Universidad de las Ciencias Informáticas
Facultad 4

Título: Análisis y Diseño del Sistema para la Gestión del Expediente del Profesor en la Universidad de las Ciencias Informáticas.

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Autor(es): **Humberto González Hernández**

Yuniesky Pérez Pérez

Tutor(es): **Ing. Henry Raúl González Brito**

Asesora: **Ing. Dayana Pupo Palacio**

Junio, 2008

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Humberto González Hernández

_____

Firma del Autor

Yuniesky Pérez Pérez

_____

Firma del Autor

Henry Raúl González Brito

_____

Firma del Tutor

DATOS DE CONTACTO

Tutor: Ing. Henry Raúl Gonzalez Brito, graduado de la Carrera Ingeniería Informática en el Instituto Politécnico José Antonio Echavarría y la Universidad de Camagüey en el marco del Plan CUJAE. Profesor Instructor de la Universidad de las Ciencias Informáticas, ha impartido diferentes asignaturas durante tres años en la Facultad 4 como Ingeniería de Software, Gestión de Software, Seguridad Informática y Metodología de la Investigación Científica. En este momento se desempeña como especialista superior en la Dirección de Producción 4 y es coordinador del Programa ERP Cuba.

Asesor(a): Ing. Dayana Pupo Palacio, Graduada de la Carrera de Ingeniería en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Recién graduada, actualmente labora como administradora de redes del departamento de redes del GRUPO DE TURISMO GAVIOTA S.A.

AGRADECIMIENTOS

A la Revolución y en especial a Fidel Castro principal forjador de lo que es hoy la universidad de excelencia, a familiares y amigos, y a todas las personas que de una forma u otra han formado parte de la realización del presente trabajo de diploma. A todos

Gracias.

DEDICATORIA

De Humberto

Especialmente a mi madre que es la razón principal por la que hoy estoy en este lugar, para ti mama, ya vez que como siempre ante todos los problemas me ahogo en vaso de agua y al final salgo adelante porque ha sido eso lo que me has sabido enseñar, levantar la cabeza y continuar con paso firme, gracias por todo y por la paciencia que has tenido.

A Dayana, mi novia, tutora, asesora, que hace un año me tuvo en su dedicatoria y hoy como bien dijo ella esta en la mia, por supuesto no podías faltar, has sido mi principal inspiración para seguir el camino.

A mi hermano Ale, que aunque yo no lo demuestre; tal vez por mi manera de ser, si te quiero con el alma y eso lo sabes.

A Amaury que ha sabido ser un padre para mi desde el principio.

A mis amigos y hermanos de la vida, en especial a Deibys que estuvo siempre conmigo en las buenas y en las malas, ya ves mi hermano lo logré.

A toda mi familia en general y a mi papá.

De Yuniesky

A toda mi familia.

A mi hermano Michel

A mis padres que lo han dado todo por mi éxito,

Pero en especial a mi nana por tener siempre confianza en mí y guiarme por el camino correcto, por darme siempre el apoyo que necesité en mis decisiones…

RESUMEN

En la Universidad de la Ciencias Informáticas, cada uno de los profesores posee un expediente docente que contiene una copia del titulo de graduado del profesor, así como de certificaciones, maestrías, diplomados, entro otros documentos de gran importancia, el cual se encuentra ubicado en el departamento de Capital Humano.

Generalmente, la gestión de la información en este departamento se realiza de forma manual, pues no existe un software que facilite dicho proceso. Es por ello, que en el presente trabajo de diploma se propone diseñar un sistema automatizado que facilite la gestión de la información relacionada con el expediente del profesor.

Para diseñar el sistema, se utiliza la metodología de desarrollo RUP con el lenguaje UML y la Suite de Rational como herramienta de modelado. Con la elaboración del diseño, se definen las clases fundamentales que permitirán que el sistema funcione satisfactoriamente, así como los atributos y métodos que deben tener las mismas, lo cual brindará al desarrollador una idea clara de lo que debe implementar.

Se espera que el sistema sea desarrollado siguiendo el diseño propuesto. Además, que permita mejoras en la gestión de la información, manejada en el departamento de Capital Humano en la Universidad de las Ciencias Informáticas.

PALABRAS CLAVE

Gestión de la información, expediente del profesor.

TABLA DE CONTENIDOS

AGRADECIMIENTOS III

DEDICATORIA IV

RESUMEN VI

INTRODUCCIÓN 1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA 5

Introducción 5
1.1 Conceptos básicos relacionados con el problema planteado 5
1.2 Sistemas automatizados existentes. 8
1.4 Tendencias actuales. 12
1.5 Tecnologías para el desarrollo de aplicaciones Web. 22
Conclusiones: 41

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA 42

Introducción 42
2.1 Objeto de estudio. 42
2.2 Objeto de automatización. 44
2.3 Información que se maneja en el negocio. 45
2.4 Propuesta de Sistema 46
2.5 Modelo del Negocio. 47
2.6 Especificación de los requerimientos del software. 59
2.7 Modelado del sistema. 64
Conclusiones 84

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA 85

Introducción 85
3.1 Diagramas de clases de análisis. 85
3.2 Diagramas de clases de diseño. 89
3.3 Diseño de la base de datos propuesta. 100
Conclusiones 103

CONCLUSIONES 104

RECOMENDACIONES 105

GLOSARIO 106

REFERENCIA BIBLIOGRÁFICA 108

BIBLIOGRAFÍA 110

INTRODUCCIÓN

En la medida en que ocurre un tránsito hacia una nueva era, se intenta llamar la atención en aspectos de vital importancia que por su novedad aún no son de un amplio dominio. Muchas veces determinadas teorías o enfoques emergentes, no necesariamente conllevan a la sustitución de otras. Son aspectos nuevos que pueden coexistir o no con enfoques anteriores.

En ocasiones, ante el surgimiento de un nuevo enfoque se maneja un conjunto de aspectos, sin realzar determinados puntos de contacto con otros que coexisten. La aplicación de cualquier teoría, técnica o enfoque debe partir de un estudio detallado del contexto en el que debe aplicarse, conociendo sus puntos de contacto y diferencias, para no provocar distorsiones en el sistema.

El desarrollo de la tecnología surgido en la década del 80, ha permitido un salto en el manejo de informaciones, no solo en términos de velocidad, sino de linealidad. Las personas han ganado en interacción con las computadoras, y se han integrado a la sociedad a un ritmo vertiginoso, en forma tal, que personas de cualquier edad las emplean para diferentes actividades que van desde el ocio y el entretenimiento hasta el cálculo científico, las decisiones, el diseño.

La sociedad industrial va dando paso a una sociedad postindustrial, también llamada sociedad de la información, o del conocimiento. En la sociedad industrial la mano de obra constituía un componente vital para el desarrollo de la producción. Así en esos años, surgieron diferentes enfoques gerenciales orientados hacia el estudio y la atención de los recursos humanos y su gestión.

El desarrollo tecnológico ha facilitado la ejecución de muchos procesos mediante máquinas y tecnologías. Hoy, mucha mano de obra se ha sustituido por equipos que elaboran, montan, ejecutan actividades programadas por el hombre sustituyéndolo en determinadas actividades.

En esta sociedad emergente tiene un papel importante el desarrollo de nuevos métodos, nuevos productos, nuevos enfoques, nuevos paradigmas. Esto solo lo puede llevar a cabo el hombre, no con sus manos, sino con sus conocimientos y su inteligencia.

Estos conocimientos deben ser estimulados. Para ello, las organizaciones deben crear espacios donde su personal pueda generar esos conocimientos tácitos, pueda explicitarlos, registrándolos, para su incorporación al quehacer organizacional.

La Gestión Documental pretende el tratamiento integral, consistente y fiable de los documentos y la información que se genera en todo centro laboral.

En la Universidad de la Ciencias Informáticas, al igual que en cualquier universidad del país, cada uno de sus profesores posee un expediente docente.

Estos expedientes docentes están ubicados en el departamento de Capital Humano donde se reflejan copias de los certificados y documentos que han ido obteniendo en su labor, dentro de los cuales se aprecian: una copia del título de graduado, títulos de diplomados y maestrías, certificados de cursos de postgrados, certificados por la participación en eventos y publicaciones, las evaluaciones anuales que se le hacen al profesor, premios obtenidos, reconocimientos, entre otros.

En la mayoría de las ocasiones, se dificulta el manejo de la información contemplada en el expediente del profesor, debido a que no existen copias de un mismo expediente y cuando más de una persona necesitan consultar el mismo expediente surge entonces la dificultad para gestionar la información.

Frecuentemente, las personas que dirigen determinados procesos docentes relacionados con el profesor deben revisar su expediente, apreciándose esto durante el proceso de categorización de un profesor, para lo cual debe irse personalmente al departamento de Capital Humano, empleando parte de su tiempo para ello, aun si solo se fuese a realizar una simple revisión del mismo.

También ocurre con frecuencia que los profesores esperan a tener acumulados una cierta cantidad de documentos para luego ir al Departamento de Capital Humano a actualizar los datos de sus propios expedientes, lo cual implica que en el tiempo que transcurre entre cada actualización, no pueda obtenerse quizás una información muy completa de cada uno de los profesores.

Evidentemente existe la necesidad, en el orden práctico, de la existencia de una herramienta automatizada que permita gestionar los expedientes de los profesores en la Universidad de las Ciencias Informáticas, garantizando la calidad y eficiencia de este proceso.

De lo antes expuesto se determinó como **problema científico** el siguiente:

¿Cómo mejorar los procesos de gestión de la información relacionados con el manejo del expediente del profesor en la Universidad de las Ciencias Informáticas, mediante un sistema automatizado?

Este problema se enmarca en el **objeto de estudio** concerniente a los procesos de gestión de la información del expediente del profesor, que se centra en diseñar un sistema para la informatización de los procesos de gestión de la información del expediente del profesor en la Universidad de las Ciencias Informáticas, definiéndose como **campo de acción**.

Basado en esa idea se define el **objetivo general** del presente trabajo: Diseñar un sistema informático que permita mejorar los procesos de gestión de la información relacionada con el expediente del profesor en la Universidad de las Ciencias Informáticas.

Los **objetivos específicos** que se proponen para alcanzar el objetivo general son los siguientes:

- Llevar a cabo un estudio de los procesos de gestión de la información relacionados con el expediente del profesor.
- Hacer un levantamiento de requisitos que permita conocer los procesos que intervienen en la gestión del expediente del profesor.
- Profundizar en el estudio de las tecnologías, metodologías y herramientas para el desarrollo de software.
- Identificar y seleccionar los artefactos que se utilizarán durante las distintas fases.
- Hacer el análisis y diseño del sistema utilizando una arquitectura que se adecue a los lineamientos de la Dirección de Informatización de la Infraestructura Productiva.

De acuerdo con los objetivos mencionados para lograr un correcto análisis y diseño de dicho sistema, es necesario llevar a cabo un conjunto de **tareas** que permitan dar cumplimiento a los mismos como son:

- Realizar búsquedas bibliográficas para el estudio de los sistemas de gestión documental.
- Analizar en detalles el negocio y obtener de la forma más concreta posible las funcionalidades del sistema a desarrollar.
- Buscar información acerca de las tendencias de las tecnologías actuales en el campo de la informática.
- Investigar profundamente en los aspectos fundamentales de la metodología y herramientas a utilizar, en las cuales estará basado el desarrollo del sistema.
- Modelar una base de datos acorde a los requerimientos de la aplicación para su mejor funcionamiento.
- Modelar los procesos que permiten la gestión del expediente del profesor.

El presente trabajo de diploma está compuesto por tres capítulos:

El Capítulo 1: Fundamentación Teórica: Aborda aspectos relacionados con los Sistemas de Gestión Documental, así como ventajas que proporciona la aplicación de dichos sistemas. Además del análisis de metodologías, lenguaje y herramientas de modelado de software.

El Capítulo 2: Características del sistema: Abarca la descripción de los procesos del negocio, identificación de los actores, los casos de uso y la relación entre ellos representada en el modelo del negocio. Se especifican los requerimientos funcionales y no funcionales del sistema. Se identifican y justifican los actores del sistema y se muestra el diagrama de Casos de Uso del Sistema.

El Capítulo 3: Análisis y diseño del sistema: Se muestran los diagramas de Clases de Análisis. Teniendo en cuenta la metodología RUP, se elaboran un grupo de artefactos que pertenecen a los flujos de trabajo de Análisis y Diseño y se muestran los diagramas de interacción correspondientes.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción.

La gestión documental también denominada gestión de documentos no es un término nuevo pero se perfila como un tema mundialmente tratado debido a la amplia cobertura de la información en la era actual y adquiere mayor importancia estratégica en el desarrollo de una organización.

En el presente Capítulo se abordan diferentes temas entre los que se encuentran la gestión documental, los sistemas de gestión documental, entre otros. Además se hace un análisis de las tendencias de las metodologías de desarrollo de software, lenguajes y herramientas que permiten la elaboración del modelado, análisis y el diseño de los sistemas.

1.1 Conceptos básicos relacionados con el problema planteado

1.1.1 ¿Qué es un documento?

Un documento es información registrada en cualquier forma o medio, creado o recibido, conservado o usado por una organización o un individuo en la transacción de negocios o la realización de asuntos y que se guardan como testimonio de dicha actividad.[1]

Constituye toda expresión en lenguaje natural o convencional y cualquier otra expresión gráfica, sonora o imagen, recogida en cualquier tipo de soporte material que puede ser en papel, en microforma o en un soporte magnético o electrónico de cualquier otro tipo.

Un documento es toda información fijada materialmente sobre un soporte, en la cual se recoge la evidencia producida por los sucesos a través del tiempo.

1.1.2 ¿Qué es un documento electrónico?

Un documento electrónico es un contenedor de información en formato digital, la cual recoge información simultáneamente de varias fuentes, en torno a específicos temas para satisfacer las necesidades de un individuo en particular.[2]

Puede estar en forma electrónica porque se ha creado mediante un programa informático de aplicación o bien porque se ha digitalizado, por ejemplo, cuando se ha escaneado a partir de un documento en papel o de una microforma.

También se puede considerar como documento electrónico toda información que puede constituir un documento de archivo cuyo tratamiento es automatizado y requiere de una herramienta específica para leerse o recuperarse.

Para su visualización es necesaria la utilización de un dispositivo electrónico. Este difiere del documento impreso en el material que lo conforma, es decir que ambos pueden contener el mismo texto pero en soportes diferentes.

Es necesario destacar que no es lo mismo documento electrónico que documento digital. Un documento digital es un documento electrónico, pero no todo documento electrónico es un documento digital, los primeros pueden ser análogos o digitales.

1.1.3 Acercamiento al concepto de gestión Documental.

El hombre ha utilizado a través de los años diferentes materiales cada vez con mejores ventajas de uso en comparación con los anteriores, con el objetivo de dejar constancia de sus actos y pensamientos.

Hoy día con el rápido crecimiento en el volumen de los documentos que se generan en las empresas profesionales ha surgido la necesidad de la creación de archivos que permitan controlar y almacenar grandes cantidades de información, y otras ventajas como el acceso a la misma en el menor tiempo posible.

Todas estas optimizaciones están favorecidas gracias al desarrollo de las tecnologías de la información y de las comunicaciones (TIC), las cuales en gran medida han dado nueva visión al manejo de la información. Esta nueva visión del manejo de la información se le conoce como gestión documental, la cual constituye una de las metas de las empresas en su afán de poder controlar grandes volúmenes de información.

En los nuevos modelos de negocio la gestión de la información, la documentación y el conocimiento se perfila como un componente estratégico de primera magnitud. [3]

La Gestión Documental de manera general es un conjunto de tareas y procedimientos que permiten coordinar y controlar sistemáticamente, todo lo relacionado con la creación, recepción, organización, almacenamiento, preservación, mantenimiento, acceso y difusión de los documentos en una organización durante su ciclo de vida.

1.1.4 Acercamiento al concepto de Sistemas de Gestión Documental (SGD).

Los SGD son programas de gestión de bases de datos que disponen de una tecnología idónea para el tratamiento de documentos científicos, culturales y técnicos.[4]

Para intentar explicar qué es un sistema de gestión documental, tenemos que insistir en dos aspectos importantes: los objetivos que pretendemos conseguir con dicho sistema y el ámbito de aplicación del proyecto. [5]

La definición de objetivos debe adecuarse a cada empresa, pero hay unos elementos básicos y comunes, que son aplicables en todos los entornos. En general, un sistema de gestión documental pretende:

- Hacer más fácil a las personas trabajar con los documentos. Cada persona sabe como encontrar en poco tiempo los documentos adecuados cuando los necesita.
- Facilitar que la información se comparta y se aproveche como un recurso colectivo, evitar duplicados y copias innecesarias.

Definir el ámbito de aplicación de un SGD desde un punto de vista práctico, nos obliga a considerar un enfoque global en el que tienen que integrarse los archivos en papel, los documentos electrónicos y las bases de datos. [5]

Un reto de los SGD es buscar soluciones y tecnologías que garanticen la fiabilidad e integridad y acceso seguro a la información que se trata.

1.2 Sistemas automatizados existentes.

1.2.1 Internacionales.

AUPAC:

Es un producto de gestión integrada de registros, expedientes, trámites, documentos, acuses de recibo, resoluciones, órganos de gobierno, vencimientos, inventario y archivos administrativo e histórico.

Entre las principales funcionalidades que ofrece AUPAC se pueden mencionar las siguientes:

- Registro de entrada y salida con certificación mecánica en validadora.
- Integración con bases de datos de terceros y enlace con otras bases de datos externas.
- Gestión automatizada de notificaciones y acuses de recibo
- Gestión documental en Word y Open Office con control de versiones y digitalización integrada
- Identificación automática de documentos con códigos de barras individualizados
- Control de vencimientos mediante tareas y alertas
- Control de traslados físicos de fondos, expedientes y acuses de recibo
- Gestión de expedientes (multi-expediente y multi-trámite) y automatización con circuitos gráficos de tramitación (workflow)
- Inventario de procedimientos
- Tratamiento de órganos de gobierno, acuerdos y resoluciones
- Inventario de bienes y gestión de patrimonio
- Terminal para liquidación de tasas
- Explotación estadística con informes dinámicos
- Cuadros de mando e indicadores de gestión integral mediante cubos OLAP
- Niveles de seguridad independientes por entidad, servicio, usuario y objeto
- Configuración, parametrización y mantenimiento administrativo integrado

Las ventajas más importantes del producto AUPAC son:

- Ofrece a la administración la organización de procedimientos, informaciones administrativas, circuitos asistidos de tramitación, estructuras departamentales, recursos humanos, etc.

- Proporciona una metodología flexible de trabajo a través de un sistema unificado que contribuye a una imagen corporativa clara.
- Transmite eficacia a los procesos de tramitación, documentación y resolución de expedientes.
- Aporta control, medición y seguimiento de los procesos administrativos para la detección de "cuellos de botella".
- Garantiza la calidad de los procedimientos implantados según las normas ISO 9000 proporcionando a la Administración valiosos instrumentos para promover la mejora continua de los procesos.
- Facilita un servicio de atención ciudadana ágil y permanente empleando las nuevas tecnologías e Internet.
- Posibilita la satisfacción para los administrantes por la mejora del trabajo realizado y para los administrados por la calidad del servicio obtenido.

DocuWare 5

El sistema DocuWare 5 (Gestión integrada de documentos) es un programa de software con tecnología avanzada para gestión integrada de documentos. Es capaz de procesar todo tipo de documentos, de fuentes diversas y dispone de procedimientos de control internos para asegurar el cumplimiento de los requisitos de auditoría de empresa y compatibilidad.

Importa documentos, los clasifica, añade un índice de texto completo y los prepara para el procesamiento posterior. El programa está dotado de funciones adicionales para gestión de registros, que garantizan la seguridad y el control de las operaciones de acceso.

Utiliza funciones de flujo de trabajo, gestión de contenido Web e integración universal para proporcionar eficaces prestaciones de gestión de contenido empresarial (ECM) que facilitan la expansión en el ámbito de una organización.

Ofrece comodidad de uso y seguridad. Hasta los más exigentes aprobarán el caudal de funciones y la facilidad de uso y administración que, combinado con la seguridad óptima y la capacidad de integración, hacen de DocuWare 5 un producto preparado para evolucionar según los requisitos de la empresa.

Ventajas que ofrece el sistema

- Un archivador central para todos los documentos
- Información disponible en todo momento en cualquier lugar
- Facilidad de búsqueda y rápida recuperación de datos
- Integración de cualquier flujo de trabajo
- Máxima seguridad

1.2.2 Nacionales

Babel

Babel es un sistema que integra las tecnologías de la información a la gestión de solicitudes de los servicios de traducción e interpretación del Centro de Información de la Empresa de Telecomunicaciones de Cuba S.A. (ETECSA).

Mediante una interfaz de comunicación amigable los usuarios pueden realizar un intercambio de datos entre todas las funciones implicadas en este proceso y así aprovechar adecuadamente las sinergias que se producen entre todas y cada una de las funciones.

Este sistema da a sus usuarios la información precisa sobre el estado de su solicitud y además, las competencias del traductor, al aumentar el valor añadido de cada recurso que interviene en el proceso. Esta herramienta de trabajo permite la organización, clasificación de la información y la recuperación de documentos con oportunas normas de seguridad.

PAPIRO

Es un producto informático de uso libre que emplea herramientas igualmente libres y permite conservar documentación de valor histórico al evitar su manipulación; pues, al digitalizarse el documento, este se consulta en formato electrónico. Incluye también la gestión de publicaciones periódicas, especialmente revistas.

La consulta de los documentos, puede ser realizada a través de una eficiente gestión que permite recuperar información por todas las variables de almacenamiento en la base de datos; mientras, la posibilidad de colocar en la red de redes dichas bases de datos y las imágenes de los documentos, no sólo democratiza la accesibilidad documental; sino, que ahorra recursos materiales y tiempo, haciendo más eficaz y eficiente el proceso de búsqueda e investigación documental.

Está concebido para correr bajo la plataforma Windows en las versiones 2000 y XP. Los formatos de imagen de los documentos están en pdf, estructura de uso común y gran poder de compresión, y para su consulta en la red se diseñó un método de gestión dinámica con páginas PHP.

El sistema se emplea en el Archivo histórico de la ciudad de Manzanillo, Granma, y si bien fue diseñado para su empleo en repositorios históricos, también puede aplicarse en archivos de gestión, resultando al mismo tiempo un intento pionero -por lo menos en Cuba-, al vincular la gestión de bases de datos con las imágenes de documentos originales digitalizados, contribuyendo de este modo a su conservación y socialización.

1.3 Propósito de los SGD.

Los SGD son desarrollados para proveer un repositorio donde los documentos pueden ser creados, clasificados, manejados y almacenados para facilitar el acceso de usuarios a los mismos.

Representan la integración de las tecnologías de la información con el tratamiento de documentos que incluye almacenamiento, manejo, recuperación y acceso a los documentos.

Los SGD brindan la posibilidad de hacer seguros, accesibles, recuperables e intercambiables los documentos que sean procesados electrónicamente, posibilitando su gestión se haga de forma más eficiente que el manejo de los mismos en formato duro.

1.3.1 Ventajas de los SGD.

La aplicación de un sistema de gestión documental permite un incremento exponencial de la productividad empresarial, ya que facilita la ubicación y el manejo de la información además que reduce en gran medida, el exceso de documentos que generalmente se conservan en las organizaciones.

- Reducción del tiempo de consulta de documentos electrónicos.
- Reducción del tiempo necesario para realizar determinados procesos.
- Acceso concurrente a un documento.
- Mejora de atención a los clientes.
- Ahorrar espacio físico, eliminando todos los espacios dedicados al archivo de documentos en papel.
- Incremento en la satisfacción de los usuarios.

1.3.2 Desventajas de los SGD.

Después de haber realizado un análisis de los disímiles beneficios que presenta un SGD, se puede apreciar que existen inconvenientes que hacen deficiente el sistema desde determinados puntos de vista:

- El intercambio de información es a nivel del documento como una única unidad de información.
- Problemas y dificultades de interoperabilidad en el intercambio de información.
- Requiere tiempo para la preparación y adaptación del usuario.
- No son lo suficientemente flexibles al diseño de un proyecto específico.

En la actual sociedad de la información los planes de gestión de documentos se deben soportar en herramientas informáticas de gestión de documentos con sus adecuados sistemas de almacenamiento y políticas de conservación.

1.4 Tendencias actuales.

En la actualidad, durante el proceso de desarrollo de software, se tienen en cuenta una serie de tendencias y tecnologías informáticas, de manera que se identifican un grupo de ellas para el desarrollo de la investigación.

Tanto el modelado de los sistemas de software como las herramientas que se usan para la elaboración de los mismos, son desde hace ya algunos años el complemento fundamental para el desarrollo de un software de calidad. No se debe dejar de mencionar que el lenguaje de modelado es también otro elemento que cumple vital importancia para poder cumplir lo antes mencionado.

1.4.1 Interfaces gráficas.

Las interfaces gráficas son una evolución natural de las interfaces modo texto y la sencillez de uso y robustez que se puede alcanzar dentro de una interfaz gráfica es mucho mayor, además que permiten diseños de mucha riqueza visual, más expresivos, flexibles y eficaces, que llegan a alcanzar tanta accesibilidad y rapidez como su antecesora.

Mediante el uso de menús se puede hacer que el usuario ejecute comandos de forma guiada, sin necesidad de conocer la sintaxis concreta del mismo. Con un diseño adecuado de los menús de una aplicación, se pueden poner a disposición del usuario decenas de comandos fácilmente accesibles y que pueden guiar al usuario.
Incluso los comandos más utilizados se pueden incluir en una barra de herramientas de fácil acceso para un uso más rápido de la interfaz.

Logrando un diseño correcto de interfaces gráficas, se pueden lograr algunas de las grandes ventajas de las interfaces de modo texto, como son la accesibilidad para personas discapacitadas, o la completa utilización del teclado.

1.4.2 Interfaces Web.

La aparición de la Web hizo posible que cualquier persona pueda ofrecer información particularizada a los demás y encontrar documentos interactivos sobre cualquier tema, relacionados unos con otros mediante enlaces que permiten navegar de página en página alrededor del mundo.

Las páginas Web supusieron la aparición de las interfaces Web, interfaces gráficas de usuario con unos elementos comunes de presentación y navegación que pronto se convirtieron en estándares.

Este tipo de interfaces sirven de intermediarias entre usuarios genéricos, no acostumbrados generalmente al uso de aplicaciones informáticas, y unos sistemas de información y procesos transaccionales que son transparentes para el usuario, debiendo posibilitar la localización de la información deseada, el entendimiento claro de las funcionalidades ofrecidas, la realización práctica de tareas específicas por parte de los usuarios y la navegación intuitiva por las diferentes páginas que forman la aplicación Web.

El diseño de las aplicaciones Web ha evolucionado con el tiempo hacia un esquema general perfectamente definido, ofreciendo unas interfaces bien diseñadas, con un conjunto de componentes gráficos y funcionales similares que hacen posible que sea cual sea el usuario que accede a una aplicación Web cualquiera, la comunicación entre ellos sea posible y efectiva.

De esta forma se han definido elementos y agrupaciones de éstos que han demostrado su utilidad y su comprensión por los usuarios, entre los que se destacan:

- Los sistemas de navegación constituyen elementos de una interfaz que permiten la navegación por las diferentes secciones y vistas que componen la aplicación Web. Generalmente se presentan como menús formados por diferentes opciones, con las que el usuario puede interactuar al hacer clic sobre cada una ellas cargando una vista o sección diferente.
- El cuerpo de la página es la parte de la interfaz Web que presenta a los usuarios información específica sobre un tema concreto. Es la parte de la página que la identifica e individualiza frente a las demás de una aplicación Web.
- El pie de página es un elemento opcional que suele contener información muy concreta. Si la página es muy alta, de tal forma que el usuario se vea obligado a utilizar la barra de desplazamiento vertical del navegador, el pie de página suele contener un menú auxiliar que permita al usuario continuar navegando por el sistema sin tener que volver a buscar el menú principal.
- Los formularios de entrada de datos en los cuales los usuarios registrarán la información solicitada para que posteriormente persista en el sistema de almacenamiento de datos que esté acoplado a dicha aplicación Web. Estos elementos generalmente tienen campos requeridos en función de los servicios que presta la aplicación Web.

1.4.2.1 Ventajas.

Las aplicaciones Web tienen disímiles ventajas, como es el caso del fácil manejo de la gestión de cambios, funcionan de igual modo independientemente de la versión del sistema operativo instalado. En lugar de crear clientes para distintos sistemas operativos, la aplicación es escrita una vez y mostrada en cualquier lugar.

Utilizan lenguajes interpretados del lado del cliente, tales como JavaScript, para añadir elementos dinámicos a la interfaz de usuario. Generalmente cada página Web individual es enviada al cliente como un documento estático, pero la secuencia de páginas provee de una experiencia interactiva.

1.4.2.2 Desventajas.

Las interfaces Web tienen ciertas limitantes en la funcionalidad del cliente. Métodos comunes en las aplicaciones de escritorio como arrastrar y soltar no están soportados por las tecnologías Web estándares.

Los desarrolladores Web comúnmente utilizan lenguajes interpretados del lado del cliente para añadir más funcionalidad, especialmente para crear una experiencia interactiva que no requiera recargar la página cada vez que se realiza una petición, algo que suele molestar a los usuarios. Desde el punto de vista de estilos, la información no es presentada de la misma forma en todos los navegadores, por ejemplo Mozilla Firefox e Internet Explorer.

Otro impedimento resulta el hecho de tener que ajustarse a un rango de capacidad, una interfaz Web excesivamente cargada de contenido se torna demasiado pesada a la hora de ser visualizada por el cliente, debido a que implica más tiempo de espera, el uso excesivo de contenido de peso es directamente proporcional al tiempo de espera del usuario al recibir una respuesta del servidor.

Aunque tecnologías como AJAX agilizan en gran medida la interacción cliente servidor es siempre oportuno el buen uso de las prácticas de diseño Web.

1.4.3 Internet.

Otro de los factores que facilita la elección de desarrollar aplicaciones Web es la Internet, que se puede definir como una red global de redes de ordenadores cuya finalidad es permitir el intercambio libre de información entre todos sus usuarios, utilizando un conjunto de protocolos, el más destacado, el TCP/IP. Una de las ventajas de Internet es que posibilita la conexión con todo tipo de ordenadores, desde los personales, hasta los más grandes que ocupan habitaciones enteras [6].

1.4.4 Navegadores Web

La difusión de los distintos sistemas a través de Internet se logra mediante los navegadores Web. Un navegador Web o *browser* es una aplicación software que permite al usuario recuperar y visualizar documentos de hipertexto, comúnmente descritos en *HTML*, desde *servidores* Web de todo el mundo a través de Internet.

La funcionalidad básica de un navegador Web es permitir la visualización de documentos de texto, posiblemente con recursos multimedia incrustados. Los documentos pueden estar ubicados en la computadora donde está el usuario, pero también pueden estar en cualquier otro dispositivo que esté conectado a la computadora del usuario o a través de Internet, y que tenga los recursos necesarios para la transmisión de los documentos.
Existen diferentes navegadores Web tales como Mozilla, Windows Internet Explorer, Firefox, Netscape, Opera, Konqueror, entre otros [7].

1.4.5 Orientación a objetos.

Un paradigma de programación es un modelo básico de diseño y desarrollo de programas que permite determinar la estructura de los mismos con la ayuda de modelos conceptuales.

Existen múltiples formas de construcción de sistemas software, entre ellas la programación estructurada, funcional, lógica, orientada a objetos, entre otras. Sin embargo la tecnología orientada a objetos es el paradigma de programación más utilizado en la actualidad ya que su consistente base teórica y la amplia gama de herramientas que permiten crear código a través de diseños orientados a objetos la convierten en la alternativa más adecuada para el desarrollo de aplicaciones.

Además fomenta la reutilización y extensión del código, permite crear sistemas más complejos y facilita el mantenimiento del software. Lo interesante es que proporciona conceptos con los cuales se modela y representa el mundo real tan fielmente como sea posible.

La importancia de hacer esta referencia radica en que para el desarrollo de software orientado a objetos no basta usar un lenguaje que lo implemente, también se necesitará realizar un análisis y diseño orientado a objetos ya que el modelado visual es fundamental para el propio desarrollo, siendo este argumento esencial para valorar la importancia de este trabajo, pues un diseño orientado a objetos maximiza la modularidad y la encapsulación.

Todo esto trae como consecuencia que el sistema se descompone en objetos con responsabilidades claramente especificadas; la extensibilidad pues da la posibilidad de ampliar la funcionalidad de la aplicación de manera sencilla y la reusabilidad, que permite reutilizar parte del código para el desarrollo de una aplicación similar.

1.4.6 Sistemas distribuidos.

En lo que respecta a la arquitectura se observa la tendencia a la utilización de sistemas distribuidos, los cuales se definen como una colección de computadores autónomos conectados por una red, y con el software distribuido adecuado para que el sistema sea visto por los usuarios como una única entidad capaz de proporcionar facilidades de computación.

Los sistemas distribuidos se implementan en diversas plataformas hardware, desde unas pocas estaciones de trabajo conectadas por una red de área local, hasta Internet [8].

1.4.7 Arquitectura de aplicación.

Una arquitectura es un entramado de componentes funcionales que aprovechando diferentes estándares, convenciones, reglas y procesos, permite integrar una amplia gama de productos y servicios informáticos, de manera que pueden ser utilizados eficazmente.

Debemos señalar que para seleccionar el modelo de una arquitectura, hay que partir del contexto tecnológico y organizativo del momento, teniendo en cuenta además que la administración de sistemas y la definición de dónde ubicar los datos, deben ser seriamente consideradas y evaluadas, ya que las facilidades de administración de sistemas son un requisito indispensable para una implantación exitosa.

Cuando se habla de aplicaciones Web tenemos que pensar en la mejor arquitectura para el control e intercambio de información a través de la red, siendo la Arquitectura Cliente / Servidor la mas significativa.

1.4.7.1 Modelo cliente/servidor (dos capas).

El modelo cliente-servidor de un sistema distribuido es el modelo más conocido y ampliamente adoptado en la actualidad.

Hay un conjunto de procesos servidores, cada uno actuando como un gestor de recursos para una colección de recursos de un tipo, y una colección de procesos clientes, cada uno llevando a cabo una tarea que requiere acceso a algunos recursos hardware y software compartidos. Los gestores de recursos a su vez podrían necesitar acceder a recursos compartidos manejados por otros procesos, así que algunos procesos son ambos clientes y servidores.

En el modelo, cliente-servidor, todos los recursos compartidos son mantenidos y manejados por los procesos servidores. Los procesos clientes realizan peticiones a los servidores cuando necesitan acceder a algún recurso. Si la petición es válida, entonces el servidor lleva a cabo la acción requerida y envía una respuesta al proceso cliente. [9]

Entre las principales características de la arquitectura cliente/servidor se pueden destacar las siguientes:

- El servidor presenta a todos sus clientes una interfaz única y bien definida
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo
- Los cambios en el servidor implican pocos o ningún cambio en el cliente [10].

1.4.7.2 Modelo cliente/servidor (tres capas).

La arquitectura de tres capas se refiere a un diseño reciente que introduce una capa intermedia al proceso. Cada capa es un proceso separado y bien definido corriendo en plataformas separadas. En la arquitectura tradicional de tres capas se instala una interfaz de usuario en la computadora del usuario final (el cliente).

La arquitectura basada en WEB transforma la interfaz de búsqueda existente (el explorador de WEB), en la interfaz del usuario final. La tercera capa generalmente es el sistema de administración de la *base de datos*. Es decir donde los datos requeridos por la capa intermedia son almacenados. La tercera capa se localiza en un *servidor* separado conocido como el *servidor* de *base de datos*.

.La parte funcional de la arquitectura de tres capas generalmente es conocida como la capa intermedia o el servidor de aplicaciones. En éste ocurren la mayoría de los procesos [11].

1.4.8 Servicios Web (Web services)

En el caso del manejo de datos se usa ampliamente lo que se conoce como Servicio Web (en inglés Web service) que no es más que una colección de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma pueden utilizar los servicios Web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos.

Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se ha creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares.

1.4.8.1 Ventajas de los servicios Web

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado.
- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.
- Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar.

Los servicios Web hacen uso de distintas tecnologías como son XML, SOAP, XSL.

1.4.9 Extensible Markup Language (XML).

XML («lenguaje de marcas extensible»), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML).

Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo, etc.

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

1.4.9.1 Ventajas de XML

- **Comunicación de datos.** Si la información se transfiere en XML cualquier aplicación podría escribir un documento de texto plano con los datos que estaba manejando en formato XML y otra aplicación recibir esta información y trabajar con ella.
- **Migración de datos.** Si trabajamos en formato XML sería muy sencillo mover datos de una *base de datos* a otra.
- **Aplicaciones Web.** Con XML hay una sola aplicación que maneja los datos y para cada navegador podemos tener una hoja de estilo o similar para aplicarle el estilo adecuado.

1.4.10 HTML

HTML es el lenguaje de marcado predominante para la construcción de páginas Web. Permite representar el contenido enriquecido en forma de texto, así como complementar el texto con objetos, ejemplo imágenes. HTML describe la estructura del contenido, y puede manejar la apariencia de un documento y su comportamiento a través de un script (por ejemplo JavaScript) (Mateu, Carles 2004).

El lenguaje HTML pretende continuamente proporcionar funcionalidades más avanzadas para crear páginas más ricas en contenido. Además se ha definido una especificación compatible con HTML, el XHTML (Extensible Hypertext Markup Language) que se suele definir como una versión XML de validación de HTML, proporcionando un esquema XML que permita validar el documento para comprobar si está bien estructurado.

El HTML dinámico (DHTML o Dynamic HTML) no es un estándar definido por el W3C, sino que es un término de mercado que utilizan Netscape y Microsoft para referirse al conjunto de nuevas tecnologías de Web. Dicho conjunto comprende:

- HTML
- Hojas de estilo (CSS)
- JavaScript

Esta conjunción de tecnologías permite ofrecer al usuario interfaces gráficas mucho más ricas y a la vez complejas, controlar formularios de forma más eficiente, brinda un número de facilidades al usuario y proporciona un intercambio más interactivo.

1.4.11 CSS

CSS es un lenguaje de hojas de estilos creado para controlar la presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar contenido y presentación, y es imprescindible para crear aplicaciones Web complejas.

Separar contenido y presentación brinda numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

El lenguaje CSS se utiliza para definir el aspecto de todos los contenidos, el formato de tablas, la separación, el color, tamaño y tipo de letra de titulares y/o textos, la tabulación con la que se muestran los elementos de una lista o menú.

Las ventajas de utilizar CSS son:

- Control centralizado de la presentación de un sitio Web completo con lo que se agiliza de forma considerable la actualización del mismo.
- Los Navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio Web, con lo que aumenta considerablemente la accesibilidad. Por ejemplo, personas con deficiencias visuales pueden configurar su propia hoja de estilo para aumentar el tamaño del texto o remarcar más los enlaces.
- Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre o incluso a elección del usuario. Por ejemplo, para ser impresa, mostrada en un dispositivo móvil, o ser "leída" por un sintetizador de voz.
- El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño (siempre y cuando no se utilice estilo en línea).

1.5 Tecnologías para el desarrollo de aplicaciones Web.

Las tecnologías para el desarrollo de Interfaces Web se dividen en dos grupos: las tecnologías por parte del servidor y las tecnologías por parte del cliente. Cada una de ellas tiene sus propias ventajas y

desventajas, por lo que para desarrollar con éxito una aplicación se deben utilizar conjuntamente y aplicar las que más se ajusten al software en desarrollo.

Las aplicaciones Web se basan en el protocolo HTTP y el lenguaje HTML. El primero proporciona una simple y sencilla implementación de un sistema de comunicaciones que permite enviar cualquier tipo de fichero de forma fácil, simplificando el funcionamiento del servidor y permitiendo que servidores poco potentes atiendan muchas peticiones y reduzcan los costes de despliegue.
El segundo brinda un mecanismo de composición de páginas enlazadas fácilmente, altamente eficientes y de uso muy simple.

1.5.1 Tecnologías por parte del servidor.

Las tecnologías por parte del servidor tienen como elemento común: el control.
El diseñador de un sistema posee control total sobre el servidor, por tanto se puede detallar organizadamente cómo ejecutar un programa del lado del servidor, pues estos programas son invisibles a los cambios que puedan surgir en el lado del cliente, por lo que una aplicación que funciona interactuando con el servidor puede producir páginas a cualquier navegador.

La principal desventaja es la velocidad de ejecución debido a que toda la interacción está relacionada con el servidor, lo que provoca demoras con respecto al tiempo de respuesta o al tiempo de recorrido por la red. Las tecnologías del servidor más utilizadas son: PHP, ASP.NET, JSP y JSF.

1.5.1.1 Personal Home Page (PHP).

PHP es un lenguaje de secuencia de comandos de servidor diseñado específicamente para la Web. Dentro de una página Web se puede incrustar código PHP que se va a ejecutar cada vez que se visite la página. El código PHP es interpretado en el servidor Web y genera código HTML conjuntamente con otro contenido.

Este es un producto de tipo open source (código abierto) por lo que se puede acceder a su código, utilizarlo, modificarlo y sin coste alguno. La comunidad de desarrolladores y programadores de este

lenguaje de programación es muy amplia por lo que continuamente aparecen mejoras en su código de lo cual se deriva la popularidad que hoy día tiene.

Es un lenguaje que está disponible en gran cantidad de sistemas operativos diferentes, desde todos los sistemas operativos de Unix tanto gratuitos como comerciales, hasta las diferentes versiones de Windows.

1.5.1.2 ASP.NET

ASP.NET es una herramienta de desarrollo Web comercializada por Microsoft como parte de su estrategia .NET para el desarrollo Web, con el objetivo de resolver las limitaciones de ASP y posibilitar la creación de software como servicio. Es la plataforma unificada de desarrollo Web que proporciona a los desarrolladores los servicios necesarios para crear aplicaciones Web empresariales.

El desarrollo de aplicaciones utilizando páginas ASP, consiste básicamente en intercalar macros o fragmentos de código dentro de los documentos HTML que sirven para crear las interfaces de usuario de las aplicaciones Web. “Los fragmentos de HTML proporcionan la parte estática de lo que ve el usuario mientras que los fragmentos de código generar la parte dinámica.” (Berzal, Fernando; Cortijo, Francisco J. & Cubero, Juan Carlos)

Una página ASP.NET no es más que un fichero HTML con extensión .aspx al que se le añade algo de código que se puede implementar utilizando distintos lenguajes interpretados. Cuando alguien accede a la página, el Internet Information Server interpreta el código que incluye la página y combina el resultado de su ejecución con la parte estática de la página ASP (la parte escrita en HTML convencional).

Una vez interpretada la página ASP, el resultado final es lo que se envía al navegador Web instalado en la máquina del usuario que accede a la aplicación.

1.5.1.3 Java Server Pages (JSP).

JSP es la forma más común de desarrollar páginas dinámicas en Java. Las JSPs son archivos HTML o XML a los cuales se insertan etiquetas con código Java. Cuando un cliente pide una página JSP del servidor y no se ha ejecutado antes, la página es inicialmente pasada al motor de JSP, el cual compila la página convirtiéndola en servlet, la ejecuta y devuelve el contenido de los resultados al cliente.

La tecnología JSP puede ser analizada desde dos puntos de vista: el diseñador Web y el programador de interfaz.

Un diseñador Web entiende la tecnología JSP como una extensión de HTML para proveer la habilidad de incluir pequeñas porciones de código Java dentro de los archivos HTML. Estas porciones de código Java generan contenido dinámicamente, que es incluido dentro de otros archivos HTML o XML.

Generalmente se crea un concepto erróneo de JSP, llegando a pensar que el código Java insertado en una página JSP es transmitido con el HTML y ejecutado por el cliente (un navegar por ejemplo), sin embargo, una página JSP es traducida a un servlet Java y ejecutada en el servidor.
Las sentencias JSP incluidas en la página se convierten en parte del servlet generado desde la página JSP. El servlet resultante es ejecutado en el servidor, no es nunca visible al usuario.

Para un programador de interfaz, la tecnología JSP es un medio de alto nivel para escribir servlets. En lugar de escribir directamente clases servlet y luego enviar HTML desde los servlets, se escriben páginas HTML con código Java insertado. El ambiente JSP toma la página y la compila dinámicamente. En cualquier momento que el cliente haga una petición a esta página al servidor Web, el servlet que fue generado desde el código JSP es ejecutado, y el resultado es enviado al usuario.

1.5.2 Tecnologías por parte del cliente.

Las tecnologías por parte del cliente se basan fundamentalmente en la rapidez que brindan a los usuarios en comparación con las tecnologías por parte del servidor, pues se validan posibles errores antes que los datos sean enviados al servidor y de esta forma se gana en consistencia y velocidad, aunque es siempre recomendable hacer validaciones en el servidor.

La programación del lado del Cliente depende del navegador o explorador Web, si existe una mínima alteración la aplicación puede no recuperarse al cambio y colapsar.

1.5.2.1 JavaScript.

JavaScript es un lenguaje de programación script que se utiliza fundamentalmente para crear páginas Web dinámicas. Es un lenguaje interpretado, por lo que no es necesario compilar los programas para ejecutarlos. Los programas escritos en JavaScript se pueden probar directamente en cualquier navegador sin necesidad de utilizar herramientas de compilación o cualquier proceso intermediario.

¿Qué es una página Web dinámica?

Es una página Web que implementa funcionalidades, acciones que se ejecutan al pulsar botones u otros elementos HTML, operaciones sencillas o complejas que luego son mostradas al usuario, así como mensajes de aviso, efectos visuales o animaciones.

JavaScript es un lenguaje interpretado, inspirado en Java, que se incluye en los documentos HTML para añadir cierta interactividad a sus contenidos, evitando tener que realizar programación en el servidor.

Es un lenguaje de sintaxis similar a Java, en cuanto a tipos de datos y estructuras de control; sin embargo, al no ser compilado, realiza un control de tipos menos estricto. Por ejemplo, no es necesario declarar las variables, y su tipo puede cambiar durante la ejecución del programa. Todas las referencias entre objetos se resuelven en tiempo de ejecución; Java lo realiza en tiempo de compilación.

Utiliza un gestor automático de memoria dinámica, que reserva espacio para crear objetos y los elimina cuando éstos ya no se utilizan. Está basado en un conjunto predefinido de objetos, que pueden ser extendidos. Sin embargo, no es posible crear nuevas clases, o establecer relaciones de herencia. Permite utilizar funciones, al estilo de los lenguajes de programación orientados a procedimientos.

1.5.2.2 AJAX

AJAX (acrónimo de Asynchronous JavaScript + XML) no es una tecnología en sí mismo. En realidad, se trata de la unión de varias tecnologías que se desarrollan de forma autónoma y que se unen de formas nuevas y sorprendentes.

Las tecnologías que forman AJAX son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

Para el desarrollo de aplicaciones AJAX se requiere de un conocimiento avanzado de todas y cada una de las tecnologías anteriores. Pero ¿cómo funciona una aplicación AJAX?

En una aplicación Web tradicional las acciones del usuario en la página desencadenan llamadas al servidor. Una vez procesada la petición del usuario, el servidor devuelve una página HTML al navegador del usuario.

Esta técnica tradicional para desarrollar aplicaciones Web funciona correctamente, pero no muestra una buena sensación al usuario. Cuando se realizan peticiones continuas al servidor el usuario tiene que esperar que se recargue la página con los cambios solicitados. Si se deben realizar peticiones continuas, la aplicación Web se convierte en algo incómodo.

AJAX se ha desarrollado para mejorar la interacción entre el usuario y la aplicación, evitando las constantes recargas de la página, debido a que el intercambio de información con el servidor se produce en un segundo plano.

Este segundo plano es la creación de un elemento intermedio, una nueva capa que mejora la respuesta de la aplicación, y que no permite que el usuario encuentre una ventana del navegador en blanco, esperando la respuesta del servidor.

Las peticiones HTTP al servidor se transforman en peticiones JavaScript que se realizan al elemento encargado de AJAX. Las peticiones más simples no requieren intervención del servidor, por lo que la respuesta es inmediata. Si la interacción del servidor requiere la respuesta del mismo, la petición se realiza de forma asíncrona mediante AJAX.

En este caso, la interacción del usuario tampoco se ve interrumpida por recargas de página o largas esperas por la respuesta del servidor.

Desde su primera definición, se han creado cientos de aplicaciones basadas en AJAX, que en la mayoría de casos pueden sustituir completamente otras técnicas como Flash y hasta casos de aplicaciones más avanzadas, incluso a complejas aplicaciones de escritorio.

1.5.3 Mapeo de Objetos a Bases de Datos (ORM).

Las bases de datos siguen una estructura relacional. Por este motivo, para acceder a la base de datos como si fuera orientada a objetos, es necesaria una interfaz que traduzca la lógica de los objetos a la lógica relacional. Esta interfaz se denomina “mapeo de objetos a bases de datos” (ORM, de sus siglas en inglés “object-relational mapping”).

Un ORM consiste en una serie de objetos que permiten acceder a los datos y que contienen en su interior cierta lógica de negocio. Una de las ventajas de utilizar estas capas de abstracción de objetos/relacional es que evita utilizar una sintaxis específica de un sistema de bases de datos concreto. Esta capa transforma automáticamente las llamadas a los objetos en consultas SQL optimizadas para el sistema gestor de bases de datos que se está utilizando en cada momento.

De esta forma, es muy sencillo cambiar a otro sistema de bases de datos completamente diferente a mitad del desarrollo de un proyecto. Estas técnicas son útiles por ejemplo cuando se debe desarrollar un

prototipo rápido de una aplicación y el cliente aun no ha decidido el sistema de bases de datos que más le conviene.

La capa de abstracción utilizada encapsula toda la lógica de los datos. El resto de la aplicación no tiene que preocuparse por las consultas SQL y el código SQL que se encarga del acceso a la base de datos es fácil de encontrar. Los desarrolladores especializados en la programación con bases de datos pueden localizar fácilmente el código.

1.5.3.1 Propel.

Propel es una de las mejores capas de abstracción de objetos/relacional disponibles en PHP 5. Es un servicio de objeto persistente y de consulta, lo que significa que Propel provee un sistema para Almacenar objetos en una base de datos y un sistema para búsqueda y restauración de objetos desde una base de datos.

Permite realizar consultas complejas y manipulación de bases de datos sin escribir una sola cláusula SQL. Hace más fácil la escritura de aplicaciones, más fácil de desplegar, y mucho más fácil para migrar si alguna vez la situación lo amerita.

Propel inicialmente implementa el patrón entrada de datos en fila para representar la base de datos. Una entrada de datos en fila brinda objetos que lucen exactamente como el registro en su estructura de registros pero puede ser accedido con los mecanismos regulares de su lenguaje de programación habitual. Todos los detalles de acceso de fuentes de datos están ocultos detrás de esta interfaz.

Sin embargo, Propel también genera las clases para cada tabla que exhibe algunas de las propiedades de la tabla del patrón datos de entrada. Una tabla de entrada de datos almacena todo el SQL para acceder a una sola tabla o vista: selecciones, inserciones, actualizaciones, y eliminaciones. Otro código llama los métodos para todas las interacciones con la base de datos.

En Propel las clases de tabla de entrada de datos son llamadas clasesPeer, mientras que las clases de filas de entrada de datos son llamadas entidad o clases objeto. Como una aplicación, Propel tiene dos componentes principales (y ahora formalmente separados):

- Un motor generador para construir sus clases y archivos SQL (generador-propel).
- Un ambiente de ejecución que proporciona herramientas para construir consultas SQL, ejecutando consultas compiladas, y herramientas para el manejo de conexiones para múltiples bases de datos simultáneamente (propel).

El ambiente de ejecución proporciona una capa de abstracciones y encapsulación de bases de datos reglas lógicas de negocios. Las clases Propel representan la capa modelo, del tradicional MVC, diseñado para encapsular cualquier nivel de validación de dato necesitado por su aplicación.

1.5.4 Metodologías para el desarrollo de software.

El desarrollo de todo software es un proceso realmente complejo, en el cual es necesario el uso de una metodología de desarrollo adecuada, para poder obtener como resultado clientes y desarrolladores satisfechos con el producto. Las metodologías son un proceso disciplinado sobre el desarrollo de software con el objetivo de hacer este más entendible y eficiente.

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software. Dentro de las clasificaciones existentes para las mismas sobresalen dos: las metodologías tradicionales y las ágiles.

Las metodologías tradicionales están basadas en normas provenientes de estándares seguidos por el entorno de desarrollo, son generalmente aplicadas a grandes proyectos e indican paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben de tener.

Además detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla.

Por otra parte se encuentran las llamadas metodologías ágiles que se centran más en la obtención del sistema sin importar cuán documentado esté el mismo, generalmente propone que los involucrados participen el proceso de desarrollo para ir mejorando las funcionalidades del sistema en producción, así como el trabajo en parejas [12].

El uso de una metodología adecuada garantiza también características primordiales en el desarrollo del software como es el tiempo, factor crítico que afecta a todo producto, el cual incide con mayor fuerza cuando no se hace un análisis profundo o se desconoce detalles del sistema a desarrollar. Otra característica que garantiza la metodología es la calidad del producto, la cual es el factor primario tanto para el cliente como para los desarrolladores.

Dentro de las metodologías tradicionales y que puede ser visto como una metodología ágil además, pues es tan configurable como deseemos, se encuentra el RUP haciendo uso del lenguaje UML.

1.5.4.1 Rational Unified Process (RUP).

El Proceso Racional Unificado (Rational Unified Process), es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, la implementación y la documentación de sistemas orientados a objetos. RUP es en realidad un refinamiento realizado por Rational Software del más genérico proceso unificado.

El RUP 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).

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

Inicio: Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema

Elaboración: Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen

Construcción: Se concentra en la elaboración de un producto totalmente operativo y eficiente y el manual de usuario

Transición: Se implementa el producto en el cliente y se entrena a los usuarios. Como consecuencia de esto suelen surgir nuevos requisitos a ser analizados.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

Sus principales características son:

Dirigido por los casos de uso. En el Proceso Unificado los casos de uso se utilizan para capturar los requisitos funcionales y para definir los contenidos de las iteraciones. La idea es que cada iteración tome un conjunto de casos de uso o escenarios y desarrolle todo el camino a través de las distintas disciplinas: diseño, implementación, prueba, etc.

Centrado en la arquitectura. El Proceso Unificado asume que no existe un modelo único que cubra todos los aspectos del sistema. Por dicho motivo existen múltiples modelos y vistas que definen la arquitectura de software de un sistema.

Iterativo e Incremental. RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Por ejemplo, una iteración de elaboración centra su atención en el análisis y diseño, aunque refina los requerimientos y obtiene un producto con un determinado nivel, pero que irá creciendo incrementalmente en cada iteración. Aunque cada iteración tiene que proponerse un incremento en el proceso de desarrollo, todas deben aportar al principal resultado de la fase en la que se desarrolla.

Enfocado en los riesgos. El Proceso Unificado requiere que el equipo del proyecto se centre en identificar los riesgos críticos en una etapa temprana del ciclo de vida. Los resultados de cada iteración, en especial

los de la fase de Elaboración, deben ser seleccionados en un orden que asegure que los riesgos principales son considerados primero.
Además plantea una forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo), pretende implementar las mejores prácticas en Ingeniería de Software con el fin de lograr crear equipos de alto rendimiento que producen proyectos más exitosos y que satisfacen las necesidades del usuario, estas son:

- Administrar requerimientos
- Usar arquitectura de componentes
- Modelar visualmente
- Verificar calidad
- Desarrollar iterativamente
- Controlar los cambios

Es importante mencionar que el ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas:

Disciplina de desarrollo. [13]

- Ingeniería de Negocios: Entendiendo las necesidades del negocio.
- Requerimientos: Trasladando las necesidades del negocio a un sistema automatizado.
- Análisis y Diseño: Trasladando los requerimientos dentro de la arquitectura de software.
- Implementación: Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- Pruebas: Asegurándose que el comportamiento requerido es el correcto y que todo lo solicitado está presente.

Disciplina de soporte. [13]

- Configuración y administración del cambio: Guardando todas las versiones del proyecto.
- Administrando el proyecto: Administrando horarios y recursos.
- Ambiente: Administrando el ambiente de desarrollo.

- Distribución: Hacer todo lo necesario para la salida del proyecto.

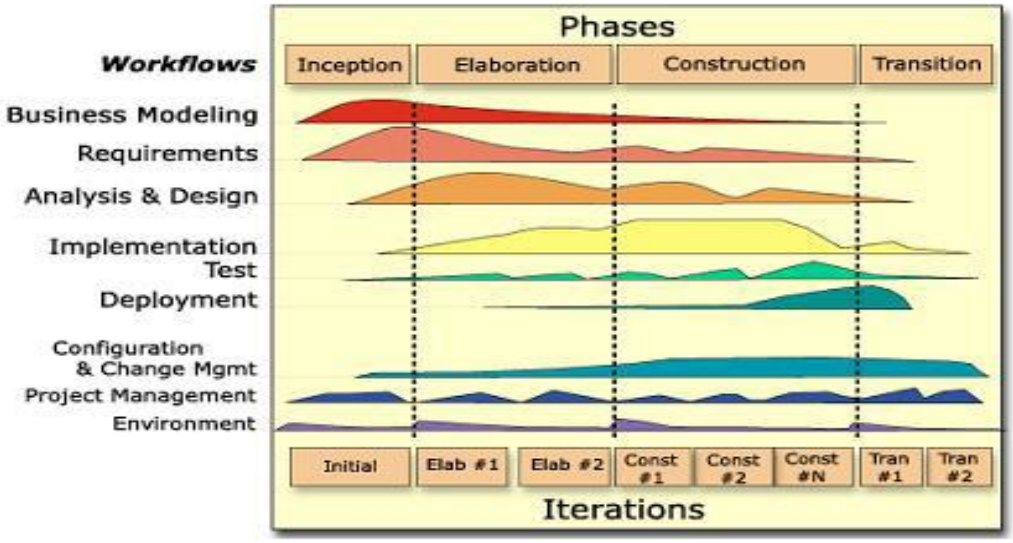

Figura 1.1 Fases e Iteraciones de la Metodología RUP.

Es recomendable que a cada una de estas iteraciones se le clasifique y ordene según su prioridad, y que cada una se convierta luego en un entregable al cliente. Esto trae como beneficio la retroalimentación que se tendría en cada entregable o en cada iteración.

Los elementos del RUP son:

- Actividades: Son los procesos que se llegan a determinar en cada iteración.
- Trabajadores: son las personas o entes involucrados en cada proceso.
- Artefactos: pueden ser documentos, modelos, o elementos de un modelo.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software. [13]

1.5.4.2 Extreme Programming (XP).

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

Esta metodología se basa en la realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuado para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico.

Es la más destacada de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.

Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

Las características fundamentales de XP son:

- **Desarrollo iterativo e incremental**: pequeñas mejoras, unas tras otras.
- **Pruebas unitarias** continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión.
- **Programación por parejas**: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto.
- **Frecuente interacción del equipo de programación con el cliente o usuario**. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
- **Corrección de todos los errores antes de añadir nueva funcionalidad**. Hacer entregas frecuentes.

- **Refactorización del código**, es decir, reescribir ciertas partes del código para aumentar su legibilidad pero sin modificar su comportamiento
- **Propiedad del código compartida**: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto.
- **Simplicidad en el código**: Cuando todo funcione se podrá añadir funcionalidad si es necesario.

La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

La simplicidad y la comunicación son extraordinariamente complementarias. Con más comunicación resulta más fácil identificar qué se debe y qué no se debe hacer. Mientras más simple es el sistema, menos tendrá que comunicar sobre este, lo que lleva a una comunicación más completa, especialmente si se puede reducir el equipo de programadores [14].

1.5.4.3 Crystal Methodologies.

Las metodologías Crystal fueron creadas por Alistair Cockburn Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos.

La familia Crystal dispone un código de color para marcar la complejidad de una metodología: cuanto más oscuro un color, más pesado es el método. Cuanto más crítico es un sistema, más rigor se requiere. El código cromático se aplica a una forma tabular elaborada por Cockburn que se usa en muchas metodologías ágiles para situar el rango de complejidad.

El desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo en equipo definidas.

Los métodos se llaman de esta forma, evocando las facetas de una gema: cada faceta es otra versión del proceso, y todas se sitúan en torno a un núcleo idéntico. Hay cuatro variantes de metodologías: Crystal Clear (Claro como el cristal) para equipos de 8 o menos integrantes; Amarillo, para 8 a 20; Naranja, para 20 a 50; Rojo, para 50 a 100.

Se promete seguir con Marrón, Azul y Violeta. La más exhaustivamente documentada es Crystal Clear (CC) que puede ser usada en proyectos pequeños de categoría D6, aunque con alguna extensión se aplica también en niveles E8 a D10. El otro método elaborado en profundidad es el Naranja, apto para proyectos de duración estimada en 2 años. Como casi todos los otros métodos, CC consiste en valores, técnicas y procesos. [15]

1.5.5 Lenguaje de modelado (UML).

UML es un lenguaje usado para especificar, visualizar y documentar los componentes de un sistema en desarrollo orientado a objetos. Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad, está apoyado en gran medida por el Object Management Group (OMG).

UML ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

Permite modelar sistemas utilizando técnicas orientadas a objetos (OO), así como especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos. Facilita la documentación de todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones, etc.).

UML es independiente del proceso, aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.

Un modelo UML está compuesto por tres clases de bloques de construcción:

Elementos: Son abstracciones de cosas reales o ficticias (objetos, acciones, etc.) que constituyen los bloques básicos de construcción.
Relaciones: Es donde se relacionan los elementos entre sí.
Diagramas: Es la representación gráfica de un conjunto de elementos con sus relaciones. Visualizan un sistema desde diferentes perspectivas.

Los diagramas que forman la base de UML y muestran la manera en que es diseñado un sistema son los siguientes:

- Diagrama de casos de uso.
- Diagrama de clases.
- Diagrama de objetos.
- Diagrama de secuencia.
- Diagrama de colaboración.
- Diagrama de estados.
- Diagrama de actividades.
- Diagrama de componentes.
- Diagrama de despliegue.

1.5.5.1 Modelado de objetos.

En la especificación del UML podemos comprobar que una de las partes que lo componen es un metamodelo formal. Un metamodelo es un modelo que define el lenguaje para expresar otros modelos.

Un modelo OO es una abstracción cerrada semánticamente de un sistema y un sistema es una colección de unidades conectadas que son organizadas para realizar un propósito específico. Un sistema puede ser descrito por uno o más modelos, posiblemente desde distintos puntos de vista.

Una parte del UML define, entonces, una abstracción con significado de un lenguaje para expresar otros modelos (es decir, otras abstracciones de un sistema, o conjunto de unidades conectadas que se organizan para conseguir un propósito).

Lo que en principio puede parecer complicado, no lo es tanto si pensamos que uno de los objetivos del UML es llegar a convertirse en una manera de definir modelos, no sólo establecer una forma de modelo, de esta forma simplemente estamos diciendo que UML, además, define un lenguaje con el que podemos abstraer cualquier tipo de modelo.

El UML es una técnica de modelado de objetos y como tal supone una abstracción de un sistema para llegar a construirlo en términos concretos. El modelado no es más que la construcción de un modelo a partir de una especificación.

Un modelo es una abstracción de algo, que se elabora para comprender ese algo antes de construirlo. El modelo omite detalles que no resultan esenciales para la comprensión del original y por lo tanto facilita dicha comprensión.

Con la creación del UML se persigue obtener un lenguaje que sea capaz de abstraer cualquier tipo de sistema, sea informático o no, mediante los diagramas, es decir, mediante representaciones gráficas que contienen toda la información relevante del sistema.

Un diagrama es una representación gráfica de una colección de elementos del modelo, que habitualmente toma forma de grafo donde los arcos que conectan sus vértices son las relaciones entre los objetos y los vértices se corresponden con los elementos del modelo.

Los distintos puntos de vista de un sistema real que se quieren representar para obtener el modelo se dibuja de forma que se resalten los detalles necesarios para entender el sistema.

1.5.6 Herramientas para el desarrollo de software.

Una vez decidida la metodología a utilizar en el proceso de desarrollo de un software, se precisa de una herramienta adecuada para poder aplicar el método y poder modelar los artefactos correspondientes a las mismas.

Dentro de estas herramientas existe un paquete representativo a nivel mundial que brinda una plataforma de trabajo, que conecta todas sus funcionalidades permitiendo no sólo documentar los artefactos, sino que brinda otras características para administrar un proyecto en toda su totalidad, este es el Rational Rose Enterprise Edition.

El **Rational Suite 2003** permite documentar un producto software brindando una serie de facilidades para la elaboración de los artefactos que propone el Proceso Unificado Racional. Está compuesto a su vez por un grupo de herramientas que se especializan en un aspecto en específico, las cuales son:

Rational Rose Enterprise Edition: Esta herramienta permite modelar visualmente un grupo de artefactos que se generan como parte de la metodología. Dividido en cuatro vistas fundamentales: Vista de Casos de Uso, Vista Lógica, Vista de Componentes y Vista de Despliegue; además, propone un grupo de artefactos predefinidos que facilitan el desarrollo de software con calidad.

Soporta completamente las especificaciones de UML, las cuales nos permite especificar, analizar y diseñar el sistema antes de codificarlo. Esta herramienta nos permite generar código en varios lenguajes de programación a partir del diseño en UML, y proporciona mecanismos para realizar la denominada Ingeniería Inversa, es decir, a partir del código de un programa podemos obtener información sobre su diseño.

Rational Requisite Pro: Esta herramienta gestiona los requerimientos funcionales y no funcionales del sistema a elaborar.

Rational Clear Case: Rational ClearCase proporciona una gestión del ciclo de vida y control de los activos de desarrollo de software.

Rational ClearQuest: proporciona un seguimiento flexible de defectos y cambios en toda la empresa. Soporte completo para consultas con generación de múltiples informes y gráficos.

Rational Soda: Automatiza la documentación del proyecto de software a lo largo de todo el ciclo de vida. Genera documentos mediante la extracción de datos solicitados directamente de los repositorios de datos de herramientas.

Existen dentro de este paquete otras herramientas con otras funcionalidades que gestionan y satisfacen todo el proceso de desarrollo de software.

Conclusiones:

Para lograr un proyecto de desarrollo de software realmente exitoso, debe lograrse la utilización de tecnologías estándares que aseguren la posibilidad de mantenimiento futuro de las aplicaciones.
Además, deben emplearse metodologías de desarrollo ágiles, que son aquellas basadas en una fuerte interacción con el cliente y usuarios, permiten obtener productos adecuados a las necesidades reales, ahorrando esfuerzo y aumentando la satisfacción del usuario final.

Como no existe un paradigma de diseño ni un lenguaje de programación que se ajuste a todas las necesidades, debe escogerse en cada caso la tecnología que mejor satisfaga los requerimientos. Para esto, obviamente, es necesario un conocimiento amplio de las ciencias de la computación.

Para efectuar el diseño del presente sistema, se determinó el uso del RUP para el análisis y diseño, una de las metodologías más utilizadas en la actualidad para el desarrollo de proyectos de software, que a su vez hace uso del Lenguaje Unificado de Modelado (UML) y se utilizó como herramienta de desarrollo el Rational Rose Enterprise Edition.
Como lenguajes de programación por parte del servidor se propone PHP y JavaScript por parte del cliente, con las grandes ventajas que presentan como principales lenguajes de programación Web. Así concluye este capitulo, dándole paso a los siguientes con el afán de posibilitar la búsqueda de una solución a la situación problémica.

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

Introducción

En este capitulo se describen los procesos vinculados con la gestión del expediente del profesor de la Universidad de las Ciencias Informáticas, así como el proceso de categorización docente del profesor. Se identifican y analizan las necesidades del usuario describiéndose los aspectos de la realidad objetiva que serán objeto de automatización, se presenta un estudio del modelo de negocio, además de una propuesta del sistema a desarrollar, especificándose los requerimientos funcionales y no funcionales del mismo.

2.1 Objeto de estudio.

El Departamento de Capital Humano, ubicado dentro de la Dirección de Recursos Humanos, es el responsable de la gestión de la información correspondiente al expediente del profesor, teniéndose en cuenta como gestión: la búsqueda y revisión, actualización, creación y entrega del expediente del profesor.

En este Departamento se encuentra el personal capacitado para la manipulación de dicha información, para lo cual no se cuenta con la existencia de un sistema automatizado que brinde facilidades a la hora de trabajar.

Dentro de este personal capacitado se puede relacionar a:

- Especialista de cuadro: es la persona autorizada para revisar y actualizar el expediente del profesor, crear un nuevo expediente en el caso que sea un nuevo profesor y entregar el expediente cuando un profesor resulta baja de la universidad.
- Miembro de un tribunal: es la persona que integra un determinado tribunal de categorización para evaluar a un profesor y darle una categoría docente.

Además, cada profesor tiene la posibilidad de solicitar la revisión de su propio expediente así como también la actualización del mismo.

Los procesos de gestión de la información del expediente del profesor, resulta ser el objeto de estudio de esta investigación.

Los procesos relacionados con esta gestión son:

- Matricular un profesor
- Categorizar un profesor
- Actualizar el expediente del profesor
- Dar baja al expediente del profesor

2.1.1 Situación actual. Su análisis crítico.

Actualmente, el mayor problema existente que impide una viable gestión de la información en el Departamento de Capital Humano, es la carencia de recursos informáticos, encabezando esta lista una aplicación informática que de seguro mejoraría el proceso de gestión.

Cada vez que un profesor desea revisar su expediente, debe presentarse personalmente en el Departamento, esperar allí a ser atendido por el personal autorizado para buscar lo que el profesor desea, luego si el profesor posee algún nuevo documento que deba ser reflejado en su expediente, se le agrega al contenido del mismo. En el caso que el profesor sea nuevo en la universidad, debe solicitar que le sea creado su expediente.

Si un miembro de un tribunal de categorización desea consultar el expediente de un profesor que este asignado a su tribunal, debe presentarse de igual forma y hacer la solicitud correspondiente; este mismo proceso debe realizarlo cualquier directivo que necesite consultar el expediente de un profesor determinado de su respectiva facultad.

El conflicto surge, cuando más de una persona necesita consultar el mismo expediente, pues al no existir copias del mismo, solo puede utilizarse por una persona a la vez, lo que trae como consecuencia que se obstaculicen el resto de los procesos que esperan por este expediente.

Además, durante el tiempo que transcurre entre una actualización y otra, por parte de un profesor, que generalmente espera a tener varios documentos para presentarse en el Departamento de Capital Humano y realizar esta acción, no se podrá obtener del expediente del profesor una información ciertamente actualizada.
Esta situación problémica solo puede ser enmendada con la creación de un sistema automatizado que agilice y facilite las funcionalidades implícitas en el proceso de gestión de la información del expediente del profesor.

2.2 Objeto de automatización.

El objeto de automatización radica en el proceso de gestión del expediente docente del profesor en la Universidad de las Ciencias Informáticas, de las actividades que involucra la gestión del mismo se propone automatizar:

Matricular un profesor: Cuando un profesor ya sea recién graduado o con previa trayectoria docente se incorpora a la universidad este debe remitirse al Departamento de Capital Humano donde se le solicitan una serie de documentos y datos necesarios para llevarle a cabo la matricula, se procede a crear su expediente.

Consultar un expediente: El profesor en cualquier momento puede pedir consultar su expediente docente para de esta manera asegurarse que esta actualizado y que el mismo recoge toda la información necesaria referente a su trayectoria docente.

Actualizar un expediente: Cuando un profesor reúne una cierta cantidad de información nueva, referente a su trayectoria docente, el mismo se presenta en el Departamento de Capital Humano para que esta información se le sea incorporada al expediente ya creado previamente, puede hacerse también en el caso que el profesor necesite modificar sus datos personales.

Dar de baja a un expediente: Es cuando el profesor va a causar baja del centro, luego de tramitar correctamente la misma en la Dirección de Recursos Humanos se presenta en el Departamento de Capital Humano con todos los documentos que indiquen que puede causar baja, para tramitar la misma,

se le da de baja y sus datos personales pasan a una lista donde aparecen todos los profesores que han brindado su servicio en el centro.

Solicitar Categorización: Una vez abierta la convocatoria de categorización por los diferentes departamentos docentes, el profesor puede optar por una de las categorías docentes definidas previamente llenando un modelo de solicitud de categorización y entregándolo al departamento de Capital Humano para su posterior análisis.

Modificar Solicitud de Categorización: Es después de solicitada la categorización correspondiente, el profesor puede modificar la misma en caso de desea optar por una categoría diferente a la que anteriormente había solicitado en el modelo.

Canelar la Solicitud de Categorización: Es cuando el profesor decide no categorizarse en la convocatoria en curso, lo informa al Departamento de Capital Humano y se le cancela la solicitud, o si en el proceso de categorización el tribunal decide que no cumple con los requisitos establecidos para la categoría por la cual opta o no aprueba alguno de los ejercicios correspondientes, se le cancela las solicitud de categoría.

Categorizar al profesor: Previa solicitud de una categoría docente el tribunal se reúne el tribunal de categorización para revisar los expedientes de los solicitantes y verificar que cumple con los requisitos establecidos para dicha solicitud, en caso que sea positivo el proceso, el profesor procede a realizar los ejercicios correspondientes a la categoría por la cual opta y se le otorga la nueva categoría en caso de obtener una calificación satisfactoria luego de la realización de los mismos.

Archivar documentos: Todos los documentos como: títulos, certificaciones, diplomas, entre otros, que avalen el desempeño y la trayectoria del profesor durante su vida como profesional docente, deben ser guardados en el expediente docente del profesor.

2.3 Información que se maneja en el negocio.

Expediente del profesor: Es el modelo mas importante del área, en el se registran todos los datos referentes a la trayectoria docente del profesor.

Modelo de Solicitud de Categoría Docente: En este modelo se registran los datos necesarios para la solicitud de una nueva categoría docente por parte del profesor, cuando un profesor estima que puede optar por una nueva categoría docente superior a la que tiene, llena este modelo y lo entrega en el área de Capital Humano para su posterior análisis.

Acta de entrega del expediente del profesor: Este modelo refleja la información y los datos necesarios referentes a la entrega oficial del expediente del profesor en el momento en que va a causar baja del centro.

Títulos o certificaciones: Constituyen documentos que el profesor tiene registrado en su expediente. Ejemplo: El Titulo Universitario, Certificados que acrediten los diferentes cursos de posgrados que el profesor a cursado o impartido, certificados de reconocimientos, certificados de: publicaciones, investigaciones, asesoría o tutoría a tesis de pregrado, maestría, doctorado.

Actas con las propuestas de otorgamiento de Categoría: Estas son las actas mediante las cuales el Tribunal de Categorización notifica si el profesor procede o no a obtener la categoría docente por la cual opto en su Modelo de Solicitud de Categoría. Existen 6 actas (Acta de Revisión, Acta de Conclusiones, Acta de la Clase, Acta del Ejercicio de Idioma, Acta del Ejercicio de PSCT, Acta de Disertación)

2.4 Propuesta de Sistema

Para darle solución a la situación problémica se ha decidido diseñar un sistema para la web que agilice y mejore el tratamiento de la información en el proceso de gestión del expediente del profesor en la Universidad de la Ciencias Informáticas, una aplicación del tipo Cliente-Servidor a la cual se tendrá acceso desde la intranet de la Universidad, previa autenticación por el dominio UCI.

El personal autorizado del Departamento de Capital Humano podrá manejar los datos y documentos referentes al expediente del profesor de manera rápida y eficiente, consultar la información, modificarla o darle baja un expediente en caso de ser necesario. Así como mantenerlo actualizado constantemente, ya

sea por los datos propios del profesor o acerca del proceso de categorización en el que se pueda encontrar inmerso el mismo, pues este es un proceso que suele durar varios meses.

Además todo profesor en la universidad, previa autenticación al sistema podrá en todo momento y desde cualquier punto de la red del centro, acceder a la información de su expediente docente, para de esta manera tener noción del estado en que se encuentra la información sobre su trayectoria docente en el centro.

2.5 Modelo del Negocio.

La disciplina de modelado de negocio es la primera que propone RUP dentro del ciclo de desarrollo de un software, tiene su mayor peso durante la fase de inicio debido a que permite conocer los procesos existentes actuales de cualquier entidad o empresa para la que se vaya a desarrollar el sistema. Es en este flujo de trabajo donde se conocen a fondo cómo son iniciadas cada una de las actividades de un sector determinado, y a través del modelado de estos procesos se obtiene una visión más amplia del negocio existente.

RUP propone, que para esta disciplina se modele el negocio siempre y cuando los procesos sean fácilmente identificables, en el caso de que no puedan percibirse se realizaría un modelo de dominio, que englobaría los principales conceptos que sean encontrados para a partir de este modelo, pasar al flujo de trabajo de requerimientos.

En el caso que nos embarga utilizamos modelo de negocio, para establecer una abstracción de la organización y con el objetivo de lograr un buen entendimiento del negocio para la construcción de un sistema correcto, a través del desarrollo de un conjunto de artefactos que permiten modelar íntegramente el mismo.

2.5.1 Actores del Negocio

Un actor del negocio es el principal beneficiado de los procesos existentes, es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo

que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados.

Actores del Negocio	Justificación
Profesor	Es aquella persona que interactúa con la entidad Departamento de Capital Humano para que se le confeccione el expediente, o en caso de que el mismo exista, consultarlo o actualizarlo, así como optar por una categoría docente y ser procesado para obtener de la misma.

Tabla 2.1 Actores del negocio

2.5.2 Trabajadores del Negocio

Trabajadores del Negocio	Justificación
Especialista de Cuadros	Trabajador del Departamento de Capital Humano encargado de llevar a cabo el proceso de gestión del expediente del profesor.
Miembro del Tribunal de Categorización.	Individuo que integra un Tribunal de Categorización encargado de llevar a cabo el proceso de categorización docente.

Tabla 2.2 Trabajadores del Negocio

2.5.2 Diagrama de Casos de Uso del Negocio.

Los diagramas de casos de uso explican gráficamente el conjunto de casos de uso del negocio, los actores y la relación entre ambos.

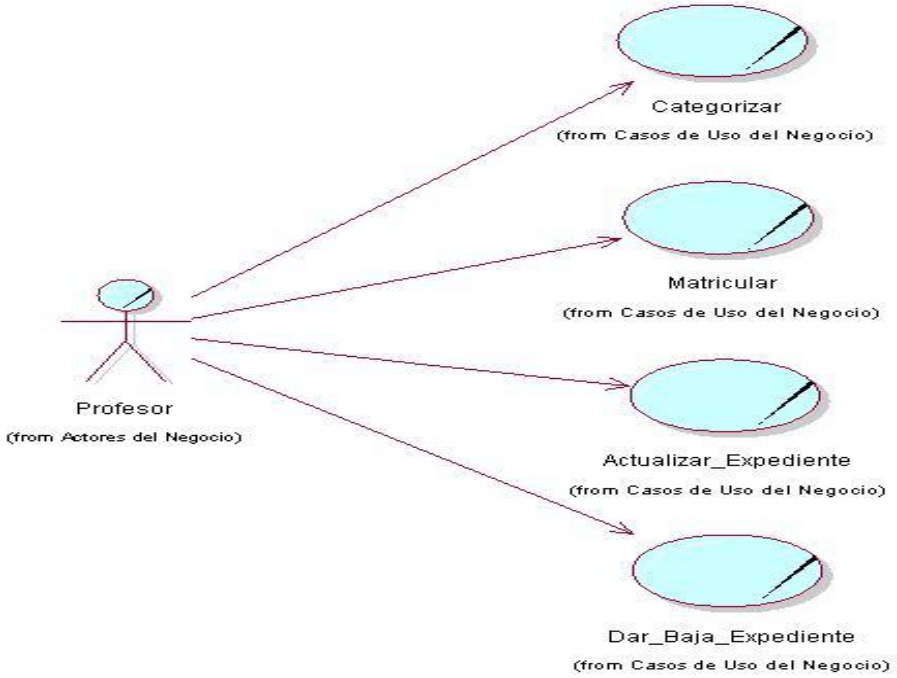

Figura 2.1 Diagrama de Casos de Uso del Negocio.

2.5.3 Descripción textual de los Casos de Uso del Negocio

Los Casos de Uso del Negocio son procesos que ocurren en el negocio, una detallada descripción de los mismos ayuda a una mejor comprensión del proceso de desarrollo de software por parte del equipo de desarrollo y el cliente.

Descripción textual Caso de Uso de Negocio: Matricular.

Caso de Uso del Negocio	Matricular
Actores	Profesor(inicia)
Trabajadores	Especialista de Cuadros
Propósito	Mantener el control y la integridad de la información referente a la trayectoria docente de los profesores de la Universidad de las Ciencias Informáticas. Así como el acceso de manera rápida y eficiente a la misma.
Resumen El Caso de Uso inicia cuando el profesor se presenta en el departamento de capital humano para solicitar su matricula, se registran los datos y documentos necesarios, se guarda el expediente en el registro de expedientes, el profesor se retira y finaliza el Caso de Uso.	
Precondiciones	El expediente no debe estar creado anteriormente. No debe haberse matriculado anteriormente.
Casos de Uso Asociados	
Flujo normal de los eventos	
Acción del Actor	**Respuesta del negocio**
1. EL profesor solicita la matricular. 3. EL profesor entrega los documentos.	2. La especialista de cuadros le solicita al profesor los documentos necesarios. 4. La Especialista de Cuadros verifica que

	estén todos los documentos necesarios. 5. Si están todos los documentos, el Especialista de Cuadros le solicita los datos necesarios.
6. El profesor le entrega los datos.	7. La Especialista de Cuadros recibe los datos y verifica que estén todos. 8. El Especialista de Cuadro crea un nuevo expediente. 9. El Especialista de Cuadros informa al profesor que el expediente fue creado y la información fue guardada con éxito.
10. El profesor se retira y finaliza el Caso de Uso.	

Flujos Alternos
4.1. Si no están todos los documentos el Especialista de Cuadro decide no continuar la matricula y finaliza el caso de uso. 7.1. Si no están todos los datos el Especialista de Cuadro solicita nuevamente los datos y regresa al 7 del flujo normal.

Poscondiciones	Queda registrado un nuevo expediente en el Departamento de Capital Humano
Prioridad	Crítico

Mejoras	La automatización de las actividades llevadas a cabo en este proceso les permitirá a los Especialistas de Cuadro crear un expediente con mayor facilidad y rapidez.

Tabla 2.3 CUN Matricular

Descripción textual Caso de Uso de Negocio: Actualizar Expediente

Caso de Uso del Negocio	Actualizar Expediente
Actores	Profesor(inicia)
Trabajadores	Especialista de Cuadros
Propósito	Actualizar la información referente a la trayectoria docente del profesor de la Universidad de manera más rápida y eficiente.
Resumen El Caso de Uso inicia cuando el profesor se presenta en el departamento de capital humano para realizar cambios en la información de su expediente docente, se registran los nuevos datos y/o documentos, se guarda el expediente en el registro de expedientes, el profesor se retira y finaliza el Caso de Uso.	
Precondiciones	El expediente debe estar creado anteriormente.
Casos de Uso Asociados	
Flujo normal de los eventos	
Acción del Actor	**Respuesta del negocio**
1. EL profesor solicita actualizar su expediente.	2. La especialista de cuadros le solicita al profesor los datos y/o documentos que se

3. EL profesor entrega los datos y/o documentos. 7. El profesor se retira y finaliza el caso de uso.	desean actualizar. 4. La Especialista de Cuadros busca el expediente. 5. La especialista de cuadros actualiza el expediente. 6. La especialista de cuadros informa que el expediente fue actualizado correctamente.
Flujos Alternos	
5. Si el expediente no existe la especialista de cuadros le informa al profesor y termina el caso de uso.	
Poscondiciones	Queda actualizado el expediente en el Departamento de Capital Humano
Prioridad	Crítico
Mejoras	

Tabla 2.4 CUN Actualizar_Expediente.

Descripción textual Caso de Uso de Negocio: Dar_Baja_Expediente.

Caso de Uso del Negocio	Dar Baja Expediente
Actores	Profesor(inicia)
Trabajadores	Especialista de Cuadros
Propósito	Poder eliminar el expediente de un profesor de la Universidad de manera más

	rápida y eficiente.
Resumen El Caso de Uso inicia cuando el profesor se presenta en el departamento de capital humano para solicitar la baja de su expediente docente, se guarda el expediente en el registro de expedientes, el profesor se retira y finaliza el Caso de Uso.	
Precondiciones	El expediente debe estar creado anteriormente.
Casos de Uso Asociados	
Flujo normal de los eventos	
Acción del Actor	**Respuesta del negocio**
1. EL profesor solicita dar baja a su expediente.	
	2. La especialista de cuadros le solicita al profesor los documentos y datos necesarios para realizar la matricula.
3. EL profesor entrega todos los documentos y datos.	
	4. La Especialista de Cuadros recibe y verifica que estén todos los documentos y datos. 5. El especialista de cuadros procede a guardar los datos personales del profesor. 6. El especialista de cuadros le entrega el expediente al profesor.
7. El profesor se retira y finaliza el Caso de	

Uso.	
Flujos Alternos	
4.1. Si no están todos los documentos y datos el especialista de cuadros le informa al profesor que faltan documentos o datos para tramitar la baja y finaliza así el caso de uso.	
Poscondiciones	Queda eliminado el expediente en el Departamento de Capital Humano
Prioridad	Crítico
Mejoras	

Tabla 2.5 CUN Dar_Baja_Expediente.

Descripción textual Caso de Uso de Negocio: Categorizar.

Caso de Uso del Negocio	Categorizar
Actores	Profesor(inicia)
Trabajadores	Especialista de Cuadros, Tribunal de Categorización.
Proposito	Llevar a cabo el proceso de categorización docente del profesor.

Resumen El Caso de Uso inicia cuando el profesor se presenta en el departamento de Capital Humano para presentar su solicitud de categorización, el Especialista de Cuadros guarda la solicitud en el expediente, luego le entrega el expediente al Tribunal de Categorización, este se reúne para verificar que cumpla con todos los requisitos para dicha categoría docente y en caso positivo procede a realizar los ejercicios correspondientes a la categorización, luego el Tribunal de Categorización decide si obtiene el profesor la categoría por la cual opta, de ser así se actualiza el expediente del profesor con la nueva categoría y las actas correspondientes a los ejercicios que realizó.	
Precondiciones	El profesor debe solicitar la categoría docente.
Casos de Uso Asociados	Actualizar_Expediente.
Flujo normal de los eventos	
Acción del Actor	**Respuesta del negocio**
El profesor entrega la solicitud de categoría.	
	2. La especialista de cuadros guarda la solicitud en el expediente.
	3. La Especialista de Cuadros asigna el expediente a un tribunal de categorización.
	4. El Tribunal de Categorización revisa el expediente.
	5. El tribunal de Categorización confecciona el Acta de Revisión.
6. El profesor realiza los ejercicios	

correspondientes a la categoría por la cual esta optando.	7. El Tribunal de Categorización realiza el Acta de ejercicio correspondiente y notifica al Especialista de Cuadros el ejercicio que fue realizado. 8. El Tribunal de Categorización realiza el Acta de Conclusiones. 9. El Tribunal de Categorización le otorga la categoría solicitada. 10. El Tribunal de Categorización le entrega al Especialista de Cuadros el expediente con las Actas correspondientes (Revisión, Ejercicio, Conclusiones). 11. El Especialista de cuadros confecciona la certificación de categoría y se la entrega al profesor. 12. El Especialista de cuadros actualiza el expediente del profesor con la nueva categoría y finaliza en Caso de Uso.

Flujos Alternos
5.1. Si el profesor no cumple con los requisitos para dicha categoría, el Tribunal da Categorización le entrega al Especialista de Cuadros el expediente del profesor junto con el Acta de Revisión y Conclusiones.

5.2 El especialista de Cuadros Actualiza el Expediente y finaliza el Caso de Uso. 7.1 Si el profesor no aprueba el o los ejercicios que debe realizar, el Tribunal de Categorización Realiza el Acta de Conclusiones. 7.1 Tribunal de Categorización le entrega al Especialista de Cuadros el expediente del profesor junto con las Actas: Revisión, Ejercicios, Conclusiones. 7.2 El especialista de Cuadros Actualiza el Expediente y finaliza el Caso de Uso. 8.1 Si al confeccionar el Acta de Conclusiones se califica de no aprobado al profesor, el Tribunal de Categorización entrega al Especialista de Cuadros el expediente, junto con las actas de los ejercicios correspondientes, el acta de revisión y el acta de conclusiones. 8.2 El Especialista de Cuadros Actualiza el Expediente con los documentos que le fueron entregados, y finaliza el Caso de Uso.	
Poscondiciones	Queda Actualizado el expediente en el Departamento de Capital Humano
Prioridad	Crítico
Mejoras	

Tabla 2.6 CUN Categorizar.

2.5.4 Diagramas de actividades.

Un diagrama de actividades describe gráficamente como están estructurados los flujos básicos o alternativos de los casos de uso del negocio. Es un grafo que en dependencia de la actividad que realiza posee un estado determinado, representando la ejecución de una sentencia de un procedimiento. Los diagramas de actividades de los casos de uso del negocio encuentran en el Anexo 1.

2.5.5 Modelo de Objetos

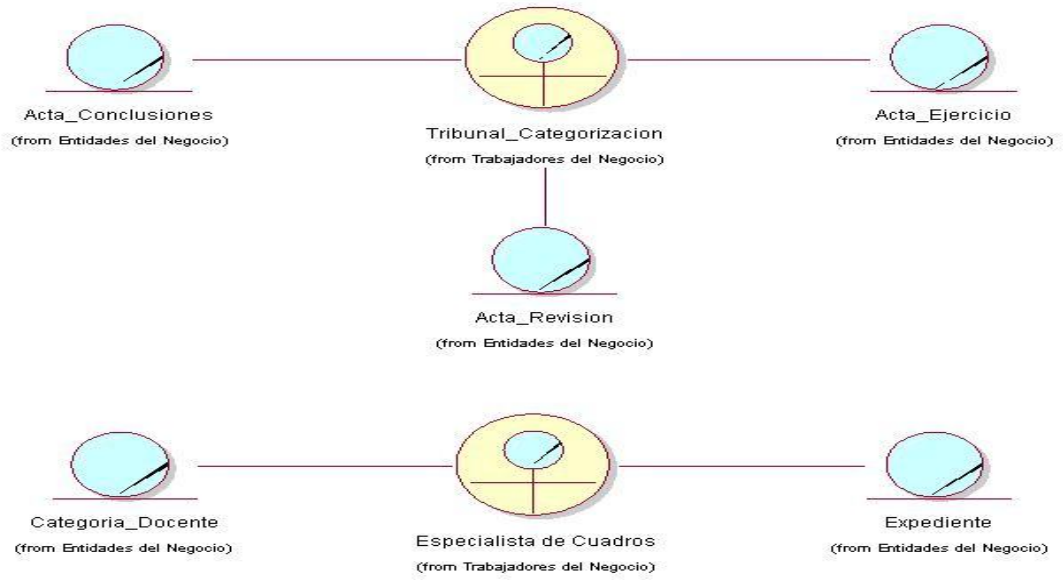

Figura 2.2 Modelo de Objeto del Negocio.

2.6 Especificación de los requerimientos del software.

La especificación de los requerimientos de un software, es el proceso de indagar sobre lo que se debe construir, y de esta forma guiar el desarrollo hacia el sistema correcto. Con una descripción suficientemente buena de los mismos podrá llegarse a un acuerdo entre el cliente (incluyendo los usuarios) y los desarrolladores sobre que debe y que no debe hacer el sistema.

2.6.1 Requerimientos funcionales.

Los requerimientos funcionales especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física, de manera que especifiquen el comportamiento de entrada y salida del sistema. A partir de los procesos de negocio estudiados y las actividades a automatizar identificadas, se pueden definir los siguientes requerimientos funcionales:

R1. Autenticar.

- R1.1 Verificar usuario.
- R1.2 Verificar contraseña.
- R1.3 Definir nivel de acceso al sistema.

R2. Gestionar Usuario.

- R2.1 Adicionar usuario.
- R2.2 Asignar Rol.
- R2.3 Modificar Usuario
- R2.4 Eliminar Usuario.

R3 Matricular profesor.

- R3.1 Registrar datos personales.
- R3.2 Registrar documentos.

R4 Dar baja al expediente de un profesor.

R5 Gestionar Datos Personales y Familiares.

- R5.1 Modificar Datos Personales y Familiares.
  - R5.1.1 Modificar Datos Personales.
  - R5.1.2 Modificar Datos Familiares.
- R5.2 Consultar Datos Personales y Familiares.
- R5.3 Registrar Datos Personales y Familiares
  - R5.3.1 Registrar Datos Personales.
  - R5.3.2 Registrar Datos Familiares.

R6 Gestionar Datos Sociales.

- R6.1 Modificar Datos Sociales.
- R6. Registrar Datos sociales.
- R6.3 Consultar Datos Sociales.
- R6.4 Gestionar Aspectos Relativos a la Salud.
  - R6.4.1 Registrar Aspectos Relativos a la Salud.
  - R6.4.2 Modificar Aspectos Relativos a la Salud.
  - R6.4.3 Consultar Aspectos Relativos a la Salud.

R7 Gestionar Datos del Currículum.

R7.1 Modificar Datos del Currículum.

R7.2 Registrar datos y documentos del currículum

R7.3 Consultar Datos del Currículum.

R7.4 Adicionar Documentos relacionados con el currículum del profesor (Certificados, Diplomas, Títulos).

R8. Gestionar Solicitud de Categorización.

R8.1 Adicionar Solicitud de Categorización.

R8.2 Modificar Solicitud de Categorización.

R8.3 Cancelar Solicitud de Categorización.

R9. Actualizar Categoría Docente.

R9.1 Registrar Actas con la propuesta de otorgamiento de categoría.

R9.2 Registrar Documento de Certificación de Categoría.

R10. Permitirle al Profesor mostrar su Expediente personal.

R11 Mostrar el expediente de un profesor dado su nombre (En caso de haber más de un profesor con el mismo nombre, mostrar un listado con todos los expedientes).

R12 Mostrar un listado con los expedientes de los profesores dado el nombre, primer apellido, sexo, número de carnet de identidad y el solapín.

R13 Mostrar un listado con los expedientes de los profesores dada la facultad y el departamento.

R14 Mostrar el listado de todos los profesores que tienen que ratificar su categoría docente dado un año.

R15 Mostrar el listado de todos los profesores que tienen que ratificar su categoría docente dada la facultad el año.

R16 Mostrar el listado de los profesores que solicitaron una categoría docente determinada, para una especialidad dada.

R17 Mostrar el listado de los profesores con una categoría docente determinada.

R18 Mostrar el listado de los profesores con una categoría docente determinada dada la facultad y la categoría docente.

R19 Mostrar el listado de los profesores con una categoría científica determinada.

R20 Mostrar el listado de todos los profesores con una categoría científica determinada dada la facultad y categoría científica.

R21. Subir documentos por parte del profesor.

R22. Aprobar los documentos que el profesor ha subido al servidor.

R23. Controlar proceso de categorización.

R23.1 Notificar en cada momento los ejercicios que se realizan en el proceso de categorización.

R23.2 Revisar los expedientes de los profesores dada la especialidad en la que solicitan una categoría docente.

R24 Permitir consultar el estado del proceso de categorización por parte del tribunal de categorización correspondiente.

R25 Permitir consultar el estado del proceso de categorización por parte del especialista de cuadros.

R26 Permitir consultar el estado del proceso de categorización por parte del profesor.

R27 Gestionar Tribunal de Categorización.

R27.1 Registrar Tribunal.

R27.2 Modificar Tribunal.

R27.3 Eliminar Tribunal.

R28 Separar profesores por tribunales de categorización.

2.6.2 Requerimientos no funcionales.

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Muestran las características que hacen al producto atractivo, usable, rápido o confiable. No alteran la funcionalidad del producto, esto quiere decir que los requerimientos funcionales se mantienen invariables sin importarle con que propiedades o cualidades se relacionen.

Requerimientos de apariencia o interfaz externa.

La aplicación deberá desarrollarse en un ambiente Web. Debiendo presentar una interfaz sencilla, amigable y de fácil uso, para que pueda ser utilizada sin mucho entrenamiento por los usuarios, teniendo en cuenta que algunos de éstos pueden no ser especialistas en informática, sin dejar de poseer un ambiente profesional.

Requerimientos de usabilidad.

El sistema deberá poder ser usado por todos los profesores de la UCI, quienes podrán tener acceso a la información correspondiente a sus expedientes docentes acorde con la necesidad y permisos que posean para ello. También podrán actualizarse los datos propios al profesor por parte del personal autorizado.

Se realizaran solicitudes de categorización por parte de los profesores, así como su cancelación si se requiere de la misma y durante el proceso de categorización se brindara información de cómo marcha el proceso. Además, aquellos trabajadores autorizados por el sistema obtendrán detallada información acerca de un determinado profesor que se desee consultar en caso de estarse desarrollando algún proceso de investigación en el cual este inmiscuido dicho profesor.

El administrador del sistema podrá principalmente agregar nuevos usuarios y roles al sistema si así se requiere. Definiéndose así niveles de usuario que se diferenciarán en las opciones que el sitio les posibilite. Basta con que los mismos cuenten con conocimientos básicos en el manejo de la computadora y el ambiente Web a grandes rasgos.

Requerimientos de rendimiento.

Para garantizar una aplicación con óptimas funcionalidades, la misma debe garantizar la rapidez de respuesta ante las solicitudes de los usuarios. El procesamiento de la información y el rápido acceso a sus páginas debe realizarse con gran velocidad. La eficiencia del producto estará determinada en el aprovechamiento de los recursos que se disponen en el modelo Cliente/Servidor y la velocidad de las consultas a la base de datos.

Requerimientos de soporte

Se le debe dar mantenimiento periódico a los servidores de bases de datos controlando la integridad de la información.

Requerimientos de seguridad y privacidad

La información debe transmitirse de manera segura, se debe garantizar la seguridad a todos los niveles (Interfaz, negocio y acceso a datos) restringiendo las funcionalidades mediante roles de usuarios garantizando que la información sea accesible al usuario autorizado.

Requerimientos de confiabilidad

La información debe transmitirse a través de canales seguros. Se debe chequear la integridad de los datos.

Requerimientos de ayudas y documentación en línea

Se debe brindar una interfaz amigable que explique las diferentes funcionalidades con que cuenta el sistema de manera rápida, además los manuales de usuario y toda la documentación actualizada de cada módulo de la aplicación.

Requerimientos de hardware.

Requerimientos para una estación de trabajo: 256Mb RAM (Recomendado 512Mb), 1GHz, 10Gb HDD. Requerimientos para un *servidor*: 512Mb RAM (Recomendado 1Gb RAM o superior), 1GHz o superior, 60Gb HDD.

Requerimientos de software.

El sistema debe correr en sistema operativo Windows. El cliente solo deberá disponer de un navegador Web (IE o Firefox).

Restricciones en el diseño y la implementación.

El diseño de la aplicación fue realizado con fines de obtener un sistema a desarrollar en entorno Web, para lo cual deberá ser implementado utilizando como lenguaje de programación del lado del cliente JavaScript y por lado del servidor PHP. Para garantizar una mejor documentación del sistema, así como el uso de última tecnología, se utiliza UML para realizar el análisis y el diseño del sistema. Como herramienta de apoyo a este lenguaje se utiliza el Racional Rose Enterprise Edition.

2.7 Modelado del sistema.

El modelado del sistema consiste en estructurar o enmarcar los respectivos requerimientos funcionales y no funcionales a través de casos de uso que reflejen las funcionalidades futuras del software a desarrollar.

2.7.1 Actores del sistema

Un actor del sistema está representado por un individuo que interactúa desde afuera del mismo teniéndose en cuenta el tipo de usuario que sea el mismo.

Descripción de los actores del sistema

Actor	Descripción
Usuario	Es aquel persona autorizada a entrar al sistema y según su rol podrá llevar a cabo acciones dentro del mismo
Administrador	Usuario del sistema que podrá insertar, modificar o eliminar usuarios y asignarles un determinado rol
Profesor	Usuario del sistema que podrá revisar su expediente personal, solicitar una categoría docente, así como gestionar sus datos familiares y referentes a su salud. Además de subir los documentos que vaya acumulando durante su trayectoria laboral y que son de vital importancia para su currículum
Revisor	Usuario del sistema que representa una especialización del profesor, al que se le permite revisar los expedientes de los profesores.
Miembro del Tribunal de Categorización	Usuario del sistema que representa una especialización del profesor y que forma parte de un tribunal de categorización en un momento determinado el cual tiene la tarea de controlar el proceso de categorización docente.

Especialista de Cuadros	Usuario del sistema encargado de llevar a cabo el proceso de gestión del expediente del profesor y que representa una especialización del revisor debido a que puede llevar a cabo la acción de revisar expedientes.

Tabla 2.7 Descripción de los actores del sistema

2.7.2 Diagrama de casos de uso del sistema.

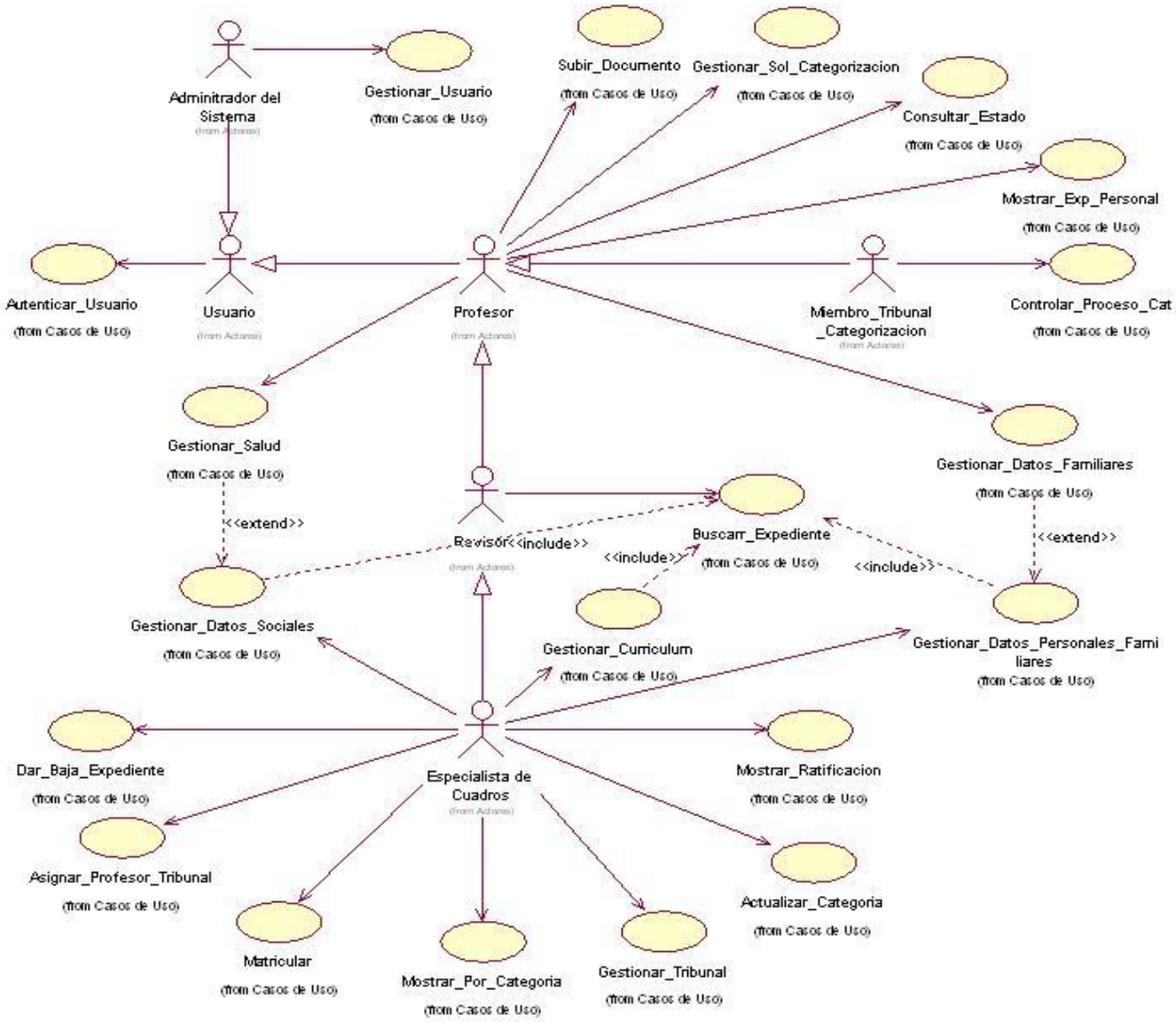

Figura 2.3 Diagrama de Casos de Uso del Sistema

2.7.3 Descripción textual de los casos de uso del sistema.

Caso de Uso del Sistema Matricular

Caso de Uso:	Matricular.
Actores:	Especialista de Cuadros (inicia).
Resumen:	El caso de uso comienza cuando el Especialista de Cuadros solicita matricular un profesor, el especialista de cuadros introduce los datos que son de carácter obligatorio para realizar la matrícula, se crea un nuevo expediente donde se registra la información y finalmente el sistema brinda la posibilidad de actualizar dicho.
Referencias	R3
Casos de Uso Asociados	---
Precondiciones	El Especialista de Cuadros se ha autenticado correctamente en el sistema. El Especialista de Cuadros ha solicitado matricular un profesor.
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
	1. El Sistema muestra la interfaz correspondiente y brinda la posibilidad de registrar los datos y documentos necesarios para realizar la matrícula.
2. El Especialista de Cuadros introduce los datos del profesor incluyendo los documentos necesarios y que son de carácter obligatorio.	3. El Sistema verifica que no existan errores en la información introducida.
	4. El sistema verifica que el

	profesor no se encuentre matriculado previamente.
	5. El Sistema registra la información.
	6. El Sistema notifica que los datos fueron registrados correctamente y brinda la posibilidad de mostrar el expediente del profesor en caso de querer actualizar el mismo.
7. Si el Especialista de Cuadros desea actualizar el expediente del profesor ver Casos de Uso, Gestionar_Datos_ Personales_Familiares, Gestionar_Datos_Sociales, Gestionar_Currículum.	

Flujos Alternos

Línea 3.

3.1 El sistema detecta errores en la información introducida y brinda la posibilidad de corregirlos.

3.2 El Especialista de Cuadros corrige los errores. Regresar a la línea 4.

Línea 4.

4.1 El Sistema comprueba e informa que el profesor ya se encuentra matriculado.

4.2 El Especialista de Cuadros presiona aceptar. Finaliza el Caso de Uso.

Línea 5.

5.1 Muestra un mensaje de error de conexión con la base de datos.

5.2 El Especialista de Cuadros acepta el error y finaliza el Caso de Uso.

* El administrador del sistema presiona cancelar y finaliza el Caso de Uso.	
Poscondiciones.	Queda matriculado un profesor.

Tabla 2.8 CUS Matricular.

Caso de Uso del Sistema Buscar Expediente.

Caso de Uso:	Buscar_Expediente.
Actores:	Revisor (inicia)
Resumen:	El Caso de Uso inicia cuando el Revisor selecciona la opción de buscar expedientes, se muestra la interfaz para dicha operación, el Revisor introduce los parámetros de búsqueda necesarios, se muestra el listado los profesores que fueron encontrados, el Sistema brinda la posibilidad de revisar los expedientes de los mismos y finaliza el caso de uso. (En caso de haber más de un profesor con el mismo nombre se muestra una lista con todos los expedientes).
Referencias	R11, R12, R13, R14
Precondiciones	El Revisor debe estar autenticado correctamente en el sistema. El Revisor solicita buscar expediente.
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
	1. El Sistema muestra la interfaz correspondiente a la búsqueda del expediente del profesor.
2. El Revisor introduce los datos necesarios para realizar la búsqueda. Nombre Usuario	3. El Sistema verifica que al menos uno de los campos contenga información y que no existan errores en los datos introducidos.

No. Solapín No. Carnet de Identidad Sexo Departamento Y Facultad en caso de ser el Revisor un Especialista de Cuadros o un directivo del centro. Selecciona uno, todos o cualquier combinación de los anteriores criterios de búsqueda y presiona buscar.	
	4. El Sistema muestra un listado con el o los profesores correspondientes al criterio de búsqueda y brinda la posibilidad de revisar el expediente de cada uno de ellos.
5. El Revisor selecciona uno o varios expedientes y presiona mostrar.	6. El Sistema muestra el o los expedientes y brinda la posibilidad de seleccionar la parte del expediente que el Revisor desee consultar. Datos Sociales Datos del Currículum (Muestra Inicialmente el primer expediente y la parte del expediente referente a: Datos Personales y Familiares).
7. El Revisor selecciona la parte del expediente que	8. El Sistema muestra la parte del

desea consultar.	expediente seleccionada. Si el revisor es un especialista de cuadros el sistema le brinda la posibilidad de registrar o modificar la información personal y familiar, social o del currículum del profesor, en ese caso, ver Casos de Uso Gestionar_Datos_Personales_Familiares, Gestionar_Datos_Sociales, Gestionar_Curriculum, respectivamente.
9. El Revisor presiona aceptar. Regresar a la línea 4, o al Caso de Uso correspondiente.	
Flujos alternos	
Línea 3. 3.1 El Sistema detecta errores en los campos introducidos o que uno de los campos está vacio y brinda la posibilidad de corregir los errores. 3.2 El Revisor corrige los errores. Regresar a la línea 3 **Línea 4.** 4.1 El Sistema muestra un error de conexión con la base de datos. 4.2 El Revisor presiona aceptar y finaliza el Caso de Uso. **Línea 8.** 8.1 El Sistema muestra un error de conexión con la base de datos. 8.2 El Revisor presiona aceptar y finaliza el Caso de Uso. * En cualquier momento el Revisor presiona Cancelar y finaliza en Caso de Uso.	
Poscondiciones.	Se muestra un listado con el expediente de los profesores.

Tabla 2.9 CUS Buscar_Expediente.

Caso de Uso del Sistema Gestionar Datos Sociales

Caso de Uso:	Gestionar_Datos_Sociales
Actores:	Especialista de Cuadros
Resumen:	El Caso de Uso comienza cuando el Especialista de Cuadros accede al expediente del profesor, el sistema brinda la posibilidad de registrar o modificar la información referente a los datos sociales del mismo.
Referencias	R6
Casos de Uso Asociados	CU Gestionar_Salud (de Extensión)
Precondiciones	El Especialista de Cuadros debe estar autenticado correctamente en el sistema. El Especialista de Cuadros accede al expediente del profesor.

Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
	1. El sistema muestra la información referente a los datos sociales del profesor y brinda la posibilidad de modificarla o agregarle información nueva.
A) Si el Especialista de Cuadros desea Registrar nuevos datos ver sección 1. B) Si el Especialista de Cuadros desea Modificar los datos ya existentes ver sección 2.	

Sección 1	
Acción del Actor	**Respuesta del Sistema**
1. El Especialista de Cuadros indica	2. El sistema brinda la posibilidad de

que desea agregar nueva información.	introducir la información.
3. El Especialista de Cuadros introduce la información e indica guardarlos.	4. El sistema valida la información introducida por el Especialista de Cuadros.
	5. El Sistema registra la información introducida por el Especialista de Cuadros.
6. El Especialista de Cuadros presiona Aceptar.	
7. Finaliza el Caso de Uso.	
Flujos alternos	
Línea 4. 4.1 El Sistema detecta errores en los datos introducidos y brinda la posibilidad de corregirlos. 4.1 El Especialista de Cuadros corrige los errores. Regresar a la línea 4 de la sección 1.	
Sección 2.	
Acción del actor.	**Respuesta del Sistema.**
	1. El sistema brinda la posibilidad de modificar la información.
2. El Especialista de Cuadros selecciona la información que desea modificar.	3. El sistema muestra la información seleccionada de manera editable.
4. El Especialista de Cuadros modifica los datos y presiona guardar.	5. El sistema verifica que la información introducida por el Especialista de Cuadros este correcta.
	6. El Sistema registra la información introducida por el Especialista de

	Cuadros.
7. El especialista de Cuadros presiona aceptar	
8. Finaliza el Caso de Uso.	
Flujos Alternos.	
Línea 5. 5.1 El Sistema detecta errores en los datos introducidos y brinda la posibilidad de corregirlos. 5.2 El Especialista de Cuadros corrige los errores. Regresar a la línea 5 de la sección 2.	
Flujos Alternos. (Caso Base)	
Según sea el caso, el sistema no puede agregar o modifica la información y muestra un mensaje indicándolo. El Especialista de Cuadros presiona aceptar y finaliza el Caso de Uso. Si el Especialista de Cuadros indica que desea pasar a otra parte del expediente y se ha adicionado o modificado alguna información, el Sistema notifica si desea guardar los cambios.	
Poscondiciones.	Quedan registrados o modificados los datos sociales en el expediente del profesor.

Tabla 2.10 CUS Gestionar_Datos_Sociales

Caso de Uso del Sistema Gestionar Currículum

Caso de Uso:	Gestionar_Curriculum
Actores:	Especialista de Cuadros
Resumen:	El Caso de Uso comienza cuando el Especialista de Cuadros accede al expediente del profesor, el sistema brinda la posibilidad de registrar o modificar la información referente al

	currículum del profesor.
Referencias	R7
Casos de Uso Asociados	Buscar_Expediente (incluido)
Precondiciones	El Especialista de Cuadros debes estar autenticado correctamente en el sistema. El especialista de cuadros accede al expediente del profesor.

Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
	1. El sistema muestra la información referente a los datos del currículum del profesor y brinda la posibilidad de modificarla o agregarle información nueva.
2. A) Si el especialista de cuadros desea registrar información nueva ver sección 1. B) Si el especialista de cuadros modificar la información ya existente ver sección 2.	

Sección 1	
Acción del Actor	**Respuesta del Sistema**
1. El especialista de cuadros indica que desea agregar nueva información.	2. El sistema brinda la posibilidad de introducir la información.
3. El Especialista de Cuadros introduce la información e indica guardar.	4. El sistema verifica que los datos introducidos estén correctos.

	5. El Sistema registra la información introducida por el Especialista de Cuadros.
6. El Especialista de Cuadros presiona Aceptar.	
7. Finaliza el Caso de Uso.	
Flujos alternos	
Línea 4. 4.1 El Sistema detecta errores en los datos introducidos y brinda la posibilidad de corregirlos. 4.1 El Especialista de Cuadros corrige los errores. Regresar a la línea 4 de la sección 1.	
Sección 2.	
Acción del actor.	**Respuesta del Sistema.**
	1. El sistema brinda la posibilidad de modificar la información.
2. El Especialista de Cuadros selecciona la información que desea modificar.	3. El sistema muestra la información seleccionada de manera editable.
4. El Especialista de Cuadros modifica los datos y presiona guardar.	5. El sistema verifica que la información introducida por el Especialista de Cuadros este correcta.
	6. El Sistema registra la información introducida por el Especialista de Cuadros.
7. El especialista de Cuadros presiona aceptar	
8. Finaliza el Caso de Uso.	
Flujos Alternos.	

Línea 5. 5.1 El Sistema detecta errores en los datos introducidos y brinda la posibilidad de corregirlos. 5.2 El Especialista de Cuadros corrige los errores. Regresar a la línea 5 de la sección 2.

Flujos Alternos. (Caso Base)	
Acción del Actor	**Acción del Sistema**
1. El profesor presenta el original de cada uno de los documentos subidos previamente por el, para de esta manera darle aprobación a los mismos.	2. El especialista de cuadros aprueba los documentos.
	3. El sistema registra la información.

Según sea el caso el sistema no puede registrar modificar la información y finaliza el Caso de Uso * En todo momento si el Especialista de Cuadros presiona cancelar finaliza en Caso de Uso. Si el Especialista de Cuadros indica que desea pasar a otra parte del expediente y se ha adicionado o modificado alguna información, el Sistema notifica si desea guardar los cambios.

Poscondiciones	Quedan registrados o modificados los datos del currículum del profesor.

Tabla 2.11 CUS Gestionar Currículum

Caso de Uso del Sistema Gestionar Datos Personales Familiares

Caso de Uso:	Gestionar_Datos_Personales_Familiares
Actores:	Especialista de Cuadros
Resumen:	El Caso de Uso comienza cuando el Especialista de Cuadros accede al expediente del profesor, el sistema brinda la posibilidad de registrar o modificar la información.
Referencias	R5
Casos de Uso Asociados	CU Gestionar_Datos_Familiares (por extensión) CU Buscar_Expediente (incluido)
Precondiciones	El Especialista de Cuadros debe estar autenticado correctamente en el sistema. El Especialista de Cuadros accede al expediente del profesor.

Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
	1. El Sistema brinda la posibilidad de registrar o modificar la información referente a los datos personales y familiares del profesor.
2. A) Si desea Registrar nuevos datos ver sección 1. B) Si desea Modificar los datos ya existentes ver sección 2.	

Sección 1	
Acción del Actor	**Respuesta del Sistema**
1. El Especialista de Cuadros indica que desea agregar nueva información.	2. El sistema brinda la posibilidad de introducir la información.
3. El Especialista de Cuadros	4. El sistema verifica que no existan

introduce la información e indica guardarla.	errores en los datos introducidos por el Especialista de Cuadros.
	5. El Sistema registra la información introducida por el Especialista de Cuadros.
6. El Especialista de Cuadros presiona Aceptar.	
7. Finaliza el Caso de Uso.	
Flujos alternos	
Línea 4. 4.1 El Sistema detecta errores en los datos introducidos y brinda la posibilidad de corregirlos. 4.2 El Especialista de Cuadros corrige los errores. Regresar a la línea 4 de la sección 1.	
Sección 2.	
Acción del actor.	**Respuesta del Sistema.**
	1. El sistema brinda la posibilidad de modificar la información referente a los datos personales y familiares del profesor.
2. El Especialista de Cuadros selecciona la información que desea modificar.	3. El sistema muestra la información seleccionada de manera editable.
4. El Especialista de Cuadros modifica los datos y presiona guardar.	5. El sistema verifica que no existan errores en los datos introducidos por el Especialista de Cuadros.
	6. El sistema notifica si realmente desea modificar dichos datos.
7. El especialista de cuadros presiona	8. El Sistema registra la información

aceptar	introducida por el Especialista de Cuadros.
9. El especialista de Cuadros presiona aplicar.	
10. Finaliza el Caso de Uso.	
Flujos Alternos. (caso base)	
Según sea el caso, el sistema no puede registrar o modificar la información y finaliza el Caso de Uso. Si el Administrador del Sistema selecciona mostrar otra parte del expediente del profesor e hizo alguna modificación, el sistema brinda la posibilidad de guardar los cambios.	
Poscondiciones.	Quedan registrados o modificados los Datos Personales y Familiares del profesor.

Tabla 2.12 CUS Gestionar Datos Personales Familiares

Caso de Uso del Sistema Controlar Proceso de Categoría

Caso de Uso:	Controlar_Proceso_Cat
Actores:	Miembro_Tribunal_Categorizacion
Resumen:	El Caso de Uso comienza cuando el Miembro del Tribunal de Categorización presiona la opción de controlar el proceso de categorización, el Sistema brinda la posibilidad de revisar los expedientes de los profesores que están asignados a dicho al tribunal del cual es miembro, o notificar los resultados del proceso de categorización para un profesor determinado, el caso de uso finaliza cuando el Miembro del Tribunal solicita salir del sistema.
Referencias	R23

Precondiciones	Debe estar abierta la convocatoria para el proceso de categorización. El Miembro del Tribunal de Categorización debe estar autenticado correctamente.
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
	1. El Sistema muestra un listado con los profesores que están asignados a dicho tribunal de categorización, brinda la posibilidad de revisar los expedientes o de notificar los resultados de los exámenes realizados hasta la fecha por cada uno de los profesores.
2. El Miembro del Tribunal de Categorización selecciona el o los profesores y la acción que desea realizar. A) Si el Miembro del Tribunal de Categorización presiona Ver Expediente ver sección 1. B) Si el Miembro del Tribunal de Categorización presiona Notificar Resultados ver sección 2.	
Sección 1	
Acción del Actor	**Respuesta del Sistema**
	1. El Sistema muestra el o los expedientes. (Muestra Inicialmente el primer expediente

	y la parte del expediente referente a: Datos Personales y Familiares).
2. El Miembro del tribunal de categorización selecciona la parte del expediente que desea revisar. Datos Sociales. Datos del currículum.	3. El Sistema carga y muestra la información referente a la parte del Expediente seleccionada.
4. El Miembro del Tribunal de Categorización presiona aceptar, regresar a la línea 2 del flujo base.	
Flujos alternos	
Sección 2.	
Acción del actor.	**Respuesta del Sistema.**
	1. El Sistema brinda la posibilidad de notificar el resultado del o los profesores seleccionados, en cada uno de los exámenes realizados hasta el momento.
2. El Miembro del Tribunal de Categorización selecciona el tipo de examen y la calificación y presiona aceptar.	3. El Sistema verifica que se llenen los campos.
	4. El Sistema registra la información introducida.
	5. El Sistema notifica que la

	información ha sido registrada correctamente.
6. El Miembro del Tribunal de Categorización presiona aceptar, regresar a la línea 2 del flujo base.	
Flujos Alternos.	
Línea 3. 3.1 El sistema detecta que existen campos vacios y brinda la posibilidad de llenarlos. 3.2 El Miembro del Tribunal de Categorización llena los campos. Regresar a la línea 3 de la sección 2. **Línea 4.**4.1 El sistema muestra un error de conexión con la base de datos. 4.2 El Especialista de Cuadros presiona aceptar y finaliza el Caso de Uso.	
Poscondiciones.	Dependiendo del caso queda registrada la calificación de cada uno de los profesores en los exámenes realizados.

Tabla 2.13 CUS Controlar Proceso de Categoría

Para abundar más en la descripción textual de los Casos de Uso del Sistema ver anexo 2

Conclusiones

En este capítulo se abordó la situación problémica que se presenta actualmente a la hora de gestionar el expediente del profesor en el Departamento de Capital Humano, definiendo la propuesta de un sistema a partir del objeto de estudio, mostrándose de una vez el diagrama de casos de uso del negocio, el del modelo de objeto del negocio y el de los casos de uso del sistema. Se definieron además, los requerimientos funcionales y no funcionales del software, así como la descripción de cada uno de los casos de usos del sistema, determinándose los actores que interactúan con el mismo.

CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA

Introducción.

A partir del diagrama de casos de uso, obtenido como parte del flujo de trabajo de requerimientos, se puede establecer una traza, con la disciplina que continúa el ciclo de vida de RUP, esta se encarga de describir en términos de clases cómo debe funcionar cada caso de uso y las colaboraciones que existen entre las clases definidas.

De esta forma se obtienen los diagramas de clases de análisis correspondientes a cada caso de uso, conformando las realizaciones de los mismos conjuntamente con los diagramas de interacción hasta llegar a refinar este análisis en el diseño esperado para la aplicación a desarrollar, con el objetivo de transformar los requerimientos, al diseño que el sistema debe tener y con la finalidad de adaptarlo al ambiente de implementación, trazando su funcionamiento.

Para el desarrollo satisfactorio de esta disciplina, la metodología propone el uso de patrones con el objetivo de lograr una arquitectura robusta para el sistema en cuestión.

3.1 Diagramas de clases de análisis.

Los diagramas de clases de análisis, expresan la definición y relación entre las clases, están compuestos por clases que pueden ser de tres tipos fundamentales: interfaz, controladora y entidad.

Las clases interfaz: son las que se encargan de brindar un entorno gráfico amigable y entendible para que el usuario interactúe con el sistema de manera que modelan la interacción entre el sistema y sus actores.

Las clases controladoras: son las que se encargan de captar toda la información que es enviada a través de las clases interfaz y guardar esos datos a través de métodos en la base de datos, coordinando la realización de uno o unos pocos casos de uso así como las actividades de los objetos que implementan la funcionalidad del caso de uso.

Las clases entidades: son las que se obtienen como resultado de las entidades definidas en el negocio estudiado y definido, aunque puede suceder que algunas de estas desaparezcan y surjan nuevas entidades, generalmente modelan información que posee larga vida y que es a menudo persistente, estas clases son las futuras tablas de la base de datos.

Los diagramas de clases de análisis además, contienen relaciones que evidencian la navegabilidad y la dependencia entre clases, de esta forma se identifican relaciones de asociación, generalización especialización, entre otras.

Diagrama de Clase del Análisis

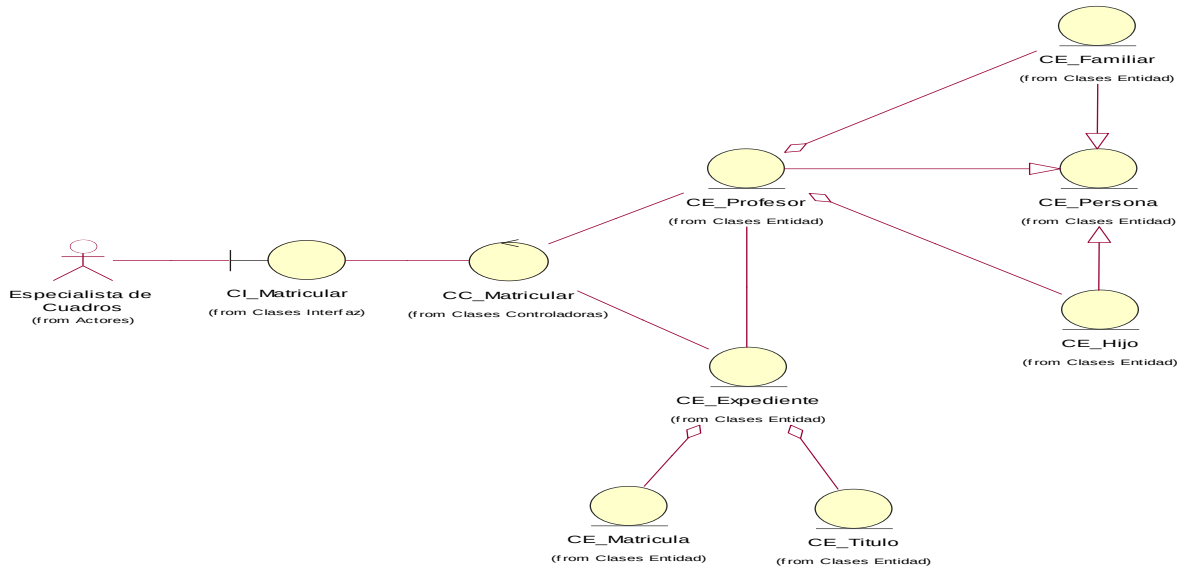

Figura 3.1 DCA Caso de Uso Matricular

Diagrama de Clase del Análisis

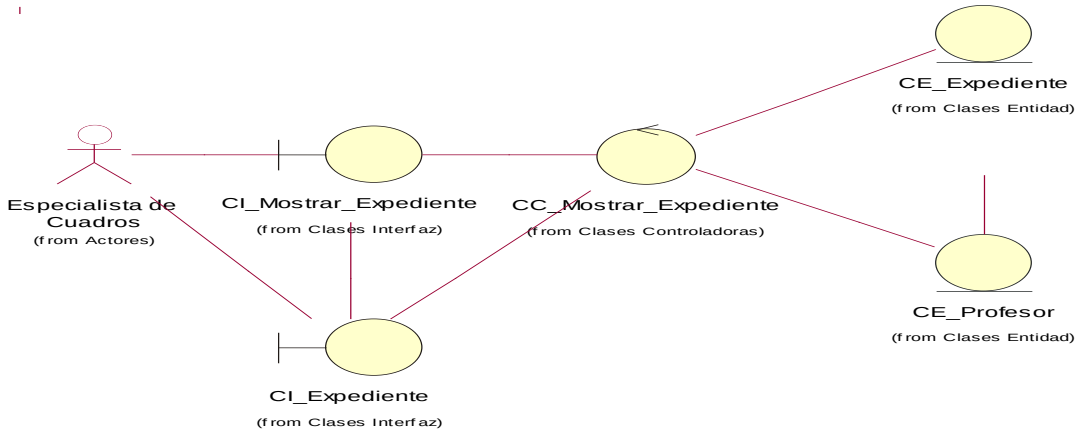

Figura 3.2 DCA Caso de Uso Buscar Expediente

Diagrama de Clase del Análisis

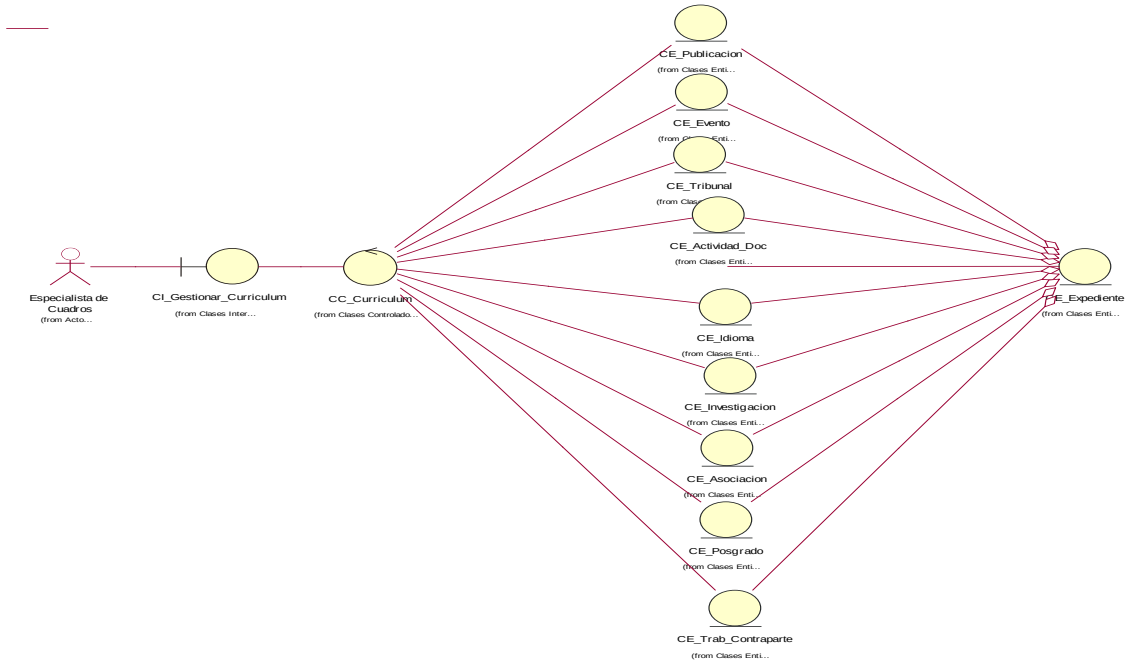

Figura 3.3 DCA Caso de Uso Gestionar Currículum

Diagrama de Clase del Análisis

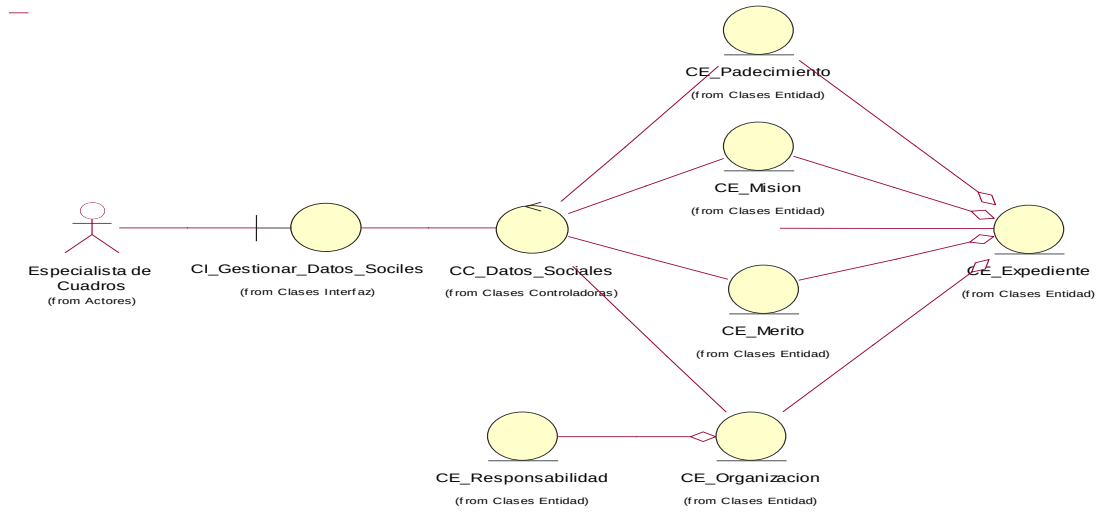

Figura 3.4 DCA Caso de Uso Gestionar Datos Sociales

Diagrama de Clase del Análisis

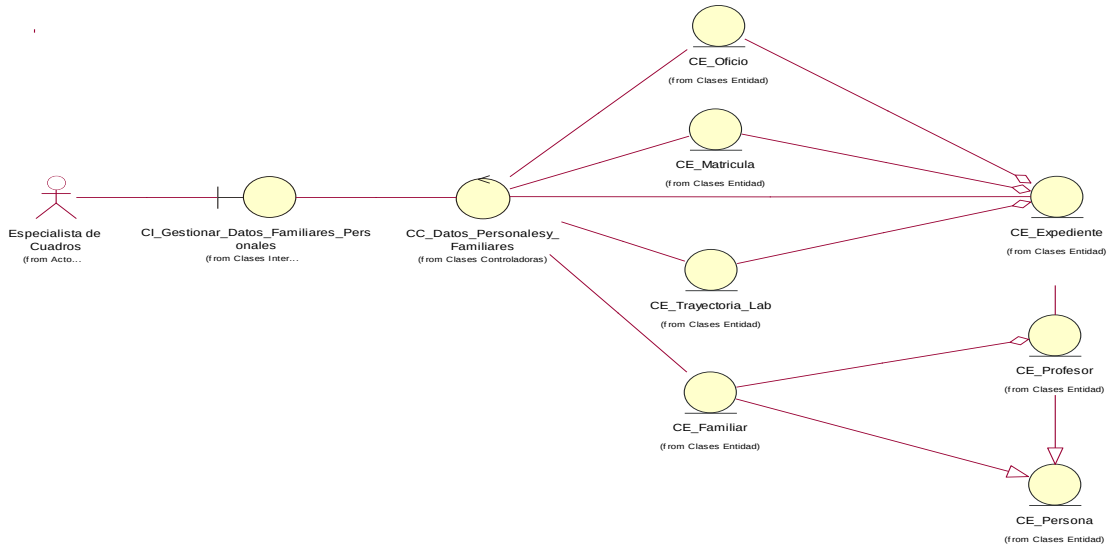

Figura 3.5 DCA Caso de Uso Gestionar Datos Personales y Familiares

Para consultar los restantes diagramas de clases del análisis ver anexo 3.

3.2 Diagramas de clases de diseño.

Para la elaboración del diseño, generalmente se utilizan un grupo de patrones o modelos para lograr objetivos específicos, dentro de los más conocidos están los GRASP (General Responsibility Asignment Software Patterns) que en su traducción al español significa patrones generales de software para asignar responsabilidades.

De manera general los patrones constituyen soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos, basados en la experiencia y que se ha demostrado que funcionan. [16]

Los patrones GRASP se utilizan con el objetivo de asignar responsabilidades a las diferentes clases que se definen en el diseño. Dentro de este grupo se identifican cinco patrones muy utilizados:

Experto: Se encarga de asignar una responsabilidad a la clase que tiene la información necesaria para cumplirla, siendo este el principio fundamental en virtud del cual se le asignan las responsabilidades a los objetos.

Creador: Se encarga de identificar quién debe ser el responsable de la creación (o instanciación) de nuevos objetos o clases.

Alta cohesión: Nos dice que la información que almacena una clase debe de ser coherente y está en la mayor medida de lo posible relacionada con la clase.

Bajo acoplamiento: Se basa en lograr tener las clases lo menos ligadas entre sí posible. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases.

Controlador: Sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado.

En los diagramas de clases que se elaboraron, no se aprecian directamente el uso de estos patrones, debido a que no hay necesidad de crear nuevas clases, simplemente se aplican a las clases definidas en el diseño, distribuyendo responsabilidades entre las mismas de forma tal que no existan muchas relaciones, que no se sobrecargue de métodos a una clase en específico pudiendo acomodarlos en otras, entre otras mejoras que brinda el uso de este grupo de patrones.

3.2.1 Diagramas de interacción.

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema, lo que conlleva modelar instancias concretas o prototípicas de clases interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento. En el contexto de las clases describen la forma en que grupos de objetos colaboran para proveer un comportamiento. [17]

3.2.1.1 Diagramas de secuencia.

El diagrama de secuencia es uno de los diagramas más efectivos para modelar interacción entre objetos en un sistema. Un diagrama de secuencia muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada método de la clase. Mientras que el diagrama de casos de uso permite el modelado de una vista de negocio del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario y mensajes pasados entre los objetos. [18]

Diagrama de Interacción

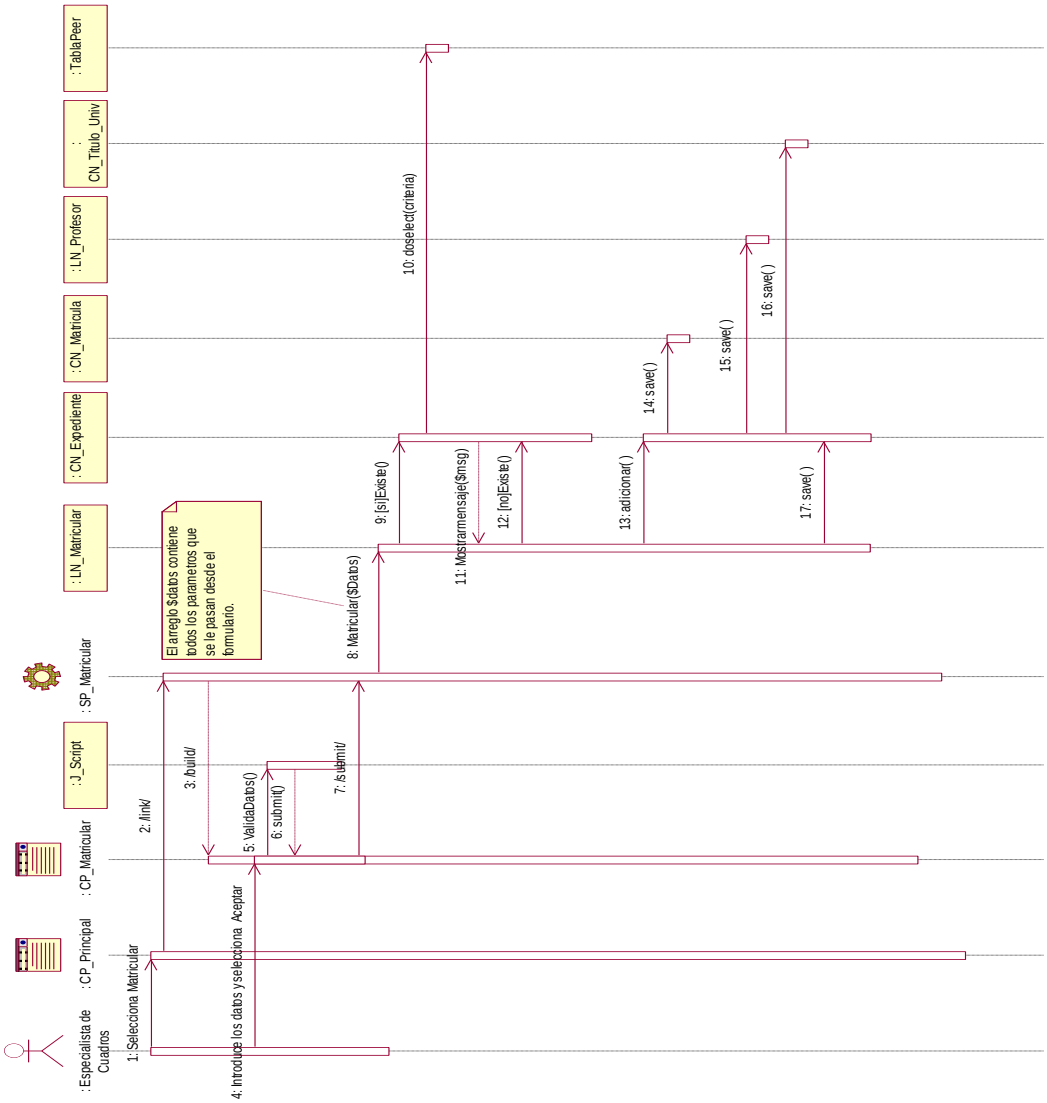

Figura 3.6 DS Caso de Uso Matricular

Diagrama de Interacción

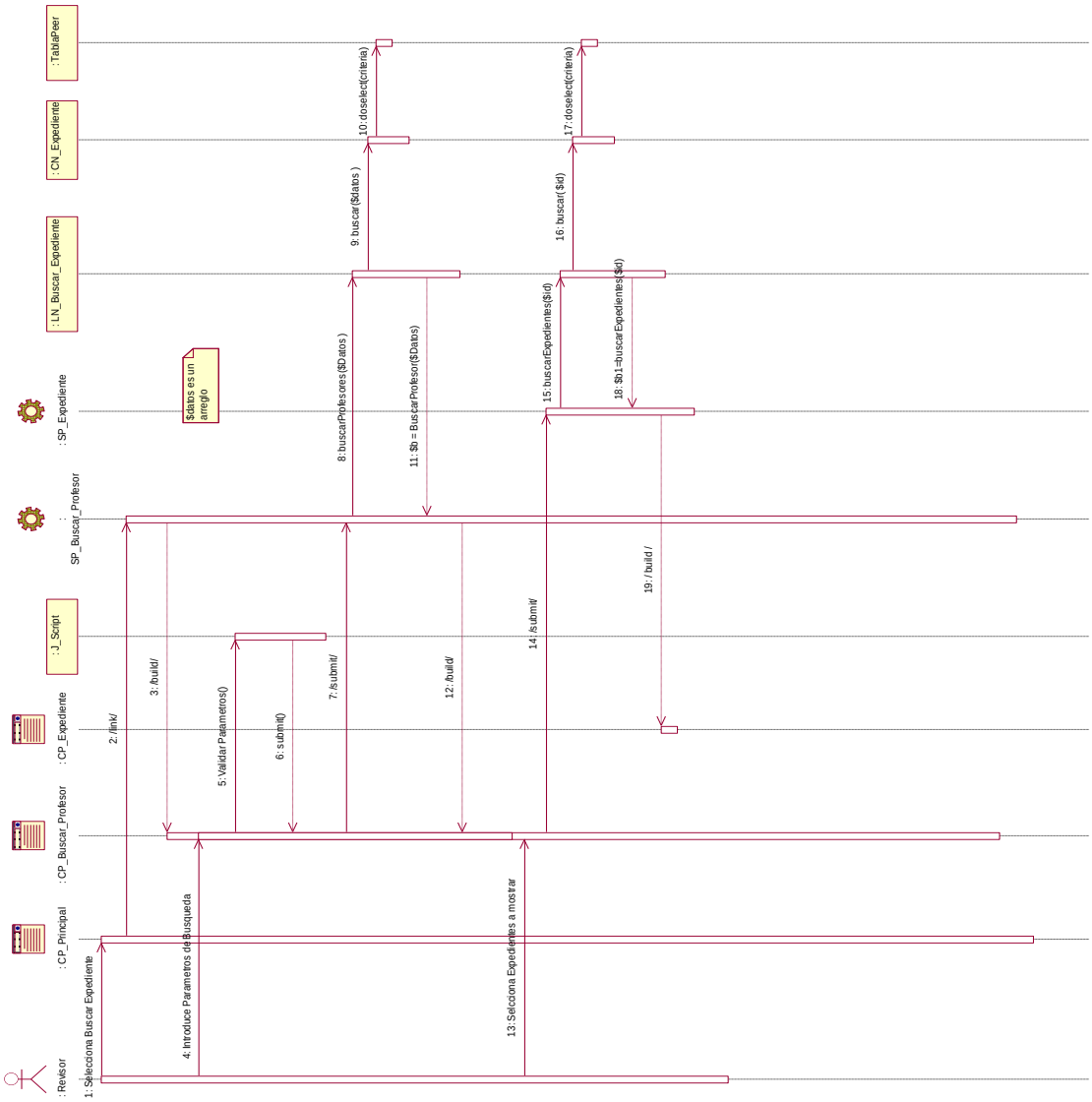

Figura 3.7 DS Caso de Uso Buscar Expediente

Diagrama de Interacción

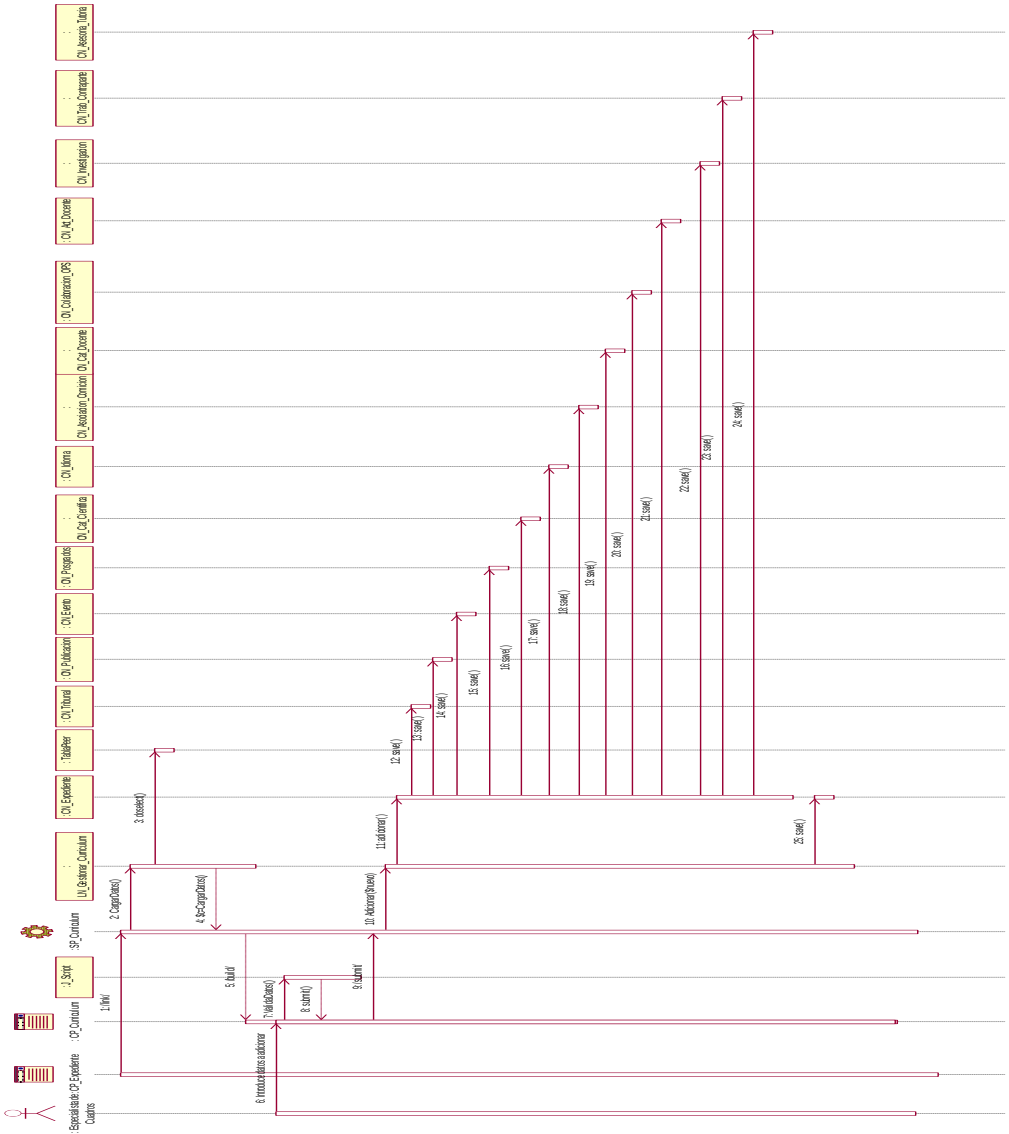

Figura 3.8 DS Caso de Uso Gestionar Currículum (Escenario Adicionar)

Diagrama de Interacción

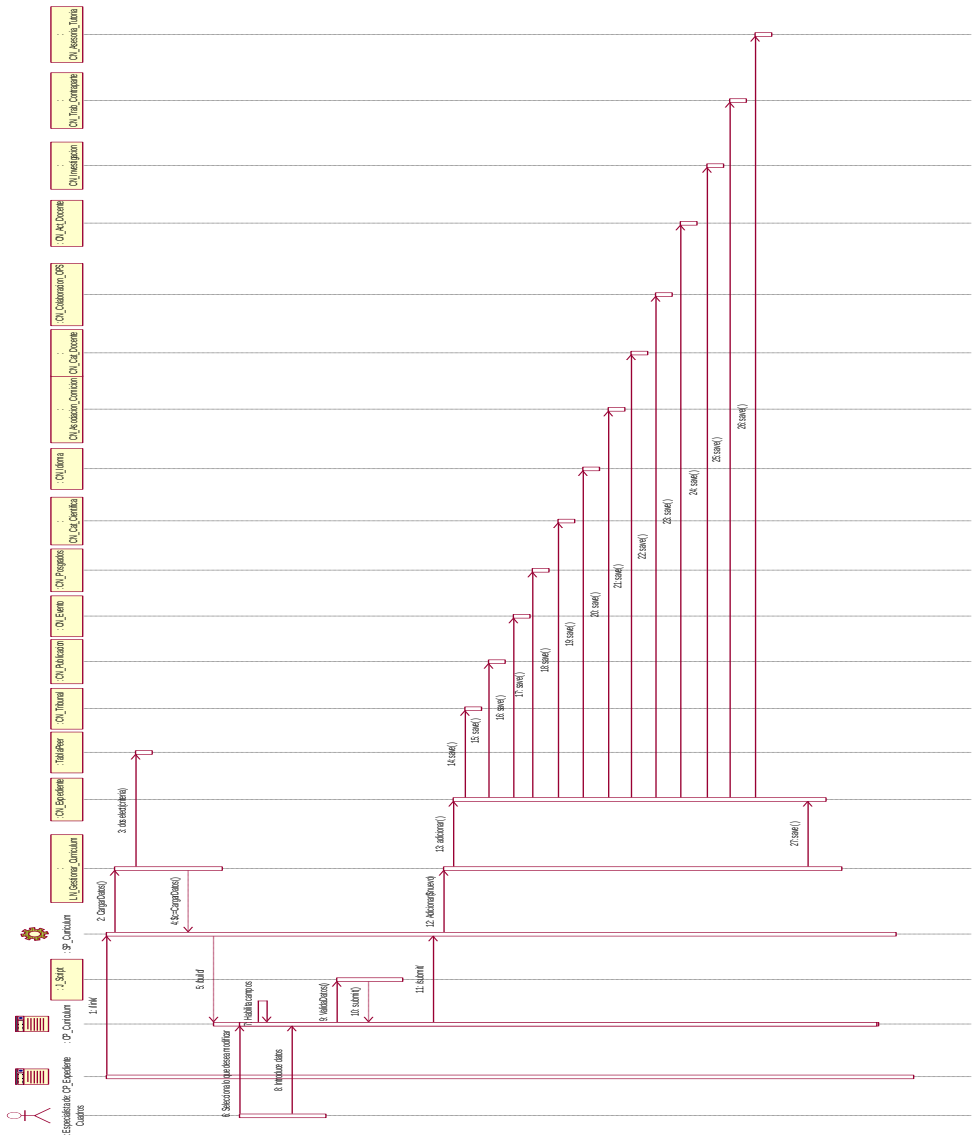

Figura 3.9 DS Caso de Uso Gestionar Currículum (Escenario Modificar)

Para consultar los restantes diagramas de secuencia ver anexo 4

3.2.2 Diagramas de clases de diseño Web.

Diagrama de Clases del Diseño

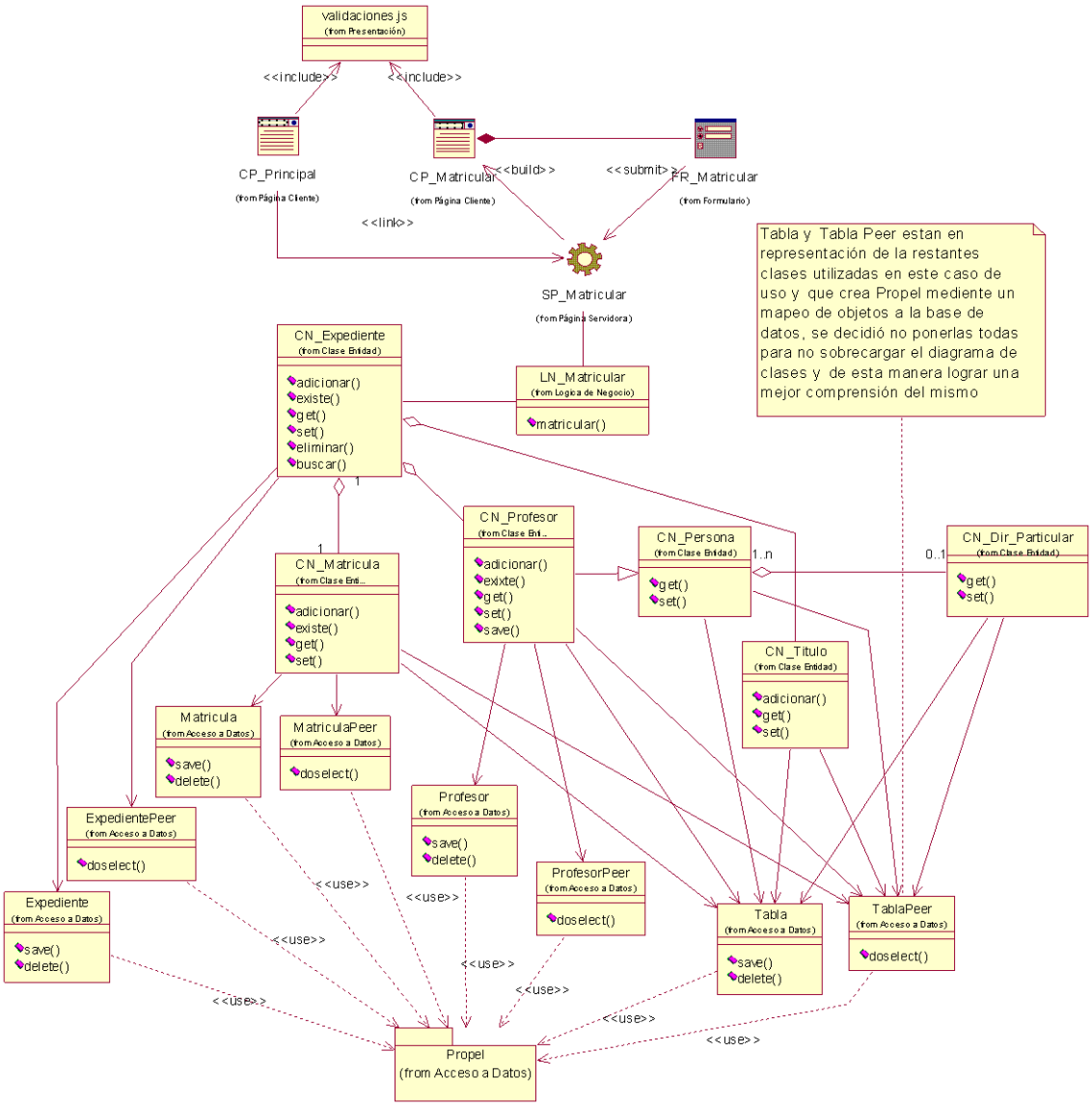

Figura 3.10 DCD Caso de Uso Matricular

Diagrama de Clases del Diseño

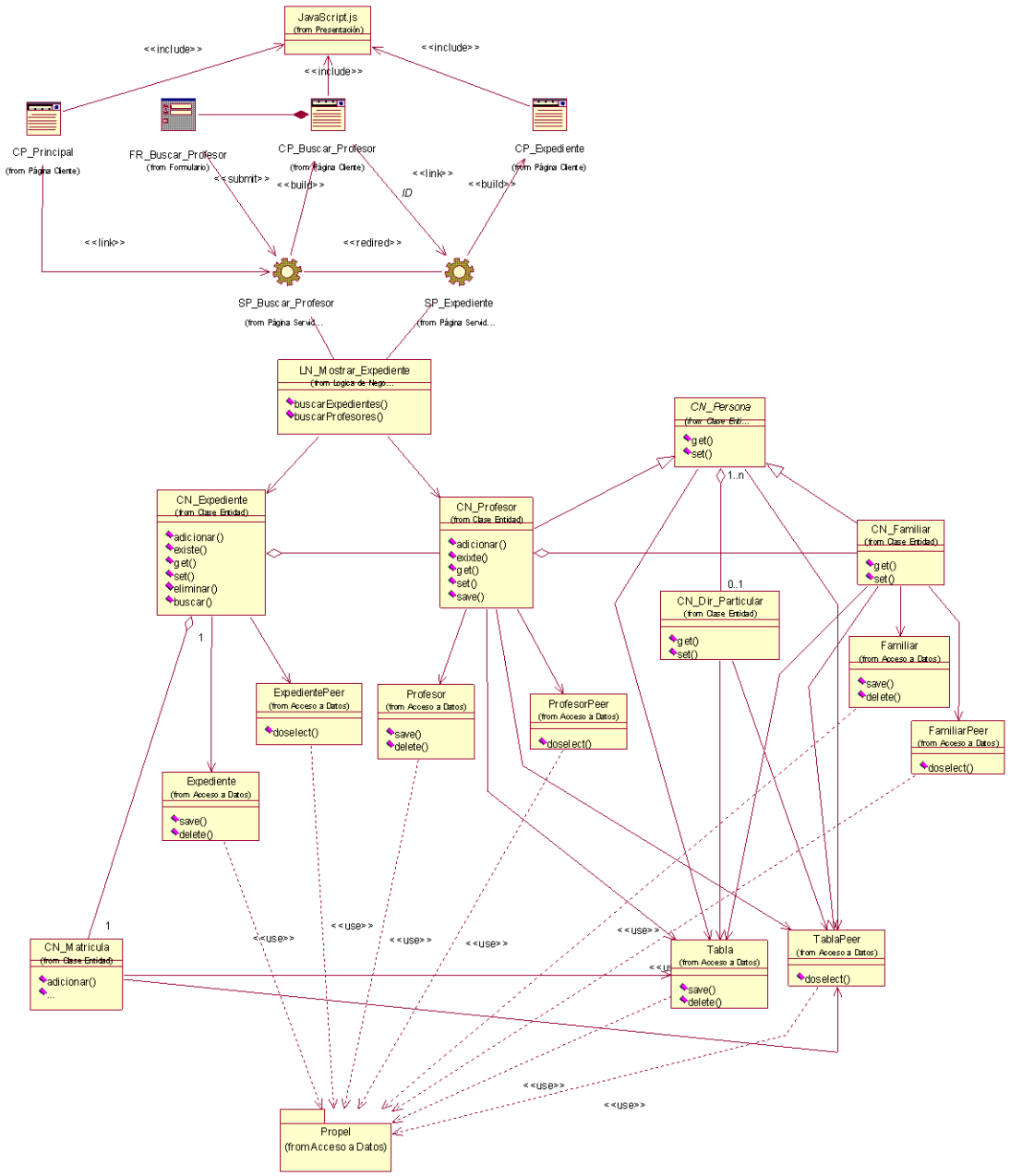

Figura 3.11 DCD Caso de Uso Buscar Expediente

Diagrama de Clases del Diseño

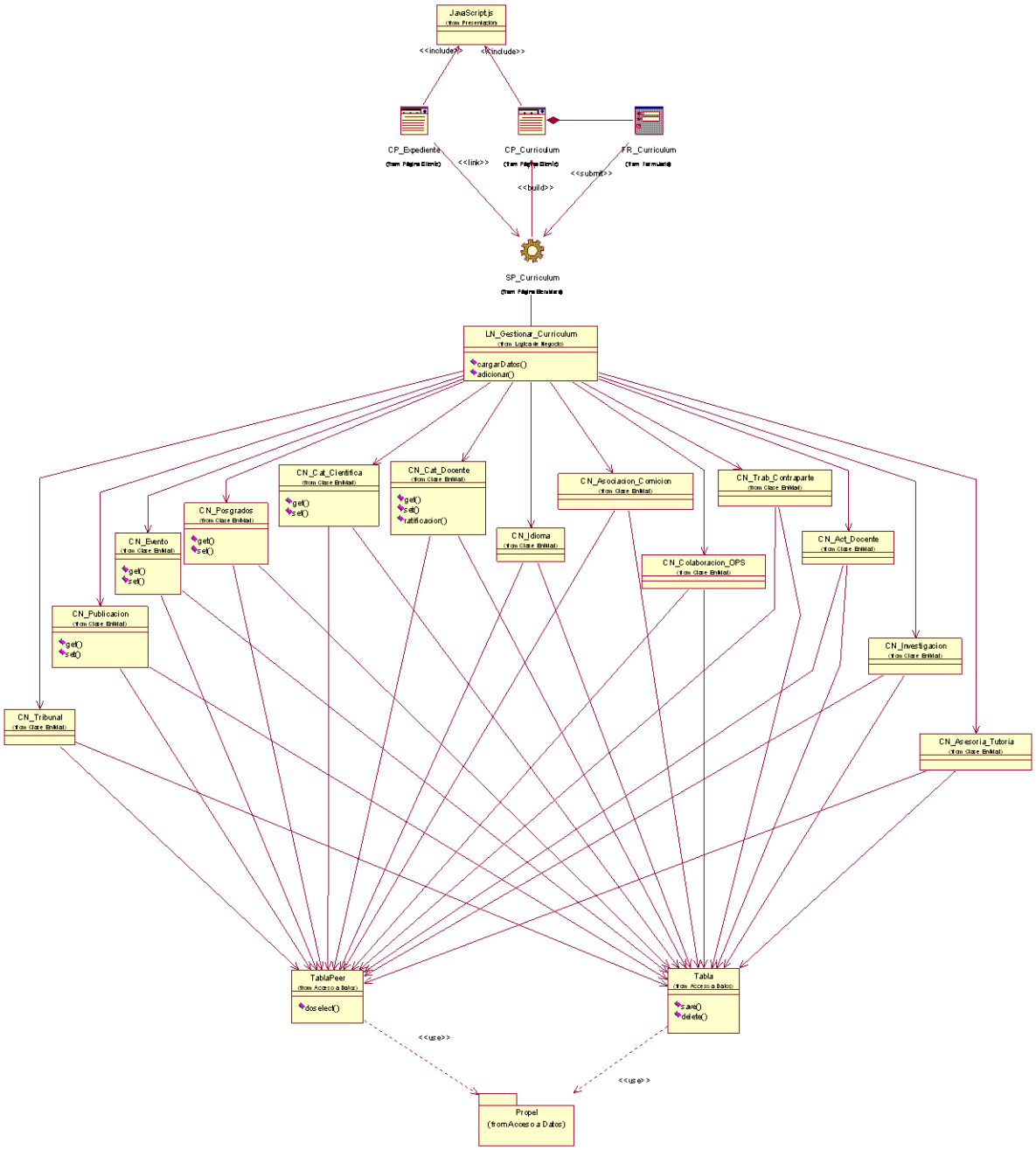

Figura 3.12 DCD Caso de Uso Gestionar Currículum

Diagrama de Clases del Diseño

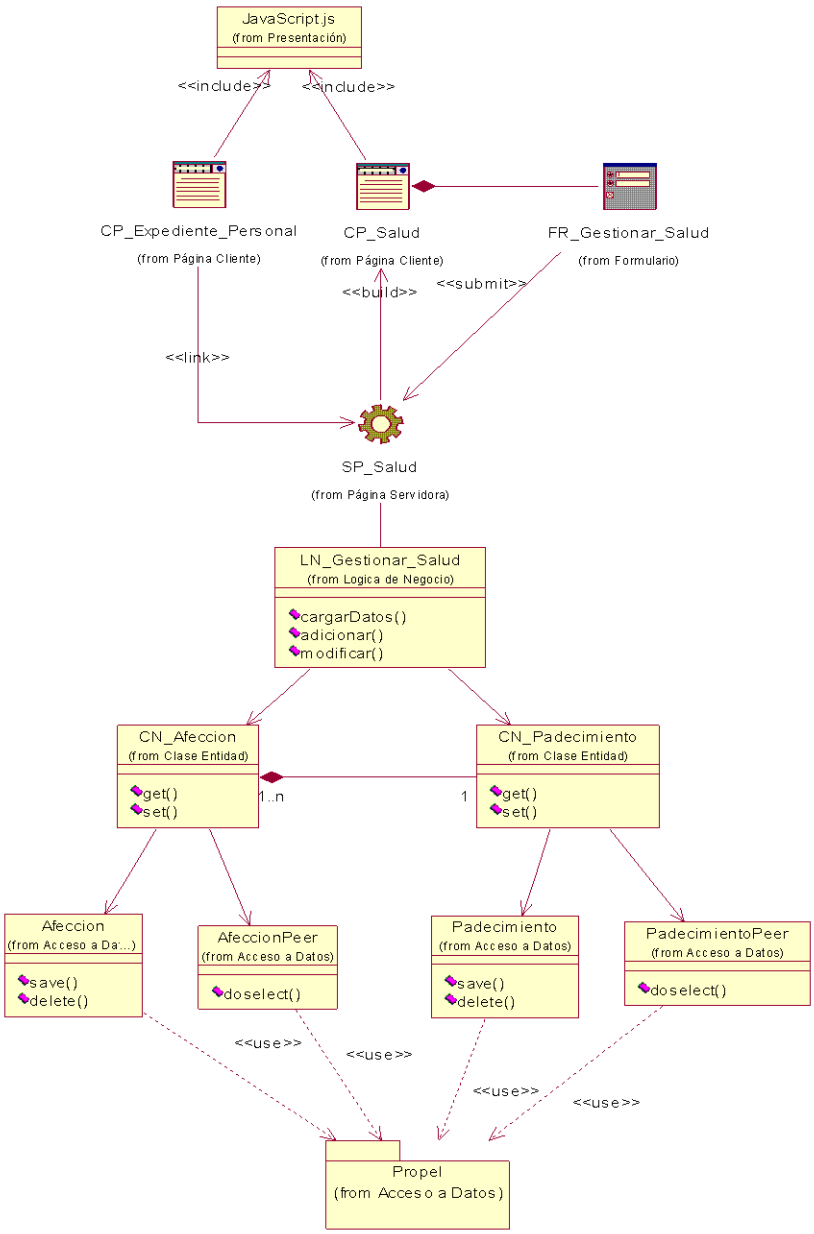

Figura 3.13 DCD Caso de Uso Gestionar Datos Sociales

Diagrama de Clases del Diseño

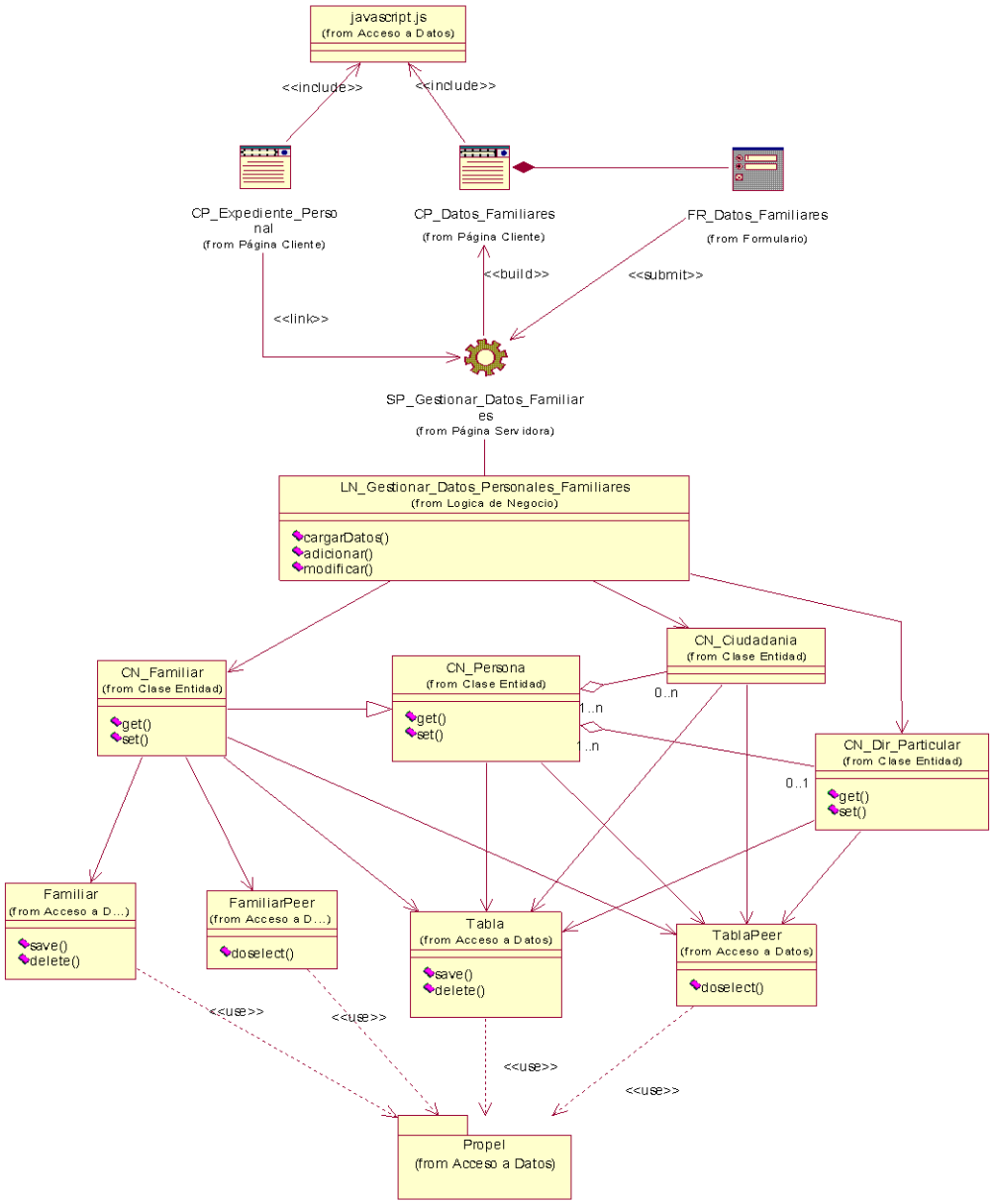

Figura 3.14 DCD Caso de Uso Gestionar Datos Personales Familiares

Para consultar los restantes diagramas de clases del diseño ver anexo 5

3.3 Diseño de la base de datos propuesta.

3.3.1 Diagrama entidad-relación de la base de datos.

El diagrama o modelo entidad-relación es una herramienta para el modelado de datos de un sistema de información. Estos modelos expresan entidades relevantes para un sistema de información, sus inter-relaciones y propiedades.

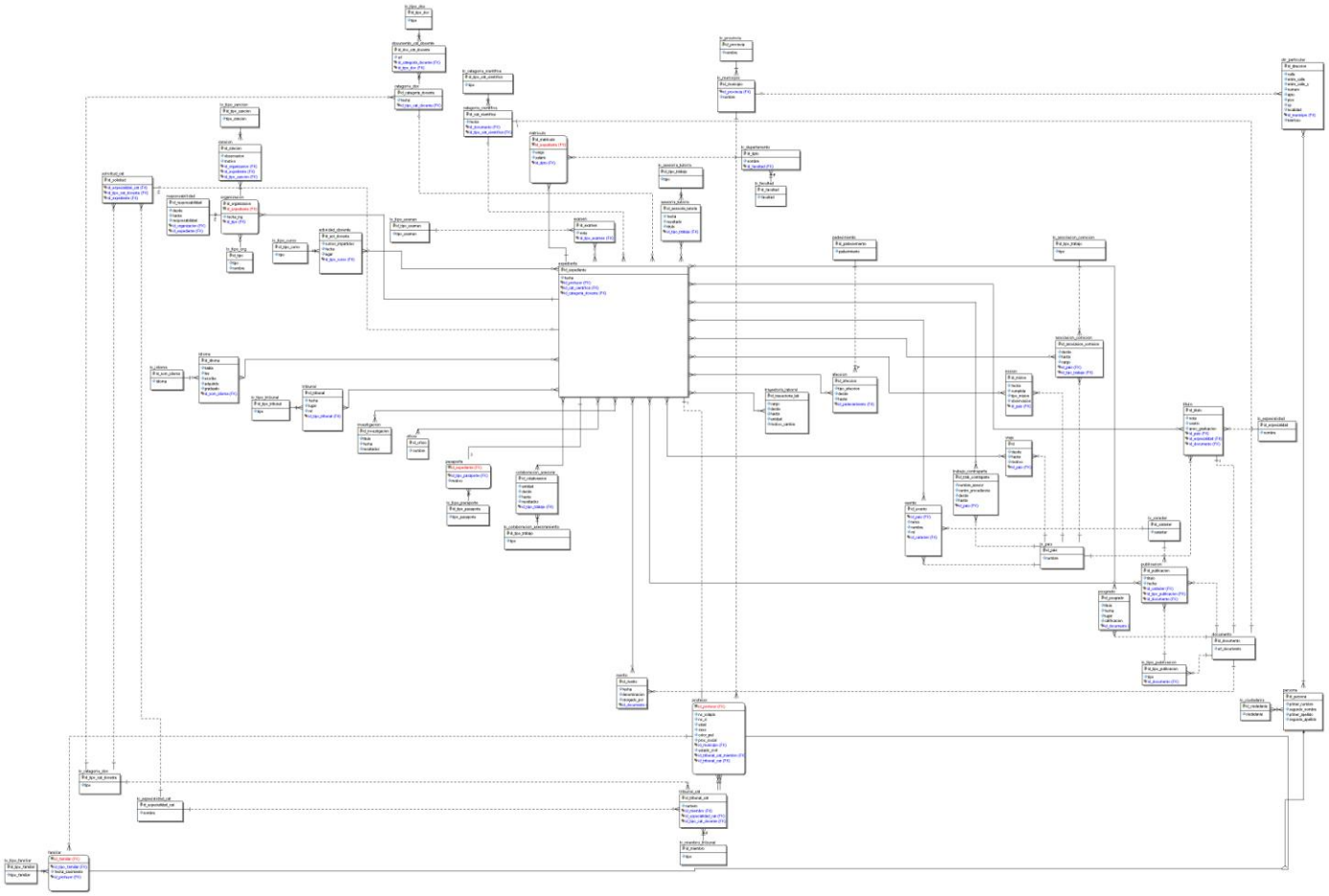

Figura 3.16 Modelo Entidad-Relación

3.3.2 Descripción de las tablas de la base de datos.

Nombre: profesor		
Descripción: Se almacenan los datos propios de un profesor.		
Atributo	**Tipo**	**Descripción**
id_profesor	INTEGER	Identificador del profesor
no_solapin	INTEGER	Número de solapín del profesor
no_ci	INTEGER	Número de carnet de identidad del profesor.
edad	INTEGER	Edad del profesor
sexo	CHAR	Sexo del profesor
color_piel	VARCHAR	Color de piel del profesor
proc_social	VARCHAR	Procedencia social del profesor
estado_civil	VARCHAR	Estado civil del Profesor
id_municipio_nac	INTEGER	Identificador del municipio donde nació (llave foránea)
id_tribunal_cat_miembro	INTEGER	Identificador del tribunal de categoría del cual es miembro (llave foránea)
id_tribunal_cat	INTEGER	Identificador del tribunal de categoría al cual esta asignado para matricularse (llave foránea)

Nombre: asesoría_tutoria		
Descripción: Se almacena la información relacionada con los trabajos de asesoría o tutoría a tesis de investigación o grado científico.		
Atributo	**Tipo**	**Descripción**
id_asesoria_tutoria	INTEGER	Identificador de la asesoría o tutoría.
titulo	VARCHAR	Título de la investigación o la tesis de grado científico.
resultado	VARCHAR	Resultado obtenido en el trabajo realizado.
fecha	DATE	La fecha en que se realizó la investigación o la tesis de grado científico.
id_tipo_trabajo	INTEGER	Identificador del tipo de trabajo realizado. (asesoría, tutoría) (llave foránea)

Nombre: asociacion_comicion		
Descripción: Se almacena la información relacionada con las asociaciones o comisiones, nacionales o internacionales a las que pertenece.		
Atributo	**Tipo**	**Descripción**
id_asociacion_comision	INTEGER	Identificador de la asociación o comisión.
desde	DATE	Fecha en la que comenzó a formar parte de la asociación o la comisión.
hasta	DATE	Fecha de fin como miembro de la asociación o la comisión.
cargo	VARCHAR	Cargo que ocupa.

id_pais	INTEGER	Identificador del país donde radica la asociación o la comisión.
id_tipo_trabajo	INTEGER	Identificador de la asociación o comisión (llave foránea)

Nombre: categoria_cientifica		
Descripción: Se almacena la información relacionada con la categoría científica.		
Atributo	**Tipo**	**Descripción**
id_cat_cientifica	INTEGER	Identificador de la categoría_cientifica
fecha	DATE	Fecha en que obtuvo la categoría.
id_documento	INTEGER	Identificador del certificado que acredita la categoría. (llave foránea)
id_tipo_cat_cientifica	INTEGER	Identificador del tipo de categoría científica. (Máster, Doctor) (llave foránea)

Nombre: categoria_doc		
Descripción: Se almacena la información relacionada con la categoría docente		
Atributo	**Tipo**	**Descripción**
id_categoria_docente	INTEGER	Identificador de la categoría docente.
fecha	DATE	Fecha en que obtuvo la categoría docente.
id_tipo_cat_docente	INTEGER	Identificador del tipo de categoría docente. (instructor, auxiliar, asistente, titular) (llave foránea)

Nombre: dir_particular		
Descripción: Se almacena la dirección particular tanto del profesor como se sus familiares.		
Atributo	**Tipo**	**Descripción**
id_direccion	INTEGER	Identificador de la dirección particular.
calle	VARCHAR	Nombre de la calle donde reside
entre_calle	VARCHAR	Nombre de la primera entrecalle
entre_calle_y	VARCHAR	Nombre de la segunda entrecalle
numero	NUMERIC	Numero de la casa
apto	VARCHAR	Apartamento
piso	VARCHAR	Piso
telefono	NUMERIC	Teléfono particular
id_municipio	INTEGER	Identificador del municipio

Para abundar mas en la descripción de las tablas del modelo entidad relación ver anexo 6

Conclusiones.

A partir de las clases de análisis definidas en función de los casos de uso arquitectónicamente significativos, se arribó al modelo de análisis, que da entrada al diseño y que está elaborado en el lenguaje del usuario, sin necesidad de especificar métodos ni atributos de las clases expuestas.

Como resultado de refinar este artefacto se alcanzó la primera aproximación del modelo de diseño. Con la elaboración del diseño, se definen las clases fundamentales que permitirán que el sistema funcione satisfactoriamente, así como los atributos y métodos que deben tener las mismas, para brindarle al desarrollador una idea clara de lo que debe implementar.

El uso de patrones de diseño contribuyó a reutilizar la experiencia de los desarrolladores, ya que clasifican y describen formas de solucionar problemas, que ocurren de forma frecuente en el desarrollo. La realización de cada caso de uso es un artefacto muy importante que da entrada al flujo de trabajo que le sigue al diseño en la metodología RUP.

CONCLUSIONES

Con el desarrollo de este trabajo se arriba a las siguientes conclusiones:

- Se llevo a cabo un profundo estudio sobre el proceso de gestión de la información relacionado con el expediente del profesor.
- Se realizó un levantamiento de requisitos del sistema de la forma más acertada posible, obteniéndose las futuras funcionalidades que requiere el sistema informático a desarrollar.
- Se profundizaron los conocimientos específicos de aquellas tecnologías, metodologías y herramientas necesarias para el desarrollo de software.
- Se obtuvieron los artefactos necesarios para lograr la concepción del producto en cuestión.
- Se logró un análisis y diseño que resulta muy necesario para continuar el trabajo en las fases posteriores del desarrollo de software.
- Se logró una arquitectura adecuada a los lineamientos de la Dirección de Informatización de la Infraestructura Productiva.

De esta forma se dio cumplimiento a los objetivos trazados para la elaboración del trabajo de diploma, obteniendo el diseño de un sistema informático, que permita mejorar el proceso de gestión de la información referente al expediente del profesor en la Universidad de las Ciencias Informáticas.

RECOMENDACIONES

En la realización del trabajo fue posible determinar algunas recomendaciones necesarias:

- Implementar el sistema diseñado utilizando las tecnologías que se ajustan a este tipo de aplicación.
- Probar y desplegar la aplicación en los Departamentos de Capital Humano de la Universidad de las Ciencias Informáticas.

GLOSARIO

Firewall: Un firewall es un dispositivo que funciona como cortafuegos entre redes, permitiendo o denegando las transmisiones de una red a la otra. Un uso típico es situarlo entre una red local y la red Internet, como dispositivo de seguridad para evitar que los intrusos puedan acceder a información confidencial. [19]

Framework: En el desarrollo de software, un *framework* es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un *framework* puede incluir soporte de programas, bibliotecas y un lenguaje de scripting entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Interface: Se le llama así a las clases que definen métodos y atributos abstractos y que luego son utilizados por otras clases para su funcionamiento. En lenguajes de programación como Java son muy utilizados.

World Wide Web Consortium: 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. [20]

Arquitectura Cliente – Servidor: La arquitectura cliente/servidor es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes. [21]

Base de datos: (DataBase). Conjunto de datos relacionados que se almacenan de forma que se pueda acceder a ellos de manera sencilla, con la posibilidad de relacionarlos, ordenarlos en base a diferentes criterios, etc. [22]

Servidor: es una computadora central, de gran capacidad, compartida por las otras computadoras de la red, llamadas Clientes o estaciones de trabajo (workstations), ya que reciben el servicio de almacenar, controlar y compartir la información contenida en el servidor. [23]

Clases: abstracciones que representan a un conjunto de objetos con un comportamiento e interfaz común. [24]

Funciones: conjunto de instrucciones que permiten procesar las variables para obtener un resultado. [25]

Orientado a objetos: Significa que el software se organiza como una colección de objetos discretos que contiene tanto estructura de datos como también un comportamiento. [26]

Lenguaje interpretado: un lenguaje de programación que fue diseñado para ser ejecutado por medio de un intérprete, en contraste con los lenguajes compilados. También se les conoce como lenguajes de script.

REFERENCIA BIBLIOGRÁFICA

1. CASAS de BARRAN, ALICIA y GRIFFIN, ANDREW. *ORGANIZACIÓN Y CONTROL DE DOCUMENTOS ADMINISTRATIVOS.* Montevideo : s.n., 2004.

2. BJÖRK, B.-C. *Document management - a key technology for the construction industry. Information and Communications Technology (ICT) in the Practice of Building and Civil Engineering, Helsinki, Finland, Association of Finnish Civil Engineers, 2001.*

3. Bustelo Ruesta, Carlota y García-Morales Huidobro, Elisa. *TENDENCIAS EN LA GESTION DE LA INFORMACION, LA DOCUMENTACION Y EL CONOCIMIENTO EN LAS ORGANIZACIONES.* s.l. : El Profesional de la Información, 2001.

4. Codina, Luís. El Profesional de la Información. *El Profesional de la Información.1993. Disponible en :* http://www.elprofesionaldelainformacion.com/contenidos/1993/mayo/qu_es_un_sistema_de_gestin_documental.html.

5. Bustelo Ruesta, Carlota. *GESTIÓN DOCUMENTAL EN LAS EMPRESAS: UNA APROXIMACIÓN.* 2000.

6. *Internet*. Disponible en: http://eia.udg.es/~ramon/ib/t3/tema3.htm#punto31

7. *Navegadores Web*. Disponible en: http://es.wikipedia.org/wiki/Navegadores_web

8. *Sistemas distribuidos*. 2006. Disponible en: http://www.augcyl.org/glol/old/N_1/sistemasdis.html

9. *Modelo Cliente/Servidor (dos capas)*. Disponible en: http://www.augcyl.org/glol/old/N_1/sistemasdis.html

10. *Modelo cliente-servidor* Disponible en: http://www.csi.map.es/csi/silice/Global71.html

11. *Modelo Cliente/Servidor (tres capas)*. 2006. Disponible en: http://www.janium.com/phpws/index.php?module=documents&JAS_DocumentManager_op=downloadFile&JAS_File_id=1

12. *Metodologías de Desarrollo*. 2007. Disponible en: http://www.um.es/docencia/barzana/IAGP/Iagp2.html#BM1

13. *Mendoza Sanchez, María A.* Metodologías de Desarrollo de Software. 2004.

14. *EXtreme Programming (XP)*. Disponible en: http://www.willydev.net/descargas/prev/TodoAgil.Pdf

15. *Crystal Methodologies* 2007. Disponible en: http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/heterodox.asp#8

16. *Patrones de diseño*. 2007. Disponible en: http://www.ingenierosoftware.com/analisisydiseno/patrones-diseno.php

17. *Diagramas de Interacción*. Disponible en:
http://www-gris.det.uvigo.es/~avilas/UML/node41.html

18. *Diagramas de Secuencia.* Disponible en:
http://www-gris.det.uvigo.es/~avilas/UML/node42.html

19. *Firewall*. 2007. Disponible en: http://www.desarrolloweb.com/articulos/513.php

20. *W3C*. 2007. Disponible en: http://www.w3c.es/Consorcio/

21. *Arquitectura Cliente – Servidor*. 2007. Disponible en: http://www.csi.map.es/csi/silice/Global71.html

22. *Base de datos*. 2007. Disponible en: http://www.mastermagazine.info/termino/4012.php

23. *Servidor*. 2007. Disponible en: http://www.mty.itesm.mx/rectoria/dda/usols/concepto1.htm

24. *Clases*. 2007. Disponible en: http://pisuerga.inf.ubu.es/lsi/Invest/Java/Tuto/I_1.htm

25. *Funciones*. 2007. Disponible en: http://www.desarrolloweb.com/manuales/20/

26. *Orientado a objetos*. 2007. Disponible en: http://www.monografias.com/trabajos5/softobj/softobj.shtml

BIBLIOGRAFÍA

BLASI, Emanuel. (mayo del 2007): *Resumen de Patrones de Diseño*

CORPORATION, R. S., *Ayuda de Rational Suite 2003. Consultada en enero del 2007.*

JACOBSON, IVAR; RUMBAUGH, JAMES; BOOCH, GRADY: *El proceso unificado de desarrollo*, Editorial Addison Wesley, 2000.

JACOBSON, IVAR; RUMBAUGH, JAMES; BOOCH, GRADY: *El lenguaje unificado de modelado. Manual de referencia*, *Editorial Addison Wesley*, 2000.

LARMAN, CRAIG: *Uml y patrones, Editorial Prentice Hall Iberoamericana*, 1999.

MOLPECERES, A: *Procesos de desarrollo: RUP, XP Y FDD*, 2002.

PRESSMAN, ROGER: *Ingeniería del Software. Un enfoque práctico*, Editorial McGraw-Hill, España, 2002.

MENDOZA SANCHEZ, MARÍA A. *Metodologías de Desarrollo de Software. 2004.*

1