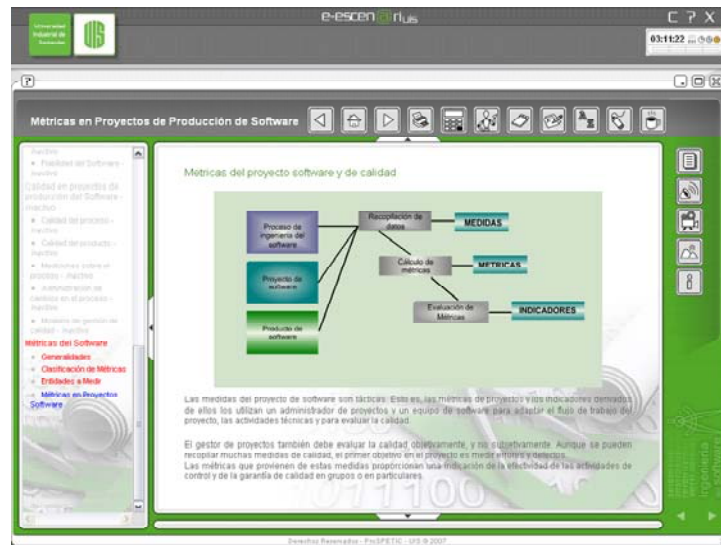


Figura 125. Núcleo de conocimiento - Métricas en proyectos de software

Título: Animación de factores determinantes de métricas de calidad en proyectos software

- **Objetivos Específicos:**
 - Ilustrar enfoques de factores de calidad desde el punto de vista de diferentes autores.
 - Identificar las medidas de calidad necesarias para aplicar métricas en proyectos de producción de software a través de los factores de calidad del software.
 - Exponer algunos criterios de medición y su relación con los diferentes enfoques de factores de calidad para el establecimiento de métricas de calidad del software.
 - Diferenciar las características de calidad del software presentes en un producto software.
- **Núcleo de Conocimiento:**

La calidad del software es una compleja mezcla de factores que varían a través de diferentes aplicaciones y según los clientes que las pidan, los factores que afecta la calidad del software se pueden categorizar en diferentes grupos según sus autores.
- **Tiempo de Dedicación:**

Para lograr que el usuario adquiera la competencia del tema métricas en proyectos de software empleando el material multimedia que se encuentra

- Explicar cómo están integradas las métricas en proyectos de software.
- **Núcleo de Conocimiento:**
El objetivo primordial de la Ingeniería del Software es producir un sistema, aplicación o producto de alta calidad. Para lograr este objetivo los ingenieros del software debe aplicar métodos efectivos junto con herramientas modernas dentro del contexto de un proceso maduro de desarrollo de software.
- **Tiempo de Dedicación:**
Para la consulta de dicho material junto con algunas de sus referencias bibliográficas se debe emplear un tiempo aproximado de 30 min.
- **Bibliografía y Documentación de Interés:**
 - [MCC 77] McCALL, J., P. RICHARDS, y G. WALTERS. “Factors in Software Quality”, 3 vols. NTIS AD-A049-014, 015, 055. Noviembre 1977.
 - [PRE 02] PRESSMAN, Roger S. “Ingeniería del Software, un enfoque práctico”. 5ª ed. Madrid, Editorial McGraw Hill, 2002. 59 p. ISBN 0-07-709677-0.
 - [CAV 78] CAVANO, J.P., y J.A. McCALL. “A Framework for the Measurement of Software Quality”. Proc. ACM Software Quality Assurance Wokshop. Noviembre 1978. 133-139 p.



Figura 129. Información de Soporte - Métricas en proyectos software

Título: Soporte Gráfico y tablas de la Temática métricas en proyectos de software.

- **Objetivos Específicos:**
 - Representar de forma gráfica el proceso para obtener medidas del software a partir de los atributos de calidad del mismo.
 - Establecer la relación entre las métricas y los factores de calidad.
 - Asociar los factores de calidad con las entidades a medir y las métricas del software.



Figura 132. Gráfica nº 3
Métricas en proyectos software

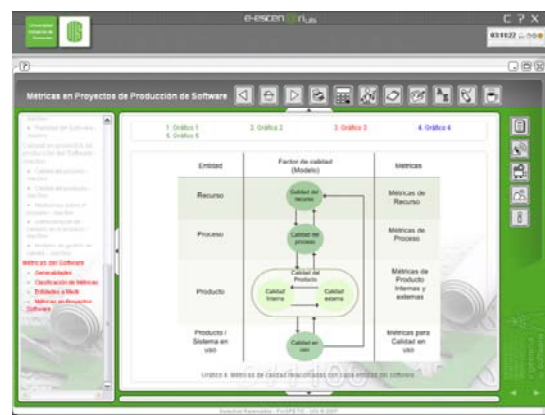


Figura 133. Gráfica nº 4
Métricas en proyectos software

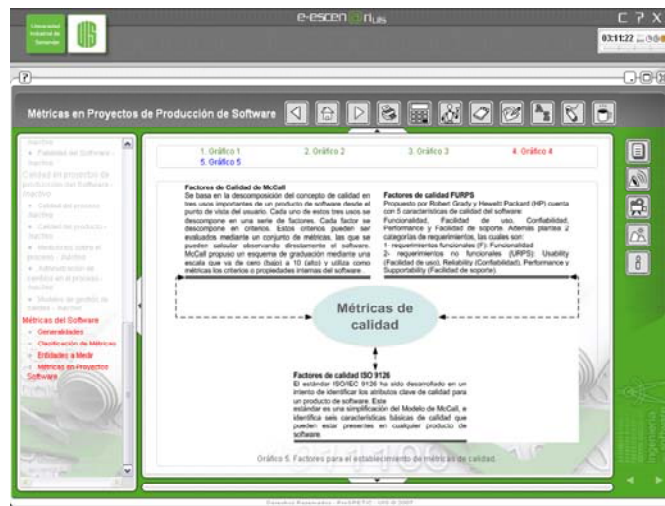


Figura 134. Gráfica nº 5 - Métricas en proyectos software

Título: Soporte de Audio para la Temática de métricas en proyectos de software

- **Objetivos Específicos:**
 - Soportar la temática de métricas en proyectos de software y métricas de calidad.
 - Exponer la utilidad de las métricas en un proyecto de producción de software.
- **Núcleo de Conocimiento:**

Las métricas en un proyecto de software se utilizan para minimizar la planificación de desarrollo guiando los ajustes necesarios que eviten

retrasos y atenúen riesgos potenciales. También se utilizan para evaluar la calidad del producto en el momento actual.

- **Tiempo de Dedicación:**
El tiempo empleado para el análisis y comprensión de los archivos de audio es aproximadamente de 15 minutos.

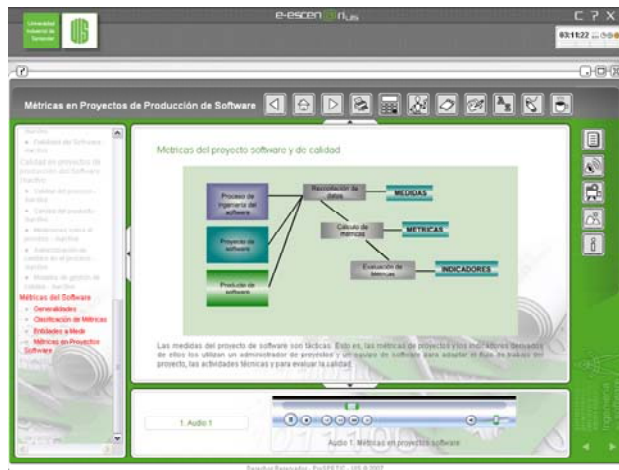


Figura 135. Archivos de audio - Métricas en proyectos software

- **Título: Gestión de conocimiento para la temática Métricas en proyectos software**
- **Objetivos Específicos:**
 - Mostrar la ficha de créditos que esta conformada por las personas responsables del desarrollo del proyecto.
 - Exponer las tablas de saber y saber hacer creadas en el diseño instruccional para que el estudiante sepa los conocimientos y destrezas que debe adquirir.
 - Presentar el DSA² de la asignatura y ubicar al estudiante dentro del diagrama en la temática “entidades a medir del software”.
- **Tiempo de Dedicación:**
El tiempo empleado para visualizar y analizar la información de gestión es de 15 minutos.
- **Bibliografía y Documentación de Interés:**
 - [MCC 77] McCALL, J., P. RICHARDS, y G. WALTERS. “Factors in Software Quality”, 3 vols. NTIS AD-A049-014, 015, 055. Noviembre 1977.

- [PRE 02] PRESSMAN, Roger S. “Ingeniería del Software, un enfoque práctico”. 5ª ed. Madrid, Editorial McGraw Hill, 2002. 59 p. ISBN 0-07-709677-0.
- [CAV 78] CAVANO, J.P., y J.A. McCALL. “A Framework for the Measurement of Software Quality”. Proc. ACM Software Quality Assurance Wokshop. Noviembre 1978. 133-139 p.
- www.sei.cmu.edu
- www.isse.gmu.edu/faculty/ofut/rsrch/abstracts/mj-coupling.html



Figura 136. Objetivos Métricas en proyectos software



Figura 137. Énfasis en la temática Métricas en proyectos software

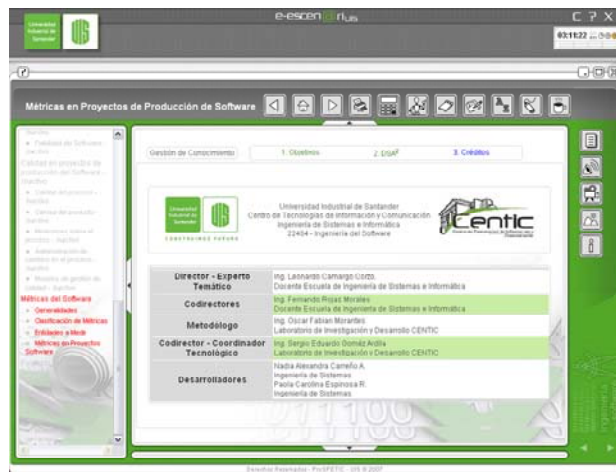


Figura138. Ficha créditos de desarrollo

5. ESTRUCTURACIÓN Y PUESTA EN FUNCIONAMIENTO DEL PORTAL WEB DEL PROFESOR

El propósito de esta etapa es mostrar el quehacer docente e investigativo del profesor UIS en Internet y de crear la cultura del trabajo en la red, el proyecto propone y desarrolla el portal del profesor con los siguientes fines:

- ❖ Estructurar los recursos (información estática) utilizados como soporte o la enseñanza/aprendizaje de las asignaturas: lecturas, presentaciones, tareas, enlaces de interés, noticias, listas de clase, fichas de alumnos etc.
- ❖ Promover la investigación, publicaciones, áreas de interés, contactos, revistas electrónicas.
- ❖ Promover la comunicación entre profesores, estudiantes e investigadores.⁷⁷

5.1. LINK INICIO (figura 139)

Se observará una página principal, como se muestra en la figura. En ésta se encuentra la presentación del docente Leonardo Augusto Camargo Corzo. Se tiene la opción del navegador (Google).

- ❖ **Noticias:** Se pueden visualizar en este link las noticias actualizadas de temas de interés del estudiante; al seleccionar alguna noticia, esta puede estar en formato PDF, PPT o enlazarla a la página Web relacionada a la Noticia.

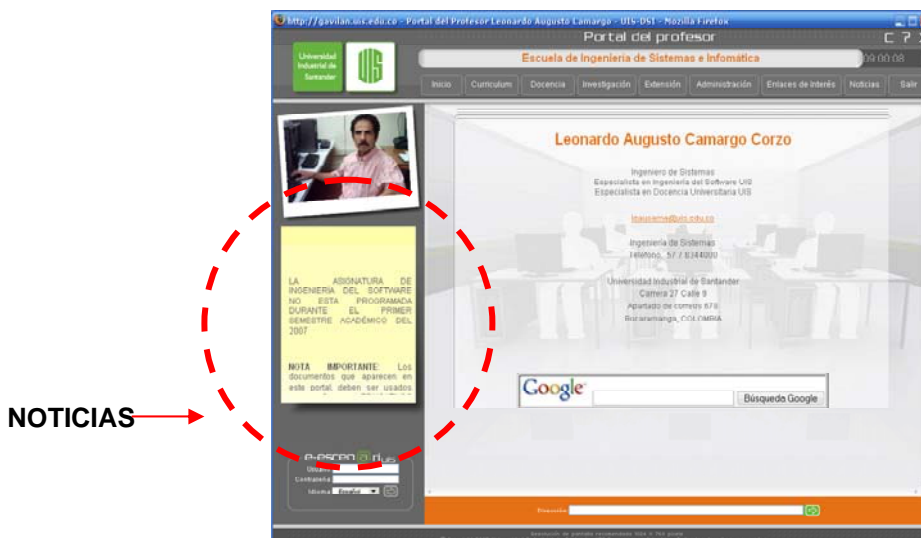


Figura 139. Aspecto general del portal del profesor

⁷⁷ Fuente portal de la profesora Clara Inés Peña <http://gavilan.uis.edu.co/~clarenes/>

6. ELABORACIÓN DE LA EVALUACIÓN DEL OBJETO DE APRENDIZAJE

Como se había planteado en el numeral 4.1.5. (SOBRE LA EVALUACIÓN DEL OBJETO DE APRENDIZAJE), la plataforma **e-escen@riUIS** cuenta con un gestor de evaluación que apoya el aprendizaje por competencias por parte del estudiante y que da soporte al docente en el proceso de valoración sobre la evolución del alumno.

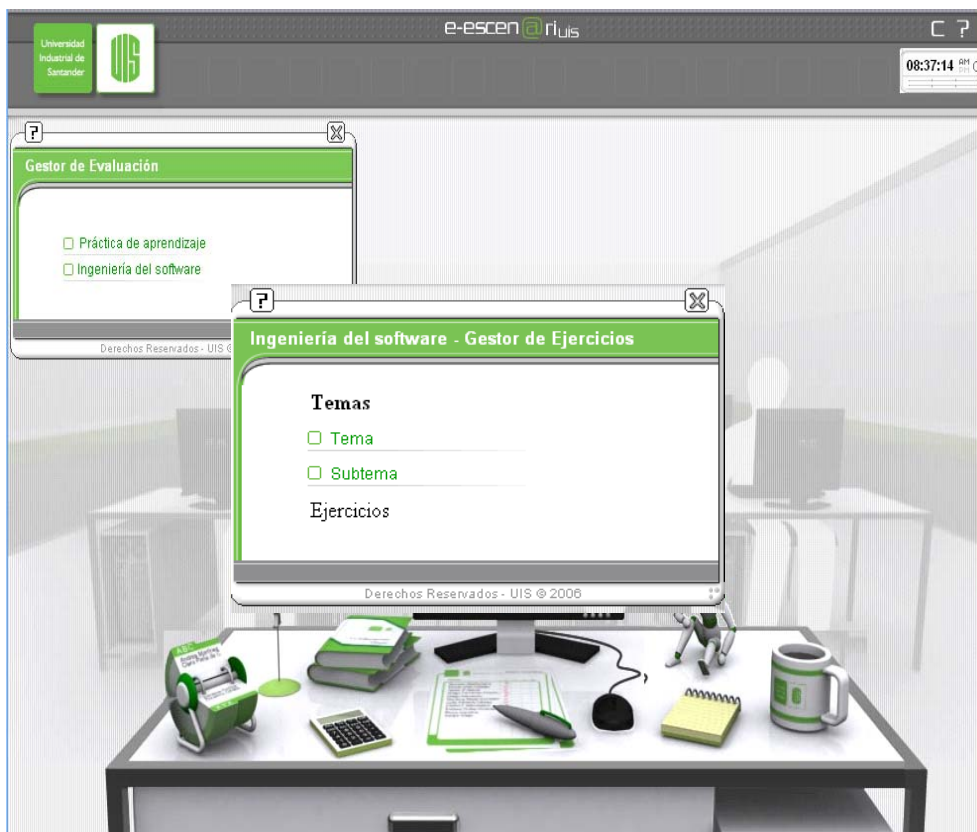


Figura 145. Gestor de ejercicios de la plataforma e-escen@riUIS

Este gestor de conocimientos agrupa las preguntas por temas y subtemas y permite abordar la evaluación por medio de 7 tipos de ejercicios diferentes: Asociación, completar, ordenar, selección, sopa de letras, cuestionario académico y pregunta abierta. Todo esto con el objetivo de ofrecer tanto al docente como al estudiante las herramientas necesarias que complementen su quehacer en el ámbito de la enseñanza-aprendizaje.

El gestor de evaluación ha sido diseñado para evaluar por competencias, éstas se incorporan de 3 maneras:

- Competencias argumentativas
- Competencias Interpretativas
- Competencias Propositivas

Con ellas se busca evaluar la capacidad del estudiante para articular conceptos y teorías, sustentar conclusiones y propuestas, encontrar sentido a textos, problemas, esquemas, proponer hipótesis y dar alternativas de solución.

A continuación se plantean algunos de los ejercicios que se han propuesto para la temática de Métricas del Software de la asignatura Ingeniería del Software y que se encuentran incorporados a la plataforma **e-escen@riUIS**.

PREGUNTA No.1

TEMA	Métricas del Software
SUBTEMA	Generalidades
CATEGORÍA	Normal
COMPETENCIA	Interpretativa
TIPO PREGUNTA	Sopa de letras

ENUNCIADO:

Encontrar en la sopa de letras 7 palabras relacionadas con la medición del software

F K N O I C I D E M A Q W D G U R W
R N L O P D T F G U S B N I M U T A
G U O N O I C A T E R P R E T N I D
I U O I R E H J A D R G N O C H Y G
E J K U C T E G A J B E S L O H I H
S A O B M A S T I C O L D A L F O O
C E L C O L T E P I E C F D E S K L
A U P D T S I N E T N A T I C D P I
L V M A I E S O E A A F O T C C O N
A R J O T N U R G M N E L R I N I D
O O G I E O S R E O I M M E O Y U I
L L D H R R I E N M I L E N N S T C
M J A M C S A N O I R E A T I A U A
S I S I L A N A I B A S O E K X J D
A G U D F H L T E R N B I C R G I O
T S D R U V S E F H M L O I J Y Y R
O V U F O R M U L A C I O N G I G D
J U E V I N D R T S C H I E R H C A

RESPUESTA:

NOI C I D E M
N O N O I C A T E R P R E T N I
I C O
E S C A L A C O L E C I O N I
S C A L A C O L E C I O N I
L A C O L E C I O N I
A C O L E C I O N I
S I S I L A N A E R
F O R M U L A C I O N

PREGUNTA No. 2

TEMA	Métricas del Software
SUBTEMA	Clasificación de Métricas
CATEGORÍA	Normal
COMPETENCIA	Argumentativa
TIPO PREGUNTA	Selección

ENUNCIADO:

Uno de los inconvenientes de las LDC (Líneas de código) es:

- A:** Resulta arduo formar al personal en su utilización
- B:** Aíslan el tamaño intrínseco del sistema de los factores del medio
- C:** Perjudican a los programas cortos pero bien diseñados
- D:** No se calculan con base en un estándar

RESULTADO:

C: Perjudican a los programas cortos pero bien diseñados

RETROALIMENTACIÓN:

Los ofensores de las medidas de LDC argumentan que éstas son dependientes del lenguaje de programación y por ende perjudican a los programas más cortos y que no pueden acomodar fácilmente lenguajes procedimentales.

PREGUNTA 3.

TEMA	Métricas del Software
SUBTEMA	Clasificación de Métricas
CATEGORÍA	Normal
COMPETENCIA	Interpretativa
TIPO PREGUNTA	Selección

ENUNCIADO:

Usted es miembro de un equipo de desarrollo de software y recibe la siguiente información:

Parámetro de medida	Cuenta	Factor de peso			
		Simple	Medio	Complejo	
No. Entradas	28	3	4	6	---
No. Salidas	32	4	5	7	---
No. Peticiones	123	3	4	6	---
No. de Archivos	59	7	10	15	---
No. de Interfaces	60	5	7	10	---
Cuenta Total =					---

Le indican, además, que resulta complejo el procesamiento interno del sistema (influencia 4) y que el método para realizar las actualizaciones de los archivos maestros tiene una influencia promedio en la complejidad (influencia 3).

Le piden calcular los puntos de función y el resultado que usted obtiene es:

- A: 1563, 84 PF
- B: 1411, 87 PF
- C: 10034, 64 PF
- D: 2172 PF

RESPUESTA:

A: 1563, 84 PF

RETROALIMENTACIÓN:

Los puntos de función se calculan con la ecuación:

$$PF = \text{Cuenta Total} * [0,65 + 0.0.1 * \Sigma Fi]$$

La Cuenta Total es la suma de todos los datos obtenidos de multiplicar cada uno de los valores de las cuentas por su respectivo factor de peso en la tabla, y ΣFi es la sumatoria de los factores de complejidad de las características de la aplicación, en este caso solo se consideraron 2 y la suma de su puntuación de complejidad ha sido estimada en 7. Al remplazar estos valores en la ecuación obtenemos el resultado indicado.

PREGUNTA 4.

TEMA	Métricas del Software
SUBTEMA	Entidades a medir
CATEGORÍA	Normal
COMPETENCIA	Propositiva
TIPO PREGUNTA	Completar

ENUNCIADO.

La primera obligación en cualquier actividad de medición de software es identificar las _____ y los _____ de interés.

RESPUESTA.

La primera obligación en cualquier actividad de medición de software es identificar las **entidades** y los **atributos** de interés.

RETROALIMENTACIÓN.

Cualquier cosa que queramos medir o predecir en un software es un atributo de cualquier entidad de un producto, proceso o recurso asociado a éste.

Cada entidad de software tiene varios atributos que pueden ser medidos. Es por eso que se necesita hacer una distinción entre atributos que son internos o externos.

PREGUNTA 5.

TEMA	Métricas del Software
SUBTEMA	Métricas en proyectos software
CATEGORÍA	Normal
COMPETENCIA	Propositiva
TIPO PREGUNTA	Ordenar

ENUNCIADO.

Exacta, software medir la parcialmente calidad del medida de forma No ya posible que es imperfecta cada es.

RESPUESTA.

No es posible medir la calidad del software de forma exacta, ya que cada medida es parcialmente imperfecta.

RETROALIMENTACIÓN.

Las medidas de calidad siempre son indirectas, ya que no se mide directamente la calidad sino algunas de sus manifestaciones. A veces, cuando se intentan obtener medidas precisas de la calidad del software se acaba frustrado por la naturaleza subjetiva de esta actividad. Para resolver este problema, se buscan medidas

cuantitativas de la calidad del software, para poder llevar a cabo un análisis objetivo.

7. EMPAQUETAMIENTO DEL OBJETO DE APRENDIZAJE

Existen varias herramientas para “empaquetar” contenidos y una buena opción gratuita y de código abierto es el editor ReLOAD⁷⁸ ó (*Reusable eLearning Object Authoring & Delivery*), se puede editar, previsualizar y empaquetar contenidos

En la figura 1, se representa el esquema general de un empaquetador de contenidos como el editor ReLOAD.

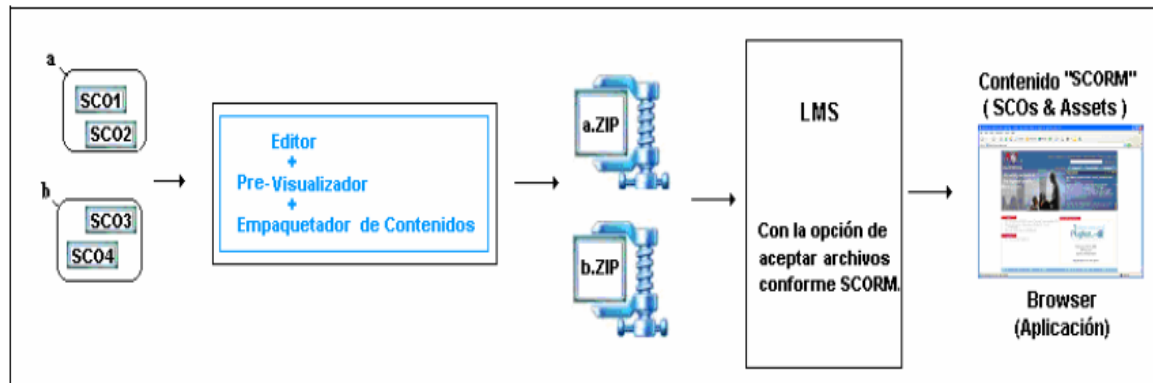


Figura 146. Modelo Generalizado del Editor ReLOAD.

Reload es un empaquetador de contenidos y editor de metadatos de código abierto, destinado a compartir material de enseñanza-aprendizaje. Con Reload Editor se puede tomar cualquier contenido electrónico (páginas Web, imágenes, video, animaciones flash, applets Java, etc.), empaquetarlo y prepararlo para almacenarlo en sitios dedicados a estos temas.

Reload permite las siguientes funciones:

- Crear, importar, editar y exportar paquetes de contenidos.
- Empaquetar contenidos creados con otras herramientas.
- Reorganizar y recatalogar los contenidos.
- Preparar los contenidos para ser almacenados en sitios destinados a tales efectos.

7.1. EJEMPLO PRÁCTICO DEL PROCESO DE EMPAQUETAMIENTO

PASOS:

- Al crear un paquete se necesita inicialmente una lista de actividades que se desean incluir, por esto el primer paso es hacer los objetos de aprendizaje con las diferentes herramientas: páginas Web con el editor correspondiente,

⁷⁸ <http://www.reload.ac.uk>

animaciones de Flash, aplicativos en Java Script, archivos de audio, documentos de texto, etc. Estos deben ser guardados en un directorio determinado que es donde irá el programa a buscarlas y donde guardará los ficheros que cree.

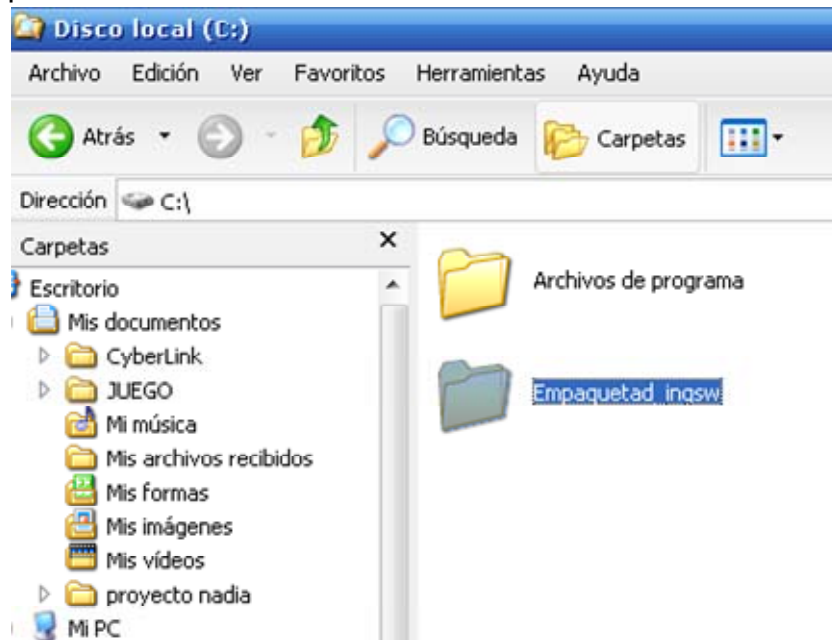


Figura 147. Empaquetamiento en SCORM con RELOAD2004, paso 1

- Ingresar al programa Reload2004 y crear un paquete nuevo. Para esto se debe seguir el procedimiento de: Archivo-Nuevo-ADL SCORM2004 Package.

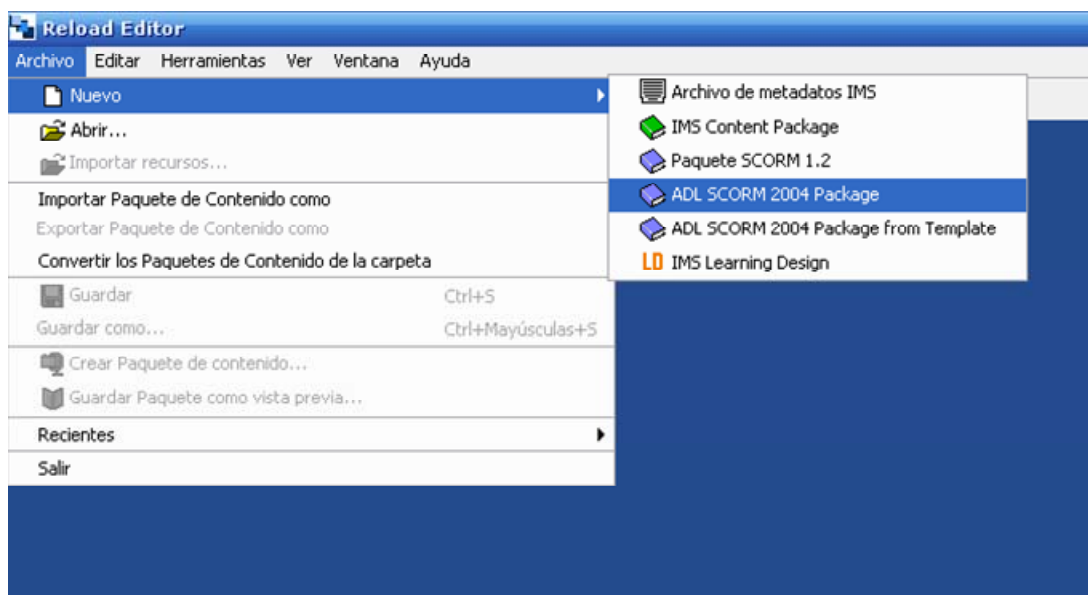


Figura 148. Empaquetamiento en SCORM con RELOAD2004, paso 2

Reload2004 solicita en qué directorio se debe crear el paquete, para esto se debe seleccionar la dirección de destino en la que se ubicó el objeto de aprendizaje.

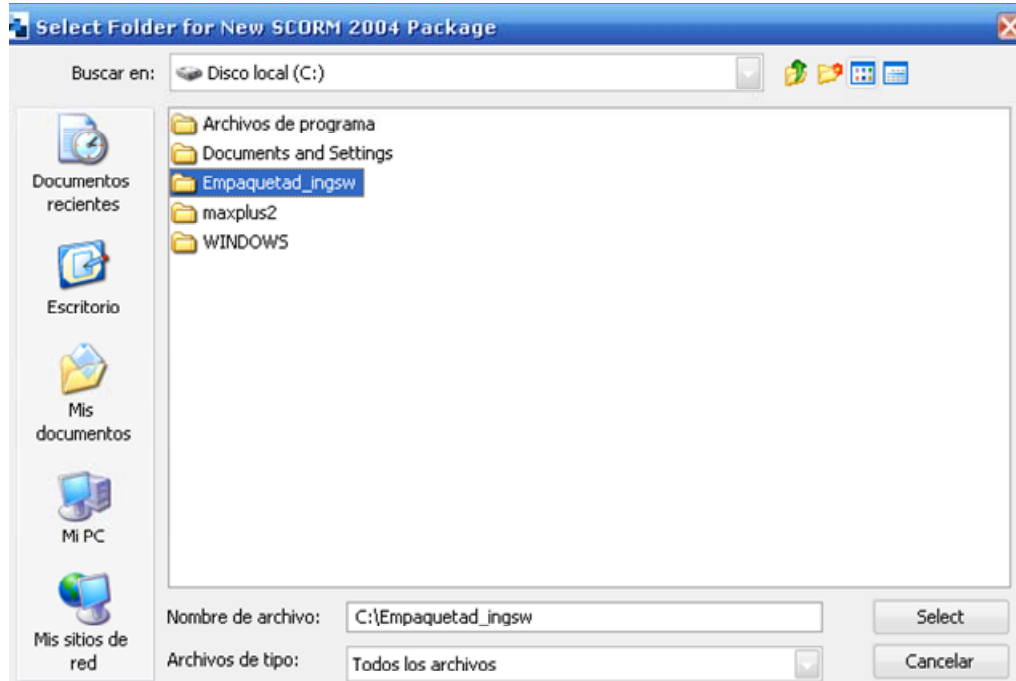


Figura 149. Empaquetamiento en SCORM con RELOAD2004, paso 2

- Reload crea los archivos necesarios para el proceso de empaquetamiento. Se puede observar que, además de los ficheros que ya existían, ahora el directorio contiene otros nuevos entre ellos se destaca el manifiesto del paquete imsmanifest.xml.

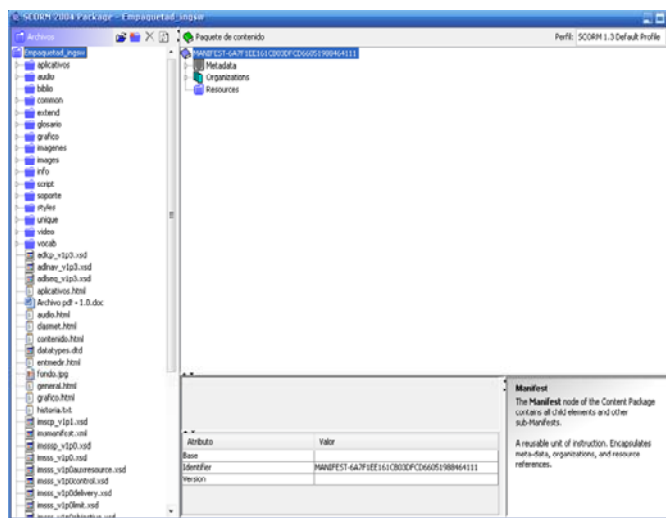


Figura 150. Empaquetamiento en SCORM con RELOAD2004, paso 3

- El paquete aun se encuentra vacío, para agregarle su contenido se debe empezar a incluir la información de los metadatos para esto se selecciona MANIFIEST y luego clic con el botón derecho sobre éste, en seguida se despliega un menú y se selecciona editar metadatos, y éste se exporta a la carpeta definida para guardar los metadatos del objetos de aprendizaje llamada metadatos, en la figura se ubica dicha carpeta, y el nombre del metadato es **t01_metadato** para este caso.

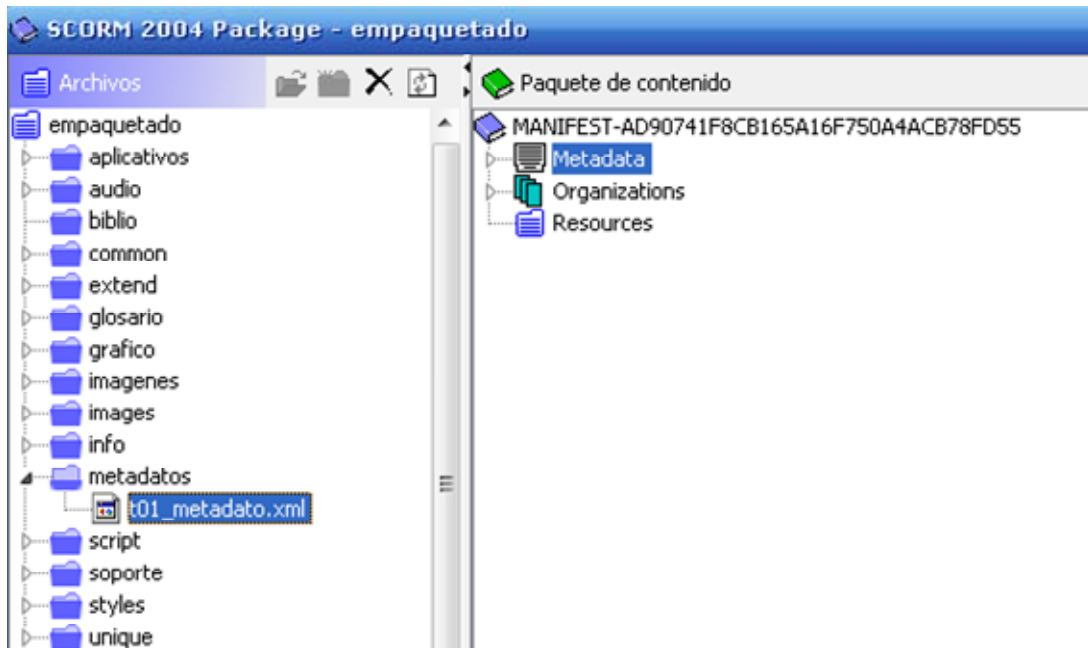


Figura 151. Empaquetamiento en SCORM con RELOAD2004, paso 4

Las metadatos son los datos informativos y estandarizados que tiene que contener todo paquete SCORM para poder ser utilizado por los diferentes LMS (eLearning Management System o entornos virtuales de enseñanza-aprendizaje).

El formato definido para e-escen@riUIS utiliza las 9 categorías de metadatos XML propuesta por el LOM (Learning Object Metadata) estándar de IEEE:⁷⁹

1. La categoría General (<general>) agrupa la información general que describe el objeto de aprendizaje de manera global.

⁷⁹ Fuente: GUÍA DE CREACIÓN METADATOS PARA LOS OBJETOS DE APRENDIZAJE PARA e-escen@riUIS

2. La categoría Ciclo de Vida (<lifeCycle>) agrupa las características relacionadas con la historia y el estado actual del objeto de aprendizaje, y aquellas que le han afectado durante su evolución.
 3. La categoría Meta-Metadatos (<metaMetadata>) agrupa la información sobre la propia instancia de metadatos, (en lugar del objeto de aprendizaje descrito por la instancia de metadatos).
 4. La categoría Técnica (<technical>) agrupa los requerimientos y características técnicas del objeto de aprendizaje.
 5. La categoría Uso Educativo (<educational>) agrupa las características educativas y pedagógicas del objeto de aprendizaje.
 6. La categoría Derechos (<rights>) agrupa los derechos de propiedad intelectual y las condiciones para el uso del objeto de aprendizaje.
 7. La categoría Relación (<relation>) agrupa las características que definen la relación entre este objeto de aprendizaje y otros objetos relacionados.
 8. La categoría Anotación (<annotation>) permite incluir comentarios sobre el uso educativo del objeto e información sobre cuándo y por quién fueron creados dichos comentarios.
 9. La categoría Clasificación (<classification>) describe este objeto de aprendizaje en relación a un determinado sistema de clasificación.
- Para ir dando forma a la estructura de aprendizaje que se le quiere dar al paquete se debe crear, ante todo, una organización. Un paquete puede tener muchas estructuras, denominadas organizaciones, esto depende del número de objetos de aprendizaje que se vayan a incluir dentro del paquete. Y luego se van agregando los ítems necesarios por cada núcleo de conocimiento.

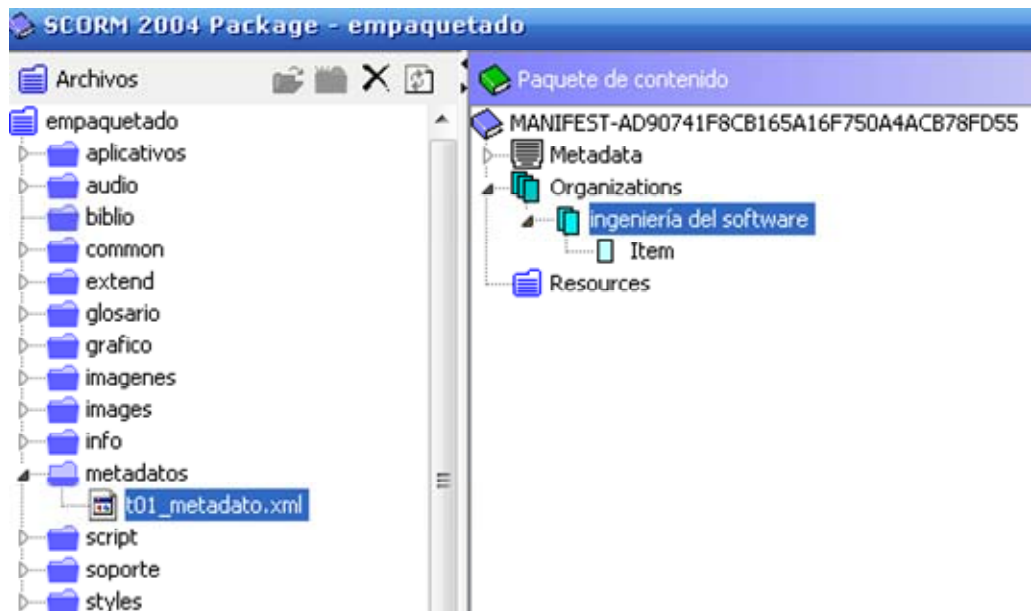


Figura 152. Empaquetamiento en SCORM con RELOAD2004, paso 5

- La organización del paquete vendrá dada por la secuencia de los contenidos que se le vayan añadiendo. Reload permite hacerlo simplemente arrastrando y soltando cada elemento desde la lista de la izquierda hasta el nombre de la organización:

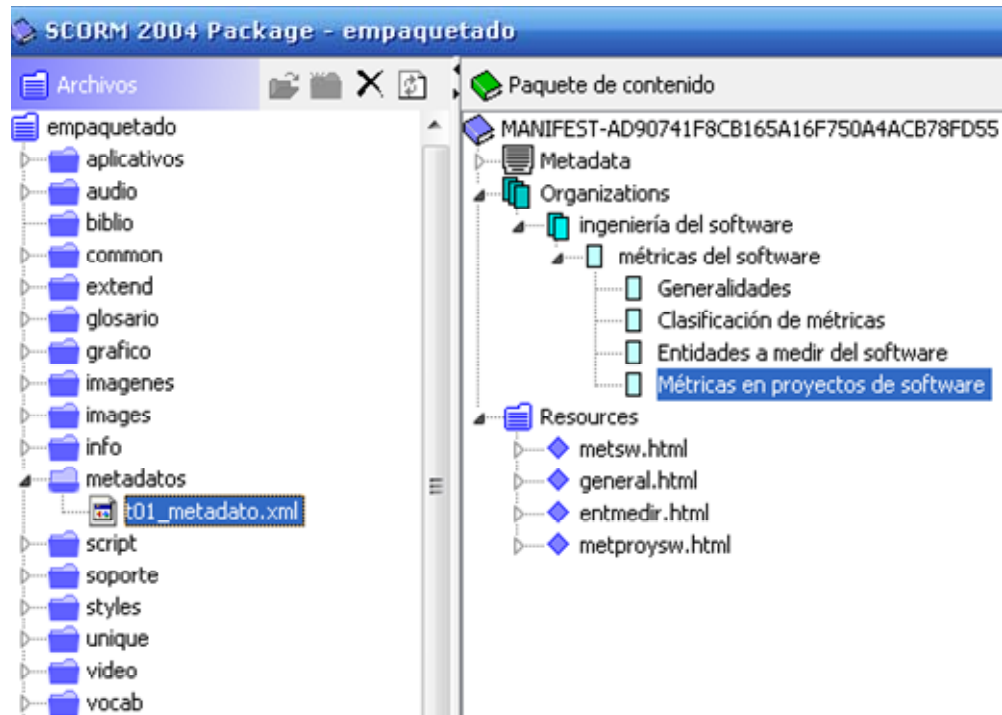


Figura 153. Empaquetamiento en SCORM con RELOAD2004, paso 6

Se puede observar que a medida que se sueltan elementos sobre la organización también se añaden automáticamente al nodo Resources. Si se despliega el nodo se podrá notar que no solamente se ha añadido el documento HTML sino también la hoja de estilo CSS y los archivos relacionados. Siempre se podrá reordenar los elementos de la organización seleccionándolos y haciéndolos subir o bajar con las flechas de la barra de herramientas.

- Reload permite ver cómo quedará la secuencia de objetos que se ha determinado a través de la organización. Para verlo se hace clic sobre el botón "Previsualización" del paquete de la barra de herramientas.

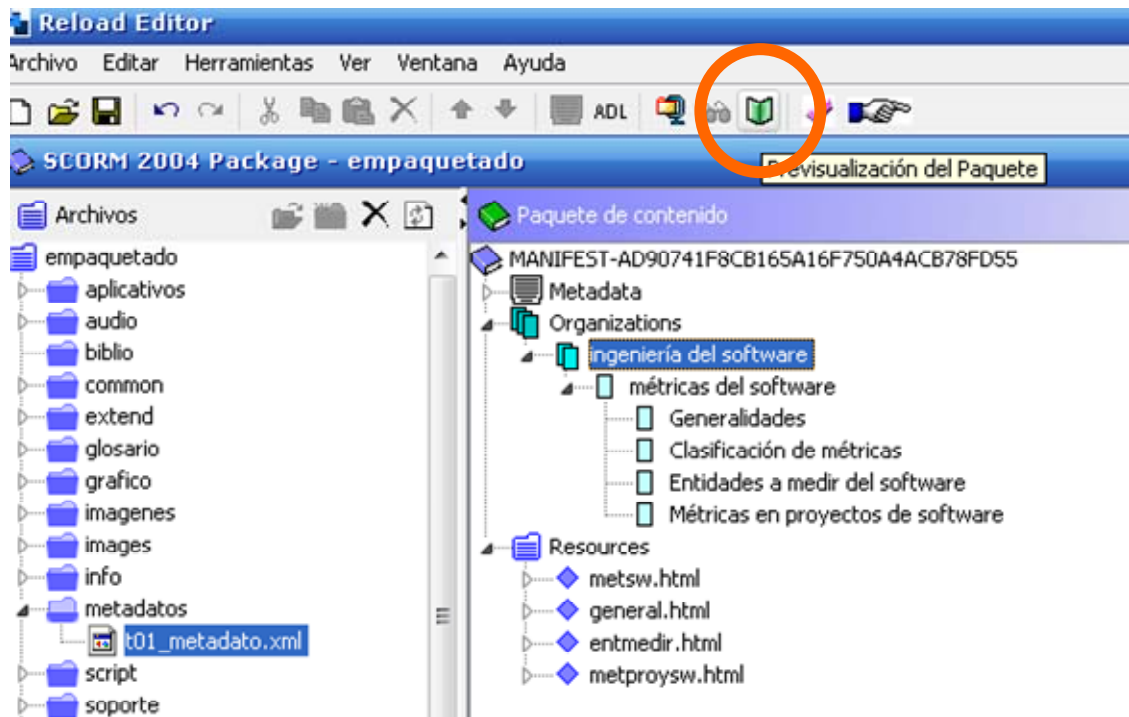


Figura 154. Empaquetamiento en SCORM con RELOAD2004, paso 7

Esto abre una ventana del navegador donde, en diferentes marcos, se puede observar la organización que se ha dado (marco izquierdo), el objeto de aprendizaje seleccionado (marco derecho) y en el marco superior una sencilla barra que permite el desplazamiento de uno a otro.

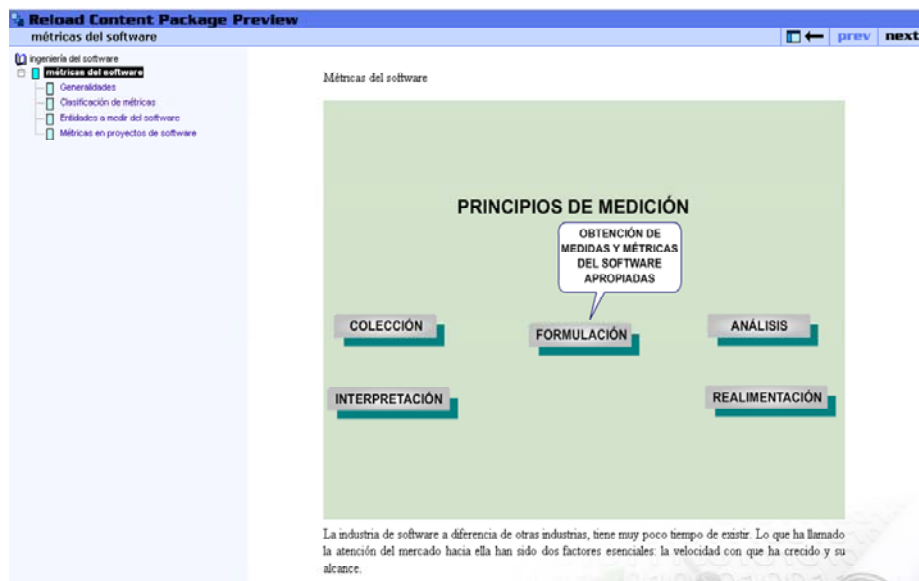


Figura 155. Empaquetamiento en SCORM con RELOAD2004

- Finalmente se hace el paquete en formato zip. Para esto, en el menú “File” se selecciona la opción “Zip Content Package...”.

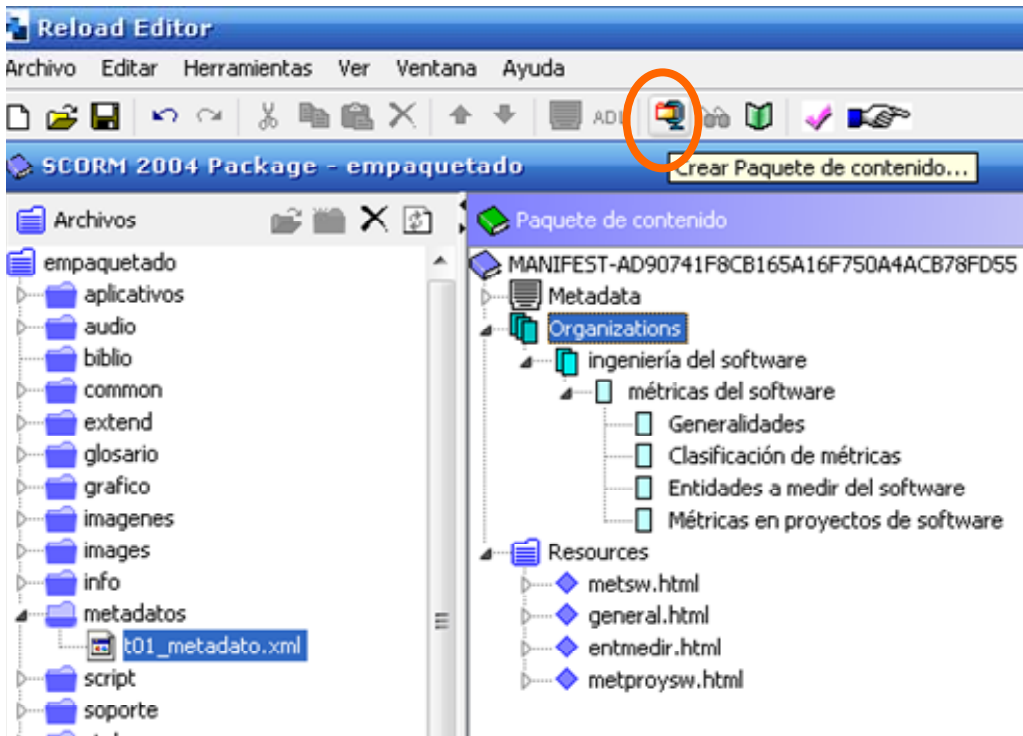


Figura 156. Empaquetamiento en SCORM con RELOAD2004, paso 8

Se elige la ubicación donde se desea guardar, se digita el nombre que se le quiere dar al paquete y se da clic en “Guardar”, con esto se generará un fichero zip que contiene todo el paquete.

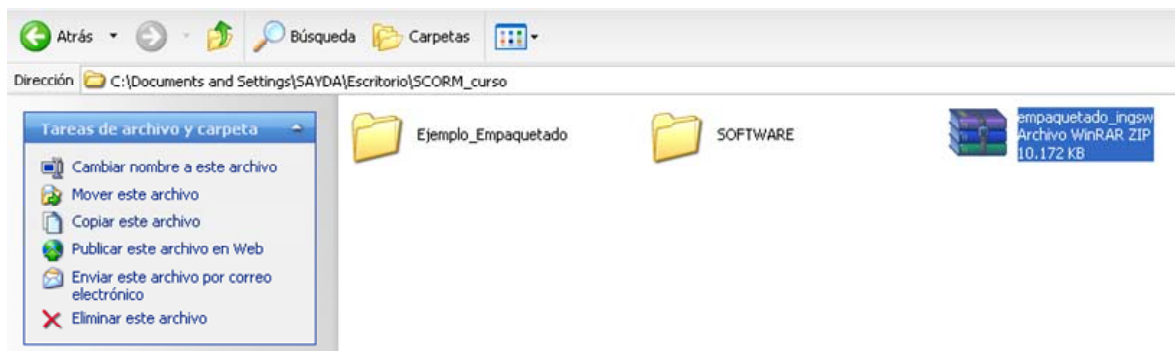


Figura 157. Visualización del fichero en .zip

CONCLUSIONES

- Es indispensable, para lograr los objetivos planteados en este tipo de proyectos, contar con un equipo de personas cuyo trabajo interdisciplinario fortalezca el desarrollo de cada etapa desde su rol de desempeño. La participación de los expertos temáticos, del metodólogo, de todo el soporte técnico de los ingenieros del CENTIC y de las desarrolladoras, enriquece el contenido y asegura el alcance de los propósitos planteados para este trabajo.
- El enfoque por competencias ayuda a centrar los procesos educativos en objetivos vinculados a un contexto social y supone un compromiso serio y profundo con la innovación. Su aporte es además fundamental a la hora de repensar y replantear un sistema educativo que registre las necesidades, los gustos, las preferencias y las opciones de los educandos.
- El diseño instruccional evidencia el proceso que establece relaciones entre los contenidos de la asignatura, las estrategias instruccionales y los resultados de aprendizaje deseados.
- Este trabajo de grado, en conjunto con el ProSPETIC, contribuye a proporcionar al estudiante y al docente estrategias que integren las actividades desarrolladas en el aula de clase con herramientas para el aprendizaje en línea.
- Con las TIC`s aparecen nuevos ambientes de aprendizaje que no pretenden sustituir a las aulas tradicionales, pero que vienen a complementarlas. El objeto de aprendizaje diseñado y desarrollado para la temática de Métricas del Software permite la profundización de los contenidos presentados en los diferentes formatos digitales que lo componen.

RECOMENDACIONES

- Es necesario dar continuidad a la construcción de objetos de aprendizaje para asegurar el soporte al proceso de enseñanza aprendizaje a través de medios didácticos que apoyen la formación en línea para el todo el contenido abordado en la asignatura Ingeniería del Software.
- Fundamentar la construcción de dichos objetos de aprendizaje en los criterios establecidos en el diseño instruccional realizado en este proyecto.
- Revisar y adecuar el diseño instruccional cuando fuere necesario, aprovechando su adaptabilidad, debido a que los programas académicos y, específicamente, la asignatura Ingeniería del Software, esta sujeta a cambios en su contenido.
- Es fundamental que los desarrolladores de los siguientes proyectos trabajen de la mano con el equipo conformado por los expertos temáticos y los ingenieros del CENTIC para asegurar la continuidad de la visión y el enfoque planteados por el ProsPETIC y la calidad de los productos resultantes.
- Mantener actualizada la información contenida en el portal del profesor y promover el acceso al mismo por parte de los estudiantes con el fin de disponer de una forma adicional de comunicación entre éste y el docente.

BIBLIOGRAFÍA

[ÁLVAREZ, 1987]

ÁLVAREZ, J.M. Dos perspectivas contrapuestas sobre el currículum y su desarrollo. Revista de Educación No. 282, enero-abril, 1987. En: [SANZ. 2004]

[ANAYA, 1999]

ANAYA, Marcela Isabel, Centro de Tecnología Educativa, Secretaría de Educación y Cultura del Estado de Sonora,
Disponible en: www.cete-sonora.gob.mx/

[ARBESÚ, 1996]

ARBESÚ, M. El sistema modular Xochimilco. En “El Sistema Modular en la Unidad Xochimilco de la Universidad Autónoma Metropolitana” México, 1996. En [SANZ. 2004]

[ARMAS, 1997]

ARMAS, Sara. Técnicas Grupales de Enseñanza-Aprendizaje. Universidad de San Carlos de Guatemala, 1997.
Disponible en: <http://medicina.usac.edu.gt/pfd/semanas/semana8/tecnicas.pdf>

[BAHAMON, 2003]

BAHAMÓN, José Hernando. El aprendizaje individual permanente: ¿cómo lograr el desarrollo de esta capacidad de los estudiantes?, Centro de Recursos Para el Aprendizaje, 2003.
Disponible en: http://www.eduteka.org/pdfdir/cartilla_aprendizaje.pdf

[BAUER, 1972]

BAUER, F. L.: <<Software Engineering>>, Information Processing, 71, North Holland Publishing Co., Amsterdam.

[BECCO, 2002]

BECCO Guillermo R. Vigotsky y Teorías sobre el Aprendizaje
Disponible en: <http://www.ciudad.com.ar/>

[CARRILLO, et al. 2005]

CARRILLO CAICEDO, Gabriel; et al. Propuesta Metodológica para el Desarrollo e Implementación de Diseños Curriculares Bajo la Visión de Competencias para Asignaturas de Programas de Formación Profesional. Universidad Industrial de Santander, 2005.

[CATALANO, et al. 2004]

CATALANO, Ana; AVOLIO, Susana; SLADOGNA, Mónica. Diseño Curricular Basado en Normas de Competencia Laboral: Conceptos y Orientaciones Metodológicas. 1º ed. - Buenos Aires: Banco Interamericano de Desarrollo, 2004.

Disponible

en:

http://www.cinterfor.org.uy/public/spanish/region/ampro/cinterfor/publ/dis_curr/pdf/dis_curr.pdf

[DEPRESBITERIS, 2002]

DEPRESBITERIS, Léa. Avaliação da aprendizagem - uma nova prática implica nova visão do ensino, 2002. Resúmenes de la Biblioteca de Documentos, Chilecalifica.

Disponible

en:

<http://www.chilecalifica.cl/califica/showCuerpo.do?cuerpo=309&barra=1&pagina=bajada-cuerpo-documento.jsp>

[DÍAZ, 1993]

DÍAZ BARRIGA, Frida. Aproximaciones Metodológicas al Diseño Curricular: Hacia una Propuesta Integral. En Tecnología y Comunicación Educativa. No 21, 1993. En [SANZ. 2004]

[DÍAZ, 1997]

DÍAZ BARRIGA, Ángel. Didáctica y Currículum, Buenos Aires, Paidós, 1997. En [SANZ. 2004]

[DÍAZ, HERNÁNDEZ. 1998]

DÍAZ BARRIGA, Frida; HERNÁNDEZ ROJAS, Gerardo. Estrategia Docentes para un Aprendizaje Significativo: Una Interpretación constructivista. México: McGRAW-HILL Interamericana, 1998.

Disponible

en:

http://www.uv.mx/cpp/doc/DC_Diaz_Barriga_Estrategias_de_ensenanza....doc

[ESTRADA, 2005]

ESTRADA, Lilia. Elaboración y Documentación de una Propuesta de Diseño Curricular bajo la Visión de Competencias para la Asignatura Mediciones Eléctricas y Estudio de su Implementación en una Plataforma e-learning, Tesis de grado. Universidad Industrial de Santander, 2005.

[FONDEF 01, 2005]

FONDEF: Fondo de Fomento al Desarrollo Científico y Tecnológico. Grupo de Ingenieros. Manual de buenas prácticas para el desarrollo de objetos de aprendizaje. Proyecto FONDEF Chile, agosto de 2005.

Disponible en: http://www.aproa.cl/1116/articles-68370_recurso_1.pdf

[FONDEF 02, 2005]

FONDEF: Fondo de Fomento al Desarrollo Científico y Tecnológico. Grupo de Ingenieros. Qué es un objeto de aprendizaje. Proyecto FONDEF Chile, agosto de 2005.

Disponible en: <http://www.aproa.cl/1116/propertyvalue-5538.html>

[GARZA, 2000]

GARZA, Eugenio. Las Técnicas Didácticas en el Modelo Educativo del Tec de Monterrey. Col. Tecnológico, Monterrey, 2000.

Disponible en: <http://tecnologiaedu.us.es/bibliovir/pdf/309.pdf>

[GÓMEZ, HERNÁNDEZ, 2006]

Edwin Gómez Jiménez, Samuel Hernández García. Diseño instruccional mediado por tecnologías de información y comunicación (TIC's), como estrategia de formación basada en competencias; para la asignatura estadística i del programa académico de ingeniería de sistemas e informática.

Disponible en: <http://tangara.uis.edu.co/biblioweb/>

[GONZÁLEZ, et al. 1992]

GONZÁLEZ, M y otros. El modelo de investigación en la acción. En “El Planeamiento Curricular en la Enseñanza Superior”, CEPES-UH. 1992. En [SANZ. 2004]

[GONZÁLEZ, 2005]

GONZÁLEZ, Ángela. Evaluación por competencias. Universidad Santo Tomás de Aquino. Colombia, 2005.

Disponible en: <http://basicamente.usta.edu.co/mainartic6.html>

[HERNÁNDEZ, et al. 1998]

HERNÁNDEZ, Carlos Augusto, et al. Exámenes de Estado: una propuesta de evaluación por competencias. Bogotá: Javegraf, 1998. En: [POSADA. 2004]

[HERNÁNDEZ, 2004]

HÉRNANDEZ POU, Priscilla. Psicología Educativa y Método de Enseñanza. Santo Domingo, República Dominicana, 2004.

Disponible en: <http://www.monografias.com/trabajos5/psicoedu/psicoedu.shtml>

[HURST, 2005]

HURST, Geoconda. Módulo Taller de Metodología de Diseño Curricular. Programa de Capacitación Modular (PCM) para Docentes de Formación Profesional de la Enseñanza Media de Honduras, 2005.

Disponible

en:

<http://www.upnfm.edu.hn/programas/prahmo/Modulos%20%20Pedagogicos/Mod.%20Dis e%C3%B1o%20Curricular.pdf>

[IEEE]

IEEE, Standards Collection: Software Engineering, IEEE Standard 610.12-1990.

Disponible en: <http://www.ieee.org>

[LANDSBERGER, 2005]

LANDSBERGER, Joe. Aprendizaje Basado en Problemas, 2005.

Disponible en: <http://www.studygs.net/espanol/pbl.htm>

[LÓPEZ, 2007]

LÓPEZ, Clara. Los Repositorios de Objetos de Aprendizaje como soporte para los entornos de e-Learning. Editorial España, Salamanca, 2007.

Disponible

en:

http://www.biblioweb.dgsca.unam.mx/libros/repositorios/objetos_aprendizaje.htm

[MERGEL, 1998]

MERGEL, Brenda. Diseño Instruccional y Teoría del Aprendizaje. Universidad de Saskatchewan Canadá, 1998.

Disponible en: http://www.educadis.uson.mx/Educa_subpaginas/Biblio-virt-art%EDculos-DI.htm

[MOCKER, SPEAR. 1982]

MOCKER, D. W., SPEAR, G. E. Lifelong Learning: Formal, Nonformal, Informal, and Self-Directed. The National Center for Research in Vocational Education, The Ohio State University, 1982. En: [LOWRY. 2003]

[ONTORIA, et al. 1997]

ONTORIA, A. et al. Mapas conceptuales - una técnica para aprender. 7ed, Madrid, Nancea, 1997. Referenciado en los resúmenes de la Biblioteca de Documentos, Chilecalifica.

Disponible

en:

<http://www.chilecalifica.cl/califica/showCuerpo.do?cuerpo=309&barra=1&pagina=bajada-cuerpo-documento.jsp>

[PEON, 1999]

Rodolfo Peón Aguirre, Programa de Educación Continua, Abierta y a Distancia de la Universidad de Sonora

Disponible en: www.educadis.uson.mx

[PÉREZ, 2000]

PÉREZ ABRIL; Mauricio. Competencia textual, competencia pragmática y competencia argumentativa. Ejes de la evaluación de producción de textos. Universidad Pedagógica Nacional, 2000.

Disponible en: <http://www.ut.edu.co/rddlct/docs/mpa-dc.doc>

[PERKINS, 1991]

PERKINS, D. Technology meets constructivism: Do they make a marriage? Educational Technology, 1991. En [SCHWIER. 1998] en [MERGEL. 1998]

[PIAGET, 1970]

PIAGET, Jean. La epistemología de las relaciones interdisciplinarias. En: [POSADA. 2004]

[POSADA, 2004]

POSADA, Rodolfo. Formación superior basada en competencias, interdisciplinaria y trabajo autónomo del estudiante. Departamento de Pedagogía, Facultad de Educación, Universidad del Atlántico Colombia. Revista Iberoamericana de Educación (ISSN: 1681-5653). 2004.

Disponible en: <http://www.rieoei.org/deloslectores/648Posada.PDF>

[PRESSMAN, 2002]

PRESSMAN, Roger S. "Ingeniería del Software, un enfoque práctico". 5ª ED. Madrid, Editorial McGraw Hill, 2002. 59 p. ISBN 0-07-709677-0.

[RAMÍREZ P., Dorys Consuelo - VERJEL A., Dania Rubiela, 2005]

Diseño y elaboración de la estructura curricular para la asignatura tratamiento de señales bajo una visión de competencias y estudio de adaptación a una plataforma e-learning. Trabajo de grado (Ingeniera Electrónica) Universidad Industrial de Santander

Disponible en: <http://tangara.uis.edu.co/biblioweb/>

[REBOLLO, 2004]

REBOLLO, Miguel .El estándar SCORM para EaD, Tesina del Máster en Enseñanza y Aprendizaje abiertos y a Distancia Universidad Nacional de Educación a Distancia

[RUIZ, 1996]

RUIZ, José María. Teoría del currículo: Diseño y desarrollo curricular. Editorial Universitas. S.A. España, 1996. En [SANZ. 2004]

[SAETTLER, 1990]

SAETTLER, Paúl. Historia de la Tecnología Educativa Americana, 1990. En: [MERGEL. 1998]

[SÁEZ, ELLIOT. 1988]

SÁEZ, M., ELLIOT, J. La investigación en Acción en España: Un Proceso que Empieza. Revista de Educación. Madrid, 1988. En [SANZ. 2004]

[SALAS, 2001]

SALAS Z. Walter A. Formación Por Competencias en Educación Superior. Una aproximación conceptual a propósito del caso colombiano –Revista Iberoamericana de Educación (ISSN: 1681-5653)

[SÁNCHEZ, 1999]

SÁNCHEZ, Luis Iván. Lineamientos Generales para el Diseño/Rediseño Curricular, UAT, México, 1999.

Disponible en:

<http://colaboracion.uat.edu.mx/rectoria/subacademica/materiales%20de%20trabajo/Flexibilidad%20Documento%202%20Final.doc>

[SANZ, 2004]

SANZ, Teresa. Modelos Curriculares. Centro de Estudios para el Perfeccionamiento de la Educación Superior. Universidad de la Habana, 2004.

Disponible en:

<http://www.upsp.edu.pe/descargas/Docentes/Antonio/revista/04/2/189404205.pdf>

[SERNA, 2003]

SERNA, José María. El Método de Casos: Reflexiones sobre el cambio en la Metodología de la Enseñanza del Derecho en México, 2003.

Disponible en: <http://www.bibliojuridica.org/libros/4/1793/9.pdf>

[TORRES, 1996]

TORRES SANTOME, Jurjo. Globalización e interdisciplinariedad: El curriculum integrado. Madrid: Morata, 1996. En [POSADA. 2004]

[TORRES, 2001]

TORRES, Gladis. Diseño Curricular: Metodología para el Perfeccionamiento del Currículo en su Esfera de Acción, UACAM, México, 2001.

Disponible en:

[http://www.uacam.mx/macad.nsf/4a24042bd57e05c980256509003e0809/73c5cc4fbd0792c586256e7f0004a495/\\$FILE/Diseno-Curricular.pdf](http://www.uacam.mx/macad.nsf/4a24042bd57e05c980256509003e0809/73c5cc4fbd0792c586256e7f0004a495/$FILE/Diseno-Curricular.pdf)

[VALENZUELA, 2000]

VALENZUELA G, Jaime Ricardo, Los Tres “Autos” del aprendizaje: Aprendizaje estratégico en educación a distancia, 2000. En [ESTRADA. 2005]

[ZABALZA, 2003]

ZABALZA, Miguel A. Competencias Docentes del Profesorado Universitario. Calidad y Desarrollo Profesional. Madrid: Nancea S.A. Editores, 2003. En: [ESTRADA. 2005]

[ZELKOVIT, 1979]

ZELKOVIT, M. V., Shaw, A. C. y GANNON, J. D.: Principles of Software Engineering and Design. Prentice Hall, 1979.

PÁGINAS WEB:

APRENDER A APRENDER. ESTRATEGIAS DE APRENDIZAJE.

Disponible en: <http://extensiones.edu.aytolacoruna.es/educa/aprender/estrategias.htm>

APRENDIZAJE BASADO EN PROYECTOS. José Miguel Molina.

Disponible en: http://www.mercadeo.com/31_aprendizajebp.htm

APRENDIENDO CON REPOSITORIOS DE OBJETOS DE APRENDIZAJE”, APROA. MANUAL DE BUENAS PRACTICAS PARA EL DESARROLLO DE OBJETOS DE APRENDIZAJE. VERSIÓN 1.

Disponible en: www.aproa.cl

APROA (APRENDIENDO CON OBJETOS DE APRENDIZAJE): METODOLOGÍA DE DESARROLLO PARA DISEÑO Y PRODUCCIÓN DE OBJETOS DE APRENDIZAJE

Disponible en: www.aproa.cl/1116/propertyvalue-5538.html

CENTRO DE TECNOLOGÍAS DE INFORMACIÓN DE INTEC. ESTÁNDARES E-LEARNING: ESTADO DEL ARTE.

Disponible en: <http://empresas.sence.cl>.

DEFINICIONES DE E-LEARNING

Disponible en: <http://empresas.sence.cl>

DRAE, DICCIONARIO DE LA REAL ACADEMIA ESPAÑOLA DE LA LENGUA.

Disponible en: <http://buscon.rae.es/draeI/>

FELDER AND SOLOMAN. LEARNING STYLES AND STRATEGIES

Disponible en: <http://www4.ncsu.edu/unity/lockers/users/f/felder/public/ILSdir/styles.htm>

GO2LEARN: UN PASO MÁS ADELANTE QUE EL RESTO.

Disponible en: <http://go2learn.cl/id.html>

MODELOS CENTRADOS EN LOS OBJETIVOS

Disponible en:

www.cepes.uh.cu/bibliomaestria/teoriaydisenocurricular/MODELOSDEOBJETIVOS.doc

PORTAL PROFESORA CLARA INES PEÑA DE CARRILLO

Disponible en: <http://gavilan.uis.edu.co/~clarenes>

PRINCIPIOS DIDÁCTICOS, APRENDIZAJE COOPERATIVO Y PROCESO DE ENSEÑANZA. Disponible en:

<http://www.monografias.com/trabajos15/principios-didacticos/principios-didacticos.shtml>

REUSABLE E-LEARNING OBJECT

Disponible en: <http://www.reload.ac.uk>

SCORM (SHARABLE CONTENT OBJECT REFERENCE MODEL): ESTÁNDARES TÉCNICOS INTERRELACIONADOS PARA DESARROLLAR ENSEÑANZA DE CONTENIDOS VÍA WEB.

Disponible en <http://www.tise.cl/archivos/tise2006/17.pdf>

SISTEMA DE TUTORÍA INTELIGENTE ADAPTATIVO CONSIDERANDO ESTILOS DE APRENDIZAJE.

Disponible en: <http://eia.udg.es/~clarenes/docs/ribie-udg-2002.pdf>

TELE LEARNING MODELS.

Disponible en: www.licefteluguqam.ca/gp/ppt/aera.pps

TYLER E HILDA TABA: MODELO RACIONAL NORMATIVO

Disponible en:

http://www.wikilearning.com/tyler_e_hilda_taba_modelo_racional_normativo_i-wkccp-10358-11.htm

UML-DIAGRAMAS DE SECUENCIA

Disponible en: <http://www.creangel.com/uml/secuencia.php>

<http://www.creangel.com/uml/intro.php>

<http://www.ultrasist.com.mx/tecnologias/uml.htm>

ANEXO A.

TEST DE ESTILOS DE APRENDIZAJE DE FELDER Y SILVERMAN

ÍNDICE DE ESTILOS DE APRENDIZAJE DE FELDER Y SOLOMAN CUESTIONARIO

Instrucciones: Encierre en un círculo la opción "a" o "b" para indicar su respuesta a cada pregunta. Por favor seleccione solamente una respuesta para cada pregunta. Si tanto "a" y "b" parecen aplicarse a usted, seleccione aquella que se aplique más frecuentemente.

1. Entiendo mejor algo:

- a) si lo practico.
- b) si pienso en ello.

2. Me considero:

- a) realista.
- b) innovador.

3. Cuando pienso acerca de lo que hice ayer, es más probable que lo haga en base a:

- a) una imagen.
- b) palabras.

4. Tengo tendencia a:

- a) entender los detalles de un tema pero no ver claramente su estructura completa.
- b) entender la estructura completa pero no ver claramente los detalles.

5. Cuando estoy aprendiendo algo nuevo, me ayuda:

- (a) hablar de ello.
- (b) pensar en ello.

6. Si yo fuera profesor, preferiría dar un curso:

- (a) que trate sobre hechos y situaciones reales de la vida.
- (b) que trate con ideas y teorías.

7. Prefiero obtener información nueva de:

- (a) imágenes, diagramas, gráficas o mapas.

(b) instrucciones escritas o información verbal.

8. Una vez que entiendo:

(a) todas las partes, entiendo el total.

(b) el total de algo, entiendo como encajan sus partes.

9. En un grupo de estudio que trabaja con un material difícil, es más probable que:

(a) participe y contribuya con ideas.

(b) no participe y solo escuche.

10. Es más fácil para mí:

(a) aprender hechos.

(b) aprender conceptos.

11. En un libro con muchas imágenes y gráficas es más probable que:

(a) revise cuidadosamente las imágenes y las gráficas.

(b) me concentre en el texto escrito.

12. Cuando resuelvo problemas de matemáticas:

(a) generalmente trabajo sobre las soluciones con un paso a la vez.

(b) frecuentemente sé cuales son las soluciones, pero luego tengo dificultad para imaginarme los pasos para llegar a ellas.

13. En las clases a las que he asistido:

(a) he llegado a saber como son muchos de los estudiantes.

(b) raramente he llegado a saber como son muchos estudiantes.

14. Cuando leo temas que no son de ficción, prefiero:

(a) algo que me enseñe nuevos hechos o me diga como hacer algo.

(b) algo que me de nuevas ideas en que pensar.

15. Me gustan los maestros:

(a) que utilizan muchos esquemas en la pizarra.

(b) que toman mucho tiempo para explicar.

16. Cuando estoy analizando un cuento o una novela:

- (a) pienso en los incidentes y trato de acomodarlos para configurar los temas.
- (b) me doy cuenta de cuales son los temas cuando termino de leer y luego tengo que regresar y encontrar los incidentes que los demuestran.

17. Cuando comienzo a resolver un problema de tarea, es más probable que:

- (a) comience a trabajar en su solución inmediatamente.
- (b) primero trate de entender completamente el problema.

18. Prefiero la idea de:

- (a) certeza.
- (b) teoría.

19. Recuerdo mejor:

- (a) lo que veo.
- (b) lo que oigo.

20. Es más importante para mí que un profesor:

- (a) exponga el material en pasos secuenciales claros.
- (b) me dé un panorama general y relacione el material con otros temas.

21. Prefiero estudiar:

- (a) en un grupo de estudio.
- (b) solo.

22. Me considero:

- (a) cuidadoso en los detalles de mi trabajo.
- (b) creativo en la forma en la que hago mi trabajo.

23. Cuando alguien me da direcciones de nuevos lugares, prefiero:

- (a) un mapa.
- (b) instrucciones escritas.

24. Aprendo:

- (a) a un paso constante. Si estudio con ahínco consigo lo que deseo.
- (b) en inicios y pausas. Me llego a confundir y súbitamente lo entiendo.

25. Prefiero primero:
- (a) hacer algo y ver que sucede.
 - (b) pensar como voy a hacer algo.
26. Cuando leo por diversión, me gustan los escritores que:
- (a) dicen claramente los que desean dar a entender.
 - (b) dicen las cosas en forma creativa e interesante.
27. Cuando veo un esquema o bosquejo en clase, es más probable que recuerde:
- (a) la imagen.
 - (b) lo que el profesor dijo acerca de ella.
28. Cuando me enfrento a un cuerpo de información:
- (a) me concentro en los detalles y pierdo de vista el total de la misma.
 - (b) trato de entender el todo antes de ir a los detalles.
29. Recuerdo más fácilmente:
- (a) algo que he hecho.
 - (b) algo en lo que he pensado mucho.
30. Cuando tengo que hacer un trabajo, prefiero:
- (a) dominar una forma de hacerlo.
 - (b) intentar nuevas formas de hacerlo.
31. Cuando alguien me enseña datos, prefiero:
- (a) gráficas.
 - (b) resúmenes con texto.
32. Cuando escribo un trabajo, es más probable que :
- (a) lo haga (piense o escriba) desde el principio y avance.
 - (b) lo haga (piense o escriba) en diferentes partes y luego las ordene.
33. Cuando tengo que trabajar en un proyecto de grupo, primero quiero:
- (a) realizar una "lluvia de ideas" donde cada uno contribuye con ideas.
 - (b) realizar la "lluvia de ideas" en forma personal y luego juntarme con el grupo para comparar las ideas.

34. Considero que es mejor elogio llamar a alguien:

- (a) sensible.
- (b) imaginativo.

35. Cuando conozco gente en una fiesta, es más probable que recuerde:

- (a) cómo es su apariencia.
- (b) lo que dicen de sí mismos.

36. Cuando estoy aprendiendo un tema, prefiero:

- (a) mantenerme concentrado en ese tema, aprendiendo lo más que pueda de él.
- (b) hacer conexiones entre ese tema y temas relacionados.

37. Me considero:

- (a) abierto.
- (b) reservado.

38. Prefiero cursos que dan más importancia a:

- (a) material concreto (hechos, datos).
- (b) material abstracto (conceptos, teorías).

39. Para divertirme, prefiero:

- (a) ver televisión.
- (b) leer un libro.

40. Algunos profesores inician sus clases haciendo un esquema de lo que enseñarán. Esos esquemas son:

- (a) algo útiles para mí.
- (b) muy útiles para mí.

41. La idea de hacer una tarea en grupo con una sola calificación para todos:

- (a) me parece bien.
- (b) no me parece bien.

42. Cuando hago grandes cálculos:

- (a) tiendo a repetir todos mis pasos y revisar cuidadosamente mi trabajo.

(b) me cansa hacer su revisión y tengo que esforzarme para hacerlo.

43. Tiendo a recordar lugares en los que he estado:

(a) fácilmente y con bastante exactitud.

(b) con dificultad y sin mucho detalle.

44. Cuando resuelvo problemas en grupo, es más probable que yo:

(a) piense en los pasos para la solución de los problemas.

(b) piense en las posibles consecuencias o aplicaciones de la solución en un amplio rango de campos.

ANEXO B.⁸⁰

TAXONOMIA DE BLOOM (VERBOS PARA ENUNCIAR SABERES)

⁸⁰ GUZMÁN., TATIANA. HERRERA., JAIRO. Diseño instruccional basado en competencias mediado por tecnologías de información y comunicación (tic's), para la asignatura inteligencia artificial del programa académico de ingeniería de sistemas e informática.

TAXONOMÍAS DE OBJETIVOS

En este apartado se presentarán dos taxonomías empleadas como referente para la construcción de la tabla de saberes, y como guía de selección de verbos para enunciar los diferentes saberes, contenidos, propósitos, actividades de enseñanza - aprendizaje y unidades de aprendizaje.

■ Taxonomía de Bloom

La taxonomía de Bloom, es una de las catalogaciones más conocidas en el diseño curricular y surgió de la propuesta de Benjamín Bloom que privilegia la técnica de objetivos cognitivo-conductuales. La taxonomía de Bloom propone seis niveles de competencia de los objetivos formativos, los cuales se presentan en la *Tabla 3*.

Clasificación taxonómica de Bloom

Nivel	Descripción
Conocimiento	Ser capaz de recordar palabras, hechos, fechas, convenciones, clasificaciones, principios, teorías, etc.
Comprensión	Ser capaz de trasponer, interpretar y extrapolar a partir de ciertos conocimientos.
Aplicación	Ser capaz de usar conocimientos o principios para resolver un problema.
Análisis	Ser capaz de identificar los elementos, las relaciones y los principios de organización de una situación.
Síntesis	Ser capaz de producir una obra personal después de haber trazado un plan de acción.
Evaluación	Ser capaz de emitir un juicio crítico basado en criterios internos o externos.

Cada nivel de la taxonomía de Bloom incluye los anteriores, es decir que para adquirir el siguiente nivel deben haberse adquirido los anteriores. La propuesta de objetivos de Bloom se centra principalmente en el nivel de conocimiento y se puede apreciar en su libro *Taxonomía de los Objetivos de la Educación*, la clasificación de las metas educacionales. *Manuales I y II*⁸¹.

⁸¹ BLOOM, Benjamín Samuel. *Taxonomía de los Objetivos de la Educación: Clasificación de las Metas Educativas. Manuales I y II*. Buenos Aires: El Ateneo, 1979.

■ Categorías del Contenido de César Coll

En el libro *Psicología y Currículum*, César Coll establece tres categorías generales de contenidos: la de hechos, conceptos y principios, la de procedimientos, y la de valores, normas y actitudes y a su vez presenta ejemplos de verbos que pueden usarse en cada categoría⁸². En la *Tabla 4* se define cada una de las categorías y los verbos recomendados.

Categorías y verbos recomendados de César Coll

Categoría 1	Categoría 2	Categoría 3
<p>Aprender hechos y conceptos significa que se es capaz de identificar, reconocer, describir y comparar objetos, sucesos o ideas.</p> <p>Aprender un principio significa que se es capaz de identificar, reconocer, clarificar, describir y comparar las relaciones entre los conceptos o hechos a que se refiere el principio.</p>	<p>Aprender un procedimiento significa que se es capaz de utilizarlo en diversas situaciones y de diferentes maneras, con el fin de resolver los problemas planteados y alcanzar las metas fijadas.</p>	<p>Aprender un valor significa que se es capaz de regular el propio comportamiento de acuerdo con el principio normativo que dicho valor estipula.</p> <p>Aprender una norma significa que se es capaz de comportarse de acuerdo con ella.</p> <p>Aprender una actitud significa mostrar una tendencia consistente y persistente a comportarse de una particular manera ante determinada clase de situaciones, objetos, sucesos o personas.</p>
VERBOS APLICABLES EN CADA CATEGORÍA		
<p>Identificar, Analizar, Señalar, Reconocer, Interferir, Resumir, Clasificar, Generalizar, Aplicar, Describir, Comentar, Distinguir, Comparar, Interpretar, Relacionar, Conocer, Recordar, Indicar, Explicar, Sacar soluciones, Enumerar, Situar (en espacio o tiempo), etc.</p>	<p>Manejar, Observar, Confeccionar, Probar, Utilizar, Elaborar, Construir, Simular, Aplicar, Demostrar, Recoger, Reconstruir, Presentar, Planificar, Experimentar, Ejecutar, Componer, etc.</p>	<p>Comportarse (de acuerdo con), reaccionar a, acceder a, conformarse con, respetar, actuar, preocuparse por, tolerar, conocer, deleitarse, apreciar, darse cuenta que, inclinarse por, prestar atención a, aceptar, obedecer, interesarse por, ser conciente de, permitir, valorar (positiva o negativamente), etc.</p>

⁸² Coll, César. *Psicología y Currículum: Una Aproximación Psicopedagógica a la Elaboración del Currículo Escolar*. 1 ed. Barcelona: Paidós, 1995.

VERBOS PARA ENUNCIAR SABERES

SABER		HACER		SER	
Verbo	Sinónimos	Verbo	Sinónimos	Verbo/actitud	Sinónimos/Explicación
Identificar	corresponder, establecer, reconocer, determinar, referir, describir, reseñar, compenetrarse, detallar, registrar	Manejar	usar, utilizar, manipular, operar, maniobrar, transformar	Comportar (se)	regirse, actuar, obrar, proceder, portarse
Analizar	estudiar, detallar, observar, separar, descomponer, averiguar, considerar, examinar, distinguir, comparar, razonar	Observar	examinar, estudiar, notar, analizar, percibir, mirar	Reaccionar (a)	oponerse, resistir, responder, evolucionar
Señalar	guiar, mostrar, , decir, distinguirse, establecer, registrar, aclarar, designar, evidenciar, indicar, recalcar, determinar, nombrar, mencionar, informar, reseñar, destacar	Confeccionar	hacer, probar, medir, elaborar, ejecutar, componer, manufacturar, fabricar	Acceder (a)	entrar, llegar, aceptar, alcanzar, someterse, , permitir
Reconocer	rememorar, recordar, investigar, examinar, observar, registrar, inspeccionar, aceptar, averiguar	Probar	justificar, demostrar, evidenciar, ensayar, comprobar	Conformar (se con)	adaptar, adecuar, ajustar, concordar, amoldarse
Inferir	originar, argumentar, razonar, entender, inducir, concluir, deducir, discurrir, derivar, relacionar, teorizar	Utilizar	usar, emplear, manejar, aplicar	Respetar	considerar, admirar, honrar
Resumir	recapitular, sintetizar	Elaborar	confeccionar, fabricar, hacer, proyectar, producir, realizar, transformar	Actuar	trabajar, ejercer, proceder, ejecutar, elaborar, intervenir
Clasificar	numerar, especificar	Construir	fabricar, cimentar, obrar	Preocupar (se)	inquietar, angustiar, fomentar, prevenir, interesarse, ocuparse, responsabilizarse
Generalizar	universalizar, pluralizar, diversificar, extender	Simular	practicar, representar, idear	Tolerar	sobrellevar, soportar, admitir, aceptar, consentir, comprender
Describir	detallar, explicar, pormenorizar, especificar, reseñar, referir, determinar, definir	Aplicar	colocar, adaptar, destinar, estudiar, administrar, emplear, manejar, usar, utilizar	Conocer	comprender, averiguar, relacionarse, entender
Comentar	esclarecer, interpretar, explicar, aclarar, parafrasear, ilustrar	Reconstruir	rehacer, reparar, reproducir, repetir	Deleitar(se)	agradar, complacerse, recrearse
Distinguir	apreciar, comprender, analizar, discernir, observar, resaltar, separar, señalar, seleccionar,	Demostrar	justificar, razonar, enseñar, probar, argumentar, declarar, evidenciar,	Apreciar	considerar, querer, valorar, respetar, tener en cuenta, tener en aprecio

SABER		HACER		SER	
Verbo	Sinónimos	Verbo	Sinónimos	Verbo/actitud	Sinónimos/Explicación
	diferenciar, reconocer, argumentar, clarificar, ver identificar, notar		exponer, señalar, mostrar, manifestar, indicar		
Comparar	cotejar, examinar, confrontar, parangonar, contrastar, equiparar, relacionar	Recoger	reunir, agrupar, recolectar, acopiar	Dar (se) cuenta	facilitar, dedicarse, aportar
Interpretar	Analizar, comentar, entender, explicar, deducir, representar, aclarar, ilustrar, definir, describir	Presentar	exponer, descubrir, relacionar, explicar, enseñar, mostrar, producir	Inclinarse (se) por	propender, apoyarse
Relacionar	enlazar, unir, relatar, describir, contar, vincular, encadenar, explicar, conectar, coordinar, referir	Planificar	proyectar, planear, programar	Prestar (atención a)	proporcionar, dar, conceder
Conocer	comprender, averiguar, saber, entender, percibir, percatarse, enterarse, dominar	Experimentar	examinar, estudiar, notar, probar, advertir, apreciar, observar, comprobar, ensayar, percibir	Aceptar	comprometerse, acceder, admitir
Recordar	mencionar, evocar, recordar, aludir, acordarse, recapitular	Ejecutar	Realizar, elaborar, emprender, verificar, efectuar, cumplir, hacer	Interesar (se por)	afanarse, apasionar, concernir, cautivar, inquietarse, preocuparse
Indicar	mostrar, orientar, sugerir, señalar, guiar, observar	Componer	arreglar, rectificar, corregir, crear, formar, reparar, hacer, constituir	Ser (conciente de)	
Explicar	aclarar, justificar, definir, argüir, esclarecer, ilustrar, decir, expresarse, declarar, elucidar, dilucidar, enseñar, interpretar, describir, razonar	Justificar	evidenciar, testimoniar, razonar, demostrar, explicar, argumentar, salvar, documentar, excusar, respaldar	Permitir	proporcionar, consentir, posibilitar, conceder
Enumerar	exponer, mencionar, listar, detallar, especificar, catalogar, numerar, enunciar, referir, nombrar	Cuantificar	medir, ponderar	Valorar	estimar, apreciar
Definir	precisar, explicar, detallar, especificar, aclarar, puntualizar, delimitar, determinar	Hallar	descubrir, obrar, encontrar, averiguar, inventar, solucionar, observar, percatar	Colaborar	Contribuir, reforzar, apoyar, contribuir, cooperar
Especificar	establecer, diferenciar, determinar, precisar, detallar, pormenorizar, enumerar, delimitar, explicar, definir, describir, relacionar, distinguir	Encontrar	hallar, inventar, descubrir	Acordar	Concertar, conciliar, pactar
Establecer	erigir, instaurar, constituir,	Interpretar	analizar, comentar,	Argumentar	Aducir, Argüir, cuestionar,

SABER		HACER		SER	
Verbo	Sinónimos	Verbo	Sinónimos	Verbo/actitud	Sinónimos/Explicación
	decretar, organizar		entender, explicar, deducir, representar		discutir
Delimitar	limitar, acotar, definir, aclarar, determinar, establecer, señalar	Identificar	establecer, unificar, reconocer, determinar, equiparar, referir, describir, reseñar, detallar, igualar, registrar	Asumir	Tomar ,adquirir
Precisar	determinar, detallar, concretar, especificar, describir, establecer	Mencionar	referir, citar, indicar, aludir, nombrar	Participar en	Tomar parte en, interesar
Nombrar	mencionar, citar, designar, denominar, aludir, señalar	Clasificar	catalogar, separar, coordinar, ordenar, organizar	Ofrecer (se)	Comprometer (se)
Referir	mencionar, citar, describir, explicar, exponer, aludir, representar, detallar, especificar	Emplear	ocupar, destinar, disponer, colocar, manejar, utilizar, servirse, valerse, usar, aplicar	Actuar en forma transigente	Consentir en parte con lo que no se cree justo, razonable o verdadero, a fin de acabar con una diferencia
Citar	aludir, mencionar, nombrar, referir, enumerar, señalar	Expresar	declarar, manifestar, hablar, reflejar, decir, significar, opinar	Actuar de manera comprensiva y tolerante	comprender
Recapitular	resumir, compendiar, recordar, reseñar, sintetizar	Evaluar	estimar, determinar, valorar, calcular, tasar	Reflexionar	Considerar, cavilar
Presentar	exponer, descubrir, relacionar, explicar, enseñar, indicar	Obtener	adquirir, alcanzar, sacar, producir, lograr, recibir, elaborar	Interesarse (por)	Curiosear, prestar atención
Ilustrar	explicar, informar, instruir, aclarar	Calcular	computar, tasar, valorar, evaluar, contar, deducir	Cooperar	Aportar, influir, colaborar, apoyar
Reseñar	describir, contar, resumir, referir, especificar, detallar	Reconocer	recordar, buscar, investigar, examinar, explorar, observar, registrar, inspeccionar	Demostrar Orden, Calidad y Precisión	
Deducir	inferir, concluir, inducir, teorizar	Enunciar	especificar, decir, exponer, relacionar, explicar, declarar, manifestar, expresar	Dirigir	Guiar, liderar
Mencionar	referir, citar, indicar, aludir, nombrar, señalar	Determinar	precisar, definir, delimitar, resolver, limitar, ordenar, describir, señalar, concluir, especificar, diagnosticar, decidir	Tomar la iniciativa	Actuar con decisión
Discernir	aclarar, distinguir, comprender,	Analizar	estudiar, detallar,	Mostrar	Creatividad

SABER		HACER		SER	
Verbo	Sinónimos	Verbo	Sinónimos	Verbo/actitud	Sinónimos/Explicación
	entender		individualizar, observar, separar, descomponer, averiguar, considerar, examinar, distinguir, comparar, investigar, indagar	capacidad de creación	
Diferenciar	distinguir, discriminar	Referir	mencionar, citar, describir, explicar, relatar, exponer	Decidir	Formar juicios, concluir, resolver
Estipular	concretar, determinar	Adoptar	practicar, acoger, ayudar, aceptar, recoger	Adaptarse (a)	Acomodarse, Avenirse a diversas circunstancias
Detallar	aclarar, señalar, definir, determinar, analizar, pormenorizar, especificar, precisar, puntualizar, referir, delimitar	Relacionar	enlazar, unir, describir, contar, vincular, explicar, conectar, coordinar, referir	Motivar (se)	Promover
Rememorar	rememorar, evocar, recordar, acordarse, recapitular	Deducir	derivar, inferir, concluir, resultar	Planificar	Planear, proyectar
Listar	enumerar, registrar, catalogar	Examinar	averiguar, observar, reconocer, analizar, verificar, comprobar, inspeccionar, estudiar, indagar, investigar	Organizar	Distribuir el tiempo, ordenar las acciones
Relacionar	Contar, referir, relatar	Estudiar	observar, analizar, investigar, examinar, preparar, aprender, formarse, instruirse, educarse	Compartir	Comunicar, participar, tomar parte, auxiliar,
Plantear	Esbozar, diseñar, idear, proyectar, proponer	Medir	valorar, calcular, evaluar, determinar, establecer, contar, medir, comprobar, calibrar	Ayudar	Aportar, influir, colaborar, apoyar, cooperar
Asociar	relacionar	Elaborar	confeccionar, fabricar, hacer, proyectar, producir, realizar, transformar	Aceptar	Aprobar, admitir, consentir
Exponer	Mostrar, presentar, explicar, interpretar	Verificar	constatar, revisar, comprobar, probar, examinar, justificar, demostrar, evidenciar, realizar, cotejar, confirmar	Discutir	Debatir, cuestionar
Señalar	Mencionar, decir, recalcar, nombrar	Efectuar	practicar, ejecutar, realizar, verificar, hacer, actuar, obrar	Invitar	Estimular, impulsar, inducir, instar

SABER		HACER		SER	
Verbo	Sinónimos	Verbo	Sinónimos	Verbo/actitud	Sinónimos/Explicación
		Transformar	cambiar, modificar, elaborar, restaurar, reformar	Actuar	Obrar ,proceder, conducirse, portarse, desenvolverse
		Realizar	elaborar, producir, proceder, concluir, crear, desarrollar, hacer, componer, ejecutar, efectuar, confeccionar	Proponer	Plantear, exponer, formular, recomendar, opinar, insinuar
		Resumir	recapitular, compendiar, condensar, sintetizar, extractar, esquematizar, compilar	Trabajar	Ejercer, elaborar, ocuparse
		Clasificar	catalogar, separar, ordenar, organizar	Mediar	Interceder, intervenir
		Describir	explicar, pormenorizar, especificar, exponer, representar, relatar	Organizar	Establecer, instaurar, emprender
		Implementar	Realizar, efectuar, hacer	Aprobar	Calificar, asentir, certificar
		caracterizar	Determinar, definir, identificar ,describir, especificar	Motivar	Infundir, incitar, promover, suscitar
		Representar	caracterizar	Dirigir	Guiar, administrar, orientar, aconsejar, conducir
		Diseñar	Planear, Proyectar, Plantear, bosquejar	Juzgar de manera crítica	Evaluar, apreciar
		Modelar	configurar	Comunicar	Participar, anunciar
		Comprobar	Corroborar, confirmar, probar	Manejar conflictos	
		Esbozar	bosquejar	Sentido Estético	
		Expresar	formular	Mostrar Disposición crítica	

NOTAS: Al plantear los saberes se debe tener presente que:

1. Los mismos verbos pueden, en algunos casos, usarse en diferentes categorías (saber, ser, hacer).
2. Un mismo contenido temático puede estar relacionado con varios saberes, en la medida en que puede ser objeto de diferentes tipos de aprendizaje.

ANEXO C.

**DOCUMENTOS RELACIONADOS CON LA REALIZACIÓN DEL DISEÑO
INSTRUCCIONAL**

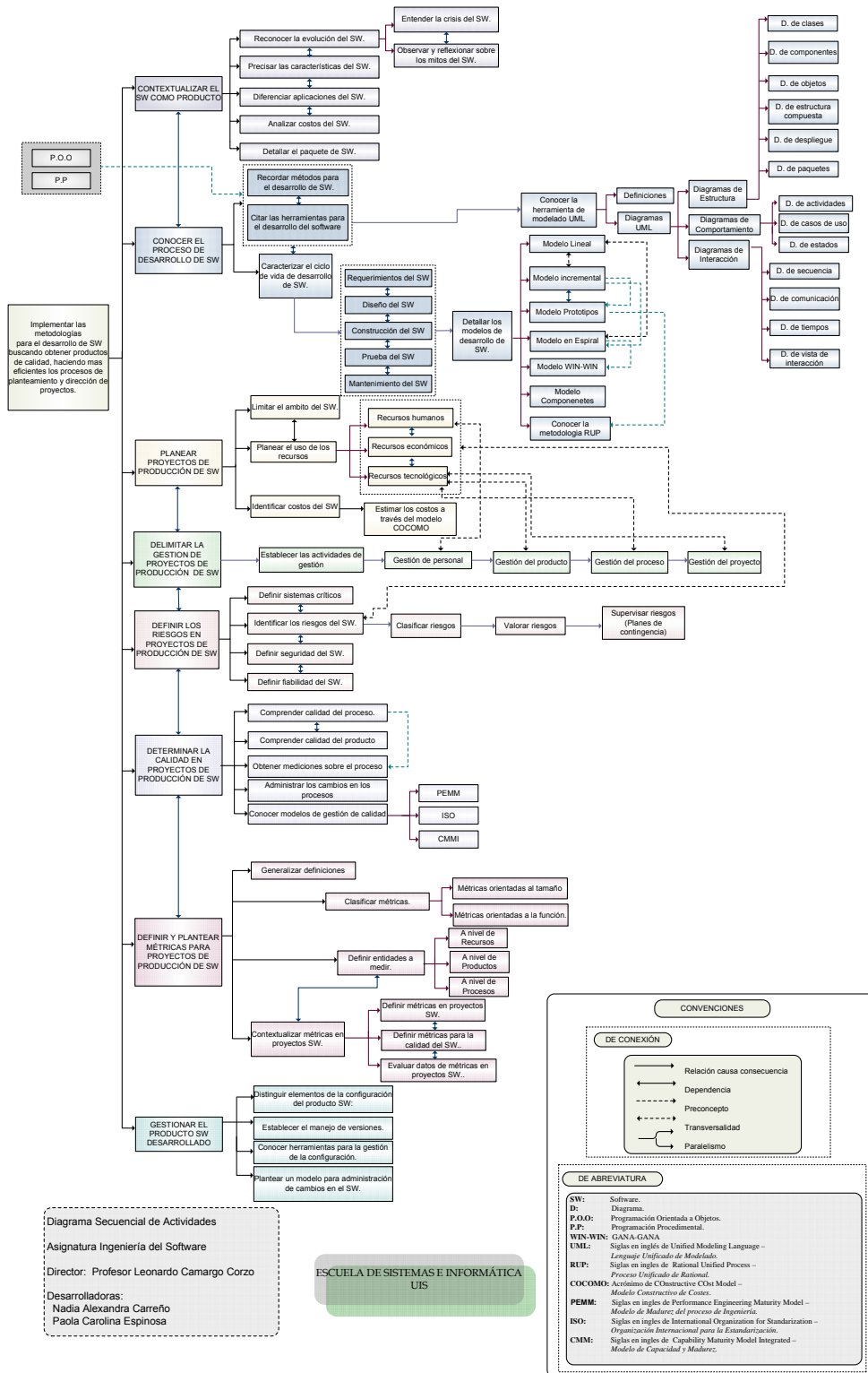


Diagrama Secuencial de Actividades
Asignatura Ingeniería del Software
Director: Profesor Leonardo Camargo Corzo
Desarrolladoras:
Nadia Alexandra Carreño
Paola Carolina Espinosa

EESCUELA DE SISTEMAS E INFORMÁTICA
UIS

TABLA DE SABERES ASIGNATURA: Ingeniería del Software	Versión Final	Página 1
CONTEXTUALIZAR EL SOFTWARE COMO PRODUCTO		
SABER	HACER	

<p>F. Precisar el concepto de software.</p> <p>G. Reconocer la evolución del software a través de la historia.</p> <p>H. Comprender el fenómeno de la crisis del software.</p> <p>I. Precisar las alternativas dadas para solucionar los problemas que surgieron al desarrollar software.</p> <p>J. Reflexionar sobre los mitos del software.</p> <p>K. Enunciar las características del software como producto.</p> <p>L. Analizar las aplicaciones del software dentro de sus ámbitos de utilidad.</p> <p>M. Analizar el software en su nueva tendencia como servicio.</p> <p>N. Enunciar los costos necesarios para el desarrollo de software.</p> <p>O. Especificar los componentes del paquete software desde la perspectiva del software como producto.</p>	<p>a. Describir los sucesos más relevantes de cada una de las etapas en la evolución del software. [1, 2]</p> <p>b. Resumir los problemas que dieron origen a la crisis del software. [1, 3]</p> <p>c. Argumentar los factores relacionados con la crisis del software. [3]</p> <p>d. Concluir las causas por las cuales se presenta insatisfacción sobre el software desarrollado. [3]</p> <p>e. Citar ideas para minimizar vulnerabilidad del software ante la crisis. [3, 4]</p> <p>f. Señalar los mitos más relevantes que existen en la elaboración de productos software. [5]</p> <p>g. Identificar los entes afectados por los mitos en el desarrollo de software. [5]</p> <p>h. Detallar las características que hacen la diferencia entre el software y cualquier otro tipo de producto. [6]</p> <p>i. Caracterizar el ámbito en el que se desarrolla el producto software. [7]</p> <p>j. Precisar el tipo de aplicación software dependiendo del ámbito en el que se desarrolla. [7]</p> <p>k. Citar nuevas posibilidades de ofrecer servicios de soporte sobre el software. [8]</p> <p>l. Listar los parámetros involucrados en el cálculo del costo total de proyectos de producción de software. [9]</p> <p>m. Entender el software como un paquete compuesto. [10]</p>
---	--

TABLA DE SABERES ASIGNATURA: Ingeniería del Software	Versión Final	Página 2
CONOCER EL PROCESO DE DESARROLLO DE SOFTWARE		
SABER	HACER	

<ol style="list-style-type: none"> 1. Detallar métodos para el desarrollo de software. 2. Definir las herramientas necesarias para el desarrollo de software. 3. Diferenciar los tres grupos de diagramas UML: de Estructura, de Comportamiento y de Interacción. 4. Comprender cada uno de los trece (13) diagramas UML: de clases, de componentes, de objetos, de estructura compuesta, de despliegue, de paquetes, de actividades, de casos de uso, de estados, de secuencia, de comunicación, de tiempos y de vista de interacción. 5. Definir el concepto de ciclo de vida de desarrollo de software. 6. Reconocer las cuatro (4) etapas fundamentales del ciclo de vida de desarrollo de software: Análisis, Diseño, Desarrollo e Implementación. 7. Distinguir los modelos utilizados en el desarrollo de software: Modelos lineal, incremental, de prototipos, en espiral, win - win, de componentes y RUP. 8. Describir las etapas de cada modelo de desarrollo de software. 9. Analizar para los diferentes modelos empleados en el desarrollo de software sus ventajas y/o desventajas. 	<ol style="list-style-type: none"> a. Identificar los métodos y herramientas para el desarrollo de software. [1, 2] b. Clasificar cada uno de los trece (13) diagramas UML en el grupo al cual pertenece. [3, 4] c. Elaborar cada uno de los trece (13) diagramas UML. [4] d. Interpretar cada uno de los trece (13) diagramas UML. [4] e. Documentar los diagramas UML. [4] f. Determinar los atributos del ciclo de vida de desarrollo de software. [5, 6] g. Pormenorizar cada una de las cuatro (4) etapas fundamentales del ciclo de vida de desarrollo de software: Análisis, Diseño, Desarrollo e Implementación. [6] h. Caracterizar cada uno de los modelos utilizados en el desarrollo de software. [5, 6, 7] i. Relacionar las etapas de cada modelo de desarrollo de software con las cuatro (4) etapas fundamentales del ciclo de vida de desarrollo de software. [6, 7, 8] j. Delimitar cada una de las etapas de los diferentes modelos de desarrollo software. [8] k. Sugerir un modelo para el desarrollo de un software que satisfaga sus características particulares. [6, 7, 8, 9] l. Elaborar la documentación de requisitos para un proyecto planteado. [8]
---	---

PLANEAR PROYECTOS DE PRODUCCIÓN DE SOFTWARE

SABER

HACER

- | | |
|---|--|
| <ul style="list-style-type: none"> C. Conocer el concepto de ámbito del software en proyectos de producción de software. D. Precisar los elementos que se deben tener en cuenta para definir el ámbito del software. E. Reconocer los recursos a planear en proyectos de producción de software: humanos, económicos, tecnológicos. F. Precisar las actividades de planificación asociadas a cada recurso en proyectos de producción de software. G. Estudiar la estimación de costos involucrados en proyectos de producción de software. H. Analizar las diferentes técnicas para la estimación de costos en proyectos de producción de software. I. Conocer el modelo de estimación de costos COCOMO. | <ul style="list-style-type: none"> a. Delimitar el ámbito del software en proyectos de producción de software. [1, 2] b. Describir las características de cada recurso involucrado en la planeación de proyectos de producción de software. [3] c. Identificar las actividades de cada recurso relacionadas con la planeación de proyectos de producción de software. [3] d. Documentar un plan de proyecto apoyado en los conceptos vistos y un ejercicio particular definido. [1...4] e. Valorar los costos involucrados en proyectos de producción de software. [5, 6] f. Plantear ejemplos sobre el uso del modelo COCOMO en la estimación de costos en proyectos de producción de software. [7] |
|---|--|

TABLA DE SABERES ASIGNATURA: Ingeniería del Software	Versión Final	Página 4
DELIMITAR LA GESTIÓN DE PROYECTOS DE PRODUCCIÓN DE SOFTWARE		
SABER	HACER	

<ol style="list-style-type: none"> 1. Entender el concepto de gestión de proyectos de producción de software. 2. Especificar las actividades de gestión de proyectos de producción de software. 3. Comprender, por medio de las 4P (personal, producto, proceso y proyecto), la gestión integral de proyectos de producción de software 4. Definir los procesos de gestión de personal en proyectos de producción de software. 5. Precisar procesos de gestión del producto en proyectos de producción de software. 6. Explicar los procesos de gestión del proceso en proyectos de producción de software. 7. Detallar los procesos de gestión del proyecto en proyectos de producción de software. 	<ol style="list-style-type: none"> a. Describir las actividades de gestión de proyectos de producción de software. [1, 2] b. Analizar problemas presentes en la gestión de proyectos de producción de software. [1, 2, 3] c. Identificar la función que desempeña cada una de las 4P (personal, producto, proceso y proyecto) en la gestión de proyectos de producción de software. [3] d. Diferenciar los procesos de gestión de las 4P en proyectos de producción de software. [4, 5, 6, 7] e. Mencionar los puntos de gestión eficiente para cada una de las 4P en proyectos de producción de software. [4, 5, 6, 7]
---	--

TABLA DE SABERES ASIGNATURA: Ingeniería del Software	Versión Final	Página 5
DEFINIR LOS RIESGOS EN PROYECTOS DE PRODUCCIÓN DE SOFTWARE		
SABER	HACER	

<ol style="list-style-type: none"> 1. Estudiar el concepto de sistema crítico en proyectos de producción de software. 2. Identificar los riesgos del software dentro del desarrollo de proyectos de producción de software. 3. Caracterizar los riesgos presentes en proyectos de producción de software. 4. Definir parámetros de supervisión de riesgos en proyectos de producción de software. 5. Definir seguridad en proyectos de producción de software. 6. Analizar la fiabilidad en proyectos de producción de software. 	<ol style="list-style-type: none"> a. Identificar un sistema crítico en proyectos de producción de software. [1] b. Diagnosticar los riesgos conocidos y predecibles en proyectos de producción de software. [2, 3] c. Diferenciar los riesgos genéricos de todo proyecto de producción de software de los riesgos específicos del proceso y del producto. [2, 3] d. Detallar planes de contingencia para supervisar los riesgos en proyectos de producción de software. [2, 3] e. Reconocer los parámetros que enmarcan la seguridad en proyectos de producción de software. [4] f. Inspeccionar la fiabilidad en proyectos de producción de software. [5]
--	---

TABLA DE SABERES ASIGNATURA: Ingeniería del Software	Versión Final	Página 6
DETERMINAR LA CALIDAD EN PROYECTOS DE PRODUCCIÓN DE SOFTWARE		
SABER	HACER	

<ol style="list-style-type: none"> 1. Establecer el concepto de calidad del software. 2. Identificar los factores necesarios para que un proceso de producción de software sea considerado de calidad. 3. Reconocer los factores que determinan la calidad de un producto de software. 4. Plantear las medidas que se deben hacer sobre los factores que determinan la calidad en un proceso de producción de software. 5. Comprender la necesidad de administrar los cambios en los factores que determinan la calidad en un proceso de producción de software. 6. Conocer la definición de modelo de calidad de software. 7. Exponer los modelos de gestión de calidad: PEMM, ISO, CMMI. 	<ol style="list-style-type: none"> a. Discutir sobre la importancia de la calidad en proyectos de producción de software. [1] b. Listar los factores que determinan la calidad en un proceso de producción y en un producto software. [2, 3] c. Evaluar cada uno de los factores que determinan la calidad en un proceso de producción y en producto software. [3] d. Efectuar mediciones sobre los factores que determinan la calidad en un proceso de producción de software. [2, 3, 4] e. Analizar las mediciones realizadas sobre los factores que determinan la calidad en un proceso de producción de software. [2, 3, 4] f. Aplicar la administración de los cambios en los factores que determinan la calidad en un proceso de producción de software. [5] g. Reconocer la importancia de adoptar un método de gestión de calidad para el desarrollo de proyectos de producción de software. [6, 7] h. Diferenciar claramente la estructura de cada uno de los modelos de gestión de calidad. [7]
---	---

TABLA DE SABERES ASIGNATURA: Ingeniería del Software	Versión Final	Página 7
DEFINIR Y PLANTEAR MÉTRICAS PARA PROYECTOS DE PRODUCCIÓN DE SOFTWARE		
SABER	HACER	

<ol style="list-style-type: none"> 1. Definir métricas, medidas e indicadores en proyectos de producción de software. 2. Establecer las Métricas del software. 3. Conocer las cinco (5) actividades (Formulación, Colección, Análisis, Interpretación y Realimentación) en el proceso de medición en proyectos de producción de software. 4. Definir las métricas orientadas al tamaño. 5. Definir las métricas orientadas a la función. 6. Describir las ventajas y desventajas de las métricas orientadas al tamaño y orientadas a la función. 7. Conocer el concepto de atributo dentro del contexto de las métricas de software. 8. Definir los atributos internos de un producto software. 9. Detallar los atributos externos de un producto software. 10. Distinguir las entidades que son objeto de medición en proyectos de producción de software a nivel de: recursos, productos y procesos. 11. Entender el concepto de métricas de proyectos de producción software. 12. Determinar métricas para la calidad en proyectos de producción de software. 13. Conocer los factores de calidad que determinan las métricas de calidad en proyectos de producción de software. 	<ol style="list-style-type: none"> a. Diferenciar los conceptos de métricas, medidas e indicadores en proyectos de producción de software. [1] b. Interpretar el concepto de métricas en el contexto de software. [2] c. Caracterizar cada una de las cinco (5) actividades en el proceso de medición en proyectos de producción de software. [3] d. Calcular métricas orientadas al tamaño. [4, 6] e. Calcular métricas orientadas a la función. [5, 6] f. Estimar los beneficios de aplicar tanto las métricas orientadas al tamaño como las orientadas a la función. [4, 5, 6] g. Clasificar cada uno de los atributos en internos y externos en proyectos de producción de software. [7, 8, 9] h. Describir cada una de las entidades a medir en proyectos de producción de software a nivel de: Recursos, Productos y Procesos. [10] i. Relacionar los atributos internos y externos con las entidades a las cuales pertenecen en proyectos de producción de software. [8, 9, 10] j. Plantear métricas para proyectos de producción de software. [11] k. Deducir los propósitos y objetivos de emplear métricas en proyectos de producción de software. [11] l. Estimar los beneficios de implementar métricas de calidad en proyectos de producción de software. [12] m. Identificar las medidas de calidad necesarias para aplicar métricas en proyectos de producción de software a través de los factores de calidad del
--	--

	<p>software. [13]</p> <p>n. Describir el proceso de evaluación y modelación del software mediante la aplicación de métricas. [1... 13]</p>
--	--

TABLA DE SABERES ASIGNATURA: Ingeniería del Software	Versión Final	Página 9
GESTIONAR EL PRODUCTO SOFTWARE DESARROLLADO		
SABER	HACER	

<ol style="list-style-type: none"> 1. Plantear el concepto de Gestión de la Configuración del Software. 2. Conocer los elementos de la configuración del producto software. 3. Identificar los procedimientos necesarios para el manejo de versiones de software. 4. Reconocer los beneficios y dificultades de la implementación de una nueva versión de software. 5. Señalar algunas herramientas para la gestión de la configuración del software. 6. Identificar los tipos de cambios que puede requerir un producto software a través del tiempo. 7. Determinar procesos para el control de cambios en el software. 	<ol style="list-style-type: none"> a. Evidenciar la necesidad de la gestión de configuración del software. [1] b. Establecer las relaciones que existen entre los diferentes elementos de la configuración del software. [2] c. Establecer políticas de implementación de nuevas versiones de software. [3, 4] d. Definir procesos y herramientas necesarios para realizar la gestión de configuración del producto software. [3, 5] e. Analizar el impacto de la introducción de un cambio en un elemento del software, sobre los demás. [6, 7] f. Plantear un modelo para la administración de cambios en el software. [6, 7]
---	---

RELACIÓN PROPÓSITOS CONTENIDOS		Versión Final	Página 1
CONTEXTUALIZAR EL SOFTWARE COMO PRODUCTO			
PROPÓSITOS	CONTENIDOS TEMÁTICOS	SABER	HACER
<p>Proporcionar información acerca de aspectos relevantes en la evolución del software.</p>	<ul style="list-style-type: none"> ➤ Conceptualización del software ➤ La evolución del software ➤ Crisis del software ➤ Los mitos del software 	<ol style="list-style-type: none"> 1. Precisar el concepto de software. 2. Reconocer la evolución del software a través de la historia. 3. Comprender el fenómeno de la crisis del software. 4. Precisar las alternativas dadas para solucionar los problemas que surgieron al desarrollar software. 5. Reflexionar sobre los mitos del software. 	<ol style="list-style-type: none"> a. Describir los sucesos más relevantes de cada una de las etapas en la evolución del software. [1, 2] b. Resumir los problemas que dieron origen a la crisis del software. [1, 3] c. Argumentar los factores relacionados con la crisis del software. [3] d. Concluir las causas por las cuales se presenta insatisfacción sobre el software desarrollado. [3] e. Citar ideas para minimizar vulnerabilidad del software ante la crisis. [3, 4] f. Señalar los mitos más relevantes que existen en la elaboración de productos software. [5] g. Identificar los entes afectados por los mitos en el desarrollo de software. [5]
<p>Identificar las características del software como producto, estudiar las aplicaciones más importantes y conocer su nuevo enfoque como servicio.</p>	<ul style="list-style-type: none"> ➤ Características del software ➤ Aplicaciones del software <ul style="list-style-type: none"> ▪ Software de sistemas ▪ Software de tiempo real ▪ Software de gestión ▪ Software de ingeniería y científico ▪ Software empotrado ▪ Software de computadoras personales ▪ Software de inteligencia artificial ▪ Software como servicio. 	<ol style="list-style-type: none"> 1. Enunciar las características del software como producto. 2. Analizar las aplicaciones del software dentro de sus ámbitos de utilidad. 3. Analizar el software en su nueva tendencia como servicio. 	<ol style="list-style-type: none"> a. Detallar las características que hacen la diferencia entre el software y cualquier otro tipo de producto. [1] b. Caracterizar el ámbito en el que se desarrolla el producto software. [2] c. Precisar el tipo de aplicación software dependiendo del ámbito en el que se desarrolla. [2] d. Citar nuevas posibilidades de ofrecer servicios de soporte sobre el software. [3]

<p>Observar los costos generales del desarrollo de proyectos de producción de software.</p>	<ul style="list-style-type: none"> ➤ Definición de costos del software ➤ Tipos de costos: <ul style="list-style-type: none"> ▪ Costos de hardware y software ▪ Costos de viajes y capacitación ▪ Costos de esfuerzo 	<p>1. Enunciar los costos necesarios para el desarrollo de software.</p>	<p>a. Listar los parámetros involucrados en el cálculo del costo total de proyectos de producción de software. [1]</p>
<p>Identificar los componentes del paquete software</p>	<ul style="list-style-type: none"> ➤ Componentes del paquete software: <ul style="list-style-type: none"> ▪ Licencias ▪ Marco legal 	<p>1. Especificar los componentes del paquete software desde la perspectiva del software como producto.</p>	<p>a. Entender el software como un paquete compuesto. [1]</p>

RELACIÓN PROPÓSITOS CONTENIDOS		Versión Final	Página 3
CONOCER EL PROCESO DE DESARROLLO DE SOFTWARE			
PROPÓSITOS	CONTENIDOS TEMÁTICOS	SABER	HACER
Recapitular sobre los conocimientos acerca de los métodos y herramientas que se usan en el proceso de desarrollo de software.	<ul style="list-style-type: none"> ➤ Métodos para el desarrollo de software: <ul style="list-style-type: none"> ▪ Programación orientada a objetos (P.O.O) ▪ Programación procedimental (P.P) ➤ Herramientas para el desarrollo de software. 	<ol style="list-style-type: none"> 1. Detallar métodos para el desarrollo de software. 2. Definir las herramientas necesarias para el desarrollo de software. 	<ol style="list-style-type: none"> a. Identificar los métodos y herramientas para el desarrollo de software. [1, 2]
Comprender un lenguaje gráfico para visualizar, especificar, construir y documentar un desarrollo de software.	<ul style="list-style-type: none"> ➤ Definición de UML ➤ Clasificación de los 13 diagramas UML en sus 3 grandes grupos: <ul style="list-style-type: none"> ▪ Diagramas de estructura ▪ Diagramas de comportamiento ▪ Diagramas de interacción ➤ Énfasis de cada uno de los grupos de diagramas UML ➤ Objetivo de cada uno de los diagramas UML <ul style="list-style-type: none"> ▪ Características de cada uno de los diagramas UML 	<ol style="list-style-type: none"> 1. Diferenciar los tres (3) grupos de diagramas UML: de Estructura, de Comportamiento y de Interacción. 2. Comprender cada uno de los trece (13) diagramas UML: de clases, de componentes, de objetos, de estructura compuesta, de despliegue, de paquetes, de actividades, de casos de uso, de estados, de secuencia, de comunicación, de tiempos y de vista de interacción. 	<ol style="list-style-type: none"> a. Clasificar cada uno de los trece (13) diagramas UML en el grupo al cual pertenece. [1, 2] b. Elaborar cada uno de los trece (13) diagramas UML. [2] c. Interpretar cada uno de los trece (13) diagramas UML. [2] a. Documentar los diagramas UML. [2]
Describir el ciclo de vida de desarrollo de software mediante sus etapas básicas.	<ul style="list-style-type: none"> ➤ Concepto de ciclo de vida del desarrollo de software. ➤ Las cuatro (4) etapas del ciclo de vida del desarrollo de software: <ul style="list-style-type: none"> ▪ Análisis ▪ Diseño ▪ Desarrollo ▪ Implementación 	<ol style="list-style-type: none"> 1. Definir el concepto de ciclo de vida de desarrollo de software. 2. Reconocer las cuatro (4) etapas fundamentales del ciclo de vida de desarrollo de software: Análisis, Diseño, Desarrollo e Implementación. 	<ol style="list-style-type: none"> a. Determinar los atributos del ciclo de vida de desarrollo de software. [1, 2] b. Pormenorizar cada una de las cuatro (4) etapas fundamentales del ciclo de vida de desarrollo de software: Análisis, Diseño, Desarrollo e Implementación. [2]

<p>Comprender la estructuración de las etapas de cada modelo de desarrollo del software.</p>	<p>➤ Modelos de desarrollo de software:</p> <ul style="list-style-type: none"> ▪ Modelo lineal ▪ Modelo incremental ▪ Modelo prototipos ▪ Modelo en espiral ▪ Modelo Win-Win ▪ Modelo componentes ▪ Metodología RUP 	<ol style="list-style-type: none"> 1. Distinguir los modelos utilizados en el desarrollo de software: Modelos lineal, incremental, de prototipos, en espiral, win - win, de componentes y RUP. 2. Describir las etapas de cada modelo de desarrollo de software. 3. Analizar para los modelos empleados en el desarrollo de software sus ventajas y/o desventajas, 	<ol style="list-style-type: none"> a. Caracterizar cada uno de los modelos utilizados en el desarrollo de software. [1, 2, 3] b. Relacionar las etapas de cada modelo de desarrollo de software con las cuatro (4) etapas fundamentales del ciclo de vida de desarrollo de software. [1, 2, 3] c. Delimitar cada una de las etapas de los diferentes modelos de desarrollo software. [2] d. Sugerir un modelo para el desarrollo de un software que satisfaga sus características particulares. [1, 2, 3] e. Elaborar la documentación de requisitos para un proyecto planteado. [1, 2, 3]
---	--	---	---

RELACIÓN PROPÓSITOS CONTENIDOS		Versión Final	Página 5
PLANEAR PROYECTOS DE PRODUCCIÓN DE SOFTWARE			
PROPÓSITOS	CONTENIDOS TEMÁTICOS	SABER	HACER
Definir el ámbito del software mediante los aspectos básicos que permitan su delimitación.	➤ Ámbito del software	<ol style="list-style-type: none"> 1. Conocer el concepto de ámbito del software en proyectos de producción de software. 2. Precisar los elementos que se deben tener en cuenta para definir el ámbito del software. 	<ol style="list-style-type: none"> a. Delimitar el ámbito del software en proyectos de producción de software. [1, 2]
Comprender las actividades para llevar a cabo la planeación de proyectos de producción de software	➤ Definición de actividades asociadas a la planificación de proyectos. <ul style="list-style-type: none"> ▪ Uso de los recursos <ul style="list-style-type: none"> • Recursos humanos • Recursos tecnológicos • Recursos económicos 	<ol style="list-style-type: none"> 1. Reconocer los recursos a planear en proyectos de producción de software: humanos, económicos, tecnológicos. 2. Precisar las actividades de planificación asociadas a cada recurso en proyectos de producción de software. 	<ol style="list-style-type: none"> a. Describir las características de cada recurso involucrado en la planeación de proyectos de producción de software. [1] b. Identificar las actividades de cada recurso relacionadas con la planeación de proyectos de producción de software. [1] c. Documentar un plan de proyecto apoyado en los conceptos vistos y un ejercicio particular definido. [1, 2]
Comprender el modelo COCOMO para la estimación de costos en proyectos de producción de software	➤ Estimación de costos ➤ Modelo COCOMO <ul style="list-style-type: none"> ▪ Básico ▪ Intermedio ▪ Avanzado 	<ol style="list-style-type: none"> 1. Estudiar la estimación de costos involucrados en proyectos de producción de software. 2. Analizar las diferentes técnicas para la estimación de costos en proyectos de producción de software. 3. Conocer el modelo de estimación de costos COCOMO. 	<ol style="list-style-type: none"> a. Valorar los costos involucrados en proyectos de producción de software. [1, 2] b. Plantear ejemplos sobre el uso del modelo COCOMO en la estimación de costos en proyectos de producción de software. [3]

RELACIÓN PROPÓSITOS CONTENIDOS		Versión Final	Página 6
DELIMITAR LA GESTIÓN DE PROYECTOS DE PRODUCCIÓN DE SOFTWARE			
PROPÓSITOS	CONTENIDOS TEMÁTICOS	SABER	HACER
<p>Analizar el significado de gestionar proyectos de producción de software mediante las actividades involucradas.</p>	<ul style="list-style-type: none"> ➤ Definición de gestión de proyectos de producción de software ➤ Actividades de gestión <ul style="list-style-type: none"> ▪ Gestión de personal ▪ Gestión de producto ▪ Gestión de proceso ▪ Gestión de proyecto 	<ol style="list-style-type: none"> 1. Entender el concepto de gestión de proyectos de producción de software. 2. Especificar las actividades de gestión de proyectos de producción de software. 3. Comprender por medio de las 4P (personal, producto, proceso y proyecto) la gestión integral de proyectos de producción de software. 4. Definir los procesos de gestión de personal en proyectos de producción de software. 5. Precisar procesos de gestión del producto en proyectos de producción de software. 6. Explicar los procesos de gestión del proceso en proyectos de producción de software. 7. Detallar los procesos de gestión del proyecto en proyectos de producción de software.. 	<ol style="list-style-type: none"> a. Describir las actividades de gestión de proyectos de producción de software. [1, 2] b. Analizar problemas presentes en la gestión de proyectos de producción de software. [1, 2, 3] c. Identificar la función que desempeña cada una de las 4P (personal, producto, proceso y proyecto) en la gestión de proyectos de producción de software. [3] d. Diferenciar los procesos de gestión de las 4P en proyectos de producción de software. [4, 5, 6, 7] e. Mencionar los puntos de gestión eficiente para cada una de las 4P en proyectos de producción de software. [4, 5, 6, 7]

RELACIÓN PROPÓSITOS CONTENIDOS		Versión Final	Página 7
DEFINIR LOS RIESGOS EN PROYECTOS DE PRODUCCIÓN DE SOFTWARE			
PROPÓSITOS	CONTENIDOS TEMÁTICOS	SABER	HACER
Conceptualizar un sistema crítico en proyectos de producción de software.	➤ Definición de sistema crítico en proyectos de producción de software.	1. Estudiar el concepto de sistema crítico en proyectos de producción de software.	a. Identificar un sistema crítico en proyectos de producción de software. [1]
Definir los riesgos presentes en el desarrollo de proyectos de producción del software.	➤ Riesgos del software ➤ Tipos de riesgos: <ul style="list-style-type: none"> ▪ Riesgos genéricos ▪ Riesgos específicos ➤ Clasificación de riesgos: <ul style="list-style-type: none"> ▪ Riesgos del proyecto ▪ Riesgos técnicos ▪ Riesgos del negocio ➤ Supervisión de riesgos	1. Identificar los riesgos del software dentro del desarrollo de proyectos de producción de software. 2. Caracterizar los riesgos presentes en proyectos de producción de software. 3. Definir parámetros de supervisión de riesgos en proyectos de producción de software.	a. Diagnosticar los riesgos conocidos y predecibles en proyectos de producción de software. [1, 2] b. Diferenciar los riesgos genéricos de todo proyecto de producción de software de los riesgos específicos del proceso y del producto. [1, 2] c. Detallar planes de contingencia para supervisar los riesgos en proyectos de producción de software. [1, 2, 3]
Analizar la fiabilidad y seguridad en proyectos de software	➤ Seguridad en proyectos de producción de software. de ➤ Fiabilidad en proyectos de producción de software. de	1. Definir seguridad en proyectos de producción de software. 2. Analizar la fiabilidad en proyectos de producción de software.	a. Reconocer los parámetros que enmarcan la seguridad en proyectos de producción de software. [1] b. Inspeccionar la fiabilidad en proyectos de producción de software. [2]

RELACIÓN PROPÓSITOS CONTENIDOS		Versión Final	Página 8
DETERMINAR LA CALIDAD EN PROYECTOS DE PRODUCCIÓN DE SOFTWARE			
PROPÓSITOS	CONTENIDOS TEMÁTICOS	SABER	HACER
<p>Identificar las actividades necesarias para determinar los requisitos de calidad de proyectos de producción software.</p>	<ul style="list-style-type: none"> ➤ Definición de calidad del proceso y del producto ➤ Parámetros para determinar la calidad del proceso y del producto ➤ Mediciones sobre el proceso de producción y el producto de software ➤ Administración de los cambios en el proceso de producción de software. 	<ol style="list-style-type: none"> 1. Establecer el concepto de calidad del software. 2. Identificar los factores necesarios para que un proceso de producción de software sea considerado de calidad. 3. Reconocer los factores que determinan la calidad de un producto de software. 4. Plantear las medidas que se deben hacer sobre los factores que determinan la calidad en un proceso de producción de software. 5. Comprender la necesidad de administrar los cambios en los factores que determinan la calidad en un proceso de producción de software. 	<ol style="list-style-type: none"> a. Discutir sobre la importancia de la calidad en proyectos de producción de software. [1] b. Listar los factores que determinan la calidad en un proceso de producción y en un producto software. [2, 3] c. Evaluar cada uno de los factores que determinan la calidad en un proceso de producción y en producto software. [3] d. Efectuar mediciones sobre los factores que determinan la calidad en un proceso de producción de software. [2, 3, 4] e. Analizar las mediciones realizadas sobre los factores que determinan la calidad en un proceso de producción de software. [2, 3, 4] f. Aplicar la administración de los cambios en los factores que determinan la calidad en un proceso de producción de software. [5]
<p>Examinar los diferentes modelos que permiten controlar la calidad en proyectos de producción de software.</p>	<ul style="list-style-type: none"> ➤ Concepto de Modelo de Calidad del software ➤ Tipos de modelos de gestión de calidad: <ul style="list-style-type: none"> ▪ PEMM ▪ ISO ▪ CMMI 	<ol style="list-style-type: none"> 1. Conocer la definición de modelo de calidad de software. 2. Exponer los modelos de gestión de calidad: PEMM, ISO, CMMI. 	<ol style="list-style-type: none"> a. Reconocer la importancia de adoptar un método de gestión de calidad para el desarrollo de proyectos de producción de software. [1, 2] b. Diferenciar claramente la estructura de cada uno de los modelos de gestión de calidad. [2]

RELACIÓN PROPÓSITOS CONTENIDOS		Versión Final	Página 9
DEFINIR Y PLANTEAR MÉTRICAS PARA PROYECTOS DE PRODUCCIÓN DE SOFTWARE			
PROPÓSITOS	CONTENIDOS TEMÁTICOS	SABER	HACER
Definir y analizar los conceptos básicos de métricas en proyectos de producción de software.	<ul style="list-style-type: none"> ➤ Definición de medidas, métricas e indicadores. ➤ Métricas aplicadas al software 	<ol style="list-style-type: none"> 1. Definir métricas, medidas e indicadores e proyectos de producción de software. 2. Establecer las Métricas del software. 3. Conocer las cinco (5) actividades (Formulación, Colección, Análisis, Interpretación y Realimentación) en el proceso de medición en un proyecto de producción de software. 	<ol style="list-style-type: none"> a. Diferenciar los conceptos de métricas, medidas e indicadores en proyectos de producción de software. [1] b. Interpretar el concepto de métricas en el contexto de software. [2] c. Caracterizar cada una de las cinco (5) actividades en el proceso de medición en un proyecto de producción de software. [3]
Conocer los diferentes enfoques para aplicar métricas a proyectos de producción de software	<ul style="list-style-type: none"> ➤ Mediciones del software: <ul style="list-style-type: none"> ▪ Métricas orientadas al tamaño ▪ Métricas orientadas a la función 	<ol style="list-style-type: none"> 1. Definir las métricas orientadas al tamaño. 2. Definir las métricas orientadas a la función. 3. Describir las ventajas y desventajas de las métricas orientadas al tamaño y orientadas a la función. 	<ol style="list-style-type: none"> a. Calcular métricas orientadas al tamaño. [1, 3] b. Calcular métricas orientadas a la función. [2, 3] c. Estimar los beneficios de aplicar tanto las métricas orientadas al tamaño como las orientadas a la función. [1, 2, 3]
Identificar las entidades y atributos para aplicar métricas en proyectos de producción de software.	<ul style="list-style-type: none"> ➤ Concepto de atributo ➤ Clasificación de los atributos: <ul style="list-style-type: none"> ▪ Atributos internos ▪ Atributos externos ➤ Entidades a medir: <ul style="list-style-type: none"> ▪ Recursos ▪ Productos ▪ Procesos 	<ol style="list-style-type: none"> 1. Conocer el concepto de atributo dentro del contexto de las métricas de software. 2. Definir los atributos internos de un producto software. 3. Detallar los atributos externos de un producto software. 4. Distinguir las entidades que son objeto de medición en proyectos de producción de software a nivel de: recursos, productos y procesos. 	<ol style="list-style-type: none"> a. Clasificar cada uno de los atributos de un producto software en internos y externos. [1, 2, 3] b. Describir cada una de las entidades a medir a nivel de: Recursos, Productos y Procesos en proyectos de producción de software. [4] c. Relacionar los atributos internos y externos con las entidades a las cuales pertenecen. [2, 3, 4]
Reconocer la importancia de aplicar métricas a proyectos de producción de software.	<ul style="list-style-type: none"> ➤ Métricas aplicadas a proyectos de producción de software ➤ Definición de métricas de calidad ➤ Recopilación de datos 	<ol style="list-style-type: none"> 1. Entender el concepto de métricas de proyectos de producción de software. 2. Determinar métricas para la calidad en proyectos de producción de software. 	<ol style="list-style-type: none"> a. Plantear métricas para proyectos de producción de software. [1] b. Deducir los propósitos y objetivos de emplear métricas en proyectos de producción de software. [1] c. Estimar los beneficios de

	<ul style="list-style-type: none"> ➤ Calculo de métricas ➤ Evaluación de datos 	<p>3. Conocer los factores de calidad que determinan las métricas de calidad en proyectos de producción de software.</p>	<p>implementar métricas de calidad en proyectos de producción de software. [2]</p> <p>d. Identificar las medidas de calidad necesarias para aplicar métricas en proyectos de producción de software a través de los factores de calidad del software. [3]</p> <p>e. Describir el proceso de evaluación y modelación del software mediante la aplicación de métricas. [1, 2, 3]</p>
--	--	--	--

RELACIÓN PROPÓSITOS CONTENIDOS		Versión Final	Página 11
GESTIONAR EL PRODUCTO SOFTWARE DESARROLLADO			
PROPÓSITOS	CONTENIDOS TEMÁTICOS	SABER	HACER
Comprender la gestión de configuración del producto software y los elementos necesarios para ejecutarla.	<ul style="list-style-type: none"> ➤ Definición de gestión de configuración del producto software ➤ Elementos de la configuración del producto software 	<ol style="list-style-type: none"> 1. Plantear el concepto de Gestión de la Configuración del Software. 2. Conocer los elementos de la configuración del producto software. 	<ol style="list-style-type: none"> a. Evidenciar la necesidad de la gestión de configuración del software. [1] b. Establecer las relaciones que existen entre los diferentes elementos de la configuración del software. [2]
Reconocer el procedimiento para implementar una versión de software	<ul style="list-style-type: none"> ➤ Definición de versión ➤ Manejo de versiones 	<ol style="list-style-type: none"> 1. Identificar los procedimientos necesarios para el manejo de versiones de software. 2. Reconocer los beneficios y dificultades de la implementación de una nueva versión de software. 	<ol style="list-style-type: none"> a. Establecer políticas de implementación de nuevas versiones de software. [1, 2]
Examinar el proceso para llevar a cabo cambios en el producto software con base en las herramientas para gestión de la configuración.	<ul style="list-style-type: none"> ➤ Herramientas para la gestión de la configuración del producto software ➤ Administración de los cambios en el producto software. 	<ol style="list-style-type: none"> 1. Señalar algunas herramientas para la gestión de la configuración del software. 2. Identificar los tipos de cambios que puede requerir un producto software a través del tiempo. 3. Determinar procesos para el control de cambios en el software. 	<ol style="list-style-type: none"> a. Definir procesos y herramientas necesarios para realizar la gestión de configuración del producto software. [1, 3] b. Analizar el impacto de la introducción de un cambio en un elemento del software, sobre los demás. [2, 3] c. Plantear un modelo para la administración de cambios en el software. [2, 3]

ACTIVIDADES DE ENSEÑANZA APRENDIZAJE		VERSIÓN FINAL		PÁGINA 1
CONTEXUALIZAR EL SOFTWARE COMO PRODUCTO				
PROPÓSITO	CONTENIDO TEMÁTICO	SABER	HACER	ACTIVIDADES
<p>Proporcionar información acerca de aspectos relevantes en la evolución del software.</p>	<ul style="list-style-type: none"> ➤ Conceptualización del software ➤ La evolución del software ➤ Crisis del software ➤ Los Mitos del software 	<ol style="list-style-type: none"> 1. Precisar el concepto de software. 2. Reconocer la evolución del software a través de la historia. 3. Comprender el fenómeno de la crisis del software. 4. Precisar las alternativas dadas para solucionar los problemas que surgieron al desarrollar software. 5. Reflexionar sobre los mitos del software. 	<ol style="list-style-type: none"> a. Describir los sucesos más relevantes de cada una de las etapas en la evolución del software. [1, 2] b. Resumir los problemas que dieron origen a la crisis del software. [1, 3] c. Argumentar los factores relacionados con la crisis del software. [3] d. Concluir las causas por las cuales se presenta insatisfacción sobre el software desarrollado. [3] e. Citar ideas para minimizar vulnerabilidad del software ante la crisis. [3, 4] f. Señalar los mitos más relevantes que existen en la elaboración de productos software. [5] g. Identificar los entes afectados por los mitos en el desarrollo de software. [5] 	<p>Argumentar los aspectos relevantes en la evolución del software.</p>
<p>Identificar las características del software como producto, estudiar las aplicaciones más importantes y conocer su nuevo enfoque como servicio.</p>	<ul style="list-style-type: none"> ➤ Características del software ➤ Aplicaciones del software <ul style="list-style-type: none"> ▪ Software de sistemas ▪ Software de tiempo real ▪ Software de gestión ▪ Software de ingeniería y científico ▪ Software empotrado ▪ Software de computadoras personales ▪ Software de inteligencia artificial ▪ Software como servicio. 	<ol style="list-style-type: none"> 1. Enunciar las características del Software como producto. 2. Analizar las aplicaciones del Software dentro de sus ámbitos de utilidad. 3. Analizar el software en su nueva tendencia como servicio. 	<ol style="list-style-type: none"> a. Detallar las características que hacen la diferencia entre el software y cualquier otro tipo de producto. [1] b. Caracterizar el ámbito en el que se desarrolla el producto Software. [2] c. Precisar el tipo de aplicación software dependiendo del ámbito en el que se desarrolla. [2] d. Citar nuevas posibilidades de ofrecer servicios de soporte sobre el software. [3] 	<p>Analizar las características y aplicaciones del software como producto y su nuevo enfoque como servicio.</p>

<p>Observar los costos generales del desarrollo de proyectos de producción de software.</p>	<ul style="list-style-type: none"> ➤ Definición de costos del software ➤ Tipos de costos: <ul style="list-style-type: none"> ▪ Costos de hardware y software ▪ Costos de viajes y capacitación ▪ Costos de esfuerzo 	<p>1. Enunciar los costos necesarios para el desarrollo de software.</p>	<p>a. Listar los parámetros involucrados en el cálculo del costo total de proyectos de producción de software. [1]</p>	<p>Generalizar los costos de desarrollo de proyectos de software.</p>
<p>Identificar los componentes del paquete software.</p>	<ul style="list-style-type: none"> ➤ Componentes del paquete software: <ul style="list-style-type: none"> ▪ Licencias ▪ Marco legal 	<p>1. Especificar los componentes del paquete Software desde la perspectiva del software como producto.</p>	<p>a. Entender el software cómo un paquete compuesto. [1]</p>	<p>Especificar los componentes del paquete software.</p>

ACTIVIDADES DE ENSEÑANZA APRENDIZAJE		VERSIÓN FINAL		PÁGINA 3
CONOCER EL PROCESO DE DESARROLLO DE SOFTWARE				
PROPÓSITO	CONTENIDO TEMÁTICO	SABER	HACER	ACTIVIDADES
Recapitular sobre los conocimientos acerca de los métodos y herramientas que se usan en el proceso de desarrollo de software.	<ul style="list-style-type: none"> ➤ Métodos para el desarrollo de software: <ul style="list-style-type: none"> ▪ Programación orientada a objetos (P.O.O) ▪ Programación procedimental (P.P) ➤ Herramientas para el desarrollo de software. 	<ol style="list-style-type: none"> 1. Detallar métodos para el desarrollo de software. 2. Definir las herramientas necesarias para el desarrollo de software. 	<ol style="list-style-type: none"> a. Identificar los métodos y herramientas para el desarrollo de software. [1, 2] 	Mencionar los métodos y herramientas útiles en el proceso de desarrollo de Software.
Comprender un lenguaje gráfico cómo herramienta para visualizar, especificar, construir y documentar un desarrollo de software.	<ul style="list-style-type: none"> ➤ Definición de UML ➤ Clasificación de los 13 diagramas UML en sus 3 grandes grupos: <ul style="list-style-type: none"> ▪ Diagramas de estructura ▪ Diagramas de comportamiento ▪ Diagramas de interacción ➤ Énfasis de cada uno de los grupos de diagramas UML ➤ Objetivo de cada uno de los diagramas UML <ul style="list-style-type: none"> ▪ Características de cada uno de los diagramas UML 	<ol style="list-style-type: none"> 1. Diferenciar los tres (3) grupos de diagramas UML: de Estructura, de Comportamiento y de Interacción. 2. Comprender cada uno de los trece (13) diagramas UML: de clases, de componentes, de objetos, de estructura compuesta, de despliegue, de paquetes, de actividades, de casos de uso, de estados, de secuencia, de comunicación, de tiempos y de vista de interacción. 	<ol style="list-style-type: none"> a. Clasificar cada uno de los trece (13) diagramas UML en el grupo al cual pertenece. [1, 2] b. Elaborar cada uno de los trece (13) diagramas UML. [2] c. Interpretar cada uno de los trece (13) diagramas UML. [2] d. Documentar los diagramas UML. [2] 	Emplear el Lenguaje Unificado de Modelado (UML) para documentar un proyecto de software.
Describir el ciclo de vida de desarrollo de software mediante sus etapas básicas.	<ul style="list-style-type: none"> ➤ Concepto de ciclo de vida del desarrollo de software. ➤ Las cuatro (4) etapas del ciclo de vida del desarrollo de software: <ul style="list-style-type: none"> ▪ Análisis ▪ Diseño 	<ol style="list-style-type: none"> 1. Definir el concepto de ciclo de vida de desarrollo de software. 2. Reconocer las cuatro (4) etapas fundamentales del ciclo de vida de desarrollo de software: Análisis, Diseño, Desarrollo e Implementación. 	<ol style="list-style-type: none"> a. Determinar los atributos del ciclo de vida de desarrollo de software. [1, 2] b. Pormenorizar cada una de las cuatro (4) etapas fundamentales del ciclo de vida de desarrollo de software: Análisis, Diseño, Desarrollo e Implementación. [2] 	Presentar las etapas que sigue el ciclo de vida en cada modelo de desarrollo de software.

	<ul style="list-style-type: none"> ▪ Desarrollo ▪ Implementación 			
<p>Comprender la estructuración de las etapas de cada modelo de desarrollo del software.</p>	<ul style="list-style-type: none"> ➤ Modelos de desarrollo de software: <ul style="list-style-type: none"> ▪ Modelo lineal ▪ Modelo incremental ▪ Modelo prototipos ▪ Modelo en espiral ▪ Modelo Win-Win ▪ Modelo componentes ▪ Metodología RUP 	<ol style="list-style-type: none"> 1. Distinguir los modelos utilizados en el desarrollo de software: Modelos lineal, incremental, de prototipos, en espiral, win - win, de componentes y RUP. 2. Describir las etapas de cada modelo de desarrollo de software. 3. Analizar para los modelos empleados en el desarrollo de software sus ventajas y/o desventajas. 	<ol style="list-style-type: none"> a. Caracterizar cada uno de los modelos utilizados en el desarrollo de software. [1, 2, 3] b. Relacionar las etapas de cada modelo de desarrollo de software con las cuatro (4) etapas fundamentales del ciclo de vida de desarrollo de software. [1, 2, 3] c. Delimitar cada una de las etapas de los diferentes modelos de desarrollo software. [2] d. Sugerir un modelo para el desarrollo de un software que satisfaga sus características particulares. [1, 2, 3] e. Elaborar la documentación de requisitos para un proyecto planteado. [1, 2, 3] 	

ACTIVIDADES DE ENSEÑANZA APRENDIZAJE		VERSIÓN FINAL		PÁGINA 5
PLANEAR PROYECTOS DE PRODUCCIÓN DE SOFTWARE				
PROPÓSITO	CONTENIDO TEMÁTICO	SABER	HACER	ACTIVIDADES
Definir el ámbito del software mediante los aspectos básicos que permitan su delimitación.	<ul style="list-style-type: none"> ➤ Ámbito del software 	<ol style="list-style-type: none"> 1. Conocer el concepto de ámbito del software en proyectos de producción de software. 2. Precisar los aspectos que se deben tener en cuenta para definir el ámbito del software. 	<ol style="list-style-type: none"> a. Delimitar el ámbito del software en proyectos de producción de software. [1, 2] 	Describir los aspectos del ámbito y las actividades necesarias para la planeación de proyectos de producción de software
Comprender las actividades para llevar a cabo la planeación de proyectos de producción de software	<ul style="list-style-type: none"> ➤ Definición de actividades asociadas a la planificación de proyectos. <ul style="list-style-type: none"> ▪ Uso de los recursos <ul style="list-style-type: none"> • Recursos humanos • Recursos tecnológicos • Recursos económicos 	<ol style="list-style-type: none"> 1. Reconocer los recursos a planear en proyectos de producción de software: humanos, económicos, tecnológicos. 2. Precisar las actividades de planificación asociadas a cada recurso en proyectos de producción de software. 	<ol style="list-style-type: none"> d. Describir las características de cada recurso involucrado en la planeación de proyectos de producción de software. [1] e. Identificar las actividades de cada recurso relacionadas con la planeación de proyectos de producción de software. [1] f. Documentar un plan de proyecto apoyado en los conceptos vistos y un ejercicio particular definido. [1, 2] 	
Comprender el modelo COCOMO para la estimación de costos en proyectos de producción de software.	<ul style="list-style-type: none"> ➤ Estimación de costos ➤ Modelo COCOMO <ul style="list-style-type: none"> ▪ Básico ▪ Intermedio ▪ Avanzado 	<ol style="list-style-type: none"> 1. Estudiar la estimación de costos involucrados en proyectos de producción de software. 2. Analizar las diferentes técnicas para la estimación de costos en proyectos de producción de software. 3. Conocer el modelo de estimación de costos COCOMO. 	<ol style="list-style-type: none"> a. Valorar los costos involucrados en proyectos de producción de software. [1, 2] b. Plantear ejemplos sobre el uso del modelo COCOMO en la estimación de costos en proyectos de producción de software. [3] 	Emplear el modelo COCOMO para la estimación de los costos en proyectos de producción de software.

ACTIVIDADES DE ENSEÑANZA APRENDIZAJE		VERSIÓN FINAL		PÁGINA 6
DELIMITAR LA GESTIÓN DE PROYECTOS DE PRODUCCIÓN DE SOFTWARE				
PROPÓSITO	CONTENIDO TEMÁTICO	SABER	HACER	ACTIVIDADES
<p>Analizar el significado de gestionar proyectos de producción de software mediante las actividades involucradas.</p>	<ul style="list-style-type: none"> ➤ Definición de gestión de proyectos de producción de software ➤ Actividades de gestión <ul style="list-style-type: none"> ▪ Gestión de personal ▪ Gestión de producto ▪ Gestión de proceso ▪ Gestión de proyecto 	<ol style="list-style-type: none"> 1. Entender el concepto de gestión de proyectos de producción de software. 2. Especificar las actividades de gestión de proyectos de producción de software. 3. Comprender por medio de las 4P (personal, producto, proceso y proyecto) la gestión integral de proyectos de producción de software. 4. Definir los procesos de gestión de personal en proyectos de producción de software. 5. Precisar procesos de gestión del producto en proyectos de producción de software. 6. Explicar los procesos de gestión del proceso en proyectos de producción de software. 7. Detallar los procesos de gestión del proyecto en proyectos de producción de software. 	<ol style="list-style-type: none"> a. Describir las actividades de gestión de proyectos de producción de software. [1, 2] b. Analizar problemas presentes en la gestión de proyectos de producción de software. [1, 2, 3] c. Identificar la función que desempeña cada una de las 4P (personal, producto, proceso y proyecto) en la gestión de proyectos de producción de software. [3] d. Diferenciar los procesos de gestión de las 4P en proyectos de producción de software. [4, 5, 6, 7] e. Mencionar los puntos de gestión eficiente para cada una de las 4P en proyectos de producción de software. [4, 5, 6, 7] 	<p>Emplear las actividades necesarias para una gestión adecuada de proyectos de producción de software</p>

ACTIVIDADES DE ENSEÑANZA APRENDIZAJE		VERSIÓN FINAL		PÁGINA 7
DEFINIR LOS RIESGOS EN PROYECTOS DE PRODUCCIÓN DE SOFTWARE				
PROPÓSITO	CONTENIDO TEMÁTICO	SABER	HACER	ACTIVIDADES
Conceptualizar un sistema crítico en proyectos de producción de software.	<ul style="list-style-type: none"> ➤ Definición de sistema crítico en proyectos de producción de software. 	1. Estudiar el concepto de sistema crítico en proyectos de producción de software.	a. Identificar un sistema crítico en proyectos de producción de software. [1]	Reconocer un sistema crítico en proyectos de producción de software.
Definir los riesgos presentes en el desarrollo de proyectos de producción de software.	<ul style="list-style-type: none"> ➤ Riesgos del software ➤ Tipos de riesgos: <ul style="list-style-type: none"> ▪ Riesgos genéricos ▪ Riesgos específicos ➤ Clasificación de riesgos: <ul style="list-style-type: none"> ▪ Riesgos del proyecto ▪ Riesgos técnicos ▪ Riesgos del negocio ➤ Supervisión de riesgos 	<ol style="list-style-type: none"> 1. Identificar los riesgos del software dentro del desarrollo de proyectos de producción de software. 2. Caracterizar los riesgos presentes en proyectos de producción de software. 3. Definir parámetros de supervisión de riesgos en proyectos de producción de software. 	<ol style="list-style-type: none"> a. Diagnosticar los riesgos conocidos y predecibles en proyectos de producción de software. [1, 2] b. Diferenciar los riesgos genéricos de todo proyecto de producción de software de los riesgos específicos del proceso y del producto. [1, 2] c. Detallar planes de contingencia para supervisar los riesgos en proyectos de producción de software. [1, 2, 3] 	Identificar los riesgos y determinar fiabilidad y seguridad en proyectos de producción de software
Analizar la fiabilidad y seguridad en proyectos de producción de software.	<ul style="list-style-type: none"> ➤ Seguridad en proyectos de producción de software. ➤ Fiabilidad en proyectos de producción de software. 	<ol style="list-style-type: none"> 1. Definir seguridad en proyectos de producción de software. 2. Analizar la fiabilidad en proyectos de producción de software. 	<ol style="list-style-type: none"> a. Reconocer los parámetros que enmarcan la seguridad en proyectos de producción de software. [1] b. Inspeccionar la fiabilidad en proyectos de producción de software. [2] 	

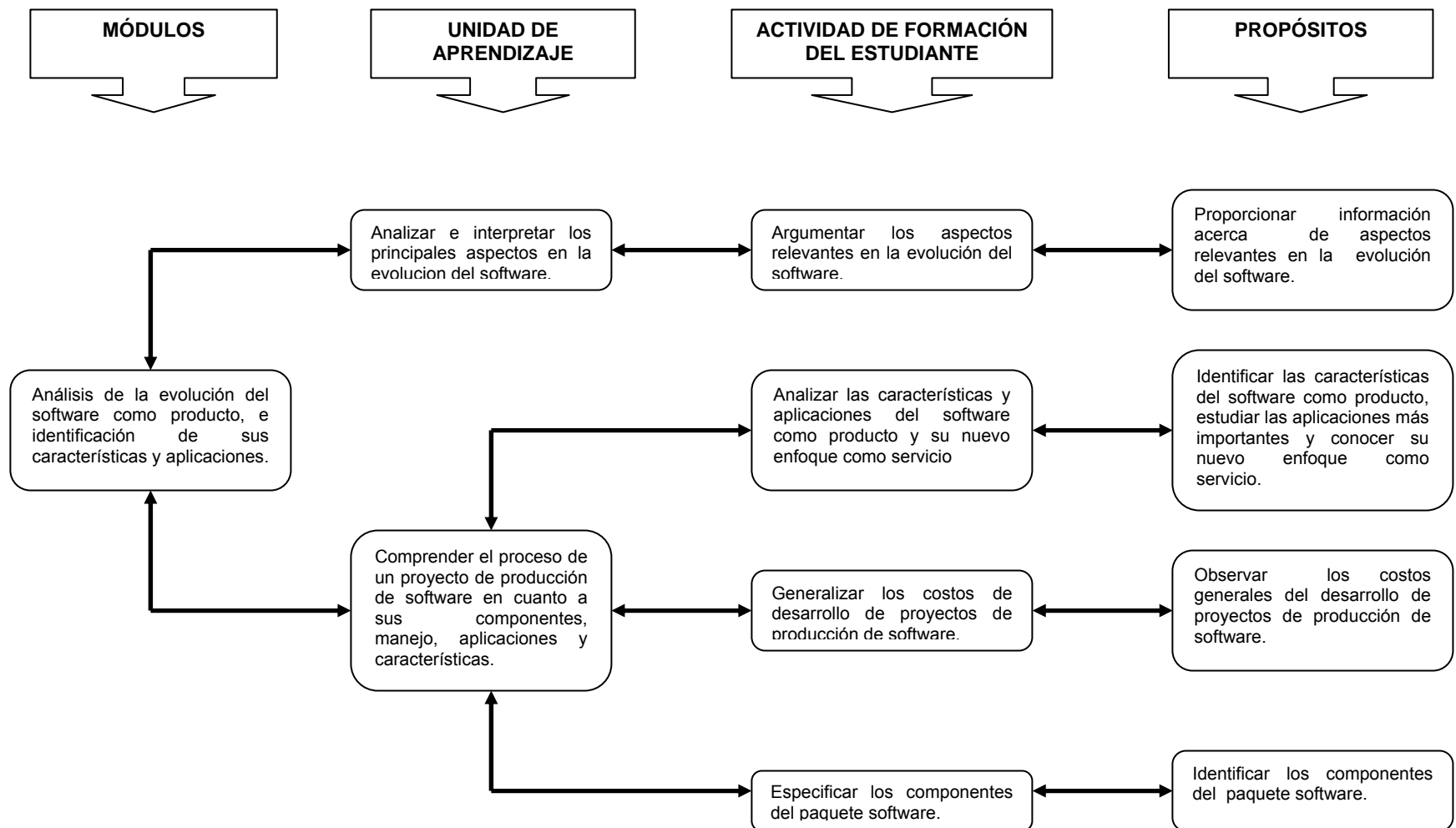
ACTIVIDADES DE ENSEÑANZA APRENDIZAJE		VERSIÓN FINAL		PÁGINA 8
DETERMINAR LA CALIDAD EN PROYECTOS DE PRODUCCIÓN DE SOFTWARE				
PROPÓSITO	CONTENIDO TEMÁTICO	SABER	HACER	ACTIVIDADES
<p>Identificar las actividades necesarias para determinar los requisitos de calidad de proyectos de producción de software.</p>	<ul style="list-style-type: none"> ➤ Definición de calidad del proceso y del producto ➤ Parámetros para determinar la calidad del proceso y del producto ➤ Mediciones sobre el proceso de producción y el producto de software ➤ Administración de los cambios en el proceso de producción de software. 	<ol style="list-style-type: none"> 1. Establecer el concepto de calidad del software. 2. Identificar los factores necesarios para que un proceso de producción de software sea considerado de calidad. 3. Reconocer los factores que determinan la calidad de un producto de software. 4. Plantear las medidas que se deben hacer sobre los factores que determinan la calidad en un proceso de producción de software. 5. Comprender la necesidad de administrar los cambios en los factores que determinan la calidad en un proceso de producción de software. 	<ol style="list-style-type: none"> a. Discutir sobre la importancia de la calidad en proyectos de producción de software. [1] b. Listar los factores que determinan la calidad en un proceso de producción y en un producto software. [2, 3] c. Evaluar cada uno de los factores que determinan la calidad en un proceso de producción y en producto software. [3] d. Efectuar mediciones sobre los factores que determinan la calidad en un proceso de producción de software. [2, 3, 4] e. Analizar las mediciones realizadas sobre los factores que determinan la calidad en un proceso de producción de software. [2, 3, 4] f. Aplicar la administración de los cambios en los factores que determinan la calidad en un proceso de producción de software. [5] 	<p>Ejecutar actividades que permitan establecer los requisitos de calidad para el proceso y producto software.</p>
<p>Examinar los diferentes modelos que permiten controlar la calidad en proyectos de producción de software.</p>	<ul style="list-style-type: none"> ➤ Concepto de Modelo de Calidad del software ➤ Tipos de modelos de gestión de calidad: <ul style="list-style-type: none"> ▪ PEMM ▪ ISO ▪ CMMI 	<ol style="list-style-type: none"> 1. Conocer la definición de modelo de calidad de software. 2. Exponer los modelos de gestión de calidad: PEMM, ISO, CMMI. 	<ol style="list-style-type: none"> a. Reconocer la importancia de adoptar un método de gestión de calidad para el desarrollo de proyectos de producción de software. [1, 2] b. Diferenciar claramente la estructura de cada uno de los modelos de gestión de calidad. [2] 	<p>Reconocer los modelos de gestión de calidad</p>

ACTIVIDADES DE ENSEÑANZA APRENDIZAJE		VERSIÓN FINAL		PÁGINA 9
DEFINIR Y PLANTEAR MÉTRICAS PARA PROYECTOS DE PRODUCCIÓN DE SOFTWARE				
PROPÓSITO	CONTENIDO TEMÁTICO	SABER	HACER	ACTIVIDADES
Definir y analizar los conceptos básicos de métricas en proyectos de producción de software.	<ul style="list-style-type: none"> ➤ Definición de medidas, métricas e indicadores. ➤ Métricas aplicadas al software 	<ol style="list-style-type: none"> 1. Definir métricas, medidas e indicadores en proyectos de producción de software. 2. Establecer las Métricas del software. 3. Conocer las cinco (5) actividades (Formulación, Colección, Análisis, Interpretación y Realimentación) en el proceso de medición en proyectos de producción de software. 	<ol style="list-style-type: none"> a. Diferenciar los conceptos de métricas, medidas e indicadores en proyectos de producción de software. [1] b. Interpretar el concepto de métricas en el contexto de software. [2] c. Caracterizar cada una de las cinco (5) actividades en el proceso de medición en proyectos de producción de software. [3] 	Describir el concepto de métricas y analizar su utilización en el proceso de medición en proyectos de producción de software.
Conocer los diferentes enfoques para aplicar métricas en proyectos de producción de software	<ul style="list-style-type: none"> ➤ Mediciones del software: <ul style="list-style-type: none"> ▪ Métricas orientadas al tamaño ▪ Métricas orientadas a la función 	<ol style="list-style-type: none"> 1. Definir las métricas orientadas al tamaño. 2. Definir las métricas orientadas a la función. 3. Describir las ventajas y desventajas de las métricas orientadas al tamaño y orientadas a la función. 	<ol style="list-style-type: none"> a. Calcular métricas orientadas al tamaño. [1, 3] b. Calcular métricas orientadas a la función. [2, 3] c. Estimar los beneficios de aplicar tanto las métricas orientadas al tamaño como las orientadas a la función. [1, 2, 3] 	
Identificar las entidades y atributos para aplicar métricas en proyectos de producción de software.	<ul style="list-style-type: none"> ➤ Concepto de atributo ➤ Clasificación de los atributos: <ul style="list-style-type: none"> ▪ Atributos internos ▪ Atributos externos ➤ Entidades a medir: <ul style="list-style-type: none"> ▪ Recursos ▪ Productos ▪ Procesos 	<ol style="list-style-type: none"> 1. Conocer el concepto de atributo dentro del contexto de las métricas de software. 2. Definir los atributos internos de un producto software. 3. Detallar los atributos externos de un producto software. 4. Distinguir las entidades que son objeto de medición en proyectos de producción de software a nivel de: recursos, productos y procesos. 	<ol style="list-style-type: none"> a. Clasificar cada uno de los atributos de un producto software en internos y externos. [1, 2, 3] b. Describir cada una de las entidades a medir a nivel de Recursos, Productos y Procesos en proyectos de producción de software. [4] c. Relacionar los atributos internos y externos con las entidades a las cuales pertenecen. [2, 3, 4] 	Señalar los atributos a medir dentro de las entidades establecidas en métricas del software

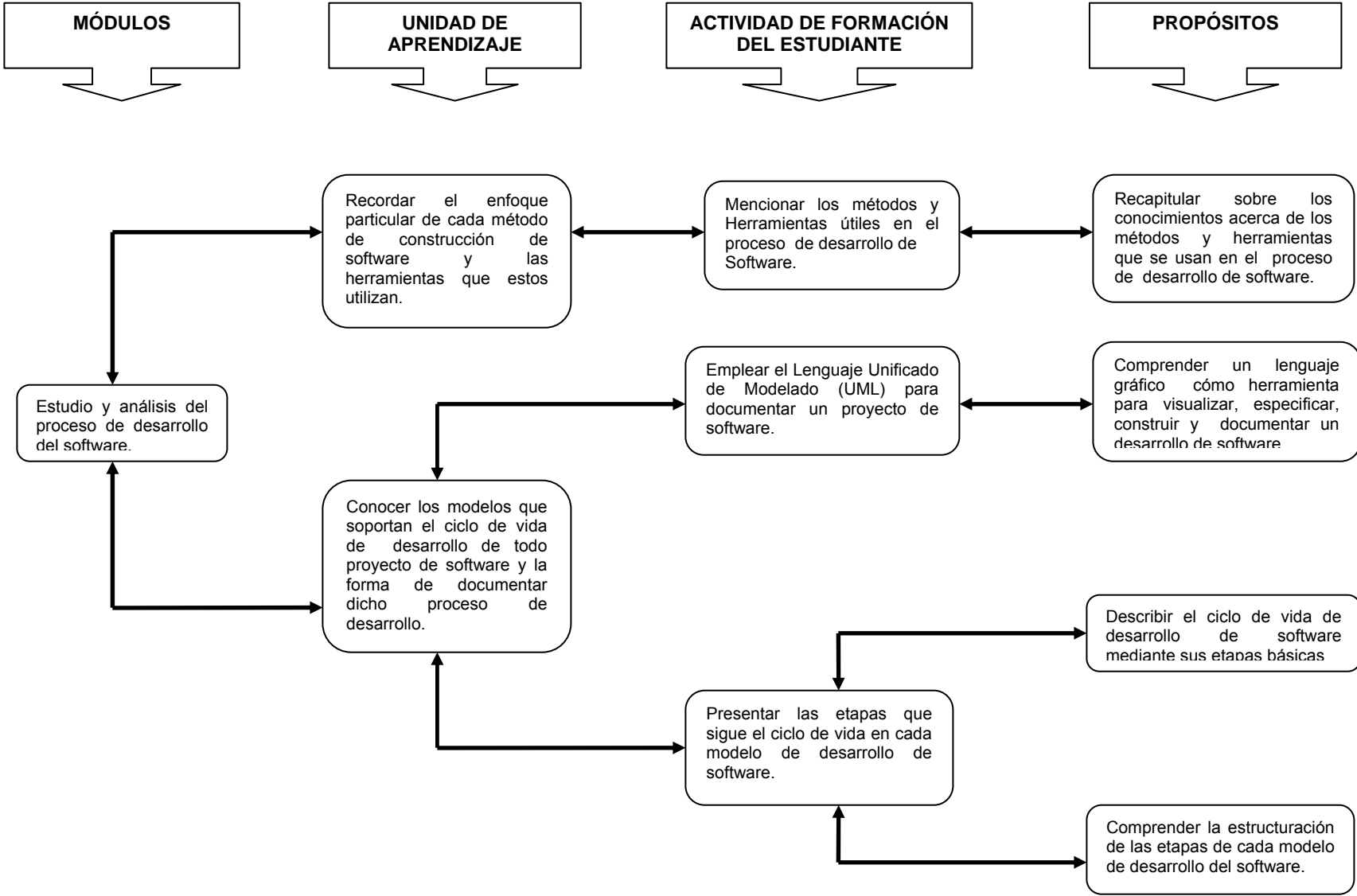
<p>Reconocer la importancia de aplicar métricas a proyectos de producción de software.</p>	<ul style="list-style-type: none"> ➤ Métricas aplicadas a proyectos de producción de software ➤ Definición de métricas de calidad ➤ Recopilación de datos ➤ Calculo de métricas ➤ Evaluación de datos 	<ol style="list-style-type: none"> 1. Entender el concepto de métricas de proyectos de producción de software. 2. Determinar métricas para la calidad en proyectos de producción de software. 3. Conocer los factores de calidad que determinan las métricas de calidad en proyectos de producción de software. 	<ol style="list-style-type: none"> a. Plantear métricas para proyectos de producción de software. [1] b. Deducir los propósitos y objetivos de emplear métricas en proyectos de producción de software. [1] c. Estimar los beneficios de implementar métricas de calidad en proyectos de producción de software. [2] d. Identificar las medidas de calidad necesarias para aplicar métricas en proyectos de producción de software a través de los factores de calidad del software. [3] e. Describir el proceso de evaluación y modelación del software mediante la aplicación de métricas. [1, 2, 3] 	<p>Exponer los beneficios de implementar métricas en proyectos de producción de software</p>
---	--	--	---	---

ACTIVIDADES DE ENSEÑANZA APRENDIZAJE		VERSIÓN 01		PÁGINA 11
GESTIONAR EL PRODUCTO SOFTWARE DESARROLLADO				
PROPÓSITO	CONTENIDO TEMÁTICO	SABER	HACER	ACTIVIDADES
Comprender la gestión de la configuración del producto software y los elementos necesarios para ejecutarla.	<ul style="list-style-type: none"> ➤ Definición de gestión de configuración del producto software ➤ Elementos de la configuración del producto software 	<ol style="list-style-type: none"> 1. Plantear el concepto de Gestión de la Configuración del Software. 2. Conocer los elementos de la configuración del producto software. 	<ol style="list-style-type: none"> a. Evidenciar la necesidad de la gestión de configuración del software. [1] b. Establecer las relaciones que existen entre los diferentes elementos de la configuración del software. [2] 	Entender el proceso de gestión para controlar los cambios en los productos de software
Reconocer el procedimiento para implementar una versión de software	<ul style="list-style-type: none"> ➤ Definición de versión ➤ Manejo de versiones 	<ol style="list-style-type: none"> 1. Identificar los procedimientos necesarios para el manejo de versiones de software. 2. Reconocer los beneficios y dificultades de la implementación de una nueva versión de software. 	<ol style="list-style-type: none"> a. Establecer políticas de implementación de nuevas versiones de software. [1, 2] 	
Examinar el proceso para llevar a cabo cambios en el producto software con base en las herramientas para gestión de la configuración.	<ul style="list-style-type: none"> ➤ Herramientas para la gestión de la configuración del producto software ➤ Administración de los cambios en el producto software. 	<ol style="list-style-type: none"> 1. Señalar algunas herramientas para la gestión de la configuración del software. 2. Identificar los tipos de cambios que puede requerir un producto software a través del tiempo. 3. Determinar procesos para el control de cambios en el software. 	<ol style="list-style-type: none"> a. Definir procesos y herramientas necesarios para realizar la gestión de configuración del producto software. [1, 3] b. Analizar el impacto de la introducción de un cambio en un elemento del software, sobre los demás. [2, 3] c. Plantear un modelo para la administración de cambios en el software. [2, 3] 	

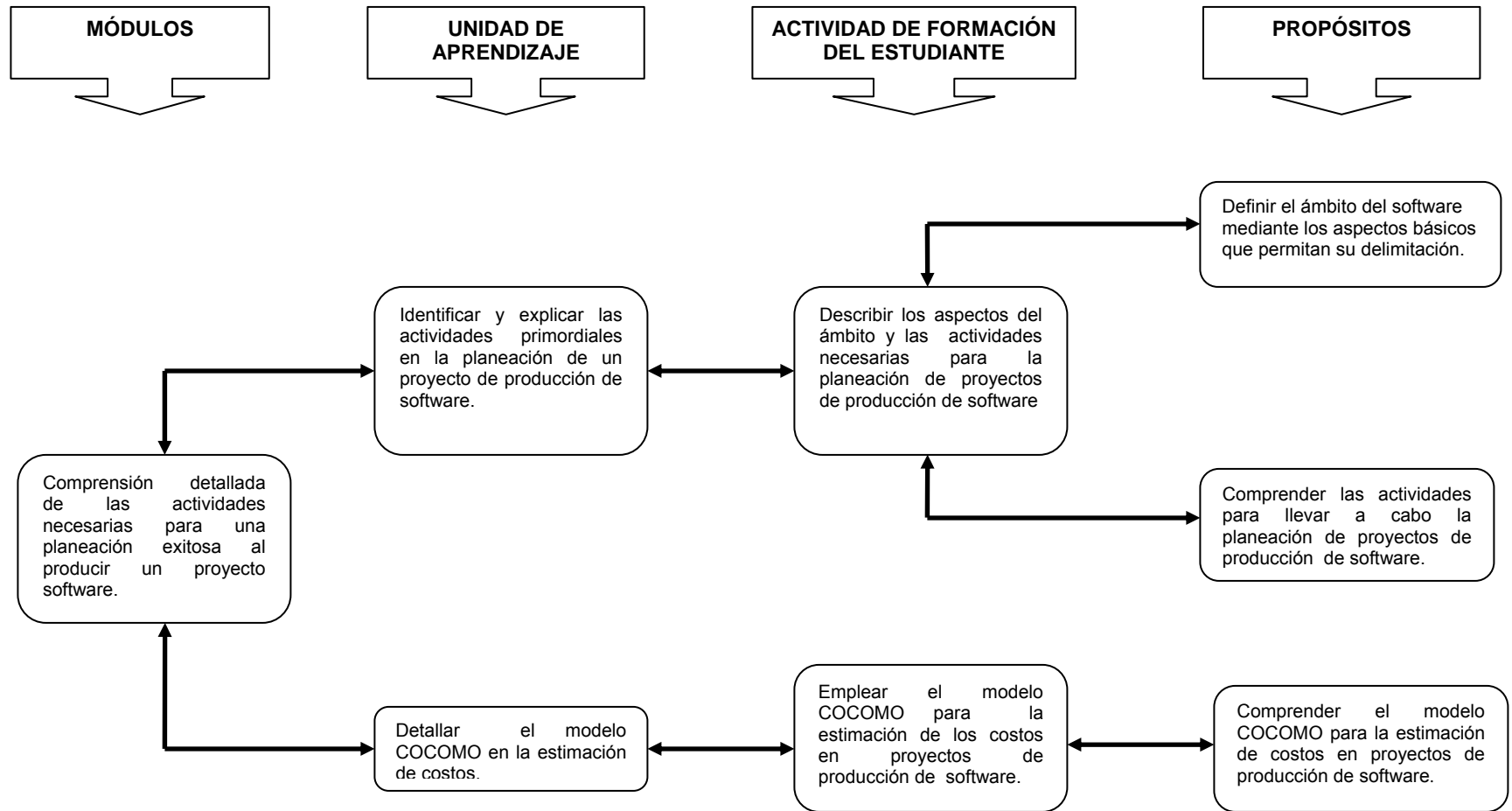
CONTEXTUALIZAR EL SOFTWARE COMO PRODUCTO



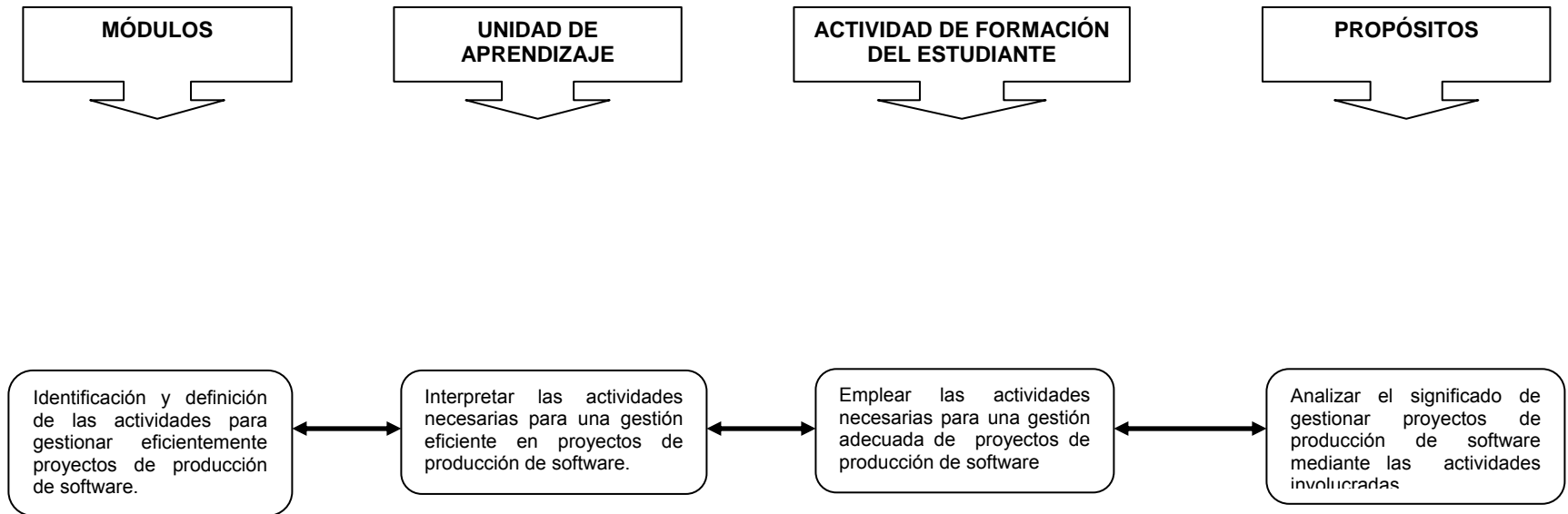
CONOCER EL PROCESO DE DESARROLLO DEL SOFTWARE



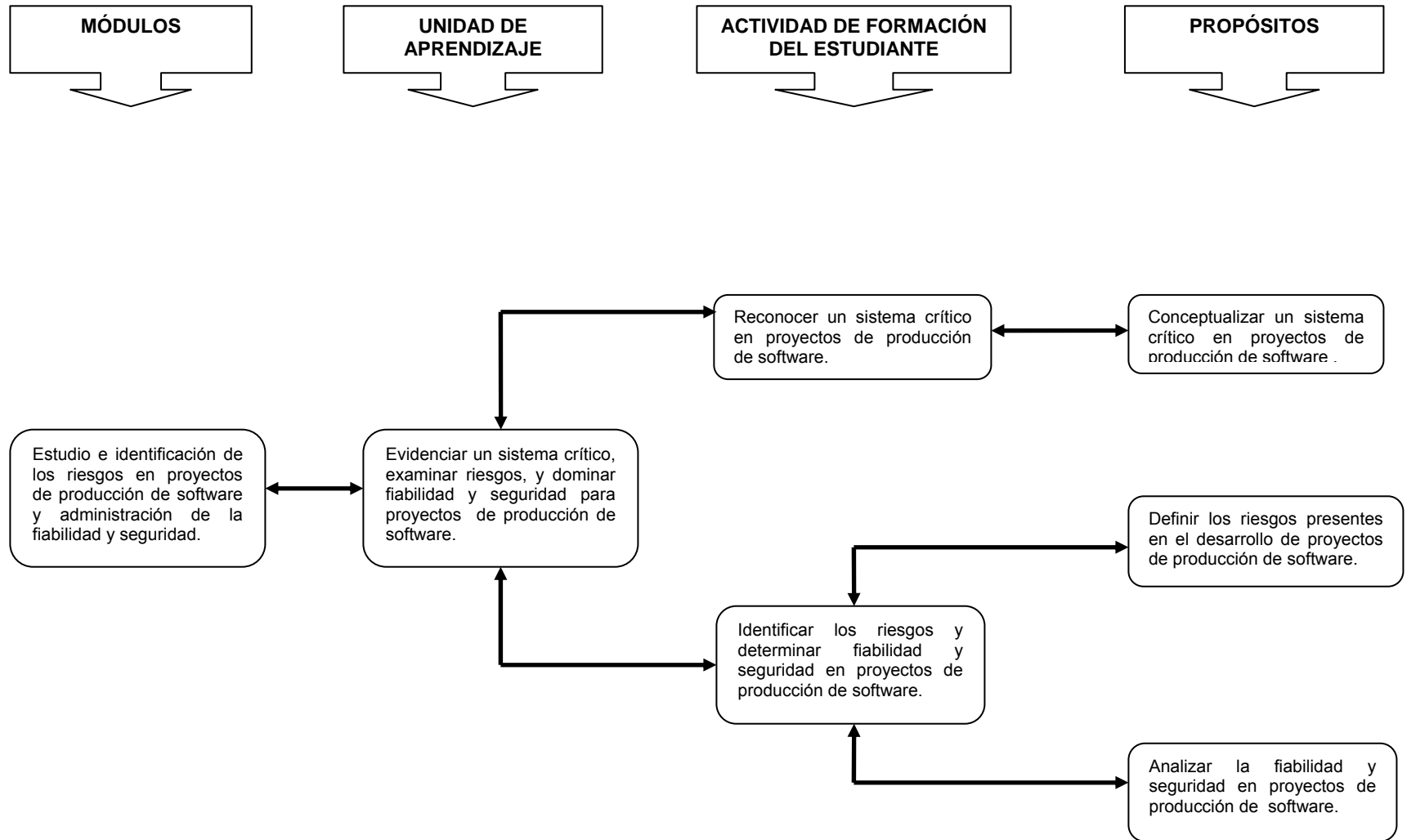
PLANEAR PROYECTOS DE PRODUCCIÓN DE SOFTWARE



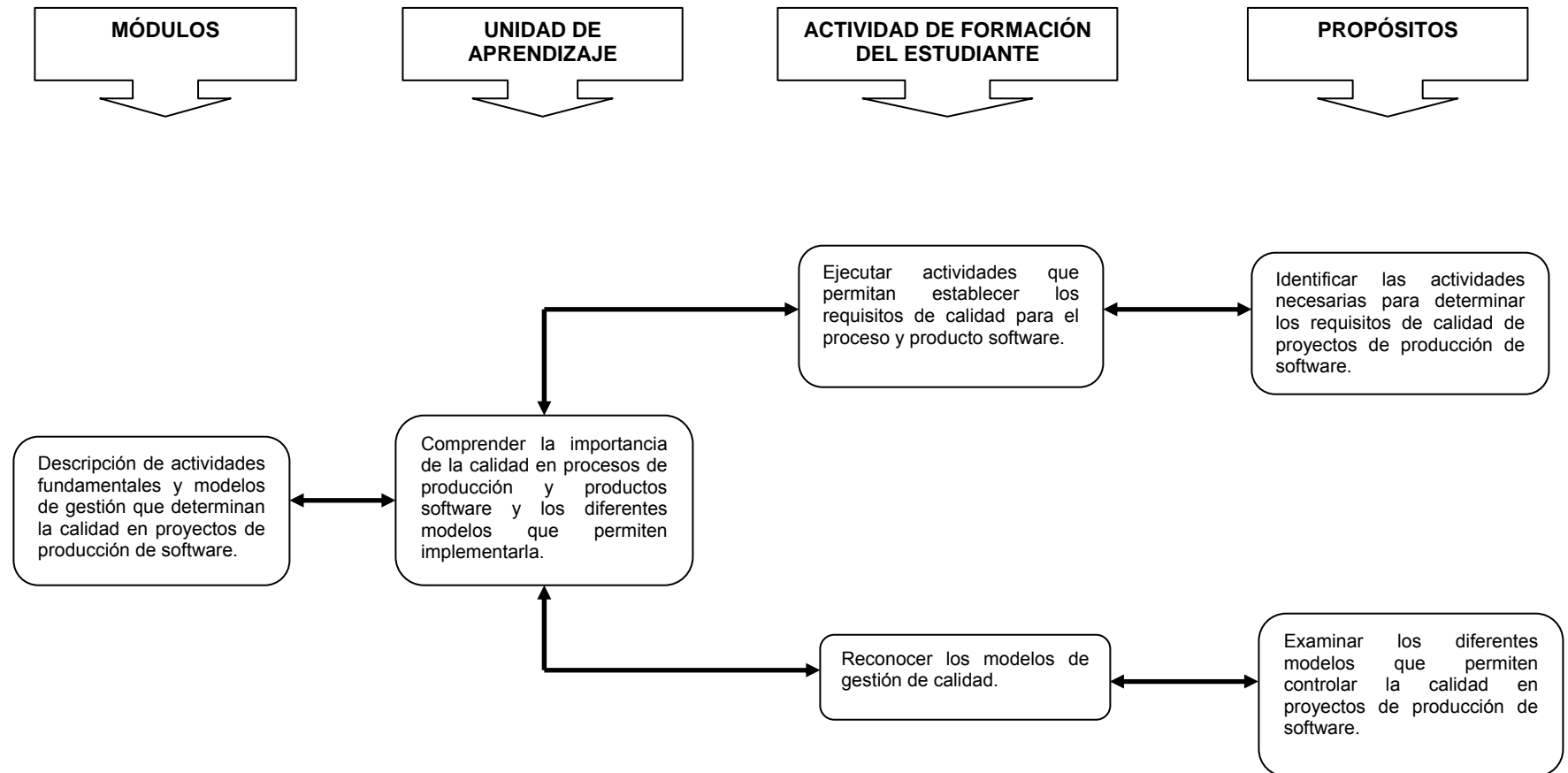
DELIMITAR LA GESTIÓN DE PROYECTOS DE PRODUCCIÓN DE SOFTWARE



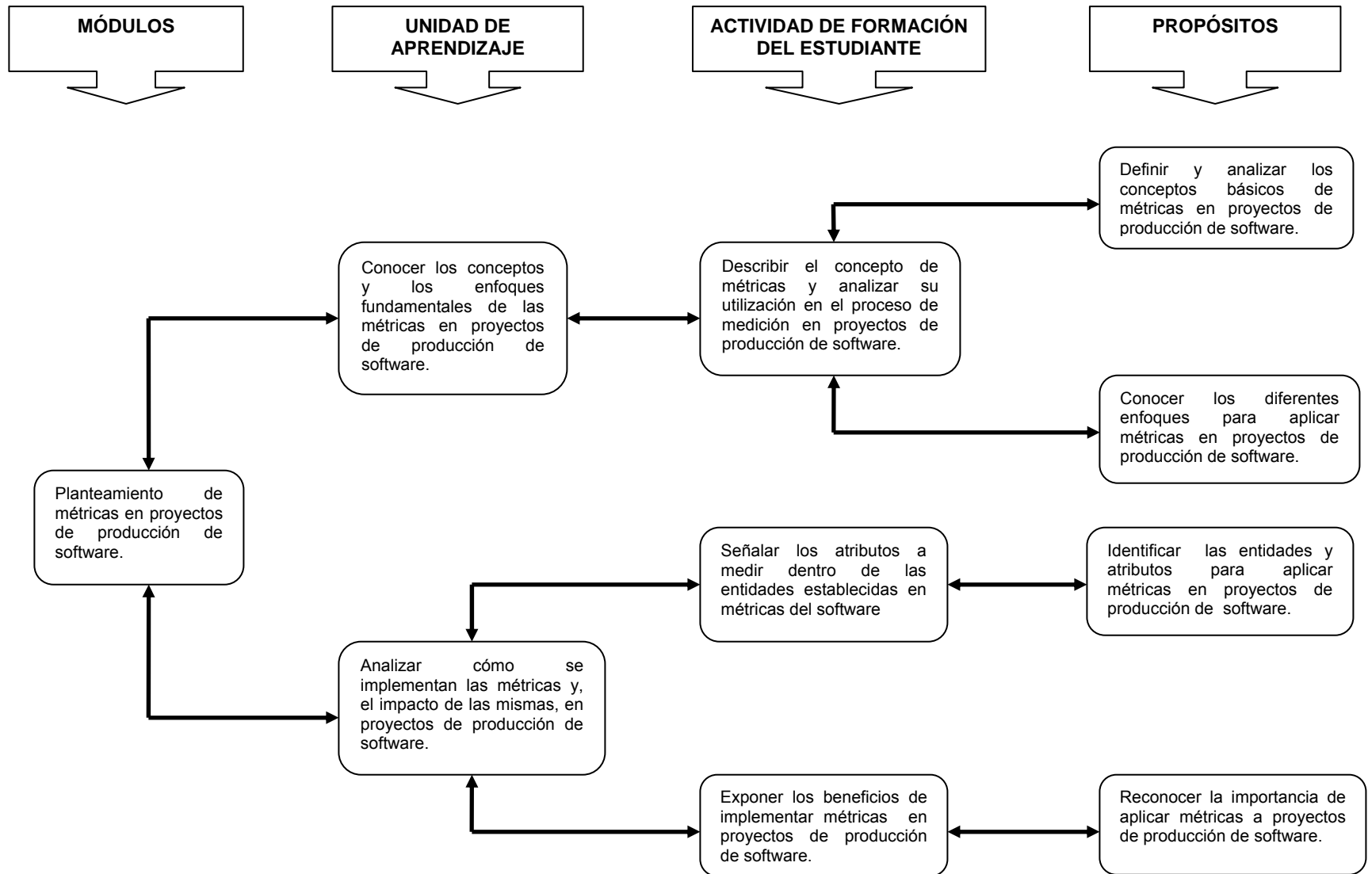
DEFINIR LOS RIESGOS EN PROYECTOS DE PRODUCCIÓN DEL SOFTWARE



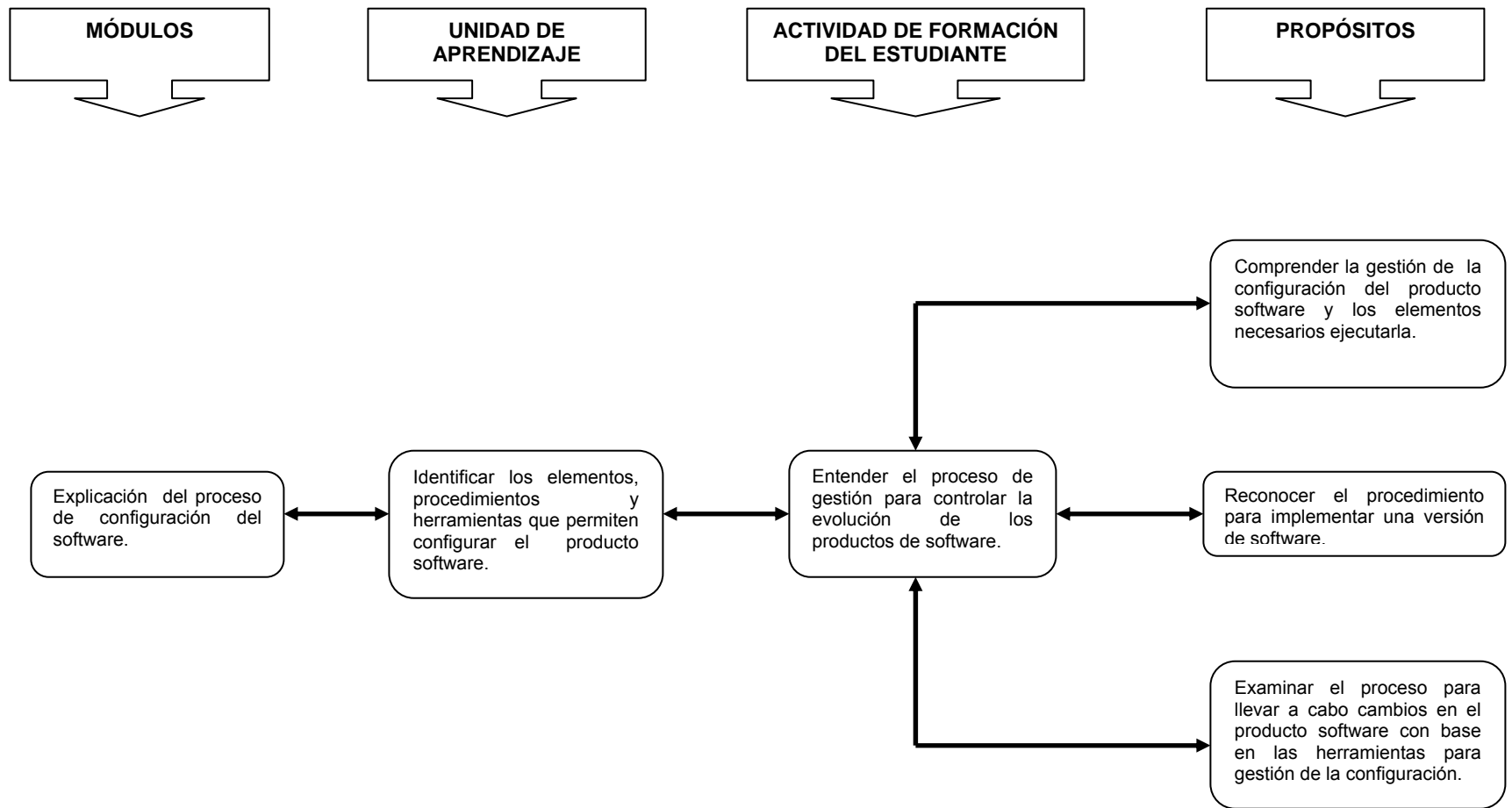
DETERMINAR LA CALIDAD EN PROYECTOS DE PRODUCCIÓN DE SOFTWARE



DEFINIR Y PLANTEAR MÉTRICAS PARA PROYECTOS DE PRODUCCIÓN DE SOFTWARE



GESTIONAR EL PRODUCTO SOFTWARE DESARROLLADO



CONTEXTUALIZAR EL SOFTWARE COMO PRODUCTO	PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Análisis del software como producto, e identificación de sus características y aplicaciones.</i>	
UNIDAD DE APRENDIZAJE	<i>Analizar e interpretar los principales aspectos en la evolución del software.</i>	
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Argumentar los aspectos relevantes en la evolución del software.</i>	

DURACIÓN DE LA ACTIVIDAD				
CRITERIOS	CONTENIDOS		METODOLOGÍA	
	CONCEPTUALES (SABER)	PROCEDIMENTALES (HACERES)	Métodos didácticos	Medios didácticos
Proporcionar información acerca de aspectos relevantes en la evolución del software.	A. Precisar el concepto de software.	F. Describir los sucesos más relevantes de cada una de las etapas en la evolución del software. (A, B)	a. Exposición	Textos: <ul style="list-style-type: none"> • Contenido en formato pdf que suministre información sobre la evolución del software mediante la explicación de los sucesos de cada una de las 5 generaciones del software, detallando el surgimiento de la crisis del software en cada una de las generaciones al igual que los mitos del software detallando sus causas. Gráficos: <ul style="list-style-type: none"> • Ilustración sobre el progreso que ha tenido el software a través de la historia mediante una línea de tiempo con base en las 5 generaciones. • Ilustración explicando mediante un paralelo los mitos del software y la realidad. Animación: <ul style="list-style-type: none"> • Representación de cada una de las etapas en la evolución del software.
	B. Reconocer la evolución del software a través de la historia.	G. Resumir los problemas que dieron origen a la crisis del software. (A, C)	b. Formulación de preguntas	
	C. Comprender el fenómeno de la crisis del software.	H. Argumentar los factores relacionados con la crisis del software. (C)	c. Consulta	
	D. Precisar las alternativas dadas para solucionar los problemas que surgieron al desarrollar software.	I. Concluir las causas por las cuales se presenta insatisfacción sobre el software desarrollado. (C)	d. Resumen	
	E. Reflexionar sobre los mitos del software.	J. Citar ideas para minimizar la vulnerabilidad del software ante la crisis. (C, D)	e. Análisis e interpretación de lectura	
		K. Señalar los mitos más relevantes que existen en la elaboración de productos software. (E)	f. Investigaciones	
		L. Identificar los entes afectados por los mitos en el desarrollo de software. (E)	g. Analogía	

CONTEXTUALIZAR EL SOFTWARE COMO PRODUCTO	PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Análisis del software como producto, e identificación de sus características y aplicaciones.</i>	
UNIDAD DE APRENDIZAJE	<i>Analizar e interpretar los principales aspectos en la evolución del software.</i>	
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Argumentar los aspectos relevantes en la evolución del software.</i>	

EVIDENCIAS DE APRENDIZAJE	EVALUACIÓN	
CONOCIMIENTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Comprende el concepto de software (A)	<ol style="list-style-type: none"> 1. Exposición 2. Prueba o examen 3. Mesa redonda 4. Actividades complementarias 	<ol style="list-style-type: none"> a. Informe (1) b. Resumen (1, 3, 4) c. Cuestionario (2) d. Preguntas informales (1)
Distingue cada etapa de la evolución del software (A, B, F)	<ol style="list-style-type: none"> 1. Exposición 2. Prueba o examen 3. Actividades complementarias 4. Seguimiento de actividades 5. Mapa conceptual 	<ol style="list-style-type: none"> a. Informe (1) b. Resumen (1, 3, 4) c. Cuestionario (2) d. Preguntas informales (1) e. Mapa conceptual (5)
Interpreta el fenómeno de la crisis del software (A, C, D, G, H, J)	<ol style="list-style-type: none"> 1. Exposición 2. Prueba o examen 3. Actividades complementarias 	<ol style="list-style-type: none"> a. Informe (1) b. Resumen (1, 3, 4) c. Cuestionario (2) d. Preguntas informales (1)
Conoce alternativas dadas para solucionar los problemas que surgen al desarrollar software. (D, I)	<ol style="list-style-type: none"> 1. Diagramas de información 2. Actividades complementarias 3. Prueba o examen 	<ol style="list-style-type: none"> a. Esquema (1) b. Resumen (2, 3) c. Ejercicios (3) d. Cuestionario (3) e. Taller de problemas (3)
Identifica los mitos del software. (E, K, L)	<ol style="list-style-type: none"> 1. Mapa conceptual 2. Actividades complementarias 3. Prueba o examen 4. Exposición 	<ol style="list-style-type: none"> a. Mapa conceptual (1) b. Cuestionario (3) c. Resumen (2, 4) d. Test (3)

DESEMPEÑO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Explica los sucesos más relevantes de cada una de las etapas en la evolución del software. (A, B)	<ol style="list-style-type: none"> 1. Mapa conceptual 2. Actividades complementarias 3. Prueba o examen 4. Exposición 5. Ensayo 	<ol style="list-style-type: none"> a. Mapa conceptual (1) b. Resumen (2) c. Relatorías (2) d. Cuestionario (3)
Describe los problemas que dieron origen a la crisis del software. (A, C)	<ol style="list-style-type: none"> 1. Debate 2. Actividades complementarias 3. Prueba o examen 4. Exposición 	<ol style="list-style-type: none"> a. Resumen (1, 2, 4) b. Cuestionario (3) c. Relatorías (2, 4)
Argumenta los factores relacionados con la crisis del software. (C)	<ol style="list-style-type: none"> 1. Mapa conceptual 2. Debate 3. Actividades complementarias 4. Prueba o examen 5. Exposición 	<ol style="list-style-type: none"> a. Mapa conceptual (1) b. Resumen (2, 3, 5) c. Toma de notas (2) d. Cuestionario (4)
Identifica las causas de insatisfacción sobre el software desarrollado (I)	<ol style="list-style-type: none"> 1. Observación 2. Prueba o examen 3. Exposición 	<ol style="list-style-type: none"> a. Lista de verificación (1) b. Test (2) c. Ejercicios (2) d. Preguntas informales (3)
Describe ideas para minimizar la vulnerabilidad del software ante la crisis. (C, G, H)	<ol style="list-style-type: none"> 1. Entrevista 2. Mesa redonda 3. Exposición 4. Prueba o examen 5. Actividades complementarias 	<ol style="list-style-type: none"> a. Cuestionario informal (1, 2, 3) b. Toma de notas (3) c. Ejercicios (4, 5) d. Test (4)
Interpreta los mitos más relevantes que existen en la elaboración de productos software. (E, K, L)	<ol style="list-style-type: none"> 1. Exposición 2. Prueba o examen 3. Actividades complementarias 	<ol style="list-style-type: none"> a. Informe (1) b. Resumen (1, 3) c. Cuestionario (2)
Identifica los entes afectados por los mitos en el desarrollo de software. (L)	<ol style="list-style-type: none"> 1. Entrevista 2. Exposición 3. Actividades complementarias 	<ol style="list-style-type: none"> a. Cuestionario informal (1) b. Resumen (2, 3) c. Preguntas informales (2) d. Relatorías (2, 3)

CONTEXTUALIZAR EL SOFTWARE COMO PRODUCTO	PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Análisis del software como producto, e identificación de sus características y aplicaciones.</i>	
UNIDAD DE APRENDIZAJE	<i>Comprender el proceso de un proyecto software en cuanto a sus componentes, manejo, aplicaciones y características.</i>	
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Analizar las características y aplicaciones del software como producto.</i>	

DURACIÓN DE LA ACTIVIDAD				
CRITERIOS	CONTENIDOS		METODOLOGÍA	
	CONCEPTUALES (SABER)	PROCEDIMENTALES (HACERES)	Métodos didácticos	Medios didácticos
Identificar las características del software como producto, estudiar las aplicaciones más importantes y conocer su nuevo enfoque como servicio.	A. Enunciar las características del software como producto. B. Analizar las aplicaciones del software dentro de sus ámbitos de utilidad. C. Analizar el software en su nueva tendencia como servicio.	D. Detallar las características que hacen la diferencia entre el software y cualquier otro tipo de producto. (A) E. Caracterizar el ámbito en el que se desarrolla el producto Software. (B) F. Precisar el tipo de aplicación software dependiendo del ámbito en el que se desarrolla. (B) G. Citar nuevas posibilidades de ofrecer servicios de soporte sobre el software. (C)	a. Conferencia por un experto b. Consulta c. Resumen d. Tareas individuales e. Análisis e interpretación de lectura f. Investigación g. Lluvia de ideas	Textos: <ul style="list-style-type: none"> • Contenido en formato pdf que exponga las características del software y sus aplicaciones (definición de: software de sistemas, de tiempo real, de gestión, de ingeniería y científico, empotrado, basado en Web y de inteligencia artificial). Gráficos: <ul style="list-style-type: none"> • Ilustración mediante esquemas explicando las características de las principales aplicaciones del software. Video: <ul style="list-style-type: none"> • Sobre aplicaciones del mundo real en empresas.

CONTEXTUALIZAR EL SOFTWARE COMO PRODUCTO	PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Análisis del software como producto, e identificación de sus características y aplicaciones.</i>	
UNIDAD DE APRENDIZAJE	<i>Comprender el proceso de un proyecto software en cuanto a sus componentes, manejo, aplicaciones y características.</i>	
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Analizar las características y aplicaciones del software como producto.</i>	

EVIDENCIAS DE APRENDIZAJE	EVALUACIÓN	
CONOCIMIENTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Distingue las características del Software como producto (A, C, D) .	<ol style="list-style-type: none"> 1. Exposición 2. Prueba o examen 3. Actividades complementarias 4. Seguimiento de actividades 5. Mapa conceptual 	<ol style="list-style-type: none"> a. Informe (1) b. Resumen (1, 3, 4) c. Cuestionario (2) d. Preguntas informales (1) e. Mapa conceptual (5)
Conoce las aplicaciones del Software dentro de sus ámbitos de utilidad (C, E, F) .	<ol style="list-style-type: none"> 1. Mapa conceptual 2. Exposición 3. Ensayo 4. Actividades complementarias 	<ol style="list-style-type: none"> a. Mapa conceptual (1) b. Resumen (2, 4) c. Relatoría (2) d. Ensayo (3)
Explica la nueva tendencia del software como servicio (C, G) .	<ol style="list-style-type: none"> 1. Prueba o examen 2. Exposición 3. Actividades complementarias 	<ol style="list-style-type: none"> a. Cuestionario (1) b. Resumen (2, 3) c. Preguntas informales (2) d. Relatoría (3)
DESEMPEÑO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Identifica las características que hacen la diferencia entre el software y cualquier otro tipo de producto. (A)	<ol style="list-style-type: none"> 1. Prueba o examen 2. Exposición 3. Actividades complementarias 4. Mapa conceptual 	<ol style="list-style-type: none"> a. Cuestionario (1) b. Resumen (2, 3) c. Informe (2) d. Mapa conceptual (4)
Precisa el tipo de aplicación software dependiendo del ámbito en el que se desarrolla. (B, F, E)	<ol style="list-style-type: none"> 1. Diagramas de información 2. Actividades complementarias 3. Prueba o examen 4. Mapa conceptual 	<ol style="list-style-type: none"> a. Tabla (1) b. Mapa conceptual (1, 4) c. Cuestionario (2, 4) d. Test (3)

CONTEXTUALIZAR EL SOFTWARE COMO PRODUCTO	PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Análisis del software como producto, e identificación de sus características y aplicaciones.</i>	
UNIDAD DE APRENDIZAJE	<i>Comprender el proceso de un proyecto de producción de software en cuanto a sus componentes, manejo, aplicaciones y características.</i>	
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Generalizar los costos de desarrollo de proyectos de producción de software.</i>	

DURACIÓN DE LA ACTIVIDAD				
CRITERIOS	CONTENIDOS		METODOLOGÍA	
	CONCEPTUALES (SABER)	PROCEDIMENTALES (HACER)	Métodos didácticos	Medios didácticos
Observar los costos generales del desarrollo de proyectos de producción de software	A. Listar los costos necesarios para el desarrollo de software.	B. Listar los parámetros involucrados en el cálculo del costo total de proyectos de producción de software (A)	a. Formulación de preguntas b. Consulta c. Tareas individuales d. Análisis y resolución de problemas e. Investigación f. Aprendizaje basado en problemas g. Taller de ejercicios h. Presentación participativa i. Debate	Textos: <ul style="list-style-type: none"> Contenido en formato pdf que exponga de forma general los costos implicados en un proyecto de producción de software. (Costos de hardware y software, costos de viajes y capacitación y costos de esfuerzo) Animación: <ul style="list-style-type: none"> Sobre la identificación de los costos del software en un ejemplo en particular.

CONTEXTUALIZAR EL SOFTWARE COMO PRODUCTO	PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Análisis del software como producto, e identificación de sus características y aplicaciones.</i>	
UNIDAD DE APRENDIZAJE	<i>Comprender el proceso de un proyecto de producción de software en cuanto a sus componentes, manejo, aplicaciones y características.</i>	
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Generalizar los costos de desarrollo de proyectos de producción de software.</i>	

EVIDENCIAS DE APRENDIZAJE	EVALUACIÓN	
CONOCIMIENTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Distingue los costos necesarios para el desarrollo de software (A)	1. Mapa conceptual 2. Diagramas de información 3. Prueba o examen 4. Exposición	a. Mapa conceptual (1) b. Esquema (2) c. Test (3) d. Cuestionario (3) e. Resumen (4)
DESEMPEÑO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Explica los parámetros involucrados en el cálculo del costo total de un proyecto de desarrollo de software. (B)	1. Actividades complementarias 2. Diagramas de información 3. Exposición 4. Prueba o examen	a. Esquema (2, 4) b. Resumen (1, 3) c. Informe (3) d. Cuestionario (4)

CONTEXTUALIZAR EL SOFTWARE COMO PRODUCTO	PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Análisis del software como producto, e identificación de sus características y aplicaciones.</i>	
UNIDAD DE APRENDIZAJE	<i>Comprender el proceso de un proyecto de producción de software en cuanto a sus componentes, manejo, aplicaciones y características.</i>	
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Especificar los componentes del paquete software.</i>	

DURACIÓN DE LA ACTIVIDAD				
CRITERIOS	CONTENIDOS		METODOLOGÍA	
	CONCEPTUALES (SABER)	PROCEDIMENTALES (HACERES)	Métodos didácticos	Medios didácticos
Identificar los componentes del paquete software.	A. Especificar los componentes del paquete Software desde la perspectiva del software como producto.	B. Entender cómo está compuesto el paquete software. (A)	<ul style="list-style-type: none"> a. Exposición b. Consulta c. Tareas individuales d. Lluvia de ideas e. Investigación f. Formulación de preguntas g. Debate h. Análisis e interpretación de lectura 	<p>Textos:</p> <ul style="list-style-type: none"> • Contenido en formato pdf acerca de los componentes del paquete software explicando el manejo de licencias y del marco legal para productos software. <p>Gráficos:</p> <ul style="list-style-type: none"> • Bosquejo que explique la labor de cada uno de los componentes dentro del paquete software.

CONTEXTUALIZAR EL SOFTWARE COMO PRODUCTO	PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Análisis del software como producto, e identificación de sus características y aplicaciones.</i>	
UNIDAD DE APRENDIZAJE	<i>Comprender el proceso de un proyecto de producción de software en cuanto a sus componentes, manejo, aplicaciones y características.</i>	
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Especificar los componentes del paquete software.</i>	

EVIDENCIAS DE APRENDIZAJE	EVALUACIÓN	
CONOCIMIENTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Distingue los componentes del paquete Software desde la perspectiva del software como producto. (A)	<ol style="list-style-type: none"> 1. Mapa conceptual 2. Diagramas de información 3. Prueba o examen 4. Exposición 	<ol style="list-style-type: none"> a. Mapa conceptual (1) b. Esquema (2) c. Test (3) d. Resumen (4)
DESEMPEÑO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Describe cómo está compuesto el software (A, B)	<ol style="list-style-type: none"> 1. Prueba o examen 2. Exposición 3. Actividades complementarias 4. Mapa conceptual 	<ol style="list-style-type: none"> a. Cuestionario (1) b. Resumen (2, 3) c. Informe (2) d. Mapa conceptual (4)

CONOCER EL PROCESO DE DESARROLLO DE SOFTWARE	PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Estudio y análisis del proceso de desarrollo del software.</i>	
UNIDAD DE APRENDIZAJE	<i>Recordar el enfoque particular de cada método de construcción de software y las herramientas que estos utilizan.</i>	
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Mencionar los métodos y herramientas útiles en el proceso de desarrollo de software.</i>	

DURACIÓN DE LA ACTIVIDAD			
CRITERIOS	CONOCIMIENTOS		METODOLOGÍA
	CONCEPTUALES (SABER)	PROCEDIMENTALES (HACER)	Métodos didácticos
Recapitular sobre los conocimientos acerca de los métodos y herramientas que se usan en el proceso de desarrollo de software.			Medios didácticos
	A. Detallar métodos para el desarrollo de software. B. Definir las herramientas necesarias para el desarrollo de software.	C. Identificar los métodos y herramientas para el desarrollo de software. (A, B)	a. Consulta b. Formulación de preguntas c. Consulta

CONOCER EL PROCESO DE DESARROLLO DE SOFTWARE	PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Estudio y análisis del proceso de desarrollo del software.</i>	
UNIDAD DE APRENDIZAJE	<i>Recordar el enfoque particular de cada método de construcción de software y las herramientas que estos utilizan.</i>	
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Mencionar los métodos y herramientas útiles en el proceso de desarrollo de software.</i>	

EVIDENCIAS DE APRENDIZAJE	EVALUACIÓN	
CONOCIMIENTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Conoce los métodos para el desarrollo de software. (A, C)	1. Exposición 2. Actividades complementarias	a. Resumen (1) b. Relatorías (1, 2)
Distingue las herramientas para el desarrollo de software. (B, C)		
DESEMPEÑO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Describe los métodos para el desarrollo del software (A, C)	1. Debate 2. Exposición	a. Resumen (1, 2) b. Preguntas informales (2)
Determina las herramientas para el desarrollo del software (B, C)		

CONOCER EL PROCESO DE DESARROLLO DE SOFTWARE		PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Estudio y análisis del proceso de desarrollo del software.</i>		
UNIDAD DE APRENDIZAJE	<i>Conocer los modelos que soportan el ciclo de vida de desarrollo de todo proyecto de software y la forma de documentar dicho proceso de desarrollo.</i>		
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Emplear el Lenguaje Unificado de Modelado (UML) para documentar un proyecto de software.</i>		

DURACIÓN DE LA ACTIVIDAD				
CRITERIOS				
Comprender un lenguaje gráfico como herramienta para visualizar, especificar, construir y documentar un desarrollo de software.	CONCEPTUALES (SABER)	PROCEDIMENTALES (HACER)	Métodos didácticos	Medios didácticos
	A. Diferenciar los tres (3) grupos de diagramas UML: de Estructura, de Comportamiento y de Interacción. B. Comprender cada uno de los trece (13) diagramas UML: de clases, de componentes, de objetos, de estructura compuesta, de despliegue, de paquetes, de actividades, de casos de uso, de estados, de secuencia, de comunicación, de tiempos y de vista de interacción.	C. Clasificar cada uno de los trece (13) diagramas UML en el grupo al cual pertenece (A, B) D. Elaborar cada uno de los trece (13) diagramas UML (B) E. Interpretar cada uno de los trece (13) diagramas UML (B) F. Documentar los diagramas UML. (B)	a. Resumen b. Investigación c. Análisis de ejercicios d. Diagramas e. Análisis y resolución de problemas f. Proyecto	Textos: <ul style="list-style-type: none"> Contenido en formato pdf acerca de los conceptos, los objetivos y la funcionalidad de cada uno de los 13 diagramas UML Gráficos: <ul style="list-style-type: none"> Ilustraciones sobre la estructura de los diagramas UML. Animaciones: <ul style="list-style-type: none"> Sobre la aplicación de los diagramas: casos de uso, clases, secuencia y de estado a un ejemplo en particular. Video: <ul style="list-style-type: none"> Explicando la notación de los 13 diferentes diagramas UML y la aplicación de cada uno dentro del desarrollo de software.

CONOCER EL PROCESO DE DESARROLLO DE SOFTWARE		PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Estudio y análisis del proceso de desarrollo del software.</i>		
UNIDAD DE APRENDIZAJE	<i>Conocer los modelos que soportan el ciclo de vida de desarrollo de todo proyecto de software y la forma de documentar dicho proceso de desarrollo.</i>		
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Emplear el Lenguaje Unificado de Modelado (UML) para documentar un proyecto de software.</i>		

EVIDENCIAS DE APRENDIZAJE	EVALUACIÓN	
CONOCIMIENTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Distingue los tres grupos de diagramas UML (A, C)	1. Exposición 2. Actividades complementarias	a. Resumen (1, 2) b. Preguntas informales (1)
Conoce cada uno de los trece (13) diagramas UML (B, C)	1. Exposición 2. Actividades complementarias 3. Diagramas de información	a. Resumen (1, 2) b. Preguntas informales (1) c. Esquema (3)
DESEMPEÑO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Indica como se clasifican los trece (13) diagramas UML en sus tres (3) grandes grupos. (B, C)	1. Exposición 2. Prueba o examen	a. Lista de verificación (1) b. Informe (1) c. Cuestionario (2) d. Ejercicios (2)
Explica cada uno de los diagramas UML (B, D, E, F)		
PRODUCTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Representa cada uno de los diagramas UML (B, D, E, F)	1. Diagramas de información 2. Actividades complementarias 3. Proyectos	a. Esquema (1) b. Resumen (2) c. Informe (3)

CONOCER EL PROCESO DE DESARROLLO DE SOFTWARE		PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Estudio y análisis del proceso de desarrollo del software.</i>		
UNIDAD DE APRENDIZAJE	<i>Conocer los modelos que soportan el ciclo de vida de todo proyecto de software y la forma de documentar dicho proceso de desarrollo.</i>		
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Presentar las etapas que sigue el ciclo de vida en cada modelo de desarrollo de software</i>		

DURACIÓN DE LA ACTIVIDAD			
CRITERIOS	CONTENIDOS		METODOLOGÍA
	CONCEPTUALES (SABER)	PROCEDIMENTALES (HACER)	Métodos didácticos Medios didácticos
Describir el ciclo de vida de desarrollo de software mediante sus etapas básicas	A. Definir el concepto de ciclo de vida de desarrollo de software. B. Reconocer las cuatro (4) etapas fundamentales del ciclo de vida del software: Análisis, Diseño, Desarrollo e Implementación.	C. Determinar los atributos del ciclo de vida de desarrollo de software. (A, B) D. Pormenorizar cada una de las cuatro (4) etapas fundamentales del ciclo de vida de desarrollo de software: Análisis, Diseño, Desarrollo e Implementación. (B)	a. Consulta b. Análisis e interpretación de lectura c. Exposición d. Investigación e. Formulación de preguntas f. Análisis de casos Textos: <ul style="list-style-type: none"> Contenido en formato pdf sobre el concepto de ciclo de vida de desarrollo de software y de las actividades que se realizan en cada una de las etapas fundamentales (análisis, diseño, desarrollo e implementación) del ciclo de vida de desarrollo de software. Gráficos: <ul style="list-style-type: none"> Diagrama que representa las etapas del ciclo de vida de desarrollo de software. Animaciones: <ul style="list-style-type: none"> Animación mostrando la evolución del software a través de las etapas de su ciclo de vida.

CONOCER EL PROCESO DE DESARROLLO DE SOFTWARE		PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Estudio y análisis del proceso de desarrollo del software.</i>		
UNIDAD DE APRENDIZAJE	<i>Conocer los modelos que soportan el ciclo de vida de desarrollo de todo proyecto de software y la forma de documentar dicho proceso de desarrollo.</i>		
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Presentar cada una de las etapas que sigue cada modelo de ciclo de vida de desarrollo de software.</i>		

EVIDENCIAS DE APRENDIZAJE	EVALUACIÓN	
CONOCIMIENTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Identifica el ciclo de vida del software. (A, B)	1. Prueba o Examen 2. Actividades Complementarias 3. Exposición.	a. Cuestionario (1) b. Resumen (2, 3) c. Preguntas informales (3)
Comprende las etapas básicas del ciclo de vida del software. (B, D)		
DESEMPEÑO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Expone el ciclo de vida del software (A, B)	1. Exposición. 2. Actividades complementarias 3. Prueba o examen	a. Informe (1) b. Resumen (1, 2) c. Test (3)
Establece los atributos del ciclo de vida del software. (A, B)		
Detalla las etapas básicas del ciclo de vida del software. (B, C)	1. Actividades complementarias 2. Diagramas de información 3. Exposición	a. Esquema (2) b. Resumen (1, 3) c. Informe (3)

CONOCER EL PROCESO DE DESARROLLO DE SOFTWARE	PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Estudio y análisis del proceso de desarrollo del software.</i>	
UNIDAD DE APRENDIZAJE	<i>Conocer los modelos que soportan el ciclo de vida de desarrollo de todo proyecto de software y la forma de documentar dicho proceso de desarrollo.</i>	
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Presentar las etapas que sigue el ciclo de vida en cada modelo de desarrollo de software.</i>	

DURACIÓN DE LA ACTIVIDAD				
CRITERIOS	CONTENIDOS		METODOLOGÍA	
	CONCEPTUALES(SABER)	PROCEDIMENTALES(HACER)	Métodos didácticos	Medios didácticos
Comprender la estructuración de las etapas de cada modelo de desarrollo del software.	<p>A. Distinguir los modelos utilizados en el desarrollo de software: Modelos lineal, incremental, de prototipos, en espiral, win - win, de componentes y RUP.</p> <p>B. Describir las etapas de cada modelo de desarrollo de software.</p> <p>C. Analizar para los modelos empleados en el desarrollo de software sus ventajas y/o desventajas.</p>	<p>D. Caracterizar cada uno de los modelos utilizados en el desarrollo de software. (A, B, C)</p> <p>E. Relacionar las etapas de cada modelo de desarrollo de software con las cuatro (4) actividades fundamentales del ciclo de vida de desarrollo de software. (A, B, C)</p> <p>F. Delimitar cada una de las etapas de los diferentes modelos de desarrollo software. (B)</p> <p>G. Sugerir un modelo para el desarrollo de un software que satisfaga sus características particulares. (A, B, C)</p> <p>H. Elaborar la documentación de requisitos para un proyecto planteado. (A, B, C)</p>	<p>a. Exposición</p> <p>b. Consulta</p> <p>c. Resumen</p> <p>d. Diagramas Mapas conceptuales</p> <p>e. Análisis y resolución de problemas</p>	<p>Textos:</p> <ul style="list-style-type: none"> Contenido en formato pdf explicando los modelos de desarrollo del software y las diferencias entre las etapas que cada uno de estos abarca. <p>Gráficos:</p> <ul style="list-style-type: none"> Esquemas que representan las etapas de cada modelo de desarrollo de software (Lineal, incremental, de prototipos, en espiral, win-win, de componentes y RUP). <p>Animaciones:</p> <ul style="list-style-type: none"> Animación sobre la forma en que interactúan las etapas de cada uno de los modelos de desarrollo de software.

CONOCER EL PROCESO DE DESARROLLO DE SOFTWARE	PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Estudio y análisis del proceso de desarrollo del software.</i>	
UNIDAD DE APRENDIZAJE	<i>Conocer los modelos que soportan el ciclo de vida de desarrollo de todo proyecto de software y la forma de documentar dicho proceso de desarrollo.</i>	
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Presentar las etapas que sigue el ciclo de vida en cada modelo de desarrollo de software.</i>	

EVIDENCIAS DE APRENDIZAJE	EVALUACIÓN	
CONOCIMIENTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Interpreta cada uno de los modelos utilizados en el desarrollo de software. (A, D)	<ol style="list-style-type: none"> 1. Prueba o examen 2. Diagramas de información 3. Actividades complementarias 	<ol style="list-style-type: none"> a. Esquema (1, 2) b. Cuestionario (1, 3) c. Test (1)
Reconoce las etapas de cada modelo de desarrollo de software. (B, E, F)	<ol style="list-style-type: none"> 1. Diagramas de información 2. Actividades complementarias 3. Prueba o examen 4. Mapa conceptual 	<ol style="list-style-type: none"> a. Esquema (1, 3) b. Mapa conceptual (1, 4) c. Cuestionario (2, 3) d. Test (3)
Razona sobre las ventajas y desventajas del uso de cada uno de los modelos utilizados en el desarrollo de software. (C, G)	<ol style="list-style-type: none"> 1. Actividades complementarias 2. Exposición 3. Prueba o examen 	<ol style="list-style-type: none"> a. Resumen (1, 2) b. Informe (2) c. Cuestionario (3)
DESEMPEÑO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Detalla cada uno de los modelos utilizados en el desarrollo de software (A, D)	<ol style="list-style-type: none"> 1. Prueba o Examen 2. Actividades Complementarias 3. Exposición. 	<ol style="list-style-type: none"> a. Cuestionario (1) b. Resumen (2, 3) c. Preguntas informales (3)
Asocia las etapas de cada modelo de desarrollo de software con las 4 etapas básicas del ciclo de vida del software. (B, E, F)		
PRODUCTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Ilustra cada uno de los modelos de desarrollo de software describiendo sus etapas (A, F)	<ol style="list-style-type: none"> 1. Diagramas de información 2. Actividades complementarias 3. Prueba o examen 4. Mapa conceptual 	<ol style="list-style-type: none"> a. Esquema (1, 3) b. Mapa conceptual (1, 4) c. Cuestionario (2, 3) d. Test (3)
Propone un modelo para el desarrollo de un software con características particulares (A, B, C, G)	<ol style="list-style-type: none"> 1. Prueba o Examen. 2. Actividades Complementarias. 3. Exposición. 	<ol style="list-style-type: none"> a. Cuestionario(1) b. Taller de Problemas (1, 2) c. Ejercicios (1, 2) d. Exposición (3)
Documenta lo requisitos para un proyecto planteado (A, B, C, H)		

PLANEAR PROYECTOS DE PRODUCCIÓN DE SOFTWARE		PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	Comprensión detallada de las actividades necesarias para una planeación exitosa al producir un proyecto software.		
UNIDAD DE APRENDIZAJE	Identificar y explicar las actividades primordiales en la planeación de un proyecto de producción de software.		
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	Describir los aspectos del ámbito y las actividades necesarias para la planeación de proyectos de producción de software		

DURACIÓN DE LA ACTIVIDAD				
CRITERIOS	CONTENIDOS		METODOLOGÍA	
	CONCEPTUALES (SABER)	PROCEDIMENTALES (HACERES)	Métodos didácticos	Medios didácticos
Definir el ámbito del software mediante los aspectos básicos que permitan su delimitación.	A. Conocer el concepto de ámbito del software en proyectos de producción de software.	C. Delimitar el ámbito del software en proyectos de producción de software. (A, B)	a. Exposición b. Conferencia por un experto c. Consulta d. Resumen e. Análisis e interpretación de lectura f. Lluvia de ideas	Textos: <ul style="list-style-type: none"> • Contenido en formato pdf acerca del concepto de ámbito del software y la información necesaria para determinarlo. Gráficos: <ul style="list-style-type: none"> • Ilustración sobre un ejemplo de ámbito del software. Video: <ul style="list-style-type: none"> • Video dirigido por un experto temático que narre como se hace la identificación del ámbito del software.
	B. Precisar los aspectos que se deben tener en cuenta para definir el ámbito del software.			

PLANEAR PROYECTOS DE PRODUCCIÓN DE SOFTWARE		PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Comprensión detallada de las actividades necesarias para una planeación exitosa al producir un proyecto software.</i>		
UNIDAD DE APRENDIZAJE	<i>Identificar y explicar las actividades primordiales en la planeación de un proyecto de producción de software.</i>		
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Describir los aspectos del ámbito y las actividades necesarias para la planeación de proyectos de producción de software</i>		

EVIDENCIAS DE APRENDIZAJE	EVALUACIÓN	
<i>CONOCIMIENTO</i>	<i>TÉCNICAS DE EVALUACIÓN</i>	<i>INSTRUMENTOS DE EVALUACIÓN</i>
Conoce los aspectos básicos que permiten la delimitación del ámbito del software. (A, B)	<ol style="list-style-type: none"> 1. Diagramas de información 2. Actividades complementarias 3. Prueba o examen 4. Mapa conceptual 	<ol style="list-style-type: none"> a. Tabla (1) b. Mapa conceptual (4) c. Cuestionario (2, 3) d. Test (3)
Define el ámbito del software. (A)	<ol style="list-style-type: none"> 1. Exposición 2. Prueba o examen 3. Actividades complementarias 	<ol style="list-style-type: none"> a. Informe (1) b. Cuestionario (2) c. Resumen (1, 3)
<i>DESEMPEÑO</i>	<i>TÉCNICAS DE EVALUACIÓN</i>	<i>INSTRUMENTOS DE EVALUACIÓN</i>
Identifica los aspectos básicos que delimitan el ámbito del software. (A, B, C)	<ol style="list-style-type: none"> 1. Prueba o examen 2. Exposición 3. Actividades complementarias 4. Mapa conceptual 	<ol style="list-style-type: none"> a. Cuestionario (1) b. Resumen (2, 3) c. Informe (2) d. Mapa conceptual (4)
Identifica el ámbito del software (C)	<ol style="list-style-type: none"> 1. Exposición 2. Prueba o examen 3. Actividades complementarias 	<ol style="list-style-type: none"> a. Informe (1) b. Cuestionario (2) c. Test (2) d. Resumen (1, 3)

PLANEAR PROYECTOS DE PRODUCCIÓN DE SOFTWARE		PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Comprensión detallada de las actividades necesarias para una planeación exitosa al producir un proyecto software.</i>		
UNIDAD DE APRENDIZAJE	<i>Identificar y explicar las actividades primordiales en la planeación de un proyecto de producción de software.</i>		
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Describir los aspectos del ámbito y las actividades necesarias para la planeación de proyectos de producción de software</i>		

DURACIÓN DE LA ACTIVIDAD				
CRITERIOS	CONTENIDOS		METODOLOGÍA	
	CONCEPTUALES (SABER)	PROCEDIMENTALES (HACER)	Métodos didácticos	Medios didácticos
Comprender las actividades para llevar a cabo la planeación de proyectos de producción de software.	A. Reconocer los recursos a planear en proyectos de producción de software: humanos, económicos, tecnológicos.	C. Describir las características de cada recurso involucrado en la planeación de proyectos de producción de software. (A)	a. Exposición b. Conferencia por un experto c. Presentación participativa d. Formulación de preguntas e. Consulta f. Resumen g. Lluvia de ideas h. Proyecto i. Investigación	Textos: •Contenido en formato pdf detallando el conjunto de actividades que globalmente se denominan planificación del proyecto (estimación de trabajo, recursos, tiempo); y definiendo cada uno de los recursos. Gráficos: •Ilustrar los recursos que se planean durante la producción de software y su interrelación (humanos, económicos, tecnológicos). Animación: •Representación animada de las actividades de planificación asociadas con cada recurso.
	B. Precisar las actividades de planificación asociadas a cada recurso en proyectos de producción de software.	D. Identificar las actividades de cada recurso relacionadas con la planeacion de proyectos de producción de software. (A) E. Documentar un plan de proyecto apoyado en los conceptos vistos y un ejercicio particular definido. (A, B)		

PLANEAR PROYECTOS DE PRODUCCIÓN DE SOFTWARE		PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Comprensión detallada de las actividades necesarias para una planeación exitosa al producir un proyecto de software.</i>		
UNIDAD DE APRENDIZAJE	<i>Identificar y explicar las actividades primordiales en la planeación de un proyecto de producción de software.</i>		
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Describir los aspectos del ámbito y las actividades necesarias para la planeación de proyectos de producción de software</i>		

EVIDENCIAS DE APRENDIZAJE	EVALUACIÓN	
CONOCIMIENTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Reconoce los recursos humanos, económicos, tecnológicos (A)	<ol style="list-style-type: none"> 1. Exposición 2. Prueba o examen 3. Mapa conceptual 4. Actividades complementarias 	<ol style="list-style-type: none"> a. Informe (1) b. Cuestionario (2) c. Test (2) d. Resumen (1, 4) e. Mapa conceptual (3)
Precisar las actividades de planificación asociadas a cada recurso. (B)	<ol style="list-style-type: none"> 1. Prueba o examen 2. Exposición 3. Actividades complementarias 4. Mapa conceptual 	<ol style="list-style-type: none"> a. Cuestionario (1) b. Resumen (2, 3) c. Informe (2) d. Mapa conceptual (4)
DESEMPEÑO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Describe las características de cada recurso. (A, C, D)	<ol style="list-style-type: none"> 1. Prueba o examen 2. Exposición 3. Actividades complementarias 4. Mapa conceptual 	<ol style="list-style-type: none"> a. Cuestionario (1) b. Resumen (2, 3) c. Informe (2) d. Mapa conceptual (4)
Identifica las actividades de cada recurso (C)	<ol style="list-style-type: none"> 1. Mapa conceptual 2. Diagramas de información 3. Prueba o examen 4. Exposición 	<ol style="list-style-type: none"> a. Mapa conceptual (1) b. Esquema (2) c. Test (3) d. Cuestionario (3) e. Resumen (4)
Describe las actividades necesarias para llevar a cabo la planeación de un proyecto de software. (C, D)		
PRODUCTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Aplica los conceptos vistos y con base en un ejercicio particular definido documenta un plan de proyecto (A, B, C, D)	<ol style="list-style-type: none"> 1. Prueba o examen 2. Diagramas de información 3. Exposición 4. Actividades complementarias 	<ol style="list-style-type: none"> a. Mapa conceptual (1) b. Esquema (2) c. Resumen (3, 4) d. Ejercicios (1, 4)

PLANEAR PROYECTOS DE PRODUCCIÓN DE SOFTWARE		PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	Comprensión detallada de las actividades necesarias para una planeación exitosa al producir un proyecto software.		
UNIDAD DE APRENDIZAJE	Detallar el modelo COCOMO en la estimación de costos.		
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	Emplear el modelo COCOMO para la estimación de los costos en proyectos de producción de software.		

DURACIÓN DE LA ACTIVIDAD				
CRITERIOS	CONTENIDOS		METODOLOGÍA	
	CONCEPTUALES (SABER)	PROCEDIMENTALES (HACER)	Métodos didácticos	Medios didácticos
Comprender el modelo COCOMO para la estimación de costos en proyectos de producción de software.	A. Estudiar la estimación de costos involucrados en proyectos de producción de software.	D. Valorar los costos involucrados en proyectos de producción de software. (A, B)	a. Consulta b. Resumen c. Tareas individuales d. Exposición e. Debate f. Proyecto g. Análisis de ejercicios	Textos: <ul style="list-style-type: none"> • Contenido en formato pdf que de a conocer el modelo de estimación de costos del software COCOMO. Gráficos: <ul style="list-style-type: none"> • Ilustrar las diferentes técnicas para la estimación de costos en un proyecto de producción de software por medio de diagramas. Animación: <ul style="list-style-type: none"> • Mostrar con animaciones las diferentes técnicas para la estimación de costos en un proyecto de producción de software. Simulador: <ul style="list-style-type: none"> • Que calcule los costos involucrados en el proceso de producción de software con base en el Modelo COCOMO
	B. Analizar las diferentes técnicas para la estimación de costos en proyectos de producción de software.	E. Plantear ejemplos sobre el uso del modelo COCOMO en la estimación de costos en proyectos de producción de software. (C)		
	C. Conocer el modelo de estimación de costos COCOMO.			

PLANEAR PROYECTOS DE PRODUCCIÓN DE SOFTWARE	PLANEACIÓN CURRICULAR	Versión Final
---	-----------------------	---------------

MÓDULO DE FORMACIÓN	<i>Comprensión detallada de las actividades necesarias para una planeación exitosa al producir un proyecto software.</i>
UNIDAD DE APRENDIZAJE	<i>Detallar el modelo COCOMO en la estimación de costos.</i>
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Emplear el modelo COCOMO para la estimación de los costos en proyectos de producción de software.</i>

EVIDENCIAS DE APRENDIZAJE	EVALUACIÓN	
CONOCIMIENTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Explica la estimación de costos involucrados en el proceso de producción de software. (A)	<ol style="list-style-type: none"> 1. Mapa conceptual 2. Diagramas de información 3. Prueba o examen 4. Exposición 	<ol style="list-style-type: none"> a. Mapa conceptual (1) b. Esquema (2) c. Cuestionario (3) d. Resumen (4)
Identifica las diferentes técnicas para la estimación de costos en un proyecto de producción de software. (A, B)	<ol style="list-style-type: none"> 1. Actividades complementarias 2. Diagramas de información 3. Exposición 4. Prueba o examen 	<ol style="list-style-type: none"> a. Esquema (2, 4) b. Resumen (1, 3) c. Informe (3) d. Test (4) e. Cuestionario (4)
Conoce el modelo de estimación de costos COCOMO. (C, E)	<ol style="list-style-type: none"> 1. Exposición 2. Prueba o examen 3. Mapa conceptual 4. Actividades complementarias 	<ol style="list-style-type: none"> a. Informe (1) b. Cuestionario (2) c. Resumen (1, 4) d. Mapa conceptual (3)
DESEMPEÑO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Explica los costos involucrados en el proceso de producción de software. (D)	<ol style="list-style-type: none"> 1. Mapa conceptual 2. Diagramas de información 3. Prueba o examen 4. Exposición 	<ol style="list-style-type: none"> a. Mapa conceptual (1) b. Esquema (2) c. Cuestionario (3) d. Resumen (4)
Plantea ejemplos sobre el uso del modelo COCOMO en la estimación de costos. (C, E)	<ol style="list-style-type: none"> 1. Diagramas de información 2. Actividades complementarias 3. Prueba o examen 4. Mapa conceptual 	<ol style="list-style-type: none"> a. Esquema (1, 3) b. Mapa conceptual (1, 4) c. Cuestionario (2, 3) d. Resumen (2)

DELIMITAR LA GESTIÓN DE PROYECTOS DE PRODUCCIÓN DE SOFTWARE	PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Identificación y definición de las actividades para gestionar eficientemente proyectos de producción de</i>	

	software.
UNIDAD DE APRENDIZAJE	<i>Interpretar las actividades necesarias para una gestión eficiente en proyectos de producción de software.</i>
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Emplear las actividades necesarias para una gestión adecuada de proyectos de producción de software</i>

DURACIÓN DE LA ACTIVIDAD				
CRITERIOS	CONTENIDOS		METODOLOGÍA	
	CONCEPTUALES (SABER)	PROCEDIMENTALES (HACER)	Métodos didácticos	Medios didácticos
Analizar el significado de gestionar proyectos de producción de software mediante las actividades involucradas.	A. Entender el concepto de gestión de proyectos de producción de software. B. Especificar las actividades de gestión de proyectos de producción de software. C. Comprender por medio de las 4P (personal, producto, proceso y proyecto) la gestión integral de proyectos de producción de software. D. Definir los procesos de gestión de personal en proyectos de producción de software. E. Precisar procesos de gestión del producto en proyectos de producción de software. F. Explicar los procesos de gestión del proceso en proyectos de producción de software. G. Detallar los procesos de gestión del proyecto en proyectos de producción de software.	H. Describir las actividades de gestión de proyectos de producción de software. (A, B) I. Analizar problemas presentes en la gestión de proyectos de producción de software. (A, B, C) J. Identificar la función que desempeña cada una de las 4P (personal, producto, proceso y proyecto) en la gestión de proyectos de producción de software. (C) K. Diferenciar los procesos de gestión de las 4P en proyectos de producción de software. (D, E, F, G) L. Mencionar los puntos de gestión eficiente para cada una de las 4P en proyectos de producción de software. (D, E, F, G)	a. Exposición b. Conferencia por un experto c. Consulta d. Tareas individuales e. Resumen f. Análisis e interpretación de lectura g. Investigación h. Mapas conceptuales i. Debate j. analogía k. lluvia de ideas	Textos: <ul style="list-style-type: none"> Contenido en formato pdf sobre los conceptos de gestión de proyectos de producción de software, especificando los recursos: los tipos de personal, de producto y definiendo las actividades de proceso y de proyecto para el mismo. Además en el formato pdf se deben describir las actividades de gestión eficiente de los cuatro (4) recursos. Gráficos: <ul style="list-style-type: none"> Ilustrar las actividades de gestión integral de cada recurso (personal, producto, proceso y proyecto). Ilustrar la relación que existe entre cada recurso por medio de conectores. Animación: <ul style="list-style-type: none"> Mostrar por medio de una animación los puntos de gestión eficiente para cada una de las 4P en proyectos de producción de software.

DELIMITAR LA GESTIÓN DE PROYECTOS DE PRODUCCIÓN DE SOFTWARE	PLANEACIÓN CURRICULAR	Versión Final
--	------------------------------	----------------------

MÓDULO DE FORMACIÓN	<i>Identificación y definición de las actividades para gestionar eficientemente proyectos de producción de software.</i>
UNIDAD DE APRENDIZAJE	<i>Interpretar las actividades necesarias para una gestión eficiente en proyectos de producción de software.</i>
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Emplear las actividades necesarias para una gestión adecuada de proyectos de producción de software</i>

EVIDENCIAS DE APRENDIZAJE	EVALUACIÓN	
CONOCIMIENTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Conoce el concepto de gestión de proyectos de producción de software. (A)	<ol style="list-style-type: none"> 1. Mapa conceptual 2. Exposición 3. Ensayo 4. Actividades complementarias 	<ol style="list-style-type: none"> a. Ensayo (3) b. Mapa conceptual (1) c. Cuestionario (2, 3) d. Resumen (2, 4)
Comprende las actividades de gestión. (B, C, H, J)	<ol style="list-style-type: none"> 1. Prueba o examen 2. Exposición 3. Actividades complementarias 4. Mapa conceptual 	<ol style="list-style-type: none"> a. Cuestionario (1) b. Resumen (2, 3) c. Informe (2) d. Mapa conceptual (4)
Define los procesos de gestión de personal (D)	<ol style="list-style-type: none"> 1. Prueba o examen 2. Exposición 3. Actividades complementarias 	<ol style="list-style-type: none"> a. Cuestionario (1) b. Resumen (2, 3) c. Preguntas informales (2) d. Relatoría (3)
Identifica los procesos de gestión del producto (E)	<ol style="list-style-type: none"> 1. Diagramas de información 2. Actividades complementarias 3. Prueba o examen 4. Mapa conceptual 	<ol style="list-style-type: none"> a. Esquema (1, 3) b. Mapa conceptual (1, 4) c. Cuestionario (2, 3) d. Test (3)
Comprende los procesos de gestión del proceso (F)	<ol style="list-style-type: none"> 1. Exposición 2. Prueba o examen 3. Mapa conceptual 4. Actividades complementarias 	<ol style="list-style-type: none"> a. Informe (1) b. Cuestionario (2) c. Resumen (1, 4) d. Mapa conceptual (3)
Explica las actividades de gestión de proyectos de producción de software. (G)	<ol style="list-style-type: none"> 1. Prueba o examen 2. Exposición 3. Actividades complementarias 	<ol style="list-style-type: none"> a. Cuestionario (1) b. Resumen (2, 3) c. Preguntas informales (2) d. Relatoría (3)
DESEMPEÑO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Identifica problemas presentes en la gestión de proyectos de producción de software. (I)	<ol style="list-style-type: none"> 1. Prueba o examen 2. Exposición 3. Actividades complementarias 4. Mapa conceptual 	<ol style="list-style-type: none"> a. Cuestionario (1) b. Resumen (2, 3) c. Informe (2) d. Mapa conceptual (4)

Explica la función que desempeña el personal, producto, proceso y proyecto en la gestión de proyectos de producción de software. (C, J, K)	<ol style="list-style-type: none"> 1. Diagramas de información 2. Actividades complementarias 3. Prueba o examen 4. Mapa conceptual 	<ol style="list-style-type: none"> a. Tabla (1) b. Mapa conceptual (1, 4) c. Cuestionario (2, 3) d. Test (3)
Diferencia los procesos de gestión de las 4P en proyectos de producción de software. (D, E, K)	<ol style="list-style-type: none"> 1. Diagramas de información 2. Actividades complementarias 3. Prueba o examen 4. Mapa conceptual 	<ol style="list-style-type: none"> a. Esquema (1, 3) b. Mapa conceptual (1, 4) c. Cuestionario (2, 3) d. Test (3)
PRODUCTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Menciona los puntos de gestión eficiente para cada una de las 4P en proyectos de producción de software. (L)	<ol style="list-style-type: none"> 1. Actividades complementarias 2. Prueba o examen 	<ol style="list-style-type: none"> a. Ejercicios (1, 2) b. Taller de problemas (1) c. Problemas (2) d. Test (2)

DEFINIR LOS RIESGOS EN PROYECTOS DE PRODUCCIÓN DE SOFTWARE		PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Estudio e identificación de los riesgos en proyectos de producción de software y administración de la fiabilidad y seguridad.</i>		
UNIDAD DE APRENDIZAJE	<i>Evidenciar un sistema crítico, examinar riesgos, y dominar fiabilidad y seguridad para proyectos de producción de software.</i>		
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Reconocer un sistema crítico en proyectos de producción de software.</i>		

DURACIÓN DE LA ACTIVIDAD				
CRITERIOS	CONTENIDOS		METODOLOGÍA	
	CONCEPTUALES (SABER)	PROCEDIMENTALES (HACER)	Métodos didácticos	Medios didácticos
Conceptualizar un sistema crítico en proyectos de producción de software.	A. Estudiar el concepto de sistema crítico en proyectos de producción de software.	B. Identificar un sistema crítico en proyectos de producción de software. (A)	a. Exposición b. Presentación participativa c. Consulta d. Resumen	Textos: <ul style="list-style-type: none"> • Contenido en formato pdf que de a conocer el concepto y las características de un sistema crítico. Simulador: <ul style="list-style-type: none"> • Simulación que según condiciones dadas del proyecto analice si es o no crítico. Video: <ul style="list-style-type: none"> • Grabación de un experto temático para que narre un caso real de un sistema que haya sido crítico.

DEFINIR LOS RIESGOS EN PROYECTOS DE PRODUCCIÓN DE SOFTWARE		PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Estudio e identificación de los riesgos en proyectos de producción de software y administración de la fiabilidad y seguridad.</i>		
UNIDAD DE APRENDIZAJE	<i>Evidenciar un sistema crítico, examinar riesgos, y dominar fiabilidad y seguridad para proyectos de producción de software.</i>		
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Reconocer un sistema crítico en proyectos de producción de software.</i>		

EVIDENCIAS DE APRENDIZAJE	EVALUACIÓN	
CONOCIMIENTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Conoce el concepto de sistema crítico en proyectos de producción de software. (A)	<ol style="list-style-type: none"> 1. Diagramas de información 2. Actividades complementarias 3. Prueba o examen 4. Mapa conceptual 	<ol style="list-style-type: none"> a. Tabla (1) b. Mapa conceptual (4) c. Cuestionario (2, 3) d. Test (3)
DESEMPEÑO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Identificar un sistema crítico en proyectos de producción de software. (A, B)	<ol style="list-style-type: none"> 1. Diagramas de información 2. Actividades complementarias 3. Prueba o examen 4. Mapa conceptual 	<ol style="list-style-type: none"> a. Esquema (1, 3) b. Mapa conceptual (1, 4) c. Cuestionario (2, 3) d. Test (3)

DEFINIR LOS RIESGOS EN PROYECTOS DE PRODUCCIÓN DE SOFTWARE	PLANEACIÓN CURRICULAR	Versión Final
---	------------------------------	----------------------

MÓDULO DE FORMACIÓN	<i>Estudio e identificación de los riesgos en proyectos de producción de software y administración de la fiabilidad y seguridad.</i>
UNIDAD DE APRENDIZAJE	<i>Evidenciar un sistema crítico, examinar riesgos, y dominar fiabilidad y seguridad para proyectos de producción de software.</i>
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Identificar los riesgos y determinar fiabilidad y seguridad en proyectos de producción de software</i>

DURACIÓN DE LA ACTIVIDAD				
CRITERIOS	CONTENIDOS		METODOLOGÍA	
	CONCEPTUALES (SABER)	PROCEDIMENTALES (HACER)	Métodos didácticos	Medios didácticos
Definir los riesgos presentes en el desarrollo de proyectos de producción de software.	A. Identificar los riesgos del software dentro del desarrollo de proyectos de producción de software.	D. Diagnosticar los riesgos conocidos y predecibles en proyectos de producción de software. (A, B)	a. Presentación participativa	Textos: <ul style="list-style-type: none"> Contenido en formato pdf que exponga y ejemplifique los tipos de riesgos presentes en la producción de software (riesgos genéricos y riesgos específicos) y explique cuales riesgos se pueden clasificar en: riesgos del proyecto, riesgos técnicos y riesgos del negocio y detalle planes de contingencia para dichos riesgos. Gráficos: <ul style="list-style-type: none"> Ilustrar parámetros de supervisión de riesgos en un proyecto de producción de software por medio de la representación grafica de los mismos. Animación: <ul style="list-style-type: none"> Caracterizar los riesgos de un proyecto de producción de software por medio de una animación (riesgos genéricos y riesgos específicos y riesgos del proyecto, riesgos técnicos y riesgos del negocio). Simulador: <ul style="list-style-type: none"> Simulación que genere planes de contingencia dados parámetros de riesgos y seguridad.
	B. Caracterizar los riesgos presentes en proyectos de producción de software.	E. Diferenciar los riesgos genéricos de todo proyecto de producción de software de los riesgos específicos del proceso y del producto. (A, B)	b. Debate	
C. Definir parámetros de supervisión de riesgos en proyectos de producción de software.	F. Detallar planes de contingencia para supervisar los riesgos en proyectos de producción del software. (A, B, C)	d. Consulta	d. Consulta	
		e. Lluvia de ideas	e. Lluvia de ideas	
		f. Analogía	f. Analogía	
		g. investigación	g. investigación	

DEFINIR LOS RIESGOS EN PROYECTOS DE PRODUCCIÓN DE SOFTWARE		PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	Estudio e identificación de los riesgos en proyectos de producción de software y administración de la fiabilidad y seguridad.		
UNIDAD DE APRENDIZAJE	<i>Evidenciar un sistema crítico, examinar riesgos, y dominar fiabilidad y seguridad para proyectos de producción de software.</i>		
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Identificar los riesgos y determinar fiabilidad y seguridad en proyectos de producción de software</i>		

EVIDENCIAS DE APRENDIZAJE	EVALUACIÓN	
CONOCIMIENTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Conoce los riesgos del software dentro del desarrollo de un proyecto. (A)	<ol style="list-style-type: none"> 1. Actividades complementarias 2. Prueba o examen 3. Mapa conceptual 	<ol style="list-style-type: none"> a. Mapa conceptual (1) b. Cuestionario (2, 3) c. Test (3)
Comprende los riesgos de un proyecto de producción de software. (A, B)	<ol style="list-style-type: none"> 1. Prueba o examen 2. Exposición 3. Actividades complementarias 4. Mapa conceptual 	<ol style="list-style-type: none"> a. Cuestionario (1) b. Resumen (2, 3) c. Informe (2) d. Mapa conceptual (4)
Define parámetros de supervisión de riesgos en un proyecto de producción de software. (B, C)	<ol style="list-style-type: none"> 1. Prueba o examen 2. Exposición 3. Actividades complementarias 	<ol style="list-style-type: none"> a. Cuestionario (1) b. Resumen (2, 3) c. Preguntas informales (2) d. Relatoría (3)
DESEMPEÑO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Explica los riesgos conocidos y predecibles (D)	<ol style="list-style-type: none"> 1. Prueba o examen 2. Exposición 3. Actividades complementarias 	<ol style="list-style-type: none"> a. Cuestionario (1) b. Resumen (2, 3) c. Preguntas informales (2) d. Relatoría (3)
Comprende planes de contingencia para supervisar los riesgos del software. (D, E, F)	<ol style="list-style-type: none"> 1. Prueba o examen 2. Exposición 3. Actividades complementarias 4. Mapa conceptual 	<ol style="list-style-type: none"> a. Cuestionario (1) b. Resumen (2, 3) c. Informe (2) d. Mapa conceptual (4)

DEFINIR LOS RIESGOS EN PROYECTOS DE PRODUCCIÓN DE SOFTWARE		PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Estudio e identificación de los riesgos en proyectos de producción de software y administración de la fiabilidad y seguridad.</i>		
UNIDAD DE APRENDIZAJE	<i>Evidenciar un sistema crítico, examinar riesgos, y dominar fiabilidad y seguridad para proyectos de producción de software.</i>		
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Identificar los riesgos y determinar fiabilidad y seguridad en proyectos de producción de software.</i>		

DURACIÓN DE LA ACTIVIDAD				
CRITERIOS	CONTENIDOS		METODOLOGÍA	
	CONCEPTUALES (SABER)	PROCEDIMENTALES (HACER)	Estrategia de enseñanza-aprendizaje	Técnica de enseñanza-aprendizaje
Analizar la fiabilidad y seguridad en proyectos de producción de software.	A. Definir seguridad en proyectos de producción de software.	C. Reconocer los parámetros que enmarcan la seguridad en proyectos de producción de software. (A)	a. Presentación participativa b. Conferencia por un experto c. Exposición d. Consulta e. Debate f. Investigación	Textos: <ul style="list-style-type: none"> • Contenido en formato pdf que explique y argumente en que consiste la seguridad y fiabilidad para un proyecto software. Animación: <ul style="list-style-type: none"> • Presentar parámetros que enmarcan la seguridad en un proyecto de producción de software . Simulador: <ul style="list-style-type: none"> • Simulación que evalúe la fiabilidad de un proyecto dependiendo de parámetros que enmarcan la seguridad.
	B. Analizar la fiabilidad en proyectos de producción de software.	D. Inspeccionar la fiabilidad en proyectos de producción de software. (B)		

DEFINIR LOS RIESGOS EN PROYECTOS DE PRODUCCIÓN DE SOFTWARE	PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Estudio e identificación de los riesgos en proyectos de producción de software y administración de la fiabilidad y seguridad.</i>	
UNIDAD DE APRENDIZAJE	<i>Evidenciar un sistema crítico, examinar riesgos, y dominar fiabilidad y seguridad para proyectos de producción de software.</i>	
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Identificar los riesgos y determinar fiabilidad y seguridad en proyectos de producción de software.</i>	

EVIDENCIAS DE APRENDIZAJE	EVALUACIÓN	
CONOCIMIENTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Conoce el concepto de seguridad en un proyecto de producción de software. (A)	1. Prueba o examen 2. Exposición 3. Actividades complementarias	a. Cuestionario (1) b. Resumen (2, 3) c. Preguntas informales (2) d. Relatoría (3)
Comprende la fiabilidad en un proyecto de producción de software. (B, D)		
DESEMPEÑO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Identifica los parámetros que enmarcan la seguridad en un proyecto de producción de software. (C)	1. Diagramas de información 2. Actividades complementarias 3. Prueba o examen 4. Mapa conceptual	a. Tabla (1) b. Mapa conceptual (1, 4) c. Cuestionario (2, 3) d. Test (3)
Distingue la fiabilidad en un proyecto de producción de software. (B, D)	1. Prueba o examen 2. Exposición 3. Actividades complementarias 4. Mapa conceptual	a. Cuestionario (1) b. Resumen (2, 3) c. Informe (2) d. Mapa conceptual (4)

DETERMINAR LA CALIDAD EN PROYECTOS DE PRODUCCIÓN DE SOFTWARE	PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	Descripción de actividades fundamentales y modelos de gestión que determinan la calidad en proyectos de producción de software.	
UNIDAD DE APRENDIZAJE	Comprender la importancia de la calidad en procesos de producción y productos software y los diferentes modelos que permiten implementarla.	
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	Ejecutar actividades que permitan establecer los requisitos de calidad para el proceso y producto software.	

DURACIÓN DE LA ACTIVIDAD				
CRITERIOS	CONTENIDOS		METODOLOGÍA	
	CONCEPTUALES (SABER)	PROCEDIMENTALES (HACERES)	Métodos didácticos	Medios didácticos
Identificar las actividades necesarias para determinar los requisitos de calidad de proyectos de producción de software.	<p>A. Establecer el concepto de calidad del software.</p> <p>B. Identificar los factores necesarios para que un proceso de producción de software sea considerado de calidad.</p> <p>C. Reconocer los factores que determinan la calidad de un producto de software.</p> <p>D. Plantear las medidas que se deben hacer sobre los factores que determinan la calidad en un proceso de producción de software.</p> <p>E. Comprender la necesidad de administrar los cambios en los factores que determinan la calidad en un proceso de producción de software.</p>	<p>F. Discutir sobre la importancia de la calidad en proyectos de producción de software. (A)</p> <p>G. Listar los factores que determinan la calidad en un proceso de producción y en producto software. (B, C)</p> <p>H. Evaluar cada uno de los factores que determinan la calidad en un proceso de producción y en producto software. (C)</p> <p>I. Efectuar mediciones sobre los factores que determinan la calidad en un proceso de producción de software (B, C, D)</p> <p>J. Analizar las mediciones realizadas sobre los factores que determinan la calidad en un proceso de producción de software. (B, C, D)</p> <p>K. Aplicar la administración de los cambios en los factores que determinan la calidad en un proceso de producción de software. (E)</p>	<p>a. Consulta</p> <p>b. Análisis e interpretación de lectura</p> <p>c. Resumen</p> <p>d. Lluvia de ideas</p> <p>e. Formulación de preguntas</p> <p>f. Debate</p> <p>g. Investigación</p>	<p>Textos:</p> <ul style="list-style-type: none"> • Contenido en formato pdf acerca de los conceptos determinantes de la calidad del software como concepto de calidad, control de calidad, garantía de calidad y costo de la calidad teniendo en cuenta sus componentes y especificando las actividades (como medición y administración) que deben realizarse sobre los factores que determinan la calidad del software <p>Animaciones:</p> <ul style="list-style-type: none"> • Animación sobre el proceso secuencial de las actividades que se deben implementar para determinar la calidad en proyectos de producción de software.

DETERMINAR LA CALIDAD EN PROYECTOS DE PRODUCCIÓN DE SOFTWARE		PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Descripción de actividades fundamentales y modelos de gestión que determinan la calidad en proyectos de producción de software.</i> <i>Comprender la importancia de la calidad en procesos de producción y productos software y los diferentes modelos que permiten implementarla.</i> <i>Ejecutar actividades que permitan establecer los requisitos de calidad para el proceso y producto software.</i>		
UNIDAD DE APRENDIZAJE			
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE			

EVIDENCIAS DE APRENDIZAJE	EVALUACIÓN	
CONOCIMIENTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Reconoce la importancia de la calidad en un proceso de producción de software (A, B)	<ol style="list-style-type: none"> Exposición Actividades complementarias Ensayo 	<ol style="list-style-type: none"> Preguntas informales (1) Resumen (1, 2) Ensayo (2)
Especifica los factores que determinan la calidad en un proceso de producción de software (B, C, G, H)	<ol style="list-style-type: none"> Exposición Prueba o examen Actividades complementarias 	<ol style="list-style-type: none"> Resumen (1, 3) Preguntas informales (1) Cuestionario (2)
Comprende la incidencia de cada uno de los factores que determinan la calidad en un proceso de producción de software, sobre la misma. (B, C, G, H)		
Establece medidas sobre los factores que permitan determinar la calidad en un proceso de producción de software. (D, I, J)	<ol style="list-style-type: none"> Exposición Actividades complementarias 	<ol style="list-style-type: none"> Resumen (1, 2) Ejercicios (2)
Reconoce el proceso de administrar los cambios en los factores que determinan la calidad en un proceso de producción de software. (E, K)	<ol style="list-style-type: none"> Mesa redonda Exposición Actividades complementarias 	<ol style="list-style-type: none"> Cuestionario informal (1) Resumen (2, 3)
DESEMPEÑO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Explica la importancia de la calidad en un proceso de producción de software (A, B)	<ol style="list-style-type: none"> Exposición Debate 	<ol style="list-style-type: none"> Preguntas informales (1) Resumen (1, 2)
Analiza los factores que determinan la calidad en un proceso de producción de software (B, C, G, H)	<ol style="list-style-type: none"> Mesa redonda Prueba o examen Actividades complementarias 	<ol style="list-style-type: none"> Cuestionario informal (1) Test (2) Resumen (1, 3)
Propone medidas sobre los factores que determinan la calidad en un proceso de producción de software (D, I, J)	<ol style="list-style-type: none"> Exposición Prueba o examen Actividades complementarias 	<ol style="list-style-type: none"> Resumen (1, 3) Ejercicios (2, 3)
Aplica medidas sobre los factores que determinan la calidad en un proceso de producción de software (D, I, J)		
Establece la administración de cambios en los factores que determinan la calidad en un proceso de producción de software. (E, K)	<ol style="list-style-type: none"> Mesa redonda Prueba o examen Actividades complementarias 	<ol style="list-style-type: none"> Cuestionario informal (1) Test (2) Resumen (1, 3)

DETERMINAR LA CALIDAD EN PROYECTOS DE PRODUCCIÓN DE SOFTWARE		PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Descripción de actividades fundamentales y modelos de gestión que determinan la calidad en proyectos de producción de software.</i>		
UNIDAD DE APRENDIZAJE	<i>Comprender la importancia de la calidad en procesos de producción y productos software y los diferentes modelos que permiten implementarla.</i>		
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Reconocer los modelos de gestión de calidad.</i>		

DURACIÓN DE LA ACTIVIDAD				
CRITERIOS	CONTENIDOS		METODOLOGÍA	
	CONCEPTUALES (SABER)	PROCEDIMENTALES (HACER)	<i>Métodos didácticos</i>	
			<i>Medios didácticos</i>	
Examinar los diferentes modelos que permiten controlar la calidad en proyectos de producción de software.	<p>A. Conocer la definición de modelo de calidad de software.</p> <p>B. Exponer los modelos de gestión de calidad: PEMM, ISO, CMMI.</p>	<p>C. Reconocer la importancia de adoptar un método de gestión de calidad para el desarrollo de proyectos de producción de software. (A, B)</p> <p>D. Diferenciar claramente la estructura de cada uno de los modelos de gestión de calidad. (B)</p>	<p><i>Métodos didácticos</i></p> <p>a. Consulta</p> <p>b. Análisis e interpretación de lectura</p> <p>c. Resumen</p> <p>d. Investigación</p> <p>e. Formulación de preguntas</p>	<p>Textos:</p> <ul style="list-style-type: none"> • Contenido en formato pdf acerca de los modelos de calidad del software (PEMM, ISO, CMMI) detallando los niveles y normas que soportan la calidad del software dentro de dichos modelos. <p>Gráficos:</p> <ul style="list-style-type: none"> • Ilustración mediante diagramas que permitan establecer la estructura de los diferentes modelos de calidad del software mediante los procesos que estos plantean para gestionar la calidad. <p>Animaciones:</p> <ul style="list-style-type: none"> • Animación que ilustre un paralelo de las diferentes etapas que sigue cada modelo de calidad del software.

DETERMINAR LA CALIDAD EN PROYECTOS DE PRODUCCIÓN DE SOFTWARE		PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Descripción de actividades fundamentales y modelos de gestión que determinan la calidad en proyectos de producción de software.</i>		
UNIDAD DE APRENDIZAJE	<i>Comprender la importancia de la calidad en procesos de producción y productos software y los diferentes modelos que permiten implementarla.</i>		
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Reconocer los modelos de gestión de calidad.</i>		

EVIDENCIAS DE APRENDIZAJE	EVALUACIÓN	
CONOCIMIENTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Comprende el concepto de modelo de calidad del software (A, C)	1. Exposición 2. Actividades complementarias	a. Preguntas informales (1) b. Resumen (1, 2)
Identifica los modelos de gestión de calidad: PEMM, ISO, CMMI (B, D)	1. Diagramas de información 2. Actividades completarias 3. Exposición	a. Cuadro sinóptico (1) b. Esquema (1) c. Resumen (2, 3) d. Preguntas informales (2)
Entiende la importancia de aplicar un modelo de gestión de calidad en un proyecto de producción de software (A, B, C)		
Reconoce la estructura de cada uno de los modelos de gestión de calidad (B, D)		
DESEMPEÑO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Expone el concepto de modelo de calidad del software (A, C)	1. Mesa redonda 2. Prueba o examen 3. Actividades complementarias	a. Cuestionario informal (1) b. Test (2) c. Resumen (1, 3)
Argumenta la importancia de aplicar un modelo de gestión de calidad en un proyecto de producción de software (A, B, C)		
Describe la estructura de cada uno de los modelos de gestión de calidad (B, D)	1. Exposición 2. Prueba o examen 3. Actividades complementarias	a. Preguntas informales (1) b. Test (2) c. Resumen (1, 3)

DEFINIR Y PLANTEAR MÉTRICAS PARA PROYECTOS DE PRODUCCIÓN DE SOFTWARE	PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Planteamiento de métricas en proyectos de producción de software.</i>	
UNIDAD DE APRENDIZAJE	<i>Conocer los conceptos y los enfoques fundamentales de las métricas.</i>	
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Describir el concepto de métricas y analizar su utilización en el proceso de medición en proyectos de producción de software.</i>	

DURACIÓN DE LA ACTIVIDAD			
CRITERIOS	CONTENIDOS		METODOLOGÍA
Definir y analizar los conceptos básicos de métricas.	CONCEPTUALES(SABER)	PROCEDIMENTALES (HACER)	Métodos didácticos
		<p>A. Definir métricas, medidas e indicadores en proyectos de producción de software.</p> <p>B. Establecer las Métricas del software.</p> <p>C. Conocer las cinco (5) actividades (Formulación, Colección, Análisis, Interpretación y Realimentación) en el proceso de medición en proyectos de producción de software.</p>	<p>D. Diferenciar los conceptos de métricas, medidas e indicadores en proyectos de producción de software. (A)</p> <p>E. Interpretar el concepto de métricas en el contexto de software. (B)</p> <p>F. Caracterizar cada una de las cinco (5) actividades en el proceso de medición en proyectos de producción de software. (C)</p>

DEFINIR Y PLANTEAR MÉTRICAS PARA PROYECTOS DE PRODUCCIÓN DE SOFTWARE	PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Planteamiento de métricas en proyectos de producción de software.</i>	
UNIDAD DE APRENDIZAJE	<i>Conocer los conceptos y los enfoques fundamentales de las métricas.</i>	
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Describir el concepto de métricas y analizar su utilización en el proceso de medición en proyectos de producción de software.</i>	

EVIDENCIAS DE APRENDIZAJE	EVALUACIÓN	
CONOCIMIENTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Comprende los conceptos de métricas, medidas e indicadores (A, C)	1. Exposición 2. Mesa redonda	a. Preguntas informales (1) b. Cuestionario informal (2)
Interpreta el concepto de métricas en el contexto de software (B, D)	1. Exposición 2. Prueba o examen 3. Actividades complementarias	a. Resumen (1, 3) b. Cuestionario (2)
Reconoce las 5 actividades básicas que apoyan el proceso de medición del software. (C, E)	1. Exposición 2. Actividades complementarias	a. Lista de verificación (1) b. Preguntas informales (1) c. Resumen (1, 2)
DESEMPEÑO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Explica las diferencias entre métricas, medidas e indicadores (A, C)	1. Exposición 2. Actividades complementarias	a. Resumen (1, 2) b. Preguntas informales (1)
Expone las métricas del software (B, C)	1. Exposición. 2. Actividades complementarias 3. Prueba o examen	a. Informe (1) b. Resumen (1, 2) c. Test (3)
Deduce los objetivos de las métricas del software (B, C)		
Establece las características que debe tener una métrica del software (B, C)	1. Diagramas de información 2. Actividades complementarias 3. Proyectos	a. Esquema (1) b. Resumen (2) c. Informe (3)
Describe cada una de las 5 actividades básicas que apoyan el proceso de medición del software. (C, E)	1. Actividades complementarias 2. Diagramas de información 3. Exposición	a. Esquema (2) b. Resumen (1, 3) c. Informe (3)

DEFINIR Y PLANTEAR MÉTRICAS PARA PROYECTOS DE PRODUCCIÓN DE SOFTWARE	PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Planteamiento de métricas en proyectos de producción de software.</i>	
UNIDAD DE APRENDIZAJE	<i>Conocer los conceptos y los enfoques fundamentales de las métricas.</i>	
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Describir el concepto de métricas y analizar su utilización en el proceso de medición en proyectos de producción de software.</i>	

DURACIÓN DE LA ACTIVIDAD				
CRITERIOS	CONTENIDO		METODOLOGÍA	
	CONCEPTUALES (SABER)	PROCEDIMENTALES (HACER)	Métodos didácticos	Medios didácticos
Conocer los diferentes enfoques para aplicar métricas a proyectos de producción de software	A. Definir las Métricas orientadas al tamaño. B. Definir las Métricas orientadas a la función. C. Describir las ventajas y desventajas de las Métricas orientadas al tamaño y orientadas a la función.	D. Calcular métricas orientadas al tamaño. (A, C) E. Calcular métricas orientadas a la función. (B, C) F. Estimar los beneficios de aplicar tanto las métricas orientadas al tamaño como las orientadas a la función. (A, B, C)	a. Análisis e interpretación de lectura b. Resumen c. Taller de ejercicios d. Simulaciones e. Consulta	Textos: <ul style="list-style-type: none"> Contenido en formato pdf explicando la clasificación de las métricas, específicamente las métricas orientadas al tamaño y a la función y la relación que existe entre las dos. Gráficos: <ul style="list-style-type: none"> Diagrama que represente la clasificación de las métricas. Ilustración sobre el cálculo de las métricas orientadas al tamaño exponiendo su procedimiento. Ilustración sobre el cálculo de las métricas orientadas a la función mostrando su procedimiento. Animaciones: <ul style="list-style-type: none"> Animación que ilustre un ejemplo del cálculo de métricas orientadas a la función. Simulador: <ul style="list-style-type: none"> Simulador que permite realizar el cálculo de métricas orientadas al tamaño y orientadas a la función.

DEFINIR Y PLANTEAR MÉTRICAS PARA PROYECTOS DE PRODUCCIÓN DE SOFTWARE	PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Planteamiento de métricas en proyectos de producción de software.</i>	
UNIDAD DE APRENDIZAJE	<i>Conocer los conceptos y los enfoques fundamentales de las métricas.</i>	
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Describir el concepto de métricas y analizar su utilización en el proceso de medición en proyectos de producción de software.</i>	

EVIDENCIAS DE APRENDIZAJE	EVALUACIÓN	
CONOCIMIENTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Conoce los enfoques para aplicar métricas en un proyecto de producción de software (A, B, D, E)	<ol style="list-style-type: none"> 1. Mesa redonda 2. Prueba o examen 3. Actividades complementarias 	<ol style="list-style-type: none"> a. Cuestionario informal (1) b. Test (2) c. Resumen (1, 3)
Comprende el concepto de métricas orientadas al tamaño (A, D)	<ol style="list-style-type: none"> 1. Exposición 2. Debate 3. Mesa redonda 	<ol style="list-style-type: none"> a. Resumen (1, 2, 3) b. Cuestionario informal (3)
Comprende el concepto de métricas orientadas a la función (B, E)		
Analiza las ventajas y desventajas de las métricas orientadas al tamaño y las métricas orientadas a la función (C, F)	<ol style="list-style-type: none"> 1. Actividades complementarias 2. Exposición 3. Prueba o examen 	<ol style="list-style-type: none"> a. Resumen (1, 2) b. Informe (2) c. Cuestionario (3)
DESEMPEÑO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Describe las métricas orientadas al tamaño (A, D)	<ol style="list-style-type: none"> 1. Exposición 2. Prueba o examen 3. Actividades complementarias 	<ol style="list-style-type: none"> a. Preguntas informales (1) b. Test (2) c. Resumen (1, 3)
Detalla las métricas orientadas a la función (B, E)		
Deduce los beneficios de aplicar tanto las métricas orientadas al tamaño como las orientadas a la función (C, F)	<ol style="list-style-type: none"> 1. Actividades complementarias 2. Exposición 3. Prueba o examen 	<ol style="list-style-type: none"> a. Resumen (1, 2) b. Informe (2) c. Cuestionario (3)
PRODUCTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Calcula métricas orientadas al tamaño (A, D)	<ol style="list-style-type: none"> 1. Prueba o Examen. 2. Actividades Complementarias. 3. Exposición. 	<ol style="list-style-type: none"> a. Cuestionario (1) b. Taller de Problemas (1, 2) c. Ejercicios (1, 2) d. Exposición (3)
Calcula métricas orientadas a la función (B, E)		

DEFINIR Y PLANTEAR MÉTRICAS PARA PROYECTOS DE PRODUCCIÓN DE SOFTWARE	PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Planteamiento de métricas en proyectos de producción de software.</i>	
UNIDAD DE APRENDIZAJE	<i>Analizar cómo se implementan las métricas y el impacto de las mismas en proyectos de producción de software.</i>	
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Señalar los atributos a medir dentro de las entidades establecidas en métricas del software</i>	

DURACIÓN DE LA ACTIVIDAD				
CRITERIOS	CONTENIDO		METODOLOGÍA	
	CONCEPTUALES (SABER)	PROCEDIMENTALES (HACER)	<i>Métodos didácticos</i>	
			<i>Medios didácticos</i>	
Identificar las entidades y atributos para aplicar métricas en proyectos de producción de software.	<p>A. Conocer el concepto de atributo dentro del contexto de las métricas de software.</p> <p>B. Definir los atributos internos de un producto software.</p> <p>C. Detallar los atributos externos de un producto software.</p> <p>D. Distinguir las entidades que son objeto de medición en proyectos de producción de software a nivel de: recursos, productos y procesos.</p>	<p>E. Clasificar cada uno de los atributos de un producto software en internos y externos. (A, B, C)</p> <p>F. Describir cada una de las entidades a medir a nivel de: Recursos, Productos y Procesos en proyectos de producción de software. (D)</p> <p>G. Relacionar los atributos internos y externos con las entidades a las cuales pertenecen (B, C, D)</p>	<p>a. Consulta</p> <p>b. Análisis e interpretación de lectura</p> <p>c. Resumen</p> <p>d. Investigación</p> <p>e. Formulación de preguntas</p>	<p>Textos:</p> <ul style="list-style-type: none"> • Contenido en formato pdf que especifique las entidades y los atributos internos y externos a medir en las métricas del software. <p>Gráficos:</p> <ul style="list-style-type: none"> • Imágenes que representen las entidades a medir en las métricas del software por medio de la forma grafica de las mismas. <p>Animaciones:</p> <ul style="list-style-type: none"> • Animación que ilustre la relación que existe entre atributos y entidades.

DEFINIR Y PLANTEAR MÉTRICAS PARA PROYECTOS DE PRODUCCIÓN DE SOFTWARE	PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Planteamiento de métricas en proyectos de producción de software.</i>	
UNIDAD DE APRENDIZAJE	<i>Analizar cómo se implementan las métricas y el impacto de las mismas en proyectos de producción de software.</i>	
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Señalar los atributos a medir dentro de las entidades establecidas en métricas del software</i>	

EVIDENCIAS DE APRENDIZAJE	EVALUACIÓN	
CONOCIMIENTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Define el concepto de atributo en el contexto de las métricas del software (A, E)	1. Exposición 2. Actividades complementarias	a. Preguntas informales (1) b. Resumen (1, 2)
Conoce el concepto de atributo interno (B, E)	1. Prueba o Examen 2. Actividades Complementarias	a. Cuestionario (1) b. Resumen (2, 3)
Conoce el concepto de atributo externo (C, E)	3. Exposición.	c. Preguntas informales (3)
Determina los recursos, productos y procesos en un proyecto de producción de software. (D, F, G)	1. Debate 2. Exposición	a. Resumen (1, 2) b. Preguntas informales (2)
Reconoce las entidades que son objeto de medición en proyectos de producción de software a nivel de: recursos, productos y procesos. (D, F, G)	1. Actividades complementarias 2. Diagramas de información 3. Exposición	a. Esquema (2) b. Resumen (1,3) c. Informe (3)
DESEMPEÑO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Describe cada una de las entidades a medir a nivel de: Recursos, Productos y Procesos. (D, F)	1. Mesa redonda 2. Prueba o examen 3. Actividades complementarias	a. Cuestionario informal (1) b. Test (2) c. Resumen (1, 3)
Establece los atributos internos que pertenecen a cada una de las entidades a medir (B, C, G)	1. Prueba o Examen 2. Actividades Complementarias	a. Cuestionario (1) b. Resumen (2, 3)
Establece los atributos externos que pertenecen a cada una de las entidades a medir (B, D, G)	3. Exposición.	c. Preguntas informales (3)

DEFINIR Y PLANTEAR MÉTRICAS PARA PROYECTOS DE PRODUCCIÓN DE SOFTWARE		PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Planteamiento de métricas en proyectos de producción de software.</i>		
UNIDAD DE APRENDIZAJE	<i>Analizar cómo se implementan las métricas y el impacto de las mismas en proyectos de producción de software.</i>		
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Exponer los beneficios de implementar métricas en proyectos de producción de software</i>		

DURACIÓN DE LA ACTIVIDAD					
CRITERIOS	CONTENIDOS		METODOLOGÍA		
	CONCEPTUALES (SABER)	PROCEDIMENTALES (HACER)	Métodos didácticos	Medios didácticos	
Reconocer la importancia de aplicar métricas a proyectos de producción de software.	<p>A. Entender el concepto de métricas en proyectos de producción de software.</p> <p>B. Determinar métricas para la calidad en proyectos de producción de software.</p> <p>C. Conocer los factores de calidad que determinan las métricas de calidad en proyectos de producción de software.</p>	<p>D. Plantear métricas para proyectos de producción de software. (A)</p> <p>E. Deducir los propósitos y objetivos de emplear métricas en proyectos de producción de software (A)</p> <p>F. Estimar los beneficios de implementar métricas de calidad en proyectos de producción de software (B)</p> <p>G. Identificar las medidas de calidad necesarias para aplicar métricas en proyectos de producción de software a través de los factores de calidad del software. (C)</p> <p>H. Describir el proceso de evaluación y modelación del software mediante la aplicación de métricas. (A, B, C)</p>	<p>a. Análisis e interpretación de lectura</p> <p>b. Resumen</p> <p>c. Investigación de preguntas</p> <p>d. Formulación de preguntas</p> <p>e. Exposición</p>	<p>Textos:</p> <ul style="list-style-type: none"> Contenido en formato pdf explicando el objetivo de aplicar métricas en proyectos de producción de software, el establecimiento de una línea base para métricas, el proceso de colección, computo y evaluación de las mismas; y las métricas de calidad del software, detallando los factores de calidad propuestos por McCall, Hewlett-Packard y la ISO, las medidas de calidad mas relevantes y la influencia de las métricas de calidad en los proyectos de producción de software. <p>Gráficos:</p> <ul style="list-style-type: none"> Diagrama que representa la incidencia de las métricas de calidad sobre las entidades del software. Diagrama que refleja, a partir de la estimación de la calidad en proyectos de software, cómo surgen las métricas de calidad. <p>Animaciones:</p> <ul style="list-style-type: none"> Animación sobre los factores 	

				de calidad del software propuestos por diferentes autores.
--	--	--	--	--

DEFINIR Y PLANTEAR MÉTRICAS PARA PROYECTOS DE PRODUCCIÓN DE SOFTWARE	PLANEACIÓN CURRICULAR	Versión Final
MÓDULO DE FORMACIÓN	<i>Planteamiento de métricas en proyectos de producción de software.</i>	
UNIDAD DE APRENDIZAJE	<i>Analizar como se implementan las métricas y, el impacto de las mismas, en proyectos de producción de software.</i>	
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Exponer los beneficios de implementar métricas en proyectos de producción de software</i>	

EVIDENCIAS DE APRENDIZAJE	EVALUACIÓN	
CONOCIMIENTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Interpreta el concepto de métricas en el contexto de un proyecto software (A, D, E)	1. Exposición 2. Actividades complementarias	a. Preguntas informales (1) b. Resumen (1, 2)
Reconoce las métricas para la calidad en un proyecto de producción de software. (B, F)	1. Prueba o examen 2. Diagramas de información 3. Actividades complementarias	a. Esquema (1, 2) b. Cuestionario (1, 3) c. Test (1)
Determina los factores de calidad que determinan las métricas en proyectos de producción de software. (C, G, H)	1. Mesa redonda 2. Prueba o examen 3. Actividades complementarias	a. Cuestionario informal (1) b. Test (2) c. Resumen (1, 3)
Analiza los propósitos y objetivos de emplear métricas en un proyecto software (A, B, F)	1. Exposición 2. Debate	a. Preguntas informales (1) b. Resumen (1, 2)
DESEMPEÑO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Detallar el concepto de métricas en el contexto de un proyecto software (A, D, E)	1. Exposición 2. Debate	a. Preguntas informales (1) b. Resumen (1, 2)
Propone métricas para un proyecto software (A, B, D)	1. Exposición 2. Prueba o examen 3. Actividades complementarias	a. Preguntas informales (1) b. Test (2) c. Resumen (1, 3)
Evalúa los beneficios de implementar métricas de calidad en un proyecto de producción de software (A, B, F)	1. Exposición 2. Debate	a. Preguntas informales (1) b. Resumen (1, 2)
Deduce medidas de calidad con base en los factores de calidad del software. (C, G)	1. Prueba o Examen. 2. Actividades Complementarias. 3. Exposición.	a. Cuestionario (1) b. Taller de Problemas (1, 2) c. Ejercicios (1, 2) d. Exposición (3)
Detalla el proceso de evaluación y modelación del software mediante la aplicación de métricas. (C, I)		

GESTIONAR EL PRODUCTO SOFTWARE DESARROLLADO	PLANEACIÓN CURRICULAR	Versión
MÓDULO DE FORMACIÓN	<i>Explicación del proceso de configuración del software</i>	
UNIDAD DE APRENDIZAJE	<i>Identificar los elementos, procedimientos y herramientas que permiten configurar el producto software.</i>	
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Entender el proceso de gestión para controlar la evolución de los productos software.</i>	

DURACIÓN DE LA ACTIVIDAD				
CRITERIOS	CONTENIDOS		METODOLOGÍA	
	CONCEPTUALES (SABER)	PROCEDIMENTALES (HACER)	Métodos didácticos	Medios didácticos
Comprender la gestión de la configuración del producto software y los elementos necesarios para ejecutarla.	A. Plantear el concepto de gestión de la configuración del software.	C. Evidenciar la necesidad de la gestión de la configuración del software. (A, B)	b. Exposición c. Consulta d. Formulación de Preguntas e. Debate	Textos: <ul style="list-style-type: none"> Contenido en formato pdf acerca de la definición de gestión de la configuración del software, de línea base para el control de cambios y exponiendo los elementos de la configuración del software. Gráficos: <ul style="list-style-type: none"> Ilustración que represente las relaciones que existen entre los diferentes elementos de la configuración del software por medio de puntos de conexión. Animación: <ul style="list-style-type: none"> Sobre el los pasos para establecer una línea base con un ejemplo de la vida real (analogía)
	B. Conocer los elementos de la configuración del producto software.	D. Establecer las relaciones que existen entre los diferentes elementos de la configuración del software. (B)		

GESTIONAR EL PRODUCTO SOFTWARE DESARROLLADO		PLANEACIÓN CURRICULAR	Versión
MÓDULO DE FORMACIÓN	<i>Explicación del proceso de configuración del software</i>		
UNIDAD DE APRENDIZAJE	<i>Identificar los elementos, procedimientos y herramientas que permiten configurar el producto software.</i>		
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Entender el proceso de gestión para controlar la evolución de los productos software.</i>		

EVIDENCIAS DE APRENDIZAJE	EVALUACIÓN	
CONOCIMIENTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Comprende la gestión de la configuración del software (A, C)	<ol style="list-style-type: none"> 1. Mesa redonda 2. Prueba o examen 3. Actividades complementarias 	<ol style="list-style-type: none"> a. Cuestionario informal (1) b. Test (2) c. Resumen (1, 3)
Identifica los elementos necesarios para realizar la gestión de la configuración del software (B, D)	<ol style="list-style-type: none"> 1. Exposición 2. Prueba o examen 3. Actividades complementarias 	<ol style="list-style-type: none"> a. Resumen (1, 3) b. Preguntas informales (1) c. Cuestionario (2)
Reconoce la necesidad de implementar los elementos de la configuración del software para gestionar la misma. (A, B, C)		
Conoce la relación que existe entre los diferentes elementos de la configuración del software (B, D)	<ol style="list-style-type: none"> 1. Exposición 2. Prueba o examen 3. Actividades complementarias 	<ol style="list-style-type: none"> a. Preguntas informales (1) b. Test (2) c. Resumen (1, 3)
DESEMPEÑO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Crea su propio concepto sobre la gestión de la configuración del software (A, C)	<ol style="list-style-type: none"> 1. Exposición 2. Debate 	<ol style="list-style-type: none"> a. Preguntas informales (1) b. Resumen (1, 2)
Deduce las ventajas de gestionar la configuración del software (A, C)		
Relaciona los elementos de la configuración del software. (B, D)	<ol style="list-style-type: none"> 1. Exposición 2. Actividades complementarias 	<ol style="list-style-type: none"> a. Preguntas informales (1) b. Resumen (1, 2)

GESTIONAR EL PRODUCTO SOFTWARE DESARROLLADO	PLANEACIÓN CURRICULAR	Versión
MÓDULO DE FORMACIÓN	<i>Explicación del proceso de configuración del software</i>	
UNIDAD DE APRENDIZAJE	<i>Identificar los elementos, procedimientos y herramientas que permiten configurar el producto software.</i>	
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Entender el proceso de gestión para controlar la evolución de los productos software.</i>	

DURACIÓN DE LA ACTIVIDAD				
CRITERIOS	CONTENIDOS		METODOLOGÍA	
	CONCEPTUALES (SABER)	PROCEDIMENTALES (HACER)	Métodos didácticos	Medios didácticos
Reconocer el procedimiento para implementar una versión de software.	<p>A. Identificar los procedimientos necesarios para el manejo de versiones de software.</p> <p>B. Reconocer los beneficios y dificultades de la implementación de una nueva versión de software.</p>	<p>C. Establecer políticas de implementación de nuevas versiones de software. (A, B)</p>	<p>a. Exposición</p> <p>b. Consulta</p> <p>c. Tareas individuales</p> <p>d. Debate</p> <p>e. Análisis e interpretación de lectura</p>	<p>Textos:</p> <ul style="list-style-type: none"> • Contenido en formato pdf acerca del control de versiones, su proceso, ventajas e inconvenientes. <p>Gráficos:</p> <ul style="list-style-type: none"> • Grafo de evolución que represente las diferentes versiones de un sistema software. <p>Animaciones:</p> <ul style="list-style-type: none"> • Animación sobre las relaciones entre componentes, variantes y versiones.

GESTIONAR EL PRODUCTO SOFTWARE DESARROLLADO		PLANEACIÓN CURRICULAR	Versión
MÓDULO DE FORMACIÓN	<i>Explicación del proceso de configuración del software</i>		
UNIDAD DE APRENDIZAJE	<i>Identificar los elementos, procedimientos y herramientas que permiten configurar el producto software.</i>		
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Entender el proceso de gestión para controlar la evolución de los productos software.</i>		

EVIDENCIAS DE APRENDIZAJE	EVALUACIÓN	
<i>CONOCIMIENTO</i>	<i>TÉCNICAS DE EVALUACIÓN</i>	<i>INSTRUMENTOS DE EVALUACIÓN</i>
Conoce los procedimientos para llevar un software de una versión a otra. (A, C)	<ol style="list-style-type: none"> 1. Prueba o Examen 2. Actividades Complementarias 3. Exposición. 	<ol style="list-style-type: none"> a. Cuestionario (1) b. Resumen (2, 3) c. Preguntas informales (3)
Identifica las ventajas e inconvenientes de implementar una nueva versión de software (B)	<ol style="list-style-type: none"> 1. Actividades complementarias 2. Exposición 3. Prueba o examen 	<ol style="list-style-type: none"> a. Resumen (1, 2) b. Informe (2) c. Cuestionario (3)
<i>DESEMPEÑO</i>	<i>TÉCNICAS DE EVALUACIÓN</i>	<i>INSTRUMENTOS DE EVALUACIÓN</i>
Dispone políticas de implementación de nuevas versiones de software (A, B, C)	<ol style="list-style-type: none"> 1. Debate 2. Exposición 	<ol style="list-style-type: none"> a. Resumen (1, 2) b. Preguntas informales (2)

GESTIONAR EL PRODUCTO SOFTWARE DESARROLLADO		PLANEACIÓN CURRICULAR	Versión
MÓDULO DE FORMACIÓN	<i>Explicación del proceso de configuración del software</i>		
UNIDAD DE APRENDIZAJE	<i>Identificar los elementos, procedimientos y herramientas que permiten configurar el producto software.</i>		
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Entender el proceso de gestión para controlar la evolución de los productos software.</i>		

DURACIÓN DE LA ACTIVIDAD					
CRITERIOS	CONTENIDOS		METODOLOGÍA		
	CONCEPTUALES (SABER)	PROCEDIMENTALES (HACER)	Métodos didácticos	Medios didácticos	
Examinar el proceso para llevar a cabo cambios en el producto software con base en las herramientas para la gestión de la configuración.	<p>A. Señalar algunas herramientas para la gestión de la configuración del software.</p> <p>B. Identificar los tipos de cambios que puede requerir un producto software a través del tiempo.</p> <p>C. Determinar procesos para el control de los cambios en el software.</p>	<p>C. Definir procesos y herramientas necesarios para realizar la gestión de configuración del producto software. (A, C)</p> <p>D. Analizar el impacto de la introducción de un cambio en un elemento del software, sobre los demás. (B, C)</p> <p>E. Plantear un modelo para la administración de cambios en el software. (B, C)</p>	<p>a. Exposición</p> <p>b. Presentación participativa</p> <p>c. Consulta</p> <p>d. Resumen</p>	<p>Textos:</p> <ul style="list-style-type: none"> • Contenido en formato pdf sobre las herramientas, tipos de cambios y procesos necesarios para la configuración del software y la realización de informes de estado de la configuración. <p>Gráficos:</p> <ul style="list-style-type: none"> • Ilustración sobre el proceso de control de cambios en el producto software por medio de un diagrama de flujo. • Diagrama de flujo de control de acceso y de sincronización en la configuración del software. <p>Animaciones:</p> <ul style="list-style-type: none"> • Animación sobre como se implementan las herramientas de la configuración del software en la gestión de la misma. • Animación describiendo los procesos para el control de cambios en el software. 	

GESTIONAR EL PRODUCTO SOFTWARE DESARROLLADO		PLANEACIÓN CURRICULAR	Versión
MÓDULO DE FORMACIÓN	<i>Explicación del proceso de configuración del software</i>		
UNIDAD DE APRENDIZAJE	<i>Identificar los elementos, procedimientos y herramientas que permiten configurar el producto software.</i>		
ACTIVIDAD DE ENSEÑANZA-APRENDIZAJE	<i>Entender el proceso de gestión para controlar la evolución de los productos software.</i>		

EVIDENCIAS DE APRENDIZAJE	EVALUACIÓN	
CONOCIMIENTO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Distingue las herramientas para la gestión de la configuración del software (A, C)	<ol style="list-style-type: none"> 1. Exposición. 2. Actividades complementarias 3. Prueba o examen 	<ol style="list-style-type: none"> a. Informe (1) b. Resumen (1, 2) c. Test (3)
Comprende los posibles cambios que puede requerir un software a través del tiempo (B, D)	<ol style="list-style-type: none"> 1. Actividades complementarias 2. Exposición 3. Prueba o examen 	<ol style="list-style-type: none"> a. Resumen (1, 2) b. Informe (2) c. Cuestionario (3)
Establece procesos para controlar cambios en el software (C, E)		
DESEMPEÑO	TÉCNICAS DE EVALUACIÓN	INSTRUMENTOS DE EVALUACIÓN
Determina procesos y herramientas para la gestión de la configuración del software. (A, C)	<ol style="list-style-type: none"> 1. Prueba o Examen 2. Actividades Complementarias 3. Exposición. 	<ol style="list-style-type: none"> a. Cuestionario (1) b. Resumen (2, 3) c. Preguntas informales (3)
Establece el impacto de realizar un cambio a un elemento del software sobre los demás. (B, D)		
Plantea un modelo para la administración de los cambios en el software (C, E)	<ol style="list-style-type: none"> 1. Actividades complementarias 2. Proyectos 	<ol style="list-style-type: none"> a. Esquema (1) b. Resumen (2) c. Informe (3)