



BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

FACULTAD DE CIENCIAS DE LA COMPUTACIÓN

"SIMULACION DE EXAMENES PARA PREPARACIÓN
DEL EXAMEN DEL CENEVAL"

TESIS PROFESIONAL

QUE PARA OBTENER EL TITULO DE:

LICENCIADO EN CIENCIAS DE LA COMPUTACION

PRESENTA:

AIDA CARMONA LUNA

ASESORA:

M.C. MELIZA CONTRERAS GONZALEZ

PUEBLA PUE.

Índice

CAPITULO I

Marco Teórico.....	6
1.1 ¿Qué es el CENEVAL?.....	6
1.1.1 Tipos de Exámenes.....	6
1.1.2 ¿Qué es el EGEL?.....	6
1.1.3 Beneficios.....	7
1.1.4 Acreditación.....	8
1.2 EGEL CCO Examen General para el Egreso de la Licenciatura.....	8
1.2.1 Características.....	9
1.2.2 Especificaciones Técnicas del Instrumento.....	9
1.2.3 Ficha Técnica.....	10
1.2.4 Estructura del EGEL CCO.....	11
1.2.5 Estructura del EGEL ICO.....	11
1.3 Ingeniería de Software.....	12
1.3.1 Metodología.....	12
1.3.2 Modelos de Desarrollo de Software.....	14
1.3.2.1 Modelo en Cascada.....	14
1.3.2.2 Modelo en Espiral.....	15
1.3.2.3 Modelo de Construcción de Prototipos.....	17
1.3.2.4 Modelo de Desarrollo Evolutivo.....	18
1.3.2.5 Modelo Incremental.....	19
1.3.2.6 Modelo RAD o DRA.....	20
1.4 Lenguaje Unificado de Modelado UML.....	23
1.4.1 Elementos de UML.....	23
1.5 Sistema de Base de Datos.....	28
1.5.1 Componentes de un Sistema de Base de Datos.....	28
1.5.2 Sistema Gestor de Base de Datos.....	28
1.6 Modelo de Datos.....	30
1.6.1 Modelo Entidad-Relación.....	31
1.6.2 Diagrama Entidad-Relación.....	32
1.6.3 Modelo Relacional.....	33
1.7 La Normalización.....	33
1.7.1 Primera Forma Normal 1FN.....	33
1.7.2 Segunda Forma Normal 2FN.....	33
1.7.3 Forma Normal de Boyce Codd (FNBC).....	34
1.7.4 Tercera Forma Normal 3FN.....	34
1.7.5 Cuarta Forma Normal 4FN.....	34
1.7.6 Quinta Forma Normal 5FN.....	34
1.8 MySQL.....	34
1.9 Lenguajes de Programación Web.....	35
1.9.1 Lenguaje HTML.....	36
1.9.2 Lenguaje Javascript.....	36
1.9.3 Lenguaje PHP.....	37
1.9.4 Lenguaje ASP.....	38
1.9.5 Lenguaje ASP. NET.....	38

1.9.6 Lenguaje JSP.....	39
CAPITULO II	
Análisis del Sistema.....	41
2.1 Introducción.....	41
2.2 Definición de Requerimientos.....	41
2.3 Planteamiento de Problema.....	42
2.3 Requerimientos Funcionales del Sistema.....	42
2.4 Modelo de Casos de Uso.....	43
2.4.1 Diagrama de Casos de Uso.....	43
2.4.3 Descripción de Casos de Uso.....	45
CAPITULO III	
Diseño del Sistema.....	50
3.1 Diseño Conceptual.....	50
3.1.1 Modelo Entidad-Relación.....	50
3.1.2 Descripción de Entidades.....	51
3.1.3 Descripción de los Atributos.....	52
3.2 Normalización.....	53
3.2.1 Tabla Áreas.....	53
3.2.2 Tabla Evaluación.....	54
3.2.3 Tabla Examen.....	54
3.2.4 Tabla Opciones.....	55
3.2.5 Tabla Planteamientos.....	55
3.2.6 Tabla Preguntas.....	56
3.2.7 Tabla Rangos.....	56
3.2.8 Tabla Subáreas.....	57
3.2.9 Tabla Usuarios.....	57
CAPITULO IV	
Implementación y Pruebas.....	58
4.1 Implementación del Sistema.....	58
4.1.1 Página Principal.....	58
4.1.1.1 Liga Acceso Administrador.....	59
4.1.1.2 Liga Acceso Coordinador.....	59
4.1.1.3 Acceso Usuario.....	60
4.1.1.4 Registro Nuevos Usuarios.....	60
4.1.2 Acceso Administrador.....	60
4.1.2.1 Página de Administrador.....	61
4.1.2.2 Muestra Usuarios.....	62
4.1.2.3 Seleccionar el Tipo de Pregunta.....	63
4.1.2.4 Creación de Pregunta Simple.....	63
4.1.2.5 Creación de Pregunta Multirreactivo.....	64
4.1.3 Acceso Coordinador.....	66
4.1.3.1 Página de Coordinador.....	66
4.1.4 Registro de Datos de Usuarios.....	67
4.1.5 Página de Bienvenida del Usuario.....	68
4.2 Pruebas del Sistema.....	69
Conclusiones y Perspectivas.....	78

INTRODUCCIÓN

El presente trabajo de tesis titulado "Simulación de Pruebas para Preparación del Examen del CENEVAL", tiene como propósito que los alumnos mejoren su preparación para presentar tal prueba además de que practiquen y mejoren sus habilidades. La acreditación de esta prueba es muy importante para los alumnos ya que actualmente algunas universidades están optando por que los alumnos realicen este tipo de pruebas con el fin de que obtengan el título de la Licenciatura que estén cursando, y en la Benemérita Universidad Autónoma de Puebla, particularmente en la Facultad de Ciencias de la Computación, la acreditación de este examen es la pauta para obtener el Título tanto de Licenciatura como de Ingeniería.

Este proyecto será realizado siguiendo la metodología del modelo de desarrollo de software denominado Modelo en Cascada, que es el modelo tradicional para la implementación de software, el cual sugiere la realización de las etapas de la siguiente forma: Análisis de Requerimientos, Diseño, Codificación, Implantación Pruebas y Mantenimiento.

Durante la fase de Análisis de Requerimientos se definirá el problema, los requerimientos funcionales del sistema y se realizarán los respectivos diagramas para una mejor interpretación de los mismos.

En la parte del diseño se realizará el diagrama Entidad-Relación para así mapearlo al Modelo Relacional y continuar con la descripción de las entidades, que forman parte fundamental del sistema, los atributos y las relaciones existentes; para así proseguir con la normalización, parte importante que debe ser tomada en cuenta para el buen funcionamiento de una base de datos.

La fase de la implantación o implementación es la parte en la que se lleva a la realidad todo lo que se ha planteado en las fases anteriores, ya que es donde se codifica el sistema, en este caso utilizaremos varias herramientas, MySQL como motor de Bases de Datos, PHP como lenguaje intermedio entre la interfaz y la base de datos.

De acuerdo con el modelo continuaremos con la fase de las pruebas, en donde a través de algunos ejemplos se podrá verificar la funcionalidad del sistema, esta fase es muy importante ya que durante su ejecución se podrá corroborar que cumple con los requerimientos que se plantearon durante la fase del análisis.

La etapa de mantenimiento en este caso será omitida ya que esta solo se puede realizar cuando el sistema ha estado en funcionamiento durante algún tiempo, es decir, esta parte no está considerada dentro de los lineamientos que se tomaron durante el análisis.

ESTRUCTURA DEL DOCUMENTO

Capítulo I

En el capítulo I titulado Marco Teórico se hace un breve resumen de lo que es el CEVEVAL y como esta conformado el examen para la acreditación de la Ingeniería y Licenciatura en Ciencias de la Computación que esta institución ofrece, así mismo se definen algunos términos de Ingeniería de Software, Bases de datos y lenguajes de programación Web, ya que haciendo uso de alguno de estos fue como se implemento este proyecto.

Capítulo II

En este capitulo llamado Análisis del Sistema se encuentran el planteamiento del problema y la definición de los requerimientos del mismo, así como el diagrama general del sistema y la descripción de los casos de uso del mismo.

Capítulo III

Este capitulo denominado Diseño del Sistema esta enfocado principalmente al diseño conceptual, diseño lógico y diseño físico del sistema, y para realizar cada una de estas etapas de realizo el Diagrama Entidad- Relación ya que con el se pudo implementar la base de datos, se realizo también la descripción de entidades y se procedió a la normalización de las mismas asegurando así el correcto funcionamiento del sistema.

Capítulo IV

El Capitulo IV llamado Implementación y Pruebas es donde se realizo la implementación del sistema, ayudado por el lenguaje de programación Web, también se muestra la interfaz del sistema y se describen sus elementos, con lo cual se pudieron realizar las pruebas con algunos ejemplos, evaluando así el funcionamiento del proyecto.

Para finalizar se describen algunas perspectivas y conclusiones que se obtuvieron de la realización del sistema.

CAPITULO I

Marco Teórico

1.1 ¿QUÉ ES EL CENEVAL?

El Centro Nacional de Evaluación para la Educación Superior (Ceneval) se construyó el 28 de abril de 1994, a partir de una recomendación de la Asamblea General de la ANUIES. Como asociación civil, el Ceneval es un organismo privado que busca autofinanciarse. Su propósito consiste en ofrecer a las instituciones educativas, públicas y privadas, diversos servicios de evaluación, en particular exámenes de ingreso a bachillerato, licenciatura y postgrado, así como exámenes de egreso para diversas carreras, que se llamaron inicialmente de Calidad Profesional y ahora de Egreso de la Licenciatura.

Para las instituciones las ventajas de estos exámenes son claras: en el aspecto técnico resuelven la dificultad de elaboración de pruebas estandarizadas que miden los conocimientos de los estudiantes; también se aprecian como instrumentos que permiten la comparación de resultados entre instituciones.

Los resultados del examen son muy importantes tanto para la universidad como para el alumno. Para la institución indica que tan preparados están saliendo los alumnos de las diferentes carreras y le permite hacer ajustes a sus planes y programas de acuerdo a la demandas de los empleadores.

Para los alumnos los resultados obtenidos de la sustentación del examen del CENEVAL sirve como un importante indicador de las competencias adquiridas durante la carrera. El examen sirve también como una carta de presentación al momento de buscar empleo [P1].

1.1.1 TIPOS DE EXÁMENES

El **CENEVAL** desarrolla, principalmente, **dos tipos de exámenes:**

Los Nacionales de Ingreso (EXANI), que evalúan las habilidades y competencias fundamentales, así como los conocimientos indispensables que debe tener quien aspira a continuar sus estudios de educación media superior y superior, así como los Exámenes Generales para el Egreso de la Licenciatura (EGEL), que evalúan los conocimientos y la información indispensables que debe mostrar un recién egresado de los estudios de licenciatura.

1.1.2 ¿QUÉ ES EL EGEL?

El EGEL es una prueba nacional, especializada por carrera profesional, que tiene como objetivo identificar la medida en que los egresados de la Licenciatura cuentan con los conocimientos y habilidades esenciales para el inicio del ejercicio profesional en el país.

Está orientado a determinar si los egresados son capaces de utilizar lo que han estudiado y aprendido en su licenciatura, en situaciones similares a las que enfrentarán en el ejercicio profesional.

La realización y actualización de los EGEL en operación, implica la puesta en práctica de un complejo proceso, metodológico, técnico y de generación de consensos en el que participan más de 600 expertos de IES, colegios y asociaciones profesionales e instancias empleadoras de todo el país, que integran los Consejos Técnicos de los distintos EGEL.

1.1.3 BENEFICIOS

Para el alumno:

- El aspirante presenta un examen confiable y válido, probado a nivel nacional, lo cual garantiza que el resultado representa el nivel académico de cada sustentante.
- Que el recién egresado cuente con un comprobante de sus conocimientos académicos expedido por una instancia externa a la institución donde haya realizado sus estudios y que sirva de apoyo a su currículum vitae.
- Conocer a detalle las características y alcance de su formación en relación con el perfil referencial de validez para su campo de ejercicio profesional.
- Aumentar la probabilidad de lograr una inserción rápida y adecuada en el mercado de trabajo.

Para la universidad:

- Proporciona a las instituciones de educación superior elementos de juicio para realizar un proceso confiable, preciso y objetivo para apoyar procesos de planeación y evaluación curricular que les permita orientar acciones que tiendan a mejorar la formación académica de los profesionales, adecuando los planes y programas de estudio.
- Informar a los principales agentes educativos (autoridades, instituciones, maestros, estudiantes y sociedad en general) acerca del estado que guardan individuos y poblaciones, respecto de las habilidades y conocimientos considerados esenciales para insertarse al campo laboral.
- Contar con una modalidad más de titulación (en las instituciones cuya reglamentación lo permita), ya vigente en algunas instituciones públicas y privadas.
- Integrar un elemento que sirva de parámetro para adoptar medidas que favorezcan el incremento de la calidad de la educación profesional de los egresados.

Para los empleadores:

- Conocer con mayor precisión el perfil profesional de los candidatos a emplear mediante elementos objetivos de juicio.
- Conocer la calidad académica de los egresados que inician su ejercicio profesional.
- Incorporar elementos de juicio respecto al desempeño esperado de los egresados que apoyen la toma de decisiones de índole laboral.

- Contar con recursos humanos de calidad profesional, de acuerdo con las necesidades nacionales [P1].

1.1.4 ACREDITACION

Para acreditar dicha prueba se deben cubrir los siguientes requisitos.

Los sustentantes que obtengan en el resultado global el puntaje en el dominio satisfactorio, el cual corresponde a por lo menos 1000 puntos, y al menos 1000 puntos en las áreas, Ciencias de la Ingeniería y en otra de las áreas, Ciencias Básicas o Ingeniería Aplicada, se les otorgará un Testimonio de Desempeño Satisfactorio (TDS)¹.

Los sustentantes que obtengan en el resultado global el puntaje en el dominio sobresaliente, el cual corresponde a por lo menos 1150 puntos, y al menos 1000 puntos en cada una de las áreas, Ciencias Básicas, Ciencias de la Ingeniería e Ingeniería Aplicada, se les otorgará un Testimonio de Desempeño Sobresaliente (TDSS)².

1. Hasta el 2003, llamado Testimonio de Desempeño Académico Satisfactorio (TDAS).

2. Hasta el 2003, llamado Testimonio de Alto Rendimiento Académico (TARA).

▪ .

1.2 EGEL CCo EXAMEN GENERAL PARA EL EGRESO DE LA LICENCIATURA EN CIENCIAS COMPUTACIONALES

Es una prueba de cobertura nacional que evalúa el nivel de conocimientos, habilidades y destrezas académicas de los egresados de la Licenciatura en Ciencias Computacionales

El examen es elaborado, bajo la coordinación del Ceneval, por un Consejo Técnico integrado por representantes de la Asociación Nacional de Instituciones de Educación en Informática, A.C., M.I. Consultoría Empresarial, S.C., el CoSNET Consejo del Sistema Nacional de Educación Tecnológica - SEP, la Cámara Nacional de la Industria de la Transformación Sector de Tecnologías de Información e instituciones de educación superior que forman el mayor número de licenciados e ingenieros en Ciencias Computacionales en el país.

El Consejo Técnico elabora el perfil referencial de validez, sobre la base de los planes y programas de estudio vigentes en el ámbito nacional, a partir del cual se diseña el examen, y garantiza la pertinencia, validez y confiabilidad del instrumento desarrollado.

Asimismo, cuenta con la participación y asesoría de un Comité Académico, integrado por un grupo de expertos de reconocido prestigio en el campo de la Ciencias de la Computación del país.

1.2.1 CARACTERÍSTICAS

Es un examen cuya naturaleza es el resultado de la decisión voluntaria, personal o institucional y es independiente de los procesos de evaluación o certificación que realizan las instituciones de educación superior.

Los resultados son confidenciales (únicamente el sustentante y la institución los conocen) y se precisan en una constancia que muestra el desempeño del sustentante en cada área evaluada, sin expresiones aprobatorias o reprobatorias.

No condiciona la expedición del título ni de la cédula profesional. Para efectos de titulación, en su caso, cada centro educativo establece el nivel requerido para ello.

Puede presentarse tantas veces como se desee, siempre y cuando se cumpla con los requisitos de inscripción.

Cuenta con una GUÍA DE EXAMEN que describe detalladamente el proceso de evaluación, con recomendaciones para la solución exitosa del examen y ejemplos de preguntas [P1].

1.2.2 ESPECIFICACIONES TÉCNICAS DEL INSTRUMENTO

Es un examen objetivo: tiene criterios de calificación unívocos y precisos, lo cual permite realizar procesos de calificación rápidos y confiables por medio de sistemas automatizados; ello es indispensable cuando se requiere evaluar a decenas de miles de sustentantes y ofrecer resultados rápidamente.

Es un examen estandarizado: cuenta con reglas fijas de diseño, elaboración y aplicación, además de que los sustentantes se enfrentan a las mismas preguntas o a preguntas equivalentes y se les evalúa bajo las mismas reglas, lo que garantiza mayor objetividad en la calificación.

El examen es de ejecución máxima: exige del sustentante su máximo rendimiento en la tarea o tareas que se le piden que ejecute, contiene reactivos de diferentes grados de dificultad y tiene un tiempo límite suficiente para contestar el instrumento en su totalidad.

Es un examen de opción múltiple (cada pregunta o problema se acompaña de tres distractores verosímiles y una respuesta correcta): que evalúa las áreas o competencias establecidas por el Consejo Técnico como esenciales y significativas a la formación del Licenciado en Ciencias Computacionales en el país; acordadas como comunes, entre las instituciones formadoras participantes y reconocidas como altamente sensibles para medir la calidad de la formación profesional en el área.

El Consejo Técnico acordó que el examen sea de naturaleza criterial para calificar el logro de las áreas incluidas en él, lo que permite comparar el puntaje obtenido por el sustentante con los estándares de calidad establecidos.

Evalúa resultados de aprendizaje (no se refiere a los elementos ni a los procesos para lograr ese aprendizaje).

Su diseño se basa en el perfil referencial de validez, establecido por los cuerpos colegiados del EGEL-CCo, ampliamente representativos de las instituciones de educación superior públicas y privadas que ofrecen la Licenciatura en Ciencias Computacionales en el país.

Dicho perfil referencial de validez ha sido organizado por áreas y subáreas haciendo uso de una tipología taxonómica de los procesos cognitivos.

Está integrado por preguntas y problemas de opción múltiple que evalúan los conocimientos y las habilidades de resolución de problemas en las áreas en Ciencias Computacionales.

1.2.3 FICHA TÉCNICA

- **ÁREAS DE CIENCIAS COMPUTACIONALES:**

Licenciaturas en Ciencias Computacionales y demás carreras afines.

- **ÁREAS DEL CONOCIMIENTO:**

El examen evalúa las áreas, Entorno Social, Matemáticas, Hardware, Software y Tratamiento de Información e Interacción, de los Modelos Curriculares de la Asociación Nacional de Instituciones de Educación en Informática, A. C.

- **NÚMERO DE REACTIVOS:**

270. Adicionalmente se puede incluir un porcentaje de reactivos en situación piloto que no son considerados para efectos de calificación del sustentante.

- **NÚMERO DE SESIONES Y DURACIÓN:**

3 sesiones de cuatro horas cada una, en dos días

- **FECHA DE INICIO DE OPERACIÓN:**

Septiembre de 1999

1.2.4 ESTRUCTURA DEL EGEL – CCO

ÁREA / Subárea	Número de Reactivos	DISTRIBUCIÓN DE REACTIVOS POR SESIÓN		
		1a.	2a.	3a.
ENTORNO SOCIAL	30			
Las organizaciones		15		
Unidades de informática, ética y legislación		15		
MATEMÁTICAS	68			
Matemáticas Básicas		20		
Matemáticas aplicadas y computacionales		48		
HARDWARE	53			
Física y electrónica				13
Arquitectura de computadoras			13	
Redes			27	
SOFTWARE	78			
Software de base				25
Programación				40
Ingeniería de Software				13
TRATAMIENTO DE INFORMACIÓN E INTERACCIÓN	41			
Tratamiento de información			20	
Interacción hombre-máquina			21	
TOTAL:	270	98	81	91

Figura 1.1 Estructura EGEL CCO

1.2.5 ESTRUCTURA DEL EGEL – ICO

ÁREA / Subárea	Número de Reactivos	DISTRIBUCIÓN DE REACTIVOS POR SESIÓN		
		1a.	2a.	3a.
ENTORNO SOCIAL	30			
Las organizaciones			15	
Unidades de informática, ética y legislación			15	
MATEMÁTICAS	47			
Matemáticas Básicas			20	
Matemáticas aplicadas y computacionales			27	
HARDWARE	86			
Física y electrónica				31
Arquitectura de computadoras				16
Redes				39
SOFTWARE	77			
Software de base		29		
Programación		34		
Ingeniería de Software		14		
TRATAMIENTO DE INFORMACIÓN E INTERACCIÓN	30			
Tratamiento de información		14		
Interacción hombre-máquina			16	
TOTAL:	270	91	93	86

Figura 1.2 Estructura EGEL ICO

1.3 INGENIERIA DE SOFTWARE

La Ingeniería de software designa el conjunto de técnicas destinadas a la producción de un programa de computadora, más allá de la sola actividad de programación. Forman parte de esta disciplina las ciencias computacionales y el manejo de proyectos, entre otros campos, propios de la rama más genérica denominada Ingeniería informática.

El software es el conjunto de instrucciones que permite al hardware de la computadora desempeñar trabajo útil. En las últimas décadas del siglo XX, las reducciones de costo en hardware llevaron a que el software fuera un componente que participa en muchos de los dispositivos usados por las sociedades industrializadas [15].

1.3.1 METODOLOGÍA

Un objetivo de décadas ha sido el encontrar procesos o metodologías predecibles y repetibles que mejoren la productividad y la calidad.

Etapas del proceso

La ingeniería de software requiere llevar a cabo numerosas tareas, dentro de etapas como las siguientes:

Análisis de requisitos

Extraer los requisitos de un producto de software es la primera etapa para crearlo. Mientras que los clientes piensan que ellos saben lo que el software tiene que hacer, se requiere de habilidad y experiencia en la ingeniería de software para reconocer requisitos incompletos, ambiguos o contradictorios. El resultado del análisis de requisitos con el cliente se plasma en el documento ERS, *Especificación de Requerimientos del Sistema*, cuya estructura viene definida por varios estándares. Asimismo, se define un diagrama de Entidad/Relación, en el que se plasman las principales entidades que participarán en el desarrollo del software.

La captura, análisis y especificación de requisitos (incluso pruebas de ellos), es una parte crucial; de esta etapa depende en gran medida el logro de los objetivos finales. Se han ideado modelos y diversos procesos de trabajo para estos fines. Aunque aun no está formalizada, ya se habla de la Ingeniería de Requisitos.

Especificación

Es la tarea de describir detalladamente el software a ser escrito, en una forma matemáticamente rigurosa. En la realidad, la mayoría de las buenas especificaciones han sido escritas para entender y afinar aplicaciones que ya estaban desarrolladas. Las especificaciones son más importantes para las interfaces externas, que deben permanecer estables.

Diseño y arquitectura

Se refiere a determinar como funcionará de forma general sin entrar en detalles. Consiste en incorporar consideraciones de la implementación tecnológica, como el hardware, la red, etc. Se definen los Casos de Uso para cubrir las funciones que realizará el sistema, y se transforman las entidades definidas en el análisis de requisitos en clases de diseño, obteniendo un modelo cercano a la programación orientada a objetos.

Programación

Reducir un diseño a código puede ser la parte más obvia del trabajo de ingeniería de software, pero no es necesariamente la porción más larga. La complejidad y la duración de esta etapa está íntimamente ligada al o a los lenguajes de programación utilizados.

Pruebas

Consiste en comprobar que el software realice correctamente las tareas indicadas en la especificación. Una técnica de prueba es probar por separado cada módulo del software, y luego probarlo de forma integral, para así llegar al objetivo. Se considera una buena practica el que las pruebas sean efectuadas por alguien distinto al desarrollador que la programó, idealmente un área de pruebas; sin perjuicio de lo anterior el programador debe hacer sus propias pruebas. En general hay dos grandes formas de organizar un área de pruebas, la primera es que esté compuesta por personal inexperto y que desconozca el tema de pruebas, de esta forma se evalúa que la documentación entregada sea de calidad, que los procesos descritos son tan claros que cualquiera puede entenderlos y el software hace las cosas tal y como están descritas. El segundo enfoque es tener un área de pruebas conformada por programadores con experiencia, personas que saben sin mayores indicaciones en que condiciones puede fallar una aplicación y que centren su atención en detalles que personal inexperto no consideraría.

Documentación

Todo lo concerniente a la documentación del propio desarrollo del software y de la gestión del proyecto, pasando por modelaciones (UML), diagramas, pruebas, manuales de usuario, manuales técnicos, etc., todo con el propósito de eventuales correcciones, usabilidad, mantenimiento futuro y ampliaciones al sistema.

Mantenimiento

Mantener y mejorar el software para enfrentar errores descubiertos y nuevos requisitos. Esto puede llevar más tiempo incluso que el desarrollo inicial del software. Alrededor de 2/3 de toda la ingeniería de software tiene que ver con dar mantenimiento. Una pequeña parte de este trabajo consiste en arreglar errores, o bugs. La mayor parte consiste en extender el sistema para hacer nuevas cosas [L6].

1.3.2 MODELOS DE DESARROLLO DE SOFTWARE

La ingeniería de software cuenta con varios modelos o paradigmas de desarrollo en los cuales se apoya para la realización de software, de los cuales se destacan éstos por ser los más utilizados y los más completos:

- Modelo en cascada o Clásico (modelo tradicional)
- Modelo en espiral (modelo evolutivo)
- Modelo de prototipos
- Modelo evolutivo
- Modelo Incremental
- RAD (Rapid Application Development)

1.3.2.1 MODELO EN CASCADA

El desarrollo en cascada, también llamado modelo en cascada, es el enfoque metodológico que ordena rigurosamente las etapas del ciclo de vida del software, de forma tal que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior, ver figura 1.1

Un ejemplo de una metodología de desarrollo en cascada es:

1. Análisis de requisitos
2. Diseño del Sistema
3. Diseño del Programa
4. Codificación
5. Pruebas
6. Implantación
7. Mantenimiento

De esta forma, cualquier error de diseño detectado en la etapa de prueba conduce necesariamente al rediseño y nueva programación del código afectado, aumentando los costos del desarrollo. La palabra *cascada* sugiere, mediante la metáfora de la fuerza de la gravedad, el esfuerzo necesario para introducir un cambio en las fases más avanzadas de un proyecto.

Si bien ha sido ampliamente criticado desde el ámbito académico y la industria, sigue siendo el paradigma más seguido al día de hoy [L6].

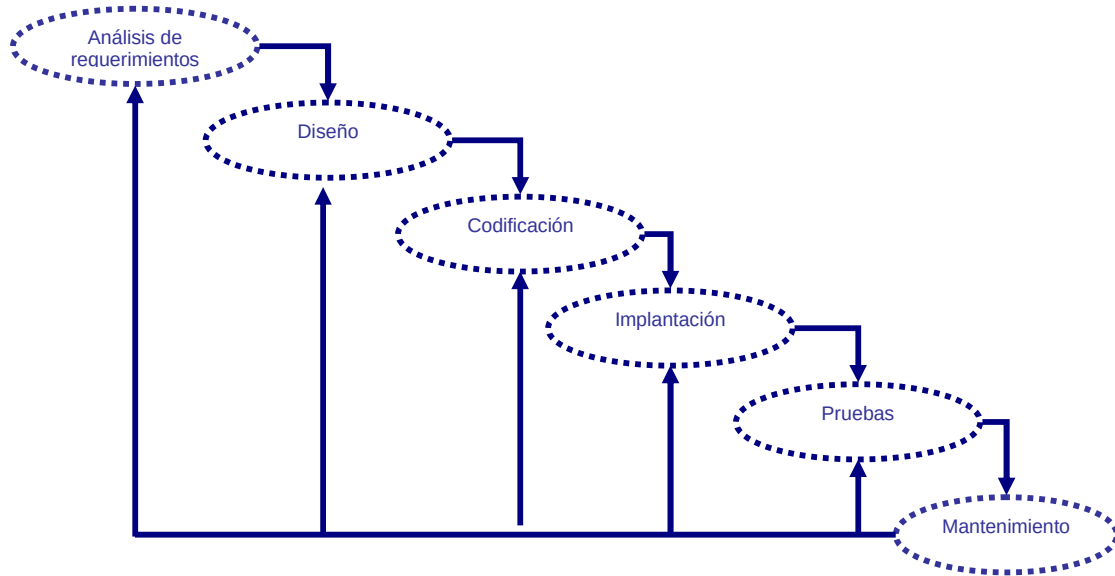


Figura 1.3 Modelo en cascada

1.3.2.2 MODELO EN ESPIRAL

El Desarrollo en Espiral es un modelo de ciclo de vida desarrollado por Barry Boehm en 1985, utilizado generalmente en la Ingeniería de software. Las actividades de este modelo son una espiral, cada bucle es una actividad. Las actividades no están fijadas a prioridad, sino que las siguientes se eligen en función del análisis de riesgo, comenzando por el bucle interior.

Para cada actividad habrá cuatro tareas:

Determinar o fijar objetivos

- Fijar también los productos definidos a obtener: requerimientos, especificación, manual de usuario.
- Fijar las restricciones.
- Identificación de riesgos del proyecto y estrategias alternativas para evitarlos.
- Hay una cosa que solo se hace una vez: planificación inicial o previa.

Análisis del riesgo

- Se estudian todos los riesgos potenciales y se seleccionan una o varias alternativas propuestas para reducir o eliminar los riesgos.

Desarrollar, verificar y validar (probar)

- Tareas de la actividad propia y de prueba.
- Análisis de alternativas, identificación y resolución de riesgos.

- Dependiendo del resultado de la evaluación de los riesgos, se elige un modelo para el desarrollo, que puede ser cualquiera de los otros existentes, como formal, evolutivo, cascada, etc.

Planificar

- Se revisa todo lo hecho, evaluándolo, y con ello se decide si se continua con las fases siguientes y se planifica la próxima actividad.

Mecanismos de control

- La dimensión radial mide el costo.
- La dimensión angular mide el grado de avance del proyecto.

Ventajas

El análisis del riesgo se hace de forma explícita y clara. Une los mejores elementos de los restantes modelos.

- Reduce riesgos del proyecto
- Incorpora objetivos de calidad
- Integra el desarrollo con el mantenimiento, etc.

Además es posible tener en cuenta mejoras y nuevos requerimientos sin romper con la metodología, ya que este ciclo de vida no es rígido ni estático.

Desventajas

- Genera mucho tiempo en el desarrollo del sistema
- Modelo costoso
- Requiere experiencia en la identificación de riesgos

Inconvenientes

Genera mucho trabajo adicional, y eso causa muchos problemas sobre todo si la compañía que esta produciendo el software es floja. Cuando un sistema falla se pierde tiempo y costo dentro de la empresa [L7].

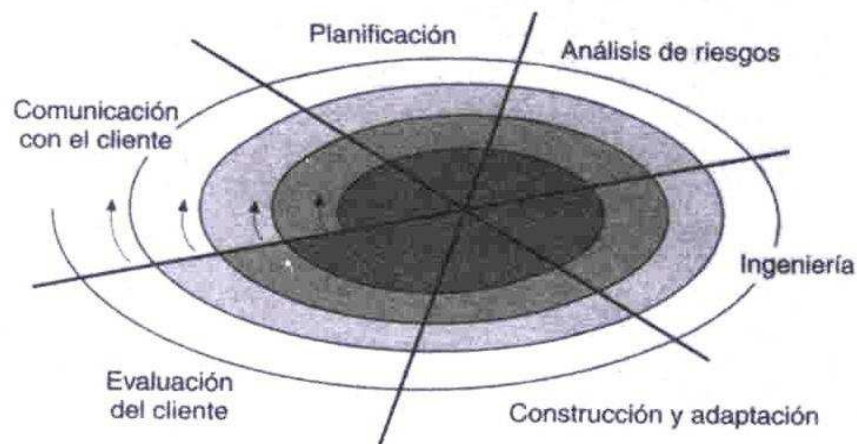


Figura 1.4 Modelo en espiral

1.3.2.3 MODELO DE CONSTRUCCION DE PROTOTIPOS

En Ingeniería de software el desarrollo con prototipado, también llamado modelo de prototipos se inicia con la definición de los objetivos globales para el software, luego se identifican los requisitos conocidos y las áreas del esquema en donde es necesaria más definición. Entonces se plantea con rapidez una iteración de construcción de prototipos y se presenta el modelado (en forma de un diseño rápido).

El diseño rápido se centra en una representación de aquellos aspectos del software que serán visibles para el cliente o el usuario final (por ejemplo, la configuración de la interfaz con el usuario y el formato de los despliegues de salida). El diseño rápido conduce a la construcción de un prototipo, el cual es evaluado por el cliente o el usuario para una retroalimentación; gracias a ésta se refinan los requisitos del software que se desarrollará. La iteración ocurre cuando el prototipo se ajusta para satisfacer las necesidades del cliente. Esto permite que al mismo tiempo el desarrollador entienda mejor lo que se debe hacer y el cliente vea resultados a corto plazo.

Ventajas

- Este modelo es útil cuando el cliente conoce los objetivos generales para el software, pero no identifica los requisitos detallados de entrada, procesamiento o salida.
- También ofrece un mejor enfoque cuando el responsable del desarrollo del software está inseguro de la eficacia de un algoritmo, de la adaptabilidad de un sistema operativo o de la forma que debería tomar la interacción humano-máquina.

La construcción de prototipos se utiliza como un modelo del proceso independiente, se emplea más comúnmente como una técnica susceptible de implementarse dentro del contexto de cualquiera de los modelos del proceso expuestos. Sin importar la forma en que éste se aplique, el paradigma de construcción de prototipos ayuda al desarrollador de software y al cliente a entender de mejor manera cuál será el resultado de la construcción cuando los requisitos estén satisfechos.

Inconvenientes

- El usuario tiende a crearse unas expectativas cuando ve el prototipo de cara al sistema final. A causa de la intención de crear un prototipo de forma rápida, se suelen desatender aspectos importantes, tales como la calidad y el mantenimiento a largo plazo, lo que obliga en la mayor parte de los casos a reconstruirlo una vez que el prototipo ha cumplido su función.
- En aras de desarrollar rápidamente el prototipo, el desarrollador suele tomar algunas decisiones de implementación poco convenientes (por ejemplo, elegir un lenguaje de programación incorrecto porque proporcione un desarrollo más rápido). Con el paso del tiempo, el desarrollador puede olvidarse de la razón que le llevó a tomar tales decisiones, con lo que se corre el riesgo de que dichas elecciones pasen a formar parte del sistema final [L6].



Figura 1.5 Modelo de construcción de prototipos

1.3.2.4 MODELO DE DESARROLLO EVOLUTIVO

En el desarrollo de software es común encontrar que el cliente solamente tenga un conjunto de objetivos globales y que no conozca los objetivos específicos del sistema, en este caso un modelo de desarrollo evolutivo es ideal.

Los pasos de especificación, desarrollo y validación se llevan a cabo al mismo tiempo a lo largo del proceso. Se comienza con la definición de los objetivos globales del sistema, estos objetivos se enfocan a las características visibles para el usuario, después de que se lleva a cabo un diseño rápido del sistema. Este modelo de desarrollo es ideal para sistemas pequeños o para sistemas de tamaño mediano con un tiempo de vida relativamente corto.

Existen 2 tipos de desarrollo evolutivo:

1.- Desarrollo exploratorio: el objetivo del proceso es trabajar con el cliente para explorar sus requerimientos y entregar el sistema final. El desarrollo empieza con las partes del sistema que se comprenden mejor. El sistema evoluciona agregando nuevos atributos acordes con las propuestas del cliente.

2.- Prototipos desechables: el objetivo del proceso de desarrollo evolutivo es comprender los requerimientos del cliente y entonces desarrollar una definición mejorada de los requerimientos del sistema. El prototipo se centra en experimentar con aquellas partes de los requerimientos del cliente que no se comprenden del todo [L6].

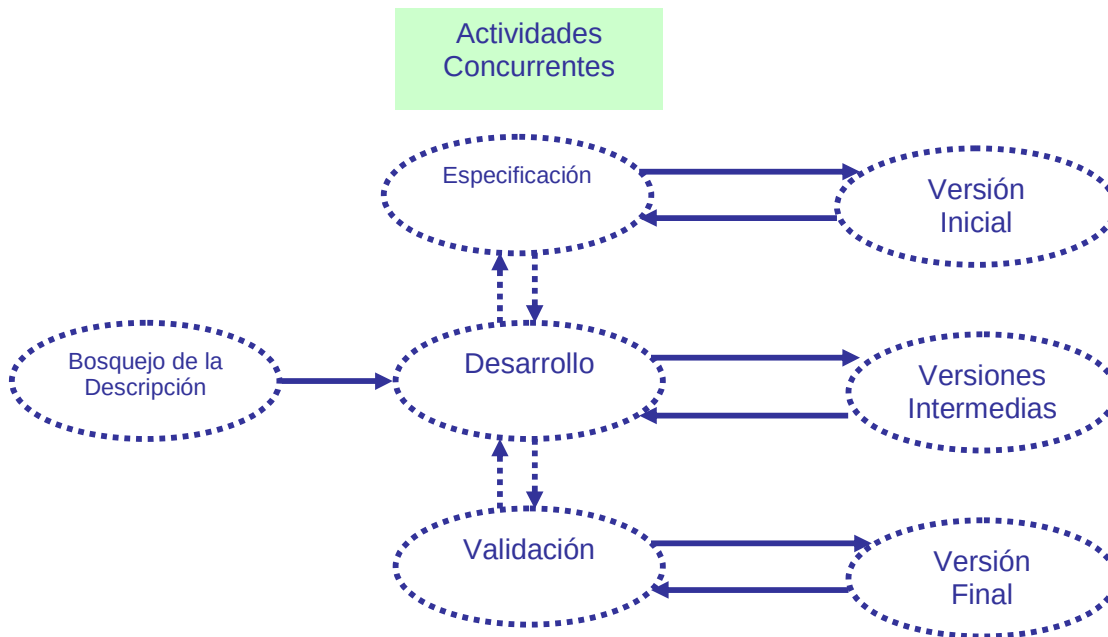


Figura 1.6 Modelo de Desarrollo Evolutivo

1.3.2.5 MODELO INCREMENTAL

El modelo incremental aplica repetidamente los elementos del modelo lineal secuencial, en cada secuencia lineal se produce un incremento en el sistema, cada incremento se construye sobre aquel que ya ha sido entregado.

El cliente identifica los requerimientos del sistema, después se clasifican dependiendo a su importancia y a continuación se definen los incrementos donde cada uno proporciona un subconjunto de funcionalidad del sistema. La asignación de servicios a los incrementos depende de la prioridad del servicio. Los servicios de prioridad más alta son los que se entregan primero al cliente.

En la entrega de cada incremento, los clientes pueden probarlo, esto ayuda a tener una idea mas clara de los requerimientos para los incrementos posteriores y para refinar la ultima versión del incremento actual.

Este modelo se basa en desarrollo y la entrega de incrementos de funcionalidad muy pequeños, en la participación del cliente en el proceso y el mejoramiento constante del código [L6].

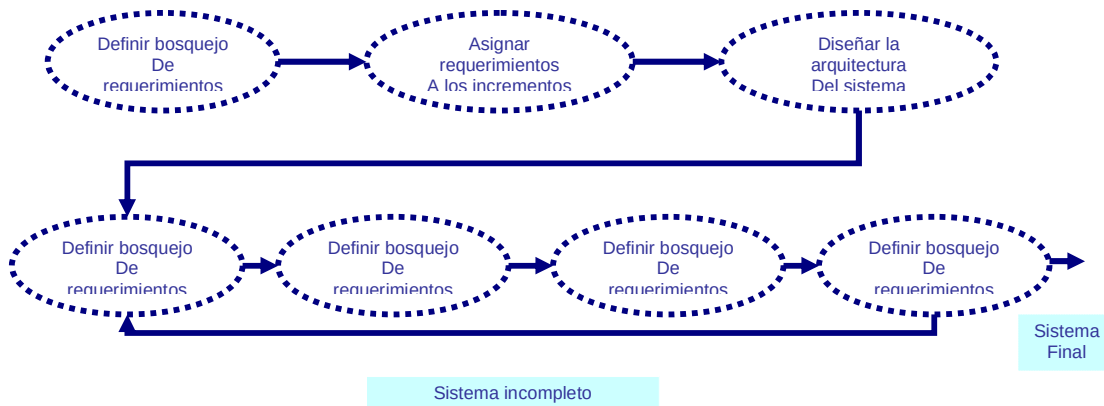


Figura 1.7 Modelo Incremental

1.3.2.6 MODELO RAD O DRA

El *Desarrollo Rápido de Aplicaciones (DRA)* como se muestra en la Figura 1.6 es un modelo de proceso del desarrollo del software lineal secuencial que enfatiza un ciclo de desarrollo extremadamente corto. El modelo DRA es una adaptación a «alta velocidad» del modelo lineal secuencial en el que se logra el desarrollo rápido utilizando una construcción basada en componentes. Si se comprenden bien los requisitos y se limita el ámbito del proyecto, el proceso DRA permite al equipo de desarrollo crear un «sistema completamente funcional» dentro de períodos cortos de tiempo (por ejemplo: de **60** a 90 días). Cuando se utiliza principalmente para aplicaciones de sistemas de información, el enfoque DRA comprende las siguientes fases:

Modelado de Gestión

El flujo de información entre las funciones de gestión se modela de forma que responda a las siguientes preguntas: ¿Qué información conduce el proceso de gestión? ¿Qué información se genera? ¿Quién la genera? ¿A dónde va la información? ¿Quién la procesa?

Modelado de datos

El flujo de información definido como parte de la fase de modelado de gestión se refina como un conjunto de objetos de datos necesarios para apoyar la empresa. Se definen las características (llamadas atributos) de cada uno de los objetos y las relaciones entre estos objetos.

Modelado del proceso

Los objetos de datos definidos en la fase de modelado de datos quedan transformados para lograr el flujo de información necesario para implementar una función de gestión. Las

descripciones del proceso se crean para añadir, modificar, suprimir, o recuperar un objeto de datos.

Generación de aplicaciones

El DRA asume la utilización de técnicas de cuarta generación. En lugar de crear software con lenguajes de programación de tercera generación, el proceso DRA trabaja para volver a utilizar componentes de programas ya existentes (cuando es posible) o a crear componentes reutilizables (cuando sea necesario). En todos los casos se utilizan herramientas para facilitar la construcción del software.

Pruebas y entrega

Como el proceso **DRA** enfatiza la reutilización, ya se han comprobado muchos de los componentes de los programas. Esto reduce tiempo de pruebas. Sin embargo, se deben probar todos los componentes nuevos y se deben ejercitar todas las interfaces a fondo.

Si una aplicación de gestión puede modularse de forma que permita completarse cada una de las funciones principales en menos de tres meses (utilizando el enfoque descrito anteriormente), es un candidato del DRA. Cada una de las funciones pueden ser afrontadas por un equipo **DRA** separado y ser integradas en un solo conjunto.

Desventajas

- Para proyectos grandes aunque por escalas, el **DRA** requiere recursos humanos suficientes como para crear el número correcto de equipos **DRA**.
- **DRA** requiere clientes y desarrolladores comprometidos en las rápidas actividades necesarias para completar un sistema en un marco de tiempo abreviado. Si no hay compromiso por ninguna de las partes constituyentes, los proyectos **DRA** fracasarán.
- No todos los tipos de aplicaciones son apropiados para **DRA**. Si un sistema no se puede modularizar adecuadamente, la construcción de los componentes necesarios para **DRA** será problemático. **Si** está en juego el alto rendimiento, y se va a conseguir el rendimiento convirtiendo interfaces en componentes de sistemas, el enfoque **DRA** puede que no funcione.
- **DRA** no es adecuado cuando los riesgos técnicos son altos. Esto ocurre cuando una nueva aplicación hace uso de tecnologías nuevas, o cuando el software nuevo requiere un alto grado de interoperatividad con programas de computadora ya existentes [L7].

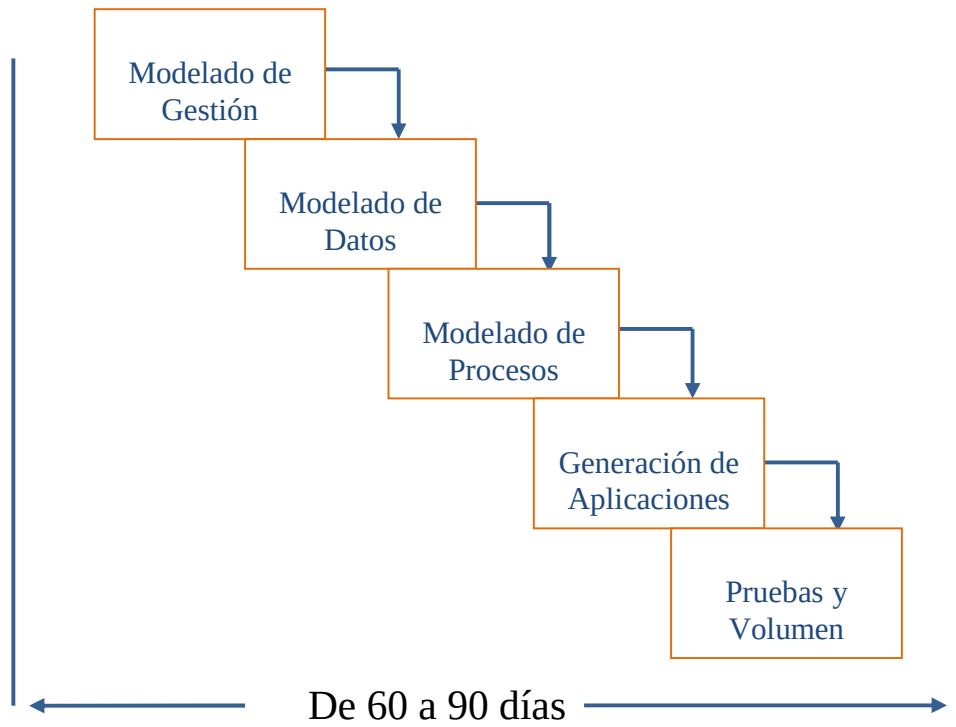


Figura 1.8 Modelo RAD o DRA.

1.4 LENGUAJE UNIFICADO DE MODELADO UML

Por sus siglas en inglés, *Unified Modeling Language* es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables [P5].

Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. Se utiliza para definir un sistema de software, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional), pero no especifica en sí mismo qué metodología o proceso usar.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.

1.4.1 ELEMENTOS DE UML

Diagrama de casos de uso

Los diagramas de casos de uso describen las relaciones y las dependencias entre un grupo de *casos de uso* y los actores participantes en el proceso.

Es importante resaltar que los diagramas de casos de uso no están pensados para representar el diseño y no puede describir los elementos internos de un sistema. Los diagramas de casos de uso sirven para facilitar la comunicación con los futuros usuarios del sistema, y con el cliente, y resultan especialmente útiles para determinar las características necesarias que tendrá el sistema. En otras palabras, los diagramas de casos de uso describen *qué* es lo que debe hacer el sistema, pero no *cómo*.

Caso de uso

Un *caso de uso* describe, —desde el punto de vista de los actores—, un grupo de actividades de un sistema que produce un resultado concreto y tangible.

Los casos de uso son descriptores de las interacciones típicas entre los usuarios de un sistema y ese mismo sistema. Representan el interfaz externo del sistema y especifican qué requisitos de funcionamiento debe tener éste.

Cuando se trabaja con casos de uso, es importante considerar las siguientes reglas:

- Cada caso de uso está relacionado como mínimo con un actor.
- Cada caso de uso es un iniciador (es decir, un actor)

- Cada caso de uso lleva a un resultado relevante (un resultado con “valor intrínseco”)

Los casos de uso pueden tener relaciones con otros casos de uso. Los tres tipos de relaciones más comunes entre casos de uso son:

- *<<include>>* que especifica una situación en la que un caso de uso tiene lugar *dentro* de otro caso de uso
- *<<extends>>* que especifica que en ciertas situaciones, o en algún punto (llamado punto de extensión) un caso de uso será extendido por otro.
- *Generalización* que especifica que un caso de uso hereda las características del “súper” caso de uso, y puede volver a especificar algunas o todas ellas de una forma muy similar a las herencias entre clases.

Actor

Un actor es una entidad externa (fuera del sistema) que interacciona con el sistema participando (y normalmente iniciando) en un caso de uso. Los actores pueden ser gente real (por ejemplo, usuarios del sistema), otras computadoras o eventos externos.

Los actores no representan a personas *físicas* o a sistemas, sino su *papel*. Esto significa que cuando una persona interacciona con el sistema de diferentes maneras (asumiendo diferentes papeles), estará representado por varios actores. Por ejemplo, una persona que proporciona servicios de atención al cliente telefónicamente y realiza pedidos para los clientes estaría representada por un actor “equipo de soporte” y por otro actor “representante de ventas”.

Descripción de casos de uso

Las descripciones de casos de uso son reseñas textuales del caso de uso. Normalmente tienen el formato de una nota o un documento relacionado de alguna manera con el caso de uso, y explica los procesos o actividades que tienen lugar en el caso de uso.

Diagrama de clases

Los diagramas de clases muestran las diferentes clases que componen un sistema y cómo se relacionan unas con otras. Se dice que los diagramas de clases son diagramas “estáticos” porque muestran las clases, junto con sus métodos y atributos, así como las relaciones estáticas entre ellas: qué clases “conocen” a qué otras clases o qué clases “son parte” de otras clases, pero no muestran los métodos mediante los que se invocan entre ellas.

Clase

Una clase define los atributos y los métodos de una serie de objetos. Todos los objetos de esta clase (instancias de esa clase) tienen el mismo comportamiento y el mismo conjunto de atributos (cada objetos tiene el suyo propio). En ocasiones se utiliza el término “tipo” en lugar de clase.

Las clases están representadas por rectángulos, con el nombre de la clase, y también pueden mostrar atributos y operaciones de la clase en otros dos "compartimentos" dentro del rectángulo.

Atributos

En UML, los atributos se muestran al menos con su nombre, y también incluyen su tipo, valor inicial y otras propiedades. Los atributos pueden visualizarse:

- + Indica atributos *públicos*
- # Indica atributos *protegidos*
- - Indica atributos *privados*

Operaciones

Las operaciones (métodos) también se muestran al menos con su nombre, parámetros y valores de retorno. Las operaciones, al igual que los atributos pueden visualizarse:

- + Indica operaciones *públicas*
- # Indica operaciones *protegidas*
- - Indica operaciones *privadas*

Diagramas de secuencia

Los diagramas de secuencia muestran el intercambio de mensajes (es decir la forma en que se invocan) en un momento dado. Los diagramas de secuencia ponen especial énfasis en el orden y el momento en que se envían los mensajes a los objetos.

En los diagramas de secuencia, los objetos están representados por líneas intermitentes verticales, con el nombre del objeto en la parte más alta. El eje de tiempo también es vertical, incrementándose hacia abajo, de forma que los mensajes son enviados de un objeto a otro en forma de flechas con los nombres de la operación y los parámetros.

Diagramas de colaboración

Los diagramas de colaboración muestran las interacciones que ocurren entre los objetos que participan en una situación determinada. Esta es más o menos la misma información que la mostrada por los diagramas de secuencia, pero destacando la forma en que las operaciones se producen en el tiempo, mientras que los diagramas de colaboración fijan el interés en las relaciones entre los objetos y su topología.

En los diagramas de colaboración los mensajes enviados de un objeto a otro se representan mediante flechas, mostrando el nombre del mensaje, los parámetros y la secuencia del mensaje. Los diagramas de colaboración están indicados para mostrar una situación o flujo programa específicos y son unos de los mejores tipos de diagramas para demostrar o explicar rápidamente un proceso dentro de la lógica del programa.

Diagrama de estado

Los diagramas de estado muestran los diferentes estados de un objeto durante su vida, y los estímulos que provocan los cambios de estado en un objeto.

- Los diagramas de estado ven a los objetos como *máquinas de estado* o autómatas finitos que pueden estar en un conjunto de estados finitos y que pueden cambiar su estado a través de un estímulo perteneciente a un conjunto finito.

Estado

Los estados son los ladrillos de los diagramas de estado. Un estado pertenece a exactamente una clase y representa un resumen de los valores y atributos que puede tener la clase. Un estado UML describe el estado interno de un objeto de una clase particular

Hay dos tipos especiales de estados: inicio y fin. Son especiales en el sentido de que no hay ningún evento que pueda devolver a un objeto a su estado de inicio, y de la misma forma no hay ningún evento que pueda sacar a un objeto de su estado de fin.

Diagrama de actividad

Los diagramas de actividad describen la secuencia de las actividades en un sistema. Los diagramas de actividad son una forma especial de los diagramas de estado, que únicamente (o mayormente) contienen actividades.

Los diagramas de actividad son similares a los diagramas de flujo procesales, con la diferencia de que todas las actividades están claramente unidas a objetos.

Los diagramas de actividad siempre están asociados a una *clase*, a una *operación* o a un *caso de uso*.

Los diagramas de actividad soportan actividades tanto secuenciales como paralelas. La ejecución paralela se representa por medio de iconos de fork/espera, y en el caso de las actividades paralelas, no importa en qué orden sean invocadas (pueden ser ejecutadas simultáneamente o una detrás de otra).

Las actividades pueden formar jerarquías, lo que significa que una actividad puede estar formada de varias actividades "de detalle", en cuyo caso las transiciones entrantes y salientes deberían coincidir con las del diagrama de detalle.

Diagramas de componentes

Los diagramas de componentes muestran los componentes del software (ya sea las tecnologías que lo forman como Kparts, componentes CORBA, Java Beans o simplemente secciones del sistema claramente distintas) y los artilugios de que está compuesto como los archivos de código fuente, las librerías o las tablas de una base de datos.

Los componentes pueden tener interfaces (es decir clases abstractas con operaciones) que permiten asociaciones entre componentes.

Diagramas de implementación

Los diagramas de implementación muestran las instancias existentes al ejecutarse así como sus relaciones. También se representan los nodos que identifican recursos físicos, típicamente una computadora así como interfaces y objetos (instancias de las clases) [P5].

1.5 SISTEMA DE BASE DE DATOS

Es un sistema que almacena datos que están relacionados. Es un repositorio en donde guardamos información integrada que podemos almacenar y recuperar.

Un conjunto de información almacenada en memoria auxiliar que permite acceso directo y un conjunto de programas que manipulan esos datos [L5].

1.5.1 COMPONENTES DE UN SISTEMA DE BASE DE DATOS

Un sistema de base de datos esta formado por cuatro elementos: datos, hardware, software y usuarios.

Hardware: constituido por dispositivos de almacenamiento como discos, tambores, cintas, periféricos de entrada y salida, buses para conectar dispositivos, controladores de dispositivos, procesadores de hardware, memoria, etc.

Software: Llamado sistema manejador de base de datos DBMS o Sistema Administrador de Base de Datos, es el software encargado de ocultar a los usuarios los detalles a nivel hardware, ofreciendo operaciones SQL para manipular la base de datos. Otros componentes de software incluyen herramientas de desarrollo de aplicaciones, generadores de informes, administrador de transacciones, etc.

Datos: los cuales están almacenados de acuerdo a la estructura externa y van a ser procesados para convertirse en información. Ya sea que el sistema permita acceso simultáneo o que se trate de un sistema monousuario, los datos deben cumplir dos propiedades: ser integrados y compartidos. Los datos se dicen integrados al considerar la base de datos como una colección de elementos con una redundancia eliminada parcialmente. Datos compartidos se refiere a que estos se encuentran disponibles para distintos usuarios, quizás al mismo tiempo [P4].

Usuarios: se consideran 3 tipos de usuarios

- **Usuario Final:** es la persona que utiliza los datos, esta persona ve datos convertidos en información:
- **Desarrollador de Aplicaciones:** es la persona que desarrolla los sistemas que interactúan con la Base de Datos.
- **DBA:** es la persona que asegura integridad, consistencia, redundancia, seguridad, este es el Administrador de Base de Datos quien se encarga de realizar el mantenimiento diario o periódico de los datos [L1].

1.5.2 SISTEMA GESTOR DE BASE DE DATOS

Un **Sistema Gestión de Bases de Datos –SGBD-** Consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a esos datos. El Objetivo primordial de un SGBD es proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer y almacenar información de la base de datos.

“El sistema de gestión de la base de datos (SGBD) es una aplicación que permite a los usuarios definir, crear y mantener la base de datos, y proporciona acceso controlado a la misma.”

Existen distintos objetivos que deben cumplir los SGBD:

Abstracción de la información

Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario.

Existen tres niveles de abstracción:

- **Nivel físico:** el nivel más bajo de abstracción describe cómo se almacenan realmente los datos. En el nivel físico se describen a detalle las estructuras de datos complejas de bajo nivel.
- **Nivel lógico:** el siguiente nivel más alto de abstracción describe qué datos se almacenan en la base de datos y qué relaciones existen entre esos datos. Los administradores que deben decidir la información que se mantiene en la base de datos, usan el nivel lógico de abstracción.
- **Nivel de vistas:** el nivel más alto de abstracción describe sólo parte de la base de datos completa. A pesar del uso de estructuras más simples en el nivel lógico, queda algo de complejidad, debido a la variedad de información almacenada en una gran base de datos. El sistema puede proporcionar muchas vistas para la misma base de datos [17].

Independencia

La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.

Consistencia

En aquellos casos en los que no se ha logrado eliminar la redundancia, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.

Seguridad

La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra segura frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado pero despistado. Normalmente, los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.

Integridad

Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada. Los SGBD proveen mecanismos para garantizar la recuperación de la base de datos hasta un estado consistente.

Respaldo

Los SGBD deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.

Control de la concurrencia

En la mayoría de entornos, lo más habitual es que sean muchas las personas que acceden a una base de datos, bien para recuperar información, bien para almacenarla. Y es también frecuente que dichos accesos se realicen de forma simultánea. Así pues, un SGBD debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias.

Tiempo de respuesta

Lógicamente, es deseable minimizar el tiempo que el SGBD tarda en darnos la información solicitada y en almacenar los cambios realizados [P6].

1.6 MODELO DE DATOS

Es una colección de herramientas conceptuales para describir los datos, las relaciones, la semántica y las restricciones de consistencia. Los diferentes modelos de datos se clasifican en tres grupos diferentes:

- Modelos lógicos basados en objetos.
- Modelos lógicos basados en registros.

- Modelos físicos.

1.6.1 MODELO ENTIDAD – RELACIÓN

El **Modelo Entidad-Relación** es un concepto de modelado para bases de datos, propuesto por Peter Chen en 1976, mediante el cual se pretende 'visualizar' los objetos que pertenecen a la Base de Datos como **entidades** (se corresponde al concepto de objeto de la Programación Orientada a Objetos) las cuales tienen unos atributos y se vinculan mediante **relaciones**.

El modelo entidad-relación se basa en los conceptos descritos a continuación para representar un modelo de la vida real.

Entidad

Representa una "cosa" u "objeto" del mundo real con existencia independiente, es decir, se diferencia unívocamente de cualquier otro objeto o cosa, incluso siendo del mismo tipo.

Atributos

Los atributos son las propiedades que describen a cada entidad en un conjunto de entidades. Un conjunto de entidades dentro de una entidad, tiene **valores** específicos asignados para cada uno de sus atributos, de esta forma, es posible su identificación unívoca.

Relación

Es una asociación entre varias entidades.

Correspondencia de cardinalidades

Dado un conjunto de relaciones en el que participan dos o más conjuntos de entidades, la correspondencia de cardinalidad indica el número de entidades con las que puede estar relacionada una entidad dada.

Dado un conjunto de relaciones binarias y los conjuntos de entidades A y B, la correspondencia de cardinalidades puede ser:

- **Uno a uno:** Una entidad de A se relaciona únicamente con una entidad en B y viceversa.
- **Uno a varios:** Una entidad en A se relaciona con cero o muchas entidades en B. Pero una entidad en B se relaciona con una única entidad en A.
- **Varios a uno:** Una entidad en A se relaciona exclusivamente con una entidad en B. Pero una entidad en B se puede relacionar con 0 o muchas entidades en A.

- **Varios a varios:** Una entidad en A se puede relacionar con 0 o muchas entidades en B y viceversa.

Claves

Es un subconjunto del conjunto de atributos comunes en una colección de entidades, que permite identificar unívocamente cada una de las entidades pertenecientes a dicha colección. Asimismo, permiten distinguir entre sí las relaciones de un conjunto de relaciones.

Dentro de los conjuntos de entidades existen los siguientes tipos de claves:

- **Superclave:** Es un subconjunto de atributos que permite distinguir unívocamente cada una de las entidades de un conjunto de entidades. Si otro atributo unido al anterior subconjunto, el resultado seguirá siendo una superclave.
- **Clave candidata:** Dada una superclave, si ésta deja de serlo removiendo únicamente uno de los atributos que la componen, entonces ésta es una clave candidata.
- **Clave primaria:** Es una clave candidata, elegida por el diseñador de la base de datos, para identificar unívocamente las entidades en un conjunto de entidades.

Los valores de los atributos de una clave, no pueden ser todos iguales para dos o más entidades [L10].

1.6.2 DIAGRAMA ENTIDAD-RELACIÓN

Formalmente, los diagramas E-R son un lenguaje gráfico para describir conceptos. Informalmente, son simples dibujos o gráficos que describen la información que trata un sistema de información y el software que lo automatiza.

Entidad

Se representa mediante un rectángulo o "caja" etiquetada en su interior mediante un **identificador**. Ejemplos de entidades habituales en los sistemas de información son: factura, persona, empleado.

Atributo

Se representan mediante un círculo o elipse etiquetado mediante un nombre en su interior. Cuando un atributo es identificativo de la entidad se suele subrayar dicha etiqueta.

Relaciones

Se representa mediante un rombo etiquetado en su interior con un **verbo**. Este rombo se debe unir mediante líneas con las entidades (rectángulos) que relaciona.

Por motivos de legibilidad, los atributos **no suelen representarse** en un diagrama entidad-relación, sino que se describen textualmente en otros documentos adjuntos [L5].

1.6.3 MODELO RELACIONAL

El **modelo relacional** concepto generado por Edgar Frank Codd, es un modelo de datos basado en la lógica de predicado y en la teoría de conjuntos. Es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Su idea fundamental es el uso de «relaciones». Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados tuplas esto es, pensando en cada **relación** como si fuese una **tabla** que está compuestas por *registros* (cada fila de la tabla sería un registro o *tupla*), y *columnas* (también llamadas *campos*) [L7].

1.7 NORMALIZACIÓN

La normalización es el proceso mediante el cual se transforman datos complejos a un conjunto de estructuras de datos más pequeñas, que además de ser más simples y más estables, son más fáciles de mantener. También se puede entender la normalización como una serie de reglas que sirven para ayudar a los diseñadores de bases de datos a desarrollar un esquema que minimice los problemas de lógica. Cada regla está basada en la que le antecede [L10].

Existen varios niveles de normalización:

1.7.1 PRIMERA FORMA NORMAL (1FN)

Una tabla está en Primera Forma Normal sólo si:

- Todos los atributos son atómicos. Un atributo es atómico si los elementos del dominio son indivisibles, mínimos.
- La tabla contiene una clave primaria
- La tabla no contiene atributos nulos

Una columna no puede tener múltiples valores. Los datos son atómicos. (Si a cada valor de X le pertenece un valor de Y, entonces a cada valor de Y le pertenece un valor de X).

1.7.2 SEGUNDA FORMA NORMAL (2FN)

Dependencia Funcional. Una relación está en 2FN si está en 1FN y si los atributos que no forman parte de ninguna clave dependen de forma completa de la clave principal. Es decir

que no existen dependencias parciales. En otras palabras podríamos decir que la segunda forma normal está completamente basada en el concepto de dependencia funcional.

1.7.3 TERCERA FORMA NORMAL (3FN)

La tabla se encuentra en 3FN si es 2FN y cada atributo que no forma parte de ninguna clave, depende directamente y no transitivamente, de la clave primaria.

1.7.4 FORMA NORMAL DE BOYCE-CODD (FNBC)

La tabla se encuentra en BCNF si cada determinante, atributo que determina completamente a otro, es clave candidata.

1.7.5 CUARTA FORMA NORMAL (4FN)

Una tabla se encuentra en 4FN si, y sólo si, para cada una de sus dependencias múltiples no funcionales $X \twoheadrightarrow Y$, siendo X una super-clave que, X es o una clave candidata o un conjunto de claves primarias.

1.7.6 QUINTA FORMA NORMAL (5FN)

Una tabla se encuentra en 5FN si:

- La tabla esta en 4FN
- No existen relaciones de dependencias no triviales que no siguen los criterios de las claves. Una tabla que se encuentra en la 4FN se dice que está en la 5FN si, y sólo si, cada relación de dependencia se encuentra definida por las claves candidatas [L7].

1.8 MYSQL

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario. Su diseño es apropiado para manejar las bases de datos y ofrece compatibilidades con PHP, Perl, C y HTML, además, cuenta con otras opciones más avanzadas que facilitan la optimización y administración de las bases de datos.

Es un sistema cliente servidor de administración de bases de datos relacionales diseñado para el trabajo tanto en Windows como en UNIX/LINUX.

MySQL es un sistema gestor de bases de datos relacional cliente-servidor de coste mínimo que incluye un servidor SQL, programas cliente para acceder al servidor, herramientas administrativas y una interfaz de programación para escribir programas, es portable y se ejecuta en sistemas operativos comerciales.

MySQL es software de fuente abierta. Fuente abierta significa que cualquier persona puede usarlo y modificarlo. Cualquier interesado puede estudiar el código fuente y ajustarlo a sus

necesidades. MySQL usa el GPL (GNU General Public License) para definir que puede hacer y que no puede hacer con el software en diferentes situaciones.

Hay tres tipos de compilación del servidor MySQL:

- Estándar: Los binarios estándares de **MySQL** son los recomendados para la mayoría de los usuarios, e incluyen el motor de almacenamiento InnoDB.
- Max (No se trata de MaxDB, que es una cooperación con SAP): Los binarios incluyen características adicionales que no han sido lo bastante probadas o que normalmente no son necesarias.
- MySQL-Debug: Son binarios que han sido compilados con información de depuración extra. No debe ser usada en sistemas en producción porque el código de depuración puede reducir el rendimiento [P6].

1.9 LENGUAJES DE PROGRAMACION WEB

Desde los inicios de Internet, fueron surgiendo diferentes demandas por los usuarios y se dieron soluciones mediante lenguajes estáticos. A medida que paso el tiempo, las tecnologías fueron desarrollándose y surgieron nuevos problemas a dar solución. Esto dio lugar a desarrollar lenguajes de programación para la web dinámica, que permitieran interactuar con los usuarios y utilizaran sistemas de Bases de Datos.

La programación Web, parte de las siglas WWW, que significan World Wide Web o telaraña mundial.

Para realizar una pagina con la programación Web, se deben tener claros, tres conceptos fundamentales los cuales son, el URL(Uniform Resource Locators), es un sistema con el cual se localiza un recurso dentro de la red, este recurso puede ser una pagina Web, un servicio o cualquier otra cosa. En resumen el URL no es más que un nombre, que identifica una computadora, dentro de esa computadora un archivo que indica el camino al recurso que se solicita.

El siguiente concepto dentro de la programación Web, es el protocolo encargado de llevar la información que contiene una pagina Web por toda la red de Internet, como es el HTTP(Hypertext Transfer Protocol).

Y por ultimo el lenguaje necesario cuya funcionalidad es la de representar cualquier clase de información que se encuentre almacenada en una pagina Web, este lenguaje es el HTML(Hypertext Markup Language).

A continuación se describe brevemente algunos de los lenguajes de programación que han tenido más auge [P3].

1.9.1 LENGUAJE HTML

Desde el surgimiento de Internet se han publicado sitios Web gracias al lenguaje HTML. Es un lenguaje estático para el desarrollo de sitios Web (acrónimo en inglés de HyperText Markup Language, en español Lenguaje de Marcas Hipertextuales). Desarrollado por el World Wide Web Consortium (W3C). Los archivos pueden tener las extensiones (htm, html).

Ventajas:

- Sencillo que permite describir hipertexto.
- Texto presentado de forma estructurada y agradable.
- No necesita de grandes conocimientos cuando se cuenta con un editor de páginas Web o WYSIWYG.
- Archivos pequeños.
- Despliegue rápido.
- Lenguaje de fácil aprendizaje.
- Lo admiten todos los exploradores.

Desventajas

- Lenguaje estático.
- La interpretación de cada navegador puede ser diferente.
- Guarda muchas etiquetas que pueden convertirse en "basura" y dificultan la corrección.
- El diseño es más lento.
- Las etiquetas son muy limitadas

1.9.2 LENGUAJE JAVASCRIPT

Este es un lenguaje interpretado, no requiere compilación. Fue creado por Brendan Eich en la empresa Netscape Communications. Utilizado principalmente en páginas Web. Es similar a Java, aunque no es un lenguaje orientado a objetos, el mismo no dispone de herencias. La mayoría de los navegadores en sus últimas versiones interpretan código Javascript.

El código Javascript puede ser integrado dentro de nuestras páginas Web. Para evitar incompatibilidades el World Wide Web Consortium (W3C) diseñó un estándar denominado DOM (en inglés Document Object Model, en su traducción al español Modelo de Objetos del Documento).

Ventajas

- Lenguaje de scripting seguro y fiable.
- Los scripts tienen capacidades limitadas, por razones de seguridad.

- El código Javascript se ejecuta en el cliente.

Desventajas

- Código visible por cualquier usuario.
- El código debe descargarse completamente.
- Puede poner en riesgo la seguridad del sitio, con el actual problema llamado XSS (significa en inglés Cross Site Scripting renombrado a XSS por su similitud con las hojas de estilo CSS).

1.9.3 LENGUAJE PHP

Es un lenguaje de programación utilizado para la creación de sitio web. PHP es un acrónimo recursivo que significa "PHP Hypertext Pre-processor". Surgió en 1995, desarrollado por PHP Group.

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas Web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor. PHP no necesita ser compilado para ejecutarse. Para su funcionamiento necesita tener instalado Apache o IIS con las librerías de PHP. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Los archivos cuentan con la extensión (php).

Ventajas

- Muy fácil de aprender.
- Se caracteriza por ser un lenguaje muy rápido.
- Soporta en cierta medida la orientación a objeto. Clases y herencia.
- Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Capacidad de expandir su potencial utilizando módulos.
- Posee documentación en su página oficial la cual incluye descripción y ejemplos de cada una de sus funciones.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Incluye gran cantidad de funciones.
- No requiere definición de tipos de variables ni manejo detallado del bajo nivel.

Desventajas

- Se necesita instalar un servidor Web.
- Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.
- Dificulta la modularización.

- Dificulta la organización por capas de la aplicación.

Seguridad

PHP es un poderoso lenguaje e intérprete, ya sea incluido como parte de un servidor Web en forma de módulo o ejecutado como un binario CGI separado, es capaz de acceder a archivos, ejecutar comandos y abrir conexiones de red en el servidor. Estas propiedades hacen que cualquier cosa que sea ejecutada en un servidor Web sea insegura por naturaleza.

PHP está diseñado específicamente para ser un lenguaje más seguro para escribir programas CGI que Perl o C, y con la selección correcta de opciones de configuración en tiempos de compilación y ejecución, y siguiendo algunas prácticas correctas de programación.

1.9.4 LENGUAJE ASP

Es una tecnología del lado de servidor desarrollada por Microsoft para el desarrollo de sitio Web dinámicos. ASP significa en inglés (Active Server Pages), fue liberado por Microsoft en 1996. Las páginas Web desarrolladas bajo este lenguaje es necesario tener instalado Internet Information Server (IIS).

ASP no necesita ser compilado para ejecutarse. Existen varios lenguajes que se pueden utilizar para crear páginas ASP. El más utilizado es VBScript, nativo de Microsoft. ASP se puede hacer también en Perl and Jscript (no JavaScript). El código ASP puede ser insertado junto con el código HTML. Los archivos cuentan con la extensión (asp).

Ventajas

- Usa Visual Basic Script, siendo fácil para los usuarios.
- Comunicación óptima con SQL Server.
- Soporta el lenguaje JScript (Javascript de Microsoft).

Desventajas

- Código desorganizado.
- Se necesita escribir mucho código para realizar funciones sencillas.
- Tecnología propietaria.
- Hospedaje de sitios web costosos.

1.9.5 LENGUAJE ASP.NET

Este es un lenguaje comercializado por Microsoft, y usado por programadores para desarrollar entre otras funciones, sitios Web. ASP.NET es el sucesor de la tecnología ASP, fue lanzada al mercado mediante una estrategia de mercado denominada .NET.

El ASP.NET fue desarrollado para resolver las limitantes que brindaba tu antecesor ASP. Creado para desarrollar Web sencillas o grandes aplicaciones. Para el desarrollo de ASP.NET se puede utilizar C#, VB.NET o J#. Los archivos cuentan con la extensión (aspx). Para su funcionamiento de las páginas se necesita tener instalado IIS con el Framework .Net. Microsoft Windows 2003 incluye este framework, solo se necesitará instalarlo en versiones anteriores.

Ventajas

- Completamente orientado a objetos.
- Controles de usuario y personalizados.
- División entre la capa de aplicación o diseño y el código.
- Facilita el mantenimiento de grandes aplicaciones.
- Incremento de velocidad de respuesta del servidor.
- Mayor velocidad.
- Mayor seguridad.

Desventajas

- Mayor consumo de recursos.

1.9.6 LENGUAJE JSP

Es un lenguaje para la creación de sitios Web dinámicos, acrónimo de Java Server Pages. Está orientado a desarrollar páginas Web en Java. JSP es un lenguaje multiplataforma. Creado para ejecutarse del lado del servidor.

JSP fue desarrollado por Sun Microsystems. Comparte ventajas similares a las de ASP.NET, desarrollado para la creación de aplicaciones Web potentes. Posee un motor de páginas basado en los servlets de Java. Para su funcionamiento se necesita tener instalado un servidor Tomcat.

Características:

- Código separado de la lógica del programa.
- Las páginas son compiladas en la primera petición.
- Permite separar la parte dinámica de la estática en las páginas Web.
- Los archivos se encuentran con la extensión (jsp).
- El código JSP puede ser incrustado en código HTML.

Elementos de JSP

Los elementos que pueden ser insertados en las páginas JSP son los siguientes:

- **Código:** se puede incrustar código "Java".

- **Directivas:** permite controlar parámetros del servlet.
- **Acciones:** permite alterar el flujo normal de ejecución de una página.

Ventajas:

- Ejecución rápida del servlets.
- Crear páginas del lado del servidor.
- Multiplataforma.
- Código bien estructurado.
- Integridad con los módulos de Java.
- La parte dinámica está escrita en Java.
- Permite la utilización se servlets.

Desventajas:

- Complejidad de aprendizaje

Esto es solo por mencionar algunos de los lenguajes de programación Web mas conocidos [P7].

CAPITULO II

Análisis del Sistema

La importancia del análisis del sistema radica principalmente en la definición específica de los requerimientos del producto de software. En este capítulo se definen detalladamente los objetivos del sistema, se especifican los casos de uso y los módulos funcionales.

Con los casos de uso se describen las funciones principales del sistema, los cuales servirán para entender claramente los requisitos del mismo.

2.1 INTRODUCCIÓN

Esta sección se concentra en la interacción con las personas para realizar el análisis de requerimientos. Esta es una actividad de persona a persona, organizada con sumo cuidado para producir la mejor aplicación. Un beneficio importante del análisis de requerimientos es la comprensión y acuerdo de la aplicación que se construirá.

Un requerimiento defectuoso (es decir el que no se repara antes de terminar el documento de requerimientos) es muy costoso, por lo cual el análisis de requerimientos es una necesidad, no un lujo [L2]. Ya que este es la base para obtener un buen diseño, en general la obtención correcta de los requerimientos permitirá obtener los beneficios y objetivos por los cuales se está construyendo el sistema.

2.2. DEFINICION DE REQUERIMIENTOS

Para construir algo primero debe entenderse lo que debe ser ese "algo". Al proceso de entender y documentar este algo se denomina "análisis de requerimientos". En general, los requerimientos expresan qué se supone que debe hacer una aplicación [L2].

Los requerimientos son expresados por medio de oraciones que empiezan con "El sistema debe requerir", "El sistema debe permitir". Por medio de esta técnica, el lenguaje se centra en los requerimientos y se establece de manera sencilla lo que el sistema debe realizar [L8].

2.3 PLANTEAMIENTO DEL PROBLEMA

Este proyecto surge de la necesidad de contar con diferentes opciones para obtener el título de alguna licenciatura o carrera profesional, ya que el CENEVAL se ofrece como otra alternativa de titulación, para esto es necesario contar con herramientas que faciliten la aprobación de dicha prueba abarcando todos los temas vistos durante el transcurso de los estudios de licenciatura.

Por esta razón resulta conveniente realizar un Simulador Web de exámenes prototipo, de manera que el alumno pueda seleccionar el tipo de examen que necesite sea de Licenciatura o Ingeniería como herramienta de preparación para el examen ofertado por el CENEVAL.

Este examen de prueba consistirá en una serie de preguntas con límite de tiempo, sin embargo el estudiante sólo conocerá su resultado hasta finalizar la prueba, donde se mostrarán los resultados con la cantidad de preguntas mostradas, el número de preguntas contestadas correctamente, el número de preguntas contestadas incorrectamente y el número de preguntas no contestadas, así mismo se desglosará una tabla en donde el alumno podrá observar las puntuaciones obtenidas en cada área y subárea, semejando el reporte dado por el CENEVAL.

2.4 REQUERIMIENTOS FUNCIONALES DEL SISTEMA

El sistema debe requerir que el usuario se identifique mediante un nombre de usuario y una contraseña.

El sistema debe permitir registrar los datos de nuevos usuarios.

El sistema debe permitir que el usuario elija su nombre de usuario el cual podrá ser mínimo de 6 caracteres y máximo de 12 caracteres, también debe elegir la contraseña con un mínimo de 6 caracteres y máximo de 12 caracteres.

El sistema debe permitir al administrador del sistema capturar un número indeterminado de preguntas con sus respectivas respuestas.

El sistema debe permitir al usuario ver su calificación.

El sistema debe permitir al usuario modificar su información, nombre, contraseña, etc.

El sistema debe permitir al coordinador ver las calificaciones de todos los usuarios, así como también ver su información.

El sistema debe permitir al usuario realizar la evaluación.

El sistema debe permitir que el usuario conteste las preguntas dentro de un tiempo determinado.

El sistema va a generar algunos reportes, tales como: el número de personas que han realizado la prueba, las calificaciones correspondientes a cada persona y el número de registros de personas.

El sistema va a generar también gráficas mostrando los resultados.

2.5 MODELO DE CASOS DE USO

Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico. Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema al mostrar como reacciona una respuesta a eventos que se producen en el mismo.

2.5.1 DIAGRAMA DE CASOS DE USO

El diagrama de casos de uso se muestra en la figura 2.1 y aquí se identifican cuales son las actividades de cada uno de los actores. En este caso existen solo 2 actores; el actor usuario y el actor administrador, quienes realizan las siguientes actividades.

Actor Administrador

- Formular preguntas y respuestas.
- Modificar información de usuarios.
- Identificación de usuario.

Actor usuario

- Identificación de usuario.
- Registrarse en el sistema.
- Realizar evaluación
- Ver calificación.

Actor Coordinador

- Ver reportes

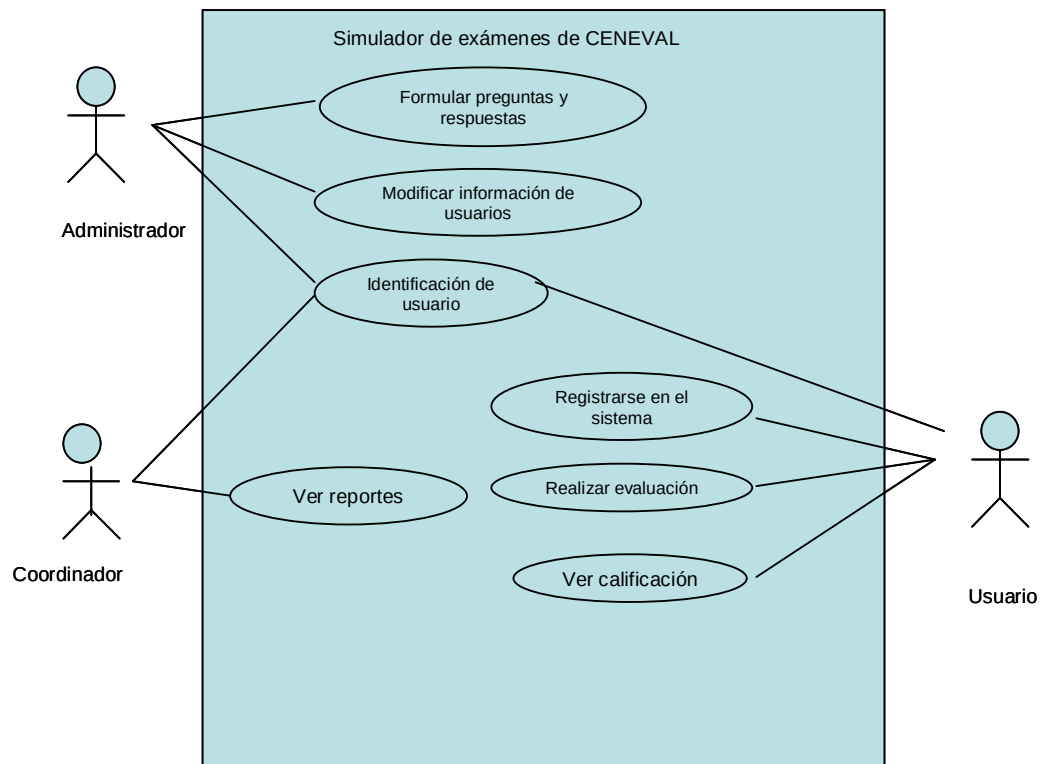


Figura 2.1 Diagrama General del Sistema

2.5.2 DESCRIPCION DE CASOS DE USO

Nombre del caso de uso:	Formular preguntas y respuestas
Descripción: El actor inicia el caso de uso. El sistema permite crear las preguntas y respuestas de la prueba.	
Actores: Administrador	
Precondiciones: El administrador debe identificarse (logearse) en el sistema.	
Flujo Normal: 1.- El sistema despliega una pantalla donde el usuario podrá elegir la opción formular nueva. 2.- El sistema mostrará un formulario que contenga el número de pregunta, la pregunta, la respuesta correcta y las respuestas opcionales. Posteriormente pulsara el botón guardar. 3.- El sistema enviará un mensaje de confirmación.	
Flujo Alternativo: No aplica.	
Poscondiciones: La pregunta y respuestas han sido guardadas en el sistema.	

Nombre del caso de uso:	Modificar información de usuario
Descripción: El actor administrador inicia el caso de uso. El sistema muestra una pantalla donde permite modificar la información de los usuarios registrados en el sistema.	
Actores: Administrador	
Precondiciones: El administrador debe identificarse (logearse) en el sistema.	
Flujo Normal: 1.- Se visualizara una pantalla en donde se desplegarán los nombres de los usuarios que se encuentran registrados en el sistema. 2.- El actor administrador seleccionará el nombre del usuario del cual desea modificar los datos. Posteriormente pulsara el botón modificar. 3.- El sistema desplegará un formulario con los datos del usuario. 4.- El actor realizará las modificaciones necesarias. Y pulsará el botón guardar 5.- El sistema desplegara un mensaje.	
Flujo Alternativo: No aplica.	
Poscondiciones: La información nueva ha sido guardada en el sistema.	

Nombre del caso de uso:	Identificación de usuario
Descripción: Los actores inician el caso de uso respectivamente. El sistema permite que el usuario introduzca su nombre de usuario y su contraseña.	
Actores: Administrador, Coordinador y Usuario	
Precondiciones: El actor administrador y el actor usuario deben estar registrados en el sistema.	
Flujo Normal: 1.- El sistema despliega un formulario con la información requerida. 2.- El actor debe introducir su nombre de usuario y su contraseña. Posteriormente debe presionar el botón Aceptar. 2.1 Si el nombre de usuario y la contraseña son correctas se desplegará la siguiente pantalla. 2.2 Si el nombre de usuario o la contraseña son incorrectas se desplegará un mensaje de error y el formulario anterior hasta que los datos sean correctos.	
Flujo Alternativo: No aplica.	
Poscondiciones: No aplica.	

Nombre del caso de uso:	Registrarse en el sistema.
Descripción: El actor usuario inicia el caso de uso. El sistema permite al usuario registrar sus datos en el sistema para realizar la prueba posteriormente.	
Actores: Usuario	
Precondiciones: No aplica	
Flujo Normal: 1.- El sistema despliega la pantalla principal. 2.- El usuario elige la opción registrarse. 3.- El sistema muestra un formulario con los datos requeridos. Posteriormente procede al llenado, y pulsa el botón guardar. 4.- El sistema comprueba la validez de los datos y los almacena.	
Flujo Alternativo: 4.- El sistema comprueba la validez de los datos, si estos no son correctos o presentan alguna redundancia, se avisa al actor de ello, permitiéndole que los corrija.	
Poscondiciones: El nuevo usuario ha sido registrado en el sistema.	

Nombre del caso de uso:	Realizar evaluación
Descripción: El actor usuario inicia el caso de uso. El sistema permite al usuario realizar la evaluación correspondiente.	
Actores: Usuario	
Precondiciones: El actor debe identificarse (logearse) en el sistema.	
Flujo Normal: 1.- El sistema muestra una pantalla donde el usuario elegirá realizar la prueba. 2.- Presionara el botón realizar evaluación. 3.- El sistema desplegará una pantalla que contiene el formulario de las preguntas y las posibles respuestas. El usuario elegirá las respuestas que el considere correctas. Una vez que termino de contestar presionará el botón siguiente. 4.- El sistema desplegará la página siguiente hasta llegar al final de la evaluación.	
Flujo Alternativo: No aplica.	
Poscondiciones: La evaluación ha sido realizada.	

Nombre del caso de uso:	Ver calificación
Descripción: El actor usuario inicia el caso de uso. El sistema permite al usuario ver la calificación correspondiente.	
Actores: Usuario	
Precondiciones: El actor debe identificarse (logearse) en el sistema y haber realizado el caso de uso Realizar evaluación.	
Flujo Normal: 1.- Una vez terminada la evaluación, el usuario presionara el botón Evaluar, y se desplegara la información correspondiente.	
Flujo Alternativo: No aplica.	
Poscondiciones: No aplica.	

Nombre del caso de uso:	Ver reportes
Descripción: El actor usuario inicia el caso de uso. El sistema permite al actor coordinador ver los reportes generados por el sistema.	
Actores: Coordinador	
Precondiciones: El actor debe identificarse (logearse) en el sistema.	
Flujo Normal: 1.- El sistema desplegará una pantalla, donde el actor, podrá elegir ver los reportes generados por el sistema.	
Flujo Alternativo: No aplica.	
Poscondiciones: No aplica.	

CAPITULO III

Diseño del Sistema

Una vez recabados y definidos todos los requerimientos del sistema se prosigue a la etapa del diseño, el *Diseño de sistemas* es el arte de definir la arquitectura de hardware y software, componentes, módulos y datos de un sistema de cómputo para satisfacer ciertos requerimientos.

El diseño de una base de datos relacional tiene por objetivo generar un conjunto de esquemas de relaciones que permiten almacenar la información con un mínimo de redundancia. El diseño esta conformado por tres etapas: diseño conceptual, diseño lógico y diseño físico.

3.1 DISEÑO CONCEPTUAL

En esta etapa se descubren entidades, atributos y relaciones. El esquema conceptual se construye a partir de la información recabada en el análisis de requerimientos. El diseño conceptual es totalmente independiente del manejador de base de datos que se escoja, de los programas de aplicación que se usen, de los lenguajes de programación Web y de los aspectos de hardware u otras consideraciones físicas que se presenten. El esquema conceptual es una fuente importante para el diseño lógico de la base de datos.

La notación más popular para esquematizar el diseño conceptual es el Diagrama Entidad – Relación como se presenta en la figura 3.1:

3.1.1 MODELO ENTIDAD – RELACIÓN

Para facilitar el entendimiento de este diagrama a continuación se describe la simbología utilizada para el Diagrama Entidad-Relación.

- " # " Llave primaria
- " * " Atributo obligatorio
- " o " Atributo opcional

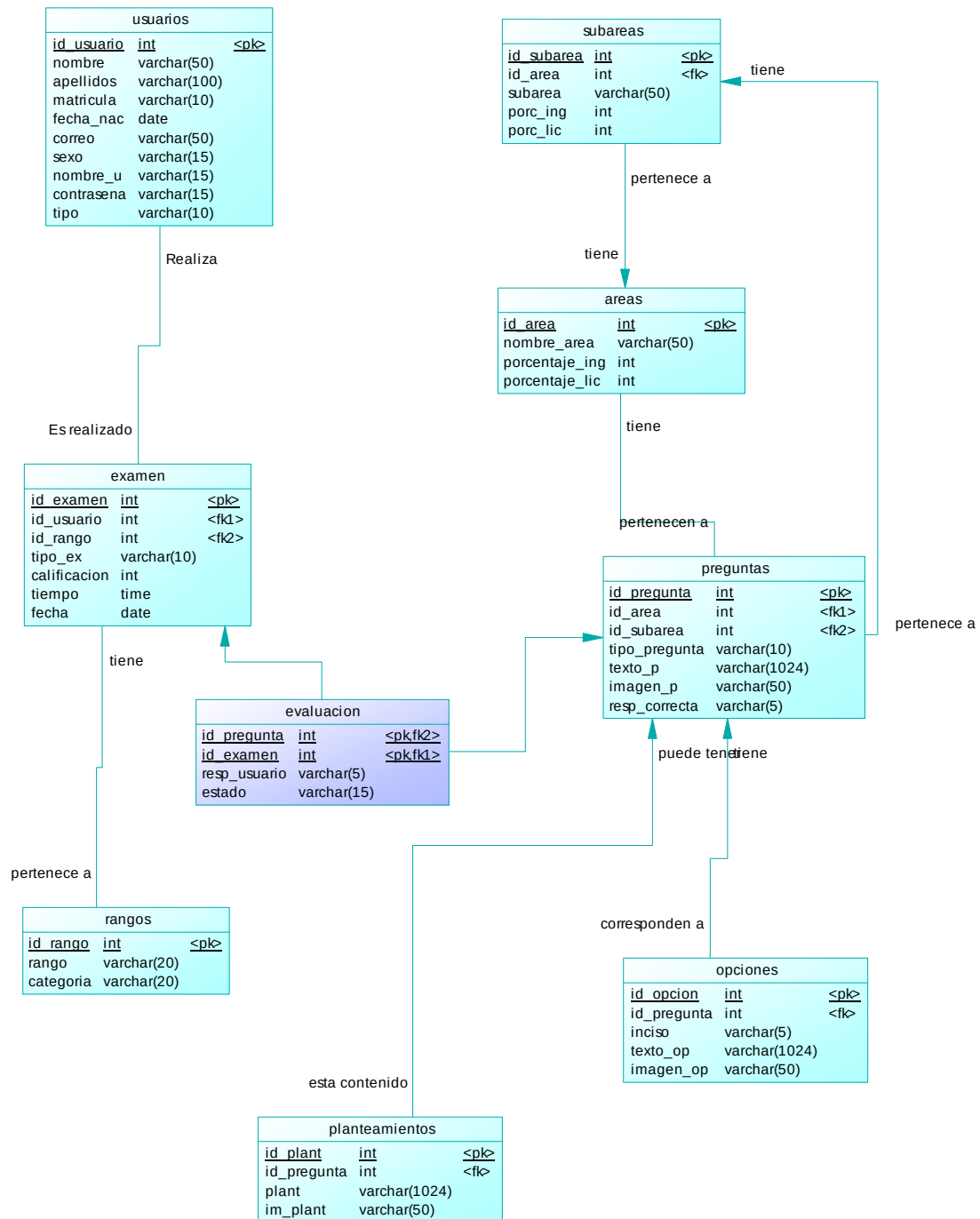


Fig. 3.1 Diagrama Entidad-Relacion

Con el diagrama de la figura 3.1 se obtienen las tablas de la base de datos del sistema.

3.1.2 DESCRIPCIÓN DE ENTIDADES

- **Áreas:** Entidad que almacena los datos de las áreas que conforman al examen.
- **Evaluacion:** Entidad generada por la relación que se establece entre las entidades Examen y Preguntas.
- **Examen:** Entidad que almacena los datos del examen.
- **Opciones:** Entidad que almacena las opciones de las preguntas del examen.
- **Planteamientos:** Esta entidad contiene los planteamientos que puede tener una pregunta del tipo multirreactivo.
- **Preguntas:** Entidad que almacena las preguntas del examen.
- **Rangos:** Entidad que almacena el tipo de rango al que pertenece cada examen de acuerdo con la calificación obtenida por el usuario.
- **Subáreas:** Entidad que almacena los datos correspondientes a las subáreas que pertenecen a cada área.
- **Usuarios:** Entidad que almacena la información personal de los individuos que van a utilizar el sistema.

3.1.3 DESCRIPCIÓN DE LOS ATRIBUTOS

El signo “#” se utiliza para describir el atributo que es LLAVE PRIMARIA, y el atributo subrayado indica que es LLAVE FORANEA.

- **Áreas** (#id_area, nombre_area, porcentaje_ing, porcentaje_lic).
- **Evaluación** (#id_examen, #id_pregunta, resp_usuario, estado).
- **Examen** (#id_examen, id_usuario, id_rango, tipo_ex, calificación, tiempo, fecha).
- **Opciones** (#id_opcion, id_pregunta, inciso, texto_op, imagen_op).
- **Planteamientos** (#id_plant, plant, im_plant).
- **Preguntas** (#id_pregunta, id_area, id_subarea, id_plant, tipo_pregunta, texto_p, imagen_p, resp_correcta).
- **Rangos** (#id_rango, rango, categoria).
- **Subáreas** (#id_subarea, id_area, subarea, porc_ing, porc_lic).
- **Usuarios** (#id_usuario, nombre, apellidos, matricula, fecha_nac, correo, sexo, nombre_u, contraseña, tipo).

Un primer paso en la fase del diseño lógico consiste en la conversión del Modelo Entidad-Relación al Modelo relacional. Siguiendo unas sencillas reglas se obtiene el esquema relacional en el cual se utiliza una simbología parecida al Modelo Entidad- Relación.

3.2 NORMALIZACIÓN

La normalización es un proceso por el cual deben pasar las tablas de toda base de datos relacional, ya que esto ayuda a evitar las anomalías de actualización, evitar redundancia y sobre todo proteger la integridad de los datos.

Las 3 reglas básicas de normalización se describen brevemente a continuación:

1FN: Las celdas de las tablas poseen valores simples y cada celda contiene un solo valor.

2FN: 1) Esta en 1FN
2) Todos los atributos (no llaves) dependen por completo de la llave primaria.

3FN: 1) Esta en 2FN
2) No existen dependencias transitivas.

Por lo tanto es necesario comprobar que las tablas antes esquematizadas cumplan con estas reglas básicas de normalización.

3.2.1 TABLA ÁREAS

1FN: La siguiente tabla contiene atributos atómicos, por tanto esta en 1FN.

#id_area	Nombre_area	Porcentaje_ing	Porcentaje_lic
----------	-------------	----------------	----------------

2FN: La tabla áreas contiene atributos que dependen totalmente de la llave primaria por tanto se encuentra en 2FN.

#id_area	Nombre_area	Porcentaje_ing	Porcentaje_lic
	↑	↑	↑

3FN: También se observa que los atributos de la tabla no poseen dependencias transitivas lo cual confirma que se encuentra en 3FN.

3.2.2 TABLA EVALUACIÓN

1FN: La siguiente tabla contiene atributos atómicos, por tanto esta en 1FN.

#Id_examen	#Id_pregunta	resp_usuario	estado
------------	--------------	--------------	--------

2FN: La tabla evaluación contiene 2 atributos que forman parte de la clave y el atributo resp_usuario y estado son dependientes de la llave, por tanto se encuentra en 2FN.

#Id_examen	#Id_pregunta	resp_usuario	estado
			

3FN: En esta tabla no existen dependencias transitivas lo que significa que cumple con la 3FN.

3.2.3 TABLA EXAMEN

1FN: La tabla contiene solo elementos atómicos, por tanto esta en 1FN

#id_examen	Id_usuario	Id_rango	Tipo_ex	calificación	Tiempo	fecha
------------	------------	----------	---------	--------------	--------	-------

2FN: En la tabla los atributos no llave dependen totalmente de la llave primaria, por tanto la tabla se encuentra en 2FN

#id_examen	Id_usuario	Id_rango	Tipo_ex	calificación	Tiempo	fecha
						

3FN: Ninguno de los atributos depende transitivamente de algún otro atributo, por tanto esta tabla se encuentra en 3FN.

3.2.4 TABLA OPCIONES

1FN: Al observar la tabla se deduce que los atributos son indivisibles, por tanto cumplen con la 1FN.

#id_opcion	<u>Id_pregunta</u>	inciso	texto_op	imagen_op
------------	--------------------	--------	----------	-----------

2FN: Esta tabla contiene atributos no llaves que dependen totalmente de la clave primaria, lo que significa que cumple con la 2FN.

#id_opcion	<u>Id_pregunta</u>	inciso	texto_op	imagen_op
		↑	↑	↑

3FN: Así mismo, entre los atributos de esta tabla no existen dependencias transitivas lo que lleva a la conclusión de que también se encuentra en 3FN.

3.2.6 TABLA PLANTEAMIENTOS

1FN: Todos los atributos de la tabla son atómicos, lo cual significa que cumple con la 1FN

#id_plant	plant	Im_plant
-----------	-------	----------

2FN: Los atributos de la tabla y que no son parte de la clave primaria, dependen absolutamente de la clave, por tanto también cumple con la 2FN.

#id_plant	plant	Im_plant
	↑	↑

3FN: Igualmente en la tabla se observa que no existen dependencias transitivas entre los atributos de la tabla por lo cual cumple con la 3FN.

3.2.7 TABLA PREGUNTAS

1FN: Esta tabla contiene atributos que no pueden dividirse, lo que indica que la tabla se encuentra en 1FN.

#Id_pregunta	<u>id_area</u>	Id_subarea	Id_plant	Tipo_pregunta	texto_p	imagen_p	resp_correcta
--------------	----------------	------------	----------	---------------	---------	----------	---------------

2FN: De igual manera, los atributos no llaves que forman parte de la tabla dependen totalmente de la llave primaria, lo que indica que cumple con la 2FN.

#Id_pregunta	<u>id_area</u>	Id_subarea	Id_plant	Tipo_pregunta	texto_p	imagen_p	resp_correcta
--------------	----------------	------------	----------	---------------	---------	----------	---------------

3FN: La 3FN nos indica que en la tabla no deben existir dependencias transitivas, y al observarla se comprueba que esta tabla cumple con este requisito.

3.2.7 TABLA RANGOS

1FN: En la siguiente tabla se observa que los atributos son indivisibles, por lo tanto cumplen con los requisitos de la 1FN.

#id_rango	rango	categoría
-----------	-------	-----------

2FN: De igual manera se observa que los atributos no llaves de la tabla dependen totalmente de la llave primaria, por lo tanto cumplen con la 2FN.

#id_rango	rango	categoría
-----------	-------	-----------

3FN: También es importante mencionar que ninguno de los atributos tiene dependencias transitivas y lo cual confirma que la tabla rangos se encuentra en 3FN

3.2.8 TABLA SUBÁREAS

1FN: Observando la siguiente tabla se corrobora que los atributos que contiene dicha tabla posee atributos atómicos lo cual confirma que se encuentra en 1FN.

#id_subarea	<u>Id_area</u>	Subarea	Porc_ing	Porc_lic
-------------	----------------	---------	----------	----------

2FN: Los atributos de esta tabla dependen de la clave primaria por lo tanto cumple con la 2FN.

#id_subarea	<u>Id_area</u>	Subarea	Porc_ing	Porc_lic
-------------	----------------	---------	----------	----------

3FN: Esta tabla no posee dependencias transitivas por tanto cumple con la 3FN.

3.2.9 TABLA USUARIOS

1FN : En esta tabla se observa que no tiene atributos repetidos, además los atributos que contiene la tabla solo guardan elementos atómicos por lo tanto comprueba que la tabla Usuarios cumple con la 1FN

#Id_usuario	nombre	apellidos	matricula	fecha_nac	correo	sexo	nombre_u	contrasena	tipo
-------------	--------	-----------	-----------	-----------	--------	------	----------	------------	------

2FN : La segunda forma normal nos dice que todos los atributos no llaves dependen de la llave primaria, por lo tanto, con la esquematización de la tabla podemos observar que se encuentra en 2FN.

#Id_usuario	nombre	apellidos	matricula	fecha_nac	correo	sexo	nombre_u	contrasena	tipo
	↑	↑	↑	↑	↑	↑	↑	↑	↑

3FN: Como se muestra en la tabla, no existe ninguna dependencia transitiva, por lo tanto la tabla Usuarios se encuentra en 3FN.

CAPITULO IV

Implementación y Pruebas

En la actualidad se requiere que los sistemas Web sean dinámicos, atrás han quedado aquellas paginas de Internet que se limitaban a cargarse y ejecutarse, con unos comportamientos inmóviles y una rigidez inaceptable, es por ello que se opto por implementar este sistema de manera que cualquier persona, no importando el lugar en el que se encuentre pueda utilizar esta herramienta que ha sido creada para apoyo de cualquier persona que desee realizar el examen del CENEVAL.

4.1 IMPLEMENTACIÓN DEL SISTEMA

Como ya se mencionó anteriormente, para la creación de la base de datos se utilizo la herramienta PhpMyAdmin en donde se crearon las tablas y campos que servirán para almacenar todos los datos de los usuarios que ingresen al sitio Web.

Una vez terminada la creación de la base de datos se procedió a la realización de la interfaz y la vinculación de esta con la base de datos [L3] [L4] [L9], mediante Dreamweaver como editor y WAMPSEVER como servidor.

4.1.1 PÁGINA PRINCIPAL

El sistema cuenta con una página principal, en la cual los usuarios podrán logearse dependiendo del tipo de usuario y también de los privilegios que éste tenga, los usuarios nuevos podrán registrar sus datos para tener acceso al simulador de exámenes.

Cada usuario tiene funciones específicas, pues existen 3 tipos: Administrador, Coordinador y Usuario, quienes podrán realizar varias operaciones las cuales se describirán más adelante.

Al inicio del sistema se mostrará la página principal de los usuarios (Fig. 4.1) y a continuación se hará una breve descripción de las secciones que la conforman.



Figura 4.1 Página Principal

4.1.1.1 LIGA DE ACCESO DEL ADMINISTRADOR



Figura 4.1.1 Liga de administrador

4.1.1.2 LIGA DE ACCESO DEL COORDINADOR



Figura 4.1.2 Liga de coordinador

4.1.1.3 ACCESO USUARIO



Accede a tu cuenta

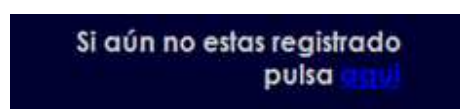


Nombre de Usuario

Contraseña

Figura 4.1.3 Acceso de usuario

4.1.1.4 REGISTRO DE NUEVOS USUARIOS

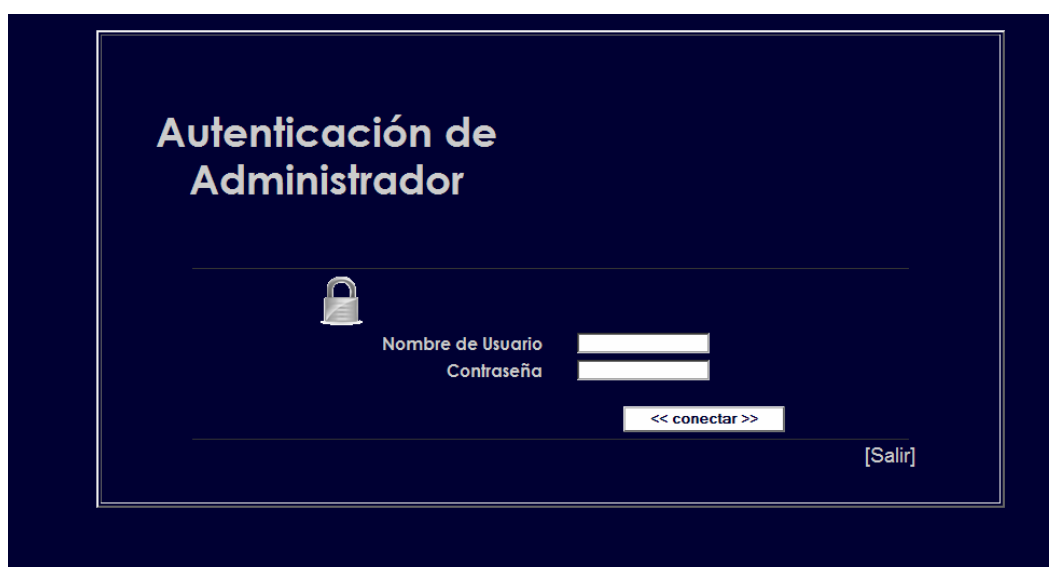


Si aún no estas registrado
pulsas 

Figura 4.1.4 Registro usuarios

4.1.2 ACCESO ADMINISTRADOR

Si el usuario pulsa la liga [Administrador] aparecerá la siguiente pagina (Figura 4.2) en donde se debe autentificar como administrador, de lo contrario no podrá acceder al sistema.



Autenticación de Administrador



Nombre de Usuario

Contraseña

Figura 4.2 Autenticación de Administrador

Una vez ingresando como administrador del sistema se muestra la siguiente pagina y a continuación se describen las operaciones que éste puede realizar:

- Buscar un usuario determinado.
- Ver todos los usuarios así como modificar y borrar la información de los mismos.
- Crear el banco de preguntas para la simulación del examen.
- Modificar y eliminar los reactivos.

4.1.2.1 PÁGINA DE ADMINISTRADOR

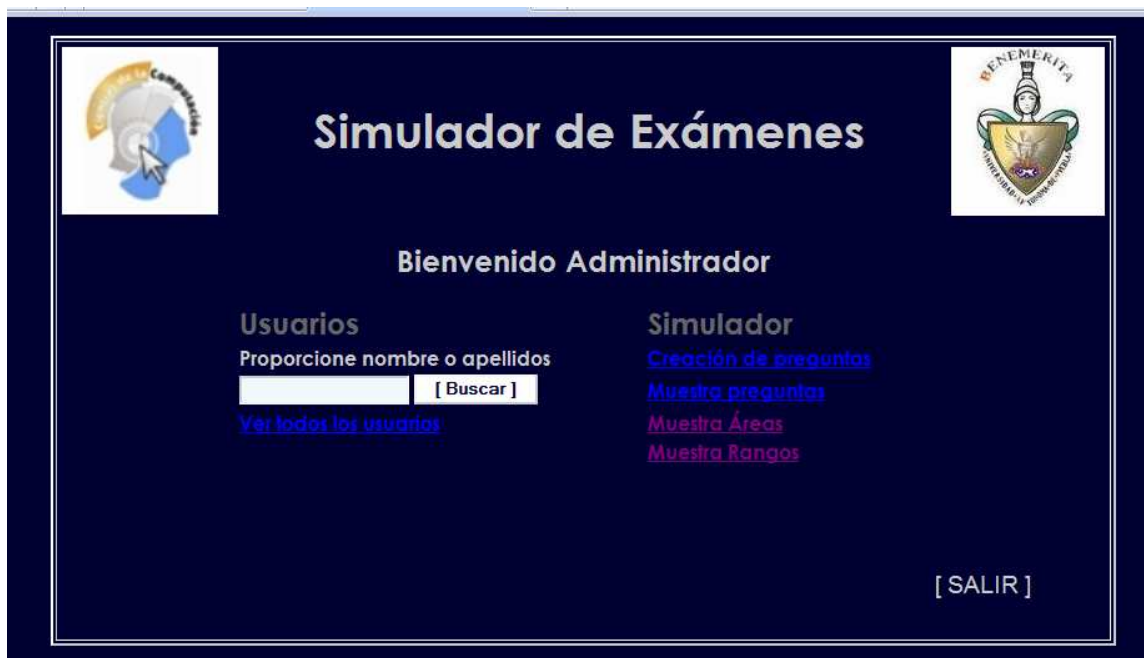


Figura 4.3 Página de Administrador

Si el Administrador desea ver la lista de los usuarios dados de alta en el sistema debe dar clic sobre **Ver todos los usuarios** y el sistema los mostrará en caso de que existan, tal como se muestra en la Figura 4.4

4.1.2.2 MUESTRA USUARIOS



NOMBRE	APELLIDOS	MATRICULA	FECHA DE NACIMIENTO	CORREO	SEXO		
Alejandro	Hernández Campos	200523456	1982-04-24	yunkax@hotmail.com	Masculino	X	
Alejandro Iván	Gutiérrez Hernández	200267546	1984-02-07	alex_ivan@hotmail.com	Masculino	X	
Alma	Torres Garcia	200424653	1985-05-24	almitoga@yahoo.com	Femenino	X	
Gilberto	Andrade Andrade	200067543	1982-02-04	andradex@gmail.com	Masculino	X	
Juan Joel	López Hernández	200312546	1985-02-11	cucoman9@hotmail.com	Masculino	X	
Luz Natalia	Oriza Maldonado	200365478	1985-07-23	luzna_om@hotmail.com	Femenino	X	
Mirelly	Sotero Vázquez	200456784	1984-12-07	miresv7@hotmail.com	Femenino	X	
Miriam	Salgado Morales	200456789	1986-03-17	miriam_sm@hotmail.com	Femenino	X	
Nely	Sánchez Reyes	200123456	1984-02-10	saren_utopia@hotmail.com	Femenino	X	

[SALIR]

Figura 4.4 Muestra Usuarios Registrados

Así mismo se muestran los iconos de Eliminar y Editar, en donde el Administrador podrá realizar tal operación al usuario que desee.

Al dar clic sobre la opción Creación de Preguntas, aparecerá la siguiente página, en donde describe una breve explicación sobre los tipos de reactivos que maneja el sistema y también el Administrador deberá elegir entre las dos opciones [REACTIVO SIMPLE] o [MULTIRREACTIVO]

4.1.2.3 SELECCIONAR TIPO DE PREGUNTA



Figura 4.5 Tipos de Reactivos

4.1.2.4 CREACIÓN DE PREGUNTA SIMPLE

En esta página se dará de alta el reactivo simple

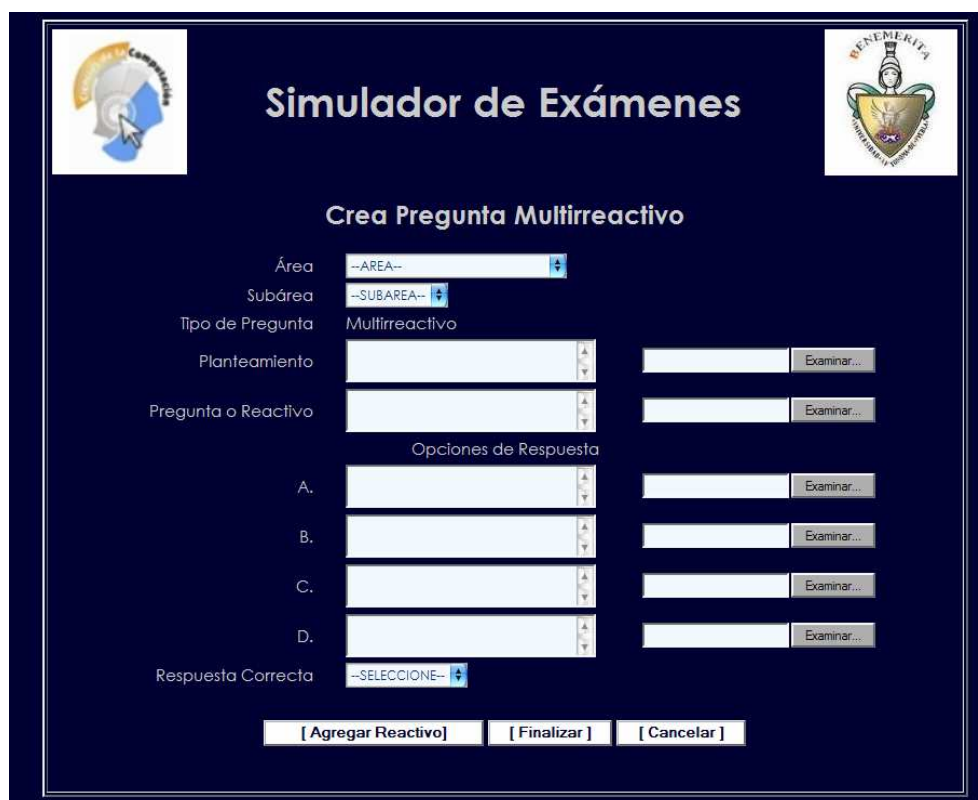


The screenshot displays a web-based simulation interface titled 'Simulador de Exámenes'. It features a dark blue background with white text and form elements. At the top, there are two logos: a circular one on the left and a shield-shaped one on the right. The main heading 'Simulador de Exámenes' is centered at the top. Below it, the section 'Crea Pregunta Simple' is highlighted. The form includes several input fields and buttons: 'Área' and 'Subárea' are dropdown menus; 'Tipo de Pregunta' is a dropdown set to 'Simple'; 'Pregunta' is a text input field with an 'Examinar...' button; 'Opciones de Respuesta' consists of four rows labeled A, B, C, and D, each with a text input field and an 'Examinar...' button; 'Respuesta Correcta' is a dropdown menu set to '--SELECCIONE--'; and at the bottom, there are two buttons labeled '[Enviar]' and '[Cancelar]'.

Figura 4.6 Crea Pregunta Simple

4.1.2.5 CREACIÓN DE PREGUNTA MULTIRREACTIVO

En la siguiente página se creará la pregunta multirreactivo, básicamente es parecido al reactivo simple, la diferencia radica en que éste cuenta con un área de texto adicional en donde se escribirá el planteamiento que identifica a las preguntas Multirreactivo, al presionar el botón [Agregar Reactivo] se agregará otro reactivo que va a contener el mismo planteamiento.



The screenshot displays a web-based form titled "Simulador de Exámenes" (Exam Simulator). The form is for creating a "Pregunta Multirreactivo" (Multiple Choice Question). It includes fields for "Área" (Area) and "Subárea" (Subarea), both with dropdown menus. The "Tipo de Pregunta" (Question Type) is set to "Multirreactivo". There are four input fields for "Planteamiento" (Stem), "Pregunta o Reactivo" (Question or Item), and four options labeled A, B, C, and D. Each option has a corresponding "Examinar..." button. At the bottom, there is a "Respuesta Correcta" (Correct Answer) dropdown menu and three buttons: "[Agregar Reactivo]", "[Finalizar]", and "[Cancelar]".

Figura 4.7 Creación de Preguntas Multirreactivo.

4.1.2.6 MUESTRA ÁREAS

Cuando el usuario desea ver las áreas existentes debe hacer clic sobre **Ver Áreas** y aparecerá la siguiente página en donde se muestran las áreas dadas de alta, así como los datos de las mismas, para conocer las subáreas asociadas es necesario dar clic sobre **Subáreas**, en este ejemplo se dio clic sobre las subáreas del área Entorno Social y los resultados se muestran en la Figura 4.9

De igual forma que en las páginas donde se muestra a los usuarios y las preguntas, en esta página se muestran los iconos para eliminar y modificar las áreas existentes, en caso de que se requiera crear una nueva área se debe hacer clic sobre [**AREA NUEVA**]



Simulador de Exámenes



[AREA NUEVA]

NUM	NOMBRE DE AREA	PORCENTAJE ING	PORCENTAJE LIC		
1	Entorno Social	10	10	Subáreas	
2	Matemáticas	17	25	Subáreas	
3	Hardware	33	20	Subáreas	
4	Software	29	30	Subáreas	
5	Tratamiento de la Información	11	15	Subáreas	

[SALIR]

Figura 4.8 Muestra Áreas

[AREA NUEVA]					
NUM	NOMBRE DE AREA	PORCENTAJE ING	PORCENTAJE LIC		
1	Entorno Social	10	10	Subáreas	
2	Matemáticas	17	25	Subáreas	
3	Hardware	33	20	Subáreas	
4	Software	29	30	Subáreas	
5	Tratamiento de la Información	11	15	Subáreas	

SUBAREAS ASOCIADAS				
NOMBRE DE SUBAREA	PORCENTAJE ING	PORCENTAJE LIC		
Las Organizaciones	5	5		
Las Unidades de Informática, Ética y Legislación	5	5		


[SALIR]

Figura 4.9 Muestra Subáreas Asociadas

4.1.2.7 MUESTRA RANGOS

Dentro de las actividades que puede realizar el administrador esta la de Ver Rangos en la cual se observan los rangos existentes para la calificación del examen y los porcentajes que a cada area corresponden de acuerdo al tipo de examen que el usuario presente, es decir estos varían, hasta la fecha el rango establecido es el que esta registrado en la guía que proporciona la página del CENEVAL, pero en caso de que estos cambiaran se puede realizar esta operación, ya que la página tiene las opciones de eliminar, modificar y crear un rango nuevo, como se muestra en la Figura 4.10



Simulador de Exámenes



[RANGO NUEVO]

Muestra Rangos

RANGO	CATEGORIA	
0-999	No Satisfactorio	 
1000-1149	Satisfactorio	 
1150-1300	Sobresaliente	 

 [SALIR]

Figura 4.10 Muestra Rangos

4.1.3 ACCESO COORDINADOR

Así mismo el coordinador debe autenticarse, para ingresar al sistema. (Fig. 4.11)

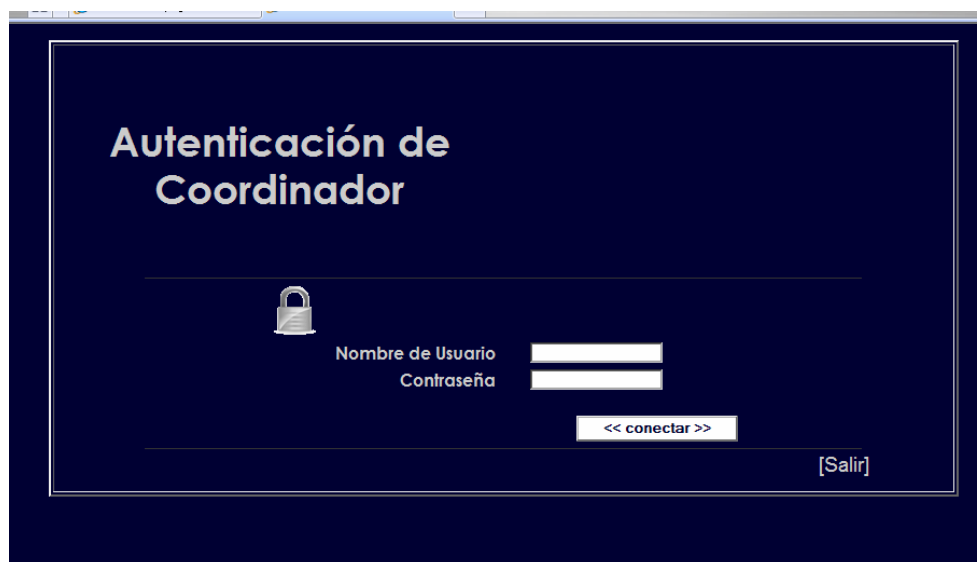


Figura 4.11 Autenticación de Coordinador

4.1.3.1 PÁGINA DE COORDINADOR

En la página de coordinador que se muestra a continuación (Figura 4.9 Coordinador), se observa que el coordinador puede ver las puntuaciones por carrera, las puntuaciones generales que han obtenido los usuarios, puntuaciones por generación, puntuaciones por área, así como puede observar las gráficas que se generan con la información antes mencionada.

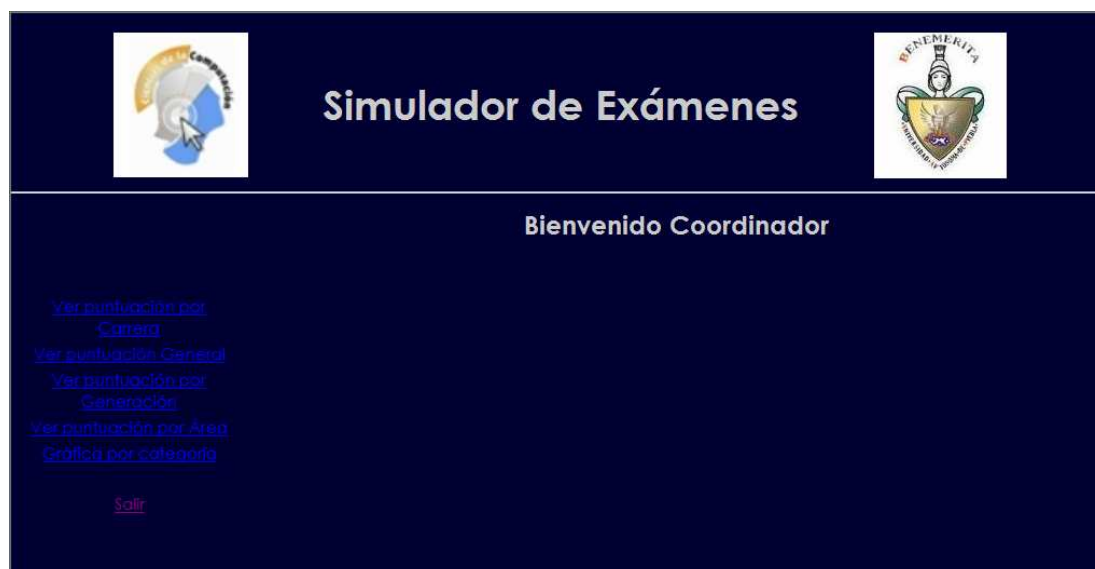


Figura 4.12 Página de Coordinador

Aunque el coordinador no puede realizar ninguna modificación a los resultados, solo puede analizarlos con el fin de verificar si el sistema tiene la funcionalidad y demanda deseada.

4.1.4 REGISTRO DE DATOS DE USUARIOS

En esta página se procede al registro de datos de usuarios nuevos, con el fin de que en futuras ocasiones, puedan ingresar al sistema y realizar la simulación del exámenes, así mismo llevar un registro de datos en donde se muestren las calificaciones obtenidas por el usuario, las veces que ha realizado la simulación y el tiempo que ha demorado en resolver la prueba, ya que con estos datos se podrán construir estadísticas en cuanto a la funcionalidad y demanda de este sistema.



The image shows a web application interface titled "Simulador de Exámenes". It features a dark blue background with a white border. In the top left corner is a logo with a stylized head and the text "CENEVAL". In the top right corner is the official coat of arms of Mexico. The main heading "Simulador de Exámenes" is centered at the top in a large, bold, white font. Below this, the section "Nuevo Registro" is centered. The form contains several input fields: "Nombre(s)" (text), "Apellidos" (text), "Matricula" (text), "Fecha de Nacimiento" (with dropdowns for "AÑO", "MES", and "DIA"), "Correo" (text), and "Sexo" (a dropdown menu showing "--Seleccione--"). Below these fields, the instruction "Selecciona tu nombre de usuario y contraseña" is centered. This is followed by "Nombre de usuario" (text) and "Contraseña" (text) fields. At the bottom of the form are two buttons: "[Enviar]" and "[Cancelar]".

Figura 4.13 Registro de datos de usuarios

Cuando el usuario ha dado sus datos y el sistema los ha insertado correctamente se muestra una página en donde se agradece por haber ingresado los datos y se hace un vínculo hacia la página principal (Figura 4.14) con el fin de que el usuario se autentifique y pueda ingresar al sistema como usuario registrado y realizar la simulación de pruebas.

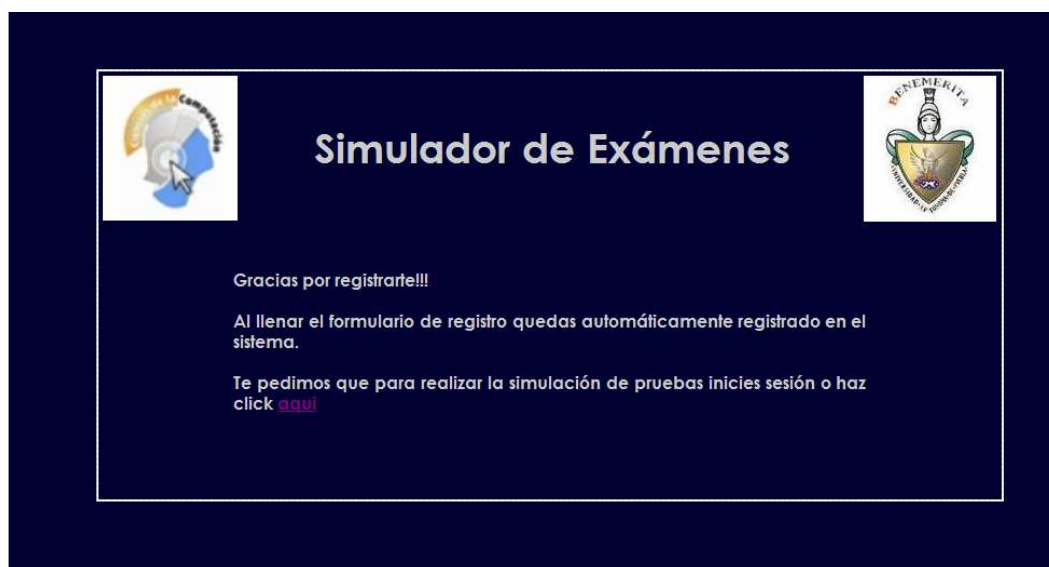


Figura 4.14 Agradecimiento por registro.

4.1.5 PÁGINA DE BIENVENIDA DEL USUARIO

Una vez que el usuario ha realizado el proceso de registro de datos puede acceder al sistema mediante su nombre de usuario y su contraseña, después de ser debidamente verificadas, la página de bienvenida del usuario es la siguiente:

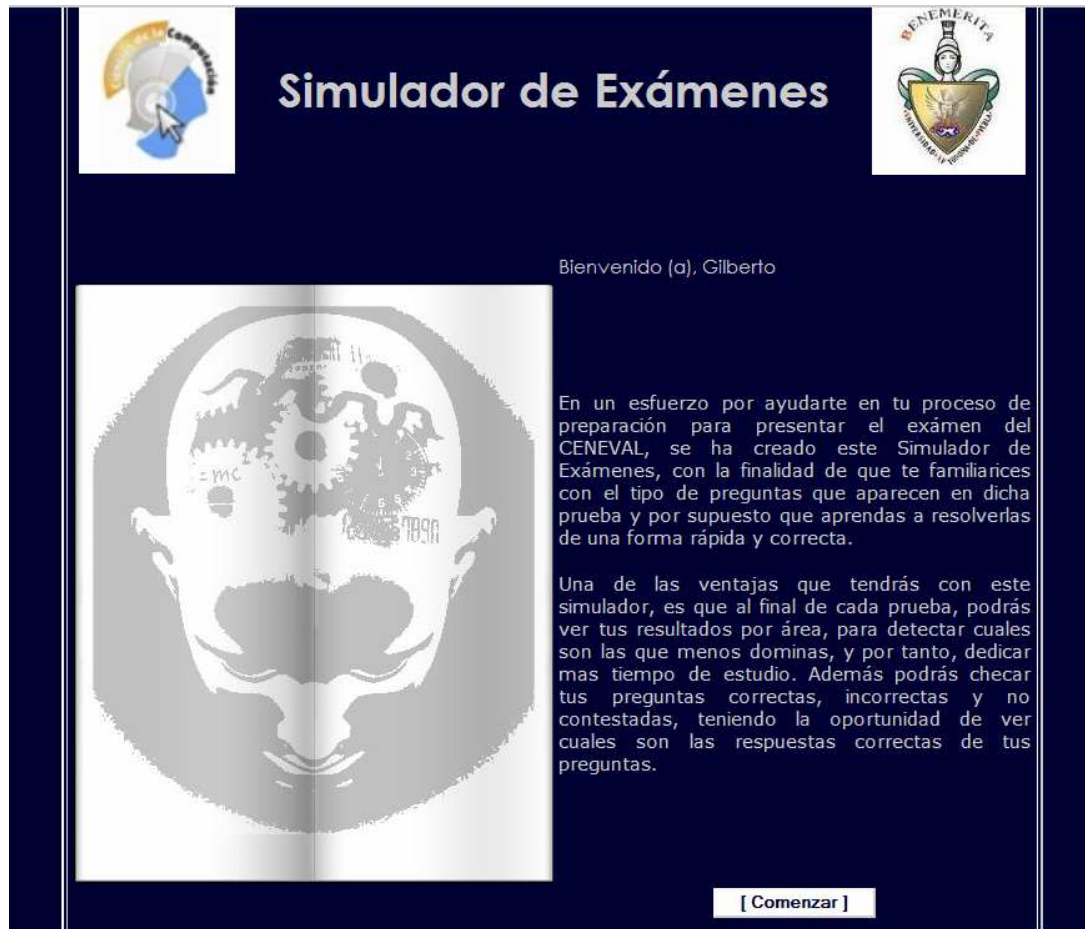


Figura 4.15 Bienvenida

4.3 PRUEBAS DEL SISTEMA

En esta sección se realizarán las pruebas del sistema, con las cuales se determinará el buen funcionamiento del mismo, así como las posibles fallas y/o errores que se puedan presentar.

Es importante señalar que los usuarios que utilizan este simulador deben estar registrados, para llevar a cabo la simulación de pruebas, además de que con su registro se podrá determinar cuanta demanda y utilidad tiene el sistema, por lo tanto se debe verificar que el nombre de usuario y contraseña correspondan a alguno que se encuentre registrado, si esto no ocurre se mostrará un mensaje. (Figura 4.16)

aquí'. At the bottom, in large red letters, is the message: 'NOMBRE DE USUARIO Y/O CONTRASEÑA INCORRECTOS'." data-bbox="191 95 808 451"/>

[Administrador] [Coordinador]

Accede a tu cuenta

Nombre de Usuario

Contraseña

<< conectar >>

Si aún no estás registrado pulsa [aquí](#)

**NOMBRE DE USUARIO
Y/O CONTRASEÑA
INCORRECTOS**

Figura 4.16 Nombre de usuario y/o contraseña incorrectos.

Si los datos fuesen correctos no hay ningún problema y el sistema muestra la página de bienvenida al usuario.

Cuando el usuario desea realizar la evaluación, sólo debe dar clic en el botón comenzar, mostrado en la figura 4.15 y se desplegará la siguiente página, donde se muestran algunas instrucciones sobre la solución del examen y que el usuario debe tener en cuenta, ya que también debe presentar el examen que le corresponde ya sea de Ingeniería o Licenciatura, ya que aunque ambos exámenes pertenecen a la carrera de Ciencias de la Computación, el porcentaje de reactivos con respecto a cada área es en algunos casos diferente y preferentemente el usuario debe realizar el que más le acomode a sus necesidades.



Figura 4.17 Instrucciones Generales

En cuestión de vista, los exámenes son parecidos lo que difiere es el número de reactivos de cada área, por lo cual en la siguiente pantalla se muestra el inicio de la evaluación, para iniciar el desplegado de preguntas el usuario debe dar clic en la liga iniciar examen. Al igual que empezará a correr cronometro el cual indicará el tiempo que esta determinado para la solución de la prueba.

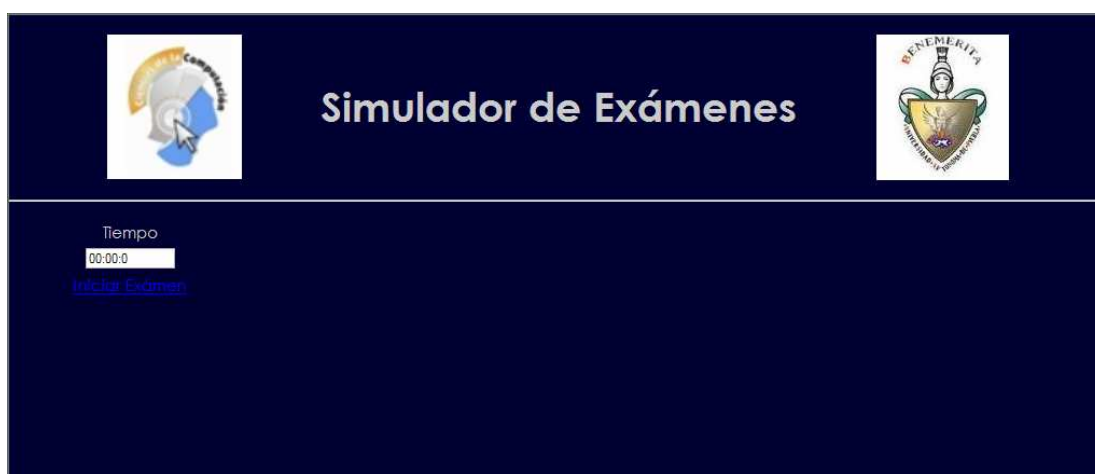


Figura 4.18 Inicio de Examen

Simulador de Exámenes

Tiempo
00:00:43.0
[Iniciar Examen](#)

Área::: Entorno Social
Subárea::: Las organizaciones

Instrucciones: Selecciona la opción correcta.

Es una organización que a través de la administración de sus recursos humanos y financieros produce bienes y/o servicios, para satisfacer las necesidades de la sociedad.

Opciones:

- ☐ A) Empresa.
- ☐ B) Asociación
- ☐ C) Fábrica.
- ☐ D) Industria.

[Siguiente]

Figura 4.19 Desplegado de preguntas

Una vez que el usuario ha terminado de contestar todas las preguntas, debe dar clic sobre el icono  para evaluar la prueba y saber los resultados que ha obtenido. Los cuales se mostraran en la Figura 4.22

Los resultados que obtuviste en la prueba son los siguientes:		
Total de Preguntas	54	
Correctas	45	
Incorrectas	7	
No respondidas	2	
Puntuación	1083	
Categoría	Satisfactorio	
INDICE GLOBAL :: 1051		
ÁREAS / SUBÁREAS	RESULTADO	
ENTORNO SOCIAL	1300	
Las organizaciones		1300
Las unidades de Ética, Informática y Legislación		1300
MATEMÁTICAS	975	
Matemáticas Básicas		650
Matemáticas Aplicadas y Computacionales		1300
HARDWARE	1083	
Física y Electrónica		1083
Arquitectura de Computadoras		867
Redes		1300
SOFTWARE	1032	
Software de Base		1300
Programación		929
Ingeniería de Software		867
TRATAMIENTO DE LA INFORMACIÓN	867	
Tratamiento de la Información		867
Interacción Hombre-Máquina		1300

Figura 4.20 Resultados del Examen

Así mismo, cuando algún usuario trate de acceder al sistema como Administrador o Coordinador, se le pedirá que se identifique, igualmente con un nombre de usuario y contraseña, ya que la información que se guardará en el sistema es confidencial.

Por lo tanto, en caso de no corresponder los nombres de usuario y/o contraseña tanto del administrador como del coordinador, se desplegará un mensaje como el que se muestra en la Figura 4.18.



Figura 4.21 Acceso Denegado.

Las siguientes pruebas realizadas fueron hechas al Menú del Administrador, en donde éste puede escoger entre varias opciones:

Si se desea realizar la búsqueda de un determinado usuario, conociendo su nombre y/o apellidos, se realiza sólo introduciendo estos datos en la caja de texto y presionado el botón [Buscar], como se muestra en la Figura 4.22

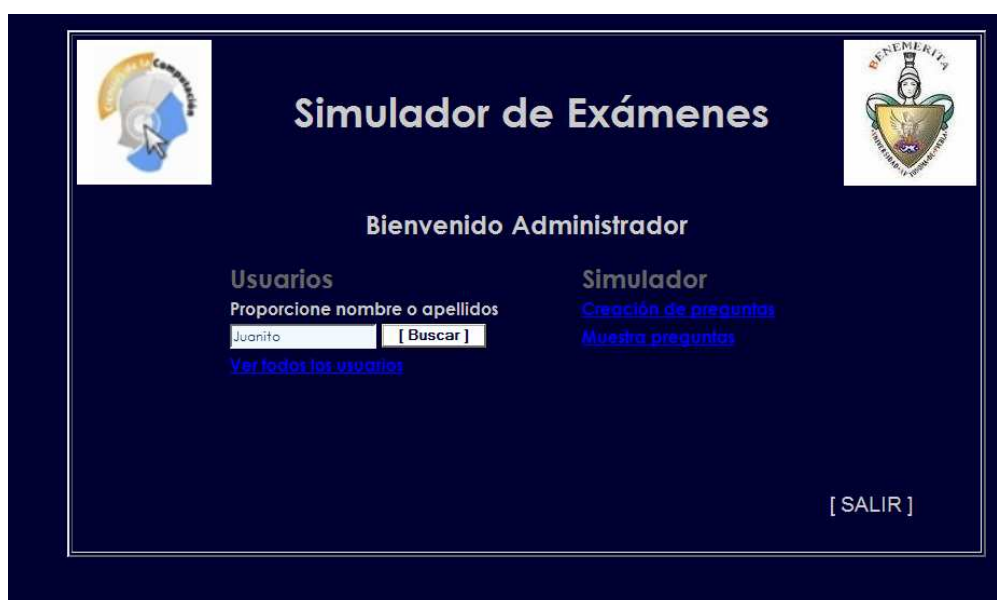


Figura 4.22 Buscar Usuario.

En caso de no encontrarse el nombre aparecerá un mensaje en el que se especifica que el usuario no se encontró registrado en el sistema. (Figura 4.23)

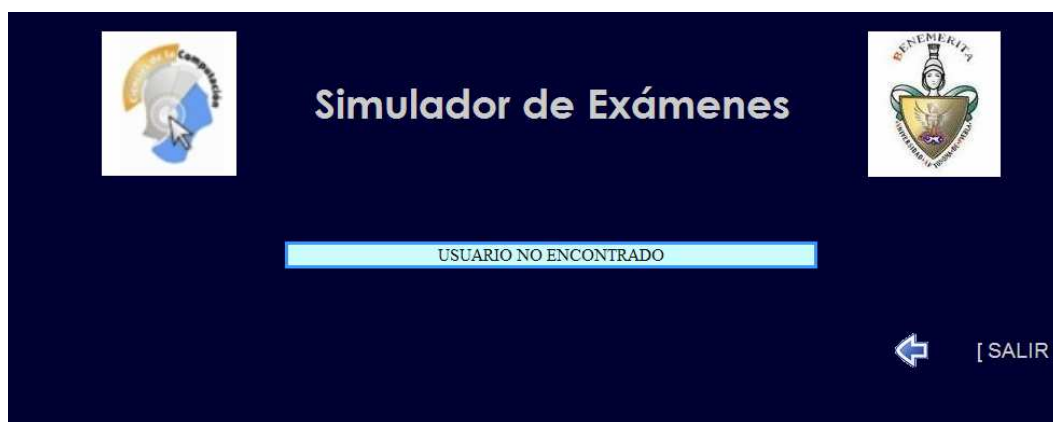


Figura 4.23 Usuario no encontrado.

Si el Administrador desea ver la información de todos los usuarios esta será desplegada en una página, en caso de no encontrarse ningún usuario registrado, el sistema desplegará un mensaje como el que se muestra en la Figura 4.13

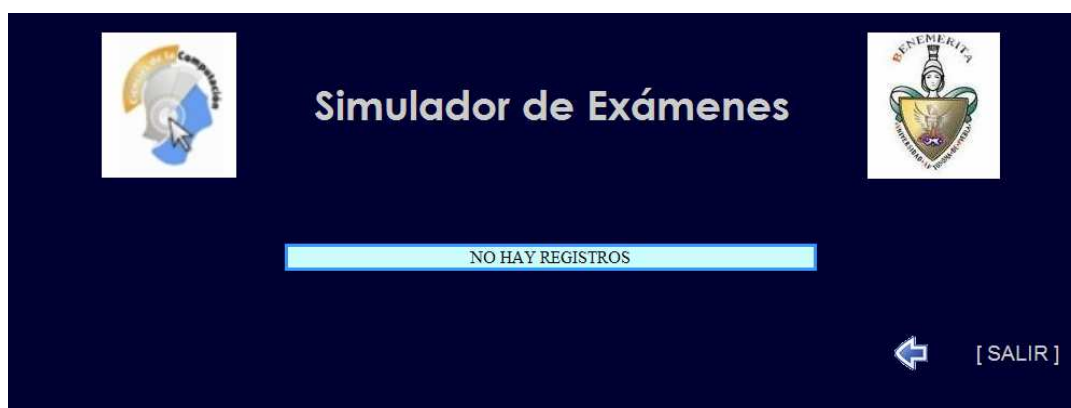
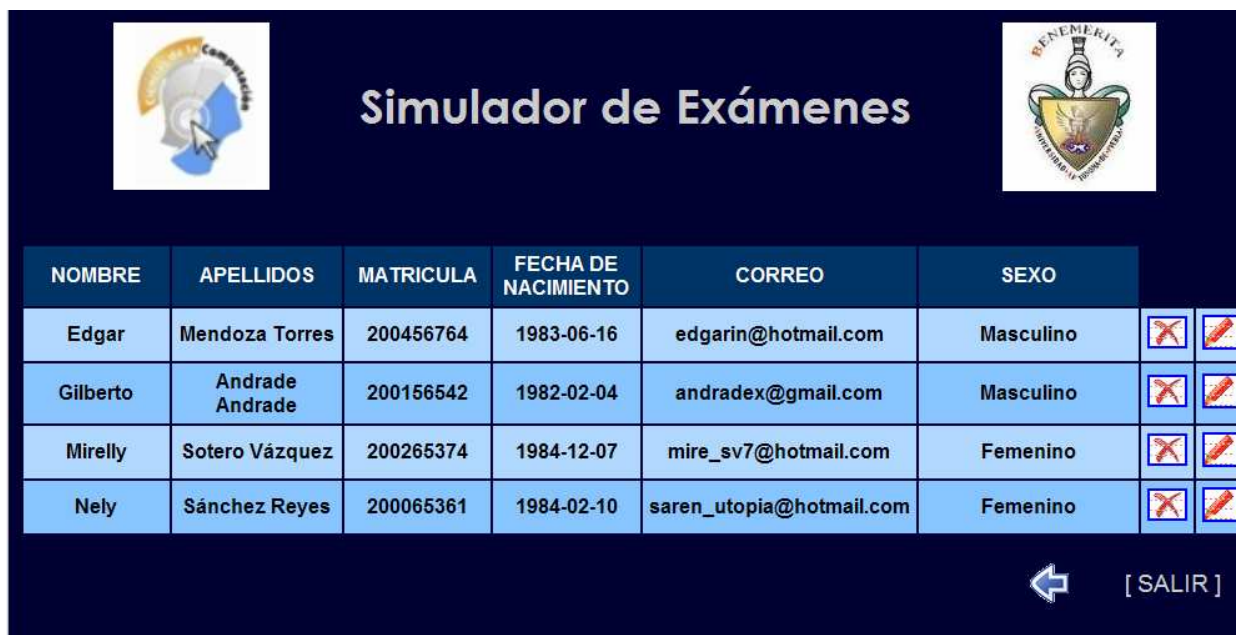


Figura 4.24 No hay registros

La página que muestra los usuarios registrados también mostrará el icono  para eliminar el usuario seleccionado, y utilizando el icono  se actualiza la información del mismo.

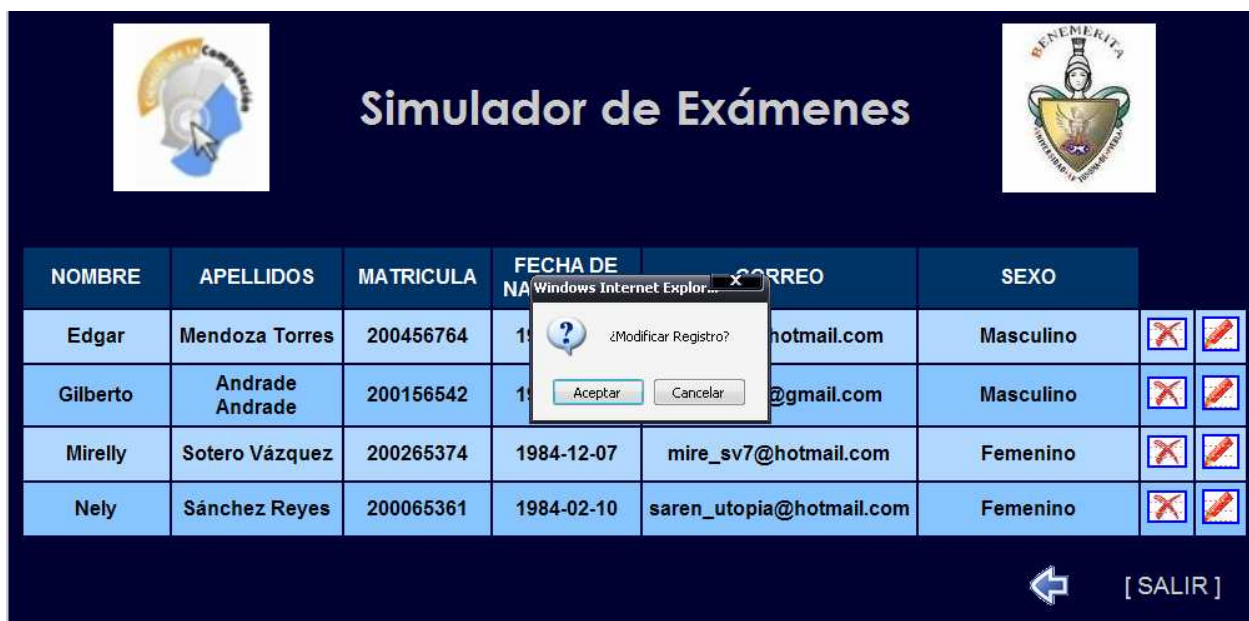


The interface features a dark blue header with two logos: a stylized head with a cursor on the left and the Benemérita logo on the right. The title 'Simulador de Exámenes' is centered in the header. Below the header is a table with six columns: NOMBRE, APELLIDOS, MATRICULA, FECHA DE NACIMIENTO, CORREO, and SEXO. The table contains four rows of user data. To the right of the table are two icons: a red 'X' and a red pencil. At the bottom right is a back arrow icon and the text '[SALIR]'.

NOMBRE	APELLIDOS	MATRICULA	FECHA DE NACIMIENTO	CORREO	SEXO
Edgar	Mendoza Torres	200456764	1983-06-16	edgarin@hotmail.com	Masculino
Gilberto	Andrade Andrade	200156542	1982-02-04	andradex@gmail.com	Masculino
Mirelly	Sotero Vázquez	200265374	1984-12-07	mire_sv7@hotmail.com	Femenino
Nely	Sánchez Reyes	200065361	1984-02-10	saren_utopia@hotmail.com	Femenino

Figura 4.25 Muestra Usuarios

En ambos casos se desplegará un cuadro de diálogo para verificar que el usuario desea realizar tal acción, como el que se muestra en la Figura 4.15



The interface is identical to Figure 4.25, but with a dialog box overlaid on the table. The dialog box has a question mark icon and the text '¿Modificar Registro?'. It has two buttons: 'Aceptar' and 'Cancelar'. The dialog box is positioned over the first row of the table.

NOMBRE	APELLIDOS	MATRICULA	FECHA DE NACIMIENTO	CORREO	SEXO
Edgar	Mendoza Torres	200456764	1983-06-16	edgarin@hotmail.com	Masculino
Gilberto	Andrade Andrade	200156542	1982-02-04	andradex@gmail.com	Masculino
Mirelly	Sotero Vázquez	200265374	1984-12-07	mire_sv7@hotmail.com	Femenino
Nely	Sánchez Reyes	200065361	1984-02-10	saren_utopia@hotmail.com	Femenino

Figura 4.26 Cuadro de Diálogo.

En seguida de confirmar que se desea realizar tal acción, se muestra la página en donde se podrá modificar la información, la cual se observa en la Figura 4.16





Simulador de Exámenes



Modifica Registro

Nombre(s)	Gilberto
Apellidos	Andrade Andrade
Matricula	200067543
Fecha de Nacimiento	1982-02-04
Correo	andradex@gmail.com
Sexo	Masculino

Figura 4.27 Modifica Registro

Cuando se ha modificado y/o eliminado algún registro se envía un mensaje de notificación de que la operación se realizó con éxito, como se muestra en la Figura 4.17





Simulador de Exámenes



NOMBRE	APELLIDOS	MATRICULA	FECHA DE NACIMIENTO	CORREO	SEXO		
Edgar	Mendoza Torres	200456764	1983-06-16	edgarin@hotmail.com	Masculino		
Gilberto	Andrade Andrade	200156542	1982-02-04	andradex@gmail.com	Masculino		
Mirelly	Sotero Vázquez	200265374	1984-12-07	mire_sv7@hotmail.com	Femenino		
Nely	Sánchez Reyes	200065361	1984-02-10	saren_utopia@hotmail.com	Femenino		

EL USUARIO FUE
MODIFICADO EXITOSAMENTE



Figura 4.28 Mensaje de Notificación.

De igual manera como se maneja la información de los usuarios, es manejada la información de los reactivos o preguntas, la página en donde se muestran las preguntas contiene el área

de la pregunta y la pregunta, también se pueden visualizar el icono  para eliminar la pregunta, el icono  para modificar y a diferencia de la página en donde se muestra la información de los usuarios, ésta contiene un icono adicional,  con el cual se visualiza toda la información de la pregunta seleccionada, como se muestra en la Figura 4.19

 Simulador de Exámenes 		
AREA	PREGUNTA	
Entorno Social	¿Cómo se clasifican las organizaciones de acuerdo con la actividad o giro que realizan?	  
Entorno Social	Es una organización que a través de la administración de sus recursos humanos y financieros produce bienes y/o servicios, para satisfacer las necesidades de la sociedad.	  
Matemáticas	¿Cuál es el orden y grado de la siguiente ecuación diferencial?	  
Entorno Social	Son actividades que se deben efectuar durante la planeación de una auditoría	  
Hardware	En el modelo OSI una de las funciones de la capa de enlace es:	  
Software	¿Cuál es el propósito del análisis lexicográfico?	  
Software	Recorriendo el árbol de izquierda a derecha, ¿Cuántos nodos se visitan hasta llegar a K, si se hace una búsqueda en profundidad (depht-first)	  
Entorno Social	Es la influencia interpersonal ejercida en una situación dirigida a la consecución de uno o diversos objetivos específicos.	  
Entorno Social	Es el conjunto de fuerzas internas o externan que generan, dirigen o mantienen una conducta hacia un objetivo determinado y tiene por objetivo impulsar y	  

Figura 4.29 Muestra Preguntas



Simulador de Exámenes



Área Matemáticas

Subárea Matemáticas Básicas

Tipo de Pregunta simple

Pregunta ¿Cuál es el orden y grado de la siguiente ecuación diferencial?

$$\left(\frac{d^3 y}{dx^3}\right)^2 + 2\frac{d^2 y}{dx^2} + x^2\left(\frac{dy}{dx}\right)^3 = 0$$

A. 3,2

B. 2,3

C. 3,3

D. 2,5

Inciso A

Correcto

Figura 4.30 Muestra Datos de Pregunta

CONCLUSIONES Y PERSPECTIVAS

Después de realizar todas las pruebas necesarias para la comprobación de la funcionalidad del sistema, se concluye que, el realizar un proyecto Web es una tarea laboriosa, ya que se deben considerar muchos factores que a simple vista no son necesarios o importantes y que gracias a las técnicas de Ingeniería de Software se puede realizar cualquier tipo de proyecto.

En este caso se realizó un sistema que va a ser de utilidad para las futuras generaciones de egresados y que les brindará mayor apoyo en el estudio y sobre todo en la práctica de conocimientos, por el momento este proyecto sólo está adecuado para la realización de exámenes de prueba, pero quizás en un futuro no muy lejano, esta sea la nueva forma de evaluación de los estudiantes y como computólogos debemos ser los pioneros en esta forma de evaluación.

Y no sólo en la facultad de Ciencias de la Computación será posible aplicar este tipo de proyecto, en cualquier otra facultad independientemente de la carrera, ya que será un medio confiable y sobre todo muy fácil y rápido.

El sistema en su funcionamiento básico, cuenta con varias consultas realizadas tanto por el administrador como por el coordinador, el administrador puede realizar varias tareas, como son: consulta, modificación y eliminación de usuarios; creación, modificación, eliminación y consulta del banco de preguntas, el cual es una parte esencial para el funcionamiento de este proyecto, el coordinador solo podrá revisar datos y analizarlos, no podrá hacer ninguna modificación. El usuario podrá ingresar sus datos, identificarse, realizar la prueba y ver sus resultados.

En la realización de este proyecto de tesis, se realizaron consultas en el lenguaje PHP y MySQL, y sobre todo el diseño de bases de datos, algo que es muy importante para la implementación de sistemas y que muchas veces no se define con claridad es la base de datos, y que es la parte indispensable para este tipo de proyectos dinámicos.

También es importante señalar que este proyecto puede tener mejoras importantes a futuro, y la flexibilidad de la base de datos permite hacer modificaciones en las áreas y subáreas que forman parte de los reactivos del mismo, y que son parte del modelo que a la fecha está vigente en la estructura del examen del CENEVAL, sin embargo en caso de que las distintas áreas y subáreas cambiasen se podría adecuar el sistema, dando de baja las áreas y subáreas del modelo anterior, pero si se desea tener datos de alta ambos modelos será necesario cambiar la estructura de la base de datos; así también se puede modificar el rango de calificaciones, en caso de que en algún momento el porcentaje sea diferente.

Otro punto importante es la realización de esta prueba en dispositivos móviles, pues esto sería de mucha utilidad ya que se ahorraría tiempo y esfuerzo.

La computación ofrece una amplia gama de posibilidades para la realización de proyectos y lo mejor es que se pueden combinar para realizar mejor cada tarea, es por ello que también

la minería de datos puede entrar en acción para la mejora de este proyecto pues se puede realizar un análisis de datos bajo esta rama y así obtener mejores resultados.

BIBLIOGRAFIA

LIBROS

- [L1] Abraham Silberschatz, Henry F. Korth, S. Sudarshan ; traducción, Fernando Sáenz Pérez, Antonio García Cordero, Jesús Correas Fernández.
Fundamentos de Bases de datos
McGraw-Hill, 2006.
- [L2] Braude, Eric J.
Ingeniería de Software: una perspectiva orientada a objetos
Alfaomega México; Febrero 2007
ISBN 970-15-0851-3
- [L3] José López Quijado
Domine PHP y MySQL Programación dinámica en el lado del servidor
Alfaomega, México, Mayo 2007
ISBN: 84-7897-751-1
- [L4] Jacobo Pavón Puertas
Creación de un portal con PHP Y MySQL
Alfaomega, México, 2007
ISBN: 978-84-7897-754-3
- [L5] David M. Kroenke; traducción Ana Elizabeth García Hernández.
Procesamiento de base de datos: fundamentos, diseño, e instrumentación
México: Person Educación, c2003.
- [L6] Sommerville, Ian
Ingeniería de Software
Addison Wesley, Mexico, 2003
ISBN: 970-26-0206-8
- [L7] Roger S. Pressman; traducción Rafael Ojeda Martín... [Et al.] ; Coordinación general y revisión técnica Luís Joyanes Aguilar.
Ingeniería del software: un enfoque práctico
Aravaca, Madrid: McGraw-Hill, c2003.
- [L8] R. Windle Daniel, Abreo L. Rene
Software requirements using the unified process: a practical approach
Pearson Education de Mexico, 2003
ISBN: 0-13-096972-9
- [L9] Francisco Charte Ojeda.
Proyectos profesionales PHP 5

Madrid: Anaya Multimedia, 2004.

- [L10] DATE, C.J.
Introducción a los sistemas de bases de datos
Pearson Educación, México, 2001
ISBN: 968-444-419-2

REFERENCIAS EN INTERNET

- [P1] www.ceneval.edu.mx
- [P2] <http://www.monografias.com/trabajos5/inso/inso.shtml>
- [P3] <http://www.angelfire.com/scifi/jzavalar/apuntes/IngSoftware.html>
- [P4] http://es.wikipedia.org/wiki/Desarrollo_de_software
- [P5] <http://docs.kde.org/kde3/es/kdesdk/umbrello/uml-elements.html>
- [P6] <http://es.wikipedia.org/wiki/DBMS>
- [P7] <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>