

UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERIA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS
INGENIERÍA EN COMPUTACIÓN



Desarrollo de un Software Educativo en entorno Web para la
asignatura de Taller de Programación Orientada a Objetos,
impartida en el Departamento de Computación y Sistemas Escuela
de Ingeniería y Ciencias Aplicadas de la Universidad de oriente
Núcleo Anzoátegui

Presentado por:

Fajardo Salas, Yitzy.

C.I.:17.092.980

Macayo Marcano, Noireth

C.I: 17.733.552

Trabajo de grado presentado como requisito parcial para optar al título de
INGENIERO EN COMPUTACIÓN

Barcelona, Julio de 2009.

UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



Desarrollo de un Software Educativo en entorno Web para
la asignatura de Taller de Programación Orientada a Objetos,
impartida en el Departamento de Computación y Sistemas
Escuela de Ingeniería y Ciencias Aplicadas de la Universidad
de oriente Núcleo Anzoátegui

Asesor:

Ing. Aquiles Torrealba
Asesor Académica

Barcelona, Julio de 2009.

UNIVERSIDAD DE ORIENTE
NÚCLEO DE ANZOÁTEGUI
ESCUELA DE INGENIERÍA Y CIENCIAS APLICADAS
DEPARTAMENTO DE COMPUTACIÓN Y SISTEMAS



Desarrollo de un Software Educativo en entorno Web para
la asignatura de Taller de Programación Orientada a Objetos,
impartida en el Departamento de Computación y Sistemas
Escuela de Ingeniería y Ciencias Aplicadas de la Universidad
de oriente Núcleo Anzoátegui



Jurado Calificador

Ing. Aquiles Torrealba

Asesor Académica

Ing. Zulirais Garcia

Jurado Principal

Ing. Pedro Dorta

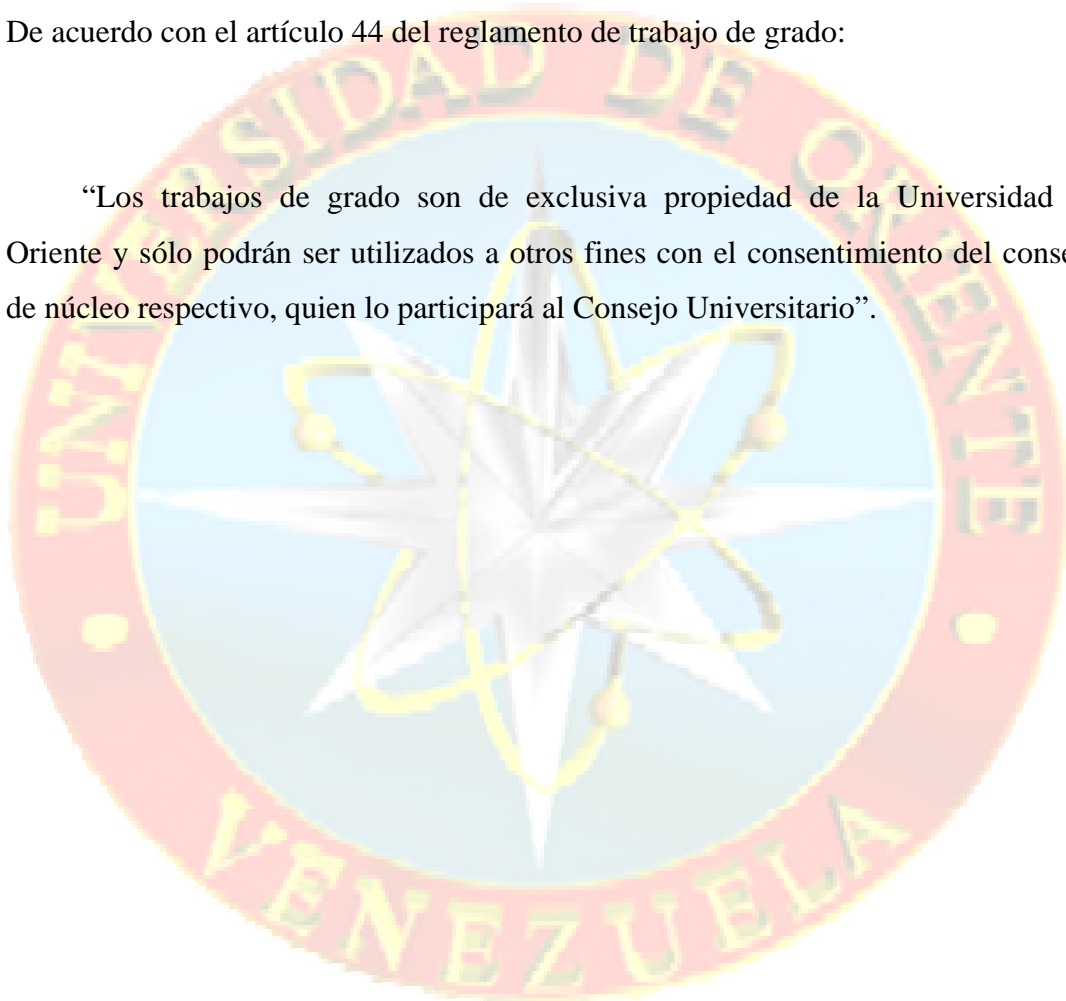
Jurado Principal

Barcelona, Julio de 2009.

RESOLUCIÓN

De acuerdo con el artículo 44 del reglamento de trabajo de grado:

“Los trabajos de grado son de exclusiva propiedad de la Universidad de Oriente y sólo podrán ser utilizados a otros fines con el consentimiento del consejo de núcleo respectivo, quien lo participará al Consejo Universitario”.



AGRADECIMIENTOS

A DIOS, por nunca abandonarme y darme la fuerza necesaria para seguir adelante siempre.

A mis mamis queridas: María Elena por todo el amor, el apoyo en cada una de mis decisiones y la ayuda incondicional eres la mejor madre. A mi tía Mercedes por su guía y sus consejos, por nunca abandonarme, por cuidarme y quererme como a una hija y por ayudarme siempre, gracias. A mi tía Milagros, por su dedicación por velar que nunca me faltara nada, cuidarme siempre y darme lo mejor de ella, gracias. Gracias a todas, las Amo.

A mi querido esposo Miguel, por estar siempre a mi lado, por enseñarme que no existe un problema que no tenga solución, por su paciencia y confianza, por su ayuda incondicional, por darme fuerzas para continuar día tras día, por quererme tanto, Gracias, Te Amo.

A mi papa, tío y compadre Lisandro, por su ayuda incondicional, por su cariño por su apoyo, por ser como un padre para mí, Gracias.

A mi compañera Noireth por su paciencia y tolerancia, por ser una amiga incondicional y por todo su esfuerzo para el desarrollo de este proyecto, y a su mama Noris por ser tan especial.

A mis hermanos Felix, Darwin, Liliana y Jhonny, por estar al pendiente y apoyarme siempre, en especial a mi hermanita Liliana por ser de gran ayuda y apoyo para la culminación de este trabajo.

A mi suegra Isaura, por siempre estar dispuesta a brindarme su ayuda en los buenos y malos momentos, gracias por tu cariño.

A mi amiga Ale con la que compartí las alegrías y las tristezas en el curso de mi carrera, y me brindo su apoyo y su ayuda cuando la necesite, a mi amigo Andrés por estar siempre dispuesto a ayudarme y compartir sus conocimientos conmigo.

DEDICATORIAS

A la profesora Zulirais Garcia por brindarme su amistad y ayudarnos en el desarrollo de este proyecto, a nuestro asesor Aquiles Torrealba por guiarnos en el transcurso de este trabajo.

A todos gracias!

Yitzy

A Dios por estar siempre allí, por darme salud, a mí y a mi familia.

A mi madre, Noris, por ser un apoyo incondicional, por estar conmigo en cada momento que podía, por ser muestra de amor sobre todas las cosas, perseverancia, lucha, sacrificio, por recordarme cada día que soy capaz de lograr todas las cosas que me proponga, por ser tan tolerante conmigo. Por supuesto por decirme todos los días que primero se comen las verdes. A Manuel porque a pesar de todos los inconvenientes siempre me ayudaste en todo lo que pudiste.

A mis hermanos Roimer, Karen y Mitzayt por estar pendiente de todo este largo camino, a Roimer por ayudar a Noris y a mí en todo momento, por la confianza para dejarme estar siempre presente en la vida de tus hijos, a Karen por estar siempre a mi lado aunque sea para pelear, a la Nena por tener siempre palabras de aliento y estar tan pendiente de Noris.

A mi primo Leonardo, por tener una confianza ciega en mí, por desprenderte de cosas preciadas para ti para ayudarme, por hacerme reír siempre con todas sus ocurrencias... Ingeniero como su papa.

A mi compañera y amiga Yitzy, por ser constancia y persistencia, por aguantarme todos los “no me parece”, por ayudarme cada vez que lo necesitaba y cuando no también. A Miguel por darme asilo en su casa, por darnos su apoyo y opinión en todo momento. A ambos por abrir las puertas de su casa y tratarme como de su familia y por tener esa preciosura de niña que se ríe cada vez que me ve. A la Sra. Maria Elena, la Sra. Mercedes, la Sra. Milagros y el Sr. Lisandro por contar con su ayuda siempre.

A mis tías Paula e Ysidra Macayo por confiar en mí y recordármelo siempre y por supuesto a mi prima Rosa por hacerme entender que no hay límites para lograr lo que se quiere.

A mi tía Paula Marcano por ser ejemplo de lucha constante y por ayudarme a solucionar cuanta duda ortográfica y gramatical tuve. A tía Irene y tío Nelson por estar siempre con mi mamá y apoyarme siempre.

DEDICATORIAS

A mis primas Desi y Sol por ser dos hermanas más. A Desi por estar en cada momento y por permitirme contar contigo a pesar de todo.

A Scalante por ser tan buena amiga, por compartir todos los momentos, malos y buenos de la carrera, por poder contar contigo. A Andrés estar dispuesto siempre a brindarme toda su ayuda y conocimiento y tener la paciencia de explicarlo.

A la profesora Zuli por ser de gran ayuda y brindarme todo su conocimiento en todo este proyecto, por regañarme cada vez que lo tenía que hacer. Al profesor Aquiles Torrealba por sus ocurrencias y por demostrar su alegría cada vez que lográbamos hacer algo.

A todos gracias por creer en mí.

Noireth

DEDICATORIA

A DIOS por ser mi guía y acompañarme en todos los momentos de mi vida.

A mi querida hija Laura, para que se sienta orgullosa de su mami, y por ser mi inspiración para la culminación de este trabajo.

A la memoria de mi abuelito Felix, que estoy segura de que donde quiera que este me cuidara y bendecirá siempre

Yitzy

DEDICATORIAS

A mi mamá para ser el motivo de su felicidad.

A lo más hermoso de la vida, mis sobrinos, Javier y Roger, para que sigan creyendo en mí y se sientan felices y orgullosos de su tía.

A mi abuela Bautista para que donde quiera que este vea este momento.

Noireth

INDICE

RESOLUCIÓN	iv
AGRADECIMIENTOS	v
DEDICATORIA	ix
INDICE	11
RESUMEN.....	14
CAPITULO1	15
Planteamiento del Problema.....	15
1.1 PLANTEAMIENTO DEL PROBLEMA	15
1.2 OBJETIVOS	19
1.2.1 Objetivo General.....	19
1.2.3 Objetivos Específicos.....	19
1.3 ANTECEDENTES DE LA INVESTIGACION.....	20
CAPITULO 2.....	22
Fundamentos Teóricos	22
2.1 INTRODUCCION	22
2.2 PROCESO DE ENSEÑANZA APRENDIZAJE.....	22
2.3 SOFTWARE	34
2.4 INGENIERIA DE SOFTWARE EDUCATIVO.....	35
2.5 MATERIALES EDUCATIVOS COMPUTARIZADOS.....	44
2.6 PROCESO UNIFICADO.....	48

2.7 LENGUAJE UNIFICADO DE CONSTRUCCION DE MODELADO	56
[17].....	56
2.8 INTERNET	59
2.9 MULTIMEDIA	60
2.10 DREAMWEAVER	60
2.11 FLASH.....	60
Capitulo 3.....	61
Fase de Inicio	61
3.1 INTRODUCCION	62
3.2 METODOLOGIA UTILIZADA.....	62
3.3 PLANIFICACION DE LA FASE DE INICIO	62
3.4 MODELO DE DOMINIO	63
3.5 IDENTIFICACION DE RIESGOS	64
3.6 FLUJO DE TRABAJO REQUISITOS	65
3.7 FLUJO DE TRABAJO ANÁLISIS	79
3.6 CONCLUSION DE LA FASE DE INICIO.....	84
CAPITULO 4	85
Fase de Elaboración	85
4.1 INTRODUCCION	85
4.2 PLANIFICACION DE LA FASE DE ELABORACION.....	85
4.3 FLUJO DE TRABAJO REQUISITOS	86
4.4 FLUJO DE TRABAJO ANALISIS	88
4.5 FLUJO DE TRABAJO DISEÑO.....	89

4.6 FLUJO DE TRABAJO IMPLEMENTACION	124
4.7 CONCLUSIÓN DE LA FASE DE ELABORACIÓN	138
CAPITULO 5.....	139
Fase de Construcción	139
5.1 INTRODUCCION	139
5.2 PLANIFICACION DE LA FASE DE CONSTRUCCION	139
5.3 ESCOGENCIA DE APLICACIONES PARA EL DESARROLLO DEL SISTEMA SETPOO.....	139
5.4 FLUJO DE TRABAJO IMPLEMENTACION	142
5.5 FLUJO DE TRABAJO PRUEBAS	144
5.4 CONCLUSIÓN DE LA FASE DE CONSTRUCCIÓN	176
5.5 PLANIFICACIÓN DE LA FASE DE TRANSICIÓN.....	176
CAPITULO 6.....	177
Manual de Usuario	177
6.1. INTRODUCCIÓN	177
6.2. INICIO DE SESIÓN	177
6.3 ELEMENTOS ESTANDAR Y DE NAVEGACIÓN.....	178
6.4. DESCRIPCIÓN DE LAS SECCIONES DE LA APLICACIÓN	181
Conclusiones y Recomendaciones	194
CONCLUSIONES	194
RECOMENDACIONES.....	196
BIBLIOGRAFIA	197
METADATOS PARA TRABAJOS DE GRADo	199

RESUMEN

El presente proyecto consiste en el desarrollo de un software educativo interactivo basado en entorno web, creado como herramienta de apoyo para la enseñanza y el aprendizaje de la unidad curricular “Taller de Programación Orientada a Objetos”, perteneciente al pensum de Ingeniería en Computación, de la Universidad de Oriente. Basándose en los principios de un curso virtual que proporcionará una nueva alternativa para suministrar el material de instrucción, que buscará el interés del estudiante hacia el contenido del software, tomando en cuenta las teorías del aprendizaje. Para la elaboración del software se empleo el Proceso Unificado de Desarrollo de Software, el cual ofrece un proceso estructurado y continuo de técnicas definidas y uso de modelos para el desarrollo de software de calidad, aunado a la metodología de Ingeniería de Software Educativo de Álvaro Galvis, que articula los aspectos educacionales, comunicacionales y computacionales involucrados en la creación de ambientes educativos multimedia apoyados con MEC’s (Materiales Educativos Computarizados). La visualización del contenido de la asignatura se realiza a través de textos y animaciones alusivas a los temas que conforman la asignatura, el sistema también cuenta con un sistema de evaluaciones que le permite al estudiante verificar el grado de asimilación alcanzado. El objetivo es crear un material que sirva de apoyo para el proceso de enseñanza-aprendizaje del profesor y el estudiante.

CAPITULO1

PLANTEAMIENTO DEL PROBLEMA

1.1 PLANTEAMIENTO DEL PROBLEMA

En la actualidad gran parte de la vida cotidiana está sujeta a la tecnología, la computación ha sido una de las principales bases fundamentales en cuanto al facilitamiento de las actividades del hombre, hoy en día la mayoría de la población dispone de un computador o una tecnología similar.

La educación es uno de los pilares más importantes en el desarrollo humano, ya que esta permite adquirir conocimientos, valores, costumbres, y formas de actuar, el hombre recibe educación desde el principio de su vida y a lo largo de ella. La educación le permite al ser humano desenvolverse en la sociedad.

En un entorno de constantes cambios y donde la revolución computacional se hace presente, en el caso particular de la Educación, esta revolución a presentado avances significativos que se pueden calificar como beneficiosos ya que tienden a la modernización de los métodos de enseñanza, es así como nace un modelo educativo basado en la computación y el mundo informático, un ejemplo de estos son: Los Software Educativos, Tutoriales Inteligentes, Adiestramiento Basado en Computador, hechos posibles gracias a herramientas de programación, dispositivos de Hardware, Elementos Multimedia, etc., los cuales unificados de manera estructurada contribuyen con la creación y distribución de programas orientados a la enseñanza de algún tema en específico.

La enseñanza asistida por computador se ha convertido en una rama de investigación importante, ya que es una disciplina que forma parte de la ingeniería de software y tiene como objetivos la incorporación de aspectos didácticos y pedagógicos dentro de las fases de análisis y diseño de software, que faciliten y garanticen la satisfacción de necesidades educativas. Es por ello que se han creado medios de aprendizaje más dinámicos, inteligentes y sofisticados que ayuden al

MARCO TEORICO

estudiante a comprender lo que desea aprender, de una manera menos abstracta y hasta divertida, los cuales ofrecen una interfaz o un micromundo, fundamentados en la aplicación de imágenes, sonido, video, texto, esquemas animados, etc., integrados en un mismo ambiente, donde el estudiante interactúa y resuelve los problemas que se le presentan para de esta manera lograr obtener algún conocimiento.

Los Software Educativos contribuyen con el mejoramiento de la calidad de la enseñanza y el aprendizaje, es por ello que son de gran utilidad al impartir conocimientos en distintos niveles educativos. Por ser la universidad el pilar fundamental en el desarrollo de profesionales a diversas áreas, es importante contar con medios que sirvan de herramienta de apoyo, tanto para estudiantes como para profesores. La Universidad de Oriente, Núcleo Anzoátegui, es una de las principales casas de estudio de Venezuela, ella ofrece una gama de carreras en las cuales destaca, tanto la Ingeniería en Computación como la Ingeniería en Sistema, especialidad donde convergen las nuevas tecnologías nacientes constantemente y cuyo objetivo fundamental persigue la formación de profesionales capacitados en las áreas de innovación tecnológica, en la misma son dictadas asignaturas que poseen un extenso y muchas veces complejo contenido programático.

Cabe destacar la unidad curricular Taller de Programación Orientada a Objetos, asignatura dictada en el Departamento De Computación y Sistema de la Escuela de Ingeniería y Ciencias Aplicadas, es la base fundamental ya provee de los conocimientos básicos acerca de la programación, así como también el estudio de las herramientas y técnicas para la resolución de problemas, la misma cuenta con un contenido programático muy completo y esto hace que sea una materia extensa y compleja, ahondado a esto, cabe destacar que cada vez es mayor el número de estudiantes que la cursan, los cuales cada uno posee un ritmo de aprendizaje diferente, donde algunos podrán asimilar o no, el contenido impartido, ocasionando que el profesor e incluso el preparador asignado no se den abasto a la hora de asistir al estudiante en clase.

MARCO TEORICO

En vista de esta situación los profesores universitarios que actualmente dictan la materia, Ing. Aquiles Torrealba e Ing. Pedro Dorta, han sugerido el desarrollo de un Software Educativo, para la asignatura Taller de programación Orientado a Objetos, impartida en el Departamento de Computación y Sistema, de la universidad de Oriente, Núcleo Anzoátegui, como herramienta de apoyo para el aprendizaje dirigido en clase y con un condensado contenido pedagógico, que corresponda a las exigencias del contenido programático vigente, con el fin de ofrecerle un material de apoyo adicional al estudiante y lograr así el nivel de asimilación y comprensión que la asignatura necesita.

Por ser el aprendizaje una etapa importante en el desarrollo de toda actividad educativa es necesario exigir para la enseñanza las más calificadas ayudas tecnológicas, es por ello que el Software resultante implementará estrategias instruccionales activas exigiéndole al aprendiz mayor participación, con el objeto de enriquecer su aprendizaje y la asimilación del mismo con eficiencia para su aplicación a nuevas situaciones o problemas, esto le permitirá la opción de interactuar con un ambiente multimedia de calidad que contribuirá a potencializar el estilo cognoscitivo presente en cada educando y a desarrollar el pensamiento lógico. De igual manera, le brindará al estudiante la posibilidad de individualizar su proceso de enseñanza-aprendizaje, sin presiones por parte del docente, ni de sus compañeros ante problemas de comprensión.

Cabe destacar que el Software Educativo que se implementará podrá ser usado a cualquier hora y en cualquier sitio, esto gracias a la amplitud de acciones que brindan las tecnologías actuales de computación las cuales harán posible que el uso de esta herramienta pueda usarse en ambientes de redes locales o distribuirse en la internet, lo único que necesitará el estudiante para poder disponer de la aplicación será un computador

El entorno multimedia e interactivo del sistema logrará captar la atención del estudiante, manteniéndolo en una constante expectativa y un permanente interés en todo el contenido del software a través del cual podrá navegar libremente, así como

MARCO TEORICO

también contar con auto-evaluaciones, amenas e interactivas donde podrá verificar de forma inmediata el grado de asimilación del contenido presentado a través de mensajes informativos y un sistema de puntuación que desplegará el programa. Así la aplicación reforzará el actual método de enseñanza con el fin de que el estudiante se sienta constructor de su propia experiencia de aprendizaje a través de la reorganización de sus esquemas de conocimiento.

En lo que respecta al educador, la aplicación estará adecuada a los nuevos paradigma de la educación y proporcionará una poderosa herramienta auxiliar de enseñanza con los requerimientos técnicos apropiados que lo ayudará considerablemente en su esfuerzo por impartir el contenido programático de la cátedra, contribuyendo así a resolver el problema existente entre el crecimiento acelerado de la demanda de estudiantes, del volumen de información y el tiempo disponible para dar cumplimiento al mismo. De la misma manera le brindara al profesor la información que requiere acerca del progreso de cada uno de sus estudiantes, ya que este tendrá acceso al historial de cada uno de ellos.

Para lograr el desarrollo de este proyecto de investigación se empleará la metodología del proceso unificado de software (RUP), el cual ofrece una guía completa para el desarrollo de software. En cuanto al modelado de la enseñanza-aprendizaje tomaremos como pautas a seguir el modelo propuesto por Álvaro Galvis para el desarrollo de software educativo. Tomando en cuenta que el programa se realizara bajo un entorno web, se hará uso de webml el cual proveerá especificaciones gráficas formales para un proceso de diseño completo que puede ser asistido por herramientas de diseño visuales.

En cuanto a la implementación de la aplicación se utilizaran como recursos herramientas de apoyo orientadas al diseño de este tipo de aplicaciones tales como: Javascript, Flash, Ajax, las cuales proporcionan un medio disponible para el desarrollo de animaciones, páginas web, software de distintas utilidades, etc.

MARCO TEORICO

El software educativo a desarrollar para el curso de Taller de Programación Orientada a Objetos, se presenta como un novedoso recurso instruccional que le permitirá al estudiante una enseñanza individual, interactiva y flexible que incrementarán aún más la asimilación del contenido programático. Así mismo, el seguimiento paso a paso de cada una de las unidades sobre las cual estará basada esta aplicación. Esta nueva modalidad de aprendizaje también contará con una serie de evaluaciones, con el objetivo de ponderar el nivel de comprensión y retención del material de estudio, a fin de ofrecer al estudiante una manera de medir el grado de preparación del que dispone para una evaluación realizada por el profesor. El uso de esta aplicación aportará al estudiante, un material didáctico e interactivo que favorecerá la asociación de ideas y la creatividad, permitirá las practicas de nuevas técnicas y la reducción del tiempo y del esfuerzo necesario para aprender, retener y comprender el contenido programático de la cátedra.

1.2 OBJETIVOS

1.2.1 Objetivo General.

Desarrollar un Software Educativo en entorno web para la asignatura de Taller de Programación Orientada a Objetos, impartida en el Departamento de Computación y Sistemas Escuela de Ingeniería y Ciencias Aplicadas de la Universidad de oriente Núcleo Anzoátegui.

1.2.3 Objetivos Específicos.

- ✓ Analizar los principios pedagógicos requeridos por el contenido programático de la asignatura con el fin de determinar el contexto en el cual se va a desarrollar la aplicación.
- ✓ Determinar los objetivos y requerimientos del sistema.
- ✓ Realizar el diseño de interfaz de la aplicación que servirá como medio de comunicación entre el programa y el usuario.

- ✓ Diseñar la estructura del software, considerando los principios o pautas establecidas en el desarrollo educativo del modelo de ingeniería de software educativo.
- ✓ Realizar la codificación e implementación del sistema.
- ✓ Efectuar pruebas, tanto de unidad como de integración, realizando las depuraciones respectivas.

1.3 ANTECEDENTES DE LA INVESTIGACION

En la Universidad de Oriente, Núcleo de Anzoátegui, los estudiantes de pregrado del departamento de Computación y Sistemas; han desarrollado diversos software para diferentes niveles de educación. Entre los cuales se encuentran:

Título: Desarrollar un software educativo para facilitar la enseñanza y aprendizaje de la asignatura Análisis y Diseño de Algoritmo específicamente para las unidades: I, II, III y V; dictada en el Departamento de Computación y Sistemas de la Escuela de Ingeniería y Ciencias Aplicadas de la Universidad de Oriente Núcleo de Anzoátegui; utilizando técnicas de diseño e implementación de la ingeniería de software educativo con el enfoque orientado a objetos.

Resumen: Este proyecto está basado en los principios de un curso virtual y se creó con el propósito de proporcionar una nueva alternativa para suministrar el material de instrucción, para desarrollarlo se tomaron en cuenta las teorías del aprendizaje y se tomo como pauta a seguir el proceso unificado de desarrollo de software así como también la metodología de software educativo propuesta por Álvaro Galvis

Autor: Veracierta T., Gabriela M. y Torres R., Pamela A.

Institución: Universidad de Oriente, Ingeniería en Computación

Nivel: Pregrado

Fecha: 2005

Título: Desarrollar un software educativo como herramienta fundamental de apoyo en el proceso enseñanza-aprendizaje de la unidad curricular Introducción a la Ingeniería en Computación de la especialidad de Ingeniería en Computación, utilizando ambiente interactivo multimedia.

Resumen: Este proyecto se ofreció con el fin de proporcionar al estudiante un recurso instruccional capaz de mejorar el nivel de retención y comprensión del contenido programático, basado en teorías de aprendizajes y estrategias instruccionales activas, se desarrollo aplicando ingeniería de software, así como la ingeniería de software propuesta por Alvaro Galvis y apoyándose en los Materiales Educativos Computacionales (MEC).

Autor: Finotello G. Lucía C. y Regardiz C. Sulmira del C.

Institución: Universidad de Oriente, Ingeniería en Computación

Nivel: Pregrado

Fecha: 2004

Título: Desarrollo de un Software Educativo para apoyar la enseñanza de la Asignatura Programación I Impartida en el Departamento de Computación y Sistemas de la Universidad de Oriente

Resumen: Este proyecto se manejó la metodología y técnicas propuestas por Álvaro Galvis para la elaboración de este tipo de software.

Autor: Velásquez R., Gloribert M. y Bonini R., Juan Carlos A

Institución: Universidad de Oriente, Ingeniería en Computación

Nivel: Pregrado

Fecha: 2003

CAPITULO 2

FUNDAMENTOS TEÓRICOS

2.1 INTRODUCCION

La enseñanza es una ayuda para que las personas aprendan. Cuando se emprende la tarea de enseñar, el educador no puede responsabilizarse de todo el aprendizaje que vaya a tener lugar. El avance de la tecnología educativa y la nueva visión acerca de los procesos de enseñanza-aprendizaje, llevan a la educación a una modalidad de ambientes de aprendizaje apoyado por los medios, diseñados para crear condiciones pedagógicas y contextuales favorables, donde el conocimiento y sus relaciones con los individuos son el factor principal para formar una “sociedad del conocimiento”.

En este capítulo se presenta los fundamentos que a nivel pedagógico y metodológico sustenta el desarrollo del software educativo que se plantea.

2.2 PROCESO DE ENSEÑANZA APRENDIZAJE

La enseñanza es el proceso mediante el cual se comunican o transmiten conocimientos especiales o generales sobre un tema específico, a través de medios diversos. El aprendizaje es la adquisición de nuevos conocimientos a partir de determinada información percibida.

El objetivo es aprendido de distintas formas por cada sujeto, porque las experiencias y las capacidades de cada individuo presentan características únicas. El aprendizaje abarca la adquisición de destrezas, hábitos y habilidades, así como actitudes y valoraciones.

El aprendizaje que puede enriquecer a la persona es el que establece una relación entre el nuevo material susceptible de ser aprendido y los conocimientos

previos del sujeto. Cuando se cumple esta condición, el sujeto le encuentra sentido a lo que estudia, lo entiende y puede lograr entonces un aprendizaje significativo.

2.2.1 Factores del Aprendizaje [6]

✓ **Motivación:** interés que tiene el alumno por su propio aprendizaje o por las actividades que le conducen a él. El interés se puede adquirir, mantener o aumentar en función de elementos intrínsecos y extrínsecos. Hay que distinguirlo de lo que tradicionalmente se ha venido llamando en las aulas motivación, que no es más que lo que el profesor hace para que los alumnos se motiven.

✓ **La maduración psicológica:** es importante saber cómo ayudar a aprender dependiendo de la edad del alumno aprenda de una forma más fácil, y saber de qué temas tratar o hablar con él.

✓ **La dificultad material:** otro factor que puede influir en el aprendizaje es lo material y esto es muy importante porque muchas veces depende de dinero la educación de nuestros hijos, tenemos que ver la forma de ayudarlos económicamente de la mejor manera para que tengan todos los materiales, de otra forma se atrasarían y no aprendieran correctamente.

✓ **La actitud dinámica y activa:** esta parte es de las que tenemos que tomar mucho en cuenta, puesto que es más fácil aprender en una clase dinámica, ósea con juegos y preguntas que ayuden a entender mejor el tema, pero claro, que el alumno este en una actitud de aprender.

✓ **Tu estado de fatiga o descanso:** es muy importante que el alumno esté en condiciones de aprender, que quiere decir esto, que este descansado, haya dormido bien, para poner la atención debida en la clase.

✓ **Capacidad intelectual:** esta capacidad es diferente en cada una de las personas, buena, regular, mala y excelente. Debemos explicar muy bien el tema para un mejor aprendizaje.

✓ **Distribución del tiempo para aprender:** toma en cuenta que la distribución de tu tiempo es muy importante para que tu mente siempre este activa para aprender.

2.2.2 Principios del Aprendizaje

✓ **Contigüidad.** Este principio afirma que la situación estímulo en la que se quiere que responda el educando debe presentarse "en proximidad temporal con la respuesta deseada".

✓ **Repetición.** Este principio afirma que la situación estímulo y su respuesta necesitan repetirse o practicarse para que el aprendizaje mejore y se retenga. Pero, existen situaciones en que la repetición no mejora el aprendizaje ni la retentiva. Por esto, es mejor considerar que la repetición no es condición fundamental del aprendizaje, sino tan sólo una práctica que garantiza la presencia de otras condiciones favorables.

✓ **Reforzamiento.** Este principio sostiene que el aprendizaje de un acto nuevo se fortalece cuando éste sigue un estado de cosas que es satisfactorio. El individuo aprende más fácilmente un acto nuevo, cuando va seguido por un acto viejo que ya ejecuta con facilidad; ejemplo: si al niño le gusta ver dibujos de animales y se quiere que aprenda a dibujarlos según este principio aprenderá la nueva habilidad de dibujar animales asociándola al hecho de ver dibujos de animales.

2.2.3 Teorías del Aprendizaje [7]

Todas las aproximaciones psicológicas al fenómeno del aprendizaje humano tienen algo que decir como fundamento para el diseño de ambientes de enseñanza /

aprendizaje. Sin embargo, los aportes no necesariamente son convergentes, como no lo es la perspectiva desde la cual se analiza el fenómeno en cada caso, ni los métodos usados para obtener el conocimiento. Si hubiera una teoría que atendiera todos los aspectos del fenómeno, que abarcara las demás teorías, no habría que estudiar las otras; pero la realidad es diferente, de ahí la necesidad de por lo menos conocer los puntos más importantes de los diferentes aportes en relación al tema. Las aproximaciones al fenómeno del aprendizaje oscilan entre dos polos: conductismo y cognoscitivismo.

Las teorías sobre el aprendizaje, son muy extensas dado la complejidad del tema, por ello se presenta en este marco teórico un resumen de los principios básicos sobre el aprendizaje y se detalla la teoría conductista-cognoscitiva desarrollada por Robert M. Gagné, utilizada en el diseño educativo de esta aplicación.

2.2.3.1 El Conductismo [8]

El conductismo iguala al aprendizaje con los cambios en la conducta observable, bien sea respecto a la forma o a la frecuencia de esas conductas. El aprendizaje se logra cuando se demuestra o se exhibe una respuesta apropiada a continuación de la presentación de un estímulo ambiental específico.

El conductismo focaliza en la importancia de las consecuencias de estas conductas y mantiene que las respuestas a las que se les sigue con un refuerzo tienen mayor probabilidad de volver a sucederse en el futuro. No se hace ningún intento de determinar la estructura del conocimiento de un estudiante, ni tampoco de determinar cuáles son los procesos mentales que ese estudiante necesita usar. Se caracteriza al estudiante como reactivo a las condiciones del ambiente y no como sucede en otras teorías, donde se considera que asume una posición activa en el descubrimiento del mismo.

2.2.3.2 El Cognitivismo [9]

Los modelos cognoscitivos señalan que el aprendizaje es la “adquisición de conocimientos” del organismo sobre el medio ambiente, en donde el organismo es un procesador de la información. El objetivo de esta propuesta es conocer los procesos cognoscitivos internos que son usados para aprender (atención, memoria y otros procesos básicos), así como los contenidos de lo aprendido (la representación del conocimiento), como determinantes esenciales de la conducta.

Esta perspectiva busca conocer y conceptualizar los procesos que intervienen entre el estímulo que recibe un sujeto y la respuesta que este da como consecuencia. Se intenta integrar un modelo del organismo en el que se observe cómo se producen las respuestas que da el sujeto para explicar cómo se generan sus estructuras intelectuales. Asimismo busca proponer modelos para analizar las relaciones entre las diferencias individuales, el aprendizaje y la ejecución, con la finalidad de conseguir la competencia de un estudiante después de una ejecución precisa.

2.2.3.3 Conductismo Cognoscitivo: La Teoría de Robert M. Gagné.

El cognitivismo (Merrill, Gagné...), basado en las teorías del procesamiento de la información y recogiendo también algunas ideas conductistas (refuerzo, análisis de tareas) y del aprendizaje significativo, aparece en la década de los sesenta y pretende dar una explicación más detallada de los procesos de aprendizaje.

Para Gagné, el aprendizaje es un proceso de cambio en las capacidades del individuo el cual produce estados persistentes y es diferente de la maduración o desarrollo orgánico. Se infiere que ha ocurrido cuando hay un cambio de conducta que perdura. El aprendizaje se produce usualmente mediante interacción del individuo con su entorno (físico, social, psicológico, etc.).

MARCO TEORICO

El estudiante representará en su mente simbólicamente el conocimiento, que se considera (igual que los conductistas) como una realidad que existe externamente al estudiante y que éste debe adquirir. El aprendizaje consiste en la adquisición y representación exacta del conocimiento externo. La enseñanza debe facilitar la transmisión y recepción por el alumno de este conocimiento estructurado.

Gagné explica que la persona que aprende pasa por diversas etapas en la que la información es transformada. El proceso se inicia con la fase de motivación (externa o interna), en la que se crea una expectativa que mueve el aprendizaje. En la fase de comprensión se llama la atención del aprendiz sobre lo que es importante, se favorece que perciba selectivamente aquello que interesa que aprenda. Cuando el aprendiz pasa de no-aprendido a aprendido, de no-ser-capaz a serlo (fase de adquisición y retención). Este incidente se da cuando la información ya transformada pasa del registro sensorial a la memoria de corto plazo y se acrecienta la estructura de la información; sin embargo, para que este aprendizaje pueda ser permanente se necesita integrar la nueva información con la estructura existente, cuando se acumula en la memoria de largo plazo lo aprendido.

El afinamiento se produce en las fases siguientes al incidente esencial, mediante la fase de recordación, que exige recuperar lo aprendido, la de generalización; que demanda transferir lo aprendido a una variedad de contexto, y la de retroinformación, la cual permite afirmar lo aprendido mediante el refuerzo o la reorientación que resulta de confrontar la expectativa con lo logrado. De esta manera, las fases del aprendizaje propuesta por Gagné amplifican el conocimiento derivado de las teorías de procesamiento de la información.

Cuando se quiere diseñar un ambiente y actividades apropiadas para obtener un resultado esperado, o evaluar lo adecuado de un ambiente y actividades de aprendizaje frente a los postulados de esta teoría, el siguiente procedimiento suele dar los resultados esperados:

- ✓ Definir claramente el objetivo o resultado que se desea lograr al finalizar el desarrollo de las actividades en el ambiente de aprendizaje; la redacción debe hacerse preferiblemente en términos observables u operacionales.

- ✓ Descomponer el objetivo final o Terminal en los objetivos subyacentes, hasta llegar al punto de partida, aquellos objetivos que se espera domine ya el aprendiz.
- ✓ Clasificar el objetivo Terminal de aprendizaje y cada uno de los subobjetivos en las categorías que corresponden.
- ✓ Diseñar un ambiente y actividades de aprendizaje que atiendan los principios que son aplicables a cada uno de los tipos de objetivos determinados. Si se trata de valorar un ambiente y actividades dados, se verifica la congruencia de éstos con los principios aplicables a cada tipo de objetivo.

Una forma de favorecer este autoperfeccionamiento de los procesos de aprendizaje es hacer conscientes a los aprendices, de los principios que rigen el aprendizaje de cada uno de los tipos de objetivos. La verbalización de estos principios ligada a situaciones vivenciales puede favorecer que se apliquen a conciencia, como una estrategia para aprender más eficientemente aquello que interesa. Trasladar el control de los aprendices, sobre el procesamiento de la información que compone a cada tipo de aprendizaje, puede también ayudar a que haya una más eficaz colaboración entre docentes y alumnos [9].

2.2.4 Fases del Aprendizaje [11]

Este modelo posibilita el entendimiento de los mecanismos internos del aprendizaje, que Gagné divide en fases o etapas.

MARCO TEORICO

✓ La primera fase es la motivación que se encuentra estrechamente ligada a los conceptos de expectativa y de refuerzo, es decir que debe existir algún elemento de motivación o expectativa para que el estudiante pueda aprender.

✓ La segunda fase es de atención y percepción selectiva, mediante la cual se modifica el flujo de información que ha llegado al registro sensorial y que pasa a la memoria. La atención en referencia a la dirección de los mecanismos de atención hacia el elemento o unidad que debe ser aprendida. La selección perceptiva, que es percibir los elementos destacados de la situación.

✓ La tercera fase es la adquisición, que comienza con la codificación de la información que ha entrado en la memoria de corto alcance y que para su ingreso en la memoria de largo alcance se transforma de la información que ha sido recibida en material simplificado, o como material verbal, o como imágenes mentales, etc.

✓ La cuarta fase es la retención o acumulación en la memoria.

✓ La quinta fase es la de recuperación de la información. En acción a estímulos externos una información que ya ha sido almacenada en la memoria de largo alcance puede ser recuperada. Esto se hace mediante un rastreo de la memoria hasta encontrar la información, utilizando la misma codificación que se empleó para almacenarla.

✓ La sexta fase es la de la generalización. La recuperación de una información almacenada puede hacerse en circunstancias diferentes a las que produjeron su almacenamiento, o en situaciones que no están en el mismo contexto del aprendizaje original, por eso es que el individuo debe ser capaz de generalizar lo aprendido en nuevas situaciones.

2.2.5 Ambientes de Aprendizaje

Los ambientes de aprendizaje son aquellos espacios en donde se crean las condiciones para que el individuo se apropie de nuevos conocimientos, de nuevas experiencias, de nuevos elementos que le generen procesos de análisis, reflexión y apropiación [12].

Los ambientes de aprendizaje son planeados para responder a diversas necesidades: el individuo que aprende en su propio espacio, el grupo que aprovecha las herramientas tecnológicas y los conocimientos en la dinámica de una interacción directa en sus integrantes, el grupo que se relaciona en un espacio virtual, etc. Aparte de este mundo de relaciones, los ambientes de aprendizaje pueden enfatizar o privilegiar uno o varios de sus componentes: los asesores, los estudiantes, los medios tecnológicos. Para ser más preciso estos ambientes dependen en gran medida de los medios para la estructuración de la propuesta pedagógica y toca a los docentes y estudiantes su consolidación y aplicación.

Existen dos formas sistemáticas para la creación y uso de ambientes de aprendizaje que Tomas Dwyer llama: enfoque algorítmico y heurístico.

2.2.5.1 Enfoque Algorítmico

Definiendo algoritmo, encontramos que es un conjunto ordenado y finito de operaciones que permite hallar la solución a un problema. En este enfoque predomina el aprendizaje por medio de la transmisión del conocimiento, desde la persona que sabe hacia la persona que lo desea aprender. Por lo que el diseñador, trata de captar secuencias bien diseñadas de aquellas actividades que orientan al aprendiz desde donde está hacia donde desea llegar. Con este tipo de material, el alumno trata de asimilar al máximo lo que se le transmite.

Este enfoque según Dwyer se orienta hacia la definición y realización de secuencias predeterminadas de actividades que, cuando se acierta en los supuestos

sobre el nivel de entrada y las expectativas de los destinatarios y cuando se llevan a cabo las actividades en la forma esperada, conducen a lograr metas mensurables también predeterminadas. El alumno, bajo este enfoque, tiene como misión asimilar al máximo las enseñanzas de su maestro, convirtiéndose en depositario de sus conocimientos y modelo de pensamiento. [13]

Puede decirse que bajo este enfoque se da una enseñanza controlada por el diseñador. El decide para qué y qué enseñar, diagnostica o lanza hipótesis a partir de las cuales, establece el cómo y el hasta dónde y con qué nivel. El aprendiz debe tratar de aprehender al máximo lo que enseña el profesor, siendo éste y los materiales de que se vale, las fuentes del conocimiento. Este ya suele estar elaborado, no hay que descubrirlo, se trata de asimilarlo.

2.2.5.2 Enfoque Heurístico

En este enfoque predomina el aprendizaje de acuerdo a la experiencia obtenida por el educando y sus descubrimientos, por lo que deberá crear sus propios modelos de pensamiento, y sus propios supuestos del mundo para luego probarlos con el material heurístico.[13]

El docente se ayuda de medios, para crear ambientes aptos para que el alumno deba explorar utilizando una serie de conjeturas, los cuales le permitan al estudiante interactuar y manipular con el objeto de conocimiento. Sin embargo, para promover el aprendizaje por descubrimiento no basta con que haya dispositivos heurístico (micromundos, ambientes interactivos para explorar) que hagan posible la vivencia en que se basan las experiencias físicas o mentales que realiza el aprendiz. Es necesario que el profesor favorezca el desarrollo de las capacidades de autogestión en el aprendiz.

2.2.6 La Enseñanza

La esencia de la enseñanza está en la transmisión de información mediante la comunicación directa o apoyada en la utilización de medios auxiliares, de mayor o menor grado de complejidad y costo. Tiene como objetivo lograr que en los individuos quede, como huella de tales acciones combinadas, un reflejo de la realidad objetiva de su mundo circundante que, en forma de conocimiento del mismo, habilidades y capacidades, lo faculten y, por lo tanto, le permitan enfrentar situaciones nuevas de manera adaptativa, de apropiación y creadora de la situación particular aparecida en su entorno. El proceso de enseñanza consiste, fundamentalmente, en un conjunto de transformaciones sistemáticas de los fenómenos en general, sometidos éstos a una serie de cambios graduales cuyas etapas se producen y suceden en orden ascendente, de aquí que se la deba considerar como un proceso progresivo y en constante movimiento, con un desarrollo dinámico en su transformación continua [1].

2.2.7 Tecnología en la Educación

Tomando en cuenta la influencia y el impacto del desarrollo tecnológico sobre la sociedad, hoy en día diversas instancias educativas se están valiendo del conjunto de nuevas tecnologías en comunicación e informática y están descubriendo sus potencialidades para hacer más eficientes las estrategias pedagógicas.

Evidentemente esta integración de las nuevas tecnologías como elementos de diversificación y mejoramiento de los entornos de aprendizaje, ha exigido a su vez un replanteamiento del proceso de enseñanza - aprendizaje y de las relaciones entre los profesores, los alumnos y el contexto.

De esta forma, el proceso de incorporación de las nuevas tecnologías al ámbito educativo solo puede ser validado dentro de la sistematización que implica la

tecnología educativa, de aquí que se resalte la importancia que juega el papel del maestro dentro de este proceso.

Un computador es uno de las principales instrumentos de apoyo para el desarrollo de procesos y destrezas cognitivas, por lo tanto se pueden utilizar de diversas formas en la educación, bien sea como tutor o como herramienta, es decir, como medio de enseñanza aprendizaje o como herramienta de trabajo.

Un aspecto muy importante que posee el computador es la interactividad hombre - máquina, la cual es una de las principales características para ofrecer algo innovador y diferente a otros medios de aprendizaje.

El computador es capaz de manipular estímulos textuales, gráficos, color, animaciones, puede interactuar con el usuario, de igual manera es capaz de procesar la información suministrada en función de la que ya posee. Esto hace posible que el usuario tenga control sobre el proceso. Además ofrece la posibilidad de individualizar en todas las dimensiones ya sean de ritmo, secuencia, metas, punto de partida y tratamiento.

El computador utilizado como mediación pedagógica para el proceso de enseñanza – aprendizaje, se desarrolla a través de las acciones del profesor, los materiales y el estudiante mismo dentro de su proceso de auto e ínter aprendizaje. La mediación pedagógica es entendida aquí como las formas de intervención educativa que posibilitan transformaciones intelectuales, materiales, personales y sociales ó como plantea Gutiérrez F y Prieto C, “es el tratamiento de contenidos y formas de expresión de los diferentes temas a fin de hacer posible el acto educativo, dentro del horizonte de una educación concebida como participación, creatividad, expresividad y relacionalidad” [12].

Este nuevo enfoque para la educación, representa una nueva alternativa de aprendizaje para el estudiante y para el profesor, ya que deja atrás las clases tradicionales, adaptándose a las nuevas corrientes tecnológicas.

2.3 SOFTWARE

Un software es todo programa o aplicación programada para realizar tareas específicas, Está formado por una serie de instrucciones y datos, que permiten aprovechar todos los recursos que el computador tiene.

El software es un producto que tiene características muy especiales, hay que tener en cuenta que es un producto que se desarrolla y se centra en su diseño, con una existencia lógica, de instrucciones sobre un soporte. Es un producto que no se gasta con el uso como otros y repararlo no significa restaurarlo al estado original, sino corregir algún defecto de origen lo que significa que el producto entregado posee defectos, que podrán ser solucionados en la etapa de mantenimiento. [13]

2.3.1 Ingeniería de Software

Es la aplicación de un modelo sistemático para la creación de un programa computarizado basado en un marco de referencia, que contiene los procesos, las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando la vida del sistema desde la definición de requisitos hasta la finalización de su uso.

2.3.1.1 Objetivos de la Ingeniería de Software.

- ✓ Suministrar a los desarrolladores las bases para construir software de alta calidad en una forma eficiente.
- ✓ Mejorar la calidad de los productos de software.
- ✓ Facilitar el control del proceso de desarrollo de software.
- ✓ Definir una disciplina que garantice la producción y mantenimiento de los productos o software desarrollado en el plazo fijado dentro del costo estimado

2.4 INGENIERIA DE SOFTWARE EDUCATIVO

Es una disciplina que forma parte de la Ingeniería de Software, que tiene como objetivo la incorporación de aspectos didácticos y pedagógicos dentro de las fases de análisis y diseño de software, que faciliten y garanticen la satisfacción de necesidades educativas.

2.4.1 Características de un software educativo

Los software educativo son elaborados para tratar las diferentes asignaturas de diversas formas (a partir de cuestionarios, facilitando una información estructurada a los alumnos, mediante la simulación de fenómenos) y ofrecer un entorno de trabajo flexible más o menos adaptado a las circunstancias de los alumnos y más o menos rico en posibilidades de interacción. Las características esenciales de un software educativo son las siguientes:

- ✓ Facilidad de uso.
- ✓ Versatilidad (adaptación a diversos contextos)
- ✓ Calidad del entorno audiovisual.
- ✓ Calidad en los contenidos (base de datos).
- ✓ Navegación e interacción.
- ✓ Originalidad y uso de tecnología avanzada.
- ✓ Capacidad de motivación.
- ✓ Adecuación a los usuarios y a su ritmo de trabajo.
- ✓ Potencialidad de los recursos didácticos.

- ✓ Fomento de la iniciativa y el autoaprendizaje.
- ✓ Enfoque pedagógico actual.
- ✓ Documentación.
- ✓ Esfuerzo cognitivo.

2.4.2 Funciones de un Software Educativo [14]

- ✓ **Función informativa.** Presentan contenidos que proporcionan informaciones objetivas estructuradas, representan la realidad y la ordenan.
- ✓ **Función instructiva.** Orientan y regulan el aprendizaje de los estudiantes, promueven el logro de objetivos educativos específicos.
- ✓ **Función motivadora.** Incluyen elementos para captar la atención de los alumnos mantener su interés.
- ✓ **Función evaluadora.** permite responder inmediatamente a las respuestas y acciones de los estudiantes, y evaluar el trabajo que se va realizando con ellos.
- ✓ **Función investigadora.** Ofrecen entornos donde investigar, buscar determinadas informaciones, cambiar los valores de las variables de un sistema, etc.
- ✓ **Función expresiva.** Permiten a los estudiantes expresarse y comunicarse con el computador y con otros compañeros a través de las actividades de los programas.
- ✓ **Función lúdica.** Trabajar con los computadores realizando actividades educativas es una labor que a menudo tiene unas connotaciones lúdicas y festivas para los estudiantes.

✓ **Función innovadora.** Los programas educativos por su versatilidad abren amplias posibilidades de experimentación didáctica e innovación educativa en el aula.

2.4.3 Una Metodología de ISE

La enseñanza asistida por computador se ha convertido en una rama de investigación importante, ya que es una disciplina que forma parte de la ingeniería de software y tiene como objetivos la incorporación de aspectos didácticos y pedagógicos dentro de las fases de análisis y diseño de software, que faciliten y garanticen la satisfacción de necesidades educativas.

En la metodología de ISE desarrollada por Álvaro Galvis, los micromundos interactivos juegan un papel clave. Es a través de ellos como se crean ambientes lúdicos para aprender y es en ellos donde se viven experiencias que sirven de base para que el aprendiz genere o apropie conocimiento, dependiendo de la manera (algorítmica o heurística) como se use el micromundo.

La metodología ISE ofrece mecanismos de análisis, diseño educativo y comunicacional, prueba piloto y de campo bastante sólidos, toda vez que se fundamentan en principios educativos, comunicacionales y de tecnología educativa de validez comprobada. Sin embargo, desde la perspectiva computacional no ha evolucionado, con lo que cabe enriquecerla tomando en cuenta los avances tecnológico en el diseño y desarrollo computacional que se han logrado en los últimos años. Estos avances permiten incluir dentro de los productos de software nuevos recursos que enriquecen el potencial de acción de los mismos y que cabe usar desde el momento de formular su diseño. [14]

En la figura 2.1 se ilustra el flujo de acción de la metodología de ISE. El ciclo de vida de una aplicación educativa puede tener dos maneras de ejecución, en función de los resultados de la etapa de análisis: en el sentido de las manecillas del reloj se

procede a diseñar, desarrollar y probar lo que se requiere para atender una necesidad. En el sentido contrario, se somete a prueba aquello que puede satisfacer la necesidad.

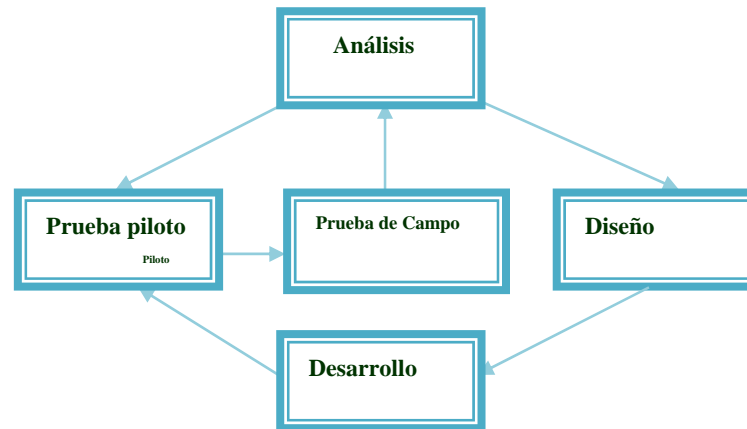


Figura 2.1 Metodología ISE propuesta por Galvis.

2.4.4 La Ingeniería de Software y el Paradigma Orientado a Objetos

El enfoque de orientación por objetos (O.O) es un paradigma que cubre el ciclo de vida de un software y que permite tener un mayor acercamiento al mundo que se modela y saber cómo funciona este mundo. Puede ser aplicado indistintamente al análisis, diseño o desarrollo de una aplicación. [14]

Las tecnologías de objetos reflejan una visión natural del mundo. Los objetos están caracterizados en clases y las clases en jerarquías; cada clase contiene un conjunto de atributos que la describen y un conjunto de operaciones que definen su comportamiento. Los objetos modelan casi todo aspecto identificable del dominio del problema; en donde las entidades externas, cosas, ocurrencias, roles, unidades organizacionales, lugares y estructuras pueden ser representadas mediante estos. Es importante destacar que, los objetos (y las clases de las que se derivan) encapsulan datos y procesos. Las operaciones de procesos son parte del objeto y se inician al pasarle un mensaje al objeto. Una definición de clase constituye la base para la reutilización en los niveles de modelado, diseño e implementación. [15]

El enfoque O.O. busca resarcir las deficiencias que se presentan en cada una de las etapas del ciclo de vida de la IS convencional, permitiendo obtener una mejor representación del mundo y de los requerimientos particulares de una aplicación en dicho mundo. Este enfoque puede ser aplicado indistintamente al análisis, diseño o desarrollo de una aplicación. No es estrictamente necesario usar el enfoque en todas las etapas del ciclo de vida de una aplicación. Si se desea, se puede elaborar un buen análisis y diseño O.O., aún cuando la implementación no necesariamente siga el mismo esquema. Sin embargo, es una excelente alternativa usar O.O. en todo el ciclo de vida, buscando aprovechar al máximo todas las bondades de este paradigma.

2.4.5 Ingeniería de Software Educativo con Enfoque Orientada a Objetos [14]

En el caso particular de la ISE, usar O.O. en todos los procesos computacionales (análisis, diseño y desarrollo) permite reflejar fácilmente en los ambientes todo aquello que es importante desde el punto de vista educativo. Esto forma parte del comportamiento del mundo y dicho comportamiento puede ser modelado claramente con este enfoque. Para establecer la estructura genérica sobre la cual se puedan "montar" micromundos lúdicos se va a tener en cuenta el conjunto de elementos mencionados por Galvis y se usa el enfoque O.O. para definir el modelo de datos.

A continuación se presenta una estructura genérica, planteada por Galvis, sobre la cual se puedan "montar" ambientes lúdicos. Se va a tener en cuenta un conjunto de elementos educativos y comunicacionales mencionados en Galvis y se usa el enfoque O.O. para definir el modelo de datos.

Siguiendo el ciclo de vida de un software, la siguiente descripción permite entender cada una de sus etapas, enriquecidas con el enfoque O.O mencionado.

2.4.5.1 Análisis

El objetivo de esta etapa es determinar el contexto en el cual se va a crear la aplicación y derivar de allí los requerimientos que deberá atender la solución interactiva, como complemento a otras soluciones basadas en uso de otros medios (personales, impresos, audio-visuales, experienciales), teniendo claro el rol de cada uno de los medios educativos seleccionados y la viabilidad de usarlos. De acuerdo con Galvis en esta etapa se establece como mínimo la siguiente información:

- ✓ Características de la población objetivo
- ✓ Conducta de entrada y campo vital
- ✓ Problema o necesidad a atender
- ✓ Principios pedagógicos y didácticos aplicables
- ✓ Justificación de uso de los medios interactivos como alternativa de solución.

2.4.5.2. Especificaciones de los Requerimientos.

Como síntesis de la etapa de análisis se deben formular los requerimientos que deberá atender el material interactivo que se desea obtener.

La especificación de requerimientos debe contener lo siguiente:

- ✓ Descripción de la Aplicación
- ✓ Las restricciones, basadas en la población objetivo, áreas de contenido, conducta de entrada, principios pedagógicos aplicables y modos de uso de la aplicación.

-
- ✓ Descripción de los posibles escenarios de interacción que tendrá el usuario.
 - ✓ Diagramas de interacción.

2.4.5.3 Diseño

El diseño del ambiente interactivo se realiza a tres niveles diferentes: educativo, comunicacional y computacional. La metodología de ISE original es fuerte en cuanto al diseño educativo y diseño comunicacional de MECs. En esta propuesta ISE-OO se van a tomar en cuenta estas fortalezas y se van a usar de manera que sean reflejadas en el diseño computacional de la aplicación y en la implementación de la misma.

Al diseñar el ambiente en el que se desarrollará la acción se deben definir claramente los elementos que se determinaron como necesarios en todo ambiente interactivo y aquellos deseables que convenga para el caso.

Se realiza el diseño usando el enfoque O.O., formalizando muchos de los aspectos relacionados con la aplicación, definiendo desde esta etapa los objetos, su comportamiento, el propósito de la aplicación, las restricciones existentes y los escenarios de interacción.

Galvis usa como base una propuesta que servirá como soporte al diseño O.O. y posterior diseño de datos e implementación de la aplicación. Se va usar UML para la notación del modelo. Se desea así obtener una arquitectura genérica para ambientes interactivos, que pueda extenderse para satisfacer necesidades de un problema en particular. Junto con la arquitectura se debe especificar la funcionalidad que el usuario tendrá sobre el modelo, para saber qué cosas puede hacer sobre él. A continuación se define cada una de las etapas del diseño: diseño educativo, diseño comunicacional diseño computacional.

2.4.5.3.1 Diseño Educativo

Tomando como punto de partida la necesidad o problema, así como la conducta de entrada y campo vital de la población objeto, se debe establecer lo que hay que enseñar o reforzar para subsanar con apoyo del software educativo las necesidades encontradas. Como resultado de la fase de diseño educativo se debe tener lo siguiente: contenido y su estructura; ambiente; sistema de evaluación. De acuerdo con Galvis el diseño educativo debe resolver los siguientes interrogantes: ¿Qué aprender con el MEC? ¿En qué ambiente aprenderlo?, ¿Cómo motivar y mantener motivados a los usuarios? ¿Cómo saber que el aprendizaje se está logrando?.

2.4.5.3.2 Diseño Comunicacional

En esta fase del proceso de diseño se define la interfaz (zona de comunicación usuario-programa) de la aplicación. En este momento se debe complementar ese bosquejo definiendo formalmente los objetos que posee cada pantalla y cuáles elementos del mundo son usados/afectados. Se toma como base la descripción macro dada en especificación. Es importante conseguir que la interfaz sea: amigable, flexible y agradable de usar; también debe ser consistente, altamente interactiva, que conlleve a tener mecanismos de comunicación entre el usuario y la aplicación. Igualmente al definir la interfaz se debe tener en cuenta: ¿cuáles dispositivos de entrada-salida conviene poner a disposición del usuario para trabajar con el Ambiente?, ¿qué zonas de comunicación entre usuario y programa debe tener el Ambiente?, ¿cuáles son las características de dichas zonas de comunicación?, ¿cómo verificar que la interfaz satisface los requisitos mínimos deseados?

2.4.5.3.3 Diseño Computacional

Al final de esta etapa se tiene como resultado, claramente definidas, cada una de las diferentes clases de objetos. Algunas clases necesitarán extenderse para ser usadas en el modelo. Además se puede dar el caso de agregar nuevas clases y relaciones al

modelo para dar mayor funcionalidad al modelo acorde con los requerimientos propios de la aplicación.

Junto al conjunto de clases, llamado también modelo estático del mundo, se debe ilustrar la lógica acerca de cómo se desarrollan cada una de las actividades en el modelo. Para ello se deben refinar los casos de uso. Para hacer esto se pueden usar diagramas de interacción que pueden ser de dos tipos: diagramas de secuencia o diagramas de colaboración. En estos diagramas ya se puede ver la secuencia de mensajes entre los diferentes objetos involucrados en cada caso de uso y se pueden modelar todas las alternativas que puedan presentarse en cada caso.

2.4.5.4 Desarrollo

En esta fase se implementa la aplicación usando toda la información obtenida anteriormente. Se toma la definición de clases y se implementa en el lenguaje y herramientas escogidas tomando en cuenta las restricciones computacionales que se tengan. Se establece la herramienta de desarrollo sobre la cual se va a implementar la aplicación. Los criterios para escogerla incluyen; costo, disponibilidad en el mercado, portabilidad de la aplicación desarrollada, facilidades al desarrollador (ambientes gráficos de desarrollo, mecanismos de depuración, manejo de versiones, etc.).

En el desarrollo se busca que el modelo del mundo sea independiente de la interfaz, esto facilita el trabajo y permite trabajar en paralelo. La interfaz se implementa usando la especificación del diseño comunicacional. En algunos ambientes de desarrollo la creación de ésta se facilita con herramientas visuales de desarrollo. En otros se tiene que programar cada uno de los elementos de la interfaz.

2.4.5.5 Prueba

Durante esta fase se pone a disposición de la población objetivo una versión beta del micromundo interactivo, con el propósito de verificar que efectivamente la aplicación satisface las necesidades y cumple con la funcionalidad definida.

2.5 MATERIALES EDUCATIVOS COMPUTARIZADOS

Los Materiales Educativos Computarizados o MEC, son todas aquellas aplicaciones dedicadas a apoyar el proceso educativo, es decir, complementar los otros medios y materiales de enseñanza – aprendizaje sin llegar a sustituirlos [1].

2.5.1 Características Esenciales de los Programas Educativos [14].

Los programas educativos pueden tratar las diferentes materias (matemáticas, idiomas, geografía, dibujo...), de formas muy diversas (a partir de cuestionarios, facilitando una información estructurada a los alumnos, mediante la simulación de fenómenos...) y ofrecer un entorno de trabajo más sensible a las circunstancias de los alumnos y más rico en posibilidades de interacción; pero todos comparten cinco características esenciales:

- ✓ Son materiales elaborados con una finalidad didáctica, como se desprende de la definición.
- ✓ Utilizan el ordenador como soporte en el que los alumnos realizan las actividades que ellos proponen.
- ✓ Son interactivos, contestan inmediatamente las acciones de los estudiantes y permiten un diálogo y un intercambio de informaciones entre el ordenador y los estudiantes.

- ✓ Individualizan el trabajo de los estudiantes, ya que se adaptan al ritmo de trabajo cada uno y pueden adaptar sus actividades según las actuaciones de los alumnos.

- ✓ Son fáciles de usar. Los conocimientos informáticos necesarios para utilizar la mayoría de estos programas son similares a los conocimientos de electrónica necesarios para usar un vídeo, es decir, son mínimos, aunque cada programa tiene unas reglas de funcionamiento que es necesario conocer.

2.5.2. Estructura Básica de los Programas Educativos [14].

La mayoría de los programas didácticos, igual que muchos de los programas informáticos nacidos sin finalidad educativa, tienen tres módulos principales claramente definidos: el módulo que gestiona la comunicación con el usuario (sistema input/output), el módulo que contiene debidamente organizados los contenidos informativos del programa (bases de datos) y el módulo que gestiona las actuaciones del ordenador y sus respuestas a las acciones de los usuarios (motor).

2.5.2.1. El Entorno de Comunicación o Interfaz.

La interfaz es el entorno a través del cual los programas establecen el diálogo con sus usuarios, y es la que posibilita la interactividad característica de estos materiales. Está integrada por dos sistemas:

El sistema de comunicación programa-usuario, que facilita la transmisión de informaciones al usuario por parte del ordenador, incluye:

- ✓ Las pantallas a través de las cuales los programas presentan información a los usuarios.

- ✓ Los informes y las fichas que proporcionen mediante las impresoras.

-
- ✓ El empleo de otros periféricos: altavoces, sintetizadores de voz, robots, módems, convertidores digitales-analógicos...
 - ✓ El sistema de comunicación usuario-programa, que facilita la transmisión de información del usuario hacia el ordenador, incluye:
 - El uso del teclado y el ratón, mediante los cuales los usuarios introducen al ordenador un conjunto de órdenes o respuestas que los programas reconocen.
 - El empleo de otros periféricos: micrófonos, lectores de fichas, teclados conceptuales, pantallas táctiles, lápices ópticos, modems, lectores de tarjetas, convertidores analógico-digitales.

2.5.2.2 Las Bases de Datos.

Las bases de datos contienen la información específica que cada programa presentará a los alumnos. Pueden estar constituidas por:

- ✓ **Modelos de comportamiento.** Representan la dinámica de unos sistemas. Distinguimos:
 - **Modelos físico-matemáticos**, que tienen unas leyes perfectamente determinadas por unas ecuaciones.
 - **Modelos no deterministas**, regidos por unas leyes no totalmente deterministas, que son representadas por ecuaciones con variables aleatorias, por grafos y por tablas de comportamiento.
- ✓ **Datos de tipo texto**, información alfanumérica.
- ✓ **Datos gráficos.** Las bases de datos pueden estar constituidas por dibujos, fotografías, secuencias de vídeo, etc.

- ✓ **Sonido.** Como los programas que permiten componer música, escuchar determinadas composiciones musicales y visionar sus partituras.

2.5.2.3 El Motor o Algoritmo.

El algoritmo del programa, en función de las acciones de los usuarios, gestiona las secuencias en que se presenta la información de las bases de datos y las actividades que pueden realizar los alumnos. Distinguimos 4 tipos de algoritmo:

- ✓ **Lineal**, cuando la secuencia de las actividades es única.
- ✓ **Ramificado**, cuando están predeterminadas posibles secuencias según las respuestas de los alumnos.
- ✓ **Tipo entorno**, cuando no hay secuencias predeterminadas para el acceso del usuario a la información principal y a las diferentes actividades. El estudiante elige qué ha de hacer y cuándo lo ha de hacer. Este entorno puede ser:
 - **Estático**, si el usuario sólo puede consultar (y en algunos casos aumentar o disminuir) la información que proporciona el entorno, pero no puede modificar su estructura.
 - **Dinámico**, si el usuario, además de consultar la información, también puede modificar el estado de los elementos que configuran el entorno.
- ✓ **Programable**, si a partir de una serie de elementos el usuario puede construir diversos entornos.
- ✓ **Instrumental**, si ofrece a los usuarios diversos instrumentos para realizar determinados trabajos.
- ✓ **Tipo sistema experto**, cuando el programa tiene un motor de inferencias y, mediante un diálogo bastante inteligente y libre con el alumno (sistemas dialogales), asesora al estudiante o tutoriza inteligentemente el aprendizaje. Su desarrollo está muy ligado con los avances en el campo de la Inteligencia Artificial.

2.6 PROCESO UNIFICADO

El proceso unificado de desarrollo nace en vista de la necesidad de crear un estándar para el desarrollo de software con la finalidad de satisfacer la creciente demanda de sistemas más grandes, modernos y poderosos y que a su vez permita disminuir el tiempo de desarrollo de los mismos. Además este proceso permite minimizar los riesgos de tener que codificar componentes que no serán utilizados correctamente o simplemente no son terminados.

El proceso unificado de desarrollo de software se apoya en tres principios básicos:

- ✓ **Dirigidos por casos de uso:** los casos de uso dirigen y controlan el proceso de desarrollo en su totalidad. “Un sistema de software ve la luz para dar servicio a sus usuarios. Por tanto, para construir un sistema con éxito debemos conocer lo que sus futuros usuarios necesitan y desean”.

Los usuarios del sistema requieren de respuestas diferentes dependiendo de lo que este solicitando, un caso de uso representa una pequeña parte de lo que puede brindar el sistema y el modelo de casos de usos agrupa todos los casos de uso que puede tener el sistema.

- ✓ **Centrado en la arquitectura:** es la pieza clave que permite comprender el sistema, organizar el desarrollo y hacer evolucionar el software.

- ✓ **Proceso iterativo e incremental:** el desarrollo se plantea de manera progresiva, de tal modo que se atenúen los riesgos y se planteen las cuestiones en el instante en el que estamos capacitados para resolverlas. Cada avance realizado en el proyecto se cuenta como una iteración, esto trae como consecuencia que una gran aplicación se descomponga en piezas más pequeñas, lo que permite visualizar la mayor parte de los detalles inherentes al desarrollo y

crecimiento de la aplicación, garantizando así, que se trabaje en forma más segura y eficiente.

El proceso unificado de desarrollo se define como un marco de trabajo genérico dirigido por caso de uso, centrado en la arquitectura, interactivo e incremental que puede especializarse para la gran variedad de las actividades necesarias para transformar los requisitos de un usuario en un sistema software.

El Proceso Unificado tiene dos dimensiones (Figura 2.2):

- ✓ Un eje horizontal que representa el tiempo y muestra los aspectos del ciclo de vida del proceso a lo largo de su desenvolvimiento.
- ✓ Un eje vertical que representa las disciplinas, las cuales agrupan actividades de una manera lógica de acuerdo a su naturaleza.

La primera dimensión representa el aspecto dinámico del proceso conforme se va desarrollando, se expresa en términos de fases, iteraciones e hitos.

La segunda dimensión representa el aspecto estático del proceso: cómo es descrito en términos de componentes del proceso, disciplinas, actividades, flujos de trabajo, artefactos y roles.

El Proceso Unificado se basa en componentes, lo que significa que el sistema en construcción está hecho de componentes de software interconectados por medio de interfaces bien definidas. El Proceso Unificado usa el Lenguaje de Modelado Unificado (UML) en la preparación de todos los planos del sistema. De hecho, UML es una parte integral del Proceso Unificado, fueron desarrollados a la par [16].

El Proceso Unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo consta de cuatro fases: Inicio, elaboración, Construcción y Transición. Cada una de las fases se subdivide a su vez en

iteraciones. Cada iteración está formada por cinco flujos de trabajo fundamentales adaptables según sea la fase para el cual se está desarrollando. Estos flujos son: Requisitos, Análisis, Diseño, Implementación y Pruebas, como se muestra en la figura 2.2.

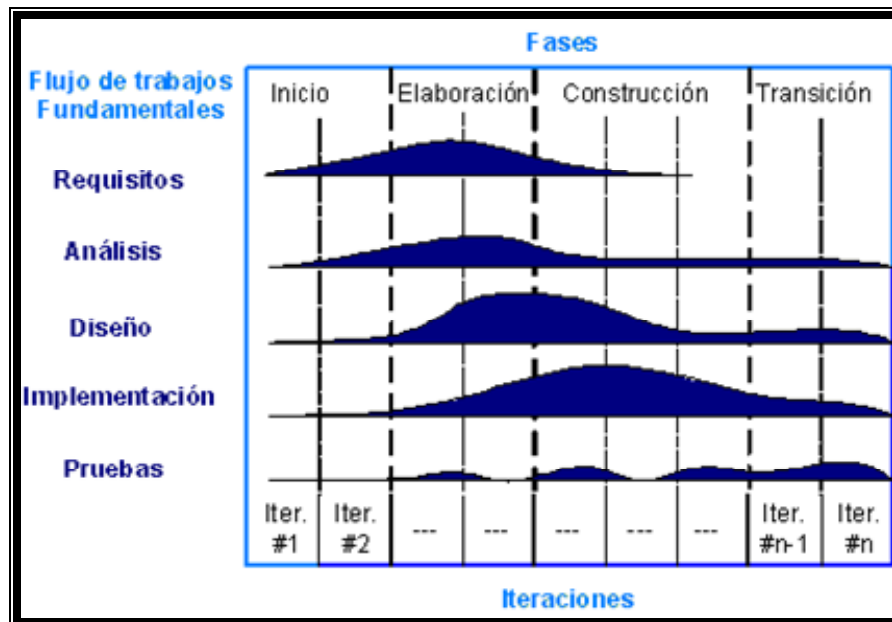


Figura 2.2 El Proceso Unificado de Desarrollo de Software.⁽¹⁶⁾

2.6.1 Fases del proceso unificado de Desarrollo de Software_[16]

El proceso unificado se divide en cuatro fases.

2.6.1.1 Fase de Inicio

El objetivo de esta fase es desarrollar el análisis de negocio hasta el punto necesario para justificar la puesta en marcha del proyecto. Para realizar este análisis se debe:

- ✓ Delimitar el ámbito del sistema propuesto, es decir, definir los límites del sistema y empezar a identificar las interfaces con sistemas relacionados que estén fuera de los límites.

- ✓ Describir o esbozar una propuesta de la arquitectura del sistema, y en especial de aquellas partes del sistema que sean nuevas, arriesgadas o difíciles.

- ✓ Identificar los riesgos críticos, es decir, los que afectan a la capacidad de construir el sistema, y determinar si podemos encontrar una forma de mitigarlos.

2.6.1.2 Fase de Elaboración

En esta fase se debe:

- ✓ Establecer una base de la arquitectura sólida (la línea base de la arquitectura) que cubre la funcionalidad del sistema significativa arquitectónicamente y las características importantes para los actores involucrados. Esta línea base consiste en los artefactos de los medios, la descripción de la arquitectura y en una implementación ejecutable de esta.

- ✓ Continuar la observación y control de los riesgos críticos que aun queden, e identificar los riesgos significativos hasta el punto de que podamos estimar su impacto en el análisis de negocio y en la apuesta económica.
- ✓ Recopilar la mayor parte de los requisitos que aun queden pendientes, formulando los requisitos como casos de uso.

- ✓ Se especifican los niveles por atributos de calidad.

2.6.1.3 Fase de Construcción.

En esta fase la línea base de la arquitectura crece hasta convertirse en el sistema completo, es decir, se crea el producto hasta obtener una versión beta. Para ello, se detallan los casos de uso y escenarios restantes, se modifica si es necesario, la descripción de la arquitectura y se continúan los flujos de trabajos a través de

iteraciones adicionales, dejando cerrados los modelos de análisis, diseño e implementación. Además se integran los subsistemas.

Generalmente esta fase requiere un mayor número de iteraciones que las fases anteriores. En muchos casos el tener un trabajo de requisito, análisis t diseño pobre implica que la construcción genere varias iteraciones, debido a que la construcción lleva una cantidad de tiempo grande. Una de las grandes ventajas de construir un software utilizando un enfoque que use múltiples fases y un desarrollo iterativo e incremental es que permite la asignación de recursos y de tiempo a lo largo de ciclo de vida.

2.6.1.4. Fase de Transición.

Esta fase se centra en implementar el producto en su entorno de operación, es decir, un número reducido de usuarios con experiencia prueba el producto e informa los defectos y deficiencias, se descubren riesgos inesperados y se eliminan ambigüedades y lagunas en la documentación de usuarios.

En esta fase no se busca reformular el producto sino que se buscan pequeñas deficiencias que pasaron desapercibidas en la fase de construcción.

2.6.2 Flujos de Trabajo.

2.6.2.1 Captura de Requisitos.

- ✓ Encontrar requisitos funcionales, no funcionales y adicionales.
- ✓ Encontrar actores y casos de uso.
 - Encontrar actores.
 - Encontrar los casos de uso.
 - Describir brevemente cada caso de uso.
 - Describir el modelo de caso de uso en su totalidad.

- ✓ Determinar la prioridad de los caso de uso.
- ✓ Detallar un caso de uso.
 - Estructurar la descripción de los casos de uso.
 - Formalizar la descripción de los casos de uso.
- ✓ Construir un prototipo de la interfaz de usuario.
 - Crear el diseño lógico de una interfaz de usuario.
 - Creación de un diseño y en prototipo físico de la interfaz de usuario.
- ✓ Estructurar el modelo de caso de uso (diagrama).
 - Identificar las descripciones de funcionalidad compartida.
 - Identificar las descripciones de funcionalidades adicionales y opcionales.
 - Identificar las relaciones entre casos de uso.

2.6.2.2 Análisis.

- ✓ Análisis de la arquitectura.
 - Identificar los paquetes de análisis.
 - Identificar las clases de entidades obvias.
 - Identificar los requisitos especiales comunes.
- ✓ Analizar un caso de uso.
 - Identificación de clase de análisis.
 - Descripción de iteraciones entre objetos del análisis.
 - Captura de requisitos especiales.
- ✓ Analizar un Paquete.

2.6.2.3 Diseño.

- ✓ Diseño de la arquitectura.

- Identificación de nodos y configuración de red.
- Identificación de subsistemas y de sus interfaces.
 - Identificación de subsistemas de aplicación.
 - Identificación de subsistemas de intermedios y de software del sistema.
 - Definición de dependencias entre subsistemas.
 - Identificación de clases de diseño relevantes para la arquitectura.
 - Identificación de clases del diseño a partir de clases del análisis.
 - Identificación de clases activas.
- ✓ Diseño de un caso de uso.
 - Identificación de clases del diseño participantes.
 - Descripción de las interacciones entre objetos del diseño.
 - Identificación de subsistemas e interfaces participantes.
 - Descripción de interacciones entre subsistemas.
 - Captura de requisitos de implementación.
- ✓ Diseño de una clase.
 - Esbozar la clase del diseño.
 - Identificar operaciones.
 - Identificar atributos.
 - Identificar asociaciones y agregaciones.
 - Identificar las generalizaciones.
 - Describir métodos.
 - Describir estados.
 - Tratar requisitos especiales.
- ✓ Diseño de un subsistema.
 - Mantenimiento de las dependencias entre subsistemas.
 - Mantenimiento de interfaces proporcionadas por el subsistema.

- Mantenimiento de los contenidos de los subsistemas.

2.6.2.4 Implementación.

- ✓ Implementación de la arquitectura.
 - Identificación de los componentes significativos arquitectónicamente.
 - Identificación de los componentes ejecutables y asignación de éstos a nodos.
- ✓ Integrar sistema.
 - Planificación de una construcción.
 - Integración de una construcción.
- ✓ Implementar un subsistema.
 - Mantenimientos de los contenidos de los subsistemas.
- ✓ Implementar una clase.
 - Esbozo de los componentes ficheros.
 - Generación de código a partir de una clase de diseño.
 - Implementación de operaciones.
 - Los componentes han de proporcionar las interfaces aplicadas.
- ✓ Realizar pruebas de unidad.
 - Realización de pruebas de especificación.
 - Realización de pruebas de estructura.

2.6.2.5 Prueba

- ✓ Planificar pruebas.
- ✓ Diseñar pruebas.
 - Diseño de los casos de uso de prueba de integración.
 - Diseño de los casos de prueba de sistema.
 - Diseño de los casos de prueba de regresión.
 - Identificación y estructuración de los procedimientos de prueba.

- ✓ Implementar pruebas.
- ✓ Realizar pruebas de integración.
- ✓ Realizar prueba de sistema.
- ✓ Evaluar prueba.

2.7 LENGUAJE UNIFICADO DE CONSTRUCCION DE MODELADO ^[17]

UML es un lenguaje basado en notación gráfica el cual permite especificar, construir, visualizar y documentar los objetos de u sistema programado Puede ser utilizado por cualquier metodología de análisis y diseño orientada por objetos para expresar las relaciones estáticas o dinámicas que existen entre ellos.

UML define varios modelos para la representación de los sistemas:

- ✓ El modelo de clases que captura la estructura estática
- ✓ El modelo de estados que expresa el comportamiento dinámico de los objetos
- ✓ El modelo de casos de uso que describe las necesidades del usuario
- ✓ El modelo de interacción que representa los escenarios y los flujos de mensajes
- ✓ El modelo de realización que muestra las unidades de trabajo
- ✓ El modelo de despliegue que precisa el reparto de procesos

La Figura 2.6 muestra los nueve tipos de diagramas diferentes definidos en UML

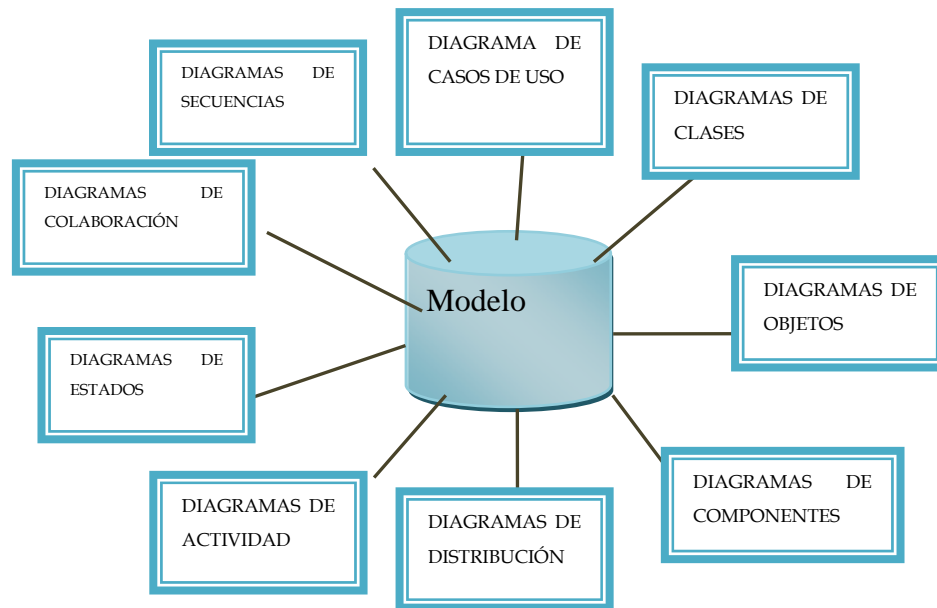


Figura 2.3 Diagramas de UML.

2.7.1. Diagramas Utilizados en UML.

Los diagramas estáticos son:

- ✓ **Diagrama de clases:** muestra las clases, interfaces, colaboraciones y sus relaciones. Son los más comunes y dan una vista estática del proyecto.
- ✓ **Diagrama de objetos:** Es un diagrama de instancias de las clases mostradas en el diagrama de clases. Muestra las instancias y como se relacionan entre ellas. Se da una visión de casos reales.
- ✓ **Diagrama de componentes:** Muestran la organización de los componentes del sistema. Un componente se corresponde con una o varias clases, interfaces o colaboraciones.

✓ **Diagrama de despliegue:** Muestra los nodos y sus relaciones. Un nodo es un conjunto de componentes. Se utiliza para reducir la complejidad de los diagramas de clases y componentes de un gran sistema. Sirve como resumen e índice.

✓ **Diagrama de casos de uso:** Muestran los casos de uso, actores y sus relaciones. Muestra quien puede hacer que y relaciones existen entre acciones (casos de uso). Son muy importantes para modelar y organizar el comportamiento del sistema.

Los diagramas dinámicos son:

✓ **Diagrama de secuencia y Diagrama de colaboración:** Muestran a los diferentes objetos y las relaciones que pueden tener entre ellos, los mensajes que se envían entre ellos. **Son dos diagramas diferentes, que se puede pasar de uno a otro sin pérdida de información**, pero que nos dan puntos de vista diferentes del sistema. En resumen, cualquiera de los dos es un Diagrama de Interacción.

✓ **Diagrama de estados:** muestra los estados, eventos, transiciones y actividades de los diferentes objetos. Son útiles en sistemas que reaccionen a eventos.

✓ **Diagrama de actividades:** Es un caso especial del diagrama de estados. Muestra el flujo entre los objetos. Se utilizan para modelar el funcionamiento del sistema y el flujo de control entre objetos.

2.7.2 Extensiones de UML para modelar Aplicaciones Web. [18]

Una extensión de UML es expresada en función de estereotipos, valores de etiquetas y restricciones combinando estos mecanismos nos permiten crear nuevos tipos de bloques de construcción que podamos usar en el modelo (tabla 1.1).

✓ **Estereotipo:** Es una extensión del vocabulario en el lenguaje. Un estereotipo nos permite agregar nuevos significados semánticos a los elementos de modelo. El estereotipo puede ser aplicado casi a cada uno de los elementos del modelo y generalmente son representados como cadena entre un par de << >>. Sin embargo también pueden representar por un icono nuevo.

✓ **Valor etiquetado:** es una extensión a la propiedad del elemento de un modelo. La mayoría de los elementos del modelo tienen propiedades asociadas con ellos mismos. Las clases, tienen nombres, visibilidad, persistencia, y otros atributos asociados con ellos. Un valor etiquetado es la definición de una nueva propiedad que puede asociarse con un elemento de modelo. Un valor etiquetado es representado en el diagrama como una cadena encerrados entre corchetes [].

✓ **Restricción:** Es una extensión a la semántica del lenguaje. Es una regla que define como el modelo se va a integrar o poner todo junto. Una restricción especifica las condiciones bajo las cuales el modelo puede ser considerado “bien formado”. Las restricciones son representadas como cadenas entre un par de llaves {}.

2.8 INTERNET

Básicamente millones de ordenadores conectados entre sí independientes unos de otros. Para que todos estos ordenadores puedan coexistir y comunicarse entre ellos, deben ponerse de acuerdo. Con este motivo fueron creados los Protocolos, que son reglas de comunicación que han de adoptarse para ser entendido por los otros ordenadores de la red. Los dos protocolos más importantes son Protocolo de Control de Transmisión (Transfer Control Protocol) y el Protocolo de Internet (Internet Protocol). Un ordenador, si maneja estos dos protocolos, no tendrá ningún problema para ser entendido por los demás ordenadores de Internet. Uno de los servicios de Internet es el World Wide Web a o llamado Web, es una herramienta basada en

hipertextos que permite recuperar y mostrar información basada en búsqueda por palabras claves.

2.9 MULTIMEDIA

Es cualquier combinación de texto, arte gráfico, sonido, animación y vídeo utilizado por computadora u otros medios electrónicos, mejorando las interfaces tradicionales basada solo en texto, proporcionando beneficios importantes que atraen y mantienen la atención y el interés. Multimedia mejora la retención de la información presentada, cuando está bien diseñada puede ser enormemente divertida.

2.10 DREAMWEAVER

Es un editor de código HTML profesional para el diseño visual y la administración de sitios y páginas Web. Con lo que podría controlar manualmente el código HTML como trabajar en un entorno de edición visual, facilitando herramientas útiles para mejorar el diseño Web. Las funciones de edición visual de Dreamweaver permiten añadir diseño y funcionalidad rápidamente sin escribir una sola línea de código. Con esta herramienta se pueden crear objetos y comandos, modificar métodos abreviados de teclado e incluso escribir código JavaScript; para ampliar las posibilidades que ofrece Dreamweaver con nuevos comportamientos, inspectores de propiedades e informes de sitios.

2.11 FLASH

Es el futuro de la verdadera Web multimedia animada e interactiva. Es un paquete completo que incluye herramientas de dibujo, de animación de imágenes, de gestión de sonidos e interactividad, y que permite crear imágenes a partir de nada, sin tener que recurrir a otro programa. Sus posibilidades son tan versátiles que van desde crear un simple logo animado hasta una página Web completa, donde la animación y la interactividad van acompañadas de un diseño gráfico que pueden hacer del conjunto algo verdaderamente llamativo. Además de crear objetos, las páginas pueden contener imágenes creadas por aplicaciones externas en formato vectorial o en mapa de bits, elementos de sonido y fragmentos de video.

CAPITULO 3
FASE DE INICIO

3.1 INTRODUCCION

El propósito de la fase de inicio es crear una vista general de los objetivos del proyecto, establecer qué es lo que se debe cubrir con el desarrollo del sistema, definir los límites dentro de los cuales se busca los riesgos críticos, con el fin de crear el nuevo enfoque del ambiente de aprendizaje desarrollado para la unidad curricular “Taller de Programación Orientada a Objetos”. En primera instancia, se inicia con una descripción de la metodología utilizada y de las necesidades educativas que deben ser atendidas y que justifican el desarrollo de esta aplicación, todo ello basándose en las técnicas de desarrollo de ambientes educativos: consulta a fuentes apropiadas, identificación de problemas educativos y análisis de posibles causas de los problemas detectados.

3.2 METODOLOGIA UTILIZADA

La creación de la herramienta de aprendizaje interactiva, Software Educativo para el Taller de Programación Orientada a Objetos, se desarrolló empleando el Proceso Unificado de Rational (RUP), junto con el Lenguaje Unificado de Modelado (UML) los cuales constituyen una metodología que ofrece un proceso estructurado de técnicas bien definidas para el análisis, implementación y documentación de sistemas orientados a objetos. Esta metodología fue complementada con Ingeniería de Software Educativo (ISE) propuesta por Álvaro Galvis, la cual logra articular debidamente los aspectos educacionales, comunicacionales y computacionales que intervienen en la creación de ambientes educativos multimedia apoyados con MEC's (Materiales Educativos Computarizados).

3.3 PLANIFICACION DE LA FASE DE INICIO

Por las características pedagógicas que debe incluir el tipo de software a desarrollar, durante la captura de requisitos se recopilará información de la asignatura Taller de Programación Orientada a Objetos, como se lleva a cabo el proceso de enseñanza-aprendizaje incluyendo los medios y materiales utilizada durante el proceso de aprendizaje. De igual manera en esta fase de definirá el ámbito del sistema.

3.4 MODELO DE DOMINIO

Un Modelo de Dominio es un artefacto de la disciplina de análisis, construido con las reglas de UML durante la fase de concepción, en la tarea construcción del modelo de dominio, presentado como uno o más diagramas de clases y que contiene, no conceptos propios de un sistema de software sino de la propia realidad física.

Los modelos de dominio pueden utilizarse para capturar y expresar el entendimiento ganado en un área bajo análisis como paso previo al diseño de un sistema, ya sea de software o de otro tipo. Similares a los mapas mentales utilizados en el aprendizaje, el modelo de dominio es utilizado por el analista como un medio para comprender el sector industrial o de negocios al cual el sistema va a servir.

En esta fase se intenta llegar a una comprensión del problema, sin entrar en detalle de implementación de la aplicación, con una representación visual de los aspectos presentes en el mundo real donde se desarrolla la asignatura Taller de Programación Orientada a Objetos.

La representación del modelo de dominio se observa en la figura 3.1, donde se aprecia la estructura del proceso de enseñanza-aprendizaje que se lleva a cabo actualmente en la asignatura Taller de Programación Orientada a Objetos.

Como muestra la figura 3.1 en el proceso intervienen estudiantes y profesores, los estudiantes inscriben la materia la cual es dictada por el profesor, el mismo recopila y prepara el material de apoyo relacionado con contenido programático correspondiente a la asignatura.

El profesor imparte la clase y guía al estudiante en el proceso de aprendizaje el cual tiene el deber de recopilar información así como de resolver ejercicios prácticos relacionada con el contenido programático con el fin de estar preparado para las evaluaciones y programas propuestos por el profesor, los cuales serán desarrollados en el computador, y permitirán evaluar los conocimientos del estudiante.

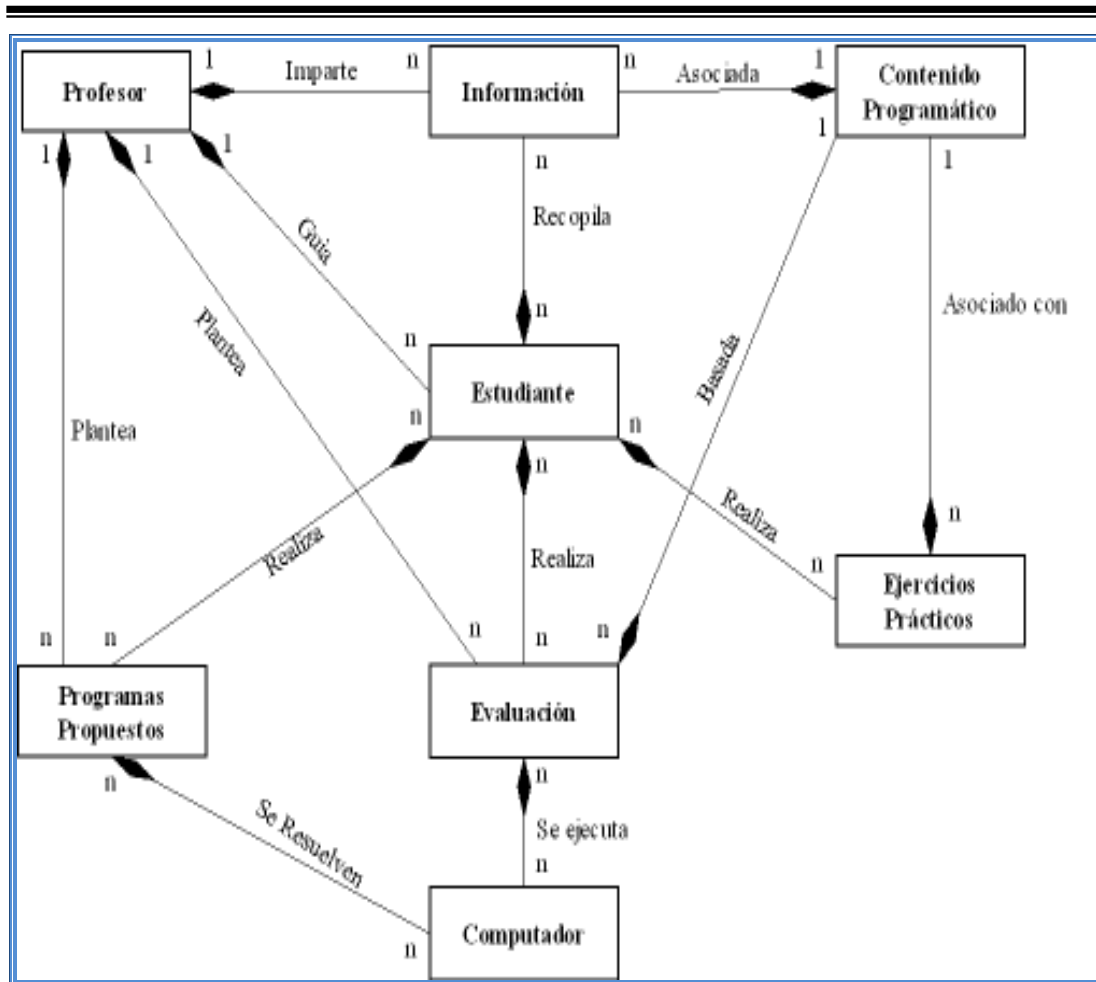


Figura 3.1 Modelo de Dominio de la Asignatura Taller de Programación Orientada a Objetos

3.5 IDENTIFICACION DE RIESGOS

Un riesgo es la probabilidad de que ocurra algún suceso adverso o no deseado en el proyecto. El tratamiento de riesgos permite la identificación así como también el plan de contingencia a seguir para mitigar los efectos del mismo sobre el proyecto.

Entre los riesgos críticos que pueden surgir durante el desarrollo de la aplicación cabe destacar:

- ✓ Escogencia de herramientas inadecuadas para el desarrollo del sistema.

- ✓ Fallas en el acceso a la base de datos.
- ✓ Fallas de comunicación entre los software escogidos para llevar a cabo la aplicación.
- ✓ La herramienta presenta poca interactividad con el usuario.
- ✓ El tiempo de respuesta a una acción requería por el usuario es considerablemente largo.
- ✓ No contar con un plan de evaluación adecuado para verificar que el estudiante ha logrado los objetivos respectivos del tema.
- ✓ El sistema no presenta ejemplos sencillos audiovisuales, enriquecidos de animaciones, gráficos, textos y cualquier otra herramienta que facilite el proceso de enseñanza-aprendizaje
- ✓ El sistema no le proporciona al usuario la facilidad de navegar a su propio ritmo de aprendizaje

Entre los riesgos secundarios para el software que tienen un impacto leve en el desarrollo del proyecto se encuentran:

- ✓ Falta de recurso, al no contar con la plataforma de Hardware y software mínimo requerido para el desarrollo de la aplicación.

3.6 FLUJO DE TRABAJO REQUISITOS

Su propósito fundamental es guiar el desarrollo del proyecto hacia la descripción de las condiciones que debe cumplir el sistema, y así obtener una arquitectura sólida. Para la representación gráfica de los elementos que componen los flujos de trabajo se

hizo uso del conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos (UML).

En este flujo de trabajo se incluyeron las siguientes actividades: analizar las necesidades educativas, comprender el contexto del sistema, capturar los requisitos funcionales y no funcionales.

3.6.1 Contexto de la aplicación

En esta etapa se identificaron los usuarios destino de este software, determinando las características de la población objetivo, paso importante para el análisis de las necesidades educativas existente que la aplicación debe satisfacer y para la especificación de requerimientos del sistema. Con base en esto, se determino la siguiente información:

3.6.1.1 Características de la población objetivo

La audiencia principal, para la cual es diseñado este software, está conformada por estudiantes de la carrera de Ingeniería en Computación de la Universidad de Oriente. Esta población está en la etapa inicial de sus estudios universitarios, tomando en cuenta que la asignatura Taller de Programación Orientada a Objetos se dicta en el tercer semestre de acuerdo al pensum de estudio vigente de esta carrera.

La población objetivo son jóvenes con edades comprendidas entre 19 a 20 años principalmente (Ver Apéndice B), de ambos sexos, con características físicas y psicológicas normales, con muchas expectativas ya que muchos de ellos poseen poco o ningún conocimiento acerca del mundo de la programación.

El software educativo será utilizado igualmente por los profesores que dictan la asignatura, los cuales están relacionados con el mundo de la computación, son intelectualmente sensitivos y de carácter abierto, que podrán usarlo como guía y herramienta de apoyo.

3.6.1.2 Medios y materiales de enseñanza y aprendizaje

El proceso de enseñanza-aprendizaje, es el que lleva a cabo un conjunto de acciones dirigidas al logro de un aprendizaje significativo y constructivo, que involucra qué es lo que se aprende y cómo se aprende, con el único fin de formar a un aprendiz o estudiante en un tema determinado, para ello existen ambientes especializados como el salón de clases, lugar en el que el profesor y el estudiante, con apoyo de los materiales y equipos de que disponen, interactúan bajo una metodología de enseñanza propuesta por el profesor, con la finalidad de activar el proceso de aprendizaje.

El medio donde se lleva a cabo la enseñanza de la asignatura Taller de programación orientada a Objetos es la planta física o laboratorio destinado y acondicionado para dar clases.

El profesor, para transmitir el contenido de la asignatura, cuenta con apenas 3 horas semanales de clases que son puramente práctica, con materiales didácticos tradicionales como lo son, el pizarrón, marcadores, las transparencias y diapositivas en algunos casos. Cada estudiante cuenta con la disposición de un computador para realizar las prácticas y evaluaciones propuestas por el profesor.

El estudiante apoya su proceso de aprendizaje de la materia, a través de las investigaciones realizadas en los libros, revistas, manuales, internet, ejemplos prácticos facilitado por el profesor, entre otros. Con el fin de complementar los conocimientos adquiridos en las clases; sin embargo, al no conseguir una bibliografía que abarque todos los tópicos de la materia, éste tiene que realizar una búsqueda que puede conllevar a información redundante, ocasionando pérdida de tiempo y dinero.

El profesor responde a las dudas de los alumnos, estimula su participación con cuestionamientos al grupo y proyectos para realizarse dentro y fuera de clase, bien sea en forma individual o grupal. También asigna horas de consultas personalizadas,

donde las nuevas inquietudes de los alumnos durante sus horas de estudio son atendidas para así complementar la enseñanza.

3.6.2 Necesidad Educativa

El análisis de las necesidades educativas es el primer paso dado del proceso de creación del ambiente de enseñanza/aprendizaje apoyado en el computador de la unidad curricular Taller de Programación orientada a Objetos, el cual responde a necesidades educativas prioritarias y relevantes determinadas en esta fase de análisis. En esta sección se identifica, a través del modelo educativo empleado en clase, las debilidades de este sistema, sus posibles causas y soluciones.

3.6.2.1 Identificación del Problema Educativo

El profesor, los alumnos y los registros académicos, son una fuente de información primaria, para detectar y priorizar aspectos problemáticos que surgen en la programación sinóptica de la asignatura Taller de Programación Orientado a Objetos. Es gracias a ellos que se puede especificar en qué punto del contenido se presenta mayor dificultad de comprensión, que impiden un rendimiento poco satisfactorio por evaluaciones, así como también los aspectos problemáticos que se presentan en el logro de los objetivos de enseñanza/aprendizaje de esta cátedra.

La metodología aplicada para la recopilación de información es la entrevista a los estudiantes, tomando en cuenta que es una población grande y se necesita recolectar cierta cantidad de datos y opiniones para lo cual este método es el más rápido y objetivo. Las encuestas son anónimas y basadas en preguntas cerradas usando criterios de Sistemas de Información. Otra fuente de información importante son los registros académicos (notas finales) de los últimos semestres cursados los cuales también son objeto de análisis.

3.6.2.2 Análisis de los Registros Académicos

El estudio del rendimiento académico de los semestres anteriores reflejado en las notas finales obtenidas por los estudiantes, arroja importante información sobre la puntuación promedio de aprobación de la asignatura y el nivel de repitencia (Ver Apéndice C). Los cuales demuestran resultados poco alentadores tanto para los profesores debido al nivel de alumnos aplazados, como para los estudiantes.

3.6.2.3 Análisis de Entrevistas a Estudiantes

Estas encuestas (Ver apéndice A) están dirigida a estudiantes que ya cursaron la asignatura en semestres anteriores, y se elabora con la finalidad de: determinar en qué medios apoyaron su proceso de aprendizaje, que unidades del contenido programático presentaron para ellos mayor nivel de dificultad; el grado de importancia que posee la asignatura para el desarrollo de la carrera y el nivel de repitencia. Igualmente se hace énfasis en recoger sus opiniones sobre el sistema actual de enseñanza así como el desarrollo de una nueva herramienta para apoyar el proceso enseñanza/aprendizaje, entre otras.

Los resultados de las encuestas se muestran en el apéndice B.

3.6.2.4 Problemas Detectados y sus Causas

Como resultado del análisis de la información recopilada en la etapa anterior, se obtiene una lista de necesidades educativas que evidencian la problemática existente en el actual proceso de enseñanza de la asignatura Taller de Programación Orientada a Objetos.

Entre los problemas detectados se encuentran:

- ✓ El sistema actual de enseñanza donde el profesor constituye la principal fuente de información, contando con el marcador y el pizarrón como únicos materiales didácticos, brinda pocas oportunidades al estudiante de adquirir o afianzar el conocimiento. Esto debido a limitaciones de la casa de estudio.

- ✓ Existen muchos conceptos complejos cuya aplicación inmediata no están normalmente a disposición del aprendiz, lo que conlleva a que se le presente cierto grado de dificultad en la comprensión.
- ✓ Las horas de clases asignadas a la asignatura son pocas por lo cual el profesor se ve limitado en tiempo al impartir todo el contenido programático y no puede en horas de clases cubrir las necesidades de conocimiento, inquietudes y curiosidades de los estudiantes sobre el tema dado.
- ✓ No todos los aprendices tienen el mismo ritmo de asimilación y captación de los conocimientos impartidos, así como también pueden no tener los conocimientos de base o carecen de motivación para estudiar los temas.
- ✓ Pese a las diversas estrategias empleada por el profesor para establecer una clase dinámica, éste no logra la participación activa de los alumnos durante las horas de clase.

3.6.2.5 Análisis de alternativas de Solución

Algunas de las necesidades pueden resolverse incluyendo la participación del profesor donde podría incorporar un nuevo modelo educativo en clase y un MEC (Material Educativo Computarizado) que incluya micromundos que sean atractivos y significantes para los aprendices en los que se puedan practicar un tópico de la materia hasta lograr un amplio nivel de dominio, ambos trabajando de forma simultánea, beneficiando de esta forma el cómo se dicta la asignatura y la adquisición de los conocimientos por parte del estudiantado.

3.6.2.6 Establecimiento del Rol del Computador

Un computador es una herramienta versátil utilizada para reforzar el aprendizaje, en el Taller de Programación Orientada a Objetos se cuenta con un determinado número

de computadores que satisfacen las necesidades de los alumnos, puesto que ya se cuenta con el recurso se aprovechará su capacidad de almacenamiento y de ejecución para la creación de micromundos interactivos donde se lleven a cabo experiencias didácticas que favorecen el aprendizaje.

3.6.2.7 Selección o Planeación de Desarrollo de MEC's

Por no identificarse Materiales Educativos Computarizados que puedan adaptarse a la enseñanza de Análisis y Diseño Educativo para la asignatura Taller de Programación Orientada a Objetos, y por contar con los medios necesarios para su planeamiento y elaboración, se desarrollará una aplicación con la finalidad de satisfacer las necesidades educativas de los estudiantes de la asignatura.

3.6.3 Requisitos Funcionales

La especificación de los requisitos funcionales contiene los aspectos del contenido o comportamiento del software que es requerido por el usuario. Tomando en consideración los aspectos de la sección anterior, para la creación del material educativo computarizado que se requiere para reforzar los conocimientos impartidos en la asignatura Taller de programación orientada a Objetos; se crea una lista de características que pueden ser implementadas en el sistema proporcionando los siguientes beneficios:

- ✓ Apoyar la labor del profesor, reduciendo su carga de trabajo repetitivo.
- ✓ Debe abarcar el contenido programático de la asignatura taller de Programación Orientada a Objetos.
- ✓ El software debe estar dotado de ricas animaciones y mecanismos de interacción que motiven a sus usuarios

FASE DE INICIO

-
- ✓ El sistema debe mostrar ejemplos sencillos audiovisuales, enriquecidos de animaciones, gráficos, textos y cualquier otra herramienta que facilite el proceso enseñanza/aprendizaje.
 - ✓ La aplicación debe ofrecerle al usuario facilidades en cuanto a su uso y manejo.
 - ✓ El sistema debe facilitar la navegación a medida que se enlazan los capítulos a las lecciones y a la evaluación.
 - ✓ Una fácil navegación a medida que pasa de un tópico a otro, para que el usuario tenga el control de decidir que desea ver o estudiar en un momento determinado.
 - ✓ El sistema le debe facilitar al usuario navegar a su propio ritmo de aprendizaje.
 - ✓ Permitir al usuario una rápida asimilación de los conceptos gracias a la variedad de medios y estímulos que utiliza.
 - ✓ El software debe ser diseñado con restricciones de entrada de usuarios, esto con la finalidad de mantener la seguridad de la información almacenada en la base de datos.
 - ✓ El software diseñado debe permitir el mantenimiento sin ninguna complicación.
 - ✓ Permitir el acceso a la información de manera ágil y eficaz, simplificando los procesos cognitivos y facilitando las estrategias de aprendizaje.
 - ✓ Permitir la revisión de sus conocimientos en las diferentes unidades del contenido programático.

- ✓ La aplicación debe contar con un manual de usuario.

3.6.4 Requisitos no funcionales

La especificación de los requisitos no funcionales incide en el diseño de los componentes de computación del software. Por este motivo, es importante que se tenga establecido con anterioridad los requerimientos de equipos y el soporte lógico necesario para el desarrollo y la operación de la aplicación.

3.6.4.1 Requisitos de Software

- ✓ Windows XP, Windows Vista
- ✓ Adobe Flash CS3
- ✓ Adobe Dreamweaver CS3.
- ✓ Microsoft Internet Explorer versión 6 en adelante o mozilla FireFox 2.7 en adelante.

3.6.4.2 Requisitos mínimos de hardware

- ✓ Procesador Intel Pentium a 600 MHz
- ✓ Mínimo 128 Mb de memoria RAM
- ✓ Tarjeta de vídeo de 16 bits de color y 4 Mb de memoria
- ✓ Teclado Windows, mouse y cornetas estándar e impresora
- ✓ Unidad de CD-ROM
- ✓ Monitor a color de 16 bits con capacidad para resolución de 800 x 600 píxeles

- ✓ 256 MB de espacio disponible en disco


3.6.5 Modelo de Casos de Usos

Un caso de uso es una secuencia de acciones que el sistema lleva a cabo para ofrecer algún resultado de valor para un actor, quien es una entidad humana o un sistema externo, que utiliza el sistema para interactuar con los casos de uso. El modelo de casos de uso es una representación de todos los actores, casos de uso y las relaciones existentes entre estos.

El modelo de casos de uso permite establecer las condiciones y posibilidades que debe cumplir el sistema a desarrollar, es decir, los casos de uso modelan los requisitos funcionales.

3.6.5.1 Actores del Sistema

Los actores del sistema representan cualquier elemento que intercambia información con el sistema SETPOO y que está fuera de él. La identificación de los actores del sistema es el primer paso para la realización del modelo de los casos de uso. En este sistema, un actor es un rol que un usuario juega con respecto al sistema, no necesariamente representa a una persona en particular, sino más bien la labor que realiza.

ACTORES	DESCRIPCIÓN
 <p data-bbox="402 1722 548 1753">Estudiante</p>	<p data-bbox="581 1537 1390 1621">Representa a la persona que va a interactuar con el software educativo, con la finalidad de adquirir conocimientos.</p> <p data-bbox="594 1646 1377 1785">El estudiante, que establecerá su propia forma de navegación dentro del software, lo utilizara para buscar información y autoevaluarse.</p>

FASE DE INICIO



 <p>Profesor</p>	<p>Persona que encuentra en el software apoyo para la enseñanza de la asignatura que dicta y evalúa su contenido. El profesor ejercerá una función de administrador de sistema pues él será el que se encargue de llevar a cabo las labores de actualizaciones, ingresos de nuevos estudiantes, entre otras.</p>
 <p>Base de Datos</p>	<p>Sistema gestor de base de datos.</p>

Tabla 3.1 Actores del sistema SETPOO

3.6.5.2 Caso de Uso General del Sistema SETPOO

Los casos de usos que representa el software educativo durante esta fase de inicio son los siguientes:

FASE DE INICIO

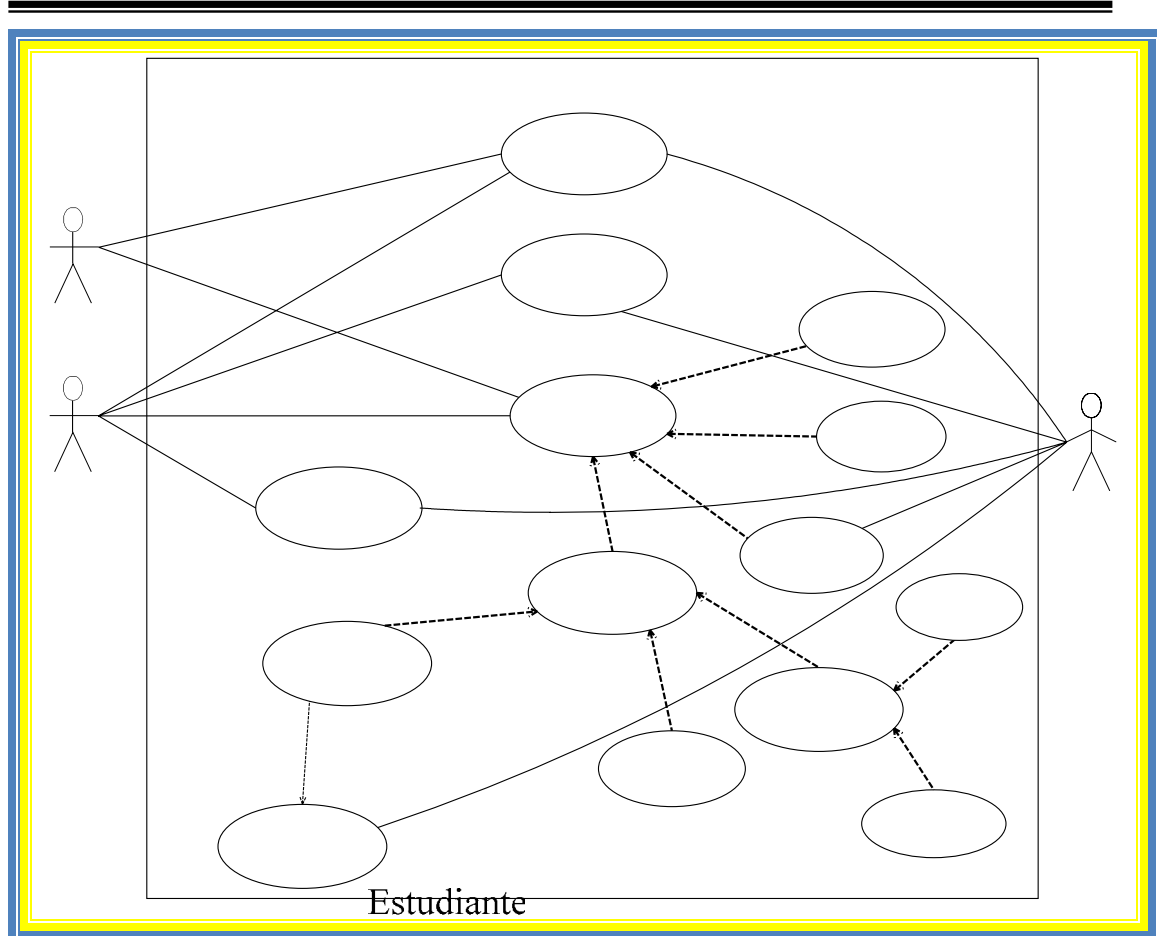


Figura 3.2 Diagrama de caso de uso del sistema SETPOO

✓ **Validar Usuario:** Es en este caso de uso donde se verifica que un usuario es válido o no, permitiéndose el acceso tanto profesores como estudiantes y estableciendo las restricciones de acuerdo al tipo de usuario que haya ingresado.

Professor

✓ **Procesar cuenta de usuario:** Es aquí donde se agregan a los nuevos estudiantes al sistema, siendo esto ejecutado por el profesor.

Procesar registro
de Estudiante

✓ **Gestor de navegabilidad:** Aquí se brinda al usuario una interfaz amigable, donde se le permita navegar por el sistema de acuerdo a sus necesidades y permisologías. <<extends>>

Evaluación

<<include>>

✓ **Actividad didáctica:** En este caso de uso el estudiante tendrá acceso a la actividad didáctica del sistema, donde podrá en práctica los conocimientos aprendidos en el transcurso de todos los capítulos.

✓ **Consultar ayuda:** Los actores utilizan el caso de uso Consultar ayuda para acceder al micromundo ayuda. El sistema desplegará los temas de ayuda en los cuales se indicarán los pasos necesarios para el manejo del software: las funciones de los distintos botones, el sistema de navegación de la aplicación, la forma como presentar la evaluación y demás acciones ejecutadas por el software. En el sistema, el micromundo ayuda siempre debe estar accesible para los actores desde cualquier escenario del software.

✓ **Actualizar perfil:** Este caso de uso le permite al estudiante actualizar su perfil, luego de que se le fue concedido una contraseña para ingresar al sistema.

✓ **Acceder capítulos:** Es aquí donde se permite al usuario poder acceder a los distintos aspectos del sistema tales como visualizar los temas, las actividades didácticas, utilizar el sonido, realizar evaluaciones, entre otras. Este caso de uso implica los aspectos más importantes del sistema pues contiene toda la parte donde el estudiante pone en marcha el proceso de enseñanza-aprendizaje.

✓ **Visualizar resumen:** Este caso de uso le da la opción al usuario de acceder al resumen del capítulo al cual esta accediendo.

✓ **Actividad Didáctica:** Le permite al estudiante realizar una actividad didáctica, que le permita verificar y recordar los conocimientos adquiridos a acerca del tema en estudio.

✓ **Evaluación:** Este caso de uso permite al estudiante realizar evaluaciones por capítulo, obteniendo una nota con la cual se verán reflejado los

conocimientos que fueron adquiridos a lo largo del capítulo y se podrán determinar los tópicos en donde fallo el estudiante.

✓ **Actualizar record de estudiante:** Luego de cada evaluación se actualizará el record del estudiante con el objeto de obtener un historial de notas.

✓ **Visualizar temas:** Cada capítulo está compuesto de temas, este caso de uso se encarga de la exploración de cada uno de ellos y las distintas actividades que se pueden llevar a cabo en ellos.

✓ **Animación:** Aquí el usuario tendrá acceso a las animaciones que servirán como ejemplo de capítulos y le permitirán al estudiante visualizar de manera grafica lo que se explica en el contenido.

✓ **Mostrar información:** El actor estudiante utiliza este caso de uso para interactuar con el tema seleccionada del capítulo activo. El sistema presenta al usuario la interfaz del tema que cuenta con una explicación teórica confiable, que incluyen textos, imágenes, animaciones y las actividades de aprendizajes.

Es importante resaltar que todos los escenarios para la exposición de los temas de cualquiera de los capítulos desarrollados presentan un patrón estándar de funcionamiento.

✓ **Mostrar ejercicios:** Es aquí donde el usuario accede a los ejercicios que se proponen en el sistema, siendo estos de mayor complejidad, con el fin de que el estudiante pueda poner en práctica los conocimientos adquiridos.

✓ **Procesar registro de estudiantes:** Este caso de uso agrupa una cantidad de aspectos del estudiante al cual el profesor puede acceder y hacer uso de esta

información si le es necesaria. Es posible acceder al record de los estudiantes, a los datos generales y a los resultados de las actividades didácticas.

3.7 FLUJO DE TRABAJO ANÁLISIS

Durante este flujo de trabajo, se analizan las necesidades obtenidas en la captura de requisitos con el objetivo de estructurar los requerimientos de manera que facilite su comprensión, su preparación, su modificación, en general, su mantenimiento; utilizando un lenguaje más formal para apuntar detalles relativos a los requisitos del sistema y de esta manera refinarlos. El propósito fundamental del análisis es resolver los casos de usos analizando los requerimientos con mayor profundidad.

3.7.1 Diagrama De Actividades

Un diagrama de actividad demuestra la serie de actividades que deben ser realizadas en un caso de uso, así como las distintas rutas que pueden irse desencadenando en el caso de uso.

Es importante recalcar que aunque un diagrama de actividad es muy similar en definición a un diagrama de flujo (típicamente asociado en el diseño de Software), estos no son lo mismo. Un diagrama de actividad es utilizado en conjunción de un diagrama de caso de uso para auxiliar a los miembros del equipo de desarrollo a entender como es utilizado el sistema y cómo reacciona en determinados eventos.

3.7.1.1 Diagrama para el caso de uso “Gestor de Navegabilidad”

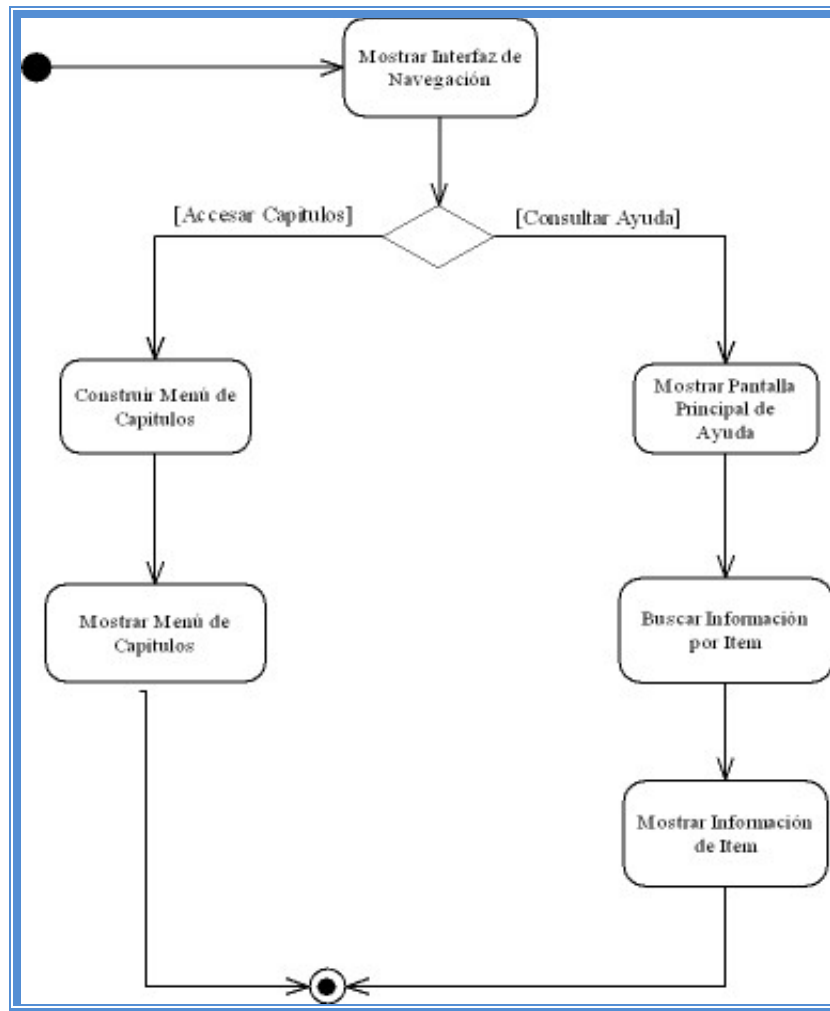


Figura 3.3 Diagrama de actividades del caso de uso: Gestor de navegabilidad

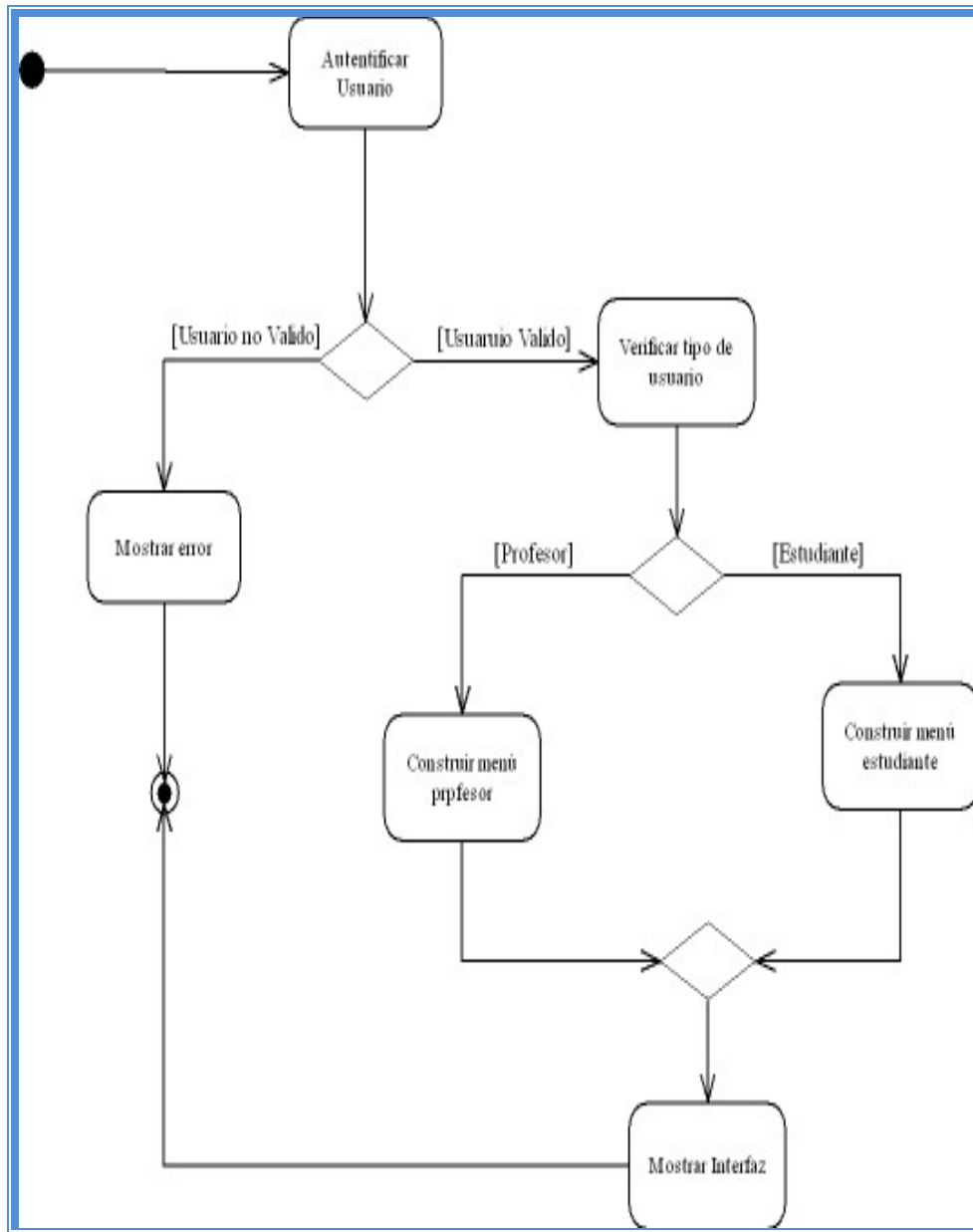
3.7.1.2 Diagrama para el caso de uso “Validar Usuario”

Figura 3.4 Diagrama de actividades del caso de uso: Validar usuario.

3.7.1.3 Diagrama de actividad para el caso de uso “Procesar Registro”

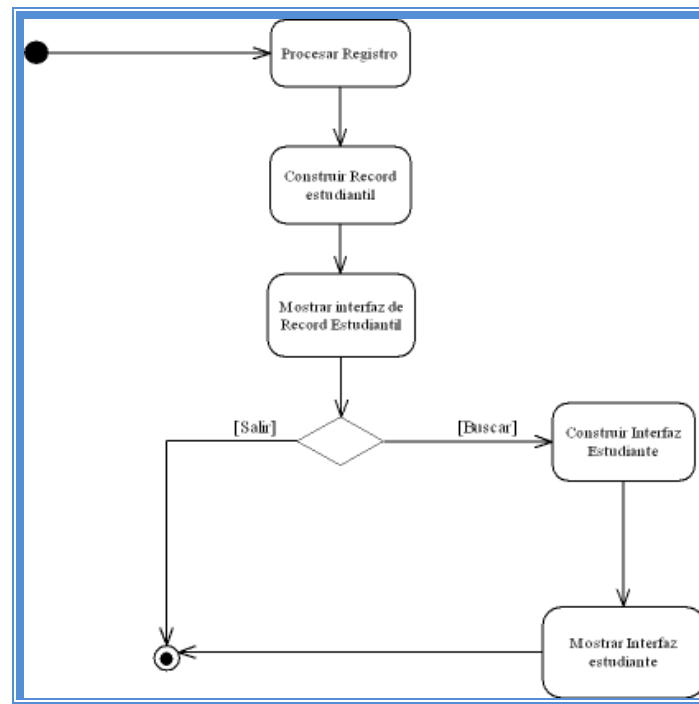


Figura 3.5 Diagrama de actividades del caso de uso: Procesar registró

3.7.2 Diagrama de Paquete

Un paquete es una agrupación de elementos del modelo que permite describir grupos de: clase, caso de uso, diagrama de colaboración o subsistemas, para organizar el modelo de análisis en piezas más pequeñas y manejables.

Un paquete se muestra gráficamente como una carpeta con etiqueta. Los paquetes subordinados se incluyen en el interior. El nombre del paquete se encuentra dentro de la etiqueta si el paquete describe sus elementos.

Una identificación inicial de los paquetes del análisis se hace de manera natural, basándonos en los requisitos funcionales y en el dominio del problema.

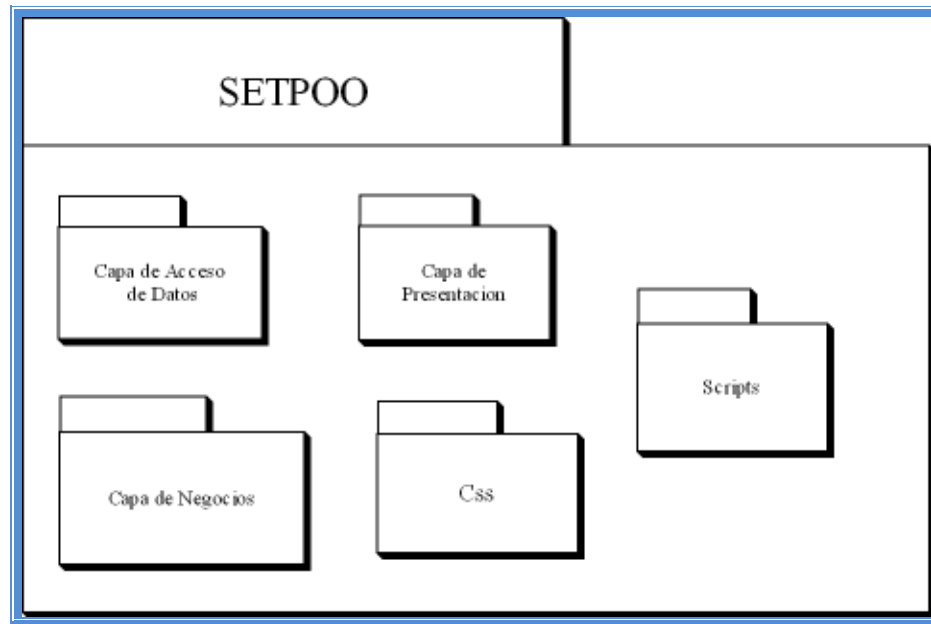


Figura 3.6 Diagrama de paquetes del sistema SETPOO

Paquetes del Sistema SETPOO

Capa de acceso de datos: En esta capa es la encargada de realizar la conexión con la base de datos del sistema.

Capa de presentación: Esta capa maneja las distintas interfaces donde el usuario puede interactuar con el sistema.

Capa de Negocios: En esta capa están contenidos todos los códigos de las distintas clases que conforman el sistema SETPOO

CSS: Este paquete contiene los archivos pertenecientes a los distintos estilos que conforman el sistema.

Scripts: El paquete de archivos scripts donde se validan todas las entradas al sistema.

3.6 CONCLUSION DE LA FASE DE INICIO

En esta fase de inicio se analizó el sistema actual para capturar los objetos más importantes dentro del contexto de SETPOO, se determinó cuáles serán los requerimientos funcionales del sistema, Además se construyó el modelo de casos de uso para la captura de los mismos, con la identificación de actores, los casos de uso y las relaciones que existen entre estos.

De acuerdo con la metodología de Ingeniería de Software Educativo de Álvaro Galvis, se dio el primer paso para el desarrollo de un material educativo computarizado (MEC). Se identificaron las deficiencias o debilidades del sistema educativo existente y sus posibles causas, para ello se usaron una serie de documentos, tales como entrevistas, encuestas y registros académicos, que arrojaron importantes conclusiones para determinar los objetivos que se han de cumplir con el desarrollo de la aplicación.

Igualmente se realizó un análisis educativo el cual permitió definir los requerimientos desde el punto de vista educativo: la población destino, el entorno de la aplicación y las limitaciones de la misma.

CAPITULO 4

FASE DE ELABORACIÓN

4.1 INTRODUCCION

Durante la fase de inicio, se realizó un análisis de las necesidades educativas de la asignatura y sus posibles soluciones. Una vez seleccionado el desarrollo de un software como solución, se debe establecer la información que debe presentar esta herramienta, los objetivos específicos a lograr por el uso de la aplicación igualmente desarrollar las estrategias requeridas para la transmisión de del conocimiento esto siguiendo la metodología para la creación de micromundos interactivos de Álvaro Galvis.

Se definen las diferentes actividades de aprendizajes y medios que se utilizarán para guiar al usuario a que se apropie del conocimiento de la unidad curricular Taller de Programación Orientada a Objetos. En esta etapa también se diseña el sistema de motivación y sistema de evaluación que sirve de base para comprobar el logro de los objetivos por parte de los aprendices.

Durante esta fase se captura los requisitos restantes para poder estimar el esfuerzo requerido en el desarrollo del proyecto. Se establece una base del diseño y las características de la arquitectura del sistema para guiar el trabajo en las siguientes fases de igual manera se continúa observando, controlando e identificando los riesgos críticos hasta que se pueda estimar su impacto en el sistema.

4.2 PLANIFICACION DE LA FASE DE ELABORACION

El objetivo principal de esta fase es alcanzar la línea base de la arquitectura, recopilando la mayoría de los requisitos que aún quedan pendientes (si los hay), estableciendo una base de la arquitectura sólida y observando si aún existen riesgos.

En esta fase los flujos de trabajo se guían a partir del modelo de casos de uso, y por el esbozo del modelo de análisis y la arquitectura candidata construidos en la fase de inicio. Al final se tiene la recopilación de todos los requisitos funcionales y la elaboración de un modelo de análisis y de diseño, lo suficientemente sólidos como para construir la arquitectura base del sistema.

En el flujo de trabajo requisitos, basado en el análisis educativo realizado en la fase anterior se realizará el diseño educativo, a través del cual se determinará el contenido educativo de la aplicación y se establecerán los aspectos relacionados a la interfaz de usuario.

Durante el flujo análisis, se analizarán los casos de uso que contribuirán con la construcción de un modelo de análisis desde el punto de vista de la arquitectura. Se especifican el modo de acceso al contenido, componentes de la aplicación y las herramientas de desarrollo a utilizar.

Al finalizar el flujo de trabajo diseño, se tendrá las clases que componen cada módulo del software y una primera versión de la interfaz de usuario.

En el flujo de implementación, podrían crearse prototipos de la interfaz de usuario, aunque este paso puede ser obviado si durante el diseño se determina que no es necesario en esta fase. El flujo de prueba, aun no será necesario en esta fase.

4.3 FLUJO DE TRABAJO REQUISITOS

El diseño de un MEC tiene el compromiso de idear y especificar una solución educativa apoyándose en el uso del computador tal que, cuando se lleve a la práctica con el tipo de usuarios a quienes se dirigen, exista la probabilidad de atender las necesidades identificadas.

El diseño de un MEC incluye tres dimensiones complementarias: la educativa, que es el corazón del MEC; la de comunicación, que hace posible una interacción eficiente entre el usuario y el programa; y la de computación propiamente dicha, que permite atender, en forma eficiente y efectiva, los requerimientos que las dos

dimensiones restantes imponen al MEC e indica cómo hacer en el computador aquello que, a nivel de educación y comunicación se requiere para atender la necesidad educativa detectada [19].

4.3.1 Requisitos Modelados como casos de Uso

Durante la reestructuración del modelo de casos de uso se agregó una nueva función disponible para el profesor obteniendo un nuevo caso de uso Gestionar Evaluación (ver figura 4.7), donde se aprecia la aparición del un nuevo caso de uso.

- ✓ Gestionar evaluación: Caso de uso utilizado por el profesor el cual le facilitara al mismo la agregación de nuevas preguntas para las evaluaciones.

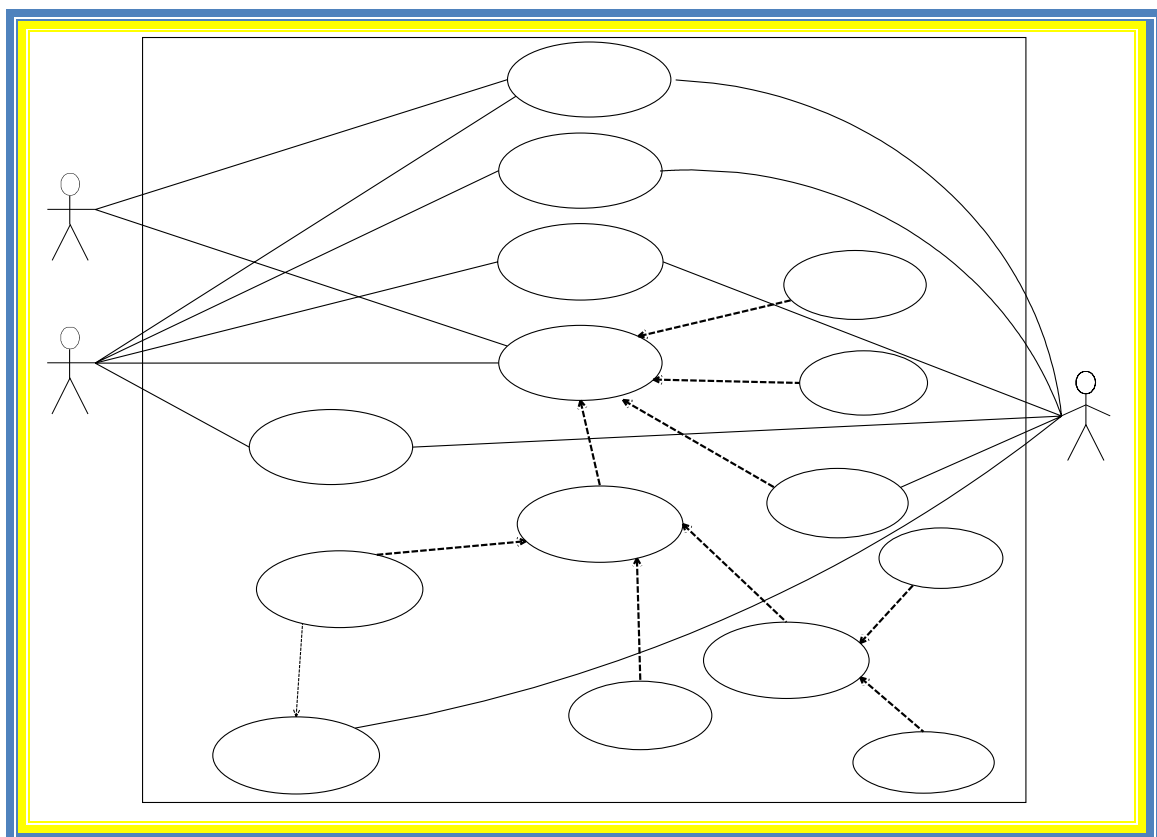


Figura 4.1 Diagrama de Casos de uso del sistema SETPOO

4.4 FLUJO DE TRABAJO ANALISIS

En la fase de inicio se desarrolló el análisis de la arquitectura sólo hasta el punto de determinar que había una arquitectura del sistema factible. Ahora en esta fase se extiende el análisis hasta el punto de que pueda servir como base a la línea principal de la arquitectura ejecutable. Para ello se realiza un ajuste al usuario profesor, ya que se le ha agregado una nueva funcionalidad a través del caso de uso gestionar evaluación. Además, se hace el estudio de los paquetes del análisis de todos los casos de uso detallados.

4.4.1 Diagrama de Actividades

Un diagrama de actividades representa los flujos de trabajo paso a paso de negocio y operacionales de los componentes en un sistema

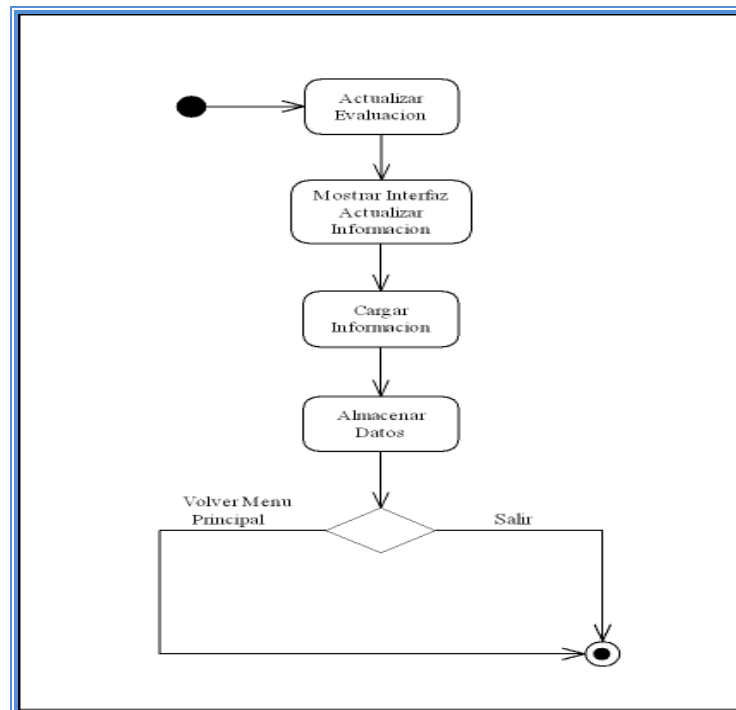


Figura 4.2 Diagrama de actividad del nuevo caso de Actualizar Evaluación

4.5 FLUJO DE TRABAJO DISEÑO

Durante el flujo de trabajo diseño se modela el sistema y se encuentra su forma, para que soporte todos los requisitos (incluyendo los requisitos no funcionales y otras restricciones), centrando la vista en casos de uso estructuralmente más representativos.

4.5.1 Diseño Educativo de un MEC.

Tomando como punto de partida la necesidad o problema educativo y las características de la población objetivo, tal como lo propone Álvaro Galvis en su Metodología para el desarrollo de Micromundos Interactivos, el objetivo del diseño educativo es estructurar la información que se mostrará al estudiante, con la finalidad de favorecer el aprendizaje de la asignatura. Para lograr esto, es importante especificar la conducta de entrada del aprendiz y el objetivo terminal o general del software, con la finalidad detallar las unidades del contenido que serán desarrollados.

Conducta de Entrada

Se denomina de esta manera al o a los comportamientos que se presupone que ya es capaz de demostrar el aprendiz cuando inicia el uso del material [19]. Esta conducta se especifica enunciando los objetivos de aprendizaje que se presuponen dominados y son relevantes para el logro del objetivo terminal.

La conducta previa que requiere el aprendiz para el uso del material educativo SETPOO es la siguiente:

Al iniciar el uso de la aplicación el estudiante debe tener capacidad del manejo de un computador, noción de los conceptos básicos a cerca de los paradigmas de la programación y conocimiento previo obtenido por medio de

asignaturas anteriormente cursadas según el pensum actual de la carrera de Ingeniería en Computación.

Objetivo General del Software.

Es un enunciado que indica lo que será capaz de hacer el aprendiz al finalizar el estudiar del MEC.

Como punto de partida para la definición del contenido, se planteo el siguiente objetivo general:

Servir de apoyo en el proceso de enseñanza y aprendizaje de los fundamentos básicos de la asignatura Taller de Programación Orientada a Objetos.

A partir de la conducta de entrada y el objetivo general del MEC, se determina lo que hay que enseñar o reforzar, descomponiendo el objetivo general en todas las tareas de aprendizaje subyacente a través de un análisis al contenido a desarrollar en el material educativo.

En el material educativo SETPOO, se distribuye el contenido del software en nueve (9) unidades como se muestra a continuación en la tabla 4.1:

Herramienta Educativa Computarizada
<p>Objetivo General:</p> <p><i>Servir de apoyo en el proceso de enseñanza y aprendizaje de los fundamentos básicos de la asignatura Taller de programación Orientada a Objetos.</i></p>
<p>Conducta de Entrada:</p> <p><i>Al iniciar el uso de la aplicación el estudiante debe tener capacidad del manejo de un computador, noción de los conceptos básicos a cerca de los paradigmas de la programación y conocimiento previo obtenido por medio de asignaturas anteriormente cursadas según el pensum actual de la carrera de Ingeniería en Computación.</i></p>
<p>Contenido:</p> <p>Unidad I: Utilización de Entorno de Desarrollo</p>

Unidad II: Datos Simples y Sentencias
Unidad III: Entrada y Salida Estándar
Unidad IV: Estructura de Selección y Control
Unidad V: Datos Compuestos (Arreglos)
Unidad VI: Apuntadores
Unidad VII: Datos Compuestos (Estructura)
Unidad VIII: Funciones
Unidad IX: Clases y Objetos

Tabla 4.1. Contenido Programático de SETPOO.

4.5.1.1 Estructura del contenido de Aprendizaje de SETPOO

La estructura del contenido de aprendizaje del software así como los objetivos asociados, se mostrara a continuación en la figura 4.3.

Tabla de Contenido Programático del sistema SETPOO 1/3

Capítulos	Objetivos Generales	Lecciones
I- Utilización de Entorno de Desarrollo	Al finalizar el capítulo el estudiante, estará en la capacidad de reconocer el entorno de programación del lenguaje C++, así como recordar la estructura básica de un programa.	1.1 Introducción a Turbo C++ 1.2 Turbo C++ como lenguaje orientado a objetos
II- Datos Simples y sentencias	Al concluir el capítulo, el estudiante será capaz de definir un dato de tipo entero, un dato de tipo flotante y clasificar los tipos asociados a estos, de igual manera reconocer la sintaxis para la declaración y asignación de	2.1 Datos enteros 2.2 Datos flotantes 2.3 Tipos de Sentencia 2.3.1 Declaraciones de datos 2.3.2 Asignación 2.3.3 Selección 2.3.4 Control

FASE DE CONSTRUCCION

	sentencias.	2.3.5 Sentencia Nula 2.4 Identificadores en C y reglas para su construcción
III- Entrada y Salida Estándar	Al concluir el capítulo el estudiante, estará en la capacidad de reconocer las funciones de entrada y salida estándar, definir y enumerar los operadores y recordar el orden de prioridad de los mismos.	3.1 Sentencias de entrada 3.2 Sentencias de salida 3.3 formatos de entrada y salida 3.4 Operadores 3.4.1 Operadores Aritméticos 3.4.2 Operadores lógicos 3.4.3 Operadores relacionales 3.4.4 Operadores de asignación 3.4.5 Operadores de autoincremento y autodecremento 3.5 Prioridades

Tabla 4.2 Planificación general de SETPOO

Tabla de Contenido Programático del sistema SETPOO 2/3

Capítulos	Objetivos Generales	Lecciones
IV- Estructura de Selección y Control	Al finalizar el capítulo el alumno, estará capacitado para definir y reconocer las sentencias de control y repetición utilizadas en C++	4.1 Estructura de selección 4.1.1 Estructura if, else 4.1.2 Anidamiento de if 4.2.3 switch 4.2 Estructura de Control 4.2.1 Sentencia while 4.2.2 Sentencia do while 4.2.3 Sentencia for
V- Datos Compuestos	Al concluir el capítulo el alumno será capaz de definir el concepto de arreglo, así como	5.1 Arreglos 5.1.1 Concepto, declaración, inicialización

FASE DE CONSTRUCCION

(Arreglos)	reconocer las diferencias entre arreglos unidimensionales y bidimensionales	5.1.2 Recorrido 5.1.3 Ordenamiento 5.2 Arreglos Multidimensionales (Matrices) 5.2.1 Conceptos, declaración Inicialización 5.2.2 Recorrido 5.2.3 Ordenamiento
VI- Apuntadores	Al finalizar el capítulo el estudiante estará capacitado para definir un apuntador y reconocer las diferencias entre acceder a la dirección y a su contenido	6.1 Definición 6.2 operadores de &,* 6.3 Arreglos y apuntadores 6.3.1 Recorridos de arreglos 6.3.2 Apuntadores a matrices
VII- Datos Compuestos (Estructura)	Al concluir el capítulo el alumno podrá definir el concepto de estructura y visualizar arreglos con estas.	7.1 Concepto y declaración 7.2 Estructuras anidadas 7.3 Arreglos y estructura 7.4 Estructura como parámetros

Tabla 4.2 Planificación General de SETPOO

Tabla de Contenido Programático del sistema SETPOO 3/3

Capítulos	Objetivos Generales	Lecciones
VIII- Funciones	Al finalizar el capítulo el estudiante logrará definir y reconocer funciones en C++ así como diferenciar una variable global de una local.	8.1 Conceptos 8.2 Funciones pre-existentes 8.3 Funciones definidas por el usuario 8.4 Funciones de cadena 8.5 Creación de función 8.6 Parámetros 8.7 Funciones con retorno 8.8 Ámbito de variable

FASE DE CONSTRUCCION

		8.8.1 Globales 8.8.2 Locales
IX- Clases y Objetos	Al finalizar el capítulo, el alumno estará en capacidad de reconocer los conceptos básicos de la programación orientada a objetos	9.1 Clases 9.1.1 Definición y atributos 9.1.2 Componentes de una clase 9.2 Constructores 9.3 Objetos 9.3.1 Definición y atributos 9.3.2 Características de un Objeto

Tabla 4.2 Planificación General de SETPOO

4.5.1.2 Objetivos Específicos del Sistema SETPOO

A continuación se presentan los objetivos específicos del sistema SETPOO

Unidad I: Utilización de Entorno de Desarrollo	
Tema	Objetivo
1.1 Introducción a Turbo C++	Conocer el entorno de desarrollo del lenguaje C++
1.2 Turbo C++ como lenguaje orientado a objetos	Reconocer el lenguaje turbo C++ como lenguaje de programación orientada a objeto
Estructura de un programa en C	Reconocer y recordar la estructura para la construcción de programas en C++

Tabla 4.3 Objetivos específicos del capítulo 1: Utilización de entorno de desarrollo

Unidad II: Datos Simples y Sentencias	
Tema	Objetivo
2.1 Datos enteros	Definir y reconocer dato de tipo entero,

FASE DE CONSTRUCCION

	y sus diferentes usos dentro de un programa.
2.2 Datos Flotantes	Definir y reconocer dato de tipo flotante, y sus diferentes usos dentro de un programa.
2.3 Tipos de Sentencia	Definir y reconocer los diferentes tipos de sentencias manejadas por C++
2.4 Identificadores en C y reglas para su construcción	Identificar y recordar las reglas básicas para la construcción de identificadores en C++

Tabla 4.4 Objetivos específicos del capítulo 2: Datos simples

Unidad III: Entrada y Salida Estándar	
Tema	Objetivo
3.1 Sentencia de entrada	Identificar las sentencias de entrada y su uso dentro en programa C++
3.2 Sentencia de Salida	Identificar las sentencias de salida y su uso dentro en programa C++
3.3 Formatos de entrada y salida	Reconocer la lista de formatos de entrada y salida permitidas por C++
3.4 Operadores	Definir y reconocer los distintos operadores permitidos es C++
3.5 Prioridades de operadores	Clasificar las prioridades de los distintos tipos de operadores, así como conocer las nociones básicas para resolver operaciones.

Tabla 4.5 Objetivos específicos del capítulo 3: Entrada y salida estándar

Unidad IV: Estructura de Selección y Control	
Tema	Objetivo
4.1 Estructuras de selección	Definir y clasificar las sentencias de selección if, else, switch
4.2 Estructuras de control	Definir y clasificar las sentencias de

FASE DE CONSTRUCCION

	control while, do while, for, break
--	-------------------------------------

Tabla 4.6 Objetivos específicos del capítulo 4: Estructuras de selección y control

Unidad V: Datos Compuesto (Arreglos)	
Tema	Objetivo
5.1 Arreglos	Definir un arreglo y reconocer el uso de estos en el entorno de desarrollo de programas
5.2 Arreglos Multidimensionales	Definir matrices, reconocer las diferencias de estas con arreglos simples así como su uso en el entorno de desarrollo de un programa

Tabla 4.7 Objetivos específicos del capítulo 5: Datos compuestos (Arreglos)

Unidad VI: Apuntadores	
Tema	Objetivo
6.1 Definición	Definir el concepto de apuntadores en C++.
6.2 Operadores *,&	Definir, diferenciar y reconocer el operador * y el operador &
6.3 Arreglos y apuntadores	Definir arreglos utilizando apuntadores

Tabla 4.8 Objetivos específicos del capítulo 6: Apuntadores

Unidad VII: Datos Compuestos (Estructura)	
Tema	Objetivo
7.1 Concepto y declaración	Definir el concepto de Estructura como datos compuestos
7.2 Estructuras anidadas	Definir y reconocer estructuras anidadas
7.3 Arreglos y estructura	Definir estructura utilizando arreglos

FASE DE CONSTRUCCION

7.4 Estructura como parámetros	Reconocer parámetros dentro del entorno de un programa
--------------------------------	--

Tabla 4.9 Objetivos específicos del capítulo 7: Datos Compuesto (Estructuras)

Unidad VIII: Funciones	
Tema	Objetivos
8.1 Concepto	Definir lo que es una función en C++
8.2 Funciones pre-existentes	Reconocer funciones pre-definidas en C++, tales como printf, scanf, etc
8.3 Funciones definidas por el usuario	Reconocer como funciones como método de resolución de problemas.
8.4 Funciones de cadenas	Reconocer funciones de cadenas pre-existentes en C++
8.5 Creación de funciones	Reconocer las reglas básicas para la creación de una función
8.6 Parámetros	Definir e identificar parámetros dentro del entorno de una función
8.7 Ámbito de variable	Definir y reconocer variables locales y globales en C++
8.8 Arreglos y Funciones	Identificar arreglos como argumentos de funciones

Tabla 4.10 Objetivos específicos del capítulo 6: Funciones

Capítulo IX: Clases y objeto	
Tema	Objetivo
9.1 Clases	Definir y crear conceptualmente clases
9.2 Constructores	Definir y crear constructores en C++
9.3 Objetos	Definir y reconocer un objeto en C++

Tabla 4.11 objetivos específicos del capítulo 9: Clases y objetos

4.5.1.3. Planificación de Actividades

El diseño educativo va desde lo más complejo hacia lo más simple en la organización de la información. Como se aprecia, la herramienta educativa SETPOO persigue un objetivo general, para el logro de este cada unidad posee un objetivo específico; de igual forma cada tema contará con una planificación de actividades con el propósito de lograr dichos objetivos.

Para el logro de los objetivos específicos de cada lección se considera importante la intervención continua y directa del aprendiz. Las situaciones de participación del estudiante son obvias, apoyados también de textos breves e imágenes significativas, que orientan al estudiante en el avance de los temas de estudio. La información impartida es completa en la medida que se lleven a cabo las interacciones planteadas, lo cual prácticamente obliga al estudiante a ser un receptor activo de su aprendizaje.

La planificación de las actividades tiene como propósito hacer efectivo los objetivos específicos de cada unidad utilizando una red semántica que relaciona los conceptos que interesa desarrollar en la aplicación. Con base en esta red se establece el contenido que soporta el material educativo, presentando los contenidos de una forma cuidadosa, tomando en cuenta que el paso de un tema a otro no debe ser forzado y manteniendo la coherencia a lo largo del material.

A continuación se describen las actividades que se desarrollaron para la planificación de las unidades del software.

4.5.1.3.1 Organización de Ítem

La organización de ítem es realizada para identificar y realzar cada una de las lecciones que componen los capítulos y cada uno de los temas que componen a las lecciones. Se desea reflejar la división de los capítulos en lecciones y de las lecciones en temas de estudio específicos.

A continuación en la tabla 4.12 se presenta la selección de ítem para el capítulo en estudio:

<p>Capítulo 9</p> <p>Clases y objetos</p> <p><i>Lección 1: Clases</i></p> <ul style="list-style-type: none">✓ <i>¿Qué es una clase?</i>✓ <i>Atributos de una clase</i>✓ <i>Estructura de una clase</i>✓ <i>Ejemplo de clases</i> <p><i>Lección 2: Constructores</i></p> <ul style="list-style-type: none">✓ <i>¿Qué es un Constructor?</i>✓ <i>Ejemplo de constructores</i> <p><i>Lección 3: Objetos</i></p> <ul style="list-style-type: none">✓ <i>¿Qué es un objeto?</i>✓ <i>Características de un objeto</i>✓ <i>Ejemplo de objetos</i>
--

Tabla 4.12 Organización de ítem del capítulo 9 Clases y Objetos

4.5.1.3.2 Resumen del Contenido de la Información

La información a ser presentada según la organización de las tablas anteriores, será resumida y expresada de manera precisa para evitar que el estudio del contenido se haga tedioso para el estudiante.

A manera de ejemplo, se presenta en la tabla 4.13 parte del resumen para el tema 9.3 Objeto, de la unidad VII: Clases y Objetos.

<i>Tema IX: Clases y Objetos</i> <i>Lección 7.3.2: Características de un Objeto</i>
<p>Un objeto se caracteriza por varios conceptos:</p> <ul style="list-style-type: none">➤ <i>Atributos:</i> estos son los datos que caracterizan al objeto. Son variables que almacenan datos relacionados al estado de un objeto.➤ <i>Métodos:</i> Los métodos de un objeto caracterizan su comportamiento, es decir, son todas las acciones (denominadas operaciones) que el objeto puede realizar por sí mismo.➤ <i>Identidad:</i> El objeto tiene una identidad, que lo distingue de otros objetos, sin considerar su estado. Por lo general, esta identidad se crea mediante un identificador que deriva naturalmente de un problema

Tabla 4.13 Resumen del contenido del tema 9.3: Objetos

4.5.1.3.3 Selección de Ejemplos e Ilustraciones

En la presentación del contenido será necesario presentar ilustraciones que muestren en forma gráfica lo explicado teóricamente, con la finalidad de facilitar al estudiante la comprensión de lo que estudia.

Las Ilustraciones son seleccionadas cuidadosamente para dar ejemplos de formas más comprensibles y agradables, rompiendo con la monotonía del marco

teórico manteniendo de esta forma la motivación del estudiante. La ilustración se presentara de dos formas:

✓ **Estática**, ilustraciones basadas en dibujos, figuras o gráficos donde el estudiante participa como observador.

✓ **Dinámica**, ilustraciones que simulan un funcionamiento a través de imágenes animadas y propician la participación del estudiante.

Ilustraciones estáticas

Las ilustraciones estáticas dentro del sistema SETPOO modelaran ejemplos ilustrados de la información contenida, la figura 4.3 muestra un ejemplo estático empleado para identificar atributos y métodos de un objeto, del capítulo 9: Clases y Objetos, la figura muestra un objeto bicicleta y una tabla donde están contenidos atributos y otra métodos, este ejemplo muestra los métodos y atributos correspondientes al objeto en cuestión.

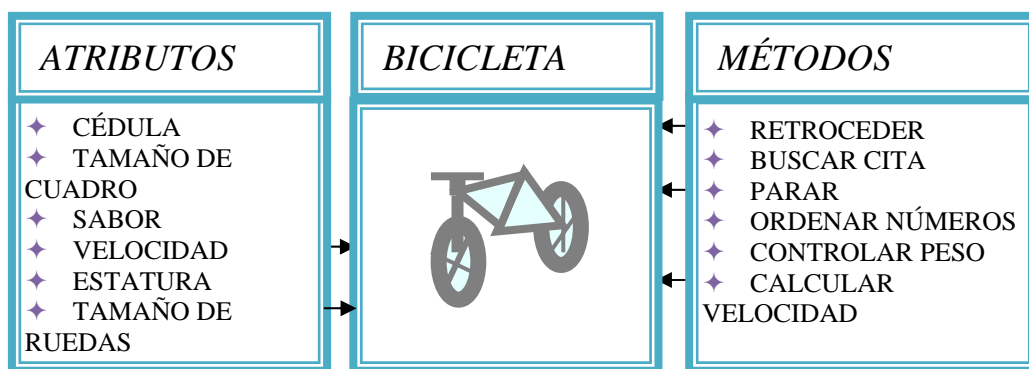


Figura 4.3 Ejercicio propuesto de la lección 9.3: Objeto del capítulo 9.

Ilustraciones Dinámicas

En el caso de las ilustraciones dinámicas el sistema SETPOO mostrara ejemplos a través de animaciones, a continuación la figura 4.4 muestra el inicio de la animación desarrollada para el capítulo 9: Clases y objetos. La cual consta de un rompecabezas el cual será armado por el usuario, al principio se mostrara de fondo la imagen de un carro y las piezas que componen el mismo, el usuario procederá a armar el objeto con las piezas disponibles en la pantalla, las cuales serán arrastradas por este y deberán ser colocadas de manera que encajen, en el caso de que la pieza no encaje en donde el usuario la coloco la pieza tendrá movilidad, hasta que sea situada en el lugar correcto. La figura 4.5 muestra la culminación de la actividad.

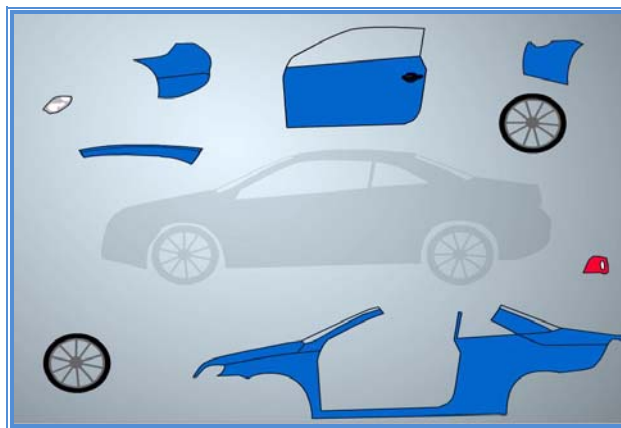


Tabla 4.4 Ejemplo de inicio de la actividad didáctica para el capítulo 9: Clases y objetos

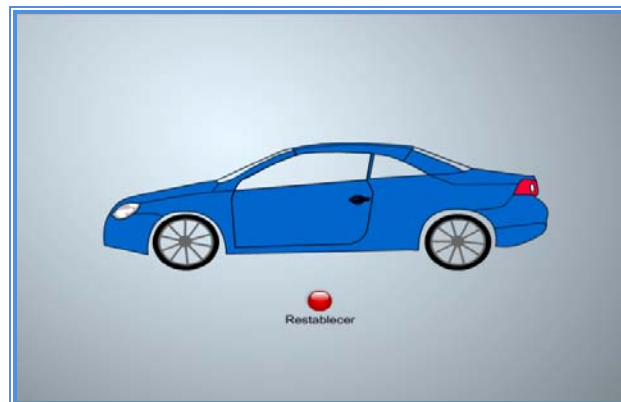


Tabla 4.5 Ejemplo finalización de la actividad didáctica para el capítulo 9: Clases y objetos

4.5.1.4 Sistema de Motivación

La presentación de SETPOO será dinámica y atractiva, desplazando imágenes coloridas y textos cortos que reflejan los temas que serán expuestos en la aplicación, invitando al usuario a interactuar con el software educativo. Una vez que el usuario acceda a aplicación encontrará una interfaz clara, sencilla y amigable, con elementos motivadores que le orienten durante la búsqueda de conocimiento.

El usuario tendrá la oportunidad de observar en la aplicación imágenes interactivas, que ilustran el contenido de aprendizaje así como animaciones explicativas de los principales conceptos expuestos en las lecciones, que requieren de la lógica y asociación de los conocimientos adquiridos por el usuario, para que así pueda entender completamente el concepto enseñado.

4.5.1.5 Sistema de Evaluación

El software educativo un tipo de evaluación que servirán de retroalimentación para el usuario. La evaluación muestra una serie de preguntas de selección múltiples, donde el usuario tiene la posibilidad de elegir de forma aleatoria la que él considere correcta; una vez finalizada la evaluación el usuario puede verificar su puntuación.

En la evaluación se respeta el tiempo de respuesta del usuario y si éste falla no será regresado al tema correspondiente, ya que él es el único responsable de decidir si debe o no reforzar sus conocimientos. Además éste podrá navegar libremente entre los diferentes capítulos y lecciones, sin tener que pasar ninguna prueba en especial (aunque se le recomienda que siga un recorrido secuencial en las lecciones), esto con el objetivo de no limitar a aquellos usuarios que tienen cierto conocimiento de la asignatura y sólo quieren estudiar algunos de los capítulos.

En el sistema SETPOO las evaluaciones de los capítulos se rigen por una serie de reglas expuestas a continuación:

FASE DE CONSTRUCCION

-
- ✓ El sistema consta de nueve (9) capítulos, cada uno de ellos con su evaluación respectiva, el estudiante podrá realizar la evaluación con el objetivo de medir sus conocimientos acerca del contenido presentado
 - ✓ El estudiante podrá realizar la evaluación del capítulo en el momento que él considere apropiado
 - ✓ No hay tiempo establecido para la culminación de la evaluación una vez activada
 - ✓ La evaluación constara de 10 preguntas, cada una de selección simple o múltiple.
 - ✓ En la pantalla aparecerá una pregunta a la vez.
 - ✓ Una vez que el estudiante seleccione su respuesta, el sistema automáticamente mostrara si su selección es correcta o incorrecta y seguirá hacia la siguiente pregunta.
 - ✓ Al finalizar las 10 preguntas el sistema le mostrara al estudiante su puntuación final.
 - ✓ Una vez presentada la evaluación, si el estudiante no quedo satisfecho con su puntuación podrá volver a presentar la misma.

A continuación la tabla 4.14 presenta un ejemplo del modelo de evaluación que presentará el sistema SETPOO.

Ejemplo de evaluación del capítulo 9: Clases y Objetos 1/3

Ejemplo de Evaluación del Capítulo 9: clases y Objetos

- 1- ¿Una clase se puede considerar como?
 - a- Un patrón para construir objetos
 - b- Una metodología para la aplicación de funciones
 - c- La estructura principal de un programa
 - d- La declaración de atributos

- 2- Es una instancia de una clase. Encapsula datos y métodos que manejan esos datos
 - a- Constructores
 - b- Clase
 - c- Objeto
 - d- Atributo

- 3- ¿Cuáles de las siguientes opciones son características de un objeto? (Elija 2)
 - a- Atributos
 - b- Identidad
 - c- Constructor
 - d- Encapsulación

- 4- Elija los atributos que pertenecen al objeto televisor. (Elija 2)
 - a- Marca
 - b- Estado del seguro
 - c- Pulgadas
 - d- sabor

Tabla 4.14 Modelo de Evaluación del sistema SETPOO

Ejemplo de evaluación del capítulo 9: Clases y Objetos 2/3

Ejemplo de Evaluación del Capítulo 9: clases y Objetos

- 5- Funciones miembros especiales que sirven para inicializar un objeto de una determinada clase al mismo tiempo que se declara
 - a- Atributo
 - b- Constructores
 - c- Métodos
 - d- Todas las anteriores

FASE DE CONSTRUCCION

<p>6- Las acciones que el objeto puede realizar por si mismo se conoce como:</p> <ul style="list-style-type: none"> a- Clase b- Atributo c- Métodos d- Constructores <p>7- Seleccione los métodos que corresponden al objeto cubo. Elija 2</p> <ul style="list-style-type: none"> a- Volumen () b- Área_superficial() c- Avanzar() d- Dimensión() <p>8- Denota las características esenciales de un objeto donde se capturan sus comportamientos</p> <ul style="list-style-type: none"> a- Abstracción b- Programación orientada a objetos c- Encapsulamiento d- Métodos
--

Tabla 4.14 Modelo de Evaluación del sistema SETPOO

Ejemplo de evaluación del capítulo 9: Clases y Objetos 3/3

<h3>Ejemplo de Evaluación del Capítulo 9: clases y Objetos</h3>
<p>9- La programación agrega nuevas ideas al mundo de la programación orientada a objetos. Estas son: Elija 2</p> <ul style="list-style-type: none"> a- Reutilización de código b- Pasos de mensaje c- Programación segura d- Procesamiento de código eficiente <p>10- Los principios básicos de la programación orientada a objetos son: Elija 2</p> <ul style="list-style-type: none"> a- Herencia b- Apuntadores c- Polimorfismo d- subrutinas

Tabla 4.14 Modelo de Evaluación del sistema SETPOO

4.5.2 Diseño Comunicacional.

La zona de comunicación en la que se realiza la interacción entre usuario y programa, es de vital importancia en un software educativo orientado a facilitar el proceso de enseñanza – aprendizaje y mantener motivado al estudiante.

La estructura del contenido a tratar en el MEC, sugiere una selección de dispositivos para una interacción efectiva con los elementos del entorno. Se selecciono como dispositivo físico de entrada el teclado y el Mouse del computador, el monitor como dispositivo de entrada y salida por presentar visualmente el entorno de la aplicación y mostrar las repuestas del software a la utilización de los dispositivos de entrada por parte del usuario.

La zona de comunicación, donde se combinan los dispositivos de entrada y salida, está compuesta por:

- ✓ **Zona de trabajo**, donde el usuario tiene a disposición lo que le sirve de base para aprender. En el caso de SETPOO las zonas de trabajo ocupan la mayor parte de la pantalla y son donde aparece la información principal que proporciona el programa y se desarrollan las actividades educativas
- ✓ **Zona de control del programa**, donde el usuario puede alterar el flujo y ritmo de ejecución de la aplicación.
- ✓ **Zona de contexto para la acción**, donde el usuario identifica en que módulo de la aplicación se encuentra.

Para el material educativo SETPOO, se estableció como zona de trabajo el centro de la pantalla (ver Figura 4.6), donde serán presentados los textos e, ilustraciones que son requeridos.

Para la zona de control se selecciono dos áreas. En una de ellas se ubicará el menú principal, que dará acceso al usuario a los distintos módulos que conforma el software. En la otra zona (área de navegación) se tendrá los elementos con los que el

usuario podrá desplazarse por el contenido mostrado en la zona de trabajo (ver figura 4.6)

La zona de contexto e identificación de usuario, como se ilustra en la figura 4.6, se ubica en la parte superior de la pantalla, donde se identifica el software.

La figura 4.6 muestra una zona de identificación de usuario, la cual servirá para indicar el usuario que está logueado en el sistema.



Figura 4.6. Zona de comunicación de la interfaz de usuario SETPOO

Esta interfaz de usuario servirá de prototipo para su desarrollo durante la fase de construcción y podrá estar sujeta a cambios si así es requerido.

4.5.2.1 Elementos Constitutivos de la interfaz

La capacidad educativa y pedagógica de la interfaz SETPOO depende de cómo se presenten los aspectos formales del contenido que será objeto del proceso de

aprendizaje y de que el estudiante logre interactuar de manera armónica con ese contenido y con todo el sistema en sí.

Para cumplir este requisito es fundamental que la inclusión u omisión de los posibles elementos que componen las pantallas (textos, imágenes, tablas, fondos, etc.) responda a una intencionalidad pedagógica, con el fin de hacer más fluida la comunicación entre el estudiante y el sistema.

A continuación se describen algunos de los elementos que componen la interfaz de SETPOO.

4.5.2.1.1 Texto

Los textos contenidos en SETPOO, son presentados de forma concisa, bien estructurada y fácil de leer. Están organizados en párrafos cortos e incluyendo sólo una idea por párrafo: la brevedad para expresar una idea con exactitud contribuye a que el estudiante reconstruya el sentido del mensaje.

Los tipos de letra elegidos contrastan con el micromundo seleccionado. Se usa la fuente Olympus para los títulos, en tamaños que varían de 12 a 18, y la fuente Georgia para los textos de las lecciones en tamaño 13, para no hacer demasiado largo el texto, ni causar dificultades de lectura a las personas que no tengan una buena visión.

El color del texto es otro aspecto que determina la legibilidad. Se ha procurado un alto contraste entre el fondo de la página y el color de texto, evitando contrastes demasiado inquietantes, para no abusar de la capacidad de atención de los lectores. Los textos que representan información son de color negro, los títulos son presentados en colores amarillo, para que contrasten e identifiquen su función.

4.5.2.1.2 Fondos

Los fondos identifican cada zona de comunicación de la interfaz, contribuyendo así a facilitar la orientación del estudiante dentro del software. El sistema SETPOO está basado en la representación del mundo griego, así que los fondos y las imágenes son estructuras alusivas a este tema.

La zona de mayor importancia pedagógica, que es la de exposición de lecciones, presenta fondo beige dando impresiones de envejecimiento, color que contrasta con el color negro de los textos, cuya función es hacer más legible la información presentada.

4.5.2.2 Diseño de pantallas

Las pantallas de SETPOO están diseñadas para que sean fáciles de utilizar, directas y no estrictas, para que así el usuario pueda hacer buen uso de la información contenida en ellas.

El sistema SETPOO contiene principalmente las siguientes pantallas:

- ✓ Pantalla de inicio
- ✓ Pantalla de explorar capítulos
- ✓ Pantalla de evaluación
- ✓ Pantalla de gestión de estudiantes

4.5.2.2.1 Pantalla de Inicio

En la pantalla inicial del sistema se muestran estatuas griegas alusivas a la escenografía en la que se pretende envolver al estudiante y motivar su curiosidad por saber que habrá después, en ella se le solicita al usuario su nombre y contraseña para

poder ingresar al sistema, con el fin de poder tener acceso a la información. La representación de esta pantalla se muestra a continuación en la figura 4.7.

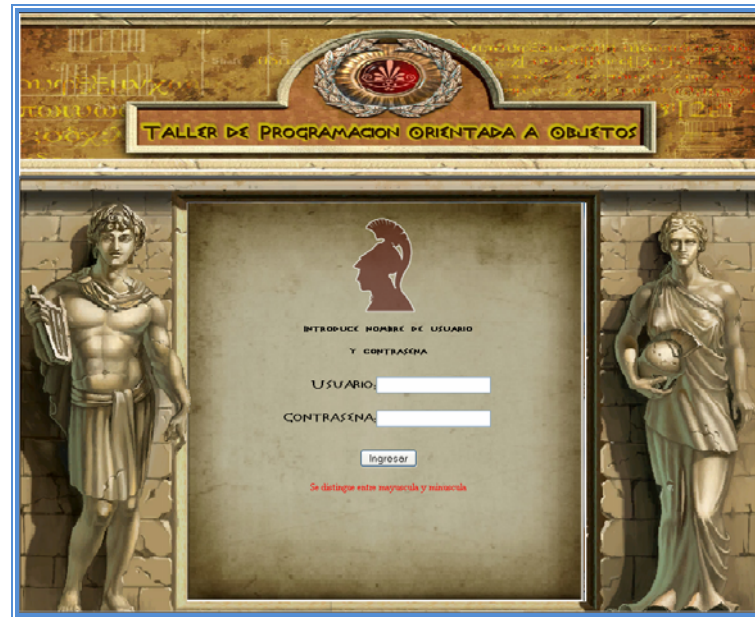


Figura 4.7 Pantalla de inicio del sistema SETPOO

4.5.2.2.2 Pantalla de explorar capítulos

La figura 4.8 muestra el diseño de la pantalla de explorar capítulos. Esta pantalla representa la interfaz principal de exposición de lecciones con todos sus elementos asociados: información teórica, animaciones y actividades de aprendizajes.

La pantalla de explorar contenidos presenta las zonas de comunicación distribuidas funcionalmente y de fácil identificación por parte del usuario. La zona de contexto está ubicada en la parte superior de la pantalla y contiene el título del capítulo que ha sido activado.

FASE DE CONSTRUCCION

La zona de control está conformada básicamente por el menú de capítulos, los botones gráficos que representan las distintas herramientas con que cuenta la aplicación (ver resumen, inicio, cerrar sesión, ayuda).

La zona de trabajo es la que ocupa el mayor espacio y está conformada por los textos de información y las animaciones interactivas.

La zona de trabajo varía su contenido según la lección que sea activada. La zona de control y de contexto es inalterable, es decir, no presenta ningún cambio a medida que se desarrollan las lecciones.



Figura 4.8 Pantalla de explorar capítulos SETPOO

4.5.2.2.3 Pantalla de Gestión de Usuario

La figura 4.9 muestra el diseño de pantalla dispuesto para el profesor, esta pantalla representa la interfaz del profesor, donde el mismo podrá agregar y gestionar a los inscritos estudiantes en el sistema.

FASE DE CONSTRUCCION

La zona de trabajo está conformada por interfaces que muestran distintas opciones seleccionadas por el profesor como: agregar estudiante, ver información de curso, buscar estudiantes y otras.

La zona de control está dispuesta del lado izquierdo de la pantalla donde se muestra el menú principal y las opciones disponibles para el profesor, por último el área de navegación donde el profesor podrá manejar su sesión.



Figura 4.9 Pantalla de Gestión de usuario para el profesor

4.5.2.2.4 Pantalla de Evaluación

La figura 4.10 muestra el diseño de la pantalla de evaluación. De acuerdo con la concepción de enseñanza y aprendizaje, la evaluación está entendida como, un momento de reflexión y análisis de la enseñanza. Partiendo de esto, esta pantalla está diseñada de una manera agradable al usuario, buscando la concentración del estudiante en este proceso.

FASE DE CONSTRUCCION

En esta pantalla se presenta la zona de contexto en la parte superior con la identificación del software y del capítulo. En la zona de trabajo se presentan las preguntas y las opciones de respuesta para ser activadas por parte del usuario. En esta zona también se presentaran los mensajes de evaluación de respuestas. La zona de control está dispuesta en la parte inferior de la pantalla y está conformada por los botones que enlazan las cada pregunta.

Por tratarse de una evaluación, el usuario no tendrá acceso al contenido, por lo que esta pantalla no presentara el menú de capítulos. Solo tendrá disponible la opción evaluar para salir de la evaluación.



Figura 4.10 Pantalla de evaluación del sistema SETPOO

4.5.3 Diagrama de clase de diseño

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de diseño de los sistemas, donde se crea el

FASE DE CONSTRUCCION

diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro.

La figura 4.11 representa de manera general el diagrama de clases del Sistema SETPOO, el cual muestra todas las clases involucradas en el desarrollo de la aplicación, las cuales representan todas las operaciones que los usuarios pueden realizar.

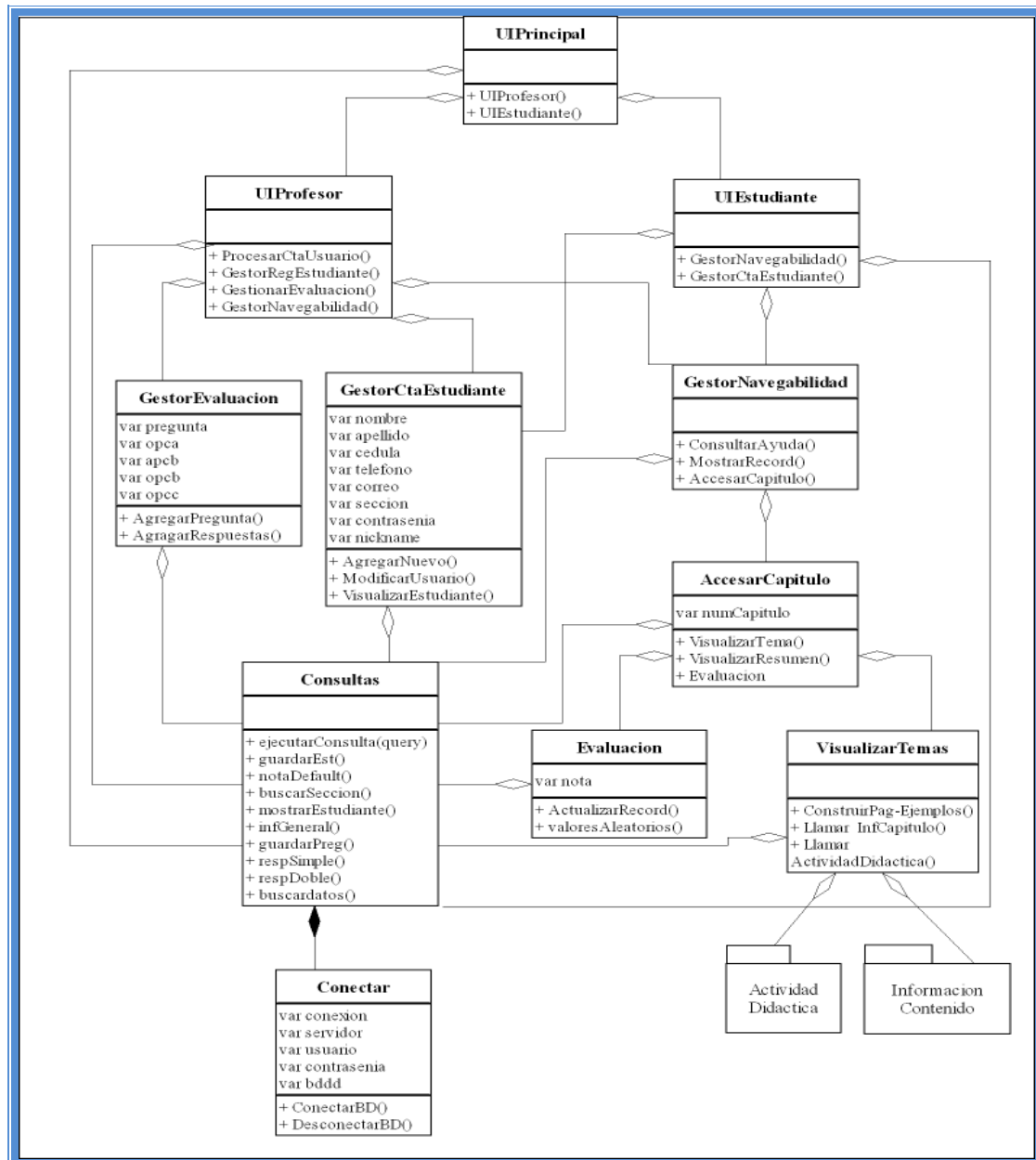


Figura 4.11 Diagrama de Clases del sistema SETPOO

4.5.4 Diagrama de Secuencia

Un diagrama de secuencia muestra las interacciones entre objetos ordenadas en secuencia temporal. Muestra los objetos que se encuentran en el escenario y la secuencia de mensajes intercambiados entre los objetos para llevar a cabo la

4.5.4.2 Diagrama de Secuencia Para el Caso de Uso Agregar Estudiante

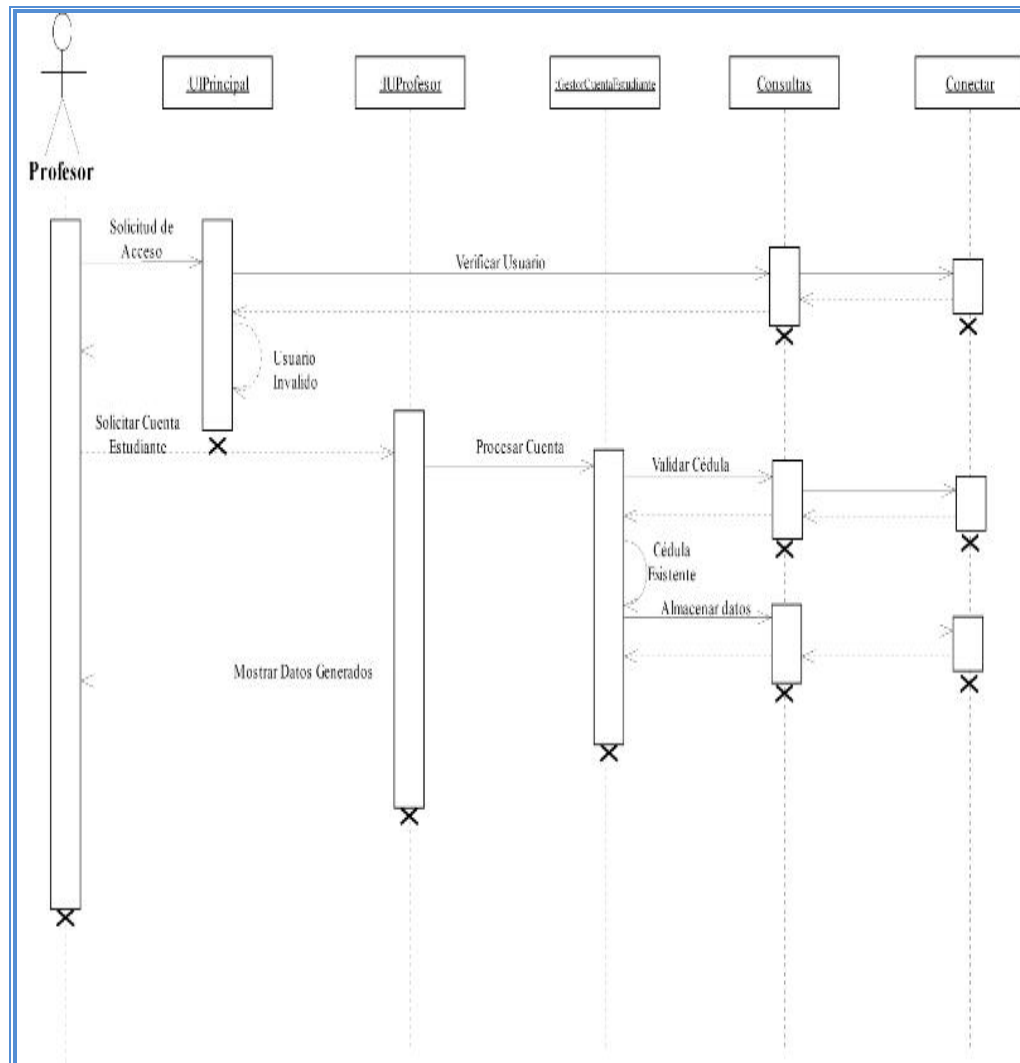


Figura 4.13 Diagrama de Secuencia Agregar Estudiante

4.5.4.3 Diagrama de Secuencia Para el Caso de uso Accesar Capitulo

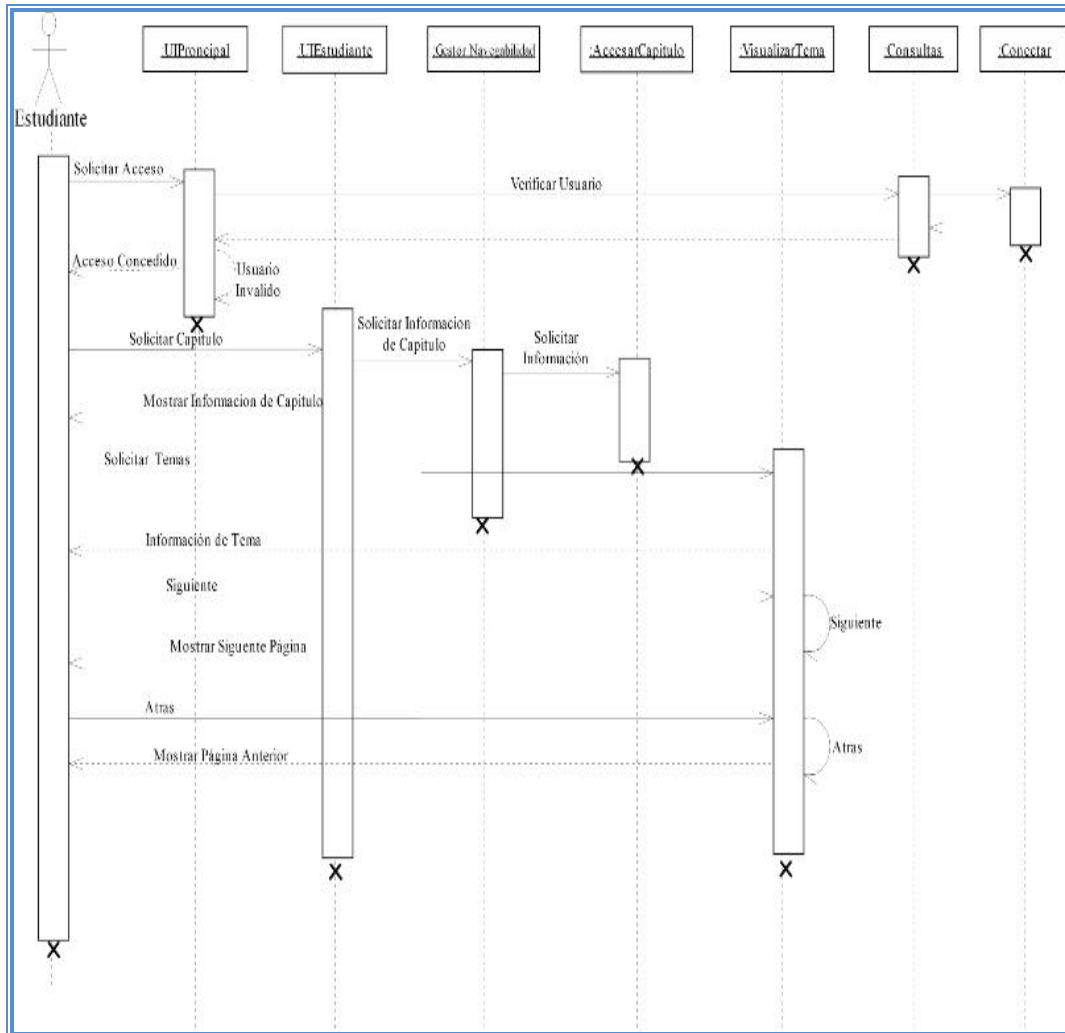


Figura 4.14 Diagrama de Secuencia Accesar capitulo

4.5.5 Diagrama de Capas

El uso de la arquitectura de capas es aplicable a muchos tipos de sistemas. Este patrón define como organizar el modelo de diseño en capas, lo cual quiere decir que los componentes de una capa solo pueden hacer referencia a componentes en capas inferiores. Este patrón simplifica la comprensión y la organización del desarrollo de

FASE DE CONSTRUCCION

sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores. Además, nos ayuda a identificar que puede reutilizarse. Un sistema con una arquitectura en capas pone a los subsistemas de aplicación individuales en lo más alto. Estos se construyen a partir de subsistemas en las capas más bajas, como son los marcos de trabajo y las bibliotecas de clases.

La capa general de aplicación contiene los subsistemas que no son específicos de una sola aplicación, sino que pueden ser reutilizados por muchas aplicaciones diferentes dentro del mismo dominio o negocio. La arquitectura de las dos capas inferiores puede establecerse sin considerar los casos de uso de que no son dependientes del negocio. La arquitectura de las dos capas superiores se crea a partir de los casos de uso significativos para la arquitectura (estas capas son dependientes del negocio).

Las capas inferiores son de aplicación general a varias aplicaciones y deben poseer interfaces más estables, mientras que las capas más altas son más dependientes de la aplicación y pueden tener interfaces menos estables. A continuación la figura 4.15 muestra el diagrama de la arquitectura del sistema SETPOO utilizando diagrama de capas.

FASE DE CONSTRUCCION

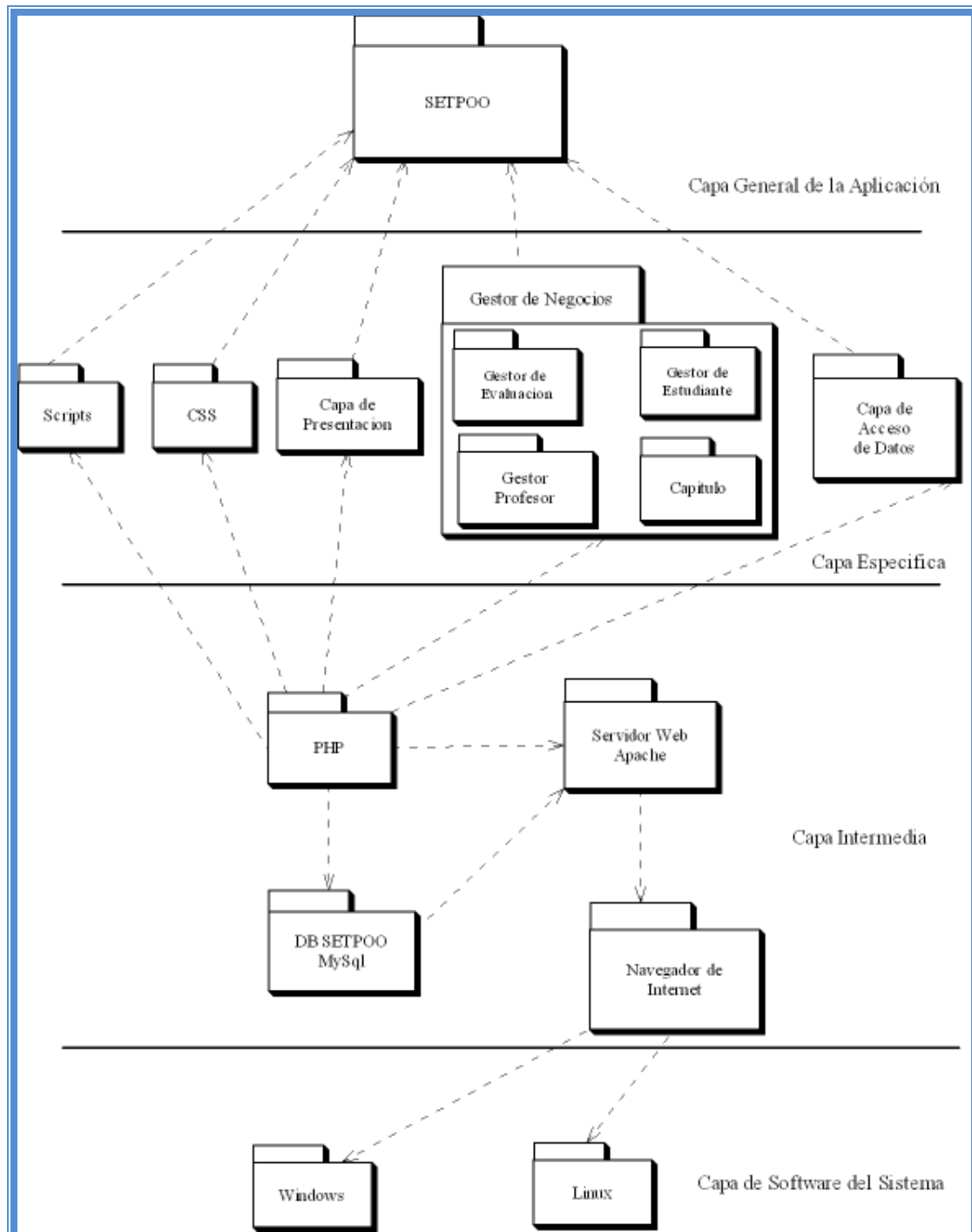


Figura 4.15 Diagrama de Capas del sistema SETPOO

4.5.6 Diagrama de Paquete de diseño

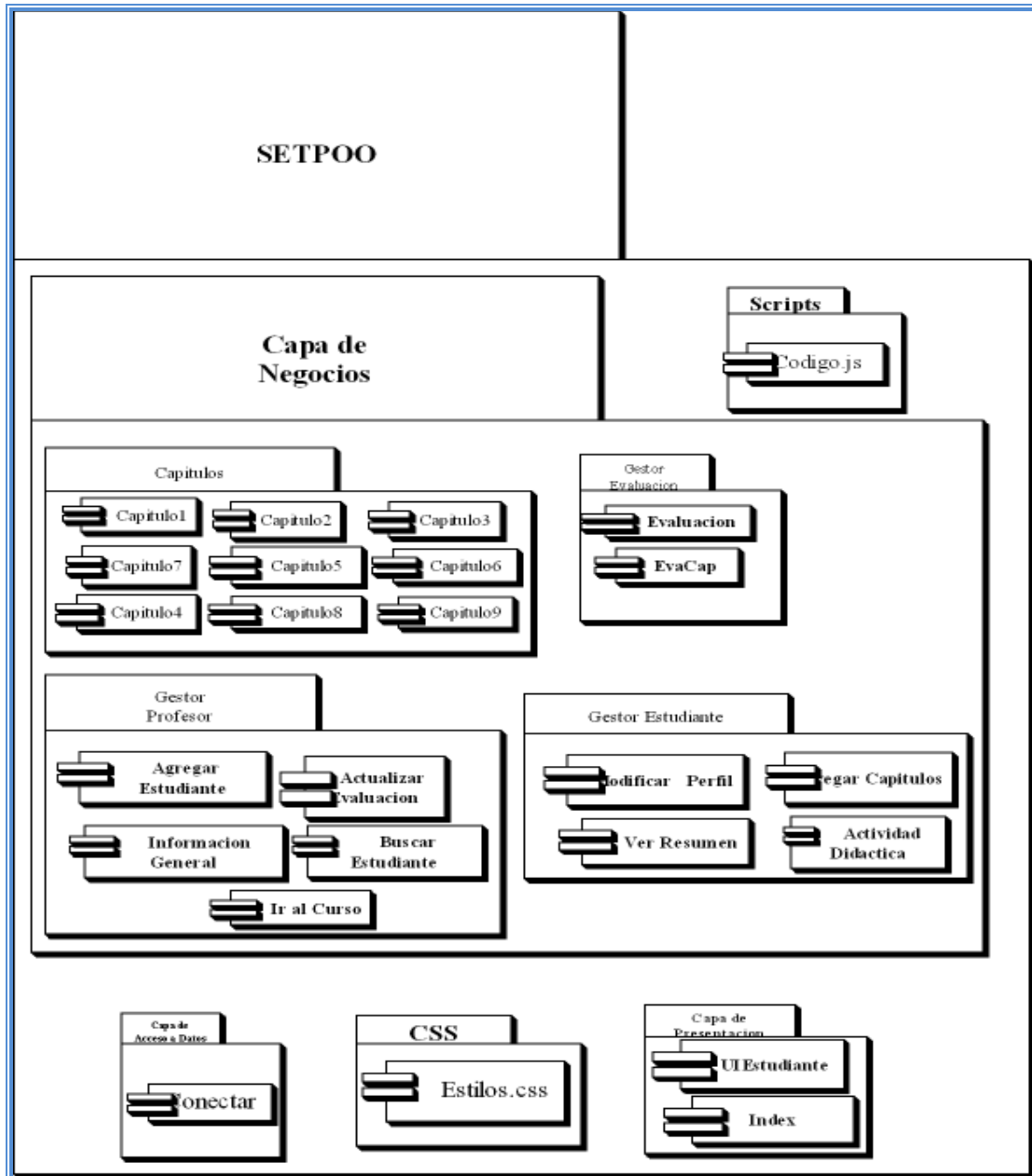


Figura 4.16 Diagrama de paquete de análisis

4.5.7 Diseño de la Base de Datos

El modelo relacional, es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. En una base de datos relacional, todos los datos se almacenan y se acceden a ellos por medio de relaciones. Las relaciones que almacenan datos son llamados "relaciones base" y su implementación es llamada "tabla". Estas bases de datos se utilizan principalmente porque ofrecen sistemas simples y eficaces para representar y manipular los datos.

En las bases de datos relacionales las relaciones representan las entidades que se consideran interesantes en la base de datos. Cada instancia de la entidad encontrará sitio en una tupla de la relación, mientras que los atributos de la relación representarán las propiedades de la entidad. A continuación se presenta el diseño de la base de dato del sistema SETPOO en la figura 4.17.

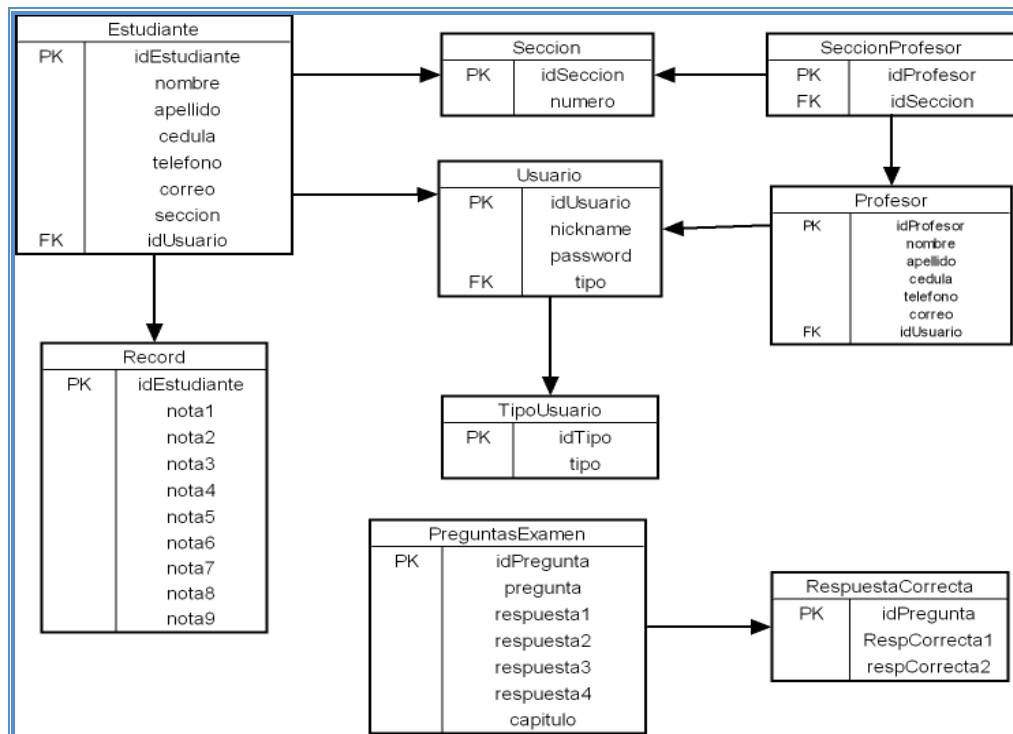


Figura 4.17 Diagrama de base de datos del sistema SETPOO

4.6 FLUJO DE TRABAJO IMPLEMENTACION

En la implementación se comenzará con el uso de los resultados del diseño y se implementará las clases pertenecientes al sistema en términos de componentes. El propósito fundamental de este flujo de trabajo es desarrollar la arquitectura y el sistema como un todo. Es en esta iteración donde se realizará el diagrama de componentes parcial de la aplicación y se implementará la interfaz inicial del estudiante, donde puede acceder a la información proporcionada por el sistema, debido a que el resto se verán completamente implementadas en la fase de Construcción.

4.6.1 Diagrama de Despliegue

Los Diagramas de Despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación.

Un nodo es un objeto físico en tiempo de ejecución que representa un recurso computacional, generalmente con memoria y capacidad de procesamiento. Un artefacto es un producto del proceso de desarrollo de software, que puede incluir los modelos del proceso (modelos de Casos de Uso, modelos de Diseño, etc.), archivos fuente, ejecutables, documentos de diseño, reportes de prueba, prototipos, manuales de usuario y más. En la figura 4.18 se muestra el diagrama de despliegue del sistema SETPOO.

FASE DE CONSTRUCCION

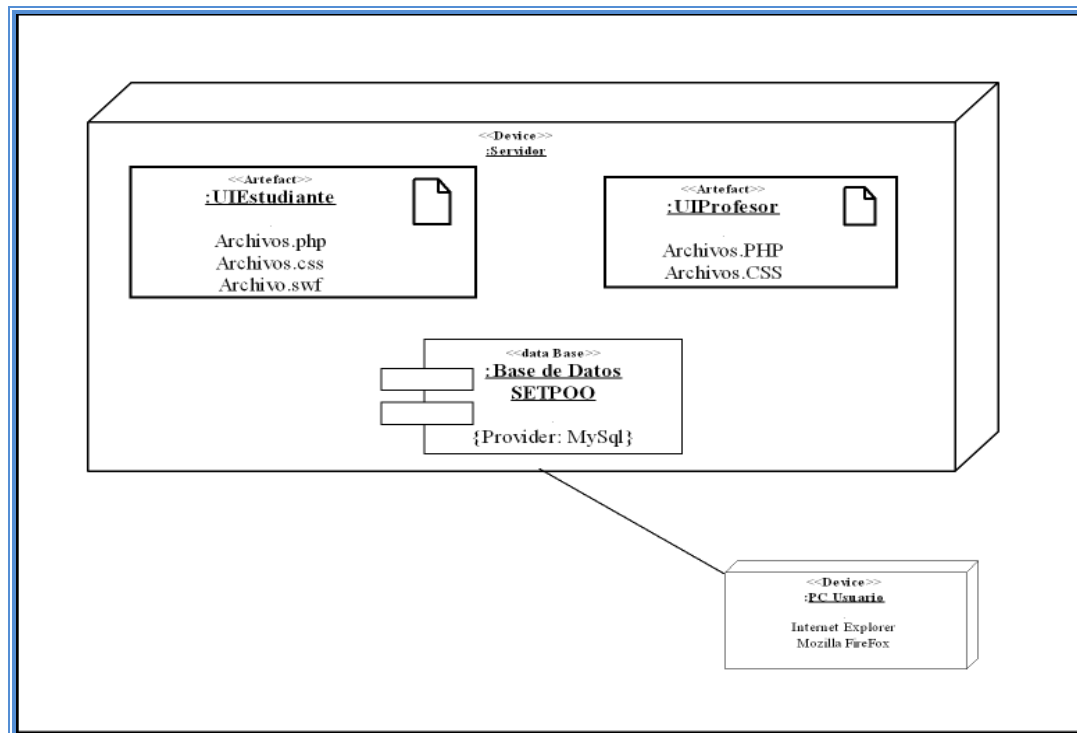


Figura 4.18 Diagrama de despliegue del sistema SETPOO

4.6.2 Diagrama de Componentes

Un diagrama de componentes representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos componentes. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. La figura 4.19 muestra el diagrama de componentes parcial del sistema SETPOO.

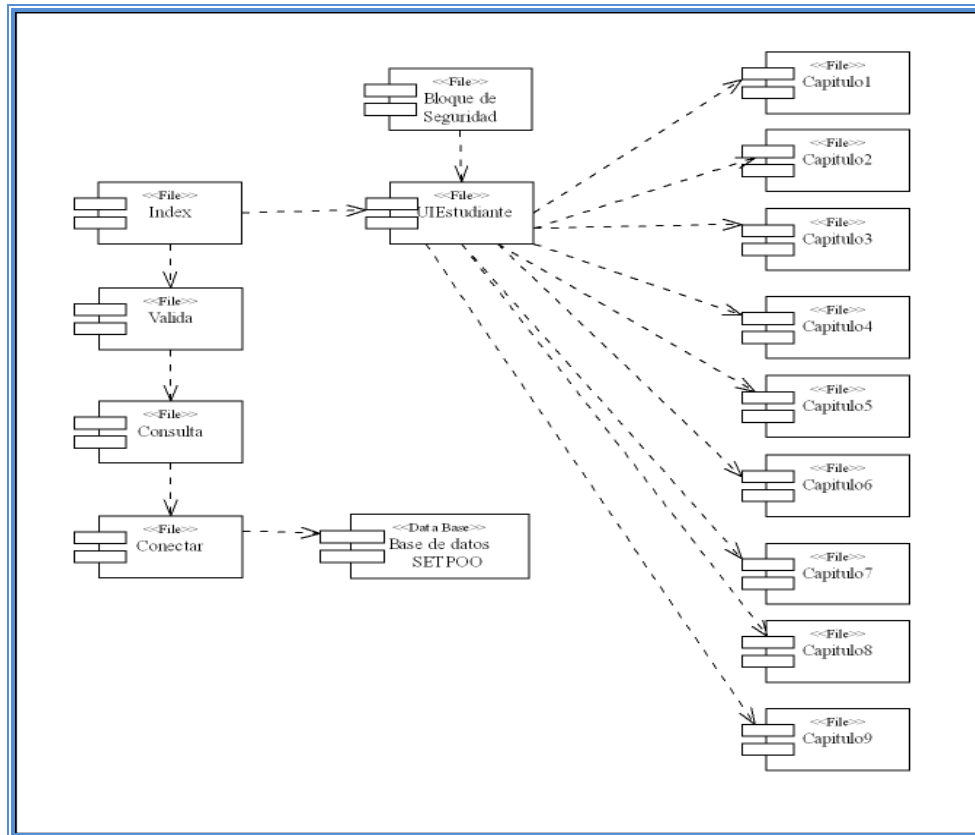


Figura 4.19 Diagrama de componentes parcial del sistema SETPOO

4.6.3 Implementación de los Componentes Asociados a la Consulta de Capítulos

En la figura 4.20 observamos la interfaz para el usuario estudiante, con esta estamos observando la ejecución de los componentes Index, IUestudiante, valida, consulta y conectar del diagrama de la figura 4.20.



Figura 4.20 Interfaz principal del estudiante sistema SETPOO

Código:

index.php

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
  <html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/Plantilla1.dwt.php"
codeOutsideHTMLIsLocked="false" --><head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"
  />
  <!-- InstanceBeginEditable name="doctitle" -->
  <title>:::SETPOO:::</title>
  <!-- InstanceEndEditable -->
  <!-- InstanceBeginEditable name="head" -->
  <LINK REL="SHORTCUT ICON" HREF="setpoo.png">
  <link href="css/Estilos.css" rel="stylesheet" type="text/css" />
  <script type="text/javascript" src="Scripts/codigo.js"></script>

```

FASE DE CONSTRUCCION

```

<!-- InstanceEndEditable -->
</head>
<body>
  <table width="944" border="1" cellspacing="0" cellpadding="0"
align="center">
  <tr>
    <td height="220" width="944" background="Imagenes/banner4.PNG"
background-repeat="no-repeat"></td>
  </tr>
  <tr>
    <td><!-- InstanceBeginEditable name="RegionEditable1" -->
      <table height="583" width="943" border="1" cellspacing="0"
cellpadding="0" background="Imagenes/login.png" class="EstiloFondo">
        <tr>
          <td align="center" valign="top">
            <p>&nbsp;</p>
            <table width="90" height="127" border="0" cellspacing="0"
cellpadding="0" background="Imagenes/3.png" class="EstiloFondo"> <tr>
              <td>&nbsp;</td>
            </tr> </table>
            <form id="form1" name="form1" method="post"
action="valida.php">
              <table width="336" height="206" border="0" cellpadding="0"
cellspacing="0" class="EstiloFondo">
                <tr>
                  <td height="56" colspan="2" align="center"
class="Estilo3"><p>Introduce nombre de usuario </p>
                  <p>y contraseña</p></td>
                </tr>
                <tr>
                  <td width="153" height="44" align="right"
class="Estilo1">Usuario:</td>
                  <td width="184"><label>
                    <input type="text" name="txtusuario" id="txtusuario" />
                  </label></td>
                </tr>
                <tr>
                  <td height="45" align="right"
class="Estilo1">Contraseña:</td>

```

FASE DE CONSTRUCCION

```

        <td><label>
            <input type="password" name="txtpass" id="txtpass" />
        </label></td>
    </tr><tr>
        <td height="61" colspan="2" align="center"><input
name="btningresar" type="submit" class="Estilo8" id="btningresar"
value="Ingresar" />
            <label></label></td>
    </tr>
</table>
</form>
<table width="300" border="0" cellspacing="0" cellpadding="0">
    <tr>
        <td align="center" class="EstiloAlerta">Se distingue entre mayuscula
y minuscula</td>
    </tr>
</table>
</td>
</tr>
</table>
<!-- InstanceEndEditable --></td>
</tr>
</table></body><!-- InstanceEnd --></html>
UIEstudiante
<?PHP
    include("BloqueDeSeguridad.php");
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
    <html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/Plantilla1.dwt.php"
codeOutsideHTMLOIsLocked="false" -->
        <head>
            <meta http-equiv="Content-Type" content="text/html; charset=utf-8"
/>
            <!-- InstanceBeginEditable name="doctitle" -->
            <title>:::Estudiante-SETPOO:::</title>
            <LINK REL="SHORTCUT ICON" HREF="setpoo.png">
            <link rel="stylesheet" type="text/css" href="css/Estilos.css"

```

FASE DE CONSTRUCCION

```

media="screen"/>
  <!-- InstanceEndEditable -->
  <!-- InstanceBeginEditable name="head" --><!-- InstanceEndEditable -->
  </head>
  <body>
    <table width="944" border="1" cellspacing="0" cellpadding="0"
align="center">
      <tr>
        <td height="220" width="944" background="Imagenes/banner4.PNG"
background-repeat="no-repeat"></td>

      </tr>
      <tr>
        <td><!-- InstanceBeginEditable name="RegionEditable1" -->
          <table width="945" height="560" border="1" cellpadding="0"
cellspacing="0">
            <td height="50" colspan="2">
              <table width="945" height="57" border="0" cellspacing="0"
cellpadding="0" background="Imagenes/menuiconos.png"
class="EstiloFondo">
                <tr>
                  <td width="397" valign="bottom" class="Estilo3">
                    <?PHP
                                $nombre= $_SESSION["nombre"];
                                $apellido= $_SESSION["apellido"];
                                echo "Bienvenido(a) $nombre $apellido";
                                ?>
                    &nbsp;&nbsp;&nbsp;</td>
                  <td width="215" valign="middle">
                    <span id="liveclock" ></span>
                    <script language="JavaScript" type="text/javascript">
                    <!--
function show5(){
if (!document.layers&&!document.all&&!document.getElementById)
return

var Digital=new Date()
var hours=Digital.getHours()

```

```

var minutes=Digital.getMinutes()
var seconds=Digital.getSeconds()

var dn="PM"
if (hours<12)
dn="AM"
if (hours>12)
hours=hours-12
if (hours==0)
hours=12

if (minutes<=9)
minutes="0"+minutes
if (seconds<=9)
seconds="0"+seconds
//change font size here to your desire
myclock="<font size='5' face='Georgia' >"+hours+": "+minutes+": "
+seconds+" "+dn+"</b></font>"
if (document.layers){
document.layers.liveclock.document.write(myclock)
document.layers.liveclock.document.close()
}
else if (document.all)
liveclock.innerHTML=myclock
else if (document.getElementById)
document.getElementById("liveclock").innerHTML=myclock
setTimeout("show5()",1000)
}
window.onload=show5
//-->
</script></td>
    <td width="104" valign="bottom">
    <a href="#">Ver Resumen</a></td>
    <td width="123" valign="bottom">
    <a
        href=UIEstudiante.php?seccion=m>Modificar
Usuario</a></td>
    <td width="106" valign="bottom">
    <a href="Logout.php">Cerrar Sesion</a></td>
</tr>

```

FASE DE CONSTRUCCION

```

        </table></td>
    </tr> <tr >
        <td
            width="246"
            height="503"
background="Imagenes/menu4.png"><table
width="246"
height="433"
border="0" cellpadding="0" cellspacing="0" valign="top" >
    <tr>
        <td
            width="246"
            height="140"
            align="center"
            valign="top"></td>
    </tr>
    <tr>
        <td height="28" align="center" valign="top"><label>
            <a href=UIEstudiante.php?seccion=a class="varios2">Capitulo
1</a>
        </label></td>
    </tr>
    <tr>
        <td height="28" align="center" valign="top"><label></label>
            <a href=UIEstudiante.php?seccion=b class="varios2">Capitulo
2</a></td>
    </tr>
    <tr>
        <td height="28" align="center" valign="top"><label>
            <a href=UIEstudiante.php?seccion=c class="varios2">Capitulo
3</a></label></td>
    </tr>
    <tr>
        <td height="28" align="center" valign="top"><label>
            <a href=UIEstudiante.php?seccion=d class="varios2">Capitulo
4</a></label></td>
    </tr>
    <tr>
        <td height="28" align="center" valign="top"><label>
            <a href=UIEstudiante.php?seccion=e class="varios2">Capitulo
5</a></label></td>
    </tr>
    <tr>
        <td height="28" align="center" valign="top"><label>
            <a href=UIEstudiante.php?seccion=f class="varios2">Capitulo
6</a></label></td>
    </tr>

```


FASE DE CONSTRUCCION

```

<tr>
  <td height="28" align="center" valign="top"><label></label>
  <a href=UIEstudiante.php?seccion=g class="varios2">Capitulo
7</a></td>
</tr> <tr>
  <td height="28" align="center" valign="top"><label>
  <a href="#" class="varios2">Capitulo 8</a></label></td>
</tr>
<tr>
  <td height="28" align="center" valign="top"><label></label>
  <a href="#" class="varios2">Capitulo 9</a></td>
</tr>
<tr>
  <td height="28" align="center" valign="top"><label></label>
  <a href=UIEstudiante.php?seccion=z
class="varios2">Evaluacion</a></td>
</tr>
</table></td>
<td><table width="697" height="506" border="0" cellpadding="0"
cellspacing="0">
  <tr>
    <td width="349" colspan="2" align="center" valign="middle"
background="Imagenes/centro14.png">
    <form id="form1" name="form1" method="post" action="">
    <?PHP
        switch($_GET['seccion'])
        {
            case "a": $archivo="Capitulo1.php";
                break;
            case "b": $archivo="Capitulo2.php";
                break;
            case "c": $archivo="Capitulo3.php";
                break;
            case "d": $archivo="Capitulo4.php";
                break;
            case "e": $archivo="Capitulo5.php";
                break;
            case "f": $archivo="Capitulo6.php";
                break;

```

FASE DE CONSTRUCCION

```

                                case "g": $archivo="Capitulo7.php";
                                    break;
                                case "m":
                                    break;
                                case "z": $archivo="Evaluacion.php";
                                    break;
                                default: $archivo="Logo.php";
                                    break;
                                }
                                include("$archivo");
                                ?> </form>                                </td>
                                </tr>

                                </table></td>
                                </tr>
                                </table>

```

```

                                <!-- InstanceEndEditable --></td>
                                </tr>
                                </table>
                                </body>
                                <!-- InstanceEnd --></html>

```

Consulta

```

<?php
require("Conectar.php");
class Consultas
{
    //Metodo que realiza consultas sql
    function ejecutarConsulta($query)
    {
        $resultado = mysql_query($query);

        if($resultado==1)
            return true;
        else if($resultado==0)
            return false;

        for($i=0 ; $fila = mysql_fetch_row($resultado) ; $i++)

```

```

        for($j=0 ; $j<count($fila) ; $j++)
            $tabla[$i][$j] = $fila[$j];
    return $tabla;
}
//Metodo para verificar usuario
function validarUsuario($nick, $clave)
{
    $query= "select tipo from usuario where (nickname ='$nick') &
(password ='$clave')";
    $conn = new Conectar();
    $conn->conectarBD();
    $tabla = $this->ejecutarConsulta($query);
    $conn->desconectarBD();
    return $tabla;
}
?>

```

Conectar.php

```

<?PHP

//Clase para realizar la conexion y desconexion con la BD
class Conectar
{
    var $conexion;
    var $servidor = "localhost";
    var $usuario = "root";
    var $pwd = "123456";
    var $bd = "setpoo";

    //Funcion conectar a BD
    function conectarBD()
    {
        $this->conexion = mysql_connect($this->servidor, $this-
>usuario, $this->pwd);
        return mysql_select_db($this->bd, $this->conexion);
    }

    //Funcion desconectar de la BD
    function desconectarBD()
    {

```

FASE DE CONSTRUCCION

```

        return mysql_close($this->conexion);
    }
}??>
valida.php

<?PHP
require("Consultas.php");
$cons= new Consultas();
$tabla=                $cons->validarUsuario($_POST['txtusuario'],
$_POST['txtpass']);
$nroFilas= count ($tabla);
if($nroFilas!=0)
{
    session_start();
    $_SESSION["autenticado"]="SI";
    if($tabla[0][0]=='1')
    {
        $infUser=        $cons->        infProf($_POST['txtusuario'],
$_POST['txtpass']);
        $_SESSION["nombre"]= $infUser[0][0];
        $_SESSION["apellido"]= $infUser[0][1];
        $_SESSION["cedula"]= $infUser[0][2];
        $_SESSION["nickname"]= $infUser[0][3];
        header ("Location: IUProfesor.php");
    }
    else
    {
        $infUser=        $cons->        infEst($_POST['txtusuario'],
$_POST['txtpass']);
        $_SESSION["nombre"]= $infUser[0][0];
        $_SESSION["apellido"]= $infUser[0][1];
        $_SESSION["cedula"]= $infUser[0][2];
        $_SESSION["nickname"]= $infUser[0][3];
        header("Location: UIEstudiante.php");
    }
}exit;
}
else
echo "Usuario no registrado";
?>

```

FASE DE CONSTRUCCION

La figura 4.21 y 4.22 muestra la interfaz de consultar capítulo 1, y a continuación se muestra el código respectivo, el cual requiere las clases antes mostrados.



Figura 4.21 Interfaz Consultar capítulo1 del sistema SETPOO



Figura 4.21 Interfaz Consultar capítulo1 del sistema SETPOO

Codigo:

```

<script src="Scripts/AC_RunActiveContent.js"
type="text/javascript"></script>
<script type="text/javascript">
AC_FL_RunContent('codebase','http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=9,0,28,0','width','698','height','504','title','Capitulo 1','src','Flash/capitulo1','quality','high','pluginspage','http://www.adobe.com/shockwave/download/download.cgi?P1_Prod_Version=ShockwaveFlash','movie','Flash/capitulo1'); //end AC code
</script><noscript><object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=9,0,28,0" width="698" height="504" title="Capitulo 1">
<param name="movie" value="Flash/capitulo1.swf" />
<param name="quality" value="high" />
<embed src="Flash/capitulo1.swf" quality="high"
pluginspage="http://www.adobe.com/shockwave/download/download.cgi?P1_Prod_Version=ShockwaveFlash" type="application/x-shockwave-flash" width="698" height="504"></embed></object>
</noscript>

```

4.7 CONCLUSIÓN DE LA FASE DE ELABORACIÓN

En la fase de elaboración se ha abarcado todo el dominio del proyecto, profundizando en los puntos críticos de la arquitectura o riesgos importantes, se han actualizado todos los productos de la fase de inicio, se tiene una visión estable de la arquitectura del sistema y los principales elementos de riesgo han sido abordados y resueltos. Con el cumplimiento de la planificación de esta fase, se han obtenido todos los elementos necesarios para completar la implementación del sistema en la siguiente fase.

CAPITULO 5

FASE DE CONSTRUCCIÓN

5.1 INTRODUCCION

La finalidad principal de la fase de construcción es alcanzar la capacidad operacional del producto. Durante esta fase todos los componentes, características y requisitos identificados y detallados en las fases anteriores deben ser implementados, integrados y probados en su totalidad, obteniendo una versión aceptable del software.

La meta de este capítulo es lograr el desarrollo del sistema con calidad de producción, mediante la implementación de toda la funcionalidad y realización de las pruebas.

5.2 PLANIFICACION DE LA FASE DE CONSTRUCCION

En general, no se identificaron nuevos requisitos durante la fase de construcción y se trabajó directamente en la implementación y las pruebas empleando como base el análisis y diseño de la fase de elaboración.

5.3 ESCOGENCIA DE APLICACIONES PARA EL DESARROLLO DEL SISTEMA SETPOO

Para llevar a cabo el desarrollo del sistema SETPOO se realizó una selección de aplicaciones que se adaptaran a las necesidades del desarrollo del software

5.3.1 Escogencia del Lenguaje de Programación

Para el desarrollo del sistema SETPO, se debe contar con un lenguaje que permita la metodología de la programación orientada a objetos, ya que esta facilita el desarrollo de software así como el mantenimiento de los mismos, igualmente permite códigos más limpios y bien estructurados.

Es por esto que se escogió como lenguaje de programación PHP (Hypertext Pre-Processor), debido a que es un lenguaje interpretado, por tanto multiplataforma, de propósito general ampliamente usado y que está diseñado especialmente para la creación de páginas web, se ejecuta en un servidor web, tomando el código en PHP

como su entrada y creando páginas web como salida, además PHP es open source y la velocidad de ejecución es relativamente alta ya que no requiere demasiados recursos del sistema.

5.3.2 Editor de Programación

Para llevar a cabo el desarrollo del sistema SETPOO se utilizó Adobe Dreamweaver CS3 (la interfaz se muestra en la figura 5.1), la cual es una herramienta robusta para la creación de aplicaciones y sitios web. Ofrece una combinación muy útil de herramientas de diseño visual, funciones de desarrollo de aplicaciones y soporte para la edición de código. Todo ello permite a los desarrolladores y diseñadores de diferentes niveles de conocimiento crear sitios y aplicaciones atractivos basados en estándares de forma rápida. Desde el soporte para el diseño basado en CSS hasta funciones de codificación manual, Dreamweaver ofrece las herramientas que los profesionales necesitan en un entorno integrado y mejorado. Los desarrolladores pueden utilizar Dreamweaver con la tecnología de servidores que deseen para crear aplicaciones de Internet eficaces que conecten a los usuarios con base de datos, servicios web y sistemas heredados.

Dreamweaver también incluye numerosas herramientas y funciones relacionadas con la codificación. Además, ayuda a crear aplicaciones web dinámicas basadas en bases de datos empleando tecnología de servidor como ASP, ASP.NET, ColdFusion, JSP y PHP.

5.3.3 Desarrollo de Animaciones

Adobe Flash CS3 (la interfaz se muestra en la figura 5.2), fue la herramienta seleccionada para llevar a cabo las animaciones correspondientes al sistema SETPOO, ya que permite el desarrollo de aplicaciones interactivas, Flash está basado en imágenes vectoriales, lo cual permite que ajustes el tamaño de las imágenes sin que ellas se píxelen.

FASE DE CONSTRUCCION

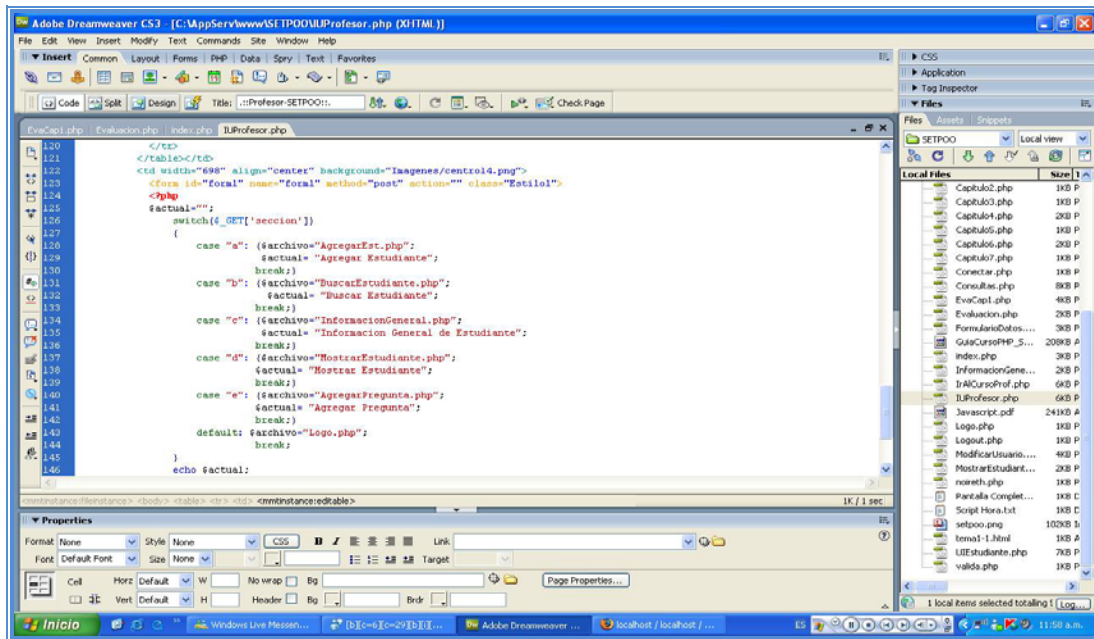


Figura 5.1 Interfaz de Adobe Dreamweaver CS3

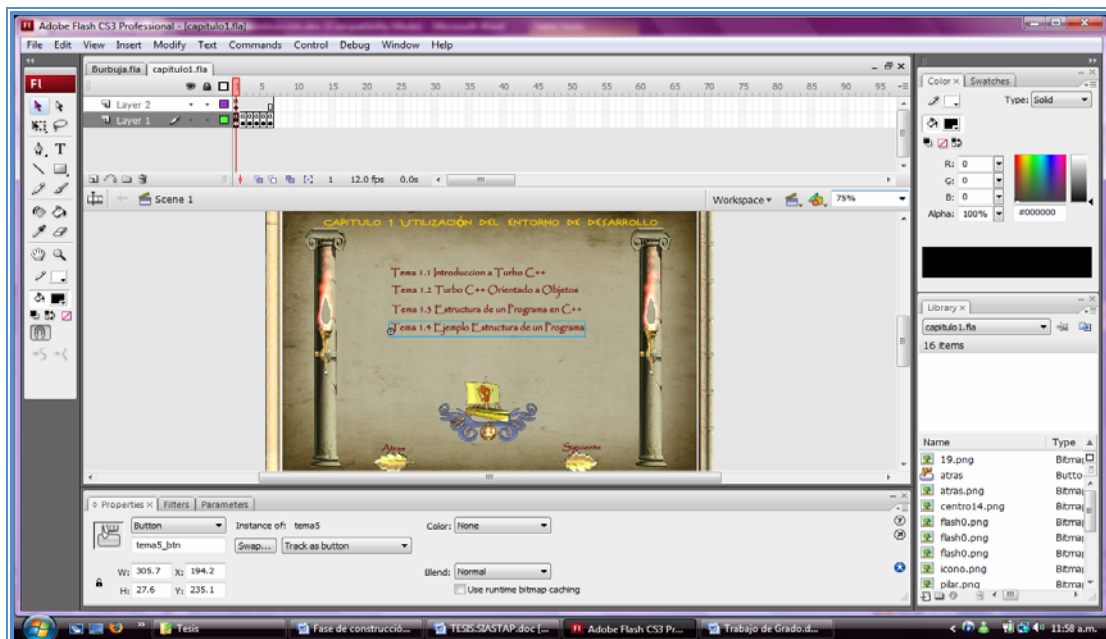


Figura 5.2 Interfaz de Adobe Flash CS3

5.3.4 Escogencia del gestor de base de datos

FASE DE CONSTRUCCION

Para la construcción de la base de datos de SETPOO, se escogió MySQL, es un sistema de gestión de base de datos relacional, multihilo y multiusuario. La interfaz se muestra en la figura 5.3.

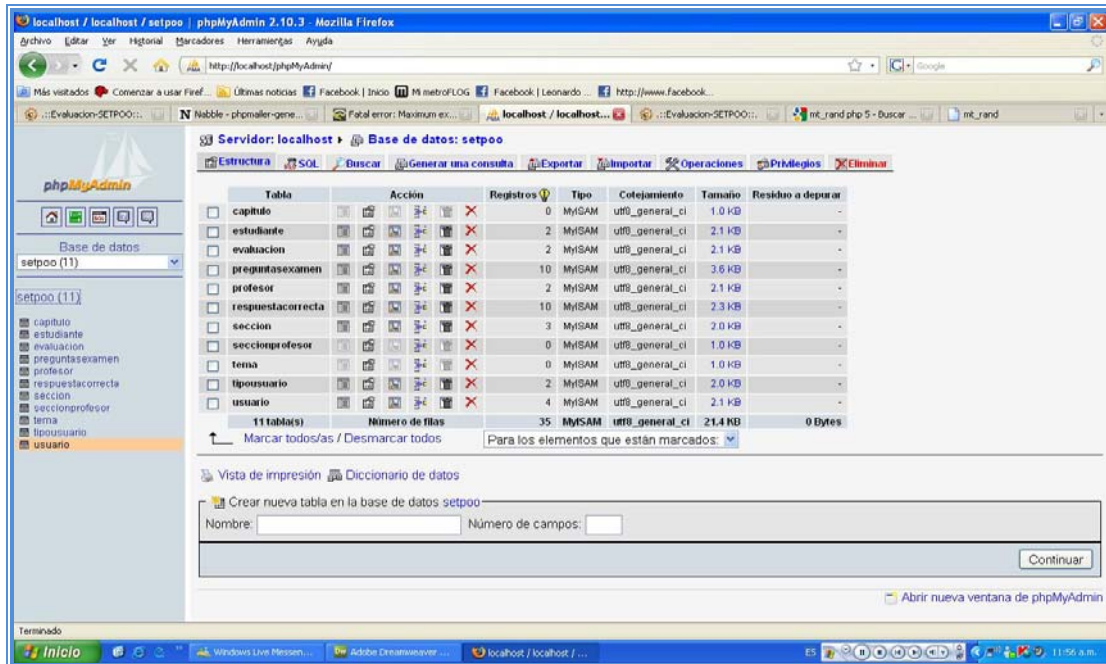


Figura 5.3 Interfaz de manejador de base de datos PHP MyAdmin

5.4 FLUJO DE TRABAJO IMPLEMENTACION

El propósito esencial de este flujo de trabajo es implementar el sistema en términos de componentes, y codificación del software, integración de módulos y la explicación acerca de las funcionalidades del sistema y su correcto uso, descubriendo así toda la estructura en forma de código abierto e identificación de las actividades que este puede realizar.

5.4.1 Diagrama de Componentes Total

FASE DE CONSTRUCCION

En esta fase se desarrolla el diagrama de componentes con la totalidad de los componentes del sistema, indicando las clases que necesitan cada uno. En la figura 5.4 podemos observar este diagrama.

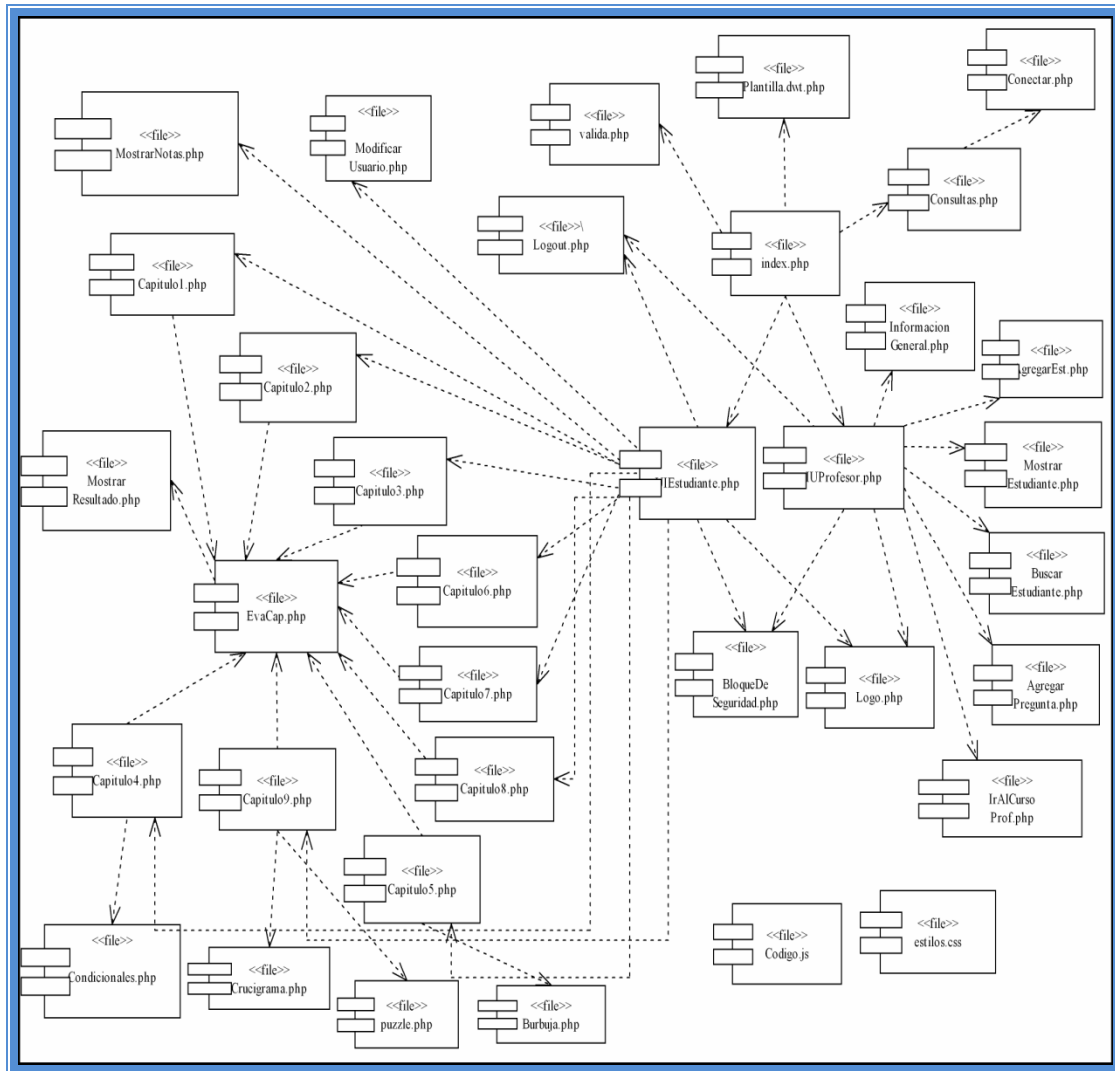


Figura 5.4 Diagrama de componentes total del sistema SETPOO

5.5 FLUJO DE TRABAJO PRUEBAS

El propósito esencial de este flujo de trabajo es comprobar el resultado de la implementación mediante las pruebas de cada construcción, así como validar y verificar el software, es decir, determinar si el sistema satisface los requisitos y funciona de la manera establecida.

5.5.1 Pruebas por Unidad

Las pruebas por unidad se aplicaron mediante la aplicación de la prueba de la caja negra sobre los diversos componentes del sistema. Para realizar este tipo de pruebas, se identifican un conjunto de valores que pueden ser introducidos por un actor, y se expresan como clases de equivalencia para poder abarcar la totalidad de las ocurrencias de un evento de inserción de datos.

A continuación en la tabla 5.1 se muestra el componente index, el cual permite el logueo de usuarios tanto profesores como estudiantes.

<i>Nº</i>	<i>Dato</i>	<i>Clase de Equivalencia</i>	<i>valido</i>	<i>Invalido</i>
1	Usuario	Carácter Numérico		X
2	Usuario	Carácter Alfanumérico		X
3	Usuario	Longitud del carácter <=30	X	
4	Usuario	Dato nulo		X
5	Contraseña	Carácter Numérico	X	
6	Contraseña	Carácter Alfanumérico	X	
7	Contraseña	Longitud del carácter <=10	X	
8	Contraseña	Dato nulo		X

Tabla 5.1 Clase de equivalencia para el componente Index

A continuación en la tabla 5.2, se representan las clases del componente agregar estudiante, el mismo se encarga de ingresar un nuevo estudiante a la base de datos.

FASE DE CONSTRUCCION

<i>Nº</i>	<i>Dato</i>	<i>Clase de Equivalencia</i>	<i>valido</i>	<i>Invalido</i>
1	Nombre	Carácter Numérico		X
2	Nombre	Carácter Alfabetico	X	
3	Nombre	Longitud del carácter <=30	X	
4	Nombre	Dato nulo		X
5	Apellido	Carácter Numérico		X
6	Apellido	Carácter Alfabeticos	X	
7	Apellido	Longitud del carácter <=30	X	
8	Apellido	Dato nulo		X
9	Cedula	Carácter Numérico	X	
10	Cedula	Carácter Alfanumérica		X
11	Cedula	Longitud del carácter <=8	X	
12	Cedula	Dato nulo		X
13	Sección	Carácter Numérico	X	
14	Sección	Carácter Alfanumérica		X
15	Sección	Longitud del carácter <=8	X	
16	Sección	Dato nulo		X

Tabla 5.2. Clase de equivalencia para el componente agregar estudiante

En la tabla 5.3, se representan las clases del componente modificar perfil perteneciente a la interface del estudiante, la cual permite al estudiante modificar datos de su perfil, tales como e-mail, teléfono, password.

<i>Nº</i>	<i>Dato</i>	<i>Clase de equivalencia</i>	<i>valido</i>	<i>Invalido</i>
1	Teléfono	Carácter Numérico	X	
2	Teléfono	Carácter Alfanumérico		X
3	Teléfono	Longitud del carácter <=11	X	
4	Teléfono	Dato nulo	X	
5	e-mail	Carácter Numérico	X	
6	e-mail	Carácter Alfanumérica	X	
7	e-mail	Longitud del carácter <=50	X	
8	e-mail	Dato nulo	X	
9	Password	Carácter Numérico	X	
10	Password	Carácter Alfanumérica	X	
11	Password	Longitud del carácter <=10	X	
12	Password	Dato nulo		X
13	Confirmar Password	Carácter Numérico	X	
14	Confirmar Password	Carácter Alfanumérica	X	
15	Confirmar Password	Longitud del carácter <=8	X	
16	Confirmar Password	Dato nulo		X
17	Confirmar Password	Datos igual a Password	X	

Tabla 5.3. Clase de equivalencia para el componente Agregar Astudiante

FASE DE CONSTRUCCION

La tabla 5.4 muestra la clase del componente buscar estudiante, la cual se muestra en la interfaz del profesor y le permite al mismo la búsqueda de estudiante por cedula.

<i>Nº</i>	<i>Dato</i>	<i>Clase de equivalencia</i>	<i>valido</i>	<i>Invalido</i>
1	Cedula	Carácter Numérico		X
2	Cedula	Carácter Alfanumérico		X
3	Cedula	Longitud del carácter <=30	X	
4	Cedula	Dato nulo		X

Tabla 5.4 clase de equivalencia para el componente Buscar Estudiante

La tabla 5.5 muestra la clase del componente Información General, la cual se muestra en la interfaz del profesor y le permite al mismo la opción de visualizar a todos los alumnos inscritos en el sistema, así mismo permite búsquedas de estudiantes de cada sección.

<i>Nº</i>	<i>Dato</i>	<i>Clase de Equivalencia</i>	<i>valido</i>	<i>Invalido</i>
1	Sección	Carácter Numérico	X	
2	Sección	Carácter Alfanumérico		X
3	Sección	Longitud del carácter <=8	X	
4	Sección	Dato nulo	X	

Tabla 5.6 clase de equivalencia para el componente Información General

La tabla 5.7 muestra la clase del componente agregar evaluación, perteneciente a la interfaz del profesor, la cual le permite al mismo agregar nuevas preguntas al sistema.

<i>Nº</i>	<i>Dato</i>	<i>Clase de Equivalencia</i>	<i>valido</i>	<i>Invalido</i>
1	Pregunta	Carácter Numérico		X
2	Pregunta	Carácter Alfanumérico	X	
3	Pregunta	Longitud del carácter <=500	X	
4	Pregunta	Dato nulo		X
5	Respuesta 1	Carácter Numérico		X
6	Respuesta 1	Carácter Alfanumérica	X	
7	Respuesta 1	Longitud del carácter <=500	X	
8	Respuesta 1	Dato nulo		X
9	Respuesta 2	Carácter Numérico		X

FASE DE CONSTRUCCION

10	Respuesta 2	Carácter Alfanumérica	X	
11	Respuesta 2	Longitud del carácter <=500	X	
12	Respuesta 2	Dato nulo		X
13	Respuesta 3	Carácter Numérico		X
14	Respuesta 3	Carácter Alfanumérica	X	
15	Respuesta 3	Longitud del carácter <=500	X	
16	Respuesta 3	Dato nulo		X
17	Respuesta 4	Carácter Numérico		X
18	Respuesta 4	Carácter Alfanumérica	X	
19	Respuesta 4	Longitud del carácter <=500	X	

Tabla 5.7 Clase de equivalencia para el componente agregar evaluación

En la tabla 5.8 y 5.9 se presentan algunos casos de prueba de caja negra, donde se incluirán datos que serán cotejados por las clases de equivalencia agregar estudiante y modificar perfil, respectivamente.

<i>Nº</i>	<i>Dato</i>	<i>Clase de Prueba</i>	<i>Salida</i>
1	Nombre	435679	Invalido
2	Nombre	Luis6	Invalido
3	Nombre	Francisco	Valido
4	Nombre	Nada	Invalido
5	Apellido	56789	Invalido
6	Apellido	Martinez2	Invalido
7	Apellido	Jiménez	Valido
8	Apellido	Nada	Invalido
9	Cedula	12359123	Valido
10	Cedula	V 19879008	Invalido
11	Cedula	219987002	Invalido
12	Cedula	Nada	Invalido
13	Sección	12	Valido
14	Sección	S 98	Invalido
15	Sección	98	Valido
16	Sección	Nada	Invalido

Tabla 5.8 Prueba de caja negra para Agregar Estudiante

<i>Nº</i>	<i>Dato</i>	<i>Clase de Prueba</i>	<i>Salida</i>
1	Teléfono	04168790989	Valido
2	Teléfono	0424 8796788	Invalido
3	Teléfono	080012567891000	Invalido
4	Teléfono	Nada	Valido
5	e-mail	1234598709	Invalido
6	e-mail	Nemesis48@gmail.com	Valido
7	e-mail	Luis78@hotmail.com	Valido
8	e-mail	nada	Valido

FASE DE CONSTRUCCION

9	Password	123456	Valido
10	Password	Rom30lope2	valido
11	Password	Nada	Invalido
12	Confirmar Password	123456	Valido
13	Confirmar Password	5rt64q5rf	valido
14	Confirmar Password	nada	Invalido
15	Confirmar Password	23415	Invalido

Tabla 5.9 Prueba de caja de negra para Modificar Perfil

5.5.2 Pruebas de Integración

El objetivo general de las pruebas de integración es, detectar las fallas de interacción entre las distintas clases que componen al sistema. Debido a que cada clase probada por separado se inserta de manera progresiva dentro de la estructura, las pruebas de integración son realmente un mecanismo para comprobar el correcto ensamblaje del sistema completo. Al efectuar la integración de los módulos, se concentra el esfuerzo en la búsqueda de fallas que puedan provocar excepciones arrojadas por los métodos; el empleo de operaciones equivocada, e invocación inadecuada de los métodos.

Luego de verificar la calidad de los componentes, se procede a comprobar la eficiencia de un conjunto de componentes integrados por fase, estas se pueden observar resaltadas en la figura 5.9 en donde se despliega el diagrama de componentes por fase de integración del sistema.

FASE DE CONSTRUCCION

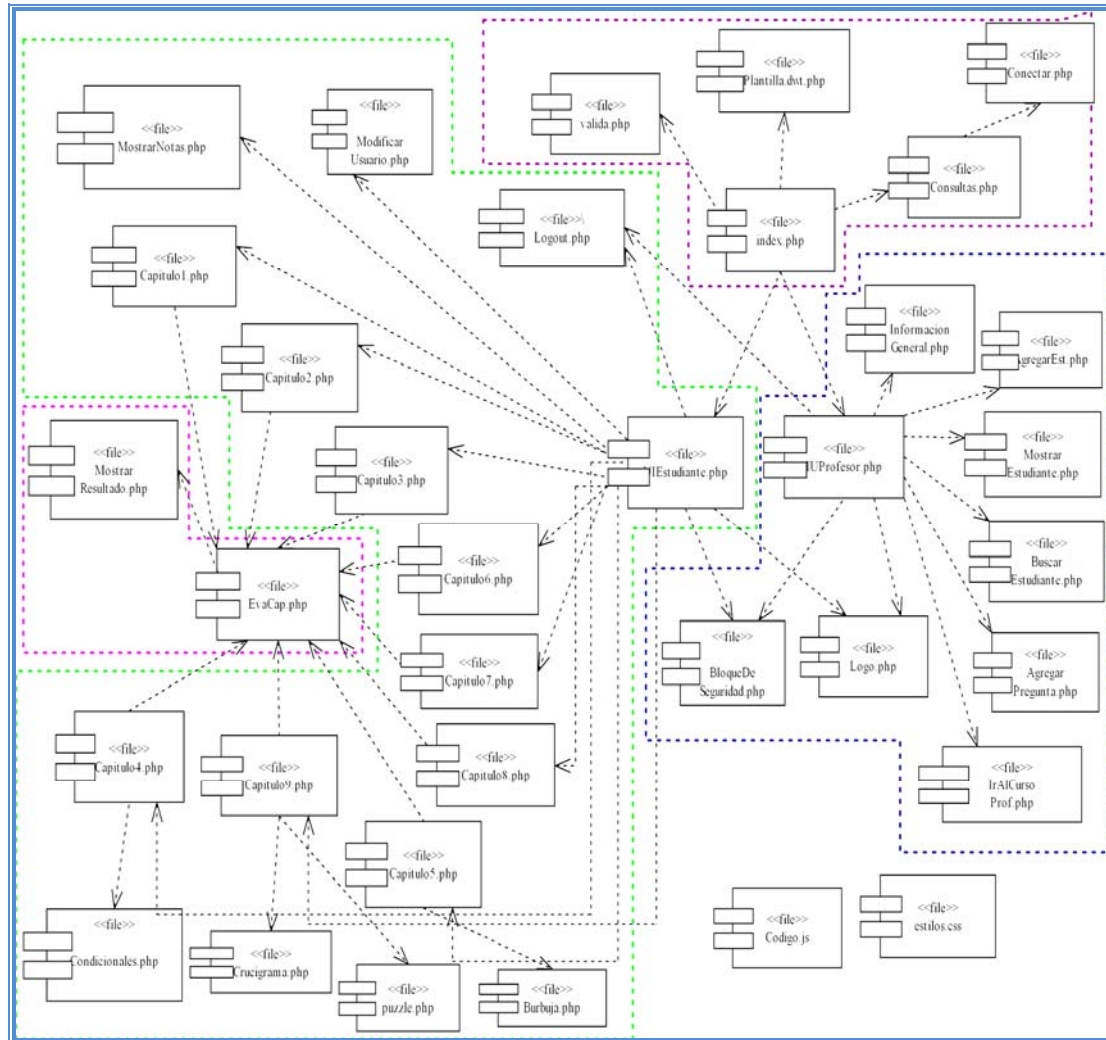


Figura 5.9 Diagrama de componentes en fases de integración

Leyenda***Integración de componente Agregar usuario:***

Este caso de prueba verifica la funcionalidad de la implementación de la fase 3 resaltada con el color azul, de la figura 5.9.

Entrada:

FASE DE CONSTRUCCION

<i>DATOS</i>	
NOMBRE:	Valentina
APELLIDO:	Ramirez
CEDULA:	17678970
SECCION:	01

Tabla 5.10. Datos de entrada para probar la integración de Agregar usuario

Resultado:

El sistema genera una clave y una contraseña con la que el nuevo estudiante puede ingresar al tutorial

Procedimiento de prueba:

- ✓ Activar la interfaz de profesor
- ✓ Acceder a la interfaz de agregar estudiante.
- ✓ Cargar los datos del nuevo estudiante de la tabla 5.10 y hacer click en el botón agregar, el sistema deberá asignar una clave y una contraseña. (Ver figura 5.10 y figura 5.11 respectivamente)
- ✓ Acceder a la interfaz información general, la cual muestra los datos de los usuarios pertenecientes al sistema. (Ver figura 5.12)
- ✓ Se realiza la comunicación con la base de datos a través del componente conexión y se observa los datos del estudiante que se acaba de ingresar. (Ver figura 5.13, figura 5.14)

A continuación se muestra la secuencia de las imágenes que muestra que procedimiento de integración.

FASE DE CONSTRUCCION

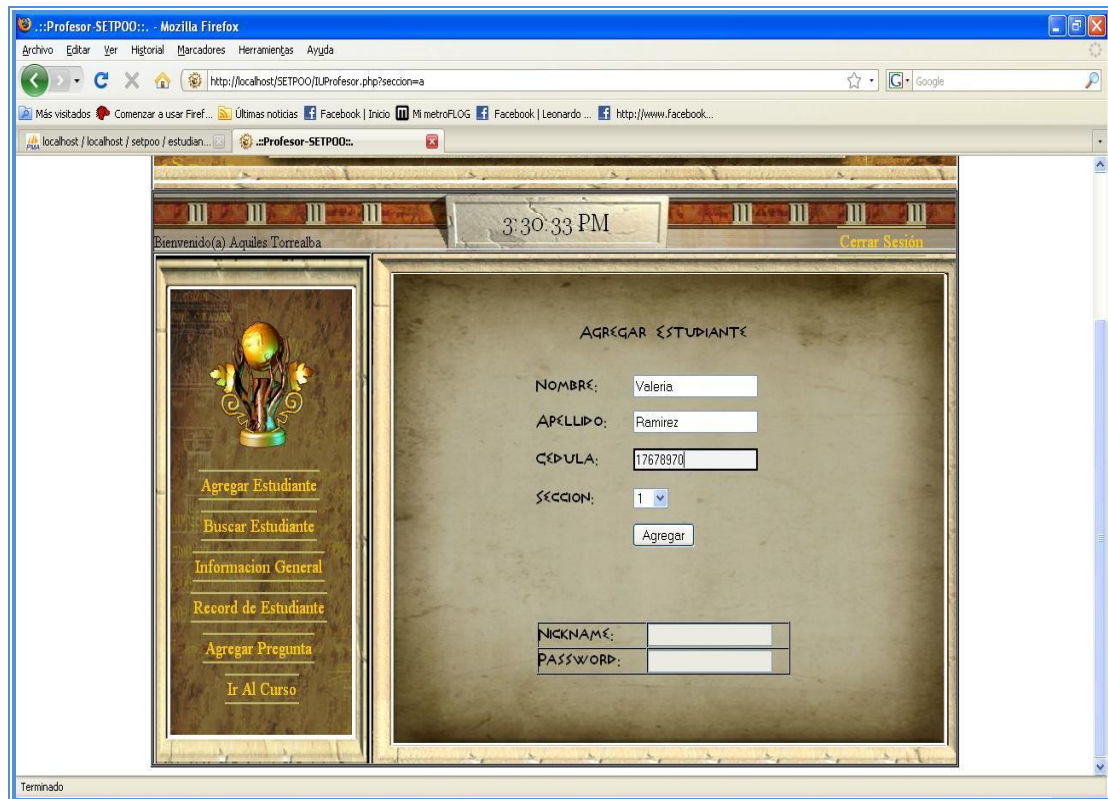


Figura 5.10 Interfaz Agregar estudiante

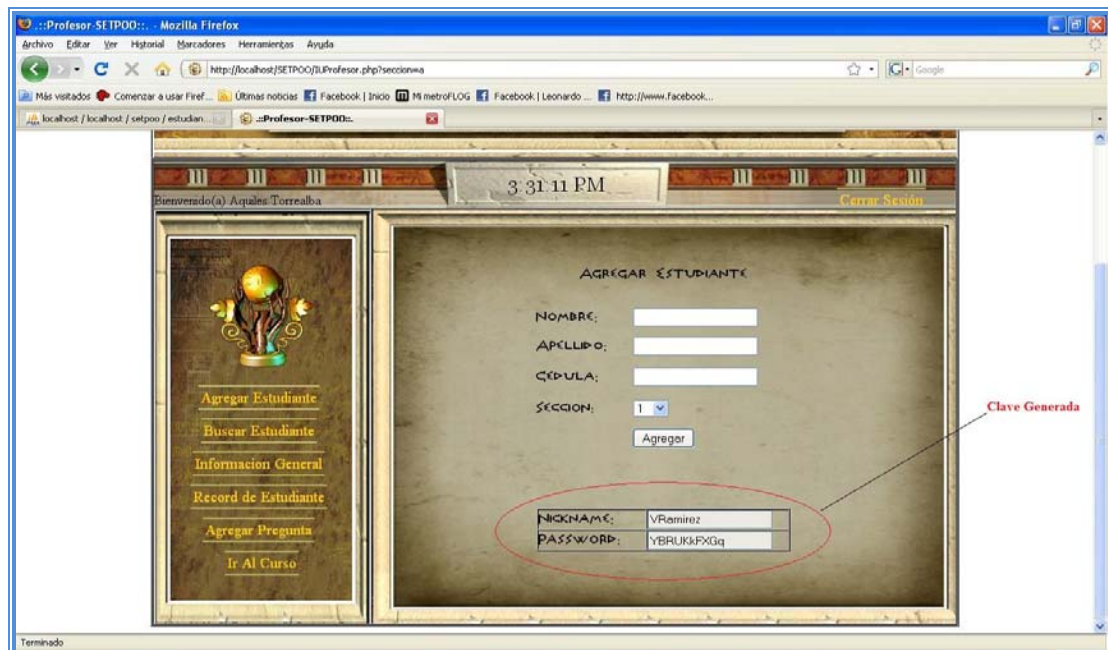


Figura 5.11 Interfaz muestra la clave y contraseña asignada al nuevo estudiante

FASE DE CONSTRUCCION



5.12 interfaz Información General

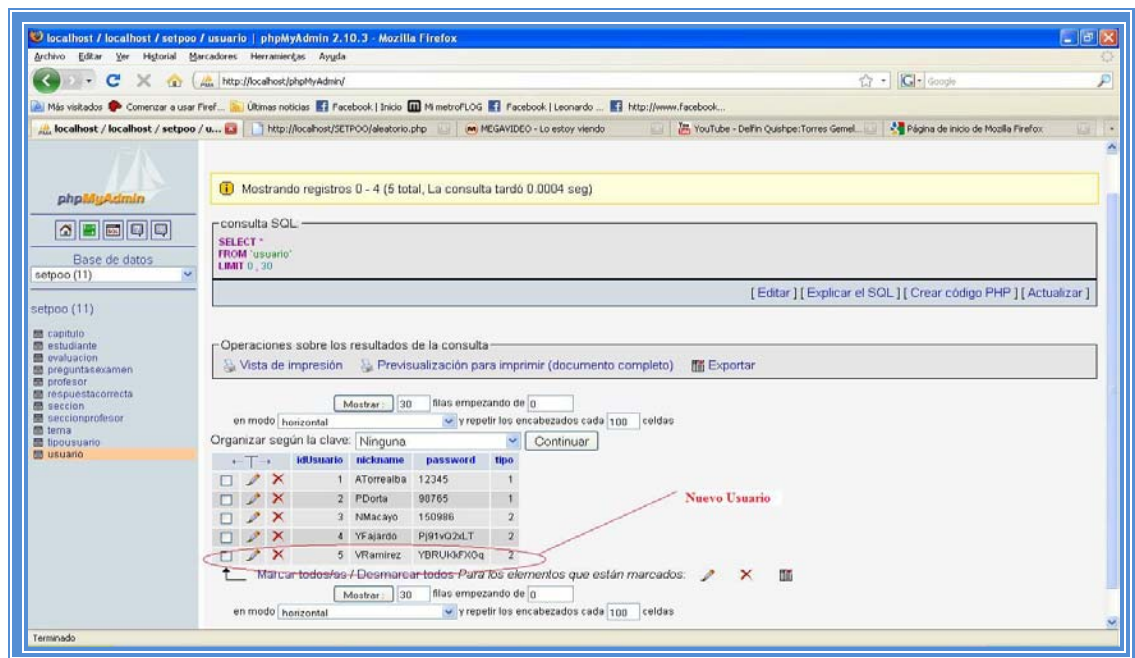


Figura 5.13 Interfaz PHP MyAdmin tabla de base de dato con nuevo usuario

FASE DE CONSTRUCCION

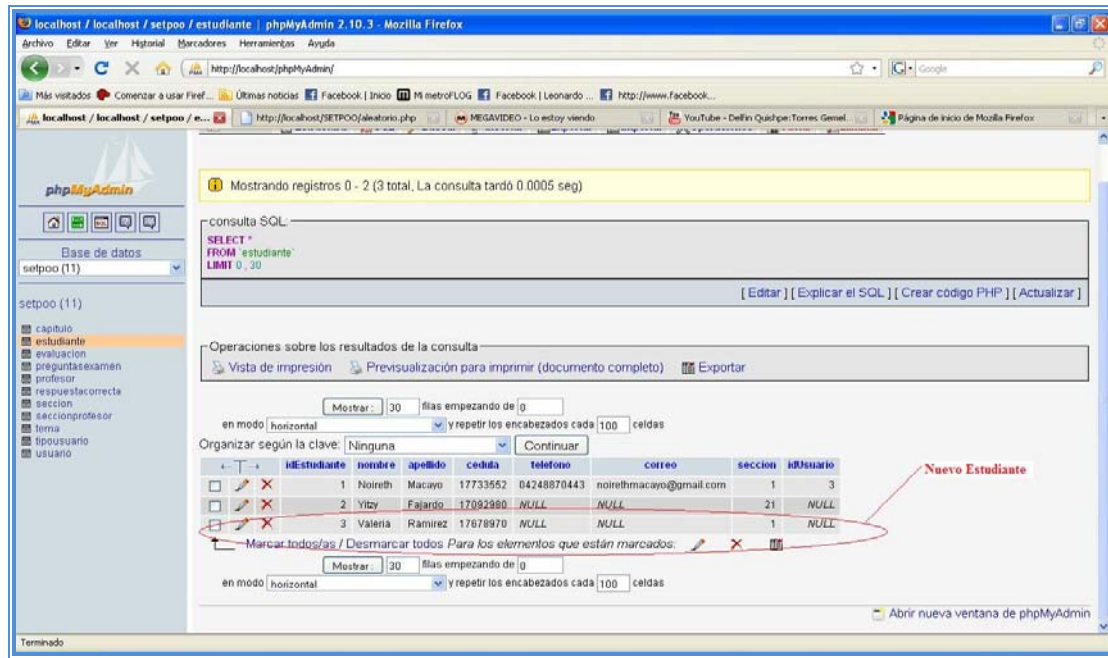


Figura 5.13 Interfaz PHP MyAdmin tabla de base de dato con nuevo estudiante

A continuación se presenta el código para cada una de las pantallas anteriores:

Código interfaz de Profesor. Figura 5.10

```
<?PHP
```

```
include("BloqueDeSeguridad.php");
```

```
?>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml"><!-- InstanceBegin
template="/Templates/Plantilla1.dwt.php"
```

```
codeOutsideHTMLIsLocked="false" --><head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"
```

```
/>
```

```
<!-- InstanceBeginEditable name="doctype" -->
```

FASE DE CONSTRUCCION

```

<title>::Profesor-SETPOO::.</title>
<LINK REL="SHORTCUT ICON" HREF="setpoo.png">
<link rel="stylesheet" type="text/css" href="css/Estilos.css"
media="screen"/>
<script type="text/javascript" src="Scripts/codigo.js"></script>
<!-- InstanceEndEditable -->
<!-- InstanceBeginEditable name="head" -->
<!-- InstanceEndEditable --></head>
<body>
<table width="944" border="1" cellspacing="0" cellpadding="0"
align="center"> <tr>
<td height="220" width="944" background="Imagenes/banner4.PNG"
background-repeat="no-repeat"></td>
</tr> <tr>
<td><!-- InstanceBeginEditable name="RegionEditable1" -->
<table width="944" height="559" border="1" cellpadding="0"
cellspacing="0">
<tr >
<td height="55" colspan="2"
background="Imagenes/menu4.png">
<table width="940" height="57" border="0" cellpadding="0"
cellspacing="0" background="Imagenes/menuiconos.png">
<tr>
<td width="350" valign="bottom">
<?PHP
$nombre= $_SESSION["nombre"];

```

FASE DE CONSTRUCCION

```
        $apellido= $_SESSION["apellido"];
        echo "Bienvenido(a) $nombre $apellido";
    ?></td>
    <td width="243" align="center">
        <span id="liveclock" ></span>
<script language="JavaScript" type="text/javascript">
    <!--
function show5(){
if (!document.layers&&!document.all&&!document.getElementById)
return

var Digital=new Date()
var hours=Digital.getHours()
var minutes=Digital.getMinutes()
var seconds=Digital.getSeconds()

var dn="PM"
if (hours<12)
dn="AM"
if (hours>12)
hours=hours-12
if (hours==0)
hours=12

if (minutes<=9)
minutes="0"+minutes
```

```

    if (seconds<=9)
    seconds="0"+seconds
    //change font size here to your desire
    myclock="<font size='5' face='Georgia' >"+hours+": "+minutes+": "
    +seconds+" "+dn+"</b></font>"
    if (document.layers){
    document.layers.liveclock.document.write(myclock)
    document.layers.liveclock.document.close()
    }
    else if (document.all)
    liveclock.innerHTML=myclock
    else if (document.getElementById)
    document.getElementById("liveclock").innerHTML=myclock
    setTimeout("show5()",1000) }
window.onload=show5
    //-->
    </script></td>
        <td width="208">&nbsp;</td>
        <td width="139" valign="bottom"><a href="Logout.php"
class="varios2">Cerrar Sesión</a></td>
    </tr>
</table></td>
</tr>
<tr >
    <td width="255" height="502"
background="Imagenes/menu4.png">

```


FASE DE CONSTRUCCION

```

<table width="230" height="393" border="0" cellpadding="0"
cellspacing="0" align="center">
  <tr>
    <td height="153" class="EstiloTablaProf">&nbsp;</td>
  </tr>
  <tr>
    <td height="40" align="center">
      <a href=IUProfesor.php?seccion=a class="varios2">Agregar
Estudiante</a>
      &nbsp;</td>
    </tr>
  <tr>
    <td height="40" align="center">
      <a href=IUProfesor.php?seccion=b class="varios2">Buscar
Estudiante</a>
      &nbsp;</td>
    </tr>
  <tr>
    <td height="40" align="center">
      <a href=IUProfesor.php?seccion=c class="varios2">Informacion
General</a>
      &nbsp;</td>
    </tr>
  <tr>
    <td height="40" align="center">

```

FASE DE CONSTRUCCION

```

        <a href=IUProfesor.php?seccion=d class="varios2">Record de
Estudiante</a>
        &nbsp;</td>
</tr><tr>
        <td align="center" height="40">
        <a href=IUProfesor.php?seccion=e class="varios2">Agregar
Pregunta</a>
        &nbsp;</td>
</tr><tr>
        <td align="center" height="40"><a href="IrAlCursoProf.php"
class="varios2">Ir Al Curso</a></td>
        </tr>
</table></td>
<td
        width="698"
        align="center"
background="Imagenes/centro14.png">
        <form id="form1" name="form1" method="post" action=""
class="Estilo1">
        <?php
        $actual="";
        switch($_GET['seccion'])
        {
                case "a": {$archivo="AgregarEst.php";
                        $actual=
Estudiante";
                                "Agregar
                                break;}
                case "b": {$archivo="BuscarEstudiante.php";

```

FASE DE CONSTRUCCION

```

                                $actual=          "Buscar
Estudiante";
                                break;}
                                case "c": {$archivo="InformacionGeneral.php";
                                $actual=          "Informacion General de
Estudiante";
                                break;}
                                case
                                "d":
                                {$archivo="MostrarEstudiante.php";
                                $actual=          "Mostrar
Estudiante";
                                break;}
                                case "e": {$archivo="AgregarPregunta.php";
                                $actual=          "Agregar
Pregunta";
                                break;}
                                default: $archivo="Logo.php";
                                break;
                                }
                                echo $actual;
                                include("$archivo");
                                ?>
                                </form> </td>
                                </tr>
                                </table>

```

```

        <!-- InstanceEndEditable --></td>
    </tr>
</table>
</body>
<!-- InstanceEnd --></html>

```

La clase requiere de otra clase, la clase Consulta

```

<?php
require("Conectar.php");
class Consultas
{
    //Metodo que realiza consultas sql
    function ejecutarConsulta($query)
    {
        $resultado = mysql_query($query);

        if($resultado==1)
            return true;
        else if($resultado==0)
            return false;

        for($i=0 ; $fila = mysql_fetch_row($resultado) ; $i++)
            for($j=0 ; $j<count($fila) ; $j++)
                $tabla[$i][$j] = $fila[$j];
        return $tabla;
    }
    //Metodo que para almacenar estudiantes
    function guardarEstudiante($nom, $ape, $ced, $sec)

```

FASE DE CONSTRUCCION

```
{
    $query = "insert into estudiante (nombre, apellido, cedula,
seccion) values ('$nom', '$ape', '$ced', '$sec');";
    $conn = new Conectar();
    $conn->conectarBD();
    $tabla = $this->ejecutarConsulta($query);
    $conn->desconectarBD();
    return $tabla;
}

//Metodo para almacenar las notas por defecto
function notaDefault()
{
    $query= "insert into evaluacion (nota1, nota2, nota3,
nota4, nota5, nota6, nota7, nota8, nota9) values (0,0,0,0,0,0,0,0,0);";
    $conn = new Conectar();
    $conn->conectarBD();
    $tabla = $this->ejecutarConsulta($query);
    $conn->desconectarBD();
    return $tabla;
}

//Metodo para buscar las secciones que existen
function buscarseccion()
{
    $query= "select numero from seccion";
    $conn = new Conectar();
    $conn->conectarBD();
    $tabla = $this->ejecutarConsulta($query);
    $conn->desconectarBD();
    return $tabla;
}
```

FASE DE CONSTRUCCION

```
}  
//Metodo para mostrar los estudiantes  
function mostrarEstudiante()  
{  
    $query= "SELECT cedula, nota1, nota2, nota3, nota4, nota5,  
nota6, nota7, nota8, nota9 FROM estudiante, evaluacion WHERE  
estudiante.idestudiante = evaluacion.idestudiante";  
    $conn = new Conectar();  
    $conn->conectarBD();  
    $tabla = $this->ejecutarConsulta($query);  
    $conn->desconectarBD();  
    return $tabla;  
}  
//Metodo que muestra la informacion general del estudiante  
function informacionGeneral()  
{  
    $query= "select nombre, apellido, cedula, correo, telefono from  
estudiante";  
    $conn = new Conectar();  
    $conn->conectarBD();  
    $tabla = $this->ejecutarConsulta($query);  
    $conn->desconectarBD();  
    return $tabla;  
}  
//Metodo para enviar las preguntas  
function guardarPregunta($preg, $resp1, $resp2, $resp3, $resp4, $cap)
```

FASE DE CONSTRUCCION

```

    { $query= "insert into preguntasexamen (pregunta, respuesta1,
respuesta2, respuesta3, respuesta4, capitulo) values ('$preg', '$resp1', '$resp2',
'$resp3', '$resp4', '$cap');"
        $conn = new Conectar();
        $conn->conectarBD();
        $tabla = $this->ejecutarConsulta($query);
        $conn->desconectarBD();
        return $tabla;
    }
//Metodo para guardar las respuesta correctas simples
function respSimple($resp)
{
    $query= "insert into respuestacorrecta (respcorre1) values
('$resp');"
    $conn = new Conectar();
    $conn->conectarBD();
    $tabla = $this->ejecutarConsulta($query);
    $conn->desconectarBD();
    return $tabla;
}
//Metodo para guardar las respuestas correctas dobles
function RespDoble($resp1, $resp2)
{
    $query= "insert into respuestacorrecta (respcorre1, respcorre2)
values ('$resp1', '$resp2');"
    $conn = new Conectar();
    $conn->conectarBD();

```

FASE DE CONSTRUCCION

```
        $tabla = $this->ejecutarConsulta($query);
        $conn->desconectarBD();
        return $tabla;
    }

    //Metodo para buscar todos los datos
    function buscarDatos($ced)
    {
        $query= "SELECT estudiante.nombre, estudiante.apellido,
estudiante.cedula, estudiante.telefono, estudiante.correo, estudiante.seccion,
evaluacion.nota1, evaluacion.nota2, evaluacion.nota3, evaluacion.nota4,
evaluacion.nota5, evaluacion.nota6, evaluacion.nota7, evaluacion.nota8,
evaluacion.nota9 FROM evaluacion, estudiante WHERE (estudiante.cedula =
'$ced') & ( estudiante.idEstudiante = evaluacion.idEstudiante )";

        $conn = new Conectar();
        $conn->conectarBD();
        $tabla = $this->ejecutarConsulta($query);
        $conn->desconectarBD();
        return $tabla;
    }

    //Metodo para verificar usuario
    function validarUsuario($nick, $clave)
    {
        $query= "select tipo from usuario where (nickname ='$nick') &
(password ='$clave)";

        $conn = new Conectar();
```


FASE DE CONSTRUCCION

```
        $conn->conectarBD();
        $tabla = $this->ejecutarConsulta($query);
        $conn->desconectarBD();
        return $tabla;
    }
} //Metodo para almacenar los usuarios con sus contraseñas
function guardarUsuario($nick, $pass)
{
    $query = "insert into usuario (nickname, password, tipo) values
('$nick', '$pass','2')";
    $conn = new Conectar();
    $conn->conectarBD();
    $tabla = $this->ejecutarConsulta($query);
    $conn->desconectarBD();
    return $tabla;
}
} //Metodo para devolver los ids de la tabla usuario
function idUsuarioGen()
{
    $query= "select idUsuario from usuario";
    $conn = new Conectar();
    $conn->conectarBD();
    $tabla = $this->ejecutarConsulta($query);
    $conn->desconectarBD();
    return $tabla;
}
} //Metodo para guardar id en tabla Estudiante
```

FASE DE CONSTRUCCION

```
function guardarId($id, $cedula)
{
    $query= "update estudiante set idUsuario= '$id' where cedula=
'$cedula'";

    $conn = new Conectar();
    $conn->conectarBD();
    $tabla = $this->ejecutarConsulta($query);
    $conn->desconectarBD();
    return $tabla;
} //Metodo para devolver informacion de Profesor
function infProf($nick, $clave)
{
    $query = "SELECT profesor.nombre, profesor.apellido,
profesor.cedula, usuario.nickname, usuario.tipo FROM usuario,
profesor WHERE (usuario.nickname = '$nick') & ( usuario.password =
'$clave' ) & ( profesor.idUsuario = usuario.idUsuario )";

    $conn = new Conectar();
    $conn->conectarBD();
    $tabla = $this->ejecutarConsulta($query);
    $conn->desconectarBD();
    return $tabla;
}
//Metodo para devolver informacion de Profesor
function infEst($nick, $clave)
{
    $query = "SELECT estudiante.nombre, estudiante.apellido,
estudiante.cedula, usuario.nickname, usuario.tipo FROM usuario,
```

FASE DE CONSTRUCCION

```

    estudiante WHERE (usuario.nickname = '$nick') & ( usuario.password
= '$clave' ) & ( estudiante.idUsuario = usuario.idUsuario );
        $conn = new Conectar();
        $conn->conectarBD();
        $tabla = $this->ejecutarConsulta($query);
        $conn->desconectarBD();
        return $tabla;
    } //Metodo para actualizar los datos del estudiante
function actDatos($tlf, $correo, $ced)
    {
        $query = "update estudiante set telefono='$tlf', correo=
'$correo' where cedula='$ced'";
        $conn = new Conectar();
        $conn->conectarBD();
        $tabla = $this->ejecutarConsulta($query);
        $conn->desconectarBD();
        return $tabla;
    }
//Metodo para actualizar contraseña
function actPass($clave, $ced, $nick)
    {
        $query= "update usuario, estudiante set password= '$clave'
where (estudiante.cedula= '$ced')&(nickname= '$nick')";
        $conn = new Conectar();
        $conn->conectarBD();
        $tabla = $this->ejecutarConsulta($query);
        $conn->desconectarBD();
    }

```

```
        return $tabla;
    }
    //Metodo para buscar las notas de un estudiante especifico
    function VerificarNotas($ced)
    {
        $query = "select nota1, nota2, nota3, nota4, nota5, nota6, nota7,
nota8, nota9 from estudiante, evaluacion where (estudiante.ced= $ced) &
(estudiante.idEstudiante= evaluacion.idEstudiante)";
        $conn = new Conectar();
        $conn->conectarBD();
        $tabla = $this->ejecutarConsulta($query);
        $conn->desconectarBD();
        return $tabla;
    }
    //Método que devuelve todas las preguntas de un capitulo
    function pregCapitulo($cap)
    {
        $query= "select pregunta, respuesta1, respuesta2, respuesta3,
respuesta4 from preguntasesamen where capitulo= $cap";
        $conn = new Conectar();
        $conn->conectarBD();
        $tabla = $this->ejecutarConsulta($query);
        $conn->desconectarBD();
        return $tabla;
    }
}
?>
```

Código de la clase Agregar Estudiante mostrada en la figura 5.11

FASE DE CONSTRUCCION

```
<link rel="stylesheet" type="text/css" href="css/Estilos.css"
media="screen"/>
<script type="text/javascript" src="Scripts/codigo.js"></script>
```

```
<p>
```

```
<?PHP
```

```
require("Consultas.php");
$guardar= new Consultas();
```

```
?>
```

```
<?PHP
```

```
function generarNickname($nombre, $apellido)
```

```
{
```

```
    $nick= $nombre{0}.$apellido;
```

```
    return $nick;
```

```
} function generarPassword($numcar)
```

```
{ $password = "";
```

```
    $caracteres = "0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ";
```

```
    $i = 0;
```

```
    while ($i < $numcar)
```

```
    {
```

```
        $char = substr($caracteres, mt_rand(0, strlen($caracteres)-1), 1);
```

```
        if(!strstr($password,$char))
```

```
        {
```

```
            $password.= $char;
```

FASE DE CONSTRUCCION

```

                $i++;
            }
        }
        return $password;
    }
?>
<?PHP
if($_POST['Agregar'])
{
    $guardar->guardarEstudiante($_POST['nombre'], $_POST['apellido'],
$_POST['cedula'], $_POST['seccion']);
    $guardar->notaDefault();
    $nick = generarNickname($_POST['nombre'],$_POST['apellido']);
    $clave= generarPassword(10);
    $guardar->guardarUsuario($nick, $clave);
    $sid= $guardar->idUsuarioGen();
    $nroFilas= count($sid);
    $sidU= $sid[$nroFilas-1][0];
    $guardar->guardarId($sidU, $cedula);
}
?></p>
<table width="300" height="182" border="0" align="center" cellpadding="0"
cellspacing="0" class="Estilo1">
<tr>
<td width="114" height="36">Nombre:</td>

```

```

<td width="186"><label>
  <input type="text" name="nombre" id="nombre" />
</label></td>
</tr>
<tr>
  <td height="35">Apellido:</td>
  <td><label>
    <input type="text" name="apellido" id="apellido" />
  </label></td>
</tr>
<tr>
  <td height="39">Cedula:</td>
  <td><label>
    <input type="text" name="cedula" id="cedula" />
  </label></td>
</tr>
<tr>
  <td height="36">Seccion:</td>
  <td><?PHP
    $tabla = $guardar->buscarSeccion();
    $nroFilas= count($tabla);
    $nroColumna= count($tabla[0]);
    echo "<select name=\"seccion\" id=\"seccion\">";
    for ($i=0; $i<$nroFilas; $i++)
    {
      echo "<option>".$tabla[$i][0]."</option>";
    }
  </td>
</tr>

```

FASE DE CONSTRUCCION

```

        }
        ?>
    </td>
</tr>
<tr>
    <td colspan="2" align="center"><label>
        <input type="submit" name="Agregar" id="Agregar" value="Agregar"
/>
    </label></td>
</tr>
</table>
<p>&nbsp;</p>
<table width="297" border="1" align="center" cellpadding="0"
cellspacing="0">
    <tr>
        <td width="126">Nickname:</td>
        <td><label>
            <?PHP
                $campo= " <input type=\"text\" name=\"textfield\"
id=\"textfield\" value=\"\$nick\" readonly/>";
                echo $campo
            ?>
        </label>
        </td>
    </tr><tr>
        <td>Password:</td>

```


FASE DE CONSTRUCCION

```

<td>
<?PHP
    $campo=    "<input    type=\"text\"    name=\"textfield\"
id=\"textfield\" value=\"\$clave\" readonly/>";
    echo $campo;
?>
</td></tr>
</table>

```

Las clase anteriores requieren de la clase Conectar

```

<?PHP
//Clase para realizar la conexion y desconexion con la BD
class Conectar
{
    var $conexion;
    var $servidor = "localhost";
    var $usuario = "root";
    var $pwd = "123456";
    var $bd = "setpoo";

    //Funcion conectar a BD
    function conectarBD()
    {
        $this->conexion = mysql_connect($this->servidor, $this-
>usuario, $this->pwd);
        return mysql_select_db($this->bd, $this->conexion);
    }
}

```

```

//Función desconectar de la BD
function desconectarBD()
{
    return mysql_close($this->conexion); }
}
?>

```

Código interfaz de Profesor Información general mostrada en la figura 5.12, esta clase también hace un llamado a la clase consulta y conectar mostradas anteriormente.

```

<link rel="stylesheet" type="text/css" href="css/Estilos.css"
media="screen"/>
<?PHP
function celdas($nombre, $i)
{
    $ renglon = "<td bgcolor=\"#FFFFFF\" class=\"Estilo2\"
valign=\"center\" align=\"center\"><input name=\"campo$i\"
type=\"text\" size=\"10\" value=\"\".$nombre.\"\" readonly/></td>";
    return $renglon;
}
?>
<table width="389" height="26" border="1" align="center" cellpadding="0"
cellspacing="0" >
<tr class="Estilo3">
<td width="60"><div align="center" >Nombre</div></td>
<td width="63" align="center"><p>Apellido</p> </td>
<td width="59" align="center">Cedula</td>

```

FASE DE CONSTRUCCION

```
<td width="109" align="center">Correo</td>
<td width="86" align="center">Telefono</td>
</tr>
<?PHP
//Realizo consulta a la base de datos
    require("Consultas.php");
    $cons = new Consultas();
    $tabla = $cons->informacionGeneral();
    $nroFilasTabla = count($tabla);
    for($i = 0; $i<$nroFilasTabla; $i++)
    {
        echo"<tr>".celdas($tabla[$i][0], 1)."".celdas($tabla[$i][1],
2)."".celdas($tabla[$i][2], 3)."".celdas($tabla[$i][3], 4)."".celdas($tabla[$i][4], 5);
    }

    //Fin de la tabla
?>
</table>
```

5.4 CONCLUSIÓN DE LA FASE DE CONSTRUCCIÓN

Durante la fase de construcción se complementó el diseño de la arquitectura base de SETPOO, y se codificaron, probaron e integraron todos los componentes del software.

Las pruebas de unidad se aplicaron mediante el método de caja negra, adicionalmente se aplicaron pruebas de integración lo cual permitió detectar y corregir fallas en el funcionamiento de los componentes y en la forma de conectarse unos a otros.

Durante la construcción, se controlaron los riesgos correspondientes a esta fase, logrando el propósito de tener lista la primera versión operativa del sistema.

5.5 PLANIFICACIÓN DE LA FASE DE TRANSICIÓN.

Al final de la fase de construcción se obtuvo una aplicación que cumple con los requisitos expresados inicialmente.

La tarea de la fase de transición para esta aplicación sería: instalación en la sala donde se imparte la asignatura, documentación de las observaciones que surgen durante esta fase.

Debido a que para este trabajo de grado no se estableció como objetivo la puesta en marcha del proyecto, esta fase no se llevara a cabo, sin embargo, dado a su duración, la etapa de instalación y prueba del sistema servirá como punto de partida para otro trabajo de grado.

CAPITULO 6

MANUAL DE USUARIO

6.1. INTRODUCCIÓN

El siguiente manual ha sido diseñado para guiar en el correcto uso y óptimo desempeño del software de enseñanza-aprendizaje SETPOO, a los estudiantes de la asignatura Taller de Programación Orientada a Objetos, los cuales darán uso a esta herramienta que fué creada con el fin de crear un material que sirva de apoyo en el proceso de aprendizaje.

Se espera que sea de completo entendimiento para el lector la descripción de los procedimientos necesarios a llevar acabo para manejar correctamente el sistema, es recomendable que ante cualquier duda se acuda a un Administrador del Sistema, para evitar cualquier conflicto

6.2. INICIO DE SESIÓN

El software SETPOO, cuenta con ciertos niveles de seguridad verificados y validados, por medio de los cuales se lleva un control de las personas que pueden acceder al mismo. Al entrar al sistema se visualiza una pantalla de bienvenida en la cual el usuario podrá empezar a navegar en el sistema, debe ingresar sus datos de inicio de sesión en los cuadros de textos especificados para esto, que vemos en la figura 1, y luego haga click en el botón “Ingresar”. Si se ingresan los datos correctamente se desplegará una pantalla acorde al usuario que se haya ingresado, profesor o estudiante. En caso de que los datos ingresados no sean válidos el sistema mostrará un mensaje de error.

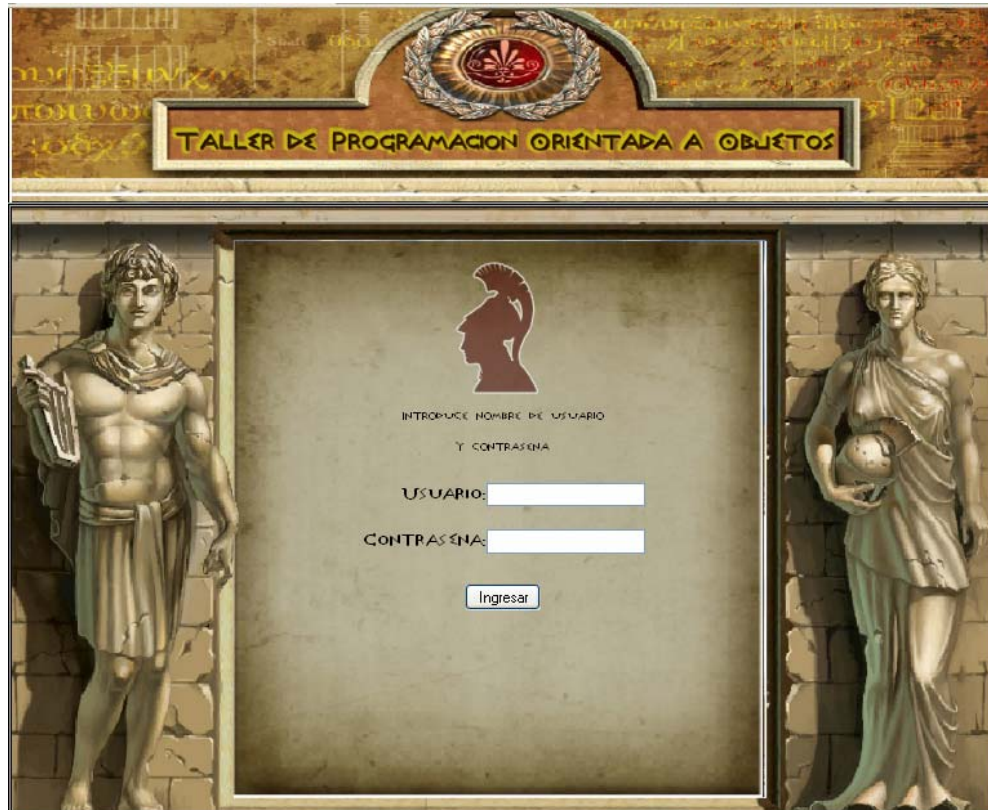


Figura 1. Pantalla de inicio de sesión

6.3 ELEMENTOS ESTANDAR Y DE NAVEGACIÓN

A continuación se explican los formatos básicos de organización y presentación de links para la navegación entre las secciones de la aplicación:

6.3.1. Encabezado

Se encuentra en la parte superior, donde se muestra la identificación del sistema. (Ver figura 2).

6.3.2. Botón “Ayuda”

En cualquier momento que lo necesite, este manual estará disponible para que lo consulte y lo guíe sobre cómo usar la aplicación. Contiene una breve descripción de las acciones que puede realizar en cada sección. Sólo debe presionar el botón “Ayuda” que se encuentra en la parte inferior izquierda (Ver figura 3).



Figura 2. Detalle del encabezado y menú de secciones

6.3.3. Menú de Secciones

Debajo del encabezado se encuentra el menú con links que le permite al usuario visitar las diferentes secciones del sistema. Estos links son contienen el nombre de cada sección, al hacer click en cada uno de ellos se desplegará si es el caso las diferentes opciones que se tienen para cada sección.



Figura 3. Detalle de Botón Ayuda



Figura 4. Detalle del Botón Cerrar Sesión

6.3.5 Botón “Cerrar Sesión”

A través de este botón podrá cerrar su sesión de usuario e ir a la pantalla de login o inicio de sesión. Está ubicado en la parte superior derecha. (Ver figura 7).

6.4. DESCRIPCIÓN DE LAS SECCIONES DE LA APLICACIÓN

Debido a que se manejan dos usuarios distintos, que son estudiante y profesor se tendrán interfaces diferentes para cada uno de ellos. Para entender fácilmente como trabaja cada una de las secciones en las interfaces de la aplicación, es necesario conocer cuáles son las funciones de cada una, además de ciertos aspectos importantes para el manejo correcto de las mismas. A continuación, se describen las acciones que contiene cada sección.

6.4.1 Secciones De La Aplicación Como Estudiante

6.4.1.1 Modificar Usuario

En esta sección el estudiante tiene la posibilidad de modificar sus datos personales, tales como número de teléfono, correo electrónico y la contraseña, al presionar el link se mostrará en pantalla el nombre, apellido, cedula y nombre de usuario y los campos que pueden ser modificados. Ver figura 5.

6.4.1.2 Ver Récord

Esta sección le permite al estudiante conocer el resultado de las evaluaciones que ha presentado y el resultado obtenido en cada una de ellas. Ver figura 6.

6.4.1.2 Resumen

Muestra al estudiante un archivo pdf con un resumen de todo el contenido de la asignatura.

The screenshot shows the user interface for 'TALLER DE PROGRAMACION ORIENTADA A OBJETOS'. The user is logged in as 'Br. Noireth Macayo'. The time is 11:02:01 PM. The 'Modificar Usuario' button is highlighted in yellow. The user profile information is as follows:

Nombre: Noireth
 Apellido: Macayo
 Cedula: 17733552
 Usuario: NMacayo
 Telefono: 0416
 Correo: @hotmail.com

Buttons: Contraseña, Confirmar, Contraseña, Modificar

Navigation menu (left): Entorno de Desarrollo, Tipos de Datos, Entrada y Salida Estándar, Estructuras de Control, Arreglos, Apuntadores, Estructura de Datos, Funciones, Clases y Objetos, Ayuda

Figura 5. Detalle del botón Modificar Usuario

The screenshot shows the user interface for 'TALLER DE PROGRAMACION ORIENTADA A OBJETOS'. The user is logged in as 'Br. Noireth Macayo'. The time is 11:09:45 PM. The 'Ver Record' button is highlighted in yellow. The user profile information is as follows:

Nombre: Noireth
 Apellido: Macayo
 Cedula: 17733552
 Usuario: NMacayo
 Telefono: 0416
 Correo: @hotmail.com

Buttons: Contraseña, Confirmar, Contraseña, Modificar

Navigation menu (left): Entorno de Desarrollo, Tipos de Datos, Entrada y Salida Estándar, Estructuras de Control, Arreglos, Apuntadores, Estructura de Datos, Funciones, Clases y Objetos, Ayuda

NOTA 1	NOTA 2	NOTA 3	NOTA 4	NOTA 5	NOTA 6	NOTA 7	NOTA 8	NOTA 9
8	0	0	0	0	0	0	0	0

Figura 6. Detalle del botón ver Record

6.4.1.3 Menú de navegación por capítulos

En esta sección el estudiante podrá navegar por los capítulos de la asignatura Taller de Programación Orientada a Objetos, este menú está conformado por links con el nombre del capítulo, se encuentra ubicado a la izquierda. Ver figura 6. Para acceder a la información del capítulo se debe hacer click al link donde se desea dirigirse. Luego se mostrará en pantalla la información del capítulo solicitado. Ver figura 7.



Figura 6. Detalle del menú de navegación por capítulos



Figura 7. Detalle del link Arreglos del menú de navegación de capítulos.

6.4.1.4 Menú de navegación por temas

En esta sección se muestra un menú correspondiente a los temas de cada capítulo, este menú está compuesto por botones identificados con el nombre de un tema, ver Figura 8. Además se muestran los botones atrás y siguiente que se utilizan igualmente para navegar por los temas. Ver figura 9.



Figura 8. Detalle de menú de navegación por temas.



Figura 9. Detalle de botones Atras y Siguinte

6.4.1.5 Actividad interactiva.

Para acceder a esta sección el estudiante debe estar navegando por los temas de un capítulo, y seleccionar el botón Actividad interactiva Ver Figura 10, la cual desplegara una nueva pestaña con la actividad correspondiente. Ver figura 11.



Figura 10. Detalle del botón Actividad Interactiva



Figura 11. Detalle Actividad Interactiva

6.4.1 Secciones De La Aplicación Como Profesor

6.4.1.3 Menú de Gestión de Estudiantes

Esta sección le ofrece al profesor la posibilidad de gestionar toda la información referente los estudiantes que estén en el sistema. Este menú está conformado por un conjunto de links ubicados a la izquierda de la pantalla. Ver figura 12.



Figura 12. Detalle del menú de Gestión de Estudiante

6.4.1.4 Agregar estudiante

Esta sección le permite al profesor agregar nuevos estudiantes al software SETPOO, para acceder a ella se debe hacer click en la sección identificada, ver Figura 13, luego se desplegara una pantalla donde el profesor pueda ingresar los datos personales de un nuevo alumno, al presionar el botón guardar se genera una contraseña y un nombre de usuario para el nuevo ingreso. Ver figura 14.

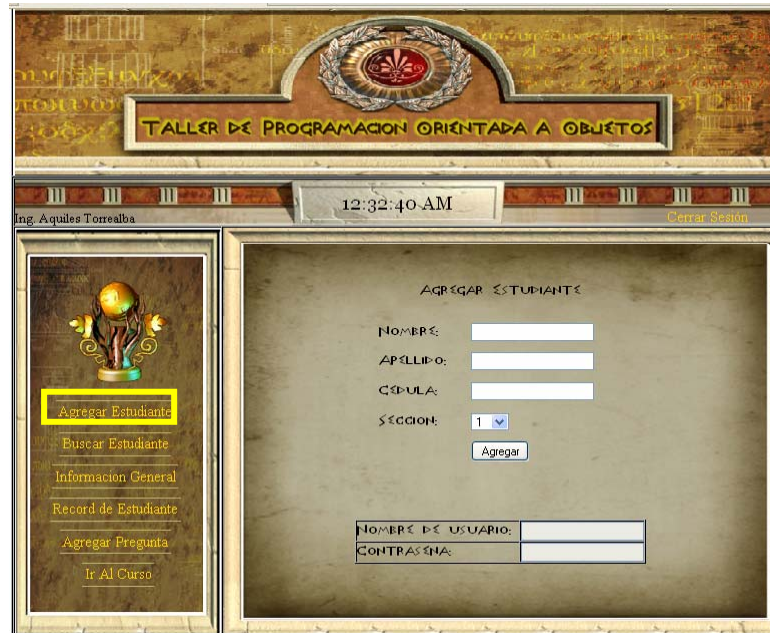


Figura 13. Detalle del link Agregar Estudiante



Figura 14. Detalle del Botón Agregar

6.4.1.5 Buscar Estudiante

Esta sección le permite al profesor buscar a un estudiante en específico por cedula, para acceder a ella se debe hacer click en el link Buscar Estudiante, ver Figura 15, el cual mostrara la interfaz donde el profesor puede ingresar los datos del estudiante que

desea buscar, al presionar el botón buscar, ver Figura 16, se mostrarán los datos personales del estudiantes, así como también las notas de las evaluaciones.

NOTA 1	NOTA 2	NOTA 3	NOTA 4	NOTA 5	NOTA 6	NOTA 7	NOTA 8	NOTA 9

Figura 15. Detalle de Buscar Estudiante

6.4.1.5 Información General

Esta sección le brinda al profesor la opción de visualizar datos personales como nombre, apellido, cedula, teléfono y correo de todos los estudiantes que se encuentran en el sistema. Ver figura 17.

TALLER DE PROGRAMACION ORIENTADA A OBJETOS

Ing. Aquiles Torrealba 12:43:40 AM Cerrar Sesión

BUSCAR ESTUDIANTES

INGRESE INFO DE CÉDULA

NOMBRE: Noireth
 APELLIDO: Macayo
 CÉDULA: 17733552
 TELÉFONO: 04248870443
 CORREO: noirethmacayo@gmail.cc
 SECCIÓN: 1

NOTA 1	NOTA 2	NOTA 3	NOTA 4	NOTA 5	NOTA 6	NOTA 7	NOTA 8	NOTA 9
8	0	0	0	0	0	0	0	0

Figura 16. Detalle Botón Buscar

TALLER DE PROGRAMACION ORIENTADA A OBJETOS

Ing. Aquiles Torrealba 12:47:52 AM Cerrar Sesión

INFORMACION GENERAL DE ESTUDIANTES

NOMBRE	APELLIDO	CÉDULA	CORREO	TELÉFONO
Noireth	Macayo	17733552	noirethmacay	04248870443
Yitzy	Fajardo	17092980		
Valenia	Ramirez	17678970		
ellys	castro	12345		
ana	vrgas	2345		
juana	perez	3577		
Maria	Marin	17890657		

Figura 17. Detalle Información General

6.4.1.6 Record de Estudiantes

Esta sección le permite al profesor visualizar las notas de todos los estudiantes que se encuentran en el software SETPOO, al hacer click en el link Información general, se despliega una lista donde se incluye las cedula junto con las notas de todos los alumnos. Ver Figura 18.

The screenshot shows the SETPOO software interface. At the top, there is a header with the course title "TALLER DE PROGRAMACION ORIENTADA A OBJETOS" and a clock displaying "12:49:56 AM". Below the header, there is a sidebar with navigation options: "Agregar Estudiante", "Buscar Estudiante", "Información General", "Record de Estudiante" (highlighted in yellow), "Agregar Pregunta", and "Ir Al Curso". The main area displays a table titled "MOSTRAR ESTUDIANTE" with columns for "CEDULA" and "NOTA 1" through "NOTA 9". The table contains six rows of student data.

CEDULA	NOTA 1	NOTA 2	NOTA 3	NOTA 4	NOTA 5	NOTA 6	NOTA 7	NOTA 8	NOTA 9
1773355	0	0	0	0	0	0	0	0	0
1709298	0	0	0	0	0	0	0	0	0
1767897	0	0	0	0	0	0	0	0	0
12345	0	0	0	0	0	0	0	0	0
2345	0	0	0	0	0	0	0	0	0
3577	0	0	0	0	0	0	0	0	0
1789065	0	0	0	0	0	0	0	0	0

Ver figura 18. Detalle Información General

6.4.1.7 Agregar Pregunta

Esta sección le permite al profesor agregar preguntas a las evaluaciones, al hacer click en el link identificado como Agregar Pregunta, ver figura 19, se desplegara una interfaz donde el profesor deberá ingresar el capítulo que desea actualizar y posteriormente debe ingresar la pregunta y las respuestas, marcando en un checkbox la respuesta correcta en caso de ser simple o marcar dos checkboxes en caso de que se múltiple.

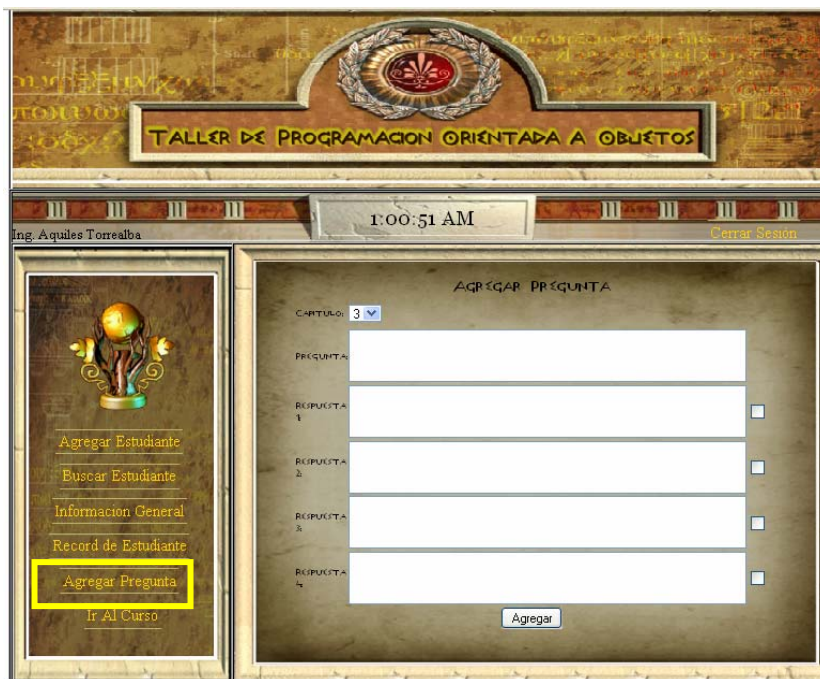


Figura 19. Detalle Agregar Pregunta

6.4.1.8 Ir al Curso

Al igual que los estudiantes, el profesor también tiene la posibilidad de visitar la información de los capítulos, ver Figura 20, con solo hacer click en el link Ir Al Curso, la cual despliega una nueva página con un menú de capítulos. Ver Figura 21. La diferencia entre la opción Ir Al Curso de profesor y navegar por el menú de capítulos del estudiante es que el profesor no puede presentar evaluaciones.



Figura 20. Detalle Ir Al Curso



Figura 21. Pantalla de Navegación por capítulos de Profesor.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

✓ Se considera que el desarrollo de SETPOO es oportuno, por ser una herramienta de estudio actualizada, pues contempla el contenido programático de la cátedra Taller de programación Orientada a Objetos, impartida en la universidad de oriente núcleo Anzoátegui.

✓ La realización de este proyecto facilitara el proceso de enseñanza-aprendizaje para la materia Taller de Programación Orientada a Objetos, ya que será utilizado como material de apoyo para los estudiantes y como sistema de control para los profesores.

✓ Para el desarrollo de SETPOO se empleo el proceso unificado, conjuntamente con la Ingeniería de Software Educativo, los cuales permitieron especificar los modelos computacionales de la aplicación, para establecer la arquitectura del software así como también la realización de un análisis robusto y la aplicación de las teorías sustantivas sobre el aprendizaje y la comunicación humana, obteniendo así un software de alto contenido pedagógico.

✓ Durante la fase de inicio se determinaron los requerimientos específicos necesarios para la elaboración del sistema y posteriormente fueron actualizados en la fase de elaboración, traduciéndose en el punto de partida para la definición de una excelente y completa funcionalidad

✓ Para obtener un resultado satisfactorio al desarrollar un software educativo es necesario aplicar los fundamentos de la metodología propuesta por Galvis (ISE) en algunos de los flujos de trabajo correspondiente a la fase del proceso unificado.

CONCLUSIONES

- ✓ Al desarrollar un software educativo, conocer las necesidades educativas es primordial para solventar aquellas que pueden resolverse mediante una herramienta educativa computarizada.
- ✓ Se desarrollo la estructura del contenido a ser presentado en el MEC acorde a las necesidades educativas definidas.
- ✓ Se estableció una interfaz adecuada, facilitando la comunicación entre el usuario y el material educativo computarizado.
- ✓ Se realizó de manera exitosa la integración, pruebas y documentación de funcionamiento del sistema, se evaluó y depuró el sistema en varias iteraciones, obteniendo un software altamente funcional, al que puede realizarse mantenimiento y actualizaciones.
- ✓ El uso de PHP y MySQL, como lenguaje de programación y Manejador de Base de Datos respectivamente, permitieron el desarrollo del sistema de manera clara y efectiva, debido a que estas tecnologías son fácilmente utilizables en cualquier entorno computacional, puesto a que sus requerimientos para el funcionamiento son mínimos y al ser tecnologías de Software Libre se pueden obtener en la Web.

RECOMENDACIONES

- ✓ Promover el desarrollo de programas educativos a nivel general, a fin de que la inclusión de las tecnologías informáticas en la educación tengan un mayor alcance. Que se pueda contar con programas educativos basados en diferentes temas, de manera que los usuarios/alumnos aprovechen todas las ventajas que este novedoso método de enseñanza les puede brindar.
- ✓ Cuando se desarrolla un software educativo, es necesario que en las fases del análisis y diseño educativo, intervengan expertos en la asignatura sobre la cual estará basado el sistema, esto para garantizar la validez de la información recopilada en estas etapas. Además esta información permitirá establecer mecanismos de resolución adecuados de las necesidades de los usuarios del sistema y apoyar cada una de las fases del desarrollo del software educativo en sólidos principios educativos.
- ✓ Dar mantenimiento y actualizar los contenidos de SETPOO a fin de que los estudiantes tengan cada día la oportunidad de enfrentarse a nuevos retos y se mantengan motivados a la utilización de la aplicación.

BIBLIOGRAFIA

[1] Veracierta T., Gabriela M. y Torres R., Pamela A.,” **Desarrollar un software educativo para facilitar la enseñanza y aprendizaje de la asignatura Análisis y Diseño de Algoritmo específicamente para las unidades: I, II, III y V; dictada en el Departamento de Computación y Sistemas de la Escuela de Ingeniería y Ciencias Aplicadas de la Universidad de Oriente Núcleo de Anzoátegui; utilizando técnicas de diseño e implementación de la ingeniería de software educativo con el enfoque orientado a objetos.**” Tesis (2005).

[2] Finotello G. Lucía C. y Regardiz C. Sulmira del C.,” **Desarrollar un software educativo como herramienta fundamental de apoyo en el proceso enseñanza-aprendizaje de la unidad curricular Introducción a la Ingeniería en Computación de la especialidad de Ingeniería en Computación, utilizando ambiente interactivo multimedia.**”, Tesis (2004).

[3] Velásquez R., Gloribert M. y Bonini R., Juan Carlos A,“**Desarrollo de un Software Educativo para apoyar la enseñanza de la Asignatura Programación I Impartida en el Departamento de Computación y Sistemas de la Universidad de Oriente**”, Tesis (2003).

[6] Wikimedia Foundation, Inc.Aprendizaje. (diciembre de 2008)

<http://es.wikipedia.org/wiki/Aprendizaje>

[7] Salcedo, P. “**Ingeniería de Software Educativo, Teoría y bases que la sustentan**”.(2002)

<http://www.inf.udec.cl/~revista/ediciones/edicion6/isetm.PDF>

[8]Peggy A. Ertmer y Timothy J. Newby. ”**Conductismo, Cognitismo y Construcciónismo**”,1993

http://ares.unimet.edu.ve/programacion/psfase3/modII/biblio/CONDUCTISMO_%20COGNITIVISMO_%20CONSTRUCTIVISMO.pdf

[9] Good T. **“Psicología Educativa Contemporánea”**. McGraw Hill. (5ta. Ed.). Madrid España. (2001).

[10] Gagne R. y Brig. L. **“La planificación de la enseñanza y sus principios”**. Editorial Trillas. México. (1976).

[11] Feroso P. **“Teoría de la educación”**. Ediciones Ceac. España. (1982).

[12] Olier K. **“Efectos de un programa de educación virtual sobre los conocimientos de los docentes acerca del uso de las nuevas tecnologías de la información y comunicación con fines educativos”**.(2001).

<http://www.cibereduca.com/temames/ponencias/sept/p69.htm>

[13]Piattini M. **“Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión”**. Rama Madrid. (1996).

[14]Gómez R., Galvis P. y Mariño O. **“Ingeniería de Software Educativo con Modelaje Orientado por Objetos: un medio para desarrollar micromundos interactivos”**.(1998)

<http://www.minerva.uevora.pt/simposio/comunicacoes/riomezmarino.html>

[15] Pressman R. **“Ingeniería de software”**. McGraw-Hill. (4ta Ed.). España (1998).

[16] Jacobson, I. Booch G. y Rumbaugh J., **“El Proceso Unificado de Desarrollo de Software”**. Primera Edición. Addison Wesley. Madrid. 2000.

[17] Jacobson, I. Booch G. y Rumbaugh J y G. Booch. **“El Lenguaje Unificado de Modelo”**. Primera Edición. Addison-Wesley. Madrid, España. 2000.

[18] Hennicker R., N. Kock y L. Mandel, **“Modeling the user interface of Web applications with UML”**,(2000)

http://www.pst.informatik.uni-muenchen.de/personen_kochn/ExtendingUML.pdf

[19] Galvis P. **“Ingeniería de Software Educativo”**. Ediciones Unidas. Primera Edición. Santa fe de Bogotá Colombia.1992.

METADATOS PARA TRABAJOS DE GRADO

TÍTULO	“Desarrollo de un Software Educativo en entorno Web para la asignatura de Taller de Programación Orientada a Objetos, impartida en el Departamento de Computación y Sistemas Escuela de Ingeniería y Ciencias Aplicadas de la Universidad de oriente Núcleo Anzoátegui”
SUBTÍTULO	

AUTOR (ES):

APELLIDOS Y NOMBRES	CÓDIGO CULAC / E MAIL
Fajardo S., Yitzy R.	CVLAC: V-17.092.980 E MAIL: yitzyfajardo@hotmail.com
Macayo M., Noireth J.	CVLAC: V- 17.733.552 E MAIL: noireth_m2@hotmail.com
	CVLAC: E MAIL:
	CVLAC: E MAIL:

PALÁBRAS O FRASES CLAVES:

ENSEÑANZA - APRENDIZAJE _____

ACTIVIDAD DIDACTICA _____

SISTEMA _____

MICROMUNDO _____

WEB _____

**METADATOS PARA TRABAJOS DE GRADO, TESIS Y
ASCENSO:**

ÀREA	SUBÀREA
Ingeniería y Ciencias Aplicadas	Ingeniería en Computación

RESUMEN (ABSTRACT):

El presente proyecto consiste en el desarrollo de un software educativo interactivo basado en entorno web, creado como herramienta de apoyo para la enseñanza y el aprendizaje de la unidad curricular “Taller de Programación Orientada a Objetos”, perteneciente al pensum de Ingeniería en Computación, de la Universidad de Oriente. Basándose en los principios de un curso virtual que proporcionará una nueva alternativa para suministrar el material de instrucción, que buscará el interés del estudiante hacia el contenido del software, tomando en cuenta las teorías del aprendizaje. Para la elaboración del software se empleo el Proceso Unificado de Desarrollo de Software, aunado a la metodología de Ingeniería de Software Educativo de Álvaro Galvis, La visualización del contenido de la asignatura se realiza a través de textos y animaciones alusivas a los temas que conforman la asignatura, el sistema también cuenta con un sistema de evaluaciones que le permite al estudiante verificar el grado de asimilación alcanzado. El objetivo es crear un material que sirva de apoyo para el proceso de enseñanza-aprendizaje del profesor y el estudiante.

METADATOS PARA TRABAJOS DE GRADO, TESIS Y**ASCENSO:****CONTRIBUIDORES:**

APELLIDOS Y NOMBRES	ROL / CÓDIGO CVLAC / E_MAIL				
Torrealba, Aquiles	ROL	CA	AS	TU X	JU
	CVLAC:	V-			
	E_MAIL				
	E_MAIL				
García, Zulirais.	ROL	CA	AS	TU	JU X
	CVLAC:	V-10.299.576			
	E_MAIL	zzzuliii@hotmail.com			
	E_MAIL				
Dorta, Pedro	ROL	CA	AS	TU	JU X
	CVLAC:	V-			
	E_MAIL				
	E_MAIL				
	ROL	CA	AS	TU	JU
	CVLAC:	V-			
	E_MAIL				
	E_MAIL				

FECHA DE DISCUSIÓN Y APROBACIÓN:

2009	08	04
AÑO	MES	DÍA

LENGUAJE.SPA

METADATOS PARA TRABAJOS DE GRADO, TESIS Y**ASCENSO:****ARCHIVO (S):**

NOMBRE DE ARCHIVO	TIPO MIME
TESIS.SETPOO.doc	Aplication/msword

CARACTERES EN LOS NOMBRES DE LOS ARCHIVOS: A B C D E F G H I J K L
M N O P Q R S T U V W X Y Z. a b c d e f g h i j k l m n o p q r s t u v w x
y z. 0 1 2 3 4 5 6 7 8 9.

ALCANCE

ESPACIAL: UDO-ANZOATEGUI (OPCIONAL)

TEMPORAL: 1 Año

TÍTULO O GRADO ASOCIADO CON EL TRABAJO:

 Ingeniero en Computación

NIVEL ASOCIADO CON EL TRABAJO:

 Pre-Grado

ÁREA DE ESTUDIO:

 Departamento de Computación y Sistemas

INSTITUCIÓN:

 Universidad de Oriente – Núcleo de Anzoátegui

**METADATOS PARA TRABAJOS DE GRADO, TESIS Y
ASCENSO:**

DERECHOS

De acuerdo con el artículo 44 del reglamento de trabajo de grado:
"Los trabajos de grado son de exclusiva propiedad de la Universidad de Oriente y sólo podrán ser utilizados a otros fines con el consentimiento del consejo de núcleo respectivo, quien lo participará al Consejo Universitario".

AUTOR
Fajardo S., Yitzy R.

AUTOR
Macayo M., Noireth J.

AUTOR

TUTOR
Torrealba, Aquiles

JURADO
García, Zulirais

JURADO
Dorta, Pedro

POR LA SUBCOMISIÓN DE TESIS