

Universidad de las Ciencias Informáticas



Facultad 8

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS

*Sistema de Gestión de la Información para la
Vice-dirección de Docencia e Investigaciones
del Policlínico - Hospital Ernesto Che
Guevara.*

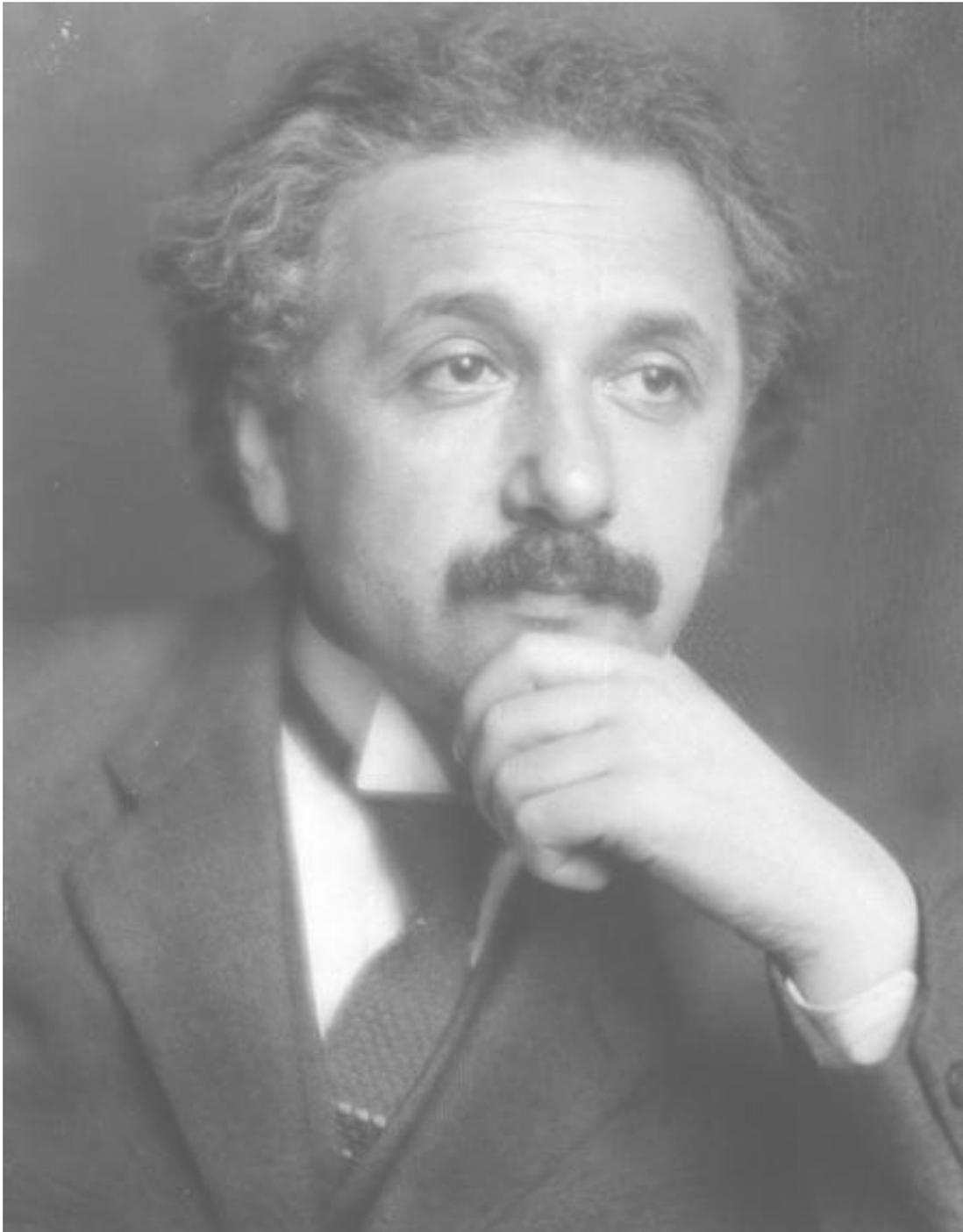
AUTORES:

Lidivel Cabrera Pérez
Oreste Alejandro Moya Rodríguez

TUTORES:

Ing. Osdalme Fuentes Colina
MSc. Edilberto Fernández Cumbá

Ciudad de La Habana, Junio del 2009
“Año del 50 aniversario del triunfo de la Revolución”



"Intenta no volverte un hombre de éxito, sino volverte un hombre de valor"

Albert Einstein.

DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores del Trabajo de Diploma Sistema de Gestión de la Información para la Vice-dirección de Docencia e Investigaciones del Policlínico – Hospital Ernesto Che Guevara y autorizamos al Policlínico – Hospital Ernesto Che Guevara y a la Universidad de las Ciencias Informáticas (UCI) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los ____ días del mes de Junio de 2009.

Firma del Autor

Lidivel Cabrera Pérez

Firma del Autor

Oreste Alejandro Moya Rodríguez

Firma del Tutor

Ing. Osdalme Fuentes Colina

Firma del Tutor

Msc. Edilberto Fernández Cumbá



Dedicatoria

A mis padres y a mi hermano por ser lo más grande que tengo en la vida. Gracias por existir.

Agradecimientos

A mi papá por ser mi gran ejemplo y enseñarme a ser fuerte y seguir adelante.

A mi mamá por ser la mejor mamá del mundo y siempre estar cuando lo necesito.

A mi hermano por ser lo que más quiero en la vida.

A mi familia y a todos mis amigos por su apoyo incondicional.

A Oreste por ser un gran amigo y por su ayuda en estos años.

A Osdalme por ser un tutor excepcional.

A Edilberto por confiar en nosotros y permitirnos desarrollar su idea.

Lidivel

Dedicatoria

A mi mamá, a mi papá, a mi mujer y a mi hijo que es lo más lindo de este mundo.

Agradecimientos

A mis padres, a ellos les debo todo lo que soy.

A Yoanna por apoyarme tanto en todo momento y por permitirme abrazar y besar a la personita más linda del mundo.

A mi hijo por ser la razón de mi vida.

A mi familia y amigos por estar ahí cuando los he necesitado.

A Lidivel, la quiero y estimo mucho por ser tan buena amiga.

A Osdalme y Edilberto por ser parte imprescindible de este trabajo. Gracias por todo el apoyo que nos brindaron.

Oreste



Resumen

La Vice-dirección de Docencia e Investigaciones del Policlínico - Hospital Ernesto Che Guevara requiere de un mayor control sobre todos los documentos e información que se procesa en ella. Por lo expuesto anteriormente es necesario desarrollar un Sistema que permita automatizar el control y la gestión de la información generada por las actividades docentes e investigativas de los trabajadores del centro. El presente trabajo se centró en el análisis, diseño e implementación de un Sistema con el objetivo de mejorar la gestión de la información y minimizar el tiempo de procesamiento de la misma, accediendo a ella en cualquier momento y desde cualquier lugar dentro del Policlínico - Hospital Ernesto Che Guevara. Para la documentación e implementación de este Sistema se siguieron los flujos de trabajo que propone el Proceso Unificado de Desarrollo, como lenguaje de programación por parte del servidor se utilizó PHP y por parte del cliente JavaScript, XML, AJAX y XHTML y se hizo uso del Lenguaje Unificado de Modelado.

Palabras claves: Sistema de Gestión de la Información, Vice-dirección de Docencia e Investigaciones, Policlínico - Hospital Ernesto Che Guevara.



INDICE

INTRODUCCIÓN	1
<i>Situación Problémica</i>	1
CAPÍTULO 1	5
FUNDAMENTACIÓN TEÓRICA	5
<i>Introducción</i>	5
1.1 – Conceptos Esenciales.	5
1.1.1 – Internet	5
1.1.2 – Web o World Wide Web	6
1.1.3 – ¿Qué es una página Web?	6
1.1.4 – Sitio Web	6
1.1.5 – ¿Qué es un Software de Gestión?	7
1.1.6 – Sistemas de Gestión de la Información.	7
1.2 – Tendencias y Tecnologías actuales.	8
1.2.1 – Arquitecturas y Patrones de Diseños	8
1.2.1.1 – Patrón Modelo – Vista – Controlador (MVC)	9
1.2.1.2 – Arquitectura en tres capas.	10
1.2.1.3 – ¿Por qué utilizar la Arquitectura en tres capas?	11
1.3 – Metodologías de Desarrollo de Software	12
1.3.1 – Extreme Programming (XP)	13
1.3.2 – Proceso Unificado del Software	14
1.3.3 – ¿Por qué el uso de RUP?	17
1.4 – Lenguaje de Modelado	17
1.4.1 – UML	17
1.5 – Tecnologías y Lenguajes de Programación del lado del Cliente	18
1.5.1 – XHTML (eXtensible HyperText Markup Language)	18
1.5.2 – Javascript	18
1.5.3 – CSS (Cascading Style Sheets)	19
1.5.4 – XML (eXtensible Markup Language ó Lenguaje extensible de marcas)	19
1.5.5 – AJAX (Javascript y XML Asíncronicos)	20
1.6 – Tecnologías y Lenguajes de Programación del Lado del Servidor	20
1.6.1 – PHP (Personal Home Page)	21
1.6.2 – ASP.NET (Active Server Pages)	21
1.6.3 – ¿Por qué usar PHP?	22
1.7 – Servidor Web	22
1.7.1 – Servidor Apache	23
1.7.2 – IIS (Internet Information Server)	24
1.7.3 – ¿Por qué utilizar un Servidor Apache?	24
1.8 – Sistemas gestores de Bases de datos	25
1.8.1 – MySQL	25
1.8.2 – PostgreSQL	26
1.8.3 – Oracle	27
1.8.4 – ¿Por qué usar MySQL?	27
1.9 – Herramientas a utilizar.	28
1.9.1 – Rational Rose Enterprise Edition	28
1.9.2 – EMS MySQL Manager	28
1.9.3 – Dreamweaver	29
1.9.4 – ER/Studio	31
1.9.5 – Aptana Studio	31
1.9.6 – Zend Studio	32
<i>Conclusiones</i>	33
CAPÍTULO 2	34
CARACTERÍSTICAS DEL SISTEMA	34
<i>Introducción</i>	34

<i>Modelo de Negocio</i>	34
2.1 – Modelo de Negocio	34
2.1.1 – Reglas del Negocio	35
2.1.2 – Actores del Negocio	35
2.1.3 – Trabajadores del Negocio	36
2.1.4 – Entidades del Negocio	36
2.1.5 – Casos de uso del Negocio	36
2.1.6 – Diagrama de Casos de uso del Negocio	37
2.1.7– Descripción de los Principales Casos de uso del Negocio	37
2.1.7.1– Descripción del Caso de uso del Negocio Gestionar proceso de capacitación de los trabajadores	37
2.1.7.2 – Descripción del Caso de uso del Negocio Gestionar proceso de superación de los trabajadores	38
2.1.7.3– Descripción del Caso de uso del Negocio Controlar Plan de Trabajo Individual de los trabajadores	39
2.1.7.4 – Descripción del Caso de uso del Negocio Gestionar Actividad Científica del Centro	40
2.1.8 – Modelo de Objetos	42
<i>Modelo de Sistema</i>	42
2.2 – Especificación de Requerimientos para el Sistema de Gestión de la Información de la VDDI	42
2.2.1 – Requisitos Funcionales	42
2.2.2 – Requisitos No Funcionales	44
2.2.3 – Actores del Sistema	45
2.2.4 – Diagrama de Casos de uso del Sistema	46
2.2.5 – Descripción de Casos de uso del Sistema	46
2.2.5.1 – Descripción del Caso de uso del Sistema Autenticar Usuario	46
2.2.5.2 – Descripción del Caso de uso del Sistema Gestionar Trabajador	47
2.2.5.3 – Descripción del Caso de uso del Sistema Buscar Trabajador	49
2.2.5.4 – Descripción del Caso de uso del Sistema Listar Trabajador	50
2.2.5.5 – Descripción del Caso de uso del Sistema Generar Reportes	51
<i>Conclusiones</i>	54
CAPÍTULO 3	55
ANÁLISIS Y DISEÑO DEL SISTEMA	55
Introducción	55
Análisis del Sistema	55
3.1 – Diagramas de clases del análisis.	55
3.1.1 – Diagrama de clases de análisis de Autenticar Usuario	56
3.1.2 – Diagrama de clases de análisis de Buscar trabajador	56
3.1.3 – Diagrama de clases de análisis de Generar Reporte (Control de Créditos)	56
3.1.4 – Diagrama de clases de análisis de Gestionar trabajador	57
3.1.5 – Diagrama de clases de análisis de Listar trabajador	57
3.2 – Diagramas de Interacción.	57
3.2.1 – Diagrama de secuencia de Autenticar Usuario	58
3.2.2 – Diagrama de secuencia de Generar Reporte (Control de Créditos)	58
3.2.3 – Diagrama de secuencia de Listar trabajador	59
3.2.4 – Diagrama de secuencia de Buscar trabajador	60
3.2.5 – Diagrama de secuencia de Gestionar trabajador	61
Diseño del Sistema	61
3.3 – Diagrama de clases del diseño.	62
3.3.1 – Diagrama de clases del diseño de Autenticar Usuario	62
3.3.2 – Diagrama de clases del diseño de Listar Trabajador	63
3.3.3 – Diagrama de clases del diseño de Generar Reportes (Control de Créditos)	63
3.3.4 – Diagrama de clases del diseño de Buscar Trabajador	64
3.3.5 – Diagrama de clases del diseño de Gestionar Trabajador	65
3.4 – Descripción de las clases del diseño.	66
3.4.1 – Descripción de las clases de Acceso a Datos	66
3.5 Diagrama de clases persistentes.	69
<i>Conclusiones</i>	69





CAPÍTULO 4	70
IMPLEMENTACIÓN	70
<i>Introducción</i>	70
4.1 – <i>Modelo de Implementación.</i>	70
4.2 – <i>Modelo de Despliegue.</i>	70
4.2.1 – Diagrama de despliegue	71
4.3 – <i>Diagrama de componentes</i>	71
4.3.1 – Diagrama de componentes	72
4.3.2 – Diagrama de componentes del CU Autenticar Usuario	72
4.3.3 – Diagrama de componentes del CU Gestionar Trabajador	73
4.3.4 – Diagrama de componentes del CU Buscar Trabajador	73
4.3.5 – Diagrama de componentes del CU Listar Trabajador	74
4.3.6 – Diagrama de componentes del CU Generar Reportes (Control de Créditos)	74
<i>Conclusiones</i>	75
CONCLUSIONES	76
RECOMENDACIONES	77
REFERENCIAS BIBLIOGRÁFICAS	78
BIBLIOGRAFÍA	81
ANEXOS	82
1 – <i>Descripciones de los Casos de uso del Sistema</i>	82
1.1 – Descripción del Caso de uso del Sistema Gestionar Usuario.	82
1.2 – Descripción del Caso de uso del Sistema Buscar Trabajador por Actividad	83
1.3 – Descripción del Caso de uso del Sistema Listar Actividad	84
1.4 – Descripción del Caso de uso del Sistema Buscar Problema	85
1.5 – Descripción del Caso de uso del Sistema Gestionar Plan de Trabajo	86
2 – <i>Diagramas de Clases de Análisis</i>	88
2.1 – Diagrama de clases de análisis de Buscar Necesidad	88
2.2 – Diagrama de clases de análisis de Listar Plan de Trabajo	88
2.3 – Diagrama de clases de análisis de Gestionar Usuario	89
2.4 – Diagrama de clases de análisis de Listar Actividad	89
3 – <i>Diagramas de Clases del Diseño</i>	90
3.1 – Diagrama de Clases del Diseño de Buscar Actividad por Trabajador	90
3.2 – Diagrama de Clases del Diseño de Gestionar INA	90
3.3 – Diagrama de Clases del Diseño de Gestionar Plan de Trabajo	90
3.4 – Diagrama de Clases del Diseño de Listar Plan de Trabajo	92
3.5 – Diagrama de Clases del Diseño de Reporte Desarrollo Individual	92
4 – <i>Diagramas de Componentes</i>	92
4.1 – Diagrama de componentes de Buscar Necesidad	93
4.2 – Diagrama de componentes de Gestionar Plan de Trabajo	93
4.3 – Diagrama de componentes de Listar Problema	93
4.4 – Diagrama de componentes de Reporte Desarrollo Individual	93
GLOSARIO DE TÉRMINOS	95



Introducción

En la actualidad el ser humano vive en una sociedad envuelta por un rápido e intenso desarrollo, el cual es evidente en varias esferas, pero sin lugar a dudas uno de los campos de mayor avance es la Industria Informática. En la denominada Sociedad de la Información, es cada vez más difícil actuar, pensar o interrelacionarse con las demás personas sin que exista un vínculo directo con los ordenadores, ese artefacto que es en el mundo actual, un compañero inseparable de muchos seres humanos. Con el auge de las Nuevas Tecnologías de la Información y las Comunicaciones (NTICs) se han creado Sistemas y formas de comunicación cada vez más flexibles y eficaces.

La República de Cuba, como el resto de los países del mundo, se ha involucrado gradualmente en esta Revolución Informática. Actualmente es cada vez mayor el número de Empresas o Instituciones Cubanas que se han visto forzadas a informatizar todos o parte de sus procesos, ya sea para poder relacionarse con otros organismos o debido a la imposibilidad de manejar el cúmulo de información que hoy tramitan. Por todo esto es que en el país se realizan innumerables esfuerzos para llevar a cabo la informatización de la sociedad cubana. Un ejemplo de estos esfuerzos realizados fue la creación de la Universidad de las Ciencias Informáticas (UCI), para fomentar el desarrollo de la Industria del Software Cubana.

En la UCI se encuentra el Policlínico - Hospital Ernesto Che Guevara que brinda servicio al personal de la Universidad, fundamentalmente a los estudiantes, profesores y trabajadores internos.

Situación Problemática

La Vice-dirección de Docencia e Investigaciones (VDDI) del Policlínico - Hospital Ernesto Che Guevara, es un ejemplo de necesidad de informatización, la misma tiene, entre sus múltiples funciones, el control de toda la actividad investigativa y docente que desarrollan los trabajadores del Centro Médico. El Policlínico - Hospital Ernesto Che Guevara posee una estructura similar a otras unidades de salud, no obstante al igual que en otros policlínicos no se dispone de alguna solución informática que permita controlar la actividad docente – investigativa de sus trabajadores. Como parte del trabajo en la VDDI se registran una serie de documentos, tales como: Control de Créditos Académicos, Control de la Cantidad de Estudiantes, Investigación de

Enseñanza y Aprendizaje, Plan de Desarrollo Individual, entre otros; los cuales en la actualidad, aún cuando son almacenados en formato digital, son muy difícil de actualizar, por ejemplo: en Postgrado, cada vez que un maestrante termina una asignatura el propio Vice-director de Docencia e Investigaciones es el que tiene que solicitar los datos referentes a los créditos que otorga la asignatura para efectuar la correspondiente actualización; además, en Pregrado es necesario llevar también un control, algunos de los datos que se necesitan controlar en esta área son: cantidad de médicos y enfermeras que están cursando determinada asignatura, en qué año se encuentra matriculado cada uno y en cual Sede reciben las asignaturas, entre otras informaciones. Para que en la VDDI sea posible actualizar toda la información existente, los Jefes de Departamentos (Vice-directores de Enfermería, Asistencia Médica y Estomatología) deben entregar dicha información personalmente al Vice-director de Docencia e Investigaciones, haciendo más lento y engorroso el proceso, pues ocurren grandes demoras en el mismo. Además, debido a que la cantidad de trabajadores del Policlínico - Hospital es muy elevada se dificulta la búsqueda de algún dato específico de un trabajador determinado, por ejemplo: un médico que desee buscar el crédito que tiene acumulado hasta ese momento tiene que, buscar al Vice-director de Docencia e Investigaciones y entre ambos dedicarle un tiempo considerable a esta tarea.

Dentro de las funciones actuales de la VDDI se encuentran:

1. Planificar, organizar y controlar el proceso de capacitación y superación de los trabajadores del Policlínico - Hospital.
2. Planificar, organizar y controlar el proceso de capacitación de los sectores a nivel del área de salud.
3. Planificar, organizar y controlar el desarrollo del programa de pregrado en medicina, estomatología, tecnología y enfermería según corresponda:
 - Control de la calidad tanto del trabajo educativo como docente de los profesores.
4. Planificar, organizar y controlar el desarrollo del programa de postgrado en medicina, estomatología, tecnología y enfermería según corresponda:
 - Garantizar que todos los aspirantes a iniciar residencia de MGI, EGI y Licenciatura en Enfermería conozcan en tiempo la fecha de inicio de la matrícula.

- Consignar en el expediente académico los méritos y deméritos de que ha sido acreedor el residente.
 - Controlar las actividades asistenciales, docentes, administrativas e investigativas del residente.
 - Planificar las actividades de educación continuada de todos los trabajadores del policlínico, según la identificación de necesidades.
 - Planificar, organizar y controlar la evaluación de competencia y desempeño de los trabajadores del policlínico y departamentos.
5. Actualizar y controlar el Plan de desarrollo individual de los profesores.
6. Planificar, organizar y coordinar la actividad científica del centro.

Estas tareas y otras más específicas que se llevan a cabo en el Departamento, son desarrolladas manualmente y con un mínimo de empleo de las tecnologías, lo cual trae como consecuencia el aumento de la complejidad, tiempo de ejecución de las acciones y seguridad de las mismas.

El **problema científico** que se plantea es el siguiente: ¿Cómo proveer a la Vice-dirección de Docencia e Investigaciones del Policlínico - Hospital Ernesto Che Guevara de un Sistema que facilite y agilice todos los procesos que se llevan a cabo en la misma?

El **objeto de estudio** de la investigación es el Proceso de Gestión de Información en la Vice-dirección de Docencia e Investigaciones de un Policlínico - Hospital y el **campo de acción** es el Proceso de Gestión de la Información generada por las actividades que se realizan en la Vice-dirección de Docencia e Investigaciones del Policlínico - Hospital Ernesto Che Guevara.

Para la investigación se plantea la siguiente **Idea a Defender**: si se desarrolla un Sistema como apoyo informático para la realización de las tareas asignadas en la Vice-dirección de Docencia e Investigaciones del Policlínico - Hospital Ernesto Che Guevara, entonces se hará más viable y eficaz todo el proceso que se desarrolla actualmente.

El **objetivo general** es lograr una mayor eficacia, rapidez y seguridad en la Gestión de la Información de la Vice-dirección de Docencia e Investigaciones del Policlínico - Hospital Ernesto Che Guevara, a través del desarrollo de un Sistema de Gestión que haga más accesible y manipulable dicha información. Como **objetivos específicos** se encuentran:



- Determinar la tecnología a utilizar para la creación del Sistema de Gestión para la Vice-dirección de Docencia e Investigaciones del Policlínico - Hospital Ernesto Che Guevara.
- Crear un diseño arquitectónico sólido que responda a las necesidades del cliente.
- Implementar un Sistema robusto y eficiente.

Para dar cumplimiento a los objetivos que se han planteado se proponen las siguientes **tareas**:

1. Realizar un profundo análisis del funcionamiento de la VDDI.
2. Realizar un levantamiento de los requerimientos que deben tomarse a consideración en la implementación del Sistema, teniendo en cuenta el funcionamiento de la VDDI y los procesos a informatizar.
3. Realizar un estudio de las tendencias actuales respecto al desarrollo de Sistemas Gestores de Información a nivel mundial.
4. Crear un diseño de Interfaz de Usuario acorde a las necesidades del cliente.
5. Diseñar una Base de Datos funcional e implementar el Sistema utilizando las tecnologías seleccionadas.
6. Realizar las pruebas de funcionalidad del Sistema.
7. Desplegar el Sistema y capacitar al usuario que lo utilizará.
8. Elaborar el Documento de Tesis.

CAPÍTULO 1

FUNDAMENTACIÓN TEÓRICA

Introducción

En el desarrollo del presente capítulo se realizará un estudio del estado del arte relacionado con el problema al cual se dará solución. Se analizan conceptos, tecnologías, metodologías y software que se utilizan actualmente a nivel mundial y que podrán ser utilizados en la resolución del problema, así como las tendencias existentes a la hora de la creación de Sistemas de Gestión de la Información.

Se enfatizará en las Metodologías de Desarrollo del Software más factibles para dirigir todo el proceso de creación del Sistema. También se realizará un estudio de diferentes tecnologías, como son: los servidores Web más utilizados, lenguajes de programación tanto de parte del cliente como del servidor, Sistemas gestores de base de datos y arquitecturas y patrones de diseños más idóneos.

Además se presentará una lista de todas las herramientas que se utilizarán para el desarrollo del Sistema de Gestión de la Información.

1.1 – Conceptos Esenciales.

1.1.1 – Internet

Internet es un conjunto de redes: redes de ordenadores y equipos físicamente unidos mediante cables, que conectan puntos de todo el mundo. Estos cables se presentan en muchas formas, desde cables de red local (varias máquinas conectadas en una oficina o campus) a cables telefónicos convencionales, digitales y canales de fibra óptica que forman las carreteras principales. ⁽¹⁾

Existen unos seis millones de ordenadores que utilizan Internet en todo el mundo y que utilizan varios formatos y Protocolos Internet: ⁽²⁾

- **Internet Protocol (IP):** protocolo que se utiliza para dirigir un paquete de datos desde su fuente a su destino a través de Internet.
- **Transport Control Protocol (TCP):** protocolo de control de transmisión, que se utiliza para administrar accesos.

- **User Datagram Protocol (UDP):** protocolo del datagrama del usuario, que permite enviar un mensaje desde un ordenador a una aplicación que se ejecuta en otro ordenador.

1.1.2 – Web o World Wide Web

Es básicamente un medio de comunicación de texto, gráficos y otros objetos multimedia a través de Internet, es decir, la Web es un Sistema de hipertexto que utiliza Internet como su mecanismo de transporte o, desde otro punto de vista, una forma gráfica de explorar Internet.

Es importante saber que Web o WWW no son sinónimo de Internet, la Web es un subconjunto de Internet que consiste en páginas a las que se puede acceder usando un navegador. Internet es la red de redes donde reside toda la información. Tanto el correo electrónico, como FTPs y juegos son parte de Internet, pero no de la Web.

La Web es, en la actualidad, un medio muy popular de publicar información en Internet, y con el desarrollo del protocolo de transferencia segura (Secured Server Protocol, HTTPs), la Web es ahora un medio de comercio electrónico donde los consumidores pueden escoger sus productos on-line y realizar sus compras utilizando la información de sus tarjetas bancarias de forma segura. ⁽³⁾

1.1.3 – ¿Qué es una página Web?

Una página de Internet o página Web es un documento electrónico, que contiene información específica de un tema en particular y que es almacenado en algún Sistema de cómputo que se encuentre conectado a Internet, de tal forma que este documento pueda ser consultado por cualquier persona que se conecte a esta red mundial. Esta información generalmente es presentada en formato HTML, y puede proveer de vínculos hacia otras páginas mediante hipervínculos. ⁽⁴⁾

1.1.4 – Sitio Web

Es un conjunto de archivos electrónicos y páginas Web referentes a un tema en particular, que incluye una página inicial de bienvenida, generalmente denominada *Homepage*, con un nombre de dominio y dirección en Internet específicos. El Sitio Web no necesariamente debe localizarse en el Sistema de cómputo de su Negocio.

Los documentos que integran el Sitio Web pueden ubicarse en un equipo en otra localidad, inclusive en otro país. El único requisito es que el equipo en el que residan los documentos esté conectado a la red mundial de Internet. ⁽⁴⁾

1.1.5 – ¿Qué es un Software de Gestión?

Las Aplicaciones o Software de Gestión son aquellas diseñadas para sustituir uno o varios procedimientos, tanto comerciales como administrativos, que habitualmente realiza una persona en una empresa o institución de forma presencial, por un software, que permita realizar al cliente los mismos procedimientos de forma no presencial o disminuir el esfuerzo empleado para los mismos. El proceso de información comercial constituye la mayor de las áreas de aplicación del Software de Gestión. Los Sistemas discretos, han evolucionado hacia el Software de Gestión, que accede a una o más bases de datos que contienen información comercial. Las aplicaciones en esta área reestructuran los datos existentes, para facilitar las operaciones comerciales o gestionar la toma de decisiones. Además de las tareas convencionales de procesamiento de datos, las aplicaciones de Software de Gestión también realizan cálculo interactivo. ⁽⁵⁾

1.1.6 – Sistemas de Gestión de la Información.

La Gestión de la Información es el proceso de analizar y utilizar la información que se ha recabado y registrado, para permitir a los administradores, de todos los niveles, tomar decisiones documentadas. La Gestión de la Información implica:

- Determinar la información que se precisa.
- Recoger y analizar la información.
- Registrarla y recuperarla cuando sea necesaria.
- Utilizarla y divulgarla.

Con el surgimiento de aplicaciones capaces de gestionar grandes volúmenes de información, surgen los Sistemas de Gestión de la Información. Éstos, definidos como conjuntos de funciones o componentes interrelacionados que forman un todo, obtienen, procesan, almacenan y distribuyen la información, manipulando los datos y consiguiendo, para una organización o empresa, la búsqueda de mejores vías para la dirección y control correspondiente de sus procesos, apoyando la toma de las decisiones en el desempeño de las funciones, de acuerdo a sus estrategias propias. ⁽⁶⁾

1.2 – Tendencias y Tecnologías actuales.

1.2.1 – Arquitecturas y Patrones de Diseño

Un patrón de diseño es una solución a un problema de diseño no trivial, que es efectiva y reusable. Los patrones son soluciones de sentido común, que deberían formar parte del conocimiento de un diseñador experto. Además, facilitan la comunicación entre diseñadores, pues establecen un marco de referencia. Por otro lado, los patrones de diseño, facilitan el aprendizaje al programador inexperto, pudiendo establecer parejas problema-solución.

En la programación orientada a objetos (POO), resulta complicado descomponer el Sistema en objetos (encapsulación, granularidad, dependencias, flexibilidad y reusabilidad), los patrones de diseño permiten identificar a los objetos apropiados de una manera mucho más sencilla. También posibilitan determinar la granularidad de los objetos.

Además, los patrones de diseño, ayudan a especificar las interfaces, identificando los elementos claves en las interfaces y las relaciones existentes entre distintas interfaces. De igual modo permite más fácilmente la especificación de la implementación. También, de forma casi automática, ayudan a reutilizar código, facilitando la decisión entre "herencia o composición". [\(7\)](#)

Estilo de Llamada y Retorno

Este estilo es interesante en los Sistemas que se centran en la interacción de un usuario con el propio Sistema. Se puede dividir la arquitectura en dos partes, la primera representa la interfaz del usuario con el que éste realiza la llamada al Sistema, la segunda contiene la lógica de Negocio que se realiza tras la correspondiente llamada del usuario. Esta familia de estilos enfatiza la modificabilidad, la escalabilidad, la reusabilidad y la usabilidad del Sistema. Al diseñar una arquitectura de este tipo, es importante saber qué datos son los que servirán para interactuar con el usuario, y cuáles servirán sólo como lógica de aplicación. En esta familia de arquitecturas se destacan los patrones modelo-vista-controlador y arquitectura en tres capas.

1.2.1.1 – Patrón Modelo – Vista – Controlador (MVC)

Definición del Patrón

- Modelo (Model): Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada. El modelo debe de preservar la integridad de los datos.
- Vista (View): Muestra la información al usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador. Interactúa con la interfaz de usuario.
- Controlador (Controller): Reciben las entradas, usualmente como eventos, e interpreta las operaciones del usuario; codificando los movimientos, pulsación de botones del ratón, pulsaciones de teclas, etc. Los eventos son traducidos a solicitudes de servicio (service requests) para el modelo o la vista. Es el que debe de controlar los eventos.

Responsabilidades de cada parte

Definidos los conceptos de Modelo, Vista y Controlador, a continuación se definirán las responsabilidades de cada uno de ellos.

- El modelo deberá:
 - Acceder a los datos persistentes, a la capa de almacenamiento de datos. Algo ideal es que éste sea independiente de la Base de datos utilizada.
 - Llevar las estadísticas del Sistema.
 - Definir las reglas del Negocio.
 - Notificar los cambios que se hayan realizado.
- El controlador deberá:
 - Recibir los eventos de entrada.
 - Realizar la acción correspondiente al evento que se ha disparado.
- Las vistas deberán:
 - Conseguir los datos que aporta el modelo y mostrárselos al usuario.
 - Saber cual es su controlador asociado, el cual puede pedirle algunos servicios típicos como el de actualizar.

Ventajas e Inconvenientes del MVC

Ventajas: ⁽⁸⁾

- Se consiguen múltiples vistas del modelo.
- Todas las vistas están sincronizadas.
- No acoplamiento, y facilidad de evolución, para cambiar las vistas y los controladores.
- La aplicación puede soportar un tipo de interfaz para cada usuario (rol).

Desventajas: ⁽⁸⁾

- La complejidad va creciendo más rápido que el tamaño de la aplicación.
- Problemas al conectar las distintas capas.
- Acceso ineficiente, pues es necesario varias llamadas al modelo para actualizar los datos.
- La vista y el controlador son específicos para una plataforma.

1.2.1.2 – Arquitectura en tres capas.

Cuando se desarrolla una aplicación cliente - servidor, la tendencia habitual es mezclar lógica con presentación. Es normal que en los formularios se implemente el acceso a la lógica de Negocio de la aplicación y la navegación a nuevos formularios. Pero esto trae consigo algunos inconvenientes, para solucionarlos se utiliza la arquitectura en tres capas.

Definición de Capas

- **Capa de Presentación:** esta es la que el usuario puede ver en su ordenador, es donde se gestionan los datos que se van a mostrar. Se intenta que en esta capa haya el mínimo de procesamiento. Esta capa se comunicará solamente con la capa de Negocio.
- **Capa de Negocio:** en esta capa está la lógica. Se recibe las peticiones del usuario, y tras ejecutar una acción, se le envía las respuestas del proceso. Esta capa se comunica como se ha dicho con la de presentación, la cual le envía peticiones, y ésta le responde con los resultados. Y también se comunica con la capa de datos, para pedirle datos.
- **Capa de Acceso a Datos:** es donde se accede a los datos. Se hace referencia a uno o más gestores de Bases de Datos (BD) que realizan el almacenamiento,

modificación y consulta de los datos. Esta capa recibe peticiones desde la capa de Negocio.

Ventajas e Inconvenientes del Modelo en tres capas

Ventajas:

- Facilita que se pueda descomponer la aplicación en varios niveles de abstracción.
- Permite la evolución del Sistema, ya que los cambios sólo deben de afectar a la capa donde se encuentre la modificación.
- Si la interfaz accede a la misma función, no se repetirá código. Lo que conlleva la ventaja de una mayor facilidad de mantenimiento de la aplicación, entre otros.
- Si se añade un nuevo formulario, no ocasionará grandes problemas, ya que los formularios ya no dependen unos de otros.
- El formulario ya no accede de forma independiente a los datos, ya que no se accede a través de ellos al modificar los datos.

Desventajas:

- No todo Sistema podrá ser estructura en capas.
- Aún pudiendo ser estructurado en capas, la separación entre una y otra no es trivial. Ya no sólo porque para un desarrollador no lo es, sino también porque muchos lenguajes y/o frameworks no están preparados para ello. [\(8\)](#)

1.2.1.3 – ¿Por qué utilizar la Arquitectura en tres capas?

Para el Sistema que se desea desarrollar, las desventajas que presenta la Arquitectura en tres capas son irrelevantes, pues es un Sistema de fácil estructuración en capas y será desarrollado con tecnología que facilite el trabajo con esta arquitectura; mientras que el MVC, presenta grandes deficiencias como son: la complejidad va creciendo más rápido que el tamaño de la aplicación, problemas al conectar las distintas capas, acceso ineficiente, pues es necesario varias llamadas al modelo para actualizar los datos, la vista y el controlador son específicos para una plataforma.

1.3 – Metodologías de Desarrollo de Software

El Proceso de Desarrollo de Software siempre incluye ciertos riesgos y dificultades, sin importar el tamaño del software a realizar. Sin el apoyo de una metodología, nunca se llegaría a satisfacer las necesidades de los clientes para los cuales se trabaja.

Las metodologías son el conjunto de procedimientos, técnicas, herramientas y un soporte documental, que ayuda a los desarrolladores a realizar un nuevo software. [\(9\)](#)

A partir de esta definición, se desprenden una serie de términos, que son necesarios para poder completar la definición dada: [\(9\)](#)

Tarea: Actividades elementales en que se dividen los procesos.

Procedimiento: Definición de la forma de ejecutar la tarea.

Técnica: Herramienta utilizada para aplicar un procedimiento.

Herramienta: Para realizar una técnica, se puede apoyar en las herramientas software que automatizan su aplicación.

Producto: Resultado de cada etapa.

Características deseables de una metodología:

1. Existencia de reglas predefinidas.
2. Cobertura total del ciclo de desarrollo.
3. Verificaciones intermedias.
4. Planificación y control.
5. Comunicación efectiva.
6. Utilización sobre un abanico amplio de proyectos.
7. Fácil formación.
8. Herramientas CASE.
9. Actividades que mejoren el proceso de desarrollo.
10. Soporte al mantenimiento.
11. Soporte de la reutilización de software.

Actualmente existen diversas Metodologías de Desarrollo de Software, entre las más utilizadas se encuentran: metodologías tradicionales o robustas como Rational Unified Process (RUP), Microsoft Solutions Framework (MSF) y Métrica surgida en el año 1989; o metodologías ágiles como Extreme Programming (XP), Scrum, Cristal Methods y Feature Driven Development. Cada una de las Metodologías antes mencionadas tienen sus particularidades y el escoger una de ellas debe ser un

proceso serio y riguroso para que el software se cree con la calidad necesaria. No obstante, las más aceptadas y utilizadas actualmente son RUP y XP. ⁽⁹⁾

1.3.1 – Extreme Programming (XP)

Las metodologías ágiles (como por ejemplo XP, SCRUM y Crystal) forman parte del movimiento de desarrollo ágil de software, que se basan en la adaptabilidad de cualquier cambio, como medio para aumentar las posibilidades de éxito de un proyecto.

De forma que una metodología ágil es la que tiene como principios que:

- Los individuos y sus interacciones, son más importantes que los procesos y las herramientas.
- El software que funciona, es más importante que la documentación exhaustiva.
- La colaboración con el cliente, en lugar de la negociación de contratos.
- La respuesta delante del cambio, en lugar de seguir un plan cerrado.

La programación extrema es una metodología de desarrollo ligera (o ágil), basada en una serie de valores y de prácticas de buenas maneras, que persigue el objetivo de aumentar la productividad a la hora de desarrollar programas.

Este modelo de programación se basa en una serie de metodologías de desarrollo de software, en la que se da prioridad a los trabajos que dan un resultado directo, y que reducen la burocracia que hay alrededor de la programación.

Una de las características principales de este método de programación, es que sus ingredientes son conocidos desde el principio de la informática. Los autores de XP han seleccionado aquellos que han considerado mejores, y han profundizado en sus relaciones y en como se refuerzan los unos con los otros. El resultado de esta selección ha sido esta metodología única y compacta. Por esto, aunque no está basada en principios nuevos, sí que el resultado es una nueva manera de ver el desarrollo de software.

El objetivo que se perseguía, en el momento de crear esta metodología, era la búsqueda de un método que hiciera que los desarrollos fueran más sencillos. ⁽¹⁰⁾

Valores de XP: ⁽¹¹⁾

- **Comunicación:** Crear software requiere de Sistemas comunicados.
- **Simplicidad:** Empezar con lo necesario y requerido y trabajar desde ahí.
- **Retroalimentación:** Del Sistema, del cliente, y del equipo.

- **Valentía:** Programa para hoy y no para mañana.
- **Respeto:** El equipo debe trabajar como uno, sin hacer decisiones repentinas.

Actividades: [\(11\)](#)

- **Codificación:** La parte mas importante de XP.
- **Pruebas:** Nunca se puede estar seguro de algo hasta haberlo probado.
- **Escuchar:** Escuchar los requisitos del cliente acerca del Sistema a crear.
- **Diseño:** Crear una estructura del diseño para evitar problemas.

Ventajas: [\(11\)](#)

- Programación organizada.
- Menor tasa de errores.
- Satisfacción del programador.
- El cliente tiene el control sobre las prioridades.
- Se hacen pruebas continuas durante el proyecto.
- La XP es mejor utilizada en la implementación de nuevas tecnologías donde los requerimientos cambian rápidamente.

Desventajas: [\(11\)](#)

- Es recomendable emplearlo solo en proyectos a corto plazo.
- Altas comisiones en caso de fallar.

1.3.2 – Proceso Unificado del Software

El Proceso Unificado del Software o RUP (Rational Unified Process), es una metodología tradicional o robusta y junto con el Lenguaje Unificado de Modelado (UML), del cual se hablará posteriormente, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de Sistemas orientados a objetos.

Algunas características de RUP son: [\(12\)](#)

- Forma disciplinada de asignar tareas y responsabilidades (quién, qué, cuándo y cómo).
- Pretende implementar las mejores prácticas en Ingeniería de Software.
- Desarrollo iterativo.
- Administración de requisitos.
- Uso de arquitectura basada en componentes.



- Control de cambios.
- Modelado visual del software.
- Verificación de la calidad del software.

El RUP es una solución de la empresa Rational Software Company. Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los Casos de uso. Incluye artefactos (que son los productos tangibles del proceso como, por ejemplo: el modelo de Casos de uso y el código fuente) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso). ⁽¹²⁾

Las características principales de RUP son:

- **Guiado/Manejado por Casos de uso:** La razón de ser de un Sistema software es servir a usuarios ya sean humanos u otros Sistemas; un Caso de uso es una facilidad que el software debe proveer a sus usuarios. Los Casos de uso reemplazan la antigua especificación funcional tradicional, y constituyen la guía fundamental establecida para las actividades a realizar durante todo el proceso de desarrollo, incluyendo el diseño, la implementación y las pruebas del Sistema.
- **Centrado en la arquitectura:** La arquitectura involucra los elementos más significativos del Sistema y está influenciada, entre otros, por plataformas software, Sistemas Operativos, gestores de bases de datos, protocolos, consideraciones de desarrollo como Sistemas heredados y requerimientos no funcionales. Es como una radiografía del Sistema que se desarrolla, lo suficientemente completa como para que todos los implicados en el desarrollo, tengan una idea clara de qué es lo que están construyendo, pero lo suficientemente simple como para que si se quita algo, una parte importante del Sistema quede sin especificar. Se representa mediante varias vistas que se centran en aspectos concretos del Sistema, abstrayéndose de lo demás. Todas las vistas juntas forman el llamado modelo 4+1 de la arquitectura, recibe este nombre porque lo forman las vistas lógica, de implementación, proceso y despliegue, más la de Casos de uso que es la que da cohesión a todas.
- **Iterativo e Incremental:** Para hacer más manejable un proyecto se recomienda dividirlo en ciclos. Para cada ciclo se establecen fases de referencia, cada una de las cuales debe ser considerada como un mini-proyecto, cuyo núcleo fundamental está constituido por una o más iteraciones

de las actividades principales básicas de cualquier proceso de desarrollo. En concreto RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable, según el proyecto, y en las que se hace un mayor o menor énfasis en las distintas actividades. [\(13\)](#)

RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al final de cada ciclo, cada ciclo se divide en fases, que finalizan con un hito, donde se debe tomar una decisión importante:

- Inicio: se hace un plan de fases, se identifican los principales Casos de uso y se identifican los riesgos.
- Elaboración: se hace un plan de proyecto, se completan los Casos de uso y se eliminan los riesgos. Durante esta fase y la de inicio se capturan el 80% de los requisitos.
- Construcción: se concentra en la elaboración de un producto totalmente operativo y eficiente, y el manual de usuario. La implementación es el flujo de trabajo fundamental en esta fase.
- Transición: se implementa el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requerimientos a ser analizados. [\(12\)](#)

En RUP se definen nueve flujos de trabajo distintos, separados en dos grupos. [\(13\)](#)

Los flujos de trabajo de ingeniería son los siguientes:

- Modelado del Negocio.
- Requisitos.
- Análisis y diseño.
- Implementación.
- Test.
- Despliegue.

Los flujos de trabajo de apoyo son:

- Administración del proyecto.
- Configuración y control de cambios.
- Entorno.

1.3.3 – ¿Por qué el uso de RUP?

RUP se basa en la documentación exhaustiva del software y prioriza bastante el análisis y diseño, lo cual permite obtener un producto de mayor calidad y que esté acorde a las especificaciones del cliente, aspectos en los cuales aventaja al XP. Además, RUP utiliza y se apoya en el Lenguaje Unificado de Modelado (UML), lenguaje el cual es muy potente y su uso facilita una mejor perspectiva de lo que se desea lograr.

1.4 – Lenguaje de Modelado

1.4.1 – UML

UML (Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un Sistema software orientado a objetos. Se ha convertido en el estándar de facto de la industria, debido a que ha sido impulsado por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh. Estos autores fueron contratados por la empresa Rational, para crear una notación unificada en la que basar la construcción de sus herramientas CASE. En el proceso de creación de UML han participado, no obstante, otras empresas de gran peso en la industria, como Microsoft, Hewlett-Packard, Oracle o IBM, así como grupos de analistas y desarrolladores.

Un modelo representa a un Sistema software desde una perspectiva específica. Cada modelo permite fijarse en un aspecto distinto del Sistema.

Algunos modelos de UML son:

- Diagrama de Estructura Estática.
- Diagrama de Casos de uso.
- Diagrama de Secuencia.
- Diagrama de Colaboración.
- Diagrama de Estados.

UML usa procesos de otras metodologías, aprovechando la experiencia de sus creadores, eliminó los componentes que resultaban de poca utilidad práctica y añadió nuevos elementos. UML es un lenguaje más expresivo, claro y uniforme que los anteriores definidos para el diseño Orientado a Objetos, que no garantiza el éxito de los proyectos, pero sí mejora sustancialmente el desarrollo de los mismos, al permitir una nueva y fuerte integración entre las herramientas, los procesos y los dominios. ⁽¹⁴⁾

1.5 – Tecnologías y Lenguajes de Programación del lado del Cliente

Los lenguajes del lado del cliente son aquellos que pueden ser directamente tramitados por el navegador y no necesitan un tratamiento previo. [\(15\)](#)

1.5.1 – XHTML (eXtensible HyperText Markup Language)

El lenguaje XHTML (Lenguaje de Marca de Hipertexto) indica al navegador donde colocar cada texto, cada imagen o cada video y la forma que tendrán estos al ser colocados en la página. Este lenguaje es utilizado para presentar la información en la red de redes.

El lenguaje consta de etiquetas que tienen esta forma: <Nombre de Etiqueta> cada etiqueta tiene un significado, por ejemplo: significa que se escriba en negrita (bold) o <p> significa un párrafo, <a> es un enlace, entre otras. Casi todas las etiquetas tienen su correspondiente etiqueta de cierre, que indica que a partir de ese punto no debe de afectar la etiqueta, por ejemplo: se utiliza para indicar que se deje de escribir en negrita. En resumen XHTML consta de una serie de etiquetas que se utilizan para definir la forma o estilo que se quiere aplicar a un documento.

1.5.2 – Javascript

Javascript es un lenguaje de programación utilizado para crear pequeños programas, encargados de realizar acciones dentro del ámbito de una página Web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Su uso se basa, fundamentalmente, en la creación de efectos especiales en las páginas y la definición de interactividades con el usuario.

Las sentencias escritas en javascript se encapsulan entre las etiquetas <script> y </script>.

Javascript es el siguiente paso, después del XHTML, que puede dar un programador de la Web que decida mejorar sus páginas y la potencia de sus proyectos. Es un lenguaje de programación bastante sencillo y pensado para realizar Sistemas con rapidez. [\(15\)](#)

1.5.3 – CSS (Cascading Style Sheets)

CSS son las siglas de Cascading Style Sheets, en español Hojas de estilo en Cascada. CSS es una tecnología que permite crear páginas Web de una manera más exacta. Gracias a esta tecnología se tiene más control sobre los resultados finales de la página, permite ejecutar acciones que no se pueden hacer utilizando solamente XHTML, como incluir márgenes, tipos de letra, fondos, colores. Incluso se puede definir estilos propios en un archivo externo a las páginas; si en algún momento se desea cambiar alguno de ellos, automáticamente se actualizarán todas las páginas vinculadas al sitio. ⁽¹⁵⁾

1.5.4 – XML (eXtensible Markup Language ó Lenguaje extensible de marcas)

Es un conjunto de reglas, que sirven para definir etiquetas semánticas, para organizar un documento. Además el XML es un metalenguaje, que te permite diseñar tu propio lenguaje de etiquetas. ⁽¹⁶⁾

Como lenguaje de anotación, las sentencias en XML consisten en una serie de etiquetas (llamadas *elementos*) con una serie de modificadores (llamados *atributos*). Las etiquetas pueden estar anidadas unas dentro de otras, pero toda etiqueta que se abra se tiene que cerrar, y siempre en el mismo orden.

Todos los documentos XML deben estar bien formados, y este es el requisito mínimo que deben cumplir los documentos. Eso significa que se debe cumplir lo siguiente: ⁽¹⁷⁾

- Todas las etiquetas deben estar equilibradas: esto es, todos los elementos que contengan datos de tipo carácter deben tener etiquetas de principio y fin.
- Todos los valores de los atributos deben ir entrecomillados (el carácter comilla simple [el apóstrofe] puede utilizarse si el valor contiene caracteres comillas dobles, y viceversa): si necesitas ambos, utiliza `'` y `"`.
- Cualquier elemento vacío (aquellos que no tienen etiqueta final como ``, `<hr>`, y `
` y otros de HTML) deben terminar con `'/>` o se deben hacer no vacíos añadiéndoles una etiqueta de fin.
- No debe haber etiquetas aisladas (`<` ó `&`) en el texto (por ejemplo: debe darse como `<` y `&`), y la secuencia `]]>` debe darse como `]]>`.
- Los elementos deben anidar dentro de sí sus propiedades (no se deben sobreponer etiquetas).

- Los ficheros bien formados sin DTD pueden utilizar atributos en sus elementos, pero éstos deben ser todos del tipo CDATA, por defecto. El tipo CDATA (*character DATA*) son caracteres.
- Los nombres de las etiquetas pueden ser alfanuméricos, comenzando con una letra.

1.5.5 – AJAX (Javascript y XML Asíncronicos)

Ajax no es una tecnología. Es realmente muchas tecnologías, cada una floreciendo por su propio mérito, uniéndose en poderosas nuevas formas. AJAX incorpora:

- Presentación basada en estándares usando XHTML y CSS.
- Exhibición e interacción dinámicas usando el Modelo de Objetos del Documento (Document Object Model).
- Intercambio y manipulación de datos usando XML and XSLT.
- Recuperación de datos asíncronica usando XMLHttpRequest.
- JavaScript: une todas las partes.

Una aplicación AJAX elimina la naturaleza “arrancar-frenar-arrancar-frenar” de la interacción en la Web, introduciendo un intermediario, un motor AJAX, entre el usuario y el servidor. Parecería que sumar una capa a la aplicación la haría menos reactiva, pero la verdad es lo contrario.

En vez de cargar un pagina Web, al inicio de la sesión, el navegador carga al motor AJAX (escrito en JavaScript). Este motor es el responsable por renderizar la interfaz que el usuario ve y por comunicarse con el servidor en nombre del usuario. El motor AJAX permite que la interacción del usuario con la aplicación suceda asíncronicamente (independientemente de la comunicación con el servidor). ⁽¹⁸⁾

1.6 – Tecnologías y Lenguajes de Programación del Lado del Servidor

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor Web, justo antes de que se envíe la página, a través de Internet, al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código XHTML resultante de la ejecución de la página del lado del servidor.

1.6.1 – PHP (Personal Home Page)

PHP es un lenguaje de script, interpretado en el lado del servidor, utilizado para la generación de páginas Web dinámicas, similar al ASP de Microsoft o el JSP de Sun, embebido en páginas XHTML y ejecutado en el servidor.

A diferencia de Java o JavaScript, que se ejecutan en el navegador, PHP se ejecuta en el servidor, por eso permite acceder a los recursos que tenga el servidor, como por ejemplo podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado es enviado al navegador.

El modo de operación del PHP es el siguiente:

- El navegador realiza una petición al servidor (se escribe la URL).
- Después el servidor ejecuta el código PHP solicitado y retorna el código XHTML generado al navegador.
- Por último el navegador muestra la respuesta del servidor.

Este tipo de iteración permite algunas operaciones complejas, como conexiones a bases de datos o ejecución de complejos programas. PHP, además de soportar un número masivo de bases de datos, incluyendo INFORMIX, ORACLE, Sybase, Solid, MySQL y PostgreSQL, también ofrece una gran variedad de funciones, que permiten desarrollar múltiples acciones que van desde enviar un e-mail o un archivo, crear una imagen en tiempo de ejecución, interactuar con diversos protocolos de comunicación, interactuar con documentos XML, autenticación, creación dinámica de documentos PDF (documentos de Acrobat Reader), entre muchos elementos.

Las principales características de PHP son: su rapidez, su facilidad de aprendizaje, su soporte multiplataforma tanto de diversos Sistemas Operativos, como Servidores Web y de bases de datos; y el hecho de que se distribuye de forma gratuita bajo una licencia abierta. ⁽¹⁹⁾

1.6.2 – ASP.NET (Active Server Pages)

ASP (Active Server Pages) es la tecnología, desarrollada por Microsoft, para la creación de páginas dinámicas del servidor. ASP se escribe en la misma página Web, utilizando el lenguaje Visual Basic Script (VBScript) o Jscript (Javascript de Microsoft).

El principio de la tecnología ASP es el VBScript, pero existe otra diversidad de lenguajes de programación que pueden ser utilizados como lo es Perl, JScript, entre otros.

Microsoft introdujo esta tecnología, llamada ASP, en diciembre de 1996, por lo que no es nada nueva. Es parte del Internet Information Server (IIS) desde la versión 3.0 y es una tecnología de páginas activas, que permite el uso de diferentes scripts y componentes en conjunto con el tradicional HTML, para mostrar páginas generadas dinámicamente, Microsoft las define con los siguientes términos: *“Las Active Server Pages son un ambiente de aplicación abierto y gratuito en el que se puede combinar código HTML, scripts y componentes ActiveX del servidor para crear soluciones dinámicas y poderosas para la Web”*. [\(20\)](#)

1.6.3 – ¿Por qué usar PHP?

Al ser un lenguaje libre, dispone de una gran cantidad de características que lo convierten en la herramienta ideal para la creación de páginas Web dinámicas:

- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF o analizar código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- Con PHP se puede hacer cualquier cosa que se desee realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas. [\(19\)](#)

1.7 – Servidor Web

Básicamente, un servidor Web sirve contenido estático a un navegador, carga un archivo y lo sirve, a través de la red, al navegador de un usuario. Este intercambio es

mediado por el navegador y el servidor que hablan el uno con el otro mediante HTTP. Se pueden utilizar varias tecnologías en el servidor para aumentar su potencia, más allá de su capacidad de entregar páginas HTML; éstas incluyen scripts CGI y seguridad SSL. [\(21\)](#)

1.7.1 – Servidor Apache

El servidor Apache es uno de los servidores Web más utilizados mundialmente. Es el servidor Web hecho por excelencia, su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa.

Apache es una muestra, al igual que el Sistema Operativo Linux, de que, el trabajo voluntario y cooperativo dentro de Internet es capaz de producir aplicaciones de calidad profesional difíciles de igualar.

La licencia Apache es una descendiente de la licencias BSD, no es GPL. Esta licencia te permite hacer lo que quieras con el código fuente (incluso productos propietarios) siempre que les reconozcas su trabajo. [\(22\)](#)

Ventajas del Servidor Apache:

- Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- Apache es una tecnología gratuita, de código fuente abierto. Esto le brinda una transparencia a este software de manera que, si se quiere ver qué es lo que se instala como servidor, se puede conocer, sin ningún secreto, sin ninguna puerta trasera.
- Apache es un servidor altamente configurable, de diseño modular. Es muy sencillo ampliar las capacidades del servidor Web Apache. Actualmente existen muchos módulos para Apache que son adaptables a éste y se encuentran disponibles en la red. Otra cosa importante es que, cualquiera que posea una experiencia en la programación de C o Perl, puede escribir un módulo para realizar una función determinada.
- Apache permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script, cuando ocurra un error en concreto.

- Tiene una alta configurabilidad en la creación y gestión de logs. Apache permite la creación de ficheros de log a medida del administrador, de este modo se puede tener un mayor control sobre lo que sucede en el servidor. [\(22\)](#)

1.7.2 – IIS (Internet Information Server)

Internet Information Server es un potente servidor Web que ofrece una infraestructura de gran fiabilidad, capacidad de manejo y escalabilidad para aplicaciones Web. IIS hace posible que las organizaciones aumenten la disponibilidad de sus sitios y aplicaciones Web y a la vez, reducir sus costes administrativos.

Además, constituye una serie de servicios para los ordenadores que funcionan con Windows. Originalmente era parte del Paquete de Opciones para Windows NT. Luego fue integrado en otros Sistemas operativos de Microsoft destinados a ofrecer servicios, como Windows 2000 o Windows Server 2003. Windows XP Profesional incluye una versión limitada de IIS.

Este servicio convierte a un ordenador en un servidor de Internet o Intranet, es decir, que en las computadoras que tienen este servicio instalado se pueden publicar páginas Web, tanto local como de forma remota (servidor Web).

El servidor Web se basa en varios módulos, que le dan capacidad para procesar distintos tipos de páginas, por ejemplo: Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl. [\(23\)](#)

1.7.3 – ¿Por qué utilizar un Servidor Apache?

El Servidor Web Apache es multiplataforma lo que lo convierte en un servidor prácticamente universal, a diferencia de su homólogo el IIS que solo es desarrollado para Windows. Apache es una tecnología gratuita de código fuente abierto, esto le da una transparencia a este software de manera que si se desea ver qué es lo que se está instalando como servidor, se puede conocer, sin ningún secreto, sin ninguna puerta trasera; y por último es altamente configurable de diseño modular por lo que es muy sencillo ampliar sus capacidades.

1.8 – Sistemas gestores de Bases de datos

Un Sistema gestor de Bases de datos se define como: el conjunto de programas que administran y gestionan la información contenida en una Base de datos. Este ayuda a realizar las siguientes acciones:⁽²⁴⁾

- Definición de los datos.
- Mantenimiento de la integridad de los datos, dentro de la Base de datos.
- Control de la seguridad y privacidad de los datos.
- Manipulación de los datos.

Un Sistema gestor de Bases de datos está compuesto de:

- El gestor de Bases de datos.
- Diccionario de datos.
- Administrador de las Bases de datos.
- Los lenguajes.

1.8.1 – MySQL

Es un Sistema de gestión de Bases de datos relacional, fue creada por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, así como también de la marca.

MySQL es un software de código abierto, licenciado bajo la GPL de la GNU, aunque MySQL AB distribuye una versión comercial, en lo único que se diferencia de la versión libre, es en el soporte técnico que se ofrece, y la posibilidad de integrar este gestor en un software propietario, ya que de otra manera, se vulneraría la licencia GPL.

El lenguaje de programación que utiliza MySQL es Structured Query Language (SQL), que fue desarrollado por IBM, en 1981, y desde entonces, es utilizado de forma generalizada en las Bases de datos relacionales.

Características Principales:

- El principal objetivo de MySQL es velocidad y robustez.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre Sistemas, puede trabajar en distintas plataformas y Sistemas Operativos.

- Cada Base de datos cuenta con 3 archivos: Uno de estructura, uno de datos y uno de índice y soporta hasta 32 índices por tabla.
- Aprovecha la potencia de Sistemas multiproceso, gracias a su implementación multihilo.
- Flexible Sistema de contraseñas y gestión de usuarios, con un muy buen nivel de seguridad en los datos.
- El servidor soporta mensajes de error en distintos lenguajes. [\(25\)](#)

1.8.2 – PostgreSQL

PostgreSQL es un Sistema de gestión de Bases de datos Objeto-Relacionales (ORDBMS), que ha sido desarrollado de varias formas desde 1977. Comenzó como un proyecto denominado Ingres, en la Universidad Berkeley, de California. Ingres fue más tarde desarrollado comercialmente por la Relational Technologies/Ingres Corporation.

PostgreSQL está ampliamente considerado como el Sistema de Bases de datos de código abierto más avanzado del mundo. Posee muchas características que tradicionalmente sólo se podían ver en productos comerciales de alto calibre.

Algunas de sus características son: [\(26\)](#)

DBMS Objeto-Relacional

PostgreSQL aproxima los datos a un modelo objeto-relacional, y es capaz de manejar complejas rutinas y reglas. Ejemplos de su avanzada funcionalidad son consultas SQL declarativas, control de concurrencia multi-versión, soporte multi-usuario, transacciones, optimización de consultas, herencia, y arreglos.

Altamente Extensible

PostgreSQL soporta operadores, funciones, métodos de acceso y tipos de datos definidos por el usuario.

Soporte SQL Comprensivo

PostgreSQL soporta la especificación SQL99 e incluye características avanzadas tales como las uniones (joins) SQL92.

Integridad Referencial

PostgreSQL soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la Base de datos.

Lenguajes Procedurales

PostgreSQL tiene soporte para lenguajes procedurales internos, incluyendo un lenguaje nativo denominado PL/pgSQL. Este lenguaje es comparable al lenguaje procedural de Oracle, PL/SQL. Otra ventaja de PostgreSQL es su habilidad para usar Perl, Python, o TCL como lenguaje procedural embebido.

Cliente/Servidor

PostgreSQL usa una arquitectura proceso-por-usuario cliente/servidor. Esta es similar al método del Apache 1.3.x para manejar procesos. Hay un proceso maestro que se ramifica para proporcionar conexiones adicionales para cada cliente que intente conectar a PostgreSQL.

1.8.3 – Oracle

Oracle es básicamente una herramienta cliente/servidor para la gestión de Bases de datos. Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio, hacen que sólo se vea en empresas muy grandes y multinacionales, por norma general. En el desarrollo de páginas Web pasa lo mismo: como es un Sistema muy caro, no está tan extendido como otras Bases de datos, por ejemplo, Access, MySQL, SQL Server y PostgreSQL.

Para desarrollar en Oracle se utiliza PL/SQL un lenguaje de quinta generación, bastante potente para tratar y gestionar la Base de datos, también por norma general se suele utilizar SQL al crear un formulario. [\(27\)](#)

1.8.4 – ¿Por qué usar MySQL?

Entre las ventajas que presenta MySQL, que lo hacen uno de los gestores más utilizados, se encuentran: velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento; bajo costo en requerimientos para la elaboración de Bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema; facilidad de configuración e instalación, soporta gran variedad de Sistemas Operativos; baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el Sistema en el que está; conectividad y seguridad. [\(25\)](#)

1.9 – Herramientas a utilizar.

1.9.1 – Rational Rose Enterprise Edition

Rational Rose Enterprise, es el producto más completo de la familia Rational Rose. Como el resto de los productos Rational Rose incluye soporte Unified Modeling Language (UML).

Rational Rose Enterprise es la mejor elección para el ambiente de modelado que soporte la generación de código a partir de modelos en Ada, ANSI C++, C++, CORBA, Java™/J2EE™, Visual C++ y Visual Basic. Como todos los demás productos **Rational Rose**, proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de alta calidad. [\(28\)](#)

Características adicionales incluidas: [\(28\)](#)

- Soporte para análisis de patrones basado en "Design Patterns: Elements of Reusable Object-Oriented Software".
- Característica de control por separado de componentes, modelo que permite una administración más granular y el uso de modelos.
- Soporte de ingeniería Forward y/o reversa para algunos de los conceptos más comunes de Java.
- La generación de código Ada, ANSI C ++, C++, CORBA, Java y Visual Basic, con capacidad de sincronización modelo-código configurables.
- Soporte Enterprise Java Beans™.
- Capacidad de análisis de calidad de código.
- El Add-In para modelado Web provee visualización, modelado y las herramientas para desarrollar aplicaciones de Web.
- Modelado UML para trabajar en diseños de Bases de datos, con capacidad de representar la integración de los datos y los requerimientos de aplicación a través de diseños lógicos y físicos.
- Capacidad de crear definiciones de tipo de documento XML para el uso en la aplicación.
- Integración con otras herramientas de desarrollo de Rational.

1.9.2 – EMS MySQL Manager

Es una herramienta de alto desempeño, para administración y desarrollo en Servidor de Base de datos MySQL. SQL Manager, para MySQL, trabaja con cualquier versión

de MySQL, desde la 3.23 hasta la 5.06 y soporta todas las características más recientes de MySQL incluyendo vistas, procedimientos almacenados y funciones, claves foráneas InnoDB, entre otras. Ofrece una gran cantidad de herramientas poderosas a usuarios experimentados, para satisfacer todas sus necesidades. SQL Manager para MySQL, tiene una nueva interface gráfica de usuario de avanzada, con un Sistema asistente bastante descriptivo, tan claro en su uso, que ni un principiante se confundirá.

Características principales: [\(29\)](#)

- Soporte completo para versiones MySQL desde la 3.23 hasta la 5.06.
- Nueva interfaz gráfica de usuario último modelo.
- Ágil navegación y administración de Base de datos.
- Administración sencilla de todos los objetos de MySQL.
- Herramientas de manipulación de datos avanzada.
- Acceso al servidor MySQL a través del protocolo HTTP.
- Potente Administración de seguridad.
- Excelentes Herramientas visuales y de texto para elaboración de consultas.
- Conexión vía reenvío de puerto local a través del túnel SSH.
- Impresionantes opciones de exportación e importación de datos.
- Diseñador Visual de Base de datos completamente rediseñado.
- Asistentes fáciles de usar para efectuar servicios MySQL.

1.9.3 – Dreamweaver

Dreamweaver es un programa fácil de utilizar que permite crear páginas Web profesionales.

Las funciones de edición visual de Dreamweaver 8, permiten agregar rápidamente diseño y funcionalidad a las páginas, sin la necesidad de programar manualmente el código XHTML.

Se puede crear tablas, editar marcos, trabajar con capas e insertar comportamientos JavaScript, de una forma muy sencilla y visual. Además incluye un software de cliente FTP completo, permitiendo entre otras cosas trabajar con mapas visuales de los sitios Web, actualizando el sitio Web en el servidor sin salir del programa.

Algunas Características: ⁽³⁰⁾

- Integración de RSS: con Dreamweaver 8 se puede integrar entradas RSS provenientes de otras páginas con sólo introducir la fuente y arrastrar y colocar los campos. De esta forma podrás introducir datos en formato XML fácil y cómodamente.
- Mejoras CSS: esta última versión ha mejorado mucho respecto a la compatibilidad y manejo de estilos de cascada. De esta forma, se ha mejorado el panel de estilos CSS, donde ahora podrás acceder a la configuración de cada uno de los estilos desde una lista mucho mejor, dotado de una cuadrícula editable desde donde podrás modificar sus propiedades. Además, Dreamweaver 8, añade una nueva barra de herramientas que proporciona la reproducción inmediata de los estilos para diferentes medios (pantalla, impresora, WebTV y PDAs).
- Accesibilidad: Dreamweaver 8 incorpora las normas de accesibilidad de prioridad 2 marcadas por la WCAG/W3C.
- Transferencia de archivos: Con Dreamweaver 8, se puede seguir trabajando con los archivos mientras el programa se comunica con el servidor, e incluye los archivos creados o modificados recientemente. Su sincronización ha mejorado notablemente siendo posible una mejor gestión de cambios, además de permitir el uso de bloqueo/desbloqueo de archivos para que estos no se sobrescriban.
- Interfaz mejorada: Los usuarios con problemas visuales, podrán acceder a una opción de aumento de la pantalla, en vista de diseño, para analizar o trabajar con difíciles anidamientos de tablas. Además de la inclusión de información visual gracias a las guías, que permitirán la medición píxel a píxel de todos los elementos.
- Nueva barra de herramientas: Se ha añadido una barra de herramientas a Dreamweaver 8, podrás encontrarla en la parte lateral izquierda del modo de código, esta barra hace mucho más accesible el código al permitir la navegación por etiquetas y su contracción. Una de las nuevas novedades es la posibilidad de añadir comentarios con un sólo clic.

1.9.4 – ER/Studio

Es una herramienta de modelado de datos fácil de usar y multinivel, para el diseño y construcción de Bases de datos a nivel físico y lógico. Direcciona las necesidades diarias de los administradores de Bases de datos, desarrolladores y arquitectos de datos que construyen y mantienen aplicaciones de Bases de datos grandes y complejas.

ER/Studio está equipado para crear y manejar diseños de Bases de datos funcionales y confiables. Ofrece fuertes capacidades de diseño lógico, sincronización bidireccional de los diseños físicos y lógicos, construcción automática de Bases de datos, documentación y fácil creación de reportes.

ER/Studio ofrece las siguientes funcionalidades: [\(31\)](#)

- Capacidad fuerte en el diseño lógico.
- Sincronización bidireccional de los diseños lógico y físico.
- Construcción automática de Base de datos.
- Reingeniería inversa de Base de datos.
- Documentación basada en HTML.
- Un Repositorio para el modelado.

En su versión 7.1, ER/Studio ayuda, mediante sus capacidades de modelado empresarial y colaboración, a las organizaciones a tener un acercamiento global a dos importantes aspectos de la administración de datos: la reducción de riesgos asociados a la seguridad de la información y el mejoramiento de la calidad del modelo de los datos. Con ER/Studio 7.1, los arquitectos de datos, pueden facilitar el cumplimiento de requerimientos y reducir el riesgo de que la información sea utilizada en forma inapropiada, utilizando capacidades mejoradas para la identificación, clasificación y comunicación de políticas alrededor de la información sensible. Adicionalmente, ER/Studio 7.1, mejoró sustancialmente los modelos de consistencia y calidad al proveer reglas de validación para reforzar estándares específicos y prácticas a través de toda la empresa. [\(32\)](#)

1.9.5 – Aptana Studio

Aptana Studio es un IDE (Integrated Development Environment) de desarrollo para aplicaciones de la Web 2.0, gratuito, código libre, con soporte Ajax, PHP, Ruby on Rails, Adobe Air, iPhone, entre otros. Con Aptana se facilita un desarrollo integrado de

Ajax con las tecnologías emergentes. Aptana está basado en el conocido entorno de desarrollo Eclipse (IDE), también Open Source. Pero, mientras que Eclipse está focalizado en el desarrollo para Java, Aptana Studio es una distribución focalizada en el desarrollo Web, con soporte a HTML, CSS y Javascript, así como opcionalmente a otras tecnologías mencionadas como PHP, Adobe Air o Ruby on Rails. Aptana Studio está disponible como una aplicación independiente o como plug-in para Eclipse. El programa se distribuye para todos los Sistemas operativos más comunes: Windows, Linux y MacOS X.

Características principales de Aptana Studio: [\(33\)](#)

- Ayudas visuales para la escritura de scripts en diversos lenguajes, como coloreado y auto escritura del código, ayudas contextuales de referencia a medida que se escribe, entre otras.
- Visualización de errores de sintaxis a medida que se escribe.
- Soporte para hacer FTP a servidores remotos, con herramientas para sincronización.
- Debug en Firefox (Debug Internet Explorer también con la versión Profesional)
- Librerías de funciones en Javascript populares en Ajax/Javascript para utilizar en los proyectos.
- Ejemplos ya creados para empezar a conocer las posibilidades de desarrollo rápidamente.
- Previsualización de estilos CSS con el editor CSS.
- Extensible a partir de plug-ins que puede crear Aptana u otras empresas y herramientas para estar al tanto de cualquier nuevo añadido.
- Extensible por Javascript, los usuarios pueden escribir scripts para realizar acciones y macros.
- Los Snippets permiten insertar fragmentos de texto que se utilizan muy a menudo.

1.9.6 – Zend Studio

Se trata de un programa de la casa Zend, impulsores de la tecnología de servidor PHP, orientada a desarrollar aplicaciones Web. El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración de código.

El programa entero está escrito en Java, lo que a veces supone que no funcione tan rápido como otras aplicaciones de uso diario. Sin embargo, esto ha permitido a Zend lanzar con relativa facilidad y rapidez versiones del producto para Windows, Linux y MacOS, aunque el desarrollo de las versiones de este último Sistema se retrase un poco más.

Zend Studio consta de dos módulos, en los cuales se dividen las funcionalidades de parte del cliente y las del servidor. Los dos módulos se instalan por separado, el módulo cliente contiene el interfaz de edición y la ayuda, permite además hacer depuraciones simples de scripts. Para disfrutar de toda la potencia de la herramienta de depuración habrá que disponer del módulo servidor, que instala Apache y PHP o, en caso de que estén instalados, los configura para trabajar juntos en depuración. [\(34\)](#)

Conclusiones

En este capítulo se realizó un estudio detallado de todas las tecnologías, metodologías, herramientas y técnicas que serán usadas en el desarrollo del software, así como una breve justificación de su uso. La decisión tomada estuvo basada en las opciones que con mayor eficiencia pueda responder a las necesidades del cliente. Las herramientas y tecnologías seleccionadas son: como gestor de Base de datos: MySQL; Servidor Web: Apache; lenguaje de programación del lado del servidor: PHP; lenguajes de programación del lado del cliente: XHTML, XML, JavaScript y AJAX y como Metodología de Desarrollo de Software: RUP con notación UML. Para apoyar el proceso de implementación del software se utilizarán herramientas de desarrollo como Dreamweaver, ZendStudio, Aptana Studio, ERStudio, EMS MySQL Manager y Rational Rose Enterprise Edition.

CAPÍTULO 2

CARACTERÍSTICAS DEL SISTEMA

Introducción

En el presente capítulo se ofrece una descripción del Negocio en el que se va a trabajar, se especifican los actores y trabajadores del Negocio, los Casos de uso, los diagramas de actividades y el modelo de objetos del Negocio.

Para el buen desarrollo del Sistema a desarrollar es necesario basarse en los requerimientos funcionales y no funcionales, a partir de los cuales se identifican las opciones del Sistema, que se representan mediante los Casos de uso del Sistema y la descripción de los mismos.

Modelo de Negocio

2.1 – Modelo de Negocio

El Modelo de Negocio es un artefacto del flujo de trabajo Modelamiento del Negocio en la Fase de Inicio de la metodología RUP. Sus objetivos son: comprender la estructura y la dinámica del centro en el cual se va a implantar el Sistema, comprender los problemas existentes en la actualidad dentro de la unidad e identificar las mejoras potenciales, asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común del centro y derivar los requerimientos del Sistema que va a soportar la organización.

Descripción de los procesos de Negocio.

El flujo y procesamiento de la información que se maneja en la Vice-dirección de Docencia e Investigaciones del Policlínico - Hospital Ernesto Che Guevara, comienza cuando alguno de los trabajadores de dicho Hospital empieza alguna actividad investigativa o docente, estos informan a los respectivos Jefes de Departamentos (Departamento de Asistencia Médica, Departamento de Enfermería y Departamento de Estomatología) de dicha actividad y a su vez los Jefes de Departamentos se lo comunican al Vice-director de Docencia e Investigaciones quien procesa y registra los datos docentes e investigativos de cada trabajador.

Esta información es recopilada y almacenada con el fin de tener el control de todos los trabajadores que se encuentren desarrollando algún tipo de actividad docente o de investigaciones, para elaborar posteriormente los distintos reportes que tiene que generar el Vice-director de Docencia e Investigaciones, tales como: Control de Créditos Académicos, Control de la Cantidad de Estudiantes, Investigación de Enseñanza y Aprendizaje y Plan de Desarrollo Individual.

2.1.1 – Reglas del Negocio

Las Reglas de Negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del Negocio:

- 1 Solo el Vice-director de Docencia e Investigaciones puede tener acceso a la información de todos los trabajadores.
- 2 Los Jefes de Departamentos tienen acceso solamente a la información de sus trabajadores.
- 3 Los trabajadores sólo tienen acceso a su Plan de Trabajo Individual.
- 4 Cada trabajador puede tener sólo un Plan de Trabajo Individual correspondiente a un determinado mes.
- 5 El Vice-director de Docencia e Investigaciones mantiene un control estricto sobre los créditos académicos alcanzado por cada uno de los trabajadores.

2.1.2 – Actores del Negocio

Un Actor del Negocio es cualquier individuo, grupo, organización o máquina que interactúa con el Negocio.

Actores del Negocio	Descripción
Trabajadores	Trabajadores del Policlínico – Hospital que estén realizando actividades de Pregrado y/o Postgrado.
Jefes de Departamentos	Jefes de cada uno de los Departamentos del Policlínico - Hospital, encargados de mantener informado al Vice-director de Docencia e Investigaciones con los datos actualizados de sus trabajadores.

Tabla 1: Actores del Negocio

2.1.3 – Trabajadores del Negocio

Define el comportamiento y responsabilidad (rol) de un individuo, grupo de individuos, Sistema automatizado o máquina, que trabajan en conjunto como un equipo. Ellos realizan las actividades y son propietarios de elementos.

Trabajadores del Negocio	Descripción
Vice-director de Docencia e Investigaciones.	Máximo encargado de la gestión de toda la información respecto a Postgrado y Pregrado de los trabajadores del centro.
Informático	Encargado del registro de la información docente e investigativa de los trabajadores del centro.

Tabla 2: Trabajadores del Negocio

2.1.4 – Entidades del Negocio

Representa un contenedor de información, algo físico que se utilice en el proceso del Negocio y que sirva para obtener información o para actualizar información.

1. Control de Créditos Académicos
2. Plan de Desarrollo Individual
3. Plan de Ciencia e Innovación Tecnológica
4. Plan de Superación Profesional
5. Identificación de Necesidades de Aprendizaje
6. Plan de Trabajo Individual
7. Listado Trabajadores Estudiantes
8. Listado Trabajadores Profesores

2.1.5 – Casos de uso del Negocio

Un Caso de uso del Negocio representa a un proceso de Negocio, por lo que se corresponde con una secuencia de acciones que producen un resultado observable para ciertos actores del Negocio.

1. Gestionar proceso de capacitación de los trabajadores.
2. Gestionar proceso de superación de los trabajadores.
3. Controlar Plan de Trabajo Individual de los trabajadores.
4. Gestionar Actividad Científica del centro.

2.1.6 – Diagrama de Casos de uso del Negocio

Un diagrama de Casos de uso del Negocio representa gráficamente a los procesos del Negocio y su interacción con los actores del Negocio.

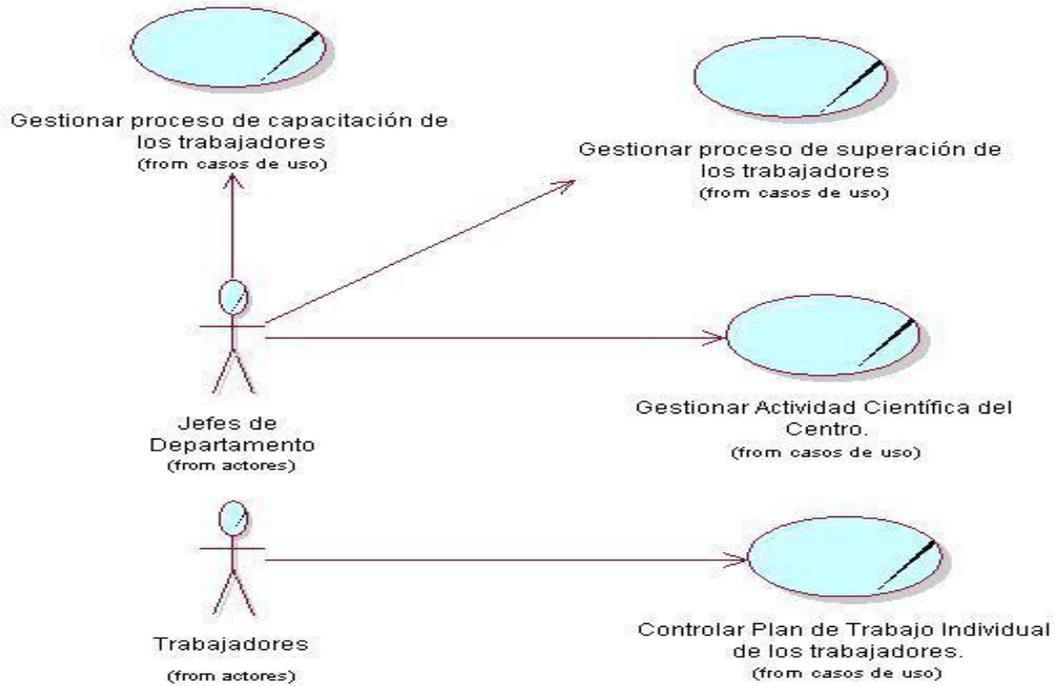


Figura 2.1 – Diagrama de Casos de uso del Negocio

2.1.7– Descripción de los Principales Casos de uso del Negocio

2.1.7.1– Descripción del Caso de uso del Negocio Gestionar proceso de capacitación de los trabajadores

Caso de uso:	Gestionar proceso de capacitación de los trabajadores
Actores:	Jefe de Departamento
Propósito:	Gestionar la información de pregrado de los trabajadores.
Resumen:	El Caso de uso se inicia cuando el Jefe de Departamento recopila y envía al Vice-director de Docencia e Investigaciones toda la información de pregrado de los trabajadores.
Casos de uso asociados:	
Flujo de Trabajo	

Acción del Actor	Respuesta del Negocio
1. El Jefe de Departamento envía los datos de los trabajadores.	2. El Vice-director de Docencia e Investigaciones actualiza el Plan de Desarrollo Individual de cada trabajador, verifica que estén todos los datos necesarios y se lo envía al Informático.
	3. El Informático imprime el Plan de Desarrollo Individual de cada uno de los trabajadores.
	4. El Vice-director de Docencia e Investigaciones envía un mensaje al recibir todos los datos necesarios.
5. El Jefe de Departamento recibe el mensaje.	
Prioridad:	Alta
Cursos Alternos	
Acción del Actor	Respuesta del Negocio
	4.1) El Vice-director de Docencia e Investigaciones solicita al Jefe de Departamento los datos que faltan y pasa a la acción 2 del flujo normal de eventos de esta sección.

2.1.7.2 – Descripción del Caso de uso del Negocio Gestionar proceso de superación de los trabajadores

Caso de uso:	Gestionar proceso de superación de los trabajadores
Actores:	Jefe de Departamento
Propósito:	Gestionar la información de postgrado de los trabajadores.
Resumen:	El Caso de uso se inicia cuando el Jefe de Departamento recopila y envía al Vice-director de Docencia e Investigaciones toda la información de postgrado de los trabajadores.
Casos de uso asociados:	
Flujo de Trabajo	

Acción del Actor	Respuesta del Negocio
1. El Jefe de Departamento envía los datos de los trabajadores.	2. El Vice-director de Docencia e Investigaciones actualiza los Créditos Académicos y el Plan de Superación Profesional de cada trabajador, verifica que estén todos los datos necesarios y se lo envía al Informático.
	3. El Informático imprime los Créditos Académicos y el Plan de Superación Profesional de cada uno de los trabajadores.
	4. El Vice-director de Docencia e Investigaciones envía un mensaje por recibir los datos satisfactoriamente.
5. El Jefe de Departamento recibe el mensaje.	
Prioridad:	Alta
Cursos Alternos	
Acción del Actor	Respuesta del Negocio
	4.1) El Vice-director de Docencia e Investigaciones solicita los datos que faltan y pasa a la acción 2 del flujo normal de eventos de esta sección.

2.1.7.3– Descripción del Caso de uso del Negocio Controlar Plan de Trabajo Individual de los trabajadores

Caso de uso:	Controlar Plan de Trabajo Individual de los trabajadores
Actores:	Trabajadores
Propósito:	Controlar Plan de Trabajo Individual de los trabajadores.
Resumen:	El Caso de uso se inicia cuando un trabajador envía su Plan de Trabajo Individual para ser revisado por el Vice-director de Docencia e Investigaciones.
Casos de uso asociados:	

Flujo de Trabajo	
Acción del Actor	Respuesta del Negocio
1. El trabajador envía su Plan de Trabajo Individual.	2. Recibe el Plan de Trabajo Individual, lo revisa y se lo envía al Informático.
	3. El Informático imprime el Plan de Trabajo Individual de cada trabajador.
	4. El Vice-director de Docencia e Investigaciones envía un mensaje por recibir los datos satisfactoriamente.
5. El Trabajador recibe el mensaje.	
Prioridad:	Alta
Cursos Alternos	
Acción del Actor	Respuesta del Negocio
	4.1) Si el Vice-director de Docencia e Investigaciones encuentra algún error en el Plan se lo informa al trabajador para ser arreglado y pasa a la acción 2 del flujo normal de eventos de esta sección.

2.1.7.4 – Descripción del Caso de uso del Negocio Gestionar Actividad Científica del Centro

Caso de uso:	Gestionar Actividad Científica del Centro
Actores:	Jefe de Departamento
Propósito:	Gestionar la información respecto a las actividades científicas que se realicen en el centro.
Resumen:	El Caso de uso se inicia cuando el Jefe de Departamento recopila y envía al Vice-director de Docencia e Investigaciones toda la información de las actividades científicas que se realicen en el centro.
Casos de uso asociados:	
Flujo de Trabajo	

Acción del Actor	Respuesta del Negocio
1. El Jefe de Departamento envía los Proyectos Investigativos desarrollados en cada departamento.	2. El Vice-director de Docencia e Investigaciones actualiza el Plan de Ciencia e Innovación Tecnológica y las Necesidades de Aprendizaje, verifica que estén todos los datos necesarios y se lo envía al Informático.
	3. El Informático imprime el Plan de Ciencia e Innovación Tecnológica y las Necesidades de Aprendizaje.
	4. El Vice-director de Docencia e Investigaciones envía un mensaje por recibir los datos satisfactoriamente.
5. El Jefe de Departamento recibe el mensaje.	
Prioridad:	Alta
Cursos Alternos	
Acción del Actor	Respuesta del Negocio
	4.1) El Vice-director de Docencia e Investigaciones solicita los datos que faltan y pasa a la acción 2 del flujo normal de eventos de esta sección.

2.1.8 – Modelo de Objetos

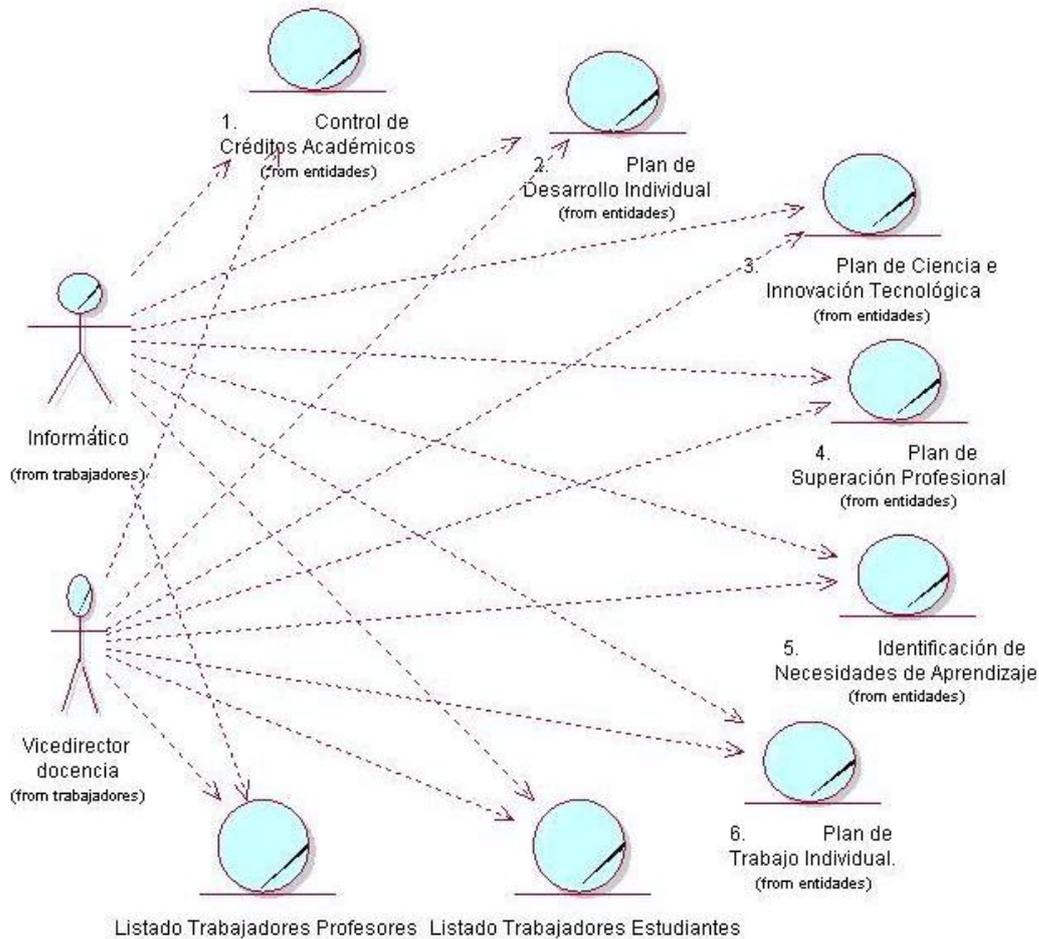


Figura 2.2 – Modelo de Objetos

Modelo de Sistema

2.2 – Especificación de Requerimientos para el Sistema de Gestión de la Información de la VDDI

2.2.1 – Requisitos Funcionales

Los requerimientos funcionales son capacidades o condiciones que el Sistema debe cumplir.

- RF1 Autenticar Usuario
 - 1.1 Verificar Privilegios
- RF2 Gestionar Usuario
 - 1.1 Crear Usuario



1.2 Modificar Usuario

1.3 Eliminar Usuario

RF3 Buscar Usuario

RF4 Listar Usuario

RF5 Gestionar Trabajador

1.1 Crear Trabajador

1.2 Modificar Trabajador

1.3 Eliminar Trabajador

RF6 Buscar Trabajador

RF7 Listar Trabajador

RF8 Gestionar Trabajador por Actividad

1.1 Adicionar Trabajador a una Actividad

1.2 Eliminar Trabajador de una Actividad

RF9 Buscar Trabajador por Actividad

RF10 Buscar Actividad por Trabajador

RF11 Gestionar Actividad

1.1 Crear Actividad

1.2 Modificar Actividad

1.3 Eliminar Actividad

RF12 Buscar Actividad

RF13 Listar Actividad

RF14 Gestionar INA

1.1 Crear Necesidad

1.2 Modificar Necesidad

1.3 Eliminar Necesidad

RF15 Buscar Necesidad

RF16 Listar Necesidad

RF17 Gestionar Plan de Ciencia e Innovación

1.1 Crear Problema

1.2 Modificar Problema

1.3 Eliminar Problema

RF18 Buscar Problema

RF19 Listar Problema

RF20 Gestionar Plan de Trabajo

- 1.1 Crear Plan de Trabajo
- 1.2 Modificar Plan de Trabajo
- 1.3 Eliminar Plan de Trabajo

RF21 Buscar Plan de Trabajo

RF22 Listar Plan de Trabajo

RF23 Generar Reportes

- 1.1 Generar Control de Créditos
- 1.2 Generar Plan de Desarrollo Individual
- 1.3 Generar Plan de Ciencia e Innovación
- 1.4 Generar Plan de Superación Profesional
- 1.5 Generar INA
- 1.6 Generar Plan de Trabajo Individual
- 1.7 Generar Listado Trabajadores Estudiantes
- 1.8 Generar Listado Trabajadores Profesores

2.2.2 – Requisitos No Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido y confiable.

RNF1 Interfaz Externa

- 1.1 Interfaz sencilla, amigable y de fácil uso para el usuario.
- 1.2 Navegación simple.

RNF2 Soporte

- 1.1 Servidor Web de fácil manejo y de código abierto.
- 1.2 Servidor de Bases de datos que soporte grandes volúmenes de información a procesar y en un tiempo mínimo.
- 1.3 Navegador capaz de soportar tecnología AJAX.

RNF3 Portabilidad

- 1.1 Para lograr un mejor funcionamiento la computadora cliente debe tener instalado el navegador Internet Explorer 6.0 o una versión superior.

RNF4 Disponibilidad

- 1.1 El Sistema debe prestar servicio las 24 horas del día.

RNF5 Seguridad

- 1.1 Autenticación de usuario necesaria para acceder al Sistema.
- 1.2 Garantizar diferentes niveles de acceso al Sistema, brindado por la asignación de roles.

RNF6 Software

- 1.1 MySQL versión 5.0 u otra versión superior que permita el uso de Bases de datos relacionales.
- 1.2 PHP versión 5.0 o superior.
- 1.3 Servidor Web Apache versión 2.0 o superior.

RNF7 Hardware

- 1.1 RAM del servidor de 256 MByte o superior.
- 1.2 Velocidad del Microprocesador de 1.42 GHz o superior.

2.2.3 – Actores del Sistema

Son todas aquellas personas u otros Sistemas que interactúan con el mismo.

Actores del Sistema	Descripción
Trabajadores	Trabajadores del Policlínico – Hospital que estén realizando actividades de Pregrado o Postgrado, los cuales acceden al Sistema para adicionar o modificar su Plan de Trabajo Individual.
Jefes de Departamentos	Jefes de los Departamentos del Policlínico – Hospital, los cuales acceden al Sistema para gestionar la información de los trabajadores que pertenecen a sus departamentos.
Administradores	Máximos encargados de la gestión de toda la información respecto a Postgrado y Pregrado de los trabajadores del centro. Dentro de estos actores entran el Vice-director de Docencia e Investigaciones y el Informático de la Vice-dirección.

Tabla 3: Actores del Sistema



2.2.4 – Diagrama de Casos de uso del Sistema

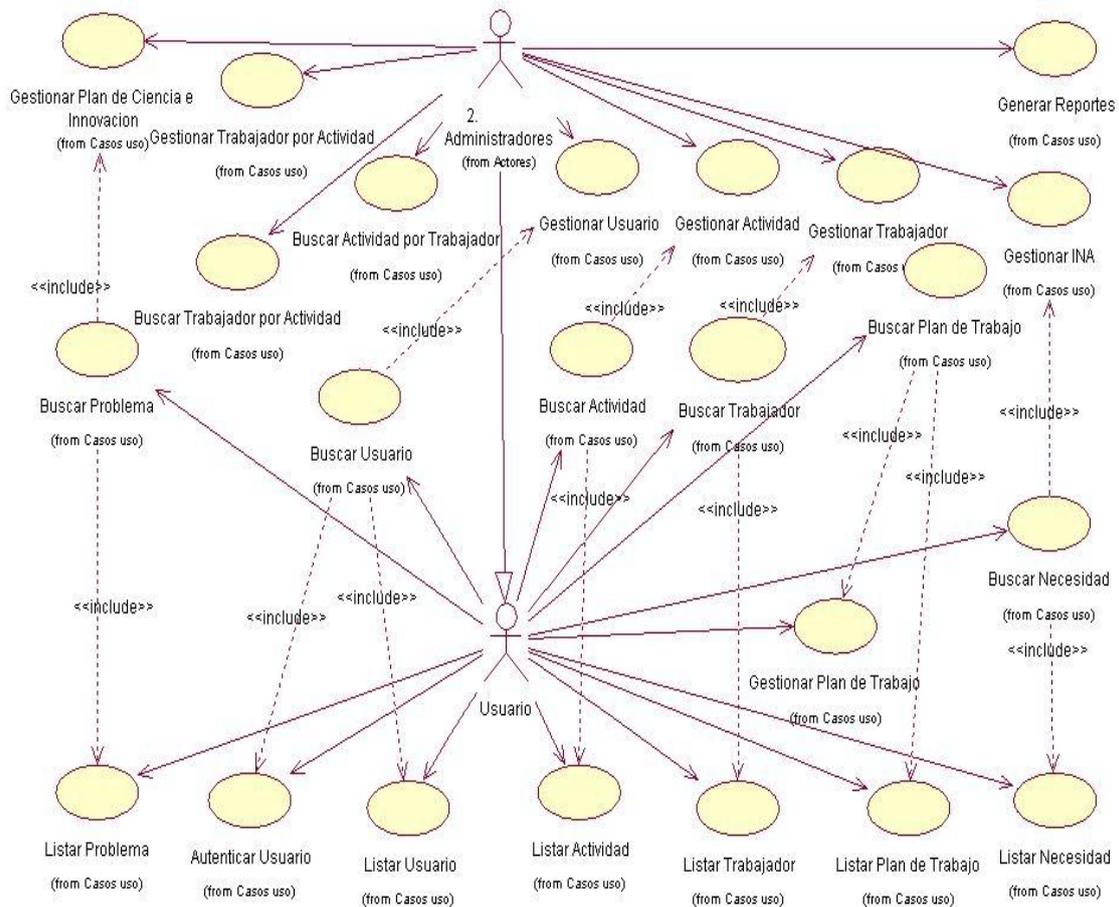


Figura 2.3 – Diagrama de Casos de uso del Sistema

2.2.5 – Descripción de Casos de uso del Sistema

La descripción de Casos de uso del Sistema se realiza con el objetivo de describir brevemente como funcionarán los Casos de uso en el Sistema, y así conseguir un mejor entendimiento de cada uno.

2.2.5.1 – Descripción del Caso de uso del Sistema Autenticar Usuario.

Nombre del CU	Autenticar Usuario
Actores	Vice-director de Docencia e Investigaciones, Informático, trabajadores, Jefes de Departamentos.
Propósito	Verificar Privilegios acorde con el tipo de usuario
Resumen	El Caso de uso lo inicia cuando algún tipo de usuario

	accede al Sistema y éste verifica los privilegios del tipo de usuario.
Referencias	RF1
Prioridad	Crítico
Sección “1.1 Verificar Privilegios”: Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario inserta usuario y contraseña.	2. El Sistema verifica que el usuario y la contraseña sean correctos y existan. 3. Muestra una interfaz distinta dependiendo el tipo de usuario y los privilegios que tiene.
Sección “Verificar Privilegios”: Flujo Alternativo	
1. Si usuario no desean realizar otra acción, termina el CU.	2. Si quedó algún campo vacío, el Sistema envía un mensaje. 3. Si el usuario o la contraseña esta incorrecta el Sistema envía un mensaje.

2.2.5.2 – Descripción del Caso de uso del Sistema Gestionar Trabajador

Nombre del CU	Gestionar Trabajador
Actores	Vice-director de Docencia e Investigaciones, Informático, Jefes de Departamentos.
Propósito	Crear, modificar o eliminar un trabajador.
Resumen	El Caso de uso se inicia el Vice-director de Docencia e Investigaciones, el Informático o el Jefe de Departamento solicitan la creación, modificación o eliminación de un trabajador.
Referencias	RF5
Precondiciones	Que el Vice-director de Docencia e Investigaciones, el Informático y el Jefe de Departamento estén autenticados en el Sistema.

Prioridad	Crítico
Sección “Crear Trabajador ”: Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Vice-director de Docencia e Investigaciones, el Informático o el Jefe de Departamento selecciona la opción para crear un nuevo trabajador.	2. El Sistema muestra la interfaz para insertar los datos del nuevo trabajador.
3. El Vice-director de Docencia e Investigaciones, el Informático o el Jefe de Departamento inserta los datos del trabajador.	4. El Sistema verifica que los campos obligatorios no estén vacíos. 5. El Sistema registra los datos.
Sección “Crear Trabajador”: Flujo Alternativo	
1. Si el Vice-director de Docencia e Investigaciones, el Informático o el Jefe de Departamento no desea realizar otra acción, termina el CU.	2. Si quedó algún campo obligatorio vacío, el Sistema envía un mensaje.
Sección “Modificar Trabajador”: Flujo Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Vice-director de Docencia e Investigaciones, el Informático o el Jefe de Departamento selecciona la opción modificar trabajador.	2. El Sistema muestra una interfaz para modificar los campos deseados.
3. El Vice-director de Docencia e Investigaciones, el Informático o el Jefe de Departamento modifica los campos deseados.	4. El Sistema verifica que los campos obligatorios no estén vacíos. 5. El Sistema registra y actualiza los nuevos datos.
Sección “Modificar Trabajador”: Flujo Alternativo	
1. Si el Vice-director de Docencia e Investigaciones, el Informático o el Jefe de Departamento no desea realizar otra acción, termina el CU.	2. Si quedó algún campo obligatorio vacío, el Sistema envía un mensaje.

Sección “Eliminar Trabajador”: Flujo Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
<p>1. El Vice-director de Docencia e Investigaciones, el Informático o el Jefe de Departamento selecciona la opción eliminar trabajador que se encontrará en forma de X al lado del nombre y los datos del trabajador.</p> <p>3. El actor confirma que desea eliminar al trabajador</p>	<p>2. El Sistema muestra una confirmación de si se desea eliminar o no el trabajador</p> <p>4. El Sistema elimina el trabajador</p>
Sección “Eliminar Trabajador”: Flujo Alternativo	
<p>1. Si el Vice-director de Docencia e Investigaciones, el Informático y el Jefe de Departamento no desea realizar otra acción, termina el CU.</p>	
Poscondiciones	<p>Los datos insertados o modificados deben actualizarse en la Base de datos.</p>

2.2.5.3 – Descripción del Caso de uso del Sistema Buscar Trabajador

Nombre del CU	Buscar Trabajador
Actores	Vice-director de Docencia e Investigaciones, Informático, trabajadores, Jefes de Departamentos.
Propósito	Buscar trabajador.
Resumen	El Caso de uso lo inicia el Vice-director de Docencia e Investigaciones y/o el Informático cuando solicita la búsqueda de los trabajadores.
Referencias	RF6
Precondiciones	Que el Vice-director de Docencia e Investigaciones, Informático, trabajadores, Jefes de Departamentos estén autenticados en el Sistema.
Prioridad	Crítico

Sección “Buscar Trabajador”: Flujo Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1. El usuario selecciona la opción buscar trabajador.	2. El Sistema muestra una interfaz para buscar el trabajador.
3. El usuario inserta el nombre o CI del trabajador deseado.	4. El Sistema muestra los datos del trabajador.
Sección “Buscar Trabajador”: Flujo Alternativo	
1. Si el usuario no desea realizar otra acción, termina el CU.	2. Si quedó algún campo obligatorio vacío, el Sistema envía un mensaje. 3. Si el trabajador no existe el Sistema muestra un mensaje.
Poscondiciones	Los datos creados o modificados deben actualizarse en la Base de datos.

2.2.5.4 – Descripción del Caso de uso del Sistema Listar Trabajador

Nombre del CU	Listar Trabajador
Actores	Vice-director de Docencia e Investigaciones, Informático, trabajadores, Jefes de Departamentos.
Propósito	Listar trabajador.
Resumen	El Caso de uso lo inicia cuando el Vice-director de Docencia e Investigaciones, Informático, trabajadores, Jefes de Departamentos solicitan la opción Listar trabajador.
Referencias	RF7
Precondiciones	Que el Vice-director de Docencia e Investigaciones, Informático, trabajadores, Jefes de Departamentos, estén autenticados en el Sistema.
Prioridad	Crítico
Sección “Listar Trabajador”: Flujo Normal de Eventos	

Acciones del Actor		Respuesta del Sistema
1. El usuario selecciona la opción listar trabajador.		2. El Sistema muestra una interfaz con el listado de los trabajadores.
Sección “Listar Trabajador”: Flujo Alternativo		
1. Si el usuario no desea realizar otra acción, termina el CU.		2. El Vice-director de Docencia e Investigaciones y/o el Informático pueden eliminar o modificar el trabajador que deseen.
Poscondiciones	Los datos creados o modificados deben actualizarse en la Base de datos.	

2.2.5.5 – Descripción del Caso de uso del Sistema Generar Reportes

Nombre del CU	Generar Reportes
Actores	El Vice-director de Docencia e Investigaciones y/o el Informático
Propósito	Generar reportes.
Resumen	El Caso de uso se inicia cuando el Vice-director de Docencia e Investigaciones y/o el Informático necesitan general algún tipo de reporte.
Referencias	RF23
Precondiciones	Que el Vice-director de Docencia e Investigaciones y/o el Informático esté autenticado en el Sistema.
Prioridad	Secundario
Sección “Generar Control de Créditos”: Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Vice-director de Docencia e Investigaciones y/o el Informático selecciona la opción generar control de créditos.	2. El Sistema muestra una interfaz en forma de hoja con el control de los créditos.

Sección “Generar Control de Créditos”: Flujo Alternativo	
1. Si el Vice-director de Docencia e Investigaciones y/o el Informático no desea realizar otra acción, termina el CU.	2. El Vice-director de Docencia e Investigaciones y/o el Informático pueden imprimir el control de créditos.
Sección “Generar Plan de Desarrollo Individual”: Flujo Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Vice-director de Docencia e Investigaciones y/o el Informático selecciona la opción generar plan de desarrollo individual.	2. El Sistema muestra una interfaz en forma de hoja con el plan de desarrollo individual.
Sección “Generar Plan de Desarrollo Individual”: Flujo Alternativo	
1. Si el Vice-director de Docencia e Investigaciones y/o el Informático no desea realizar otra acción, termina el CU.	2. El Vice-director de Docencia e Investigaciones y/o el Informático pueden imprimir el plan de desarrollo individual.
Sección “Generar Plan de Ciencia e Innovación”: Flujo Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Vice-director de Docencia e Investigaciones y/o el Informático selecciona la opción generar plan de ciencia e innovación.	2. El Sistema muestra una interfaz en forma de hoja con el plan de ciencia e innovación.
Sección “Generar Plan de Ciencia e Innovación”: Flujo Alternativo	
1. Si el Vice-director de Docencia e Investigaciones y/o el Informático no desea realizar otra acción, termina el CU.	2. El Vice-director de Docencia e Investigaciones y/o el Informático pueden imprimir el plan de de ciencia e innovación.
Sección “Generar Plan de Superación Profesional”: Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Vice-director de Docencia e Investigaciones y/o el Informático	2. El Sistema muestra una interfaz en forma de hoja con el plan de superación

selecciona la opción generar plan de superación profesional.	profesional.
Sección “Generar Plan de Superación Profesional”: Flujo Alternativo	
1. Si el Vice-director de Docencia e Investigaciones y/o el Informático no desea realizar otra acción, termina el CU.	2. El Vice-director de Docencia e Investigaciones y/o el Informático pueden imprimir el plan de superación profesional.
Sección “Generar INA”: Flujo Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Vice-director de Docencia e Investigaciones y/o el Informático selecciona la opción necesidad de aprendizaje.	2. El Sistema muestra una interfaz en forma de hoja con la información de la necesidad de aprendizaje.
Sección “Generar INA”: Flujo Alternativo	
1. Si el Vice-director de Docencia e Investigaciones y/o el Informático no desea realizar otra acción, termina el CU.	2. El Vice-director de Docencia e Investigaciones y/o el Informático pueden imprimir la información de la necesidad de aprendizaje.
Sección “Generar Plan de Trabajo Individual”: Flujo Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Vice-director de Docencia e Investigaciones y/o el Informático selecciona la opción generar plan de trabajo individual.	2. El Sistema muestra una interfaz en forma de hoja con el plan de trabajo individual.
Sección “Generar Plan de Trabajo Individual”: Flujo Alternativo	
1. Si el Vice-director de Docencia e Investigaciones y/o el Informático no desea realizar otra acción, termina el CU.	2. El Vice-director de Docencia e Investigaciones y/o el Informático pueden imprimir el plan de trabajo individual.
Sección “Generar Listado Trabajador-Estudiante”: Flujo Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Vice-director de Docencia e	2. El Sistema muestra una interfaz en

Investigaciones y/o el Informático selecciona la opción generar listado trabajador-estudiante.	forma de hoja con el listado de los trabajadores que son estudiantes.
Sección “Generar Listado Trabajador-Estudiante ”: Flujo Alternativo	
1. Si el Vice-director de Docencia e Investigaciones y/o el Informático no desea realizar otra acción, termina el CU.	2. El Vice-director de Docencia e Investigaciones y/o el Informático pueden imprimir el listado de los trabajadores que son estudiantes.
Sección “Generar Listado Trabajador-Profesor”: Flujo Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Vice-director de Docencia e Investigaciones y/o el Informático selecciona la opción generar listado trabajador-profesor.	2. El Sistema muestra una interfaz en forma de hoja con el listado de los trabajadores que son profesores.
Sección “Generar Listado Trabajador-Profesor”: Flujo Alternativo	
1. Si el Vice-director de Docencia e Investigaciones y/o el Informático no desea realizar otra acción, termina el CU.	2. El Vice-director de Docencia e Investigaciones y/o el Informático pueden imprimir el listado de los trabajadores que son profesores.
Poscondiciones	El reporte debe estar completo.

Conclusiones

En el presente capítulo se realizó toda la descripción del proceso de Negocio y Sistema, obteniendo sus respectivos diagramas de Casos de uso, la descripción de cada caso, sus relaciones y el modelo de objetos del Negocio. También se determinaron las actividades a automatizar por el Sistema, a través de los requerimientos funcionales y con los que se obtuvieron los Casos de uso del Sistema. Además se describieron los requerimientos no funcionales que definen las cualidades que el Sistema debe cumplir.

CAPÍTULO 3

ANÁLISIS Y DISEÑO DEL SISTEMA

Introducción

En este capítulo se realiza el análisis y el diseño del Sistema donde se modelan los principales Casos de uso seleccionados para la próxima iteración del producto, se define el Diagrama de clases del análisis de la aplicación y se especifica qué clases del análisis toman parte del Caso de uso, las relaciones entre ellas y las entidades. Posteriormente mediante el Diagrama de diseño se lleva a cabo todo el flujo de los procesos. En esta etapa se obtiene el modelo de clases de análisis y en el diseño, los Diagramas de secuencia y los Diagramas de clases para el diseño Web.

Análisis del Sistema

El modelo de análisis ayuda a refinar y a estructurar los requisitos, permite entender mejor los aspectos internos del Sistema, ofrece un mayor poder expresivo y una mayor formalización a través de sus diagramas.

3.1 – Diagramas de clases del análisis.

En el diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa las cosas del mundo real, no de la implementación automatizada de las mismas. Además se representa una abstracción de una o varias clases:

Interfaz: Modelan la interacción entre el Sistema y sus actores.

Entidad: Modelan la información que persiste en el tiempo o tiene una larga vida.

Control: Realizan la coordinación, secuenciado de transacciones y el control sobre otros objetos del Sistema.

3.1.1 – Diagrama de clases de análisis de Autenticar Usuario

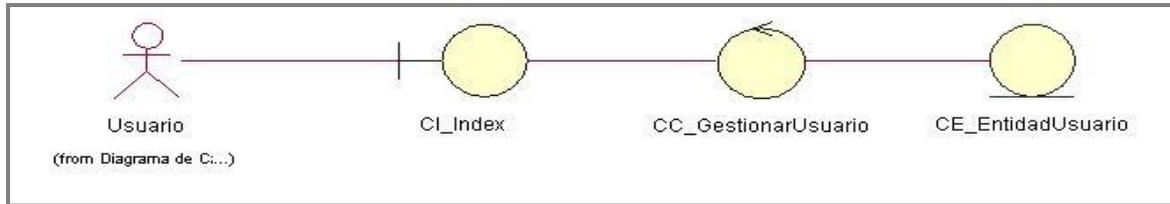


Figura 3.1 – Diagrama de clases de análisis de Autenticar Usuario

3.1.2 – Diagrama de clases de análisis de Buscar Trabajador

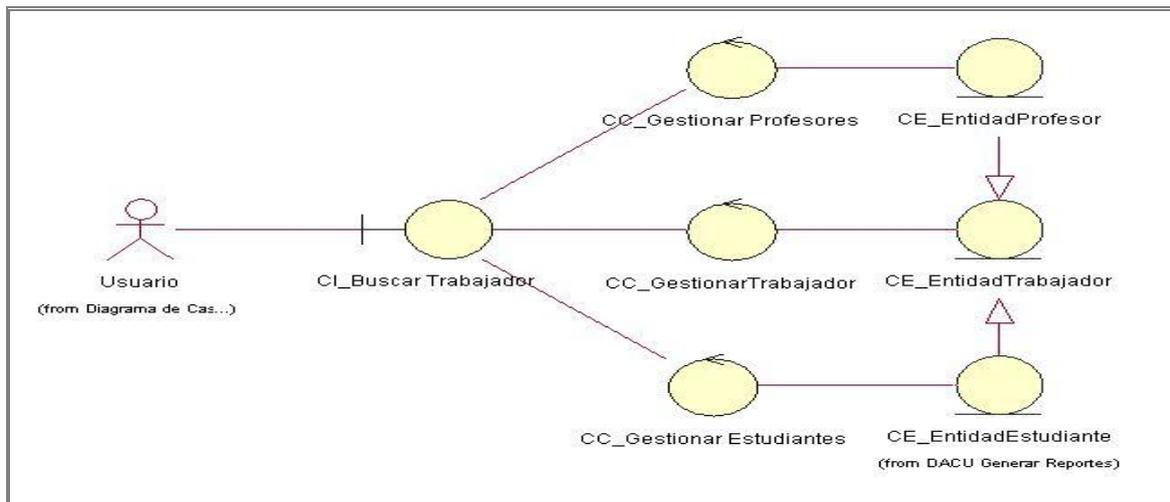


Figura 3.2 – Diagrama de clases de análisis de Buscar Trabajador

3.1.3 – Diagrama de clases de análisis de Generar Reporte (Control de Créditos)

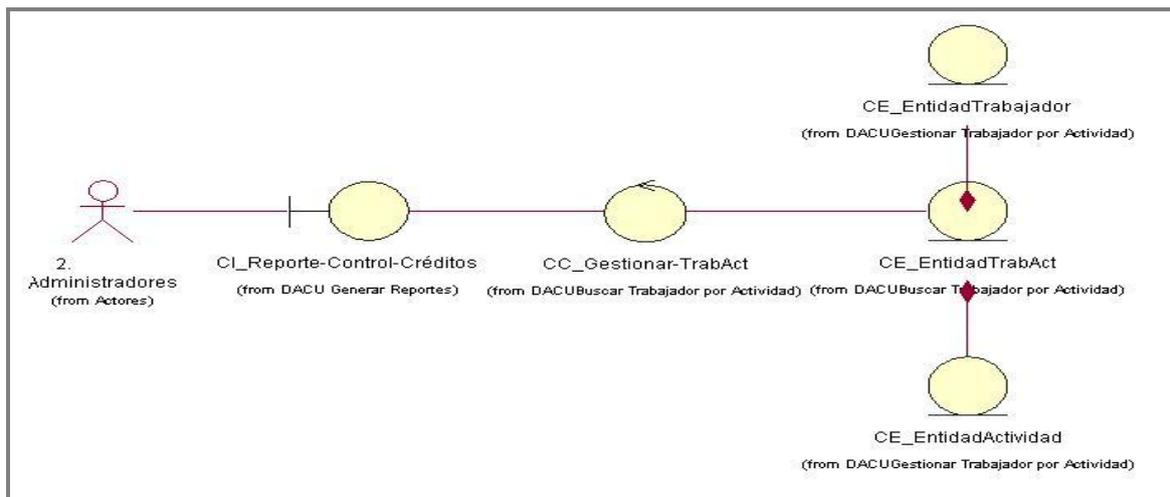


Figura 3.3 – Diagrama de clases de análisis de Generar Reporte (Control de Créditos)



3.1.4 – Diagrama de clases de análisis de Gestionar Trabajador

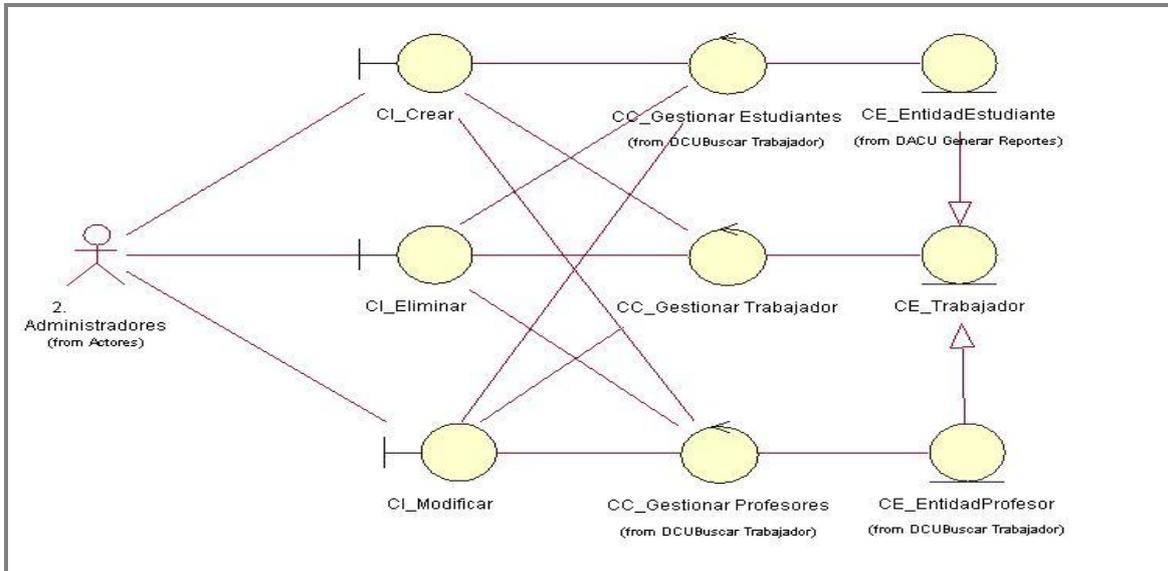


Figura 3.4 – Diagrama de clases de análisis de Gestionar Trabajador

3.1.5 – Diagrama de clases de análisis de Listar trabajador

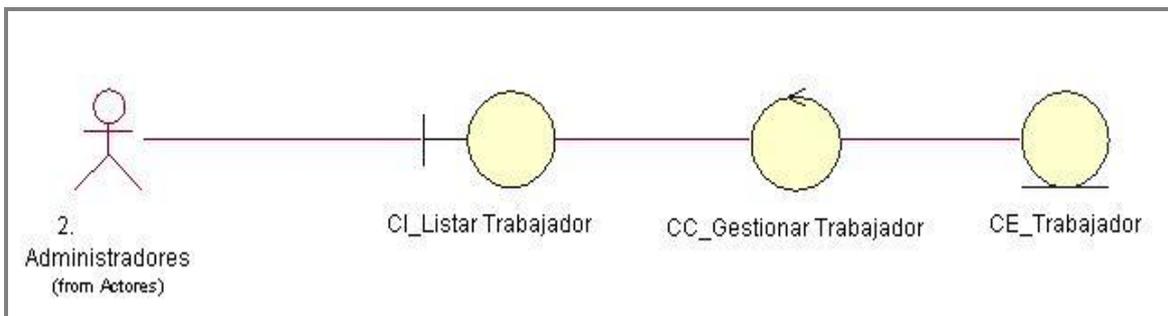


Figura 3.5 – Diagrama de clases de análisis de Listar Trabajador

3.2 – Diagramas de Interacción.

El Diagrama de interacción se utiliza para modelar los aspectos dinámicos del Sistema, consta de un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos. Los Diagramas de interacción contienen: objetos, enlaces y mensajes. Tienen como función principal describir cómo interactúan las clases con sus respectivos objetos del análisis, mediante Diagramas de colaboración o de secuencia, este último contiene las instancias de los actores, los objetos del diseño y las transmisiones de mensajes entre ellos.



3.2.1 – Diagrama de secuencia de Autenticar Usuario

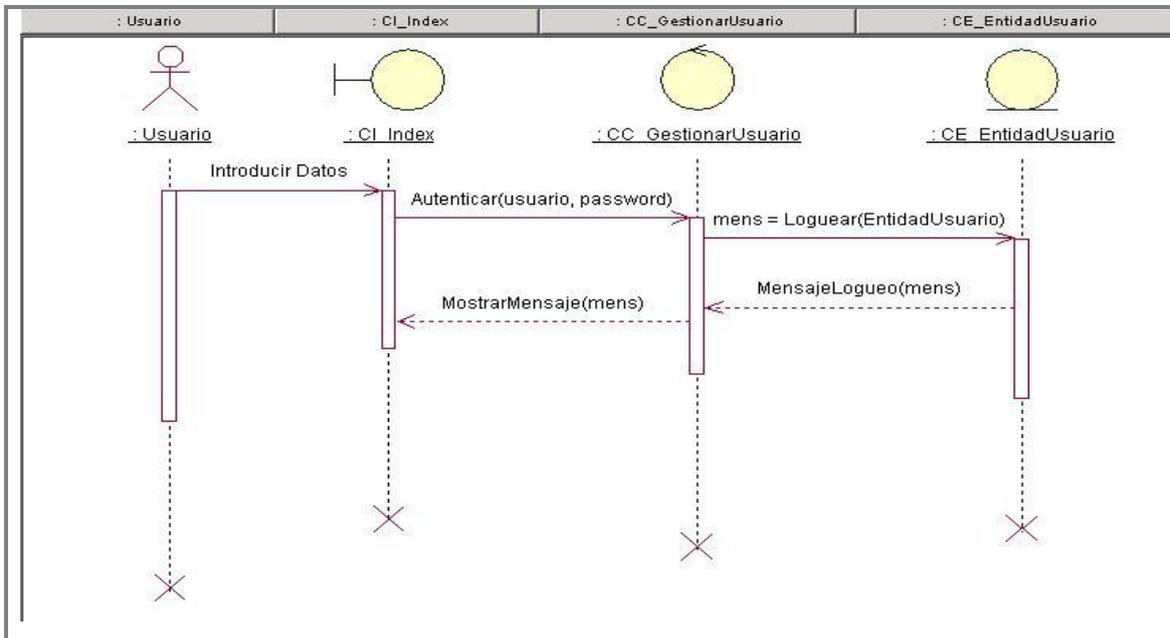


Figura 3.6 – Diagrama de secuencia de Autenticar Usuario

3.2.2 – Diagrama de secuencia de Generar Reporte (Control de Créditos)

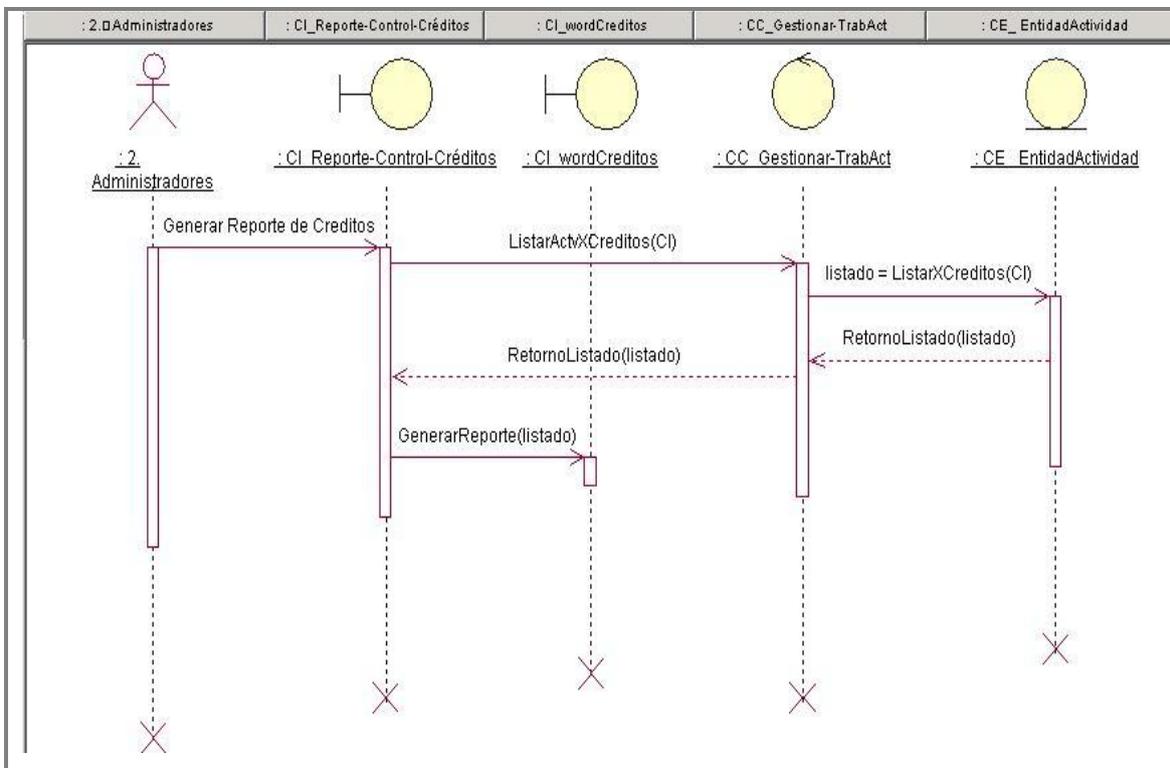


Figura 3.7 – Diagrama de secuencia de Generar Reporte (Control de Créditos)



3.2.3 – Diagrama de secuencia de Listar Trabajador

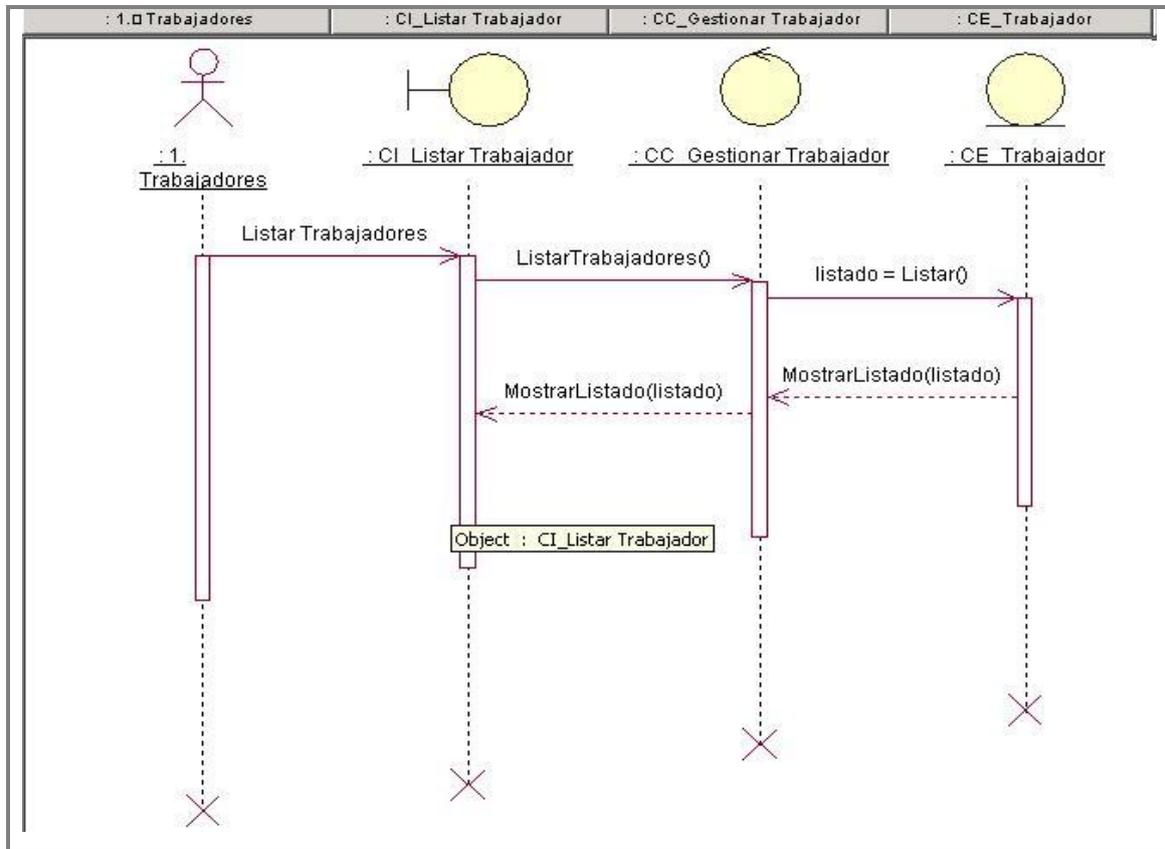


Figura 3.8 – Diagrama de secuencia de Listar Trabajador



3.2.4 – Diagrama de secuencia de Buscar Trabajador

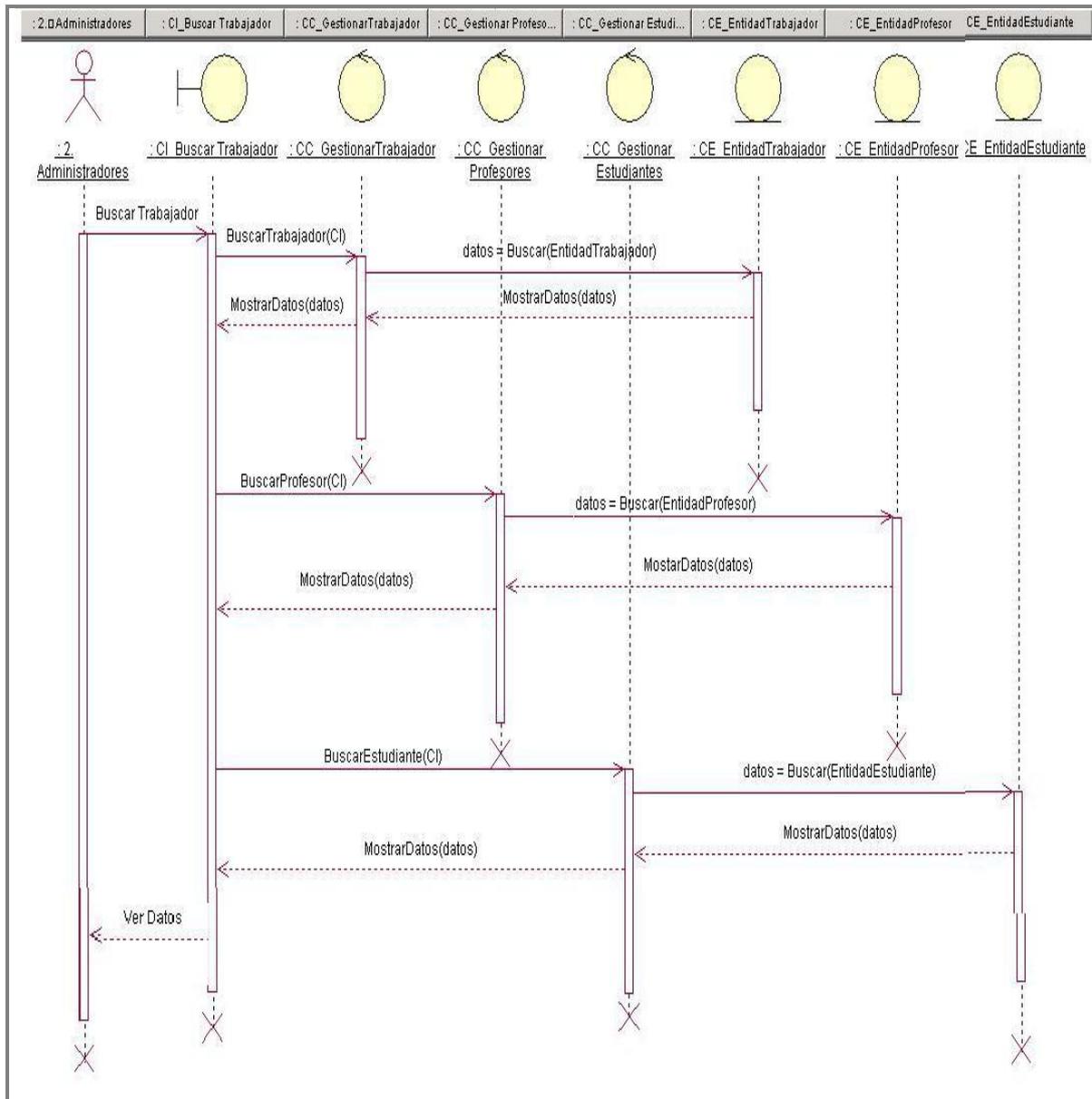


Figura 3.9 – Diagrama de secuencia de Buscar Trabajador



3.2.5 – Diagrama de secuencia de Gestionar Trabajador

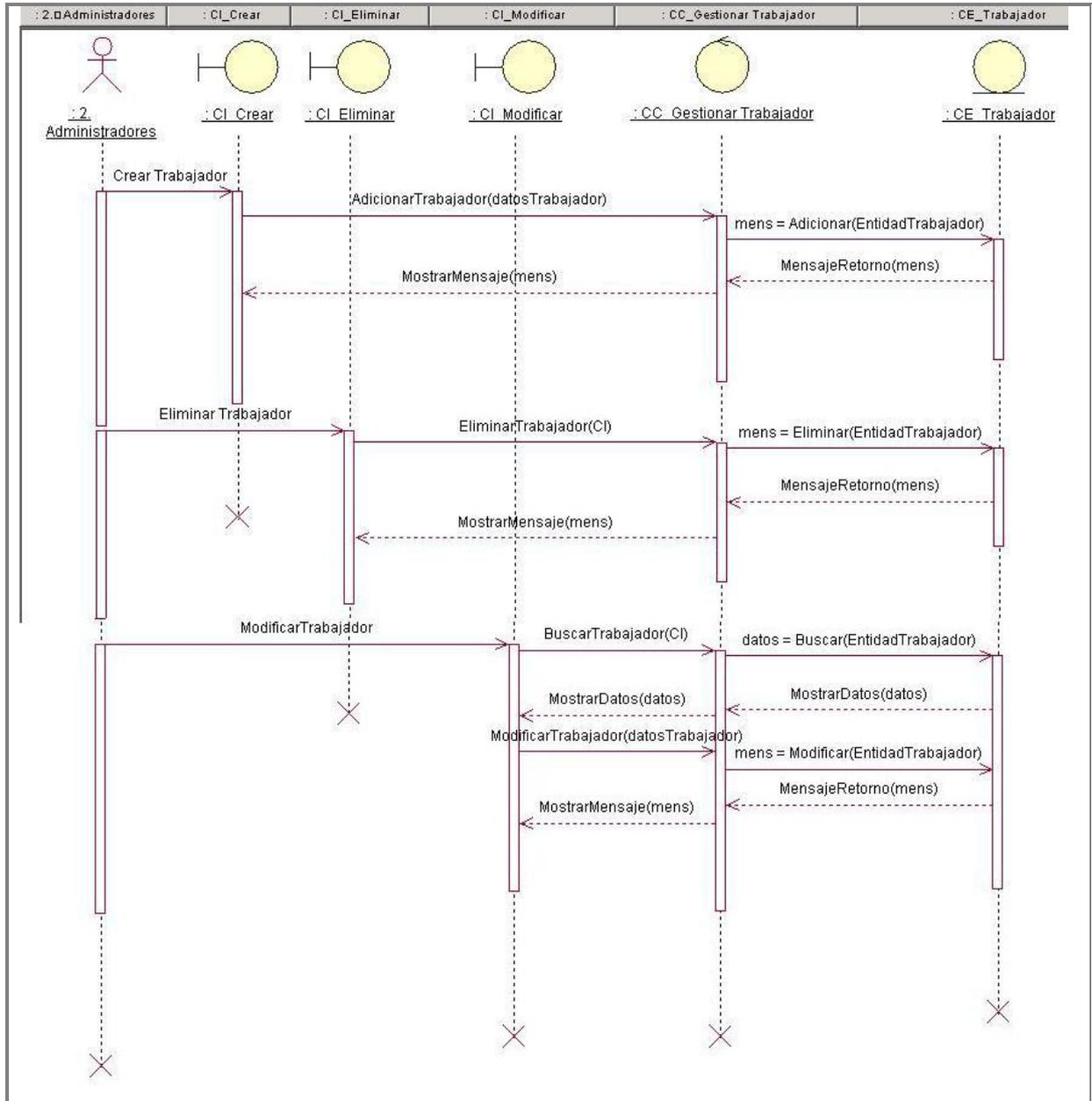


Figura 3.10 – Diagrama de secuencia de Gestionar Trabajador

Diseño del Sistema

Después del desarrollo del modelo de análisis, en el diseño se modelará el Sistema y se encontrará la forma para que soporte todos los requisitos, se puede adquirir una comprensión más profunda de los requisitos no funcionales y permite crear una entrada apropiada y un punto de partida para las actividades de implementación.





3.3 – Diagrama de clases del diseño.

Los diagramas de clases del diseño describen gráficamente las especificaciones de las clases del software y contienen las clases, atributos, métodos, navegabilidad y dependencias existentes entre ellas. En las aplicaciones Web, éste representa la colaboración entre las páginas donde cada página lógica puede ser representada como una clase.

3.3.1 – Diagrama de clases del diseño de Autenticar Usuario

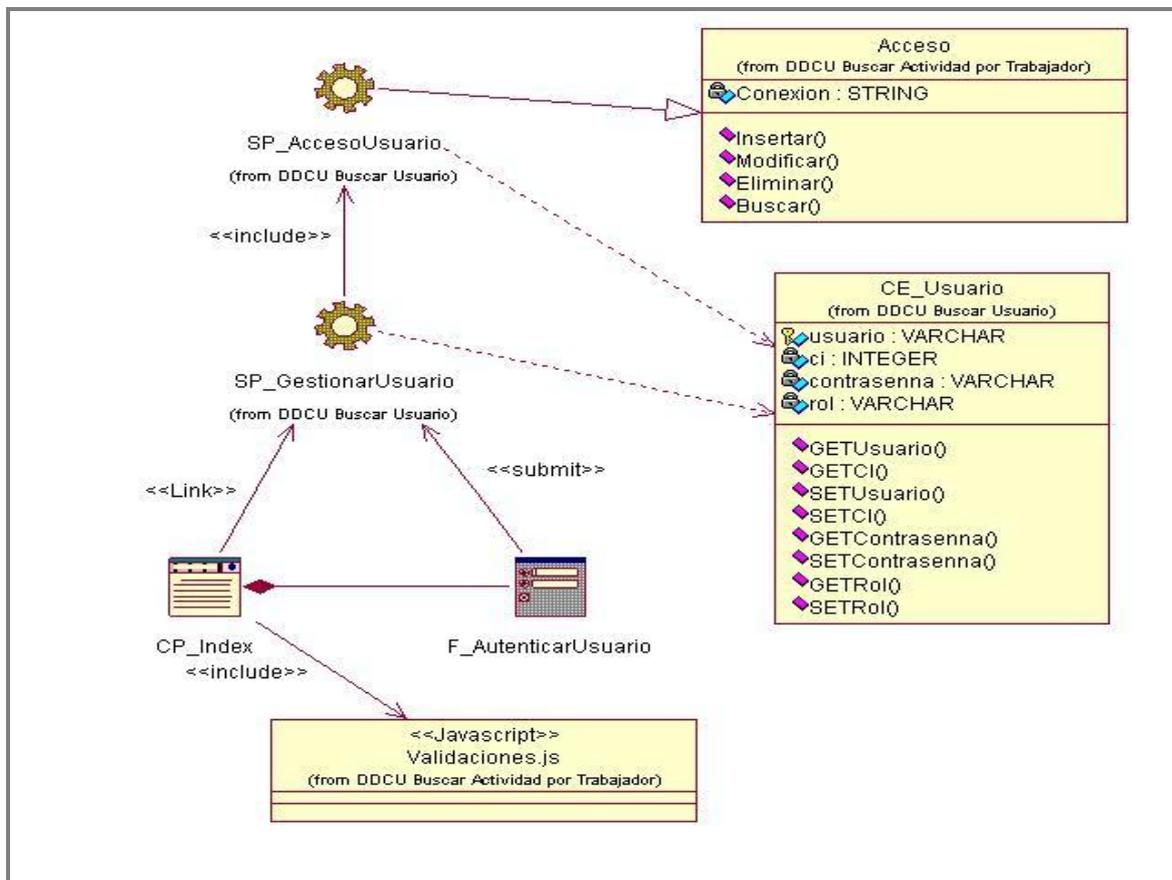


Figura 3.11 – Diagrama de clases del diseño de Autenticar Usuario

3.3.2 – Diagrama de clases del diseño de Listar Trabajador

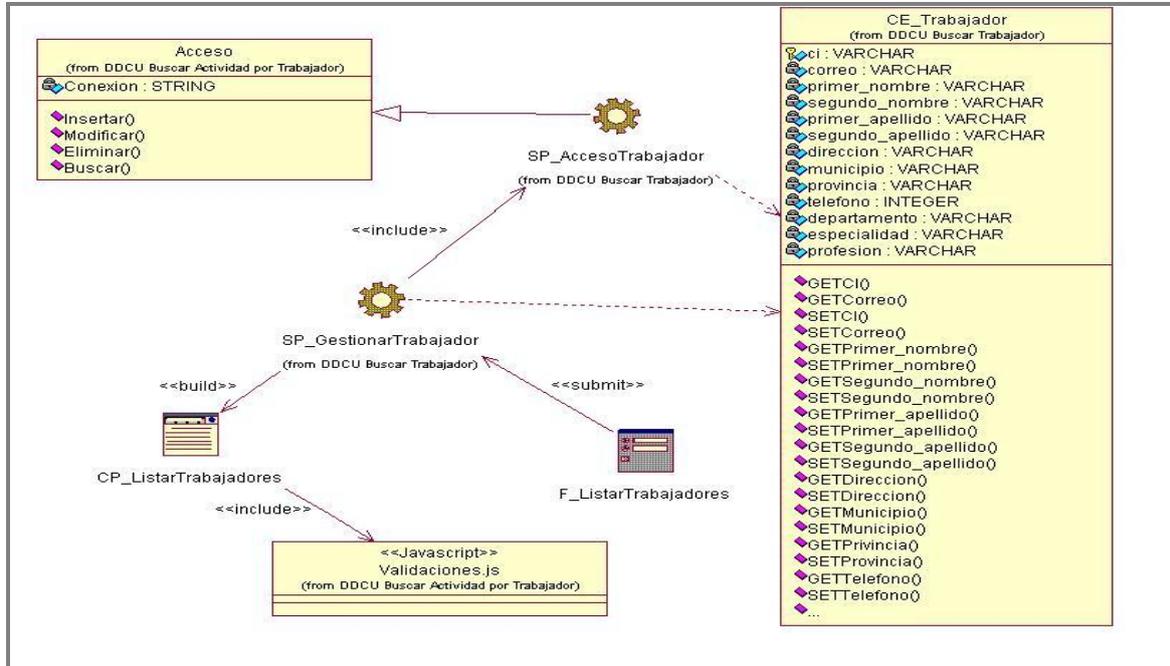


Figura 3.12 – Diagrama de clases del diseño de Listar Trabajador

3.3.3 – Diagrama de clases del diseño de Generar Reportes (Control de Créditos)

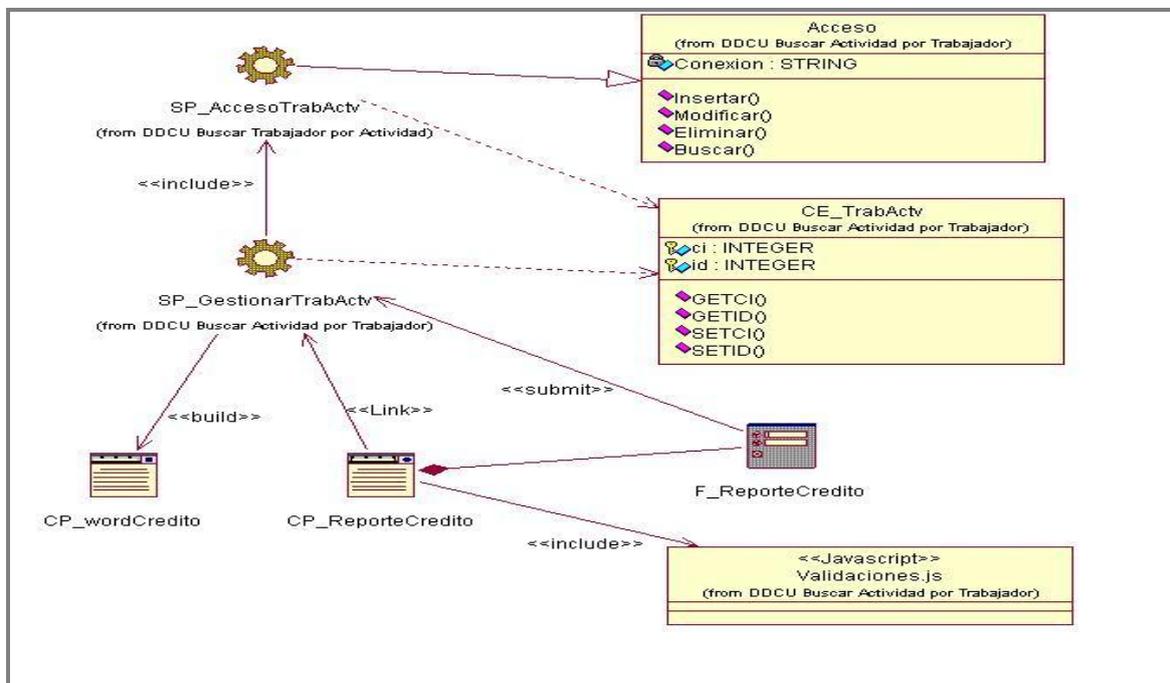


Figura 3.13 – Diagrama de clases del diseño de Generar Reportes (Control de Créditos)



3.3.4 – Diagrama de clases del diseño de Buscar Trabajador

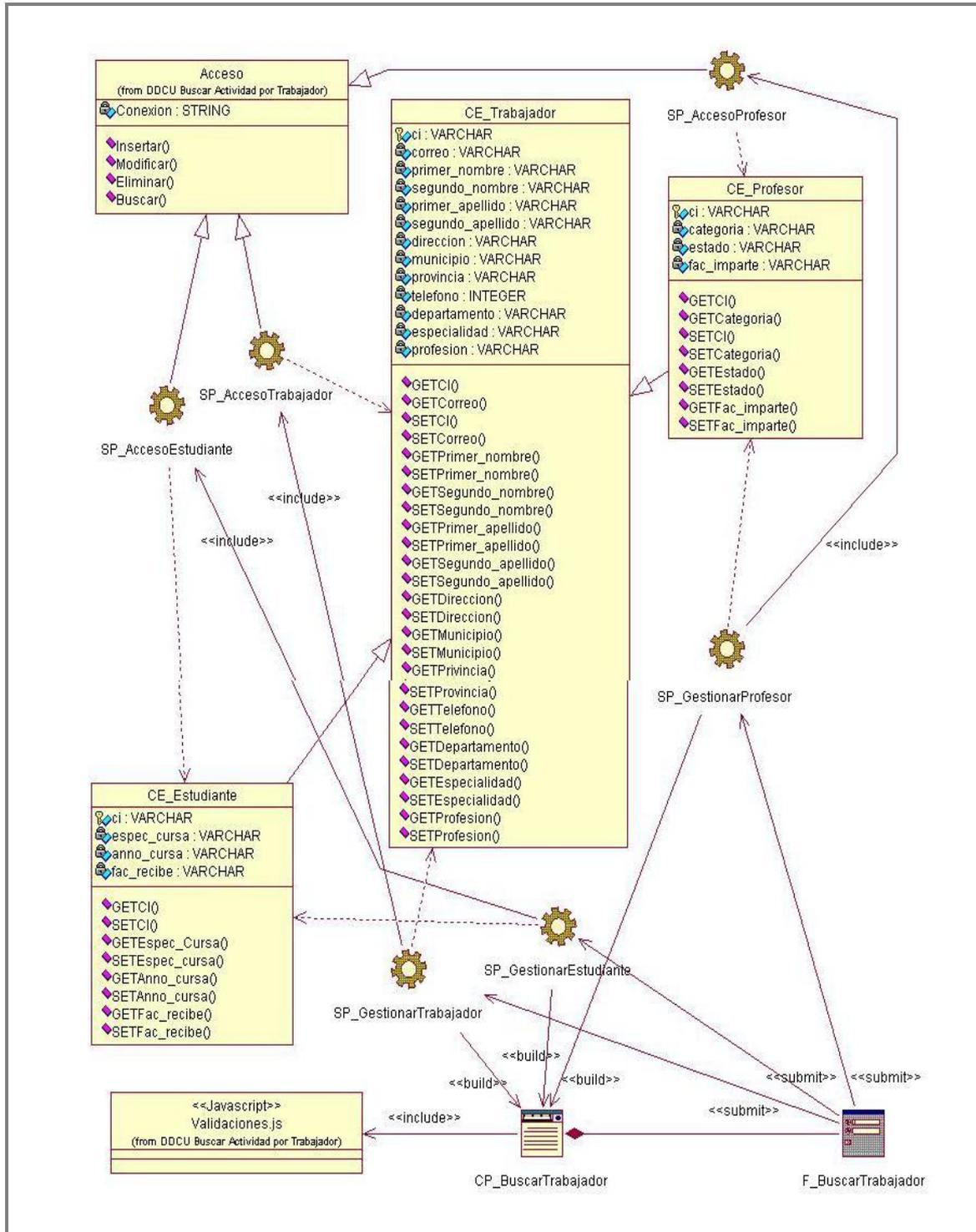


Figura 3.14 – Diagrama de clases del diseño de Buscar Trabajador



3.3.5 – Diagrama de clases del diseño de Gestionar Trabajador

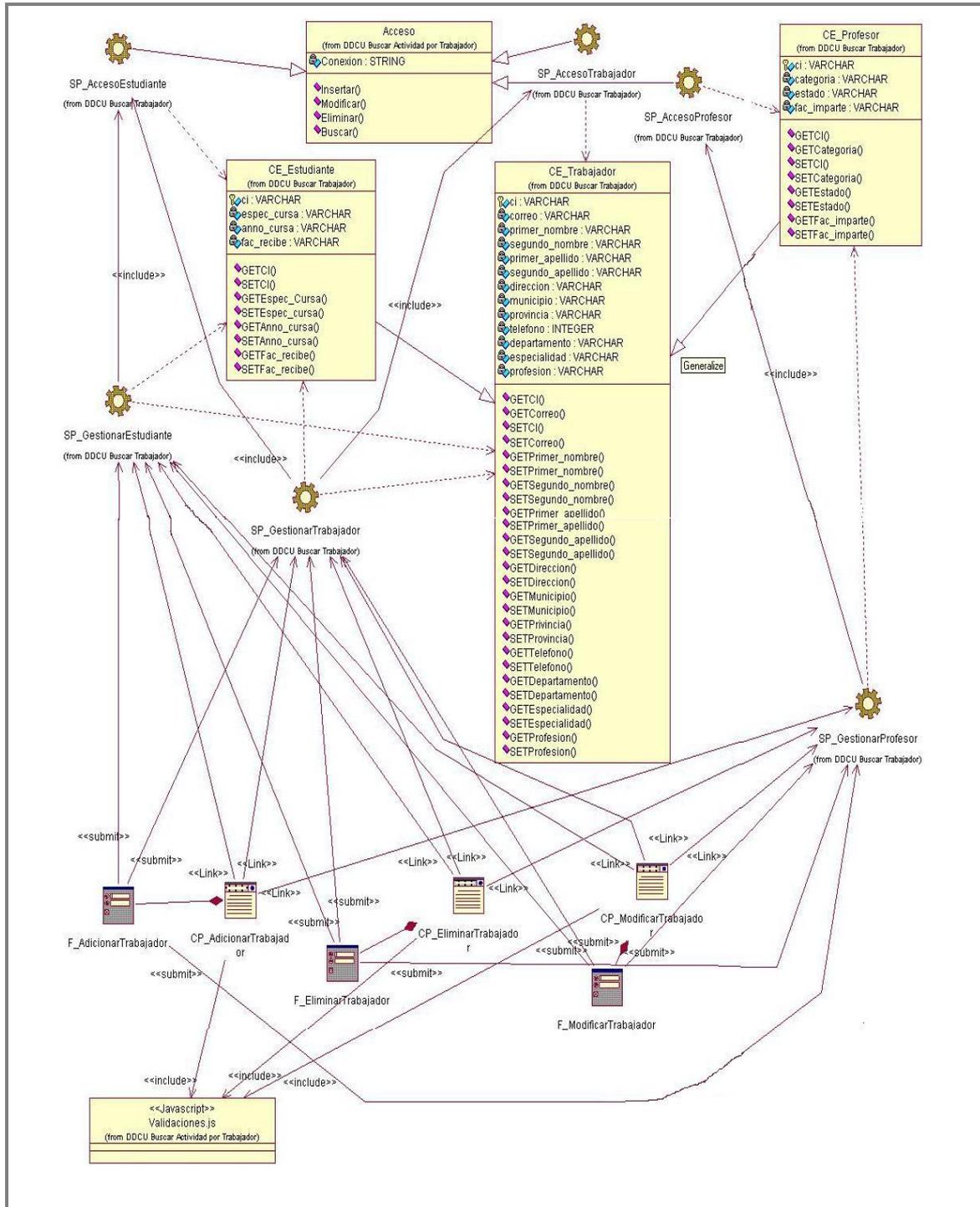


Figura 3.15 – Diagrama de clases del diseño de Gestionar Trabajador

3.4 – Descripción de las clases del diseño.

La Descripción de las clases del diseño se realiza para brindar una panorámica más detallada de los atributos y métodos representados en el Diagrama de clases de diseño, se hace una descripción de cada una por separado.

3.4.1 – Descripción de las clases de Acceso a Datos

Nombre: Acceso	
Tipo de clase : Controladora	
Responsabilidad:	
Nombre:	Insertar(Objeto) Modificar(Objeto) Eliminar(Objeto) Buscar(Objeto)
Descripción:	Es una clase controladora abstracta para heredar de ella la conexión y redefinir los métodos abstractos.

Nombre: AccesoActividad	
Tipo de clase : Controladora	
Responsabilidad:	
Nombre:	Insertar(EntidadActividad) Modificar (EntidadActividad) Eliminar(EntidadActividad) Buscar(EntidadActividad) Listar() ListarXCreditos(ci)
Descripción:	Esta clase hereda la conexión de Acceso y redefine sus métodos.

Nombre: AccesoTrabajador	
Tipo de clase : Controladora	

Responsabilidad:	
Nombre:	Insertar(EntidadTrabajador) Modificar (EntidadTrabajador) Modificar_Trab(EntidadTrabajador,ci_viejo) Eliminar(EntidadTrabajador) Buscar(EntidadTrabajador) BuscarPorNombre (EntidadTrabajador) Listar()
Descripción:	Esta clase hereda la conexión de Acceso y redefine sus métodos.

Nombre: AccesoUsuario	
Tipo de clase : Controladora	
Responsabilidad:	
Nombre:	Insertar(EntidadUsuario) Modificar (EntidadUsuario) Eliminar(EntidadUsuario) Buscar(EntidadUsuario) ModificarUser(EntidadUsuario,usuario_viejo) Listar()
Descripción:	Esta clase hereda la conexión de Acceso y redefine sus métodos.

Nombre: GestionarActividad	
Tipo de clase : Controladora	
Responsabilidad:	
Nombre:	Adicionar(nombre, tipo, fecha, cumplimiento, sede, creditos) ModificarActividad(id, nombre, tipo, fecha, cumplimiento, sede, creditos) BuscarActividad(id) ListarActividad()

	ListarActvXCreditos(ci_trab) EliminarActividad(id)
Descripción:	Esta clase permite gestionar la información referente a las actividades.

Nombre: GestionarTrabajador	
Tipo de clase : Controladora	
Responsabilidad:	
Nombre:	Adicionar(ci,primer_nombre,segundo_nombre,primer_apellido,segundo_apellido,correo,telefono,direccion,municipio,provincia,dpto, especialidad, profesion) ModificarTrabajador(ci,primer_nombre,segundo_nombre,primer_apellido,segundo_apellido,correo,telefono,direccion,municipio,provincia,dpto,especialidad,profesion,ci_viejo) BuscarTrabajador(ci) BuscarTrabajadorPorNombre(pnombre, snombre) ListarTrabajador() EliminarTrabajador(ci)
Descripción:	Esta clase permite gestionar la información referente a los trabajadores

Nombre: GestionarUsuario	
Tipo de clase : Controladora	
Responsabilidad:	
Nombre:	Autenticar(user,pass) Adicionar(user,ci,password,user_rol) ModificarUsuario(usuario,pass,rol,usuario_viejo) BuscarUsuario(user) ListarUsuario() EliminarUsuario(usuario)
Descripción:	Esta clase permite gestionar la información referente a los usuarios.

3.5 Diagrama de clases persistentes.

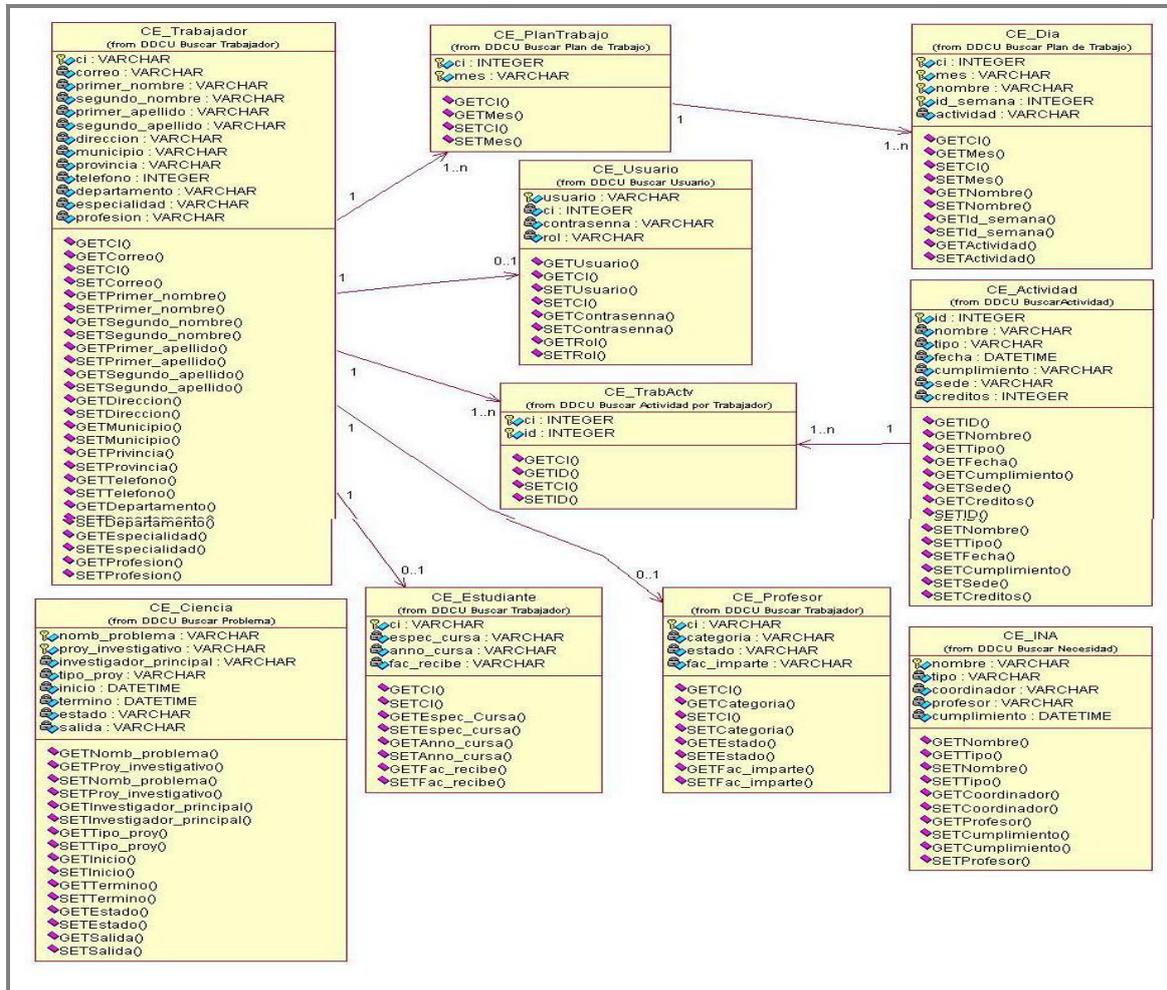


Figura 3.16 – Diagrama de clases persistentes

Conclusiones

En el capítulo se presentó el Modelo de Análisis, considerado la entrada fundamental para las actividades de diseño y éste a su vez se esfuerza en conservar la estructura del Sistema y que sirve como esquema para la implementación.

Al obtener el Modelo de Análisis y de Diseño se describe la realización física de los Casos de usos. Además, se tiene en cuenta como los requisitos funcionales, no funcionales y otras restricciones relacionadas con la implementación, impactan en el Sistema desarrollado. Asimismo los resultados obtenidos, dan entrada a la realización del Flujo de Trabajo de Implementación.

CAPÍTULO 4

IMPLEMENTACIÓN

Introducción

Partiendo de los resultados del diseño, en el presente capítulo se analiza la disciplina de implementación, se presentan el Diagrama de despliegue y el Diagrama de componentes como objetivo primordial de esta fase. Además de que se lleva a cabo el proceso de pruebas siguiendo el método de caja negra.

4.1 – Modelo de Implementación.

El Modelo de Implementación describe cómo los elementos del modelo de diseño y las clases, se implementan en términos de componentes, ficheros de código fuente, ejecutables etc. Los Diagramas de despliegue y componentes conforman lo que se conoce como un Modelo de Implementación al describir los componentes y construir su organización y dependencia entre nodos físicos en los que funcionará la aplicación.

4.2 – Modelo de Despliegue.

El Modelo de Despliegue agrupa los objetos que describen la distribución física del Sistema en términos de funcionalidad entre los nodos. Cada nodo representa un recurso que interviene en el despliegue del Sistema.

Describe la arquitectura física del Sistema durante la ejecución en términos de procesadores, dispositivos y componentes de software. Además, describen la topología del Sistema, es decir, la estructura de los elementos de hardware y software que ejecuta cada uno de ellos.



4.2.1 – Diagrama de despliegue

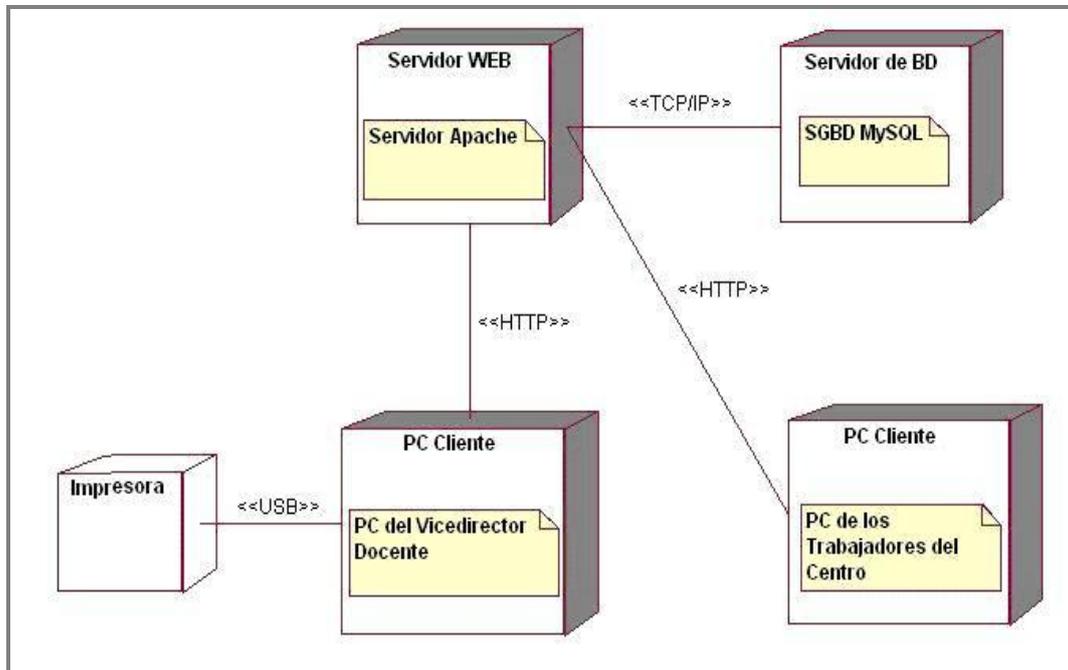


Figura 4.1 – Diagrama de despliegue

4.3 – Diagrama de componentes

Un Diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios o ejecutables. Se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un Diagrama de componentes serán componentes y paquetes.



4.3.1 – Diagrama de componentes

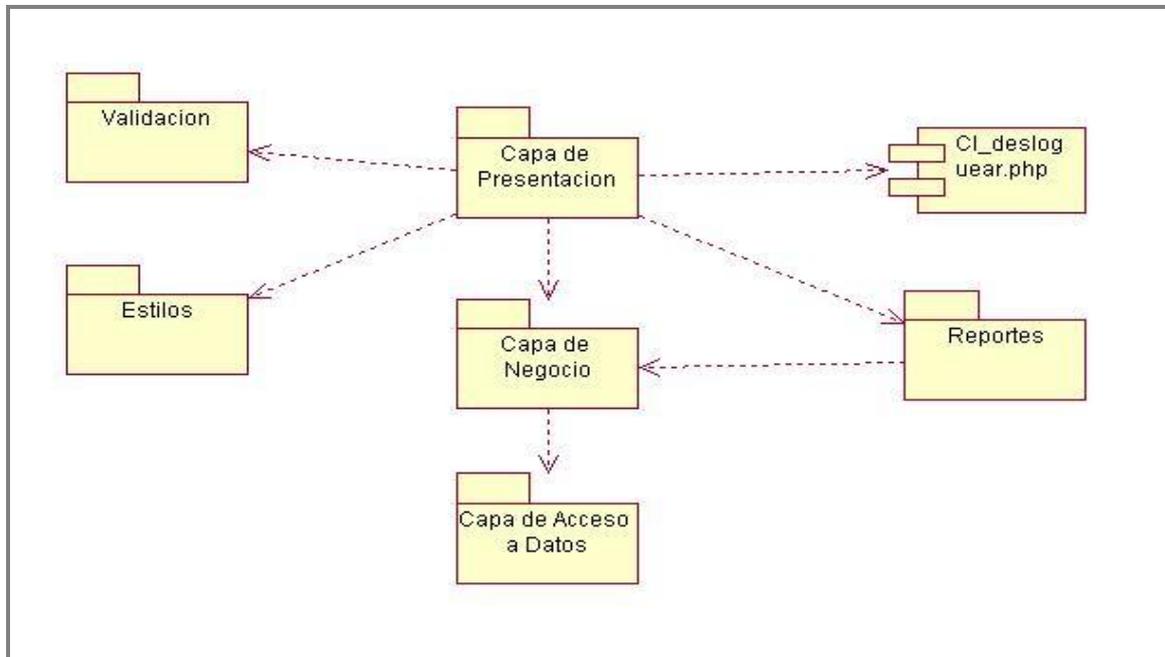


Figura 4.2 – Diagrama de componentes

4.3.2 – Diagrama de componentes del CU Autenticar Usuario

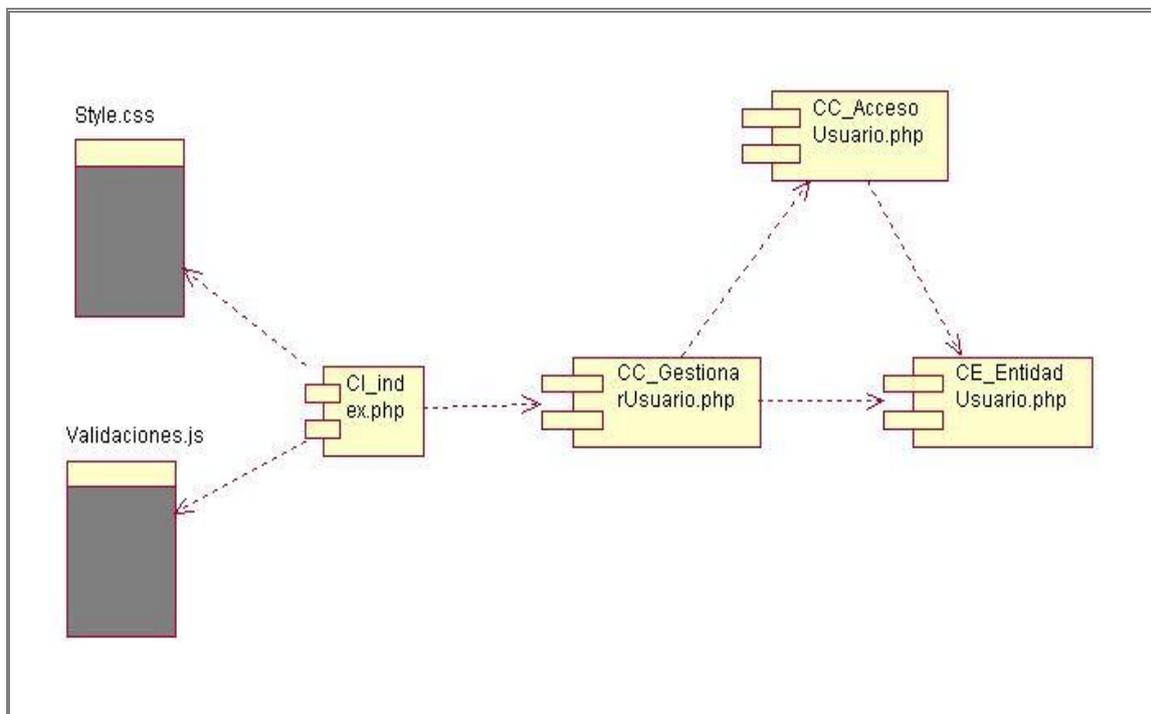


Figura 4.3 – Diagrama de componentes del CU Autenticar Usuario

4.3.3 – Diagrama de componentes del CU Gestionar Trabajador

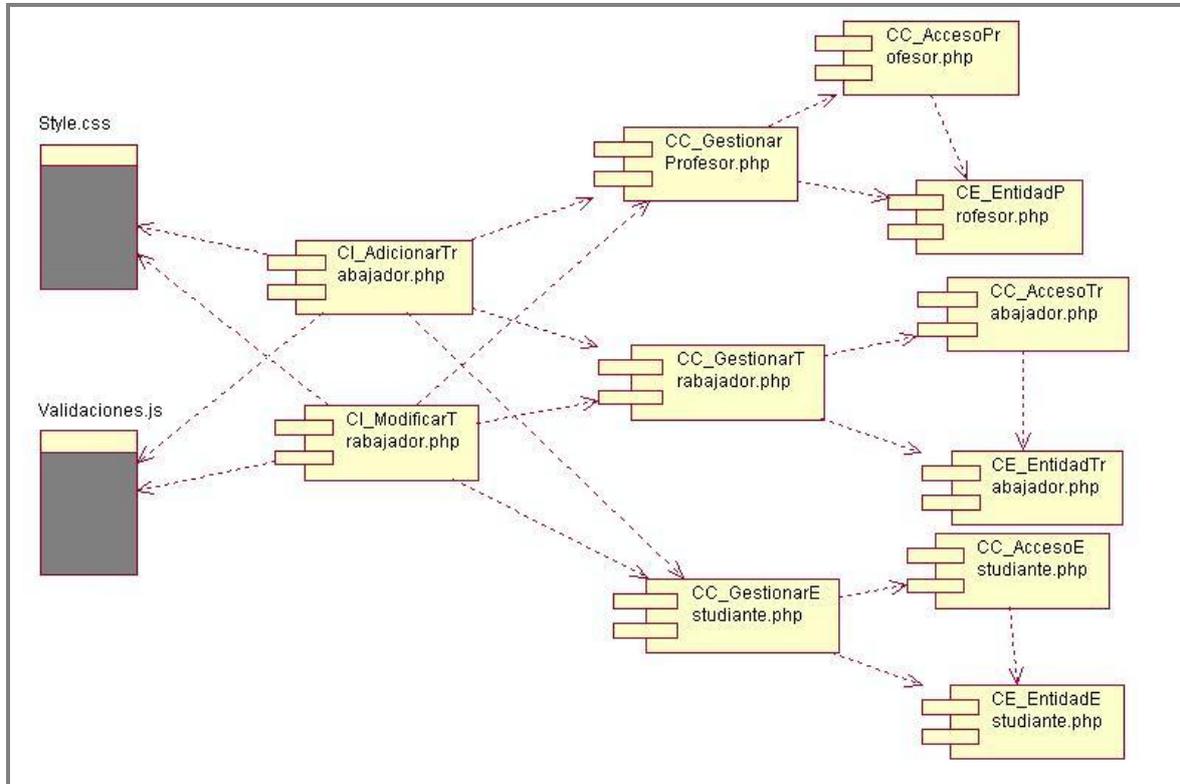


Figura 4.4 – Diagrama de componentes del CU Gestionar Trabajador

4.3.4 – Diagrama de componentes del CU Buscar Trabajador

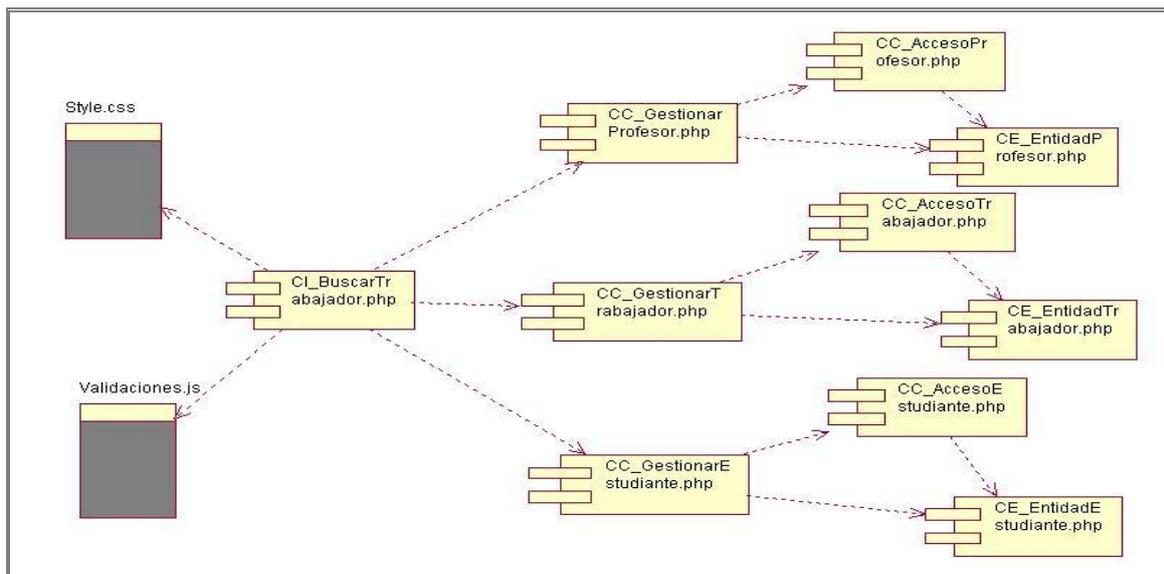


Figura 4.5 – Diagrama de componentes del CU Buscar Trabajador

4.3.5 – Diagrama de componentes del CU Listar Trabajador

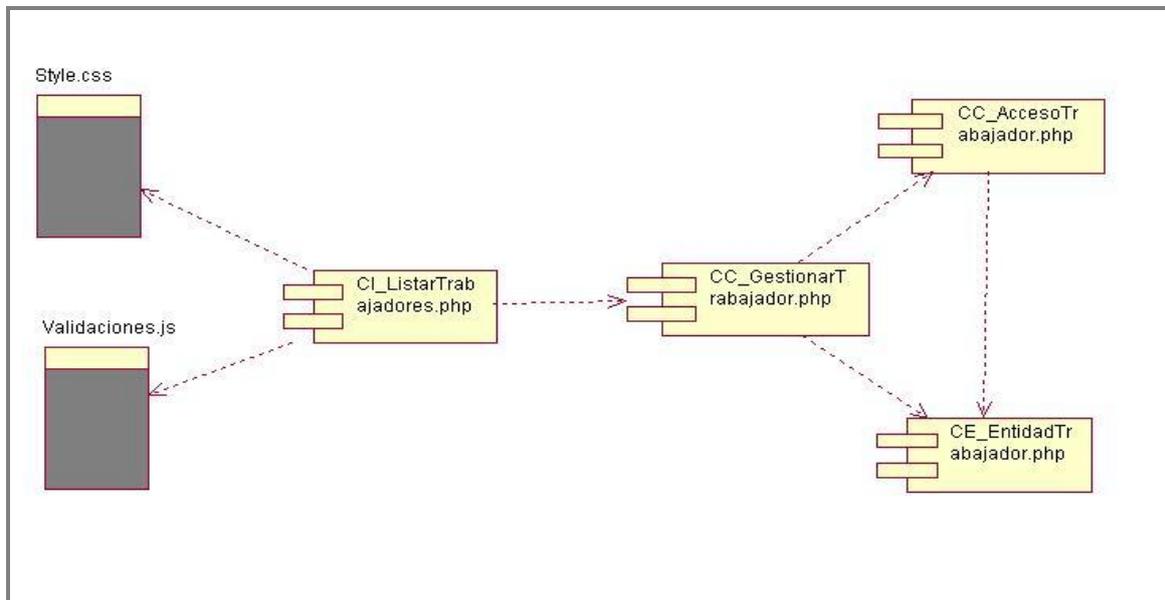


Figura 4.6 – Diagrama de componentes del CU Listar Trabajador

4.3.6 – Diagrama de componentes del CU Generar Reportes (Control de Créditos)

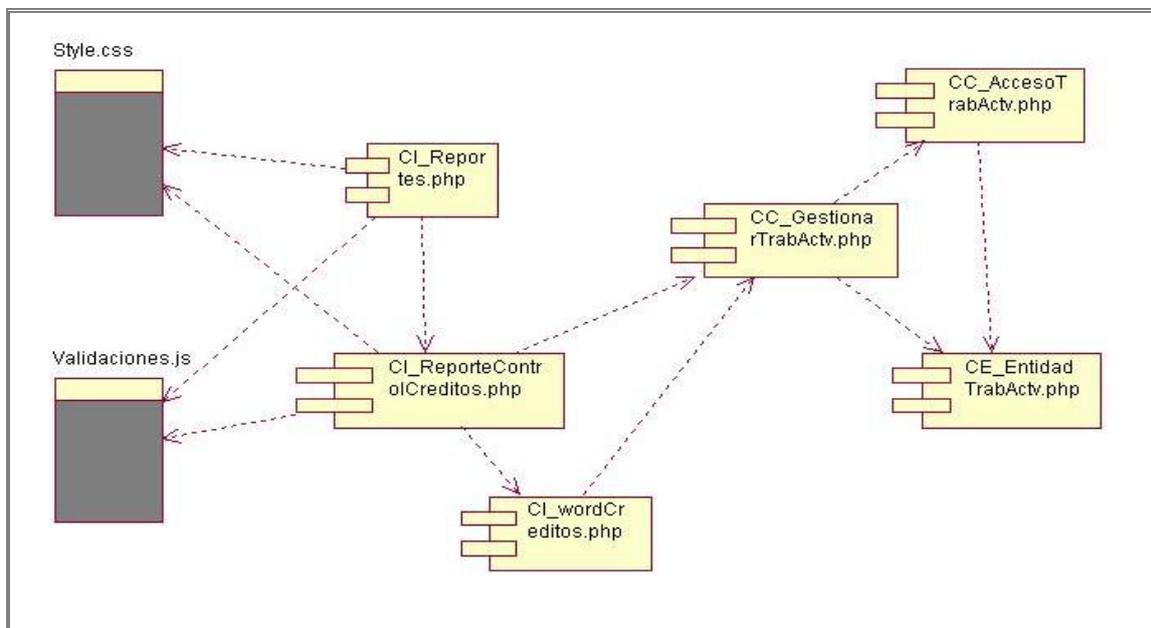


Figura 4.7 – Diagrama de componentes del CU Generar Reportes (Control de Créditos)



Conclusiones

En este capítulo se expuso la fase de Implementación verificando todos los requisitos establecidos por el cliente. A través de dos formas de representación como la del Diagrama de componentes que muestra la organización y la dependencia entre un conjunto de componentes y el Diagrama de despliegue, que son los complementos de los Diagramas de componentes, se provee la vista de implementación del Sistema.

Esta etapa se caracteriza por brindar resultados visibles para el cliente ya que queda implementada la aplicación con sus principales funcionalidades.



Conclusiones

Con la culminación del presente trabajo se puede constatar la importancia de la utilización de las tecnologías de la informática y las comunicaciones para la agilización y optimización de los procesos llevados a cabo en cualquier organización, empresa o institución.

El aporte de este trabajo se encuentra en la mejora y eficacia que se logrará en los procesos realizados en la Vice-dirección de Docencia e Investigaciones a partir del desarrollo del software, cumpliendo así los objetivos trazados. La aplicación presentada, basada en la solución propuesta, satisface las necesidades especificadas en los requisitos funcionales; además de lograr una seguridad y protección al nivel requerido por la Vice-dirección de Docencia e Investigaciones del Policlínico – Hospital Ernesto Che Guevara.



Recomendaciones

- Continuar con el estudio de las nuevas tecnologías y tendencias para mejorar las potencialidades de la aplicación.
- Implementar un nuevo Módulo del Sistema que gestione toda la información generada por las actividades de los estudiantes de medicina que acogerá el Policlínico – Hospital Ernesto Che Guevara.
- Crear Sistemas para la gestión de los procesos en otros departamentos del Policlínico – Hospital.
- Utilizar el Sistema en otras unidades de salud similares al Policlínico – Hospital.

Referencias Bibliográficas

1. Microservos. [En línea] 17 de 03 de 2005. [Citado el: 27 de 01 de 2009.] <http://www.microservos.com/archivo/internet/que-es-internet.html>.
2. Diseño de páginas Web, alojamiento, posicionamiento y dominios en Alicante, España. [En línea] 2009. [Citado el: 27 de 01 de 2009.] <http://www.masadelante.com/faq-internet.htm>.
3. masadelante.com. *Diseño de páginas Web, alojamiento, posicionamiento y dominios en Alicante, España*. [En línea] 2009. [Citado el: 27 de 01 de 2009.] <http://www.masadelante.com/faq-www.htm>.
4. **Pozo, Juan R.** HTML con Clase. [En línea] 2003 - 2008. [Citado el: 27 de 01 de 2009.] <http://html.conclase.net/articulos/navegadores>.
5. **Pressman, Roger S.** *Ingeniería del software. Un enfoque práctico*. La Habana : Felix Varela, 2005.
6. **Bartle, Phil.** Potenciación Comunitaria. *Información para la gestión y Gestión de la Información*. [En línea] [Citado el: 27 de 01 de 2009.] <http://www.scn.org/mpfc/modules/mon-miss.htm>.
7. Introducción al diseño con patrones - El Rincón del Programador. [En línea] 2002-2008. [Citado el: 27 de 01 de 2009.] <http://www.elrincondelprogramador.com/default.asp?pag=articulos/leer.asp&id=29>.
8. **Pablo López, Juan, Pablo Martín, Juan y de la Rosa, Javier.** MorfeoProject, Modelos Avanzados de Comunicación de Recursos. [En línea] 19 de 11 de 2007 . [Citado el: 28 de 01 de 2009.] http://forge.morfeo-project.org/wiki/index.php/D.3.2.2_Modelos_avanzados_de_comunicaci%C3%B3n_de_recursos#Arquitectura_Modelo-Vista-Controlador.
9. **UCI.** Actividad #3. *Historia de la Informática: Introducción al desarrollo del Software*. [En línea] 2008-2009. [Citado el: 28 de 01 de 2009.]
10. Universidad del Valle, Programación Extrema. [En línea] 2009. [Citado el: 28 de 01 de 2009.] <http://eisc.univalle.edu.co/materias/WWW/material/lecturas/xp.pdf>.
11. homepages. *Tecnológico de Monterrey, Campus Monterrey, Programación Extrema*. [En línea] 2008. [Citado el: 28 de 01 de 2009.] <http://homepages.mty.itesm.mx/al1031357/XP.ppt>.
12. Zoho. *Universidad de Palermo: Administración de Proyectos de Software*. [En línea] 2009. [Citado el: 28 de 01 de 2009.] <http://writer.zoho.com/public/27201/38205>.
13. Grupo de Usuarios Rational del Perú – PRUG. [En línea] 24 de 06 de 2006. [Citado el: 28 de 01 de 2009.] <http://prug.solucionesracionales.com/node/12>.

14. **Ferré Grau, Xavier y Sánchez Segura, María Isabel.** Clikear.com. [En línea] 2004. [Citado el: 28 de 01 de 2009.] <http://www.clikear.com/manuales/uml/>.
15. **Torre, Aníbal de la.** Lenguajes del Lado servidor o cliente. [En línea] 2006. [Citado el: 28 de 01 de 2009.] http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html.
16. maestrosdelWeb. [En línea] 11 de 05 de 2005. [Citado el: 29 de 01 de 2009.] <http://www.maestrosdelWeb.com/editorial/flashxml/>.
17. **Merelo Cuervos, Juan Julián.** Introducción al Lenguaje XML. [En línea] 2004. [Citado el: 29 de 01 de 2009.] <http://geneura.ugr.es/~jmerelo/xml/>.
18. **Garrett, Jesse James.** Ajax: un nuevo acercamiento a las aplicaciones Web. [En línea] 18 de 02 de 2005. [Citado el: 29 de 01 de 2009.] <http://www.uberbin.net/archivos/internet/ajax-un-nuevo-acercamiento-a-aplicaciones-Web.php>.
19. **Rodas Hinostroza, Raúl.** LinuxCentro.net. [En línea] 22 de 02 de 2007. [Citado el: 29 de 01 de 2009.] <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>.
20. **Ángel Álvarez, Miguel.** desarrolloWeb.com. *Que es ASP.* [En línea] 09 de 05 de 01. [Citado el: 29 de 01 de 2009.] <http://www.desarrolloWeb.com/articulos/393.php>.
21. masadelante.com. *Diseño de páginas Web, alojamiento, posicionamiento y dominios en Alicante, España.* [En línea] 2009. [Citado el: 30 de 01 de 2009.] <http://www.masadelante.com/faq-servidor-Web.htm>.
22. ciberaula. [En línea] 2006. [Citado el: 30 de 01 de 2009.] http://linux.ciberaula.com/articulo/linux_apache_intro/.
23. Internet INformation Server. [En línea] 11 de 07 de 2007. [Citado el: 30 de 01 de 2009.] <http://www.microsoft.com/spain/windowsserver2003/technologies/webapp/iis.msp>.
24. desarrolloWeb.com. [En línea] 31 de 07 de 2007. [Citado el: 30 de 01 de 2009.] <http://www.desarrolloWeb.com/articulos/Sistemas-gestores-bases-datos.html>.
25. MySQL. *Universidad Autónoma del estado de Morelos.* [En línea] [Citado el: 30 de 01 de 2009.] <http://www.uaem.mx/posgrado/mcruz/cursos/miic/MySQL.pdf>.
26. eAprende.com. [En línea] [Citado el: 30 de 01 de 2009.] <http://www.eaprende.com/gestor-de-basededatos-mysql-postresql-sqlite.html>.
27. desarrolloWeb.com. [En línea] 19 de 07 de 2002. [Citado el: 30 de 01 de 2009.] <http://www.desarrolloWeb.com/articulos/840.php>.
28. GSInnova. [En línea] 2007. [Citado el: 30 de 01 de 2009.] <http://www.rational.com.ar/herramientas/roseenterprise.html>.

29. Free Download Manager. [En línea] 20 de 06 de 2006. [Citado el: 30 de 01 de 2009.]
http://www.freedownloadmanager.org/es/downloads/SME_Gerente_de_MySQL_37542_p/.
30. aulaClic. *Conceptos Básicos de Dreamweaver 8*. [En línea] 08 de 2006. [Citado el: 30 de 01 de 2009.] http://www.aulaclic.es/dreamweaver8/t_1_1.htm.
31. Taringa! *Inteligencia Colectiva*. [En línea] [Citado el: 30 de 01 de 2009.]
http://www.taringa.net/posts/downloads/1166221/ER-Studio-7_1---Modelado-de-Bases-de-Datos-E_R.html.
32. BureaudePrensa.com. [En línea] 24 de 07 de 2006. [Citado el: 30 de 01 de 2009.]
http://bureaudeprensa.com/es/view.php?bn=bureaudeprensa_software&key=1153755965
33. desarrolloWeb.com. [En línea] 16 de 11 de 2007. [Citado el: 30 de 01 de 2009.]
<http://www.desarrolloWeb.com/articulos/aptana-studio.html>.
34. desarrolloWeb.com. [En línea] 04 de 06 de 2003. [Citado el: 30 de 01 de 2009.]
<http://www.desarrolloWeb.com/articulos/1178.php>

Bibliografía

1. **Alonso, E. M.** Tutorial de HTML. [En línea] [Citado el: 28 de 01 de 2009.] http://gias720.dis.ulpgc.es/Gias/Cursos/Tutorial_html/indice.htm.
2. **Arenas, M. I.** GENEURA. [En línea] [Citado el: 10 de 02 de 2009.] <http://geneura.ugr.es/~maribel/php/>.
3. **Castro, I. A.** TutorialPHP.net. [En línea] 2007. [Citado el: 25 de 01 de 2009.] <http://tutorialphp.net/>.
4. **García, J. J.** Tutorial HTML. [En línea] 06 de 06 de 1996. [Citado el: 21 de 01 de 2009.] <http://www.um.es/psibm/tutorial/>.
5. **Ivar Jacobson, G. B.** *El Proceso Unificado de Desarrollo*.
6. MySQL. [En línea] [Citado el: 10 de 02 de 2009.] <http://dev.mysql.com/doc/refman/5.0/es/index.html>.
7. MySQL. [En línea] [Citado el: 10 de 02 de 2009.] <http://dev.mysql.com/doc/refman/5.0/es/tutorial.html>.
8. **Pozo, J. R.** HTML con clase. [En línea] 04 de 08 de 2003. [Citado el: 04 de 02 de 2009.] <http://html.conclase.net/tutorial/html/>.
9. Tutorial de Javascript. [En línea] [Citado el: 20 de 01 de 2009.] <http://www.lcc.uma.es/~eat/services/html-js/manual13.html>.
10. **Valente, M.** HTMLPOINT.com. [En línea] [Citado el: 15 de 01 de 2009.] <http://www.htmlpoint.com/javascript/tutorial/>.
11. WebEstilo. [En línea] [Citado el: 20 de 01 de 2009.] <http://www.webestilo.com/javascript/>.

Anexos

1 – Descripciones de los Casos de uso del Sistema

1.1 – Descripción del Caso de uso del Sistema Gestionar Usuario.

Nombre del CU	Gestionar Usuario	
Actores	Vice-director de Docencia e Investigaciones, Informático.	
Propósito	Crear, modificar o eliminar un usuario.	
Resumen	El Caso de uso se inicia cuando el Vice-director de Docencia e Investigaciones y/o el Informático solicitan la creación, modificación o eliminación de un usuario.	
Referencias	RF2	
Precondiciones	Que el Vice-director de Docencia e Investigaciones y/o el Informático esté autenticado en el Sistema.	
Prioridad	Crítico	
Sección “Crear Usuario”: Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Vice-director de Docencia e Investigaciones y/o el Informático selecciona la opción para crear un nuevo usuario.	2. El Sistema muestra la interfaz para insertar los datos del nuevo usuario.	
3. El Vice-director de Docencia e Investigaciones y/o el Informático insertan los datos del usuario.	4. El Sistema verifica que no existan campos vacíos. 5. El Sistema registra los datos.	
Sección “Crear Usuario”: Flujo Alternativo		
1. Si el Vice-director de Docencia e Investigaciones y/o el Informático no desean realizar otra acción, termina el CU.	2. Si quedó algún campo vacío, el Sistema envía un mensaje.	
Sección “Modificar Usuario”: Flujo Normal de Eventos		

Acciones del Actor	Respuesta del Sistema
1. El Vice-director de Docencia e Investigaciones y/o el Informático selecciona la opción modificar usuario.	2. El Sistema muestra una interfaz para modificar los campos deseados.
3. El Vice-director de Docencia e Investigaciones y/o el Informático modifican los campos deseados.	4. El Sistema verifica que los campos obligatorios no estén vacíos. 5. El Sistema registra y actualiza los nuevos datos.
Sección “Modificar Usuario”: Flujo Alternativo	
1. Si el Vice-director de Docencia e Investigaciones y/o el Informático no desea realizar otra acción, termina el CU.	2. Si quedó algún campo obligatorio vacío, el Sistema envía un mensaje.
Sección “Eliminar Usuario”: Flujo Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Vice-director de Docencia e Investigaciones y/o el Informático selecciona la opción eliminar usuario que se encontrará en forma de X al lado del nombre y los datos de el usuario. 3. El actor confirma eliminar el usuario.	2. El Sistema realiza una confirmación de si se desea eliminar el usuario. 4. El Sistema elimina el usuario.
Sección “Eliminar Usuario”: Flujo Alternativo	
1. Si el Administrador no desea realizar otra acción, termina el CU.	
Poscondiciones	Los datos insertados o modificados deben actualizarse en la Base de datos.

1.2 – Descripción del Caso de uso del Sistema Buscar Trabajador por Actividad

Nombre del CU	Buscar Trabajador por Actividad
Actores	Vice-director de Docencia e Investigaciones, Informático, trabajadores, Jefes de Departamentos.
Propósito	Buscar Trabajador por Actividad.
Resumen	El Caso de uso lo inicia el Vice-director de Docencia e Investigaciones, Informático, trabajadores, Jefes de

	Departamentos, cuando solicitan la búsqueda algún trabajador por actividad.
Referencias	RF9
Precondiciones	Que el Vice-director de Docencia e Investigaciones, Informático, trabajadores, Jefes de Departamentos estén autenticados en el Sistema.
Prioridad	Crítico
Sección “Buscar Trabajador por Actividad”: Flujo Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1. El usuario selecciona la opción buscar Trabajador por Actividad.	2. El Sistema muestra una interfaz para buscar el trabajador.
3. El usuario inserta el ID de la actividad.	4. El Sistema muestra los trabajadores que cursan en dicha actividad.
Sección “Buscar Trabajador por Actividad”: Flujo Alternativo	
1. Si el usuario no desea realizar otra acción, termina el CU.	2. Si quedó algún campo obligatorio vacío, el Sistema envía un mensaje. 3. Si la actividad no existe el Sistema muestra un mensaje.
Poscondiciones	Los datos creados o modificados deben actualizarse en la Base de datos.

1.3 – Descripción del Caso de uso del Sistema Listar Actividad

Nombre del CU	Listar Actividad
Actores	Vice-director de Docencia e Investigaciones, Informático, trabajadores, Jefes de Departamentos.
Propósito	Listar actividad.
Resumen	El Caso de uso lo inicia cuando el Vice-director de Docencia e Investigaciones, Informático, trabajadores, Jefes de Departamentos solicitan la opción Listar actividad.
Referencias	RF13

Precondiciones	Que el Vice-director de Docencia e Investigaciones, Informático, trabajadores, Jefes de Departamentos, estén autenticados en el Sistema.	
Prioridad	Crítico	
Sección “Listar Actividad”: Flujo Normal de Eventos		
Acciones del Actor		Respuesta del Sistema
1. El usuario selecciona la opción listar actividad.		2. El Sistema muestra una interfaz con el listado de las actividades.
Sección “Listar Actividad”: Flujo Alternativo		
1. Si el usuario no desea realizar otra acción, termina el CU.		2. El Vice-director de Docencia e Investigaciones y/o el Informático pueden eliminar o modificar las actividades que deseen.
Poscondiciones	Los datos creados o modificados deben actualizarse en la Base de datos.	

1.4 – Descripción del Caso de uso del Sistema Buscar Problema

Nombre del CU	Buscar Problema
Actores	Vice-director de Docencia e Investigaciones, Informático, trabajadores, Jefes de Departamentos.
Propósito	Buscar problema.
Resumen	El Caso de uso lo inicia el Vice-director de Docencia e Investigaciones, Informático, trabajadores, Jefes de Departamentos, cuando solicitan la opción buscar problema.
Referencias	RF18
Precondiciones	Que el Vice-director de Docencia e Investigaciones, Informático, trabajadores, Jefes de Departamentos estén autenticados en el Sistema.
Prioridad	Crítico
Sección “Buscar Problema”: Flujo Normal de Eventos	

Acciones del Actor	Respuesta del Sistema
1. El usuario selecciona la opción buscar problema.	2. El Sistema muestra una interfaz para buscar problema.
3. El usuario inserta el nombre del problema y el proyecto investigativo.	4. El Sistema muestra los datos del problema.
Sección “Buscar Problema”: Flujo Alternativo	
1. Si el usuario no desea realizar otra acción, termina el CU.	2. Si quedó algún campo obligatorio vacío, el Sistema envía un mensaje. 3. Si el problema no existe el Sistema muestra un mensaje.
Poscondiciones	Los datos creados o modificados deben actualizarse en la Base de datos.

1.5 – Descripción del Caso de uso del Sistema Gestionar Plan de Trabajo

Nombre del CU	Gestionar Plan de Trabajo
Actores	Vice-director de Docencia e Investigaciones, Informático, Jefes de Departamentos y trabajadores.
Propósito	Crear, modificar o eliminar un plan.
Resumen	El Caso de uso se inicia cuando el Vice-director de Docencia e Investigaciones y/o el Informático solicitan la creación, modificación o eliminación de un plan.
Referencias	RF20
Precondiciones	Que el Vice-director de Docencia e Investigaciones, Informático, Jefes de Departamentos y trabajadores estén autenticados en el Sistema.
Prioridad	Crítico
Sección “Crear Plan de Trabajo”: Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Vice-director de Docencia e Investigaciones, Informático, Jefe de Departamento o trabajadores selecciona	2. El Sistema muestra la interfaz para insertar los datos del nuevo plan.

la opción para crear un nuevo plan.	
3. El Vice-director de Docencia e Investigaciones, Informático, Jefe de Departamento o trabajadores insertan los datos del plan.	4. El Sistema verifica que no existan campos obligatorios vacíos. 5. El Sistema registra los datos.
Sección “Crear Plan de Trabajo”: Flujo Alternativo	
1. Si el Vice-director de Docencia e Investigaciones, Informático, Jefes de Departamentos o trabajadores no desean realizar otra acción, termina el CU.	2. Si quedó algún campo obligatorio vacío, el Sistema envía un mensaje.
Sección “Modificar Plan de Trabajo”: Flujo Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Vice-director de Docencia e Investigaciones, Informático, Jefes de Departamentos o trabajadores selecciona la opción modificar plan.	2. El Sistema muestra una interfaz para modificar los campos deseados.
3. El Vice-director de Docencia e Investigaciones, Informático, Jefes de Departamentos o trabajadores modifican los campos deseados.	4. El Sistema registra y actualiza los nuevos datos.
Sección “Modificar Plan de Trabajo”: Flujo Alternativo	
1. Si el Vice-director de Docencia e Investigaciones, Informático, Jefes de Departamentos o trabajadores no desea realizar otra acción, termina el CU.	
Sección “Eliminar Plan de Trabajo”: Flujo Normal de Eventos	
Acciones del Actor	Respuesta del Sistema
1. El Vice-director de Docencia e Investigaciones y el Informático selecciona la opción eliminar plan que se encontrará en forma de X al lado del nombre y los datos del plan.	2. El Sistema muestra una confirmación de si se desea eliminar o no el plan.
3. El actor confirma que desea eliminar el plan.	4. El Sistema elimina el plan.
Sección “Eliminar Plan de Trabajo”: Flujo Alternativo	



1. Si el Administrador no desea realizar otra acción, termina el CU.	
Poscondiciones	Los datos insertados o modificados deben actualizarse en la Base de datos.

2 – Diagramas de clases de análisis

2.1 – Diagrama de clases de análisis de Buscar Necesidad

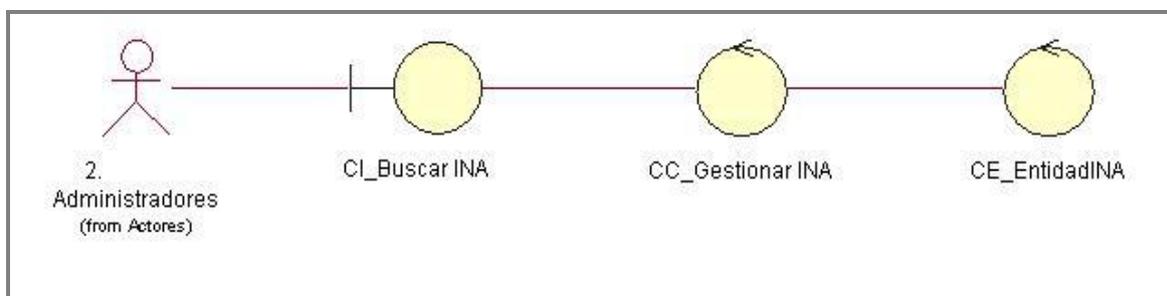


Figura 2.1– Diagrama de clases de análisis de Buscar Necesidad

2.2 – Diagrama de clases de análisis de Listar Plan de Trabajo

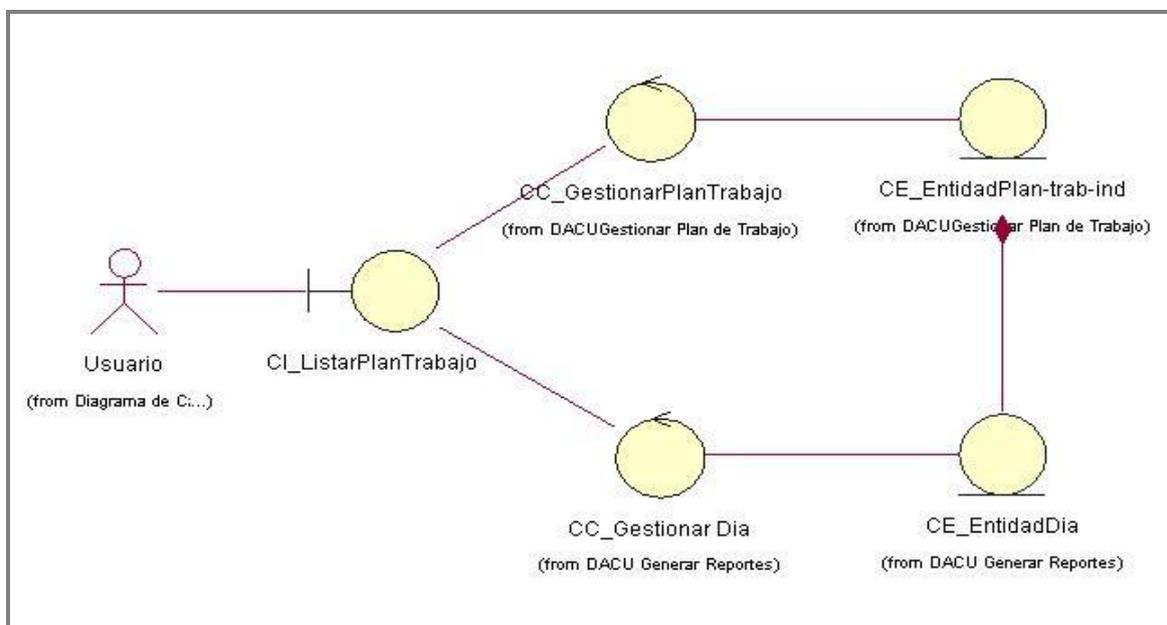


Figura 2.2– Diagrama de clases de análisis de Listar Plan de Trabajo



2.3 – Diagrama de clases de análisis de Gestionar Usuario

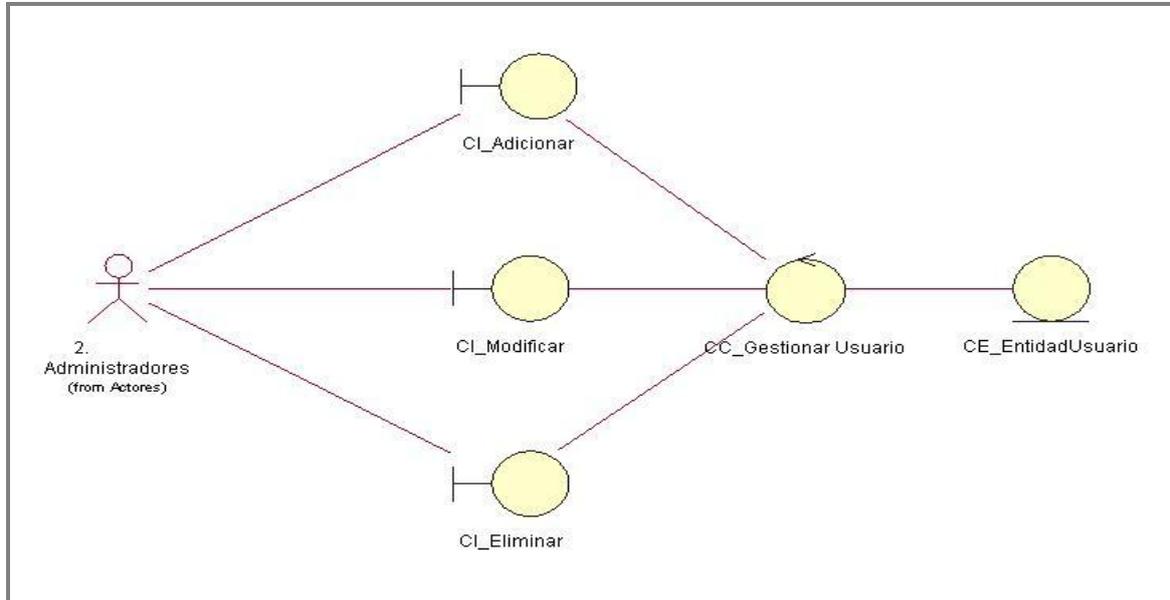


Figura 2.3– Diagrama de clases de análisis de Gestionar Usuario

2.4 – Diagrama de clases de análisis de Listar Actividad

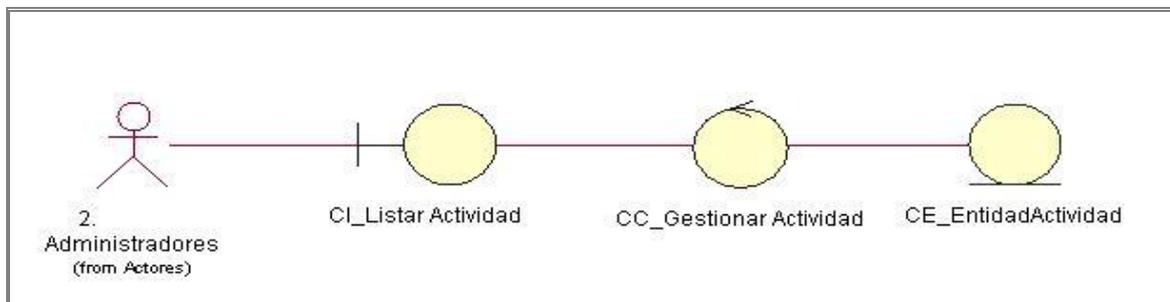


Figura 2.4– Diagrama de clases de análisis de Listar Actividad

3 – Diagramas de Clases del Diseño

3.1 – Diagrama de Clases del Diseño de Buscar Actividad por Trabajador

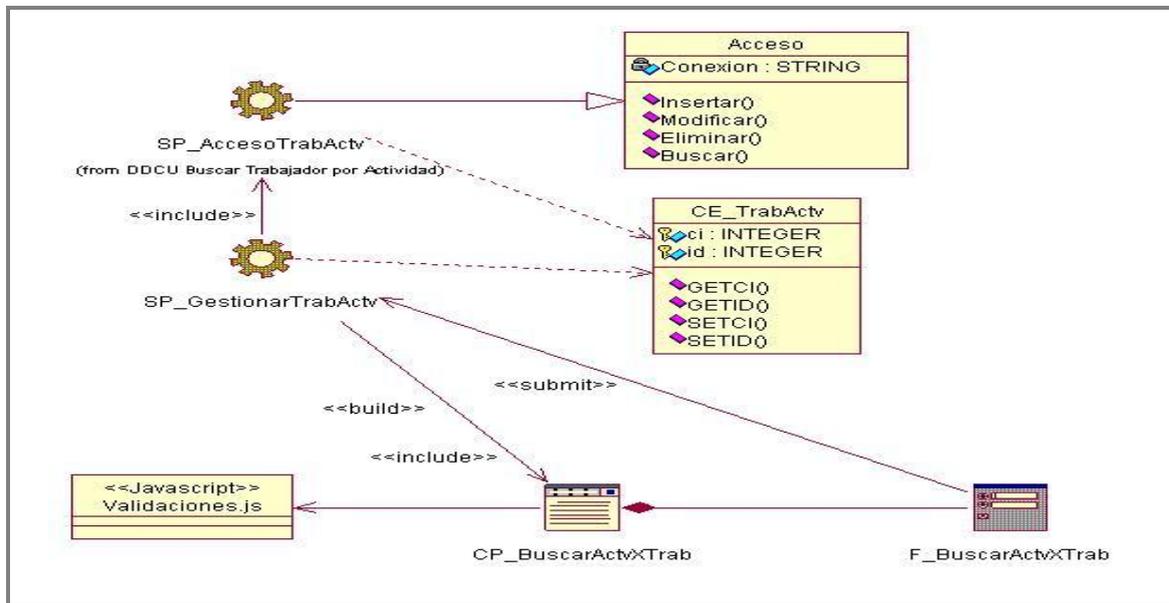


Figura 3.1– Diagrama de Clases del Diseño de Buscar Actividad por Trabajador

3.2 – Diagrama de Clases del Diseño de Gestionar INA

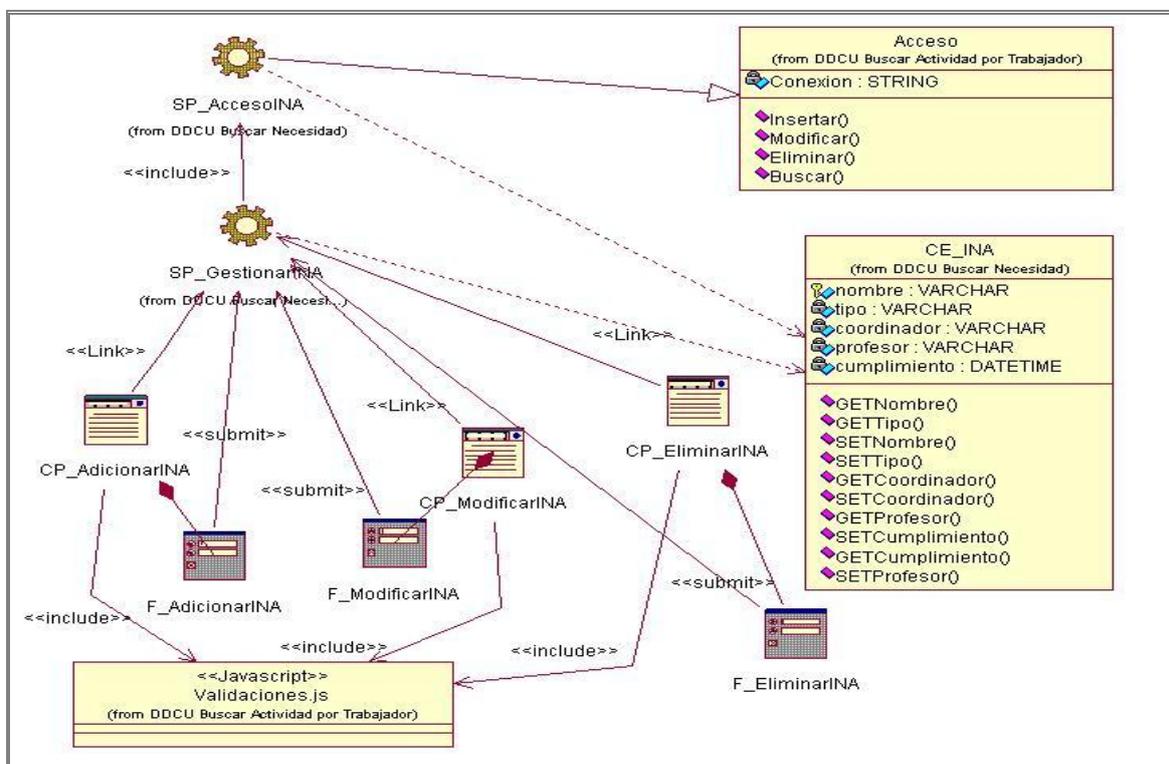


Figura 3.2– Diagrama de Clases del Diseño de Gestionar INA



3.3 – Diagrama de Clases del Diseño de Gestionar Plan de Trabajo

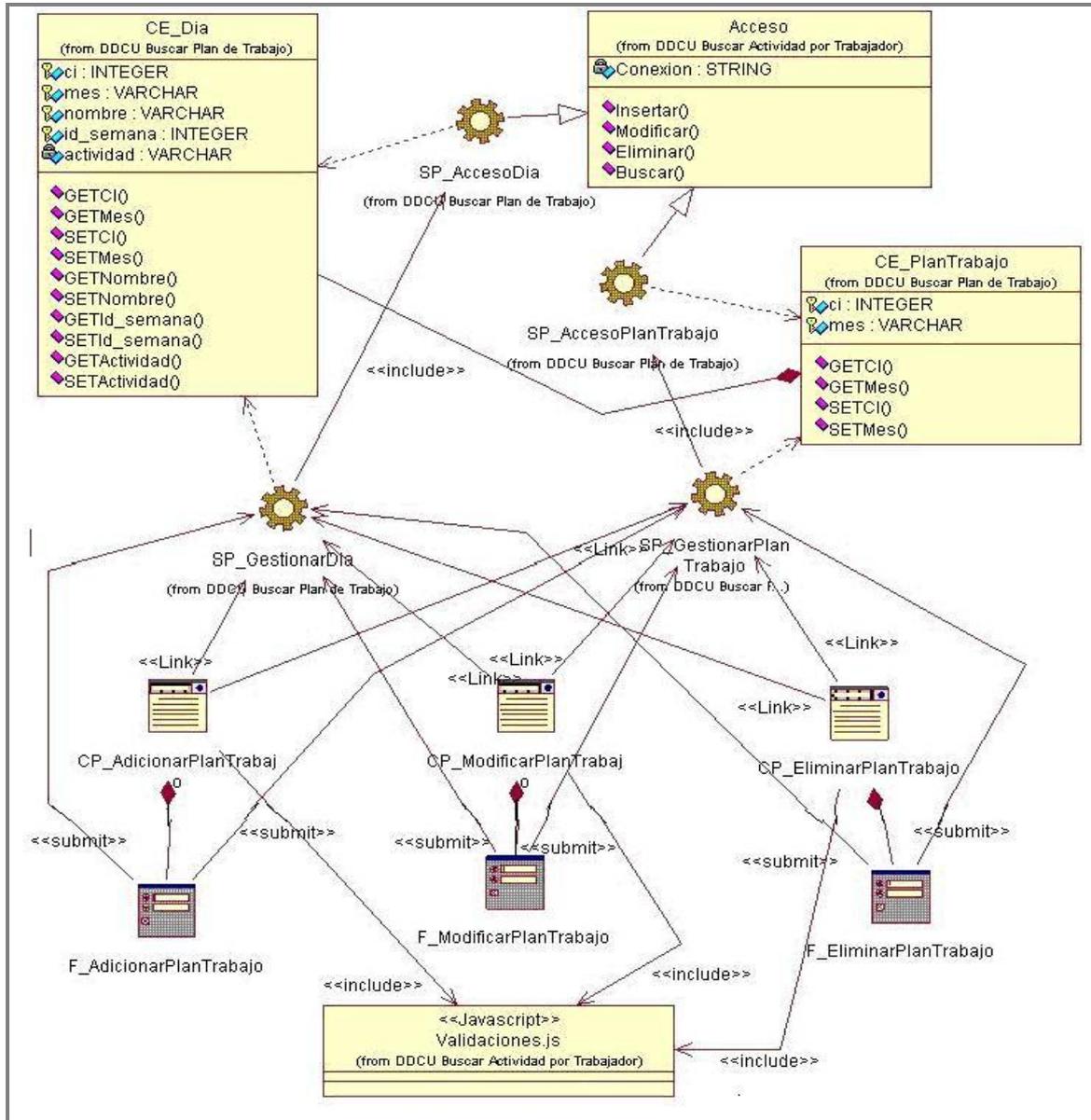


Figura 3.3– Diagrama de Clases del Diseño de Gestionar Plan de Trabajo

3.4 – Diagrama de Clases del Diseño de Listar Plan de Trabajo

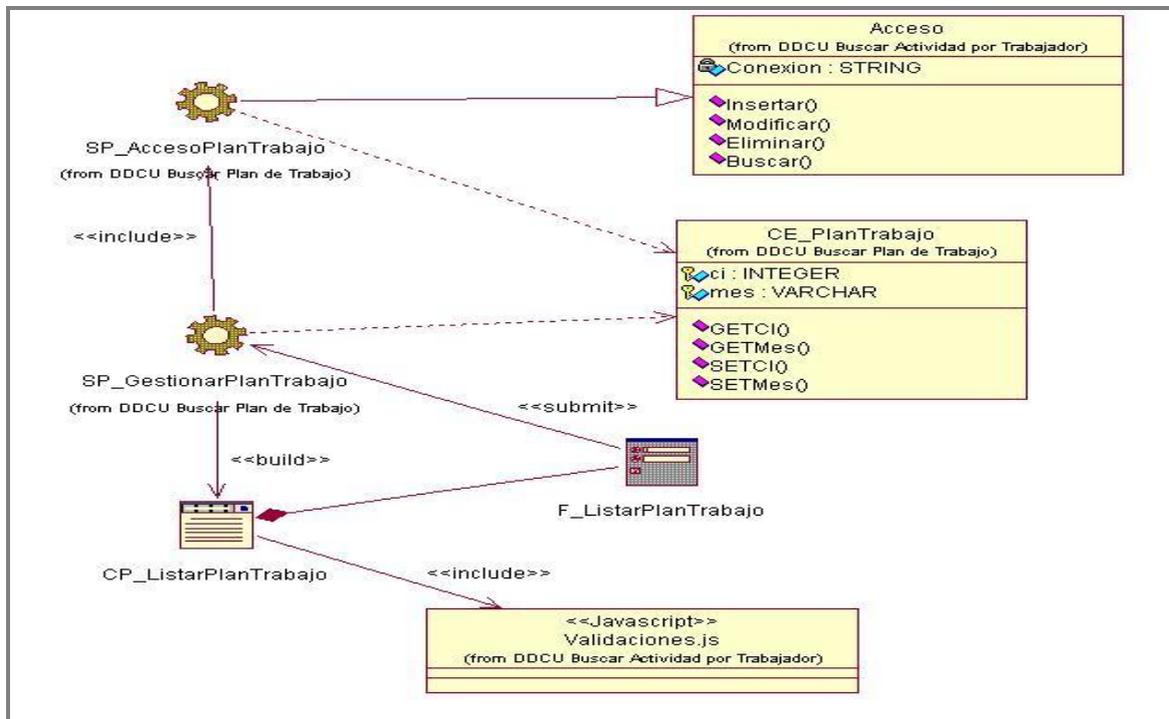


Figura 3.4– Diagrama de Clases del Diseño de Listar Plan de Trabajo

3.5 – Diagrama de Clases del Diseño de Reporte Desarrollo Individual

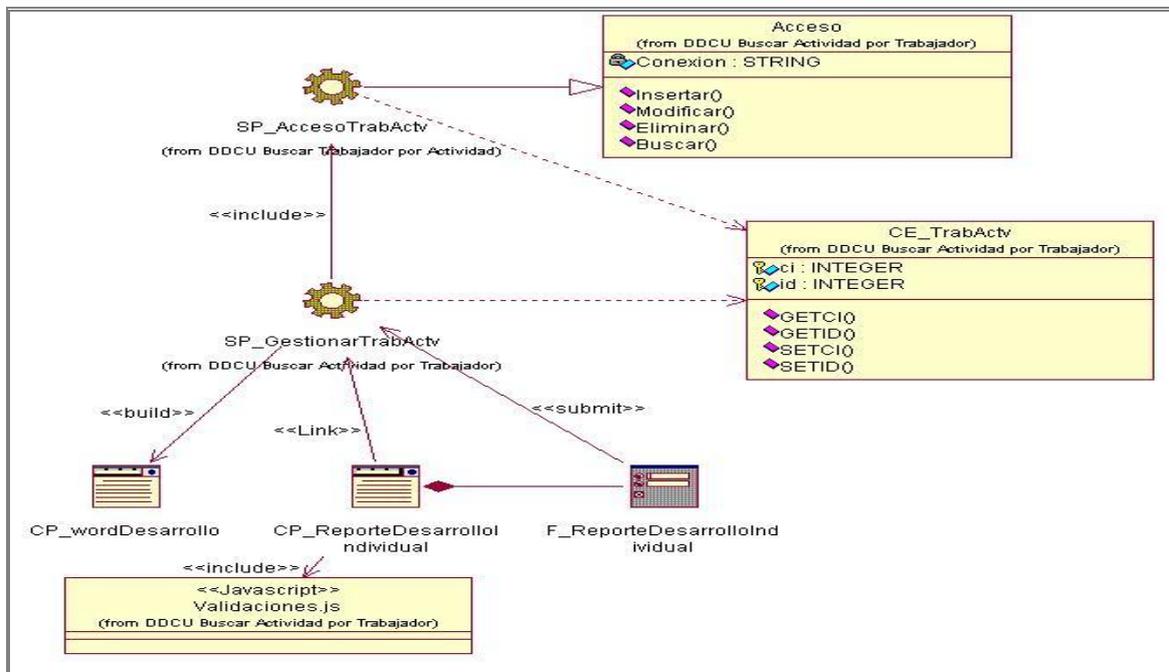


Figura 3.5– Diagrama de Clases del Diseño de Reporte Desarrollo Individual

4 – Diagramas de Componentes

4.1 – Diagrama de componentes de Buscar Necesidad

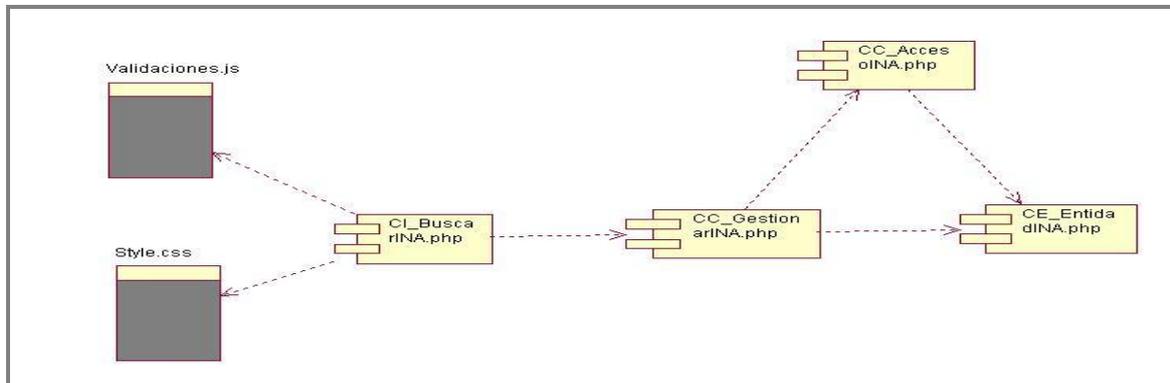


Figura 4.1– Diagrama de componentes de Buscar Necesidad

4.2 – Diagrama de componentes de Gestionar Plan de Trabajo

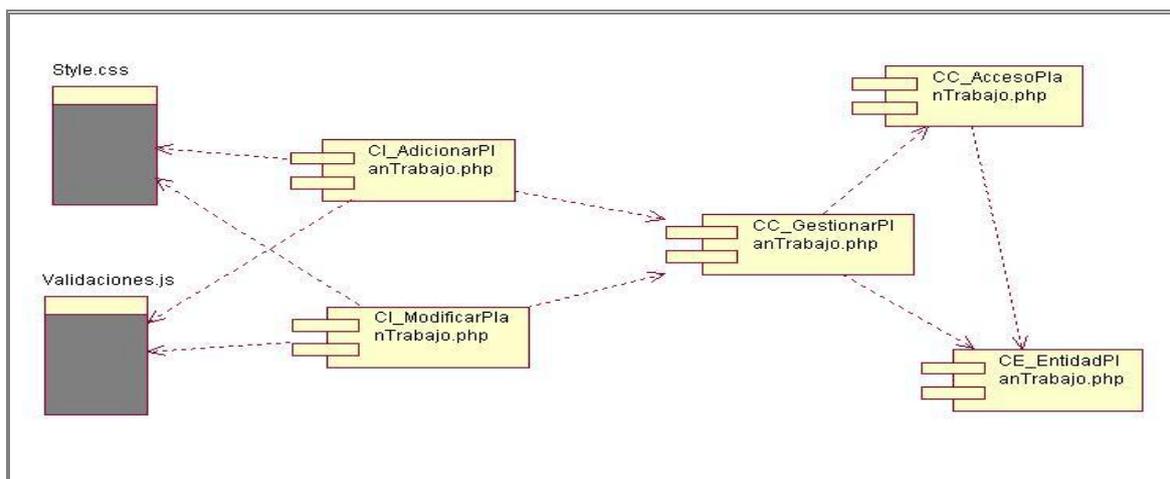


Figura 4.2– Diagrama de componentes de Gestionar Plan de Trabajo

4.3 – Diagrama de componentes de Listar Problema

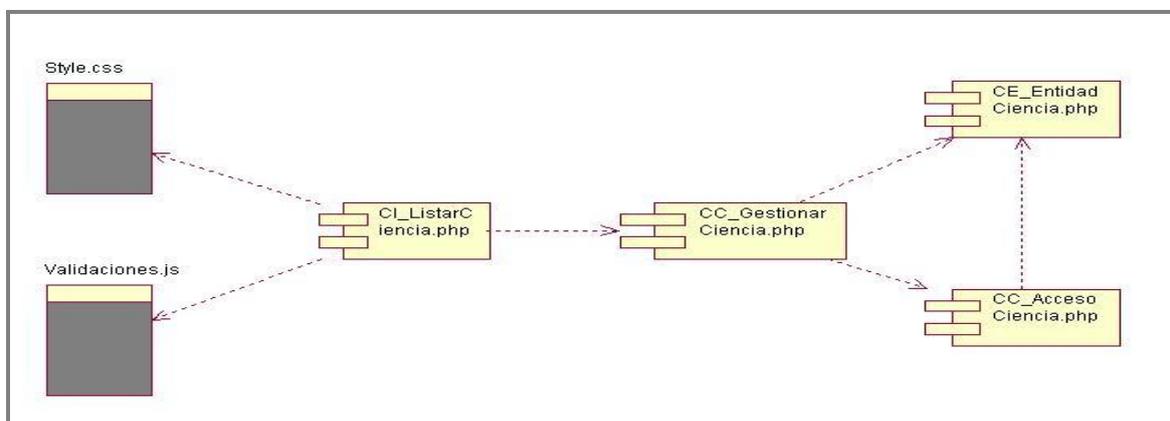


Figura 4.3– Diagrama de componentes de Listar Problema



4.4 – Diagrama de componentes de Reporte Desarrollo Individual

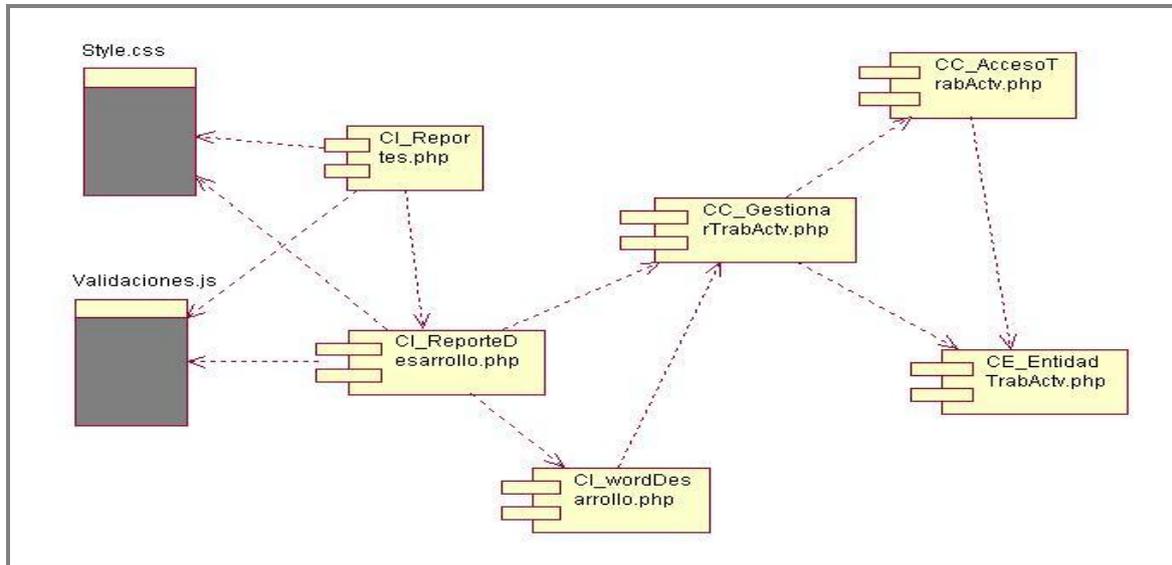


Figura 4.4– Diagrama de componentes de Reporte Desarrollo Individual

Glosario de Términos

XSLT: XSLT o Transformaciones XSL es un estándar que presenta una forma de transformar documentos XML en otros e incluso a formatos que no son XML.

XMLHttpRequest: También referida como XMLHttpRequest (Lenguaje de Marca Extensible / Protocolo de Transferencia de Hipertexto), es una interfaz empleada para realizar peticiones HTTP y HTTPS a servidores WEB. Para los datos transferidos se usa cualquier codificación basada en texto, incluyendo: texto plano y XML. El uso más popular de esta interfaz es proporcionar contenido dinámico y actualizaciones asíncronas en páginas WEB mediante tecnologías construidas sobre ella como por ejemplo AJAX.

JSP: Es un acrónimo de Java Server Pages, que en español sería Páginas de Servidor Java. Es una tecnología orientada a crear páginas web con programación en Java. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java.

URL: Significa Uniform Resource Locator, es decir, localizador uniforme de recurso y se refiere a la dirección única que identifica a una página web en Internet.

ActiveX: ActiveX es una tecnología de Microsoft para el desarrollo de páginas dinámicas. Tiene presencia en la programación del lado del servidor y del lado del cliente, aunque existan diferencias en el uso en cada uno de esos dos casos.

PDF: Acrónimo del inglés Portable Document Format, en español sería formato de documento portátil, es un formato de almacenamiento de documentos, desarrollado por la empresa Adobe Systems.

BSD: Es la licencia de software otorgada principalmente para los Sistemas BSD (Berkeley Software Distribution). Pertenece al grupo de licencias de software Libre. Esta licencia tiene menos restricciones en comparación con otras como la GPL estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

GPL: La Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License o simplemente su acrónimo del inglés GNU GPL, es una licencia creada por la Fundación de Software Libre a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

Perl: es un lenguaje de programación diseñado por Larry Wall en 1987. Perl toma características de varios lenguajes de programación. Estructuralmente, Perl está basado en un estilo de bloques y fue ampliamente adoptado por su destreza en el procesado de texto y no tener ninguna de las limitaciones de los otros lenguajes de script.

MySQL AB: es una compañía de software fundada en 1995, creadora del Sistema administrador de Bases de datos relacionales MySQL, y una de las más grandes empresas de software libre del mundo. El 16 de enero de 2008 Sun Microsystems adquirió la empresa, por lo que se prevee que su cartera de productos se integre con la de esta compañía.

GNU: Es un acrónimo recursivo que significa GNU No es Unix (GNU is Not Unix). El proyecto GNU fue iniciado por Richard Stallman con el objetivo de crear un Sistema Operativo completamente libre: el Sistema GNU.

IBM: International Business Machines o IBM (conocida coloquialmente como el Gigante Azul) es una empresa que fabrica y comercializa herramientas, programas y servicios relacionados con la informática. Tiene su sede en Armonk (Estados Unidos) y está constituida como tal desde el 15 de junio de 1911, pero lleva operando desde 1888.

Residencia: Es la labor de un médico que se halla adscripto por un período de cinco años en un servicio, sumergido en un programa de dedicación intensiva, con el objeto de lograr un dominio completo, profundo y rápido de una especialidad.

MGI: Medicina General Integral

EGI: Enfermería General Integral

SSH: Son las siglas de Secure SHell. Lo que te ofrece es una consola en un ordenador remoto con los privilegios que tenga la cuenta con la que conectes. Es decir, si en tu PC tienes varias cuentas, puedes conectar desde otro ordenador al tuyo con cualquiera de esas cuentas y sus respectivos privilegios, como pudiera ser la cuenta root, administrador o la de un usuario normal sin poder de administración. Y todo esto con encriptación de datos.

RSS: Es un formato para la sindicación de contenidos de páginas web. Sus siglas responden a Really Simple Syndication. To syndicate literalmente significa syndicar (formar parte de un sindicato. En inglés tiene otro significado: "publicar artículos simultáneamente en diferentes medios a través de una fuente a la que pertenece".

PDA: Periódicos, artículos en revistas especializadas en informática, anuncios en diversos medios, artículos en revistas de medicina y libros sobre informática médica son algunas fuentes posibles de información sobre la aplicación de dispositivos y programas informáticos para asistentes digitales personales (PDAs).

WCAG: Los documentos denominados Pautas de Accesibilidad al Contenido en la Web (WCAG) explican cómo hacer que el contenido Web sea accesible para personas con discapacidad. El término "contenido" Web normalmente hace referencia a la información contenida en una página Web o en una aplicación Web, incluyendo texto, imágenes, formularios y sonido.

W3C: El Consorcio World Wide Web (W3C) es un consorcio internacional donde las organizaciones miembros, personal a tiempo completo y el público en general, trabajan conjuntamente para desarrollar estándares Web.

AJAX: AJAX (Asynchronous Javascript and XML) traducido como indica el título, no es más que una forma de programar aplicaciones interactivas para web.

Mac OS: Mac OS, es la abreviatura de Macintosh Operating System (Sistema Operativo de Macintosh), es el nombre del primer Sistema Operativo de Apple para los ordenadores Macintosh.