

**HERRAMIENTA SOFTWARE PARA LA ENSEÑANZA-
APRENDIZAJE DE LA GEOMETRIA DESCRIPTIVA.
GEDISO 1.0**

**FELIX FERNANDO GALVAN CANCHILA
GIOVANNI LOPEZ MOLINA**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE CIENCIAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA**

2004

**HERRAMIENTA SOFTWARE PARA LA ENSEÑANZA-
APRENDIZAJE DE LA GEOMETRIA DESCRIPTIVA.
GEDISO 1.0**

**FELIX FERNANDO GALVAN CANCHILA
GIOVANNI LOPEZ MOLINA**

**Proyecto de grado para optar al título de
Ingeniera de Sistemas**

**Director
Ing. HUGO HERNANDO ANDRADE SOSA.**

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE CIENCIAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA**

2004

DEDICATORIA

Le agradezco primero a Dios, por darme la fuerza, esperanza y fé necesaria para el desarrollo del proyecto.

Luego a mis padres, por su apoyo emocional, económico de carácter incondicional que me impulsaron a perseverar por encima de mis dificultades, familiares y amigos.

A mi novia Mery, que me dio los ánimos de seguir adelante cada vez que sentí desfallecer.

Félix Fernando Galván Canchila.

DEDICATORIA

Le agradezco a Dios por permitirme desarrollar este proyecto, dándome la oportunidad de demostrar mis capacidades, para la ejecución del mismo.

A mis padres y familiares por darme su apoyo incondicional, quines me dieron la fuerza necesaria para seguir adelante con mi carrera y llevar a buen término este proyecto.

Giovanni López Molina.

AGRADECIMIENTOS

La satisfacción no solo reconocer que se cumplió con la meta propuesta, para comenzar con un nuevo objetivo. Sino en reconocer los obstáculos y superarlos uno por uno. Por eso damos gracias a todos los que me ayudaron a conseguir este triunfo:

A nuestras familias, en especial a nuestros padres, Felix, Omaira, Fabio, Marleny, ya que con su paciencia, entrega y apoyo incondicional hicieron posible la realización de este trabajo de grado.

Al Ing. Hugo Andrade sosa, por sus valiosos aportes, esmero y dedicación. Fue parte fundamental en el desarrollo del proyecto.

A nuestro codirector Edgar Augusto Sarmiento León, por su colaboración en el área de la Geometría Descriptiva, perdiendo el desarrollo del proyecto.

A todo el Grupo de Investigación en Modelos y Simulación (SIMON), por sus sugerencias, críticas constructivas y su apoyo hacia la búsqueda de nuevos retos, especialmente a Jimena, y Emiliano Lince.

A nuestros compañeros de Universidad, por el conocimiento compartido, y amistad brindada en todo momento, especialmente a Juan Pablo, Martha Ligia y Efrén García.

RESUMEN

TITULO* HERRRAMIENTA SOFTWARE PARA LA ENSEÑANZA-APRENDIZAJE DE LA GEOMETRIA DESCRIPTIVA

AUTORES

Galván Canchila Félix Fernando
López Molina Giovanni**

PALABRES CLAVES

Geometría Descriptiva; Software; Enseñanza-Aprendizaje; Proceso Unificado; Editor de Imágenes; intranet.

RESUMEN

GEDIOS0 1.0 Software de Geometría Descriptiva, este proyecto permite llevar a los estudiantes los temas necesarios para la comprensión de la asignatura Geometría Descriptiva, orientando a la práctica estos contenidos, con el desarrollo de ejercicios generales y específicos de acuerdo a la carrera que cursen los diferentes estudiantes. Su objetivo fundamental es formular una propuesta general para desarrollar una herramienta software, que sirva de apoyo al proceso de enseñanza-aprendizaje de la asignatura Geometría Descriptiva en la escuela de Diseño Industrial de la Universidad Industrial de Santander, con el fin de mejorar el dinamismo de la cátedra y facilitar el aprendizaje e implementar un primer prototipo.

Con la ejecución del siguiente proyecto se pretende dar soporte al proceso de enseñanza-aprendizaje de la geometría descriptiva, ya que permite a los alumnos y profesores interactuar con una nueva experiencia donde la tecnología sirve de apoyo a la actividad académica sirviéndole como estímulo y como punto de referencia para la utilización de este tipo de herramientas computacionales utilizadas como apoyo en sus labores cotidianas como profesores y estudiantes.

Como resultado de este proceso se obtuvo GEDISO 1.0 una herramienta software constituida por los módulos de teoría, evaluación, ejercicios y administración donde los estudiantes pueden consultar los contenidos teóricos y desarrollar su habilidad en la geometría descriptiva mediante ejemplos de aplicación y ejercicios propuestos para este fin.

* TRABAJO DE GRADO.

** Facultad de Ingenierías Físico-Mecánicas.
Ingeniería de Sistemas.
Hugo Hernando Andrade Sosa

ABSTRACT

TITLE* SOFTWARE TOOL FOR DESCRIPTIVE GEOMETRY TEACHING-LEARNING.

AUTHORS

Galván Canchila Félix Fernando
López Molina Giovanni**

KEY WORDS

Descriptive Geometry; Software; Teaching-Learning; Unified Process; Image Editor; Intranet.

ABSTRACT

GEDIOSO 1.0 a Descriptive Geometry Software, this Project allows students to handle necessary topics for the comprehension of the Descriptive Geometry subject, taking into practical account the contents, with general and specific exercise development according to each student's coursing career. Its fundamental objective is to formulate a general proposal to develop a software tool, that serves to support the teaching-learning process of the Descriptive Geometry subject in the Industrial Design School of the "Universidad Industrial de Santander", with the finality of improving the class's dynamic activities and facilitate the learning and also to implement a first prototype.

With the fulfillment of the following project we pretend to give support to the teaching-learning process of descriptive Geometry, knowing that it allows students and teacher to interact with a new experience where technology helps as a support to the academic activity as an encouragement and a reference point for the use of this type of computational tool used as a support in their daily chores as students and teachers.

As a result of this process GEDISO 1.0 was obtained, a software tool constituted with theoretical, assessments, exercises and administration modules where students are able to consult their theoretical contents and develop their descriptive geometric ability through application examples and proposed exercises for this finality.

* THESIS PROJECT.

** Physical-Mechanic Engineering Faculty. Systems Engineering.
Hugo Hernando Andrade Sosa

CONTENIDO

INTRODUCCIÓN.....	1
--------------------------	----------

PARTE I – FUNDAMENTOS.....	3
-----------------------------------	----------

CAPITULO 1. ASPECTOS GENERALES.....	4
--	----------

1.1 Objetivos.....	4
1.1.1 Objetivo general.....	4
1.1.2 Objetivo específicos.....	4
1.2 Justificación	5
1.2.1 Antecedentes y descripción del problema.....	6
1.3 Impacto	8
1.4 Descripción del problema	9
1.5 Contenidos generales del documento	10

CAPITULO 2. MARCO TEORICO	13
--	-----------

2.1 Aplicaciones Web	13
2.1.1 Funcionamiento de una aplicación Web	13
2.1.2 Evolución de los sitios Web	14
2.1.2.1 Sitios Web de primera generación	14
2.1.2.2 Sitios Web de segunda generación	14
2.1.2.3 Sitios Web de tercera generación	15
2.1.2.4 Paginas Web estáticas	16
2.1.2.5 Html	16

2.1.3	Tecnologías disponibles para el desarrollo	17
2.1.3.1	Programación del lado del cliente	19
2.1.3.2	Programación del lado del servidor	21
2.2	Bases de datos	26
2.2.1	Mysql	26
2.3	Software de aplicaciones educativas	27
2.4	Soluciones basadas en tecnologías internet	28
2.4.1	Internet	29
2.4.2	Intranet	29
2.4.3	Extranet	31
2.5	Arquitectura del software	32
2.5.1	Cliente-Servidor	33
2.5.2	Múltiples capas	34
2.5.2.1	Capa de datos	34
2.5.2.2	Capa de aplicación o de negocio	35
2.5.2.3	Capa de presentación	35

CAPITULO 3. MARCO METODOLOGICO37

3.1	Ciclo de vida del desarrollo software	38
3.1.1	Cascadas	38
3.1.1.1	Cascada pura	38
3.1.1.2	Cascadas modificadas	39
3.1.2	Desarrollo rápido de aplicaciones	41
3.1.3	Prototipado	42
3.1.3.1	Prototipado simple	43
3.1.3.2	Prototipado evolutivo	44
3.1.4	Entrega por etapas	44
3.1.5	Espiral	46
3.1.6	Proceso unificado	48
3.2	Selección del ciclo de vida	54
3.3	Proceso Unificado	56
3.4	Uml	61

<u>PARTE II – FASE DE INICIO.....</u>	65
Capítulo 4. PLANEACION	66
4.1 Planificación de la fase de inicio	66
4.2 Flujos de trabajo	67
4.2.1 El flujo de trabajo de los requisitos	67
Capítulo 5. VIABILIDAD DEL PROYECTO: UNICA ITERACIÓN	68
5.1 Actividad: Capturar de requisitos candidatos	68
5.2 Actividad: Identificar riesgos críticos	71
5.3 Actividad: Comprender el contexto de la herramienta	73
5.3.1 Descripción del contexto	73
5.3.2 Que es la geometría descriptiva	74
5.3.3 Informática educativa	74
5.3.4 Modelo de caso de uso del negocio	76
5.3.5 Modelo del dominio	78
5.4 Actividad: Representar los requisitos como caso de uso	79
5.4.1 Encontrar actores	79
5.4.2 Encontrar los casos de uso	82
5.5 Actividad: Detallar un caso de uso	87
5.6 Actividad: Estructurar el modelo de caso de uso	90
5.7 Actividad: Capturar requisitos adicionales	91
5.8 Análisis	92
5.8.1 Análisis de la arquitectura	92
5.8.1.1 Paquetes del análisis general	92
5.8.2 Identificación de clases de entidad obvias	97
5.8.3 Análisis de caso de uso	98
5.8.3.1 Identificación de clases del análisis	98

5.9 El flujo de trabajo del diseño	100
5.9.1 Actividad: Diseño de la arquitectura	100
5.9.1.1 Identificación de nodos y configuraciones de red.....	100
5.9.1.2 Identificación de subsistemas y de sus interfaces	101
 Capítulo 6. EVALUACIÓN DE LA FASE DE INICIO	104
6.1 Cumplimiento de los criterios	104
6.2 Desarrollo del modelo de caso de uso	105
6.3 Vista de la arquitectura	106
6.4 Cumplimiento de criterios	107
 <u>PARTE III - FASE DE ELABORACION</u>	<u>108</u>
 Capítulo 7. PLANEACION.....	109
7.1 Flujos, actividades y productos a entregar	109
 Capítulo 8. ARQUITECTURA ESTABLE.....	110
8.1 Actores y casos de uso.....	110
8.1.1 Casos de uso.....	110
8.2 Análisis.....	118
8.2.1 Análisis de la arquitectura.....	118
8.2.2 Análisis de casos de uso.....	121
8.2.2.1 Paquete de gestión consultar temas.....	122
8.2.2.2 Paquete de gestión resolver ejercicios.....	123
8.2.2.3 Paquete de gestión evaluar tema.....	124
8.2.2.4 Paquete de gestión consultar registros.....	125
8.2.2.5 Paquete de gestión de administración.....	126
8.3 Diseño.....	127

8.3.1	Diseño de la arquitectura.....	128
8.3.1.1	Nodos y configuraciones de red.....	129
8.3.1.2	Subsistemas.....	130
8.3.2	Diseño del subsistema de gestión de sesiones.....	131
8.3.3	Diseño guiado por el modelo de arquitectura de tres capas.....	134
8.3.4	Diseño de la interfaz.....	134
8.3.5	Diseño de la base de datos.....	139
8.3.6	Software de la herramienta y de la capa intermedia.....	142
8.4	Implementación.....	144
8.4.1	Software del sistema.....	144
Capítulo 9. EVALUACION.....		146
9.1	Desarrollo del modelo de casos de uso.....	146
9.2	Cumplimiento de criterios.....	148
<u>PARTE IV – FASE DE CONSTRUCCION.....</u>		<u>150</u>
Capítulo 10. FLUJO, ACTIVIDADES Y PRODUCTOS A ENTREGAR.....		152
Capítulo 11. ESTÁNDARES DE IMPLEMENTACIÓN.....		153
11.1	Convenciones de codificación.....	153
11.2	Perfiles de usuario.....	155
11.3	Convenciones de interfaz.....	157
11.4	Mensaje del sistema.....	157
11.5	Subsistemas de implementación.....	159
Capítulo 12. SUBSISTEMAS GENERALES.....		160
12.1	Subsistema de gestión de sesiones.....	160

12.1.1	Implementación.....	161
12.1.2	Pruebas.....	163
12.1.2.1	Casos de prueba.....	163
12.1.2.2	Procedimientos de prueba.....	164
12.2	Subsistema de gestión de interfaz.....	165
12.2.1	Implementación.....	165
12.2.1.1	Primer grupo de paginas.....	166
12.2.1.2	Segundo grupo de paginas.....	166
12.2.1.3	Formularios en delphi.....	167
12.2.2	Pruebas.....	169
12.2.2.1	Casos de prueba.....	169
12.2.2.2	Procedimiento de prueba.....	169
Capitulo 13. SUBSISTEMAS DE APLICACIÓN.....		171
13.1	Subsistema de teoría.....	171
13.1.1	Implementación.....	172
13.1.2	Pruebas.....	174
13.1.2.1	Casos de prueba.....	174
13.1.2.2	Procedimiento de prueba.....	175
13.2	Subsistema de ejercicios.....	175
13.2.1	Implementación.....	176
13.2.2	Pruebas.....	179
13.2.2.1	Casos de prueba.....	179
13.2.2.2	Procedimiento de prueba.....	179
13.3	Subsistema de registro.....	180
13.3.1	Implementación.....	180
13.3.2	Pruebas.....	182
13.3.2.1	Casos de prueba.....	182
13.3.2.2	Procedimiento de prueba.....	182
13.4	Subsistema de evaluación.....	184
13.4.1	Implementación.....	185
13.4.2	Pruebas.....	187
13.4.2.1	Casos de prueba.....	187

13.4.2.2	Procedimiento de prueba.....	187
13.5	Subsistema de Administración.....	188
13.5.1	Implementación.....	189
Capítulo 14. EVALUACIÓN.....		192
14.1	Resumen del capítulo.....	192
14.2	Desarrollo del modelo de caso de uso.....	192
CONCLUSIONES.....		195
RECOMENDACIONES.....		197
BIBLIOGRAFIA.....		199

LISTA DE TABLAS

Tabla 1. Generaciones en el desarrollo de sitios Web.....	16
Tabla 2. Lenguajes del lado del Cliente.....	20
Tabla 3. Lenguajes del lado del servidor.....	23
Tabla 4. Comparación de lenguajes del lado del servidor.....	23
Tabla 5. Ejemplo de vista de un archivo de PHP en el servidor y en el cliente.....	26
Tabla 6. Resumen de los principales Modelos de ciclo de vida de desarrollo del..... Software.....	53
Tabla 7. Guía para la selección del ciclo de vida a utilizar en un proyecto.....	55
Tabla 8. Productos y actividades de cada uno de los flujos de trabajo.....	60
Tabla 9. Diagramas UML utilizados para la realización de la herramienta..... GEDISO 1.0.....	64
Tabla 10. Listado de requisitos candidatos.....	71
Tabla 11. Listado de riesgos críticos.....	73
Tabla 12. Descripción del proceso de negocio consultar temas.....	78
Tabla 13. Actores de la herramienta.....	82
Tabla 14. Descripción del modelo general de casos de uso.....	85
Tabla 15. Descripción del caso de uso escoger tema.....	86
Tabla 16. Descripción del caso de uso resolver ejercicio.....	87
Tabla 17. Descripción detallada del caso de uso escoger tema.....	89
Tabla 18. Descripción detallada del caso de uso escoger subtema.....	90
Tabla 19. Casos de usos identificados en la fase de inicio.....	91
Tabla 20. Requisitos adicionales.....	91
Tabla 21. Casos de usos identificados hasta ahora.....	106
Tabla 22. Descripción del modelo de casos de uso: diagrama de ver Información de.... los Temas.....	111
Tabla 23. Descripción del modelo de caso de uso: diagrama de consultar temas....	112
Tabla 24. Descripción del modelo de caso de uso: diagrama de resolver ejercicio...	113
Tabla 25. Descripción del modelo de caso de uso: diagrama de evaluar tema.....	114
Tabla 26. Descripción del modelo de caso de uso: diagrama de consultar..... Registros.....	114
Tabla 27. Descripción del modelo de caso de uso: diagrama de administrar..... Grupo.....	115
Tabla 28. Descripción del modelo de caso de uso: diagrama de administrar.....	

Ejercicios.....	116
Tabla 29. Descripción del modelo de caso de uso: diagrama de administrar.....	
Estudiantes.....	116
Tabla 30. Descripción del modelo de caso de uso: diagrama de administrar otras.....	
Opciones.....	117
Tabla 31. Zonas de la herramienta GEDISO 1.0.....	136
Tabla 32. Descripción de las tablas en el diseño de la base de datos.....	140
Tabla 33. Casos de uso identificados hasta el final de la fase de elaboración.....	148
Tabla 34. Estándares para las variables de los formularios de datos.....	154
Tabla 35. Opciones de trabajo de la herramienta, por su perfil.....	157
Tabla 36. Subsistemas de implementación.....	159
Tabla 37. Código para el inicio de sesión y registro de las variables.....	161
Tabla 38. Variables de sesión registradas para cada usuario que acceda a la.....	
Herramienta.....	161
Tabla 39. Sesión de validación de acceso a la herramienta.....	162
Tabla 40. Código php que permite validar el ingreso de un usuario al sistema.....	
Con sus respectivos permisos.....	163
Tabla 41. Prueba para validar el ingreso de un usuario a la herramienta.....	165
Tabla 42. Código php que permite validar email de un usuario.....	168
Tabla 43. Prueba para validar el ingreso de un usuario a la herramienta.....	170
Tabla 44. Código php para el ingreso de anexos.....	174
Tabla 45. Prueba para el ingreso de los anexos.....	175
Tabla 46. Código en delphi para solucionar problemas por parte de los.....	
Estudiantes.....	178
Tabla 47. Prueba del almacenamiento de los datos de las practicas al solucionar.....	
Los ejercicios por parte de los estudiantes.....	180
Tabla 48. Código php que permite la visualización del historial de trabajo de los.....	
Estudiantes.....	181
Tabla 49. Prueba para la inserción de los registros de trabajo, desde el modulo de.....	
Problemas.....	184
Tabla 50. Código php que permite asignar las notas a los estudiantes.....	186
Tabla 51. Prueba para buscar la evaluación del estudiante.....	188
Tabla 52. Código php que permite la creación de un grupo de trabajo.....	191
Tabla 53. Casos de uso de la herramienta GEDISO 1.0.....	194

LISTA DE FIGURAS

Figura 1. Programación en el lado del cliente.....	17
Figura 2. Programación en el lado del servidor.....	18
Figura 3. Esquema mixto, programación en el cliente y en el servidor.....	18
Figura 4. Muestra como se interpreta php.....	25
Figura 5. Alcance de intranet, extranet, e internet.....	29
Figura 6. Arquitectura cliente – servidor.....	33
Figura 7. Arquitectura de tres capas. Caso más general de arquitecturas múltiples...	34
Figura 8. Modelo de cascada pura.....	39
Figura 9. Modelo de cascada con frases solapadas.....	40
Figura 10. Modelo de cascada con subproyectos.....	40
Figura 11. Modelo de cascada con reducción de riesgos.....	41
Figura 12. Modelo de desarrollo de aplicaciones rápidas (DRA).....	41
Figura 13. Modelo de prototipo simple.....	43
Figura 14. Modelo de prototipo evolutivo.....	44
Figura 15. Modelo de entrega por etapas.....	45
Figura 16. Modelo de espiral.....	47
Figura 17. Modelo de proceso unificado.....	49
Figura 18. Modelo de caso de uso del negocio.....	76
Figura 19. Modelo del dominio.....	79
Figura 20. Modelo de casos de uso general.....	83
Figura 21. Modelo de caso de uso, casos de uso por actor.....	86
Figura 22. Diagrama de estados del caso de uso: escoger tema.....	88
Figura 23. Diagrama de estados del caso de uso: escoger subtema.....	89
Figura 24. Identificación de paquetes del análisis a partir de clases de dominio.....	93
Figura 25. Paquete de análisis gestión de temas a partir de casos de uso.....	94
Figura 26. Paquete de análisis gestión ejercicios a partir de casos de uso.....	94
Figura 27. Paquete de análisis gestión evaluar a partir de casos de uso.....	95
Figura 28. Paquete de análisis gestión consultar registros a partir de casos de uso.....	95
Figura 29. Paquete de análisis gestión adm. grupo a partir de casos de uso.....	96
Figura 30. Paquete de análisis gestión adm. Ejercicios a partir de casos de uso.....	96
Figura 31. Dependencias y capas de paquetes de análisis.....	97
Figura 32. Clases de entidades derivadas del modelo de dominio.....	97
Figura 33. Clases de análisis del caso de uso asignar notas.....	98

Figura 34. Diagrama de colaboración de las clases de análisis, casos de uso.....	
Asignar grupos.....	99
Figura 35. Diagrama de despliegue de GEDISO 1.0.....	101
Figura 36. Identificación de subsistemas a partir de los paquetes de análisis.....	103
Figura 37. Modelo de caso de uso, diagrama de información de los temas.....	111
Figura 38. Modelo de caso de uso, diagrama de consultar temas.....	111
Figura 39. Modelo de caso de uso, diagrama de resolver ejercicios.....	112
Figura 40. Modelo de caso de uso, diagrama de evaluar tema.....	113
Figura 41. Modelo de caso de uso, diagrama de consultar registros.....	114
Figura 42. Modelo de caso de uso, diagrama de administrar grupo.....	115
Figura 43. Modelo de caso de uso, diagrama de administrar ejercicios.....	115
Figura 44. Modelo de caso de uso, diagrama de administrar estudiantes.....	116
Figura 45. Modelo de caso de uso, diagrama de administrar otras funciones.....	117
Figura 46. Paquete de análisis de gestión resolver ejercicios.....	119
Figura 47. Paquetes de análisis de gestión de Adm., gestión de Adm. de grupo,.....	
gestión de Adm. de ejercicios, Adm. de usuarios. Estos últimos contenidos	
en el primero.....	119
Figura 48. Paquete de análisis gestión de consultar registro.....	120
Figura 49. Paquete de análisis gestión evaluar tema.....	120
Figura 50. Paquete de análisis gestión consultar temas.....	121
Figura 51. Diagrama de colaboración de las clases de análisis para el caso de uso.....	
consultar subtemas.....	123
Figura 52. Diagrama de colaboración de las clases de análisis para el caso de uso.....	
escoger ejercicio.....	124
Figura 53. Diagrama de colaboración de las clases de análisis para el caso de uso.....	
consultar notas evaluación.....	125
Figura 54. Diagrama de colaboración de las clases de análisis para el caso de uso.....	
crear Informes.....	126
Figura 55. Diagrama de colaboración de las clases de análisis para el caso de uso.....	
administrar grupos.....	127
Figura 56. Arquitectura en capas.....	129
Figura 57. Subsistema de diseño.....	131
Figura 58. Diagrama de secuencia de las interacciones entre subsistemas.....	133
Figura 59. Diseño general de la interfaz de la herramienta GEDISO 1.0, modulo de.....	
teoría.....	138
Figura 60. Diseño general de la interfaz de la herramienta GEDISO 1.0.....	138
Figura 61. Diagrama entidad – relación.....	141

Figura 62. Mensaje en una ventana propia del navegador.....	158
Figura 63. Mensaje en la misma incrustada en la misma pagina que generó el..... error.....	158
Figura 64. Mensaje en una ventana nueva. Mensaje critico.....	158
Figura 65. Interfaz que permite ingresar a la herramienta.....	163
Figura 66. Interfaz que permite ingresar la información del usuario.....	169
Figura 67. Interfaz que permite el ingreso de los anexos.....	174
Figura 68. Interfaz que permite realizar los problemas.....	178
Figura 69. Interfaz que permite la visualización del historial.....	182
Figura 70. Interfaz que permite asignar la nota a los estudiantes.....	186
Figura 71. Interfaz que permite la creación de un grupo de trabajo.....	191

LISTA DE ANEXOS

ANEXO A – Requisitos candidatos.....	201
ANEXO B – Modelo del negocio basado en casos de uso.....	206
ANEXO C – Modelo de caso de uso de la herramienta.....	208
ANEXO D – Diccionario de datos.....	218
ANEXO E – Escenarios de prueba.....	226

INTRODUCCIÓN

La influencia de la tecnología está transformando el orden mundial desde la economía de los países hasta sus sistemas educativos. En el campo educativo ha incursionado con fuerza en el desarrollo de herramientas computacionales que ayudan en el proceso de enseñanza-aprendizaje en escuelas y universidades, utilizando como medio de difusión internet.

Por esto la Universidad Industrial de Santander (UIS), esta apoyando el uso de la tecnología para beneficio de la comunidad universitaria. Por esta razón se han creado las intranets, las cuales son el medio utilizado para difundir el uso de nuevas herramientas logrando una comunicación eficiente.

El Grupo de Investigación en Modelos y Simulación (SIMON)¹ en una labor interdisciplinaria apoya el desarrollo de herramientas software que contribuyen a la solución de problemas presentes en otras áreas del conocimiento, es de resaltar la colaboración con el proceso académico facilitando la interactividad entre la comunidad educativa, al crear un medio de comunicación entre profesores y estudiantes, permitiendo llevar un seguimiento de los progresos de los estudiantes en las labores realizadas en las aulas de clase de la Escuela de Diseño Industrial, específicamente en la asignatura titulada Geometría Descriptiva.

El grupo SIMON en el área de la informática educativa durante los últimos ocho años ha trabajado en la aplicación del Pensamiento de Sistemas y la Dinámica de Sistemas en la educación, dando a conocer sus resultados con artículos y ponencias en eventos académicos, siendo su principal proyecto la propuesta MAC², software motivado por una preocupación por llevar un paradigma de pensamiento a través de una teoría tecnológica instrumentalizada por la informática y el apoyar el aprendizaje.

La Metodología propuesta para la construcción y desarrollo del software es la del Proceso Unificado de Desarrollo de Software, ya que permite un desarrollo paralelo al proceso de investigación e intervención, mediante prototipos que se depuran en el

¹ Grupo adscrito a la Escuela de Ingeniería de Sistemas e Informática de la UIS. <http://azulejo.uis.edu.co/web/investigacion/grupos/SIMON/index.html>

² MAC 1-11. Micromundo de Simulación para el Aprendizaje de Ciencias de la Naturaleza para los grados 1-11. "La Informática Y El Cambio En La Educación: Una Propuesta Ilustrada Con Ambientes De Modelado Y Simulación Con Dinámica De Sistemas: Proyecto MAC." Propuesta presentada en el Primer Encuentro Iberoamericano de Dinámica de Sistemas. Monterrey. México. 2003.

mismo proceso investigativo y que a su vez, son útiles en las acciones de aplicación de los mismos conceptos y procedimientos. Según Jacobson y Booch (2000) “esta metodología se caracteriza por estar dirigida por casos de uso, centrada en la arquitectura y por ser iterativa e incremental”

La información que se ofrece en el presente documento ha sido clasificada principalmente en cuatro partes esenciales: La **parte I**, se establecen los fundamentos generales del proyecto. Plantea los objetivos, justificación y beneficios que este proyecto ofrece, también es presentada la información teórica y el marco metodológico a seguir:

Las siguientes tres partes se centran en mostrar el desarrollo del proyecto a través de cada una de las fases expuestas por la metodología escogida. Estas son inicio, elaboración y construcción. En la **parte II**, Se presenta la fase de inicio, la cual busca desarrollar el análisis de negocio hasta el punto necesario para justificar la puesta en marcha del proyecto. La fase de elaboración, presentada en la **parte III**, guiará al lector en la obtención de una arquitectura estable que guiará la construcción de la herramienta. Por ultimo la fase de construcción, la cual se encuentra en la **parte IV**, presenta las iteraciones necesarias para obtener el sistema propuesto. Para finalizar, se presentan las conclusiones y observaciones del trabajo realizado. Así como los anexos respectivos.

FUNDAMENTOS

En este informe que se presenta a continuación se explicaran las ideas fundamentales y desarrollo del proyecto. Su objeto es el de dar claridad a los objetivos que se desean alcanzar definiendo aspectos de primordial importancia como es la justificación del trabajo desarrollado y el impacto de este tiene ante la comunidad universitaria de la UIS, especialmente a los estudiantes de primer nivel de las diferentes ingenierías que tienen en su pensum académico la asignatura Geometría Descriptiva, además se explicara el marco teórico y la metodología seguida en la obtención de los objetivos.

Los objetivos específicos que se presentan son:

- ◆ Presentar la justificación del proyecto.
- ◆ Formular y describir detalladamente los objetivos del proyecto.
- ◆ Analizar el impacto del proyecto y las proyecciones esperadas.
- ◆ Explicar los alcances de este documento, así como la profundidad con que se tratará cada tema.
- ◆ Hacer una síntesis general de los aspectos a tratar en el documento.
- ◆ Formular el marco teórico necesario para el entendimiento de este trabajo de grado.
- ◆ Definir un marco metodológico para el desarrollo de este proyecto.

En esta sección del documento se explicara temas y relaciones esenciales para la comprensión y seguimiento del resto del proyecto, por tanto, es necesario que en este punto se respondan las preguntas que surgen al respecto del objetivo y la naturaleza de este proyecto.

Capítulo 1.

ASPECTOS GENERALES

Para dar una idea clara de la magnitud del trabajo de grado que se plantea, es necesario enunciar las bases fundamentales que permitan dar una perspectiva hacia el objetivo final. Es por eso que este capítulo marca el comienzo de un proceso completo en donde cada etapa se realiza con el fin de dar los parámetros necesarios para seguir avanzando hasta cumplir todas las metas propuestas.

1.1 OBJETIVOS.

1.1.1 OBJETIVO GENERAL.

Formular una propuesta general para desarrollar una herramienta software, que sirva de apoyo al proceso de enseñanza-aprendizaje de la asignatura Geometría Descriptiva en la escuela de Diseño Industrial de la Universidad Industrial de Santander, con el fin de mejorar el dinamismo de la cátedra y facilitar el aprendizaje e implementar un primer prototipo.

1.1.2 OBJETIVOS ESPECIFICOS.

➤ Formular una propuesta para desarrollar una herramienta software, que sirva de apoyo al proceso de enseñanza-aprendizaje de la geometría descriptiva en todas las áreas de trabajo de la asignatura tales como: teoría, práctica, y evaluación. Utilizando como metodología el proceso unificado de desarrollo de software para el análisis y diseño de la herramienta, dando origen a una secuencia de prototipos que serán la base para la creación de nuevos proyectos, utilizando la misma metodología de trabajo.

➤ Desarrollar un primer prototipo de la herramienta software propuesta que desarrolle las siguientes áreas de trabajo: *área de teoría, área de práctica, área de*

evaluación y área de administración. Con el siguiente soporte técnico mínimo: ambiente multimedia para el estudio conceptual de la geometría descriptiva apoyado por animaciones de los ejemplos presentados, ejercicios propuestos para resolver paso a paso realizados en un editor de imágenes, así como los procesos de actualización y mantenimiento de los contenidos y la base de datos de la herramienta.

➤ Proveer la herramienta con contenidos básicos iniciales que permitan desarrollar un curso de geometría descriptiva, tales contenidos serán: adecuación de ejercicios resueltos del papel al medio computacional por medio de animaciones; contenidos teóricos de la geometría descriptiva; alimentar la base de datos con los listados de cursos, alumnos, profesores y ejercicios para resolver paso a paso de manera que ayuden al usuario a comprender como se resolverán las diferentes clases de ejercicios.

➤ Prueba de receptividad con el fin de observar el desempeño del software, nivel de aceptación, facilidad de manejo y demás características que permitan depurar el prototipo y proponer mejoras a contemplar en futuras versiones.

1.2 JUSTIFICACIÓN.

La educación superior en Colombia, se enfrenta al reto de mantener sus modelos educativos a la par con los avances de la tecnología y metodologías de enseñanza contemporáneas, con el fin de que sus estudiantes sean competitivos a nivel internacional y su nivel de aprendizaje sea optimo.

En los primeros niveles de los programas profesionales se presenta un alto grado de dificultad y aceptación por parte de los estudiantes, lo que conlleva en algunos casos a una desorientación en los objetivos a cumplir en el desarrollo de su carrera profesional. En nuestro caso particular, la geometría descriptiva es considerada el lenguaje del ingeniero y los problemas en su enseñanza están presentes en el día a día, por lo cual se busca una solución que dinamice la cátedra y acerque el estudiante a sus conceptos de una manera ágil y agradable para este, presentándose la informática educativa como una excelente alternativa.

La enseñanza-aprendizaje de la geometría descriptiva presenta su problema fundamental en el aprendizaje por parte de los alumnos, pero que se traduce en un problema de enseñanza, puesto que, aunque son procesos diferentes, son complementarios en cuanto ambos buscan un mismo objetivo como es la aprehensión del conocimiento por parte del estudiante.³

Este problema se palpa en la dificultad que se observa en el alumno para relacionar un objeto que esta en 3D con el dibujo, que esta en 2D, es decir de abstraer la información del objeto para realizar un dibujo o viceversa.

En la búsqueda de soluciones a esta problemática, la informática educativa surge como una herramienta útil y versátil para implementar herramientas software que apoyen y dinamicen el proceso de enseñanza-aprendizaje.

1.2.1 ANTECEDENTES Y DESCRIPCIÓN DEL PROBLEMA.

En el proceso enseñanza-aprendizaje aplicado a la geometría descriptiva se han detectado algunas dificultades debido a la complejidad de las abstracciones espaciales que se deben realizar para procesar la solución de los problemas típicos, siendo los de mayor preponderancia:

- ❖ El requerimiento de un alto grado de habilidad por parte del profesor para exponer esas abstracciones utilizando casos de la vida real y profesional al tablero de clase.
- ❖ La gran cantidad de cursos que son abiertos cada periodo para satisfacer la demanda académica donde se dicta la asignatura.
- ❖ La gran variedad de metodologías de enseñanza y el inadecuado uso del lenguaje técnico propio de la asignatura, repercuten en los contenidos haciéndolos divergir de acuerdo a la metodología que cada profesor utiliza o cree conveniente para dictar la asignatura.

³ Tomado de Monografía para optar el título de especialista en docencia universitaria: La comprensión de la geometría descriptiva por Edgar Augusto sarmiento.

Debido a la inasistencia o no cumplimiento de las horas de consulta por parte de la gran mayoría de los estudiantes, la interacción entre profesor y estudiante es casi nula fuera del salón de clases creando una falta de compromiso por parte del estudiante en el aprendizaje de la asignatura, también se encontró que no todos los estudiantes se encuentran en el mismo nivel de conocimiento previo de la geometría descriptiva, presentándose en los cursos estudiantes avanzados así como estudiantes sin ningún tipo de conocimiento previo sobre el contenido de la asignatura, provocando contratiempos y lentitud para desarrollar los contenidos del curso en los tiempos previstos y en ocasiones dejando vacíos de conocimientos a los estudiantes menos avanzados.

Siendo conscientes de esta realidad la escuela de Diseño Industrial se propuso crear una herramienta que sirviera de apoyo al proceso modernización del proceso enseñanza-aprendizaje aplicado a la geometría descriptiva, por tal motivo en el año 2000 se propuso el desarrollo del proyecto ***“Modernización de la geometría descriptiva”*** que consistió en un programa donde se actualizaba la estructura de la asignatura a través de un "Sistema Integrado" que utilizara herramientas computacionales como apoyo al proceso enseñanza-aprendizaje de la asignatura Geometría Descriptiva.

Proyecto de carácter académico, pedagógico, investigativo e interdisciplinario que dio origen a Glíptica 1.0, como un componente software integrado, en una propuesta metodológica en el que las clases cotidianas de Geometría Descriptiva I y II se rediseñan utilizando como soporte tecnologías computacionales y sistemas de información⁴.

El proyecto se fue concluido en el año 2001, pero luego de ser implantado en dos equipos para su prueba, debido a inconvenientes técnicos y de desarrollo como virus en los equipos y problemas de instalación llevaron a la inutilización del software para su normal empleo.

Ante esta perspectiva el director del proyecto se vio en la necesidad de retomar el proyecto, buscando el apoyo del grupo SIMON de investigación encabezado por el Ing. Hugo Andrade Sosa quien participo como colaborador en el proyecto original, ***glíptica 1.0***, decidiéndose entonces presentarlo a consideración del grupo SIMON “la

⁴ Tomado del libro de requisitos de la Herramienta Software Glíptica 1.0, desarrollado para la escuela de Diseño Industrial.

creación de una nueva herramienta software que satisfaga las necesidades del problema propuesto”.

De acuerdo a los referentes anteriormente descritos, se pretende crear una herramienta software que le garantice al estudiante tener acceso fácil y rápido a los conceptos de la geometría descriptiva necesarios para fortalecer su formación académica.

El software ha desarrollar debe permitir al estudiante mejorar su habilidad para hacer abstracciones espaciales mediante la solución de problemas, utilizando para tal fin:

- ◆ Una excelente ayuda teórica.
- ◆ Una serie de problemas resueltos como complemento.
- ◆ Una serie de problemas propuestos para resolver que complementen las actividades realizadas en clase.
- ◆ Una ayuda como hoja de trabajo, para desarrollar elementos gráficos.

1.3 IMPACTO.

Con la ejecución del siguiente proyecto se pretende dar soporte al proceso de enseñanza-aprendizaje que se lleva a cabo en la Universidad Industrial de Santander, especialmente en la asignatura Geometría Descriptiva que se dicta en la escuela de Diseño Industrial, ya que permite a los alumnos y profesores interactuar con una nueva experiencia donde la tecnología sirve de apoyo a la actividad académica sirviéndole como estímulo y como punto de referencia para la utilización de este tipo de herramientas computacionales utilizadas como apoyo en sus labores cotidianas como profesores y estudiantes.

El proyecto permite llevar a los estudiantes los temas necesarios para la comprensión de la asignatura Geometría Descriptiva, orientando a la práctica estos contenidos, con el desarrollo de ejercicios generales y específicos de acuerdo a la carrera que cursen los diferentes estudiantes, observar ejemplos de los diferentes temas y permitir a los

profesores evaluar las actividades de los alumnos al utilizar la herramienta en las horas asignadas para trabajar.

El desarrollo de esta primera versión de la herramienta GEDISO 1.0 pretende ser la base para el mejoramiento y desarrollo de nuevos software que mejoren el proceso de enseñanza-aprendizaje entre alumnos y profesores, permitiendo en un futuro la creación de una herramienta que permita a los estudiantes explotar sus capacidades para el desarrollo de los diferentes ejercicios que se le puedan presentar en la asignatura y en su vida diaria, ampliando el alcance de este tipo de software a otras universidades del país o del mundo.

Otro aspecto no menos importante, es la seguridad de la herramienta, la cual permite identificar el usuario y verificar que se encuentre registrado en la herramienta, permitiéndole navegar por una o más secciones autorizadas, evitando que se realice un acceso de forma indebida a sitios de la herramienta restringidos por el perfil de usuario que se le ha asignado.

Además, el uso del Proceso Unificado como metodología para el desarrollo de la herramienta, se constituye en una guía para el grupo de investigación SIMON, ya que, aunque no es el primer proyecto desarrollado con esta metodología dentro del grupo, se plantea su seguimiento desde el desarrollo de su plan de proyecto hasta la finalización del proyecto y entrega del producto terminado. Otorgando una guía seria y completa para el desarrollo de futuras versiones.

La documentación presentada en este libro se desarrolla no solo mostrando los resultados o productos entregados sino presentando una explicación de las actividades realizadas para la elaboración del proyecto, lo cual resulta de gran utilidad para aquellos estudiantes interesados en seguir esta metodología y que desean tener un punto de referencia para su adaptación a un desarrollo en particular.

1.4 DESCRIPCIÓN DEL DOCUMENTO.

Muchas veces al recorrer la biblioteca, en la sección de tesis de grado, se encuentran infinidad de documentos que cuentan con una gran cantidad de folios, planos, citas, anexos y otros artículos que varían de acuerdo al tipo de trabajo que se realiza. Esto

produce gran emoción, ya que se cree hallar un buen trabajo de grado que sirva de guía para la elaboración del documento final de una Tesis de este tipo.

Pero solo basta con realizar un estudio rápido a esta clase de memorias para darse cuenta que ha sido una transcripción exagerada de temas, definiciones y otros ítem, que se encontraría en cualquier libro relacionado con la materia. En ocasiones no se presenta un resumen, sino reales transcripciones de otros textos. Entonces se debe cuestionar si en verdad para poder entender dicho proyecto se necesita leer toda esa información que los autores están presentando.

Debido a que las persona que consultan este libro puede variar entre gente con una alto grado de conocimiento en el desarrollo de software educativo con tecnologías intranet-Internet, o el vasto público de los no especialistas en estos temas, se hace necesario decirle al lector en general, que puede esperar encontrar en las memorias de este trabajo de grado y por supuesto que no.

Por tanto este documento, ha sido concebido de tal forma que tenga un sentido práctico, sin caer al extremo de demostrar cada hipótesis o razonamiento, pero dando los fundamentos necesarios para que el lector sienta que es un trabajo serio basado en un estudio cuidadoso de los temas tratados. Como se sabe, el desarrollo de las nuevas herramientas con tecnologías de Intranet e Internet ha aumentado considerablemente y han convertido su estudio en un tema realmente complejo, sin embargo se decidió no expandir las bases teóricas de este documento mas allá de los conceptos prácticos indispensables para un buen entendimiento de este caso en especial.

Ante todo, se definen los términos que se utilizan y se presentan comparaciones entre las diferentes opciones y tecnologías existentes, para posteriormente presentar al lector las razones que motivaron las elecciones realizadas para el desarrollo del proyecto.

1.5 CONTENIDO GENERALES DEL DOCUMENTO.

La información que se ofrece en el presente documento ha sido clasificada principalmente en cuatro partes esenciales.

En la **parte I**, se establecen los fundamentos generales del proyecto. Es la razón de ser de este trabajo. En el **capítulo 1** se empieza por presentar los aspectos generales. Es el punto donde necesariamente se debe informar al lector todo lo relacionado con este trabajo, de donde surge la idea, que necesidades se están supliendo y que beneficios ofrece. Se presenta detalladamente los objetivos que se desean alcanzar así como se ofrece una explicación clara de los alcances de este documento escrito y lo que se debe esperar de él.

Hacia el **capítulo 2**, es presentada la información teórica considerada como necesaria para una buena comprensión del proyecto, dentro de esta información se encuentra los conceptos generales sobre tecnologías Web, Tecnologías de desarrollo de software educativo, las soluciones y los productos disponibles y las diferentes arquitecturas de software que pueden servir de base para un desarrollo de este tipo.

Para poder avanzar a través del proyecto, es necesario presentar un marco metodológico a seguir. Este se presenta en el **capítulo 3**. Su importancia radica en que de esta manera se asegura una canalización de los esfuerzos realizados para lograr los objetivos propuestos. Primero se presentan los principales ciclos de vida para el desarrollo de software, de manera que se presente al lector los aspectos claves por los cuales se seleccionó el proceso unificado como ciclo de vida a seguir, presentando una introducción a la metodología escogida.

Las siguientes tres partes se centran en mostrar el desarrollo del proyecto a través de cada una de las fases expuestas por la metodología escogida, estas son inicio, elaboración y construcción. En el desarrollo de este libro se ha puesto atención en mostrar los progresos obtenidos al aplicar el proceso unificado como metodología, más que en presentar solo los productos o artefactos que se obtendrían al finalizar el proyecto aquí planteado. Cada parte consta de un capítulo inicial para la planeación de la fase, y un capítulo final para la evaluación de la misma, en el capítulo de evaluación de la fase, se resalta el cumplimiento de cada uno de los objetivos planteados y presenta al lector un breve resumen de los avances obtenidos.

En la **parte II**, Se presenta la fase de inicio, fase que busca desarrollar el análisis de negocio hasta el punto necesario para justificar la puesta en marcha del proyecto. No es un estudio completo de la herramienta propuesta, sino que en ella se busca el porcentaje de casos de uso necesarios para fundamentar el análisis del negocio inicial, con el objetivo de delimitar el alcance de la herramienta propuesta. **El capítulo 5**

muestra el desarrollo de las actividades a través de los flujos de trabajo necesarios para lograr los objetivos planteados, los **capítulo 4 y 5** presentan, respectivamente, la planeación y evaluación de esta fase.

La fase de elaboración, presentada en la **parte III**, guiará al lector en la obtención de una arquitectura estable para guiar la herramienta y poder planificar la fase de construcción con gran precisión. Igual que para la **parte II** del libro, la planeación de esta fase se encuentra en el **capítulo 7**, el desarrollo de los flujos de trabajo se encuentran en el capítulo central de esta parte, el **capítulo 8**, y su evaluación en el **capítulo 9**.

Por ultimo la fase de construcción, la cual se encuentra en la **parte IV**, presenta la iteración para obtener el primer prototipo de software propuesto. En la parte final, **parte V**, se presentan las conclusiones y observaciones del trabajo realizado, además, se generan las recomendaciones necesarias con miras a la adecuación de mejoras en el futuro y las razones por las cuales no se implementaron en este trabajo.

Para finalizar y con objeto de brindar una información completa, se entregan los anexos necesarios para el total entendimiento de este proyecto. Entre los anexos se encuentran la lista de requisitos candidatos, el modelo de negocio y de casos de uso, las opciones de la herramienta, así como las pruebas de unidad realizadas a la herramienta. Por ultimo, en los anexos se encuentran los manuales de usuario de la herramienta.

Capítulo 2.

MARCO TEÓRICO

Este capítulo es donde se plasman los aspectos principales para una buena comprensión de la teoría necesaria en el desarrollo de este proyecto de grado. Es una transcripción sencilla y corta de la fundamentación teórica en las tecnologías de programación, Internet, redes, y un poco sobre educación; estos son algunos de los temas esenciales que el lector debe tener claro antes de empezar el desarrollo general de este trabajo de grado.

2.1 APLICACIONES WEB.

Para comprender las implicaciones de los conceptos de las aplicaciones Web, se necesita establecer la diferencia entre aplicaciones Web y simples scripts:

“Un script es una utilidad, y como tal no tienen ningún contexto. No sabe nada de los demás scripts del sistema. Sin embargo, una aplicación esta diseñada para realizar tareas mas avanzadas. Una aplicación necesita mantener un estado y ejecutar transacciones, ya que es interactiva. Como suele requerir más interacción de usuario que un simple script, deberá preocuparse también por la seguridad y usabilidad”⁵

A continuación se profundiza en este aspecto, explicando su funcionamiento, su evolución y las tecnologías disponibles para su desarrollo.

2.1.1 FUNCIONAMIENTO DE UNA APLICACIÓN WEB.

El funcionamiento de una aplicación Web es un ejemplo típico de la arquitectura de capas, en donde múltiples clientes se conectan a un servidor (en algunos casos varios) en forma simultánea. Como en todo esquema por capas debe existir un protocolo que especifique de que forma se comunican e intercambian datos el cliente y el servidor, el protocolo utilizado en una aplicación Web es el protocolo HTTP que funciona “encapsulado” sobre el protocolo TCP/IP.

⁵ Ratschiller, Tobias y Till Gerken. Prentice. Creación de aplicaciones Web con PHP4. Pearson Educación, S.A., Madrid, 2001. Pagina 129.

Básicamente este protocolo es iniciado por el cliente con un “request”, es decir un pedido de un recurso determinado, que es casi siempre contestado por el servidor con el envío de una respuesta (“response”) que incluye un código indicando si el pedido pudo o no, ser resuelto por el servidor.

2.1.2 EVOLUCIÓN DE LOS SITIOS WEB.

Desde la aparición de la Web, distintos prototipos de sitios han sido predominantes a través del tiempo, progresando hasta dar lugar, en la actualidad, a verdaderas aplicaciones y sistemas de información con base en esta tecnología.

A continuación se presenta una reseña de las principales generaciones de los sitios Web desarrollados, de acuerdo a sus características comunes predominantes. Por ultimo se presenta la **tabla 1.** que resume estas generaciones.

2.1.2.1 SITIOS WEB DE PRIMERA GENERACIÓN.

En el desarrollo Web de primera generación las páginas se desarrollaban, se alojaban en el servidor y este se encargaba de enviar las páginas al navegador. Es un modelo basado en páginas estáticas, en donde predominaba el uso de texto, enlaces a otras páginas y listas para enumerar cosas. Frecuentemente se usaban líneas horizontales para separar contenidos y páginas de gran extensión vertical con gran cantidad de texto. Si bien no eran visualmente atractivos estos desarrollos estaban enfocados a funcionar en forma veloz y entregar al usuario gran cantidad de información interrelacionada.

2.1.2.2 SITIOS WEB DE SEGUNDA GENERACIÓN.

La segunda generación implicó una revolución en lo visual, a medida que se descubría en Internet un nuevo potencial para los negocios comerciales, el hecho de “capturar” usuarios se torno una premisa y por ello se le dio gran importancia al aspecto visual. Las páginas con capas visualmente atractivas fueron más y más populares, el uso abundante de imágenes planas y animadas y elementos multimedia se volvió común, aparecieron páginas que controlaban el estilo y la posición en la que los mismos se ubicaban. Para dar respuesta a estas necesidades de los desarrolladores surge

DHTML y JavaScript, permitiendo tener gran control de la presentación de las páginas en el navegador del usuario.

Otra característica importante de esta segunda generación fue la aparición explosiva de más y más aplicaciones a la medida. Los servicios que ofrecía un sitio se volvían factores importantes para la atracción de usuarios: chats, foros de discusión, banners, contadores, etc. Y empezaron a aparecer las fallas del protocolo CGI, a medida que las aplicaciones eran más grandes y la cantidad de usuarios crecía exponencialmente.

2.1.2.3 SITIOS WEB DE TERCERA GENERACIÓN.

La tercera generación de sitios Web, que abarca hasta la fecha actual, siguió basada en lo visual. El gran cambio vino en la forma cómo se generaba la información. Las páginas estáticas que dominaban el 100% de los sitios de primera generación y la mayoría de la segunda, fueron reemplazadas por páginas dinámicas que el servidor Web generaba en el momento que eran solicitadas, a partir de información que generalmente se guardan en una base de datos. Estos sitios “dinámicos” permiten actualizar la información e incluso cambiar completamente la forma en que se muestran dichos datos en velocidades asombrosas. Los sitios de tercera generación facilitaron las aplicaciones interactivas, la información en tiempo real. Las aplicaciones empezaron a desarrollarse también usando otras tecnologías dejando de lado el protocolo CGI. Aplicaciones en ASP, mod_perl o PHP, más poderosas y eficientes que sus pares CGI, son el estándar de este tipo de sitios.

	Primera Generación	Segunda Generación	Tercera Generación
Tipos de paginas	Estáticas en su totalidad	La mayoría Estáticas. Una pequeña porción de paginas dinámicas	Generalmente Dinámicas
Tecnología utilizada	<ul style="list-style-type: none"> • HTML 	<ul style="list-style-type: none"> • Capas • DHTML • JavaScript, VisualScript • Hojas de Estilos • CGI 	<ul style="list-style-type: none"> • Bases de Datos • PHP • ASP • Mod_perl
Características	<ul style="list-style-type: none"> • Su objetivo era entregar gran cantidad de información de forma rápida 	<ul style="list-style-type: none"> • Su objetivo era capturar nuevos clientes para los negocios. • Desarrollo de aplicaciones a la medida. • Revolución en lo visual 	<ul style="list-style-type: none"> • Creación aplicaciones interactivas con el cliente. • Su principal interés es mantener información actualizada en tiempo real. • Continúa la

			preocupación por lo visual.
--	--	--	-----------------------------

Tabla 1. Generaciones en el desarrollo de sitios Web.

2.1.2.4 PÁGINAS WEB ESTÁTICAS.

Las páginas Web son textos ASCII escritos en el lenguaje HTML (Hypertext Markup Language) que se transfieren entre los servidores de WWW y los navegadores de los clientes mediante el protocolo HTTP (Hypertext Transfer Protocol). Las páginas Web eran estáticas, en el sentido de que, a efectos de usuario, el único proceso realizado era el de visualización de sus contenidos (escritos en lenguaje HTML) por parte del explorador del cliente. Las páginas estáticas forman la base necesaria para la presentación de datos en diferentes tipos de situaciones. También influye decisivamente la sencillez con que se pueden crear, instalar y mantener. Uno de los lenguajes que presenta páginas Web estáticas es HTML, a continuación se presentan algunas características de este lenguaje.

2.1.2.5 HTML.

HTML (Hypertext Markup Language) es el lenguaje que se utiliza para crear páginas Web. Consta de un conjunto de elementos, denominados etiquetas o marcas, que se utilizan para incluir texto, dibujos, tablas, cabeceras, etc.

Las marcas que se introducen en un documento HTML no son visibles cuando se muestran en un navegador (browser o explorador). Cuando un usuario solicita una página HTML a un servidor Web, este envía la página tal cual (incluidas las etiquetas y los atributos). En el momento en que el explorador recibe la página, interpreta las etiquetas que esta contiene junto con los atributos de las mismas, mostrando al usuario el resultado final (sin etiquetas).

2.1.3 TECNOLOGÍAS DISPONIBLES PARA EL DESARROLLO.

Para desarrollar aplicaciones y dotar a las páginas Web de funcionalidad se puede trabajar tanto en el lado del cliente como en el lado del servidor, las variantes son:

Programación en el cliente:

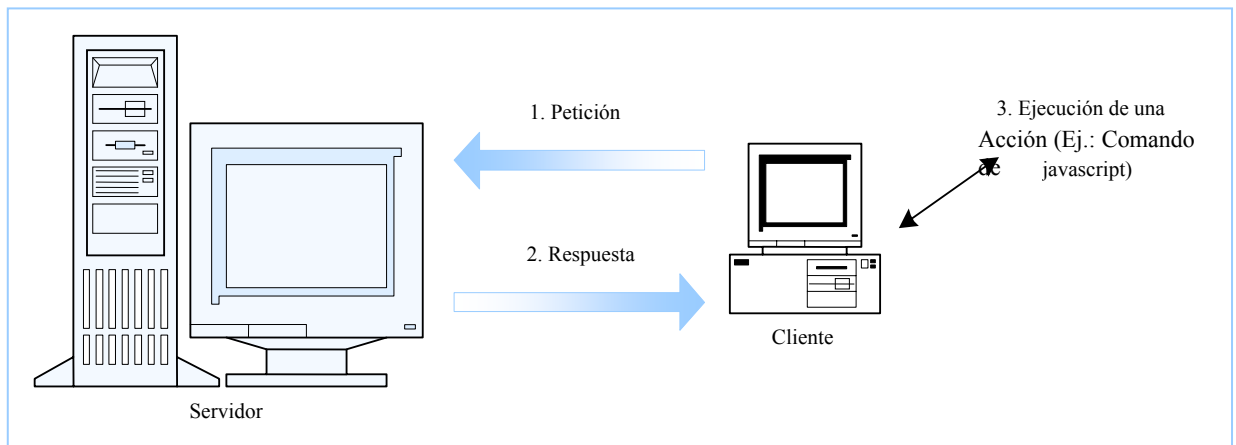


Figura 1. Programación en el lado del cliente.

1. El navegador envía un request.
2. El servidor envía un response que contiene código que el navegador entiende.
3. El navegador interpreta el código enviado por el servidor y realiza una determinada acción.

Programación en el servidor:

1. El navegador envía un request.
2. El servidor ejecuta una aplicación que realiza una determinada acción.
3. El servidor envía el resultado de dicha aplicación al cliente.
4. El navegador muestra el resultado recibido del servidor.

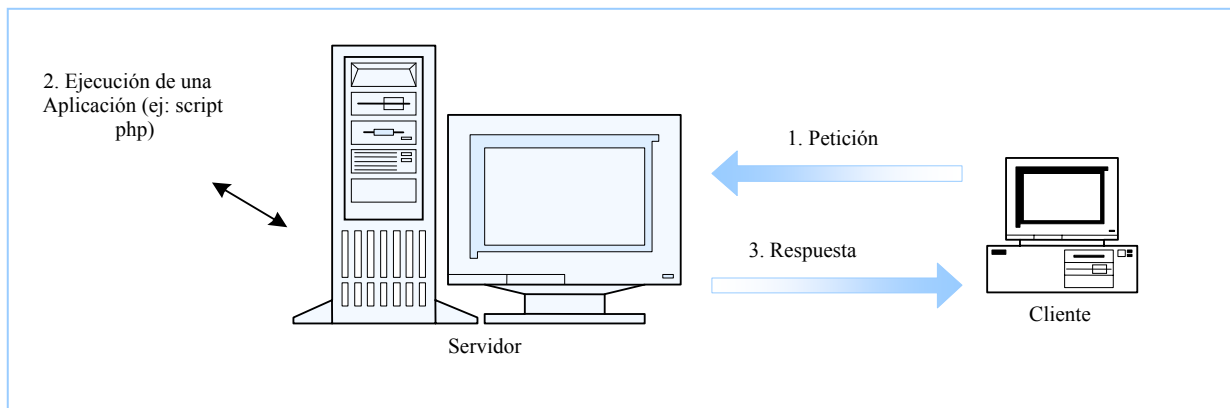


Figura 2. Programación en el lado del Servidor.

Esquema mixto: (programación en el cliente y en el servidor)

1. El navegador envía un request.
2. El servidor ejecuta una aplicación que realiza una determinada acción.
3. El servidor envía el resultado de dicha aplicación al cliente conteniendo código a interpretar por el navegador.
4. El navegador interpreta el código enviado por el servidor y realiza una determinada acción.

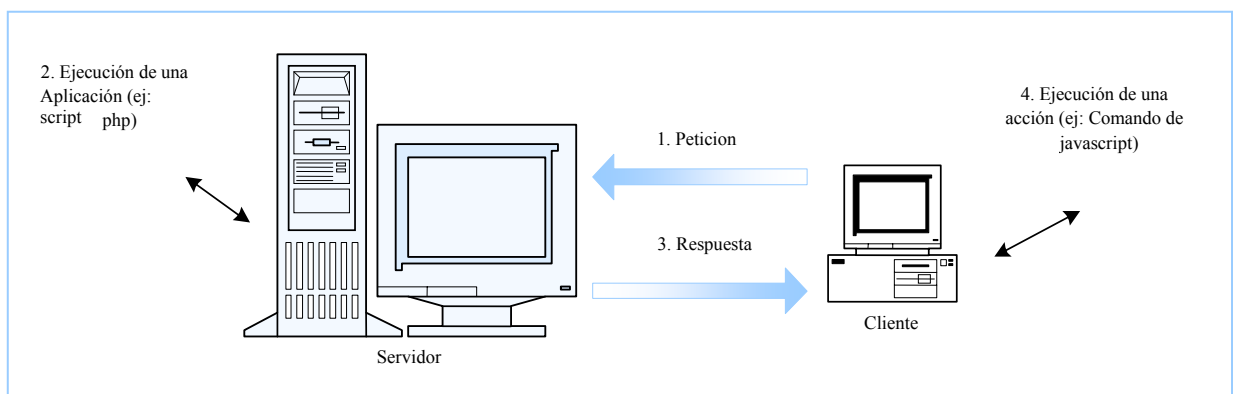


Figura 3. Esquema mixto. Programación en el cliente y en el servidor.

La programación del lado del cliente tiene como principal ventaja que la ejecución de la aplicación se delega al cliente, con lo cual se evita recargar al servidor de trabajo. El

servidor sólo envía el código, y es tarea del navegador interpretarlo. La gran desventaja de esta metodología es que el código que el servidor envía es “sensible” a qué cosas puede o no hacer el navegador. El usuario puede, por ejemplo, decidir deshabilitar una funcionalidad del navegador que es necesaria para que se ejecute un determinado servicio o peor aún, navegadores distintos pueden interpretar el mismo código de distintas formas. Típicamente Netscape y Microsoft, que producen los dos navegadores más usados del mercado, no están de acuerdo sobre como se implementan diversas tecnologías en el cliente.

Programar del lado del servidor tiene como gran ventaja que cualquier cosa puede hacerse sin tener en cuenta el tipo de cliente, ya que la aplicación se ejecuta en el servidor que es un ambiente controlado. Una vez ejecutada la aplicación, el resultado que se envía al cliente puede estar en un formato “normalizado” que cualquier cliente puede mostrar. La desventaja reside en que el servidor se sobrecarga de trabajo ya que además de servir páginas, es responsable de ejecutar aplicaciones. A menudo esto redundante en requisitos de hardware mayores a medida que el servidor ejecuta más y más servicios.

Sin embargo, debido a las incompatibilidades existentes y a la posibilidad de que el usuario controle que cosas se ejecutan y cuales no, la programación del lado del cliente no es recomendable y debe limitarse a código altamente estándar que pueda interpretarse de cualquier forma en cualquier navegador, lo cual obliga a ejecutar la gran mayoría de las aplicaciones y servicios del lado del servidor.

2.1.3.1 PROGRAMACIÓN DEL LADO DEL CLIENTE.

Una página Web se considera dinámica en el cliente cuando está diseñada para que se interprete (o ejecute) código relacionado con la página en el equipo del usuario. Las ventajas que presenta este tipo de páginas respecto a las páginas dinámicas en el servidor son:

- ◆ Liberan al servidor de una gran carga de cómputo.
- ◆ En general, disminuyen la cantidad de ancho de banda empleado en comunicaciones.
- ◆ Normalmente ofrecen respuestas inmediatas al usuario.
- ◆ Permiten la validación interactiva de datos en formularios.

- ◆ Permiten utilizar recursos locales como el ratón, CPU, memoria, etc.

Los inconvenientes que presentan son:

- ◆ A menudo son dependientes del navegador (por ejemplo, Netscape no interpreta Visual Basic Script). Incluso existen diferencias en las versiones de cada lenguaje que soportan los diferentes navegadores.
- ◆ En ocasiones son poco seguras (por ejemplo, utilizando aplicaciones ActiveX de forma indiscriminada).
- ◆ A veces los navegadores no se pueden actualizar con facilidad para cubrir nuevas tecnologías (por ejemplo, al utilizar algunos modelos de Web TV).
- ◆ No pueden resolver situaciones en las que la propia naturaleza del problema es centralizada (por ejemplo, consultar en una base de datos centralizada sin acudir a ejecuciones en servidor).

A continuación se presenta la tabla 2 que resume los principales lenguajes para la programación de los sitios Web en el lado del cliente.

Lenguaje	Descripción
HTML	Hipertext Markup Language. Lenguaje que se utiliza para la creación de páginas Web. Consta de un conjunto de elementos, denominados etiquetas o marcas, que se utilizan para incluir texto, dibujos, tablas, cabeceras, etc.
DHTML	Extensiones de HTML que cubren el uso de hojas de estilo en cascada (CSS), uso de layers o capas para el posicionamiento y visibilidad de elementos en el navegador, programación con lenguajes de scripts y modelado de objetos de documentos (DOM).
JavaScript	Lenguaje interpretado incrustado en las páginas Web para permitir ejecutar código en el cliente a través del manejo de objetos y la captura de eventos producidos en el navegador.
VBScript	Lenguaje interpretado similar a JavaScript proporcionado por Microsoft.
Applets de Java	Pequeña aplicación accesible en un servidor Internet, que se transporta por la red, se instala automáticamente y se ejecuta como parte de un documento Web.
Actives	Tecnología de Microsoft que permite ejecutar un programa que siga este estándar en los equipos de los usuarios al ser vinculado como objeto en una página Web. Los programas ActiveX dependen del sistema operativo Windows, por lo tanto no funcionan fuera de este ambiente.

Tabla 2. Lenguajes del lado del Cliente.

En este proyecto serán utilizados los tres primeros lenguajes para la programación del lado del cliente.

2.1.3.2 PROGRAMACIÓN DEL LADO DEL SERVIDOR.

Se caracterizan porque los cálculos asociados a las páginas se realizan en el equipo servidor. De esta manera, los visualizadores de los clientes reciben únicamente instrucciones HTML. Por tanto, la complejidad del sistema se traslada a los equipos servidores, liberando de carga a los equipos de los usuarios.

Las principales situaciones en las que este mecanismo resulta adecuado son:

- ◆ Cuando se necesita acceder a información centralizada (habitualmente situada en bases de datos localizadas en el equipo servidor).
- ◆ Cuando los procesos que se ejecutan, por razones de seguridad, no se traspasan a los equipos de los usuarios (por ejemplo, la validación de fondos en una tarjeta bancaria).
- ◆ Cuando no se desea que los usuarios posean el código asociado a la página Web (por ejemplo, si un estudio de ingenieros de caminos realiza una página Web que resuelve, previo pago, problemas de cálculos de estructuras).
- ◆ Cuando se desea liberar de cálculos a los equipos de los clientes (por ejemplo, un sitio Web que genera animaciones tridimensionales y posee equipos especializados para este propósito).
- ◆ Cuando se desea estar seguro de que todos los clientes van a poder ver correctamente nuestras páginas Web, independientemente del tipo y versión de su visualizador y del equipo que posean.

Para el desarrollo de aplicaciones del lado del servidor existen 3 grandes metodologías, utilizar el protocolo CGI, utilizar una API provista por el Web-servidor o bien utilizar un “módulo” del Web servidor.

◆ El protocolo CGI:

El protocolo CGI (Common Gateway Interface) fue creado para establecer un protocolo estándar de comunicación entre el servidor Web y cualquier lenguaje de programación de forma tal que desde el lenguaje “x” puedan recibirse datos que el usuario envía usando el método “POST” o “GET” y además el resultado de la

aplicación sea enviado por el servidor al navegador. Típicamente para recibir datos se usa alguna biblioteca o módulo del lenguaje elegido que implementa el protocolo CGI y para enviar datos simplemente se envían a la salida estándar desde el lenguaje elegido y el servidor Web se encarga de redireccionar esto al navegador.

De esta forma pueden realizarse aplicaciones para un sitio Web en casi cualquier lenguaje. Los lenguajes interpretados rápidamente ganaron terreno ya que tienen un ciclo de desarrollo en tiempo inferior a los lenguajes compilados y son más fáciles de depurar dentro del ambiente CGI.

Los lenguajes no interpretados (C, C++) tienen como ventaja que requieren menos recursos del servidor al generarse el proceso CGI (no hace falta un intérprete) y además suelen ser más veloces en su ejecución (no se necesita interpretar nada), sin embargo el desarrollar y depurar suelen ser tareas complejas y no siempre se justifica el esfuerzo si la aplicación es pequeña.

La desventaja de las aplicaciones CGI consiste en que el servidor debe realizar un proceso, y ejecutar la aplicación o bien el intérprete de la aplicación, y este ciclo, que se cumple cada vez que se ejecuta la aplicación, CGI consume muchos recursos y en general es costoso en tiempo para el servidor.

Uso de una API del servidor:

Otra técnica factible consiste en utilizar una API (application programming interface) provista por el servidor Web para desarrollar aplicaciones, es decir que el servidor provee un lenguaje en el cual se pueden desarrollar aplicaciones. Este esquema es más eficiente que el anterior ya que el servidor Web es el encargado de ejecutar las aplicaciones en forma directa sin necesidad de crear un proceso. Las desventajas son sin embargo importantes: En primer lugar las aplicaciones creadas en este marco no son portables ya que sólo pueden ejecutarse en un servidor Web determinado, esto es una gran desventaja frente a las aplicaciones CGI que podían una vez desarrolladas ejecutarse en cualquier servidor. La segunda gran desventaja es que frecuentemente un error de programación de una aplicación podría ocasionar que el servidor deje de funcionar, genere un error, pierda memoria u otros problemas. Esto ocasiona que este tipo de aplicación no sea confiable.

Uso de un Módulo del Servidor Web

La tecnología más reciente para la ejecución de aplicaciones consiste en anexar a un servidor Web “módulos” que le permitan interpretar un determinado lenguaje. De esta forma se logra eficiencia ya que el servidor no necesita crear un nuevo proceso por cada aplicación que ejecuta. Las aplicaciones son portables ya que son desarrolladas en un lenguaje estándar que no depende del servidor Web. Las aplicaciones son confiables ya que si bien pueden producir un error en el lenguaje en que están diseñadas, si el módulo es sólido, dichos errores no pueden comprometer al servidor.

En las siguientes tablas (Tabla 3 y Tabla 4) se presenta un resumen de los principales lenguajes con una descripción de su uso en programación del lado del servidor así como una breve comparación de los mismos.

Lenguaje	Descripción
Perl	Practical Extraction and Report Language. Lenguaje interpretado creado con el objetivo principal de simplificar las tareas de administración de un sistema UNIX. Hoy en día se ha convertido en un lenguaje de propósito general.
Python	Es un lenguaje interpretado que permite escribir programas pequeños, utilizado en desarrollo web para la creación de CGI
C, C++	Utilizado para la creación de CGI
PHP	PHP es un lenguaje interpretado diseñado para el desarrollo de sitios dinámicos. La distribución más popular de PHP es como módulo para el servidor Apache, aunque puede funcionar como un interprete para ejecutar aplicaciones CGI.
ASP	Active Server Pages. Tecnología creada por Microsoft destinada a la creación de sitios web.
JSP	Java Server Pages (JSP). Es un lenguaje interpretado insertado en paginas web y basada en Java para el desarrollo de sitios dinámicos.
Mod_perl	Módulo de Perl para el servidor web
Mod_python	Módulo Python para el servidor web

Tabla 3. Lenguajes del lado del servidor.

	CGI (interpretado)	CGI (compilado)	API del servidor	Módulo del servidor
Ejemplos	Perl, Python	C, C++	Netscape Enterprise	PHP, ASP, JSP, Mod_perl, Mod_python, FastCGI
Tiempo de Desarrollo	Corto	Largo	Medio	Corto
Depuración	Sencilla	Compleja	Compleja	Sencilla
Confiabilidad	Alta	Alta	Baja	Alta
Eficiencia	Baja	Media	Alta	Alta

Tabla 4. Comparación de lenguajes del lado del servidor.

Para el desarrollo de este proyecto se ha escogido PHP para el desarrollo del lado del servidor por tanto a continuación se realiza una pequeña introducción a este lenguaje.

PHP (Hypertext Preprocessor)

PHP es un lenguaje de scripting Web de código fuente abierto. Es el acrónimo de Personal Home Page cuando fue creado en 1994 por Rasmus Lerdorf para hacer un seguimiento de los visitantes de su currículum online. Como su utilidad y sus capacidades empezaron a crecer (a la vez que empezaba a ser utilizado en más situaciones profesionales), su significado cambió por el de PHP: Hypertext Preprocessor (PHP: procesador de Hipertexto). Básicamente esta definición viene a significar que PHP manipula los datos antes de que estos se conviertan en HTML, el lenguaje de marcado de Hipertexto. El PHP es un lenguaje embebido en páginas HTML y que se ejecutan en el servidor. Productos similares y propietarios son Active Server Pages (ASP) de Microsoft, ColdFusion de Allaire y Java Server Pages (JSP) de Sun.

PHP es un lenguaje de Scripting del lado del servidor, de plataforma cruzada e incrustado en HTML. En primer lugar del lado del servidor significa que todo el código PHP se produce en el servidor (en oposición a cuando se refiere al lado del cliente que es el ordenador con el que se visualiza el sitio Web). El Servidor es simplemente el ordenador que alberga las páginas que se ven cuando se acude a una dirección desde un navegador. Cuando se declara que es de plataforma cruzada se está indicando que se puede utilizar en ordenadores que ejecutan Unix, Windows, Macintosh, OS/2 y otros sistemas operativos (se está hablando del sistema operativo del servidor y no del cliente). PHP no solo puede funcionar en casi cualquier sistema operativo, sino que, a diferencia de otros lenguajes de programación, puede trasladar su trabajo de una plataforma a otra con pocos o ningún cambio en absoluto. Como es lógico, también se podrá desarrollar PHP en cualquier sistema operativo, al igual que sucede con HTML.

PHP es fácil de aprender comparado con otros mecanismos para obtener la misma funcionalidad. A diferencia de JSP o CGI basados en C, PHP no requiere un conocimiento exhaustivo del lenguaje de programación. A diferencia de Perl, PHP tiene una sintaxis fácil de comprender y a diferencia de ASP, no requiere conocer más de un lenguaje de programación o de la instalación de módulos externos o comerciales

para realizar tareas más complicadas no previstas en el lenguaje más usado (Visual Basic Script).

La mayoría de las funciones más útiles están predefinidas en PHP:

- ◆ Acceso a bases de datos: ODBC, Oracle, Postgres, SQL Server, MySQL, Informix, Interbase, SyBase, mSQL, dBase entre otros.
- ◆ Conectividad: HTTP, FTP, COM, YP/NIS, SNMP, Sockets, CORBA, LDAP.
- ◆ Servicios Correo y Noticias: POP, IMAP, SMTP, NNTP.
- ◆ Manejo de Textos y Gráficos: XML, HTML, PDF, GD, Flash.
- ◆ Funciones Matemáticas.
- ◆ POSIX: semáforos, memoria compartida, acceso a ficheros, expresiones regulares, cronómetros, etc.
- ◆ Comercio Electrónico: Cybercash, Verisign.
- ◆ Formularios.
- ◆ Encriptación y Compresión: MD5, Gzip, Bzip2, OpenSSL, etc.



Figura 4. Muestra como se interpreta PHP.

Una página PHP es una página normal HTML que con unas marcas especiales le indican al servidor que deben interpretarse, como se muestra en la siguiente tabla.

Ejemplo.php (SERVIDOR)	Ejemplo.php (CLIENTE)
<pre> <html> <head> <title>Saludos</title> </head> <body> <p>Hola, <? \$nombre = "Felix"; \$apellido = "Galvan"; echo "soy \$nombre \$apellido"; ?> </p> </body> </html> </pre>	<pre> <html> <head> <title>Saludos</title> </head> <body> <p>Hola, soy Felix Galvan </p> </body> </HTML> </pre>

Tabla 5. Ejemplo de vista de un archivo de PHP en el servidor y en el cliente.

Cuando un cliente solicita esta página, el servidor Web la procesa en forma secuencial desde el principio al final buscando secciones PHP limitadas por `<? y ?>`. En caso de encontrarlas, las ejecuta. Si todo se ejecuta de forma normal, producirá la pagina HTML que se muestra en la parte derecha de la Tabla 5.

El resultado es idéntico a si hubiese sido escrito el texto manualmente. Esto tiene algunas consecuencias importantes:

- ◆ PHP puede ser agregado rápidamente al código HTML producido por editores HTML interactivos.
- ◆ PHP facilita la interacción entre diseñadores y programadores.
- ◆ No se necesita re-escribir cada línea de código HTML en un lenguaje de programación.
- ◆ PHP reduce costes y aumenta la eficiencia.

2.2 BASES DE DATOS.

2.2.1 MYSQL.

Su principal objetivo de diseño fue la VELOCIDAD. Se sacrificaron algunas características esenciales en sistemas más "serios" con este fin. Otra característica importante es que consume POCOS RECURSOS, tanto de CPU como de memoria. Posee licencia GPL a partir de la versión 3.23.19.

Las ventajas de esta base de datos son:

- ◆ Mayor rendimiento.
- ◆ Mayor velocidad tanto al conectar con el servidor como al servir selects y demás.
- ◆ Mejores utilidades de administración (backup, recuperación de errores, etc).
- ◆ Aunque se cuelgue, no suele perder información ni corromper los datos.
- ◆ Mejor integración con PHP.
- ◆ No hay límites en el tamaño de los registros.
- ◆ Mejor control de acceso, en el sentido de qué usuarios tienen acceso a qué tablas y con qué permisos.
- ◆ MySQL se comporta mejor que Postgres a la hora de modificar o añadir campos a una tabla "en caliente".

Los inconvenientes de esta base de datos son:

- ◆ No soporta transacciones, "roll-backs" ni subselects.
- ◆ No considera las claves ajenas.
- ◆ Ignora la integridad referencial, dejándola en manos del programador de la aplicación.

2.3 SOFTWARE DE APLICACIONES EDUCATIVAS.

Las aplicaciones software de carácter educativo han tomado un auge importante en las vidas de las personas, ya que se está utilizando gran cantidad de herramientas software para fortalecer el proceso de enseñanza-aprendizaje en cualquier tema factible de ser tratado bajo esta premisa.

El grupo SIMON se ha convertido en uno de los pioneros en el desarrollo de herramientas con fines educativos, conocidos son los MAC o micro mundos para el aprendizaje que se han enfatizado en el área de las ciencias naturales, dirigida a los estudiantes de escuelas y colegios otorgando un apoyo a los profesores en la enseñanza de esta área del saber.

Al centrarse en la Geometría Descriptiva se encuentra que existe una gran escasez en el desarrollo de herramientas software que cubran específicamente este campo de la ciencia, en la actualidad existen herramientas software que se encuentran en el mercado para el estudio de la Geometría, las cuales son de carácter matemático y no

tratan de manera clara la Geometría Descriptiva, por tanto se esta tratando un área poco desarrollada en el carácter computacional, porque al realizar la búsqueda de herramientas para este tema sólo se encuentran solicitudes de personas que desean tener este tipo de software.

Por tanto esta es una rama del saber que necesita ser explotada en el carácter computacional, y como ingenieros de sistemas estamos en la disponibilidad de llenar esa área con la creación de software de calidad que favorezcan el aprendizaje de la Geometría Descriptiva, Además se comenzaría a desarrollar el manejo de software educativo a nivel universitario, llevando la educación superior a tomar nuevas metodologías para la enseñanza de las áreas del conocimientos que abarca, dando un paso a lo que de seguro será la educación en un futuro, apoyada por herramientas computacionales.

2.4 SOLUCIONES BASADAS EN TECNOLOGIAS INTERNET.

Internet es la red de computadoras más grande del mundo. Esta red de redes mundial provee correo electrónico, noticias, acceso a control remoto, transferencia de archivos y otros servicios a nivel mundial. El Internet está basado en el protocolo TCP/IP el cual es un estándar de comunicación global.

Por otra parte Intranet es un término relacionado con la implementación del Internet y su tecnología dentro de la red interna de una compañía, término similar es el de Extranet el cual está relacionado con la implementación de la tecnología del Internet para beneficio exclusivo de comunicación entre negocios y en donde el público en general no tiene acceso. Este concepto involucra la extensión del Intranet para incluir usuarios que se encuentran fuera del firewall de una entidad.

Por lo tanto, la diferencia básica entre Intranet, Extranet e Internet radica en las restricciones de acceso de los usuarios y no en la tecnología empleada. Las Intranet y Extranet son posibles gracias a la difusión de las tecnologías Internet.

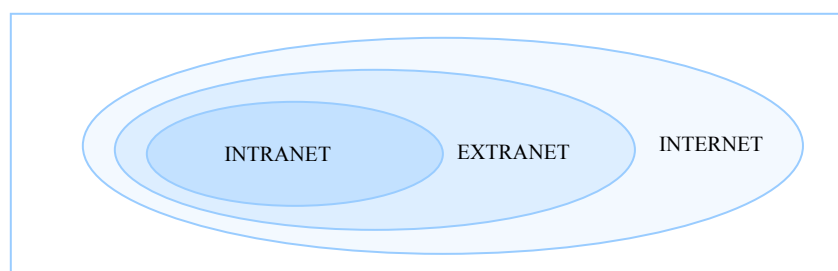


Figura 5. Alcance de Intranet, Extranet e Internet

Esto hace que en algunos casos las diferencias entre una Extranet y una Intranet corporativa sean mínimas e incluso se permita a los usuarios de la Extranet el acceso a gran parte de los servicios típicamente reservados para uso interno de una empresa.

2.4.1 INTERNET.

El desarrollo y la evolución de Internet ha sido significativo, convirtiéndose en un nuevo concepto de comunicación global. Inicialmente las organizaciones buscaron "estar presentes" en Internet, hoy pueden "vender y comprar", desde información hasta productos y servicios. Si una organización decide utilizar Internet, es necesario tomar en cuenta aspectos como el conocimiento de la tecnología Internet, el mercado o público objetivo, la tecnología del negocio: Fortalezas y debilidades, el valor agregado a ofrecer, la competencia, las alianzas, etc.

2.4.2 INTRANET.

Las intranets nacen de la posibilidad de aprovechar las ventajas de la Web para el funcionamiento interno de la empresa. Una Intranet se puede definir una red privada de ordenadores, que utiliza la tecnología, estándares y productos de Internet para compartir información entre ellos⁶. De aquí viene que Internet e Intranet se parezcan tanto sintéticamente pero que difieran semánticamente. Internet se resume en un conjunto de ordenadores interconectados a los que puede acceder cualquier usuario, mientras que Intranet es un concepto de red corporativa y por tanto privada.

La función de una Intranet es la de compartir datos y periféricos. La forma de tratar los periféricos en una Intranet no difiere de la utilizada en redes convencionales. Estos se deberán instalar en ordenadores accesibles para quien deba usarlos. Pero es en los

⁶ SIMINIANI, Mariano. Intranets, Empresa y Gestión Documental, Editorial McGraw Hill. p. 44

datos donde radican las mayores diferencias entre las redes de siempre y las Intranets.

En el caso de las redes convencionales, los datos se encuentran en un dispositivo de almacenamiento (discos duros, discos compactos, cintas, etc.) y son los usuarios quienes van a buscarlos. Cuando el usuario ya tiene los datos debe ser él quien los trate, ya sea con un programa o bien manualmente para obtener una información que le sea útil. En el caso de una Intranet el usuario siempre está tratando con la información de una forma directa y sencilla, mientras que un usuario de una red convencional debe tratar los datos de los que dispone para poder obtener una información.

Algunos de los beneficios que ofrece una red local son los siguientes:

- ◆ Optimiza la información unificándola y facilitando su tratamiento.

Una Intranet bien desarrollada, otorga permisos diferentes a cada empleado. Por eso, cuando un trabajador introduce información en la base de datos del servidor, éste ya sabe dónde y cómo debe publicar dicha información. Así pues, la información proveniente de un departamento se ubicará homogénea y ordenadamente en el espacio que le corresponda y quedará accesible para el resto de empleados con acceso a esa área.

- ◆ Acelera el paso de la gestión de la información a la gestión del conocimiento.

Cuando se posee una Intranet, la información se tiene a la mano, se evita la gestión física de documentos. En la Intranet es el trabajador el que se informa por voluntad propia para llevar a cabo correctamente la tarea que se le encomienda.

- ◆ Es una herramienta de grupo.

La Intranet permite que se erradiquen las típicas reuniones informativas que suponen un gasto de tiempo enorme para la empresa. Los foros, las carteleras de anuncios estratégicamente colocados en la Intranet son realmente útiles para llevar a cabo la tarea.

- ◆ Ahorro en la empresa.

Con una Intranet se disminuye drásticamente el uso de papel en la empresa. Los informes, los contactos, las notas, los memorandos etc. se leen en la pantalla del

ordenador. Del mismo modo, se ahorra en teléfono. Dado que la mayor parte de las empresas operan con una tarifa plana profesional, los costes en llamadas de teléfono (que ahora van a ser suplantadas por el e-mail, más consistente que el mensaje oral).

◆ Información actualizada, reciente.

La información actualizada resulta sumamente atrayente y culturalmente se entiende como signo de eficacia: constantemente, los directivos repiten en la prensa que, cada día, dedican un tiempo a la lectura de noticias porque les ayuda a mejorar en su trabajo. Así pues, nos esforzamos en estar al corriente de las últimas novedades. Y de eso se debe aprovechar la organización: si consigue que los trabajadores vean en la Intranet una fuente fiable y actualizada de información, disminuirán en buena medida los rumores.

◆ La Intranet es escalable.

Una Intranet no es un sistema inamovible. La red interna ya no es una maraña de hubs e hilos conductores en un local, sino una Red basada en el protocolo TCP/IP y que goza de las mismas ventajas: conectividad, apertura del sistema...y escalabilidad. Por eso la Intranet que crece más allá de la propia empresa se puede convertir en extranet, por eso las nuevas redes internas no tienen fronteras, por eso son enteramente modificables, rápidas y ajustables a las necesidades puntuales de la organización.

2.4.3 EXTRANET.

Una extranet es una solución Internet que permite el intercambio de información de manera permanente, confidencial y segura con personas fuera de la organización como socios de negocios, clientes, proveedores, etc y que posee las siguientes características generales:

Es un sistema de acceso restringido basado en tecnologías Internet. La comunicación se da solo entre la empresa (dueña del sistema) y su entorno. Las otras empresas usuarias de la Extranet no necesariamente se comunican entre sí.

Como se dijo anteriormente la única diferencia significativa entre una Extranet y una Intranet es que en esta última generalmente sólo acceden empleados de la empresa, mientras que a una Extranet se permite el acceso tanto a empleados como a clientes,

proveedores, colaboradores, etc. Por tanto se puede pensar en una Extranet como una extensión de la Intranet a usuarios externos autorizados.

2.5 ARQUITECTURA DE SOFTWARE.

La arquitectura de un sistema es la vista conceptual de la estructura de este. Toda aplicación contiene código de presentación, código de procesamiento de datos y código de almacenamiento de datos. Por tanto su arquitectura se define según como este distribuido este código.

En los últimos años la arquitectura de los sistemas de software desarrollados ha evolucionado junto con el resto de la tecnología informática. Hace pocos años, cuando los mainframes eran una novedad, se desarrollaban sistemas monolíticos en donde un sistema no conocía nada más allá de su entorno de operación.

Cuando la tecnología de redes aparece y se difunde, la industria no tardó en darse cuenta de la ventaja de desarrollar sistemas que contaran con la capacidad de interactuar con otros sistemas residentes en otras máquinas dentro de la red. De aquí surge el modelo Cliente-Servidor donde un "Cliente" solicita servicios de un "Servidor" el cual gestiona las solicitudes de varios clientes a la vez.

Luego llegó el auge de Internet y el desarrollo de nuevas tecnologías para software por componentes, estas se puede construir una aplicación distribuida que reside en uno o más servidores en la red y además se disminuyó la necesidad de software cliente ya que generalmente se utiliza uno ya estandarizado: los navegadores de Internet tales como Netscape o Internet Explorer, entre otros.

Las organizaciones están aprovechando estas tecnologías que permiten a sus usuario el tener acceso sencillo y casi universal a sus aplicaciones corporativas sin la necesidad de incurrir en todos los gastos de mantenimiento que conlleva el modelo cliente-servidor como fue planteado en sus inicios.

Para que todo esto funcione, se requiere de incrementar lógica de programación del lado del servidor ya que la funcionalidad del cliente es mínima. Aquí es donde surge el concepto de sistema multicapa o de "n" capas como una metodología para el diseño de sistemas distribuidos. Esta arquitectura se hizo popular a principios de los años 90

y en la actualidad se ha afianzado como la arquitectura de software de aplicación empresarial.

2.5.1 CLIENTE – SERVIDOR.

Esta es una versión simplificada de la arquitectura en capas, donde la capa de aplicación se encuentra repartida entre el cliente y el servidor. Los sistemas basados en la arquitectura cliente/servidor están formados por dos partes lógicas o capas: un servidor que proporciona servicios, y un cliente que solicita servicios del servidor o servidores. Los dos, juntos, forman un sistema de computación completo con una clara división de responsabilidades.

Un cliente es un proceso que envía un mensaje a un proceso servidor, solicitando que realice determinada tarea. Los procesos cliente normalmente gestionan la porción de interfaz de usuario de la aplicación, validan los datos introducidos por el usuario, realizan las solicitudes a los servidores y, a veces, ejecutan cierta lógica de negocio.

Un servidor es un proceso que contesta a la solicitud del cliente realizando la tarea propuesta por éste. Los clientes gestionan recursos compartidos como ficheros, impresoras, enlaces de comunicación, bases de datos, etc.

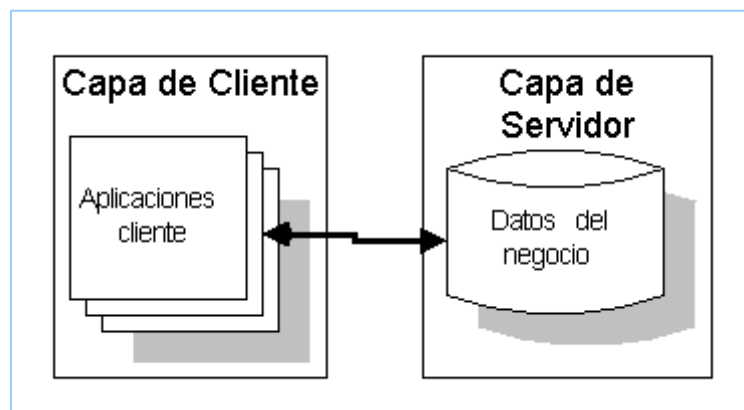


Figura 6. Arquitectura Cliente - Servidor.

2.5.2 MÚLTIPLES CAPAS.

Toda aplicación contiene código de presentación, código de procesamiento de datos y código de almacenamiento de datos, la arquitectura de las Aplicaciones difiere según como esta distribuido este código.

Una arquitectura multicapa consiste en dividir la funcionalidad del sistema total en capas lógicas que pueden ser encapsuladas como componentes que interactúan entre ellos a alto nivel y supone una mayor escalabilidad de la aplicación, un mantenimiento menor y un incremento de la reutilización de componentes.

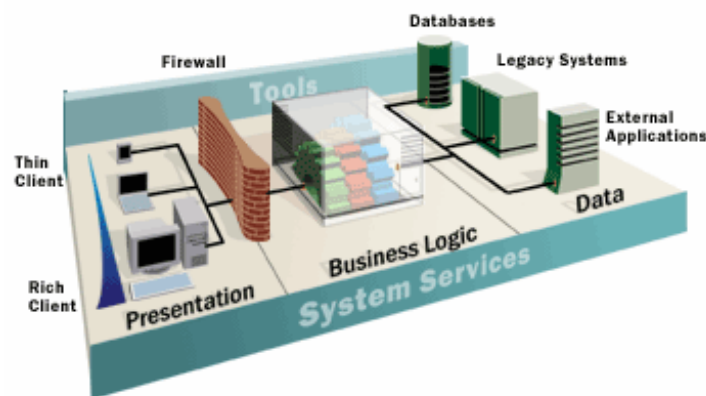


Figura 7. Arquitectura de Tres capas. Caso mas general de arquitecturas multicapas.

El número de capas típico que se utiliza en esta arquitectura son tres, capa de datos, de negocio y de presentación. Así en la capa de datos encapsulamos todas las funciones de base de datos, las funciones de interfaz de usuario en la capa de "Presentación" y finalmente toda la lógica de operación en la capa de "Negocios".

Esta encapsulación tiene como objeto ofrecer una colección más simple de funciones que permiten desarrollar aplicaciones a alta velocidad y bajo costo. El costo de mantenimiento del sistema disminuye notablemente ya que una modificación en una capa no debe afectar a las demás.

2.5.2.1 CAPA DE DATOS.

El nivel de servicios de datos es responsable de: Almacenar, recuperar y mantener los datos así como verificar la integridad de los mismos.

Los servicios de datos tienen una variedad de formas y tamaños, incluyendo los sistemas de administración de bases de datos relacionales, servidores de correo electrónico y sistemas de archivos.

2.5.2.2 CAPA DE APLICACIÓN O DE NEGOCIOS.

La capa de aplicación es el “puente” entre un usuario, representado en la capa de presentación, y los servicios de datos, que se encuentran en la capa de datos. Los servicios de esta capa responden a peticiones del usuario (u otros servicios de negocios) para ejecutar una tarea de este tipo. Cumplen con esto aplicando procedimientos formales y reglas de negocio a los datos relevantes. Esto aísla al usuario de la interacción directa con la base de datos.

El nivel de servicios de negocios es responsable de: Recibir la entrada del nivel de presentación, interactuar con los servicios de datos para ejecutar las operaciones de negocios para los que la aplicación fue diseñada a automatizar (por ejemplo, la preparación de puestos por ingresos, el procesamiento de ordenes y así sucesivamente) y de enviar el resultado procesado al nivel de presentación.

2.5.2.3 CAPA DE PRESENTACIÓN.

Los servicios de presentación proporcionan la interfaz necesaria para presentar información y reunir datos. También aseguran los servicios de negocios necesarios para ofrecer las capacidades de transacciones requeridas e integrar al usuario con la aplicación para ejecutar un proceso de negocios.

Los servicios de presentación generalmente son identificados con la interfaz de usuario, y normalmente residen en un programa ejecutable localizado en la estación de trabajo del usuario final. El cliente proporciona el contexto de presentación, generalmente un navegador como Microsoft Internet Explorer o Netscape, que permite ver los datos remotos a través de una capa de presentación HTML, u otro tipo de aplicación.

Mediante el uso de capas, se separa la programación que da acceso a los datos en las bases de datos de otros contenidos del sistema. Esto ayuda a asegurar que los desarrolladores estén libres para enfocarse en escribir su lógica de negocios en

componentes sin preocuparse acerca de cómo se muestra la salida. Recíprocamente, esto da libertad a los diseñadores de usar herramientas familiares para modificar la interfaz.

La capa de servicios de presentación es responsable de: Obtener información del usuario, enviar la información del usuario a los servicios de negocios para su procesamiento, recibir los resultados del procesamiento de los servicios de negocios y presentar estos resultados al usuario.

Aunque el diseño de las interfaces depende exclusivamente de quien lo hace y hacia quien va dirigido, existen ciertos lineamientos básicos a tener en cuenta que determinan que tan utilizable va a ser el sistema, por ejemplo, si la interfaz de usuario es pobre o no tiene estándares claros, el usuario no se sentirá cómodo al usar la aplicación, o peor aún, dejará de usarla.

Capítulo 3.

MARCO METODOLOGICO

Para construir software de alta calidad es necesario definir una metodología o proceso⁷ de desarrollo de software como marco de trabajo que contenga cada una de las tareas a realizar que permitan entregar un producto software altamente fiable, robusto y eficaz, entre otras características.

Los estilos de desarrollo del producto varían entre las diferentes clases de proyectos, necesitando diferentes clases de tareas y distintos órdenes de las mismas. La elección equivocada de un modelo de ciclo de vida puede ocasionar omisión de tareas o una secuenciación inapropiada de las mismas, lo cual va en contra de la planificación y eficiencia del proyecto, una buena elección por el contrario, es una garantía de que el esfuerzo se utiliza eficientemente.

La función principal de un modelo de ciclo de vida es establecer el orden en el que se especifica, se realizan los prototipos, se diseña, se implementa, revisa, prueba y se realizan otras actividades en un proyecto. Establece los criterios que se utilizan para determinar el paso de una actividad a otra⁸.

Existen varios modelos con diferentes enfoques, cada uno con sus ventajas e inconvenientes. Primeramente se describirán los modelos mas importantes, ya sea por su uso extendido o por su aporte a una nueva concepción del desarrollo de software, incluyendo al final una tabla comparativa entre ellos (Tabla 6. Resumen de los principales Modelos de Ciclo de Vida de Desarrollo del Software). Luego se presentará una guía para la elección del paradigma mas adecuado para este proyecto.

Por ultimo se hablara de la metodología utilizada, el Proceso Unificado de desarrollo de software, los modelos seleccionados para el desarrollo de la herramienta propuesta y se presentan los conceptos relacionados con este modelo, así como las fases y actividades que intervienen en el mismo y los productos resultantes durante la aplicación de este proceso.

⁷ Define quién está haciendo qué, cuándo y cómo alcanzar un determinado objetivo. En la Ingeniería del Software el objetivo es construir un producto software o mejorar uno existente [1]

⁸ McConnell, Steve. Desarrollo y Gestión de Proyectos Informáticos. Primera Edición. McGraw-Hill. España, 1997. Pág. 146.

3.1 CICLO DE VIDA DEL DESARROLLO SOFTWARE.

3.1.1 CASCADAS

Se caracteriza por:

- ◆ Es ideal para una versión de mantenimiento de un producto existente o migración de un producto existente a una nueva plataforma.
- ◆ No proporciona resultados tangibles en forma de software hasta el final del ciclo de vida.
- ◆ Funciona correctamente cuando los requerimientos de calidad dominan sobre los de costes y planificación.
- ◆ No permite flexibilidad en los cambios, es decir, no permite volver atrás para corregir errores.
- ◆ Es un modelo poco apropiado para un proyecto de desarrollo rápido.
- ◆ Requiere definición completa de los requerimientos antes de comenzar el diseño de la arquitectura.
- ◆ Existe revisión al final de cada etapa.
- ◆ Las etapas son discontinuas, es decir, no se solapan.

3.1.1.1 CASCADA PURA.

En el modelo de cascada pura un proyecto progresa a través de una secuencia ordenada de pasos llamadas etapas, las cuales van desde el concepto inicial de software hasta la prueba del sistema. Sus etapas no se solapan y una vez iniciada una de ellas, es difícil devolverse a la anterior. Al final de cada etapa se realiza una evaluación para determinar si se puede pasar a la siguiente o se continúa en la actual hasta su cumplimiento.

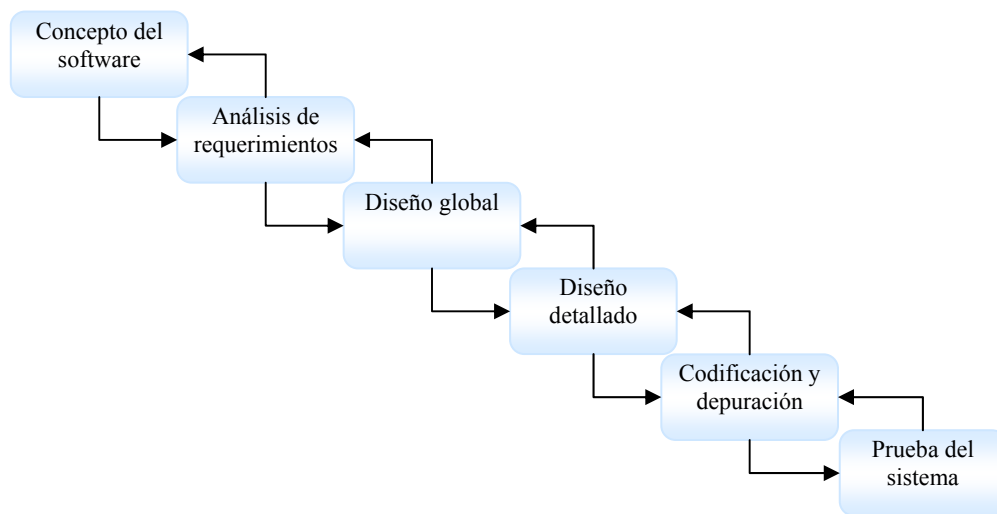


Figura 8. Modelo de Cascada pura.

El modelo en cascada no proporciona resultados tangibles en forma de software hasta que finaliza su ciclo. Sin embargo cuando se está familiarizado con el modelo, la documentación generada en cada etapa proporciona suficientes indicaciones del progreso del proyecto. Es por esto que se dice que “el modelo en cascada está dirigido por documentos; es decir, los productos principales del trabajo que se pasan de etapa en etapa son documentos”⁹.

3.1.1.2 CASCADAS MODIFICADAS.

La mayoría de los inconvenientes del modelo de cascada pura son ocasionados por el tratamiento secuencial de sus etapas, sin permitir que una de ellas pueda iniciarse sin terminar la anterior, es decir, no permite etapas solapadas en su modelo. Las siguientes variaciones del método buscan solucionar este problema introduciendo pequeñas modificaciones al mismo.

La primera variación es llamada Cascada con Fases Solapadas la cual sugiere un mayor grado de solapamiento. Por ejemplo, se debería tener el diseño global y parte del detallado antes de considerar completo el análisis de requerimientos. El problema con este modelo es que debido al solapamiento entre sus capas, los hitos¹⁰ son más ambiguos, haciendo más difícil la gestión del progreso. Además la realización de

⁹ McConnell, Steve. Desarrollo y Gestión de Proyectos Informáticos. Primera Edición. McGraw-Hill. España, 1997. Pág. 148.

¹⁰ Un hito es un punto de referencia que marca un acontecimiento importante en un proyecto y que se utiliza para controlar el progreso del mismo.

actividades en paralelo unido a una mala comunicación puede llevar a suposiciones incorrectas e ineficacia en el desarrollo. Por lo tanto esta variación es aplicable básicamente, a proyectos pequeños y bien definidos.



Figura 9. Modelo de Cascada con Fases Solapadas.

La segunda variación es Cascada con subproyectos. En ella, se trata de dividir el sistema en subsistemas lógicamente independientes que se puedan tratar como proyectos separados y puedan proseguir a su propio ritmo. Esto evitaría el problema del modelo cascada pura que debe esperar a concluir la etapa de diseño detallado de todas las partes del sistema antes de poder empezar a implementar. El problema principal de este nuevo enfoque es la presencia de interdependencias imprevistas entre los módulos.

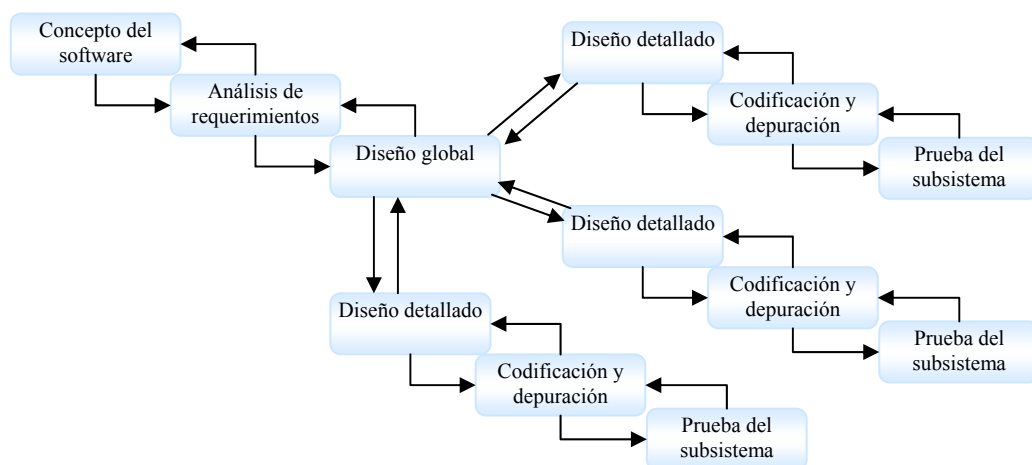


Figura 10. Modelo de Cascada con subproyectos.

La tercera modificación es llamada Cascada con Reducción de Riesgos, la cual elimina el problema de tener que comprender la totalidad de requerimientos antes de continuar con el diseño de la arquitectura. Modificando el modelo de cascada pura se puede colocar una espiral antes de iniciar sus etapas, para reducir el riesgo de los requerimientos. Durante esta espiral se controlan los riesgos principalmente del análisis de requerimientos y diseño de la arquitectura a través de prototipos de interfaz de usuario o de un sistema antiguo donde el usuario pueda exponer y aclarar sus requerimientos. Esta modificación se usa cuando el producto depende del desarrollo de un núcleo de alto riesgo para el sistema.

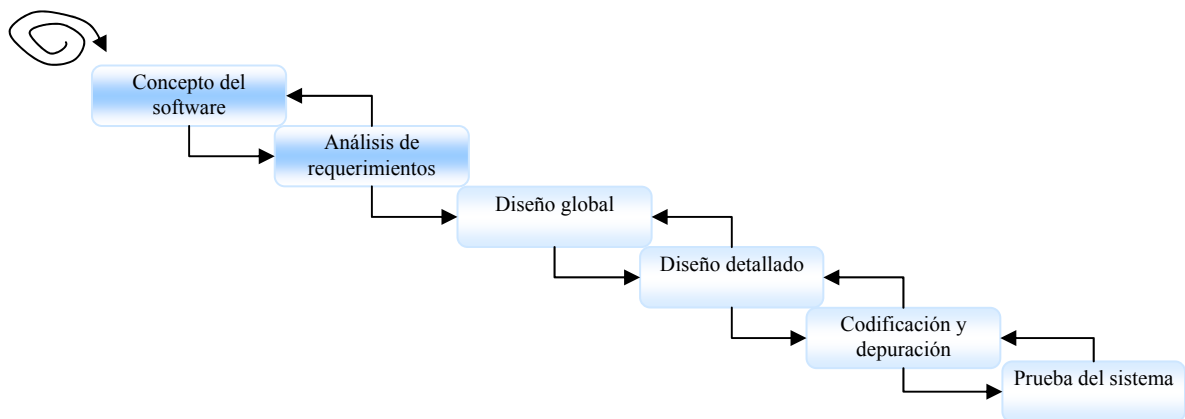


Figura 11. Modelo de Cascada con Reducción de riesgos.

3.1.2 DESARROLLO RAPIDO DE APLICACIONES.

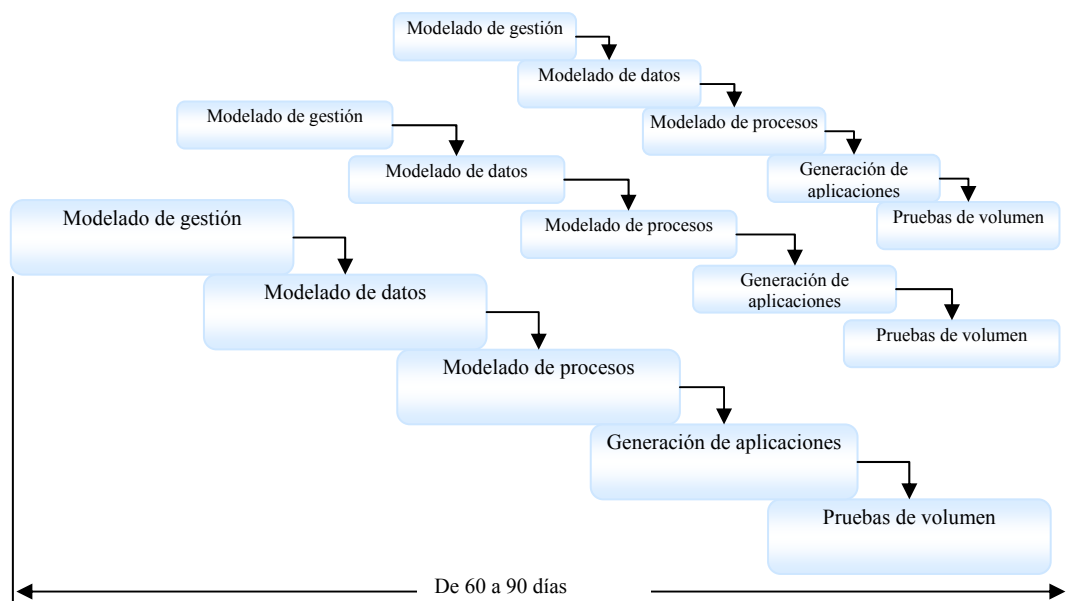


Figura 12. Modelo de Desarrollo de Aplicaciones Rápidas (DRA).

El modelo DRA es una adaptación a alta velocidad del modelo en cascada en el que se logra el desarrollo rápido utilizando un enfoque de construcción basado en componentes reutilizables y utilizando técnicas de cuarta generación en lugar de software con lenguajes de programación de tercera generación. En la siguiente figura se muestran las fases que comprende este modelo.

En proyectos donde se comprenden bien los requisitos, se limita correctamente el ámbito del proyecto, y el sistema se puede dividir en módulos, el proceso DRA permite al equipo de desarrollo crear un sistema completamente funcional dentro de periodos cortos de tiempo (60-90 días).

Se caracteriza por:

- ◆ DRA necesita que el sistema se pueda modularizar adecuadamente.
- ◆ DRA exige que los riesgos técnicos no sean altos, esto ocurre cuando se hace uso de tecnologías nuevas o cuando el software requiere un alto grado de interoperatividad con programas de computadora ya existentes.
- ◆ Requiere clientes y desarrolladores comprometidos en las rápidas actividades necesarias para completar un sistema en un marco de tiempo abreviado.

3.1.3 PROTOTIPADO.

Se caracteriza por:

- ◆ Se utiliza cuando los requerimientos cambian con rapidez, cuando el cliente es reacio a especificar el conjunto de requerimientos o cuando no se identifica el área de aplicación.
- ◆ Es útil cuando los desarrolladores no están seguros de la arquitectura o los algoritmos adecuados a utilizar.
- ◆ Proporciona al cliente signos visibles de progreso.
- ◆ Proporciona un alto grado de flexibilidad en respuesta a la realimentación del cliente y poco control por parte de la directiva.
- ◆ Imposibilidad de conocer al comienzo del proyecto lo que se tardará en crear un producto aceptable, no se sabe cuántas iteraciones se tendrán que realizar.
- ◆ El énfasis inicial se encuentra en los aspectos más visibles del sistema, después se vuelve atrás y se completan los huecos de las bases del sistema.

3.1.3.1 PROTOTIPADO SIMPLE

En este modelo, también llamado prototipo desechable, se construyen prototipos o acercamientos al sistema que sirven para identificar los requisitos del cliente. El prototipo se convierte en el primer sistema después de varias iteraciones cuando se estabilizan los requisitos. Sin embargo, este sistema debe ser nuevamente reconstruido dado que en su construcción no se tienen en cuenta aspectos de calidad y facilidad de mantenimiento.

El paradigma de construcción de prototipos comienza con la recolección de requisitos conocidos, se realiza un diseño rápido centrado en los aspectos visibles para el cliente y se construye el primer prototipo. Este será evaluado por el cliente y se utilizará para refinar los requisitos del software a desarrollar. Se construye un nuevo prototipo y se continúa esta secuencia hasta que se satisfacen las necesidades del cliente y el desarrollador comprende lo que necesita hacer.

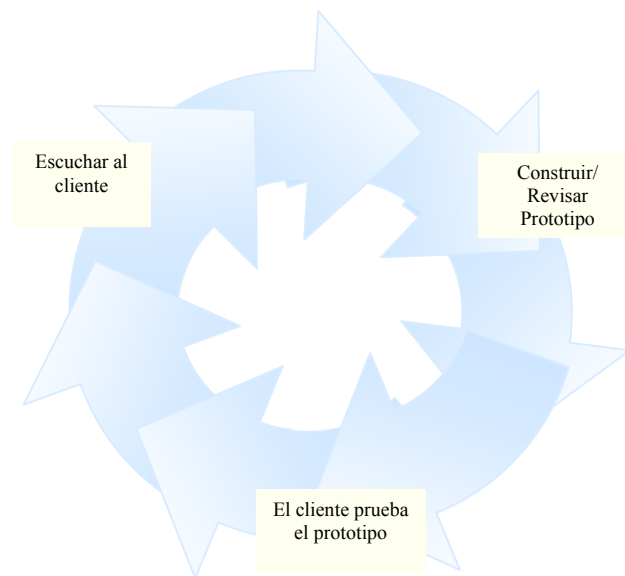


Figura 13. Modelo de Prototipo simple.

Para el desarrollo de este modelo se debe tener presente que este será solo un sistema inicial y el cliente debe estar consiente de esto. Este modelo es útil cuando no se definen los requisitos fácilmente. Además cuando se necesita probar la eficacia de un algoritmo, la capacidad de adaptación de un sistema operativo, o la forma como debería ser la interfaz de usuario.

3.1.3.2 PROTOTIPADO EVOLUTIVO

El prototipado evolutivo es un modelo que toma sus bases del prototipado desechable pero posee mayores controles sobre la calidad y desarrolla primero las áreas de mayor riesgo del sistema, de tal forma que el prototipo pueda ser tomado como producto final una vez se llegue a su fin, es decir, en este modelo se desarrolla el concepto del sistema a medida que avanza el proyecto. El prototipo evolutivo es un enfoque donde se desarrolla primero las partes seleccionadas del sistema y luego el resto a partir de estas partes. A diferencia de otros tipos de prototipado, en el evolutivo no se descarta el código del prototipo; lo transforma en el código entregado finalmente. El desarrollo de prototipos continúa hasta que se decide que el prototipo es lo suficientemente bueno y se puede entregar como producto final.

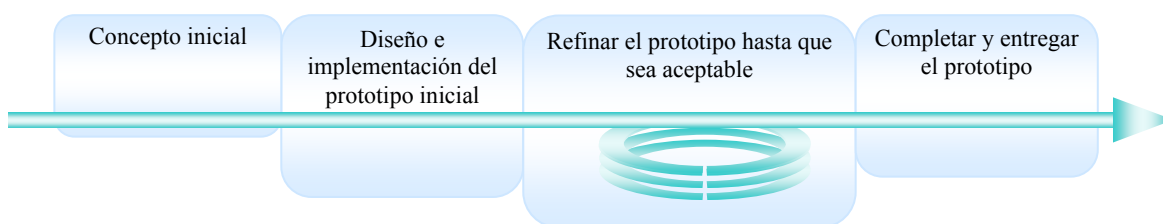


Figura 14. Modelo de Prototipo Evolutivo.

Este modelo genera signos visibles de progreso, pero se puede correr el riesgo de caer en el esquema de codificar y corregir; sin ninguna planificación, ni gestión.

3.1.4 ENTREGA POR ETAPAS

En este modelo no se entrega el producto total al final del proyecto sino que se muestra al cliente en etapas refinadas sucesivamente proporcionando una funcionalidad útil antes de entregar el 100% del proyecto. Primero se realiza la definición del concepto del software, el análisis de requerimientos y la creación del diseño global de una arquitectura como en el modelo cascada. A continuación se procede a realizar el diseño detallado, la codificación, depuración y prueba dentro de cada etapa.

Este modelo no funciona sin una planificación adecuada tanto para niveles técnicos como para niveles de gestión. En un nivel de gestión, se debe asegurar que las etapas

que se planifican son significativas para el cliente y que el trabajo se distribuye entre el personal del proyecto de tal forma que pueden completar su trabajo a tiempo. En un nivel técnico, hay que asegurarse de que se han tenido en cuenta todas las dependencias técnicas entre diferentes componentes de un producto para evitar retrasos por dependencias no previstas.

Se caracteriza por:

- ◆ Se conoce exactamente que es lo que se va a construir cuando se proceda a construirlo.
- ◆ El software se entrega por etapas sucesivas a lo largo del proyecto.
- ◆ Necesita una planificación adecuada tanto para niveles técnicos como para niveles de gestión.
- ◆ Se planifica para adaptarse a pocas solicitudes de modificación.
- ◆ El énfasis inicial se pone en el núcleo del sistema, que está constituido por funciones de bajo nivel que probablemente no van a ser modificadas por la realimentación del cliente.

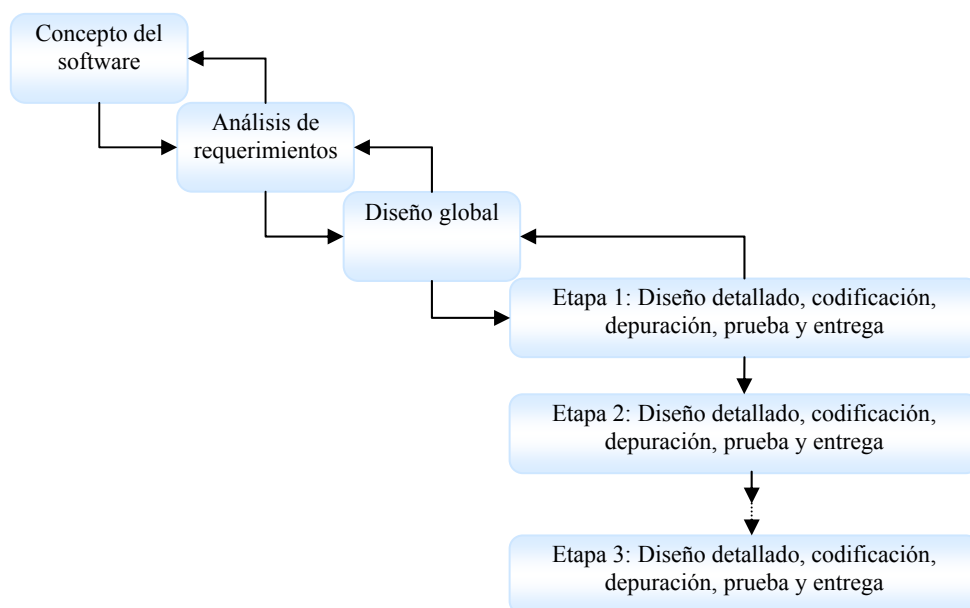


Figura 15. Modelo de Entrega por etapas.

3.1.5 ESPIRAL.

“El modelo de espiral es un modelo de ciclo de vida orientado a riesgos que divide un proyecto software en miniproyectos. Cada miniproyecto se centra en uno o más riesgos importantes hasta que todos éstos estén controlados.”¹¹. El concepto “riesgo” puede referirse a requerimientos o arquitecturas poco comprensibles, problemas de ejecución importantes o con la tecnología subyacente.

Se caracteriza por:

- ◆ Divide un proyecto software en miniproyectos.
- ◆ Cada miniproyecto se centra en uno o más riesgos importantes hasta que todos estén controlados.
- ◆ Los costos suben, los riesgos disminuyen.
- ◆ Desventaja: requiere de una gestión concienzuda, atenta y que exige conocimientos profundos.
- ◆ Cada iteración lleva consigo 6 pasos que se muestran en negrilla en los bordes exteriores de la figura que se muestra abajo:
- ◆ Determinar objetivos, alternativas y límites.
- ◆ Identificar y resolver riesgos.
- ◆ Evaluar las alternativas.
- ◆ Generar las entregas en cada iteración y comprobar que sean correctas.
- ◆ Planificar la siguiente iteración.
- ◆ Establecer un enfoque para siguiente iteración (si se decide ejecutarla)

¹¹ McConnell, Steve. Desarrollo y Gestión de Proyectos Informáticos. Primera Edición. McGraw-Hill. España, 1997. Pág. 153.

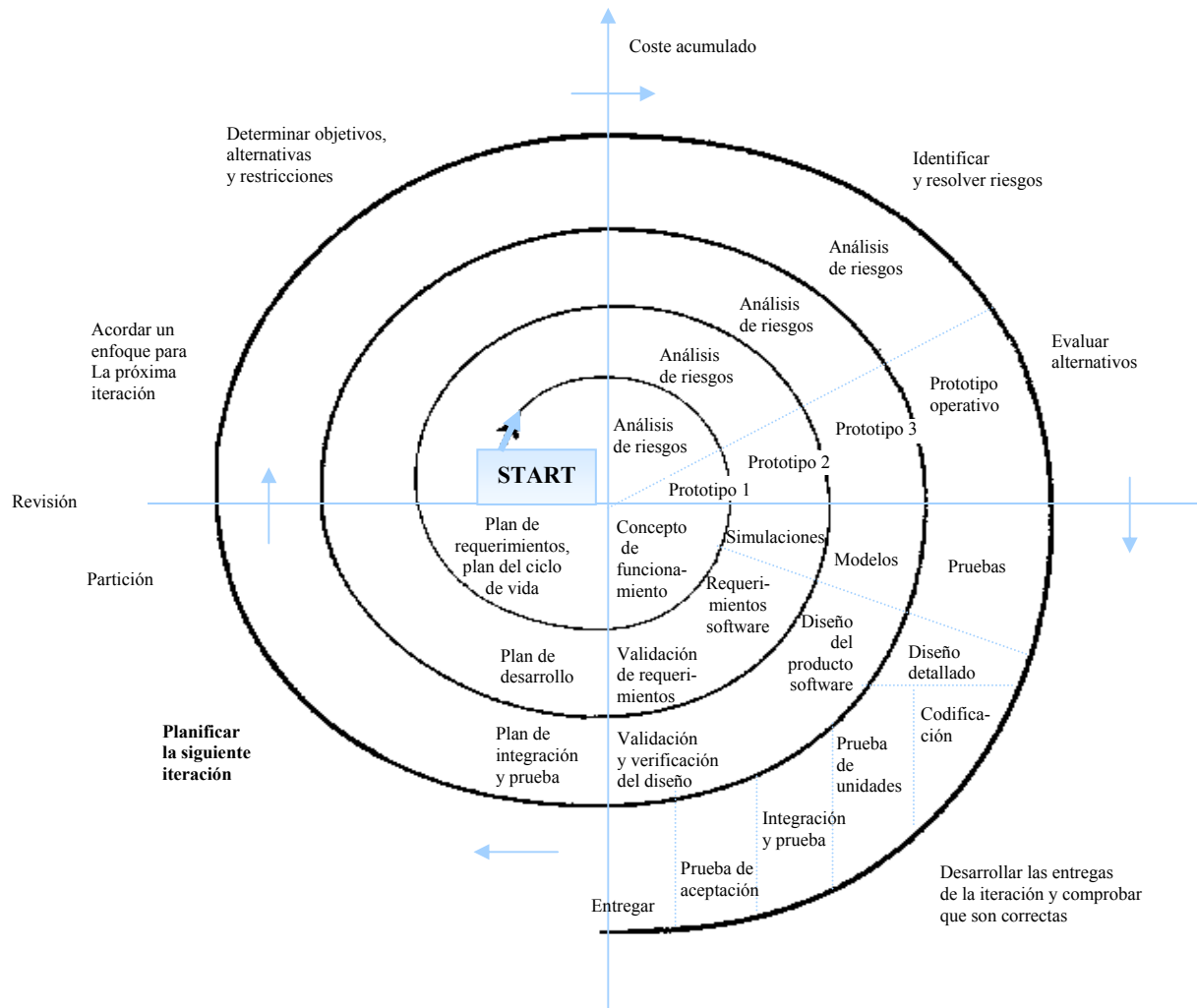


Figura 16. Modelo de Espiral.

El método parte de una escala pequeña en medio de la espiral, se localizan los riesgos, se genera un plan para manejarlos, y a continuación se establece una aproximación a la siguiente iteración. Después de controlar todos los riesgos más importantes, el modelo en espiral finaliza con un ciclo de vida en cascada, con prototipado o con otro modelo. También se pueden incorporar otros modelos de ciclo de vida en sus iteraciones.

Este método requiere una mayor planificación y un mayor seguimiento del proyecto, pero a cambio se obtiene una gran mejora en la visibilidad del progreso y en la reducción del riesgo.

3.1.6 PROCESO UNIFICADO.

El proceso unificado está basado en componentes y se sostiene sobre tres ideas básicas: casos de uso, arquitectura y desarrollo iterativo e incremental. Para hacer que estas ideas funcionen, se necesita un proceso, que tenga en cuenta ciclos, fases, flujos de trabajo, gestión del riesgo, control de calidad, gestión de proyecto y control de la configuración.

Un desarrollo iterativo, guiado por los casos de uso y centrado en la arquitectura, construye un software mediante pequeños incrementos, y añade cada incremento a la acumulación previa de incrementos de tal forma que siempre se tenga una construcción ejecutable. La arquitectura proporciona la estructura sobre la cual guiar las iteraciones mientras que los casos de uso definen los objetivos y dirigen el trabajo de cada iteración.

De esta manera el proceso reduce el riesgo de grandes retrasos en la entrega de un producto, se fijan metas más inmediatas por lo cual se puede controlar mejor el avance del proyecto.

El Proceso Unificado divide el proceso de desarrollo en ciclos, donde se obtiene una nueva versión del producto al final de cada ciclo. Cada ciclo se divide en cuatro Fases: Inicio, Elaboración, Construcción, y Transición. Cada una de estas fases concluye con un hito bien definido donde deben tomarse decisiones respecto al proyecto como la reestructuración del cronograma de trabajo. Cada una de estas fases se divide a su vez en iteraciones.

Cada iteración sigue la estructura de un pequeño ciclo de vida en cascada, pasando a través de los cinco flujos de trabajo fundamentales: requisitos, análisis, diseño, implementación y prueba. En la iteración también incluye la planificación que precede a los flujos de trabajo y la evaluación que va detrás de ellos.

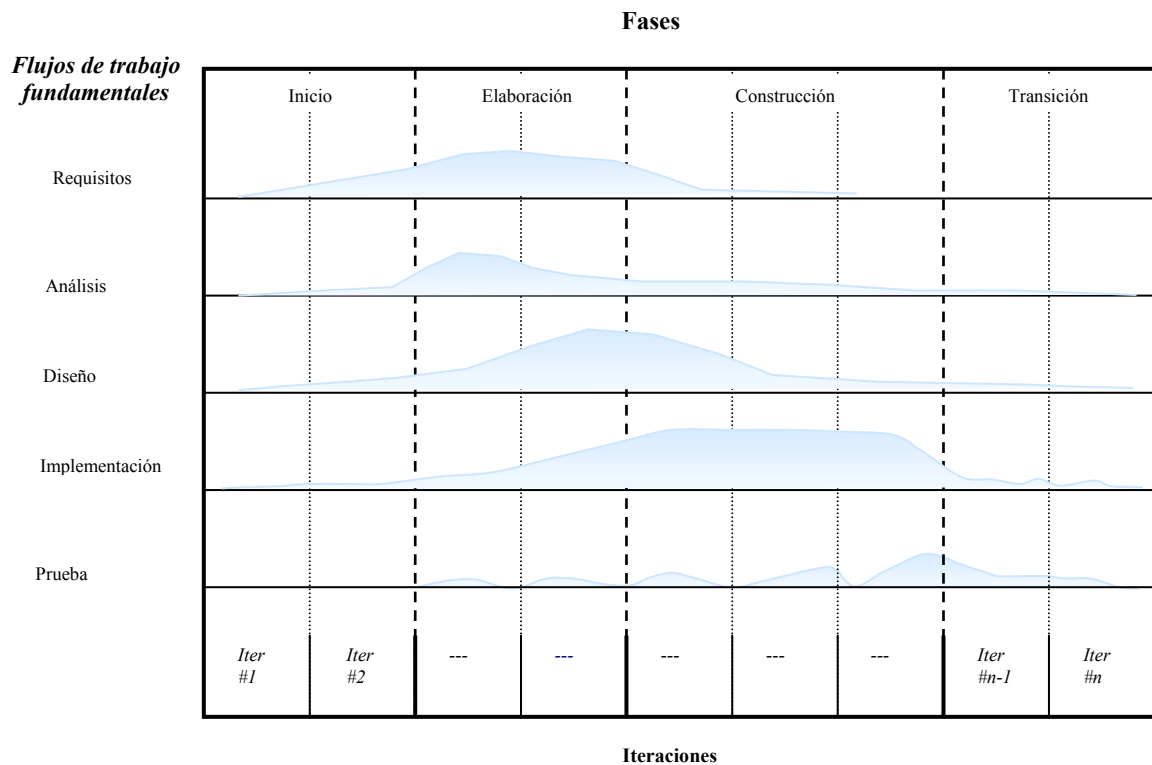


Figura 17. Modelo de Proceso Unificado.

Después de haber realizado la descripción de los diferentes modelos de desarrollo de software considerados como los más importantes, es necesario la creación de una tabla que contenga una descripción de todos ellos, mostrando factores importantes como sus debilidades y ventajas, entre otras características, haciendo más fácil su comprensión, y por ello mostrando algunas de las razones por la cual se escogió la metodología del proceso unificado.

	SE UTILIZA CUANDO	APLICACIÓN DE OTROS MODELOS	VENTAJAS	CONTACTO CON EL CLIENTE	DETECCIÓN DE ERRORES	PROBLEMAS Y RIESGOS
CASCADA PURA	<ul style="list-style-type: none"> • los requerimientos se pueden definir fácil y rápidamente a partir del conocimiento del área en estudio. • Se tiene una definición estable del producto, con variaciones mínimas de los requisitos. • Se trabaja con tecnologías conocidas • Se Construye una versión de mantenimiento de un producto. • Se migra un producto a una nueva plataforma. • Los requerimientos de calidad dominan sobre los de costes de planificación • Se dispone de personal inexperto en el manejo de una metodología en particular. 	Es una plantilla para otros métodos	<ul style="list-style-type: none"> • Presenta el proyecto con una estructura que ayuda a minimizar el esfuerzo inútil. • Ayuda a localizar errores en las primeras etapas del proyecto a un bajo costo (Si se tiene una definición estable del producto) • Permite enfrentarse a la complejidad de forma ordenada (En Proyectos complejos comprendidos totalmente) 	El cliente tiene contacto con el producto solo hasta el final del desarrollo.	Las pruebas para detección de errores se hacen al final del ciclo.	<ul style="list-style-type: none"> • No proporciona resultados de software hasta el final del ciclo • El cliente tiene contacto con el producto solo hasta el final del desarrollo. • Dificultad para especificar exactamente los requerimientos al comienzo del proyecto. • No permite flexibilidad en los cambios. Olvidar algo puede suponer un error costoso. • Discontinuidad de sus etapas, hasta no terminar una no se da inicio a la siguiente. • Gran cantidad de documentación.

DRA	<ul style="list-style-type: none"> • Se comprenden bien los requisitos del cliente. • Se limita el ámbito del proyecto. • Se puede separar el desarrollo en módulos. • Proyectos donde no se requiera un alto rendimiento. • Se posee un buen conocimiento de las herramientas de desarrollo. 	Adaptación de Cascada pura basado en componentes	<ul style="list-style-type: none"> • Desarrollo de proyectos en periodos cortos de tiempo (60-90 días) • Enfatiza en el uso de componentes reutilizables. 	Al concluir un modulo se entrega un producto de prueba al cliente.	Al final de cada modulo o subproyecto.	<ul style="list-style-type: none"> • En proyectos grandes se requieren suficientes personas para crear los equipos. • Requiere clientes y desarrolladores comprometidos en completar un sistema en corto tiempo. • No es recomendable cuando los riesgos técnicos son altos: por el uso de tecnologías nuevas o si se requiere interoperatividad con programas ya existentes.
PROTOTIPADO EVOLUTIVO	<ul style="list-style-type: none"> • Los requerimientos cambian con rapidez • El cliente no especifica los requerimientos. • No se identifica fácilmente el área de aplicación • No se esta seguro de la arquitectura o los algoritmos adecuados a utilizar 	Es una plantilla para otros métodos	El producto final cumplirá con todos los requerimientos del cliente.	Se presentan prototipos al cliente, continuando el desarrollo con los cambios sugeridos hasta ser aceptado como producto final.	Cada vez que el cliente prueba el prototipo	<ul style="list-style-type: none"> • Imposibilidad de conocer al comienzo del proyecto lo que se tardara en crear un producto aceptable. • Se puede degenerar fácilmente en un desarrollo sin ninguna planeación (con el esquema de codificar y corregir)

ESPIRAL	Es aplicable a todo tipo de desarrollo de software, especialmente en aquellos donde no se conocen bien los requerimientos del sistema.	<ul style="list-style-type: none"> • Puede finalizar con un modelo en cascada, prototipado u otro modelo • Puede incorporar otros modelos en sus iteraciones. 	<ul style="list-style-type: none"> • Las primeras iteraciones son menos costosas • Mientras los costos suben los riesgos bajan. • Proporciona bastante control de gestión. • Reducción continua del nivel de riesgo en el proyecto. • Mejor visibilidad de progreso. 	Hasta que se complete todo el número especificado de iteraciones requeridas para completar el producto.	Las pruebas se hacen al final de cada ciclo.	<ul style="list-style-type: none"> • Requiere mayor planificación • Requiere mayor seguimiento del proyecto. • Modelo complicado. • Puede ser difícil definir hitos y objetivos de comprobación que indiquen si está preparado para pasar al siguiente nivel. • El cliente no siempre está predispuesto a ser parte integral del desarrollo de software
ENTREGA POR ETAPAS O INCREMENTAL	<ul style="list-style-type: none"> • Se conoce exactamente lo que se va a construir. • El personal no está disponible para una implementación completa. 	Combina elementos del modelo en cascada con la filosofía iterativa de construcción por prototipos	<ul style="list-style-type: none"> • Presenta signos tangibles de progreso en el proyecto. • Proporciona una funcionalidad útil al cliente antes de entregar el 100% del proyecto. • Se pueden entregar las prestaciones más importantes al inicio. 	El producto se entrega por partes como resultado de cada una de las etapas de desarrollo.	Las pruebas se hacen al final de cada etapa	No funciona sin una planificación adecuada tanto para niveles técnicos como para niveles de gestión.

<p>PROCESO UNIFICADO</p>	<ul style="list-style-type: none"> Es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto. 	<p>En cada una de sus fases e iteraciones utiliza el método en cascada. Al inicio se puede utilizar prototipos de prueba.</p>	<ul style="list-style-type: none"> El proceso unificado es un marco de trabajo genérico. Busca eliminar los riesgos críticos primero. Reduce el costo de riesgos. Presenta signos visibles de progreso rápidamente. Proporciona funcionalidad al cliente antes de entregar todo el proyecto. Se entregan prestaciones importantes al inicio Proporciona suficiente control de gestión 	<p>Se entrega un incremento al final de cada iteración, en especial en la etapa de construcción.</p>	<p>Las pruebas se hacen al final de cada iteración. Además al final de cada fase se realiza una evaluación del plan de trabajo.</p>	<ul style="list-style-type: none"> Requiere mayor planificación Requiere mayor control de gestión. Se necesita definir su arquitectura antes de dar inicio a las iteraciones de construcción.
---------------------------------	--	---	--	--	---	--

Tabla 6. Resumen de los principales Modelos de Ciclo de Vida de Desarrollo del Software.

3.2 SELECCIÓN DEL CICLO DE VIDA.

El modelo de vida más efectivo depende del contexto en el que se utilice. Determinados modelos de ciclos de vida son considerados más rápidos que otros, pero cada uno de ellos será más rápido en determinadas situaciones, y más lento en otras, para seleccionar el modelo de ciclo de vida más conveniente se debe examinar el proyecto y responderse algunas preguntas que ayudara a escoger la metodología adecuada para ejecutar el proyecto.

Después de haber contestado esa serie de preguntas, se creara una tabla¹² que proporciona una guía para decidir que ciclo de vida utilizar. La efectividad de cualquier modelo dependerá de cómo se implemente. Pero si se conoce que el modelo es débil en un área particular, se puede controlar este inconveniente al principio de su planificación y compensarlo creando una mezcla entre uno o mas modelos descritos.

Capacidades del modelo	Cascada Pura	Cascada Modificada	DRA	Prototipo Evolutivo	Espiral	Entrega por etapas	Proceso Unificado
Trabaja con poca identificación de los requerimientos	Malo	Medio a Excelente	Malo	Excelente	Excelente	Malo	Excelente
Trabaja con poca comprensión sobre la arquitectura	Malo	Medio a Excelente	Malo	Malo a Medio	Excelente	Malo	Malo o Medio
Genera un sistema altamente fiable	Excelente	Excelente	Medio	Medio	Excelente	Excelente	Excelente
Genera un sistema con amplio desarrollo	Excelente	Excelente	Medio a Excelente	Excelente	Excelente	Excelente	Excelente
Gestionar riesgos	Malo	Medio	Excelente	Medio	Excelente	Medio	Excelente
Estar sometido a una planificación predefinida	Medio	Medio	Excelente	Malo	Medio	Medio	Excelente
Requiere poco tiempo de gestión	Malo	Excelente	Medio	Medio	Medio	Medio	Malo a Medio
Permite modificaciones a medio camino	Malo	Medio	Medio	Excelente	Medio	Malo	Excelente
Ofrece a los clientes signos visibles de progreso	Malo	Medio	Excelente	Excelente	Excelente	Medio	Excelente
Ofrece a la directiva signos visibles de progreso	Medio	Medio de Excelente	Excelente	Medio	Excelente	Excelente	Excelente

¹² Esta tabla ha sido tomada de McConnell con algunas modificaciones, eliminando algunos métodos que no se tomaran en cuenta en este trabajo de grado y añadiendo otros como DRA y Proceso Unificado.

Requiere poca sofisticación para los directivos y desarrolladores.	Medio	Malo a Medio	Medio	Malo	Malo	Medio	Excelente
--	-------	--------------	-------	------	------	-------	-----------

Tabla 7. Guía para la selección del ciclo de vida a utilizar en un proyecto.

Luego de analizar las preguntas planteadas y siguiendo la guía de la Tabla 7 Guía para la selección del ciclo de vida a utilizar en un proyecto. Se ha elegido el proceso unificado como metodología a seguir en este proyecto. Para el desarrollo del sistema planteado se tiene un buen conocimiento de los requisitos, así como de la arquitectura.

Esto último es especialmente importante, ya que el proceso unificado requiere un conocimiento de la misma en sus primeras iteraciones y esta se debe presentar estable durante el desarrollo del proyecto para poder aprovechar los beneficios de este método. Por otra parte el proceso unificado, ayuda a reducir riesgos de forma temprana y permite mostrar rápidamente signos visibles de progreso en el desarrollo del sistema. Aspecto importante en la realización de este proyecto.

Por ultimo, el proceso unificado requiere una amplia gestión, hecho que se convierte en un punto a favor para este caso en particular, dado que permitirá a los clientes y en especial, al director de proyecto mantener un constante seguimiento del mismo durante todo su desarrollo.

Es importante resaltar que el desarrollo de este proyecto estará enmarcado dentro de las 3 primeras fases del proceso unificado, dejando fuera la fase de transición que llevará a cabo en el marco de otro proyecto de grado, dadas las implicaciones directas que esta fase tiene en el tamaño y duración total y a la no disponibilidad inmediata del ambiente operativo para su implantación. Además, se utiliza el proceso unificado con algunas modificaciones al mismo para adaptarse al tipo de arquitectura escogida para el sistema.

Entre estas modificaciones se destaca la eliminación de los modelos de clases y de objetos y un mayor énfasis en el modelo de subsistemas dado que se va seguir la técnica de programación basada en procedimientos y no basada en objetos.

Si bien, el Proceso Unificado es utilizado con mayor énfasis para el desarrollo de proyectos con tecnología orientada a objetos, el proceso en si, “es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para

diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto.”¹³ Es este hecho, el que permite una adaptación del Proceso Unificado a este ambiente de desarrollo en particular.

3.3 PROCESO UNIFICADO.

El proceso unificado, metodología usada en este proyecto, es un proceso de desarrollo de software que proporciona normas para el desarrollo eficiente de software de calidad dentro de los plazos y presupuestos planeados. Entendiendo proceso de desarrollo de software como el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software.

El Proceso unificado es esta dirigido por casos de uso, centrado en la arquitectura, y es iterativo e incremental.

- ◆ *Dirigido por casos de uso:* El desarrollo del software se centra en la importancia del desarrollo para el usuario y no en términos de funciones que debe cumplir el sistema. Los casos de uso dirigen el proceso durante todos los flujos de trabajo de las distintas fases. Un caso de uso es una descripción de un conjunto de secuencias de acciones que un sistema lleva a cabo, un fragmento de su funcionalidad y que proporciona a un resultado interés para un actor determinado, donde un actor puede ser un usuario, un sistema o un rol.
- ◆ *Centrado en la arquitectura:* Al describir la arquitectura se obtiene una mayor comprensión del sistema, se organiza el desarrollo y se fomenta la reutilización. Esta arquitectura abarca la organización del sistema software, los elementos estructurales que compondrán el sistema y sus interfaces, así como su comportamiento y colaboraciones entre elementos.
- ◆ *Es iterativo e incremental:* Un proceso iterativo permite una comprensión creciente de los requerimientos, a la vez que se va haciendo crecer el sistema abordando las tareas más riesgosas primero. El trabajo de desarrollo se divide de manera planeada en partes más pequeñas llamadas iteraciones lo cual

¹³ Jacobson, Ivar. Booch, Grady. Rumbaugh, James. El Proceso Unificado de Desarrollo de Software. Primera edición. Addison Wesley. España, 2000. Pág. 4

genera progresivamente un incremento en el proyecto total. Si una iteración cumple sus objetivos el desarrollo continúa con la siguiente iteración, en caso contrario, se revisa las decisiones previas y se prueba un nuevo enfoque.

Como se dijo anteriormente, el Proceso Unificado divide el proceso de desarrollo en ciclos el cual se divide en cuatro Fases: Inicio, Elaboración, Construcción, y Transición. Cada una de estas fases concluye con un hito bien definido donde deben tomarse decisiones respecto al proyecto como la reestructuración del cronograma de trabajo. Cada una de estas fases se divide a su vez en iteraciones. Cada iteración sigue la estructura de un pequeño ciclo de vida en cascada, pasando a través de los cinco flujos de trabajo fundamentales: requisitos, análisis, diseño, implementación y prueba. En la iteración también incluye la planificación que precede a los flujos de trabajo y la evaluación que va detrás de ellos.

A continuación se presenta la Tabla 8 que resume las principales actividades que se realizan en cada flujo de trabajo durante las diferentes fases, así como una descripción de los aspectos más importantes a tener en cuenta en cada actividad de acuerdo en la fase donde se desarrolle. Al final de la tabla, se anexan los productos resultantes de cada fase del ciclo de desarrollo del proyecto.

FLUJO DE TRABAJO	ACTIVIDAD	INICIO	ELABORACIÓN	CONSTRUCCIÓN
Requisitos	Enumerar los requisitos candidatos	Las características provenientes de todas las fuentes de información se convierten en requisitos candidatos.	Se capturan la mayoría de los requisitos restantes para poder estimar el tamaño del proyecto.	Se concluye la recopilación de requisitos
	Comprender el contexto del sistema	Se usa el modelo de negocio para entender los procesos realizados dentro del ámbito del proyecto.		
	Capturar requisitos funcionales	Estos se representaran como casos de uso (CU)		
	Capturar requisitos no funcionales	Especifican propiedades del sistema, como dependencias de la plataforma, restricciones del entorno o de la implementación, etc.		
	Encontrar los actores y casos de uso	Solo los necesarios para describir el ámbito del sistema o la arquitectura candidata, ignorando las alternativas o caminos dentro de cada CU.	Su objetivo es comprender el 80% de los casos, buscando aquello que sea significativo desde el punto de vista de la arquitectura.	Conclusión de la descripción de los CU.
	Priorizar los casos de uso	Se realiza en la medida que se encuentran los CU. Permite planificar iteraciones futuras.	Se prioriza de acuerdo al nuevo conjunto de CU, según los riesgos percibidos y el orden en que se decida seguir el desarrollo.	A medida que identificamos CU, los añadimos a la clasificación con el objetivo de establecer su prioridad.
	Detallar los caso de uso	Solo aquellos necesarios para comprender el ámbito del sistema y los que sean de riesgo para la viabilidad del proyecto. Su detalle evitará pasar por alto los requisitos necesarios y críticos del sistema.	No se detallan en su totalidad CU seleccionados, sino que se limita el detalle a los CU y escenarios que necesitamos para esta fase.	Terminar el detalle de los CU faltantes.
	Prototipar la interfaz de usuario			Este prototipo se convierte en la especificación de la interfaz de usuario del sistema.
	Estructurar el modelo de casos de uso		En esta fase se buscan similitudes, simplificaciones y oportunidades para mejorar la estructura del modelo de CU.	Se pueden introducir mejoras sobre la estructura del modelo de CU, pero de aquellos que no se han desarrollados.
Análisis	Análisis de la arquitectura	Utilizando el conjunto inicial de CU detallado, se construye la primera versión del modelo de análisis para estas partes del sistema.	Se extiende este análisis que puede servir de base para la arquitectura. Se realiza una partición del sistema en paquetes de análisis empleando la arquitectura de capas.	
	Análisis de caso de uso	Se busca determinar los recursos compartidos entre los CU de relevancia en esta fase.	Solo se analiza los CU que aporten a la arquitectura del sistema y aquellos que sean dependientes entre si.	En cada iteración de esta fase se ampliara el modelo de análisis con los CU que sean incluidos en esta.

	Análisis de paquetes		Se busca garantizar que cada paquete de análisis cubran los CU analizados hasta esta fase. Así como que sean independientes entre si, tanto como sea posible.	Se refinan los paquetes que se identificaron en la anterior fase para acomodar los CU faltantes.
Diseño	Diseño de la arquitectura	Se desarrolla un esbozo inicial de la vista de la arquitectura del modelo de diseño, que realice los CU de interés como colaboraciones entre subsistemas. Se elige el software del sistema.	Se incluye en el modelo de diseño la arquitectura en capas, los subsistemas, interfaces y realizaciones de CU de aquellos arquitectónicamente significativos.	Se determina la conveniencia de realizar subsistemas completos a pesar de que alguna de sus partes pertenezcan a CU de baja prioridad
	Diseño de un caso de uso		Los CU importantes para esta fase se diseñan como subsistemas y se describen las interacciones entre estos.	Se diseñan los CU que surgen en esta fase como parte de los subsistemas existentes..
	Diseño de subsistemas		Se diseñan los subsistemas resultantes del diseño de la arquitectura. Si es necesario, se actualiza la vista de la arquitectura del modelo de diseño	Normalmente no se añaden nuevos subsistemas pero si se incluyen mejoras a estos.
Implementación	Implementación de la arquitectura		Se realiza la implementación de la arquitectura del sistema y se describe en la vista de arquitectura del modelo de implantación.	
	Integrar el sistema		Se integran las construcciones de los CU implementados en esta fase.	Se realizan e integran las capas inferiores de la arquitectura y luego las superiores.
	Implementar un subsistema		Se inicia la implementación de las partes cruciales para la arquitectura del sistema.	Se implementan todos los subsistemas.
	Realizar pruebas de unidad		Se realizan estas pruebas sobre los casos implementados.	Se realizan estas pruebas y se corrige el diseño y la implementación de cada componente de ser necesario.
Prueba	Planificar prueba		Se seleccionan los objetivos que evaluarán la línea base de la arquitectura.	Se establecen objetivos para cada construcción resultante de una iteración y luego para el sistema total
	Diseñar prueba		Con base en estos objetivos y en los CU implementados se diseña la prueba	Determinar como probar los requisitos del usuario en el programa.
	Realizar prueba de integración		Se verifica las interfaces entre componentes implementados y que trabajen bien conjuntamente.	Se realizan pruebas de integridad añadiendo cada construcción y probando que todos los componentes trabajen correctamente de forma conjunta.

<p>PRODUCTOS</p>	<ul style="list-style-type: none"> • Una lista de características. • Un esbozo de los modelos que representan la primera versión del modelo de casos de uso, el modelo de análisis y el modelo de diseño. • Un primer esquema de la descripción de una arquitectura candidata. • Un primer borrador de un plan para el proyecto en su totalidad. 	<ul style="list-style-type: none"> • Nueva versión de todos los modelos: casos de uso, análisis, diseño, despliegue e implementación. • Una línea base de la arquitectura. • Una descripción de la arquitectura, incluyendo vistas de los modelos de casos de uso, análisis, diseño, despliegue e implementación. • El plan de proyecto para las fases de construcción. • Versión inicial del manual de usuario. 	<ul style="list-style-type: none"> • El sistema software ejecutable. • Todos los modelos del sistema • La descripción de la arquitectura. • Manuales de usuario, técnicos y administrativos
-------------------------	--	---	---

Tabla 8. Productos y Actividades de cada uno de los flujos de trabajo

3.4 UML.

A mediados de los noventa existían métodos de análisis y diseño, lo que suponía que los mismos conceptos tenían distinta notación según el método de que se tratara. Ante esta situación, en 1994 Booch, Rumbaugh y Jacobson decidieron unificar sus métodos dando lugar a UML. Esta unificación fue promovida por el OMG de tal manera que UML se convirtió en la notación estándar para la descripción de métodos software. 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: Proporciona un vocabulario y unas reglas que se centran en la representación conceptual y física de un sistema, y que indican cómo crear y leer modelos bien formados. Sin embargo, no dice qué modelos crear ni cuándo se deberían crear, ésta es la tarea del proceso de desarrollo de software.
- ◆ Es un lenguaje para visualizar: Es un lenguaje gráfico que mezcla gráficos y texto, pero es algo más que un simple montón de símbolos. De hecho, detrás de cada símbolo en la notación UML hay una semántica bien definida, de manera que un desarrollador puede escribir un modelo en UML, y otro desarrollador, o incluso otra herramienta, puede interpretar ese modelo sin ambigüedad.
- ◆ 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: No es un lenguaje visual, pero sus modelos pueden conectarse de forma directa con una gran variedad de lenguajes de programación. Es posible establecer correspondencias desde un modelo UML a un lenguaje de programación como Java o C++, o incluso a tablas en una base de datos relacional o al almacenamiento persistente en una base de datos orientada a objetos. Permite ingeniería directa e inversa.

- ◆ Es un lenguaje para documentar: cubre toda la documentación de la arquitectura de un sistema y todos sus detalles. También proporciona un lenguaje para expresar requisitos y pruebas del software. Finalmente, UML proporciona un lenguaje para modelar las actividades de planificación de proyectos y gestión de versiones.

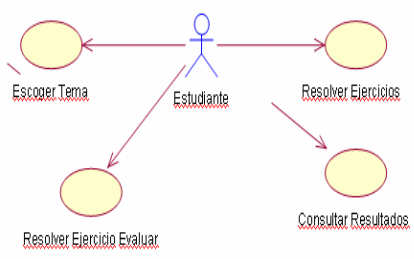
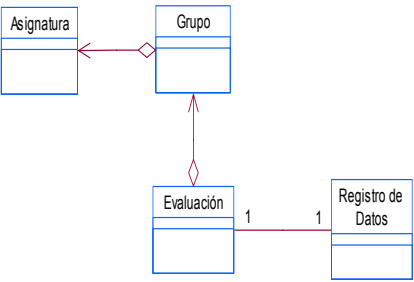
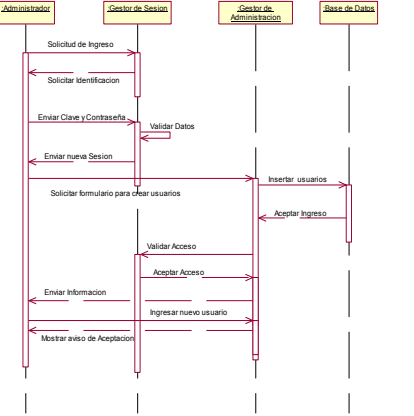
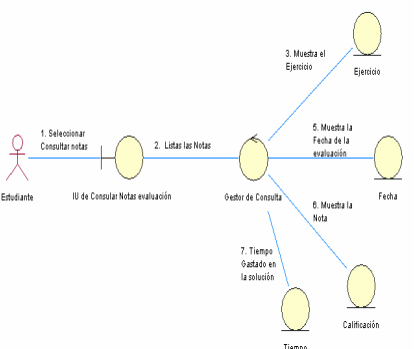
Desde la consolidación de UML como lenguaje estándar para el modelado se ha definido un buen número de procesos para el desarrollo de aplicaciones orientadas a objetos que utilizan este lenguaje como medio de expresión de los diferentes modelos que se crean durante el ciclo de vida. Las características principales deseables en cualquier proceso software basado en UML son:

- ◆ Un proceso debe ser iterativo e incremental, y debe centrarse en los aspectos críticos en las primeras iteraciones para minimizar riesgos.
- ◆ Debe estar guiado por los requisitos (casos de uso). Los requisitos cambian a lo largo del desarrollo del proyecto y el proceso debe estar preparado para identificar nuevos requisitos a lo largo de todo el ciclo de vida, ya que es difícil que puedan capturarse todos los requisitos antes de empezar la implementación.
- ◆ Debe utilizar arquitecturas basadas en componentes.
- ◆ Debe existir un control de cambios del software. La ausencia de un control de cambios hace que el proceso degenera rápidamente en un caos. Si se hace un control de cambios se solucionan parte de las dificultades principales del desarrollo de software, como la comunicación entre equipos de desarrollo, la consistencia, la interferencia entre miembros de un equipo que trabajan en paralelo, etc.

En todos los procesos basados en UML el concepto de caso de uso juega un papel primordial, ya que se emplea para definir los requisitos funcionales del sistema y en torno a ellos se articulan todas las etapas del proceso.

A continuación se presenta un resumen de los diagramas UML utilizados en la realización de este proyecto. Cabe resaltar que la conveniencia de utilizar un

determinado diagrama está determinada por el aporte que este da al entendimiento del sistema. No está determinada ni por la metodología, ni por el propio lenguaje UML.

DIAGRAMA	DESCRIPCIÓN	EJEMPLO
De Casos de Uso	Representa las funciones del sistema desde el punto de vista del usuario. Se utilizan para la comunicación con los usuarios y para expresar de forma clara y sencilla los requisitos.	 <p>Figura 1. Modelo de Casos de uso – Casos de uso por actor.</p>
De Clase	Bosquejan la arquitectura estática del sistema en función de clases y sus asociaciones.	
De Secuencia	Representan un conjunto de elementos de un sistema que interaccionan entre ellos organizados en secuencias de tiempo.	
De Colaboración	Son parecidos a los diagramas de secuencia pero dan mayor libertad para distribuir los objetos. Representan la interacción entre elementos de un sistema y se organiza con respecto a espacio y tiempo. Es una representación espacial de los objetos, sus enlaces y sus interacciones. Junto con los de secuencias se denominan, diagramas de interacción.	

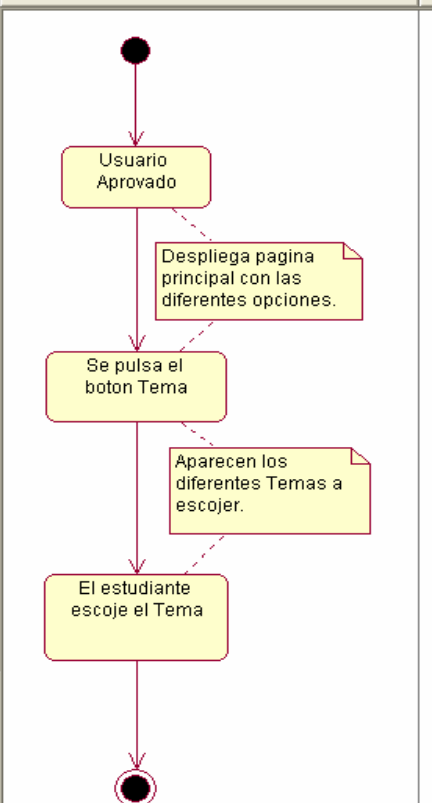
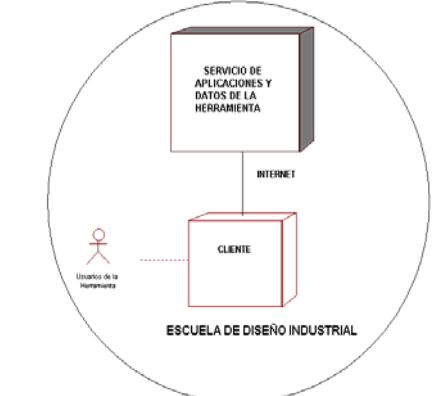
<p>De Estado</p>	<p>Representan el estado de un elemento del sistema, las condiciones y respuestas que da a ciertos eventos a los que es expuesto.</p>	 <pre> stateDiagram-v2 [*] --> UsuarioAprobado UsuarioAprobado --> SePulsaBotonTema : Despliega pagina principal con las diferentes opciones. SePulsaBotonTema --> ElEstudianteEscojeTema : Aparecen los diferentes Temas a escojer. ElEstudianteEscojeTema --> [*] </pre>
<p>De Despliegue</p>	<p>Muestra un conjunto de nodos y sus relaciones. Representa el despliegue de los componentes y subsistemas sobre los dispositivos físicos.</p>	 <pre> graph TD subgraph ESCUELA_DE_DISEÑO_INDUSTRIAL [ESCUELA DE DISEÑO INDUSTRIAL] CLIENTE[CLIENTE] SERVICIO[SERVICIO DE APLICACIONES Y DATOS DE LA HERRAMIENTA] CLIENTE --- INTERNET[INTERNET] --- SERVICIO end Usuario[Usuario de la Herramienta] -.-> CLIENTE </pre>

Tabla 9. Diagramas UML utilizados para la realización de la Herramienta GEDISO 1.0.

PARTE II

FASE DE INICIO

El proceso de desarrollo de software, esta basado en un conjunto de actividades que permiten transformar los requisitos de los usuarios en una herramienta software, después de haber encontrado esos requisitos el paso a seguir de acuerdo a la metodología elegida es la fase de inicio.

El objetivo de esta fase es desarrollar el análisis del negocio hasta el punto necesario para justificar la puesta en marcha del proyecto, es por eso que ésta es considerada como una de las más importantes, ya que en este momento es cuando se han aclarado las necesidades primordiales entre los usuarios y desarrolladores, llevando a comprender totalmente la problemática que se presenta, determinando las características, alcances y ámbito de la herramienta software.

Estos tópicos darán a los clientes y al resto de personas involucradas en el proyecto, la seguridad de que los desarrolladores, “serán capaces de contener los riesgos críticos y formular una arquitectura candidata para hacer el análisis inicial del negocio”¹⁴, garantizando la definitiva ejecución del proyecto.

La importancia que tiene la fase de inicio para este proyecto, es establecer durante su ejecución, la traducción de los conceptos e ideas confusas que no se han aclarado con el desarrollo de las primeras indagaciones sobre el tema, causadas en alguna forma por la problemática planteada en el desarrollo de una primera herramienta, que no satisfizo las necesidades del cliente y la aparición de nuevas ideas, que se dieron con el transcurso del tiempo, para transformarlas en características funcionales que harán parte de la herramienta software que se desea construir, llegando a plantear un primer esquema de la arquitectura de la herramienta.

Con base al cumplimiento de la fase anterior será posible iniciar la ejecución de las fases de elaboración y construcción, las cuales darán como resultado el desarrollo de una herramienta software funcional.

¹⁴ El proceso unificado del desarrollo del software, Addison Wesley, Cap. 13, Pág. 327

Capítulo 4.

PLANEACION

Para el desarrollo de esta fase se reunirá toda la información recopilada en la exploración realizada con clientes y comunidad científica, organizándola de tal forma que pueda ser útil. En esta fase únicamente se tendrán en cuenta las actividades correspondientes a los flujos de trabajo fundamentales los cuales son: requisitos, análisis y diseño.

4.1 PLANIFICACION DE LA FASE DE INICIO.

Para el inicio de esta fase se realizaron una serie de consultas sobre herramientas software existentes en el mercado con características similares a la pensada en desarrollar, se encontró en la actualidad una herramienta software instalada en la Escuela de Diseño Industrial, la cual se encuentra en un estado no funcional, debido a una serie de problemas técnicos y de instalación de la misma.

La información que fue recopilada permitió crear un esquema general donde se tratara de identificar las principales necesidades o características requeridas por los usuarios y algunas otras detectadas por los desarrolladores para la creación de una nueva herramienta software.

Al culminar con esta planificación, se debe tener un listado de las características de la herramienta:

- ◆ Los requisitos candidatos.
- ◆ La comprensión del contexto de la herramienta.
- ◆ Representar los requisitos funcionales convenientes, y los requisitos no funcionales haciendo énfasis en el flujo de trabajo, donde se identifican y se detallan los casos de uso pertinentes de la fase.
- ◆ Listado de los riesgos críticos, ordenándolos por un nivel de seriedad o por su influencia en el desarrollo llevando a la creación de un plan de acción a seguir para contenerlos.

- ◆ Igualmente se debe tener el modelo del negocio que permite identificar los casos de usos, los actores, las entidades que describen los procesos de negocio de una empresa, todo esto se utilizara para comprender el contexto de la herramienta.

4.2 FLUJOS DE TRABAJO

4.2.1 EL FLUJO DE TRABAJO DE LOS REQUISITOS.

En este flujo de trabajo es donde se realiza una descripción de los requisitos del software, conociendo las condiciones o capacidades que la herramienta debe cumplir, además los desarrolladores identifican los casos de uso y actores que definen el ámbito en que se va a trabajar, con lo anterior se obtiene la arquitectura candidata, y las características distribuidas en las diferentes actividades descritas en la fase de inicio, por tanto en este primer flujo se lleva a cabo la mayor parte del trabajo en esta fase .

Capítulo 5.

VIABILIDAD DEL PROYECTO: UNICA ITERACION.

En este capítulo se describirá brevemente cómo se procede para realizar cada una de las actividades involucradas en esta fase, para luego ejecutarlas en el proyecto. Para el desarrollo de cada actividad y de los artefactos generados en su desarrollo sólo se presenta lo necesario para el entendimiento de la metodología y para orientar al lector hacia la lectura y análisis de los productos entregados, los cuales se presentan como anexos.

Este esquema se seguirá durante el desarrollo del resto del libro, sin embargo en el capítulo se explicará con mayor detalle los aspectos relacionados con la obtención de cada uno de los artefactos¹⁵ y productos¹⁶. En los siguientes capítulos sólo se realizarán explicaciones cuando se trate de nuevas actividades o artefactos no presentados hasta el momento o cuando varíen con respecto a las descritas en la fase de inicio.

5.1 ACTIVIDAD: CAPTURA DE REQUISITOS CANDIDATOS.

A través de las diferentes reuniones que se realizan entre clientes, usuarios y desarrolladores surgen ideas que pueden ser de utilidad para el desarrollo de la herramienta, convirtiéndose en verdaderos requisitos, por tanto se ve la necesidad de tener una lista o tabla donde se registre un listado de los requisitos candidatos obtenidos, organizados por los módulos de los cuales se compone la herramienta a desarrollar.

¹⁵ Información tangible que es creada, modificada y usada al realizar las actividades de cada fase por las personas que intervienen en el desarrollo del sistema. Ejemplos de artefacto son los modelos, los diagramas, los prototipos de interfaz y los procedimientos de prueba.

¹⁶ Artefactos finales que forman parte del sistema terminado y que se crean durante la vida del proyecto, como modelos, código fuente, ejecutables, documentación.

Los campos de la tabla que se describen a continuación, tienen un nombre corto y una pequeña descripción o información, que ayudan a entender su importancia en la herramienta a desarrollar, los campos son:

- ◆ **Nombre del requisito:** Característica detectada.
- ◆ **Descripción:** Información importante del requisito.
- ◆ **Estado:** (Propuesto, aprobado, incluido o validado).
- ◆ **Prioridad:** (Crítico, importante o secundario).
- ◆ **Nivel de Riesgo:** (Crítico, significativo u ordinario).

A continuación se presenta la lista de características candidatas.

	NOMBRE	DESCRIPCIÓN	ESTADO	PRIORIDAD	NIVEL DE RIESGO
TEORIA	Anexo de documentos a la herramienta, por parte de los profesores.	La Herramienta permite agregar documentos como anexo a la información que la herramienta posee. Este trabajo solo puede ser elaborado por profesores, ya que se verifica su clave y contraseña.	Aprobado	Importante	Ordinario
	Ver los diferentes documentos que han sido anexados	La Herramienta permite a los diferentes usuarios ver los anexos que los profesores han suministrado como complemento a la información teórica que ya se tiene.	Aprobado	Importante	Ordinario
	La creación de ejercicios resueltos paso a paso	La herramienta en el modulo de teoría contara con ejercicios resueltos paso a paso, a manera de ejemplo de acuerdo a los temas vistos en la asignatura.	Aprobado	Crítico	Significativo
	La generación de listado de temas	El usuario de la herramienta tiene la capacidad de escoger en un listado el tema que desea consultar.	Aprobado	Secundario	Ordinario
PRACTICA	Verificación del registro de los usuarios de la herramienta.	Los usuarios antes de escoger el ó los tipos de problemas que desean desarrollar, deben ser registrados o verificar que sean usuarios autorizados, para proceder a utilizar la herramienta.	Aprobado	Importante	Ordinario

EVALUACIÓN	La generación de listado de ejercicios a resolver.	La herramienta debe permitir a todos los usuarios escoger en un listado los diferentes problemas que van a desarrollar en el tiempo asignado de trabajo por el administrador de la herramienta.	Aprobado	Secundario	Ordinario
	Solución de ejercicios a resolver por parte de los estudiantes	Este es un problema que concierne solo a los desarrolladores de la herramienta, buscando la mejor forma de crear la solución de los ejercicios, proporcionando a los estudiantes los elementos necesarios para la solución de los mismos.	Aprobado	Critico	Critico
	Registro de lo hecho en la hora de practica.	La herramienta debe permitir registrar las actividades realizadas por parte de los estudiantes en las horas asignadas de trabajo: el tiempo gastado por ejercicio, cantidad de ejercicios resueltos, etc.	Aprobado	Secundario	Ordinario
	Verificación del registro de los usuarios de la herramienta.	Los usuarios antes de escoger el ó los tipos de problemas que desea desarrollar, deberán ser registrados o verificar que sea usuario autorizado, para proceder a utilizar la herramienta.	Aprobado	Importante	Ordinario
	Solución de ejercicios a resolver por parte de los estudiantes	Este es un problema que concierne solo a los desarrolladores de la herramienta, buscando la mejor forma de crear la solución de los ejercicios, proporcionando a los estudiantes los elementos necesarios para la solución de los mismos	Aprobado	Critico	Critico
	Registro de lo hecho en la hora de evaluación.	La herramienta debe permitir registrar todas las actividades realizadas por parte de los estudiantes en las horas asignadas de evaluación , el tiempo gastado por ejercicio, cantidad de ejercicios resueltos, etc.	Aprobado	Importante	Ordinario
	La generación de listado de ejercicios a resolver.	La herramienta debe permitir la selección a los diferentes usuarios que se encuentren en el modulo de evaluación los ejercicios a resolver de acuerdo a un listado de los diferentes problemas que se encuentran almacenados en la base de datos de la herramienta y dado el tema a evaluar.	Propuesto	Importante	Ordinario

	Consulta de notas.	Los estudiantes y el profesor podrán consultar las notas correspondientes. Los estudiantes podrán acceder a sus notas en cualquier momento, después de haber sido calificadas sus evaluaciones.	Propuesto	Secundario	Ordinario
ADMINISTRACIÓN	Ingresar y/o modificar usuarios.	La herramienta debe permitir la creación de nuevos usuarios, así como la modificación de los datos que se encuentren registrados de los mismos.	Aprobado	Critico	Ordinario
	Actualizar los ejercicios a resolver en el modulo de practica y evaluación	El administrador estará en la capacidad de ingresar nuevos ejercicios a la base de datos que contiene la herramienta, para evitar el aprendizaje de estos por parte de los estudiantes que manejan la herramienta.	Aprobado	Critico	Significativo
	Administración de grupos.	Al inicio de cada semestre se debe permitir abrir un grupo y asignar el profesor y alumnos que pertenezcan a él. También agregar y eliminar estudiantes, cambiar profesores y modificar los datos asociados a dicho grupo.	Aprobado	Importante	Ordinario
OTRAS	Bajos costos en el desarrollo y la implantación.	El software debe desarrollarse e implantarse con mínimos costos, sin afectar su funcionamiento.	Aprobado	Importante	Ordinario

Tabla10. Listado de requisitos candidatos.

5.2 ACTIVIDAD: IDENTIFICAR RIESGOS CRITICOS.

Los riesgos son un factor a tener en cuenta cuando se realiza un software, ya que estos influyen en el desarrollo normal de un proyecto. Por lo anterior, se ve la necesidad de crear una lista de posibles riesgos que determinan la capacidad de construir el software tratando de ordenarlos de tal forma que sea fácil identificar los que conduzcan a fallas graves del software, estos deben ser mitigados para poder ofrecer planes de contingencia y determinar un objetivo de calidad.

Los riesgos críticos que se mencionan a continuación, son los que afectan la construcción de la herramienta software. Se presentan planteando las posibles contingencias que permitan reducir o eliminar totalmente los efectos perjudiciales durante el desarrollo del software.

- ◆ **Descripción:** Una breve descripción o explicación corta del riesgo.
- ◆ **Prioridad:** (crítico, significativo o rutinario).
- ◆ **Impacto:** Indica las partes de la herramienta que se verán afectadas.
- ◆ **Monitor:** Persona encargada de realizar un seguimiento a un riesgo persistente.
- ◆ **Responsabilidad:** indica la persona o unidad responsable de eliminar el riesgo.
- ◆ **Contingencia:** Plan que se ejecuta en caso de aparecer un riesgo.

DESCRIPCIÓN	NIVEL DE RIESGO	IMPACTO	MONITOR	RESPONSABILIDAD	CONTINGENCIA
El plan de proyecto no es aprobado por el comité de proyectos de grado	Critico	Global	Autores	Autores y Director del proyecto	Apelar a otros mecanismos de aprobación
Problemas con el manejo del lenguaje de programación	Rutinario	Global	Autores	Autores	Estudio
Desconocimiento de la plataforma de desarrollo	Rutinario	Global	Autores	Autores	Estudio
Problemas con la parte grafica para el desarrollo los problemas	Critico	Global	Autores	Autores	Estudio
Desconocimiento del almacenamiento de archivo gráficos por parte de la base de datos	Rutinario	Global	Autores	Autores	Estudio y búsqueda de información
Actualmente no	Significativo	Global	Director del	Director del proyecto	Gestión con la

se cuenta con un equipo que actúe como servidor			proyecto		Escuela
---	--	--	----------	--	---------

Tabla 11. Listado de riesgos críticos.

5.3 ACTIVIDAD: COMPRENDER EL CONTEXTO DE LA HERRAMIENTA

El objetivo de definir el contexto de la herramienta es obtener un mayor conocimiento del ámbito en el cual se desarrollara el proyecto, así como delimitar los alcances del software. Hay por lo menos dos aproximaciones para expresar el contexto de un software en una forma utilizable para desarrolladores de software: modelado del dominio y modelado del negocio.

5.3.1 DESCRIPCIÓN DEL CONTEXTO

Las personas encargadas de realizar una herramienta software en este caso estudiantes de pregrado, a medida que modelan el negocio aprenden sobre el contexto de la herramienta software a desarrollar, con lo cual estarán en la capacidad de realizar el modelo del negocio, donde se especificara que procesos serán soportados por la herramienta a crear, para esto se establecerán las capacidades requeridas en cada proceso; las personas que los ejecutan; las tareas que tienen a cargo y el nivel de responsabilidad que poseen cada uno para los procesos.

En este caso se desea llegar a comprender totalmente el contexto de la herramienta software a crear, para obtener el modelo del negocio, el cual será de ayuda para la creación de un diseño amplio y completo que cumpla con todas las necesidades encontradas en el proceso de búsqueda de requisitos candidatos.

El contexto de trabajo será el aula de clase donde se desarrolla o se lleva a cabo una clase de la asignatura *geometría descriptiva*, observando la capacidad que poseen los alumnos para comprender la información entregada por los profesores en cada clase y como esta es utilizada para la solución de problemas, pero además de esto nos

aventuraremos en una nueva área importante en la actualidad en el desarrollo de cualquier asignatura a través de medios computacionales, *la informática educativa*, considerada como el futuro de la educación en escuelas, colegios y universidades.

5.3.2 QUE ES LA GEOMETRÍA DESCRIPTIVA.

La geometría descriptiva es un medio de expresión y comunicación indispensable, tanto en el desarrollo de procesos de investigación científica como en la comprensión gráfica de proyectos tecnológicos cuyo último fin sea la creación y fabricación de un producto.

Su función esencial en estos procesos consiste en ayudar a formalizar y visualizar lo que se está diseñando o descubriendo, y contribuye a proporcionar desde una primera concreción de posibles soluciones hasta la última fase del desarrollo, donde se presentan los resultados en planos definitivamente acabados.

La geometría descriptiva debe orientarse hacia la construcción sistemática de criterios relativos a la comprensión y operación de la especialidad, reconociendo a la geometría como instrumento de control y operación objetual, como una herramienta básica en el fortalecimiento de un pensamiento coherente y creativo.

La geometría descriptiva se encuentra con la excepción de la ingeniería de Sistemas en todos los planes curriculares de las ingenierías en la Universidad Industrial de Santander, ya que proporciona una forma técnica de ver la realidad y del como llevarla a la representación en el diseño, razón por la cual se hace relevante la búsqueda de alternativas, que como esta contribuyan al mejoramiento de su proceso de enseñanza-aprendizaje.

5.3.3 INFORMATICA EDUCATIVA.

La incorporación de nuevas tecnologías en la sala de clase repercute directamente en el proceso en su conjunto, es decir, en la enseñanza y el aprendizaje. Es por ello que el concepto de la Educación se ve enriquecido con nuevos aportes metodológicos propuestos por la Informática Educativa.

La herramienta que se incorpora definitivamente en los establecimientos educativos es el computador, en torno al cual se debe centrar la atención, valorando la tremenda influencia motivadora que presenta en el aula, se pasa de métodos convencionales de

dictar las clases, a métodos no convencionales donde el maestro es un guía o facilitador del aprendizaje utilizando las tecnologías de información. Si se pretende resumir en pocas palabras y en forma directa dichas influencias se debe mencionar; el aporte significativo que pueden tener en una primera instancia los software educativos que se encuentran hoy en día en el mercado, por otro, la utilización de esta herramienta por parte del maestro como apoyo administrativo a la labor diaria del quehacer docente y, por último, el apoyo de esta herramienta como elemento importante en la aplicación de un nuevo método pedagógico en el aula, enriquecido con todas las bondades que presenta para la educación la red de redes Internet.

La informática educativa es un tema relativamente nuevo en nuestro ámbito de trabajo, por tanto no es fácil encontrar y dar una solución a esta nueva rama de la educación que comienza a surgir con gran fuerza, la cual se impondrá en un futuro como la metodología más fuerte para la enseñanza-aprendizaje de cualquier materia.

La *informática educativa* esta orienta a conformar *ambientes de aprendizaje* para que los estudiantes tengan experiencias tendientes al *cambio conceptual*, esta se presenta como un *dispositivo* auxiliar del enseñante e incluso como una eventual objetivación de su saber, o de algunas dimensiones del mismo.

La *informática educativa* está orientada al *incremento de la competencia cognitiva de los aprendices*, mediante métodos estructurados de guía y evaluación de los procesos de apropiación conceptual de los mismos, en todos los casos, de manera más o menos explícita, expone y realiza proyectos de formación modernos, que atienden a las expectativas de autonomía, autodeterminación y autorreflexión como divisa para que el sujeto se constituya, como héroe de su propio relato.

Como conclusión de se puede afirmar que *La informática educativa*, trata de reconstruir escenas del mundo real vale decir, pedagógico o educativo o de enseñanza para simular las formas de interacción de los sujetos, dentro de los mentados entornos, a efecto de que como en el mundo real se produzca aprendizaje por la interacción, cambio conceptual por la operación constructiva del sujeto en la interacción, transformación en las condiciones iniciales mediante cambios de estado que opera el sujeto sobre el “mundo” simulado.

Investigaciones realizadas llevaron a la convicción de que la *informática educativa* tiene más bien un valor instrumental para atender los procesos de enseñanza, casos concretos son los de las matemáticas, la geometría y la física¹⁷.

5.3.4 MODELO DE CASOS DE USO DEL NEGOCIO

De acuerdo con la descripción del contexto donde se desarrollara la herramienta software se han determinado una serie de casos de uso que ayudaran a definir el modelo del negocio y por lo tanto el contexto de la herramienta. Hay que recordar que el modelo del negocio puede ser la base para la especificación de los requisitos más importantes de la herramienta que dará soporte al negocio, siendo por tanto el propio negocio lo que determine los requisitos.

El objetivo del modelado del negocio es describir los procesos, con el propósito de comprenderlos. Para este caso solo se modelan los procesos que soportará la herramienta software y de forma general, en este modelo se establecen los trabajadores, responsabilidades y operaciones que llevan a cabo.

A continuación se presenta el modelo de casos de uso del negocio, representando a cada proceso del negocio como un caso de uso dentro del modelo.

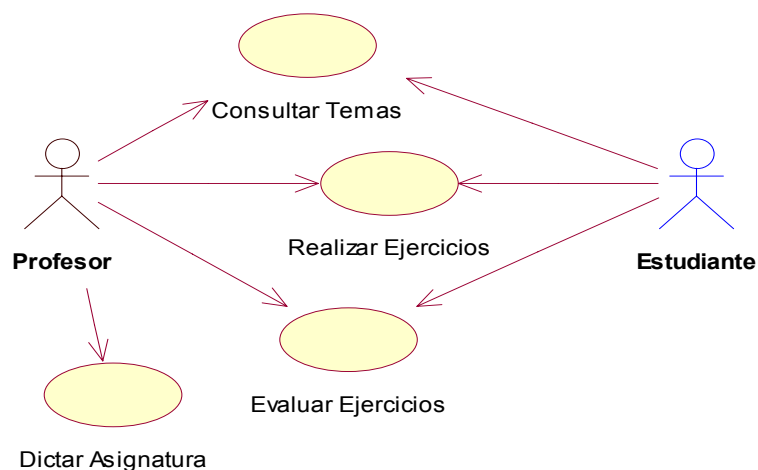


Figura 18. Modelo de casos de uso del negocio.

¹⁷ Informática educativa vs. Pedagogía Computacional, GERMÁN VARGAS GUILLÉN, JOSÉ GONZÁLEZ FLÓREZ, UNIVERSIDAD PEDAGÓGICA NACIONAL., documento Web.

Para comprender un poco más el modelo del negocio, se describirá un caso de uso del negocio a través de la siguiente tabla con los respectivos campos identificados:

- ◆ **Proceso del negocio:** Nombre del caso de uso o proceso.
- ◆ **Objetivo:** objetivo que se intenta conseguir.
- ◆ **Descripción:** Enuncia las actividades del proceso.
- ◆ **Prioridad:** Importancia del proceso, puede ser Critico, importante o secundario
- ◆ **Riesgos:** Errores o fallos que pueden ocurrir al ejecutar este proceso
- ◆ **Posibilidades:** Cambios o mejoras futuras al proceso.
- ◆ **Tiempo:** Tiempo aproximado de ejecución
- ◆ **Costo:** Costo aproximado de ejecución

Proceso de Negocio	CONSULTAR TEMAS
Objetivo	Consultar información correspondiente a los temas de la asignatura.
Descripción	<ol style="list-style-type: none"> 1. Se consulta la base de datos para verificar si el usuario esta registrado, si no lo esta debe registrarse para hacer uso de la herramienta. 2. Permite tener un listado de todos los temas que se trataran en el transcurso del semestre, donde el estudiante escogerá el tema que desee consultar. 3. Mostrara toda la información que el usuario desee consultar sobre el tema escogido, donde los conceptos que se presentan estarán organizados para evitar problemas de comprensión. 4. Presentara ejercicios resueltos como ejemplos a manera de secuencias de imágenes, para que el estudiante observe una forma de realizar los problemas con características generales, estos se encontraran al finalizar todo el tema escogido.
Prioridad	Importante
Riesgos	Rutinario
Posibilidades	Este proceso se podría mejorar, si las páginas donde se contiene la información se pudieran modificar de acuerdo a las mejoras de los conceptos, y con la opción de modificar los ejemplos que estas tienen, para darle un carácter más dinámico.
Tiempo de ejecución	Depende de la cantidad de información que se desee montar en la herramienta y la complejidad de los ejemplos a crear (4 semanas – 6 semanas).

Costo de ejecución	Depende de los mismos factores que para el tiempo de ejecución.
---------------------------	---

Tabla12. Descripción del proceso de negocio consultar temas.

5.3.5 MODELO DEL DOMINIO

Un modelado del dominio describe los conceptos importantes del contexto como, “objetos del dominio los cuales representan las “cosas” que existen o los eventos que suceden en el entorno en el cual trabajara la herramienta”¹⁸.

El modelo del dominio que se tiene de la herramienta se obtuvo de las diversas reuniones con los usuarios finales donde se obtuvieron un numero determinado de clases de las cuales se toman las más preponderantes para la creación del negocio, las otras se guardan como definiciones en un glosario de términos.

Los objetos en este modelo de información fluyen entre las actividades de un caso de uso del negocio y representan la relación que hay entre las diferentes clases del dominio, entregando una buena base para crear el primer modelo de contexto, donde se incluirá los conceptos, las relaciones, descritas mediante un diagrama de clases de UML.

¹⁸ El proceso unificado del desarrollo del software, Addison Wesley, Pág. 6.5, Pág. 112

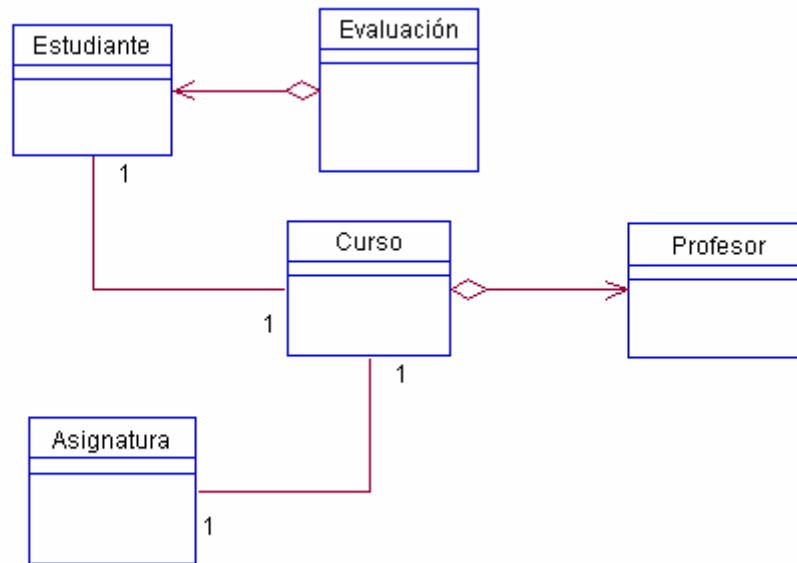


Figura 19. Modelo del domino

5.4 ACTIVIDAD: REPRESENTAR LOS REQUISITOS COMO CASOS DE USO

5.4.1. ENCONTRAR ACTORES

La identificación de los casos de usos y los actores de la herramienta es una de las actividades más importantes y decisivas para la obtención de los requisitos apropiados, siendo esta una actividad de analistas, en este caso estudiantes de pregrado encargados de desarrollar la herramienta, apoyados por las personas involucradas en la ejecución del proyecto.

A partir del modelo del negocio es posible obtener los actores de la herramienta de manera sencilla y directa. En primer lugar se identifica un actor por cada trabajador y por cada cliente del negocio, que se convertirá en usuario de la herramienta, con el fin de identificar los actores que representan sistemas expertos, de mantenimiento y operación del software.

Una vez encontrados todos los roles de un actor en el negocio, uno por cada caso de uso del negocio en el que participa, se encuentran los casos de uso de los actores. A

cada papel de un trabajador o cliente del negocio le concierne un actor al cual se le asocia por cada caso de uso del negocio en el que participe.

A medida que se avanza en la comprensión de contexto de la herramienta, se van encontrando los actores involucrados los cuales poseen uno o varios roles que se convertirán en casos de uso. Los actores interactúan a través de los casos de uso y es importante definir sus responsabilidades y necesidades en la herramienta, para evitar la definición de varios actores que cumplan el mismo papel.

Para encontrar los actores se parte del modelo de negocio y se siguen los siguientes criterios:

- ◆ Identificar al menos a un usuario que pueda representar al actor candidato, de esta manera se tendran los actores relevantes.
- ◆ Exista una coincidencia mínima entre los roles que desempeñan las instancias de los diferentes casos de uso en relación con la herramienta, esto para evitar que dos o más actores tengan en esencia los mismos roles. De ser así se tratara de crear un actor generalizado que tenga asignados los roles comunes a los actores que se solapan.

A continuación se presenta la siguiente tabla de actores, a estos se les asignan un nombre y se describe brevemente los papeles de cada actor y para que utiliza la herramienta:

NOMBRE	DESCRIPCIÓN	RESPONSABILIDADES (PAPELES QUE JUEGA)	NECESIDADES (PARA QUE UTILIZA EL SISTEMA)
Profesor	Es la persona que orienta la actividad académica de la asignatura	<ul style="list-style-type: none">• Desarrollar las clases asignadas en su carga académica.• Proporciona información para la asignación de horarios de trabajo, para los alumnos a su cargo.	<ul style="list-style-type: none">• Evaluar a los estudiantes las veces que lo crea necesario.• Resuelve ejercicios para verificar el buen funcionamiento de la herramienta.• Consultar información de algunos de los temas que componen el temario de su asignatura.

		<ul style="list-style-type: none"> • Resolver las dudas e inquietudes de los alumnos respecto del contenido y ejercicios de la herramienta. 	<ul style="list-style-type: none"> • Enviar los resultados de las evaluaciones hecha a los estudiantes. • Observar los registros de los ejercicios resueltos por los estudiantes, realizando los comentarios necesarios.
Estudiante	Es la persona que utiliza la herramienta	<ul style="list-style-type: none"> • Los estudiantes tienen una hora de trabajo la cual deben cumplir, para generar información que será evaluada. • Consultar la información que desee en el modulo de teoría. • Realizar las evaluaciones y actividades que le sean propuestas por el profesor. 	<ul style="list-style-type: none"> • Soluciona los ejercicios del modulo de problemas para verificar su grado de conocimiento. • Realiza las evaluaciones, cuando los profesores así lo quieran. • Estudiar o repasar los conceptos que se le entregan en el aula de clase. • Consultar las notas y observaciones que le sean enviadas por parte del profesor.
Administrador	Es la persona que se encargará de la administración de la herramienta, y del mantenimiento de la información.	<ul style="list-style-type: none"> • Realizar la administración de los usuarios (creación, modificación, consulta y eliminación). • Asignar y modificar permisos de acceso a los usuarios. • Mantenimiento de grupos de estudiantes. • Mantenimiento de la base de datos donde se encuentran almacenados los problemas a resolver. • Asignara permisos a personas ajenas a la clase para que trabajen en la 	<ul style="list-style-type: none"> • Utiliza la herramienta para crear, modificar y eliminar usuarios, grupos y ejercicios. • Actualizara la base de datos donde se encuentran los problemas a resolver.

		herramienta. • Asignara los horarios respectivos de trabajo a los estudiantes	
invitado	Representa a cualquier persona que no pertenezca a la clase y utilice la herramienta.	• Registrarse en la base de datos de la herramienta, para poder hacer uso de esta.	• Consultar la información de los temas que desee en el modulo de teoría. • Entrar al modulo de problemas y resolver los ejercicios que el desee.

Tabla 13. Actores de la herramienta

5.4.2 ENCONTRAR LOS CASOS DE USO

Conociendo que el punto de partida es el modelo del negocio, se encuentran las actividades donde participan los actores de la herramienta, proponiendo un caso de uso para cada rol de cada individuo que participa en la realización del modelo del negocio y que utiliza información de la herramienta.

Algunos casos de uso no se obtendrán directamente a partir de los casos de uso del modelo del negocio, estos nuevos casos de uso se detectaran al describir los casos de uso identificados adquiriendo un mayor conocimiento sobre los requisitos que deben ser soportados, y representarían funciones que debe llevar a cabo el software para lograr un objetivo asociado con un caso de uso ya existente.

Normalmente los casos de uso detectados de esta manera serán de soporte, puesto que no surgen directamente de la descripción de los procesos del negocio.

Algunos de los candidatos no llegarán a ser casos de uso por sí mismos; en cambio, podrán ser partes de otros casos de uso, hay que recordar que se intenta crear casos de uso fáciles de modificar, revisar, probar y manejar unitariamente.

Para determinar si un caso de uso es o no parte de otro, se debe tener en cuenta que sea completo por si mismo, es decir que no siempre se ejecute como continuación de otro caso de uso.

Los siguientes son dos criterios útiles para la identificación de los casos de uso:

- ❖ Cada ejecución satisfactoria de un caso de uso debe proporcionar algún valor, al actor iniciador del caso de uso, para alcanzar su objetivo, esto permite evitar encontrar casos de uso demasiado pequeños.
- ❖ El resultado de un caso de uso debe proporcionar valor a un actor en concreto, de tal manera que el caso de uso no resulte demasiado grande.

Cada caso de uso abarca un gran numero de funciones asociadas a una parte de la problemática global, por consiguiente, es necesario descomponerlos posteriormente para poder entender de manera detallada las características y los alcances del software a desarrollar. A continuación se presentará un diagrama de casos de uso general, en el cual se presenta una primera aproximación a los casos de uso de la herramienta.

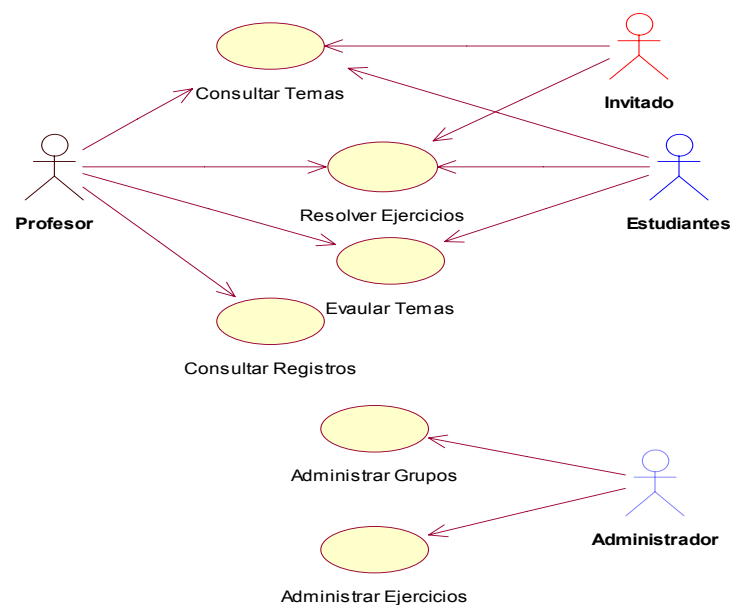


Figura 20. Modelo de casos de uso general.

En la siguiente tabla, se describirá el modelo anterior para efectos de una mejor comprensión.

Descripción del modelo general de casos de uso
<p>La herramienta ofrece a los usuarios diferentes tipos de servicio, que se utilizan en la asignatura Geometría descriptiva los cuales servirán de base para mejorar la dinámica de la clase, apoyando al proceso de enseñanza-aprendizaje que se presenta en la escuela de Diseño Industrial. Por lo tanto se ofrecen varios servicios que en este diagrama se presentan como casos de uso:</p> <ul style="list-style-type: none"> • El caso de uso – Consultar Temas – Este es utilizado por los actores participantes del proyecto “Profesor, estudiante e invitado” con el fin de buscar los temas que desee consultar o estudiar. Por tanto el estudiante en cualquier momento después de haber recibido la clase magistral dada por el profesor y si algunos conceptos mostrados en clase no están claros, el estudiante puede acercarse a la herramienta y consultar los conceptos que no comprendió, de igual manera puede hacerlo el invitado que desea conocer sobre los temas mas importantes que maneja la geometría descriptiva, claro que antes de consultar la información deben estar registrados en la base de datos de la herramienta, dado el caso si no lo está se registra y luego puede consultar la información que necesite, esta será desplegada en un listado para que el usuario escoja el tema que desee. De igual manera lo hará el profesor buscando los temas que necesite para aclarar alguna inquietud. Los temas estarán presentados de acuerdo al listado de temas que la escuela tiene para dictar la asignatura. Además de esto se encuentra el servicio de consultar ejemplos que se encuentra en el modulo de teoría y se encuentra organizado de la misma forma como se presentan los temas. Al final de cada tema se encuentra un vinculo que mostrara un ejemplo clásico de cómo resolver un ejercicio de acuerdo al tema que se esta tratando, esto ayudara a observar una forma de cómo resolver los ejercicios que se presentan en la asignatura. • El caso de uso – Resolver Ejercicios – Es utilizado por los actores participantes en el proyecto “profesor, estudiante e invitado”. En este modulo estos actores podrán resolver ejercicios los cuales serán entregados por temas organizados de la misma forma en que se encuentran en el modulo de teoría. Para llegar a utilizar este modulo los actores deben estar registrados en la base de datos de la herramienta, para poder resolver los problemas que ellos escojan, sino están registrados lo deben hacer para trabajar en este modulo. Después de estar registrados los actores escogerán el tema en que desean trabajar, aparecerá luego un listado de los ejercicios que puede resolver. Desplegándose el enunciado del problema y la hoja de dibujo correspondiente para resolverlo. • El caso de uso – Evaluar Tema – Este servicio esta desarrollado especialmente para los estudiantes, con el fin de que los profesores evalúen los temas que ya se han tratado en la asignatura. Los profesores en una de las horas asignadas por el administrador como hora de trabajo, podrán realizar la evaluación a los estudiantes, entonces entraran al modulo de evaluación, seleccionándose los ejercicios a resolver por los estudiantes, los resultados de las actividades realizadas en este modulo

serán guardados y entregados para su correspondiente verificación por parte del profesor que luego enviara los resultados. A este modulo no podrá entrar el invitado ya que el permiso otorgado no le permite utilizar este modulo.

- El caso de uso – Consultar Registros – Es un servicio al que tienen acceso el profesor y el administrador donde consultaran los registros almacenados en la base de datos de las horas de trabajo realizadas por los estudiantes. Estos registros le mostraran el desempeño de los estudiantes al resolver los problemas que aparecen en el modulo de ejercicios, también podrá ser utilizado en cierto momento como un método de evaluación, observando las actividades realizadas por cada alumno en su hora de trabajo.
- El caso de uso – Administrar Grupos – Esta es una función exclusiva del administrador de la herramienta, donde almacenara los listados de los grupos de estudiantes que los profesores le entregan, asignándole el debido permiso y las horas de trabajo respectivas para cada grupo.
- El caso de uso - Administrar Ejercicios – Es también exclusivo del administrador, quien estará encargado del mantenimiento y actualización de los ejercicios almacenados en la base de datos. El administrador estará en capacidad de almacenar nuevos ejercicios los cuales son entregados por los profesores y verificar que se cumpla la rotación de los ejercicios, con el fin de que los estudiantes no memoricen los ejercicios.

Tabla 14. Descripción del Modelo General de Casos de Uso.

En el diagrama anterior, aun cuando los casos de uso presentados deberán ser descompuestos para presentar una mayor claridad de lo que se pretende realizar, este presenta una visión global del software, y permite desde un comienzo reconocer 6 subsistemas o zonas, como se llamaran en este proyecto. Este aspecto es de bastante trascendencia en la planificación de las fases y en la arquitectura de la herramienta.

A continuación se presenta el diagrama de casos de uso para el actor estudiante y se describen un caso de uso en el que participa. Se remite al ANEXO C -*Modelo de Caso de Uso de la Herramienta*, página 208, para observar los demás casos de uso.

La descripción de cada caso de uso depende de la sencillez o complejidad del mismo, de esta forma si en ocasiones el caso de uso queda completamente explicado con su nombre, y no tiene caminos alternativos ni restricciones adicionales que deban ser resaltadas entonces solo bastará con una descripción breve, en nuestro caso será necesaria una descripción paso a paso del mismo.

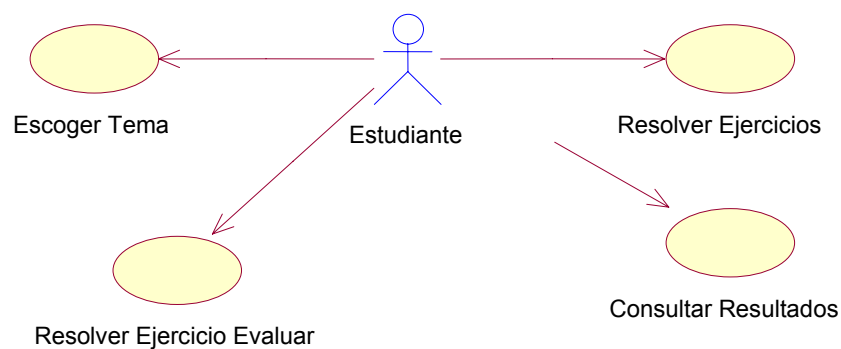


Figura 21. Modelo de Casos de uso – Casos de uso por actor.

Nombre	Escoger Tema
Breve Descripción	El estudiante utiliza este caso de uso para consultar o aclarar los conceptos vistos en la clase.
Descripción Paso a Paso Inicial	<p>Dado que ya conoce el tema al cual quiere consultar y que el estudiante tiene permiso para acceder al modulo de teoría :</p> <ol style="list-style-type: none"> 1. El estudiante inscrito en la asignatura llama este caso de uso, después de estar registrado en la base de datos de la herramienta, aparecerá luego la página principal donde se encuentran los diferentes módulos, entre ellos el de teoría, el estudiante realiza clic en el botón teoría. 2. Luego el software le mostrara a los clientes un listado de temas que se pueden escoger para su uso. 3. El Cliente escoge el tema a consultar. <p>El caso de uso finaliza y le da paso al caso de uso consultar subtemas. El CU termina cuando el estudiante se sale del modulo, desea consultar el ejemplo que se encuentra al final de cada tema, desea salir de la herramienta o escoger un subtema.</p>

Tabla 15. Descripción del Caso de uso Escoger Tema.

Nombre	Resolver Ejercicios
Breve Descripción	El estudiante utiliza este caso de uso para prepararse para las posibles evaluaciones que se desarrollan en la clase magistral o en la herramienta cuando el profesor lo desee.

Descripción Paso a Paso Inicial	<p>Para realizar este caso de uso el estudiante ya debió haber consultado los temas para aclarar dudas y mejorar sus conocimientos.</p> <ol style="list-style-type: none"> 1. El estudiante ya debió haber escogido el tema y el subtema con los cuales les mostrara una serie de ejercicios para resolver. 2. La herramienta Internamente relaciona los temas y los subtemas con los ejercicios almacenados en la B.D, mostrando un listado de los ejercicios a resolver. 3. El ejercicio se despliega en pantalla, mostrando su enunciado, las ayudas para resolverlo y un cronometro de tiempo que indica cuanto tardo desarrollando el ejercicio. 4. El estudiante procede a solucionarlo, en el momento de terminarlo escogerá el siguiente ejercicio, que es el siguiente en la lista. <p>Si el Estudiante lo desea puede salir del modulo y realizar otra operación terminando con el caso de uso.</p>
--	---

Tabla 16. Descripción del Caso de uso Resolver Ejercicio.

5.5 ACTIVIDAD: DETALLAR UN CASO DE USO.

Esta actividad sólo es necesaria cuando existen uno o varios casos de uso que aún no se han entendido y que pueden ser definitivos en el desarrollo de la herramienta. “El objetivo principal de detallar cada caso de uso es describir su flujo de sucesos en detalle, incluyendo como comienza, termina e interactúan con los actores”¹⁹. Donde un suceso es una secuencia de acciones que se ejecutan en una instancia del caso de uso, por tanto para detallar un caso de uso se deben tener en cuenta todas las alternativas o excepciones del camino básico.

Para poder detallar un caso de uso se tienen dos aspectos los cuales son: su descripción y su diagrama, pudiendo éste último ser un diagrama de estados, de actividad o de colaboración según sea el nivel de detalle que se quiera presentar.

¹⁹ El proceso unificado del desarrollo del software, Addison Wesley, Pág. 7.4.3., Pág. 147

En nuestro caso se utilizara un diagrama de estados y se pretende detallar el caso de uso Escoger Tema con el fin de dar una mayor claridad a la manera como los estudiantes y los usuarios de la herramienta deben realizar el proceso de escogencia de temas y de subtemas que se presentaran, estos de acuerdo al temario de la asignatura.

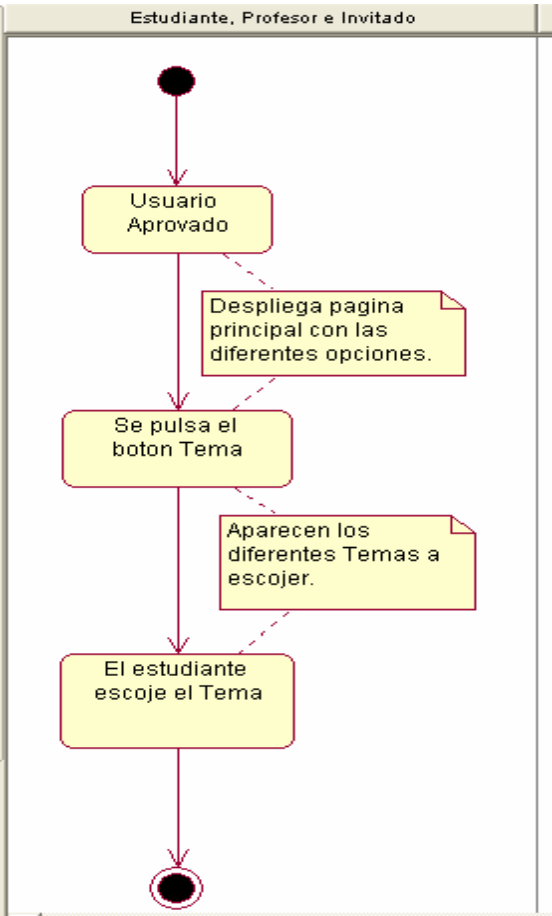


Figura 22. Diagrama de estados del caso de uso: Escoger Tema

Caso de Uso	Escoger tema.
Precondición	La persona que desea escoger un tema que se encuentre en la herramienta, antes que todo debe estar matriculado en la asignatura ya que esto le implica la asignación un permiso que le otorgara acceso directo a la herramienta y a sus servicios, sino lo esta debe registrarse para poder hacer uso del software.
Descripción o Flujo de Sucesos	El estudiante inscrito en la asignatura llama este caso de uso, después de estar registrado en la base de datos de la herramienta, aparecerá luego la página principal donde se encuentran los diferentes módulos, entre ellos el de teoría, el estudiante realiza clic en el botón teoría. Luego el software le mostrara a los clientes un listado de temas que se pueden escoger para su uso.

	<p>El Cliente escoge el tema a consultar.</p> <p>El caso de uso finaliza y le da paso al caso de uso consultar subtemas.</p>
Caminos Alternativos	En este caso no se encuentra un camino alternativo a seguir para llegar a finalizar este caso de uso, el usuario en cualquier momento puede dar por terminado el caso de uso y realizar alguna otra función que la herramienta le preste.
Poscondiciones	El caso de uso termina cuando se ha realizado la escogencia de los temas.
Requisitos especiales	Para acceder a este caso de uso lo que se necesita es registrarse en la base de datos.

Tabla 17. Descripción detallada del Caso de uso Escoger tema.

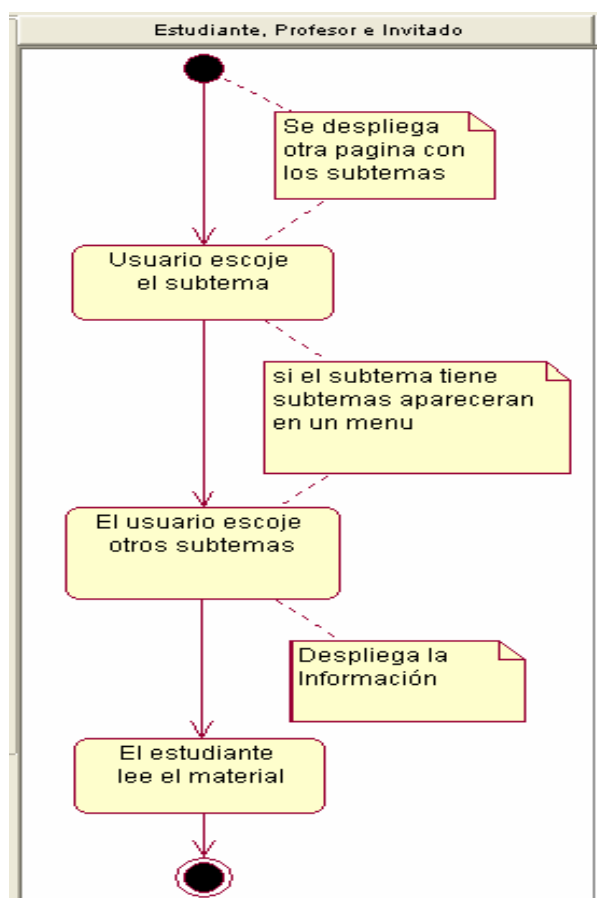


Figura 23. Diagrama de estados del caso de uso: Escoger Subtema

Caso de Uso	Escoger subtema.
Precondición	La persona que desea escoger un subtema que se encuentre en la herramienta, antes que todo debe estar registrado en la B. D, haber entrado al modulo de teoría y escogido un tema a consultar, con lo cual llegara al caso de uso escoger subtema.
Descripción o Flujo de Sucesos	<p>1. Después que el Cliente escoge el tema a consultar, la herramienta relaciona este tema con el que se encuentran almacenado en la B. D y busca todos los subtemas que contiene ese tema.</p> <p>2. El software despliega un listado de todos los subtemas que corresponde a ese</p>

	<p>tema a maneja de un menú lateral.</p> <p>3. El cliente selecciona el subtema que desee consultar, estos de acuerdo a un listado de temas que los profesores manejan en sus clases y el cual esta almacenado en la B .D de la herramienta.</p> <p>4. se despliega la información que la persona deseo consultar.</p> <p>5. El caso de uso finaliza.</p>
Caminos Alternativos	Si el usuario al momento de escoger el tema este no tiene subtemas pasara paso 5. El usuario en cualquier momento puede dar por terminado el caso de uso y realizar alguna otra función que la herramienta le preste.
Poscondiciones	El caso de uso termina cuando se ha realizado la escogencia de los subtemas que se desean observar.
Requisitos especiales	Para acceder a este caso de uso lo que se necesita es registrarse en la base de datos y haber escogido un tema para consultar.

Tabla 18. Descripción detallada del Caso de uso Escoger subtema.

5.6 ACTIVIDAD: ESTRUCTURAR EL MODELO DE CASOS DE USO

Paquetes de casos de uso	Casos de uso	Actores	Prioridad
Consultar Temas	Verificar Contraseña del Usuario.	Estudiante, profesor e invitado.	Importante
	Escoger Tema.	Estudiante, profesor e invitado.	Importante
	Consultar Ejemplos	Estudiante, profesor e invitado.	Importante
Resolver Ejercicios	Verificar Contraseña del Usuario	Estudiante, profesor e invitado.	Importante
	Escoger Tema Ejercicio.	Estudiante, profesor e invitado.	Importante
	Escoger Ejercicio	Estudiante, profesor e invitado.	Critico
Evaluar Tema	Verificar Contraseña del Estudiante.	Estudiante	Importante
	Escoger Tema Evaluar.	Estudiante	Importante
	Escoger Ejercicio Evaluar.	Estudiante	Importante
	Guardar solución.	Estudiante	Critico
Consultar Registros	Verificar Contraseña del profesor.	Profesor.	Importante
	Ver Registros del Trabajo de los Estudiantes.	Profesor.	Importante
	Crear Informes.	Profesor.	Importante

	Eliminar Informes	Profesor.	Critico
	Guardar Informes	Profesor.	Critico
Administrar Grupos	Crear Grupos.	Administrador.	Importante
	Modificar Grupos.	Administrador.	Importante
	Eliminar Grupos.	Administrador.	Importante
Administrar Usuarios	Crear Usuarios	Administrador.	Importante
	Modificar Usuarios	Administrador.	Importante
	Eliminar Usuarios	Administrador.	Importante
Administrar Ejercicios.	Insertar los Ejercicios en la base de datos.	Administrador.	Importante

Tabla 19. Casos de Usos Identificados en la Fase de Inicio.

5.7 ACTIVIDAD: CAPTURAR REQUISITOS ADICIONALES

Los requisitos adicionales son aquellos requisitos no funcionales que no pueden asociarse a ningún caso de uso en concreto sino que por el contrario tienen impacto en varios casos de uso o en ninguno. En esta actividad se pretende listar una serie de requisitos no funcionales para especificar algunas propiedades del software que hasta este punto se han determinado. A continuación se presenta una lista de estos requisitos.

Requisitos de Interfaz	<ol style="list-style-type: none"> 1. Se deberá realizar una interfaz que permita rápidamente acceder al sistema. 2. El sistema debe tener pocos requerimientos para las maquinas de los usuarios.
Requisitos de Plataforma Hardware	<ol style="list-style-type: none"> 3. Servidor: PC Pentium III mínimo 256 RAM 4. Clientes: PC mínimo Pentium
Requisitos de Red	<ol style="list-style-type: none"> 5. Utilización de la red interna de la Universidad Industrial.
Restricciones de Diseño	<ol style="list-style-type: none"> 6. Se deberá separar la capa presentación de los datos de la lógica de la herramienta.
Restricciones de Implementación	<ol style="list-style-type: none"> 7. Software del Servidor: Sistema Operativo: Windows 2000. Servidor Web: Apache. 8. Software del Cliente: Netscape 4.0 o Internet Explorer 5.0 o mayor.

Tabla 20. Requisitos Adicionales.

5.8 ANÁLISIS.

Cuando se termina el flujo de trabajo de los requisitos, aun quedan algunos detalles por comprender y varios temas sin resolver; es por ello que resulta importante el flujo del análisis, ya que a través de este se logran resultados básicos para el comienzo del flujo de diseño. Los objetivos generales del flujo de trabajo del análisis son analizar los requisitos, refinarlos y estructurarlos en un modelo de objetos que ayude a conseguir una comprensión mas precisa de los requisitos y que además sirva como primera impresión del modelo de diseño. En esta fase, el resultado es un modelo inicial del análisis, en el cual se estructuran los requisitos de manera que facilite su entendimiento, preparación, modificación y mantenimiento, con lo cual se soluciona aspectos relativos a las interferencias, concurrencia y conflictos entre los casos de uso, logrando establecer un modelo con independencia entre los casos de uso.

En el modelo de análisis se tratara de describir la herramienta desde el lenguaje del desarrollador. Realizando una vista interna de las clases y paquetes que posee; encontrando redundancias y definiendo lo que realiza cada caso de uso; mostrando su respectivo análisis con el fin de definir con precisión los casos de uso y otorgando una ayuda para guiar el establecimiento de la arquitectura candidata. Por lo tanto, en esta fase, el modelo solo será un esbozo inicial que será refinado en las fases posteriores, en especial en la fase de elaboración.

5.8.1 ANÁLISIS DE LA ARQUITECTURA

El objetivo del análisis de la arquitectura es presentar un esbozo del modelo de análisis y de la arquitectura mediante la identificación de los paquetes del análisis y las clases de entidad obvias obtenidas a partir del modelo de dominio del negocio.

5.8.1.1 PAQUETES DEL ANÁLISIS GENERAL

Debido a la creciente complejidad del modelo de análisis, es necesario dividirlo en partes pequeñas y manejables denominadas paquetes del análisis los cuales proporcionan un medio para organizar los artefactos del modelo de análisis en piezas manejables.

Estos paquetes se crean con base en los requisitos funcionales, representados en casos de uso y en las entidades del dominio del negocio, los casos de uso que lo componen deberían estar fuertemente relacionados y ser débilmente acoplados, o sea, que la dependencia de sus contenidos debe ser mínima.

Para ello se asignan los casos de uso a un paquete concreto siguiendo los siguientes criterios:

- ◆ Agrupar los casos de uso que dan soporte a un determinado proceso de negocio.
- ◆ Cada uno representaría un interés específico del análisis.
- ◆ Agrupar los casos de uso que dan soporte a un determinado actor de la herramienta.
- ◆ Se Elaboran a partir de los requisitos funcionales y de las características del negocio.

Se identifican paquetes de análisis más generales que agrupan aspectos, entidades o funcionalidades compartidas entre dos o más paquetes, estos a partir del modelo del negocio y de los casos de uso encontrados.

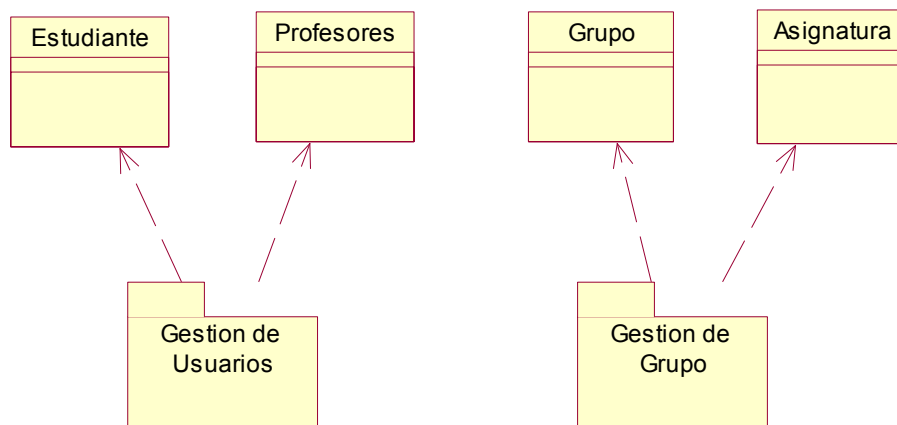


Figura 24. Identificación de paquetes del análisis a partir de clases de dominio.

El paquete “Gestión de usuarios”, contiene todos los casos de uso que soportan el proceso de negocio de aprendizaje-enseñanza de la herramienta, utilizado especialmente por los estudiantes de las diferentes carreras que ven la asignatura de Geometría Descriptiva. El paquete “Gestión de Grupo”, Contiene los casos de uso que soportan el proceso de negocio de Administración de la misma.

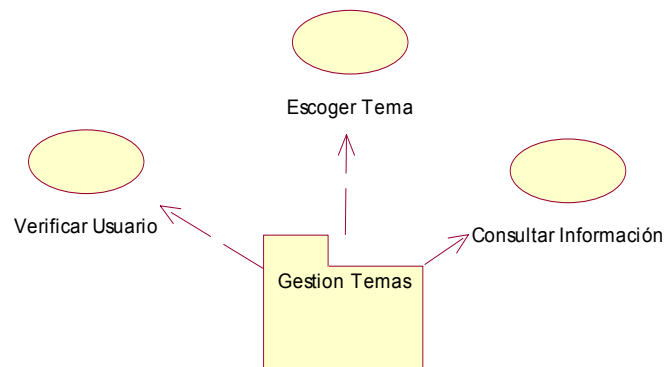


Figura 25. Paquetes de análisis Gestión Temas a partir de casos de uso

El paquete “Gestión de Temas”, contiene los casos de usos que soporten el proceso de entregar la información que el usuario requiere para desempeñarse con éxito en la asignatura.

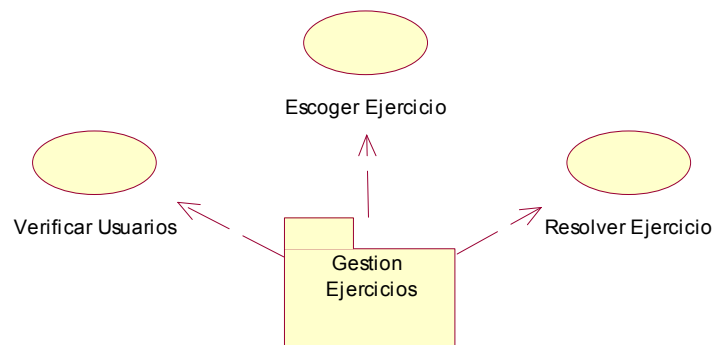


Figura 26. Paquetes de análisis Gestión Ejercicios a partir de casos de uso

El paquete “Gestión de Ejercicios”, contiene los casos de usos que soporten el proceso de aprendizaje por parte de los estudiantes, con el desarrollo de los diferentes ejercicios que contiene la herramienta.

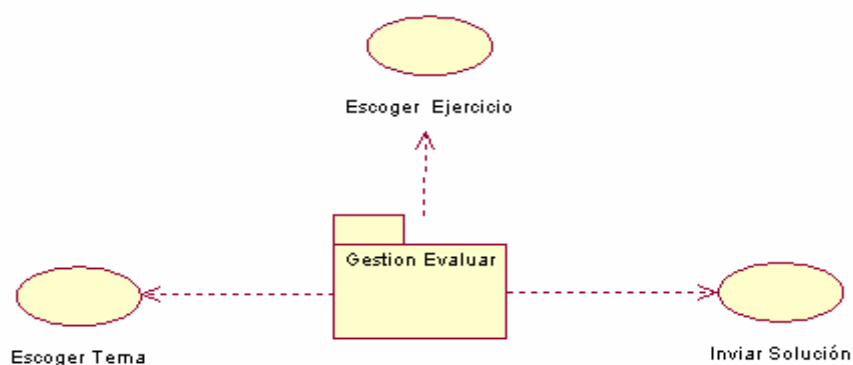


Figura 27. Paquetes de análisis Gestión Evaluar a partir de casos de uso

El paquete “Gestión de Evaluar”, contiene los casos de usos que soporten el proceso de evaluación realizado por los estudiantes, en los horarios asignados por los profesores, constituyendo este proceso como parte importante para la calificación final de cada estudiante.



Figura 28. Paquetes de análisis Gestión consultar registros a partir de casos de uso

El paquete “Gestión consultar Registros”, contiene los casos de usos que soporten el proceso de evaluación realizado por parte de los profesores, tomando los registros de trabajo de cada estudiante, con el fin de entregar una calificación al final del semestre.

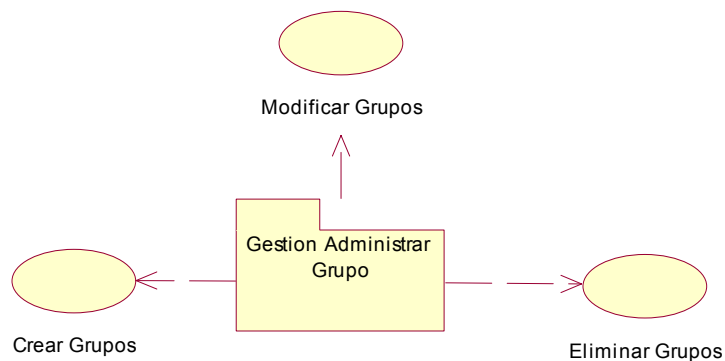


Figura 29. Paquetes de análisis Gestión Adm. Grupo a partir de casos de uso

El paquete “Gestión Adm. Grupo”, contiene los casos de usos que soporten el proceso de administración de los grupos de clase que estarán frecuentando el laboratorio de informática donde se encuentra la herramienta, este proceso es realizado por parte del administrador.

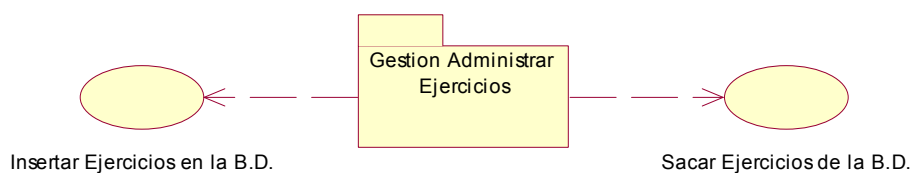


Figura 30. Paquetes del análisis Gestión Adm. Ejercicios a partir de casos de uso

El paquete “Gestión Adm. Ejercicios”, contiene los casos de usos que soporten el proceso de administración de los ejercicios que contiene la herramienta, este proceso es realizado por el administrador.

Todos estos paquetes evolucionarán y se reestructurarán a medida que se analicen más casos de uso y se determinen las clases de análisis y probablemente se convertirán en subsistemas en el modelo de diseño. En la figura anterior se presentan los paquetes del análisis obtenido del modelo de dominio del negocio y de los casos de uso del software.

El siguiente diagrama muestra las dependencias entre los paquetes obtenidos anteriormente, como se observa, los paquetes de gestión Temas y gestión Ejercicios

utilizan el paquete Gestión de usuarios debido a que una asignación esta vinculada directamente a los usuarios que trabajan con la herramienta.

Así mismo el paquete Gestión de Administrar Grupo encarga de mantener actualizada la información de los grupos que cada semestre manejan la herramienta, por lo tanto se necesita del paquetes de Gestión de Cursos.

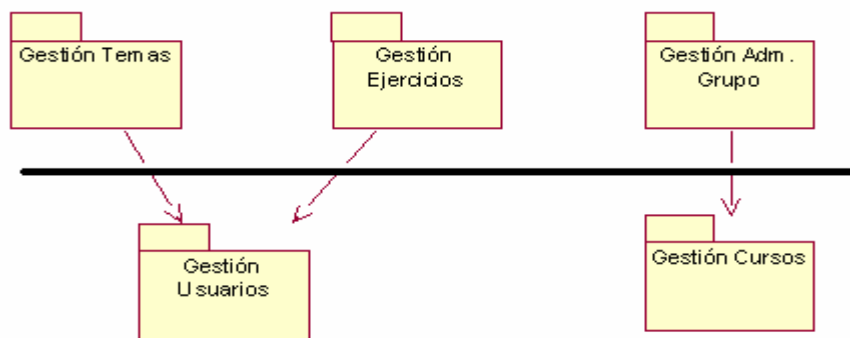


Figura 31. Dependencias y Capas de paquetes de análisis.

5.8.2 IDENTIFICACIÓN DE CLASES DE ENTIDAD OBIAS.

En esta fase se deben definir algunas clases de entidad, las cuales han sido extraídas con base en el modelo de dominio realizado durante el flujo de análisis, preferiblemente aquellas que sean significativas para la arquitectura, utilizando estas clases para modelar la información que es persistente y se convertirán en entidades del diagrama entidad relación en el modelo de diseño y en estructuras como tablas en la base de datos en la fase de implementación.

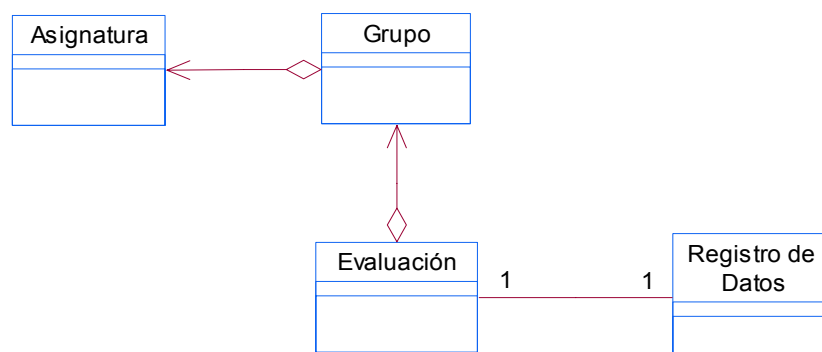


Figura 32. Clases de Entidades Derivadas del Modelo de Dominio.

5.8.3 ANÁLISIS DE CASOS DE USO.

En la fase de inicio se necesita refinar o analizar los casos de uso que puedan afectar la viabilidad del proyecto y aquellos que sean fundamentales en la búsqueda de la arquitectura correcta, para ello es suficiente con una pequeña proporción de los casos de uso. Para realizar este análisis primero se identificarán las clases del análisis y como se relacionan entre si para llevar a cabo el flujo de sucesos del caso de uso.

5.8.3.1 IDENTIFICACIÓN DE CLASES DEL ANÁLISIS.

Para la identificación de las clases del análisis se tomará el caso de uso Asignar grupos.

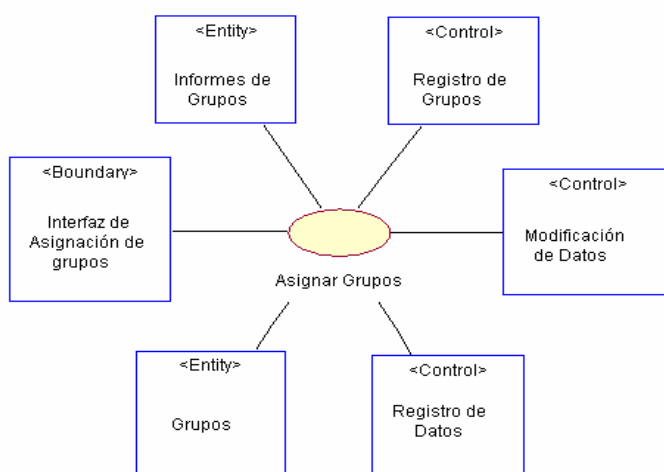


Figura 33. Clases de Análisis del Caso de Uso Asignar Grupos.

Como se ve en la figura se han identificado dos clases de entidad²⁰: Grupos e Informe de Grupos, estas dos clases son obtenidas del modelo de negocio y de la misma descripción del caso de uso realizada durante la captura de requisitos.

En la primera se guarda los datos correspondientes a los grupos que los profesores van ha dirigir y trabajar con la herramienta en la sala de cómputo, en esta información

²⁰ Las clases de entidad, se utilizan para modelar información persistente que después guiarán el desarrollo del modelo entidad relación y la posterior implementación de la base de datos.

se encuentra el día, la hora, año, semestre, etc, correspondiente a los grupos de trabajo, en la segunda esta toda la información que se almacena en la base de datos requerida para crear un informe de grupos, consulta hecha por los profesores.

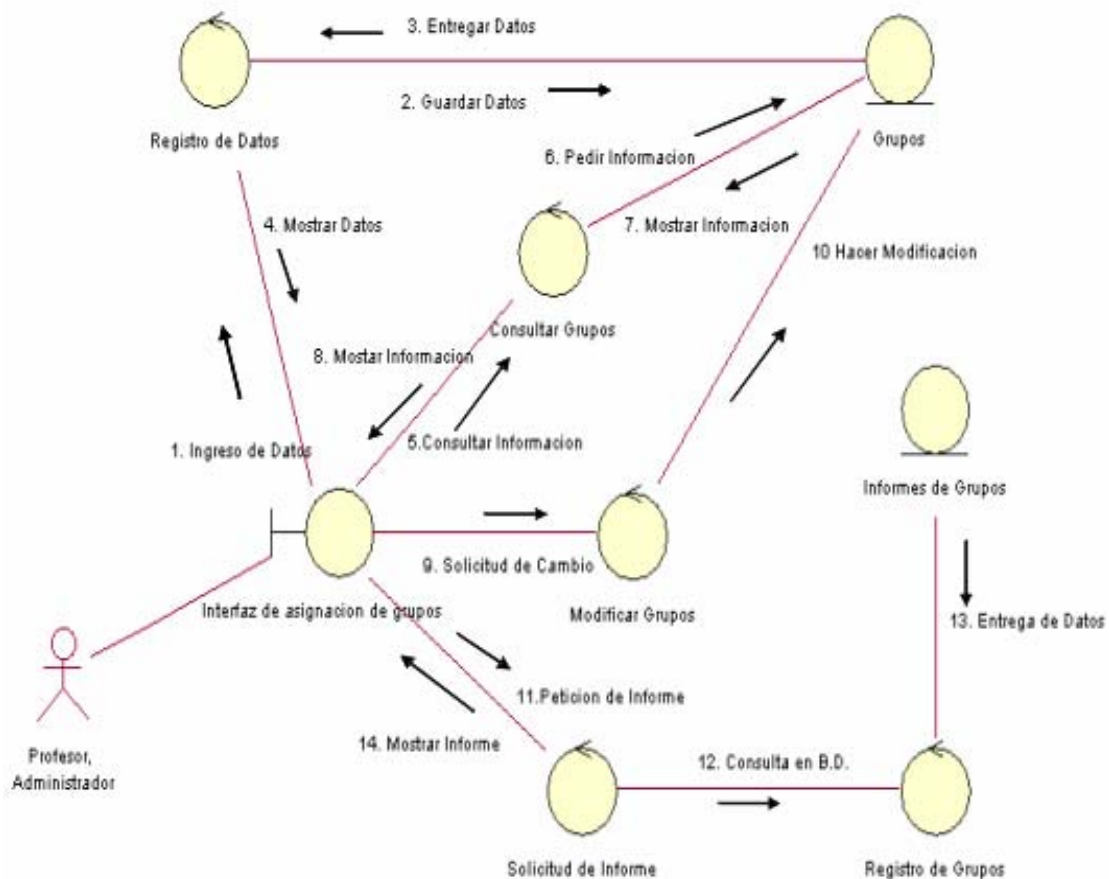


Figura 34. Diagrama de Colaboración de las Clases de Análisis, Caso de Uso Asignar Grupos.

Se encuentra igualmente una clase de interfaz²¹ llamada interfaz de Asignación de Grupos, a través de la cual el administrador podrá guardar la información de los grupos suministrada por los profesores, almacenando esta información en la base de datos de la herramienta.

²¹ Las clases de Interfaz se utilizan para modelar la interacción del actor con el sistema, tales como recibir o enviar información.

5.9 EL FLUJO DE TRABAJO DEL DISEÑO

En esta fase, el objetivo principal del flujo de trabajo de diseño es esbozar un modelo de diseño de la arquitectura candidata, con objeto de incluirlo en la descripción de la arquitectura preliminar. El diseño de casos de uso e interfaces se deja para fases posteriores. A lo largo del flujo de trabajo del diseño, los desarrolladores identificarán, a medida que evolucione el diseño, nuevos candidatos para ser subsistemas, interfaces, clases y mecanismos de diseño genérico, para luego ser refinados y mantenerlos.

5.9.1 ACTIVIDAD: DISEÑO DE LA ARQUITECTURA

Para el diseño de la arquitectura se realizará un esbozo de los modelos de diseño y despliegue identificando los nodos, configuraciones de red, subsistemas e interfaces y clases de diseño relevantes para la arquitectura y mecanismos de diseño que tratan requisitos comunes, tal como se capturaron durante el análisis sobre las clases.

5.9.1.1 IDENTIFICACIÓN DE NODOS Y CONFIGURACIONES DE RED

Las configuraciones físicas de red suelen tener una gran influencia sobre la arquitectura del software. En este proyecto en particular se utilizará una arquitectura de tres capas en la cual la interacción con el cliente se deja en una capa, la funcionalidad de la base de datos en otra y la lógica del negocio o de la aplicación en la tercera. Este sistema será ejecutado sobre un nodo servidor, donde residen la capa de datos y la capa lógica; y varios nodos cliente donde se ejecuta la capa de presentación.

En el Nodo servidor se encuentra la capa de datos y la de lógica o aplicaciones, aquí se encuentran almacenados los datos del software, en los nodos clientes se ejecuta la capa de presentación. La comunicación entre el nodo del cliente y del servidor se

realiza mediante el protocolo TCP/IP dentro de la Intranet de la Escuela de Diseño Industrial.

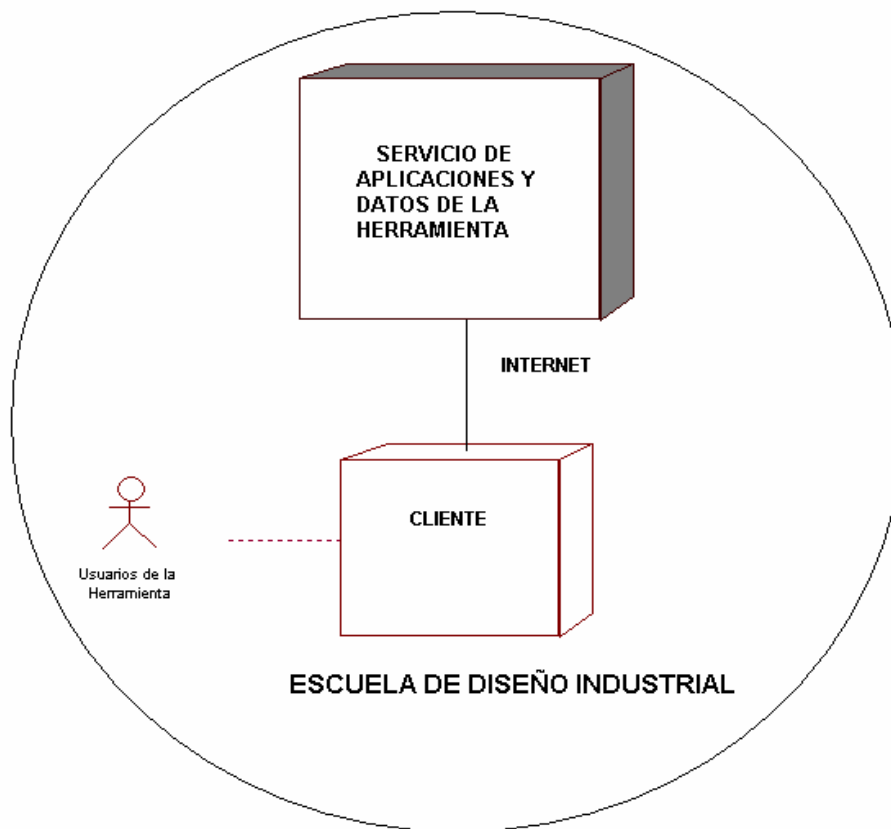
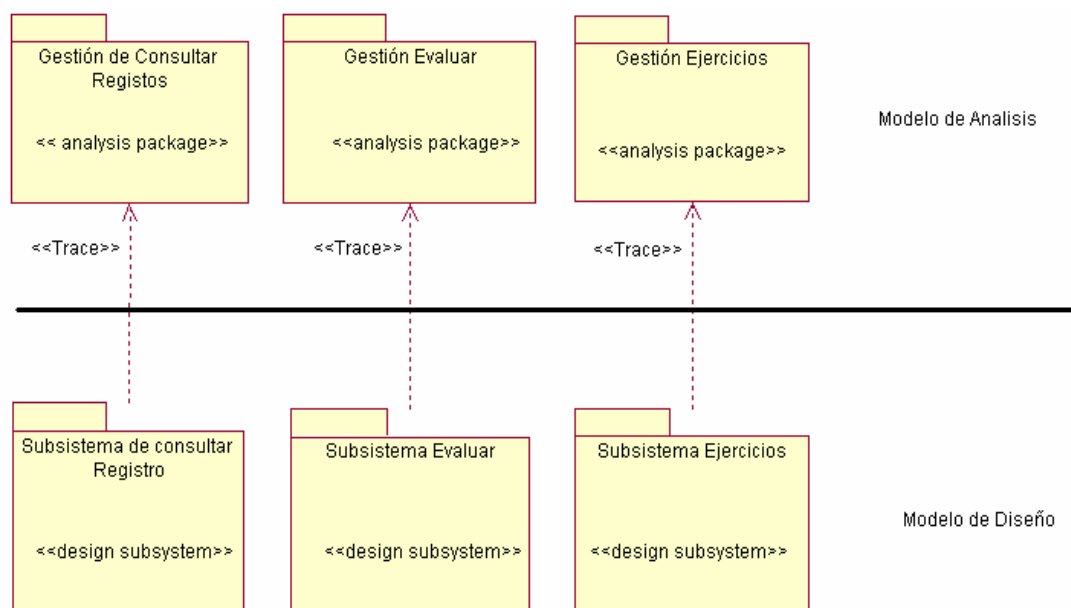
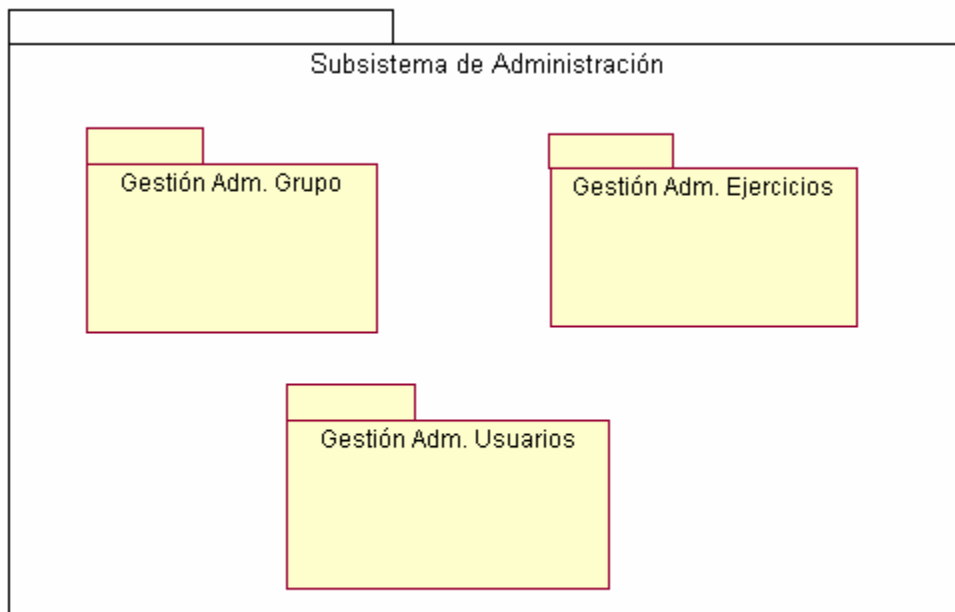


Figura 35. Diagrama de Despliegue de GEDISO 1.0.

5.9.1.2 IDENTIFICACIÓN DE SUBSISTEMAS Y DE SUS INTERFACES

Los subsistemas constituyen un medio para organizar el modelo de diseño en piezas manejables, en este caso se han identificado inicialmente como una forma de dividir el trabajo de diseño, dado que se pueden identificar fácilmente a partir de los requerimientos iniciales de la herramienta. Estos subsistemas han sido tomados a partir de los paquetes de análisis, los cuales han evolucionado para convertirse en subsistemas de diseño. Se observa en las siguientes figuras los subsistemas de diseño encontrados.



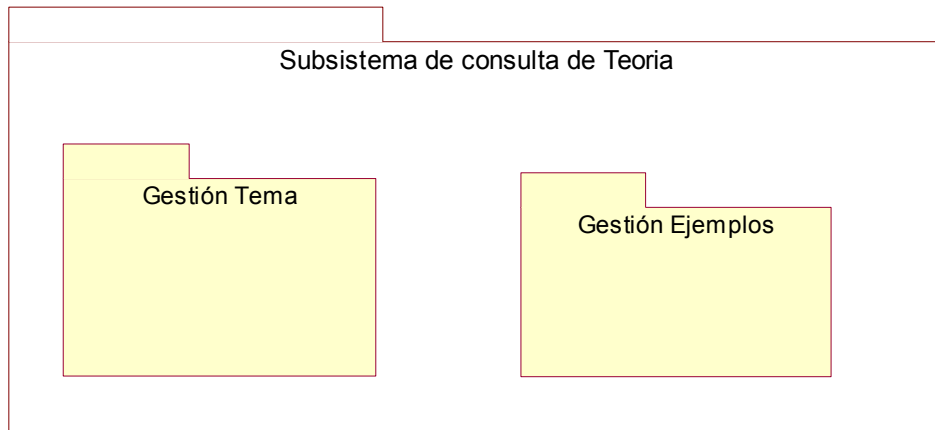


Figura 36. Identificación de subsistemas a partir de los paquetes de análisis.

CAPÍTULO 6

EVALUACIÓN DE LA FASE DE INICIO

Si se pretende conocer los beneficios de la forma de trabajo iterativo, es necesario evaluar los avances al final de cada iteración o fase. En la fase de inicio el resultado de esta evaluación es la decisión de si seguir adelante o abandonar el proyecto.

6.1 CUMPLIMIENTO DE LOS CRITERIOS

El desarrollo de esta fase estuvo orientado a visualizar el alcance del proyecto y a determinar la viabilidad del mismo. Para ello, se realizó un estudio básico del negocio, a través de los diagramas de casos de uso y de objetos, de los objetivos de la organización. Este estudio permitió tener una visión global del ámbito del software: de los actores y los casos de uso principales que deberían ser tomados en cuenta por el mismo; y de esta manera lograr una mejor comprensión de sus principales requisitos.

Además se limitó el alcance de la herramienta, se determinó que este proyecto apoyara la actividad académica dentro de la escuela de Diseño Industrial, en especial dentro del aula de clase de la asignatura Geometría Descriptiva, Generando un espacio para buscar información, para la realización de Ejercicios, brindando un canal para que los profesores lleven un control de los avances en los estudiantes por el desarrollo de evaluaciones y consultando el registro de las horas de trabajo para obtener una realimentación oportuna.

En esta fase se determinó una arquitectura candidata que sea viable para el desarrollo del proyecto, dadas las limitaciones de tiempo y dinero, pero sin sacrificar la robustez y seguridad del software. Se determinó realizar una herramienta en un ambiente Web para que sea de fácil acceso de cualquier usuario sin necesidad de demasiados requerimientos en su equipo y sin la necesidad de realizar una instalación de software adicional en su sistema.

Además se obtiene una independencia total con el software escogido en el servidor y los usuarios podrán contar con una interfaz conocida que les permita una fácil adaptación al sistema. Por ultimo se analizo cada uno de los riesgos más importantes que pueden surgir al realizar este proyecto, eliminándolos cuando fuera posible o mitigando su impacto en los casos restantes.

6.2 DESARROLLO DEL MODELO DE CASOS DE USO

Dada la importancia de los casos de uso para el progreso del proyecto, se presenta una Tabla con los casos de uso que hasta el momento han sido identificados. Para cada uno de ellos, se define el nivel de conocimiento y un estado, los cuales son los siguientes:

- ◆ *Identificado:* Se ha determinado la conveniencia de crear este caso de uso según los criterios presentados para encontrar casos de uso relevantes.
- ◆ *Descrito:* Se ha realizado una descripción breve del caso de uso.
- ◆ *Diseñado e Implementado:* Estos dos estados han sido agrupados dado que en algunas situaciones antes de llevarse a cabo la implementación de un caso de uso se realiza su diseño a través de diagramas de secuencia.

En los demás casos con el análisis de los casos de uso es suficiente para realizar su implementación.

Es importante destacar que no necesariamente todos los casos de uso requieren un análisis exhaustivo para su total comprensión, por su sencillez son comprendidos rápidamente y por lo tanto solo requieren una breve descripción del mismo.

Casos de Uso Identificados	Identificado	Descrito	Diseñados e implementados	¿Se comprende Completamente?
Verificar Registro del estudiante.	F. Inicio	F. Inicio		SI
Escoger Tema	F. Inicio	F. Inicio		SI
Consultar Ejemplos	F. Inicio	F. Inicio		SI

Escoger Tema Ejercicio	F. Inicio	F. Inicio		SI
Escoger Ejercicio	F. Inicio	F. Inicio		SI
Escoger Tema Evaluación	F. Inicio	F. Inicio		SI
Escoger Ejercicio Evaluar	F. Inicio	F. Inicio		SI
Ver registros	F. Inicio	F. Inicio		SI
Crear Informes	F. Inicio	F. Inicio		SI
Crear Grupos.	F. Inicio	F. Inicio		SI
Modificar Grupos.	F. Inicio	F. Inicio		SI
Eliminar Grupos.	F. Inicio	F. Inicio		SI
Insertar los Ejercicios en la base de datos.	F. Inicio	F. Inicio		SI
Sacar los Ejercicios de la base de datos.	F. Inicio	F. Inicio		SI

Tabla 21. Casos de Uso identificados hasta ahora.

6.3 VISTA DE LA ARQUITECTURA

Como se ha dicho desde que se presentó el Proceso Unificado para el desarrollo de este proyecto, esta metodología esta centrada en la arquitectura la cual abarca la organización del sistema software, los elementos estructurales que compondrán el sistema y sus interfaces, así como su comportamiento y colaboraciones entre elementos. Al describir la arquitectura se obtiene, por lo tanto, una mayor comprensión del sistema, se organiza el desarrollo y se fomenta la reutilización.

Por tanto se construye una arquitectura que permita implementar los casos de uso del sistema de una forma económica, además se debe tener en cuenta otros factores como el sistema operativo y de bases de datos a escoger, los productos para el desarrollo del sistema, los sistemas heredados, los estándares, políticas y los requisitos no funcionales en general.

En esta fase, esta vista esta limitada solo a la búsqueda de una arquitectura candidata, la cual se soporta solo sobre los casos de uso más importantes para el cliente, con mayor riesgo o que afectan la viabilidad del proyecto.

6.4 CUMPLIMIENTO DE LOS CRITERIOS

El desarrollo de esta fase estuvo orientado a visualizar el alcance del proyecto y a determinar la viabilidad del mismo. Para ello, se realizó un estudio básico del negocio, a través de los diagramas de casos de uso y de objetos, de los objetivos de la organización y de la experiencia como estudiante en un ambiente universitario. Este estudio permitió tener una visión global del ámbito del sistema: de los actores y los casos de uso principales que deberían ser tomados en cuenta por el mismo; y de esta manera lograr una mejor comprensión de sus principales requisitos.

Además se limitó el alcance del sistema. Se determinó que este proyecto apoyara la actividad académica dentro de la asignatura Geometría Descriptiva, en especial dentro de un aula de clase creando espacios de para compartir información, realizar ejercicios, y brindando un canal para que los profesores realicen evaluaciones que ayudaran a conocer el grado de conocimiento de los estudiantes.

En esta fase se determinó una arquitectura candidata que sea viable para el desarrollo del proyecto dado las limitaciones de tiempo y dinero, pero sin sacrificar la robustez y seguridad del sistema. Se determinó realizar el sistema en un ambiente Web para que sea de fácil acceso por cualquier usuario sin necesidad de demasiados requerimientos en su equipo y sin la necesidad de realizar una instalación de software adicional en su sistema.

Además se obtiene una independencia total con el software escogido en el servidor y los usuarios podrán contar con una interfaz conocida que les permita una fácil adaptación al sistema. Por último se analizó cada uno de los riesgos más importantes que pueden surgir al realizar este proyecto, eliminándolos cuando fuera posible o mitigando su impacto en los casos restantes.

FASE DE ELABORACION

El objetivo principal de la fase de elaboración es construir una arquitectura estable para guiar el software a lo largo de su vida futura. Para lograr el objetivo mencionado se identifican alrededor del 80% de los casos de uso, para describir detalladamente entre el 40% y el 80% de todos los casos de uso. Esta fase también lleva el estudio de la herramienta propuesta al punto de planificar la fase de construcción con gran precisión.

Para llegar a estas instancias, se ha formulado una arquitectura inicial, que permite conocer como construirla para que abarque las partes difíciles y novedosas del proyecto²², identificando los riesgos más serios, estudiándolos para encontrar las soluciones y realizando un análisis del negocio detallado que permita entrar en la siguiente fase de desarrollo del software.

Los objetivos específicos de esta fase para llevar a cabo el desarrollo del proyecto sin tener que pensar en la viabilidad del proyecto son:

- ◆ Establecer una base de la arquitectura sólida para guiar el trabajo durante la fase de construcción y transición, así como en las posteriores generaciones de la herramienta.
- ◆ Continuar la observación y control de los riesgos críticos que aún queden, e identificar los nuevos riesgos que puedan aparecer, tratando de conocer el impacto que puedan tener en el desarrollo del proyecto.
- ◆ Recopilar los requisitos que aun quedan pendientes y formular lo casos de uso para el 80 por ciento de los requisitos funcionales.

Hacia el final de esta fase, se pretende haber acumulado la información necesaria para planificar la fase de construcción, obteniéndose la información necesaria para tener un análisis del negocio claro y concreto.

²² Tomado del Proceso Unificado de desarrollo de software, Ivar Jacobson, pagina 345.

Capítulo 7.

PLANEACION

Al comienzo de la fase de elaboración, se recibe de la fase de inicio un modelo de casos de uso parcialmente completo, una descripción de la arquitectura candidata, unos bosquejos de un modelo de análisis, un modelo de diseño que servirá de guía y punto de partida para iniciar la búsqueda de una arquitectura estable que sea consistente durante la construcción de la herramienta.

Para cumplir con esta fase se adoptará un punto de vista general de la herramienta, en algunos casos, en donde predominen los riesgos técnicos, se necesitará profundizar para establecer una arquitectura sólida, tomando decisiones en ésta basados en la comprensión del sistema en su totalidad: su ámbito, sus requisitos funcionales y no funcionales.

7.1 FLUJOS, ACTIVIDADES Y PRODUCTOS A ENTREGAR

En la fase de elaboración se busca una arquitectura estable que soporte el desarrollo posterior en la fase de construcción, “durante el diseño de la arquitectura, se recopilara, analizara, se diseñara, se probara e implementara solo los requisitos relevantes desde el punto de vista de la arquitectura”²³ de la herramienta.

Para lograr este fin se deben obtener nuevos casos de uso y detallar los necesarios para entenderlos, además realizar un análisis de los paquetes que componen la herramienta y el posterior diseño de los subsistemas.

Como resultado de esta fase se entregara una nueva versión de todos los modelos: casos de uso, análisis, diseño, despliegue e implementación, también, se obtendrá una descripción de la arquitectura, la lista de riesgos actualizada y al final de la fase se deben definir el diseño de la base de datos e interfaz.

²³ Tomado del Proceso Unificado de desarrollo de software, Ivar Jacobson, pagina 349.

Capítulo 8.

ARQUITECTURA BASE

Durante este capítulo se describirá brevemente como se procede para realizar cada una de las actividades involucradas en esta fase y que no hayan sido descritas en la viabilidad del proyecto.

Para el desarrollo de cada actividad se debe establecer prioridades sobre los casos de uso y realizar actividades de análisis, diseño e implementación, a nivel de la arquitectura de la herramienta, también se desarrollaran pruebas a los casos de usos que compone la línea base de la arquitectura.

Los productos generados solo presentaran lo necesario para el entendimiento de la metodología permitiendo orientar al lector hacia el análisis de los productos generados en esta fase, los cuales se presentan como anexos al final de este libro.

8.1 ACTORES Y CASOS DE USO

Para esta actividad se identifican los casos de uso adicionales no identificados en la fase de inicio. Se es necesario comprender alrededor del 80 por ciento de los casos de uso para alcanzar los objetivos de esta fase, no es necesario detallarlos todos, solo aquellos que permitan no pasar por alto nada que pueda tener un impacto en la arquitectura y que ayude a cumplir con los objetivos planteados en esta fase.

8.1.1 CASOS DE USO

En la fase de inicio se mostró un diagrama de casos de uso general, que sirvió de aproximación a los casos de uso del sistema presentando una visión global del software, para poder tener una mayor facilidad de encontrar los casos de usos estos fueron agrupados de acuerdo a los paquetes de los cuales forman parte, lo que se hizo en la fase de inicio de la herramienta

Para esta segunda fase el diagrama de casos de uso que se entregara será descompuesto y ampliado para que se ajusten a los criterios de esta, obteniendo una mayor comprensión de estos.

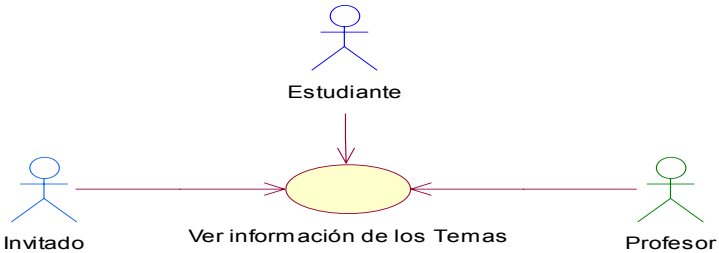


Figura 37. Modelo de Casos de Uso. Diagrama de Información de los Temas.

Descripción del Modelo Ver información de los Temas
<p>La herramienta permite ver a los actores, Profesor, Estudiante, ver la información de los temas interés correspondiente al pensum académico de la asignatura geometría descriptiva. Esto se hace a través del caso de uso ver información de los Temas. En este caso de uso el navegante solicita la información a la herramienta quien se encarga de enviarla al cliente toda la teoría que necesite sobre un tema específico. Para ello se utiliza el protocolo http utilizado por las paginas Web que contiene la herramienta.</p>

Tabla 22. Descripción del Modelo de Casos de Uso: Diagrama de ver Información de los Temas.

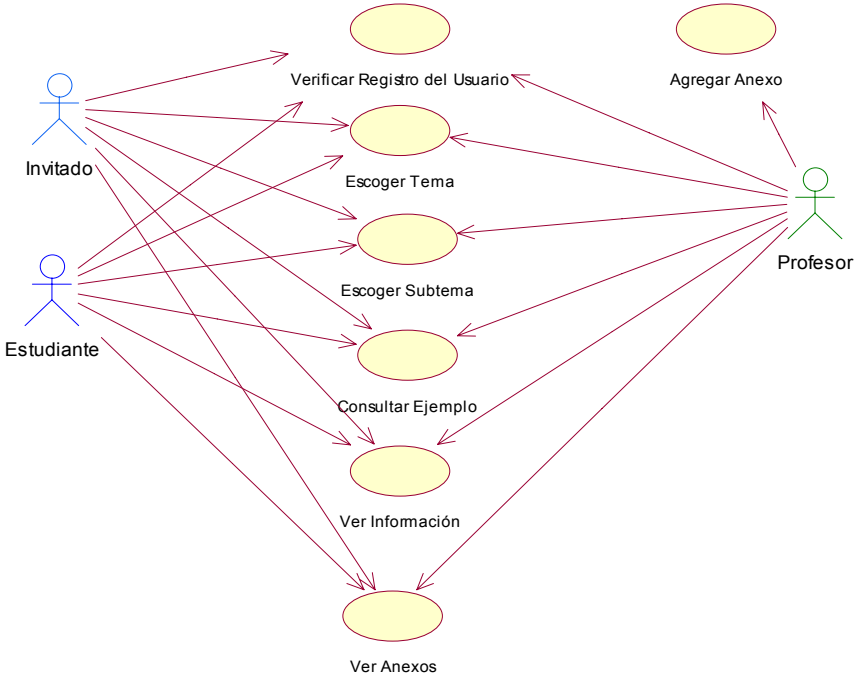


Figura 38. Modelo de Casos de Uso. Diagrama de consultar Temas.

Descripción del Modelo de Consultar Temas
<p>La Herramienta ofrece a los estudiantes la opción de consultar todos los temas que se encuentran en el pensum académico de la asignatura.</p> <p>Para una mejor comprensión de la teoría los temas están organizados por de mayor importancia en la asignatura y estos a su vez contienen los subtemas respectivos. Así, los diferentes contenidos pueden ser accedidos por los estudiantes sin ninguna dificultad, dado que estos fueron desarrollados en ambiente Web, conocido por la mayoría de las personas.</p> <p>El profesor también tiene acceso a los diferentes temas que contiene la herramienta de la misma manera que lo hacen los estudiantes. Además este caso de uso le permitirá corroborar sus conocimientos en algún tema de interés. De la misma manera que lo hace el profesor y el estudiante el software permitirá la entrada a un invitado que puede consultar la información que desee, además de consultar los ejemplos que este modulo contiene.</p> <p>El profesor tiene la opción de agregar algún anexo a manera de comentario para completar o aclarar la información que la herramienta posee.</p>

Tabla 23. Descripción del Modelo de Casos de Uso: Diagrama de Consultar Temas.

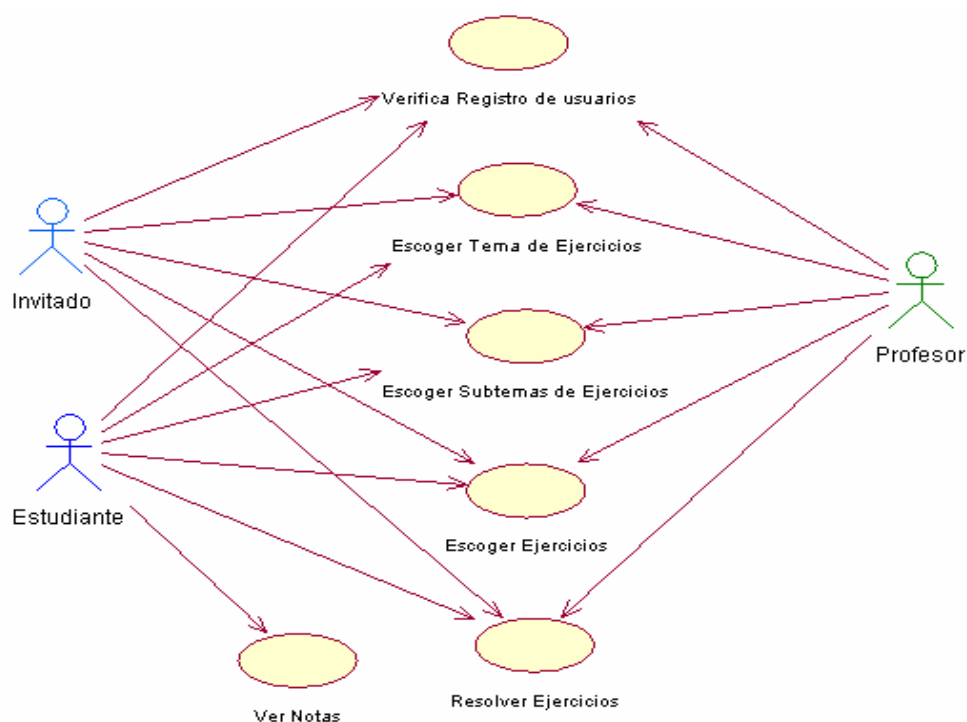


Figura 39. Modelo de Casos de Uso. Diagrama Resolver Ejercicios.

Descripción del Modelo Resolver Ejercicios
<p>La Herramienta permite a los estudiantes resolver una gran cantidad de ejercicios ya sean generales o específicos, que ayudaran a los alumnos ha comprender y mecanizar las diferentes maneras de cómo tomar un camino de solución para las diferentes clases de ejercicios, basados estos en las características del enunciado de cada problema.</p> <p>Los Ejercicios que el estudiante resuelva o trabaje durante la horas de trabajo en la herramienta, serán registrados de manera automática en la B.D, cada ves que finalice su sección de trabajo, los datos que se almacenan serán (ejercicio trabajado, el tiempo gastado por ejercicio, entre otros datos).</p> <p>Esto permite a los profesores, llevar un control sobre el estudiante de que es lo que ellos hacen en las horas de laboratorio y su desempeño con la herramienta, dando un soporte para evaluar al estudiante si así el profesor lo desea.</p>

Tabla 24. Descripción del Modelo de Casos de Uso: Diagrama de Resolver Ejercicios.

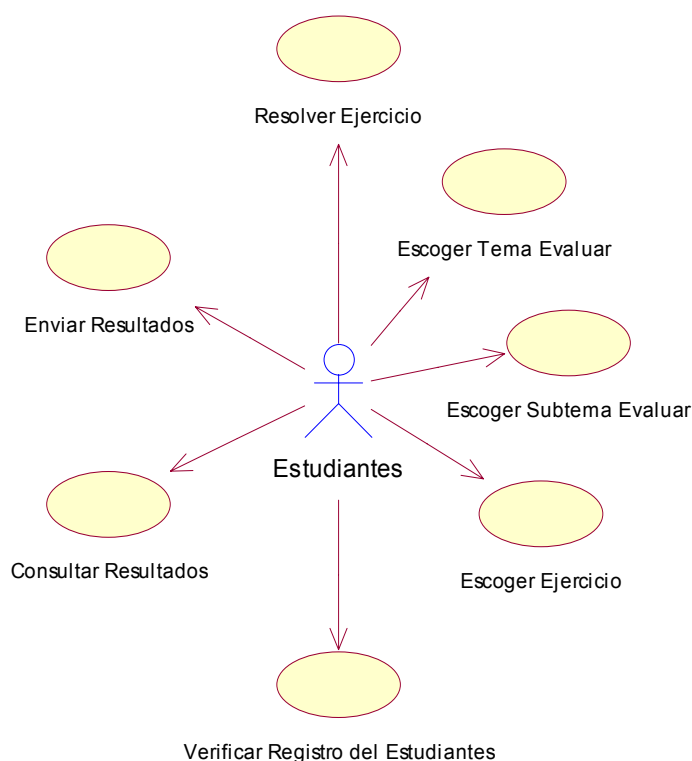


Figura 40. Modelo de Casos de Uso. Diagrama de Evaluar Tema.

Descripción del Modelo Evaluar Tema.
<p>La Herramienta permite a los usuarios realizar una serie de problemas que serán calificados por parte de los profesores entregando una nota que se computara con el resto para entregar una nota final cuando se termine el semestre académico.</p> <p>Las Evaluaciones se realizaran de varias maneras, habrá problemas de selección basados en la teoría que posee la herramienta, ejercicios a resolver. Después de haber terminado con la evaluación el estudiante guardara los resultados para entregarlo al profesor para su calificación. En este mismo modulo los estudiantes consultaran las notas de las evaluaciones que ya han sido calificadas por los profesores.</p>

Tabla 25. Descripción del Modelo de Casos de Uso: Diagrama de Evaluar Tema

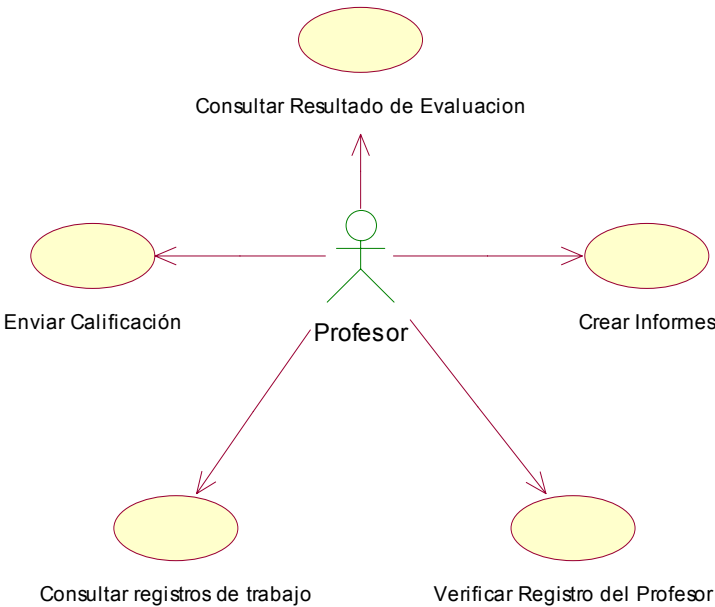


Figura 41. Modelo de Casos de Uso. Diagrama de Consultar Registros.

Descripción del Modelo Consultar Registros
<p>La herramienta permite a los profesores consultar los diferentes registros almacenados en la base de datos de los ejercicios realizados por los estudiantes en sus horas de trabajo, con estos datos ellos pueden crear informes, teniendo un historial de cada uno de los estudiantes, de lo hecho durante el semestre por cada uno de ellos. Estos registros también pueden ser considerados como un factor de calificación, ya que estos muestran, el aprovechamiento que los estudiantes le dieron a la herramienta.</p>

Tabla 26. Descripción del Modelo de Casos de Uso: Diagrama de Consultar Registros

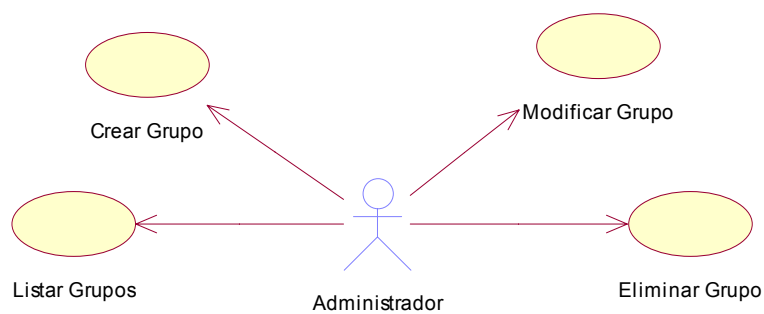


Figura 42. Modelo de Casos de Uso. Diagrama de Administrar Grupo

Descripción del Modelo Administrar Grupo
<p>La herramienta permite al administrador crear, modificar, eliminar y listar los diferentes grupos de trabajo que soportara la herramienta. Los grupos serán creados de acuerdo al listado entregado por los profesores y estos son los mismos grupos de clase que ellos manejan, a estos el administrador les asignara un horario de trabajo los cuales los estudiantes deben cumplir.</p> <p>En el momento en que algún estudiante cancele la asignatura, el profesor debe notificar al administrador para que modifique el grupo en el cual se retiro el estudiante.</p>

Tabla 27. Descripción del Modelo de Casos de Uso: Diagrama de Administrar Grupo.

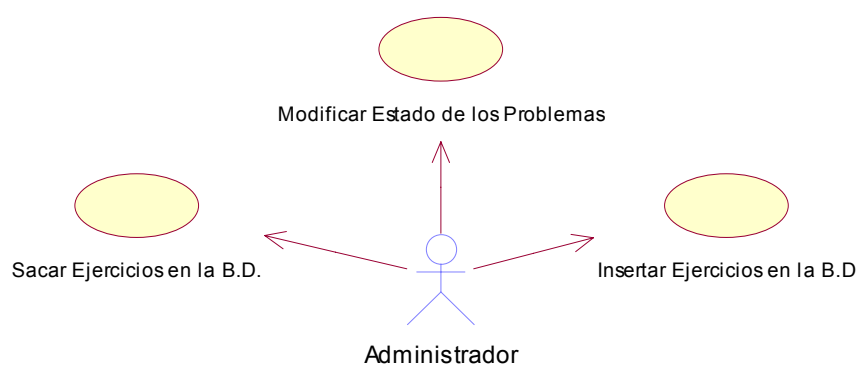


Figura 43. Modelo de Casos de Uso. Diagrama de Administrar Ejercicios.

Descripción del Modelo Administrar Ejercicios

La Herramienta permite al administrador sacar e insertar los ejercicios en la Base de Datos. Los ejercicios que contiene la herramienta están organizados de dos grupos principales, generales y específicos, estos de acuerdo a las diferentes carreras que tienen en su pensum la asignatura geometría descriptiva, aparte de estos habrán problemas de selección que ayudaran a verificar los conocimientos teóricos de los estudiantes. Cuando los problemas son agregados a la Base de Datos se les da un estado de activo o no activo el cual puede ser modificado por el administrador, con el fin de evitar que siempre estén funcionando los mismos ejercicios en la Base de Datos de la herramienta.

Tabla 28. Descripción del Modelo de Casos de Uso: Diagrama de Administrar Ejercicios.

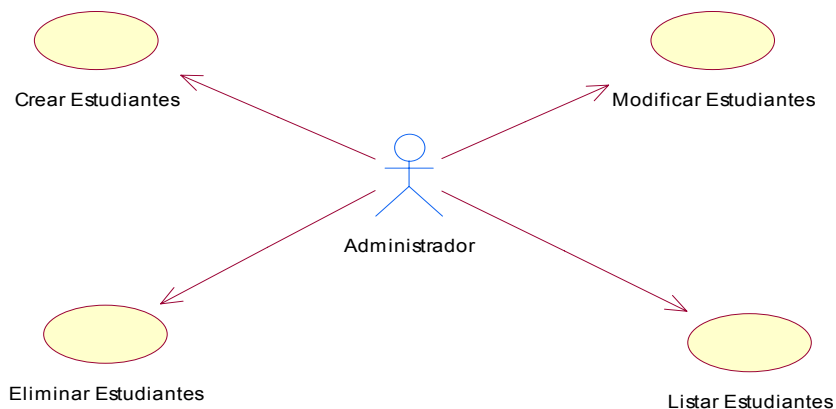


Figura 44. Modelo de Caso de Uso. Diagrama de Administrar Estudiantes

Descripción del Modelo Administrar Estudiantes

La herramienta permite al administrador crear, modificar, eliminar y listar los diferentes estudiantes que se encuentran matriculados para cursar el área de geometría descriptiva. Los estudiantes serán creados de acuerdo al listado entregado por los profesores y estos son los mismos que forman las clases magistrales que los ellos manejan, a estos el administrador les asignara un grupo de trabajo que tiene el mismo código entregado por la universidad ejemplo (a1).

En el momento en que algún estudiante cancele la asignatura, el profesor debe notificar al administrador para eliminarlo de la base de datos y del grupo al cual pertenecía.

Tabla 29. Descripción del Modelo de Casos de Uso: Diagrama de Administrar Estudiantes.

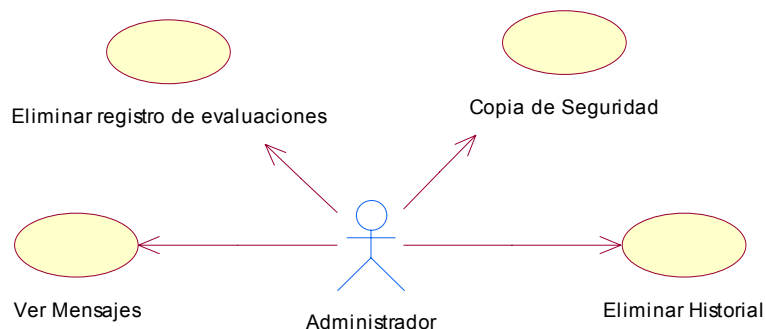


Figura 45. Modelo de Casos de Uso. Diagrama de Administrar otras funciones.

Descripción del Modelo Administrar otras opciones
<p>La Herramienta permite al administrador tener otras opciones como son eliminar registros de la base de datos que la harían pesada, esta eliminación se ejecutarían al final de cada semestre para liberar espacio y permitir el almacenamiento de nueva información, pero esta solo se realizaría si la base de datos se tornara lenta al devolver la información requerida de tal manera el administrador podría presidir de este servicio, pero para mayor seguridad en la perdida de datos se tendrá la opción de realizar copias de seguridad, llegado el caso de un colapso de la base de datos de la herramienta, se tendrán copias de seguridad que serán ejecutadas en un intervalo de tiempo prudente. Además de esto puede ver los mensajes que los usuarios envían relacionados a las fallas encontradas en el sistema o recomendaciones que estos tienen para mejorar la herramienta.</p>

Tabla 30. Descripción del Modelo de Casos de Uso: Diagrama de Administrar otras opciones.

En el transcurso de esta fase se han encontrado nuevos casos de usos, que han sido ubicados en los diagramas anteriores, mejorando la descripción de los casos de usos. Estos nuevos casos han sido descritos como parte del modelo del paquete en el cual se encuentran detallados en el ANEXO C Modelo de Caso de Uso de la Herramienta de este libro.

Es importante recordar que la claridad de la descripción de cada caso de uso depende también de la sencillez o complejidad del mismo, en algunas ocasiones solo bastará con una breve descripción del caso de uso, ya que este quedara explicado completamente con su nombre, y no tiene caminos alternativos, ni restricciones que deban ser destacadas, de lo contrario será necesaria una descripción más detallada.

8.2 ANÁLISIS

Durante la fase de inicio se realizó un borrador del modelo de análisis, ahora en la fase de elaboración se depurará este modelo, desechando componentes que eran considerados primordiales, pero que con un nuevo análisis se encontró que no lo son. En esta fase se trabajara con los casos de uso que son significativos desde el punto de vista de la arquitectura y con aquellos casos de uso complejos que se necesitan clarificar para comprender mejor los detalles de la herramienta.

Por lo tanto en esta fase se continúa con el análisis de la arquitectura, de los casos de uso, de las clases de análisis y de los paquetes del sistema, ocupándonos de los casos de uso más significativos desde el punto de vista de la arquitectura.

8.2.1 ANÁLISIS DE LA ARQUITECTURA

En la fase de inicio se desarrollo el análisis de la arquitectura solo hasta determinar que había una arquitectura factible. Ahora, se extenderá este análisis de la arquitectura hasta punto de que pueda servir como plataforma a una línea base de la arquitectura ejecutable.

Para lograr este objetivo se realizara una partición de la herramienta en los paquetes de análisis, trabajando sobre la vista de la arquitectura del modelo de casos de uso, identificando los paquetes específicos de la aplicación o los paquetes más importantes, proporcionando un medio para organizar los requisitos funcionales.

Por tanto en esta nueva fase, se añadirán los nuevos casos de uso a cada una de los paquetes existentes y se rediseñarán o se crearán nuevos de acuerdo con los criterios que se plantearon en la fase de inicio y que son los que se muestran a continuación.

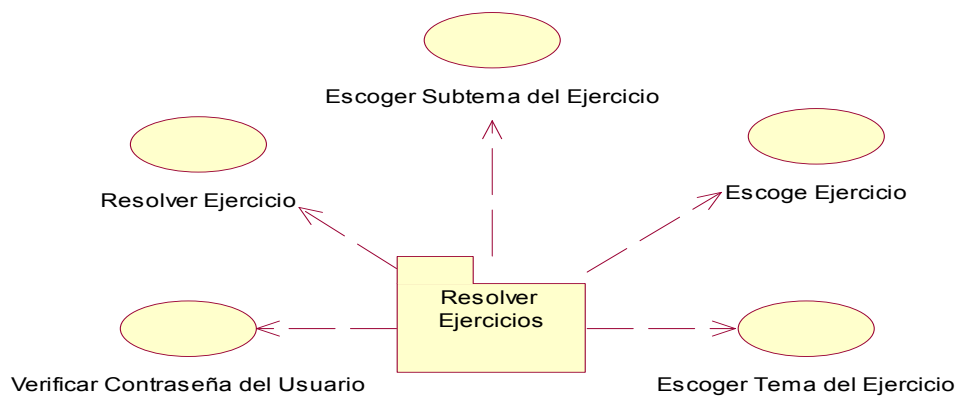


Figura 46. Paquete de análisis Gestión Resolver Ejercicio.

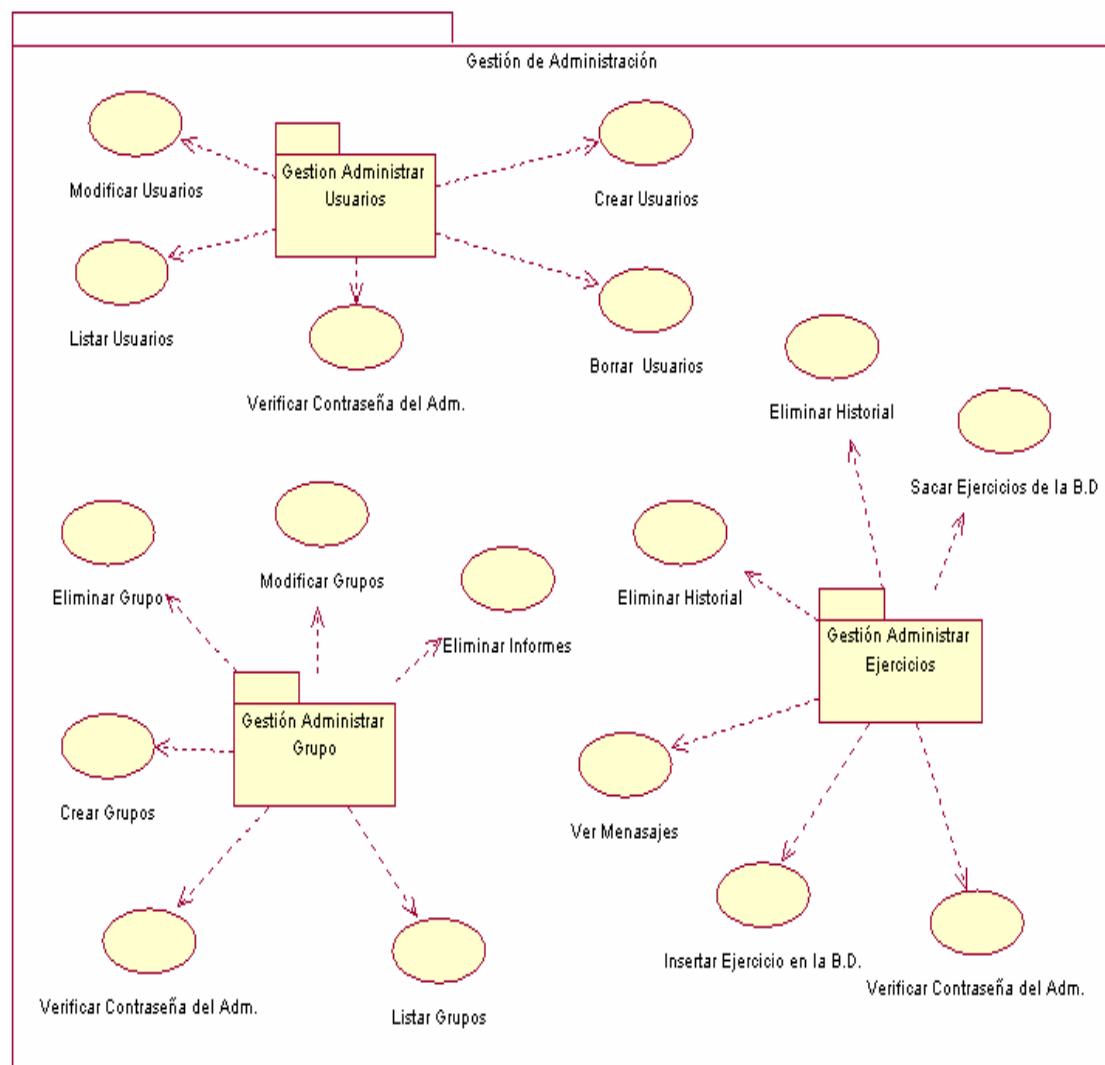


Figura 47. Paquetes de análisis de Gestión de Adm., Gestión de Adm. de Grupo, Gestión de Adm. de Ejercicios, Adm. de Usuarios. Estos últimos contenidos en el primero.

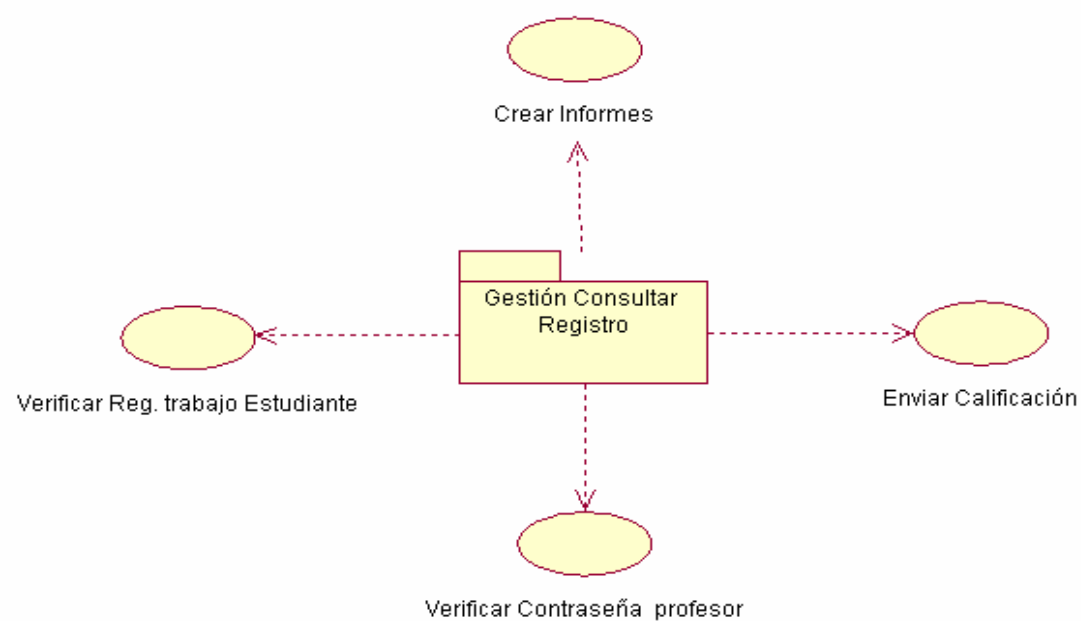


Figura 48. Paquete de análisis Gestión de Consultar Registro.

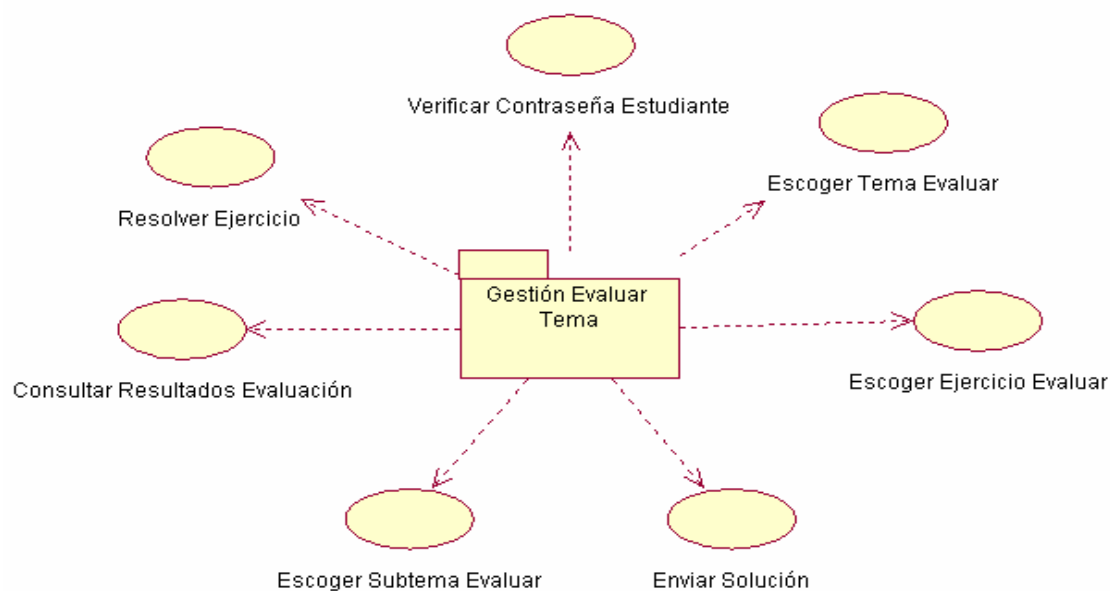


Figura 49. Paquete de análisis Gestión Evaluar Tema.

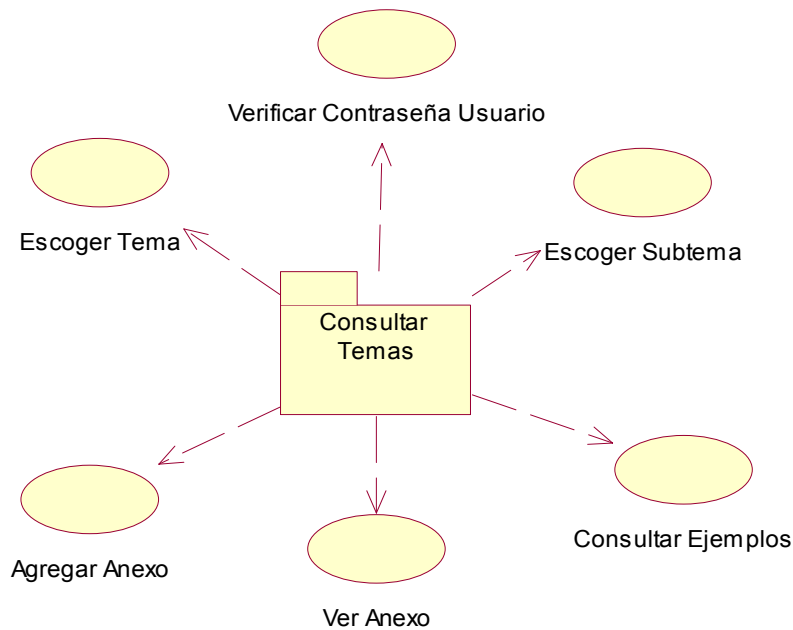


Figura 50. Paquete de análisis Gestión Consultar Temas

8.2.2 ANÁLISIS DE CASOS DE USO

Numerosos casos de uso no son claramente comprensibles tal y como están descritos en el modelo de casos de uso, por tanto deben ser refinados en función de las clases del análisis que existen en el ámbito de los requisitos pero que no se implementan necesariamente de forma directa.

Los casos de uso que deben ser refinados son aquellos que sean importantes, complejos, o que influyen sobre otros casos de uso, mejorados desde el punto de vista de la arquitectura o para la comprensión de los requisitos. Los otros casos de uso, no se refinan ni analizan, ya que solo se necesita una comprensión de lo que son y de que no tengan ningún impacto para poder ser implementados.

No es necesario describir con detalle los casos de uso mas significativos, solo hasta el punto de que sean comprensibles para cualquier persona que desee consultar el documento del proyecto, luego de haber descrito los casos de uso, se identificaran las clases del análisis, se asignara responsabilidades y como se relacionan entre si para llevar a cabo el flujo de sucesos del caso de uso.

Para este fin se han utilizado diagramas de colaboración, ya que el objetivo que se pretende es identificar requisitos y responsabilidades sobre las clases, mostrando sus relaciones y atributos.

Como ya se tienen algunos paquetes de análisis, se utilizarán durante el transcurso de esta fase, para cada uno de estos paquetes se tomaron los casos de uso que eran de importancia desde el punto de vista de la arquitectura y que aportarían para crear la línea base de la misma.

8.2.2.1 PAQUETE DE GESTION CONSULTAR TEMAS

En este paquete se ha seleccionado el caso de uso Escoger Subtemas, para este caso se puede observar el diagrama de la Figura 50, que contiene tres clases de entidad: Subtemas, Ejemplos y Anexos. La primera de estas clases, subtemas, contiene la información de cada uno de los subtemas que contiene el plan de estudio de la asignatura geometría descriptiva. La clase Ejemplos muestra un ejemplo al finalizar la consulta de la información, este ejemplo está basado respecto a la información escogida. La tercera clase de entidad, la clase Anexos es donde los profesores usuarios de la herramienta podrán ubicar los diferentes Anexos sobre los temas tratados por la herramienta.

La Clase de control gestor de subtemas, se encarga de seleccionar los subtemas, correspondientes al tema escogido, y al finalizar de consultar la información se escogerá el ejemplo respectivo. La clase de control gestor de Anexos se encarga de insertar los diferentes documentos que los profesores tengan sobre los temas, para completar la información que ya se tiene, también les permitirá a los todos usuarios ver los anexos que se encuentran almacenados en la B.D.

Por último, la clase de interfaz denominada Interfaz de Usuario (IU) del módulo de teoría, se encarga de permitir al usuario seleccionar los subtemas que desea, así como la (IU) de Agregar Anexo que se encarga de mostrar el formulario para insertar los anexos que los profesores desean a la información que ya se tiene, por último (IU) ver Anexos que permite a los usuarios de la herramienta ver los Anexos que se tienen.

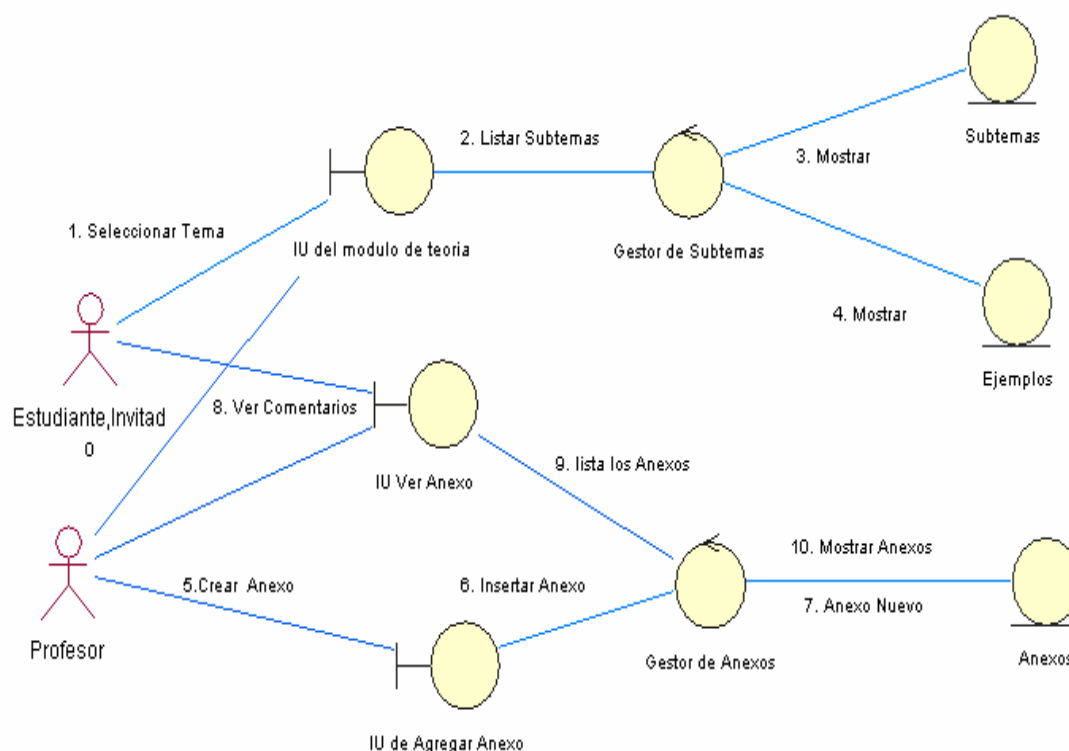


Figura 51. Diagrama de Colaboración de las Clases de Análisis para el Caso de Uso Consultar Subtemas.

8.2.2.2 PAQUETE DE GESTION RESOLVER EJERCICIOS

Para el paquete de Gestión de Resolver Ejercicios se ha seleccionado para su análisis el caso de uso Escoger Ejercicio. Como se puede observar en la Figura 52, este caso de uso, contiene dos clases de entidades: Problemas y teóricos. La primera clase contiene los problemas que los estudiantes resolverán como lo harían en el aula de clase, la segunda clase de entidad tiene los problemas teóricos que se les entregaran a los estudiantes para verificar si han comprendido la información teórica de la herramienta.

Se tiene el gestor de ejercicio que se encarga de seleccionar los ejercicios problemas y teóricos, esto de acuerdo al tema y subtema escogido para trabajar. La (IU) escoger Ejercicio se encarga de permitir al usuario escoger los ejercicios que los estudiantes deseen.

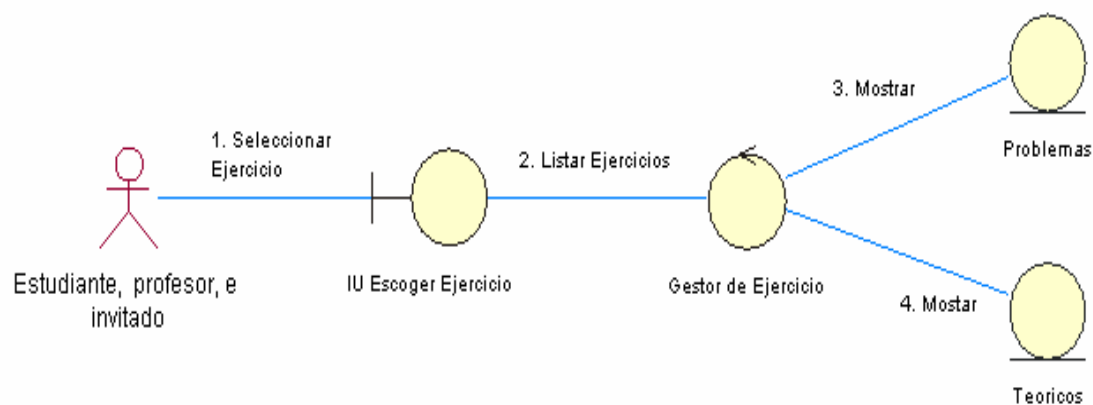


Figura 52. Diagrama de Colaboración de las Clases de Análisis para el Caso de Uso Escoger Ejercicio.

8.2.2.3 PAQUETE DE GESTION EVALUAR TEMA

El paquete de Gestión evaluar tema es uno de los más importantes en la arquitectura de la herramienta, sin embargo se decidió realizar el análisis del caso de uso consultar notas evaluaciones, ya que es importante mostrar los resultados de las evaluaciones realizadas por los profesores en las horas de trabajo con la herramienta, este paquete es de los más complicados de realizar y que contribuyen en el diseño de la arquitectura, por su complejidad solo se realizara en una parte, dejando una puerta abierta para la ejecución de esta en otros proyectos.

En este paquete se ha seleccionado el caso de uso Consultar notas evaluación. Para este caso, se puede observar en el diagrama de la Figura 53 que contiene cuatro clases de entidad: Ejercicio, Fecha, Calificación, Tiempo. Las clases de entidad Ejercicio, contienen los ejercicios de los cuales trato la evaluación realizada por los estudiantes.

La clase de entidad fecha, Hora contiene la fecha y hora en la cual se realizo la evaluación, la entidad tiempo contiene el tiempo gastado por el estudiante para la solución del los problemas y por ultimo la entidad calificación se encuentra la nota que el estudiante obtuvo en la evaluación.

La clase de control Gestor de consulta, permite observar todas las entidades mencionadas y por ultimo la (IU) de consultar notas evaluación, le permite al estudiante verificar los resultados obtenidos en las diferentes evaluaciones.

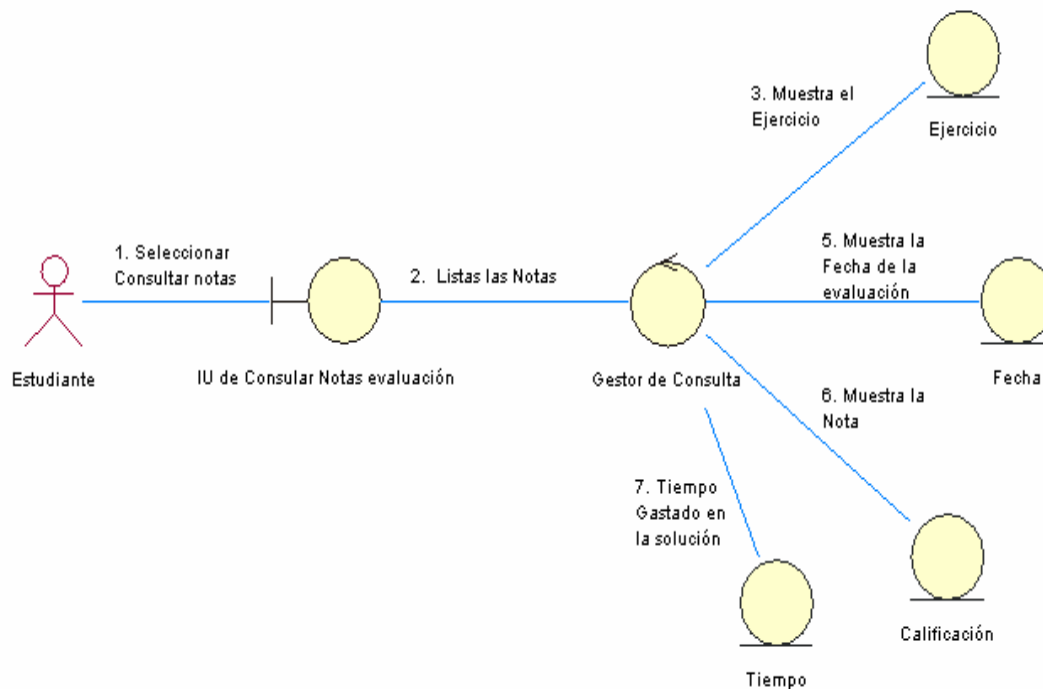


Figura 53. Diagrama de Colaboración de las Clases de Análisis para el Caso de Uso Consultar notas evaluación

8.2.2.4 PAQUETE DE GESTION CONSULTAR REGISTROS

Los casos de uso pertenecientes a este paquete, no representan un considerable grado de complejidad para el desarrollo de la herramienta. Sin embargo, por la misma razón que se ha detallado el caso de uso Escoger Ejercicio del paquete Gestión resolver ejercicio, que es el de obtener las principales clases que forman parte de este paquete para contribuir especialmente en el diseño del modelo entidad-relación.

Como se puede observar en la Figura 54, este caso de uso crear informe, cuenta con tres clases de entidad: La entidad curso que corresponde al informe que el profesor desea de los diferentes cursos que el tiene a su cargo. La segunda clase de entidad estudiante corresponde al informe de los diferentes estudiantes, de lo que ellos

hicieron en las horas de trabajo y por ultimo la clase entidad evaluación son los informes de las diferentes evaluaciones que el profesor realizo durante el semestre.

La clase de control gestor de informe, se encarga de seleccionar los diferentes tipos de informe que el profesor desee crear, y por último (IU) crear informe que les permite a los profesores la creación de informes cuando los necesite.

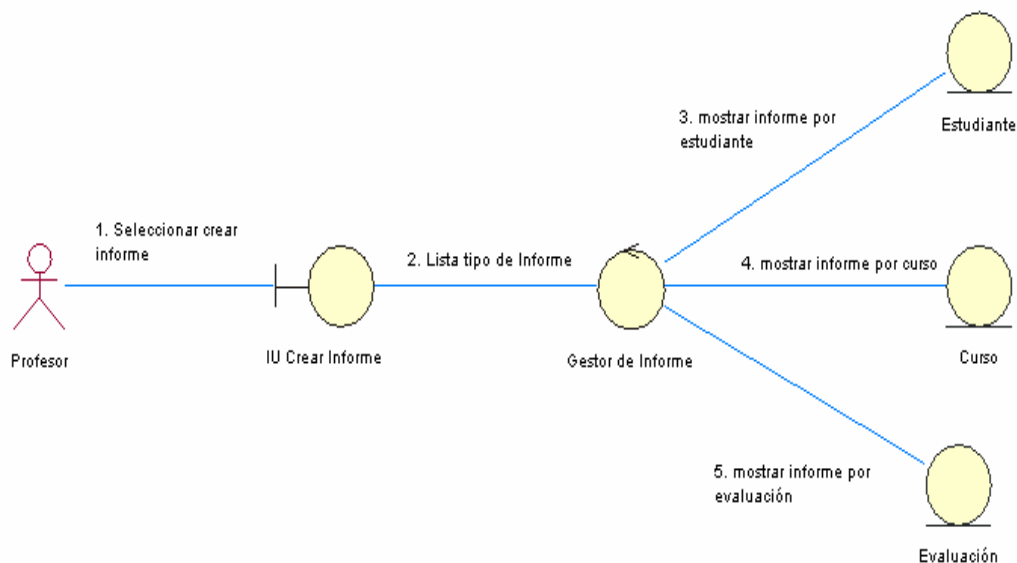


Figura 54. Diagrama de Colaboración de las Clases de Análisis para el Caso de Uso Crear Informes.

8.2.2.5 PAQUETE DE GESTION DE ADMINISTRACIÓN

El paquete de Gestión de administración hace parte de los más importantes en la arquitectura de la herramienta, ya que esta gestión esta encarga de mantener el buen funcionamiento y mantener actualizada la herramienta. Como se puede observar en la Figura 55, este caso de uso, cuenta con cinco clases de entidad: grupos, profesores, estudiantes, horarios, y fecha.

La clase de entidad grupo corresponde a los grupos de trabajo que se crearan por parte del administrador, la entidad profesor corresponde a la persona encargada de dirigir el grupo creado, la entidad estudiantes son los alumnos que componen los grupos de trabajo, la entidad día son los días asignados a los grupos para trabajar en

el software y por ultimo la entidad horarios que es la hora asignada por el administrador para trabajar con la herramienta.

La clase de control de grupos, se encarga de la creación de los diferentes grupos de trabajo de la herramienta, con sus respectivas entidades. Por ultimo la interfaz de administrar grupo le permite al administrador de la herramienta crear y administrar los diferentes grupos de trabajo.

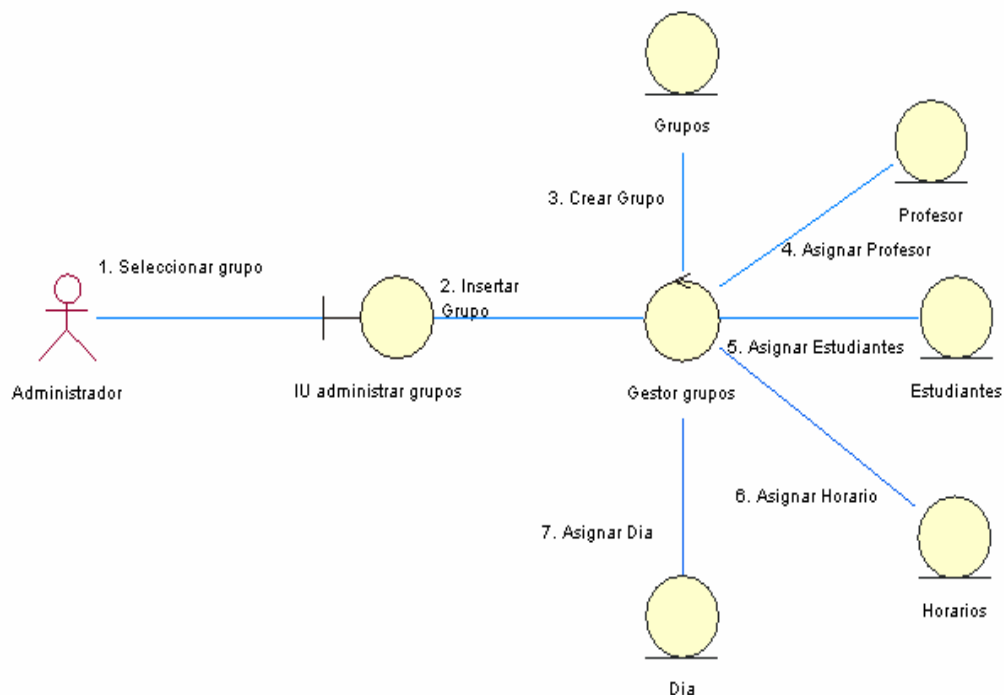


Figura 55. Diagrama de Colaboración de las Clases de Análisis para el Caso de Uso Administrar Grupos.

8.3 DISEÑO

En esta fase se diseñara e implementara menos del 10 por ciento de los casos de uso, esto es solo una fracción del total de casos de uso identificados en esta fase. En el diseño se modela la herramienta y se encuentra su forma a modo de soporte, todos los requisitos funcionales, no funcionales y otras restricciones. Una parte esencial en el diseño es el resultado del análisis, el cual proporciona una comprensión mas

detallada de los requisitos, y asigna una estructura la cual se debe de conservar cuando se da forma a la herramienta.

El diseño tiene como propósito comprender en profundidad aspectos relacionados con los requisitos no funcionales y restricciones relacionadas con los lenguajes de programación, los componentes, sistemas operativos entre otros, además es un punto de partida para descomponer el trabajo de implementación en actividades manejables, contribuyendo a crear una arquitectura estable y sólida.

En la fase de elaboración solo se diseñarán una pequeña fracción del total de casos de uso identificados, en particular, solo aquellos arquitectónicamente significativos. Al igual que los paquetes de análisis, los subsistemas durante el diseño, son críticos para definir las vistas de la arquitectura.

Como se verá en las siguientes secciones, a partir de las clases de análisis se puede obtener el diseño de las interfaces y de la base de datos. El diseño de la estructura de información (modelo entidad-relación) se obtiene a partir de las clases de entidad y el diseño de la interfaz de usuario para cada caso de uso, a partir de las clases de interfaz.

8.3.1 DISEÑO DE LA ARQUITECTURA

El objetivo del diseño de la arquitectura es esbozar los modelos de diseño, despliegue y su arquitectura mediante la identificación de los nodos, configuraciones de red, subsistemas, interfaces, software de la herramienta y capa intermedia.

La siguiente grafica ilustra cada una de las capas de la arquitectura del software:

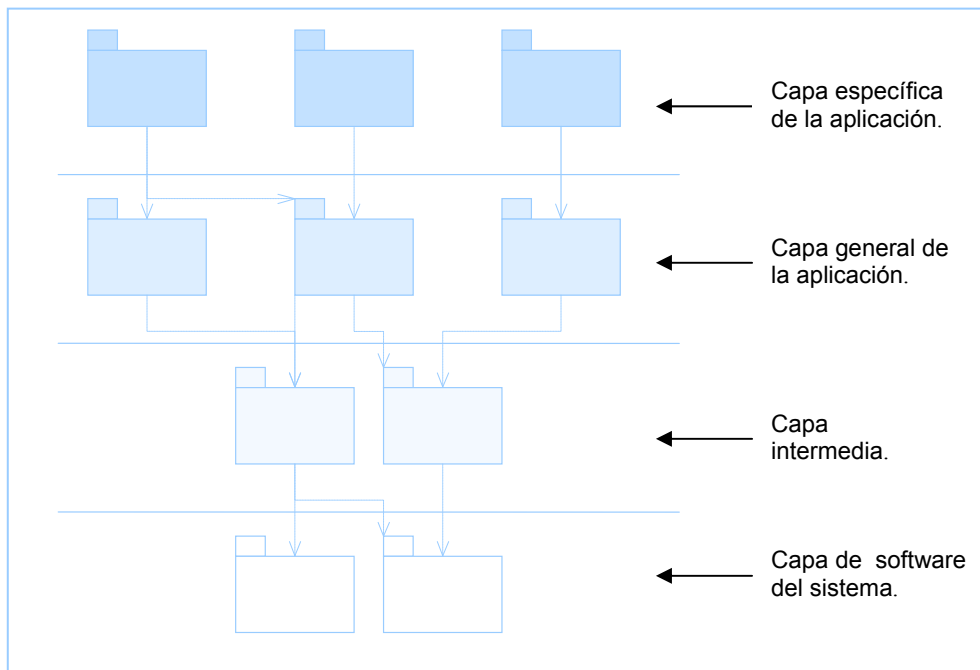


Figura 56. Arquitectura en capas.

8.3.1.1 NODOS Y CONFIGURACIONES DE RED

El diagrama de despliegue para el software se presentó por primera vez en la anterior fase, en él se puede observar que se cuenta con dos clases de nodos: Un nodo servidor que tiene la capa de datos además de la de lógica de aplicaciones y los nodo cliente que se encargará de la capa de presentación en un modelo de arquitectura de tres capas, como el que se utilizará para la distribución del software. Dado que el software será desarrollado con tecnología Web se utilizaran los protocolos http y TCP/IP para la comunicación entre los nodos.

Los requerimientos de hardware de cada uno de estos nodos se describen a continuación:

➤ **Nodo Servidor**

Para la puesta en marcha del Software se requiere un equipo servidor cuyas características serán definidas en el plan preliminar de la fase de Transición que será elaborado al terminar la fase de construcción, sin embargo como servidor para la fase de desarrollo se utilizará un PC con las siguientes características:

- ◆ Procesador Pentium III de 500MHz o superior
- ◆ Memoria RAM de 256MB
- ◆ Disco Duro de 10G
- ◆ Tarjeta de Red

Para el uso de la herramienta dentro de la universidad se aprovechará la infraestructura de la escuela de diseño industrial, la cual actualmente cuenta con un centro de cableado principal, disponiendo de un total de 25 puntos de acceso aproximadamente.

➤ **Nodo Cliente**

El nodo cliente debe contar con acceso a la red de área local (LAN) del edificio de diseño industrial, los requerimientos para los equipos clientes son:

- ◆ Procesador Pentium II de 333MHz o superior
- ◆ Memoria RAM de 64MB
- ◆ Tarjeta de Red

8.3.1.2 SUBSISTEMAS

Los subsistemas constituyen un medio para organizar el modelo de diseño en pieza manejables, además ayudan a dividir el trabajo de diseño y posteriormente el de implementación.

Primero se identifican los subsistemas a partir de los paquetes de análisis. En el flujo de trabajo de análisis se obtuvieron 5 paquetes, realizando una descomposición adecuada del software en paquetes, utilizándolos como base para los subsistemas de diseño. Por lo tanto se cuenta con los subsistemas de teoría, ejercicios, evaluación, consulta de registro, y de administración.

Existen dos nuevos subsistema que se puede observar en la Figura 57, El primero: Gestión de Sesiones se encarga de controlar el acceso al software por parte de los usuarios. Aunque aun no ha sido tratado este aspecto, se dará una breve explicación: Todos los actores para poder acceder a cada uno de estos subsistemas deben estar

en sesión, es decir deben haber validado su ingreso a la herramienta, y deben tener permiso para acceder al subsistema y a un caso de uso en particular, por tanto, como todos los subsistemas comparten esta funcionalidad, se agrupo esta característica en este nuevo subsistema.

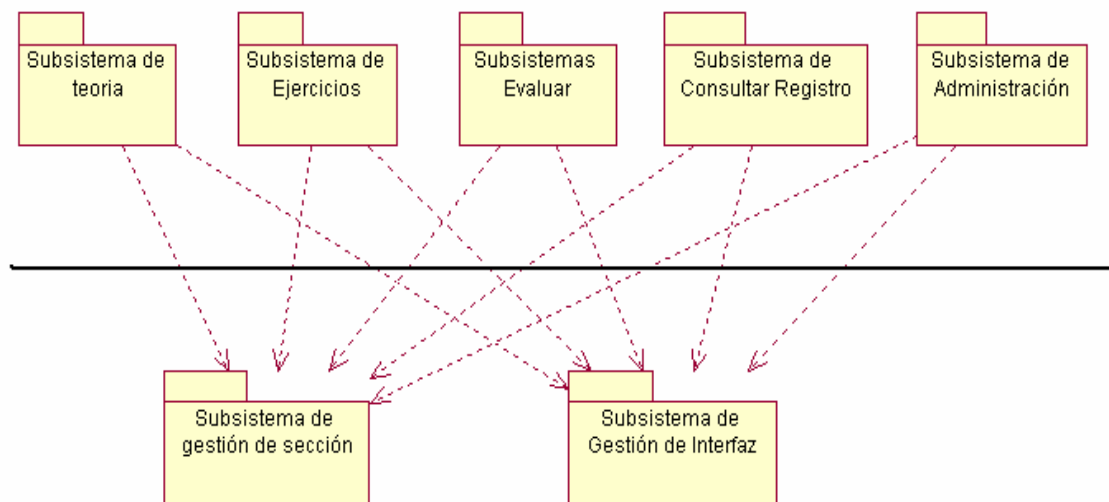


Figura 57. Subsistemas de Diseño.

Otro subsistema a destacar, es el Subsistema de Gestión de Interfaz, este subsistema, se explicará mas adelante junto con el modelo de arquitectura de 3 capas, por ahora, solo se resaltará que este subsistema permitirá dividir la capa lógica de la capa de presentación, razón por la cual es utilizado por todos los subsistemas que necesiten comunicarse con algún actor.

Dado que estos dos últimos subsistemas no son accedidos directamente por los usuarios, sino que son utilizados por los demás subsistemas, se denominan subsistemas generales de aplicación y por esta razón han sido ubicados bajo la línea horizontal resaltando esta diferencia.

8.3.2 DISEÑO DEL SUBSISTEMA DE GESTIÓN DE SESIONES

El protocolo http no tiene ningún mecanismo para mantener el estado; por eso http es un protocolo libre de contexto o sin estado, las peticiones individuales no están relacionadas unas con otras. Un servidor Web no puede distinguir fácilmente entre los usuarios y no sabe nada de las sesiones de los usuarios, por tanto, es necesario encontrar un mecanismo para solucionar este inconveniente.

Como se mencionó el subsistema de Gestión de sesiones se encarga de Validar el ingreso al software por parte de un actor, el cual deben tener permiso para acceder a una zona o subsistema determinado, así como a un caso de uso en particular.

Para explicar con más detalle el funcionamiento de este subsistema se hará uso del siguiente diagrama de secuencia. Sin embargo este diagrama no contiene todos los detalles, restricciones y caminos alternativos, ya que su objeto es mostrar al lector de una manera sencilla la forma como se realizará la validación de ingreso a la herramienta y como otros subsistemas accederán al sistema de Gestión de Sesión para a validar el acceso a los casos de uso que solicita el actor.

Como se observa en el diagrama de secuencia de la Figura 58, cuando un usuario quiere ingresar al software debe primero solicitar su ingreso al mismo, el subsistema de gestión de sesiones se encargará de solicitar al usuario la información necesaria (Clave y Contraseña) que será verificada en la base de datos. Si el ingreso es correcto el gestor de sesiones le enviará un identificador de sesión²⁴ al usuario, que consta de un número aleatorio que lo identifica y de ciertos datos propios del usuario.

Cuando el usuario, en este caso un estudiante, desea hacer uso de un proceso para llevar a cabo un caso de uso de cualquier subsistema, se debe verificar que dicho usuario tenga permiso para su ejecución y para acceder o modificar a la información que este maneja. Esto es de especial importancia, dado que para la comunicación entre el cliente y servidor se utiliza tecnología Internet, exponiendo los problemas de seguridad propios de esta tecnología.

En el siguiente diagrama de secuencia, se observan este tipo de validaciones cuando un estudiante desea llevar a cabo el caso de uso de enviar calificación. Para este caso de uso, se necesita realizar dos validaciones: la primera antes de enviar las calificaciones que verifica si el usuario es un profesor.

La segunda cuando el profesor envía los datos de las notas, reforzando la seguridad de la herramienta, ya que evita el problema de que un usuario malicioso, sin validarse correctamente en el software, trate de modificar las notas. Esto se podría realizar

²⁴ El termino Sesión para una instancia de un usuario que visita un sitio donde se mira una o mas paginas. Secuencia de paginas que un usuario visita en un sito Web desde que entra hasta que lo abandona.

dato que comúnmente los navegadores Web pueden mantener localmente información sobre una página luego de haberla visitado. Ese usuario, podría aprovechar estas paginas temporales y a través del formulario de envío utilizado por algún usuario, correctamente validado, enviar una nueva información para la adulteración de las notas.

En la etapa de implementación, estas sesiones serán realizadas con ayuda del modulo de PHP para el manejo de las mismas. Esto se explicará con más detalle en la fase de construcción cuando se implemente este subsistema.

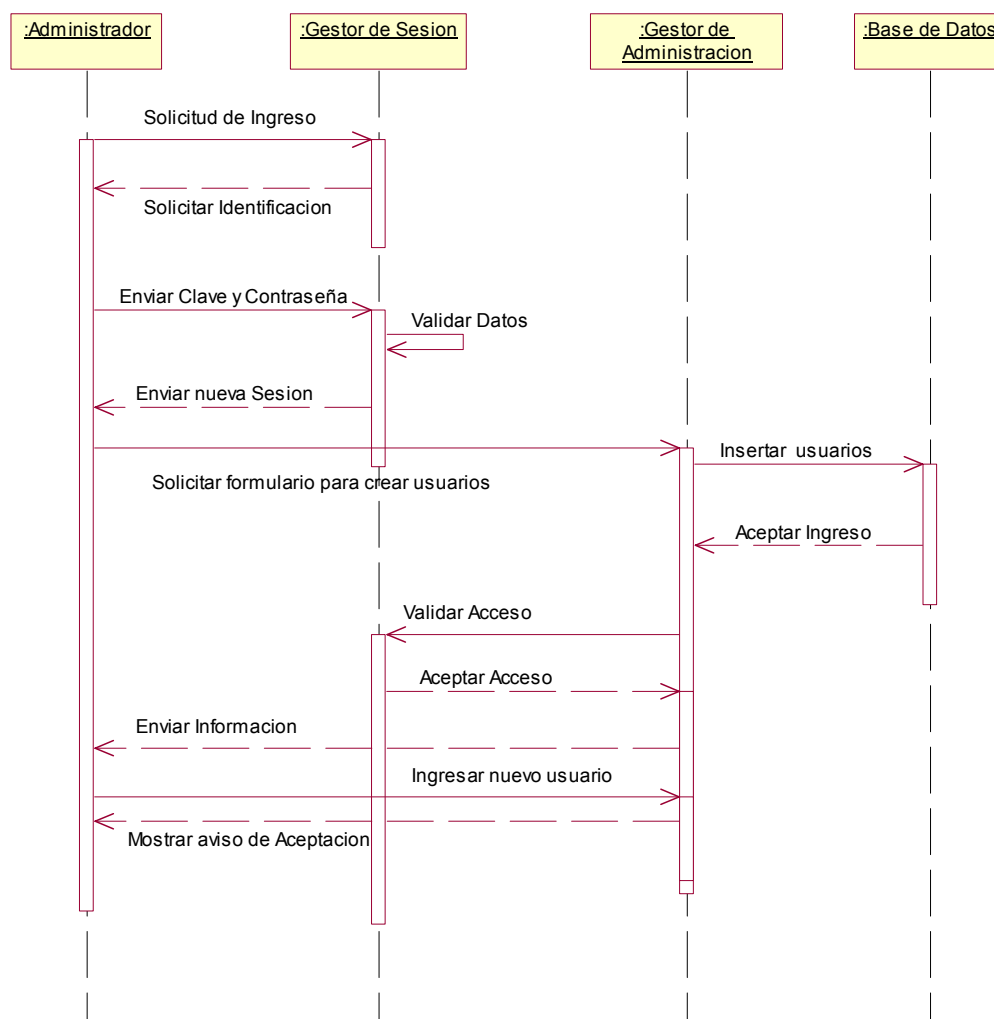


Figura 58. Diagrama de Secuencia de las interacciones entre subsistemas.

8.3.3 DISEÑO GUIADO POR EL MODELO DE ARQUITECTURA DE TRES CAPAS

La herramienta GEDISO 1.0 se desarrolla a partir del concepto de arquitectura en tres capas, que consiste en dividir la funcionalidad del software total en capas lógicas que pueden ser agrupadas para interactuar entre ellas a alto nivel, así en la capa de datos agrupadas todas las funciones de base de datos, las funciones de interfaz de usuario en la capa de Presentación y finalmente toda la lógica de operación en la capa Lógica.

Este hecho permite mayor flexibilidad y reutilización de partes de código, facilitando hacer modificaciones sencillas y rápidas a cualquiera de estas capas sin afectar el funcionamiento de las demás, esto permite el mantenimiento y evolución del sistema, ya que este puede ser actualizado, sin tener que reconstruirlo en su totalidad.

La capa de datos esta determinada por la base de datos y las funciones para el manejo de la misma, entre estas funciones están las que manejan la integridad referencial de las tablas, funciones almacenadas dentro de la base de datos, así como las interfaces de inserción, modificación y eliminación de registros a la misma que son desarrolladas mediante scripts en PHP. La capa lógica o de negocio contendrá todos los scripts y funciones que llevan a cabo la lógica del software teniendo en cuenta los casos de uso del sistema.

8.3.4 DISEÑO DE LA INTERFAZ

Esta es una etapa importante en el desarrollo de aplicaciones, la interfaz de usuario es la que va a determinar que tan utilizable va a ser un software, si la interfaz de usuario es pobre o no tiene estándares claros, el usuario no se sentirá cómodo al usar la aplicación, o peor aún, dejará de usarla.

Para el desarrollo de la interfaz se siguieron los siguientes lineamientos:

- ◆ **Consistencia:** Todas las pantallas de la aplicación Web deben de tener una distribución consistente de imágenes, texto y controles gráficos. Ante acciones de usuario comunes el software debe mostrar mensajes con la misma estructura sin importar en que parte de la herramienta ocurra el evento.

- ◆ **Flujo de pantallas:** El paso de una pantalla a otra debe de ser coherente con el trabajo que intente realizar el usuario.
- ◆ **No sobre poblar las pantallas:** Al presentar gran cantidad de controles gráficos en una misma pantalla dificultan la comprensión de la misma. Si se requiere obtener información por parte del usuario, es recomendable dividir en diferentes pantallas sucesivas la captura de información.
- ◆ **Agrupar elementos relacionados:** Una buena práctica en la creación de interfaces de usuario es la agrupación de elementos que estén relacionados entre sí, delimitando por un rectángulo o algún otro tipo de imagen los datos comunes.

Existen lineamientos que son específicos al desarrollo de aplicaciones Web, una buena referencia es “Los siete pecados mortales de un sitio Web y por qué deben de evitarse”²⁵ de Jesse Berst, que describe los grandes errores que se pueden cometer en el diseño de un sitio Web y estos son:

- ◆ **Navegación inconsistente:** A veces se hace clic en una barra de menú lateral, otras es un menú desplegable. La ubicación de los títulos, gráficos, vínculos, etc. debe de ser consistente en todas las páginas del sitio.
- ◆ **Vínculos rotos.** Vínculos que no llevan a ningún lado, todos los vínculos deben estar perfectamente direccionados.
- ◆ **Sitios que requieren de un navegador específico:** Es molesto toparse con sitios que están diseñados para un navegador específico, así como los que requieren que se baje un plug-in²⁶ determinado sin ofrecer una versión del documento que no requiera de dicho plug-in.
- ◆ **No poner información de contacto:** dirección de la entidad, números telefónicos, etc.

²⁵ Jesse Berst; -Seven Deadly Website Sins-. http://www.zdnet.com/anchordesk/story/story_1716.html. 17 Abril 2001

²⁶ Componente de software que agrega funcionalidad a un navegador sin necesidad de recom-pilarlo. Un ejemplo típico de plug-in es el de Macromedia Flash, el cual es usado para visualizar animaciones vectoriales en páginas Web. Éste plug-in puede ser instalado para el Internet Explorer y Netscape Navigator.

- ◆ **Sitios que abren otras ventanas de navegación:** Es molesto para el navegador ingresar a un sitio y que éste abra nuevas ventanas de navegación con vínculos a otros sitios.
- ◆ **Símbolos de “En construcción”:** La mayoría de los sitios en el Web están en constante evolución, Es poco profesional indicar que el sitio está en construcción.
- ◆ **Uso de marcos:** También conocidos como frames, traen consigo problemas con el uso de los botones Anterior y Siguiente de los navegadores, muchas veces son creados de un tamaño fijo y la información contenida en ellos no es visible completamente, etc. En nuestro caso se utilizaran algunos en la presentación de las paginas teóricas, se trabajo de tal forma que se evitan los problemas que estos frames contiene para las paginas Web.

En la herramienta GEDISO 1.0 existen varios actores, los cuales serán reflejados como perfiles de usuario, cada uno de los usuarios del software pertenecerá a solo uno de estos perfiles, y cada perfil tiene acceso a ciertos casos de uso de un subsistema.

En la Tabla 31 se muestran los subsistemas ya obtenidos y las zonas de la herramienta a las que dieron origen.

Subsistema	Zona	Descripción
De Teoría	Informativa	En esta zona se encuentran opciones generales, tales como información de la asignatura, información del software, ver ejemplos resueltos, agregar anexos.
De Ejercicio	Trabajo	Permite a los usuarios resolver los diferentes problemas que la herramienta le suministre
De Evaluar	Evaluación	Permite a los estudiantes realizar las evaluaciones que los profesores les indique, escogiendo entre resolver problema y ejercicios teóricos.
De consultar registro	Consulta de datos	Maneja la opción de consultar toda la información con respecto a lo hecho por los estudiantes en la horas de trabajo y evaluaciones realizadas
De Administración	Administrativa	Maneja las opciones del subsistema de administración.

Tabla 31. Zonas de la Herramienta GEDISO 1.0.

Para continuar con la descripción de la interfaz, en la Figura 59 se observa el Diseño General de la interfaz de la Herramienta GEDISO 1.0 para el modulo de teoría, que esta dividida en cuatro secciones:

En la sección superior (1) se encuentra el logotipo del software y un encabezado para las páginas. En la sección izquierda (2) estarán las opciones para escoger el tema del contenido que el usuario desee consultar. Además los subtemas aparecen al ser seleccionado el tema específico al cual pertenecen y al final del contenido de los subtemas se encontrara un link que permitirá observar el ejemplo respectivo a cada tema.

En la sección derecha (3) de la interfaz estará el contenido principal de la aplicación, es decir el desarrollo de cada una de las opciones de la sección (2). Por ultimo en la sección inferior (4) se encuentra un navegador que permitirá desplazarse ágilmente por la paginas que se encontrara en el modulo de teoría, esta sección solo será disponible para este modulo.

Para facilitar el desarrollo de las páginas, se utilizarán hojas de estilo en cascadas, que permitirá definir las propiedades de presentación de los diferentes elementos de una página Web. Estos estilos serán definidos para cada zona, permitiendo modificar fácilmente el aspecto (colores, tipo y tamaño de letra, entre otros) de una zona en especial.

Para el resto de las interfases, las secciones son las mismas solo que en vez de tener temas a seleccionar en la sección (2) se tendrán las diferentes opciones de trabajo, que mostraran sus contenidos en la sección (3).

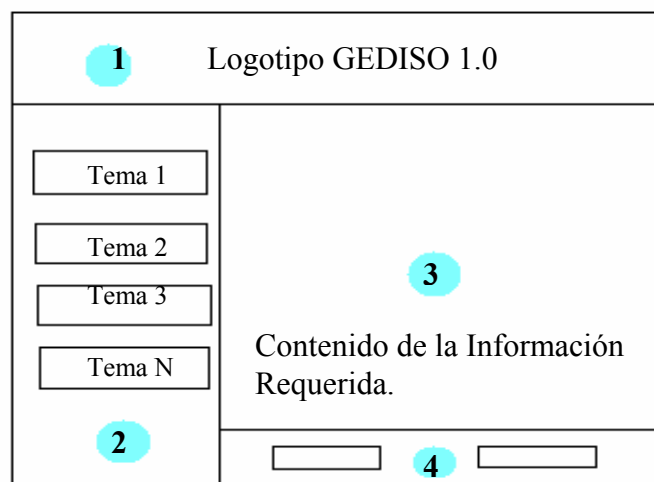


Figura 59. Diseño General de la interfaz de la herramienta GEDISO 1.0, modulo de teoría.

Además para facilitar la identificación de cada uno de las temas por el usuario, estas serán identificadas cada una con un color especial.

Lo anterior corresponde solo al modulo de teoría, para los demás módulos la distribución de la interfaz es similar, solo difiere en que solo tendrá tres secciones, tales como se ve en la figura.

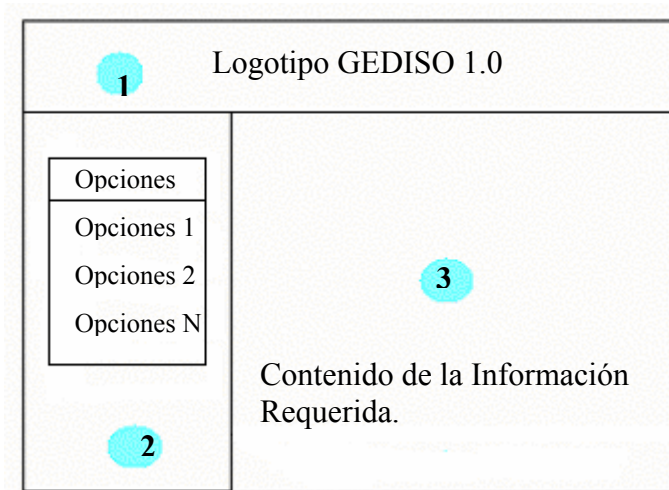


Figura 60. Diseño General de la interfaz de la herramienta GEDISO 1.0.

8.3.5 DISEÑO DE LA BASE DE DATOS

Diseñar clases de entidad que representen información persistente generalmente implica el uso de tecnologías de bases de datos específicas²⁷. Cada una de las clases de análisis generalmente tiene una estrecha correspondencia con cada una de las entidades del diagrama entidad-relación. Por ejemplo la clase de entidad grupo, corresponde con la entidad grupo del diagrama mostrado en la Figura 61 Sin embargo existen ciertas excepciones.

La clase de entidad evaluación ha sido dividida en dos entidades en el diagrama entidad relación: la primera, evaluación, guarda la información general de la misma. La segunda, nota _ evaluación, guarda la información necesaria (como su nota) sobre el desarrollo de la evaluación por parte de un alumno de un grupo en particular.

Otra excepción es la clase de entidad asignación, la cual no formará parte del diagrama Entidad - Relación ya que se refiere a los documentos enviados por los estudiantes, en su lugar se destinará un espacio en el servidor donde estarán alojados estos archivos, y para acceder a ellos solo se necesitará el nombre de archivo deseado, almacenado en la tabla progresos.

En la Tabla 32. Descripción de las tablas en el Diseño de la Base de Datos Se presenta un resumen general de las entidades presentes en el diagrama, además en el D, Diccionario de Datos página 218, contiene la información referente a la base de datos y sus elementos.

Tabla	Información
Usuario	Todos los usuarios de sistema para todos los actores
Materia	Correspondiente a las materias de Geometría Descriptiva I y II
Grupo	Grupos creados semestralmente para cada una de las materias.
Estudiantes	Estudiantes que pertenecen a cada uno de los grupos.
Evaluación	Evaluaciones realizadas por los profesores.
nota _ evaluación	Nota para cada una de las evaluaciones de cada alumno.
Profesores	Profesores de cada unos de los grupos.
Temas	Temas que contiene la herramienta.

²⁷ Jacobson, Ivar. Booch, Grady. Rumbaugh, James. El Proceso Unificado de Desarrollo de Software. Primera edición. Addison Wesley. España, 2000. Pág. 244.

Subtemas	Subtemas correspondientes a los diferentes temas que contiene la herramienta.
Problemas	Es el conjunto de los problemas generales y específicos que contiene la herramienta para ser desarrollados por los diferentes usuarios.
Comentarios	Son los anexos que los profesores ingresan de acuerdo al tema o subtema de los cuales desean hablar
Historial	Es el registro de las actividades hechas por los estudiantes en el modulo de problemas.
Practicas	Se encuentra almacenadas las prácticas de los estudiantes.
Permiso	Permisos para cada uno de los usuarios.
Evaluacion_toerica	Corresponde a las respuestas a una serie de preguntas sobre los distintos temas que tiene la herramienta.
Mensajes	Son los distintos mensajes que los usuarios entregan sobre la herramienta.
Problemas_teóricos	Son las preguntas sobre los distintos temas que tiene la herramienta.

Tabla 32. Descripción de las tablas en el Diseño de la Base de Datos.

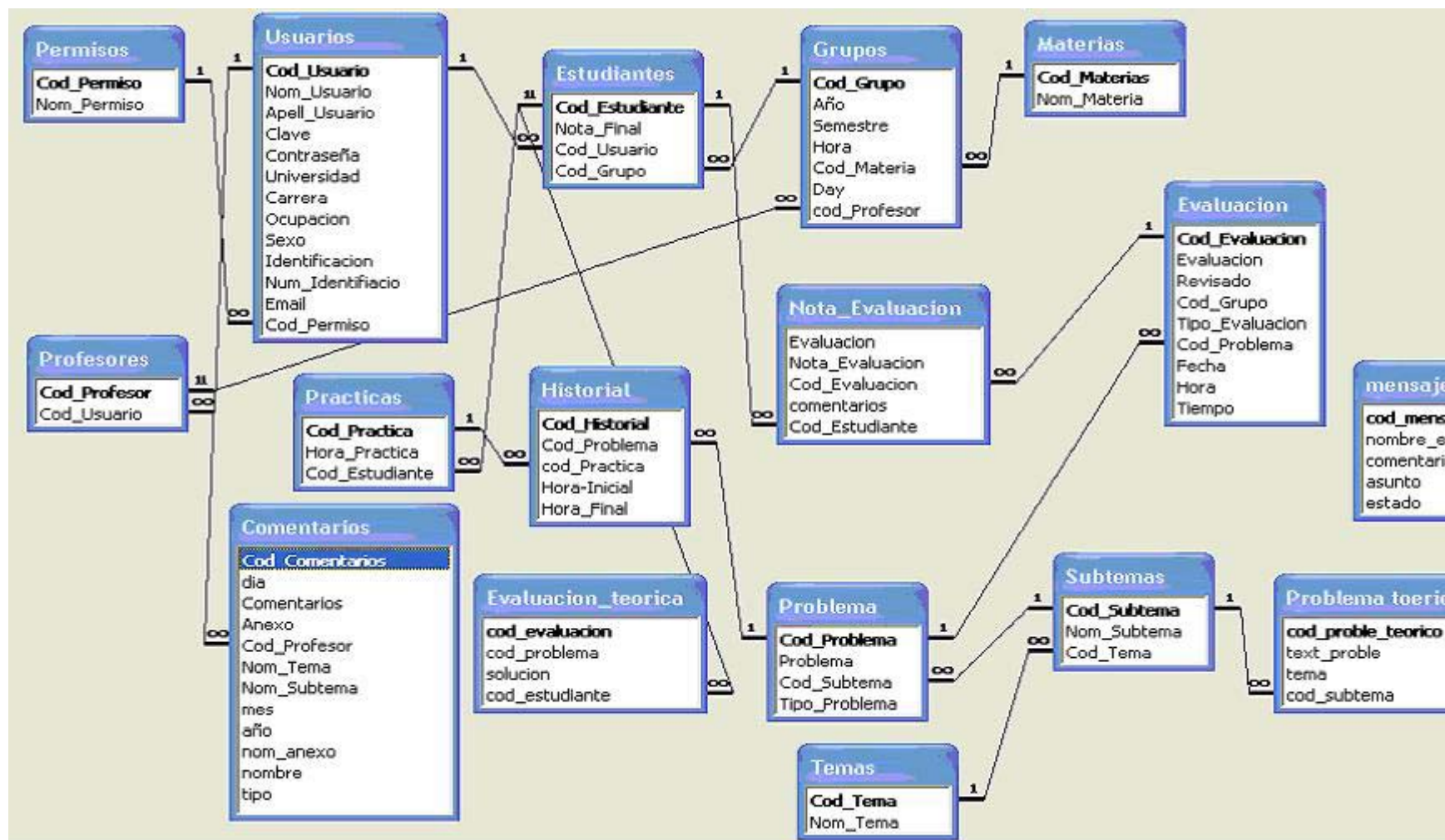


Figura 61. Diagrama Entidad - Relación.

8.3.6 SOFTWARE DE LA HERRAMIENTA Y DE LA CAPA INTERMEDIA.

El software de la herramienta y la capa intermedia constituyen los cimientos de un software, y como toda la funcionalidad de esta descansa sobre diferentes plataformas como: sistemas operativos, sistemas de gestión de bases de datos, software de comunicaciones etc. La elección de productos software es objetivo de la fase de inicio y elaboración.

La herramienta software GEDISO 1.0, se desarrolla bajo el sistema operativo **WINDOWS 2000**, que se caracteriza por:

Se basa en la fuerza de Windows NT® para brindar confianza; una mayor seguridad, la habilidad de operar de manera integrada con su red actual.

Windows 2000 Profesional esta diseñado con el objetivo de dar a sus clientes Confiabilidad, mejorando su protección a los archivos del sistemas, brinda una infraestructura segura que le permite seleccionar los niveles de protección para la información y las aplicaciones que necesita.

Independientemente del servidor que usted use, Windows 2000 Profesional maneja una interfaz estándar que permite a los usuarios hacer su trabajo, sin tener que preocuparse de que servidor estén utilizando.

El software estará basado en **PHP (Hypertext Preprocessor)**, que es un lenguaje interpretado de alto nivel embebido en páginas HTML, ejecutado desde el servidor. La mayor parte de su sintaxis ha sido tomada de C, Java con algunas características específicas de si mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. Una de sus características más potentes es su soporte para gran cantidad de bases de datos. Entre su soporte pueden mencionarse InterBase, mSQL, MySQL, Oracle, Informix, PosgreSQL, entre otras. PHP también ofrece la integración con las varias bibliotecas externas, que permiten que el desarrollador haga casi cualquier cosa desde generar documentos en pdf hasta analizar código XML. Además con el apoyo de **MYSQL** para el manejo de base de datos, único sistema gratuito en el mundo que soporta base de datos objeto y SQL, algunas de sus características son: Mayor rendimiento, mayor velocidad tanto al conectar con el servidor como al servir selects y demás, mejores utilidades de

administración (backup, recuperación de errores, etc.), no suele perder información ni corromper los datos, mejor integración con PHP, No hay límites en el tamaño de los registros, mejor control de acceso, en el sentido de qué usuarios tienen acceso a qué tablas y con sus respectivos permisos.

Se utilizará el servidor para páginas **Web Apache**, Apache es uno de los mejores servidores de Webs utilizados en la red Internet desde hace tiempo, únicamente le hace competencia un servidor de Microsoft, el IIS por lo que éste servidor es uno de los mayores triunfos del software libre.

Es un servidor de Web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos (HTTP 1.1)

- ◆ Implementa los últimos protocolos, aunque se base en el HTTP / 1.1
- ◆ Puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo y con la API de programación de módulos.
- ◆ Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches.

Es utilizado en la actualidad por su eficiencia, funcionalidad y velocidad y que de acuerdo al estudio hecho por Netcraft²⁸ es el servidor WWW más popular del momento, ya que un 62.55%²⁹ de los servidores lo utilizan.

Se maneja DELPHI que cuenta con todas las características para crear aplicaciones con avanzadas interfaces de usuario, servicios locales y distribuidos y acceso a todo tipo de orígenes de datos, esta se utilizara para la creación del motor de dibujo que se utilizara para resolver parte de los ejercicios, en el modulo de problemas y evaluación, los demás serán implementados bajo FLASH mx, el cual es estándar para la creación de animaciones y gráficos vectoriales para uso en Internet.

Todas las herramientas mencionadas anteriormente son productos “hechos en Internet”, lo cual se traduce en costo nulo excepto el Windows 2000 y delphi que necesariamente requiere licencia de funcionamiento representando un costo y con un grupo de desarrolladores de excelente calidad, logrando una solución económica sin que el software pierda su robustez y seguridad.

²⁸ El Netcraft, encuesta de servidores Web, es un estudio de los servidores Web usados en los computadores conectados a Internet. Para ello se consultan sistemáticamente la mayor cantidad posible de servidores que proveen un servicio HTTP .

²⁹ Crecimiento de Servidores WWW Abril 2001. Este estudio se realizó entre 28.669.939 sitios conectados a Internet.

8.4 IMPLEMENTACIÓN

Este flujo de trabajo implementa y prueba los componentes arquitectónicamente significativos a partir de los elementos de diseño obtenidos. El resultado es la línea base de la arquitectura, implementada normalmente a partir de menos del 10 por ciento de los casos de uso.

La mayor parte de la arquitectura del software es capturada durante el diseño, la implementación tiene como propósito desarrollar la arquitectura y el sistema como un todo. Dado el enfoque incremental del proceso unificado la implementación se dará a través de una sucesión de pasos pequeños y manejables.

En esta fase, la implementación se enfoca solo a instalar y configurar el software de la herramienta así como de asignar los componentes necesarios a cada uno de los nodos detallados en el diagrama de despliegue.

8.4.1 SOFTWARE DEL SISTEMA

Para la configuración del servidor se instaló el software descrito en el flujo de trabajo de diseño³⁰:

Sistema operativo **WINDOWS 2000**. Además se realizó su configuración de red de la siguiente manera:

- ◆ Dirección IP: 192.168.30.34
- ◆ Mascara de Red: 255.255.255.0
- ◆ Puerta de enlace o Gateway: 192.168.34.1
- ◆ DNS: 192.168.19.2 –192.168.19.6

La dirección IP suministrada es una dirección no valida en Internet. La dirección valida será asignada una vez sea terminada la etapa de construcción y se inicie la transición del sistema.

³⁰ Para mas información consulte 8.3.6 Software del sistema y de la capa intermedia en al pagina 142.

Como servidor páginas Web se instaló y configuró Apache; **PHP (Hypertext Preprocessor)**, como lenguaje de scripts que se ejecuta en el servidor y para el manejo de la base de datos se instaló y configuró **MYSQL**.

La base de datos diseñada en el flujo de análisis fue implementada en este motor de base de datos según el diagrama entidad - relación de la Figura 61. Los tipos de datos de los atributos correspondientes serán detallados en el diccionario de datos del Anexo D, Diccionario de Datos Página 218. Además se crearon las restricciones y funciones que permitirán preservar la integridad referencial entre las tablas.

Capítulo 9.

EVALUACIÓN

El cumplimiento de la fase de evaluación se basa en determinar si la arquitectura escogida es estable y sólida como para guiar el trabajo a lo largo de la fase de construcción y durante toda la vida útil del sistema. Esta arquitectura debe poder adaptarse al cambio y permitir un fácil crecimiento del sistema.

Para lograr esta arquitectura se deben recopilar la mayoría de casos de uso, para obtener una visión global del sistema y analizar aquellos que tengan impacto sobre la arquitectura. Además se debe seguir el control de los riesgos críticos surgidos desde la fase de inicio y de aquellos que surjan durante el desarrollo de esta fase.

9.1 DESARROLLO DEL MODELO DE CASOS DE USO

Para concluir este capítulo y llevar una continuidad con la fase anterior y mostrar claridad con respecto al desarrollo de los casos de uso encontrados hasta el momento se presentará la siguiente tabla que contiene los casos de uso identificados hasta el momento especificando la fase en la que fueron hallados cada uno, así como el proceso en que están, es decir, si solo fueron identificados, o ya se describieron, detallaron y si su análisis ha sido satisfactorio y no se necesita de un mayor detalle para su total comprensión.

Es importante recordar que no necesariamente todos los casos de uso requieren un análisis exhaustivo para su total comprensión, o más bien, por su sencillez son comprendidos rápidamente y por lo tanto solo requieren una breve descripción del mismo sin profundizar en su análisis.

Casos de Uso Identificados	Identificado	Descrito	Analizado	Diseñados e implementados	¿Se comprende completamente?
Verificar Registro del Usuario.	Inicio	Inicio			Si
Escoger Tema	Inicio	Inicio			Si
Escoger Subtema	Elaboración	Elaboración	Elaboración		Si
Consultar Ejemplos	Inicio	Inicio	Elaboración		Si

Agregar Anexos	Elaboración	Elaboración			Si
Ver Anexos	Elaboración	Elaboración			Si
Verificar Registro del Usuario.	Elaboración	Elaboración	Elaboración		Si
Escoger Tema Ejercicio	Inicio	Inicio	Inicio		Si
Escoger Subtema Ejercicio	Elaboración	Elaboración			Si
Escoger Ejercicio	Inicio	Inicio			Si
Ver Problemas	Elaboración	Elaboración	Elaboración		Si
Ver registros por parte de los profesores.	Inicio	Inicio	Inicio - Elaboración		No
Verificar Registro del Usuario.	Elaboración	Elaboración			Si
Escoger Tema Evaluación	Inicio	Inicio			Si
Escoger Subtema Evaluación	Elaboración	Elaboración			Si
Escoger Ejercicio Evaluar	Inicio	Inicio			Si
Enviar Ejercicio para calificación	Elaboración	Elaboración			No
Ver Nota de evaluaciones por parte de los estudiantes.	Elaboración	Elaboración			No
Crear Informes por parte de los profesores.	Inicio	Inicio	Inicio - Elaboración		Si
Verificar Registro del Administrador.	Elaboración	Elaboración			Si
Crear Grupos.	Inicio	Inicio	Elaboración		Si
Modificar Grupos.	Inicio	Inicio	Elaboración		Si
Eliminar Grupos.	Inicio	Inicio	Elaboración		Si
Listar Grupos.	Elaboración	Elaboración	Elaboración		Si
Crear usuario	Elaboración	Elaboración	Elaboración		Si
Modificar usuario	Elaboración	Elaboración	Elaboración		Si
Borrar usuario	Elaboración	Elaboración	Elaboración		No
Listar Usuarios	Elaboración	Elaboración	Elaboración		Si
Insertar los Ejercicios	Inicio	Inicio	Elaboración		Si

en la base de datos.					
Sacar los Ejercicios de la base de datos.	Inicio	Inicio	Elaboración		Si
Eliminar el historial creado en el semestre.	Elaboración	Elaboración	Elaboración		No
Eliminar registro de evaluaciones	Elaboración	Elaboración	Elaboración		No
Ver mensajes	Elaboración	Elaboración	Elaboración		Si

Tabla 33. Casos de Uso identificados hasta el final de la fase de Elaboración.

9.2 CUMPLIMIENTO DE CRITERIOS.

En varias oportunidades, se ha hecho mención del objetivo de esta fase, el cual es encontrar la arquitectura estable que guíe el desarrollo del sistema. En la búsqueda de este objetivo se han enfocado las tareas de los diferentes flujos de trabajo, obteniendo la vista de la arquitectura estable.

Durante esta fase se identificaron la mayor parte de los requisitos del sistema, se verificó la importancia de los actores encontrados en la fase de inicio y se detalló gran parte de los casos de uso. Dado que la mayoría de estos casos de uso no tienen complicaciones ni excepciones mayores en su desarrollo, fue suficiente con una descripción de los mismos para ser comprendidos totalmente y dar paso a su implementación en la fase de construcción.

Sin embargo, se analizaron los casos de uso claves para tener una visión global del sistema. Para ello se escogió un caso de uso relevante de cada paquete de análisis, que reuniera los aspectos más importantes de los demás casos de uso de dicho paquete. Así se obtuvo de forma directa el diagrama Entidad-Relación y la interfaz de usuario.

También, gracias a los casos de uso y a los subsistemas de diseño obtenidos a partir de los paquetes de análisis, se obtuvo fácilmente las opciones que debería soportar el sistema y la manera como estas se organizarían para brindar al usuario una fácil visualización de las mismas.

Los diferentes perfiles del sistema surgieron de los actores del sistema, que además contribuyeron a verificar que la línea base de la arquitectura satisficiera las necesidades de todos los usuarios.

Además, la arquitectura elegida permite una fácil expansión del sistema gracias al diseño por subsistemas, resulta fácil adicionar un nuevo subsistema o una nueva característica sin tener que reestructurar toda la aplicación. También el diseño por capas permite realizar fácilmente modificaciones en la lógica de la aplicación sin comprometer la capa de presentación ni la de datos, igualmente se puede modificar la interfaz sin afectar la capa de aplicación o sea la lógica del sistema.

PARTE IV

FASE DE CONSTRUCCIÓN

La mayor parte del trabajo de identificación de características y análisis de funcionalidad fue realizada en las fases de inicio y elaboración. Ahora en la fase de construcción, el énfasis estará en la implementación y pruebas del software, hasta conseguir una versión ejecutable de la herramienta lo suficientemente satisfactoria como para permitir la primera entrega del producto a los usuarios.

Adicionalmente, durante la fase de construcción se crearán diversos documentos que complementan la herramienta, tales como la guía del usuario final. La construcción del sistema se podrá considerar terminada cuando se pueda garantizar la realización de todos los casos de uso y además, que no surgirán riesgos incontrolables al utilizar la aplicación.

El objetivo general de esta fase es alcanzar un producto listo para ser distribuido como versión beta y ser sometido a pruebas en esta fase y en la transición. En este momento los requisitos y la arquitectura son estables y solamente en algunos casos se necesitarán realizar pequeños cambios a ellos.

Los objetivos específicos de esta fase son:

- ◆ La extensión de la identificación, descripción y realización de todos los casos de uso.
- ◆ El mantenimiento de la integridad de la arquitectura, modificándola solo cuando sea necesario.
- ◆ La monitorización de los riesgos críticos y significativos encontrados durante las dos primeras fases y su mitigación si se materializan.
- ◆ La creación de los manuales de usuario y de administración del sistema.

Al comienzo de la fase de construcción es importante realizar su planeación para determinar el número de iteraciones necesarias para el desarrollo del software, así como para establecer los objetivos y productos resultantes de cada iteración.

En pocas palabras, en la fase de elaboración, se analizaron los casos de uso, y paquetes que eran significativos desde el punto de vista de la arquitectura y al final se logró obtener un modelo del análisis casi completo al que se le añadirán simplemente actualizaciones menores en la fase de construcción. Con respecto al modelo de diseño se tiene una situación semejante, ya que no se añadirán nuevos subsistemas sino que se conservarán los que ya existen en forma de esqueleto en la línea base de la arquitectura.

CAPÍTULO 10.

FLUJOS, ACTIVIDADES Y PRODUCTOS A ENTREGAR

Como se ha dicho, el principal producto a entregar en esta fase es la primera versión con capacidad operativa de la herramienta software GEDISO 1.0, además de las versiones finales de modelos de casos de uso y de la descripción de la arquitectura.

Dado que la atención principal de esta fase recae sobre los flujos de implementación y pruebas, sus actividades serán resaltadas en los capítulos de esta parte del libro, los demás flujos solo se mencionarán en caso estrictamente necesario. Los productos de esta fase se encuentran en los anexos correspondientes³¹

³¹ Anexo C- Modelo de Casos de Uso de la Herramienta, pagina 209, Anexo D- Diccionario de datos pagina 211.

Capítulo 11.

ESTANDARES DE IMPLEMENTACIÓN

11.1 CONVENCIONES DE CODIFICACIÓN.

Debido a la importancia que tiene el brindar mantenimiento a una aplicación específica, es importante fijar ciertos estándares para facilitar esta labor en el futuro. En esta sección se establecerán estándares para obtener un código fácil de entender y mantener en la herramienta GEDISO 1.0.

En cuanto a la construcción de las sentencias se tuvieron en cuenta las siguientes pautas:

- Dado que en PHP, lenguaje utilizado para el desarrollo, es sensitivo al uso de mayúsculas y minúsculas, se conserva el estándar seguido por el lenguaje, es decir, las palabras reservadas del lenguaje como funciones y variables propias de PHP se escribirán en minúscula, salvo las excepciones dadas por el mismo lenguaje.
- Para la elección de nombres de variables se utiliza el siguiente estándar:
 - ◆ Si la variable almacena información sobre un campo, una tabla o un registro de la base de datos, entonces la variable llevará el nombre del objeto de la base de datos al que hace referencia. Las variables de sesión también seguirán este estándar.
 - ◆ Las demás variables, describirán el contenido de los datos almacenados en ella y que faciliten la lectura del código.
 - ◆ Las variables tomadas de los controles propios de un formulario Web, seguirán el esquema presentado por el primer estándar sobre la elección de nombres dado que estos controles generalmente se encargan de obtener información que será almacenada en la base de datos. Sin embargo para distinguirlas de las demás variables se les antepone un prefijo dependiendo del control que utilizan. Por ejemplo la variable `txtnom_usuario` tiene el prefijo `txt` porque utiliza un control de texto para mostrar o recibir el nombre de un usuario. Los prefijos utilizados se encuentran en la Tabla 34 ubicada a continuación.

Control	Prefijo
Caja de Texto	Txt, php
Lista o menú	Cbo, php
Botón	Btn, php
Botón	buttonEntrada, delphi
Caja de texto	EditClave, delphi
Lista o menu	ComboboxTema, delphi

Tabla 34. Estándares para las variables de los formularios de datos.

➤ Con el objetivo de obtener un código de fácil lectura se seguirán las siguientes reglas:

- ◆ Usar el sangrado de las sentencias para indicar la subordinación.
- ◆ Colocar las etiquetas de bloque (<?, ?>, {,} , etc.) en líneas separadas
- ◆ Evitar el uso de complicadas comparaciones condicionales y usar paréntesis para clarificar las expresiones lógicas y aritméticas y en lo posible simplificarlas.
- ◆ Agrupar cantidad grande de código dentro de un bloque colocándolas en líneas consecutivas y dejando una línea en blanco entre todas las demás.
- ◆ Separar las cabeceras de la función con una línea en blanco.
- ◆ Integrar comentarios de bloque en el código, utilizando la misma sangría que el bloque de código al que se refiere cada uno.

➤ Para adicionar de comentarios de explicación del código se seguirá el siguiente estándar de acuerdo a los dos tipos principales de comentarios:

- ◆ Comentarios de línea. Se colocan directamente dentro del código y explican cuestiones que allí surjan.

➤ En cuanto a los Datos de Entrada y Salida, se tuvo en cuenta:

- ◆ Validar los datos de entrada antes de realizar operaciones en la base de datos.
- ◆ Realizar la comprobación de cada una de las operaciones que manejan los datos como por ejemplo las instrucciones de inserción, borrado y/o modificación.

11.2 PERFILES DE USUARIO

Es un aspecto de vital importancia, de ahí que se definieran perfiles de usuario para el acceso a la herramienta GEDISO 1.0 que permiten desplegar las opciones de cada menú de acuerdo al perfil asignado.

Es necesario que todos los usuarios que van a hacer uso de este sistema obtengan una clave y una contraseña, una vez el usuario haya obtenido la clave y la contraseña podrán navegar a través de las diversas opciones que ofrece la herramienta, de acuerdo al perfil que le corresponda, evitando así que el usuario pueda acceder a una opción sin tener el permiso respectivo.

Los perfiles de usuario son:

- ◆ Administrador (ADM): esta persona será la encargada de toda la herramienta, es la única que tiene acceso a todas las opciones.
- ◆ Profesor (PROF): lo obtendrán todos los profesores que dictan Geometría Descriptiva.
- ◆ Estudiante (EST): lo obtendrán todos los estudiantes que se encuentren matriculados en el programa.
- ◆ Invitado (INV): El invitado solo puede ver las opciones del módulo de teoría, no se despliegan opciones de trabajo para el, es personal ajeno a la universidad.

Las opciones que se desplegarán de acuerdo al perfil se encuentran en la Tabla 35.

OPCIÓN	PERFIL			
	ADM	PROF	EST	INV
Herramienta software GEDISO 1.0				
ADMINISTRACIÓN				
Crear usuarios	x			
Modificar usuarios	x			

OPCIÓN	PERFIL	ADM	PROF	EST	INV
Listar usuarios		x			
Eliminar usuarios		x			
Crear grupos		x			
Modificar grupos		x			
Listar grupos		x			
Eliminar grupos		x			
Insertar problemas		x			
Insertar teóricos		x			
Borrar Problemas		x			
Cambiar Estado del Problema		x			
Eliminar historial		x			
Eliminar registros de evaluación		x			
Ver mensajes		x			
Copia de Seguridad		x			
EVALUACION					
Evaluación teórica		x	x	x	
Evaluar ejercicios		x	x	x	
Ver evaluación		x	x		
Asignar notas		x	x		
Ver notas		x	x	x	
PROBLEMAS					
Solucionar ejercicios		x	x	x	
Ver registros		x	x		
Ver problemas		x	x		
TEORIA					
Ver información		x	x	x	x
Agregar anexos			x		
Ver anexos		x	x	x	x
OTRAS					
Cambiar contraseña		x	x	x	x
Datos personales		x	x	x	x
Enviar comentario		x	x	x	x
Ayuda		x	x	x	x

OPCIÓN	PERFIL	ADM	PROF	EST	INV
Registrarse		x	x	x	x

Tabla 35. Opciones de trabajo de la herramienta, por su perfil.

11.3 CONVENCIONES DE INTERFAZ

Para la interacción con el usuario se tuvieron en cuenta los siguientes aspectos genéricos en el diseño de la interfaz según Roger Presman en su libro “Ingeniería del Software”³² y Tobias Ratschiller y Till Gerken en su libro “Creación de Aplicaciones Web con PHP 4”³³:

- ◆ Preguntar por la verificación de cualquier acción destructiva que no sea común.
- ◆ Utilizar verbos de acción simples o frases verbales cortas para definir las operaciones que realizará el usuario.
- ◆ Utilizar formatos de presentación que permitan una asimilación rápida de la información.
- ◆ Utilizar etiquetas consistentes y abreviaciones estándar.
- ◆ Producir mensajes de error significativos.
- ◆ Utilizar Mayúsculas y minúsculas, tabulaciones y agrupaciones de texto para ayudar a la comprensión.
- ◆ Usar eficientemente el espacio disponible en la pantalla.
- ◆ Minimizar el número de acciones de entrada de datos que debe realizar el usuario.
- ◆ Cuando se produzcan mensajes de error se debe indicar las consecuencias del mismo.
- ◆ El formato para presentación de mensajes de error debe ser suficientemente llamativo.

11.4 MENSAJES DEL SISTEMA

Los mensajes de error seguirán un estándar para su presentación y serán mostrados en una de las siguientes formas:

³² Pressman, Roger. Ingeniería del software. Un enfoque práctico. Cuarta edición. McGraw Hill. España, 1998. Capítulo 4. Pagina 479-500.

³³ Ratschiller, Tobias. Gerken, Till. Creación de Aplicaciones Web con PHP 4. Primera edición. Prentice Hall. España, 2001. Pagina 8-25.

- ◆ *En una ventana propia del navegador.* Estos mensajes son generados por la capa de presentación, y se deben a validaciones o preguntas realizadas al usuario antes de enviar los datos al servidor para su procesamiento.



Figura 62. Mensaje en una ventana propia del navegador.

- ◆ *En una pagina nueva, que generó el error.* Este tipo de mensajes se muestra cuando el error generado no es crítico para el sistema. Un ejemplo es cuando un usuario no introduce toda la información necesaria en un formulario o cuando la introduce incorrectamente.



Figura 63. Mensaje en la misma incrustada en la misma pagina que generó el error.

- ◆ *En una pagina nueva.* Cuando sea un error o problema crítico. Un ejemplo de este caso es cuando un usuario desea ingresar al sistema de manera ilegal.



Figura 64. Mensaje en una ventana nueva. Mensaje critico.

11.5 SUBSISTEMAS DE IMPLEMENTACIÓN

Los subsistemas de implementación están relacionados con los subsistemas de diseño, y se podrá seguir el trazado uno a uno entre estos. Los subsistemas proporcionan una forma de organizar los componentes del modelo de implementación, como scripts y funciones, en trozos más manejables.

Es importante entender que un subsistema de implementación se manifiesta a través de un “mecanismo de empaquetamiento” concreto en un entorno de implementación determinado, tales como un paquete en Java, un proyecto en Visual Basic o un directorio de archivos en C++ y en PHP.

	Directorio	Subsistema de diseño	Descripción
1	Zona de Teoría	De Teoría	En este Directorio se encuentra el modulo de teoría con toda la información correspondiente al pensul académico de la asignatura geometría descriptiva I y II.
2	Zona de Ejercicios	De Ejercicios	En este directorio se encuentra el módulo de problemas que contiene los ejercicios a realizar por parte del estudiante.
3	Zona de Evaluación	De Evaluación	En este directorio se encuentra el módulo de evaluación en donde se evaluarán los conocimientos de los estudiantes.
4	Zona de Administración	De Administración	En este directorio se encuentra el módulo que lleva a cabo tareas administrativas del sistema, tales como creación de usuarios, materias y grupos. Además otras opciones avanzadas.
5	Zona de Registros	De Registro	En este directorio se encuentra el módulo que maneja los progresos académicos de los estudiantes: registro de prácticas y notas de evaluaciones para cada estudiante. Este no se encuentra como una zona aparte sino que esta dentro de otras zonas como en le ejercicios y evaluación.

Tabla 36. Subsistemas de Implementación.

CAPÍTULO 12.

SUBSISTEMAS GENERALES

12.1 SUBSISTEMA DE GESTIÓN DE SESIONES

Durante la fase de elaboración, parte III de este libro, se diseñaron los subsistemas que formarían parte del sistema GEDISO 1.0. Presentándose por primera vez el subsistema de gestión de sesiones³⁴ y se explicó su funcionamiento general a través de un ejemplo expresado en el diagrama de secuencia de la Figura 58. Diagrama de Secuencia de las interacciones entre subsistemas.

En esta sección se muestran otros aspectos importantes de este módulo en su implementación para el caso particular del desarrollo con PHP, el cual soporta el concepto de sesiones, básicamente es un sistema que preserva ciertos datos en una serie de accesos, permitiendo construir aplicaciones más personalizadas.

Cuando el usuario, desea hacer uso de un proceso para llevar a cabo un caso de uso de cualquier subsistema, se debe verificar que dicho usuario tenga permiso para su ejecución y para acceder o modificar la información que este maneja. Dada la naturaleza del sistema es de vital importancia evitar el ingreso de usuarios no autorizados al sistema.

A cada visitante que acceda al sistema se le asigna un único `cod_usuario`, el conocido inicio de sesión. Éste se almacena en una cookie del usuario o bien se propaga con la URL. El soporte de sesión le permite transportar tantas variables como se desee a través de las solicitudes del cliente.

Como se dijo anteriormente, hay dos formas de propagar un `cod_usuario` de sesión: por cookies o por parámetros URL. El módulo de sesión soporta ambos métodos. Los Cookies son ideales, pero su permanencia no la controla el servidor, sino el cliente. El segundo método integra el `cod_usuario` de sesión en las URL. PHP es capaz de hacer esto de forma transparente cuando se ha compilado con `-enable-trans-sid`. Si se

³⁴ Mirar sección 8.3.2 - Diseño del Subsistema de Gestión de Sesiones, en la pagina 131.

activa esta opción las URL relativas se cambiarán automáticamente para contener el cod_usuario de sesión.

12.1.1 IMPLEMENTACIÓN

En la herramienta GEDISO 1.0, cuando un usuario desea ingresar al sistema, envía su clave y su contraseña de acceso, la herramienta valida esta información y crea una nueva sesión registrando las variables de la Tabla 37, que serán preservadas mientras se encuentre dentro del sistema. Para el registrar las variables se utiliza el siguiente código:

```

1. session_start();
2. validación de la clave y contraseña
3. Si la validación es satisfactoria
    o //Registrar variables
    o $_SESSION['clave']=$clave;
    o $_SESSION['contraseña']=$contraseña;
    o $_SESSION['cod_usuario']= $userpermi;
    o $_SESSION['inicioseccion']=$usertipo;
4. Si la validación no es satisfactoria
    o Mensaje de error, e interrumpe el acceso al sistema.

```

Tabla 37. Código para el inicio de sesión y registro de las variables

	Variables de sesión	Descripción
1	cod_usuario	identificador del perfil del usuario.
2	Clave	Clave del usuario en sesión. Utilizado para personalizar el sitio para el usuario.
3	Inicio sección	identificador del usuario en sesión. Utilizado para consultas en la base de datos y para el control de accesos del usuario a las diferentes opciones.
4	Contraseña	Contraseña del usuario en sesión. Utilizado para personalizar el sitio para el usuario.

Tabla 38. Variables de sesión registradas para cada usuario que acceda a la Herramienta.

Para implementar este subsistema se ha construido en lenguaje PHP los procesos que permiten cumplir con las funciones dadas a este subsistema. A continuación se valida que la clave y la contraseña que el usuario ingresó sean válidos y se registran las variables de sesión necesarias para su identificación. En la Tabla 37. se encuentra el

segmento de código php que registra dichas variables y en la Figura 65. se presenta la página que permite el ingreso al sistema.

Para verificar el acceso a cada modulo de la herramienta, se le asignaran permisos de acceso por cada uno de los perfiles que se manejan, esto debido a que un modulo puede contener funciones que son utilizadas por diferentes usuarios.

Los permisos serán establecidos al inicio de cada sesion, se verificara el acceso al modulo, si es correcto presenta la pagina, de lo contrario no entra, regresando a el lugar en que intento acceder a la pagina, el código revisa el perfil del usuario en sesión (cod_usuario) y crea una nueva sesión, registrando las variables de sesión propias del mismo que serán preservadas mientras se encuentre dentro del sistema.

```
//permisos para esta pagina
$usertipo=$_SESSION['cod_usuario'];
if ($usertipo !=1)
{
    print("<SCRIPT language=JavaScript>window.location=\"Paginas Modulos.php\"
</script>");
}

// Revisa si la sesión existe y si es legal su acceso.

Mensaje de error, e interrumpe el acceso al sistema.
```

Tabla 39. Sección de Validación de Acceso a la Herramienta

A continuación se presentara el código completa de la pagina inser.php, que contiene las variables de sección.

Código PHP	Descripción
<pre>// verifica la existencia del usuario en el sistema \$txtclave = \$_POST["txtclave"]; \$txtcontraseña = \$_POST["txtcontrase"]; \$query =("SELECT clave, contraseña FROM usuario WHERE clave ='\$txtclave' && contraseña ='\$txtcontraseña"); \$consulta= mysql_query (\$query,\$link); if(!mysql_fetch_array(\$consulta)) { echo("<script language=javascript>alert('Alerta, su clave o contraseña son incorrectas.');<!--></script>"); \$imprimir=1; \$_SESSION['imprimir']=\$imprimir; \$_SESSION['txtclave']=\$txtclave; print("<script language=javascript>window.location=\"PAGINAINICIO.php\" </script>"); } else {</pre--> </pre>	<p>Con esta sección de código se verifica que la clave y la contraseña de usuario existan en la base de datos y se registran dichas variables, realizando una consulta para conocer el tipo de usuario al que pertenece y se registra esta variable la cual permite el acceso del usuario para ejecutar ciertos casos de uso.</p>

<pre> \$codtipo="SELECT cod_usuario, cod_permiso FROM usuario WHERE clave=\$txtclave"; \$codtipo1=mysql_query(\$codtipo,\$link); \$infotipo=mysql_fetch_array(\$codtipo1); \$usertipo=(\$infotipo["cod_usuario"]); \$userpermi=(\$infotipo["cod_permiso"]); \$clave=\$txtclave; \$contraseña=\$txtcontraseña; \$_SESSION['clave']=\$clave; \$_SESSION['contraseña']=\$contraseña; \$_SESSION['cod_usuario']= \$userpermi; \$_SESSION['inicioseccion']=\$usertipo; // Registra en la sesión el id y la clave de usuario echo("<SCRIPT language=JavaScript>window.location=\"Paginas Modulos.php\" </script>"); } </pre>	
---	--

Tabla 40. Código PHP que permite validar el ingreso de un usuario al sistema con sus respectivos permisos

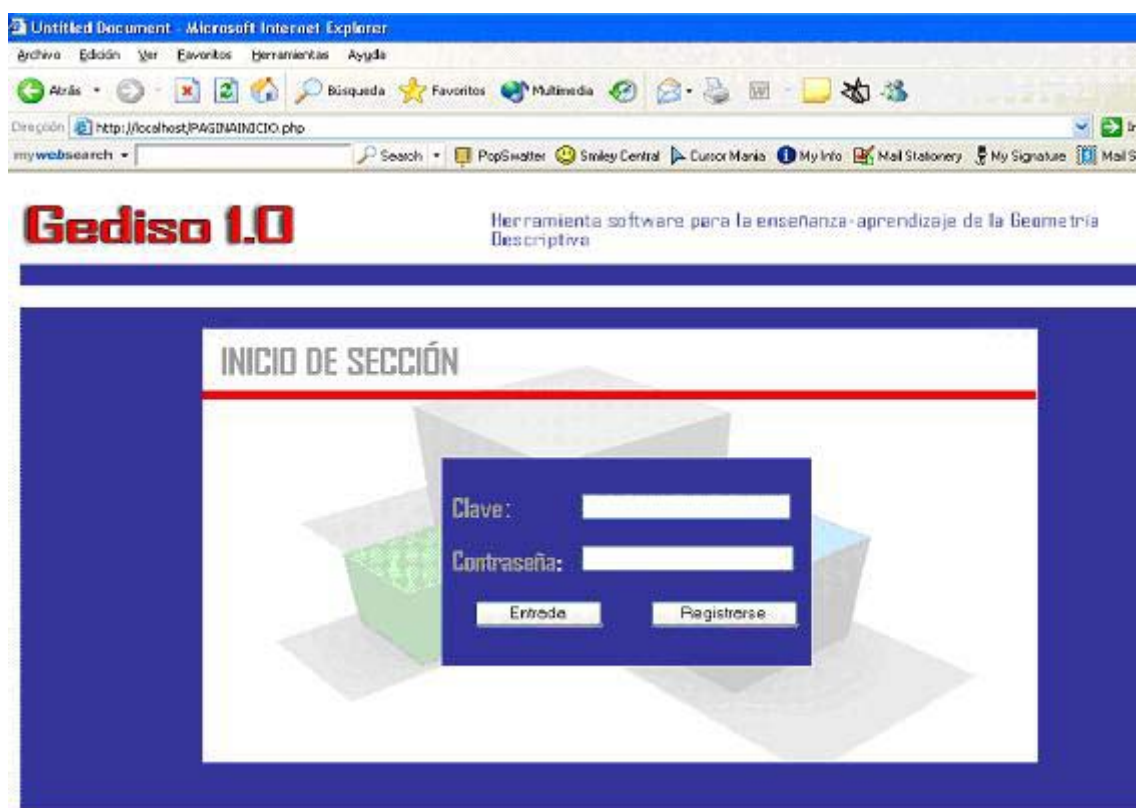


Figura 65. Interfaz que permite ingresar a la Herramienta.

12.1.2 PRUEBAS


12.1.2.1 Casos de Prueba

- Validar que el usuario que desea ingresar al sistema exista en la base datos

- ◆ Verificar el tipo de usuario al que pertenece para determinar a cuales casos de uso puede tener acceso.

12.1.2.2 Procedimiento de Prueba

Para este subsistema se validara el caso de prueba: Validar que el usuario que desea ingresar al sistema exista en la base datos, para ello se ingresa un nombre de usuario válido y una contraseña inválida, en este caso debe aparecer un mensaje de error. Posteriormente se ingresan tanto nombre de usuario como contraseña válidos y debe permitir el ingreso al sistema.

Datos Entrada	<p>Caso 1: Usuario: gupar (Dato Válido) Contraseña: XXX (Dato inválido)</p> <p>Caso 2 : Usuario: gupar (Dato Válido) Contraseña: xxxxxx (Dato Válido)</p>
Resultados	<p>Caso 1:</p>  <p>Caso 2:</p>

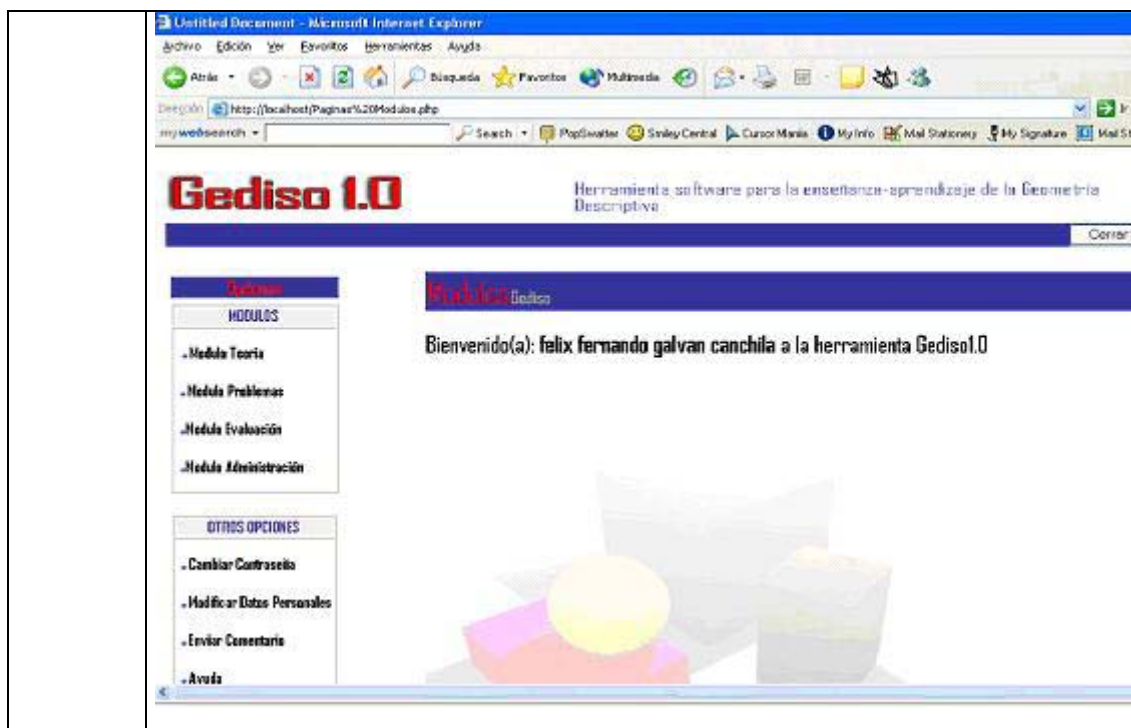


Tabla 41. Prueba para validar el ingreso de un usuario a la Herramienta.

12.2 SUBSISTEMA DE GESTIÓN DE INTERFAZ

Dado que este subsistema es importante desde el punto de vista de la arquitectura, en la fase de elaboración se diseñó e implementó parte del mismo. El principal objetivo de este subsistema es permitir la comunicación entre los subsistemas y los usuarios, para tal fin, se construye un conjunto de páginas Web que faciliten estos procesos.

La interfaz se construye utilizando php y javascript, lenguajes destinados a proporcionar la comunicación entre los usuarios y las aplicaciones así como las validaciones de información que el usuario ingresa.

12.2.1 IMPLEMENTACIÓN

Para este subsistema se construyeron dos grupos de páginas, el primero corresponde a las páginas Web de teoría de la herramienta que son .html, el segundo corresponde a las páginas Web creadas en php para el manejo de la información que tiene la herramienta y por último formularios en delphi que están encargados del desarrollo del módulo de problemas y evaluación.

12.2.1.1 Primer grupo de páginas:

- ◆ Pagina principal donde se desplegara la información (paginas temas.htm): Esta se encuentra conformada por tres frames, los cuales contienen: el menú donde se encuentran los temas que se van a consultar llamado (contenidos.htm); la sección donde se mostrara los temas que contienen el pensum de Geometría Descriptiva (información temas.htm); por ultimo la sección donde irán el nombre de la herramienta, un logo y una imagen (Encabezado.htm).
- ◆ Información de los temas: En esta parte se encuentra un número considerable de páginas que contienen todos los temas de Geometría Descriptiva I y II, además del glosario sobre los temas, prefacio, bibliografía, etc. Algunas de las páginas son: cilindros.htm, conos.htm, azumit.htm, etc.

12.2.1.2 Segundo grupo de páginas:

- ◆ paginaincial.php: Es la primera página que recibirán los usuarios de la herramienta, es solo de presentación.
- ◆ paginainicio.php: Se piden la clave y contraseña de los usuarios de la herramienta.
- ◆ Registrarse.php: Si es un usuario externo de la herramienta se puede registrar para luego hacer uso de esta.
- ◆ Paginas módulos.php: Se encuentran las diferentes opciones que el usuario puede trabajar dependiendo del tipo de usuario.
- ◆ Modulo administración.php: Es un modulo solo para el administrador de la herramienta.
- ◆ Crear usuario.php: Se crean los distintos usuarios de la herramienta.
- ◆ Modificar usuarios.php: Aquí se modifican los usuarios que han sido creados.
- ◆ Eliminar Usuarios.php: Se eliminan los diferentes usuarios de la herramienta.
- ◆ Listas usuarios.php: Lista los estudiantes usuarios de la herramienta dependiendo del curso en el que están.
- ◆ Crear grupos.php: Se crean los diferentes grupos de trabajo de los estudiantes.
- ◆ Modificar grupos: Modifica los grupos de trabajo que ya se crearon.
- ◆ Eliminar grupos.php: Elimina los grupos que ya se han creado.
- ◆ Listar grupos.php: Lista los diferentes grupos de acuerdo a su director de grupo.
- ◆ Inserproble.php: Se insertan los problemas que los estudiantes van a trabajar.
- ◆ Inserprobleteoricos.php: Permite insertar los diferentes problemas teóricos a la herramienta.

- ◆ Modifica estado problema.php: Permite al administrador cambiar el estado del problema de activo a no activo.
- ◆ Borrar problemas.php: Dado el caso de que los problemas almacenados en la base de datos sean incorrectos, estos se pueden eliminar.
- ◆ Eliminar Historial.php: Elimina todo los registros que se almacenan de las diferentes practicas que realizan los estudiantes.
- ◆ Eliminar registro evaluaciones.php: Elimina los registros creados por las diferentes evaluaciones que se han realizado.
- ◆ Backup tablas.php: Realiza una copia de seguridad de toda la base de datos, que puede ser carga de nuevo en cualquier momento.
- ◆ Ver mensajes: Permite ver al administrador los mensajes enviados por los usuarios debido a un error de la herramienta.
- ◆ Ver registros.php: Se observa el registro de las prácticas que los estudiantes hacen esta solo puede ser vista por el profesor.
- ◆ Ver problemas.php: Se pueden observar los diferentes problemas que tiene la herramienta, es exclusiva para los profesores.
- ◆ Ver evaluacion.php: Permite al profesor ver las evaluaciones hechas por los estudiantes.
- ◆ Asignar notas.php: En esta el profesor asigna las notas a las diferentes evaluaciones.
- ◆ Ver notas.php: Los diferentes estudiantes podrán ver sus notas de las diferentes evaluaciones.
- ◆ Cambio de password.php: Utilizada para cambiar la contraseña de los usuarios.
- ◆ Datos personales.php: Los usuarios consultan sus datos personales para ser modificados.
- ◆ Enviar correo.php: Si los usuarios tienen una inquietud o encuentran un error en la herramienta pueden enviar un correo como comentario para informar a los desarrolladores de los problemas.
- ◆ Comen.php: Se utiliza para agregar algún anexo sobre los diferentes temas que tiene la herramienta, solo lo utiliza el profesor.
- ◆ Ver anexos.php: Es de uso particular y es para ver los diferentes anexos sobre los temas de la herramienta.

12.2.1.3 Formularios en Delphi:

- ◆ Principal delphi form: Es el formulario principal que recibe a los usuarios, que les da la opción a los usuarios de realizar ejercicios o tener una evaluación.

- ◆ Verificar delphi form: Si lo que desea es realizar ejercicios este formulario realiza la verificación de su clave y contraseña, si son correctas los deja seguir al formulario de trabajo.
- ◆ Verificación delphi form: Si la opción del usuario es realizar una evaluación este formulario se encarga de verificar que sea un usuario activo de la herramienta, si lo es puede realizar la evaluación.
- ◆ Ejercicios delphi form: Este formulario muestra ciertos datos de los usuarios y luego les permite escoger tema, subtemas y ejercicios para resolver, en este formulario se encuentra la hoja de dibujo que es donde resolverá los problemas.
- ◆ Evaluar delphi form: Este formulario permite realizar una evaluación de los diferentes temas que contiene la herramienta, para luego almacenar el resultado en la base de datos, los cuales luego serán revisados por un profesor.
- ◆ Datos delphi form: Este es modulo que realiza la comunicación con la base de datos para poder ejecutar las sentencias SQL, que tienen los demás formularios.

Para este subsistema se presenta los procesos realizados validar el ingreso de la información de un ente al sistema. El código javascript permite hacer dichas validaciones y mostrar los errores cometidos. En la figura 66. se encuentra la página en la cual se ingresa los datos de los usuarios en mención.

Código javascript	Descripción
<pre> <script language="JavaScript"> else if (frmusuario.txtemail.value.indexOf('@', 0) == -1 frmusuario.txtemail.value.indexOf('.', 0) == -1) { alert("Dirección de e-mail inválida"); frmtes.txtcorrente.focus(); } </pre>	<p>Esta sección de código hace parte de una función que valida que el ingreso de los datos, en este código se valida el email que incluya el símbolo @ y el punto, características que tiene todo correo electrónico. En caso contrario muestra el mensaje de error y el cursor se sitúa en el campo destinado para ello (txtemail)</p>

Tabla 42. Código PHP que permite validar el email de un usuario

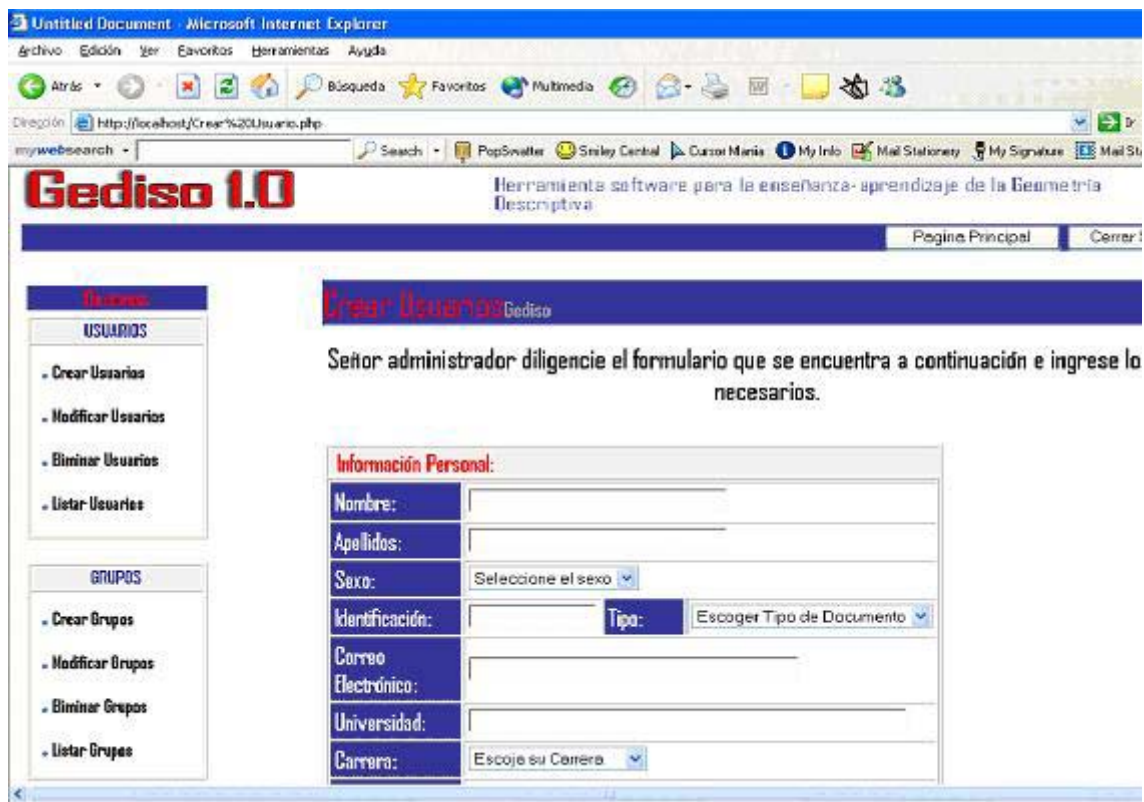


Figura 66. Interfaz que permite Ingresar la información del usuario

12.2.2 PRUEBAS

12.2.2.1 Casos de prueba

Para este subsistema se utilizan pruebas negativas, es decir, pruebas que intentan provocar que el sistema falle.

- ◆ Validar que los datos ingresados correspondan en la base datos
- ◆ Verificar la no existencia de campos nulos donde es necesaria la información.

12.2.2.2 Procedimiento de prueba

A continuación se presenta el caso de prueba: Validar que los datos ingresados correspondan con el formato asignado. Este proceso se puede observar en la tabla 43. El procedimiento para realizar esta prueba consiste en ingresar datos que no correspondan al formato de los campos, por ejemplo, ingresar letra en campos numéricos, no incluir el símbolo @ en los correos, incluir caracteres no válidos, etc.

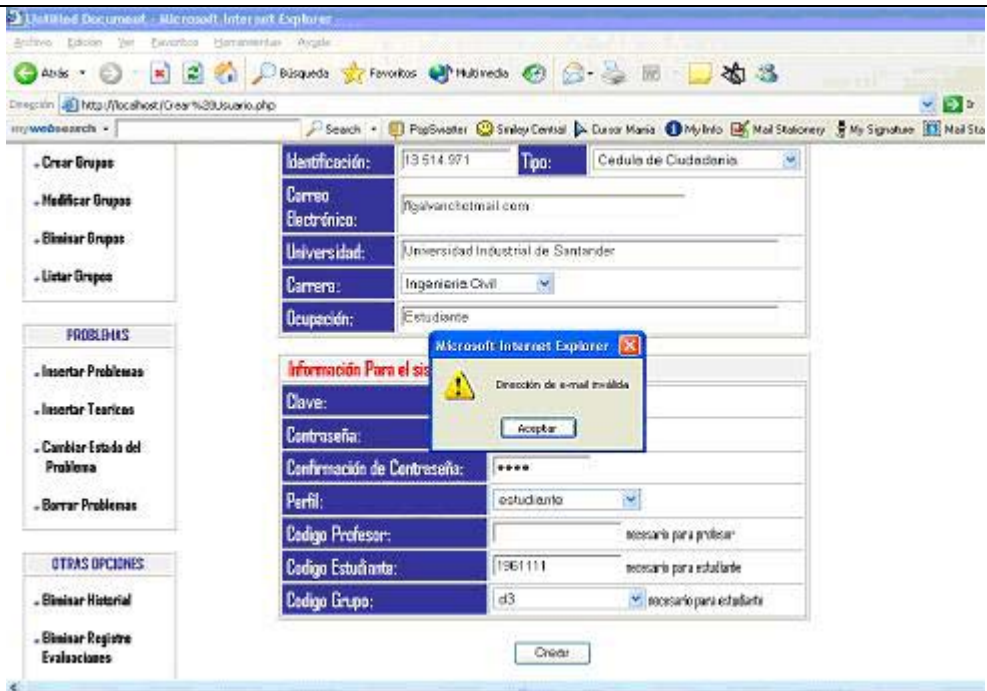
Datos Entrada	<p>Nombre: Felix Fernando Apellido: Galván Canchila Sexo: Masculino. Identificación: 13.514.971 Tipo de documento: Cedula de Ciudadanía. Correo Electrónico: ffgalvanc@hotmail.com. Universidad: UIS. Carrera: Ingeniería Civil. Ocupación: Estudiante. Clave: gato Contraseña: gato Confirmación Contraseña: gato Perfil: Estudiante Código Profesor: Código Estudiante:1961111 Código Grupo: d3</p>
Resultados	 <p>The screenshot shows a web browser window with a URL of http://localhost/Crear%20Usuario.php. The page displays a form with the following fields and values:</p> <ul style="list-style-type: none"> Identificación: 13 514 971 Tipo: Cedula de Ciudadania Correo Electrónico: ffgalvanc@hotmail.com Universidad: Universidad Industrial de Santander Carrera: Ingeniería Civil Ocupación: Estudiante Clave: gato Contraseña: gato Confirmación de Contraseña: **** Perfil: estudiante Código Profesor: (empty) Código Estudiante: 1961111 Código Grupo: d3 <p>An error message box is displayed over the form, stating "Dirección de e-mail inválida" (Invalid email address). The browser window also shows a sidebar with navigation links like "Crear Grupos", "Modificar Grupos", "Eliminar Grupos", "Listar Grupos", "PROBLEMAS", "Insertar Problemas", "Insertar Teóricos", "Cambiar Estado del Problema", "Borrar Problemas", "OTRAS OPCIONES", "Eliminar Historial", and "Eliminar Registro Evaluaciones".</p>

Tabla 43. Prueba para validar el ingreso de un usuario a la Herramienta.

Capítulo 13.

SUBSISTEMAS DE APLICACIÓN.

En la fase de elaboración se identificaron los subsistemas específicos de aplicación. Durante esta fase se implementan estos subsistemas con el fin de asegurar que cumplen su papel en cada construcción. Posteriormente se presentan las pruebas que permiten verificar el funcionamiento de los mismos.

Para no extender el documento en exceso, solo se presenta la implementación y pruebas de una funcionalidad de subsistema. La totalidad de las pruebas se encuentran en el Anexo E, Escenario de Pruebas de la Página 226.

13.1 SUBSISTEMAS DE TEORIA.

Para llevar a cabo la realización de este subsistema, se utilizó la combinación del lenguaje PHP, HTML y FLASH que permite realizar los procesos necesarios para la consulta de la información que contiene la herramienta y el ingreso de anexos que estarán contenidos en la base de datos, que luego serán revisados por los usuarios de la herramienta. Los anexos estarán en la tabla: *comentarios*.

El proceso antes mencionado requiere la información que Gestión de Interfaz para mostrar al usuario el resultado del proceso de búsqueda de información para su estudio.

Para la realización de este subsistema se realizó una página principal constituida por frames que mostraran un menú de los diferentes temas de la herramienta y los respectivos contenidos de estos, los temas que la herramienta abarca son: Prefacio, Marco Teórico, Sistema de Representación, El Punto, Líneas, Planos, Poliedros, Vistas Dirigidas, Desarrollos, Intersección, Glosario, Bibliografía, además al final de cada capítulo se encuentra un ejemplo en Flash que se desplegara, mostrando el desarrollo de un ejercicio paso a paso. Por último se encontraron páginas realizadas en PHP que agregarán los anexos a la Base de Datos y también permitirán verlos cuando los usuarios así lo requieran.

13.1.1 IMPLEMENTACIÓN

El código PHP utilizado para implementar este subsistema está inmerso en 2 páginas Web:

- ◆ Comen.php.
- ◆ VER ANEXOS.php.

Se ha elegido en este caso, los procesos para la inclusión de anexos a la base de datos, donde este procedimiento solo puede ser hecho por los diferentes profesores autorizados por la herramienta, además de la página que permite la búsqueda y observación de estos. En la tabla 44. se encuentra un segmento de php que permite el ingreso de los anexos a la base de datos y en la figura 67. se encuentra la página que permite el ingreso de estos a la tabla comentarios de la base de datos.

Código PHP	Descripción
<pre> <?php //\$_SESSION['name_archivo_extern']=\$_FILES['archivo']['tmp_name']; \$_SESSION['name_archivo']=\$_FILES['archivo']['name']; \$nombre = \$_FILES['archivo']['name']; // Esta variable guardará el tipo de archivo que se intenta subir \$tipo_archivo = \$_HTTP_POST_FILES['archivo']['type']; // Esta variable guardará el tamaño del archivo que se intenta subir \$tamano_archivo = \$_HTTP_POST_FILES['archivo']['size']; if (!(strpos(\$tipo_archivo, "pdf") strpos(\$tipo_archivo, "msword"))) { echo("<SCRIPT language=JavaScript>alert('La extensión del archivo no es correcta. Se permiten archivos .pdf .msword '); </script>"); \$txt_anexo=\$_POST['txt_anexo']; \$_SESSION['txt_anexo']=\$txt_anexo; print("<SCRIPT language=JavaScript>window.location=\"comen.php\""); </script>"); } else { \$ruta=\$directorio.\$_FILES['archivo']['name']; // comprueba si existe un archivo con el mismo nombre // en cuyo caso, no deja subirlo if(!file_exists(\$ruta)) { // comprueba que el tamaño del archivo no sea mayor de 1 mb if(\$tamano_archivo < 1024000) { </pre>	<p>Este segmento de código php realiza la inserción de la información de los anexos en la tabla comentario si se ha presionado el botón de inserción, teniendo en cuenta que previo a la inserción se ha llamado a la pagina conec.php que contiene la función conectarse(), la cual permite hacer la conexión con la base de datos.</p>

<pre> // funcion que sube el archivo if (move_uploaded_file(\$_FILES['archivo']['tmp_name'], \$directorio.\$_FILES['archivo']['name'])) { \$_SESSION['comentario']="http://localhost/Archivos de programa/Apache Group/Apache/htdocs/Archinexos/".\$_SESSION['name_archivo'] ; \$comentario=\$_SESSION['comentario']; \$seleprofe= "select cod_usuario from usuario where clave ='\$txt_clave'"; \$selepro= mysql_query (\$seleprofe,\$link); \$sele=mysql_fetch_array(\$selepro); \$profesor=(\$sele["cod_usuario"]); \$seleccion= "select cod_profesor from profesor where cod_usuario= '\$profesor'"; \$selecc= mysql_query(\$seleccion,\$link); \$selec= mysql_fetch_array(\$selecc); \$profe=(\$selec["cod_profesor"]); \$sql = "INSERT INTO comentarios (dia,mes,año,comentario,nom_tema,nom_subtma,anexo, nom_anexo,cod_profesor,nombre,tipo)"; \$sql .= "VALUES ('\$cbo_dia','\$cbo_mes','\$cbo_año', '\$comentario','\$tema','\$subtema','\$txt_comentario','\$txt_anexo',' \$profe','\$nombre','\$tipo_archivo');"; \$insertar = mysql_query(\$sql,\$link); echo("<SCRIPT language=JavaScript>alert('El archivo ha sido agregado exitosamente .');</script>"); \$_SESSION['txt_anexo']=""; print("<SCRIPT language=JavaScript>window.location='\"comen.php\"' </script>"); } else { echo("<SCRIPT language=JavaScript>alert('No se consiguio subir el archivo'); </script>"); \$txt_anexo=\$_POST['txt_anexo']; \$_SESSION['txt_anexo']=\$txt_anexo; print("<SCRIPT language=JavaScript>window.location='\"comen.php\"' </script>"); } } else { echo("<SCRIPT language=JavaScript>alert(' ERROR: El archivo exceso el tamaño máximo de 1 Mb'); </script>"); \$txt_anexo=\$_POST['txt_anexo']; \$_SESSION['txt_anexo']=\$txt_anexo; print("<SCRIPT language=JavaScript>window.location='\"comen.php\"' </script>"); } } else // existe ya uno con ese nombre { </pre>	
---	--

<pre> echo("<SCRIPT language=JavaScript>alert('Ya existe un archivo con ese nombre, cambialo por favor'); </script>"); \$txt_anexo=\$_POST['txt_anexo']; \$_SESSION['txt_anexo']=\$txt_anexo; print("<SCRIPT language=JavaScript>window.location='comen.php' </script>"); } }?> </pre>	
--	--

Tabla 44. Código PHP para el ingreso de los anexos.

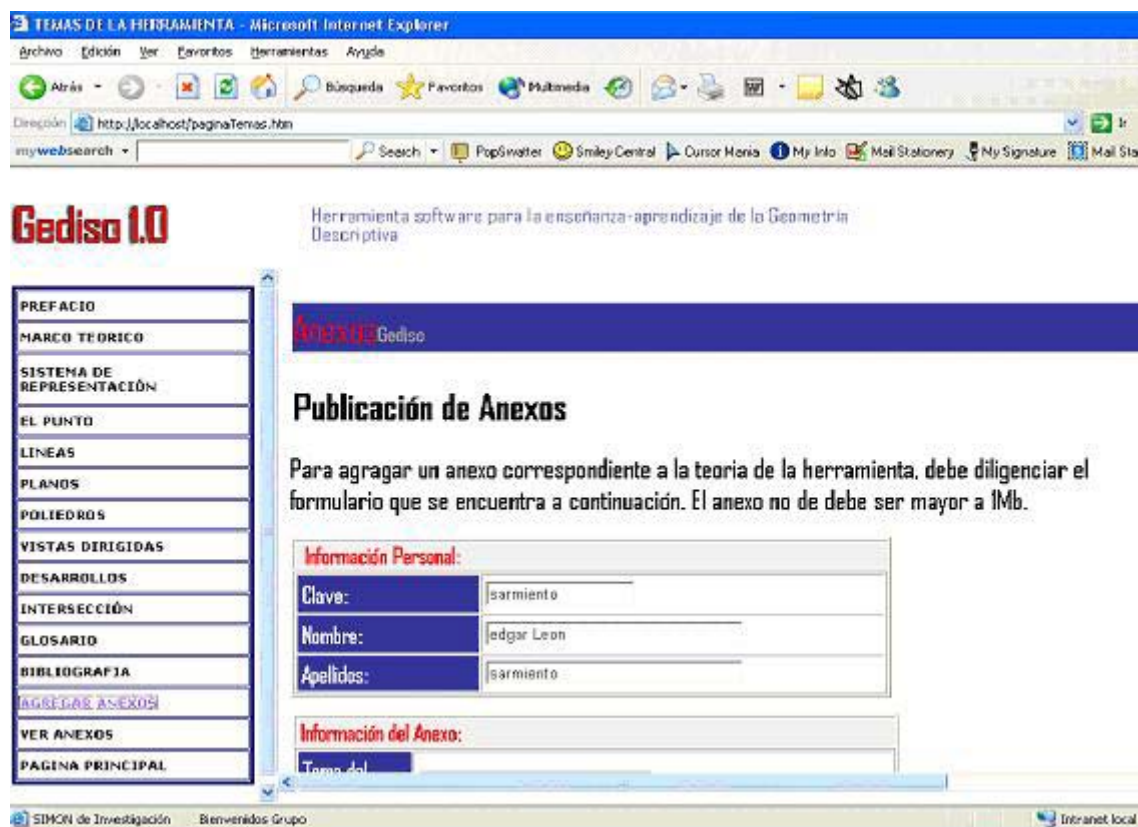


Figura 67. Interfaz que permite el ingreso de los anexos.

13.1.2 PRUEBAS

13.1.2.1 Casos de prueba

- ◆ Correcto almacenamiento de los anexos en la base de datos
- ◆ Conexión y selección correcta de la base de datos³⁵
- ◆ Evitar el ingreso a personal no autorizado

³⁵ Este caso de prueba es común a todos los subsistemas generales de aplicación, por ende solo se realizara prueba en este subsistema.

◆ Integridad transaccional

En esta parte solo se mostrará el caso de prueba: Correcto almacenamiento de los anexos en la base de datos.

13.1.2.2 Procedimiento de Prueba

El procedimiento para realizar esta prueba es bastante sencillo, por medio de la interfaz se incluye la información que permite anexar el documento, y después se procede a verificar en la base de datos que dicha información corresponda con las tablas y los campos correspondientes.

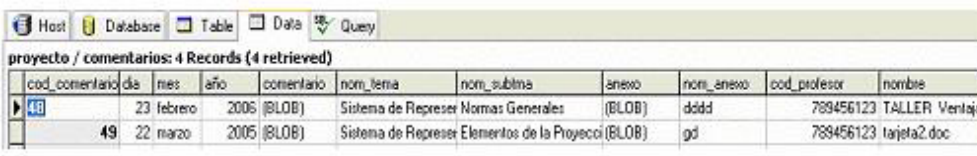
Datos Entrada	Estos datos corresponden al ente: Cód_comentario : 48 dia :23 mes :febrero año :2006 comentario :{Blob} nom_tema : Sistemas de Representación nom_subtema :Normas Generales anexo :{Blob} nom_anexo :dddd cod_profesor :789456123 nombre :TALLER Ventaja Corporativa.doc tipo :application/msword																																	
Registro en la base de datos	 <p>proyecto / comentarios: 4 Records (4 retrieved)</p> <table><tr><th>cod_comentario</th><th>dia</th><th>mes</th><th>año</th><th>comentario</th><th>nom_tema</th><th>nom_subtema</th><th>anexo</th><th>nom_anexo</th><th>cod_profesor</th><th>nombre</th></tr><tr><td>48</td><td>23</td><td>febrero</td><td>2006</td><td>(BLOB)</td><td>Sistema de Represer</td><td>Normas Generales</td><td>(BLOB)</td><td>dddd</td><td>789456123</td><td>TALLER Ventaja</td></tr><tr><td>49</td><td>22</td><td>marzo</td><td>2005</td><td>(BLOB)</td><td>Sistema de Represer</td><td>Elementos de la Proyecci</td><td>(BLOB)</td><td>gd</td><td>789456123</td><td>tarjeta2.doc</td></tr></table>	cod_comentario	dia	mes	año	comentario	nom_tema	nom_subtema	anexo	nom_anexo	cod_profesor	nombre	48	23	febrero	2006	(BLOB)	Sistema de Represer	Normas Generales	(BLOB)	dddd	789456123	TALLER Ventaja	49	22	marzo	2005	(BLOB)	Sistema de Represer	Elementos de la Proyecci	(BLOB)	gd	789456123	tarjeta2.doc
cod_comentario	dia	mes	año	comentario	nom_tema	nom_subtema	anexo	nom_anexo	cod_profesor	nombre																								
48	23	febrero	2006	(BLOB)	Sistema de Represer	Normas Generales	(BLOB)	dddd	789456123	TALLER Ventaja																								
49	22	marzo	2005	(BLOB)	Sistema de Represer	Elementos de la Proyecci	(BLOB)	gd	789456123	tarjeta2.doc																								

Tabla 45. Prueba para el ingreso de los anexos

Como se puede observar en la tabla 45, la información se ha almacenado correctamente en la tabla indicada.

13.2 SUBSISTEMAS DE EJERCICIOS

Este subsistema tiene como finalidad poner en práctica todos los conocimientos adquiridos por los estudiantes para resolver los diferentes problemas que se presenten. En ella los profesores podrán ver los diferentes problemas que la herramienta tiene.

Así mismo los estudiantes podrán acceder al formulario de trabajo que se realizó bajo Delphi por medio de un ejecutable, en estos se realizarán los diferentes ejercicios que están almacenados en la base de datos de la herramienta.

13.2.1 IMPLEMENTACIÓN

Este subsistema al igual que el anterior, se implementa en lenguaje PHP, para el manejo de la información (problemas) se utiliza la tabla, problemas, el código está embebido en las páginas, y la para la solución de los problemas se manejaron formularios en Delphi los cuales reportan una serie de datos que serán almacenados en la base de datos para que luego sean consultados por los profesores:

- ◆ Ver problemas.php.
- ◆ Ejercicios delphi form.

Para este subsistema se ha elegido presentar la implementación de los procesos utilizados para el desarrollo de los problemas que se encuentran almacenados en la base de datos de la herramienta. En la tabla 46, se puede observar un segmento de código en Delphi que permite realizar este proceso, teniendo en cuenta y validando que dicho proceso solo se puede realizar si el usuario está autorizado.

Código en Delphi	Descripción
<pre> public { Public declarations } Mapa:TMapa; Pintor1:TPintor; Linea1,LineaPara:Tlineas; Punto1:TPuntos; dx,dy:Integer; procedure CrearPintorYMapa(AnchoMap,AltoMap:Integer); procedure CambiarZoom(ZoomX,ZoomY:real); procedure GuardarIMagen(nombre:String); procedure CargarImagen(nombre:String); Function LineaPerpendicular(lineas:Tlineas;P:TPoint):Tlineas; Function LineaParalela(lineas:Tlineas;incx,incy:Integer):Tlineas; end; var FEvaluar: TFEvaluar; implementation uses Ejercicios, BaseDatos,DatosLinea; {\$R *.DFM} Function </pre>	<p>En esta sección de código se muestra como se realizan los ejercicios con ayuda de los diferentes componentes de dibujo que la herramienta tiene</p>

```

TFEvaluar.LineaPerpendicular(Lineas:TLineas;P:TPoint):TL
ineas;
var v,m,mpe,b1,b2,rx:real;
begin
  with Lineas do
  begin
    V:=(PosFin.x-PosIni.x);
    if V=0 then V:=0.000000000001;
    m:=(PosFin.y-PosIni.y)/V;
    b1:=PosIni.y-m*PosIni.x;
  end;
  with result do
  begin
    Lineas.Nombre:='Per';
    Lineas.PosIni.y:=P.y;
    Lineas.PosIni.x:=P.x;
    if m <> 0 then
    begin
      mpe:=-1/m;
      b2:=P.y-mpe*P.x;
      rx:=(b2-b1)/(m-mpe);
      Lineas.PosFin.x:=round(rx);
      Lineas.PosFin.y:=round(mpe*rx+b2);
    end
    else
    begin
      lineas.PosFin.x:=p.x;
      Lineas.PosFin.y:=Lineas.PosIni.Y;
    end;
  end;
  result:=Lineas;
end;

Function
TFEvaluar.LineaParalela(Lineas:TLineas;incx,incy:Integer):
TLineas;
begin
  Lineas.PosIni.x:=Lineas.PosIni.x + incx;
  Lineas.PosIni.y:=Lineas.PosIni.y + incY;
  Lineas.PosFin.x:=Lineas.PosFin.x + incx;
  Lineas.PosFin.y:=Lineas.PosFin.y + incY;
  result:=Lineas;
end;

procedure
TFEvaluar.CrearPintorYMapa(AnchoMap,AltoMap:Integer);
begin
  panel4.Width:=AnchoMap;
  panel4.Height:=AltoMap;
  Mapa:=Tmapa.create(AnchoMap,AltoMap,Caption);
  Pintor1:=TPintor.Create(PaintBox1.Canvas,Mapa);
end;

Procedure TFEvaluar.CambiarZoom(ZoomX,ZoomY:real);
begin
  Pintor1.CambiarEscala (ZoomX,ZoomY);
  panel4.Height:=round(mapa.MaxY);
  panel4.Width:=round(Mapa.MaxX);
  Pintor1.DibujarMapa;
end;

Procedure TFEvaluar.GuardarIMagen(nombre:String);
var ElNombre:String;
begin
  ElNombre:=nombre;
  ImagenParaGuardar:=TImage.Create(nil);
  ImagenParaGuardar.Height:=PaintBox1.Height;

```

<pre> ImagenParaGuardar.Width:=PaintBox1.Width; try Pintor1.Canvas:=ImagenParaGuardar.Canvas; pintor1.DibujarMapa; ImagenparaGuardar.Picture.SaveToFile(EINombre); finally ImagenParaGuardar.Free; Pintor1.Canvas:= PaintBox1.Canvas; end; end; Procedure TFEvaluar.CargarImagen(nombre:String); var Destino,Origen:TRect; EINombre:String; begin ELNombre:=nombre; try ImagenParaCargar:=TImage.Create(nil); ImagenParaCargar.Height:=PaintBox1.Height; ImagenParaCargar.Width:=PaintBox1.Width; Destino:=Rect(0,0,PaintBox1.Width,Paintbox1.Height); Origen:=Rect(0,0,IMagenParaCargar.Width,IMagenParaCar gar.Height); ImagenParaCargar.Picture.LoadFromFile(ELNombre); PaintBox1.Canvas.CopyRect(Destino,ImagenParaCargar.C anvas,Origen); finally ImagenParaCargar.Free; end; end; </pre>	
---	--

Tabla 46. Código en Delphi para solucionar problemas por parte de los estudiantes.

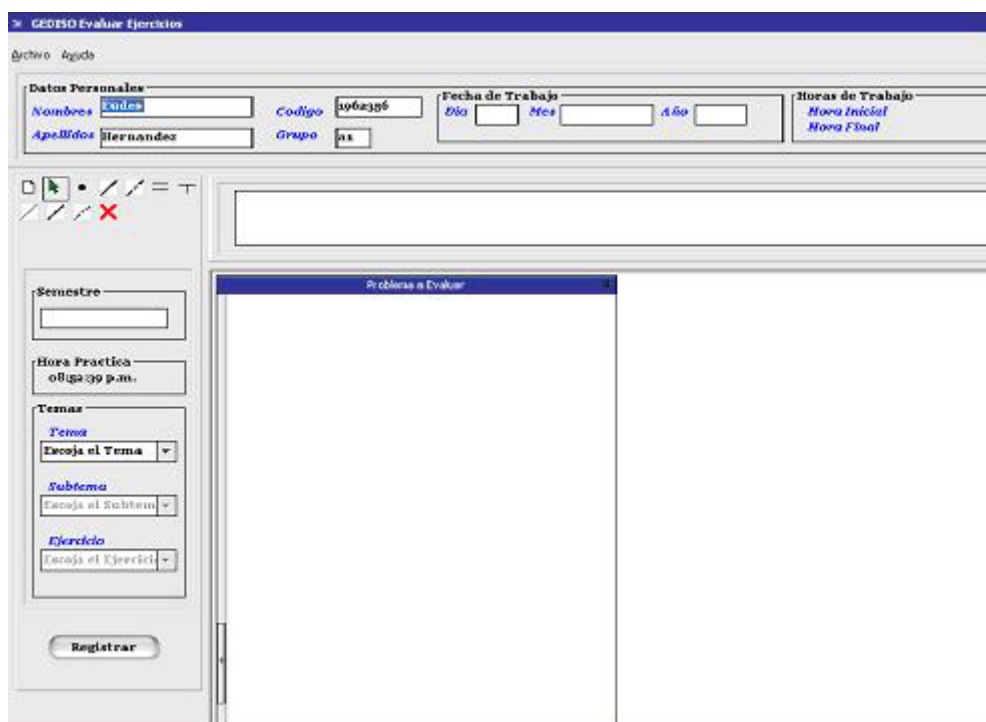


Figura 68. Interfaz que permite realizar los problemas.

13.2.2 PRUEBAS

13.2.2.1 Casos de Prueba

- ◆ Correcto almacenamiento de información en la base de datos
- ◆ Conexión y selección correcta de la base de datos³⁶
- ◆ Evitar ambigüedad en los datos
- ◆ Integridad transaccional

13.2.2.2 Procedimiento de Prueba

En este subsistema se muestra el caso de prueba: Evitar ambigüedad en los datos, el cual se puede observar en la tabla 47. Existe una situación en la que se debe tener especial cuidado, es cuando un problema es solucionado al tiempo por dos o mas estudiantes, el proceso debe asegurar que los datos queden bien almacenados y con el usuario correcto y puedan se visualizados por los profesores. Para realizar la prueba, se parte de un problema el cual es cargado de la base de datos. Se espera que inmediatamente sea terminado el problema se almacene los datos en la tablas practicas e historial.

Datos Entrada

³⁶ Este caso de prueba es común a todos los subsistemas generales de aplicación, por ende solo se realizara prueba en este subsistema.



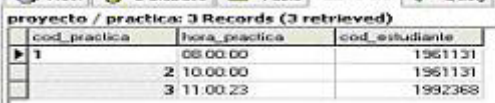
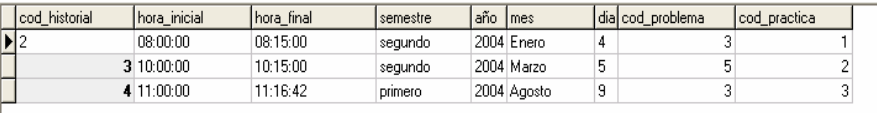
Registro en la base de datos	Registros antes de solucionar el problema
	
	
	Registros después de haber solucionado el problema
	
	

Tabla 47. Prueba del almacenamientos de los datos de las practicas al solucionar los ejercicios por parte de los estudiantes

13.3 SUBSISTEMA DE REGISTROS

Por medio de este subsistema, los profesores que dictan la asignatura Geometría Descriptiva I y II tienen la capacidad de realizar un seguimiento de las actividades realizadas por los estudiantes en el subsistema de problemas. Adicionalmente permitirá a los profesores llevar un control de los progresos de los estudiantes a través del curso. Este subsistema permite también la consulta de las evaluaciones realizadas por parte de los estudiantes según el estado en el que se encuentren (sin revisar, revisado), los profesores tendrían acceso a las mismas y asignar una nota por el trabajo realizado por el estudiante si así lo desean.

13.3.1 IMPLEMENTACIÓN

Para implementar este subsistema se ha construido en lenguaje PHP, los procesos que permiten cumplir con las funciones dadas a este subsistema. Estos procesos involucran las tablas de historial, evaluaciones.

Este subsistema emplea los subsistemas de sesiones, interfaz y de publicación para realizar todas sus actividades y las páginas que maneja son las siguientes:

- ◆ ver registros.php.
- ◆ ver evaluaciones.php

Se ha elegido que la herramienta permita presentar los registros de las horas de trabajo de los estudiantes. En la tabla 48. se encuentra un segmento de código php que permite ver los registros por estudiantes (en la tabla: *historial*), la información relacionada con las practicas asignadas a los grupos de trabajo; por ejemplo: código del problema, hora de inicio, hora final, día, mes, practica. Adicionalmente, en la figura 69. se encuentra la página que permite ver la información anteriormente mencionada.

Código PHP	Descripción
<pre> <?php \$txttestudi=\$_POST["txttestudi"]; \$practica="select * from practica where cod_usuario='\$txttestudi'"; \$practi=mysql_query(\$practica,\$link); \$pract=mysql_fetch_array(\$practi); \$codpract=\$pract["cod_practica"]; if ((!\$pract)) { echo("<SCRIPT language=JavaScript>alert('Error!, El codigo del estudiante es incorrecto o no ha realizado practicass. '); </script>"); print("<SCRIPT language=JavaScript>window.location=\"ver registros.php\" </script>"); } \$codpract=\$pract["cod_practica"]; if (\$codpract!=" ") { \$sql= "select cod_practica from practica where cod_estudiante='\$codpract'"; \$result=mysql_query(\$sql,\$link); while (\$roww=mysql_fetch_array(\$result)) { \$perusua=(\$roww["cod_practica"]); \$congrupo= "SELECT * from historial where cod_practica='\$perusua'"; \$congrup= mysql_query(\$congrupo,\$link); while (\$row=mysql_fetch_array(\$congrup)) { echo '<tr><td>'. \$row["cod_problema"]. '</td>'; echo '<td>'. \$row["hora_inicial"]. '</td>'; echo '<td>'. \$row["hora_final"]. '</td>'; echo '<td>'. \$row["dia"]. '</td>'; echo '<td>'. \$row["mes"]. '</td>'; echo '<td>'. \$row["cod_practica"]. '</td></tr>'; } } } } ?> </pre>	<p>Una vez se ha hecho click en el botón de "buscar" cuyo nombre es btnbuscar, se realiza una consulta que permite buscar el código del problema que el estudiante soluciono o trabajo, la hora de inicio, hora final de este, el día, el mes y el código de la practica, todo esto se hace con una búsqueda a la base de datos en las tablas practicas e historial, se verifica que el estudiantes al cual se le consulta su trabajo exista realizando la búsqueda satisfactoriamente. Es de aclarar que la conexión a la base de datos se ha hecho en otra sección de la página para efectos de poder utilizarla en otras validaciones.</p>

Tabla 48. Código PHP que permite la visualización del historial de trabajo de los estudiantes



Figura 69. Interfaz que permite la visualización del historial

13.3.2 PRUEBAS

13.3.2.1 Casos de prueba

- ◆ Correcto almacenamiento de información en la base de datos
- ◆ Conexión y selección correcta de la base de datos³⁷
- ◆ Evitar ambigüedad en los datos
- ◆ Integridad transaccional

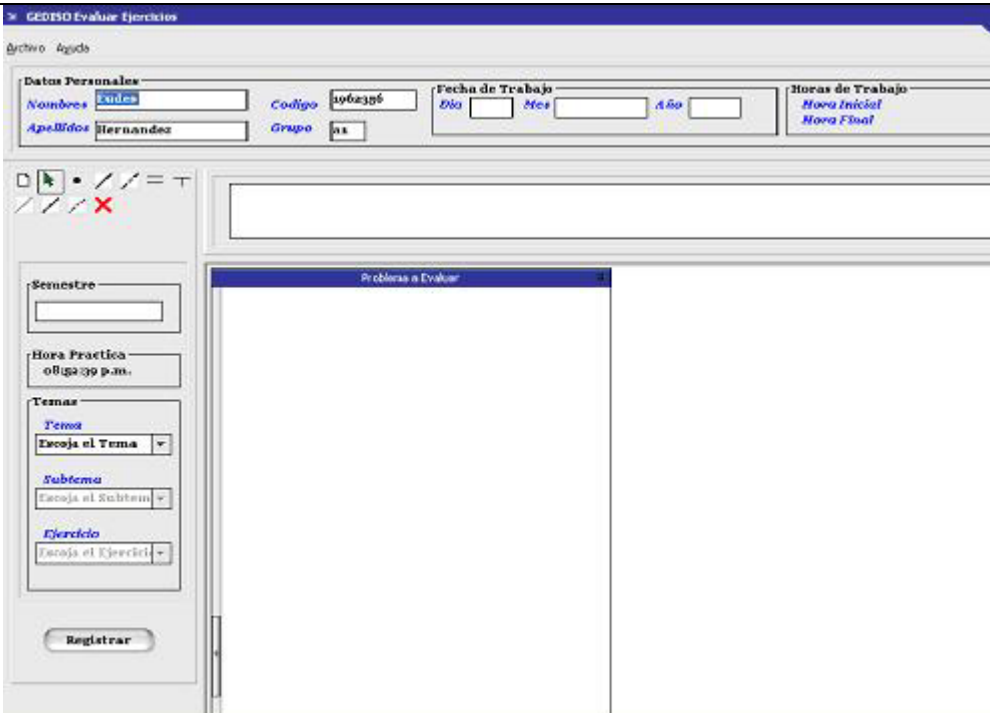
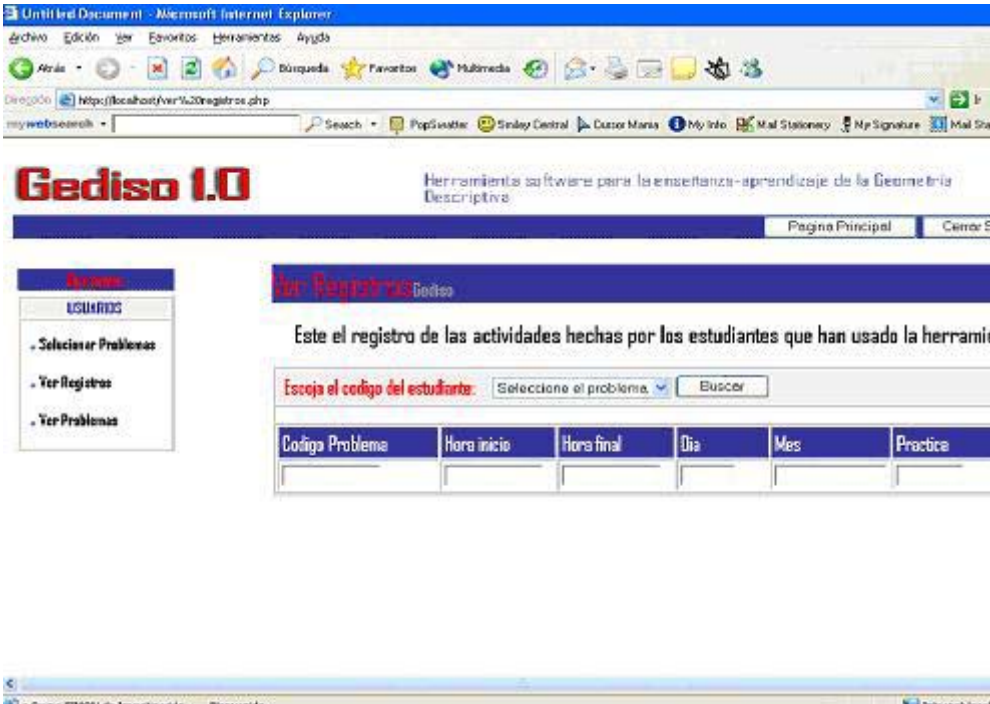
13.3.2.2 Procedimiento de prueba

En este subsistema se muestra el caso de prueba: Mantener la Integridad Transaccional, el cual se puede observar en la tabla 49. Para observar este proceso se muestra la tabla historial y el formulario de práctica antes de ingresar los registros.

Posteriormente los datos se ingresan a la base de datos después de haber seleccionado un nuevo ejercicio, en dicho formulario se pide seleccionar el tipo de

³⁷ Este caso de prueba es común a todos los subsistemas generales de aplicación, por ende solo se realizara prueba en este subsistema.

problema, el tema y el subtema a que corresponden, se verifican los datos y se muestra el problema a solucionar en la hoja de dibujo para que pueda ser trabajado, una vez terminado el problema se escogerá otro, automáticamente se registran una serie de datos en las tablas historial y practicas correspondientes a las actividades realizadas por el estudiante, dicha actualización se muestra igualmente en la tabla 49.

<p>Datos Entrada</p>	
<p>Registro en la base de datos</p>	<p>Registros antes de efectuar cambios</p> 

Registros después de efectuar cambios

The screenshot shows a web browser window displaying the 'Gediso 1.0' application. The page title is 'Herramienta software para la enseñanza-aprendizaje de la Geometría Descriptiva'. The main content area is titled 'Ver Registros Gediso' and contains the text 'Este es el registro de las actividades hechas por los estudiantes que han usado la herrami'. Below this text is a search bar with the label 'Escriba el código del estudiante:' and a dropdown menu showing '1962356'. To the right of the search bar is a 'Buscar' button. Below the search bar is a table with the following data:

Código Problema	Hora inicio	Hora final	Día	Mes	Practica
8	02:17:59 p.m.	02:18:00 p.m.	1	Juño	30
8	09:46:38 p.m.	09:47:47 p.m.	5	Juño	67
8	09:46:38 p.m.	09:47:48 p.m.	5	Juño	67
8	09:46:38 p.m.	09:47:49 p.m.	5	Juño	67
8	09:46:38 p.m.	09:47:49 p.m.	5	Juño	67
8	09:48:04 p.m.	09:48:05 p.m.	5	Juño	68
6	10:01:12 p.m.	10:01:14 p.m.	5	Juño	71
8	10:02:04 p.m.	10:02:06 p.m.	5	Juño	72

Tabla 49. Prueba para la inserción de los registros de trabajo, desde el modulo de problemas.

13.4 SUBSISTEMA DE EVALUACION

Esta zona se encarga de realizar las evaluaciones a los diferentes estudiantes, la herramienta software permitirá los profesores ver todas las evaluaciones que los estudiantes realicen, revisaran las evaluaciones pendiente, para luego cambiarlas a revisados entregando una nota o un comentario por el trabajo realizado por parte de los estudiantes. Así mismo los estudiantes podrán acceder a al modulo donde se ejecutan las evaluaciones para revisar las notas de las evaluaciones.

Esta zona esta permitida únicamente para los profesores, estudiantes y administrador, el administrador podrá realizar labores de mantenimiento sobre los problemas que contendrá la herramienta, pero no sobre las evaluaciones y las notas de cada estudiante.

13.4.1 IMPLEMENTACIÓN

Para implementar este subsistema se ha construido en lenguaje PHP y DELPHI los procesos que permiten cumplir con las funciones dadas a este. Estos procesos involucran las tablas de problemas, estudiante, nota_evaluación.

En la ejecución de este se emplea los subsistemas de sesiones e interfaz para realizar todas sus actividades, las páginas y formularios que maneja son las siguientes:

- ◆ Ver evaluaciones.php
- ◆ Asignar notas.php
- ◆ Ver notas.php
- ◆ Evaluar delphi form.
- ◆ Problemas teoricos.php

Para este subsistema se ha elegido presentar la implementación de los procesos utilizados para asignar las notas a los diferentes estudiantes. En la tabla 50. se puede observar un segmento de php que permite realizar este proceso, teniendo en cuenta y validando que dicho proceso solo se puede realizar si el usuario esta autorizado. Primero se realiza una búsqueda en la tabla evaluaciones, mostrando lo realizado por el estudiante y luego establecer una nota por su trabajo.

Código PHP	Descripción
<pre><?php session_start (); session_start("clave","contraseña","usertipo"); include("conec.php"); // Incluye la función que permite la conexión a la base de datos \$link=Conectarse(); \$inicioseccion=\$_SESSION['inicioseccion']; if (\$inicioseccion==0) { print("<SCRIPT language=JavaScript>window.location=\"PAGINAINICIO.php\" </script>"); } \$usertipo=\$_SESSION['cod_usuario']; if (\$usertipo ==4 and \$usertipo==3) { echo("<SCRIPT language=JavaScript>alert('No tiene permiso a esta pagina.');</pre>	<p>Con esta sección de código se pretende dar una idea de cómo se realiza el proceso de asignación de notas por parte de los profesores a las evaluaciones realizadas por los estudiantes. En la página de asignotas.php, se llena el formulario escogiendo a un estudiante y el profesor le asigna una nota de acuerdo a su evaluación.</p>

<pre> \$txtnotas=\$_POST['txtnotas']; \$text_evaluacion=\$_POST['text_evaluacion']; \$textcomentarios=\$_POST['textcomentarios']; \$txttestudiante=\$_POST['txttestudiante']; if(\$txtnotas!="") { \$insernotas= "insert into nota_evaluacion (evaluacion, nota_evaluacion, cod_evaluacion, comentarios, cod_estudiante) values ('\$evaluacion','\$txtnotas','\$text_evaluacion','\$textcomentarios','\$txttestu diante')"; \$inserestu=mysql_query(\$insernotas,\$link); echo("<SCRIPT language=JavaScript>alert('La nota se paso satisfactoriamente.'); </pre>	
--	--

Tabla 50. Código PHP que permite asignar la nota a los estudiantes

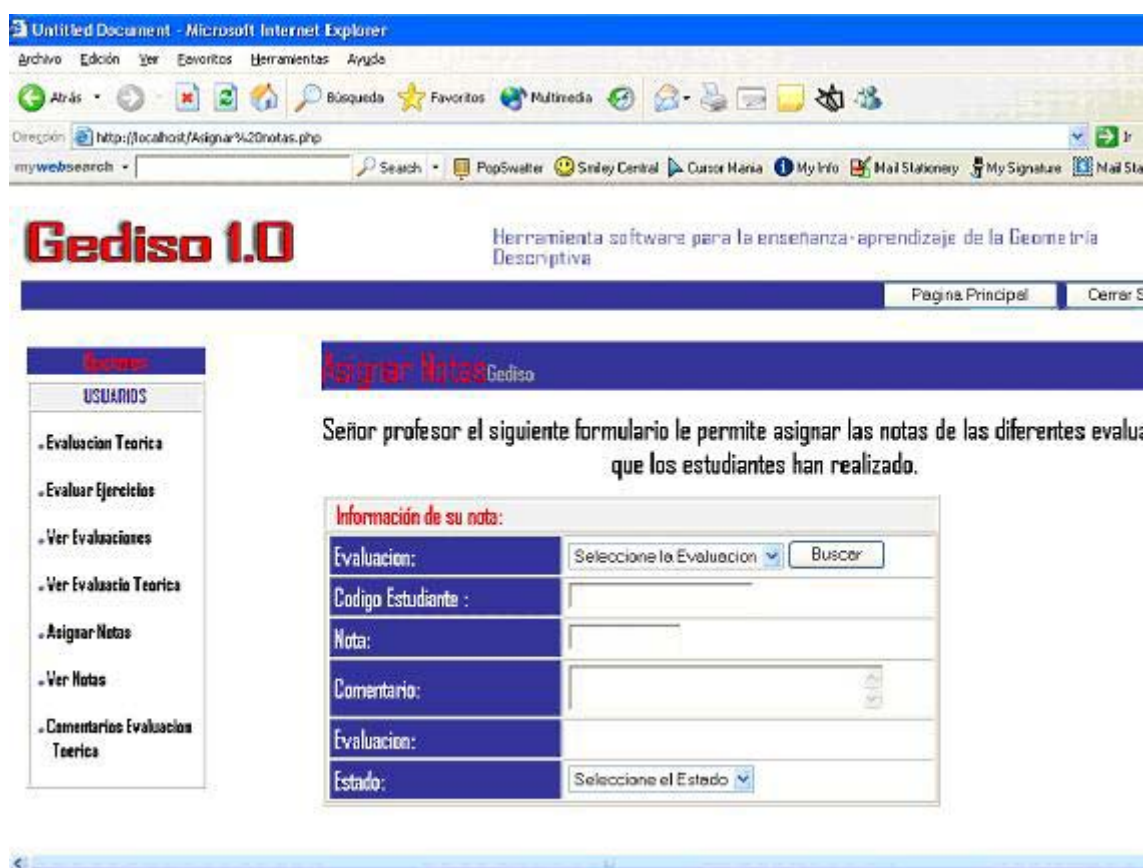


Figura 70. Interfaz que permite asignar la nota a los estudiantes

13.4.2 PRUEBAS

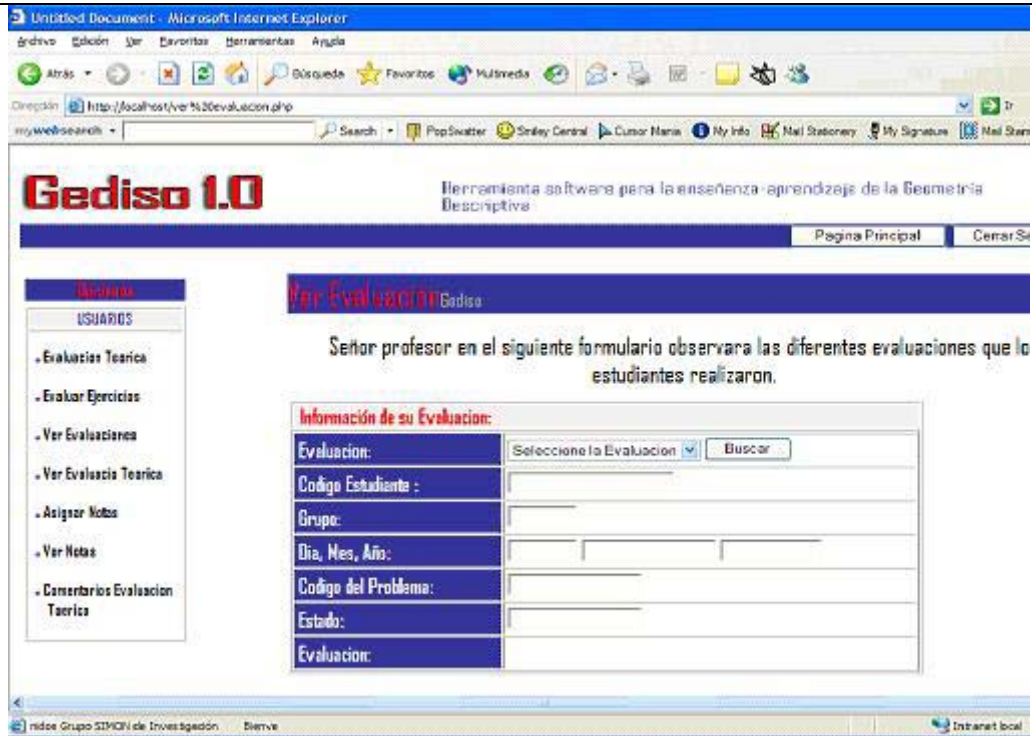
13.4.2.1 Casos de prueba

- ◆ Correcto almacenamiento de los anexos en la base de datos
- ◆ Conexión y selección correcta de la base de datos³⁸
- ◆ Evitar el ingreso a personal no autorizado
- ◆ Integridad transaccional

En esta parte se muestra el caso de prueba Congruencia entre los criterios de búsqueda y los resultados de la misma.

13.4.2.2 Procedimiento de Prueba

Como en todos los subsistemas generales de aplicación, la información (en este caso las diferentes evaluaciones) proviene de la interfaz, una vez tomada esta información se procede a realizar la búsqueda de los diferentes documentos que cumplen con las condiciones dadas.

Datos Entrada	
------------------	--

³⁸ Este caso de prueba es común a todos los subsistemas generales de aplicación, por ende solo se realizara prueba en este subsistema.

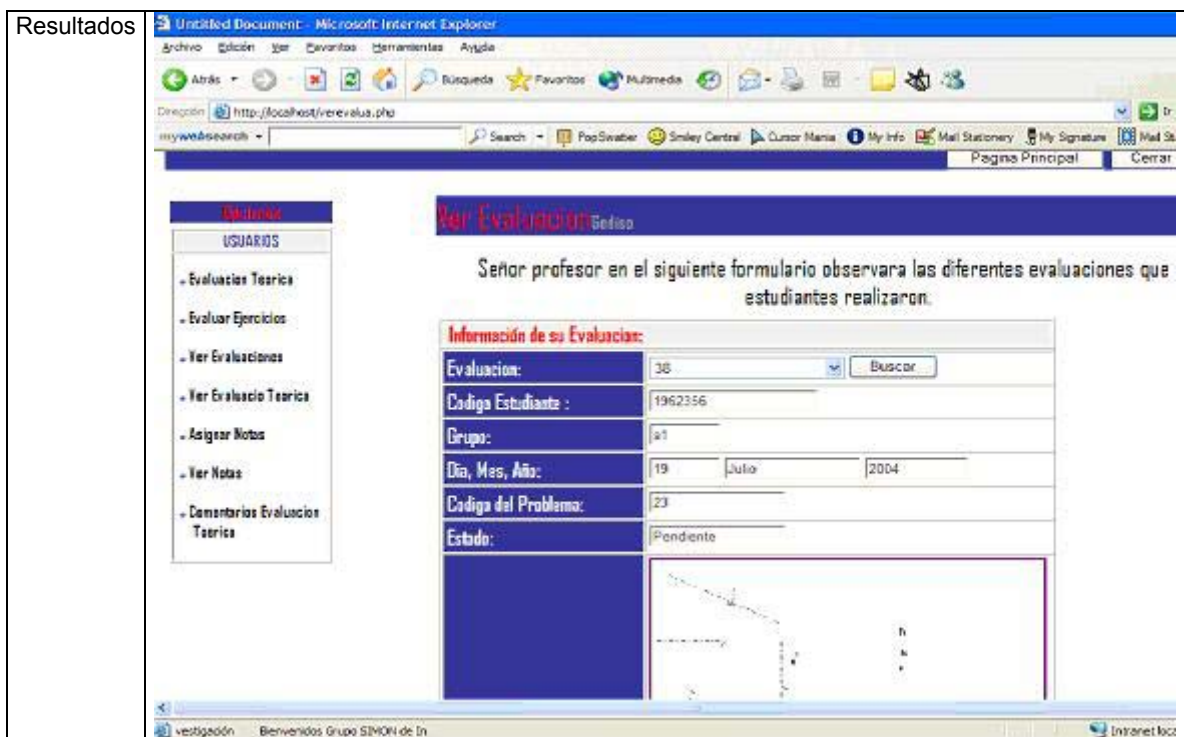


Tabla 51. Prueba para buscar la evaluación de un estudiante

La prueba ha sido satisfactoria, dado que los resultados que arrojo la búsqueda son consistentes con los criterios elegidos.

13.5 SUBSISTEMA DE ADMINISTRACIÓN

Este subsistema se encarga del manejo de usuarios, grupos. Además a través de este subsistema se realiza el ingreso, borrado, cambio de estado de los problemas, también el borrado del historial de trabajo que elimina todos los registros de las todas las practicas que se hicieron en el semestre y la eliminación de todas las evaluaciones que se trabajaron con el fin de mantener depurada la base de datos de información innecesaria. También cuenta con la opción de realizar una copia de seguridad de los diferentes registros que se borran de la base de datos.

La información sobre los anexos de cada profesor será mantenida en la base de datos, dado que son de utilidad para actualizar la información que la herramienta ya posee.

13.5.1 IMPLEMENTACIÓN

Este subsistema al igual que los anteriores, se implementa en lenguaje PHP, para el manejo de la información en las tablas involucradas, tales como usuarios, grupos, problemas, estudiantes, profesores, el código está embebido en las páginas:

- ◆ Crear usuario.php
- ◆ Crear grupos.php
- ◆ Eliminar usuarios.php
- ◆ Eliminar grupos.php
- ◆ Listar usuarios.php
- ◆ Listar grupos.php
- ◆ Modificar usuarios.php
- ◆ Modificar grupos.php
- ◆ Eliminar historial.php
- ◆ Eliminar registro evaluaciones.php
- ◆ Insertar problemas.php

Se ha elegido presentar la implementación de los procesos para crear grupo. En la tabla 52. se encuentra un segmento de php que permite la inclusión (en la tabla: *grupos*), de la información relacionada con los grupos de trabajo, por ejemplo, grupo, materia, código del profesor, semestre, año, horario del curso, día. Adicionalmente, en la figura 71. se encuentra la página que permite el ingreso de la información antes mencionada.

Código PHP	Descripción
<pre><?php // Validaciones de los datos ingresados \$consul = "SELECT cod_grupo FROM grupo WHERE cod_grupo='\$txtcod_grupo'"; \$query=mysql_query(\$consul,\$link); if (mysql_fetch_array(\$query)) { echo("<SCRIPT language=JavaScript>alert('Error!, Ya existe un grupo con ese codigo.');</pre>	<p>Una vez se ha hecho click en el botón de “crear grupo” cuyo nombre es btncrear, se realiza una consulta que revisa que no exista otro grupo con ese código, además buscar el código del profesor exista y luego se hace la inserción de los datos en la tabla grupos, se verifica que se haya creado el grupo satisfactoriamente. Es de aclarar que la conexión a la base de datos se ha hecho en otra sección de la página para efectos de poder utilizarla en otras validaciones.</p>

```

$tipoa = "SELECT * FROM profesor WHERE
cod_profesor ='$txtcod_profesor'";
$tipo = mysql_query($tipoa,$link);
if (!mysql_fetch_array($tipo))
{
echo("<SCRIPT language=JavaScript>alert('Error!, No
existe un profesor con ese codigo. '); </script>");
$_SESSION['txtcod_grupo']=$txtcod_grupo;
$_SESSION['txtcod_profesor']=$txtcod_profesor;
$_SESSION['txtaño']=$txtaño;
$_SESSION['txthora']=$txthora;
$_SESSION['txtdia']=$txtdia;
$_SESSION['creargrupo']=1;
print("<SCRIPT
language=JavaScript>window.location=\\"Crear
Grupos.php\\" </script>");

//
print("No existe un profesor con ese codigo");
}
else {
$cons = "SELECT cod_materia FROM materias
WHERE nom_materia ='$cbo_materia'";
$query= mysql_query($cons,$link);
$conss=mysql_fetch_array($query);
$codmate=($conss["cod_materia"]);
$insertar      ="INSERT INTO grupo
(cod_grupo,año,semestre,hora,dia,cod_materia,cod_pr
ofesor) values
('$txtcod_grupo','$txtaño','$cbo_semestre','$txthora','$tx
tdia','$codmate','$txtcod_profesor')";
$insert = mysql_query($insertar,$link);
$insertcodigo = "insert into estudiante (cod_grupo)
values ('$txtcod_grupo')";
$insertco =mysql_query($insertcodigo,$link);
//}
if (!$insert && !$insertco)
{
echo("<SCRIPT language=JavaScript>alert('Error en la
inserción. '); </script>");
$_SESSION['txtcod_grupo']=$txtcod_grupo;
$_SESSION['txtcod_profesor']=$txtcod_profesor;
$_SESSION['txtaño']=$txtaño;
$_SESSION['txthora']=$txthora;
$_SESSION['txtdia']=$txtdia;
$_SESSION['creargrupo']=1;
print("<SCRIPT
language=JavaScript>window.location=\\"Crear
Grupos.php\\" </script>");
//
print("Error en la inserción");
}
else
{
echo("<SCRIPT language=JavaScript>alert('El Grupo
se ha creado satisfactoriamente. '); </script>");
$_SESSION['txtcod_grupo']=' ';
$_SESSION['txtcod_profesor']=' ';
$_SESSION['txtaño']=' ';
$_SESSION['txthora']=' ';
$_SESSION['txtdia']=' ';
$_SESSION['creargrupo']=0;
print("<SCRIPT
language=JavaScript>window.location=\\"Crear
Grupos.php\\" </script>");
//print("El Grupo se ha creado satisfactoriamente");
}
}

```

<pre> } } // Cerrar conexion mysql_close(\$link); ?> </pre>	
---	--

Tabla 52. Código PHP que permite la creación de un grupo de trabajo

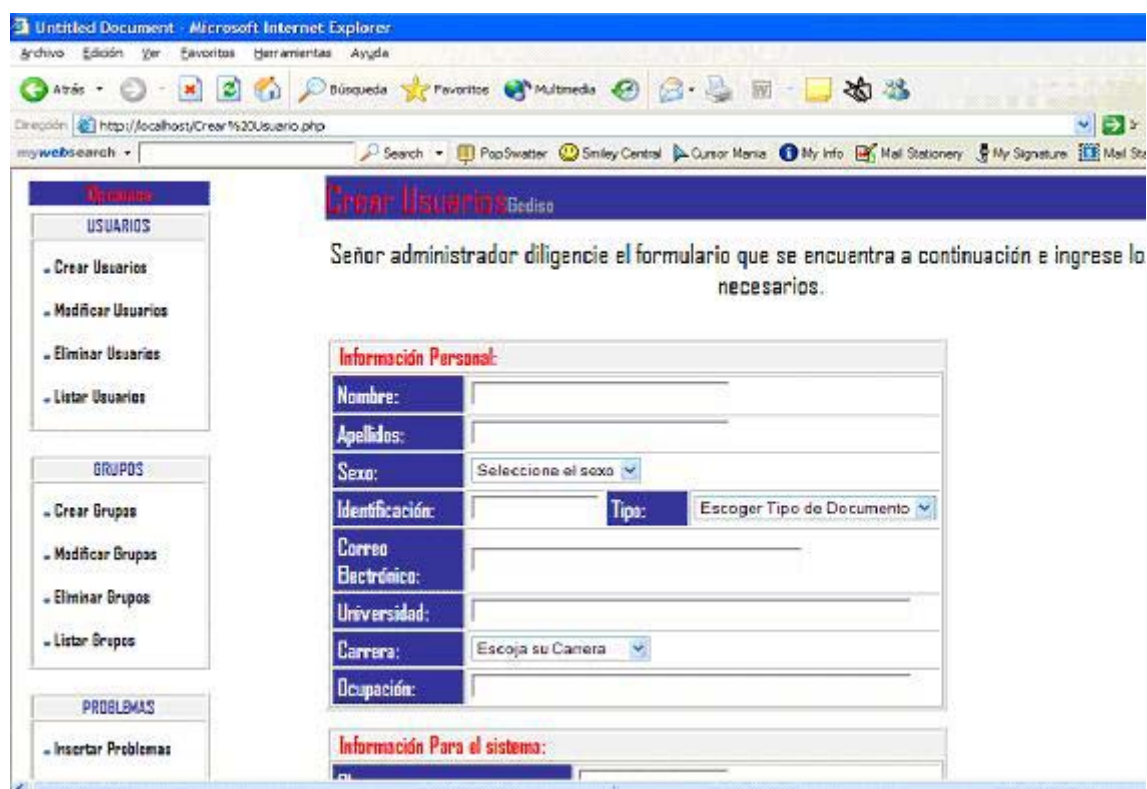


Figura 71. Interfaz que permite la creación de un grupo de trabajo

CAPÍTULO 14.

EVALUACION

La evaluación de esta fase se realiza con base en los resultados de las pruebas y en la revisión final del modelo de casos de usos que permite comprobar que el producto ha alcanzado su capacidad operativa inicial. Por lo tanto, el principal producto que se obtiene de esta fase es la herramienta software ejecutable.

La vista de la arquitectura durante esta fase tiene pocas variaciones ya que se definió desde la fase de elaboración y en la construcción sólo se realizaron leves modificaciones y actualizaciones a la misma.

14.1 RESUMEN CAPÍTULO

En este capítulo se presentaron los subsistemas que permiten, el buen funcionamiento de la Herramienta software, así como el manejo posterior de la información normativa. Para cada subsistema se realizó la implementación y las pruebas de los procesos involucrados en el funcionamiento de los mismos.

Debido a que no se desea ser repetitivos en el documento, solo se presentaron segmentos de código de una funcionalidad por cada subsistema y un caso de prueba a la misma, las demás pruebas se puede encontrar en el anexo E, Escenarios de prueba, la aplicación se puede encontrar en el server donde esta instalado el software y la documentación completa en el grupo de investigación SIMON.

14.2 DESARROLLO DEL MODELO DE CASOS DE USO

En esta fase se completó el modelo de casos de uso, el cual se encuentra en el Anexo C – Modelo de Casos de Uso de la Herramienta, pagina 208. Sin embargo para continuar con el esquema de la fase anterior y a modo de resumen se presentará la siguiente tabla que contiene todos los casos de uso identificados durante el desarrollo de este proyecto y la fase en que fueron identificados, descritos, analizados e implementados.

Casos de Uso Identificados	Identificado	Descrito	Analizado	Diseñados e implementados	¿Se comprende completamente?
Verificar Registro del Usuario.	Inicio	Inicio		Construcción	Si
Escoger Tema	Inicio	Inicio		Construcción	Si
Escoger Subtema	Elaboración	Elaboración	Elaboración	Construcción	Si
Consultar Ejemplos	Inicio	Inicio	Elaboración	Construcción	Si
Agregar Anexos	Elaboración	Elaboración		Construcción	Si
Ver Anexos	Elaboración	Elaboración		Construcción	Si
Verificar Registro del Usuario.	Elaboración	Elaboración	Elaboración	Construcción	Si
Escoger Tema Ejercicio	Inicio	Inicio	Inicio	Construcción	Si
Escoger Subtema Ejercicio	Elaboración	Elaboración		Construcción	Si
Escoger Ejercicio	Inicio	Inicio		Construcción	Si
Ver Problemas	Elaboración	Elaboración	Elaboración	Construcción	Si
Ver registros por parte de los profesores.	Inicio	Inicio	Inicio - Elaboración	Construcción	Si
Verificar Registro del Usuario.	Elaboración	Elaboración		Construcción	Si
Escoger Tema Evaluación	Inicio	Inicio		Construcción	Si
Escoger Subtema Evaluación	Elaboración	Elaboración		Construcción	Si
Escoger Ejercicio Evaluar	Inicio	Inicio		Construcción	Si
Enviar Ejercicio para calificación	Elaboración	Elaboración		Construcción	Si
Ver Nota de evaluaciones por parte de los estudiantes.	Elaboración	Elaboración		Construcción	Si
Crear Informes por parte de los profesores.	Inicio	Inicio	Inicio - Elaboración	No desarrollado	Si
Verificar Registro del	Elaboración	Elaboración		Construcción	Si

Administrador.					
Crear Grupos.	Inicio	Inicio	Elaboración	Construcción	Si
Modificar Grupos.	Inicio	Inicio	Elaboración	Construcción	Si
Eliminar Grupos.	Inicio	Inicio	Elaboración	Construcción	Si
Listar Grupos.	Elaboración	Elaboración	Elaboración	Construcción	Si
Imprimir Lista Grupos	Elaboración	Elaboración	Elaboración	Construcción	Si
Crear usuario	Elaboración	Elaboración	Elaboración	Construcción	Si
Modificar usuario	Elaboración	Elaboración	Elaboración	Construcción	Si
Borrar usuario	Elaboración	Elaboración	Elaboración	Construcción	Si
Listar Usuarios	Elaboración	Elaboración	Elaboración	Construcción	Si
Imprimir Lista Usuarios	Elaboración	Elaboración	Elaboración	Construcción	Si
Insertar los Ejercicios en la base de datos.	Inicio	Inicio	Elaboración	Construcción	Si
Sacar los Ejercicios de la base de datos.	Inicio	Inicio	Elaboración	Construcción	Si
Cambiar Estado de los problemas	Elaboración	Elaboración	Elaboración	Construcción	Si
Copia de seguridad del historial	Elaboración	Elaboración	Elaboración	Construcción	Si
Eliminar el historial creado en el semestre.	Elaboración	Elaboración	Elaboración	Construcción	No
Copia de seguridad del registro de evaluaciones	Elaboración	Elaboración	Elaboración	Construcción	Si
Eliminar registro de evaluaciones	Elaboración	Elaboración	Elaboración	Construcción	No
Ver mensajes	Elaboración	Elaboración	Elaboración	Construcción	Si
Copia de Seguridad	Elaboración	Elaboración	Elaboración	Construcción	Si

Tabla 53. Casos de Uso de la herramienta GEDISO 1.0

CONCLUSIONES

El presente trabajo de grado ha sido un proceso de aprendizaje y enriquecimiento personal, del cual se puede concluir que:

- Aunque la investigación se realizó en la Escuela de Diseño Industrial de la UIS, se observó que el manejo de herramientas software que ayuden o presten soporte al proceso de enseñanza-aprendizaje en esta escuela se reduce a la utilización de programas de Diseño Asistido por Computador (CAD), los cuales son difíciles de manejar y no se complementan con una ayuda teórica fuerte. Adicionalmente, este es un campo poco explorado en el proceso educacional de nuestro país, por esto se proponen futuros desarrollos en el campo y así ser líderes en el Diseño, Implementación y Promoción de este tipo de software que contribuyan a la dinámica y mejoramiento de dicho proceso.
- El proceso Unificado de Desarrollo de Software, es una metodología apropiada para obtener un software de calidad dentro de los plazos de tiempo y presupuesto determinados. Sin embargo, hay que recordar que el seguimiento de una metodología no implica realizar una secuencia de pasos única que garantice el éxito del proyecto, se debe apropiarse la metodología al caso particular. Esto se observó durante el presente trabajo de grado, en el cual, la herramienta se construyó progresivamente incluyendo requisitos, complementando los modelos, implementando y probando las facilidades, que dieron como resultado a Gediso 1.0.
- Se debe incentivar más el proceso de investigación para detectar áreas en las que la aplicación de la Ingeniería de Sistemas, no sólo brinde apoyo para optimizar procesos, sino promueva cambios metodológicos que aporten al desarrollo Institucional y organizacional.
- El desarrollo centrado en casos de uso proporciona beneficios al proyecto, ya que se encargan de enlazar todas las actividades y dirigir el proceso de desarrollo a través de sus flujos y sus fases. Además, proporciona un medio sistemático e intuitivo para capturar los requisitos centrándose en el valor añadido que éstos tienen para cada usuario del sistema.

- No se pretende que la aplicación de la herramienta desarrollada genere de inmediato un cambio en la cultura del proceso de enseñanza- aprendizaje, pero sí, que se den los primeros pasos hacia un proceso que facilite el aprendizaje e involucre a toda la comunidad universitaria en la necesidad de la utilización de este tipo de herramientas software.
- La construcción de cada subsistema se realizó a través de iteraciones incrementales, esto hizo evidente el seguimiento de estándares de implementación para el diseño de la interfaz, para la documentación del código y para la organización de los subsistemas en el servidor, facilita la integración de un nuevo subsistema a la herramienta actual, así como la reutilización de código y funciones que pertenecen a los subsistemas existentes. Con la integración de las anteriores actividades se obtiene un software de fácil mantenimiento y que evoluciona a través del tiempo.
- Un aspecto para destacar del presente trabajo de grado es su adaptabilidad a cualquier Universidad que posean en su pénsum la asignatura Geometría Descriptiva, por los aspectos comunes que se manejan como son los temas ya que estos han variado poco a través de los tiempos, por ende, es importante identificar características comunes en diversas universidades, con lo cual se puede evitar el desarrollo de varias herramientas software que finalmente hacen lo mismo. Ver documento de sigis.

RECOMENDACIONES

La culminación del presente trabajo de grado no significa que la labor de investigación haya terminado, por el contrario marca el inicio de un proyecto el cual debe continuar ya que existe mucho por explorar, aprender y mejorar. A continuación se exponen una serie de recomendaciones con este propósito.

- Una recomendación que se debe tener presente al realizar herramientas con desarrollos Web, es la supervisión de la seguridad. Se deberá programar de manera defensiva, con únicamente un supuesto: siempre van a existir personas mal intencionadas, no se debe suponer que la aplicación será utilizada de una manera, se debe obligar que únicamente se pueda utilizar de manera correcta. También se debe verificar la configuración de todo el software que se utiliza: el sistema operativo, el manejador de base de datos, el servidor Web, y todo lo que intervenga en el funcionamiento de la aplicación.
- Se recomienda mejorar el proceso de creación de copias de seguridad de la información almacenada en la B.D. antes de realizar el mantenimiento de las tablas.
- Entre las posibles mejoras que se pueden realizar, la de mayor prioridad consiste en ampliar las capacidades del motor de dibujo para que alcance el manejo de los temas que no se abarcaron en este prototipo, como planos y vistas. La anterior mejora permitirá a su vez ampliar la cobertura de los módulos de problemas y evaluación, proyectando a futuro una herramienta CAD propia y acercando a GEDISO en un tutorial inteligente de Geometría Descriptiva.
- Se propone que se desarrollen las opciones de informes por parte del profesor que esta a cargo de los distintos grupos de trabajo y que a partir de ellos se pueda obtener reportes estadísticos del progreso de los estudiantes en la asignatura.
- Una recomendación para el módulo de teoría es que permita que los anexos formen parte de manera automática de los contenidos básicos y no como un documento adjunto; y que las explicaciones para la solución de los ejemplos se desplieguen de manera interactiva con animaciones y sonidos.

- Con la participación de especialistas en diseño grafico mejorar la presentación de la información de los diversos módulos de trabajo, dando una mejor distribución al espacio y haciendo más atractiva la interfaz de usuario.
- Se recomienda ampliar las opciones para que los profesores señalen los errores en las evaluaciones presentadas por los estudiantes.
- Se recomienda que GEDISO posea un editor de páginas Web propio para la fácil actualización de todos los contenidos de la herramienta.
- Se recomienda desarrollar una actualización en línea de las páginas que contiene la herramienta, desarrollando un editor de páginas Web.

BIBLIOGRAFÍA

- ◆ BOBADILLA, Jesús, ALCOCER, Alejandro. Superutilidades para Webmasters. McGraw Hill. Primera edición. España, 1999.
- ◆ CANTU Marco, Mastering Delphi 6, copyright © 2001 SIBEX, inc, Alameda CA.
- ◆ EVANS, Tim. Construya su propia Intranet. Segunda edición. Prentice Hall. México, 1997.
- ◆ ISRAEL Marc, JONES J. Steven, SQL server 2000, Desing Study Guide, copyright © 2001 SIBEX, inc, Alameda CA.
- ◆ JACOBSON, Ivar, BOOCH, Grady, RUMBAUGH, James. El Proceso Unificado de Desarrollo de Software. Primera edición. Addison Wesley. España, 2000.
- ◆ -----. El Lenguaje Unificado de Modelado. Primera edición. Addison Wesley. España, 1999.
- ◆ -----. Utilización de Dreamweaver Mx. copyright © 2002 Macromedia inc. Primera edición, junio 2002.
- ◆ -----. Utilización de Flash. copyright © 2002 Macromedia inc. Primera edición, marzo 2002.
- ◆ MCCONNELL, Steve. Desarrollo y Gestión de Proyectos Informáticos. Primera Edición. McGraw-Hill. España, 1997.
- ◆ PERDUE, Tim. Artículo: Best Practices: Database Transactions.
- ◆ PRESSMAN, Roger. Ingeniería del software. Un enfoque práctico. Cuarta edición. McGraw Hill. España, 1998.
- ◆ RATSCHILLER, Tobias. GERKEN, Till. Creación de Aplicaciones Web con PHP 4. Primera edición. Prentice Hall. España, 2001.

- ◆ SAETHER, Stig, AULBACH, Alexander y SCHMID, Egon. Manual de PHP. Quinta edición. Grupo de documentación de PHP, 2001.
- ◆ TEIXEIRA Steve, PACHECO Xavier, Guia de desarrollo Delphi 5. Volumen I y II Pearson Educación, Madrid, 2000.
- ◆ VARGAS German, GONZALES Jose, Informática educativa vs pedagogía computacional, Universidad Pedagógica Nacional, 1999.
- ◆ www.phpbuilder.com, Sitio de Desarrolladores en PHP.
- ◆ www.apache.org. Sitio oficial de Apache.
- ◆ www.php.net. Sitio oficial de PHP.
- ◆ www.desarrolloweb.com, Búsquedas de php, javascript, y sentencias en mysql.
- ◆ <http://iimec.ucr.ac.cr/~revista/articulos/1-2002/archivos/informatica.pdf>. Búsqueda de informática educativa.

ANEXO A.

REQUISITOS CANDIDATOS.

	NOMBRE	DESCRIPCIÓN	ESTADO	PRIORIDAD	NIVEL DE RIESGO
TEORIA	Anexo de documentos a la herramienta, por parte de los profesores.	La Herramienta permitirá agregar documentos como anexo a la información que la herramienta posee. Este trabajo solo puede ser elaborado por profesores, ya que se verificara que su clave y contraseña.	Aprobado	Importante	Ordinario
	Ver los diferentes documentos que han sido anexados	La Herramienta le permitirá a los diferentes usuarios ver los anexos que los profesores han suministrado como complemento a la información que ya se tiene.	Aprobado	Importante	Ordinario
	La creación de ejercicios resueltos paso a paso a manera de ejemplos.	La herramienta en el modulo de teoría contara con ejercicios resueltos paso a paso, a manera de ejemplo de acuerdo a los temas vistos en la asignatura.	Aprobado	Critico	Significativo
	La generación de listado de temas	El usuario de la herramienta tendrá la capacidad de escoger en un listado el tema que desea consultar o estudiar.	Aprobado	Secundario	Ordinario
	El desarrollo de un glosario de los temas que abarca la asignatura.	El usuario de la herramienta al encontrar un tema del cual no conoce su significado, puede acceder a este, pulsando el link que las palabras importantes poseen, mostrando su significado.	Aprobado	Secundario	Ordinario
	Almacenamiento de los anexos en el servidor	Se debe destinar un espacio en el servidor para almacenar los anexos que se suben en la página de crear anexos. El profesor, se encargara de su almacenamiento físico.	Aprobado	Importante	Significativo
	Descarga correcta del anexo desde el Servidor	Se debe Verificar que el documento pedido por el usuario sea el correcto, y el cual se debe abrir en la herramienta apropiada, para su mejor lectura.	Aprobado	Importante	Significativo
	Verificar el correcto enlace de las páginas, que contiene la información.	La herramienta debe tener una secuencia lógica para la presentación de la información que esta tiene sobre la asignatura, verificando que sus enlaces funcionen y sean los correctos.	Aprobado	Secundario	Ordinario

PRACTICA	Verificación del registro de los usuarios de la herramienta, el invitado no tiene permiso.	Los usuarios antes de escoger el ó los tipos de problemas que desean desarrollar, deberán ser registrados o verificar que sean usuarios autorizados, para proceder a utilizar la herramienta.	Aprobado	Importante	Ordinario
	La generación de listado de ejercicios a resolver.	La herramienta debe permitir a todos los usuarios escoger en un listado los diferentes problemas que van ha desarrollar en el tiempo asignado de trabajo por el administrador de la herramienta.	Aprobado	Secundario	Ordinario
	Solución de ejercicios a resolver por parte de los estudiantes	Este es un problema que concierne solo a los desarrolladores de la herramienta, buscando la mejor forma de crear la solución de los ejercicios, proporcionando a los estudiantes los elementos necesarios para la solución de los mismos.	Aprobado	Critico	Critico
	Registro de lo hecho en la hora de practica.	La herramienta debe permitir registrar las actividades realizadas por parte de los estudiantes en las horas asignadas de trabajo: el tiempo gastado por ejercicio, cantidad de ejercicios resueltos, etc.	Aprobado	Secundario	Ordinario
	Verificar que el listado de temas y subtemas corresponda a los ejercicios.	Se debe tener cuidado en direccionar de forma correcta los ejercicios que los usuarios piden para su desarrollo, en las horas de trabajo.	Aprobado	Critico	Critico
	Verificar que la datos guardados en la B.D. sean del usuario que trabajo en esa hora de practica.	Se debe verificar que la información que guarde la herramienta correspondiente al trabajo realizado por el estudiante, se almacene de forma correcta y completa.	Aprobado	Critico	Critico
	Los profesores tendrán la opción de ver los problemas que la herramienta tiene.	Solo los profesores podrán ver los diferentes problemas que la herramienta tiene, validando su código de permiso, los problemas se mostraran con el enunciado y la imagen correspondiente si la tiene.	Aprobado	Importante	Significativo

	Los profesores podrán ver los registros de trabajo de cada estudiante.	La herramienta permitirá solo a los profesor ver los diferentes registros, de todos las actividades realizadas por ellos en la horas de practica con el software.	Aprobado	Importante	Significativo
	La creación de diferentes informes	La herramienta permitirá la creación de informes, por estudiantes, por grupos, por practicas, evaluaciones.	Propuesto	Importante	Significativo
EVALUACIÓN	Verificación del registro de los usuarios de la herramienta.	Los usuarios antes de escoger el ó los tipos de problemas que desea desarrollar, deberán ser registrados o verificar que sea usuario autorizado, para proceder a utilizar la herramienta.	Aprobado	Importante	Ordinario
	Solución de ejercicios a resolver por parte de los estudiantes	Este es un problema que concierne solo a los desarrolladores de la herramienta, buscando la mejor forma de crear la solución de los ejercicios, proporcionando a los estudiantes los elementos necesarios para la solución de los mismos	Aprobado	Critico	Critico
	Registro de lo hecho en la hora de evaluación.	La herramienta debe permitir registrar todas las actividades realizadas por parte de los estudiantes en las horas asignadas de evaluación , el tiempo gastado por ejercicio, cantidad de ejercicios resueltos, etc.	Aprobado	Importante	Ordinario
	La generación de listado de ejercicios a resolver.	La herramienta debe asignar de forma aleatoria a los diferentes usuarios que se encuentren en el modulo de evaluación los ejercicios a resolver de acuerdo a un listado de los diferentes problemas que se encuentran almacenados en la base de datos de la herramienta y dado el tema a evaluar.	Propuesto	Importante	Ordinario
	Consulta de notas.	Los estudiantes y el profesor podrán consultar las notas correspondientes. Los estudiantes podrán acceder a sus notas en cualquier momento, después de haber sido calificadas sus evaluaciones.	Propuesto	Secundario	Ordinario

	Aparte de la evaluación con ejercicios se realizara una teórica.	La herramienta permitirá realizar una evaluación teórica sobre los diferentes temas que la herramienta posee, mostrando el enunciado del problema y el estudiante dará su respuesta y la sustentara.	Aprobado	Importante	Significativo
	La Herramienta permitirá ver las distintas evaluaciones de los estudiantes.	La Herramienta permitirá solo a los profesores ver las evaluaciones hechas por los estudiantes.	Aprobado	Critico	Critico
	La herramienta permitirá asignar notas a las evaluaciones.	La herramienta permitirá solo a los profesores asignar notas a las evaluaciones que los estudiantes realizaron.	Aprobado	Critico	Critico
	Verificar el correcto almacenamiento de la evaluación en la B.D.	Se verificara que las evaluaciones sean almacenadas de forma correcta en la base de datos, correspondiente a lo realizado por cada estudiante.	Aprobado	Critico	Critico
ADMINISTRACIÓN	Ingresar y/o modificar usuarios.	La herramienta debe permitir la creación de nuevos usuarios, así como la modificación de los datos que se encuentren registrados de los mismos.	Aprobado	Critico	Ordinario
	Actualizar los ejercicios a resolver en el modulo de practica y evaluación	El administrador estará en la capacidad de ingresar nuevos ejercicios a la base de datos que contiene la herramienta, para evitar el aprendizaje de estos por parte de los estudiantes que manejan la herramienta, esto se realiza con el cambio del estado del problema.	Aprobado	Critico	Significativo
	Administración de grupos.	Al inicio de cada semestre se debe permitir abrir un grupo y asignar el profesor y alumnos que pertenezcan a él. También agregar y eliminar estudiantes, cambiar profesores y modificar los datos asociados a dicho grupo.	Aprobado	Importante	Ordinario
	Ingresar ejercicios de problemas y teóricos.	La herramienta en este modulo permitirá insertar en la Base de Datos, los diferentes ejercicios, sean teóricos o problemas que serán resueltos por los estudiantes.	Aprobado	Importante	Significativo

	Sacar Ejercicios de la base de datos	La herramienta permitirá al administrador sacar los ejercicios de la base de datos, llegado el caso de que estén mal planteados.	Aprobado	Importante	Ordinario
	Eliminar historial de trabajo.	La herramienta permite borrar el historial de trabajo realizado por los estudiantes durante el semestre, para liberar espacio en la B.D.	Aprobado	Importante	Significativo
	Eliminar historial de evaluaciones	La herramienta permitirá borrar el historial creado por las distintas evaluaciones que los usuarios realizaron.	Aprobado	Importante	Significativo
	Copia de Seguridad	La herramienta permitirá crear una copia de seguridad de toda la base de datos por si llegase a ocurrir perdida de datos.	Aprobado	Importante	Ordinario
	Ver mensajes	El administrador vera los diferentes mensajes que los usuarios tienen sobre la herramienta.	Aprobado	Secundario	Ordinario
OTRAS	Bajos costos en el desarrollo y la implantación.	El software debe desarrollarse e implantarse con mínimos costos, sin afectar su funcionamiento.	Aprobado	Importante	Ordinario
	Insertar mensajes	El software le permite a los usuarios enviar mensajes al administrador de problemas que este tiene.	Aprobado	Secundario	Ordinario

ANEXO B.

MODELO DEL NEGOCIO BASADO EN CASOS DE USO.

Proceso de Negocio	Realizar Ejercicios
Objetivo	Realizar en las horas de práctica los ejercicios que la herramienta les suministra, para comprender la metodología de desarrollo de los diferentes tipos de ejercicios.
Descripción	<ol style="list-style-type: none">1. Se consulta la base de datos para verificar si el usuario esta registrado, si no lo esta debe registrarse para hacer uso de la herramienta.2. El usuario ejecuta el programa de práctica de problemas.3. El software le despliega parte de su información personal, y el listado de temas de los ejercicios.4. Al escoger el tema se despliega, los subtemas, a su vez después de haber escogido el subtema se muestra un combo con los diferentes problemas.5. Por ultimo el usuario escoge el problema y luego se despliega la información de este, para ser resuelto por el estudiante.
Prioridad	Importante
Riesgos	Critico
Posibilidades	Este proceso puede ser mejorado al poseer un canal continuo de comunicación con la teoría donde las personas fácilmente puedan consultarla la información y despejar dudas que se les presente en ese momento, o comunicación con una archivo que contenga una gran cantidad de ejercicios resueltos.
Tiempo de Ejecución	Depende del tamaño de la complejidad del programa y de la cantidad de pruebas y ejercicios que se creen. (2 semana – 3 meses)
Coste de Ejecución	Depende de los mismos factores que para el tiempo de ejecución

Proceso de Negocio	Evaluar Ejercicios
Objetivo	El profesor realizar la evaluación del tema que el crea conveniente, para conocer el grado de conocimiento de los estudiantes en la materia.
Descripción	<ol style="list-style-type: none">1. Se consulta la base de datos para verificar si el usuario esta registrado, si no lo esta debe registrarse para hacer uso de la herramienta.2. El estudiante entra en el modulo de evaluación.3. Se ejecuta el programa de evaluación.4. Se despliega información del usuario.5. Aparece el listado de temas, luego el de subtema dependiendo del tema escogido.6. Aparece el listado de problemas.7. El estudiante escoge uno para su solución.
Prioridad	Importante
Riesgos	Critico
Posibilidades	----

Tiempo de Ejecución	Depende del tamaño de la complejidad del programa y de la cantidad de pruebas y ejercicios que se creen. (2 semana – 3 meses)
Coste de Ejecución	Depende de los mismos factores que para el tiempo de ejecución

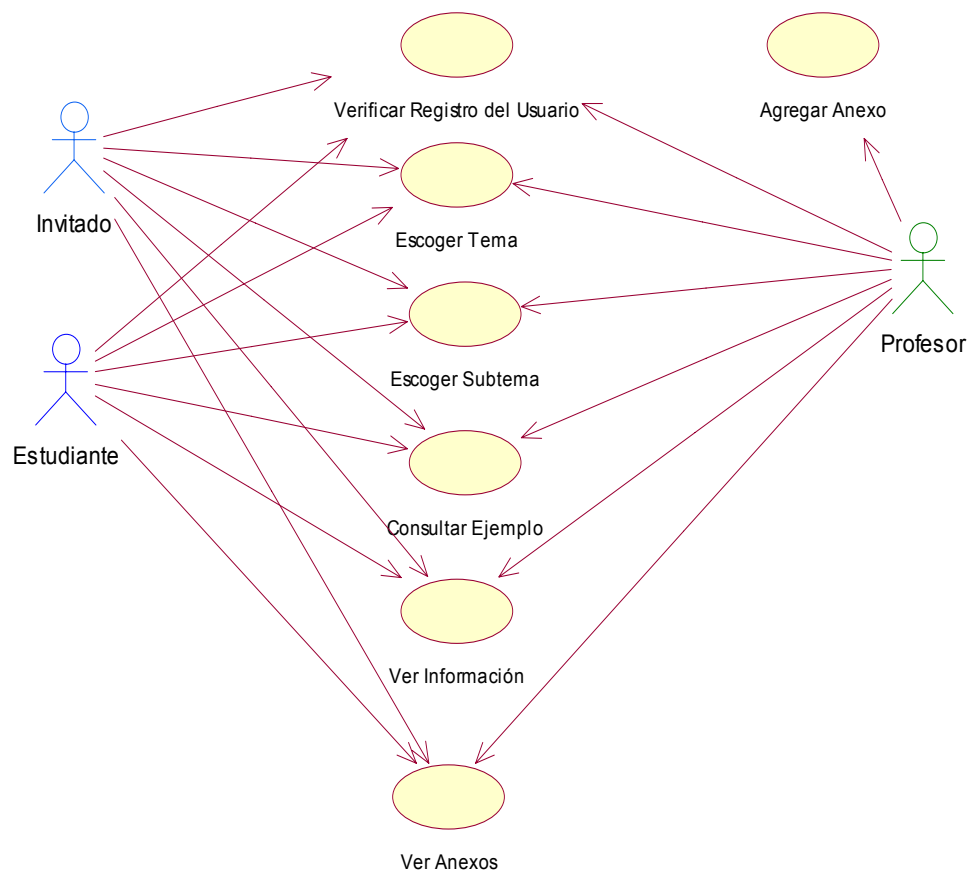
ANEXO C

MODELO DE CASO DE USO DE LA HERRAMIENTA.

Este anexo recopila el modelo de casos de uso del sistema en su totalidad, los casos de uso identificados han sido agrupados según el paquete de análisis al que correspondan.

Para cada paquete se presenta su diagrama del modelo de casos de uso y luego una descripción general de dicho modelo. A continuación se presenta una descripción breve de cada caso de uso y por ultimo se detallan aquellos que necesitan una explicación mas detallada para su comprensión.

◆ PAQUETE DE GESTIÓN DE CONSULTA DE TEMAS.



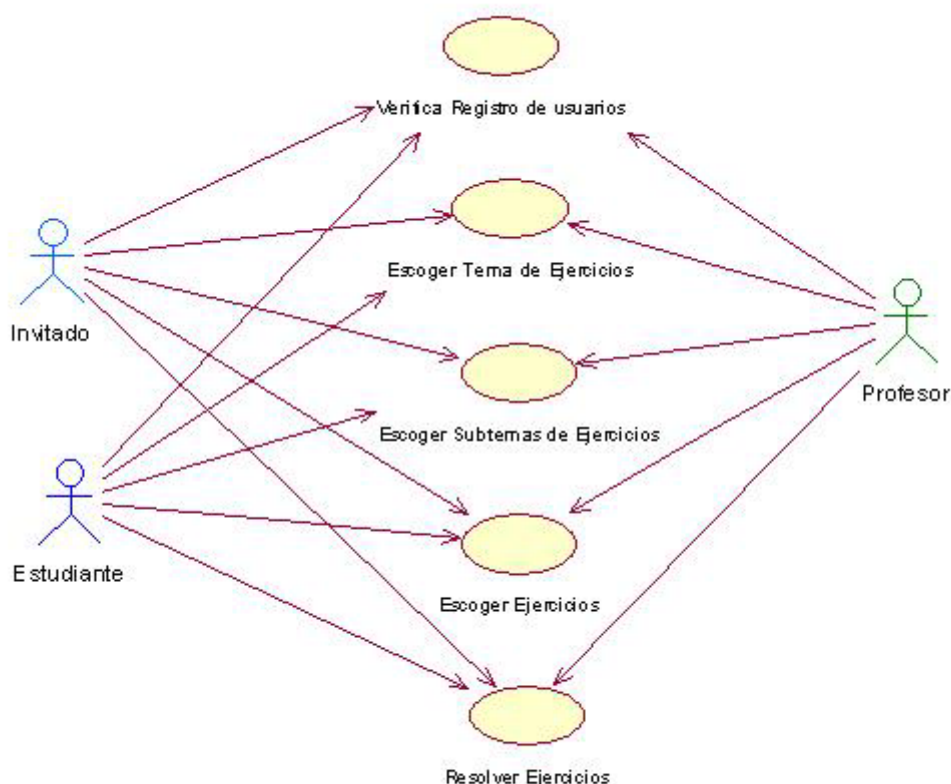
Descripción del Modelo de Consultar Temas
<p>La Herramienta ofrece a los estudiantes la opción de consultar todos los temas que se encuentran en el pensum académico de la asignatura geometría descriptiva.</p> <p>Para una mejor comprensión de la teoría estos están organizados por los temas de mayor importancia en la asignatura y estos a su vez contienen los subtemas respectivos. Así, los diferentes contenidos pueden ser accedidos por los estudiantes sin ninguna dificultad, dado que estas fueron desarrolladas en ambiente Web, conocido por la mayoría de las personas.</p> <p>El profesor también tiene acceso a los diferentes temas que contiene la herramienta de la misma manera que lo hacen los estudiantes. Además este caso de uso le permitirá corroborar sus conocimientos en algún tema de interés. De la misma manera que los hace el profesor y el estudiante el software permitirá la entrada a un invitado que puede consultar la información que desee, además de consultar los ejemplos que este módulo contiene.</p> <p>El profesor tiene la opción de agregar algún anexo a manera de comentario para completar o aclarar la información que la herramienta posee.</p>

Casos de Uso Identificados	Breve Descripción
Verificar el registro de los usuarios	Se utiliza este caso de uso para verificar que el usuario que entra a la herramienta es válido y que permisos posee.
Escoger tema	Se este caso de uso para escoger los temas que la herramienta suministra.
Escoger Subtema	Este caso de uso permite escoger subtemas de acuerdo al tema que se ha escogido.
Consultar Ejemplos	Este caso de uso permite consultar los distintos ejemplos que la herramienta tiene, estos están al final de cada tema.
Ver Información	Este caso de uso permite desplegar la información de el usuario requiera.
Ver Anexos	Este caso de uso permite ver los diferentes anexos que la herramienta tiene almacenados.
Agregar Anexos.	Este caso de uso permite solo a los profesores agregar los diferentes anexos a la herramienta.

Nombre	Agregar Anexos
Breve Descripción	El Profesor utiliza este caso de uso para publicar un anexo de un tema determinado el cual crea que es importante para su conocimiento por parte de los alumnos.
Descripción Paso a Paso Inicial	<p>Dado que ya existe el tema al cual quiere que pertenezca el artículo y dado que el Profesor es el encargado de subir documentos de carácter importante:</p> <ol style="list-style-type: none"> 1. El Profesor escoge el tema y el subtema que al que va a pertenecer el anexo. 2. El Profesor digita el nombre del anexo, escoge la fecha y escribe un comentario del mismo. 3. Una vez el Profesor envía el anexo, el sistema lo pone a disposición de las personas

	<p>que tengan permiso de acceder a este modulo.</p> <p>El CU termina cuando el profesor sube el anexo al servidor y recibe una notificación de que la transacción se realizó correctamente o que no pudo ser realizada.</p>
--	---

◆ PAQUETE DE GESTIÓN DE RESOLVER EJERCICIOS



Descripción del Modelo Resolver Ejercicios

La Herramienta permite a los estudiantes resolver una gran cantidad de ejercicios ya sean generales o específicos, que ayudaran a los alumnos ha comprender y mecanizar las diferentes maneras de cómo tomar un camino de solución para las diferentes clases de ejercicios, basados estos en las características del enunciado de cada problema.

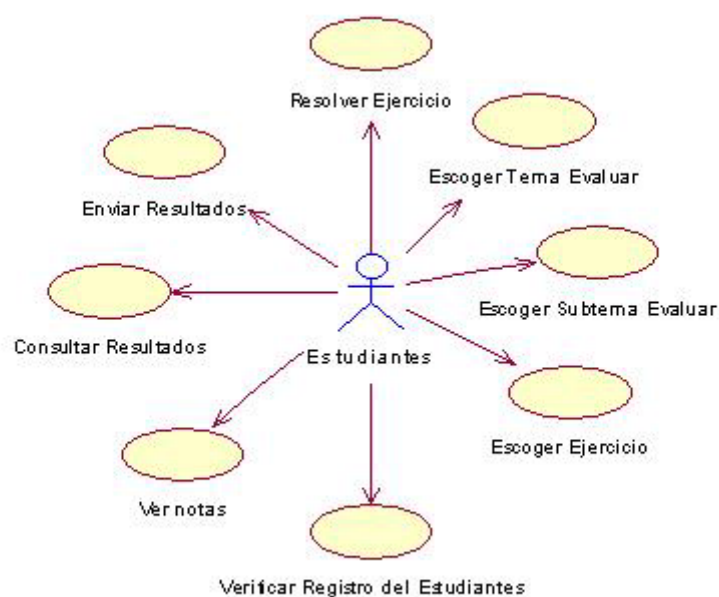
Los Ejercicios que el estudiante resuelva o trabaje durante la horas de trabajo en la herramienta, serán registrados de manera automática en la B.D, cada ves que finalice su sección de trabajo, los datos que se almacenan serán (ejercicio trabajado, el tiempo gastado por ejercicio, entre otros datos).

Esto les permitirá a los profesores, llevar un control sobre el estudiante de que es lo que ellos hacen en las horas de laboratorio y su desempeño con la herramienta, dando un soporte para evaluar al estudiante si así el profesor lo desea.

Casos de Uso Identificados	Breve Descripción
Verificar el registro del usuario	Se utiliza este caso de uso para verificar que el usuario que entra a la herramienta es valido y que permisos posee.
Escoger tema de ejercicios	Permite al usuario escoger el tema de los ejercicios a resolver.
Escoger Subtema de ejercicios	Permite escoger los subtemas de los ejercicios de acuerdo al tema escogido.
Escoger ejercicio	Permite seleccionar el problema a trabajar de acuerdo a la relación tema, subtema.
Resolver ejercicio	Permite al usuario resolver el problema en una hoja de trabajo que posee la herramienta.

Nombre	Resolver Ejercicio
Breve Descripción	Los usuarios de la herramienta excepto el invitado tienen permiso para este modulo, en el se resolverán problemas sobre los diversos temas que se presentan en el software.
Descripción Paso a Paso Inicial	<p>Para solucionar un ejercicio el usuario entra al modulo de problemas, que es una aplicación de delphi, donde se despliegan algunos datos personales del usuario:</p> <ol style="list-style-type: none"> El usuario escoge el tema a trabajar. El usuario escoge el subtema de acuerdo al tema, y a su vez escoge el problema, que resolverá. Una vez seleccionado lo anterior se despliega una hoja de trabajo que tiene una serie de componentes que le ayudaran para la solución del problema. <p>El CU termina cuando el usuario da por terminado el problema y quiere solucionar otro.</p>

PAQUETE DE GESTIÓN DE EVALUACION

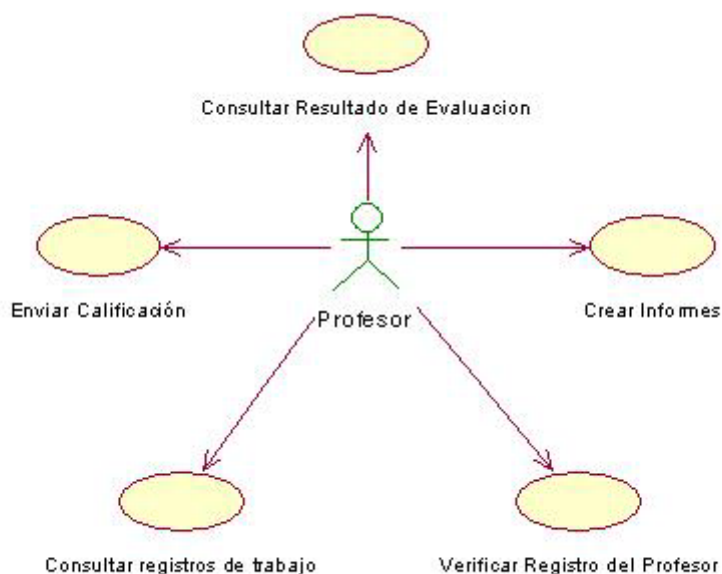


Descripción del Modelo Evaluar Tema.
<p>La Herramienta permite a los usuarios realizar una serie de problemas que serán calificados por parte de los profesores entregando una nota que se computara con el resto para entregar una nota final cuando se termine el semestre académico.</p> <p>Las Evaluaciones se realizaran de varias maneras, habrá problemas de selección basados en la teoría que posee la herramienta, ejercicios a resolver. Después de haber terminado con la evaluación el estudiante guardara los resultados para entregarlo al profesor para su calificación. En este mismo modulo los estudiantes consultaran las notas de las evaluaciones que ya han sido calificadas por los profesores.</p>

Casos de Uso Identificados	Breve Descripción
Verificar el registro del usuario	Se utiliza este caso de uso para verificar que el usuario que entra a la herramienta es valido y que permisos posee.
Escoger tema evaluar	Permite al usuario escoger el tema de la evaluación
Escoger subtema evaluar	Permite escoger el subtema a evaluar.
Escoger ejercicio	Permite seleccionar el problema a evaluar.
Resolver ejercicio	Permite resolver la evaluación de acuerdo al ejercicio seleccionado
Enviar resultados	El software se encarga de forma automática de enviar los resultados de la evaluación a la base de datos, además capturando algunos datos correspondientes al usuario.
Consultar resultados	Permite al profesor revisar las distintas evaluaciones que los alumnos realizaron.

Nombre	Enviar resultados
Breve Descripción	Cuando se realiza una evaluación y estudiante da por terminado la solución del problema, escogiendo el siguiente a resolver la herramienta envía la trabajado por el estudiante a la base de datos y otros datos que luego serán consultados por los profesores.
Descripción Paso a Paso Inicial	<p>Para enviar los resultados el usuario da por terminado el problema y decide resolver otro ejercicio.</p> <ol style="list-style-type: none"> 1. El usuario termina el problema. 2. El software toma ciertos datos y los envía la base de datos. <p>El CU termina cuando el usuario da por terminado el problema y quiere solucionar otro.</p>

◆ PAQUETE DE GESTIÓN DE CONSULTA DE REGISTROS



Descripción del Modelo Consultar Registros

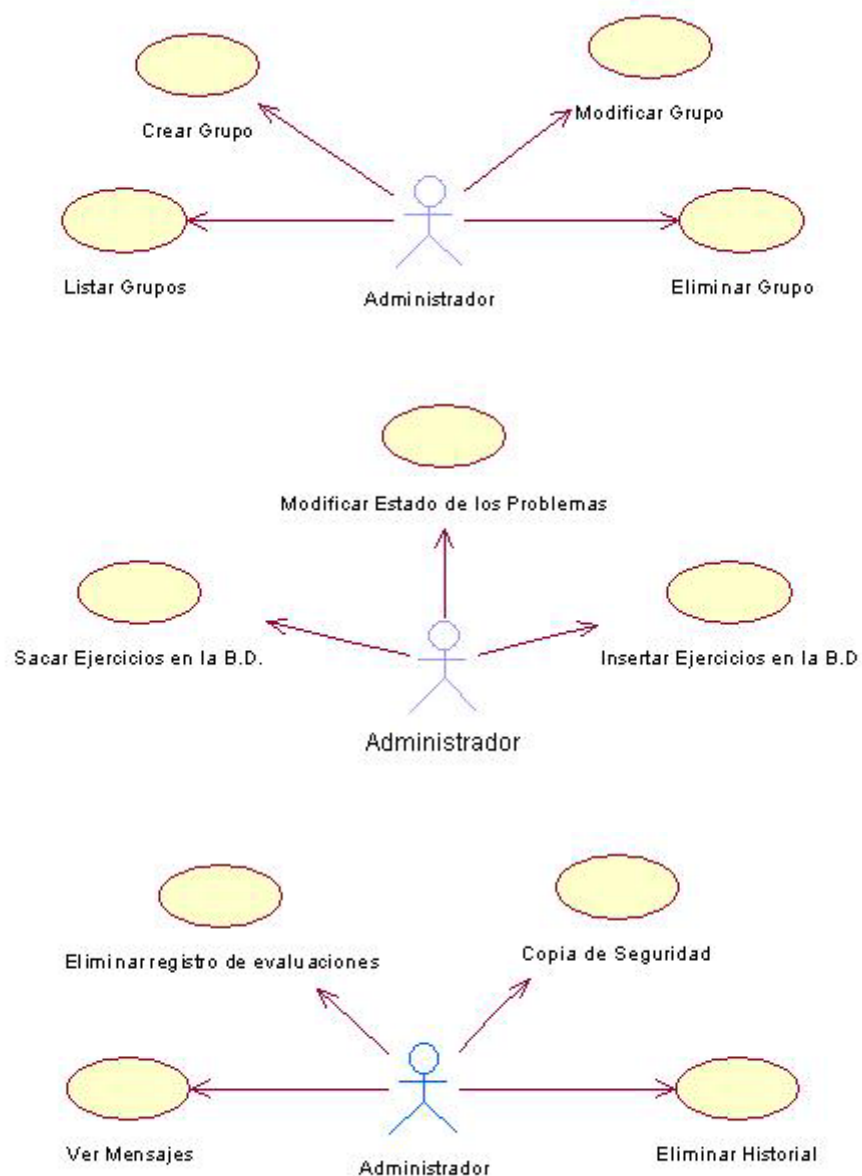
La herramienta permite a los profesores consultar los diferentes registros almacenados en la base de datos de los ejercicios realizados por los estudiantes en sus horas de trabajo, con estos datos ellos pueden crear informes, teniendo un historial de cada uno de los estudiantes, de lo hecho durante el semestre por cada uno de ellos. Estos registros también pueden ser considerados como un factor de calificación, ya que estos muestran, el aprovechamiento que los estudiantes le dieron a la herramienta.

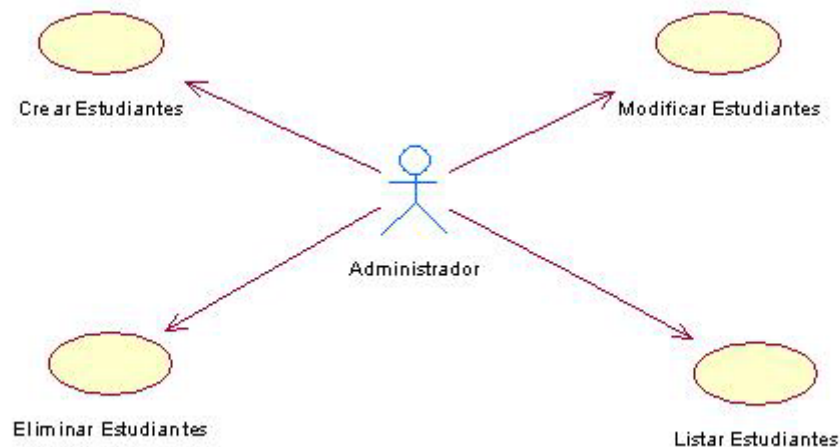
Casos de Uso Identificados	Breve Descripción
Verificar registro del profesor	Se utiliza este caso de uso para verificar que el usuario que entra a la herramienta es valido y que permisos posee.
Consultar resultado de evaluación	Permite ver los resultados de todas las evaluaciones hechas por los estudiantes.
Crear informes	Permite crear informes de los registros que la herramienta posee.
Enviar calificación	Permite enviar las notas de las evaluaciones que han sido calificadas por los profesores.
Consultar registros de trabajo	Permite observar los registros de los problemas solucionados en el modulo de problemas, por cada estudiante.

Nombre	Consultar resultado de evaluación
Breve Descripción	El profesor cuando lo desee puede verificar las soluciones de los problemas que los usuarios resolvieron en el modulo de evaluación, ya que están almacenadas en la base de datos del

	software.
Descripción Paso a Paso Inicial	<p>Para revisar los resultados de la evaluación los profesores escogerán el código de un estudiante, la información guardada en la base de datos de las evaluaciones hechas por el se desplegaran, el profesor escoger una y la revisara.</p> <ol style="list-style-type: none"> 3. Se despliega la imagen de la evaluación hecha por el estudiante. 4. El profesor la revisa para y mira si es correcta la solución o no. <p>El CU termina cuando el profesor da por terminado la revisión de los problemas y quiere realizar otra acción.</p>

◆ PAQUETE DE GESTIÓN DE ADMINISTRACION.





Descripción del Modelo Administrar Grupo

La herramienta permite al administrador crear, modificar, eliminar y listar los diferentes grupos de trabajo que soportara la herramienta. Los grupos serán creados de acuerdo al listado entregado por los profesores y estos son los mismos grupos de clase que ellos manejan, a estos el administrador les asignara un horario de trabajo los cuales los estudiantes deben cumplir.

En el momento en que algún estudiante cancele la asignatura, el profesor debe notificar al administrador para que modifique el grupo en el cual se retiro el estudiante.

Descripción del Modelo Administrar Ejercicios

La Herramienta permite al administrador sacar e insertar los ejercicios en la Base de Datos. Los ejercicios que contiene la herramienta están organizados de dos grupos principales, generales y específicos, estos de acuerdo a las diferentes carreras que tienen en su pensul la asignatura geometría descriptiva, aparte de estos habrán problemas de selección que ayudaran ha verificar los conocimientos teóricos de los estudiantes.

Quando los problemas son agregados a la Base de Datos se les da un estado de activo o no activo el cual puede ser modificado por el administrador, con el fin de evitar que siempre estén funcionando los mismos ejercicios en la Base de Datos de la herramienta.

Descripción del Modelo Administrar otras opciones

La Herramienta permite al administrador tener otras opciones como son eliminar registros de la base de datos que la harían pesada, esta eliminación se ejecutarían al final de cada semestre para liberar espacio y permitir el almacenamiento de nueva información, pero esta solo se realizaría si la base de datos se tornara lenta al devolver la información requerida de tal manera el administrador podría presidir de este servicio, pero para mayor seguridad en la perdida de datos se tendrá la opción de realizar copias de seguridad, llegado el caso de un colapso de la base de datos de la herramienta, se tendrán copias de seguridad que serán ejecutadas en un intervalo de tiempo prudente.

Además de esto puede ver los mensajes que los usuarios envían relacionados a las fallas encontradas en el sistema o recomendaciones que estos tienen para mejorar la herramienta.

Descripción del Modelo Administrar Estudiantes

La herramienta permite al administrador crear, modificar, eliminar y listar los diferentes estudiantes que se encuentran matriculados para cursar el área de geometría descriptiva. Los estudiantes serán creados de acuerdo al listado entregado por los profesores y estos son los mismos que forman las clases magistrales que los ellos manejan, a estos el administrador les asignara un grupo de trabajo que tiene el mismo código entregado por la universidad ejemplo (a1).

En el momento en que algún estudiante cancele la asignatura, el profesor debe notificar al administrador para eliminarlo de la base de datos y del grupo al cual pertenecía.

Casos de Uso Identificados	Breve Descripción
Crear Usuario	Permite la creación de usuarios en la herramienta, asignándoles un perfil.
Modificar Usuario	Permite modificar un usuario existente.
Eliminar Usuario	Permite borrar un usuario existente.
Listar Usuario	Permite listar los usuarios existentes por los diferentes grupos.
Ver Mensaje	Permite ver los mensajes que los estudiantes envían sobre el software
Eliminar Historial	Permite borrar el historial de trabajo en el modulo de problemas.
Eliminar Registro de Evaluaciones.	Permite borrar el historial de las evaluaciones realizadas por los estudiantes.
Copia de Seguridad	Permite al administrador crear un Backup dado un tiempo prudente para evitar perdidas de información en la base de datos.
Crear Grupo	Permite Crear un grupo y vincular estudiantes a este grupo.
Modificar Grupo	Permite Modificar un grupo.
Eliminar Grupo	Permite Borrar un grupo existente.
Listar Grupo	Permite listar los grupos de acuerdo a los diferentes profesores que los tienen a cargo.
Insertar Ejercicios	Permite insertar los ejercicios en la base de datos.
Sacar Ejercicios	Permite retirar los ejercicios de la base de datos.
Modificar Estado de Ejercicios	Permite modificar el estado de los problemas para poder ser visibles en la herramienta.

Nombre	Crear Grupo
Breve Descripción	El Administrador utiliza el caso de uso para abrir un nuevo grupo, asignarle un profesor.
Descripción	Una vez que inicia un nuevo semestre, el administrador podrá utilizar este caso de uso con base en los datos proporcionados por la dependencia de admisiones de la universidad.
Paso a Paso Inicial	1. El administrador le asigna un profesor al grupo.

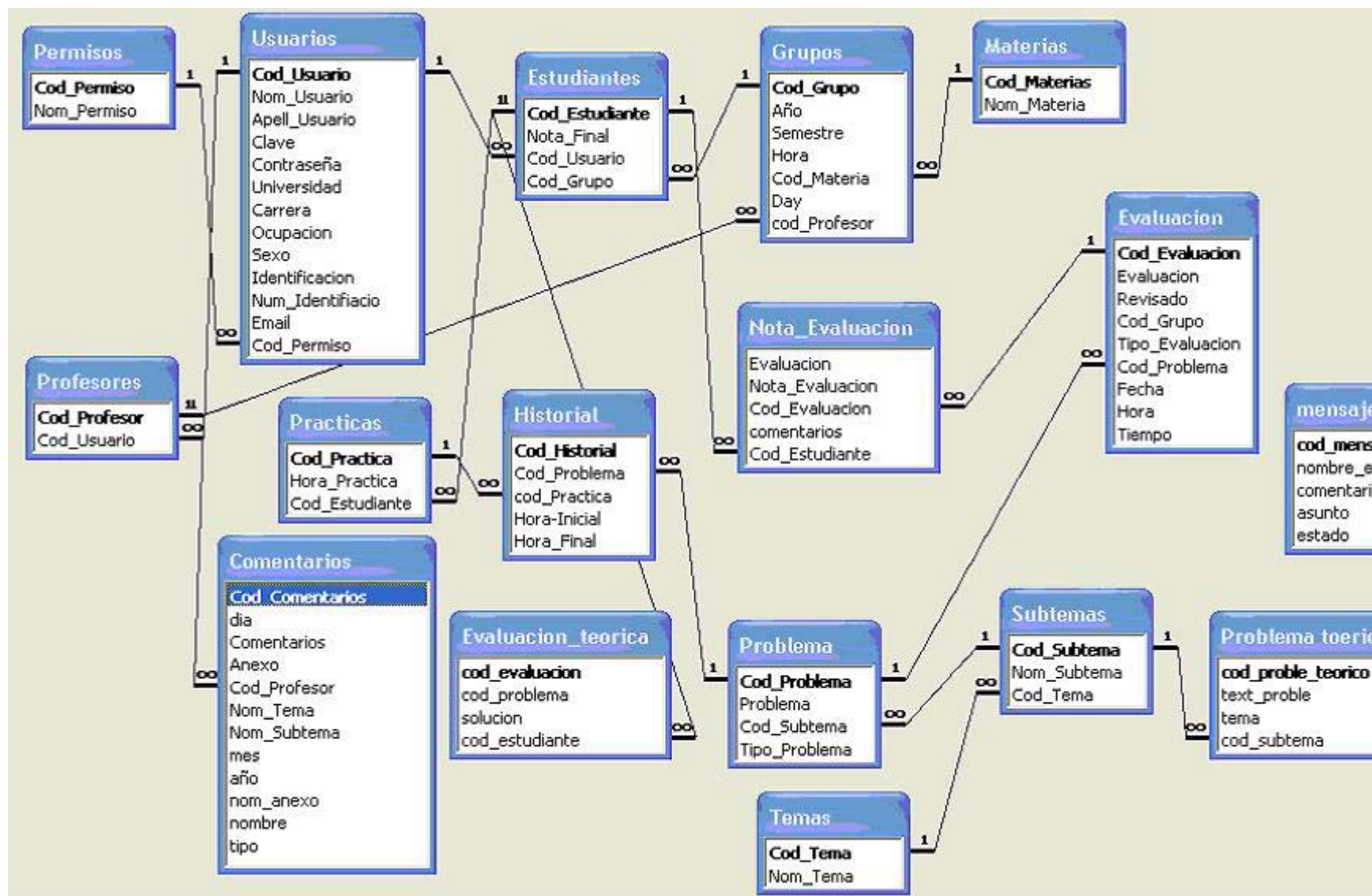
	<p>2. El administrador selecciona la materia que se dictara, semestre, año, horario, día.</p> <p>Todos estos datos se almacenan en la base de datos.</p> <p>El CU termina cuando el administrador da por terminado la creación del grupo y quiere realizar otra acción.</p>
--	---

ANEXO D.

DICCIONARIO DE DATOS

TABLAS DE LA BASE DE DATOS

Tabla	Información
usuario	Todos los usuarios de sistema para todos los actores
materias	Geometría Descriptiva I y Geometría Descriptiva II
grupos	Grupos creados semestralmente para cada una de las materias
estudiantes	Estudiantes que pertenecen a cada uno de los grupos
evaluacion	Evaluaciones registradas por cada uno de los estudiantes
Nota_evaluacion	Nota para cada una de las evaluaciones de cada estudiante de un grupo
comentarios	Son los anexos que los profesores han subido a la base de datos.
temas	Temas de los cuales esta compuesta las asignaturas Geo. Descriptiva I y II
mensajes	Son los mensajes enviados por los usuarios de la herramienta, al administrador.
practicas	Corresponde a las prácticas hechas por los estudiantes.
historial	Es el registro de todos los problemas hechos por los estudiantes en sus prácticas.
Evaluacion_teorica	Corresponde a la solución de los problemas teóricos.
Problema teóricos	Son los problemas teóricos que se les entregan a los estudiantes para ser resueltos.
profesores	Todos los profesores que tienen a cargo un grupo
subtemas	Son los subtemas que cada tema posee.
Permisos	Permisos necesarios para acceder a los módulos de la herramienta.
problemas	Corresponde a los problemas que serán resueltos por los estudiantes.



➤ INFORMACIÓN DE LAS TABLAS

◆ TABLA ESTUDIANTE.

Campo	Tipo	Longitud	Requerido	Llave Primaria	Default	Descripción
Cod_estudiante	int4	4	Si	Si		identificador de un estudiante
Nota_final	float	3	No	No		Nota final del estudiante, suma de todas las notas.
Cod_usuario	int4	4	Si	No	3	identificador que referencia al usuario-estudiante
Cod_grupo	Char	3	Si	No		Identificador del grupo a que pertenece el estudiante.

Llave Foránea	Referencia a:	
	Tabla	Llave Primaria
Cod_usuario	usuario	Cod_usuario
Cod_grupo	grupo	Cod_grupo

◆ TABLA COMENTARIOS.

Campo	Tipo	Longitud	Requerido	Llave Primaria	Default	Descripción
Cod_comentario	int	3	Si	Si	autonumérico	Identificador del anexo
dia	int	2	Si			Día en el que se subió el anexo
mes	varchar	10	Si			Mes en el cual se creo el anexo
año	int	4	Si			Año en el que se creo el anexo
comentario	longblob		Si			Ruta donde se almacena el anexo
Nom_tema	varchar	100	Si			Nombre del tema relacionado al anexo
Nom_subtema	varchar	100	Si			Nombre del subtema relacionado al anexo
anexo	mediumblob		Si			Comentario acerca del anexo
Nom_anexo	varchar	50	Si			Nombre que el profesor le da al anexo
Cod_profesor	int4	4	Si			Código del profesor que sube el anexo
nombre	varchar	50	Si			Nombre del documento anexado
tipo	varchar	30	Si			Tipo de extensión del documento anexado

Llave Foránea	Referencia a:	
	Tabla	Llave Primaria
Cod_profesor	profesor	Cod_profesor

◆ TABLA GRUPO.

Campo	Tipo	Longitud	Requerido	Llave Primaria	Default	Descripción
Cod_grupo	char	3	Si	Si		Identificador del grupo
año	varchar	4	Si			Año en que se crea el grupo
semestre	varchar	10	Si			Semestre en el que se crea el grupo

Hora	varchar	16	Si			Hora en la cual se trabaja con la herramienta.
dia	varchar	15	Si			Día en el cual el grupo trabaja con la herramienta.
Cod_materia	int4	4	Si			Código de la materia que se trabaja en el grupo.
Cod_profesor	int4	4	Si			Código del profesor que esta a cargo del grupo.

Llave Foránea	Referencia a:	
	Tabla	Llave Primaria
Cod_materia	materias	Cod_materia
Cod_profesor	profesor	Cod_profesor

◆ TABLA EVALUACION.

Campo	Tipo	Longitud	Requerido	Llave Primaria	Default	Descripción
cod_evaluacion	Int3	3	Si	Si	autonumérico	Identificación de la evaluación
Evaluacion	longblob		Si			Resultado de la evaluación
revisado	varchar	5	Si			Estado de la evaluación.
Tipo_evaluacion	varchar	20	Si			Tipo de evaluación, teórica o práctica.
año	varchar	4	Si			Año en que se realizo la evaluación
mes	varchar	11	Si			Mes en la que se realizo
dia	char	2	Si			Día en la que se realizo
hora	varchar	10	Si			Hora en la que se realizo la evaluación
Cod_problema	Int3	3	Si			Código correspondiente al problema
Cod_grupo	char	3	Si			Grupo al cual pertenece el estudiante, que realizo la evaluación.

Llave Foránea	Referencia a:	
	Tabla	Llave Primaria
Cod_problema	problemas	Cod_problema
Cod_grupo	grupo	Cod_grupo

◆ TABLA EVALUACION_TEORICA.

Campo	Tipo	Longitud	Requerido	Llave Primaria	Default	Descripción
Cod_evaluacion	Int3	3	Si	Si	autonumérico	Identificador de la evaluación teórica.
Cod_problema	Int3	3	Si			Código del problema que soluciono
solucion	text		Si		3	Respuesta al problema
Cod_estudiante	Int3	3	Si			Código del estudiante que hizo la evaluacion.

Llave Foránea	Referencia a:	
	Tabla	Llave Primaria
Cod_problema	problemas	Cod_problema
Cod_estudiante	estudiante	Cod_estudiante

◆ TABLA DE HISTORIAL.

Campo	Tipo	Longitud	Requerido	Llave Primaria	Default	Descripción
cod_historial	Int3	3	Si	Si	autonumérico	Identificación del historial
Hora inicial	varchar	15	Si			Hora en que inicio la evaluación.
Hora final	varchar	16	Si			Hora en que termino la evaluación
semestre	varchar	12	Si			Semestre en que se realizo la evaluación.
año	varchar	4	Si	2004		Año en que se realizo la evaluación
mes	varchar	11	Si			Mes en la que se realizo
dia	char	2	Si			Día en la que se realizo
Cod_practica	Int4	4	Si			Código de la practica en que se desarrollo la evaluación.
Cod_problema	Int3	3	Si			Código correspondiente al problema

Llave Foránea	Referencia a:	
	Tabla	Llave Primaria
Cod_problema	problemas	Cod_problema
Cod_practica	practica	Cod_practica.

◆ TABLA DE MATERIAS.

Campo	Tipo	Longitud	Requerido	Llave Primaria	Default	Descripción
Cod_materia	Int4	4	Si	Si		Identificador de la materia que tiene la herramienta.
Nom_materia	Char	30	Si			Nombre de la materia, correspondientes al pensul académico de la Geometría Descriptiva.

◆ TABLA DE MENSAJES.

Campo	Tipo	Longitud	Requerido	Llave Primaria	Default	Descripción
Cod_mensaje	Int3	3	Si	Si	autonumérico	Identificador del mensaje enviado por el usuario.
Nombre_estudi	varchar	30	Si			Nombre del estudiante que envió el mensaje
comentario	text		Si			Texto del mensaje que se envía.
asunto	varchar	30	Si			El asunto del mensaje.
estado	varchar	12	Si		pendiente	Si el mensaje ha sido revisado por el administrador o no.

◆ TABLA NOTA_EVALUACION.

Campo	Tipo	Longitud	Requerido	Llave Primaria	Default	Descripción
Cod_nota_evalu	int	3	Si	Si	autonumérico	Identificador correspondiente a la nota de

acion						la evaluación
Evaluación	longblob		Si			Resultado de la evaluación
Nota_evaluación	float		No			Nota de las evaluaciones realizadas
Cod_evaluación	Int3	3	Si			Código de la evaluación a la cual corresponde la nota.
comentarios	text		No			Comentarios hechos por los profesores de la evaluación.
Cod_estudiante	int4	4	Si			Código del estudiante que realizo la evaluación.

Llave Foránea	Referencia a:	
	Tabla	Llave Primaria
Cod_estudiante	estudiante	Cod_estudiante
Cod_evaluacion	evaluación	Cod_evaluacion

◆ TABLA DE PERMISOS.

Campo	Tipo	Longitud	Requerido	Llave Primaria	Default	Descripción
Cod_permiso	Int3	3	Si	Si	3	identificador del permiso a que corresponde cada usuario.
Nom_permiso	Char	20	Si			Nombre del permiso que tienen los usuarios de la herramienta.

◆ TABLA DE TEMAS.

Campo	Tipo	Longitud	Requerido	Llave Primaria	Default	Descripción
Cod_tema	Int3	3	Si	Si		Identificador de los temas en que esta compuesta la herramienta, de acuerdo al pemsul académico.
Nom_tema	Char	100	Si			Nombre de los temas que tiene la herramienta.

◆ TABLA DE SUBTEMAS.

Campo	Tipo	Longitud	Requerido	Llave Primaria	Default	Descripción
Cod_subtema	Int3	3	Si	Si		Identificador de los subtemas en que esta compuesta la herramienta, dependiendo de los temas.
Nom_subtema	Char	100	Si			Nombre de los subtemas que tiene la herramienta.
Cod_tema	Int3	3	Si			Identificador de los temas en que esta compuesta la herramienta, de acuerdo al pemsul académico.

Llave Foránea	Referencia a:	
	Tabla	Llave Primaria
Cod_tema	temas	Cod_tema

◆ TABLA DEL PROFESOR.

Campo	Tipo	Longitud	Requerido	Llave Primaria	Default	Descripción
Cod_profesor	Int3	3	Si	Si		Es el código que se le es asignado al profesor, por la universidad.
Cod_usuario	Int4	4	Si			Es el código que la herramienta le ha asignado al profesor cuando es creado.

Llave Foránea	Referencia a:	
	Tabla	Llave Primaria
Cod_usuario	usuario	Cod_usuario

◆ TABLA PRÁCTICA.

Campo	Tipo	Longitud	Requerido	Llave Primaria	Default	Descripción
Cod_practica	Int3	3	Si	Si		Es el código que se le es asignado a la práctica que el estudiante realiza.
Hora_practica	time		Si		00:00:00	Es hora de inicio de la práctica.
Cod_estudiante	Int4	4	Si			Es el código que se le asigna al estudiante por la universidad.

Llave Foránea	Referencia a:	
	Tabla	Llave Primaria
Cod_estudiante	estudiante	Cod_estudiante

◆ TABLA DE PROBLEMAS.

Campo	Tipo	Longitud	Requerido	Llave Primaria	Default	Descripción
Cod_problema	Int4	4	Si	Si	autonumérico	identificador correspondiente al problema, que es insertado por el administrador.
problema	longblob		Si			Imagen inicial del problema
Texto_problema	text		Si			Corresponde al enunciado del mismo
Tipo_problema	varchar	30	Si			Es el tipo de problema, generales o específicos por carreras
Cod_subtema	Int3	3	Si			Código del subtema a que corresponde al problema.
nombre	varchar	30	Si			Es el nombre de la imagen del problema.
tipo	varchar	30	Si			Es el tipo de la imagen almacenada en la base de datos.
estado	varchar	12	SI			Describe el estado de los problemas activos o no activos

Llave Foránea	Referencia a:	
	Tabla	Llave Primaria
Cod_subtema	subtemas	Cod_subtema

◆ **TABLA DE PROBLEMAS TEORICOS.**

Campo	Tipo	Longitud	Requerido	Llave Primaria	Default	Descripción
Cod_proble_teor	int	3	Si	Si	autonumérico	identificador correspondiente al problema teórico
Text_proble	text		Si			Texto del problema que corresponde a los temas vistos por la herramienta
tema	varchar	30	Si			Es el tema a que corresponde el problema.
Sub_tema	varchar	60	Si			Es el subtema a que corresponde el problema.

◆ **TABLA DE USUARIO.**

Campo	Tipo	Longitud	Requerido	Llave Primaria	Default	Descripción
Cod_usuario	int4	4	Yes	Yes	autonumérico	Identificador del usuario
Nom_usuario	text	var	No	No		Nombre del usuario
Apell_usuario	text	var	No	No		Apellidos del usuario
clave	text	var	No	No		Login o identificador de acceso del usuario al sistema.
contraseña	text	var	No	No		Password o Clave de acceso del usuario al sistema.
universidad	text	var	No	No		Universidad a la cual corresponde el usuario.
carrera	text	var	No	No		Carrera que cursa o curso el usuario
ocupacion	text	var	No	No		Ocupación del usuario en la actualidad.
sexo	text	var	No	No		Sexo del usuario: Masculino o Femenino
identificacion	text	var	No	No		identificador que referencia el tipo de identificación del usuario
Num_identificacion	text	var	No	No		Numero de Identificación del usuario.
Email	text	var	No	No		Correo Electrónico del usuario
Cod_permiso	char	1	No	No		identificador que referencia el permiso del usuario.

Llave Foránea	Referencia a:	
	Tabla	Llave Primaria
Cod_permiso	permisos	Cod_permiso

ANEXO E

ESCENARIOS DE PRUEBA.

Este anexo presenta las principales pruebas realizadas para comprobar el correcto funcionamiento del sistema. Las pruebas se encuentran organizadas por subsistemas y según el orden en que fueron presentadas en los capítulos 12 Y 13 de la fase de construcción.



Para el desarrollo de estas pruebas la base de datos ha sido poblada con datos ficticios. Para cada una de estos casos también se realizaron pruebas de validación de todos los datos solicitados al usuario.


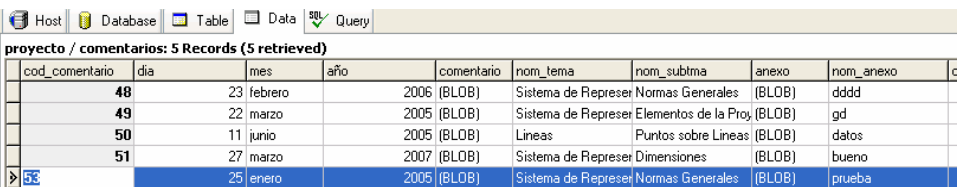

➤ SUBSISTEMA GESTION DE CONSULTAR TEMAS.

Casos de prueba

- ◆ Caso 1: Validar campos no vacíos, al ingresar anexos
- ◆ Caso 2: Validar tipo de archivos permitidos de los anexos
- ◆ Caso 3: Validar que no se creen duplicados
- ◆ Caso 4: Correcto almacenamiento en la base de datos.
- ◆ Caso 5: Validar tipo de usuario permitido.
- ◆ Caso 6: Ver los documentos seleccionados.

Pagina: Ingresar Anexos		
Para esta página se realizarán los casos 1, 2, 3, 4 y 5 por su pertinencia con la misma; el caso 6 no se realiza pues aquí si no en otra pagina.		
Caso 1: Validar campos no vacíos		
Este caso de prueba se realiza en todas las páginas donde el usuario ingresa información, pero solo se muestra para esta página ya que el proceso es el mismo y no es relevante repetirlo.		
Datos de entrada	Clave:	galvao
	Nombre:	felix fernando
	Apellidos:	galvan canchila
	Tema Anexo	Sistema de representación
	Subtema Anexo	Normas Generales
	Nombre Anexo:	
	Fecha /día/mes/año:	25/enero/2005

	Comentario:	Leer con atención
	Anexar el Archivo:	F:\Proyecciones múltiples.doc
Resultados	<p>En esta página, Todos los campos son obligatorios ya que son de vital importancia para el sistema, pero los primeros 3 campos son llenados de forma automática por la herramienta, como todos son obligatorios en caso de no ingresar la información se presenta un mensaje de error como el siguiente que ocurre cuando se deja el campo correspondiente al nombre del anexo vacío.</p> <div><div>Microsoft Internet Explorer</div><div> Por favor, escriba el nombre del anexo.</div><div>Aceptar</div></div>	
Caso 2 : Validar tipo de archivos permitidos		
Datos de Entrada	Clave:	galvao
	Nombre:	felix fernando
	Apellido:	galvan canchila
	Tema Anexo:	Sistema de representación
	Subtema Anexo:	Normas Generales
	Nombre Anexo:	prueba
	Fecha /día/mes/año:	25/enero/2005
	Comentario:	Leer con atención
	Anexar el Archivo:	F:\Temp.zip
Resultados	<p>En este caso para Ingresar los anexos a la base de datos solo se aceptan documentos .pdf o .msword, en caso contrario presenta el siguiente mensaje de error y se devuelve a la página de ingresar anexos.</p> <div><div>Microsoft Internet Explorer</div><div> La extensión del archivo no es correcta. Se permiten archivos .pdf .msword</div><div>Aceptar</div></div>	
Caso 3 : Verificar que no se creen duplicados		
Datos de Entrada	Clave:	galvao
	Nombre:	felix fernando
	Apellido:	galvan canchila
	Tema Anexo:	Sistema de representación
	Subtema Anexo:	Normas Generales
	Nombre Anexo:	prueba
	Fecha /día/mes/año:	25/enero/2005
	Comentario:	Leer con atención
	Anexar el Archivo:	F:\bulma-628[1].pdf
Resultados	<p>Como esta información se incluye solo una vez y para evitar que se creen duplicados, antes de insertar el archivo en la base de datos se hace una consulta para verificar que no existan este documento, de existir se presenta el siguiente mensaje de error y se devuelve a la misma página</p>	

		
Caso 4 : Correcto almacenamiento en la base de datos		
Datos de Entrada	Clave:	galvao
	Nombre:	felix fernando
	Apellido:	galvan canchila
	Tema Anexo:	Sistema de representación
	Subtema Anexo:	Normas Generales
	Nombre Anexo:	prueba
	Fecha /día/mes/año:	25/enero/2005
	Comentario:	Leer con atención
Anexar el Archivo:	F:\Proyecciones múltiples.doc	
Resultados		
	En caso de no ocurrir errores se hace la inserción y se devuelve a la misma página.	
Caso 5 : Validar tipo de usuario permitido		
Resultados	Para este proceso que tiene el software solo les permite a los profesores y al administrador subir los anexos a la base de datos, de lo contrario si otro usuario desea ingresar se presentara el siguiente error.	
		
Caso 6: Ver los documentos seleccionados		

El software les permite a todos los usuarios de este ver los diferentes anexos que los distintos profesores han subido a la base de datos.

Resultados



Después de seleccionarlo se despliega en otra ventana, de esta manera.




➤ SUBSISTEMA GESTION DE ADMINISTRACION.


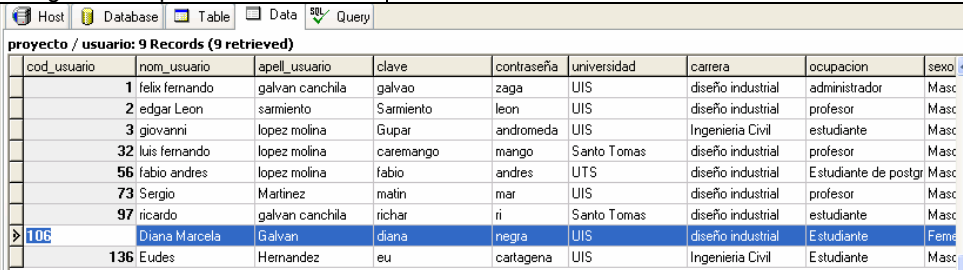
De la misma manera que se realizaron las pruebas para la creación de usuarios en el modulo de administración, de igual forma se realizaron para los diferentes procesos


que el administrador tiene a su cargo, por tal motivo no aparecen en el documento, para no hacerlo monótono.

Casos de prueba

- ◆ Caso 1: Validar campos no vacíos, al ingresar anexos
- ◆ Caso 2 Validar que no se creen duplicados
- ◆ Caso 3 Correcto almacenamiento en la base de datos.
- ◆ Caso 4 Validar tipo de usuario permitido.

Pagina : Crear Usuario		
Para esta página se realizarán los casos 1, 2, 3, 4.		
Caso 1: Validar campos no vacíos		
Datos de entrada	Nombre:	Diana Marcela
	Apellidos:	Galvan
	Sexo:	Femenino
	# Identificación:	35,456,125
	Identificación:	Cedula de Ciudadanía
	Email:	diana@hotmail.com
	Universidad:	UIS
	Carrera:	Fisioterapeuta
	Ocupación:	Fisioterapeuta
	Clave:	diana
	Contraseña:	*****
	Confirmar Contraseña:	*****
	Perfil:	
	Código del Profesor:	
Código del Estudiante:		
Código del Grupo:		
Resultados	<p>En esta página, la mayoría de los campos son obligatorios ya que son de vital importancia para el sistema, los campos # identificación, identificación, Email, ocupación, universidad son campos opcionales, como todos los demás son obligatorios en caso de no ingresar la información se presenta un mensaje de error como el siguiente que ocurre cuando se deja el campo correspondiente al perfil sin seleccionar.</p> <div><div>Microsoft Internet Explorer</div><div><div></div><div>Por favor, escoja el perfil del usuario.</div></div><div>Aceptar</div></div>	
Caso 2 : Verificar que no se creen duplicados		

Datos de Entrada	Nombre:	Diana Marcela
	Apellidos:	Galvan
	Sexo:	Femenino
	# Identificación:	35,456,125
	Identificación:	Cedula de Ciudadanía
	Email:	diana@hotmail.com
	Universidad:	UIS
	Carrera:	Fisioterapeuta
	Ocupación:	Fisioterapeuta
	Clave:	diana
	Contraseña:	*****
	Confirmar Contraseña:	*****
	Perfil:	invitado
	Código del Profesor:	
	Código del Estudiante:	
Código del Grupo:		
Resultados	En este caso para Ingresar un usuario nuevo a la base de datos se verifica que la clave del usuario sean únicas, en caso contrario presenta el siguiente mensaje de error y se devuelve a la página de ingresar usuarios.	
		
Caso 3 : Correcto almacenamiento en la base de datos		
Datos de Entrada	Nombre:	Diana Marcela
	Apellidos:	Galvan
	Sexo:	Femenino
	# Identificación:	35,456,125
	Identificación:	Cedula de Ciudadanía
	Email:	diana@hotmail.com
	Universidad:	UIS
	Carrera:	Diseño industrial
	Ocupación:	Estudiante
	Clave:	diana
	Contraseña:	*****
	Confirmar Contraseña:	*****
	Perfil:	invitado
	Código del Profesor:	
	Código del Estudiante:	
Código del Grupo:		
Resultados		
	En caso de no ocurrir errores se hace la inserción y se devuelve a la misma página.	
Caso 4 : Validar tipo de usuario permitido		

Resultados	<p>Para este proceso que tiene el software solo les permite al administrador crear los distintos usuarios en la base de datos, de lo contrario si otro usuario desea ingresar se presentara el siguiente error.</p> 
------------	--

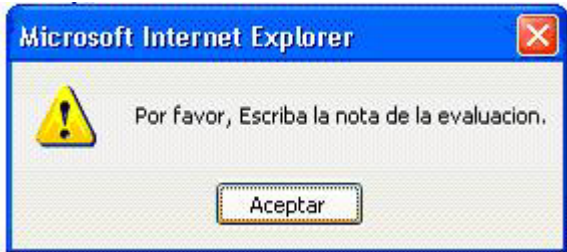
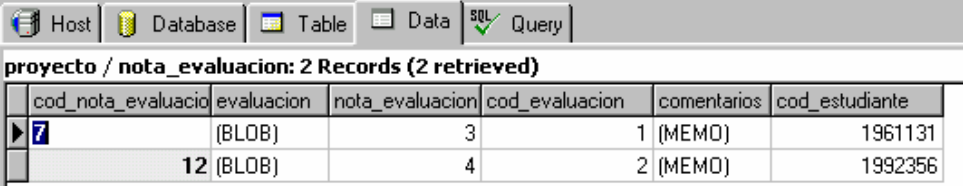
➤ SUBSISTEMA GESTION DE REGISTRO.

La siguiente prueba se realizara con los datos que se encuentran almacenados en las distintas tablas de la base de datos, con el fin de constatar que las diferentes opciones que los profesores tienen a su cargo con la herramienta se pueden realizar, de forma sencilla y rápida.

Casos de prueba

- ◆ Caso 1: Validar campos no vacíos, cuando se tiene que ingresar algún dato.
- ◆ Caso 2 Validar que no se creen duplicados.
- ◆ Caso 3 Correcto almacenamiento en la base de datos.
- ◆ Caso 4 Validar tipo de usuario permitido.

Pagina : Asignar Nota		
Para esta página se realizarán los casos 1, 2, 3, 4.		
Caso 1: Validar campos no vacíos		
Datos de entrada	Evaluación:	2
	Código del Estudiante:	1992356
	Nota:	
	Comentario:	
	Evaluación:	
	Estado:	Pendiente
Resultados	En esta página, la mayoría de los campos son obligatorios ya que son de vital importancia para el sistema, el campo Comentario es opcional, como todos los demás son obligatorios	

	<p>en caso de no ingresar la información necesaria se presenta un mensaje de error como el siguiente que ocurre cuando se deja el campo correspondiente al estado sin seleccionar.</p> 
Caso 2 : Verificar que no se creen duplicados	
Datos de Entrada	Los datos que se tienen para este caso son los mismos que para el anterior.
Resultados	En este caso cuando se ingresa una nota a la base de datos es necesario cambiar el estado de la evaluación, al realizar esta operación se evita que se lleguen a duplicar calificaciones de las evaluaciones, ya que las evaluaciones que se muestran en esta pagina son las evaluaciones pendientes y al haber cambiado su estado ya no aparece.
Caso 3 : Correcto almacenamiento en la base de datos	
Datos de Entrada	Evaluación:
	Código del Estudiante:
	Nota:
	Comentario:
	Evaluación:
Resultados	Estado:
	2
	1992356
	4
	Buen trabajo
	Revisado
Resultados	
	En caso de no ocurrir errores se hace la inserción y se devuelve a la misma página.
Caso 4 : Validar tipo de usuario permitido	
Resultados	<p>Para este proceso que tiene el software solo les permite al profesor dar las diferentes notas a los distintos estudiantes que han realizado alguna evaluación.</p> 