

ESTRATEGIA DIDÁCTICA VIRTUAL PARA APOYAR EL APRENDIZAJE DEL
CONCEPTO DE REPRESENTACIÓN EN PROCESOS DE
DISEÑO DE SOFTWARE

NAYIBE DEL CARMEN RUÍZ CHAGÜI

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO - MECÁNICAS
MAESTRÍA EN INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2013

ESTRATEGIA DIDÁCTICA VIRTUAL PARA APOYAR EL APRENDIZAJE DEL
CONCEPTO DE REPRESENTACIÓN EN PROCESOS DE
DISEÑO DE SOFTWARE

NAYIBE DEL CARMEN RUÍZ CHAGÜI

Trabajo de grado para optar al título de
Magister en Ingeniería de Sistemas e Informática

Director:

SONIA C. GAMBOA SARMIENTO

PhD en educación

Codirector:

LUIS CARLOS GÓMEZ FLÓREZ

MSc. en Informática

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO - MECÁNICAS
MAESTRÍA EN INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2013

DEDICATORIA

A Dios por permitirme que hoy pueda hacer estas dedicatorias:

A mi padre que siempre colocó sus esperanzas en la educación, como su herencia más valiosa y que estoy segura que desde el cielo me acompañó en este proceso.

A mi madre y mis hermanas a quienes debo lo que soy.

A Jesús Enrique y Marianita mis grandes tesoros y ángeles que me ha dado Dios, por su apoyo y el tiempo que me regalaron para llevar a cabo este proyecto.

A mi esposo Jesús por ser un gran compañero en esta aventura y por el apoyo incondicional, y sin el cual no hubiera sido posible culminar.

Nayibe Ruiz Ch.

AGRADECIMIENTOS

A la profesora Sonia Cristina Gamboa Sarmiento, por sus enseñanzas, dedicación, paciencia y ser mi guía en el desarrollo de este proyecto. Además por darme la oportunidad de entrar en el campo de la investigación y hacer mi primera publicación.

Al profesor Luis Carlos Gómez Flórez, por creer siempre en mí, enseñarme a ser una mejor profesional, amar la Ingeniería de Sistemas y darme su voto de confianza incondicional y el espacio para desarrollar la investigación.

A la profesora Martha Vitalia Corredor Montagut, quien con su trabajo ha sido de gran ayuda en el desarrollo de este proyecto y un ejemplo a seguir.

Al profesor Fernando Ruíz por su confianza y respaldo.

A mis amigos Mayda González, Ernesto Galvis y Eliecer Pineda, porque me acompañaron incondicionalmente en este proceso.

A Jorge Iván Torres, por su apoyo moral y quien siempre respondió a mis dudas en el aspecto técnico.

A los integrantes del grupo de investigación STI – con los cuáles compartí los estudios de maestría, en especial a Pedro Javier Trujillo, Yesid Olave, Marcela Sequeda, Marlene Guerrero, Marú, Laura Pinto, Luis Fernando Sierra, Diego Mantilla, Daniel Rubiano, Carlos Humberto Carreño y Alexander Bustamante.

A la Escuela de Ingeniería de Sistemas e Informática por brindarme el espacio para desarrollar el proyecto.

Y a todos los que de una forma directa o indirecta colaboraron y creyeron que esto se lograría; aunque no aparezcan en esta página les estoy inmensamente agradecida por compartir esta aventura con altibajos e interrupciones.

TABLA DE CONTENIDO

INTRODUCCIÓN.....	12
1. DESCRIPCIÓN DEL PROYECTO.....	14
1.1 PLANTEAMIENTO DEL PROBLEMA.....	14
1.2 JUSTIFICACIÓN.....	19
1.3 OBJETIVOS.....	20
1.3.1 Objetivo General.....	20
1.3.2 Objetivos Específicos.....	20
2. MARCO CONCEPTUAL.....	22
2.1 MODELADO.....	22
2.1.1 Modelo.....	22
2.2 REPRESENTACIÓN.....	23
2.3 DIDÁCTICA.....	25
2.3.1 Transposición Didáctica.....	26
2.3.2 Dispositivo Didáctico.....	28
2.3.3 Estrategia de Portafolios.....	29
2.4 HEURÍSTICA.....	31
2.5 TEORÍAS DE APRENDIZAJE.....	32
2.5.1 El Constructivismo.....	33
2.5.2 Teoría de Jean Piaget.....	34
2.5.3 Teoría de Vigotsky.....	35
2.5.4 Teoría de Ausubel.....	36
2.6 INGENIERÍA DE SOFTWARE.....	37
2.6.1 Modelos de proceso de software.....	38
2.6.2 Actividades de desarrollo de Ingeniería del Software.....	40
2.7 LENGUAJE DE MODELADO UNIFICADO – UML.....	41
2.8 TECNOLOGÍAS DE INFORMACIÓN Y EDUCACIÓN.....	42
2.8.1 e-learning.....	42
2.8.2 Plataformas <i>e-learning</i>	44
2.9 INFORMÁTICA EDUCATIVA.....	58
3. ESTRATEGIA DIDÁCTICA.....	62
3.1 DETERMINACIÓN DE ELEMENTOS DE REPRESENTACIÓN.....	62
3.2 HEURÍSTICOS.....	77
3.3 SELECCIÓN DE LAS TIC.....	91
4. DISPOSITIVO COMPUTACIONAL.....	96
4.1 PORTAFOLIO DIGITAL - <i>UISport</i>	96
4.2 EXPERIENCIA CON <i>UISport</i>	109
5. CONCLUSIONES.....	111
6. RECOMENDACIONES.....	114
7. REFERENCIAS BIBLIOGRÁFICAS.....	115
ANEXOS.....	120
ANEXO A. INSTALACIÓN DE <i>UISport</i>	120
ANEXO B. DESCRIPCIÓN DE CASOS DE USO DE <i>UISport</i>	124

LISTA DE FIGURAS

Figura 1. Cuadrado Mágico	24
Figura 2. Actividades del desarrollo de software orientado a objetos	62
Figura 3. Modelos del sistema	63
Figura 4. Ingeniería de Requerimientos	64
Figura 5. Tipos de escenarios.....	65
Figura 6. Tipos de Relaciones	66
Figura 7. Plantilla RAD – Documento de Análisis de Requerimientos	68
Figura 8. Elementos del Sistema Propuesto	69
Figura 9. Submodelos de la etapa de análisis.....	70
Figura 10. Actividades del análisis	71
Figura 11. Productos del diseño del sistema.....	75
Figura 12. Plantilla SDD – Documento del diseño del sistema.....	76
Figura 13. Estadísticas de Moodle	92
Figura 14. Bloque <i>e-portafolio UISport</i>	97
Figura 15. Pantalla inicial de UISport	98
Figura 16. Diagrama de Casos de Uso General de <i>UISport</i>	99
Figura 17. Caso de Uso – <i>ObtencionyAnálisis</i>	102
Figura 18. Caso de Uso – <i>DefinirCasoUso y DefinirEscenario</i>	103
Figura 19. Caso de Uso - <i>Diseño</i>	103
Figura 20. Caso Uso – <i>Evidencia</i>	103
Figura 21. Caso de Uso - <i>Vista</i>	104
Figura 22. Caso de Uso – <i>Docente</i>	105
Figura 23. Botón de Heurístico	105
Figura 24. Heurístico de análisis	106
Figura 25. Heurístico para el diseño	106
Figura 26. Pestaña de Evidencias	108
Figura 27. Pestaña del Docente.....	108

LISTA DE TABLAS

Tabla 1. Comparación entre LMS y LCMS.....	46
Tabla 2. Herramientas de las plataformas <i>e-learning</i>	49
Tabla 3. Plataformas privativas.....	55
Tabla 4. Plataformas de distribución libre	56
Tabla 5. Requerimientos No funcionales	67
Tabla 6. Criterios de diseño	72
Tabla 7. Heurística de Abbott	77
Tabla 8. Relación de heurísticos con las diferentes actividades de procesos de diseño de software.....	79
Tabla 9. Funcionalidades del dispositivo <i>UISport</i>	100

RESUMEN

TÍTULO: ESTRATEGIA DIDÁCTICA VIRTUAL PARA APOYAR EL APRENDIZAJE DEL CONCEPTO DE REPRESENTACIÓN EN PROCESOS DE DISEÑO DE SOFTWARE*

AUTOR: RUIZ CHAGUI, Nayibe del Carmen**

PALABRAS CLAVE: Ingeniería de software, representación, modelado, didáctica, portafolio digital, Moodle

DESCRIPCIÓN: El diseño de software, como actividad de ingeniería, busca representar fenómenos de la realidad plasmados en modelos para ser llevados a sistemas de información que respondan a requerimientos determinados; lo anterior requiere aprender un conjunto de habilidades de modelado y notaciones correspondientes. Para ello es necesario diseñar estrategias didácticas para el aprendizaje de tales conceptos y habilidades por parte de estudiantes de ingeniería de sistemas.

Las tecnologías de información y comunicación (TIC), por su parte, se han constituido en un valioso recurso de apoyo a los procesos de enseñanza y aprendizaje; entre ellos los entornos virtuales de aprendizaje (VLE), los cuales permiten implementar estrategias didácticas para el aprendizaje autónomo de conceptos y habilidades. A partir de los principios teóricos del constructivismo, los planteamientos de Piaget, Vygotsky y Ausubel, y de los heurísticos propuestos por Bruegge & Dutoit que soportan el modelado de las actividades de desarrollo de software, se desarrolló un dispositivo computacional basado en la estrategia didáctica portafolio e incorporado en la plataforma Moodle.

El portafolio digital busca apoyar objetivos de aprendizaje enmarcados en el concepto de *representación*, mediante la documentación del desarrollo de un proyecto de clase; en él los estudiantes registran las evidencias obtenidas en las etapas de obtención de requerimientos, análisis y diseño del proyecto software. Además el profesor cuenta con una herramienta para realimentar, gestionar el proceso de aprendizaje y fomentar las habilidades de modelado, comunicación y el trabajo en equipo en los estudiantes.

* Trabajo de investigación

** Facultad Fisicomecánicas, Maestría en Ingeniería de Sistemas e Informática. Director: GAMBOA SARMIENTO, Sonia Cristina. Codirector: GOMEZ FLOREZ, Luis Carlos.

SUMMARY

TITLE: VIRTUAL TEACHING LEARNING STRATEGY TO SUPPORT THE CONCEPT OF REPRESENTATION IN PROCESS DESIGN SOFTWARE

AUTHOR: RUIZ CHAGUI, Nayibe del Carmen**

KEYWORDS: Software engineering, representation, modeling, teaching, digital portfolio ,Moodle

DESCRIPTION: The design of software such as engineering activity, seeks to represent reality phenomena embodied in models to be taken to information systems that meet specific requirements, this requires learning a skill set of modeling notations and relevant. This requires designing teaching for learning such concepts and skills by students of systems engineering strategies.

The information and communication technologies (ICT), meanwhile, have become a valuable resource to support teaching and learning, including virtual learning environments (VLE), which allow to implement teaching strategies autonomous learning of concepts and skills. From the theoretical principles of constructivism, the ideas of Piaget, Vygotsky and Ausubel, and heuristic proposed by Bruegge & Dutoit supporting modeling activities of software development, a computational device based on the teaching strategy developed portfolio and incorporated into the Moodle platform.

The digital portfolio seeks to support learning objectives within the concept of representation, by documenting the development of a class project, students recorded him the evidence obtained in the stages of requirements elicitation, analysis and design of software project. Also, the teacher has a tool to feedback, manage the learning process and encourage modeling skills , communication and teamwork in students.

* Trabajo de investigación

** Facultad Fisicomecánicas, Maestría en Ingeniería de Sistemas e Informática. Director: GAMBOA SARMIENTO, Sonia Cristina. Codirector: GOMEZ FLOREZ, Luis Carlos.

INTRODUCCIÓN

Esta investigación se ubica en el campo de la informática educativa, la cual busca dar un buen uso de las tecnologías de información y comunicación (TIC) en la educación. El objetivo fue implementar la estrategia didáctica de portafolios en un dispositivo computacional instalado en el entorno virtual de aprendizaje Moodle, para apoyar el aprendizaje del concepto de representación en procesos de diseño de software, necesario para la creación de software.

El ingeniero de sistemas para resolver situaciones problema realiza abstracciones de la realidad y hace representaciones de esta que son plasmadas en modelos para ser llevados a sistemas de información o de gestión de conocimiento. Para disminuir la complejidad de un problema se requiere realizar representaciones adecuadas y que describan dicha realidad lo mejor posible en un lenguaje simbólico a través de modelos.

Las metodologías orientadas a objetos para desarrollo de software facilitan la definición del dominio del problema y de la solución con el fin de disminuir la complejidad inherente al software. En el mercado existen herramientas informáticas que apoyan el modelado de software como lo es UML (Lenguaje Unificado de Modelado) pero se requiere que quién realiza los diagramas UML y su uso eficiente, no sólo posea el conocimiento de esto sino que desarrolle una habilidad.

En este contexto el concepto de *representación* no sólo debe ser aprendido sino que debe ser una habilidad a desarrollarse dentro de la Ingeniería de Software. La universidad como entorno social que produce y enseña conocimiento es el espacio propicio para desarrollar proyectos de investigación como el que se describe en el presente documento. Este se encuentra estructurado de la siguiente manera:

En el capítulo 2 se describe la investigación propuesta en cuanto a la situación problemática, se presenta la justificación del proyecto y los objetivos que orientan el desarrollo de la investigación.

Los fundamentos teóricos sobre los que se soporta esta investigación se exponen en el capítulo 3, el cual se ha subdividido de la siguiente manera: inicialmente se expone lo concerniente al objeto de estudio de esta investigación, a saber, los conceptos modelado y representación; luego se presentan los elementos pedagógicos sobre los que se fundamenta la estrategia didáctica; posteriormente se propone una revisión sobre el enfoque de ingeniería de software que se asume, y del Lenguaje Unificado de Modelado (UML) como notación correspondiente; por último se expone lo referente a los recursos utilizados de las TIC.

El capítulo 4 muestra el proceso de diseño de la estrategia didáctica, en el que se engranan los enfoques de pedagogía, didáctica y TIC, y se materializan en un dispositivo computacional.

Los procesos de diseño e implementación de este dispositivo se describen en el capítulo 5, cuya descripción se hace siguiendo el Proceso Unificado como metodología. Además se ilustra la experiencia realizada con el dispositivo en el curso de Ingeniería de software I del programa de pregrado de Ingeniería de Sistemas e Informática, durante el primer semestre del 2012.

Por último se presentan las conclusiones y recomendaciones resultado del desarrollo de esta investigación.

1. DESCRIPCIÓN DEL PROYECTO

1.1 PLANTEAMIENTO DEL PROBLEMA

Los cambios que se han dado a nivel social y cultural a partir de los años cincuenta, sobre todo en los países desarrollados y que han tenido un gran impacto en la forma de producción del conocimiento científico, se deben en gran parte a lo que Lyotard ha denominado *condición postmoderna*, y que describe como “El estado de la cultura después de las transformaciones que han afectado a las reglas de juego de la ciencia, de la literatura y de las artes a partir del siglo XIX” (1987, pág. 6). Esta condición también ha suscitado que las sociedades aborden y vean la realidad a partir del vertiginoso auge de las TIC, que ha afectado la forma en que se realizan las actividades, las formas de comunicarse, el lenguaje utilizado, las maneras de hacer negocios, transacciones, el volumen de datos e información, el almacenamiento y procesamiento de los mismos, entre otros (cf. Gamboa Sarmiento, 2010, pág. 68).

Por su parte la producción de conocimiento científico pasó de ser generado principalmente por individuos a ser producto del trabajo mancomunado de colectividades; así, el volumen de información que se requiere para producirlo ha hecho que se consideren nuevas formas amparadas por las TIC para gestionarlo.

El uso cada vez mayor de las TIC como soporte a los procesos de enseñanza y aprendizaje en el aula, implican unas condiciones para los actores involucrados en el proceso, de manera que el profesor debe revisar, diseñar, administrar y guiar las actividades que garanticen el cumplimiento del propósito de aprendizaje, así como hacer seguimiento y realimentación de las mismas; los estudiantes, por otra parte, como protagonistas de su aprendizaje deben participar individualmente, y aportar y colaborar al resto de participantes, lo cual implica invertir y administrar el tiempo sujeto a unas actividades programadas.

Las TIC, tales como la Internet, las redes sociales, las plataformas que apoyan el aprendizaje (*e-learning*) configuran, en la actualidad el panorama de las

herramientas que apoyan los procesos de enseñanza y aprendizaje; según Zabalza (2004) en el proceso educativo tanto profesor como estudiante juegan un rol diferente, por una parte el profesor pasa a tener un rol de diseñador de ambientes de aprendizaje, a ser un guía y facilitador; y por otra parte, el estudiante toma un rol más activo como protagonista de dichos procesos.

La universidad, como entorno social y centro que privilegia, no solo la producción de conocimiento, sino su enseñanza, es un espacio propicio para que se reflexione sobre el conocimiento y se genere un valor agregado que contribuya al desarrollo social y económico de la sociedad. Uno de los propósitos en la didáctica es pensar las maneras en que el conocimiento científico pueda ser enseñado; según (Vargas & Gamboa, 2008, pág. 63) en la *condición postmoderna* el diseño de estrategias didácticas consiste en diseñar ambientes de aprendizaje: espacios en los que profesores y estudiantes interactúen con elementos pensados para favorecer el aprendizaje de temas específicos.

El Proyecto Institucional de la Universidad Industrial de Santander –PI-UIS– (2000) dice en su visión “[...] La UIS es actor principal del desarrollo económico y social de la región y ejemplo de democracia, convivencia, autonomía y libertad responsable. Es lugar de consulta sobre las tendencias y desarrollos en el campo de las ciencias, los avances tecnológicos, las necesidades del mundo del trabajo y los deseos de bienestar de la comunidad”, esto contempla la enseñanza, la incorporación y el uso de las TIC para potenciar sus procesos académicos. La Alta Dirección de la UIS ha considerado, como una de las áreas estratégicas, la incorporación de las TIC para fortalecer los procesos de docencia, investigación y extensión.

Así mismo, la Escuela de Ingeniería de Sistemas e Informática de la UIS en su Proyecto Educativo, en concordancia con la misión de la Universidad, expresa su voluntad de ser “[...] una organización que tiene como propósito la formación de personas de alta calidad ética, política y profesional; la generación y adecuación

de conocimientos; la conservación y reinterpretación de la cultura y la participación activa liderando procesos de cambio por el progreso y mejor calidad de vida de la comunidad” (2000), por lo que busca formar profesionales reflexivos, proactivos y que brinden soluciones a las problemáticas de la región y, además, define como objeto de estudio de la Ingeniería de Sistemas, la información y el conocimiento. Así, el Ingeniero de sistemas se ocupa de proponer modelos o representaciones del mundo real a partir de las cuales se proponga sistemas de información o de gestión de conocimiento que satisfagan situaciones problemáticas.

En este contexto, el concepto de *representación* en procesos de desarrollo de *software* tendría que ser aprendido, no solamente como un conocimiento propio de la *ingeniería del software*, sino como una habilidad que debe ser desarrollada. El mercado ofrece herramientas informáticas que permiten hacer *modelado* bajo los estándares de representación que ofrece la notación UML¹, pero su uso eficiente, la elaboración de diagramas UML, requiere de que quien los realiza tenga tanto conocimientos como habilidades de modelado que deben ser aprendidas en los espacios académicos diseñados para este fin, de manera que un estudiante frente a una situación problemática, lleve a cabo una adecuada *representación* de dicha situación que pueda ser plasmada en un modelo, el cual describe la situación en ese lenguaje simbólico. Es necesario, entonces, diseñar ambientes didácticos en los cuales se propicie el aprendizaje tanto de conceptos como de habilidades que le permita realizar abstracciones adecuadas y plantear modelos que correspondan a situaciones reales.

El uso de ambientes o entornos virtuales de aprendizaje –VLE²– dentro de los procesos de enseñanza cada vez es mayor; éstos cuentan con servicios que permiten gestionar cursos, gestionar usuarios, hacer seguimiento del aprendizaje, generar informes, realizar y gestionar evaluaciones, realizar foros de discusión,

¹ Lenguaje Unificado de Modelado

² *Virtual Learning Environment*.

entre otros; permiten diseñar estrategias didácticas pensadas para el aprendizaje de temáticas específicas y con unos propósitos claros, apoyados por las TIC y guiados por la experiencia del profesor, hace que se cuente con la posibilidad de que los estudiantes desarrollen las competencias necesarias para las cuales fueron diseñados (Corredor, Arbeláez & Perez, 2010, pág. 31).

En la UIS cada vez es mayor el uso de VLE en el acompañamiento de la educación presencial; prueba de ello es el desarrollo de proyectos como *e-scen@ric-uis* y *El portal del profesor* promovidos desde el CENTIC³, el desarrollo del proyecto titulado “Diseño instruccional basado en competencias, mediado por TIC para la asignatura Ingeniería del software del programa académico de Ingeniería de Sistemas e Informática” (Carreño, 2007), como parte del macroproyecto *ProSPETIC* dirigido por Clara Inés Peña de Carrillo y concebido en el año 2005, en el que se realizó el diseño y el correspondiente desarrollo de objetos virtuales de aprendizaje para la temática *Métricas del software de la asignatura Ingeniería del Software I*. La herramienta *Meiweb*, dirigida desde el 2006 por el profesor Manuel Guillermo Flórez y que se encuentra en su cuarta versión, permite administrar algunas asignaturas de la Escuela de Ingeniería de Sistemas e Informática. Por otra parte la herramienta desarrollada por el grupo de software Calumet y administrada por el profesor Luis Ignacio González fue creada en el año 2008 para mantener en comunicación a los usuarios, administrar archivos, contenidos de asignaturas y proyectos de grado de las escuelas de Ingeniería de Sistemas, Geología y Petróleos que actualmente hacen uso de ella. El uso de la plataforma *Moodle* por parte de Cededuis⁴, de la escuela de Ingeniería Eléctrica y Electrónica, la Facultad de Ciencias y otras unidades académicas. Otros proyectos realizados en la UIS consisten en propuestas para desarrollar ambientes virtuales de aprendizaje (AVA) y objetos virtuales de

³ Centro de Tecnologías de Información y Comunicación de la UIS

⁴ Centro para el desarrollo de la docencia en la UIS

aprendizaje (OVA) para el apoyo de enseñanza de diferentes asignaturas a nivel de primaria, secundaria y pregrado.

En el contexto nacional se han desarrollado propuestas como la realizada por el profesor Zapata Jaramillo (2007), de la Universidad Nacional, sede Medellín, en la que plantea para la enseñanza de la Ingeniería del Software el uso de técnicas, juegos y recursos que no utilizan computadoras. En el contexto internacional se tiene la llevada a cabo en la Universidad de las Ciencias Informáticas (UCI) en Cuba, donde se ha creado toda una estrategia didáctica para enseñar *Ingeniería del Software* a través de la televisión.

La presente investigación se ubica en el campo de la informática educativa y, tiene como fin diseñar una estrategia didáctica que sea implementada en un dispositivo didáctico computacional que apoye el aprendizaje del concepto de *representación* en procesos de diseño de software, necesario para llevar a cabo procesos de creación de software, usando como espacio la asignatura Ingeniería del Software I del programa de Ingeniería de Sistemas e Informática, en un ambiente de aprendizaje que permita engranar la pedagogía, la didáctica y las TIC. Para ello se apoyará en los principios de la teoría de aprendizaje constructivista que busca que el estudiante construya su propio conocimiento. Dentro de estos fundamentos se tuvieron en cuenta, en primer lugar, los planteamientos de Piaget según los cuales la formación de estructuras mentales tiene en cuenta cómo el sujeto va equilibrando su conocimiento, en los planteamientos de Vygotsky, que hace referencia a que es en la zona de desarrollo próximo donde el estudiante adquiere su conocimiento a través de su experiencia y de la ayuda de otros o de sistemas simbólicos como las TIC; y, en tercer lugar están los postulados de Ausubel que plantea la necesidad de ofrecer experiencias de forma que los estudiantes relacionen los conocimientos previos con la nueva información para que puedan lograr un aprendizaje significativo.

Este dispositivo será publicado en un VLE de distribución libre. Los VLE ofrecen potencialidades que deben ser reflexionadas no sólo desde el punto de vista tecnológico, sino desde el punto de vista pedagógico, ya que tales tecnologías son los medios o recursos y no el fin de la formación. Además, se espera que la divulgación de esta investigación muestre un panorama y algunas características a tener en cuenta en proyectos similares que busquen apoyar cursos que trabajan en modalidad presencial apoyados por las TIC.

Se espera que la divulgación de esta investigación muestre un panorama y algunas características a tener en cuenta en proyectos similares que busquen apoyar cursos que trabajan en modalidad presencial apoyados por las TIC.

La investigación busca resolver la pregunta: ¿Cómo puede representarse computacionalmente una estrategia didáctica que permita el aprendizaje del concepto de *representación* en procesos de diseño de software?

1.2 JUSTIFICACIÓN

La UIS en su proyecto Institucional y en sus programas de apoyo a la investigación y extensión plantea ampliar su base de conocimientos para mejorar los procesos académicos, por ello se hace pertinente emprender proyectos de investigación, que tengan como propósito la incorporación de TIC como apoyo en los programas académicos. Contar con apoyo académico y económico y con una infraestructura tecnológica permite que dichos proyectos puedan llevarse a cabo.

Los profesores y estudiantes de los cursos relacionados con diseño de software son beneficiarios directos de los productos obtenidos a través de esta investigación, pues podrán contar con herramientas TIC construidas para soportar los procesos de aprendizaje, de manera que sea posible alcanzar los objetivos propuestos en la asignatura, y lograr en el estudiante un aprendizaje significativo que contribuya a su formación como profesional.

En concordancia con el Proyecto Educativo de la Escuela de Ingeniería de Sistemas e Informática (UIS, 2008, pág. 9), realización de este proyecto de investigación contribuye a desarrollar en los estudiantes competencias para plantear soluciones a problemas en un contexto real y por otra parte mejorar el quehacer del profesor, mediante un uso adecuado de las TIC.

El desarrollo de proyectos de investigación de este tipo busca contribuir a desarrollar en los estudiantes competencias que les permitan plantear soluciones a situaciones problemáticas en un contexto real y, por otra parte, mejorar el quehacer profesional, haciendo un uso adecuado de las TIC.

Esta propuesta de investigación se realiza en el marco del proyecto “Desarrollo y validación de una estrategia de aprendizaje basada en el uso de plataformas virtuales para el curso de Ingeniería del Software I”, financiado por la Vicerrectora de Investigación y Extensión de la UIS –VIE– (Código 5553), lo cual le da un respaldo y es un espacio propicio para llevar a feliz término dicha investigación.

1.3 OBJETIVOS

1.3.1 Objetivo General

Diseñar una estrategia didáctica que apoye el aprendizaje del concepto de *representación* en procesos de diseño de software, para ser implementada en un dispositivo computacional que opere en un ambiente de aprendizaje virtual.

1.3.2 Objetivos Específicos

- Caracterizar elementos y actividades propios del concepto de *representación* en procesos de diseño de software.
- Diseñar una estrategia didáctica bajo el enfoque constructivista para el aprendizaje de elementos y actividades propios del concepto de *representación* en procesos de diseño de software.

- Definir las TIC que permitan implementar la estrategia didáctica concebida como ambiente de aprendizaje virtual.
- Diseñar el dispositivo computacional que materialice la estrategia didáctica propuesta para el aprendizaje del concepto de representación en procesos de diseño de software, usando los elementos con los que cuenta un VLE.
- Validar la estrategia didáctica mediante el desarrollo del dispositivo computacional diseñado.

2. MARCO CONCEPTUAL

2.1 MODELADO

Una de las formas de comprender la realidad es a través de los modelos, éstos son útiles para manejar la complejidad de los problemas. “La ingeniería de software es una actividad de modelado” (Bruegge & Dutoit, Object-oriented Software Engineering, 2010, pág. 5), esta se realiza cuando se diseña un sistema. La ciencia a través del modelado se contesta preguntas específicas acerca de un conjunto de fenómenos, para ello desarrolla abstracciones que le permiten comprenderlos y clasificarlos en conceptos, es decir, encontrar propiedades comunes a dichos fenómenos (cf. Bruegge & Dutoit, pág. 6).

2.1.1 Modelo.

Es una *representación* abstracta de un sistema a cierto nivel que toma los aspectos relevantes, omitiendo detalles que no hacen parte del contexto, esto depende de la finalidad del modelo.

Los modelos se componen de la semántica que contiene toda la información que da significado al modelo y, la notación que es la presentación visual de este, con el fin de que pueda ser comprensible y verificable por las personas (Rumbaugh, Jacobson, & Grady, 2004, pág. 15).

Un modelo es una *representación* simplificada de la realidad en la que aparecen algunas de sus propiedades, un ejemplo de ello es el globo terráqueo, el cual es la manera más exacta de representar al planeta Tierra, pero es menos práctico que un mapa, a la hora de usarse en geografía.

En el software los diseñadores requieren comprender el dominio del problema, dar solución a este y comprender el ambiente en que operará el sistema. Por lo tanto modelar es la acción que permite hacer una descripción lo más exacta posible de un sistema y de las actividades llevadas a cabo en él. “Un modelo es una

descripción de la estructura genérica y del significado de un sistema” (Rumbaugh, Jacobson, & Grady, 2004, pág. 17). El modelo toma los aspectos relevantes de la realidad para resolver un problema e ignora los aspectos no esenciales, a esta capacidad para “encapsular y aislar la información del diseño y ejecución” se le llama *abstracción* (Joyanes Aguilar, 2000, pág. 10).

La *abstracción* es esencial para diseñar software, es la clasificación de los fenómenos en conceptos. El modelado es el desarrollo de abstracciones que sirven para contestar preguntas específicas acerca de un conjunto de fenómenos.

Cuando un proceso es modelado, con ayuda de una representación gráfica (diagrama de proceso), es posible apreciar, con facilidad, las relaciones existentes entre las actividades, las cuales no son explícitas en la realidad, analizar cada actividad, definir puntos de unión entre procesos, así como identificar los subprocesos comprendidos.

Los modelos en el diseño de un sistema software (Rumbaugh, Jacobson, & Grady, 2004, págs. 11-12) permiten:

- Tener un conocimiento claro del sistema a desarrollar por parte de todos los involucrados y determinar los requerimientos de este.
- Proponer diferentes soluciones a un bajo costo antes de realizar el desarrollo final.
- Bosquejar diversos diseños de acuerdo con las vistas que se requieran del sistema, lo cual permite generar los correspondientes productos o artefactos a obtener, tales como clases, procedimientos, interfaces, bases de datos, entre otros.

2.2 REPRESENTACIÓN

Etimológicamente procede del latín *representatio*, que es la acción y efecto de hacer presente con figuras o mediante palabras algo, por lo tanto, puede ser una idea o imagen que sustituye a la realidad.

La anterior concepción fue aclarada por Greco en 1995, que la definía como el hacer presente o traer de nuevo algo que no está literalmente, esto implica volver a presentar un suceso y recrear el escenario donde éste se da. Esta acción consta de tres elementos: el humano que realiza el acto y los procesos mentales a través del cerebro, el mundo externo que comprende la naturaleza y, el nivel de la *representación*, el cual se constituye a partir de la relación entre los dos anteriores. (cf. Castellaro, 2011, págs. 63-64)

Uno de los primeros pasos para resolver un problema es la creación de una *representación* de este. Una *representación* del problema “es una estructura cognitiva que corresponde a un problema dado, construido por el solucionador sobre la base de conocimientos relacionados con el dominio y su organización” (Chi, Feltovich, & Glaser, 2010, pág. 121). Esto es llamado espacio del problema, donde se encuentran las posibles soluciones a éste.

Un ejemplo de lo anterior es la estrategia mostrada por Simon (2006, pág. 158) para resolver el juego “*scrabble* numérico”; en vez de tomar nueve cartas, es pensar en un cuadrado mágico donde hay números, del uno al nueve y donde cada fila, columna o diagonal, formada por una terna de números, deben sumar quince, ver la figura 1. Entonces, puede verse que al cambiar de *representación* se hace más fácil el juego ya que para algunos, el juego de tres en línea es más conocido.

Figura 1. Cuadrado Mágico

4	9	2
3	5	7
8	1	6

Los problemas pueden ser descritos en forma verbal a través del lenguaje natural o en lenguaje matemático, si son problemas físicos se pueden representar mediante planos, mapas, dibujos de ingeniería, prototipos o modelos

tridimensionales (Simon, pág. 161). Los problemas relacionados con actividades o acciones pueden representarse con diagramas de flujo y programas; y los sistemas software pueden ser representados a través de un lenguaje de modelado como UML. Por lo tanto, buscar una solución para un problema significa representarlo (Amarel, 1996, págs. 112-114).

Gallego (1998, págs. 168-172) sostiene que cada individuo posee unos conocimientos previos y al enfrentarse a una realidad nueva la analiza, clasifica, selecciona y transforma esas representaciones de la realidad en conocimiento nuevo que más adelante se conforma en saberes, que se relacionan con las concepciones previas, lo que favorece la creación de jerarquías de conocimiento. Lo anterior es supremamente importante a la hora de establecer una estrategia pedagógica debido a que en toda experiencia educativa hay que identificar y tener en cuenta los presaberes de los estudiantes.

2.3 DIDÁCTICA

Etimológicamente proviene del verbo *didaskhein*, que significa enseñar, instruir, explicar. Su esencia es el estudio de los procesos de enseñanza y aprendizaje, que pretende la formación y el desarrollo instructivo-formativo de los estudiantes; busca la reflexión y el análisis de los procesos de enseñanza y aprendizaje y de la docencia. Junto con la pedagogía, busca la explicación y la mejora permanente de la educación y de los hechos educativos y se ocupan del qué, el para qué y el cómo enseñar. Algunos de los componentes que juegan un papel importante en el logro de los propósitos de la didáctica son el profesor, el estudiante, el contexto del aprendizaje y el currículo.

Juan Amós Comenio llamado Comenius y conocido como el padre de la Pedagogía, introdujo el término como sustantivo en 1640 para designar “el arte de enseñar”, es decir, el conjunto de medios y procedimientos que buscan conocer, saber algo, generalmente una ciencia, una lengua, un arte. En su obra maestra

Didáctica Magna (2003) publicada a mediados del siglo XVII colocó los cimientos para la educación y la enseñanza, entre los cuales se encuentran:

- El aprendizaje se da gradualmente de lo fácil a lo complejo y para que sea exacto y perdure debe empezar a través de los sentidos.
- La educación debe ser universal “Enseñar todo a todos y totalmente”, usando la lengua materna y eliminando el castigo físico.
- En la enseñanza se ha de mostrar la utilidad de lo que se trabaje, propone la enseñanza mutua y la autonomía en el aprendizaje.

Según Vargas & Gamboa (2008, pág. 55), en la didáctica en la condición postmoderna se dan dos cambios. Por una parte, se va de la enseñanza al aprendizaje y, por otra parte, del profesor instructor al orientador.

Los espacios donde se materializa la intención didáctica tienen los siguientes elementos (cf. Vargas Guillén & Gamboa Sarmiento, 2008, pág. 63):

- Una intención libre de enseñar por parte del profesor.
- La motivación por aprender que tiene el estudiante de aprender.
- El *background* del profesor y los conocimientos previos del estudiante.
- La temática a desarrollar en el aprendizaje
- Los dispositivos didácticos o de enseñanza.
- La verificación del aprendizaje por parte del estudiante y gestionada por el profesor.

2.3.1 Transposición Didáctica.

A partir de Verret (1975, pág. 139) que define la didáctica como “*La transmisión de aquellos que saben a aquellos que no saben. De aquellos que han aprendido a aquellos que aprenden*”; se da pie a estudiar la transformación del “objeto de saber a enseñar hasta convertirse en un objeto de enseñanza” y aparece el concepto de *Transposición didáctica* retomado por el didacta de la matemáticas Chevallard (1998, pág. 16), y que se refiere al conjunto de transformaciones de objetos del saber seleccionados como objetos enseñables y esto requiere de una

transformación previa, en la cual primen estos tres aspectos: (1) el *logro*, es decir en el saber transmitido se hace una depuración de aspectos que no son relevantes para el estudiante tales como ensayos, intentos fallidos en el proceso de investigación, (2) la *continuidad* ya que no se tienen en cuenta las interrupciones, lo cual supone una trayectoria con éxito en el tiempo de las investigaciones y (3) la *síntesis* que permite tomar lo más relevante de la investigación (Verret, 1975, pág. 141).

Para Verret no todos los saberes son escolarizables, debido a que se debe fraccionar el saber científico en saberes más especializados para su enseñanza y muchas veces esto descontextualiza (Halté, 1998, pág. 173) y no le permite tener una trayectoria histórica de dicho saber; esto ha sido reafirmado por didactas de otras disciplinas tales como lo menciona Calliot (1996, pág. 229) en las ciencias sociales donde afirman que los saberes no son sólo los dados por una comunidad científica sino que son contruidos por la universidad y/o las necesidades de la sociedad; Pitetjean (1998, pág. 25) lo plantea en el aprendizaje del idioma francés en donde los saberes enseñables no se pueden reducir sólo a los saberes escolares, ya que las competencias que se desarrollan en el aprendizaje de una lengua dependen de la interacción social, al hablarla, comprenderla, escribirla y leerla; y para Tozzi (1995, pág. 240) en la enseñanza de la filosofía donde ésta hace parte de las obras reconocidas, de la historia, el pensamiento y la reflexión de los hombres.

Por lo tanto, los autores sostienen que en áreas cercanas a las matemáticas y, las que dependen de la investigación y procesos de laboratorio, es fácil determinar esos saberes científicos o esenciales que pueden ser enseñables y utilizar la transposición didáctica y que para otras disciplinas no es aconsejable dicha transposición sin descontextualizar o fragmentar el saber, para lo cual proponen determinar unos saberes esenciales que son el punto de partida de la enseñanza (cf. Gómez Mendoza, Transposición didáctica, 2005).

2.3.2 Dispositivo Didáctico.

Del latín *dispositus* “dispuesto”, aparato o mecanismo que desarrolla determinadas acciones. Su nombre está vinculado a que dicho artificio está dispuesto para cumplir con un objetivo o acción prevista y, artificio hace alusión que es creado de manera artificial y no natural. Según Souto (2005) ese artificio es concebido teniendo en cuenta que el dispositivo tiene carácter productor, transformador, lo cual se vincula a la intencionalidad de provocar aquellos cambios que se buscan y otros no esperados, cambios a nivel social, institucional, grupal, o individual. El dispositivo se pone a disposición para provocar en otros, aptitud o disposición, entre otras palabras “abre el juego”.

Según Vargas (2002, pág. 8) el término dispositivo se puede tomar como un artefacto tecnológico que está ahí, dispuesto porque se requiere para cumplir un fin específico con criterios de eficiencia, vigencia, practicidad, estética, costos, entre otros.

Al ser un *objeto tecnológico*, eso denota dos cosas: que es real y que se puede interactuar con él; y lo tecnológico implica que responde a una construcción basada en una metodología probada, sistemática y aceptada como válida con el fin de que resuelva un problema específico o actúe de una manera específica en una situación dada.

El concepto de dispositivo didáctico hace hincapié en el hecho en que, una situación de aprendizaje no se produce al azar, sino que la genera un dispositivo que sitúa a los estudiantes en un objetivo claro, el cual deben alcanzar a través de una actividad, un proyecto o problema a resolver. No existe un dispositivo general, todo depende de la disciplina, de las temáticas específicas, el nivel de los estudiantes y de la experiencia del profesor.

Para Estela Quintar (Salcedo Casallas, 2009, pág. 5) el uso del concepto dispositivo “es un artificio que tiene como función articular, conectar, poner en movimiento sistemas orgánicos, materiales o simbólicos”. Lo cual le da un grado de complejidad y su función es la de “bisagra entre mundos simbólicos y/o sistemas materiales u orgánicos”, lo cual le da el carácter de unir y colocar en acción, es decir, es un activador de procesos subjetivos e intersubjetivos.

La didáctica es el proceso por el cual, a través de dispositivos, objetivos de aprendizaje e intereses de los aprendices se materializa la enseñanza. Cuando se toman conocimientos y saberes de las ciencias, de las artes, de las técnicas o las tecnologías y se ponen en términos o elementos que sirvan para que ellos puedan ser enseñados se habla de *Transposición Didáctica*. (Vargas Guillén & Gamboa Sarmiento, 2008, pág. 55).

2.3.3 Estrategia de Portafolios.

Esta estrategia ha sido usada tiempo atrás en las artes, la arquitectura y la fotografía, para ilustrar el trabajo realizado, a los clientes, pero se ha convertido en una herramienta valiosa en el aula ya que permite a los estudiantes poner en evidencia el avance en los procesos de aprendizaje, comprender los temas de aprendizaje, hacer una reflexión y autoevaluación sobre estos; a los profesores les permite llevar un seguimiento del aprendizaje del estudiante. A continuación se muestran algunas definiciones citadas por Danielson & Abrutyn (1997, pág. 3) para el término de portafolio:

Es un registro del aprendizaje que se concentra en el trabajo del alumno y en su reflexión sobre esta tarea. Mediante un esfuerzo cooperativo entre el alumno y el personal docente se reúne un material que es indicativo del progreso hacia los resultados esenciales” (*National Education Association*, 1993, pág. 41).

Es una selección deliberada de los trabajos del alumno que cuenta la historia de sus esfuerzos, su progreso o sus logros. En él deben incluirse la participación del alumno en la elección de su contenido, los criterios de la selección y las pautas para juzgar sus méritos, así como las evidencias de su procesos de autorreflexión” (Arter, 1990, pág. 27)

Un portafolio es algo más que una mera caja llena de cosas. Se trata de una colección sistemática y organizada de evidencias utilizadas por el docente y los alumnos para supervisar la evolución del conocimiento, las habilidades y las actitudes de estos últimos en una materia determinada. (Vayrus, 1990, pág. 48)

Estos conceptos denotan que es un proceso reflexionado entre el docente y el estudiante, donde se cuenta la historia de manera coherente del proceso de aprendizaje llevado a cabo por el estudiante. Los trabajos llamados artefactos o evidencias no son seleccionados al azar sino que muestran de una forma consciente los logros alcanzados por el estudiante durante el proceso de aprendizaje (Jacobson & Florman, 2011, pág. 6).

Según Jacobson & Florman (2011), por lo general un portafolio de aprendizaje se compone de los siguientes elementos:

- Las evidencias o artefactos seleccionados por los estudiantes que son el resultado de logros de aprendizaje concretos. Pueden incluir trabajos escritos, diseños, grabaciones, aplicaciones multimediales, entre otros, que le permiten documentar su aprendizaje.
- La reflexión del estudiante sobre las evidencias entregadas, incluye observaciones sobre qué representa dicha evidencia y por qué fue seleccionada.
- La realimentación por parte del profesor tanto a las evidencias como a la reflexión realizada por el estudiante.

Por lo tanto, el portafolio no sólo es utilizado para el aprendizaje del estudiante sino que le permite realizar procesos de autoevaluación ya que debe seleccionar de un conjunto de evidencias aquellas que cumplan con unos criterios establecidos de común acuerdo con el docente y esto lo lleva a que realice una autovaloración de su trabajo, en términos cuantitativos y cualitativos, resultado de una reflexión, sobre los logros y las debilidades en el aprendizaje. Esto le da un valor adicional al aprendizaje porque el estudiante realiza procesos *metacognitivos*, es decir, conoce bien la forma en que realiza las representaciones mentales internas de una realidad, cómo funciona y, le permite autocontrolar y autorregular su propio aprendizaje para saber actuar ante situaciones similares. Por su parte el profesor puede evaluar de manera diferente a la tradicional, tanto el aprendizaje como su enseñanza teniendo en cuenta la experiencia vivida por los estudiantes (Klenowski, 2005) .

2.4 HEURÍSTICA

Etimológicamente (Hernandez, 2005, págs. 100-103), heurística es un término que aparece desde la Grecia antigua, pero es utilizado directamente a partir de la segunda mitad del siglo XX, significa descubrimiento o invención debido a la reflexión y no al azar. Cuando se usa como sustantivo se refiere al arte o ciencia del descubrimiento, si se usa como adjetivo, se refiere a cosas más específicas como las estrategias heurísticas, esto conlleva a hablar de los métodos o vías del descubrimiento científico.

Un heurístico se describe como el conocimiento explícito y no explícito necesario para organizar la secuencia de operaciones del pensamiento, tales como buscar, encontrar y relacionar con el fin de sugerir soluciones o estrategias a un problema dado, en situaciones donde no se tiene un conocimiento profundo de la temática relacionada al problema a resolver. La heurística no es la solución es una sugerencia o camino para indagar e investigar y poder atacar el problema. (Kumar & Gupta, 2011, pág. 459)

En ingeniería, una heurística es un método basado en la experiencia que puede utilizarse como ayuda para resolver problemas de diseño, desde calcular los recursos necesarios hasta planificar las condiciones de operación de los sistemas. Mediante el uso de heurísticos, es posible resolver más rápidamente problemas conocidos o similares a otros conocidos.

El diseño de software, en especial el guiado por el paradigma orientado a objetos, busca plantear modelos como solución a una problemática dada, que se relacionan entre sí y que son independientes de las estructuras de programación, para ello cuenta con principios, patrones de diseño y con la propuesta de diferentes autores, por mencionar algunos: Arthur Riel (1996) que planteó una lista de heurísticos para determinar los diferentes elementos tales como clases, objetos, herencia, entre otros; Johnson y Foote (1988) dieron las directrices para la reutilización de código; Nielsen y Loranger (2006) para el diseño de interfaces web (Ibíd, pág. 460-461).

Esta investigación se desarrolló con base en los heurísticos propuestos por Bruegge y Dutoit, reconocidos catedráticos en el área de Ingeniería de Software desde 1989, que en la actualidad trabajan en la Universidad Carnegie Mellon y en la Universidad Técnica de Munich, su trabajo se hace valioso ya que desarrollan proyectos relacionados junto con empresas y aplicados a situaciones reales (Bruegge & Dutoit, Object-oriented Software Engineering, 2010).

2.5 TEORÍAS DE APRENDIZAJE

Las teorías más representativas son: la conductista, la cognitivista y la constructivista, que a través de la historia han evolucionado dando lugar a diferentes clasificaciones. La primera se basa en los experimentos de Pavlov sobre el condicionamiento clásico y su esencia se basa en que el origen del conocimiento son las sensaciones captadas a través de los sentidos y que todo sujeto tiene una respuesta ante un estímulo que debe llevar un refuerzo. La segunda teoría, cuyo inicio se le atribuye a Gagné, se basa en el conductismo en

cuanto a los refuerzos y le agrega el estudio de las condiciones internas que intervienen en el aprendizaje. Por último, el constructivismo se basa en la teoría genética que toma los conceptos *actividad mental constructivista*, *competencia cognitiva* y *capacidad de aprendizaje*; tiene en cuenta tanto los procesos psicológicos subyacentes al desarrollo y al aprendizaje, como las características de la situación de enseñanza y aprendizaje. (Gros, 1997, págs. 31,43,81).

En esta investigación se tomaron algunos elementos del enfoque constructivista de los autores Piaget, Ausubel y Vigostky, los cuales se explican a continuación.

2.5.1 El Constructivismo.

Los conocimientos se construyen y no se transmiten o reproducen (Gros, pág. 82), por lo que los estudiantes deben participar activamente en la elaboración de las estructuras de conocimiento. Según este enfoque el aprendizaje depende de lo que la persona conoce previamente y llama la atención en que es importante tener en cuenta:

- Los procesos que componen y articulan el aprendizaje de un contenido como una unidad.
- El diseño de actividades que contextualizan los conocimientos, de manera que la persona, al ponerlos en práctica, afianza el aprendizaje y la posterior transferencia, teniendo en cuenta la complejidad de las situaciones reales.
- Que el conocimiento se construye a partir de la experiencia por lo tanto el estudiante se encuentra todo el tiempo en situación de darle sentido y significado a cada vivencia.
- El error es considerado como un factor que realimenta el aprendizaje.

Según (Serrano & Pons, 2011, pág. 2), independiente de la clasificación de los tipos de constructivismos, existen unos postulados que los recogen a todos de manera explícita o implícita y son:

a) Un constructivismo cognitivo que tiene sus raíces en la psicología y la epistemología genética de Piaget,

b) Un constructivismo de orientación socio-cultural (constructivismo social, socio-constructivismo o co-constructivismo) inspirado en las ideas y planteamientos Vigotskyanos.

c) Un constructivismo relacionado al construccionismo social de Berger y Luckmann (2001) y, a los enfoques posmodernos en psicología que sitúan el conocimiento en las prácticas discursivas (Edwards, 1997; Potter, 1998).

Estas formas del constructivismo comparten la premisa de que el conocimiento es un proceso de construcción del sujeto y no un despliegue de conocimientos innatos, ni son copiados de conocimientos dados en el mundo externo, pero difieren en cuestiones epistemológicas esenciales de la forma como se construye ese conocimiento, tanto a nivel subjetivo como intersubjetivo.

Los siguientes elementos desde el constructivismo buscan promover y apoyar los procesos de aprendizaje: El estudiante desempeña un rol activo frente al conocimiento que aprende que le exige la interpretación de la información suministrada por el entorno; la construcción del conocimiento es un proceso individual y depende del desarrollo cognitivo del que está aprendiendo y de los presaberes que posea. (Aguilar, Corredor, Porras, Ewert, & Ramón, 2003, pág. 30).

2.5.2 Teoría de Jean Piaget.

Se soporta en que el desarrollo cognoscitivo del niño está determinado por el desarrollo de sus estructuras mentales. Así, para comprender los procesos de aprendizaje y diseñar o pensar procesos de formación es necesario reconocer la naturaleza y el funcionamiento de las estructuras mentales tanto del niño como del adulto. Piaget concibe la formación del pensamiento como un desarrollo progresivo cuya finalidad es alcanzar cierto equilibrio con respecto al entorno en cada etapa del desarrollo. Él dice: “el desarrollo es un perpetuo pasar de un estado de menor equilibrio a un estado de equilibrio superior”.

Piaget afirma que el pensamiento de los niños es de características muy diferentes al de los adultos. Su teoría cuenta con elementos como el esquema, la organización, la adaptación, el equilibrio en el sujeto cognoscente, los cuales regulan las interacciones del sujeto con la realidad, ya que a su vez sirven como marcos asimiladores mediante los cuales la nueva información es incorporada a la persona, a lo cual se llama asimilación y mediante un proceso evolutivo se da la *acomodación* de las estructuras mentales y los conocimientos de manera que con el tiempo se da lo que se llama *Proceso de Equilibración* del sujeto con su medio (Marro, 1982, págs. 23-25).

2.5.3 Teoría de Vigotsky.

Vigotsky concibe el aprendizaje de cada sujeto como un proceso sociocultural que, está relacionado con el medio y los círculos sociales en los cuales se desarrolla el aprendiz. La construcción de conocimiento es el resultado del aprendizaje mediante experiencias, es decir que no se transmite de una persona a otra, de manera mecánica como si fuera un objeto, sino por medio de la interacción del sujeto con la realidad. Por lo tanto, el contexto juega un papel central pues la interacción social se convierte en el motor del desarrollo.

Vigotsky introduce el concepto de *zona de desarrollo próximo (ZDP)* que es la distancia entre el nivel real de desarrollo, determinado por la capacidad de resolver independientemente un problema como individuo, y el nivel de desarrollo potencial, determinado a través de la resolución de un problema bajo la guía de un experto o por el acompañamiento de otros sujetos o sistemas simbólicos. Los dos aspectos a tener en cuenta son: la importancia del contexto social y la capacidad de imitación por parte del que aprende, en esta teoría se sustenta que aprendizaje y desarrollo son dos procesos que interactúan. El aprendizaje se produce más fácilmente en situaciones colectivas, por ejemplo, la interacción con los padres facilita el aprendizaje (Ruiz Carrillo & Estrevel Rivera, 2010, págs. 138-140).

2.5.4 Teoría de Ausubel.

Ausubel introduce el concepto de *aprendizaje significativo* para diferenciarlo del repetitivo o memorístico; enfatiza en que los conocimientos previos o presaberes del estudiante juegan un papel fundamental en la adquisición de nuevo conocimiento. La significatividad sólo es posible si se relacionan los nuevos conocimientos con los que ya posee el sujeto. Aprender es sinónimo de comprender, y lo que se comprende lo recordará más fácilmente porque queda integrado a su estructura de conocimientos. El término *significativo* se refiere a aquel material o temática que potencialmente puede ser aprendido con significado y sentido para el que lo internaliza. (Ausubel, *et al*, 1983). Algo que ofrecen las TIC es posibilitar el diseño y la publicación de materiales realmente significativos, los cuales permiten plantear actividades que motiven la participación de los estudiantes, además, las TIC dan acceso a gran cantidad de información. Los profesores facilitan la coherencia ente los presaberes del estudiante y los nuevos conocimientos. (Corredor Montagut, Arbeláez López, & Perez Angulo, 2010, pág. 81).

Es de resaltar que los anteriores postulados aportan al desarrollo de la investigación, en el sentido de que el uso de entornos virtuales, permite que el estudiante pueda construir su propio conocimiento, pueda ejercitarse y aprender a través de situaciones prácticas; pueda interactuar con otros que le permiten confrontar, argumentar y debatir sus concepciones; además de poder caracterizar la población hacia la cual va dirigida la estrategia y los objetivos de aprendizaje, por otra parte también se propicia el aprendizaje colaborativo, el cual busca lograr la construcción de conocimiento a través del trabajo en equipo.

Las TIC propenden porque el proceso de aprendizaje para el estudiante pase de ser receptorista y memorístico de información, a realizar procesos de búsqueda, análisis y contextualización de la información que lo llevan a ser responsable de su propio aprendizaje; y para el profesor de ser transmisor a facilitador o

acompañante del aprendizaje del estudiante, es decir, un mediador (Aguilar, *et al*, 2003, pág. 34).

2.6 INGENIERÍA DE SOFTWARE

Según el estándar internacional ACM⁵, la Ingeniería del Software es una disciplina que se define como “la aplicación de un enfoque sistemático (ordenado), disciplinado y cuantificable al desarrollo, operación y mantenimiento de software, esto es, la aplicación de la ingeniería en el área del software” (IEEE, SMBOK - Guide to the Software Engineering Body of knowledge, 2004, pág. 1).

Según Bruegge y Dutoit (2010, pág. 5), la Ingeniería de Software es un actividad de modelado, los modelos son usados para manejar la complejidad inherente al software y buscar una solución aceptable a una situación problema dada, (Weitzenfeld, 2005, pág. 13).

El *SweBOK*⁶ ha propuesto los siguientes objetivos: caracterizar los contenidos y proveer acceso a través de las temáticas al conjunto de conocimientos de la disciplina, con el fin de dar un lugar a la Ingeniería del Software respecto a otras disciplinas (IEEE, 2004, pág. 19) y proveer una base para su desarrollo curricular y la creación de materiales de certificación.

Ha planteado diez áreas de conocimiento y un capítulo adicional que proporciona una visión general de las áreas de conocimiento de las disciplinas fuertemente relacionadas (Ibíd, pág. 26).

- Requisitos de software

⁵ *Association Computing Machine*

⁶ *Guide to the Software Engineering Body of Knowledge*

- Diseño de software
- Construcción de software
- Pruebas de software
- Mantenimiento de software
- Gestión de la configuración
- Gestión de la ingeniería de software
- Proceso de ingeniería de software
- Herramientas y métodos de la ingeniería de software
- Calidad del software

2.6.1 Modelos de proceso de software

Un modelo de proceso o modelo de desarrollo o ciclo de vida del software permite dar solución a una situación problema mediante la ejecución de unas fases o etapas conocidas como ciclo de vida.

Existen modelos clásicos como:

- Cascada.
- Incremental.
- Espiral.

Y algunos de los más recientes

- Ganar y Ganar (*win-win*).
- Programación Extrema (XP).
- Proceso de desarrollo de software Unificado o proceso unificado (UP).

A continuación se describe el proceso unificado, el cual fue utilizado dentro de la investigación para el desarrollo del dispositivo computacional (Weitzenfeld, 2005, págs. 56,195).

Proceso Unificado (UP), por sus siglas en inglés *Unified Process*. Tiene sus orígenes en la década de 1980, fue propuesto por Booch, Jacobson y Rumbaugh

y es una extensión al proceso *Objectory (object factory)*⁷, actualmente esta metodología luego de refinarse y documentarse hace parte del Proceso Unificado de Rational (RUP).

Se basa principalmente en la especificación de requerimientos del sistema a través de casos de uso; es centrado en la arquitectura del sistema, es decir, integra las necesidades percibidas por todos los usuarios, es importante para la conceptualización y gestión de todo el proceso de desarrollo de software y da una vista del diseño completo, tomando las características más importantes de este; seguido de un proceso iterativo e incremental que integra importantes rangos de tiempo llamados ciclos, fases, flujos de trabajo, disminución del riesgo, control de calidad y administración del proyecto (Jacobson, Booch, & Rumbaugh, 2000, págs. 3-8).

Posee cuatro fases (Sommerville, 2005, pág. 76):

Inicio: Se define la necesidad y se evalúa su factibilidad, teniendo en cuenta todos los involucrados.

Elaboración: se da inicio al proyecto y su planificación. En esta fase, se realiza el desarrollo del plan de proyecto, la identificación de riesgos claves, el diseño de la arquitectura del sistema, la determinación de los requerimientos del sistema y la especificación de los casos de uso.

Construcción: Comprende el diseño del sistema, la programación y las pruebas. En esta fase se determinan los subsistemas y la integración entre estos, así como la documentación para los usuarios.

Transición: se realizan procesos de instalación, corrimiento bajo un sistema operativo y el correspondiente mantenimiento para el correcto funcionamiento.

⁷ Jacobson, Christensen, Jonsson & Overgaard. *Object-Oriented Software Engineering. A Use-Case Driven Approach*. 1992. Addison-Wesley.

Cada fase consta a su vez de varias iteraciones. Durante cada iteración, se ejecutan diversas actividades en paralelo denominadas *flujos de trabajo*. Las actividades se realizan en cada iteración pero con diferentes necesidades específicas de acuerdo con la fase y son llamadas *flujos de trabajo de ingeniería*.

Se basa en los siguientes principios:

- El éxito de un sistema, depende de la claridad y el conocimiento de las necesidades de los usuarios.
- Las arquitecturas de los sistemas pueden verse desde diferentes vistas.
- En los proyectos grandes y/o complejos se permite dividir el trabajo en componentes.

2.6.2 Actividades de desarrollo de Ingeniería del Software.

Las actividades de desarrollo de Ingeniería del Software buscan manejar la complejidad mediante la construcción de modelos de los dominios del problema o del sistema (Bruegge & Dutoit, Object-oriented Software Engineering, 2010, págs. 16-20). Se destacan las siguientes actividades:

- **Obtención de requerimientos.** Corresponde a la definición del propósito del sistema. Busca definir los requerimientos del sistema que son las características y restricciones del sistema, que satisfacen las necesidades del cliente, el resultado es una descripción en términos de actores y casos de uso. Los actores representan entidades externas que interactúan con el sistema y, los casos de uso son secuencias de eventos generales que describen todas las acciones posibles entre un actor y el sistema.
- **Análisis.** Busca obtener un modelo del sistema que pretende ser correcto, completo, consistente y que no sea ambiguo. Aquí se formaliza la especificación de requerimientos y se examinan detalladamente las condiciones de frontera. Los En el análisis orientado a objetos, los desarrolladores construyen un modelo que describe el dominio de la aplicación, que se extiende para describir la manera en que interactúan los actores y el sistema para manipular el modelo del dominio de

la aplicación. El modelo de análisis describe el sistema por completo desde el punto de vista de los actores y sirve como la base de la comunicación entre el cliente y los desarrolladores.

- **Diseño del sistema.** Durante esta etapa los desarrolladores transforman el modelo de análisis en un modelo de diseño del sistema tomando los requerimientos no funcionales; permite tener información acerca de la estructura interna del sistema. En el desarrollo de esta actividad se formulan los objetivos de diseño del proyecto y se descompone el sistema en subsistemas.
- **Diseño de objetos.** Corresponde a la definición de los objetos que relacionan el modelo de análisis y la plataforma hardware y software definida durante el diseño del sistema, con el fin de alcanzar los objetivos de diseño.
- **Implementación.** Los desarrolladores traducen el modelo de objetos en un lenguaje de programación de alto nivel, lo que permite la generación de los archivos de código fuente.
- **Pruebas.** Los desarrolladores encuentran las diferencias entre el sistema y los modelos de este mediante la comparación del modelo de diseño de objetos con cada objeto y subsistema.

2.7 LENGUAJE DE MODELADO UNIFICADO – UML

El lenguaje de modelado unificado (UML) es el resultado de la unificación de la técnica de modelado de objetos (OMT) de Rumbaugh (1991), Booch (1994) e Ingeniería de software orientada a objetos (OOSE- Jacobson, 1992). El objetivo de UML es proporcionar una notación estándar que puede ser usada por todos los métodos orientados a objetos y ha seleccionado e integrado los mejores elementos de precedentes notaciones. (Jacobson, págs. 29-33)

A continuación se describen cada uno de los diagramas UML para la representación de dichos modelos.

- **Diagramas de caso de uso:** Representan la funcionalidad del sistema desde el punto de vista del usuario. Son utilizados en la obtención de requerimientos y el análisis pues definen las fronteras del sistema.
- **Diagramas de clase:** Representan la estructura estática del sistema. Las clases son abstracciones que encapsulan las características y el comportamiento de un conjunto de objetos, los cuales son instancias de las clases. Estos diagramas describen el sistema en términos de clases, objetos, atributos, operaciones y sus asociaciones.
- **Diagramas de secuencia:** Representan el comportamiento del sistema, visualizan la comunicación entre objetos y permiten identificar objetos faltantes en los casos de uso. Representan las interacciones entre los objetos.
- **Diagramas gráfica de estado:** Describen el comportamiento dinámico de un objeto individual como un número de estados y las transiciones entre estados.
- **Diagramas de actividad:** Describen el comportamiento del sistema en términos de actividades, que son estados que contienen la ejecución de un grupo de operaciones.

2.8 TECNOLOGÍAS DE INFORMACIÓN Y EDUCACIÓN

2.8.1 e-learning.

Se entiende por *e-learning* o enseñanza en línea al espacio donde profesores y estudiantes participan de forma remota, utilizando los servicios de internet y las bondades de las TIC con el fin de lograr un ambiente educativo interactivo. El término es más utilizado en el entorno empresarial donde ha sido usado para el entrenamiento de los empleados y en el ámbito educativo donde se utiliza para apoyar los procesos de enseñanza, aprendizaje y evaluación, por lo que se habla de virtualización, enseñanza o educación en línea (Ruipérez, 2003, pág. 18). El *e-learning* busca mejorar el conocimiento y el desempeño de los usuarios a través del uso de las TIC mediante el uso de redes, la publicación de la información

utilizando los servicios de internet y el uso de paradigmas de aprendizaje diferentes a los tradicionales (Rosenberg, 2002, pág. 28).

Según (Boneu, 2007, pág. 37) el origen del *e-learning* se ha dado a partir de la evolución de diferentes herramientas de aprendizaje a distancia basado en computador, tales como:

- El aprendizaje basado en computador o CBT (*computer based training*) que fue usado por diversas organizaciones e instituciones educativas hacia los años 60.
- El entrenamiento basado en Internet o IBT (*Internet based training*), que aparece como el siguiente paso en la evolución de los sistemas de aprendizaje a distancia, donde la información podía enviarse a los usuarios a través de Internet o de la intranet.
- El entrenamiento basado en la web o WBT (*web based training*), hace uso de la web, medio por el cual los usuarios reciben la información, participan en actividades educativas y acceden al llamado campus virtual.

Un sistema *e-learning* consta de los siguientes elementos (Ibíd, pág. 38-41):

- Los *Sistemas de Comunicación*, los cuales pueden ser síncronos, es decir, que generan comunicación en tiempo real, entre los usuarios y dirigida por el instructor tales como el chat o videoconferencias y los asíncronos, cuya comunicación no es tiempo real y están dados por el ritmo de cada usuario y los aportes quedan registrados en algunas de estas herramientas como el correo electrónico y los foros.
- Las *plataformas de e-learning* son el software que se encuentra alojado en el servidor y cuya finalidad es la gestión de usuarios, cursos, archivos, actividades de evaluación y servicios de mensajería y/o comunicación.
- Los *contenidos o courseware*, es el material de aprendizaje puesto por el profesor a disposición de los estudiantes, estos pueden tener diferentes formatos, teniendo en cuenta los objetivos del curso.

2.8.2 Plataformas *e-learning*.

Las características básicas con que cuenta un sistema *e-learning* son:

- Interactividad: El usuario también es consciente que es responsable activo de su formación.
- Flexibilidad: Adaptable a las necesidades, los contenidos y modelos pedagógicos de la organización donde se desea implantar.
- Escalabilidad: capacidad de la plataforma de *e-learning* de funcionar igualmente con un número pequeño o grande de usuarios.
- Estandarización: A través de los estándares se pueden reutilizar los cursos por otros, además permite la durabilidad, la actualización y el seguimiento del material educativo que hace parte de los cursos. Uno de dichos estándares es el modelo SCORM (*Sharable Content Object Reference Model*), desarrollado por ADL⁸ en conjunto con otras organizaciones y que busca satisfacer para los objetos de aprendizaje accesibilidad a través de tecnologías web, adaptabilidad en función de las necesidades de las personas y de las organizaciones, permanencia e interoperabilidad independiente de la plataforma y que sea reutilizable y de fácil intercambio entre sistemas de gestión del aprendizaje.

Otras características generales que se encuentran en las plataformas de *e-learning*: son de código abierto y plataforma gratuita; permite arquitectura multiidioma; los lenguajes más utilizados para el desarrollo de webs dinámicas son PHP, Java, Perl y Python; cuentan con una gran documentación y un amplio grupo de apoyo a las comunidades, así como foros de discusión que permiten resolver las dudas de los diferentes usuarios.

⁸ *Advanced Distributed Learning Network*: <http://www.adlnet.org>

Por otra parte las plataformas *e-learning* basadas en web también han evolucionado, como se ilustra en las siguientes etapas.

- *Primera etapa:* Los CMS (*Content Management System* o *Course Management System*) son las plataformas más básicas y permiten la generación de sitios web dinámicos, a través de la gestión de información de diferente tipo como textos, imágenes y multimedia en línea. En esta etapa no se consideran las herramientas colaborativas tales como foros o chats, ni la comunicación en tiempo real.
- *Segunda etapa:* Los LMS (*Learning Management System*) aparecen a partir de los CMS y proporcionan un entorno web de intercomunicación con múltiples usuarios, tales como los foros y chats. Están orientados a la enseñanza y el aprendizaje, proporcionan herramientas para la gestión de contenidos, cursos, actividades, noticias y son adaptables a las necesidades de la organización.

También se usa para los LMS, los términos Ambiente Virtual de Aprendizaje (AVA) o Entorno Virtual de Aprendizaje (*Virtual learning environment*), los dos términos son similares, pero los VLE se enfocan menos en la gestión del aprendizaje. Según Brandon Hall (2001, pág. 29) el entorno virtual de aprendizaje es un software integrado y diseñado para facilitar el aprendizaje de una organización, incluye funciones de administración del aprendizaje, la gestión de cursos, pero no el seguimiento de cursos que no hayan sido creados con el formato o estándar de dicho entorno.

Para la descripción de este proyecto se utilizará el término Entorno Virtual de aprendizaje o su sigla en inglés VLE (*Virtual Learning Environment*). Un VLE permite materializar los espacios didácticos diseñados que no requieren de un espacio físico, ni de la presencialidad de los usuarios ya que se trabaja remotamente usando las bondades de internet. Es un punto de encuentro que permite a los estudiantes (aprendices) y profesores (facilitadores) interactuar con la finalidad de lograr un objetivo aprendizaje común.

Un VLE puede ser utilizado como:

- Sistema de Distribución de Información, en este caso el espacio virtual se utiliza como complemento de las clases presenciales para enriquecerlas con diferentes materiales (apuntes, láminas, artículos, ponencias, libros, presentaciones, entre otros), a los cuales pueden acceder los estudiantes desde cualquier lugar y en cualquier momento. En este caso el VLE ofrece la ventaja de tener permanentemente actualizada la información.
- Sistema de Formación, en este caso el VLE posibilita ofrecer un entorno dinámico y flexible donde se publican y acompaña el desarrollo de actividades que hacen parte de los procesos de enseñanza, aprendizaje y evaluación.
- *Tercera etapa:* los LCMS (*Learning Content Management System*) integran las funcionalidades de los CMS y los LMS y añaden técnicas de gestión de conocimiento con el fin de que las organizaciones mejoren sus prácticas y puedan generar, en forma fácil los contenidos y materiales educativos.

En la tabla 1 se muestran una comparación entre dos tipos de *e-learning*, LMS y LCMS.

Tabla 1. Comparación entre LMS y LCMS

USOS	LMS ó VLE	LCMS
Usuarios a los que va dirigido	Responsables de los cursos, administradores de formación, profesores o	Diseñadores de contenidos, diseñadores instruccionales, directores
Proporciona	Cursos, eventos de capacitación para estudiantes	Contenidos para el aprendizaje, y soporte a usuarios
Manejo de clases, formación centrada en el	Sí (pero no siempre)	No
Administración	Cursos, eventos de capacitación y estudiantes	Contenidos para el aprendizaje, soporte en el

USOS	LMS ó VLE	LCMS
Análisis de competencias	Sí	Sí (en algunos casos)
Informe del rendimiento de los participantes en el seguimiento de la	Enfoque principal	Enfoque secundario
Colaboración entre usuarios	Sí	Sí
Mantiene una base de datos de los usuarios y sus perfiles	No siempre	No siempre
Agenda de eventos	Sí	No
Herramientas para la creación de contenidos	No	Sí
Organización de contenidos reutilizable	No siempre	Sí
Herramientas para la evaluación integrada para	Sí (la gran mayoría)	Sí (la gran mayoría)
Herramienta de flujo de trabajo	No	Sí (en algunas ocasiones)
Comparte datos del estudiante con un	Sí	No
Evaluación dinámica y aprendizaje adaptativo	No	Sí
Distribución de contenido, control de navegación e	No	Sí

Fuente: (Boneu, 2007, p. 40)

Las plataformas *e-learning* se pueden agrupar según su funcionalidad (Boneu, 2007, págs. 41-44) así:

⁹ *Enterprise Requeriment Planning*

- a) Herramientas orientadas al aprendizaje
- b) Herramientas orientadas a la productividad
- c) Herramientas para la implicación de los estudiantes
- d) Herramientas de soporte
- e) Herramientas destinadas a la publicación de cursos y contenidos
- f) Herramientas para el diseño de planes de estudio
- g) Sistemas para la gestión del conocimiento en el ámbito educativo

En la tabla 2, se muestran las herramientas correspondientes, algunas plataformas no cuentan con algunas de ellas.

Tabla 2. Herramientas de las plataformas *e-learning*

FUNCIÓN	HERRAMIENTA	DESCRIPCIÓN
Orientadas al aprendizaje	Foros	Permiten el debate sobre diversas temáticas durante el desarrollo del curso. Los foros pueden estar organizados cronológicamente, por categorías o temas y permiten o no adjuntar archivos.
	Buscador de foro	Facilitan la selección y localización de las intervenciones, en los foros, que incluyan el patrón de búsqueda indicado.
	<i>e-portafolio</i>	El portafolio digital o electrónico, permite hacer el seguimiento a la evidencias del aprendizaje de los participantes, teniendo acceso a los trabajos realizados como parte de las actividades formativas. Los trabajos pueden estar en diferentes formatos.
	Intercambio de archivos	Permiten a los usuarios subir archivos y compartirlos con los profesores u otros estudiantes del curso y en múltiples formatos de archivos; los formatos de los archivos pueden ser HTML, Word, Excel, Acrobat, entre otros.
	Chat	Herramienta de comunicación síncrona para el intercambio de mensajes entre los participantes
	Correo electrónico	Mensajería o comunicación asíncrona permite que un correo electrónico pueda ser leído o enviado desde un curso. Estos mensajes pueden hacerse dentro del curso o usar direcciones de correo externas.
	Presentación multimedia	Estos servicios (vídeo, videoconferencia, pizarra electrónica, entre otros). Una videoconferencia permite la comunicación entre dos usuarios cualesquiera. Una pizarra electrónica es de comunicación asíncrona, puede ser utilizada por el profesor con sus estudiantes en una clase virtual tal y como puede ser también la compartición de aplicaciones o el chat de voz.

	<i>Blog</i>	También llamado notas en línea. Permite a los estudiantes y profesores efectuar anotaciones en un diario donde se hacen comentarios sobre algún texto, actividades, calendario, etc.
	Wikis	Facilitan la elaboración colaborativa de documentos en línea. Gracias a los wikis el conocimiento ya no se apoya sólo en las fuentes clásicas, sino que cuenta con diferentes opiniones.
	Calendario	Permiten la revisión del progreso y sirven para hacer la planificación en el tiempo del curso con respecto a las actividades de este.
	Ayuda	Brinda asesoría en el uso de la plataforma a los participantes, por lo general incluyen tutoriales, manuales de usuario, ayuda en línea, por teléfono o correo electrónico.
	Buscador de cursos	Facilitan la selección y localización de los cursos con un patrón de búsqueda.
	Trabajo fuera de línea	Son mecanismos de sincronización para que los estudiantes puedan trabajar en forma desconectada de la plataforma y lo pueda hacer localmente; además permite que el estudiante al volverse a conectar se sincronice y actualice con la última vez que se desconectó.
	Control de publicación	Controlan la publicación de páginas al llegar a una fecha determinada y dejarlas inaccesibles las que hayan caducado; así como comprobar enlaces rotos.
	Noticias	Mantienen informado de las novedades a los usuarios de la plataforma y de los cursos.
	Avisos	Cada vez que sucede un evento en la plataforma que concierne al usuario, tales como la actualización de páginas, mensajes a foros y envío automático, se genera un mensaje automático desde la plataforma, informándole los cambios que se han producido.

	Sindicación de contenidos	RSS10, News, PodCast, son algunos de los que prestan estos servicios y permiten incorporar a la plataforma contenidos de forma sindicada, que son ofrecidos desde el exterior o desde la propia plataforma, incluso crear contenidos que pueden sindicarse desde otras plataformas usando archivos formato XML. Gracias a los <i>feeds</i> (programas o sitios que permiten leer fuentes RSS), algunos ejemplos de servicios de noticias y periódicos son: ABC, El país, El mundo, Universia, Noticias de la Universidad complutense de Madrid.
Enfocadas a los estudiantes	Grupos de trabajo	Permiten organizar un curso en grupos, de forma que proporciona un espacio para cada uno de ellos, donde el profesor asigna las actividades correspondientes.
	Autovaloraciones	Los estudiantes pueden practicar, revisar test en línea, y conocer sus valoraciones. Éstas no son contabilizadas por el profesor.
	Grupos de estudio	Permite al estudiante hacer grupos de estudio, clubes o equipos de trabajo colaborativos.
	Perfil del estudiante	Permite el registro de la información personal y de interés del estudiante, la foto, las preferencias, el trabajo en un curso y anunciarse.
Soporte	Autenticación de usuarios	Proceso que permite el acceso de los usuarios a los cursos, a través de un nombre de usuario y contraseña. Incluye el procedimiento por el cual se crean y mantienen las respectivas cuentas.
	Registro de estudiantes	Este proceso se puede efectuar de diferentes formas, lo puede realizar el profesor o los estudiantes, dependiendo de cómo se haya configurado se puede inscribir o borrar del curso.
	Auditoría	Permite consultar todas las acciones realizadas por los usuarios en la plataforma, así como obtener estadísticas sobre la utilización. Estas herramientas son utilizadas por los administradores

¹⁰ Really Simple Syndication

	Administración del curso	Permiten a los profesores administrar la progresión de una clase a través del material del curso y a los estudiantes comprobar sus progreso, con los trabajos, test, pruebas, etc.
	Apoyo al creador de cursos:	Soporte a los creadores de cursos, en la administración de éstos, a través de foros de discusión, ayudas en línea, teléfono o correo electrónico
	Calificaciones	Permiten de forma cuantitativa conocer el rendimiento y trabajo de los estudiantes de un curso.
	Seguimiento del estudiante	Proporcionan un análisis sobre el uso de los recursos y materiales que hacen los estudiantes del curso.
	Reutilización y posibilidad de compartir contenidos	Los contenidos creados en un curso se pueden compartir con otro profesor, en otro curso y otro sitio. El sistema debe permitir compartir un sistema de archivos y repositorios de contenidos abiertos.
	Plantillas de curso	Utilidades para crear la estructura de un curso en línea.
	Administración del currículo	Proporciona un currículo personalizado a los estudiantes, basado en los prerrequisitos del programa educativo o actividades, en trabajos previos o resultado de los test.
	Personalización del entorno	Permite cambiar la apariencia gráfica y visual de los cursos, de esta manera se pueden colocar imágenes y logos institucionales.
	Herramientas para el diseño de la educación	Ayudan a los creadores de cursos a crear secuencias de aprendizaje, plantillas o asistentes.

	Conformidad con el diseño de la educación	Soporta los estándares (IMS, AICC y ADL) para la compartición de materiales de aprendizaje con otras plataformas <i>e-learning</i>
Gestión de conocimiento en el ámbito educativo	Sistemas integrados de conocimiento	Proporcionan mecanismos de trabajo colaborativo y facilitan la interacción y colaboración de los usuarios, mediante foros o grupos de discusión, representan la estructura del conocimiento de forma jerárquica o árbol de conocimiento o como redes jerárquicas de nodos conectados entre sí por relaciones.
	Sistemas mediadores de información	Proveen a sus usuarios de una interfaz para realizar consultas vía web; las fuentes de conocimiento son heterogéneas y distribuidas, aunque da la apariencia de estar utilizando un sistema centralizado y homogéneo. Para la descripción e indexación de las fuentes de conocimiento y de su contenido, se utilizan lenguajes que permiten relacionar la información, como es el caso del estándar RDF ¹¹
	Repositorios	Librerías digitales que combinan las TIC y el almacenamiento digital de información para reproducir, emular librerías convencionales, como son la colección, catalogación, administración y difusión de información bibliográfica. Pueden recolectar información desde fuentes distribuidas y permiten construir al usuario su propia biblioteca digital.

¹¹ RDF, *Resource description framework*, tecnología para describir metadatos en la web recomendado por el consorcio W3C (www.w3.org/RDF)

	Ontologías	La orientación de los sistemas de gestión del conocimiento basados en ontologías que soportan sistemas de búsqueda automática del conocimiento y facilitan la toma de decisiones aplicando un motor de inferencia a bases de datos estructuradas ontológicamente. El vocabulario específico de clasificación no suele ser universal y eso presenta problemas de acceso en muchos casos. Se dan solapamientos de los elementos en más de un sitio.
	Folcsonomías	El término folcsonomía (folc+taxo+nomía) significa “clasificación gestionada por el pueblo”. De una forma sencilla, los usuarios guardan o clasifican los elementos y le asignan palabras clave (<i>tags</i>), que pueden ser compartidas con otros usuarios.

Fuente: Autor

El proyecto está concebido bajo plataformas libres, sin embargo, se enumeran algunas soluciones propietarias o privativas, que existen en el mercado, ver tabla 3, con su correspondiente enlace y autor.

Tabla 3. Plataformas privativas

Nombre	Autor	País	Enlace
<i>WebCT</i>	Murray Goldberg. Universidad De British Columbia (UBC)	Canadá	http://www.latined.com/
<i>Blackboard</i>	Fusión Blackboard Inc. y WebCT	Estados Unidos	http://www.blackboard.com/
<i>Angel</i>	ARTI ¹² con Universidad de	Estados Unidos	www.angelllearning.com
<i>First Class</i>	Opent Text	Estados Unidos	http://www.firstclass.com

Fuente: Autor

Las características de cada una aparecen en la siguiente descripción.

- **WebCT.** (*Web Course Tools*, o Herramientas para Cursos Web) Es usado por las instituciones para la capacitación y aprendizaje en línea. La flexibilidad de las herramientas para el diseño de clases hace este entorno muy atractivo en la creación de cursos en línea, tanto para usuarios principiantes como experimentados. Los instructores pueden añadir a sus cursos varias herramientas interactivas tales como: tableros de discusión o foros, sistemas de correos electrónicos, chats, entre otros. Cuenta con ayudas (*wizards*) para la creación de contenidos en línea, y emplea plantillas predefinidas. El 12 de octubre de 2005, *BlackBoard Incorporation* y WebCT anunciaron su fusión.
- **Blackboard.** Es el resultado de la fusión de la compañía *Blackboard Inc.* con WebCT, en el año 2005. Tiene diferentes presentaciones según las necesidades de la institución. Permite el cargue de archivos por individuo, incorporar la parte

¹² *Advanced Research & Technology Institute*

evaluativa con preguntas tipo falso/verdadero, llenar en espacios, apareamiento, selección múltiple, ordenamiento y preguntas cortas/ensayos.

A fecha del 2005, desarrolló y licenció aplicaciones de programas empresariales y servicios relacionados a más de 200 instituciones educativas en más de 60 países. Es usado para administrar aprendizaje en línea, procesar transacciones y comercio electrónico.

- **Angel.** Tiene un buen esquema de seguridad. Todos sus elementos tales como cursos, mensajes, entre otros, cuentan con motor de búsqueda. Los estudiantes pueden descargar el contenido del curso, enviar mensajes y sincronizar el calendario con PDA¹³. Se puede acceder al sistema empleando protocolos de acceso Inalámbricos. Para los grupos de trabajo cuenta con un directorio de archivos de grupo, foros de discusión, salón de chat, lista de correo de grupo, muestreos de opinión, tareas, actividades, logros, calendario de eventos compartido, intercambio de archivos, asignación de líder de grupo. Las evaluaciones soportan diferentes tipos de preguntas.

- **FirstClass.** Es un plataforma que se reconoce por su seguridad en la parte de correo electrónico y mensajería unificada, cumple con los estándares SCORM servidor de *Blogs*, *Podcast*, BBS y foros. Tiene y redes de comunicación instantánea para desastres naturales.

Dentro de las soluciones libres, se encuentran las siguientes, ver tabla 4:

Tabla 4. Plataformas de distribución libre

Nombre	Autor	País	Enlace
ATutor	Universidad de Toronto	Canadá	www.atutor.ca
Claroline	Universidad de Louvain	Bélgica	www.claroline.net
Dokeos	Empresa Belga y certificada	Bélgica	www.dokeos.com
Ilias	Universidad de Colonia	Alemania	www.ilias.de

¹³ *Personal Digital Assistant*

Nombre	Autor	País	Enlace
Manhattan	Western New England College	Estados Unidos	http://manhattan.sourceforge.net
Sakai	Universidad de Michigan y Universidad de Indiana	Estados Unidos	http://www.sakaiproject.org/
Moodle	Martin Dougiamas	Australia	www.moodle.org

Fuente: Adaptado de (Delgado Cejudo, 2003, pág. 30)

A continuación aparece una breve descripción de las más conocidas, por su gran aceptación y el soporte que brindan a sus usuarios.

- **ATutor.** Es considerado un LCMS ya que tiene la capacidad de creación de cursos y entornos de aprendizaje en línea. Su objetivo es lograr la accesibilidad y adaptabilidad. Cumple con las especificaciones de accesibilidad W3C e incluye servicio de email interno que permite hacer envíos a correos externos. Los estudiantes pueden compilar el contenido del curso en páginas de fácil impresión y utilizar el servicio de chat; la institución puede crear su propia presentación dentro del sistema y los instructores pueden crear contenido lineal o no lineal. Maneja multiidioma.

- **Claroline** es un groupware¹⁴ asíncrono y colaborativo, que permite montar ambientes educativos virtuales en forma rápida y con conocimientos mínimos tanto para la instalación como para la administración del mismo.

- **Dokeos** se encuentra bajo la licencia GNU GPL, el desarrollo es internacional y colaborativo. Certificado por la OSI¹⁵ la característica para administrar contenidos incluye distribución de contenidos, calendario, proceso de entrenamiento, chat, audio y video, administración de pruebas y almacenamiento de registros. Ha sido traducido a más de 31 idiomas y usado por más de cien organizaciones. Posee una interfaz

¹⁴ Conjunto de aplicaciones que se integran bajo un solo proyecto y un trabajo de muchos usuarios de forma concurrentes en el desarrollo

¹⁵ *Open Source Initiative (Organización dedicada a la promoción de código abierto)*

flexible y de fácil uso, cuenta con herramientas de traducción en línea; el intercambio de documentos entre los alumnos y profesor de manera privada; permite realizar conferencias en línea.

- **Sakai** A este proyecto se han unido Instituto Tecnológico de Massachusetts y la Universidad de Standford, junto a la Iniciativa de Conocimiento Abierto (OKI) y el consorcio uPortal. El Proyecto se consolidó con generosa ayuda de la Fundación Mellon. Se creó una fundación para manejar el proyecto, a la que pertenecen más de 100 universidades. Posee funcionalidades tales como comunicación entre profesores y alumnos, lector de noticias RSS, distribución de material docente, realización de exámenes, gestión de trabajos, entre otros.

- **Moodle** Corresponde al acrónimo de *Modular Object-Oriented Dynamic Learning Environment* (Entorno de Aprendizaje Dinámico y Modular Orientado a Objetos). Es un proyecto diseñado para dar soporte a un marco de Educación social constructivista, se distribuye gratuitamente como Software libre bajo la licencia GNU, aunque tiene algunas restricciones gracias a los derechos de *Copyleft*. Tiene una interfaz sencilla, ligera, eficiente, y compatible. Cuenta con los servicios para administrar cursos y usuarios. Ofrece actividades para los cursos tales como: foros, glosarios, cuestionarios, recursos, consultas, encuestas, tareas, chats, talleres, entre otros. Posee un módulo de tareas donde el profesor puede hacer seguimiento, posee herramientas de evaluación automática. Es una plataforma de gran aceptación y cuenta con una comunidad de soporte y documentación.

2.9 INFORMÁTICA EDUCATIVA

Es un campo que emerge de la sinergia que se da entre la Informática y la Educación para dar solución a tres problemas básicos:

- Aplicar la informática en procesos educativos.
- Aplicar educación en Informática.
- Asegurar el desarrollo del propio campo.

El término también hace alusión a la introducción de la informática al currículo escolar a través de varios enfoques propuestos por Sánchez (1992) y citado por Carvajal (2002, pág. 8):

- El aprendizaje acerca del computador comúnmente llamado alfabetización.
- El aprendizaje por medio del computador, es decir programas de ejercitación, tutoriales, entre otros.
- El computador como herramienta instruccional.
- El computador como una herramienta para pensar (Inteligencia artificial).
- Administración del aprendizaje con el computador.

Lo anterior implica que en el ambiente escolar la informática es vista desde diferentes puntos de vista y los profesores se enfrentan a resolver problemas que emergen del mundo y la participación de la Informática en la Educación que se ha extendido principalmente al nivel del uso de las TIC.

Estas no pueden dejarse de asociar a una mayor atención al impacto individual, la educación a distancia y el surgimiento de nuevas formas de enseñar, otros beneficios son el aumento de la capacidad de almacenamiento; y el potencial que representan estas herramientas para crear actividades que sin ellas no podrían llevarse a cabo. Por lo tanto, los profesores como uno de los actores principales cambian su rol dentro del aula, no son transmisores de conocimiento sino que ayudan a los estudiantes a construir el propio conocimiento en los estudiantes y sus procesos de enseñanza le demandan unas habilidades en el uso y búsqueda de recursos de información, todo esto enmarcado en una planeación de actividades que busquen un aprendizaje significativo (cf. Salinas, 2004, págs. 3-6).

La función principal de la Informática educativa es ayudar a la construcción del saber a través del buen uso de las TIC. Algunas bondades de las TIC en los procesos de enseñanza y aprendizaje (Gamboa Sarmiento, 2004, págs. 41-43) son:

- Crean un espacio dinámico que permite la gestión y actualización a bajo costo de información en diferentes formatos, sin límites de espacio y tiempo.
- Apoyan el ofrecimiento de actividades que permiten el desarrollo en los estudiantes de habilidades como la síntesis, la argumentación, así como la selección, organización y búsqueda de información.
- Favorece el aprendizaje colaborativo, el profesor puede hacer realimentaciones al proceso y el estudiante asume un papel más activo y autónomo en su aprendizaje.
- Permite un mejoramiento de la comunicación entre los diversos actores del proceso educativo mediante el ofrecimiento de mecanismos bidireccionales, tanto asíncronos como síncronos, para la interacción entre los participantes de una experiencia.

Según Aguilar, *et al* (2003, pág. 32 -36) la incursión de las TIC, en todos los niveles y entornos educativos para promover el aprendizaje, merece especial atención lo que plantea la UNESCO en relación a que los establecimientos de educación superior están llamados a ser ejemplo del buen uso de las TIC como apoyo al proceso de transformación de la información en conocimiento, igualmente, insiste en que el rol del docente que sin dejar de ser indispensable debe ser, diferente ya que las tecnologías cambian la forma de enseñar, representar, aprender y construir el conocimiento. Puede decirse que las TIC favorecen el aprendizaje por las siguientes razones:

- Son agentes simbólicos que permiten almacenar grandes cantidades de información permitiendo a la mente organizar, recuperar, analizar y procesar diferente tipo de información, gracias a las estructuras de almacenamiento, repositorios, metadatos y metabuscadores que ayudan a tener acceso a información clasificada, eficiente y significativa.
- La interactividad y el uso de diferentes herramientas de gestión, control y planificación de actividades hace que se generen nuevas formas de aprender,

donde los estudiantes tienen un rol más participativo y activo, siendo conscientes de su propio aprendizaje.

- Ofrecen la posibilidad de manejar diferentes formatos de presentación de los datos, tales como texto, gráficos, elementos multimediales, lo que hace que la información sea dinámica y que el usuario pueda realizar interpretaciones e inferencias que generan conocimiento.
- Ofrecen servicios y herramientas que favorecen el trabajo colaborativo pues se tiene un espacio de aprendizaje donde se puede interactuar, compartir, reflexionar, llegar a consensos y propuestas en común sin que existan barreras espacio-temporales. Además, se facilita el apoyo de los docentes y la consulta a expertos en el área, lo que enriquece los procesos de aprendizaje y formación de los estudiantes.
- Permiten el aprendizaje y la participación en procesos de formación a personas que por motivos de trabajo, geográficos o impedimentos físicos no pueden asistir a una institución educativa de manera presencial.

En suma, con el auge de las TIC han generado cambios en el desarrollo de los procesos de enseñanza, aprendizaje y evaluación, así como en el rol del docente como mediador, guía o acompañante del aprendizaje de los estudiantes, quienes deben asumir de manera autónoma el protagonismo del proceso de formación, ya que ellos manejan, administran los tiempos y el ritmo de aprendizaje de acuerdo con sus intereses y necesidades. Esto implica que el docente y el estudiante interactúan de acuerdo con unos tiempos y espacios diferentes a los que se dan en la educación presencial, por lo tanto, el profesor debe dedicar más tiempo a la revisión de consultas, trabajos y dudas que presentan los estudiantes en cualquier momento y lugar.

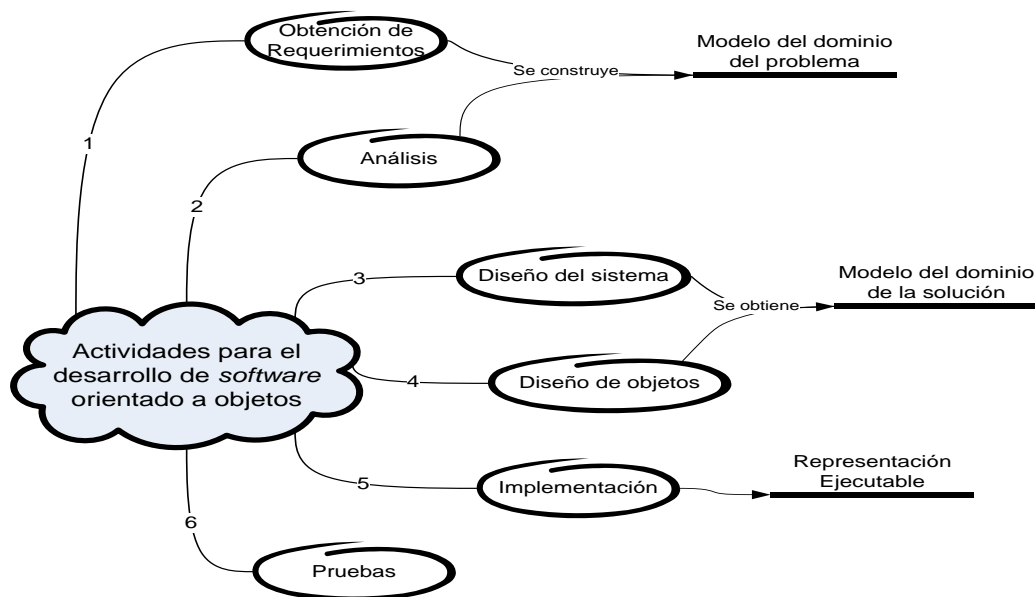
3. ESTRATEGIA DIDÁCTICA

3.1 DETERMINACIÓN DE ELEMENTOS DE REPRESENTACIÓN

Manejar la complejidad de una situación problema en Ingeniería requiere plantear el dominio del problema y de las posibles soluciones a este; los métodos orientados a objetos facilitan y combinan las actividades de modelado de los dos dominios y da como resultado el modelo de la solución que es una extensión del modelo del problema y permite manejar la complejidad inherente al software (Bruegge & Dutoit, Object-oriented Software Engineering, 2010, pág. 8). El desarrollo de software orientado a objetos consta de seis actividades como se observa en la figura 2.

A partir de las actividades 1 y 2 los ingenieros de software junto con el cliente construyen el *modelo del dominio del problema*. En la etapa 3 y 4 los ingenieros descomponen el sistema en partes más pequeñas, las analizan, realizan propuestas de solución más detalladas y se selecciona la más adecuada lo que da como resultado el *modelo de dominio de la solución*. Por último este modelo de solución se traslada a una representación ejecutable.

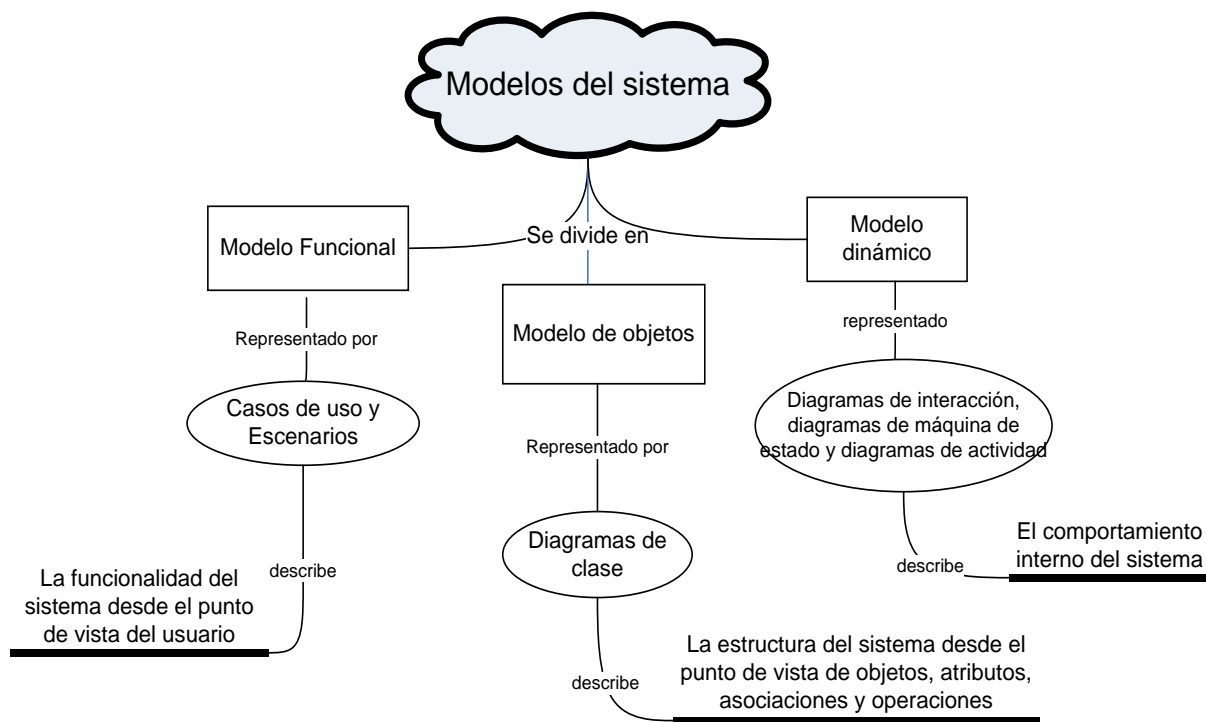
Figura 2. Actividades del desarrollo de software orientado a objetos



Fuente: Autor

La arquitectura software del sistema es un conjunto de vistas que permiten a los diferentes usuarios comprender mejor los modelos del sistema, organizar el proyecto software, fomentar la reutilización y permitir la evolución del sistema dada la flexibilidad del mismo (Jacobson, Booch, & Rumbaugh, 2000, págs. 56-60). La figura 3 muestra los diferentes modelos UML de un sistema (Bruegge & Dutoit, Object-oriented Software Engineering, 2010, pág. 30).

Figura 3. Modelos del sistema



Fuente: Autor

En la presente investigación se seleccionaron las etapas que se centran en procesos de diseño, las cuales son: *Obtención de requerimientos, análisis, diseño del sistema y diseño de objetos*. A continuación se describen los elementos, actividades y su correspondiente *representación* en UML que hacen parte de dichas etapas.

➤ **Obtención de requerimientos** (Bruegge & Dutoit, 2010, págs. 121-124). Se enfoca en describir el propósito del sistema y expresa la visión que tiene el usuario

del sistema. Consta de dos actividades principales que son la especificación de los requerimientos y el análisis (ver figura 4).

La especificación es la actividad más compleja debido a que su elaboración requiere para su elaboración de varios participantes, tanto usuarios como clientes, que tienen diferentes niveles de conocimiento. Para responder a lo anterior se usan los *escenarios* y los *casos de uso*. Los escenarios permiten describir un ejemplo o las posibles situaciones de cómo usar el sistema teniendo en cuenta la interacción entre usuarios y el sistema y los *casos de uso* son una abstracción que describe una clase de escenario.

Figura 4. Ingeniería de Requerimientos

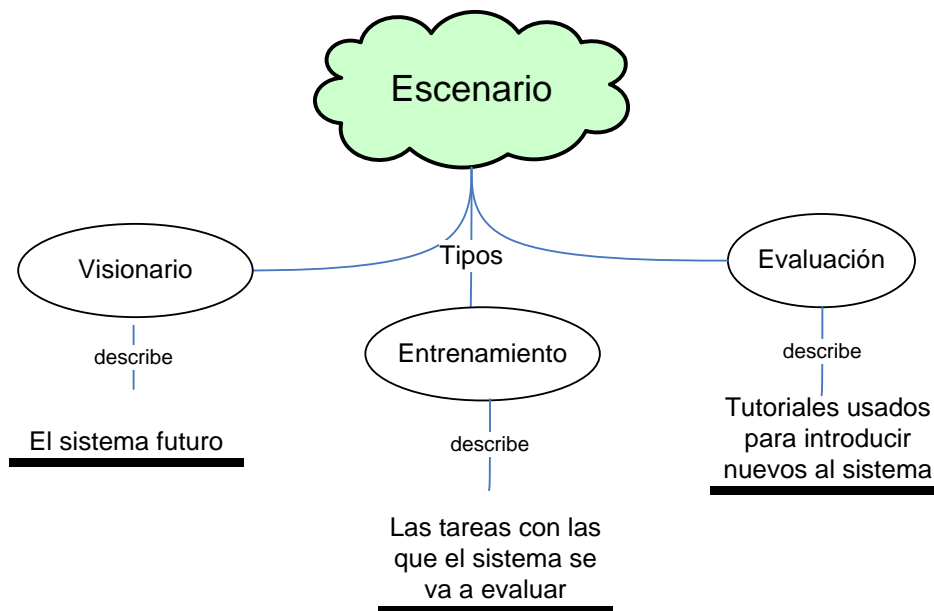


Fuente: Autor

La obtención de requerimientos incluye las siguientes actividades:

- **Identificar Actores.** Entidades externas que interactúan con el sistema, puede ser un humano o uno físico. Los actores incluyen los roles de un usuario. La identificación de actores permite definir las fronteras del sistema.
- **Identificar Escenarios.** Descripción concreta a manera de narración sobre lo que los usuarios hacen y experimentan al usar el sistema, enfocada e informal de una sola característica del sistema desde el punto de vista de un solo actor. Son representaciones más claras para los usuarios y clientes. Es una instancia de un caso de uso. Ver la clasificación en la figura 5.

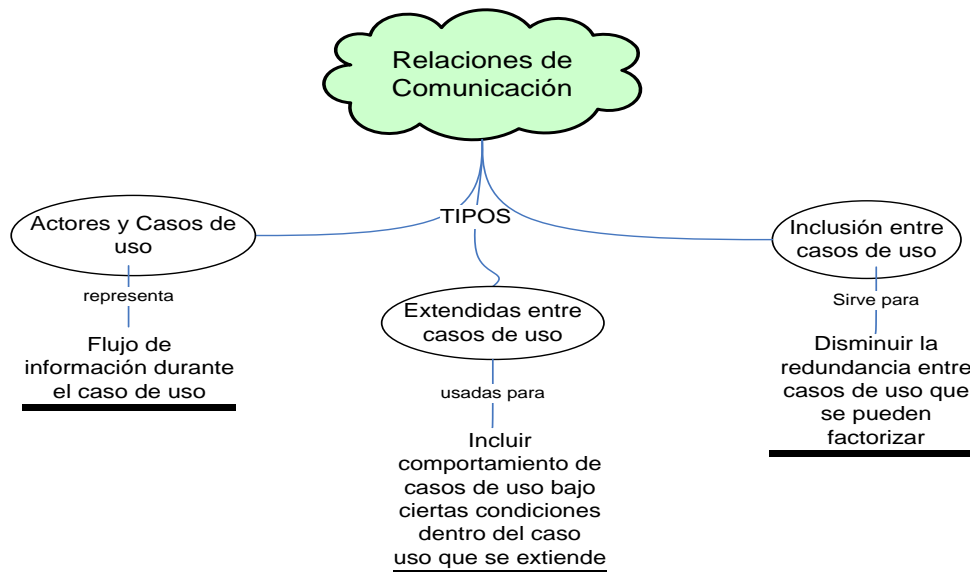
Figura 5. Tipos de escenarios



Fuente: Autor

- **Identificar relaciones entre actores y casos de uso.** Las relaciones de comunicación permiten bajar la complejidad y aumentar la comprensión del modelo. En la figura 6., se muestran los tipos de comunicación.

Figura 6. Tipos de Relaciones



Fuente: Autor

- **Identificación inicial de los objetos de análisis.** Se identifican objetos participantes en cada caso de uso. Son los conceptos del dominio de aplicación. Los desarrolladores identifican, nombran y describen los objetos y los reúne en un glosario, este se incluye en la identificación del sistema y en los manuales del usuario. De aquí se obtiene el modelo de análisis inicial.
- **Identificación de requerimientos no funcionales.** Estos describen aspectos del sistema visibles para el usuario que no están relacionados en forma directa con el comportamiento funcional del sistema. Pueden ser: apariencia de la interfaz del usuario, requerimientos de tiempo de respuesta y aspectos de seguridad. Se definen al tiempo que los funcionales ya que tienen un gran impacto en el desarrollo y costo del sistema. Estos incluyen una gran variedad de requerimientos que se pueden aplicar a diferentes aspectos del sistema, desde la usabilidad hasta el desempeño. El modelo FURPS+¹⁶ es usado por el proceso

¹⁶ Es un acrónimo usando la primera letra de las categorías de los requerimientos: *Functionality, Usability, Reliability, Performance, and Supportability*. El signo + indica las subcategorías adicionales. El modelo FURPS fue originalmente propuesto por Grady en 1992).

unificado (Jacobson et al., 200) y ofrece las siguientes categorías de requerimientos no funcionales.

Tabla 5. Requerimientos No funcionales

CATEGORÍA	DESCRIPCIÓN
Usabilidad	Es la facilidad con la que el usuario puede aprender a operar, realizar las entradas e interpretar las salidas de un sistema o componente. Incluyen el alcance de la ayuda en línea y el nivel de la documentación del usuario. Aspectos de usabilidad de la interfaz como colores de los temas, logos y fuentes.
Confiabilidad	Es la capacidad de un sistema o componente de llevar a cabo las funciones establecidas por un período de tiempo. Ejemplo: Tiempo promedio aceptable para fallos y capacidad de resistir ataques de seguridad específicos. Esta categoría es reemplazada por la <i>Confianza</i> , la cual incluye la <i>Robustez</i> (El grado en que un sistema o componente puede funcionar correctamente en presencia de entradas inválidas o condiciones extremas del ambiente) y la <i>Seguridad</i> .
Rendimiento	Son los requerimientos concernientes a los atributos cuantificables del sistema, tales como el tiempo de respuesta, <i>Producción</i> (qué tanto trabajo puede el sistema desarrollar en un intervalo de tiempo). <i>Disponibilidad</i> (el grado en el cual un sistema o componente está funcionando y es accesible cuando es requerido para su uso) y <i>Precisión</i>
Soportabilidad	Es la facilidad de cambiar el sistema después de desarrollado, por ejemplo <i>Adaptabilidad</i> (la capacidad de cambiar el sistema frente a conceptos del dominio de la aplicación adicionales), <i>Mantenimiento</i> (Capacidad de cambiar el sistema a nueva tecnología o para corregir defectos), cambiar convenciones

CATEGORÍA	DESCRIPCIÓN
	internacionales como lenguajes, unidades, formatos. <i>Portabilidad</i> (Facilidad de un sistema o componente para ser transformado de un ambiente de software o hardware a otro).

Fuente: Autor

De las actividades de obtención de requerimientos y análisis se obtiene el *Documento de análisis de requerimientos* (RAD por sus siglas en inglés).

Figura 7. Plantilla RAD – Documento de Análisis de Requerimientos

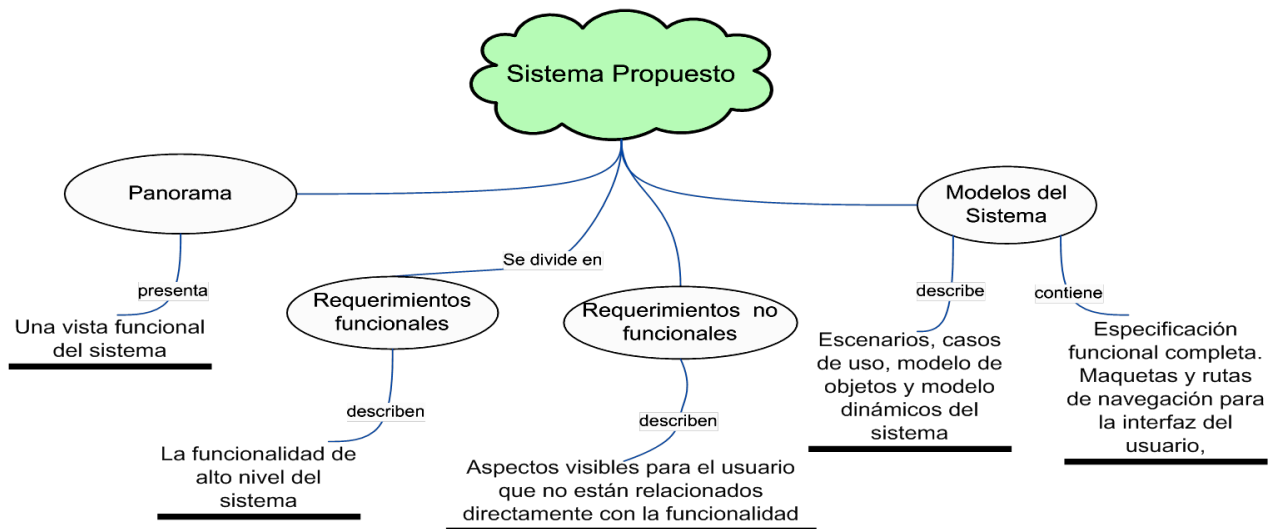
1. Introducción
1.1. Propósito del sistema
1.2. Alcance del sistema
1.3. Objetivos y criterios de éxito
1.4. Definiciones, siglas y abreviaturas
1.5. Referencias
1.6. Panorama
2. Sistema actual
3. Sistema Propuesto
3.1. Panorama
3.2. Requerimientos funcionales
3.3. Requerimientos no funcionales
3.3.1. Usabilidad
3.3.2. Confiabilidad
3.3.3. Rendimiento
3.3.4. Soportabilidad
3.3.5. Implementación
3.3.6. Interfaz
3.3.7. Empaquetamiento
3.3.8. Legal
3.4. Modelos del sistema
3.4.1. Escenarios
3.4.2. Modelos de caso de uso
3.4.3. Modelo de objetos
3.4.4. Modelos dinámicos
3.4.5. Interfaz de usuario: rutas de navegación y maquetas de pantallas
4. Glosario

Fuente: (Bruegge & Dutoit, 2010, pág. 152)

El RAD describe por completo el sistema desde el punto de vista de requerimientos funcionales y no funcionales, ver figura 8 y consta de tres secciones principales:

- **Introducción:** Busca proporcionar un panorama resumido de las funciones del sistema, objetivos y razones para su desarrollo, el alcance y las referencias al contexto de desarrollo (por ejemplo, enunciado de problemas relacionados, referencias a sistemas existentes, estudios de factibilidad)
- **Sistema actual:** Describe el estado actual de las cosas. Si el nuevo sistema reemplazará a uno existente, en esta sección se describe la funcionalidad y problemas del sistema actual. En caso de no existir sistema, esta sección describe cómo se realizan las tareas en este momento.
- **Sistema propuesto:** Documenta la obtención de requerimientos y el modelo de análisis del nuevo sistema. Esta se subdivide en 4 subsecciones, ver figura 8.

Figura 8. Elementos del Sistema Propuesto

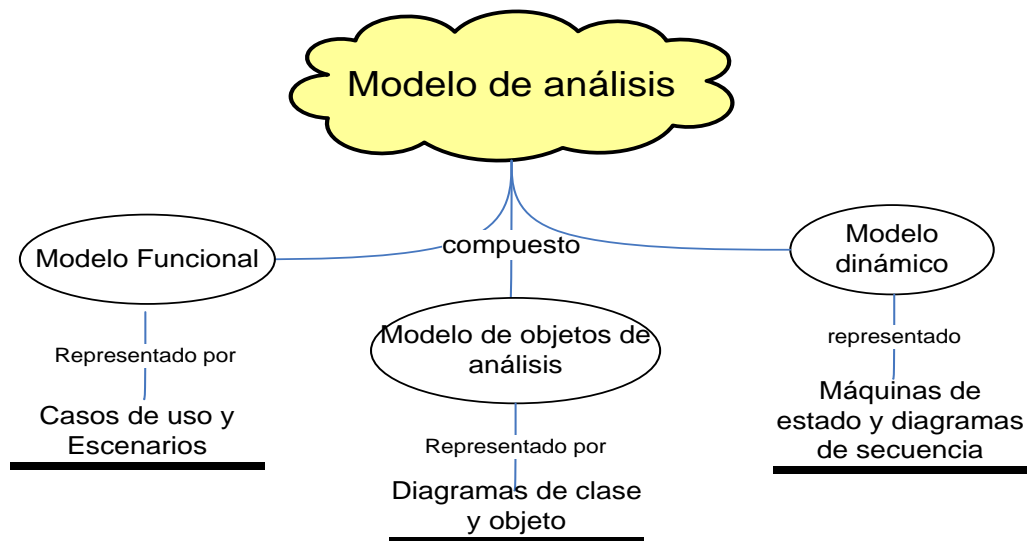


Fuente: Autor

Análisis. Busca obtener un modelo del sistema que pretenda ser correcto, completo, consistente y verificable, es decir no ambiguo. Aquí se formaliza la

especificación de requerimientos y se examina, con más detalle, las condiciones de frontera. Los desarrolladores en el *análisis orientado a objetos* construyen un modelo que describe el dominio de la aplicación. Luego se extiende para describir la manera en que interactúan los actores y el sistema para manipular el modelo del dominio de la aplicación. El *modelo de análisis* describe el sistema por completo desde el punto de vista de los actores y sirve como la base de la comunicación entre el cliente y los desarrolladores. Está compuesto por tres modelos ver la figura 9.

Figura 9. Submodelos de la etapa de análisis



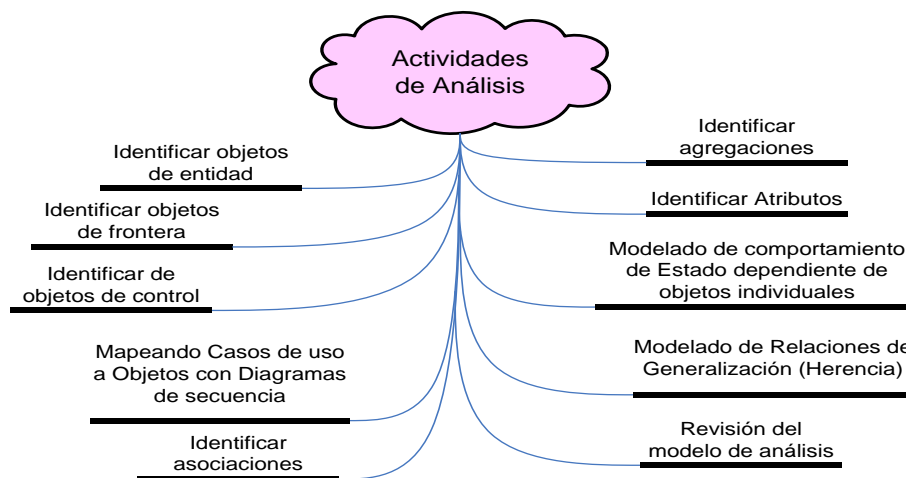
Fuente: Autor

El *modelo funcional* resultado de la obtención de requerimientos y que es refinado en esta etapa, el *modelo de objetos de análisis* es un diccionario visual que se enfoca en los principales conceptos que son manipulados por el sistema y visibles al usuario, así como de sus propiedades y relaciones; y el *modelo dinámico* se enfoca en el comportamiento del sistema.

La figura 10 muestra las actividades del *análisis* que permiten pasar de casos de uso y escenarios producidos en la obtención de requerimientos hacia un *modelo de análisis* y se describen a continuación:

- La definición de *objetos de entidad* conduce al *modelo de análisis* inicial.
- Los *objetos de frontera* representan la interfaz del sistema con los actores, sin ser muy detallado. En cada caso de uso, cada actor interactúa por lo menos con un *objeto frontera*.
- Los *objetos de control* son los responsables de la coordinación entre objetos de entidad y de frontera. Por lo general no tienen un equivalente en el mundo real. Se crea al inicio del caso de uso y se elimina cuando este termina.
- Una alternativa para identificar interacciones entre objetos son las tarjetas CRC (Clases, Responsabilidades y Colaboradores).
- Una asociación muestra la relación entre dos o más clases. Estas tienen las propiedades de un nombre, un rol a cada extremo y una multiplicidad en cada extremo.
- Las relaciones de Herencia o generalización son usadas para eliminar la redundancia del modelo de análisis.

Figura 10. Actividades del análisis



Fuente: Autor

Esta fase se centra en documentar con detalle los ítems mencionados en el documento RAD específicamente el de *modelo de objetos* (ver

Figura 7. Plantilla RAD) incluyendo el diccionario de datos, los diagramas de clase y los *modelos dinámicos*.

Diseño del sistema. Durante esta etapa los desarrolladores transforman el *modelo de análisis* en un *modelo de diseño* del sistema tomando los requerimientos no funcionales y las restricciones descritas en el enunciado del problema y en el RAD, para tener información acerca de la estructura interna del sistema. El modelo de diseño (Bruegge & Dutoit, 2010, pág. 223) refleja las estrategias para la construcción del sistema, los subsistemas y la correspondencia entre el hardware y el software. El diseño de sistemas no es un proceso algorítmico, requiere ingeniería.

El sistema se subdivide para manejar la complejidad. Encontrar subsistemas durante el diseño, es similar a encontrar objetos durante el análisis. Las actividades que los desarrolladores realizan en esta etapa son:

- **Identificar los objetivos de diseño:** Identificar y priorizar las cualidades que se deben optimizar. Estas se deben especificar de manera explícita y se obtienen de los requerimientos no funcionales y del cliente. (Ver tabla 6)

Tabla 6. Criterios de diseño

TIPO	CRITERIO	DEFINICIÓN
Desempeño	Tiempo de Respuesta	¿Qué tan rápido es atendida una petición del usuario después de que esta se ha emitido?
	Producción	¿Cuántas tareas puede realizar el sistema en un período de tiempo fijo?
	Memoria	¿Cuánta memoria es requerida para ejecutar el sistema?

Solidez	Robustez	Capacidad de mantenerse ante datos inválidos del usuario
	Confiabilidad	Diferencia entre el comportamiento especificado y el observado
	Disponibilidad	Porcentaje del tiempo del sistema en que puede ser usado para realizar tareas normales
	Tolerancia a fallos	Capacidad para funcionar bajo condiciones erróneas
	Seguridad	Capacidad para resistir ataques maliciosos
	Inocuidad	Capacidad de no poner en riesgo la vida humana, aun en presencia de errores y fallas.
Costo	De desarrollo	Costo del desarrollo del sistema
	De entrega	Costo de la instalación del sistema y entrenamiento
	De actualización	Costo de migración de datos. Este criterio es resultado de los requerimientos de compatibilidad en retrospectiva
	De mantenimiento	Costo para la corrección de errores y mejoras al sistema
	De administración	Costo requerido para la administración del sistema
Mantenimiento	Extensibilidad	¿Qué tan fácil es agregar funcionalidad o nuevas clases al sistema?
	Modificabilidad	¿Qué tan fácil es cambiar la funcionalidad del sistema?
	Adaptabilidad	¿Qué tan fácil es portar el sistema a diferentes dominios de aplicación?

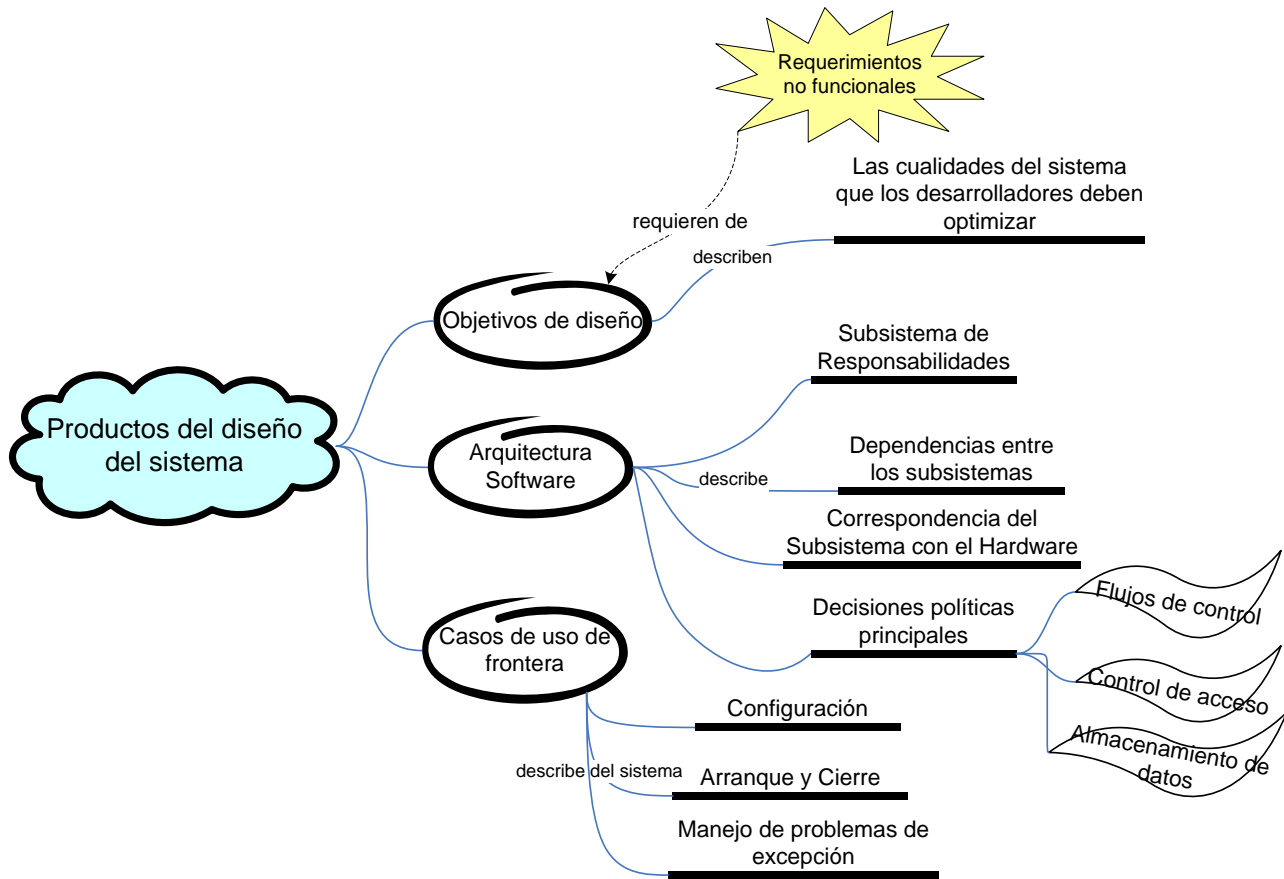
	Portabilidad	¿Qué tan fácil es portar el sistema a diferentes plataformas?
	Legibilidad	¿Qué tan fácil es comprender el sistema leyendo el código?
	Rastreabilidad de requerimientos	¿Qué tan fácil es establecer la correspondencia entre el código y los requerimientos específicos?
Usuario Final	Utilidad	¿Qué tan bien soporta el sistema el trabajo del usuario?
	Usabilidad	¿Qué tan fácil es para el usuario usar el sistema?

Fuente: Adaptado de (Bruegge & Dutoit, 2010, págs. 193-196)

- **Diseñar la descomposición inicial de subsistemas:** Descomponer el sistema en partes más pequeñas basadas en los casos de uso y los modelos del análisis. Usar estilos de arquitectura estándar, como punto de partida durante esta actividad. Esta se deriva de los requerimientos funcionales.
- **Refinar la descomposición de subsistemas:** La descomposición inicial usualmente no satisface todos los objetivos de diseño. Se refina hasta que todos los objetivos sean satisfechos.

Los productos más importantes de la etapa de diseño son: los Objetivos de diseño, la Arquitectura Software y los casos de uso de Frontera, ver Figura 11.

Figura 11. Productos del diseño del sistema



Fuente: Autor

El diseño del sistema se registra en el *Documento de Diseño del Sistema* (SDD por sus siglas en inglés). Este describe los objetivos de diseño propuestos para el proyecto, la descomposición en subsistemas con diagramas de clase UML, la correspondencia entre el hardware y el software con diagramas de despliegue UML, la administración de datos, el control de acceso, los mecanismos de flujo de control y las condiciones de frontera. El SDD permite definir las interfaces entre desarrolladores y como referencia cuando se necesita revisar las decisiones en el nivel de arquitectura. La figura 12, ilustra una plantilla de un SDD.

Figura 12. Plantilla SDD – Documento del diseño del sistema

1. Introducción
1.1. Propósito del sistema
1.2. Objetivos de diseño
1.3. Definiciones, siglas y abreviaturas
1.4. Referencias
1.5. Panorama
2. Arquitectura del software actual
3. Arquitectura del software propuesto
3.1. Panorama
3.2. Descomposición en subsistemas
3.3. Correspondencia entre hardware y software
3.4. Administración de datos persistentes
3.5. Control de acceso y seguridad
3.6. Control de software global
3.7. Condiciones de frontera
4. Servicios de subsistemas
Glosario
Referencias

Fuente: (Bruegge & Dutoit, 2010, pág. 285)

La plantilla está organizada en tres secciones:

- **Introducción.** Busca dar un breve panorama de la arquitectura de software y los objetivos de diseño. También proporciona referencias a otros documentos e información de rastreabilidad, por ejemplo, documento de análisis de requerimientos (RAD) relacionados, referencias a sistemas existentes, restricciones con un impacto en la arquitectura de software.
- **Arquitectura del software actual.** Describe la arquitectura del sistema que se está reemplazando. Si no hay un sistema anterior, esta sección puede reemplazarse con la referencia de arquitecturas actuales de sistemas similares. Incluye la descomposición del sistema, el flujo de control global, las políticas de manejo de errores y los protocolos de comunicación entre subsistemas,

- **Arquitectura del software propuesto.** Documenta el modelo de diseño del nuevo sistema y muestra la arquitectura software de cada componente, las relaciones entre hardware y software, los datos que deben ser almacenados, el control de acceso y la seguridad.
- **Servicios de subsistemas:** Describe los servicios proporcionados por cada subsistema desde el punto de vista de las funciones.

3.2 HEURÍSTICOS

Teniendo en cuenta los elementos de *representación* en las etapas de proceso de diseño mencionados anteriormente, se tomaron los heurísticos de Bruegge y Dutoit que sirven de guía al estudiante a determinar los diferentes elementos de una situación problema, realizar los diagramas UML y correspondientes modelos para dichas etapas.

La tabla 7 ilustra la heurística de Abbott es utilizada para establecer la relación entre elementos del lenguaje natural y elementos del modelo orientado a objetos.

Tabla 7. Heurística de Abbott

Parte del habla	Componente del modelo
Sustantivo propio	Objeto (Alicia)
Sustantivo común	Clase (Persona)
Verbo de acción	Operación (crea, envía)
Verbo de ser	Herencia (es un tipo de)
Verbo de tener	Agregación (incluye)
Verbo modal	Restricciones (debe ser)
Adjetivo	Atributo

Fuente: (Bruegge & Dutoit, Object-oriented Software Engineering, 2010, pág. 180)

La tabla 8, aparece la relación de cada una de las etapas del proceso de diseño de software, la representación utilizada y los correspondientes heurísticos propuestos por Bruegge y Dutoit.

Tabla 8. Relación de heurísticos con las diferentes actividades de procesos de diseño de software

FASE	ACTIVIDAD	REPRESENTACIÓN	HEURÍSTICO / PREGUNTAS
1. Obtención de Requerimientos	1.1. Identificar actores	<ul style="list-style-type: none"> • Descripción en lenguaje natural 	<ul style="list-style-type: none"> • ¿Cuáles grupos de usuarios son apoyados por el sistema para realizar su trabajo? • ¿Cuáles grupos de usuarios ejecutan las funciones principales del sistema? • ¿Cuáles grupos de usuarios realizan funciones secundarias, como el mantenimiento y la administración? • ¿Interactuará el sistema con algún sistema de hardware o software externo?
	1.2. Identificar escenarios	<ul style="list-style-type: none"> • Descripción en lenguaje Natural 	<ul style="list-style-type: none"> • ¿Cuáles son las tareas que el actor desea que realice el sistema? • ¿Qué información consulta el actor? ¿Quién crea esos datos? ¿Se pueden modificar o eliminar? ¿Quién lo hace? • ¿Qué cambios externos necesita informar el actor al sistema? ¿Con qué frecuencia? ¿Cuándo? • ¿Cuáles eventos necesita el actor que le informe el sistema? ¿Con cuánta latencia?
	1.3. Identificar casos de uso y refinarlos	<ul style="list-style-type: none"> • Descripción en lenguaje Natural 	<ul style="list-style-type: none"> • Los casos de uso deben ser nombrados con verbos. El nombre del caso de uso debería indicar lo que el usuario

FASE	ACTIVIDAD	REPRESENTACIÓN	HEURÍSTICO / PREGUNTAS
		<ul style="list-style-type: none"> • Diagrama UML 	<p>desea lograr.</p> <ul style="list-style-type: none"> • Los actores deben ser nombrados con sustantivos. • La frontera del sistema debe ser clara. • Los pasos que debe realizar el usuario y el sistema deben diferenciarse e identificarse correctamente. • Los pasos del caso de uso en el flujo de eventos debe estar en frases en voz activa, esto hace que sea explicito quien lleva a cabo la acción. • La relación causal entre pasos sucesivos debe ser claro. • Un caso de uso debe describir una transacción del usuario completa. • Las excepciones deben ser descritas separadamente. • Un caso de uso NO describe la interfaz de usuario del sistema. Esto hace que el enfoque de los pasos llevados a cabo por el usuario es mejor direccionarlo con maquetas visuales. • Un caso de uso no debe exceder dos o tres páginas de longitud. Por otra parte las relaciones <i>incluye</i> y <i>extiende</i> se pueden descomponer en casos de uso más pequeños.

FASE	ACTIVIDAD	REPRESENTACIÓN	HEURÍSTICO / PREGUNTAS
			<p>*Pueden ser usados también para escribir escenarios.</p> <ul style="list-style-type: none"> • Use escenarios para comunicarse con los usuarios y para validar la funcionalidad. • Haga un corte vertical reducido (escenario) para determinar preferencias de interacción del usuario. • Haga un corte horizontal (varios escenarios no detallados) para definir el alcance del sistema y validarlo con el usuario. • Use maquetas sólo como apoyo visual. • Presente varias alternativas al usuario. • Detalle un corte vertical amplio (escenario) cuando el alcance del proyecto y las preferencias del usuario estén bien comprendidas.
	1.4. Identificar relaciones entre casos de uso	<ul style="list-style-type: none"> • Diagramas UML 	<ul style="list-style-type: none"> • Use relaciones extendidas para comportamientos excepcionales, opcionales o que rara vez suceden. • Use relaciones de inclusión para comportamientos que se comparten entre dos o más casos de uso.
	1.5. Identificar inicialmente objetos de análisis	<ul style="list-style-type: none"> • Lenguaje natural • Diagramas UML 	<p>Para identificar <i>objetos participantes</i>:</p> <ul style="list-style-type: none"> • Términos que los desarrolladores o los usuarios necesitan aclarar para comprender el caso de uso. • Nombres recurrentes en los casos de uso.

FASE	ACTIVIDAD	REPRESENTACIÓN	HEURÍSTICO / PREGUNTAS
			<ul style="list-style-type: none"> • Entidades del mundo real de las cuales el sistema debe dar cuenta. • Procesos del mundo real de los cuales el sistema debe dar cuenta. • Casos de uso • Orígenes o destinos de datos • Artefactos de interfaz • Siempre hay que usar términos del dominio de la aplicación. • Se deben generar <i>objetos participantes</i> para cada caso de uso. • Si dos casos de uso se refieren al mismo concepto, el objeto debe ser el mismo. • Si dos objetos comparten el mismo nombre y no representan el mismo concepto se debe renombrar uno de los dos o ambos con el fin de que sea clara su diferencia. • Realizar una lista de chequeo de los <i>objetos participantes</i> para verificar que estén completos los casos de uso. <p>Revisión cruzada de casos de uso y objetos participantes:</p> <ul style="list-style-type: none"> • ¿Cuál caso de uso crea a este objeto? (durante cuáles caso

FASE	ACTIVIDAD	REPRESENTACIÓN	HEURÍSTICO / PREGUNTAS
			<p>de uso se dan al sistema los valores de los atributos del objeto)</p> <ul style="list-style-type: none"> • ¿Cuáles actores pueden tener acceso a esta información? • ¿Cuáles caso de uso modifican y destruyen ese objeto? • ¿Cuál actor puede iniciar estos casos de uso? • ¿Es necesario este objeto?
	1.6. Identificar requerimientos NO funcionales	<ul style="list-style-type: none"> • Descripción en Lenguaje natural 	Ver Tabla 5. Requerimientos No funcionales
2. Análisis	2.1 Identificar objetos Entidad	<ul style="list-style-type: none"> • Diagrama UML 	<p>Usar la denominada heurística de Abbott (Tabla 7. Heurística de Abbott)</p> <ul style="list-style-type: none"> • Términos que los desarrolladores o usuarios necesitan clarificar para entender los casos de uso. • Nombres recurrentes en los casos de uso • Entidades del mundo real de las que necesita el sistema dar cuenta. • Actividades del mundo real de las que necesita el sistema dar cuenta.

FASE	ACTIVIDAD	REPRESENTACIÓN	HEURÍSTICO / PREGUNTAS
			<ul style="list-style-type: none"> • Fuentes o destinos de datos
	2.2 Identificar objetos de Frontera	<ul style="list-style-type: none"> • Diagrama UML 	<ul style="list-style-type: none"> • Identificar controles de la interfaz del usuario que este necesita para iniciar los casos de uso. • Identificar formularios que los usuarios necesitan para entrar datos al sistema. • Identificar noticias y mensajes que el sistema usa para responder al usuario • Cuando múltiples actores están involucrados en un caso de uso, identificar terminales de actor para referirse a la interfaz de usuario en consideración. • No modelar aspectos visuales de la interfaz con objetos frontera (use mejor maquetas) • Siempre hay que usar los términos del usuario para describir interfaces, no usar términos de la solución o de la implementación.
	2.3 Identificar Objetos de Control	<ul style="list-style-type: none"> • Diagrama UML 	<ul style="list-style-type: none"> • Identificar un objeto de control por cada caso de uso. • Identificar un objeto de control por actor en el caso de uso • La vida de un objeto de control debe cubrir la extensión del

FASE	ACTIVIDAD	REPRESENTACIÓN	HEURÍSTICO / PREGUNTAS
			<p>caso de uso o la extensión de la sesión de usuario. Si es difícil identificar el comienzo y el fin de la activación de un objeto de control, es posible que el caso de uso correspondiente no tenga bien definidas las condiciones de entrada y salida.</p>
	<p>2.4 Mapeando casos de uso a objetos con diagramas de secuencia</p>	<ul style="list-style-type: none"> • Diagramas UML 	<ul style="list-style-type: none"> • La primera columna debe corresponder al actor que inicia el caso de uso. • La segunda columna debe ser un objeto de frontera (que usa el actor para iniciar el caso de uso). • La tercera columna debe ser un objeto de control que maneja el resto del caso de uso. • Los objetos de control son creados por objetos de frontera que inician casos de uso. • Los objetos de frontera son creados por objetos de control. • Los objetos de entidad son accedidos por objetos de control y de frontera. • Los objetos de entidad NUNCA tienen acceso a los objetos de frontera o control, esto hace que sea más fácil compartir objetos de entidad entre casos de uso.
	<p>2.5 Identificar</p>	<ul style="list-style-type: none"> • Diagramas UML 	<ul style="list-style-type: none"> • Examine las frases verbales

FASE	ACTIVIDAD	REPRESENTACIÓN	HEURÍSTICO / PREGUNTAS
	asociaciones		<ul style="list-style-type: none"> • Nombre con precisión a las asociaciones y los roles • Use calificadores lo mayor posible para identificar espacios de nombres y atributos principales. • Elimine cualquier asociación que pueda derivarse de otras asociaciones. • No se preocupe por la multiplicidad hasta que el conjunto de asociaciones sea estable. • Muchas asociaciones hace que el modelo sea ilegible.
	2.6 Identificar agregaciones	<ul style="list-style-type: none"> • Diagramas UML 	<ul style="list-style-type: none"> • Existen dos tipos de agregaciones de composición y de compartida. • La <i>agregación de composición</i> indica que la existencia de las partes dependen del todo. • La <i>agregación compartida</i> indica que el todo y las partes pueden existir independientemente. • Son usadas con frecuencia en la interfaz del usuario para ayudar al usuario a navegar en muchas instancias.
	2.7 Identificar Atributos	<ul style="list-style-type: none"> • Diagramas UML 	<ul style="list-style-type: none"> • Examine las frases posesivas. • Represente el estado almacenado como un atributo de un objeto de entidad • Describa cada atributo.

FASE	ACTIVIDAD	REPRESENTACIÓN	HEURÍSTICO / PREGUNTAS
			<ul style="list-style-type: none"> • No represente un atributo como un objeto, en vez de ello use una asociación. • No pierda tiempo describiendo detalles hasta que la estructura del objeto sea estable.
	2.8 Modelado del comportamiento de Estado-Dependiente de objetos individuales	<ul style="list-style-type: none"> • Diagramas UML 	<ul style="list-style-type: none"> • Los diagramas de máquina de estado representan el comportamiento desde la perspectiva de un solo objeto. • No se necesita construir máquinas de estado para cada clase del sistema, solo para los objetos que tienen una vida extendida y un comportamiento no trivial, esto se da casi siempre para <i>objetos de control</i>, con menos frecuencia para <i>objetos de entidad</i> y casi nunca para <i>objetos de frontera</i>.
	2.9 Modelado de relaciones de Herencia	<ul style="list-style-type: none"> • Diagrama UML 	<ul style="list-style-type: none"> • Si dos o más clases comparten atributos o comportamientos, las similitudes son consolidadas en una superclase.
	2.10 Revisión del modelo de análisis	<ul style="list-style-type: none"> • Diagrama UML 	<p>Preguntas para asegurarse que el modelo sea correcto:</p> <ul style="list-style-type: none"> • ¿El glosario de objetos de entidad es entendible por el usuario? • ¿Las clases abstractas corresponden a conceptos en el nivel de usuario? • ¿Todas las descripciones están de acuerdo con las

FASE	ACTIVIDAD	REPRESENTACIÓN	HEURÍSTICO / PREGUNTAS
			<p>definiciones de los usuarios?</p> <ul style="list-style-type: none"> • ¿Todos los objetos de entidad y frontera tienen como nombres frases nominativas significativas? • ¿Todos los casos de uso y objetos de control tienen como nombres frases verbales significativas? • ¿Todos los casos de error están descritos y manejados? <p>Preguntas para asegurarse que el modelo esté completo:</p> <ul style="list-style-type: none"> • Para cada objeto: ¿Lo necesita algún caso de uso? ¿En qué caso de uso se crea? ¿En cuál se destruye? ¿Se puede tener acceso a él desde un objeto de frontera? • Para cada atributo: ¿Cuándo se asigna? ¿Cuál es su tipo? ¿Debe ser un calificador? • Para cada asociación: ¿Cuándo se le recorre? ¿Por qué se escogió esa multiplicidad específica? ¿Pueden calificarse las asociaciones con multiplicidades de uno a muchos o de muchos a muchos? • Para cada objeto de control: ¿Tiene las asociaciones necesarias para tener acceso a los objetos que participan en su caso de uso correspondiente? <p>Preguntas para asegurarse que el modelo sea consistente:</p>

FASE	ACTIVIDAD	REPRESENTACIÓN	HEURÍSTICO / PREGUNTAS
			<ul style="list-style-type: none"> • ¿Hay varias clases o casos de uso con el mismo nombre? • ¿Las entidades con nombres similares indican fenómenos similares? • ¿Están descritas todas las entidades con el mismo nivel de detalle? • ¿Hay objetos con atributos y asociaciones similares que no están en la misma jerarquía de generalización? <p>Preguntas para asegurarse que el sistema descrito por el modelo de análisis sea realista:</p> <ul style="list-style-type: none"> • ¿Hay alguna característica novedosa en el sistema? ¿En dónde se han realizado estudios o prototipos para asegurar su factibilidad? • ¿Pueden satisfacerse los requerimientos de desempeño y confiabilidad? ¿Se verificaron esos requerimientos mediante algún prototipo ejecutado en el hardware seleccionado?
3. Diseño del Sistema	3.1 Definición de objetivos de diseño	<ul style="list-style-type: none"> • Descripción en lenguaje Natural 	Se deben tener en cuenta los siguientes criterios de diseño posibles (Tabla 6. Criterios de diseño) agrupados en los siguientes tipos:

FASE	ACTIVIDAD	REPRESENTACIÓN	HEURÍSTICO / PREGUNTAS
			<ul style="list-style-type: none"> • Criterios de desempeño. • Criterios de solidez. • Criterios de costo. • Criterios de mantenimiento. • Criterios de usuario final.
	<p>3.2 Descomposición del sistema en subsistemas</p>	<ul style="list-style-type: none"> • Diagrama UML 	<ul style="list-style-type: none"> • Asigne a los objetos identificados en un caso de uso al mismo subsistema • Cree un subsistema dedicado para los objetos que se usan para mover datos entre subsistemas. • Minimice la cantidad de asociaciones que cruzan fronteras de subsistemas • Todos los objetos que están en el mismo subsistema deben estar relacionados desde el punto de vista funcional.

3.3 SELECCIÓN DE LAS TIC

Para ello se tuvo en cuenta las bondades de las TIC mencionadas en el capítulo 3.8, el gran auge de las herramientas *e-learning* en especial los entornos virtuales de aprendizaje (VLE), y la incursión de estos como apoyo en el desarrollo de los procesos de enseñanza y aprendizaje que se realizan en modalidad presencial. Estos factores son elementos propicios para el uso de las TIC como un medio más no como el fin de la enseñanza.

La plataforma *e-learning* utilizada fue la de Moodle (*modular object oriented dynamic learning environment*), ambiente de aprendizaje dinámico y modular orientado a objetos. A continuación se describe en detalle.

3.3.1 Características de Moodle

El Moodle posee las siguientes características y ventajas (cf. Moodle, 2013):

- Es un sistema de administración de cursos en línea
- Es un ambiente de aprendizaje diseñado para comunidades de aprendizaje
- Fue diseñado como un paquete software cuyo recursos y actividades fueron implementados para apoyar experiencias educativas que tiene en cuenta los siguientes principios pedagógicos:
 - **Constructivismo:** sostiene que los seres humanos construimos nuevos conocimientos, de manera activa y en interacción con el entorno.
 - **Construccionismo:** afirma que el aprendizaje es más efectivo cuando se construye algo que ha de compartirse con los demás.
 - **Constructivismo social:** los grupos favorecen la construcción de productos y/o artefactos que comparten un propósito común (cf. Ferreiro, 2001).
 - **Conectados y separados:** El estudiante participa y hace sus aportes individuales desde su lógica, pero también puede escuchar, debatir y reflexionar con el otro.

- Desarrollado con código abierto y licenciamiento libre.
- Puede ser instalado en diferentes plataformas de hardware y software.
- Soporta grandes comunidades de usuarios y sirve tanto para diferentes estilos de aprendizaje en el ámbito educativo y empresarial.
- Cuenta con diferentes actividades de aprendizaje y publicación de recursos.
- Permite la comunicación y el aprendizaje colaborativo.
- Es compatible con diferentes estándares y herramientas tales como SCORM, LAMS, eLGG, OpenD, entre otros.
- Es de fácil personalización, permite a los profesores administrar sus cursos y facilita las tareas de administración.
- Posee más de 80.000 sitios registrados, millones de usuarios, en más de 200 países, maneja alrededor de 80 idiomas, soportado por un equipo de desarrolladores certificados en Australia y toda una comunidad que respalda su crecimiento exponencial, ver figura 13.

Figura 13. Estadísticas de Moodle

Moodle Statistics	
Registered sites	86,268
Countries	237
Courses	7,802,868
Users	73,092,184
Teachers	1,297,013
Enrolments	76,499,235
Forum posts	129,652,858
Resources	69,500,415
Quiz questions	196,862,264

Fuente: página oficial *www.moodle.org*, Agosto de 2013

3.3.2 Estructura de Moodle

Está organizado a manera de cursos. Estos son básicamente, páginas o áreas dentro de Moodle en donde los profesores pueden publicar sus recursos y actividades a los estudiantes. Éstas pueden tener diferentes disposiciones, pero usualmente incluyen un número de secciones ubicadas en el pánel central de la pantalla en donde se muestran los materiales y las actividades programadas, además de *bloques* laterales que ofrecen información o características extra.

Los cursos pueden tener contenido para un año de estudios, para una sesión única o para cualquier variante, dependiendo del establecimiento o el maestro. Pueden ser usados por un maestro o por un grupo de maestros.

Existen diferentes formas de inscripción de los estudiantes, que puede ser realizada por el profesor o por el administrador. Los cursos se pueden organizar por categorías. Además es posible la inclusión de usuarios invitados quienes pueden revisar el material más no participan dentro de las actividades del curso.

Un profesor de un curso Moodle puede seleccionar ítems de tres diferentes elementos, que le ayudan a los procesos de enseñanza y aprendizaje, estos son: *Actividades*, *Recursos* y *Bloques*.

- **Actividades.** Permiten la organización de diversas experiencias educativas que favorecen la participación e interacción de los estudiantes con el profesor o con los demás compañeros, así como el desarrollo de foros, entrega de tareas, responder exámenes, la construcción colaborativa de documentos mediante el uso de *wikis*. Las actividades pueden calificarse. Un profesor añade las actividades al activar la edición y puede elegir la sección del curso donde desea hacerlo. Estas aparecen usualmente en el área central del curso.

- **Recursos.** Los recursos permiten la publicación de información para apoyar las actividades de aprendizaje de los participantes del curso; el profesor puede publicar carpetas, archivos de diferente tipo, el libro de texto, enlaces a sitios de interés, videos sobre la temática. Un recurso difiere de una actividad, en que mediante el recurso el estudiante accede a información pero no interactúa con ésta, solamente puede consultarla, leerla o descargarla.
- **Bloques.** Los bloques proporcionan información extra sobre cómo acceder a los recursos y las actividades, información sobre el avance del estudiante y los participantes en el curso, el acceso al servicio de mensajería, en fin información para la gestión del curso. Los bloques son similares a los "*widgets*" tales como, canales de noticias RSS, resultados de examen, calendario, enlaces a blogs, términos del glosario o archivos privados. También existe un bloque HTML simple, que puede personalizarse como lo desee el profesor.

El profesor puede añadirlos al activar la edición y elegir del menú desplegable de "Añadir un bloque" que usualmente está al fondo del lado derecho en la página del curso.

Teniendo en cuenta los elementos propios del proceso de diseño y habiendo revisado los principios pedagógicos se determinó implementar la estrategia didáctica de *Portafolios* con un enfoque constructivista, con el fin de que se pudiera llevar el registro y desarrollo de un proyecto de software enfatizando en las etapas propias del diseño de software como lo son la obtención de requerimientos, análisis y diseño. Además el portafolio digital le permite al estudiante hacer un proceso reflexionado, ser autónomo de su aprendizaje y desempeñar un rol activo dentro del mismo. Por su parte el docente cuenta con una herramienta para hacer seguimiento y realimentación del aprendizaje y enriquece su labor docente a través de mejores prácticas.

Este capítulo evidencia el cumplimiento de los tres primeros objetivos específicos planteados en la investigación.

4. DISPOSITIVO COMPUTACIONAL

La profesora Souto (2005) señala que un dispositivo debe reunir ciertas condiciones:

- **Es un revelador.** Porque permite que en su interior se fomenten, revelen, desplieguen, diversos significados de carácter implícito y explícito, provenientes de lo subjetivo, lo intersubjetivo, lo social y de relaciones entre las representaciones de los sujetos con el saber.
- **Es un analizador.** En tanto que brinda la posibilidad de reflexionar lo revelado y ello requiere un espacio propicio que lo facilite, que sea receptivo. Esto significa que debe tener capacidad de receptividad para transformar las percepciones y los sentimientos subjetivos en conocimiento. Y de crecimiento de todos los participantes.
- **Organizador técnico.** Administra los recursos, las actividades, el tiempo, los espacios y las personas, llevando a transformaciones de relaciones sociales, de aprendizaje colectivo.

4.1 PORTAFOLIO DIGITAL - *UISport*

El portafolio digital –*UISport* es el resultado del engranaje de los elementos mencionados en el capítulo 4, que se desarrolló con el uso de la herramienta de programación libre php y conexión a las bases de datos a través de *MySql*.

Las etapas de la metodología utilizada para el desarrollo del aplicativo fueron:

- **Inicio:** En esta etapa se definió la funcionalidad del portafolio digital llamado *UISport*, tomando en cuenta los elementos determinados para la estrategia didáctica, el uso de los principios pedagógicos y los heurísticos para el modelado orientado a objetos.

- **Elaboración:** En esta fase se refinaron los casos de uso, se realizó el modelo de análisis, la descripción de la arquitectura y los flujos de trabajo.
- **Construcción:** el dispositivo fue construido en la versión 2.4.1 de Moodle2, a manera de *bloque*, de forma que sea posible utilizarlo como actividad dentro de un curso cuando el profesor lo desee. Es importante resaltar que UISport ha sido instalado, sin inconvenientes, en cursos implementados en versiones posteriores de Moodle.

Por ser un bloque debe previamente copiarse en el servidor para que pueda ser reconocido junto a los bloques estándar que trae Moodle (ver anexo 1).

Se encuentra alojado en un hosting público que hace parte del grupo de investigación en Sistemas y Tecnologías de la Información (STI), el cual es <http://gruposti.org>.

Cabe mencionar que se trabajaron cuatro prototipos del dispositivo, que se fueron refinando a partir del uso de los estudiantes y la experiencia del profesor, hasta llegar a una versión que cumplía con los requerimientos. En la figura 14, se muestra una imagen del bloque *e-portafolio*, incorporado dentro de un curso.

Figura 14. Bloque *e-portafolio* UISport



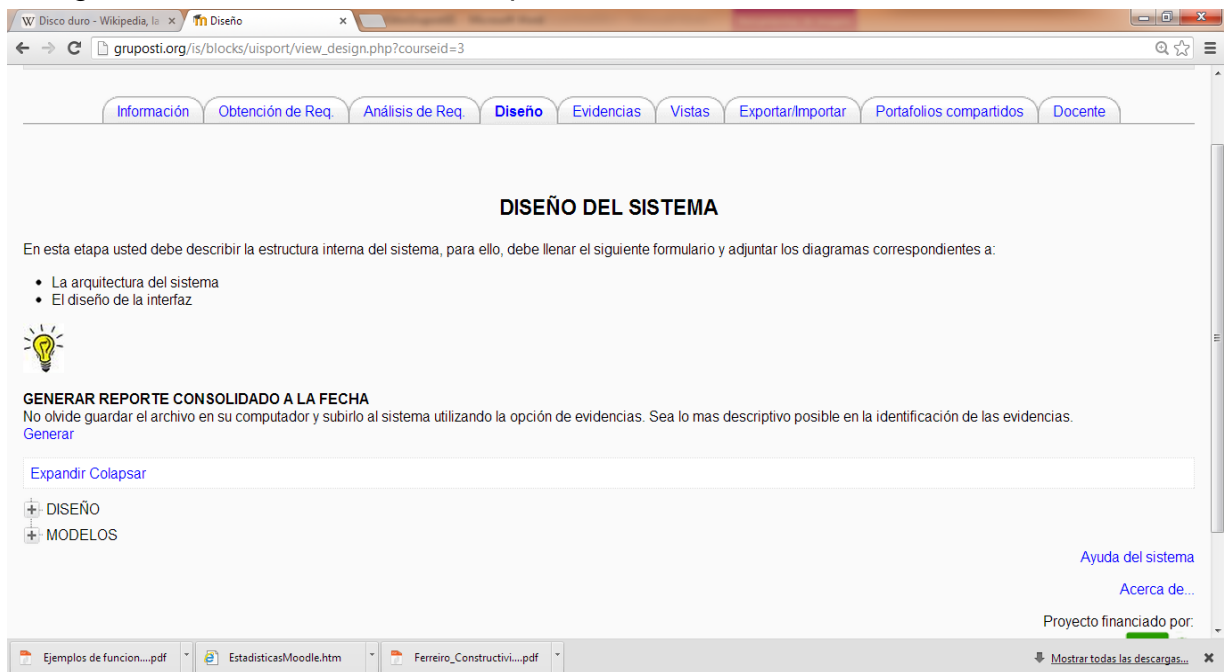
Fuente: Autor

Los aspectos de seguridad del *e-portafolio* son suministrados por la plataforma Moodle y este es incorporado a manera de bloque dentro de la misma lo que permite que sea utilizada en diferentes cursos.

- **Transición:** El dispositivo fue incorporado en Moodle como apoyo al curso presencial de Ingeniería de Software I del programa de pregrado de Ingeniería de Sistemas e Informática, *UISport* a partir del primer semestre del 2012. Su uso ha permitido realizar pruebas de funcionalidad con los estudiantes y profesores, así como realimentar la experiencia que se describe más adelante.

Al ingresar en cualquier opción aparece la siguiente pantalla (ver figura 15), esquematizado por pestañas que permiten la fácil navegación.

Figura 15. Pantalla inicial de UISport

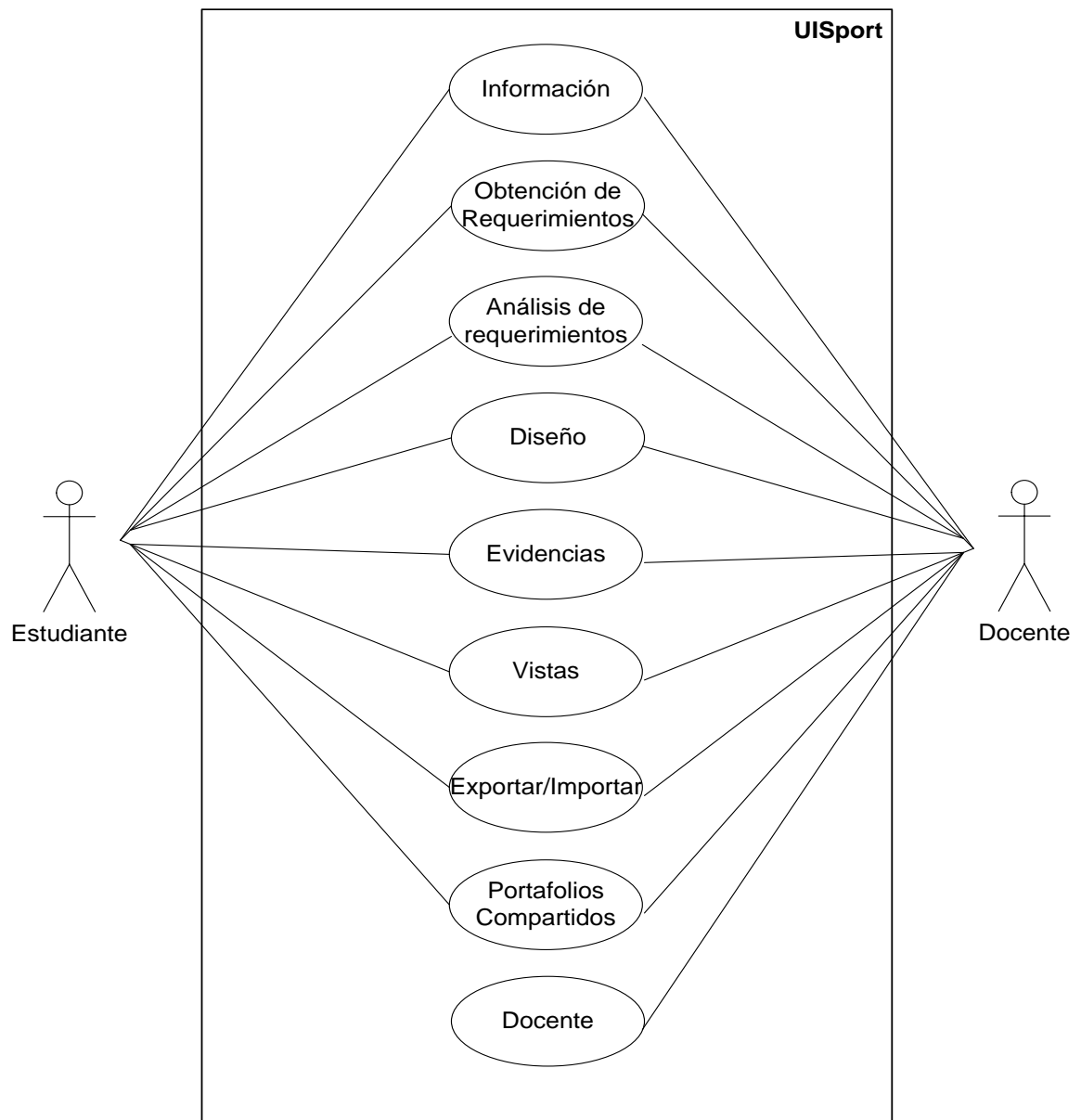


Fuente: Autor

La figura 16, muestra el diagrama general de casos de uso del dispositivo, estos fueron pensados en las etapas que involucran los procesos de diseño de software. Los usuarios deben estar previamente inscritos en un curso, por lo tanto, la seguridad, administración y el acceso son manejados por Moodle.

El dispositivo computacional tiene dos tipos de usuario: Estudiante y Docente y con los permisos asignados por los roles de Moodle dentro de un curso.

Figura 16. Diagrama de Casos de Uso General de *UISport*



Fuente: Autor

En la tabla 9, se describen las funciones del dispositivo *UISport* que corresponden a las diferentes pestañas.

Tabla 9. Funcionalidades del dispositivo *UISport*

Caso de Uso	Descripción
-------------	-------------

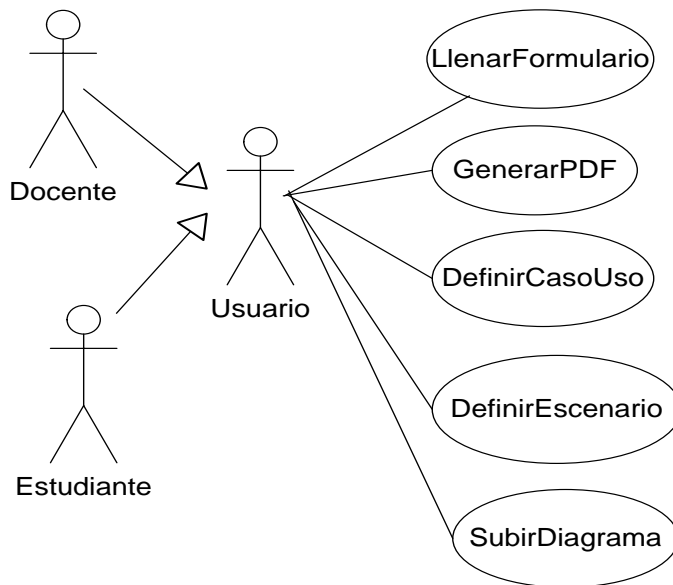
Caso de Uso	Descripción
Obtención Requerimientos	Permite definir los elementos del sistema tales como propósito, alcance y descripción del sistema actual, así como definir los escenarios, casos de uso y le permite al usuario generar el documento de análisis de requerimientos (RAD) en formato PDF con los campos diligenciados por el usuario
Análisis de Requerimientos	Permite actualizar el RAD, plantear el modelo de objetos y el modelo dinámico y generar nuevamente el documento RAD.
Diseño	Se registran los objetivos del diseño, se define la arquitectura del sistema y se suben los diagramas correspondientes a ésta
Evidencias	Permite subir a la plataforma las evidencias tales como documentos, diagramas y archivos relacionados con los productos que se generan en cada etapa.
Vistas	El usuario puede configurar varios esquemas de presentación con diferentes elementos o evidencias, configurar las vistas que desee y compartirlas con usuarios del curso o fuera de este. Además, permite hacer comentarios sobre las evidencias.
Exportar/Importar	Exporta o importa, a formato SCORM, las evidencias que el estudiante desee con el fin de tener copias de seguridad del portafolio personal.
Portafolios Compartidos	Presenta un listado de las vistas que han sido compartidas. Se puede revisar, hacer comentarios, los cuales quedan enlazados en las Vistas.
Docente	Permite al docente, tener una visión general de las

Caso de Uso	Descripción
	actividades y evidencias realizadas por los estudiantes del curso, así como el grupo al que pertenece. También permite administrar el portafolio, crear tipos de evidencia, crear copias de seguridad y reiniciar el portafolio.

Fuente: Autor

Las figuras 17 a la 22, muestran los casos de uso para cada funcionalidad y en el anexo 2, se muestra la descripción y documentación de los casos de uso y escenarios correspondientes.

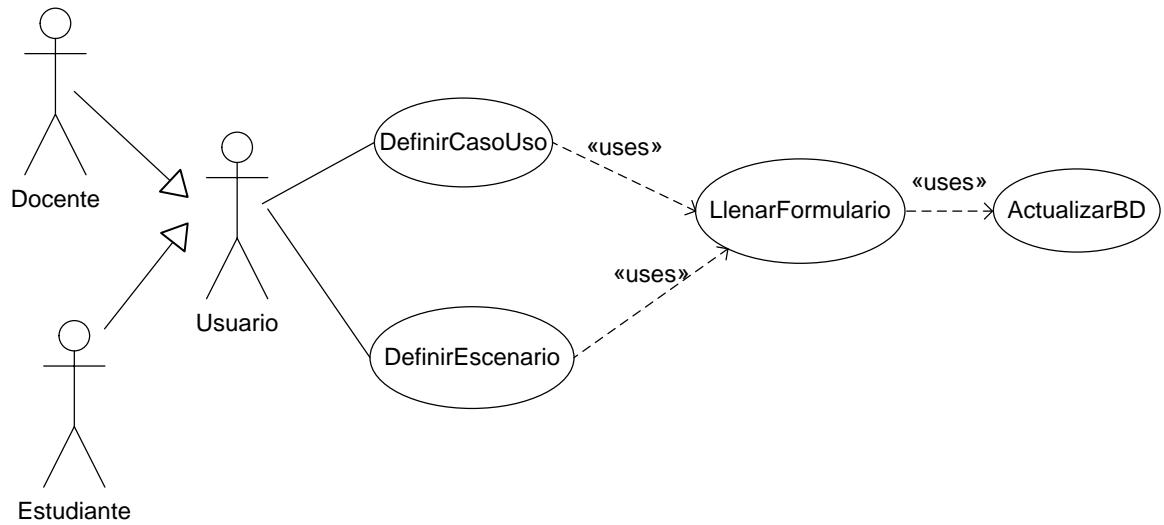
Figura 17. Caso de Uso – *Obtención y Análisis*



Fuente: Autor

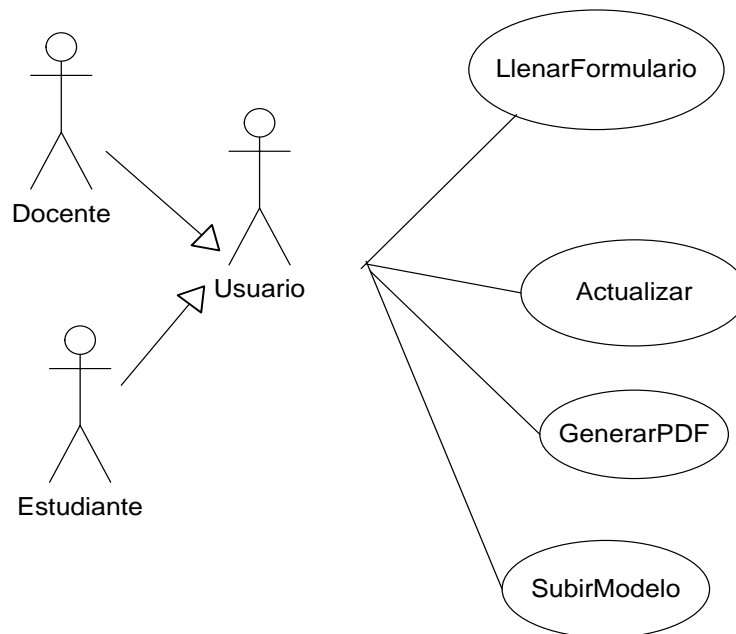
La figura 18 muestra en detalle los casos de uso *DefinirCasoUso* y *DefinirEscenario*, del caso de uso anterior *Obtención y Análisis*.

Figura 18. Caso de Uso – *DefinirCasoUso* y *DefinirEscenario*



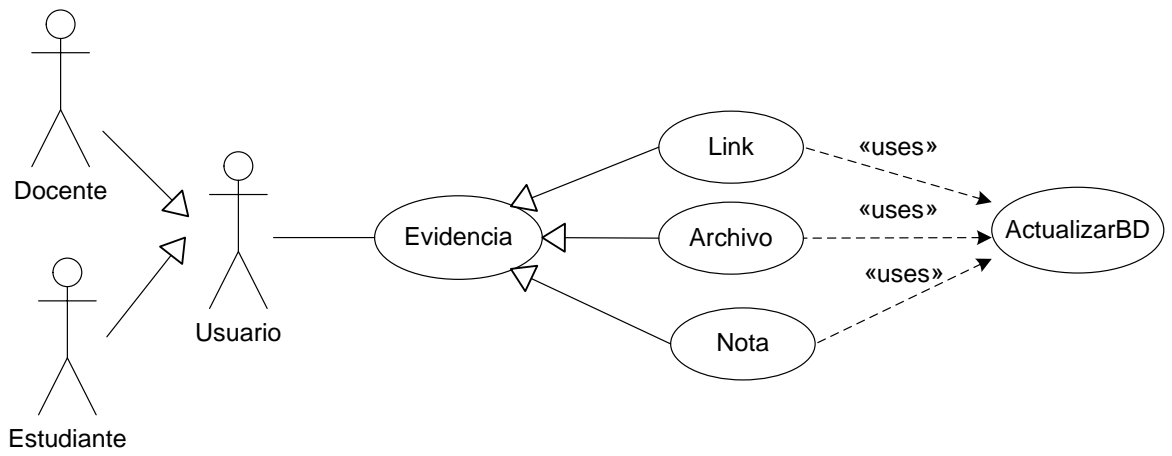
Fuente: Autor

Figura 19. Caso de Uso - *Diseño*



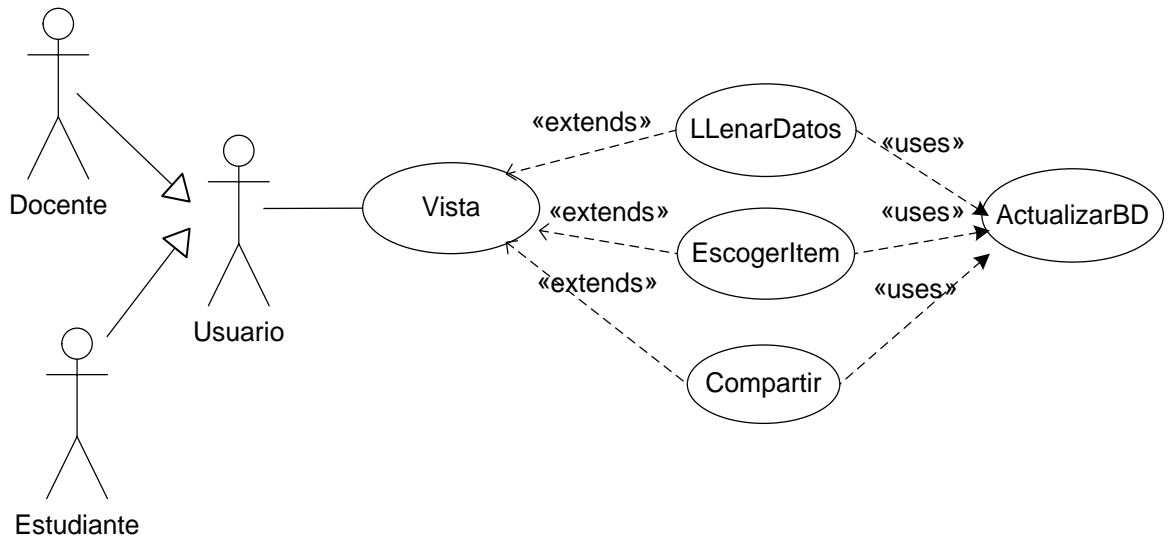
Fuente: Autor

Figura 20. Caso Uso – *Evidencia*



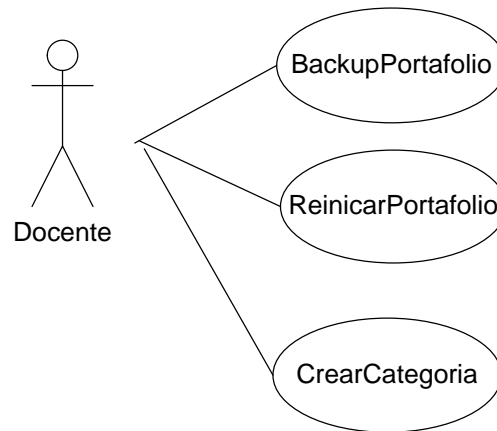
Fuente: Autor

Figura 21. Caso de Uso - Vista



Fuente: Autor

Figura 22. Caso de Uso – Docente



Fuente: Autor

Cada pestaña cuenta con el icono de heurístico que ayuda al usuario en el modelado y le indican los elementos cada etapa del proceso de diseño (figura 23).

Figura 23. Botón de Heurístico



Fuente: Autor

Al hacer clic en el heurístico remite al usuario a una página web donde aparece el heurístico correspondiente a la etapa de diseño que éste diligenciando, ver figuras 24 y 25.

Figura 24. Heurístico de análisis

The screenshot shows a web browser window with the URL `gruposti.org/ris/blocks/uisport/heuristico2.html`. The page has a blue header with the word "ANÁLISIS" in white. Below the header, there are navigation links for "Concepto", "Documentación del análisis", and "Heurísticos". The main content area is titled "CONCEPTO" and contains a paragraph explaining the goal of analysis: to obtain a model of the system that is correct, complete, consistent, and verifiable. It mentions the use of *analysis oriented to objects* and the *analysis model* to describe the domain and actor interactions. A caption "Figura 1. Submodelos del análisis" is followed by a diagram. The diagram shows a central yellow cloud labeled "Modelo de análisis" connected to three grey ovals: "Modelo Funcional", "Modelo de objetos de análisis", and "Modelo dinámico". The word "compuesto" is written between the top oval and the other two. Below "Modelo Funcional" is the text "Representado por Casos de uso y Escenarios". Below "Modelo de objetos de análisis" is "Representado por Diagramas de clase". Below "Modelo dinámico" is "representado Máquinas de estado y diagramas de secuencia". The browser's taskbar at the bottom shows several open applications and the system clock indicating 01:09 p.m. on 29/05/2013.

Fuente: Autor

Figura 25. Heurístico para el diseño

generalización son usadas para eliminar la redundancia del modelo de análisis.
[Ir al Inicio](#)

DOCUMENTACIÓN DEL ANÁLISIS

Este se centra en documentar con detalle los ítems mencionados en el documento RAD específicamente el 3.4.3 Modelo de objetos incluyendo en este el diccionario de datos y los diagramas de clase; así como el 3.4.4 Modelos dinámicos.

Tabla 1. Heurísticos para la Etapa Análisis.

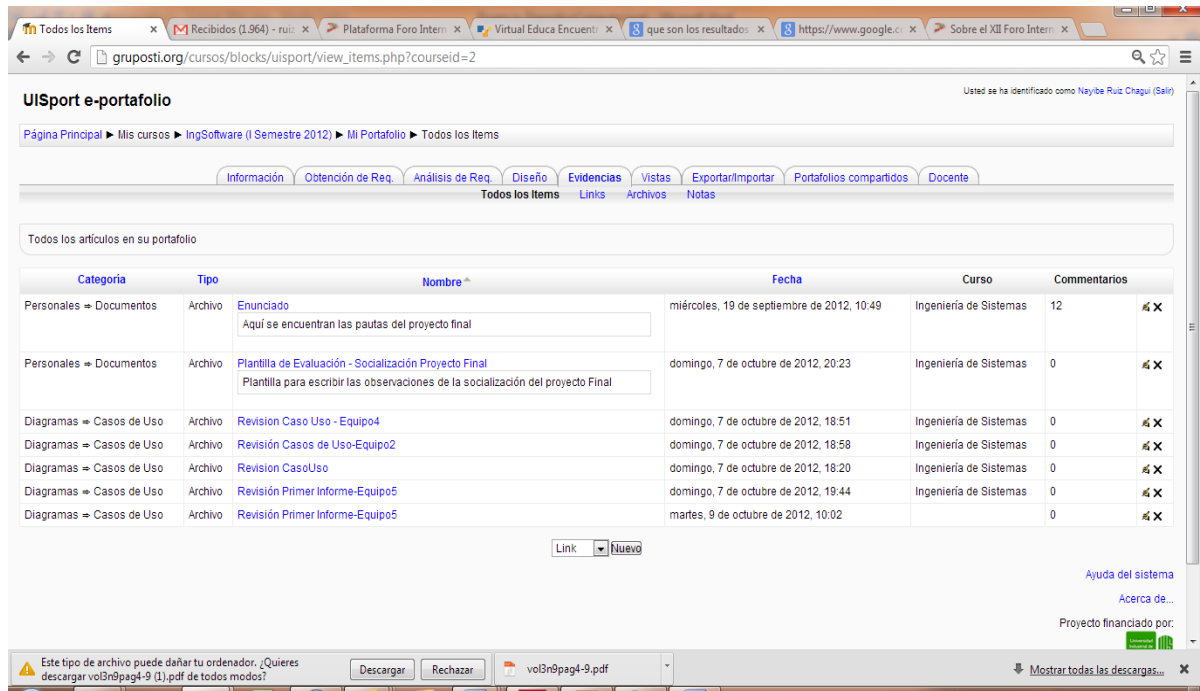
ACTIVIDAD	HEURÍSTICOS O PREGUNTAS
Identificar objetos Entidad	<p>Usar la tabla 2 . Denominada la heurística de Abbott</p> <p>Términos que los desarrolladores o usuarios necesitan clarificar para entender los casos de uso.</p> <p>Nombres recurrentes en los casos de uso.</p> <p>Entidades del mundo real de las que necesita el sistema dar cuenta.</p> <p>Actividades del mundo real de las que necesita el sistema dar cuenta.</p> <p>Fuentes o destinos de datos</p>
Identificar objetos de Frontera	<p>(Bruegge & Dutoit, 2010, pág. 183)</p> <p>Identificar controles de la interfaz del usuario que este necesita para iniciar los casos de uso.</p> <p>Identificar formularios que los usuarios necesitan para entrar datos al sistema.</p> <p>Identificar noticias y mensajes que el sistema usa para responder al usuario</p> <p>Cuando múltiples actores están involucrados en un caso de uso, identificar terminales de actor para referirse a la interfaz de usuario en consideración.</p>

Fuente: Autor

Los usuarios cuentan con un registro de las diferentes evidencias entregadas, la fecha y el usuario que realiza la entrega en cada una de las etapas de diseño del software, esto le permite al docente hacer comentarios y realimentar el proceso (Ver figura 26).

Cabe anotar que los estudiantes forman equipos de trabajo para desarrollar todo el proceso de diseño de un proyecto software de una situación problema real planteada al inicio del curso. Los estudiantes pueden crear vistas de las evidencias que desea compartir con los diferentes integrantes del curso y hacer comentarios de las mismas.

Figura 26. Pestaña de Evidencias



Fuente: Autor

El docente cuenta con una pestaña de administración (ver figura 27) que lista los portafolios compartidos que son configurados en vistas de presentación por los estudiantes, con sus correspondientes evidencias y equipos responsables. Además, puede hacer una copia de los portafolios de todo el curso.

Figura 27. Pestaña del Docente



Fuente: Autor

4.2 EXPERIENCIA CON *UISport*

El dispositivo computacional está siendo utilizado desde el primer semestre del 2012 como apoyo al curso presencial “Ingeniería de software I”, del programa de pregrado de Ingeniería de Sistemas e Informática.

Los cursos están conformados en promedio por veinte (20) estudiantes de los niveles quinto a séptimo. Los estudiantes realizan, durante el semestre, el desarrollo de un proyecto software que resuelve una situación problema real, esto fue muy interesante ya que a parte de los heurísticos, se cuenta con casos de estudio empresariales reales que han sido documentados por Bruegge y Dutoit como guía y apoyo al desarrollo del proyecto.

Para los estudiantes contar con un recurso de aprendizaje que les aporta a un área de conocimiento de la Ingeniería de Sistemas, como lo es el diseño y desarrollo de software, les ha permitido tener un aprendizaje significativo, realizar mejores prácticas de su ejercicio profesional, que se ve reflejado en el desarrollo del proyecto de grado.

Por otra parte, los estudiantes manifiestan que los beneficios que pueden obtener con *UISport* superan los inconvenientes en cuanto a interfaz o navegabilidad de la herramienta, ya que ello se refleja al enfrentarse a las pruebas *SaberPro* donde fue favorable el desempeño en esta área de conocimiento.

Además, el uso de la herramienta favorece el desarrollo de competencias para el trabajo colaborativo lo que se refleja a través de la bitácora de las evidencias y participaciones que registra la herramienta.

Al estar incorporado *UISport* en un VLE, permitió no sólo usar el *e-portafolio*, sino contar con todo el resto de actividades y recursos que ofrece Moodle, se apoyó el curso presencial haciendo un buen uso de las TIC.

Con este capítulo se da cumplimiento a los dos últimos objetivos de la investigación.

5. CONCLUSIONES

La implementación de la estrategia de *portafolio* en un dispositivo computacional contribuyó a que:

- El profesor cuente con un recurso pedagógico para realizar su labor de forma gestionada, en todos los momentos del proceso: la socialización, la evaluación y realimentación del conocimiento.
- El profesor se constituye en un gestor de aprendizaje, y, específicamente, en un diseñador de ambientes para el aprendizaje tanto del concepto de representación como de metodologías específicas para el diseño de software.
- Los estudiantes encuentran en *UISport* un ambiente que les resulta conocido y que les ofrece una variedad de recursos para lograr su aprendizaje. Por su parte, las TIC les ofrecen fácil acceso a gran cantidad de información y elementos como foros, wikis o chats, que implican la lectura y escritura de textos, lo cual fomenta en ellos habilidades comunicativas de redacción, de lectoescritura y comprensión, así como el manejo de la argumentación y el discurso usando un lenguaje técnico propio del saber que se está aprendiendo.
- Los estudiantes del programa de Ingeniería de Sistemas que cursan Ingeniería del Software I cuentan con una herramienta que favorecerá el aprendizaje del concepto de *representación* en procesos de diseño de software, pues permite que ellos puedan proponer, confrontar y realimentar los modelos a partir de los que propongan los participantes en el curso. Además, se posibilita el ejercicio de llevar la teoría a la práctica lo que favorece el desarrollo de competencias del perfil profesional.

- A través de este proyecto de investigación la Escuela de Ingeniería de Sistemas e Informática y la Maestría en Ingeniería de Sistemas e Informática de la UIS, contribuyen a la formación en los campos temáticos de la Ingeniería del Software, Informática Educativa y la Pedagogía.

- El desarrollo de esta experiencia permitirá al grupo de investigación STI continuar trabajando, a nivel de pregrado y maestría en proyectos que tengan como propósito la aplicación y el estudio del uso de las TIC en procesos de enseñanza y aprendizaje en Ingeniería, de forma que sean referentes para la académica interesada en el tema.

- Lo significativo de la investigación no está en el desarrollo del aplicativo, sino cómo fue posible llevar el diseño de la estrategia didáctica a materializarse en un dispositivo computacional o portafolio digital, el cual en su parte conceptual es una valiosa estrategia de enseñanza y aprendizaje.

- A su vez, el desarrollo de la investigación busca profundizar en el área de conocimiento de la ingeniería de software y se espera que la divulgación de esta, muestre un panorama y algunas características a tener en cuenta en proyectos similares que busquen apoyar, con los servicios de las TIC, los procesos de enseñanza, aprendizaje y evaluación durante el desarrollo de las asignaturas de los programas de pregrado que se ofrecen en modalidad presencial.

- El desarrollo de la investigación generó los siguientes resultados:
 - (1) Realización del proyecto de extensión financiado por la vicerrectoría de investigación y extensión de la UIS, con código VIE 5556 y titulado “Incorporación de una estrategia didáctica para el aprendizaje de la asignatura de Ingeniería de software I sobre una plataforma virtual”

(2) Participación en el primer Congreso sobre tecnología, educación y sociedad (CTES2012) organizado por el centro de Estudios e Investigaciones para el desarrollo Docente (CENID) y del cual se realizó una publicación en la “Revista Iberoamericana para la investigación y el desarrollo educativo” (disponible en el siguiente enlace: http://www.ride.org.mx/index.php?option=com_content&view=article&id=32).

(3) Participación en el XII Foro de Investigadores en Informática Educativa – Versión Internacional organizado por RIBIE-Col (Junio de 2013) y posterior publicación en la revista “Estado del arte de la informática Educativa en Colombia”.

(4) Aceptación de ponencia y artículo en el congreso “Innovación en investigación y educación en ingeniería: factores claves para la competitividad global”. Organizado por WEEF 2013 y ACOFI celebrado en septiembre del presente año en Cartagena.

6. RECOMENDACIONES

- Se requiere hacer un proceso de reingeniería con el fin de corregir detalles que fueron manifestados por los usuarios en aspectos tales como la interfaz y navegabilidad.
- Incorporar al dispositivo un aplicativo que permita elaborar diagramas UML de manera sencilla y en un formato compatible.
- Crear espacios en la escuela de Ingeniería de Sistemas e Informática para profundizar en el ámbito de los ambientes virtuales de aprendizaje y las plataformas *e-learning*, lo que se puede lograr a través de los grupos de investigación y la realización de proyectos de pregrado.
- Alentar a que otros profesores de la universidad, que apoyan cursos con características similares a que incursionen en el uso de las TIC para apoyar su quehacer docente, lo cual implica una planeación y asignación de recursos y tiempo.

7. REFERENCIAS BIBLIOGRÁFICAS

- Aguilar, E., Corredor, M., Porras, H., Ewert, C., & Ramón, J. (2003). *Aula Virtu@l. Una alternativa en educación superior*. Bucaramanga: Ediciones UIS.
- Amarel, S. (1996). On the Mechanization of Creative Processes. *IEEE Spectrum* 3, 112-114.
- Amós Comenio, J. (2003). *Didáctica Magna*. México: Porrúa.
- Boneu, J. M. (2007). Plataformas abiertas de e-learning para el soporte de contenidos educativos abiertos. *Revista de Universidad y Sociedad del Conocimiento (RUSC)*, Volumen 4 (Nº 1), 36-47.
- Bruegge, B., & Dutoit, A. (2010). *Object-oriented Software Engineering* (Tercera ed.). Mexico: Pearson Education.
- Bruegge, B., & Dutoit, A. (2010). *Object-oriented Software Engineering* (Tercera ed.). Mexico: Pearson Education.
- Caillot, M. (1996). La théorie de la transposition didactique est-elle transposable? En C. Raisy, M. Caillot, & débats, *Au-delà des didactiques, le didactique débats autour de concepts fédérateurs* (págs. 20-35). París: De Boeck Université.
- Cardelli, J. (2004). Reflexiones críticas sobre el concepto de Transposición Didáctica de Chevallier. *Scielo - Cuadernos de Antropología Social* (19), 49-61.
- Carvajal Villaplana, A. (2002). La informática Educativa: Una reflexión crítica. *Revista Electrónica Actualidades Investigativas en Educación*, 1-21.
- CESLCAM. (s.f.). *Estudio de herramientas de elearning*. Recuperado el 25 de Junio de 2012, de Centro de Excelencia de Software Libre de la Castilla - La Mancha: <http://es.scribd.com/doc/73202956/08-E-Learning-CESLCAM>

Chevallard, Y. (1998). La transposición Didáctica. Del saber sabio al saber enseñado. *Psicología Cognitiva y Educación* , 1-28.

Chi, M. T., Feltovich, P. J., & Glaser, R. (11 de Febrero de 2010). *Categorization and representation of physics problems by experts and novices*. Recuperado el 20 de Septiembre de 2012, de <http://onlinelibrary.wiley.com/>: http://onlinelibrary.wiley.com/doi/10.1207/s15516709cog0502_2/pdf

Corredor Montagut, M., Arbeláez López, R., & Perez Angulo, M. I. (2010). *Enseñanza en línea: otra opción para la formación en el ámbito universitario*. Bucaramanga: Publicaciones UIS.

Danielson, C., & Abrutyn, L. (1997). *An Introduction to Using Portfolios in the Classroom*. Alexandria, Virginia: Association for Supervision and Curriculum Development.

Delgado Cejudo, S. (2003). *Elearning Análisis de Plataformas Virtuales*. España: Universidad de Valencia.

Ferreiro, R. (2001). *Más allá de la teoría: El Aprendizaje Cooperativo*. (R. Magister, Ed.) Recuperado el 23 de Enero de 2013, de <http://redtalento.com/Articulos/WEBSITE%20Revista%20Magister%20Articulo%206.pdf>

Gallego Badillo, R. (1998). *Discurso Constructivista sobre las tecnologías*. Bogotá: Cooperativa Editorial Magisterio.

Gamboa Sarmiento, S. (2004). *Creatividad y Entornos Virtuales de Aprendizaje*. Bogota: Universidad Pedagógica Nacional.

Gros, B. (1997). *Diseños y programas educativos - Pautas pedagógicas para la elaboración de software*. Barcelona: Ariel Educación.

- Hall, B. (2001). *Online Education Systems: Discussion and Definition of Term*. Recuperado el 25 de Junio de 2012, de <http://www.studymontor.com/Terms.pdf>
- Halté, J. (1998). L'espace didactique et la transposition. *Pratiques* , 171-192.
- Hernandez, I. (2005). *Estética, Ciencia y Tecnología*. Bogotá: Pontificia Universidad Javeriana.
- IEEE. (2004). *SMBOK - Guide to the Software Engineering Body of knowlegde*. California.
- IEEE. (2002). *Standard for Learning Object Metadata*. New York: IEEE Inc.
- Jacobson, I., Booch, G., & Rumbaugh, J. (2000). *El proceso unificado de desarrollo de software*. Madrid: Addison Wesley.
- Jacobson, W., & Florman, J. (2011). Thriving in Academe - How can learning portfolios help you and your students? *National Education Association* , Vol. 28 (N° 5), 5-7.
- Joyanes Aguilar, L. (2000). *Programación Orientada a Objetos*. Madrid: McGrawHill.
- Klenowski, V. (2005). *Desarrollo de portafolios par el aprendizaje y la evaluación*. Madrid: Narcea, S.A.
- Kumar, R., & Gupta, D. (2011). Object Oriented Design. *International Journal of Engineering Science and Technology* , 459-463.
- Liotard, J. F. (1987). La Condición Postmoderna un informe de conocimiento. En J. Valcarce (Ed.). Madrid, España: Ediciones Cátedra S.A.
- Marro, F. (1982). Repercusiones de la teoría de Piaget en los diseños curriculares. *Infancia y Aprendizaje: Journal for the Study of Education and Development* (19-20), 21-38.

Moodle. (29 de Enero de 2013). *www.moodle.org*. Recuperado el 16 de Mayo de 2013, de <http://docs.moodle.org/all/es/Filosofía>

Petitjean, A. (1998). La transposition didactique en français. *Pratiques* , 7-33.

Rosenberg, M. (2002). *E-learning. Estrategias para transmitir conocimiento en la era digital*. Bogotá: McGrawHill.

Ruipérez, G. (2003). *Educación Virtual y eLearning*. Madrid: Fundación Auna.

Ruiz Carrillo, E., & Estrevel Rivera, L. B. (2010). Vigotsky: la escuela y la subjetividad. *Pensamiento Psicológico* , 8 (15), 135-145.

Rumbaugh, J., Jacobson, I., & Grady, B. (2004). *The Unified Modeling Language Reference Manual* (2nd Edition ed.). Addison-Wesley Professional.

Salcedo Casallas, J. R. (Enero-Junio de 2009). *Pedagogía de la potencia y didáctica no parametral: entrevista a Estela Quintar*. Recuperado el 25 de Julio de 2011, de http://tariacuri.crefal.edu.mx/rieda/ene_jun_2009/aula_magna/aula_magna_art1_p5.htm

Sánchez Ilabaca, J. (1992). *Informática Educativa* (2a ed.). Santiago de Chile: Universitaria.

Serrano, J. M., & Pons, R. M. (2011). El Constructivismo hoy: enfoques constructivistas en educación. *Revista Electrónica de Investigación Educativa* , 13 (1), 1-27.

Simon, H. A. (2006). *Las ciencias de lo artificial*. Granada: COMARES.

Sommerville, I. (2005). *Ingeniería de Software* (Séptima ed.). Madrid: Pearson Education.

Souto, M., Tenaglia, G., & Volpin, G. (2005). Una propuesta curricular basada en los conceptos de "Dispositivo" y "Formación". *V Coloquio Internacional sobre gestión universitaria en america del sur*, (págs. 1-46). Mar de Plata.

Tozzi, M. (1995). De la philosophie à son enseignement : le sens d'une didactisation. En M. Develay, *Savoirs scolaires et didactiques des disciplines. Une encyclopédie pour aujourd'hui*. (págs. p.237-260). París: ESF Éditeu.

UIS. (2008). Proyecto Educativo EISI. En E. d. Informática. Bucaramanga: Ediciones UIS.

Vargas Guillén, G. (2002). *El disponer*. Recuperado el Marzo de 2011, de <http://profesorvargasguillen.wordpress.com/2011/06/>

Vargas Guillén, G., & Gamboa Sarmiento, S. (2008). Didáctica en la condición postmoderna. De las competencias a la cooperación. *Revista Científica Guillermo de Ockham*, Vol. 6 (Nº 1), 51-59.

Vargas Guillén, G., & Gamboa Sarmiento, S. (2008). Didáctica en la condición postmoderna. De las competencias a la cooperación. *Revista Científica Guillermo de Ockham*, Vol. 6 (Nº 1), 51-59.

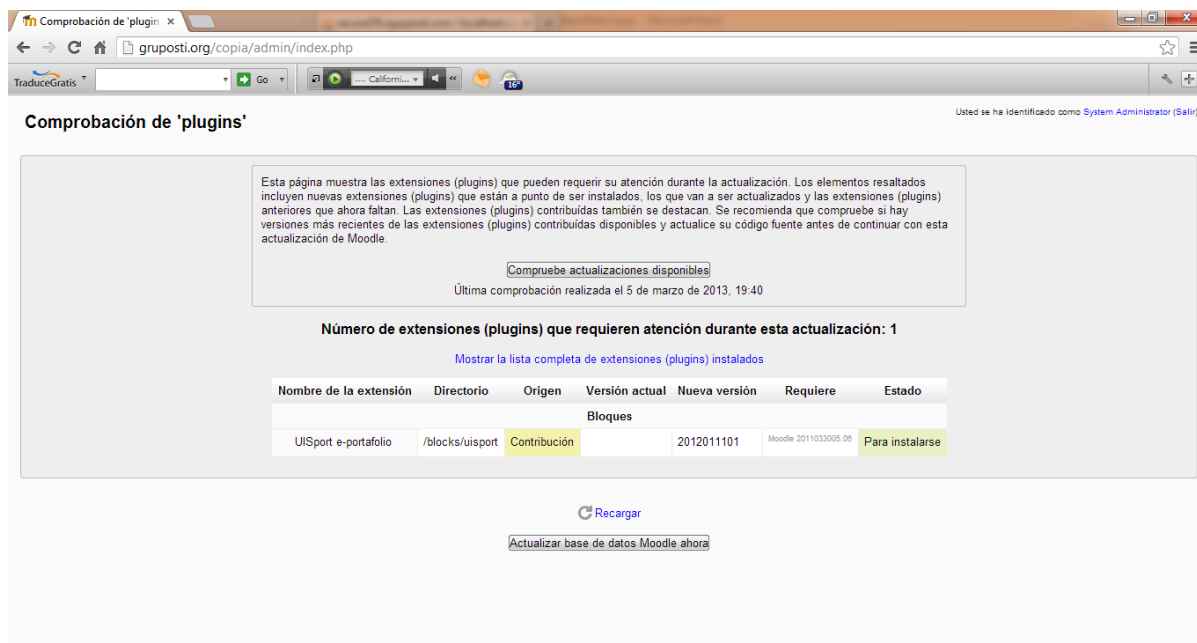
Verret, M. (1975). *Le temps des études, Paris*,. París: Librairie Honoré Champion.

Weitzenfeld, A. (2005). *Ingeniería de Software orientada a Objetos con UML, Java e Internet*. México: Thomson.

Zabalza, M. (2004). *La enseñanza universitaria. El escenario y sus protagonistas*. (segunda ed.). Madrid: Narcea.

ANEXO A. INSTALACIÓN DE *UISport*

1. Teniendo ya un curso creado en Moodle y después de copiar en el servidor en la carpeta de *blocks*, la carpeta *UISport* con el instalador del portafolio, se ingresa a la plataforma de Moodle como administrador y aparece un mensaje que informa sobre la actualización y suministra la información del bloque a instalar.



Comprobación de 'plugins'

Esta página muestra las extensiones (plugins) que pueden requerir su atención durante la actualización. Los elementos resaltados incluyen nuevas extensiones (plugins) que están a punto de ser instalados, los que van a ser actualizados y las extensiones (plugins) anteriores que ahora faltan. Las extensiones (plugins) contribuidas también se destacan. Se recomienda que compruebe si hay versiones más recientes de las extensiones (plugins) contribuidas disponibles y actualice su código fuente antes de continuar con esta actualización de Moodle.

[Compruebe actualizaciones disponibles](#)
Última comprobación realizada el 5 de marzo de 2013, 19:40

Número de extensiones (plugins) que requieren atención durante esta actualización: 1

[Mostrar la lista completa de extensiones \(plugins\) instalados](#)

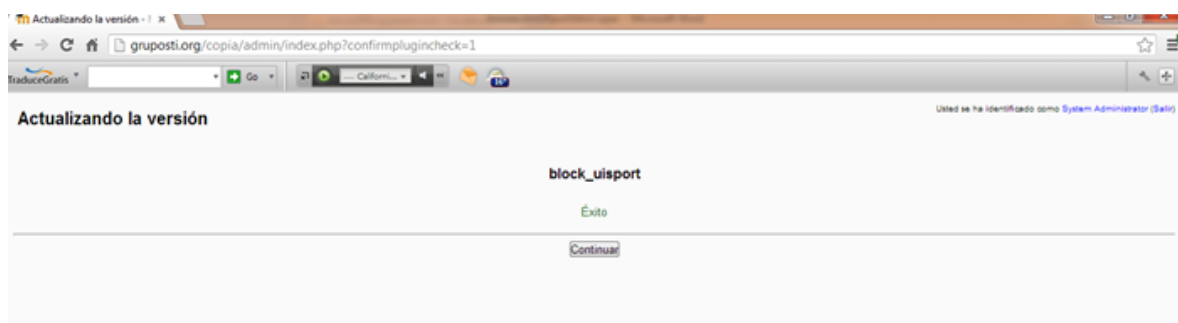
Nombre de la extensión	Directorio	Origen	Versión actual	Nueva versión	Requiere	Estado
Bloques						
UISport e-portafolio	/blocks/uisport	Contribución		2012011101	Moodle 2011033005.06	Para instalarse

[Recargar](#)

[Actualizar base de datos Moodle ahora](#)

Fuente: Autor

2. Se hace clic en el botón final para actualizar la base de datos y se da clic en continuar para que siga con el proceso de actualización.



Actualizando la versión

block_uisport

Éxito

[Continuar](#)

3. Reporta el mensaje de que se ha realizado con éxito el proceso mediante la siguiente pantalla.



Fuente: Autor

4. Si se desea, se puede registrar el sitio en *moodle.org* para las estadísticas.



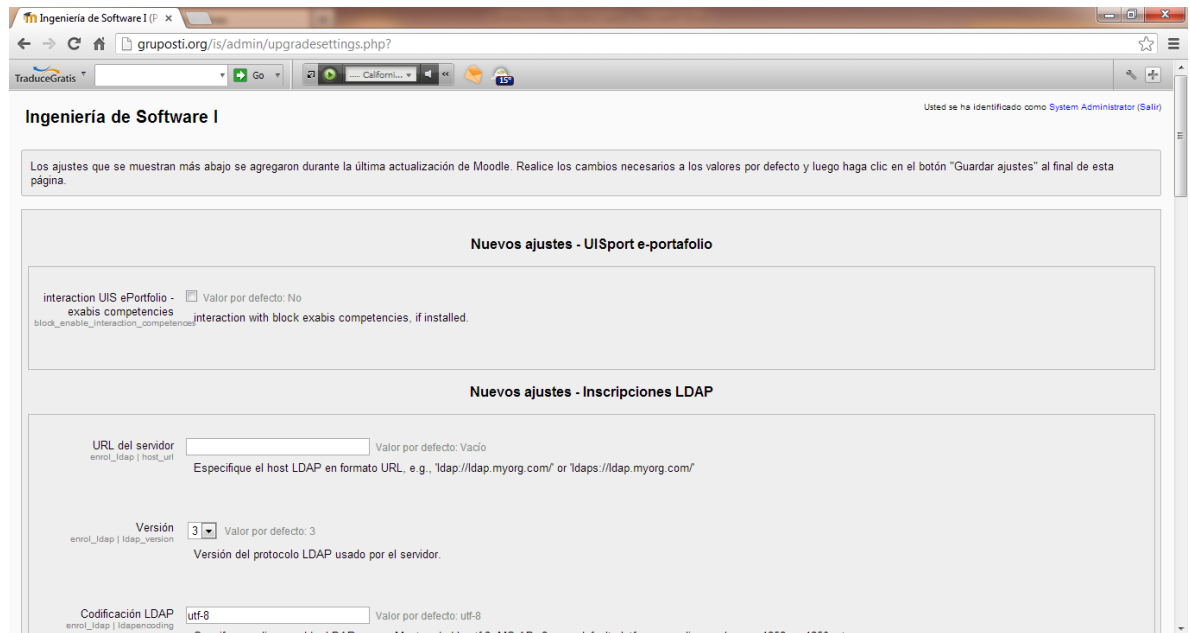
Fuente: Autor

5. Después de guardar los cambios aparece la siguiente imagen



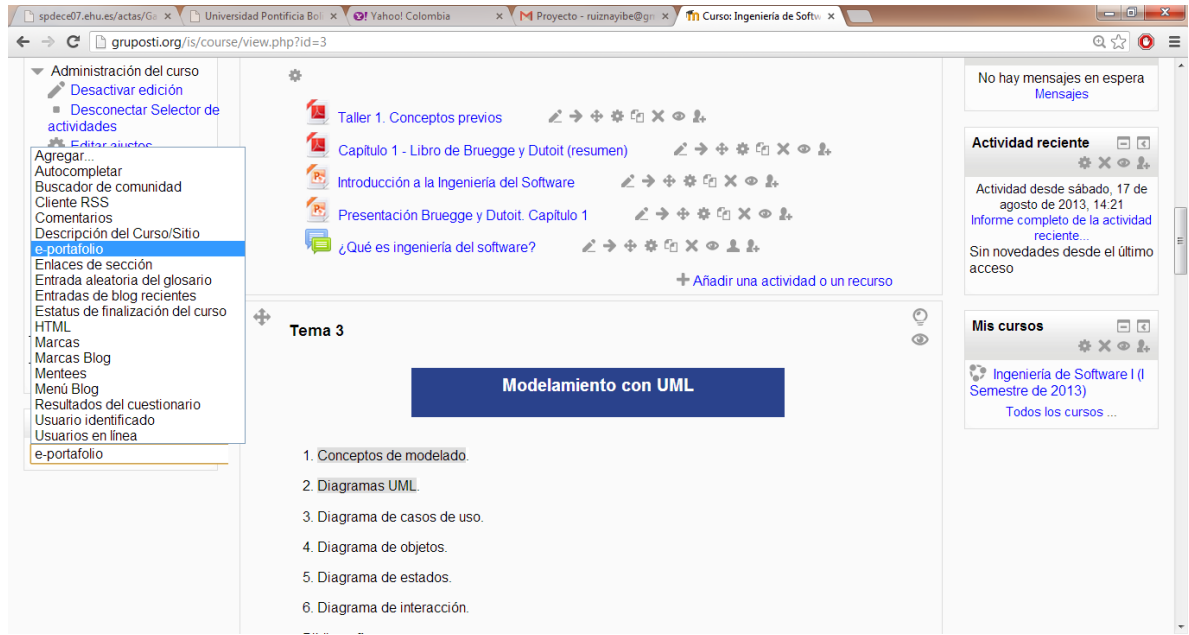
Fuente: Autor

6. Aparecen las opciones para configurar el bloque dentro del curso y queda disponible para su uso. En caso de que se desee editar o eliminar hay que ingresar como administrador a la sección de bloques.



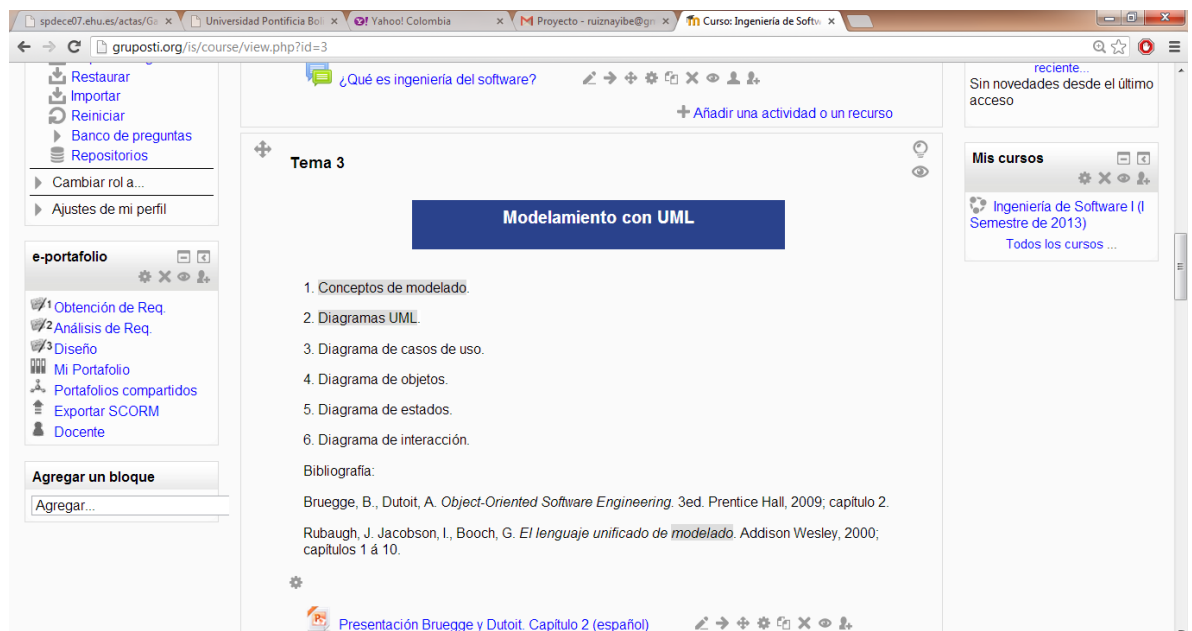
Fuente: Autor

7. Estando en rol de administrador o profesor, se agrega el bloque, dentro del curso, el cual aparece en la lista de bloques (ver imagen).



Fuente: Autor

8. Para este caso aparece en el panel izquierdo, donde se puede configurar la posición y queda listo para usar el dispositivo.



ANEXO B. DESCRIPCIÓN DE CASOS DE USO DE *UISport*

Caso de Uso – *ObtencionyAnálisis*

<i>Nombre del caso de uso:</i>	ObtencionyAnálisis
<i>Actor Participante:</i>	Usuario: Estudiante o Docente
<i>Condición Inicial o de Entrada:</i>	El Usuario ingresa al curso de Moodle y hace clic en el bloque <i>UISport</i>
<i>Flujo de eventos:</i>	<ol style="list-style-type: none"> 1. El Usuario hace clic en la pestaña de <i>Obtención de Requerimientos</i> o <i>Análisis de Requerimientos</i> 2. El Usuario llena cada uno de los campos del formulario correspondiente a la etapa de análisis y diseño del sistema y hace clic en el botón Guardar cambios 3. El Usuario hace clic en la opción Generar y se genera un documento en formato .pdf dando la opción para guardarlo en un medio de almacenamiento. 4. El Usuario puede escoger al caso de uso DefinirCasoUso o DefinirEscenario 5. El Usuario hace clic en SubirDiagrama y le permite escoger el archivo que contenga el diagrama o modelo correspondiente al caso de uso.
<i>Condición de Salida:</i>	UISport informa que se han guardado los ítems llenados del formulario al Usuario y que el archivo .pdf ha sido generado con éxito.
<i>Requerimientos Especiales</i>	El usuario debe estar registrado en el curso en rol de estudiante o profesor.

Fuente: Autor

Caso de Uso detallado – *DefinirCasoUso*

<i>Nombre del caso de uso:</i>	DefinirCasoUso o DefinirEscenario
<i>Actor Participante:</i>	Usuario: Estudiante o Docente
<i>Condición Inicial o de Entrada:</i>	El Usuario se encuentra dentro del bloque <i>UISport</i> en la pestaña <i>Obtención de Requerimientos</i> o <i>Análisis de Requerimientos</i>

<i>Flujo de eventos:</i>	<ol style="list-style-type: none"> 1. El Usuario hace clic en <i>DefinirCasoUso</i> o <i>DefinirEscenario</i> 2. El Usuario llena cada uno de los campos del formulario correspondiente al Caso de Uso o Escenario y hace clic en el botón Guardar. Además puede Actualizar los campos. 3. El Usuario hace clic en Generar, para que los datos queden actualizados en el archivo .pdf.
<i>Condición de Salida:</i>	UISport informa que se han guardado los ítems llenados con éxito. Los datos son actualizados en el archivo .pdf generado.
<i>Requerimientos Especiales</i>	El usuario debe estar registrado en el curso en rol de estudiante o profesor. La modificación o actualización de Casos y Uso y Escenarios sólo se puede hacer desde la pestaña de <i>Obtención de Requerimientos</i> .

Fuente: Autor

Caso de Uso - *Diseño*

<i>Nombre del caso de uso:</i>	Diseño
<i>Actor Participante:</i>	Usuario: Estudiante o Docente
<i>Condición Inicial o de Entrada:</i>	El Usuario ingresa al curso de Moodle y hace clic en el bloque <i>UISport</i>
<i>Flujo de eventos:</i>	<ol style="list-style-type: none"> 1. El Usuario hace clic en la pestaña de <i>Diseño</i>. 2. El Usuario llena cada uno de los campos del formulario de diseño del sistema y hace clic en el botón Guardar cambios, además puede Actualizar campos que hayan sido editados. 3. El Usuario hace clic en la opción Generar y se genera un documento en formato .pdf dando la opción para guardarlo en un medio de almacenamiento. 4. El Usuario hace clic en SubirModelo y le permite escoger el archivo que contenga el diagrama o modelo correspondiente a la etapa de diseño.
<i>Condición de Salida:</i>	UISport informa que se han guardado los ítems llenados del formulario al Usuario , que el archivo .pdf ha sido generado con éxito y los diagramas o modelos aparecen en la pestaña registrados.

<i>Requerimientos Especiales</i>	El usuario debe estar registrado en el curso en rol de estudiante o profesor.
----------------------------------	---

Fuente: Autor

Caso de Uso - Evidencia

<i>Nombre del caso de uso:</i>	Evidencia
<i>Actor Participante:</i>	Usuario: Estudiante o Docente
<i>Condición Inicial o de Entrada:</i>	El Usuario ingresa al curso de Moodle y hace clic en el bloque <i>UISport</i>
<i>Flujo de eventos:</i>	<ol style="list-style-type: none"> 1. El Usuario hace clic en la pestaña de <i>Evidencia</i> 2. El Usuario escoge una de las tres opciones de evidencia: <i>Link</i>, <i>archivo</i> o <i>Nota</i> 3. El Usuario llena los campos correspondientes al nombre, categoría o tipo y selecciona el archivo a subir a la plataforma. 4. El Usuario hace clic en Guardar cambios
<i>Condición de Salida:</i>	UISport informa que se han subido las evidencias con éxito y muestra en la pantalla la evidencia con la fecha, categoría y la opción de modificar o eliminar.
<i>Requerimientos Especiales</i>	El usuario debe estar registrado en el curso en rol de estudiante o profesor. Sólo el creador de la evidencia puede eliminar la misma.

Fuente: Autor

Caso de Uso - Vista

<i>Nombre del caso de uso:</i>	Vista
<i>Actor Participante:</i>	Usuario: Estudiante y/o Docente
<i>Condición Inicial o de Entrada:</i>	El Usuario ingresa al curso de Moodle y hace clic en el bloque <i>UISport</i>

<i>Flujo de eventos:</i>	<ol style="list-style-type: none"> 1. El Usuario hace clic en la pestaña de <i>Evidencia</i> 2. El Usuario hace clic en Agregar Vista 3. El Usuario llena los campos correspondientes al nombre de la vista y selecciona las evidencias o elementos que desea que contenga la vista 4. El Usuario hace clic en Compartir y <i>UISport</i> le muestra los usuarios de Moodle con los cuales puede compartir la vista. 5. El Usuario hace clic en Guardar cambios
<i>Condición de Salida:</i>	<i>UISport</i> informa que se ha creado la vista con éxito y muestra en la pantalla los datos, fecha de creación de la vista y, los usuarios que tienen acceso a ella y la opción de modificar o eliminar.
<i>Requerimientos Especiales</i>	El usuario debe estar registrado en el curso en rol de estudiante o profesor. Sólo el creador de la vista puede eliminar la misma.

Fuente: Autor

Caso de Uso - *Docente*

<i>Nombre del caso de uso:</i>	Docente
<i>Actor Participante:</i>	Docente
<i>Condición Inicial o de Entrada:</i>	El Docente ingresa al curso de Moodle y hace clic en el bloque <i>UISport</i>
<i>Flujo de eventos:</i>	<ol style="list-style-type: none"> 1. El Docente hace clic en la pestaña de <i>Docente</i> 2. El Docente observa el listado de los estudiantes, el grupo correspondiente y las evidencias que haya registrado. 3. El Docente puede escoger la opción Generar y con ello realiza una copia de seguridad de todos los portafolios de sus estudiantes, en un archivo en .pdf. 4. El Docente escoge la opción Editar, en cuyo caso puede agregar o modificar una categoría para los diferentes tipos de evidencias a manejar, <i>UISport</i> le informa que los cambios se han realizado con éxito. 5. El Docente escoge la opción Reiniciar y con ello elimina todos los registros del portafolio actual con el fin de iniciar con un nuevo curso, <i>UISport</i> le pide la

	confirmación de la acción.
<i>Condición de Salida:</i>	<ul style="list-style-type: none"> • UISport informa que se ha creado con éxito el archivo en .pdf de la copia de todos los portafolios. • Muestra en pantalla la nueva categoría o los cambios realizados a éstas. • Muestra el mensaje de eliminación de todos los portafolios..
<i>Requerimientos Especiales</i>	El Docente debe estar registrado en el curso en rol de profesor y sólo él puede ver esta pestaña.

Fuente: Autor