

APLICACIÓN WEB PARA GESTIONAR Y MONITOREAR LA MIGRACIÓN DE DATOS DEL SISTEMA ASSETS AL SISTEMA TRABAJADORES EN LA UCI



AUTORES

Neysis Hernández Díaz
Frank Alain Castro Sierra



TUTOR

Ing. Luis Alberto Pérez Masjuan

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

Ciudad Habana | julio | 2007.
"Año 49 de la Revolución"

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 firmamos la presente a los ____ días del mes de Julio del año 2007.

Frank Alain Castro Sierra

Autor

Neysis Hernández Díaz

Autor

Luis Alberto Pérez Masjuan

Tutor

DATOS DE CONTACTO

Síntesis del Tutor Ing. Luís Alberto Pérez Masjuán

Profesión: Ingeniero informático

Años de graduado: 1

Síntesis del Asesor Ing. José Fidalgo Hidalgo

Profesión: Ingeniero informático

Años de graduado: 1

DEDICATORIA

Neysis

A **mis padres** por su dedicación, paciencia, confianza y por quererme tanto.

A **mi hermana y su bebé** por su cariño insuperable.

A **mi novio** por su amor y su apoyo incondicional.

Frank

A **mi mamá** que hoy no se encuentra conmigo.

AGRADECIMIENTOS

Quiero agradecer en primer lugar a mi mamá por hacerme dado la vida, por estar siempre a mi lado y apoyarme en mis decisiones, a mi papá por su confianza e inculcarme ese espíritu luchador y esa pasión por el trabajo.

A mi hermana por ese cariño tan grande que lo expresa en múltiples peleas, pero que solo me hace quererla cada día más y a mi sobrino querido por ser mi lucecita inspiradora y sacar de vez en cuando ese pedacito de niñez que aún hay en mí.

A mi novio a quien no solo agradezco sus consejos y el apoyo constante, sino además, y sobre todas las cosas su confianza en mí y su eterna espera.

A mis amigas Yoenia, Bisleydis, Ameirys, Ariadne y Katia por ser amigas insustituibles, y compañeras de muchas alegrías y tristezas. A mis compañeros de aula del 9504 por su compañía en estos 5 años tan maravillosos de mi vida.

A Niobis por ser mi amiga, guía y ejemplo en estos años.

A la Revolución por permitirme construir y hacer realidad mis sueños, a la UCI por haberme dado el privilegio de formarme como profesional. A Fidel por sus grandes ideas y por su confianza en los jóvenes. A Dios por ser mi fiel confesor, confidente y por guiar mi camino.

A mi familia por apoyarme siempre en todo.

A todos aquellos que pusieron su granito de arena en mi formación como persona y profesional
MUCHAS GRACIAS.

Neysis

AGRADECIMIENTOS

A mi mamá, mi papá, mi hermana, Misbel y Ariel por siempre apoyarme y darme ánimos para realizar mis estudios.

A mis familiares por apoyarme en mi carrera.

A Camejo, Yenier, Yero y Diuber por ser amigos durante estos años de preuniversitario y Universidad.

A Carlos Manuel Rodríguez Riverol por ser más que amigo hermano y por apoyarme en todo este tiempo.

A mis amigos y compañeros de la Universidad por haber compartido todos estos años juntos.

A Liskeny y Thompson por su ayuda incondicional y sus consejos de diseño.

A Alberto, Sandro, Denis y Pochi por el apoyo en todo momento que los necesité.

A la Revolución y la UCI por formarme como profesional.

A todos los que no menciono y me han ayudado a ser quien soy.

Frank

RESUMEN

El proyecto que aquí se presenta consiste en el desarrollo de una Aplicación Web para el Sistema de migración de datos del Sistema ASSETS al Sistema trabajadores de la Universidad de Ciencias Informáticas (UCI).

La necesidad de desarrollar la aplicación surge con la idea de crear una Ciudad Digital donde la mayoría de los procesos se encuentren automatizados, con este objetivo la institución cuenta con una infraestructura que soporta el desarrollo de sistemas informáticos capaces de gestionar gran cantidad de información con un costo mínimo de tiempo de respuesta y con la posibilidad de integración entre estos.

El documento está estructurado en cinco capítulos que abordan un estudio de los principales conceptos asociados al objeto de estudio, las tecnologías existentes y la seleccionada para el desarrollo de la aplicación, la propuesta de solución descrita en detalles y un estudio e factibilidad y costos.

Con la implantación de la Aplicación Web se da solución a un problema de primer orden del centro, pues dentro de las tareas fundamentales del instituto se encuentra el control y gestión de la información referente a los trabajadores de la Universidad. Esta solución permite al departamento de Informatización de la Universidad llevar un control de las altas, bajas, modificaciones y de los errores ocurridos durante la migración de los datos.

Palabras claves:

- Aplicación Web
- Migración de datos
- Base de datos
- Servicio Web

INDICE

INTRODUCCIÓN	1
Capítulo 1: Fundamentación Teórica.....	6
1.1 Aplicaciones Web	7
1.2 Servicios Web	8
1.2.1 SOAP (Simple Object Access Protocol).....	9
1.2.2 WSDL (Servicios Web Definition Language)	10
1.2.3 UDDI (Universal Description, Discovery and Integration)	10
1.2.4 XML (Extensible Markup Language).....	10
1.2.5 Algunas ventajas de los servicios Web.....	11
1.2.6 Algunas desventajas de la utilización de los servicios Web	11
1.2.7 Modelo de acceso a Servicios Web	12
1.3 Bases de Datos.....	12
1.4 Información	14
1.5 Gestión de la información	15
1.6 Migración de Datos	15
1.7 Sistemas existentes vinculados al campo de acción	16
1.7.1 Sistema ASSETS	16
1.7.2 Sistema Trabajadores	16
1.7.3 Sistema Migración de Datos	17
1.8 Situación Problemática	17
1.9 Solución propuesta	18
Conclusiones	19
Capítulo 2: Tendencias y tecnologías actuales	20
2.1 Herramientas propuestas para el desarrollo de la aplicación.....	21
2.1.1 J2EE (Java 2 Enterprise Edition)	21
2.2 Fundamentación de la plataforma seleccionada: Microsoft .Net	22
2.3 Lenguajes de programación para la Web	24

2.3.1 Personal Home Page (PHP)	24
2.3.2 Active Server Pages (ASP).....	24
2.3.3 ASP.Net	25
2.3.4 Processing Estraction Report Language (PERL).....	25
2.3.5 C#	26
2.3.6 Java Server Pages (JSP).....	26
2.3.7 JavaScript	27
2.3.8 Visual Basic Script (VBS).....	27
2.4 Fundamentación del lenguaje y la tecnología.....	27
2.4.1 Fundamentación de la tecnología utilizada: ASP.Net.....	28
2.4.2 Fundamentación del lenguaje utilizado: C#	29
2.5 Sistemas Gestores de Base Datos (SGBD)	30
2.5.1 SQL Server 2000	30
2.5.2 Oracle	31
2.5.3 MySQL 4.0	32
2.5.4 PostgreSQL	32
2.5 Fundamentación del Sistema Gestor de Base de Datos (SGBD) seleccionado: SQL Server 2000..	33
2.6 Arquitecturas.....	35
2.6.1 Arquitectura 3 capas	35
2.6.2 Arquitectura SOA	36
2.7 Fundamentación de la utilización de arquitectura: Arquitectura de 3 capas.....	37
2.8 Metodologías de desarrollo de software	38
2.8.1 Extreme Programming (XP).....	38
2.8.2 Microsoft Solution Framework (MSF).....	39
2.8.3 Rational Unified Process (RUP).....	40
2.8.4 Scrum.....	40
2.9 Fundamentación de la Metodología de desarrollo de software seleccionada: RUP.....	41
2.9.1 Selección del lenguaje de modelación: Lenguaje de Modelación Unificado (UML Unified Modeling Language)	42
Conclusiones	42

Capítulo 3: Análisis y Diseño del Sistema	43
3.1 Entorno donde trabajará el sistema	44
3.2 Modelo de Dominio	44
3.2.1 ¿Cuándo se aplica un modelo de Dominio?	44
3.2.2 Formas típicas de las clases del dominio	44
3.2.3 ¿Cómo determinar el entorno organizacional? (LARMAN 1999).....	45
3.3 Conceptos principales del entorno.....	45
3.4 Requerimientos	46
3.4.1 Requisitos Funcionales	47
3.4.2 Requisitos No Funcionales	48
3.5 Descripción del sistema. Modelo de Casos de uso del sistema.....	50
3.5.1 Modelo de Caso de Uso del Sistema.....	52
3.5.2 Expansión de los Casos de Uso	52
3.6 Diagramas de Secuencia.....	63
Conclusiones	65
Capítulo 4: Construcción de la solución propuesta	66
4.1 Diagrama de clases.	67
4.2 Diagramas de Clases Diseño Web	71
4.3 Principios de Diseño	73
4.3.1 Estándares de la interfaz de la aplicación.....	73
4.3.2 Tratamiento de excepciones	74
4.3.3 Estándares de Codificación	74
4.3.4 Declaraciones	75
4.3.5 Diseño de la BD	75
4.4 Generalidades de la Implementación	76
4.4.1 Modelo de Despliegue	76
4.4.2 Modelo de Componentes	77
4.4.3 Prueba del sistema propuesto	78
Conclusiones	82
Capítulo 5: Estudio de la factibilidad del proyecto	83

5.1 Planificación.....	84
5.1.1 Objetivos de la planificación.....	84
5.2 Estimación Costos	85
5.3 Beneficios tangibles e intangibles	91
5.3.1 Beneficios tangibles	91
5.3.2 Beneficios intangibles	91
5.4 Análisis de costos y beneficios	93
Conclusiones	93
CONCLUSIONES	94
RECOMENDACIONES	95
BIBLIOGRAFÍA.....	97
ANEXOS	98
GLOSARIO	108

INDICE DE TABLAS

Tabla 1: Usuarios del Sistema	51
Tabla 2: Descripción Caso de Uso Autenticar Usuario.....	52
Tabla 3: Descripción Caso de Uso Gestionar Usuarios.....	53
Tabla 4: Descripción Caso de Uso Buscar Bajas	54
Tabla 5: Descripción Caso de Uso Buscar Altas	57
Tabla 6: Descripción Caso de Uso Buscar Modificaciones	59
Tabla 7: Descripción Caso de Uso Buscar Errores	62
Tabla 8: Descripción Caso de Uso Buscar Reportes.....	62
Tabla 9: Caso Prueba Usuario y contraseña correcta	79
Tabla 10: Usuario incorrecto y contraseña correcta	79
Tabla 11: Usuario incorrecto y contraseña incorrecta	80
Tabla 12: Usuario correcto y contraseña incorrecta	80
Tabla 13: Complejidad de los actores.....	86
Tabla 14: Complejidad de los Casos de Usos	86
Tabla 15: Factor de Complejidad Técnica	87
Tabla 16: Factor Ambiente.....	88
Tabla 17: Estimación del Esfuerzo	90

INDICE DE FIGURAS

Figura 1: Modelo de Dominio.....	46
Figura 2: Diagrama de Casos de Usos del Sistema.....	52
Figura 3: Caso de Uso Buscar Bajas, Escenario Buscar General.....	64
Figura 4: Caso de Uso Buscar Bajas, Escenario Buscar Áreas.....	64
Figura 5: Caso de Uso Buscar Bajas, Escenario Buscar Cargo.....	65
Figura 6: Arquitectura en Capas.....	67
Figura 7: Subpaquete Interfaz Usuario.....	68
Figura 8: Subsistema Gestor de Resultados.....	69
Figura 9: Subsistema Entidades.....	70
Figura 10: Subpaquete Acceso a Datos.....	70
Figura 11: Diagrama Clases Diseño Web Caso de Uso Autenticar.....	71
Figura 12: Diagrama Clases Diseño Web Caso de Uso Buscar Altas.....	71
Figura 13: Diagrama Clases Diseño Web Caso de Uso Buscar Bajas.....	72
Figura 14: Diagrama Clases Diseño Web Caso de Uso Buscar Modificaciones.....	72
Figura 15: Clases Diseño Web Caso de Uso Buscar Errores.....	73
Figura 16: Modelo de Datos.....	76
Figura 17: Modelo de Despliegue.....	77
Figura 18: Modelo de Componentes.....	78
Figura 19: Prueba Caja Negra.....	79
Figura 20: Prueba de caja blanca. Cálculo de la complejidad ciclométrica.....	81

INTRODUCCIÓN

Cuba ha atravesado por más de cuatro décadas una situación difícil, debido al bloqueo genocida implantado por Estados Unidos de América (EUA), el cual ha puesto en práctica disímiles formas de dominación económica, política y militar; esto nos ha privado de un desarrollo tecnológico avanzado en distintas esferas, es por ello que nos vemos en la necesidad de desarrollar con nuestros propios esfuerzos y recursos, logros en el campo de las Tecnologías de la Información y las Comunicaciones (TIC).

Nuestro país se encuentra inmerso en un proceso de informatización de la sociedad, con el propósito de extender el desarrollo informático hacia todos los lugares del país. La Universidad de las Ciencias Informáticas (UCI) constituye una institución abanderada en este sentido, en corto plazo se ha proyectado como la más desarrollada en el uso de las TIC, encaminada a convertirse en la primera “Ciudad Digital” del país; en el marco de esta realidad se han realizado múltiples productos para lograr informatizar todas las áreas del recinto universitario.

Este trabajo resulta de gran importancia para la UCI porque tributa a su Programa de Informatización, el contenido muestra como resultado una Aplicación Web para la gestión y control de la migración de datos.

En la UCI existe un Sistema de migración de datos que permite la importación de los datos entre la base de datos (BD) del ASSETS y la BD de Trabajadores de la institución, este sistema es el encargado de mantener actualizados los datos de los trabajadores, pero no se conoce en detalles las acciones que realiza sino una información general. En las condiciones actuales del funcionamiento de dicho software (SW) surge el **problema** de la inexistencia de una Aplicación Web capaz de lograr la gestión y control de los datos migrados por el Sistema de migración de la UCI.

Después de haber realizado un estudio se plantea como **objeto de estudio** de este trabajo: La gestión de la información en las aplicaciones Web para la visualización de datos, de donde se deriva el **campo de**

acción: Los procesos de gestión de información que surgen como resultado del flujo de la migración de datos en la UCI.

Como **idea a defender** se parte de la siguiente premisa, si se desarrolla una Aplicación Web, basada en un Sistema Gestor de Bases de Datos (SGBD) potente y eficaz como el SQL Server 2000, se podrá identificar cada inserción y eliminación que se realice con los trabajadores de la UCI, se podrá obtener reportes de las acciones realizadas de forma general y los errores ocurridos durante la ejecución del Sistema de migración. Además se brindará un servicio Web a la comunidad universitaria para lograr la reutilización de los métodos implementados en otras aplicaciones que necesiten de este servicio.

De esta manera el **objetivo general** a lograr es: Implementar una Aplicación Web para gestionar y controlar los datos importados diariamente por el Sistema de migración.

Para dar cumplimiento a los objetivos generales se realizarán las siguientes **tareas investigativas**:

- Realizar un estudio detallado del funcionamiento de la aplicación existente.
- Comprobación de la validez del problema existente.
- Estudio del modelo de datos de las bases de datos utilizadas por el Sistema de migración.
- Estudio del proceso de migración de datos y gestión de la información.
- Estudio de la tecnología para el desarrollo de aplicaciones Web.
- Estudio de las metodologías existentes para la modelación del ciclo de vida del software.
- Estudio de los diferentes lenguajes de programación.
- Estudio de la factibilidad, costos y beneficios del software.
- Análisis y estudio del funcionamiento de los servicios Web.
- Análisis y estudio de la arquitectura SOA.
- Análisis y definición de las necesidades de funcionamiento de la interfaz visual.
- Análisis y definición de las necesidades de funcionamiento del servicio Web que brindará la aplicación.
- Selección de la Metodología de Análisis y Diseño de sistemas informáticos, que facilite la creación y garantice la calidad del sistema.

- Selección de las herramientas para llevar a cabo el proyecto y la elección de la plataforma en la que se desarrollará la aplicación.
- Selección del lenguaje de implementación.
- Diseño de una base de datos que soporte las funcionalidades del sistema.
- Definir la forma en que será implementada la aplicación o sea la arquitectura.
- Implementación de una aplicación Web capaz de gestionar la migración de los datos.
- Implementación del servicio Web.
- Validación del sistema.
- Pruebas del sistema.

Es preciso subrayar que cada disciplina, y dependiendo de el tipo de investigación que se realice, determinar cuales serán los métodos de investigación propicios para su objeto de estudio, durante la investigación científica se utilizaron los siguientes métodos.

Métodos Teóricos: Teniendo en cuenta el proceso de análisis y síntesis basados en la posibilidad de descubrir las características generales del objeto de investigación, así como las relaciones esenciales entre ellas, se mencionan los métodos teóricos utilizados.

- “Método Histórico”: Se utiliza este método con el fin de conocer los antecedentes, etapas significativas, funcionamiento del sistema de migración de datos y las necesidades que sustenta.
- “Método Lógico”: Con el conocimiento profundo de la esencia del sistema de migración de datos se analiza la lógica interna de su desarrollo y se expresa la esencia del objeto de estudio.

Métodos Empíricos:

- Entrevistas a expertos del Departamento de Informatización de la UCI en el manejo de la aplicación que hoy existe en el centro, como vía para obtener información precisa acerca del verdadero funcionamiento de dicho software.
- Entrevista a profesores y otras personas que laboran en la UCI, que han presentado problemas en cuanto a las modificaciones de sus datos.

- Análisis de documentos, se basa en la revisión de documentos utilizados durante la investigación. Observación, se realizaron visitas al Departamento de Informatización para observar los procesos, lográndose clasificar y entender la situación problemática.

La herramienta seleccionada para la confección de este trabajo lo constituye Visual Studio .Net 2005. Esta herramienta contiene un conjunto de tecnologías que Microsoft ha integrado en una plataforma común con el objetivo de facilitar el desarrollo de nuevos servicios para el beneficio de los clientes.

Como base del desarrollo de la solución se ha utilizado la Metodología del Proceso Unificado de Desarrollo de Software (RUP). Es un proceso de desarrollo de software que contiene un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software. Utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema software. Garantiza la elaboración de todas las fases de un producto de software orientado a objetos.

Para la creación de la BD se utilizará el SQL Server 2000 que es una solución integral de base de datos y análisis, SQL Server 2000 ofrece el rendimiento, escalabilidad y confiabilidad que se requieren.

Resúmenes de los capítulos a tratar:

Capítulo 1. Fundamentación Teórica: se describen los procesos actuales que se realizan para conocer la migración de los datos desde la BD del ASSETS hacia el Departamento de Informatización en la UCI, para ello se muestran los problemas que existen en la actualidad y algunos sistemas existentes en otras partes, los cuales pueden aportar conocimientos en la posible solución que se plantea.

Capítulo 2. Tendencias y tecnologías actuales: el mismo aborda las tecnologías necesarias para la creación y desarrollo de este trabajo, el Visual Studio .Net herramienta en la cual será desarrollada la nueva aplicación, el Lenguaje Unificado de Modelado (UML) como soporte de la modelación de la solución propuesta y el Proceso Unificado de Desarrollo de Software (RUP) como base en el desarrollo de la solución.

Capítulo 3. Presentación de la solución propuesta: se propone la solución al problema planteado teniendo en cuenta la descripción de los procesos del dominio y la identificación de los casos de uso del sistema.

Capítulo 4. Construcción de la solución propuesta: se construye la solución propuesta al desarrollo de la aplicación pero ya en su sentido más completo, así como el análisis y diseño de los sistemas que incluye todos los diagramas necesarios, la “Vista Estática” del sistema y sus respectivos Diagramas de Clase y Diagramas de Presentación; así como los modelos de “Implementación” y de “Despliegue” para completar la modelación del sistema.

Capítulo 5. Estudio de Factibilidad: se especifica el tipo de estimación realizada en el trabajo así como la planificación, los costos y beneficios tangibles e intangibles.

Capítulo 1

Fundamentación Teórica

En el siguiente capítulo se describen los procesos actuales que se realizan para lograr la migración de los datos, así como los conceptos asociados para un mejor dominio del problema, para ello se muestran los problemas que existen en la actualidad y la necesidad de la creación de una Aplicación Web, que de una forma u otra responda a las exigencias, cada día más crecientes, de responsabilidad, calidad y rapidez en todo proceso de migración y visualización de datos, ya que no es solamente de suma importancia la administración de los datos sino también la manipulación que se le da a los mismos.

1.1 Aplicaciones Web

Con el surgimiento de Internet surgió la necesidad de publicar la información, diversos son los métodos que hoy día existen con este fin. Las aplicaciones Web son las encargadas de manejar el estado del negocio y de gestionar datos almacenados con ayuda de algún Sistema Gestor de Base de Datos (SGBD) con el fin de poder ser observadas por disímiles usuarios.

Una aplicación Web es un sistema informático que los usuarios utilizan accediendo a un servidor Web a través de Internet o de una intranet. (WIKIPEDIA 2007)

Las aplicaciones Web generan dinámicamente una serie de páginas Web en un formato estándar (HTML, XHTML), razón por la cual pueden ser accedidas desde un navegador Web.

En la actualidad con el desenfrenado desarrollo de las TIC, se ha abierto paso a un mundo más desarrollado en cuanto al acceso de la información y la posibilidad de acceder a ella desde cualquier sitio, convirtiéndose en un paso muy importante para la creación de aplicaciones Web rápidas, ligeras y robustas.

La Web fue creada para desplegar información, esta reposa de manera estática en los servidores, la cual es accedida a través de un navegador, valiéndose del protocolo HTTP (Hyper Text Transfer Protocol). En términos más simples, una aplicación Web es un Sistema Web que permite a los usuarios ejecutar la lógica de negocio a través de un Navegador (Browser).

El modo de crear las páginas HTML ha variado mucho, pues se pueden generar páginas dinámicas, para ello se han creado múltiples tecnologías como CGI (Common Gateway Interface) que fue la primera técnica utilizada para que el contenido de las páginas Web se generara de manera dinámica, el Fast-CGI es una solución similar al CGI, solo que propone la creación de un solo proceso persistente por cada programa Fast-CGI en lugar de por cada solicitud del cliente, pero con la aparición de las páginas dinámicas en servidor se entra en una nueva forma de trabajo, la cual está orientada al trabajo del diseñador Web, quien no necesariamente conoce de lenguajes de programación.

Partiendo del funcionamiento de la Web donde el usuario mediante un navegador o cliente hace una petición de una página Web a un servidor Web, encargado de recepcionar dicha petición, analizarla y enviar la respuesta pertinente al cliente en formato HTML a través de la red, se puede definir la Web como una especialización y concreción de las aplicaciones Cliente/Servidor. Hasta ahora se ha planteado como componentes de la arquitectura Web, un navegador o cliente, el servidor Web y la red física que es quien posibilita la comunicación entre los componentes anteriores.

La integración de un nuevo elemento a la arquitectura Web, una aplicación que se ejecuta en el servidor, le brinda a esta arquitectura la potencialidad de manejar la lógica de negocio a través de las Aplicaciones Web, nombre que reciben los sistemas Web que cuentan con esta funcionalidad. Este nuevo elemento es el encargado de controlar el estado del negocio y de gestionar los datos almacenados.

En la actualidad se ha generalizado el uso de aplicaciones Web debido a las grandes posibilidades que brindan y que los clientes sólo necesitan un navegador, capaz de interpretar código con formato HTML, para hacer uso de ellas: no tienen que instalar ningún componente de software adicional.

1.2 Servicios Web

Permiten la comunicación entre aplicaciones sin importar el lenguaje en que este implementado, el sistema operativo, el navegador que se utilice o la ubicación donde se encuentre. Son de gran utilidad porque permiten la publicación de funciones que pueden ser utilizadas por otras aplicaciones a través de los protocolos estándar.

Los servicios Web permiten a las aplicaciones invocar servicios de negocios mediante un mecanismo basado en estándares (usando XML y HTTP). (FERGUSON *et al.* 2003)

Los servicios Web, no son aplicaciones con una interfaz gráfica con la que las personas puedan interactuar, sino que son software accesible en Internet (o en redes privadas que usen tecnologías

Internet) por otras aplicaciones. De esta forma podemos desarrollar aplicaciones que hagan uso de otras aplicaciones que estén disponibles en Internet interaccionando con ellas.(TOLEDANO 2003)

Los servicios Web son fáciles de publicar, invocar y utilizar; los usuarios no necesitan un estudio bien detallado de los mismos, solo es necesario tener un conocimiento básico de su manipulación. El funcionamiento de los servicios Web está basado en un conjunto de estándares que permiten su creación, registro, ubicación y utilización. Además pueden comunicarse entre sí, ya que están basados en XML que puede ser interpretado por cualquier aplicación.

El principal motivo para utilizar servicios Web es que se basan en HTTP sobre TCP (Transmission Control Protocol) en el puerto 80, dado que las organizaciones protegen sus redes mediante firewalls para garantizar la seguridad de que no se filtren en sus sistemas. Este puerto es el que los navegadores usan y los servicios Web se vinculan por el mismo, por la simple razón de que no resultan bloqueados.

Existen tecnologías que han surgido como estándares para los servicios Web, a continuación se hablan de ellas.

1.2.1 SOAP (Simple Object Access Protocol)

Protocolo de Acceso Simple a Objetos: Especificación XML para la formación de los mensajes intercambiados entre los sistemas distribuidos y la red. Este protocolo deriva de un protocolo creado por David Winer, XML-RPC en 1998. Los mensajes debían tener un formato determinado empleando XML para encapsular los parámetros de la petición. El mensaje está compuesto de tres partes: un sobre, encabezado y el cuerpo. El sobre envuelve al mensaje y contiene el encabezado y el cuerpo; el encabezado es un elemento opcional que provee información para el enrutamiento del mensaje; el cuerpo contiene datos etiquetados como XML.

1.2.2 WSDL (Servicios Web Definition Language)

Lenguaje de descripción de servicios Web: Especificación XML para la formación del documento de descripción de un servicio Web. Identifica los métodos, funciones y parámetros necesarios para invocar un determinado servicio. Así, un usuario puede crear una aplicación cliente que comunica con el servicio Web.

1.2.3 UDDI (Universal Description, Discovery and Integration)

Integración, Descubrimiento y Descripción Universal: Es un elemento básico sobre el que se asientan los servicios Web, hace posible que empresas pueden tanto publicar como encontrar servicios Web. UDDI provee un mecanismo para que los negocios se "describan" a si mismos y los tipos de servicios que proporcionan y luego se pueden registrar y publicarse en un Registro UDDI. Tales negocios publicados pueden ser buscados, consultados o "descubiertos" por otros negocios utilizando mensajes con SOAP. La mayoría de los estándares utilizados para crear servicios utilizan XML para describir interfaces de aplicaciones.

1.2.4 XML (Extensible Markup Language)

Lenguaje extensible de etiquetas (extensible por que no es un formato prefijado como HTML): Describe una clase de objetos de datos llamados documentos XML y describe parcialmente el comportamiento de los programas que los procesan.

Es ante todo, un metalenguaje que permite diseñar un lenguaje propio de etiquetas para múltiples clases de documentos, que representa una manera distinta de hacer las cosas, más avanzada, cuya principal novedad consiste en permitir compartir los datos con los que se trabaja a todos los niveles, por todas las aplicaciones y soportes, permitiendo a la información, una gran "portabilidad" a través de plataformas tecnológicas.

Es un estándar para describir datos y crear etiquetas. Las características especiales son la independencia de datos y de la separación de los contenidos de su presentación.

Los documentos XML se componen de unidades de almacenamiento llamadas entidades (entities), que contienen datos analizados (parsed) o sin analizar (unparsed). Los datos analizados se componen de caracteres, algunos de los cuales forman los datos del documento y el resto forman las etiquetas. Las etiquetas codifican la descripción de la estructura lógica y de almacenamiento del documento. XML proporciona un mecanismo para imponer restricciones en la estructura lógica y de almacenamiento.

1.2.5 Algunas ventajas de los servicios Web

- Garantizan el fácil manejo entre aplicaciones de software, independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- 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 firewalls sin necesidad de cambiar sus políticas de seguridad.
- Permiten publicar desde diferentes lugares geográficos disímiles servicios.
- Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar.
- El cliente no necesita conocer nada acerca de la implementación del servicio al que está accediendo, salvo la definición WSDL.
- Pueden funcionar sobre múltiples protocolos de transporte, como por ejemplo HTTP, SMTP o FTP.

1.2.6 Algunas desventajas de la utilización de los servicios Web

- No tienen un alto grado de desarrollo para realizar transacciones como otros estándares abiertos de computación distribuida.
- Pueden esquivar medidas de seguridad basadas en firewalls.

1.2.7 Modelo de acceso a Servicios Web

Existe una serie de actividades esenciales que deben cumplirse para brindar un entorno de servicio Web que permita explotar sus ventajas:

- El servicio Web debe ser creado, sus interfaces y métodos de invocación deben ser definidos.
- Debe ser publicado en repositorios de intranet o de Internet para que los potenciales usuarios los localicen.
- Debe ser localizado para poder ser invocado.
- Debe ser invocado por el usuario cliente.
- Un servicio Web puede ser dado de baja del repositorio si ya no está más publicado.

Por lo tanto una arquitectura de servicios Web requiere tres operaciones fundamentales: publicar, encontrar y utilizar. Los usuarios encuentran los servicios Web publicados y luego los utilizan. Ver Anexo 1

Desde el punto de vista tecnológico la Arquitectura Orientada a Servicios (SOA) juega un papel muy importante a la hora de proveer beneficios a un conjunto de tecnologías alternativas ya que los servicios Web constituyen los cimientos de su construcción y se han transformado en una tecnología de vanguardia sobre la cual están puestas todas las expectativas de la mayor parte de las organizaciones y proveedores para el desarrollo de aplicaciones de negocios.

1.3 Bases de Datos

El desarrollo avanzado de las TIC ha permitido un avance desenfrenado de las bases de datos (BD). En la actualidad gran parte de la gestión de la información en las empresas y organizaciones gira alrededor de las bases de datos. La mayoría de las aplicaciones deben almacenar, recuperar y operar con datos,

siendo este un punto crítico a tener en cuenta a la hora de diseñar un sistema. Los datos cada día se hacen más importantes en las empresas e instituciones.

Desde hace mucho tiempo la mayoría de la información de todo el mundo está contenida en bases de datos, de aquí su merecida importancia y desarrollo vertiginoso para poder estar a la par con las necesidades de administración y transacciones que se realizan a diario, es por ello la necesidad de conocer su concepto e importancia.

Conjunto de datos interrelacionados entre sí, almacenados con carácter más o menos permanente en la computadora. O sea, que una BD puede considerarse una colección de datos variables en el tiempo. (GARCIA 1999)

Conjunto de registros (unidades de información relevante) ordenados y clasificados para su posterior consulta, actualización o cualquier tarea de mantenimiento mediante aplicaciones específicas.(ALFIN 2006)

Conjunto de datos organizados entre los cuales existe una correlación y que están almacenados con criterios independientes de los programas que los utilizan. La filosofía de las bases de datos es la de almacenar grandes cantidades de datos de una manera no redundante y que permita las posibles consultas de acuerdo a los derechos de acceso. (*GLOSARIO DE TÉRMINOS SOBRE ADMINISTRACIÓN PÚBLICA* 2006)

Las bases de datos no son más que la recopilación de datos integrados y organizados almacenados en una computadora, con el fin de permitir su uso a diferentes aplicaciones (sistemas o programas) con múltiples finalidades. Permite un acceso rápido a los datos según su nivel de acceso.

Ventajas de utilizar Bases de datos.

- Normalizar los datos: Permite minimizar la redundancia de los datos, agiliza y garantiza la actualización de los mismos.

- Evitar redundancia de datos: Como política de las bases de datos, estos se tratan de guardar en un único lugar y cuando existe la necesidad de acceder a ellos, se hace por medio de relaciones entre los mismos.
- Evitar inconsistencias de datos: Dado la importancia que generan las bases de datos y que son utilizadas en las organizaciones, se puede garantizar la no existencia de inconsistencias de datos.
- Garantizar la integridad de los datos: Se puede garantizar que la información contenida en las bases de datos es correcta en todo momento.
- Garantizar la seguridad de los datos: Debido a que los accesos a la base de datos tanto para usuarios como para aplicaciones están dados por medio de permisos, se garantiza que nadie sin ser autorizado pueda acceder a la información. Además las vistas también colaboran en el tema de la seguridad y restricciones de accesos.
- Compartir los datos: Dado que todos los datos de una base de datos se almacenan en un mismo lugar lógico (la base de datos) estos se pueden compartir sin problema entre distintos usuarios y aplicaciones.
- Facilidad de modificar los datos: Los datos se guardan en un único lugar, sin redundancia es por ello que a la hora de hacer una modificación es mas fácil de realizar.

Se utilizan estos conceptos en el marco de este trabajo debido al nivel de importancia que encierran las BD en las cuales se almacenan toda la información de la Universidad.

1.4 Información

La información es un mensaje significativo que se transmite de la fuente a los usuarios, es la expresión material del conocimiento con fines de uso. La información está destinada a resolver determinados problemas. Debe estar, entonces, disponible públicamente y servir para el desarrollo individual y corporativo. Se encuentra presente en todos los niveles de actividad y ramas de la economía, la política y la sociedad.

Conocimiento que el sujeto recibe o descubre (fortuitamente o a través de la investigación) y que representa o registra en cualquier forma para su posterior vínculo con otros hechos y fenómenos.(PAZ 1994).

Consiste en la simple recopilación de datos. Conjunto o totalidad de ciertos datos y hechos conocidos que se obtienen como resultado del análisis, síntesis y evaluación de otros datos.(PAZ 1994)

1.5 Gestión de la información

La gestión de información es el proceso que se encarga de suministrar los recursos necesarios para la toma de decisiones, así como para mejorar los procesos, productos y servicios de las organizaciones. Es el proceso mediante el cual se obtiene, despliega o utiliza una variedad de recursos básicos para apoyar los objetivos de la organización. Su utilidad esta dada en función de su aporte a los procesos de toma de decisiones, creación de productos y solución de problemas, entre otros aspectos.(FERNÁNDEZ and SUÁREZ 1996)

Es todo lo relacionado con la obtención de la información adecuada, en la forma correcta, para la persona indicada, al costo adecuado, en el tiempo oportuno, en el lugar apropiado, para tomar la acción correcta. (WOODMAN 1985)

1.6 Migración de Datos

La entrega adecuada de la información no solamente depende de las capacidades del hardware o del software, sino de la capacidad de la organización para poder manipular la información. Una buena importación de los datos proporciona una información eficaz, proporciona a los usuarios información oportuna, precisa e importante.

La migración de BD es un proceso que se realiza para mover o trasladar los datos almacenados en un origen de datos a otro.(CORREA 2006)

La migración de datos no es más que el proceso de traslado de unos datos determinados que se encuentran almacenados hacia otro sitio de destino. Esto es lo que permite la interacción entre dos BD conectadas para obtener en una, la actualización de otra.

1.7 Sistemas existentes vinculados al campo de acción

Existen diferentes sistemas que permiten un mejor entendimiento del objeto de estudio, aportan elementos fundamentales y constituyen un importante eslabón del desarrollo e implementación de la propuesta de solución que se brinda más adelante.

1.7.1 Sistema ASSETS

ASSETS es un Sistema de Gestión Integral estándar y parametrizado que permite el control de los procesos de Compras, Ventas, Producción, Taller, Inventario, Finanzas, Contabilidad, Presupuesto, Activos Fijos, Útiles y Herramientas y Recursos Humanos. Para el control de los recursos humanos tiene una base de datos (BD) única en la cual se almacenan todos los datos del personal que labora en la UCI. Esta BD tiene gran seguridad por la importancia de los datos que almacena.

1.7.2 Sistema Trabajadores

Es el sistema que permite almacenar todos los datos actualizados de los trabajadores existentes en la UCI en una BD, en esta se encuentran los trabajadores que hoy día laboran en la UCI. La actualización de la BD se realiza a través de un Sistema de migración de datos que importa la información de los trabajadores que pertenecen al centro, los trabajadores que prestan servicios a la Universidad (tercerizados y eventuales) se actualizan mediante una aplicación Web destinada a esta funcionalidad. Este es utilizado por la Dirección de Informatización para poder garantizar al trabajador el acceso a los servicios que presta la Universidad, dentro de los cuales se encuentran acceso al comedor, obtención del solapín, etc.

1.7.3 Sistema Migración de Datos

Es un sistema automatizado, realiza la migración de datos desde la BD espejo del ASSETS hacia la BD Trabajadores, realizando consultas de la información que se desea, migrando así los datos transformados desde el sistema ASSETS hacia el de trabajadores, sin mostrar cuales han sido los datos manipulados y a que persona se le han aplicado los cambios, solo muestra datos generales de las cantidades de bajas, altas. Actualiza los datos de la BD trabajadores a través de servicios Web que están implementados con esa función.

1.8 Situación Problemática

Desde la creación de la Universidad de Ciencias Informáticas hasta hoy el personal ha ido en aumento, sobrepasando la cifra de 14 200 personas (trabajadores, estudiantes). Esto trae consigo la necesidad de llevar a cabo un control estricto de los servicios que se prestan a la población universitaria. Se han tomado las medidas necesarias para que no existan libertades en los servicios que se prestan, para ello se utilizan las TIC, método que se impone para poder lograr una sociedad más desarrollada.

Existe un Sistema de Migración de datos (ASSETS-Trabajadores) puesto en marcha para garantizar una migración de los datos seleccionados, hacia la BD Trabajadores, esto se hace porque para garantizar una buena seguridad de la información. No se puede acceder a la BD ASSETS del Departamento de Capital Humano pues ahí se encuentran todos los datos relacionados con los trabajadores incluyendo las nóminas del pago.

Después de haber consultado con los expertos en materia de trabajo y desarrollo del software existente y de haber realizado algunas entrevistas al personal que labora en el centro se puede afirmar que dicho software, que hasta hoy se mantiene en ejecución no es del todo eficiente pues se hace imposible conocer las transformaciones que se han realizado en los datos de algún trabajador que labora en la Universidad, creando así una imposibilidad para garantizar la eficiencia en cuanto a la denegación y

acceso de los servicios de la institución (acceso al comedor, obtención del solapín, entrada y salida al centro) .

También crea algunos imprevistos al cambiar algún trabajador de área, nomenclador o al realizar alguna modificación de los datos del personal existente en la Universidad, pues el Sistema de Trabajadores no puede definir cuales datos han sido modificados o insertados. Para acceder a la información de este sistema es necesario ir a la aplicación o sea que no se puede acceder desde otro sitio.

El software debiera actualizarse diariamente para saber los últimos acontecimientos ocurridos y no lo hace, es por eso que no se sabe el estado de la aplicación como se desea. Cualquier error que no pueda ser detectado a tiempo, crea una situación desagradable en el funcionamiento del centro e insatisfacción en algún trabajador.

1.9 Solución propuesta

Después de haber analizado la problemática planteada, el objeto de estudio y teniendo en cuenta los objetivos, se concluye que de acuerdo a la situación actual en la que se encuentra el sistema de migración de datos se hace sumamente necesario la creación de una Aplicación Web capaz de gestionar y monitorear la migración de datos del sistema ASSETS al sistema Trabajadores la misma contará con un módulo encargado de gestionar la información relacionada con la plantilla de trabajadores de la Universidad, en este caso las bajas, las altas y las modificaciones realizadas a algún trabajador. La información recuperada se almacenará en una BD con vista a brindar reportes relacionados con la migración de datos entre los dos sistemas, los principales reportes que brindará la herramienta serán:

- Funcionamiento del sistema de migración.
 - Hora de ejecución.
 - Errores durante la ejecución
- Modificaciones realizadas a la plantilla de trabajadores de la Universidad.
 - Altas a trabajadores.
 - Bajas a trabajadores.

- Cambios en los datos de los trabajadores.
- Reporte generales por fecha.

La herramienta constará de un módulo para gestionar los permisos a la información gestionada por el mismo y brindará servicios Web que permitan a los usuarios utilizar esta aplicación desde otros sitios. El mismo brindará los servicios implementados por la aplicación para el conocimiento de la información obtenida desde cualquier otra parte de la Universidad.

Conclusiones

En este capítulo se detallaron las condiciones y problemas actuales que rodean al objeto de estudio enmarcado en este trabajo; y a través de los conceptos y definiciones planteadas, se determinaron las condiciones específicas relacionadas al problema. Se ha indagado como se lleva a cabo el negocio para así poder plantear la propuesta de solución.

Capítulo 2

Tendencias y tecnologías actuales

El desarrollo de la tecnología y los medios electrónicos ha ido en aumento en estos últimos años, se ha perfeccionado las herramientas de confección de softwares, las cuales se convierten en una herramienta indispensable para acelerar los procesos de las empresas permitiendo elevar la calidad y eficiencia de la misma.

En este capítulo se relacionan aspectos sobre las herramientas que se han escogido para la implementación y documentación del sistema a desarrollar, teniendo en cuenta el auge desenfrenado que han causado en el mundo actual. Apoyándonos en comparaciones, ventajas y desventajas de las mismas se podrá determinar la propuesta para el funcionamiento de la aplicación Web.

2.1 Herramientas propuestas para el desarrollo de la aplicación.

Se realiza un estudio detallado sobre las diferentes plataformas para la construcción del sistema, sin olvidar los disímiles lenguajes de programación que se pueden utilizar en la Web y los potentes sistemas gestores de bases de datos (BD), teniendo en cuenta comparaciones con otras herramientas y las novedades en este campo se hace el siguiente estudio para determinar las propuestas.

2.1.1 J2EE (Java 2 Enterprise Edition)

Es una plataforma que define un estándar para el desarrollo de las aplicaciones multicapas. Maneja algunas de las funciones de las aplicaciones de forma automática sin necesidad de realizar una programación compleja. Constituye una alternativa para enfrentar las crecientes necesidades de crear aplicaciones multinivel para las organizaciones. J2EE soporta un único lenguaje de programación (Java).

Esta constituye un súper conjunto de Java 2 Standard Edition (J2SE) por lo que tiene todas sus características, entre las que se encuentran la portabilidad basada en el lema “Write Once, Run Anywhere”, acceso a bases de datos basado en la Application Program Interface (API), JDBC y la tecnología CORBA para la interacción con los recursos empresariales existentes así como un modelo de seguridad para la protección de los datos de las aplicaciones.

J2EE ha sido diseñada para aplicaciones distribuidas que son construidas con base en componentes (unidades funcionales de software), los cuales interaccionan entre sí para formar parte de una aplicación J2EE. Un componente de esta plataforma debe formar parte de una aplicación y ser desplegado en un contenedor, o sea en la parte del servidor J2EE que le ofrece al componente ciertos servicios de bajo nivel y de sistema (tales como seguridad, manejo de concurrencia, persistencia y transacciones). Como ves, J2EE no es sólo una tecnología, sino un estándar de desarrollo, construcción y despliegue de aplicaciones. (GUTIÉRREZ 2003)

2.1.2 .Net

Se define como la competencia de J2EE, surgió años más tarde los que la hace más dotada, debido a que ha podido probarse en aplicaciones reales y se han corregido los errores, esto la hace más fiable y segura. Además incorpora tecnologías nuevas y soporta más de 40 lenguajes de programación (C#, C++, ASP, ASP.Net, etc.)

Microsoft. Net constituye el conjunto de tecnologías que ha desarrollado Microsoft en los últimos años con el objetivo de crear una plataforma sencilla y potente que permita distribuir software en forma de servicios, garantizando que puedan comunicarse y combinarse unos con otros de manera totalmente independiente de la plataforma, del lenguaje de programación y de el modelo de componentes con los que hayan sido desarrollados. La infraestructura.Net no es más que el conjunto de todas las tecnologías que conforman el nuevo entorno para desarrollar (tanto servicios Web como aplicaciones tradicionales, aplicaciones de consola, aplicaciones de ventanas, servicios de Windows NT, etc.), y ejecutar aplicaciones escalables y distribuidas.

La plataforma .Net es un nuevo entorno de programación especialmente diseñado para la creación de aplicaciones y de servicios Web. El programador no trabaja directamente contra un sistema operativo, sino que lo hace frente al Common Language Runtime (CLR) o máquina virtual común. Se trata de un programa, que se puede ejecutar, en principio, en cualquier sistema operativo, y que provee una serie de servicios que se pueden usar desde diferentes lenguajes de programación.

2.2 Fundamentación de la plataforma seleccionada: Microsoft .Net

J2EE y .Net son plataformas muy utilizadas para el desarrollo de aplicaciones, ambas presentan características formidables, sin embargo se decide utilizar plataforma .Net debido a las siguientes razones:

- Dispone de Visual Studio. Net calificado con un rasgo innovador pues posee una extensa librería de clases que proporciona soporte para acceso a datos (relacionales y XML), servicios de directorio, colas de mensajes, etc. Simplifica la creación y consumo de servicios Web.

- Facilidad de desarrollar aplicaciones distribuidas basadas en servicios Web.
- La mayor parte de las tareas que odian los programadores se realizan automáticamente, sin requerir un gran esfuerzo.
- Dispone del Framework .Net que es una infraestructura sobre la que se reúne todo un conjunto de lenguajes y servicios que simplifican enormemente el desarrollo de aplicaciones.
- Ofrece un entorno de ejecución altamente distribuido, que permite crear aplicaciones robustas y escalables.
- El código puede ser escrito en cualquier lenguaje de programación compatible con .Net.
- Seguridad de acceso al código pues se puede especificar que una pieza de código solo tenga permisos de lectura de archivos pero no de escritura.
- Es posible aplicar distintos niveles de seguridad al código, de forma que se puede ejecutar código procedente del Web sin tener que preocuparse si esto va a estropear el sistema.
- Garantiza el mantenimiento de las aplicaciones y esto lo hace el Framework .Net de forma automática.

La mejor forma de resumir las características de la plataforma .Net es enumerar los servicios que proporciona el CLR a todas las aplicaciones desarrolladas para la misma. Entre éstas destacan las siguientes:

- Sencillo modelo de programación.
- Tratamiento homogéneo de errores mediante excepciones.
- Desarrollo interlenguaje.
- Ejecución multiplataforma.
- Gestión automática de memoria con recolección de basura.
- Aislamiento de procesos.
- Soporte multihilo.
- Seguridad avanzada basada en el usuario y la procedencia del código.
- Interoperabilidad con código antiguo.

2.3 Lenguajes de programación para la Web

Existen diferentes lenguajes de programación para las aplicaciones Web, cada uno posee sus características específicas. Entre los más sobresalientes se encuentran PERL, ASP, PHP, Java, JSP, los módulos CGI, etc. Del lado del cliente se encuentran principalmente el JavaScript y el Visual Basic Script, que son los encargados de aportar dinamismo a la aplicación en los navegadores. A continuación se presenta una serie de características de los diferentes lenguajes de programación en la Web de más auge en la actualidad.

2.3.1 Personal Home Page (PHP)

Es un lenguaje de programación usado frecuentemente para la creación de contenidos para sitios Web con los cuales se puede programar las páginas HTML y los códigos de fuente. Es un lenguaje de programación del lado del servidor, gratuito e independiente de plataforma, rápido, con una gran librería de funciones, mucha documentación y está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. Actualmente se encuentra en su versión 5.

Con este lenguaje se pueden realizar múltiples tipos de aplicaciones Web debido a la extensa librería de funciones con la que está dotado. La librería de funciones cubre desde cálculos matemáticos complejos hasta tratamiento de conexiones de red, por poner dos ejemplos. Algunas de las más importantes capacidades de PHP son: compatibilidad con las bases de datos más comunes como MySQL, Oracle, Informix.

2.3.2 Active Server Pages (ASP)

Es una tecnología del lado del servidor desarrollada por Microsoft para implementar páginas interactivas en los navegadores Web y poder utilizar diversos componentes ya desarrollados como algunos controles ActiveX. ASP ha logrado llegar a ser el líder en la construcción de aplicaciones de comercio electrónico. Es rápido, eficiente, poderoso y extremadamente fácil de utilizar.

Posee algunos rasgos característicos como reemplazar la forma tradicional de intercambiar información entre usuarios. ASP se ejecuta sobre el servidor y envía datos de regreso al cliente (a través del navegador). ASP es compatible con múltiples plataformas y funciona con IIS (Internet Information Services). En la actualidad ASP cuenta con un potente sucesor ASP.Net.

2.3.3 ASP.Net

Es la última versión de la tecnología ASP pero no es una versión aumentada de esta sino que es enteramente una nueva tecnología para el lado del servidor. ASP.Net es la parte principal del Framework .Net de Microsoft. También contiene un nuevo sistema de entradas de controles orientado a objetos, como las cajas de lista programables y los controles de la validación.

Incluye algunas cosas nuevas como:

- Soporta los mejores lenguajes
- Controles programables.
- Componentes basados en XML
- Autenticación de usuarios con cuentas y roles.
- Alta escalabilidad
- Código compilado
- Fácil configuración y despliegue.
- Utiliza ADO.Net, soporta C#, C++ y Visual Basic (VB) pero no Visual Basic Script (VBS)

2.3.4 Processing Estraction Report Language (PERL)

Es un lenguaje de programación interpretado, muy utilizado para construir aplicaciones CGI para la Web, se trata de un lenguaje de programación muy práctico para extraer información de archivos de texto y generar informes a partir del contenido de los ficheros.

Es un lenguaje de libre uso, es decir, gratuito. Antes estaba muy asociado a la plataforma Unix, pero en la actualidad está disponible en otros sistemas operativos como Windows. Permite efectuar búsquedas de secuencias de caracteres. Perl se utiliza con fuerza en la escritura de procesos CGI (Common Gateway Interface) instalados en un servidor Web o para el desarrollo de procesos de mantenimiento de las actividades de un servidor.

No es muy recomendado para desarrollar procesos de puro cálculo científico o programas que necesitan una gran velocidad y precisión de cálculo o elaboraciones numéricas complejas pero se convierte en una necesidad para quien tenga que manejar un sitio Web que no esté compuesto sólo de texto e imagen.

2.3.5 C#

Es el nuevo lenguaje de propósito general orientado a objetos creado por Microsoft para su nueva plataforma .Net, es un lenguaje bastante joven, sin embargo esto no significa que esté inmaduro, todo lo contrario, al mismo se han incorporado las mejores características de otros lenguajes así como nuevas potencialidades, por otra parte se plantea que el compilador de C# es el más depurado y optimizado de los incluidos en el Framework.Net. Es un lenguaje diseñado para lograr una combinación idónea de simplicidad, expresividad y desempeño eficiente.(SECO 2001)

2.3.6 Java Server Pages (JSP)

Es una tecnología orientada a crear páginas Web con programación en Java. Con JSP se puede crear aplicaciones Web que se ejecuten en variados servidores Web, de múltiples plataformas, ya que Java es en esencia un lenguaje multiplataforma. Las páginas JSP están compuestas de código HTML/XML mezclado con etiquetas especiales para programar scripts de servidor en sintaxis Java. Por tanto, las JSP se pueden escribir con el editor HTML/XML habitual.

2.3.7 JavaScript

Es un lenguaje orientado a eventos e interpretado, es decir, no requiere compilación. El navegador del usuario se encarga de interpretar las sentencias JavaScript contenidas en una página HTML y ejecutarlas adecuadamente.

Es un lenguaje de programación compacto y orientado a objeto destinado al desarrollo de aplicaciones Internet que actúa a modo de complemento del HTML. JavaScript está destinado para la implementación de programas pequeños, es ideal para tareas repetitivas y de control de eventos. (HERNÁNDEZ 2005)

Es un lenguaje orientado a objetos. Aunque el modelo de objetos de JavaScript está reducido y simplificado, incluye los elementos necesarios para que los guiones puedan acceder a la información de una página y puedan actuar sobre la interfaz del navegador. (HERNÁNDEZ 2005)

2.3.8 Visual Basic Script (VBS)

Es un lenguaje script de Microsoft derivado de su paquete de programación orientado a objeto. Es la última incorporación a la familia de lenguajes "Basic" que ya comprende a VBA y sobretodo a Visual Basic. Permite extender los límites del lenguaje HTML, enlazando los scripts a las páginas Web, para que respondan a los eventos provocados por el usuario ejecutando un código.

Microsoft Visual Basic Scripting Edition lleva la ejecución de secuencias de comandos a una variedad de entornos, incluida la ejecución de secuencias de clientes Web en Microsoft Internet Explorer y la ejecución de secuencias de servidores Web en Servicios de Microsoft Internet Information Server(IIS).

2.4 Fundamentación del lenguaje y la tecnología

Luego de haber realizado un estudio detallado y profundo de los diferentes lenguajes de programación y las comparaciones realizadas en cuanto a las características principales de cada uno se hace la siguiente selección: ASP .Net como tecnología incluyendo C# como lenguaje de programación.

2.4.1 Fundamentación de la tecnología utilizada: ASP.Net

ASP .Net es literalmente una nueva plataforma atada íntimamente al sistema operativo para todas sus tareas, lo que lo hace potencialmente tan rápido de ejecutar como cualquier otro código compilado. Eso y su innegable y fácil acceso al resto de los productos Microsoft puede ser una ventaja competitiva.

ASP.Net ofrece varias ventajas importantes acerca de los modelos de programación Web anteriores:

- Eficacia y flexibilidad. Debido a que se basa en Common Language Runtime (CLR), la eficacia y la flexibilidad de toda esa plataforma .Net se encuentra disponible para los programadores de aplicaciones Web.
- Simplicidad. ASP.Net facilita la realización de tareas comunes, desde el sencillo envío de formularios y la autenticación del cliente hasta la implementación y la configuración de sitios.
- Facilidad de uso. Emplea un sistema de configuración jerárquico, basado en texto, que simplifica la aplicación de la configuración al entorno de servidor y las aplicaciones Web.
- Escalabilidad. ASP.Net ha sido diseñado teniendo en cuenta la escalabilidad, con características específicas, con el fin de mejorar el rendimiento en entornos agrupados y de múltiples procesadores.
- Disponibilidad. El motor de tiempo de ejecución controla y administra los procesos de cerca, por lo que si uno no se comporta adecuadamente (filtraciones, bloqueos), se puede crear un proceso nuevo en su lugar, lo que ayuda a mantener la aplicación disponible constantemente para controlar solicitudes.
- Seguridad. Con la autenticación de Windows integrada y la configuración por aplicación, se puede tener la completa seguridad de que las aplicaciones están a salvo.
- Programación multilinguaje. El programador no solamente puede utilizar VBScript o JScript para programar las páginas ASP sino que puede utilizar cualquiera admitido por la plataforma .Net, lo que incluye un gran abanico de lenguajes como C#, Visual Basic.Net, JScript.Net, COBOL.Net, etc.

- Separación lógica-presentación. ASP.Net separa ambas capas debido a una técnica denominada código trasero, con la que pueden separarse en ficheros independientes la presentación y el código de las páginas.
- ASP.Net es independiente del lenguaje, por lo que puede elegir el lenguaje que mejor se adapte a la aplicación.

2.4.2 Fundamentación del lenguaje utilizado: C#

Combina los mejores elementos de múltiples lenguajes de amplia difusión como C++, Java, Visual Basic o Delphi y permite al igual que estos lenguajes programar fácilmente a bajo nivel. Es un lenguaje orientado a objetos, posee una sintaxis similar a la de C++, implementa una fuerte política de seguridad de tipos, elimina la utilización de punteros aunque mantiene una reserva en caso extremo de su uso. Posee mecanismos como los índices y la instrucción foreach, que hacen más fácil e intuitivo el trabajo, elimina la herencia múltiple (ofrece el uso de interfaces) y facilita el trabajo con propiedades y eventos.

A continuación, algunas características de este lenguaje que lo hacen generar componentes de sistema duraderos:

- Gran robustez, gracias a la recolección de elementos no utilizados y a la seguridad en el tratamiento de tipos.
- Seguridad implementada por medio de mecanismos de confianza intrínsecos del código.
- Plena compatibilidad con conceptos de meta datos extensibles.
- Interactúa con otros lenguajes, entre plataformas distintas, y con datos heredados debido a la plena interoperabilidad por medio de los servicios de COM+ 1.0 y .Net Framework con un acceso limitado basado en bibliotecas, compatibilidad con XML para interacción con componentes basados en tecnología Web y capacidad de control de versiones para facilitar la administración y la implementación.

2.5 Sistemas Gestores de Base Datos (SGBD)

Ante la notable demanda de soluciones informáticas para la progresiva informatización de todas las organizaciones con la necesidad de optimizar servicios y productos, han surgido diferentes gestores de bases de datos, estos son programas que permiten manejar la información de modo sencillo y que prestan servicios para el desarrollo y el manejo de bases de datos.

Los SGBD ofrecen un control centralizado de la información teniendo como objetivos evitar la redundancia de los datos, mejorar los mecanismos de seguridad de los mismos y la privacidad, mantener la integridad de los datos realizando las validaciones necesarias y mejorar la eficacia de acceso a los datos.

Los SGBD se pueden definir como un paquete generalizado de software, que se ejecuta en un sistema computacional anfitrión, centralizando los accesos a los datos y actuando de interfaz entre los datos físicos y el usuario. Las principales funciones que debe cumplir un SGBD se relacionan con la creación y mantenimiento de la base de datos, el control de accesos, la manipulación de datos de acuerdo con las necesidades del usuario, el cumplimiento de las normas de tratamiento de datos, evitar redundancias e inconsistencias y mantener la integridad. (GARCIA 1999)

Es necesario conocer las características, ventajas y desventajas de cada sistema gestor de base de datos, es por ello que se debe analizar los productos que más se destacan y de acuerdo a la necesidad, inclinarnos por la mejor opción. Algunas de estas alternativas son Oracle y MySQL, que comercialmente son más fuertes en el mundo del software libre, se aprecian opciones tan completas como SQL Server 2000, gestor muy usado en la Web.

2.5.1 SQL Server 2000

Es un lenguaje que se utiliza para las consultas y programación de la base de datos, para acceder a los datos y para consultar, actualizar y gestionar sistemas de bases de datos relacionales. SQL Server 2000 es uno de los sistemas más rápidos disponibles, integrado y optimizado en el ambiente Windows de Microsoft.

Se instala con un alto nivel de seguridad por defecto, beneficiándose de la seguridad integrada de Microsoft Windows 2000, aunque también soporta la autenticación SQL.

Microsoft SQL Server 2000 es un Servidor de Base de Datos y herramienta de Análisis de la información. Proporciona seguridad, fiabilidad y escalabilidad para poner en marcha cualquier aplicación en un tiempo pequeño, destacando sus sencillas tareas de administración y su capacidad de analizar la información.

SQL Server permite la creación de procedimientos almacenados, los cuales consisten en instrucciones que se almacenan dentro de una base de datos de SQL Server, realizados en lenguaje SQL. Son procedimientos que se guardan semicompilados en el servidor y que pueden ser invocados desde el cliente. Se ejecutan más rápido que instrucciones SQL independientes. (ARSYS 2006)

2.5.2 Oracle

Es básicamente una herramienta cliente/servidor para la gestión de bases de datos. Posee escalabilidad para soportar un gran número de usuarios y cargas de trabajo de alto volumen de transacciones, el optimizador de consultas determina dinámicamente los caminos de acceso y métodos de enlace más eficientes para cada consulta. Además soporta efectivamente ambientes mixtos de carga de trabajo caracterizados por actividades simultáneas de consulta y actualización.

Posee lectura de multiversión sin bloqueo, siempre proporciona a los usuarios resultados consistentes, al mismo tiempo que no penaliza al rendimiento en la actividad concurrente de actualización. También garantiza la autenticidad apropiada de los usuarios y la privacidad e integridad de los datos, y permiten manejar la asignación de privilegios y monitorear las operaciones de la base de datos a lo largo de toda la empresa.

2.5.3 MySQL 4.0

Es un sistema de gestión de base de datos, multihilo y multiusuario, posee código fuente abierto y es muy usado en el mundo. Su ingeniosa arquitectura lo hace extremadamente rápido y fácil de personalizar. Posee recuperación automática ante fallas. MySQL es muy utilizado en aplicaciones Web como MediaWiki o Drupal. MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación.

Funciona sobre múltiples plataformas, además las opciones de índices de texto completo han sido movidas al archivo de configuración de MySQL, así que solamente se tienen que hacer las adecuaciones necesarias y reiniciar MySQL para que los cambios tengan efecto. Muchos fallos en las búsquedas de texto completo han sido corregidos también. En cambio, las tablas usan bloqueo a nivel de filas para mejorar de manera impresionante el rendimiento.

Teniendo en cuenta que MySQL es usado en un gran número de sistemas Web como PostNuke y Slashcode, los desarrolladores de MySQL implementaron una caché de consultas para acelerar las que son ejecutadas comúnmente. Esta caché simplemente almacena las consultas SELECT ejecutadas recientemente y sus resultados en memoria. Utilizando MySQL se pueden definir llaves foráneas entre tablas relacionadas para asegurarse de que un registro no puede ser eliminado de una tabla si aún está siendo referenciado por otra tabla.

2.5.4 PostgreSQL

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

En 1986 otro equipo dirigido por Michael Stonebraker de Berkeley continuó el desarrollo del código de Ingres para crear un sistema de bases de datos objeto-relacionales llamado Postgres. En 1996, debido a un nuevo esfuerzo de código abierto y a la incrementada funcionalidad del software, Postgres fue renombrado a PostgreSQL, tras un breve periplo como Postgres95. El proyecto PostgreSQL sigue actualmente un activo proceso de desarrollo a nivel mundial gracias a un equipo de desarrolladores y contribuidores de código abierto.

PostgreSQL es un servidor de base de datos relacional libre, liberado bajo la licencia BSD. Es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL, Firebird y MaxDB), así como sistemas propietarios como Oracle o DB2. Mediante un sistema denominado MVCC (Acceso concurrente multiversión) PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos.

PostgreSQL soporta funciones que retornan "filas", donde la salida puede tratarse como un conjunto de valores que pueden ser tratados igual a una fila retornada por una consulta. Las funciones pueden ser definidas para ejecutarse con los derechos del usuario ejecutor o con los derechos de un usuario previamente definido.

Además provee soporte para: números de precisión arbitraria, texto de largo ilimitado, figuras geométricas (con una variedad de funciones asociadas), direcciones IP, claves ajenas también denominadas llaves ajenas o llaves Foráneas (foreign keys), vistas, herencia de tablas, tipos de datos y operaciones geométricas.

2.5 Fundamentación del Sistema Gestor de Base de Datos (SGBD)

seleccionado: SQL Server 2000

Luego de analizar y estudiar cada uno de los SGBD descritos anteriormente, se realiza la selección de SQL Server 2000 porque es el mejor SGBD para Windows, es compatible con las tecnologías se utilizarán para el desarrollo de la Aplicación Web y puede aportar muchos beneficios. Dado estas serie de características y teniendo en cuenta que la mayoría de las Bases de Datos (BD) de la Universidad están

sobre este gestor y los desarrolladores de la Aplicación tienen experiencia en el trabajo con este SGBD se reafirma la selección de SQL Server 2000.

SQL Server 2000 es un potente motor de bases de datos de alto rendimiento capaz de soportar millones de registros por tabla con una interfaz intuitiva y con herramientas de desarrollo integradas como .Net, incorpora un modelo de objetos totalmente programable (SQL-DMO) con el se puede desarrollar cualquier aplicación que manipule componentes de SQL Server, es decir, para crear bases de datos, tablas, backups, etc., todo lo que se puede hacer desde el administrador del SQL Server y se puede hacer no solo en Visual C++ sino también en Visual Basic, ASP y por supuesto en .Net. SQL Server ostenta marcas de referencia en cuanto a escalabilidad y confiabilidad, que son críticas para el éxito de bases de datos de gran tamaño.

El SQL Server permite lograr una gran velocidad en el procesamiento de transacciones, y agilidad en todas sus operaciones.

Algunas de sus características principales:

- Las herramientas y los programas se ejecutan mucho más rápido y están diseñados para que produzcan menos impacto en las operaciones del servidor.
- Las copias de seguridad activas, proporcionan copias de seguridad en línea de alto rendimiento con un impacto mínimo en los sistemas en funcionamiento.
- Arquitectura de almacenamiento en disco permite la escalabilidad desde bases de datos de equipos portátiles, también a bases de datos empresariales de tamaño de terabyte.
- El optimizador de consultas con múltiples fases busca el plan óptimo de consultas para mejorar el rendimiento de consultas complejas.
- Permite el acceso y la realización de consultas desde URL a través de HTTP.
- Amplía la búsqueda de texto completo con alto rendimiento a documentos formateados a través de un mecanismo ampliable de filtro.

- Proporciona optimización automática de consultas y soporta consultas en múltiples servidores en la configuración distribuida de base de datos de SQL Server. Aprovecha al máximo las ventajas del hardware del SMP.
- Simplifica la configuración y gestión de un cluster de caídas. Permite que las bases de datos permanezcan online durante la mayoría de las operaciones. Activa backups instantáneos sin servidor y diferenciales.
- Extrae estadísticas mediante el análisis rápido de una muestra, habilitando el optimizador de consultas para utilizar la información más reciente e incrementar la eficacia de las consultas.
- Los servicios de transformación de datos sirven para importar, exportar y transformar datos heterogéneos.

2.6 Arquitecturas

2.6.1 Arquitectura 3 capas

Hace mucho tiempo se hacia innecesario la programación por capas, hoy día es un rasgo innovador en cada aplicación pues es de vital importancia para el desarrollo de aplicaciones robustas, con un repositorio de objetos se ahorra los costos de desarrollo, mantenimiento y se garantizan componentes probados y sólidos, aunque existen programadores que prefieren la eficiencia y calidad del código por encima de la eficiencia del desarrollo. Con la arquitectura de tres capas se garantiza que el usuario nunca conozca lo que está sucediendo verdaderamente detrás de la fachada.

Hasta el momento la arquitectura de 3 capas se proyecta como la más utilizada. Separa los datos de una aplicación, la interfaz de usuario y la lógica en tres componentes distintos. Donde la interfaz de usuario (Capa de Presentación) encierra todo lo que interactúa con los usuarios, aplicaciones externas o servicios, mientras que la lógica de negocios (Capa de Negocios) agrupa cálculos u otros procesos de negocios, ejecución de reglas de negocios, validación de datos relacionados al negocio, flujos y procesos, así como la de acceso a datos (Capa de Servicios de Datos) encapsula la manipulación de datos y la ejecución de las reglas de datos relacionales.

Una buena arquitectura se implementa con el fin de facilitar los requerimientos de mantenimiento, reusabilidad, escalabilidad, y robustez de las aplicaciones para garantizar que los cambios internos en cualquier capa ocasionarán los menores cambios posibles en las capas restantes.

Otra característica importante sería la posibilidad de migrar de servidor de bases de datos con un mínimo de cambios en el sistema, en tal caso, estos se concentrarían en la capa de datos, quizás hubiera que hacer pequeños ajustes en la capa de negocio, pero nunca en la capa de presentación.

2.6.2 Arquitectura SOA

Es necesario abundar sobre esta arquitectura pues el servicio Web que se desarrollará, será integrado a la Arquitectura SOA de la Universidad.

Últimamente cuando se habla de servicios Web se emplea también el término SOA, ¿qué es? SOA es un acrónimo de Arquitectura Orientada al Servicio (Service-Oriented Architecture). SOA es un método para diseñar y construir soluciones software muy independientes (poco acopladas). La funcionalidad sería accesible programáticamente por otras aplicaciones a través de interfaces publicados y que puedan ser descubiertos. Los servicios Web representan una implementación de una Arquitectura Orientada al Servicio.

Básicamente una arquitectura orientada al servicio es una colección de servicios. Estos servicios se comunican entre ellos, la comunicación puede involucrar simplemente el paso de datos o la coordinación de alguna actividad entre varios servicios. Las arquitecturas orientadas a servicios no es algo novedoso. Para mucha gente la primera de estas surgió en el pasado con el uso de DCOM o los ORBs (Object Request Brokers) de CORBA.

Y ¿qué es un servicio? Si una arquitectura orientada al servicio debe ser efectiva, se necesita un entendimiento claro del término servicio. Un servicio es una función que está bien definida, es auto contenido y no depende del contexto o el estado de otros servicios. La tecnología de los servicios Web es la tecnología de conexión más apropiada para las arquitecturas orientadas a servicios.

La revolución de SOA está cambiando la forma en que se consideran cómo se pueden estructurar y administrar las empresas. No obstante, se necesitará más que tecnología para llegar a la futura empresa. A decir verdad, la adopción de SOA está creando nuevos desafíos acerca de cómo se debería regir la orientación del servicio, cómo se lo debería adoptar, cómo afecta culturalmente a una organización y otros asuntos. La Universidad de las Ciencias Informáticas (UCI) tiene como meta para próximos años una integración plena de todas sus aplicaciones hacia SOA.

2.7 Fundamentación de la utilización de arquitectura: Arquitectura de 3 capas

Después de haber realizado un estudio basado en la arquitectura de 3 capas, se alega que es la mejor arquitectura para la implementación de la aplicación Web debido a las serie de características que se describen a continuación:

- Las llamadas de la capa de interfaz al igual que la de lógica del negocio son más flexibles ya que sólo se necesita transferir parámetros a la capa intermedia.
- Aplicaciones más robustas debido al encapsulamiento.
- Mantenimiento y soporte más sencillo (es más sencillo cambiar un componente que modificar una aplicación completa).
- Mayor flexibilidad (se pueden añadir nuevos módulos para dotar al sistema de nueva funcionalidad).
- Alta escalabilidad. La principal ventaja de una aplicación distribuida bien diseñada es su buen escalado, es decir, que puede manejar muchas peticiones con el mismo rendimiento simplemente añadiendo más hardware. El crecimiento es casi lineal y no es necesario añadir más código para conseguir esta escalabilidad.
- Con la arquitectura de tres capas, la interfaz del cliente no es requerida para comprender o comunicarse con el receptor de los datos. Por lo tanto, esa estructura de los datos puede ser modificada sin cambiar la interfaz del usuario.

- El código de la capa intermedia puede ser reutilizado por múltiples aplicaciones si está diseñado en formato modular. Esto puede reducir los esfuerzos de desarrollo y mantenimiento, así como los costos de migración.
- La separación de roles en tres capas, hace más fácil reemplazar o modificar una capa sin afectar a los módulos restantes.
- Separando la aplicación de la base de datos, hace más fácil utilizar nuevas tecnologías de agrupamiento y balance de cargas.
- Separando la interfaz del usuario de la aplicación, libera de gran procesamiento a la estación de trabajo y permite que las actualizaciones de la aplicación sean centralizadas en el servidor de aplicaciones.
- Los ambientes de tres capas pueden incrementar el tráfico en la red y requerir más balance de carga y tolerancia a las fallas.

2.8 Metodologías de desarrollo de software

2.8.1 Extreme Programming (XP)

Es una de las metodologías de software más exitosas en la actualidad. Se utiliza para proyectos a corto plazo. El producto final es fabricado por pocos desarrolladores y tiene dentro de sus principios involucrar en el equipo de trabajo al cliente. Tiene como unos de sus objetivos fundamentales la entrega del producto en corto tiempo. *Ver Anexo 2*

XP tiene las siguientes características:

- Pruebas Unitarias: Se basa en las pruebas realizadas a los principales procesos, realiza pruebas de las fallas que pudieran ocurrir, es como si se adelantara a obtener los posibles errores.
- Refabricación: Se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.

- Programación en pares: Propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento.

Lo fundamental en este tipo de metodología es:

- La comunicación, entre los usuarios y los desarrolladores
- La simplicidad, al desarrollar y codificar los módulos del sistema
- La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales

2.8.2 Microsoft Solution Framework (MSF)

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.(SANCHEZ 2006)

MSF tiene las siguientes características:

- Adaptable: Usado en cualquier parte del mundo pero su uso es limitado a las características específicas de un lugar.
- Escalable: Puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas a más.
- Flexible: Es utilizada en el ambiente de desarrollo de cualquier cliente.
- Tecnología Múltiple: porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

La Metodología MSF se adapta a proyectos de cualquier dimensión y de cualquier tecnología, se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un

proyecto, Modelo de Arquitectura del Proyecto, Modelo de Equipo, Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación. *Ver Anexo 3*

2.8.3 Rational Unified Process (RUP)

El Proceso Unificado es un proceso de desarrollo de Software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Sin embargo, el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organización, diferentes niveles de aptitud y diferentes tamaños de proyecto. Ver

El Proceso Unificado está basado en componentes, lo cual quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas. Es un proceso que define claramente quien, cuando y que debe hacerse, y como su enfoque está basado en modelos utiliza un lenguaje bien definido para tal fin, el UML.(EVANS 2003)

La metodología RUP esta dividida en cuatro fases de desarrollo del producto:

- Inicio: El Objetivo en esta etapa es determinar la visión del proyecto.
- Elaboración: En esta etapa el objetivo es determinar la arquitectura óptima
- Construcción: En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial, o sea crear el producto.
- Transición: El objetivo es llegar a obtener el release del proyecto

Ver Anexo 4

2.8.4 Scrum

Scrum divide un proyecto en iteraciones (que ellos llaman carreras cortas) de 30 días. Antes de que comience una carrera se define la funcionalidad requerida para esa carrera y entonces se deja al equipo para que la entregue. El punto es estabilizar los requisitos durante la carrera.

Sin embargo la gerencia no se desentiende durante la carrera corta. Todos los días el equipo sostiene una reunión corta (quince minutos), llamada scrum, donde el equipo discute lo que hará al día siguiente, en particular muestran a los bloques de la gerencia (los impedimentos para progresar que se atraviesan y que la gerencia debe resolver). También informan lo que se ha hecho para que la gerencia tenga una actualización diaria de dónde va el proyecto. (SCHWABER and BEEDLE 2005)

2.9 Fundamentación de la Metodología de desarrollo de software seleccionada: RUP

Se puede concluir además, que lo más importante antes de elegir la metodología que se usará para la implementación de un software, es determinar el alcance que tendrá y luego de ahí ver cual es la que más se acomoda a la aplicación que se desea implementar.

Luego de haber analizado las diferentes Metodologías de desarrollo de software antes descritas se arriba a la conclusión de utilizar RUP porque sin dudas es más adaptable para proyectos de largo plazo.

RUP además incorpora el concepto de "mejores prácticas" para la ingeniería de software, definido por cinco características fundamentales: (SCHWABER and BEEDLE 2005)

- Dirigido por casos de uso. El desarrollo está dirigido a satisfacer las necesidades de los usuarios del sistema expresadas en casos de uso.
- Centrado en la arquitectura. El desarrollo se centra en una arquitectura bien definida, con relaciones claras entre sus distintos componentes.
- Iterativo. El problema y la solución se organizan en pequeñas piezas, de manera que cada iteración se dirige específicamente al desarrollo de un conjunto de ellas.
- Incremental. Cada iteración se construye sobre la base creada por las iteraciones anteriores, agregándole capacidades al sistema.
- Controlado. El proceso se planifica y en cada momento está claro lo que debe hacerse.

2.9.1 Selección del lenguaje de modelación: Lenguaje de Modelación Unificado (UML Unified Modeling Language)

Es un lenguaje gráfico para visualizar, especificar, construir y documentar cada una de las partes que comprende el desarrollo de software. Posee formas de modelar conceptos como por ejemplo las funciones del sistema, además de otras particularidades como la de escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables. Usa procesos de otras metodologías, aprovechando la experiencia de sus creadores. Es un lenguaje de modelado orientado a objetos.

El UML es un lenguaje para la construcción de modelos; no guía al desarrollador en la forma de realizar el análisis y diseño orientados a objetos ni le indica cual proceso de desarrollo adoptar.(LARMAN 1999)

UML prescribe un conjunto de notaciones y diagramas estándar para modelar sistemas orientados a objetos, describe la semántica esencial de lo que estos diagramas y símbolos significan. Mientras que ha habido muchas notaciones y métodos usados para el diseño orientado a objetos, ahora los modeladores sólo tienen que aprender una única notación.

Conclusiones

Luego de haber realizado un estudio de las tecnologías y de haber plasmado las características fundamentales de cada una de ellas y teniendo en cuenta las estructuras de la Universidad de las Ciencias Informática en cuanto las tecnologías que se utilizan, se puede afirmar que la selección realizada es la más óptima para proporcionar una solución eficiente al problema planteado.

Capítulo 3

Análisis y Diseño del Sistema

En el siguiente capítulo se realiza la descripción de la solución propuesta. Se describe la utilización de un modelo de dominio, puesto que el sistema no da soporte para negocio alguno debido a que el sistema es muy engorroso y además no posee interfaz visual. Este modelo de dominio contiene la información necesaria, o sea las principales entidades del dominio y sus relaciones para un mayor entendimiento.

Además se enumeran los requisitos Funcionales y No Funcionales que debe tener el sistema que se propone, lo que permite hacer una concepción general del sistema, e identificar mediante un Diagrama de Caso de Uso, las relaciones de los actores que interactúan con el sistema y las secuencias de acciones que se ejecutan.

3.1 Entorno donde trabajará el sistema

El entorno donde funcionará la aplicación Web es en el Departamento de Informatización de la Universidad, pues es precisamente este departamento el encargado de informatizar todas las áreas de dicho centro, este departamento necesita la aplicación Web debido a que los problemas que actualmente existen con la información que se maneja del personal que labora en la institución es debido a que no se puede realizar las actualizaciones de los datos de manera ágil debido a las ineficiencias del sistema de migración de datos.

3.2 Modelo de Dominio

Un modelo del dominio describe los conceptos importantes del contexto como objetos del dominio y enlaza estos objetos unos con otros. La identificación y la asignación de un nombre para estos objetos ayudan a desarrollar un glosario de términos que permitirán comunicarse mejor a todos los que están trabajando en el sistema. Más adelante, los objetos del dominio ayudan a identificar algunas de las clases que se analizan y diseñan en el sistema (*GLOSARIO DE TÉRMINOS SOBRE ADMINISTRACIÓN PÚBLICA 2006*)

3.2.1 ¿Cuándo se aplica un modelo de Dominio?

- Los flujos de información son difusos (múltiples orígenes, solo eventos, sucesos).
- Imposibilidad de determinar subsistemas (exceso de interconexiones).
- Solapamiento de responsabilidades.
- Múltiples responsabilidades.
- Difícil establecimiento de reglas de funcionamiento.

3.2.2 Formas típicas de las clases del dominio

- Objetos del entorno que representan cosas que se manipulan.
- Objetos del mundo real y conceptos de los que el futuro sistema debe de hacer seguimiento.

- Sucesos que ocurrirán o han ocurrido.

Un modelo conceptual es una descripción del dominio de un problema real, no es una descripción del diseño del software.

3.2.3 ¿Cómo determinar el entorno organizacional? (LARMAN 1999)

- Determinar organigrama.
- Determinar eventos.
- Determinar objetos reales.
- Determinar conceptos.
- ¿Quién o quienes participan?

La Aplicación Web a desarrollar, permitirá a los usuarios seleccionar la información que deseen visualizar así como introducir un patrón de búsqueda (rango de fecha, cargo, dirección, municipio, categoría docente, si es becado o no, antigüedad, etc.) y la posibilidad de conocer los reportes generados por el sistema para obtener una información más general de los datos.

3.3 Conceptos principales del entorno

Universidad: Es el centro donde se desarrolla el entorno organizacional.

Departamento de Informatización: Es el encargado de informatizar todas las áreas.

Departamento de Capital Humano: Es el encargado de registrar a todos los trabajadores.

BD ASSETS: Almacena todos los datos de los trabajadores.

Sistema Trabajadores: Es quien tiene la información actualizada de los trabajadores.

Sistema de Migración: Es quien migra los datos de los trabajadores de la BD ASSET a la BD Trabajadores.

Reporte: son los reportes generales que emite el Sistema de Migración

WS Trabajadores: Mediante él se actualizan los datos en la BD Trabajadores.

El modelo del dominio se describe mediante diagramas UML, específicamente con un diagrama clases conceptuales significativas en el dominio del problema.

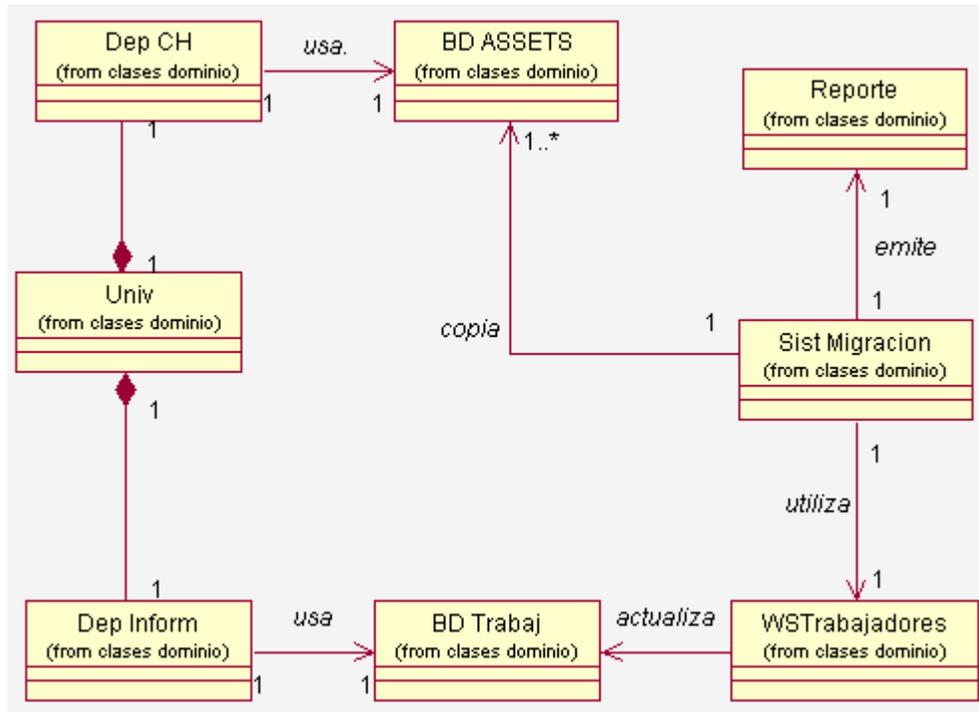


Figura 1: Modelo de Dominio

3.4 Requerimientos

Antes de empezar un proyecto es necesario saber cuales son los objetivos y funcionalidades que debe cumplir. La fase de análisis consiste en detectar estas funcionalidades y expresarlas en forma de requerimientos. El análisis de requerimientos es la primera etapa de un proyecto software, en ella se tratan de definir las condiciones o capacidades necesarias para uno o varios usuarios con el fin de solucionar un problema o conseguir un objetivo.

Para la creación global del sistema se necesita comprender todos los objetivos y necesidades del usuario. En primer lugar, se ha de especificar el comportamiento externo del sistema desde el punto de vista del usuario, en forma de requisitos. La determinación de los requerimientos se haya en base a la experiencia

de hablar con los usuarios finales sobre sus necesidades y/o analizando el sistema software existente. Los requerimientos de usuario se pueden expresar en lenguaje natural, organizados por categorías.

Después de haber enfatizado en los conceptos importantes que rodean al objeto de estudio, se puede comenzar a analizar ¿Qué funciones debe cumplir el sistema para que se cumplan los objetivos planteados? Para ello existen dos tipos de requerimientos:

- **Funcionales:** Las funciones que el sistema deberá ser capaz de realizar. Dentro de ellos se incluyen las acciones que podrán ser ejecutadas por el usuario, las acciones ocultas que debe realizar el sistema, y las condiciones extremas a determinar por el sistema.
- **No Funcionales:** Las condiciones extremas a determinar por el sistema, además las características que debe tener la aplicación. Ejemplo: sistema operativo, sistema gestor de base de datos, sistema de archivos, ergonómicos (interfaz gráfica, etc.), de rendimiento, de tiempo, formato de entrega, etc.

3.4.1 Requisitos Funcionales

R1. Autenticar usuario

R 1.1 Comparar nombre de usuario y contraseña.

R 1.2 Mostrar opciones según el rol que tenga asignado.

R2. Gestionar datos de usuarios.

R2.1 Actualizar usuarios que interactúan con el sistema.

R3. Buscar y visualizar reporte de la ejecución de la aplicación dado un rango de fechas.

R4. Buscar y visualizar errores ocurridos durante la ejecución del sistema teniendo en cuenta la fecha escogida.

R5. Buscar y visualizar bajas realizadas dado un rango de fecha.

R6. Buscar y visualizar altas realizadas dado un rango de fecha.

R7. Buscar y visualizar modificaciones realizadas dado un rango de fecha.

3.4.2 Requisitos No Funcionales

Usabilidad:

- El sistema debe exigir a los usuarios autenticarse antes de poder usar sus funcionalidades.
- El sistema podrá ser utilizado solo por las personas autorizadas por el administrador.
- El sistema debe proporcionar, dado un usuario las funcionalidades a que tiene acceso en la aplicación.
- Facilidad de uso.
- Además garantizará una conexión rápida y segura con la base de datos que contendrá la información, lo que permitirá facilidades de actualización y mantenimiento desde cualquier lugar.

Rendimiento:

- El sistema debe ser capaz de brindar la información necesaria y mostrar la información con el criterio de búsqueda seleccionado por el usuario (rango de fecha, cargo, dirección, municipio, categoría docente, si es becado o no, antigüedad, etc.).
- La eficiencia del producto estará determinada por la velocidad de transmisión de la red y su nivel de congestión, además de la capacidad de procesamiento del servidor de datos.

Seguridad:

- El ingreso a la aplicación será mediante el mecanismo de autenticación de usuarios, permitiendo acceder solamente al personal autorizado a utilizar el mismo.
- Una vez autenticado se accederá exclusivamente al módulo que se le autorizó, según el rol que desempeña para garantizar que la información sea vista y actualizada solamente por quien tiene derecho a ella.
- El sistema debe permitir a los administradores agregar usuario, eliminar y modificar además que puedan ejecutar sus permisos (total o parcialmente) ante determinadas situaciones.

- Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos, las personas que no están registradas no podrán acceder al sistema.

Utilización de recursos:

- El sistema utilizará BD para guardar todos los datos sobre las bajas, altas y también para generar el reporte de errores en general.
- Capacidad de respuesta, obtener los resultados a mostrar en el menor tiempo posible.

Disponibilidad:

- El sistema estará disponible las 24 horas del día, solo se actualizará BD una sola vez al día.

Soporte:

- El sistema permitirá su extensibilidad, pudiéndosele agregar nuevos módulos o modificar los existentes para lograr mejores prestaciones en el momento que se requieran realizar cambios.
- Las pruebas del sistema se realizarán en la UCI, las mismas permitirán evaluar en la práctica las funcionalidades y las ventajas de este nuevo producto, así como los errores.
- El Departamento de Informatización será el encargado tanto de la publicación como del mantenimiento de la aplicación. Además garantizará la integridad de los datos que se almacenan en la base de datos ya que es información relacionada con el personal de la Universidad.
- La información deberá estar disponible a los usuarios en todo momento, limitada solamente por las restricciones que estos tengan de acuerdo a la política de seguridad del sistema.

Software:

- .Net Framework Redistributable. Requerimientos mínimos: Microsoft Windows 2000, Windows XP Professional, Microsoft Windows XP, Microsoft Windows NT 4.0.
- Microsoft SQL Server 2000.

- Servidor con Internet Information Service (IIS).

Hardware:

- Procesador: Intel Pentium class, 90 MHz o mayor.
- Memoria RAM: es aconsejable más de 32 megabytes (MB).
- Disco duro: 50 GB de espacio de disco duro.
- Buen funcionamiento de la red soportando hasta 100 Mbps de velocidad.

Interfaz o Apariencia:

- El sistema debe tener una apariencia profesional, sin gran cantidad de imágenes, debe ser ágil, intuitiva, muy legible y simple de usar, teniendo en cuenta las características de los usuarios hacia los cuales va dirigido el mismo.
- Deberá ser consistente con el mundo real de manera que los conceptos manejados sean conocidos y familiares por los usuarios.
- La interfaz estará diseñada de modo tal que el usuario pueda tener en todo momento el control de la aplicación, lo que le permitirá ir de un punto a otro dentro de ella con gran facilidad.
- Se garantizará que la aplicación sea lo más interactiva posible. Se usarán los colores claros como: azul y blanco.

3.5 Descripción del sistema. Modelo de Casos de uso del sistema.

Después de haber realizado una descripción del sistema, se da paso a la descripción del modelo de casos de uso del sistema haciendo uso de las facilidades que brinda UML, se formulan las funcionalidades del sistema y representación mediante un diagrama, para ellos es de vital importancia definir los actores y los casos de uso que representarán las responsabilidades del mismo.

Un caso de uso constituye una técnica utilizada para describir el comportamiento del sistema, a través de un documento narrativo que define la secuencia de acciones que obtienen resultados de valor para un actor que utiliza un sistema para completar un proceso, sin importar los detalles de la implementación.

Un caso de uso es un documento narrativo que describe la secuencia de un actor (agente externo) que utiliza un sistema para completar un proceso.

Los actores se definen como los roles que puede tener un usuario, pueden ser humanos, otros sistemas, máquinas, hardware, etc. que interactúan con un sistema para de esta forma intercambiar datos, aunque en algunos casos pueden constituir un recipiente pasivo de información. (MOLINA and IBÁÑEZ 2001)

Un actor no es parte del sistema, es un rol de un usuario, que puede intercambiar información o puede ser un recipiente pasivo de información y representa a un ser humano, a un software o a una máquina que interactúa con el sistema. En este caso con el sistema interactúan cinco actores que se definen a continuación:

En este sistema interactúan 3 actores que se definen a continuación.

Tabla 1: Usuarios del Sistema

Actores	Justificación
Administrador	Es el administrador del sistema, es quien autoriza a los usuarios (especialistas del departamento de informatización) a tener acceso a la información de la aplicación.
Especialista	Es un usuario generalizado, especialista del Departamento de Informatización que podrá ver la información que brinda la aplicación.
Sistema Migración	Es un sistema que actualiza la base de datos de la aplicación Web.

3.5.1 Modelo de Caso de Uso del Sistema

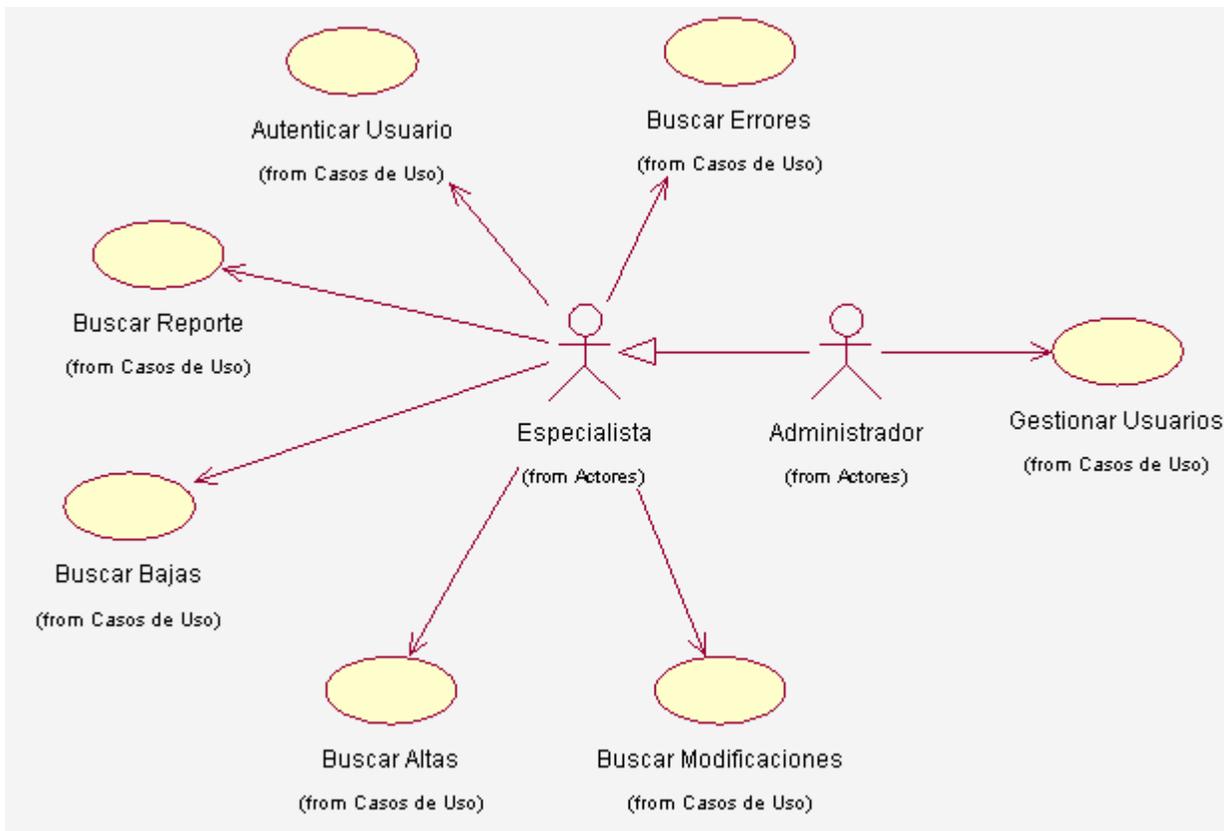


Figura 2: Diagrama de Casos de Usos del Sistema

3.5.2 Expansión de los Casos de Uso

Tabla 2: Descripción Caso de Uso Autenticar Usuario

Caso de Uso	Autenticar Usuario.
Actores	Especialista
Propósito	Permitir autenticarse.
Resumen	El Caso de Uso se inicia cuando el usuario introduce los datos(usuario y contraseña), estos se verifican y termina cuando se le da permiso o no.
Referencias	R1,R2
Precondiciones	

Curso Normal de Eventos	
Acción del Actor	Respuesta del Sistema
	Muestra los controles necesarios para que el usuario introduzca su usuario y contraseña.
1. Introduce los datos requeridos (usuario y contraseña) y acepta la operación.	2. Valida los datos introducidos por el usuario. En caso de que los datos no sean correctos, ver CA1.
	3. Muestra la información a la que tiene acceso según su rol.
Curso Alterno CA1.	
Muestra mensaje de error y retorna al paso 1.	
Poscondiciones	Se permite el acceso a funcionalidades según los privilegios
Requisitos Especiales	
Prioridad	Crítico
Interfaz	Ver Anexo 6

Tabla 3: Descripción Caso de Uso Gestionar Usuarios

Caso de Uso	Gestionar Usuario
Actores	Administrador
Propósito	Dar permisos de acceso al sistema, actualizar usuarios y asignar roles.
Resumen	El caso de uso comienza cuando el administrador abre el sistema para actualizar un usuario, realiza la actividad deseada y termina el caso de uso cuando envía la información.
Referencias	R2,R3
Precondiciones	
Curso Normal de Eventos	
Acción del Actor	Respuesta del Sistema
	Muestra los controles necesarios para que el Administrador registre o actualice los datos de un usuario.
1. El Administrador selecciona la opción Insertar usuario, si selecciona la opción Cambiar Rol CA1.	2. Muestra los controles necesarios para insertar un nuevo usuario.
3. Introduce los datos solicitados en los campos mostrados y acepta la operación.	4. Agrega el usuario al sistema.

	5. Muestra un mensaje de terminada la operación.
Curso Alterno CA1	
	2. Muestra la información de los usuarios del sistema.
3. Selecciona la opción de Cambiar Rol. Si selecciona la opción Eliminar ver CA2.	4. Muestra las opciones a cambiar.
5. Cambia los datos necesarios y acepta la operación.	6. Muestra un mensaje de si desea aceptar o no la operación.
	7. Actualiza los datos del usuario.
	8. Muestra un mensaje de terminada la operación.
Curso Alterno CA2	
	2. Muestra un mensaje, si desea realizar o no esta operación. En caso de no aceptar CA3.
	3. Elimina el usuario.
	5. Muestra un mensaje de terminada la operación.
Curso Alterno CA3	
Retorna al paso 2 del CA1	
Poscondiciones	Se ha cambiado el rol de un usuario del sistema.
Prioridad	Secundario
Interfaz	Ver Anexo 7

Tabla 4: Descripción Caso de Uso Buscar Bajas

Caso de Uso	Buscar Bajas
Actores	Especialista
Propósito	Mostrar las bajas solicitadas.
Resumen	El caso de uso se inicia cuando un usuario selecciona la opción de observar bajas y termina con la visualización de la información que desea observar.
Referencias	
Precondiciones	Estar autenticado
Curso Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Selecciona opción de mostrar bajas	2. Muestra el listado de las diferentes búsquedas que puede realizar.

<p>3. Selecciona la opción de búsqueda de bajas generales. Si selecciona la opción de búsqueda por dirección ver CA1, Si selecciona la opción de búsqueda por áreas ver CA2, Si selecciona la opción de búsqueda por cargo ver CA3, Si selecciona la opción de búsqueda por sexo ver CA4, Si selecciona la opción de búsqueda por color de la piel ver CA5, Si selecciona la opción de búsqueda por municipio ver CA6, Si selecciona la opción de búsqueda por categoría docente ver CA7, Si selecciona la opción de búsqueda por antigüedad ver CA8, Si selecciona la opción de búsqueda del personal docente ver CA9, Si selecciona la opción de búsqueda si es becado CA10,</p>	<p>4. Muestra los controles para seleccionar el rango de fecha.</p>
<p>5. Selecciona la fecha deseada y acepta la operación.</p>	<p>6. Muestra la información solicita por el usuario.</p>
<p>Curso Alterno CA1</p>	
	<p>4. Muestra los controles para seleccionar el rango de fecha y el tipo de dirección de trabajo.</p>
<p>5. Selecciona la fecha, dirección deseada y acepta la operación.</p>	<p>6. Muestra la información solicita por el usuario.</p>
<p>Curso Alterno CA2</p>	
	<p>4. Muestra los controles para seleccionar el rango de fecha y el tipo de área de trabajo.</p>
<p>5. Selecciona la fecha, área deseada, acepta la operación y retorna al paso 6.</p>	
<p>Curso Alterno CA3</p>	
	<p>4. Muestra los controles para seleccionar el rango de fecha y el tipo de cargo.</p>
<p>5. Selecciona la fecha, cargo deseado, acepta la operación. Luego paso 6.</p>	
<p>Curso Alterno CA4</p>	
	<p>4. Muestra los controles para seleccionar el rango de fecha y el tipo de sexo.</p>
<p>5. Selecciona la fecha, sexo deseado, acepta la operación y retorna al paso 6.</p>	
<p>Curso Alterno CA5</p>	

	4. Muestra los controles para seleccionar el rango de fecha y el color de la piel.
5. Selecciona la fecha, el color de la piel deseado, acepta la operación y retorna al paso 6.	
Curso Alterno CA6	
	4. Muestra los controles para seleccionar el rango de fecha y el municipio donde radica.
5. Selecciona la fecha, municipio, acepta la operación y retorna al paso 6.	
Curso Alterno CA7	
	4. Muestra los controles para seleccionar el rango de fecha y la categoría docente.
5. Selecciona la fecha, categoría docente deseada, acepta la operación y retorna al paso 6.	
Curso Alterno CA8	
	4. Muestra los controles para seleccionar el rango de fecha y la antigüedad en la profesión.
5. Selecciona la fecha, los años de antigüedad en la profesión, acepta la operación y retorna al paso 6.	
Curso Alterno CA9	
	4. Muestra los controles para seleccionar el rango de fecha y si es docente.
5. Selecciona la fecha y si es docente, acepta la operación y retorna al paso 6.	
Curso Alterno CA10	
	4. Muestra los controles para seleccionar el rango de fecha y si es becado en la universidad.
5. Selecciona la fecha y si es becado, acepta la operación y retorna al paso 6.	
Poscondiciones	Se muestra la información solicitada por el usuario.
Requisitos Especiales	
Prioridad	Secundario
Interfaz	Ver Anexo 8

Tabla 5: Descripción Caso de Uso Buscar Altas

Caso de Uso	Buscar Altas	
Actores	Especialista	
Propósito	Mostrar las altas solicitadas.	
Resumen	El caso de uso se inicia cuando un usuario selecciona la opción de observar altas y termina con la visualización de la información que desea observar.	
Referencias		
Precondiciones	Estar autenticado	
Curso Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. Selecciona opción de mostrar altas	2. Muestra el listado de las diferentes búsquedas que puede realizar.	
3. Selecciona la opción de búsqueda de altas generales. Si selecciona la opción de búsqueda por dirección ver CA1, Si selecciona la opción de búsqueda por áreas ver CA2, Si selecciona la opción de búsqueda por cargo ver CA3, Si selecciona la opción de búsqueda por sexo ver CA4, Si selecciona la opción de búsqueda por color de la piel ver CA5, Si selecciona la opción de búsqueda por municipio ver CA6, Si selecciona la opción de búsqueda por categoría docente ver CA7, Si selecciona la opción de búsqueda por antigüedad ver CA8, Si selecciona la opción de búsqueda del personal docente ver CA9, Si selecciona la opción de búsqueda si es becado CA10,	4. Muestra los controles para seleccionar el rango de fecha.	
5. Selecciona la fecha deseada y acepta la operación.	6. Muestra la información solicita por el usuario.	
Curso Alterno CA1		
	4. Muestra los controles para seleccionar el rango de fecha y el tipo de dirección de trabajo.	
5. Selecciona la fecha, dirección deseada y acepta la operación.	6. Muestra la información solicita por el usuario.	
Curso Alterno CA2		
	4. Muestra los controles para seleccionar el rango de fecha y el tipo de área de trabajo.	

5. Selecciona la fecha, área deseada, acepta la operación y retorna al paso 6.	
Curso Alterno CA3	
	4. Muestra los controles para seleccionar el rango de fecha y el tipo de cargo.
5. Selecciona la fecha, cargo deseado, acepta la operación. Luego paso 6.	
Curso Alterno CA4	
	4. Muestra los controles para seleccionar el rango de fecha y el tipo de sexo.
5. Selecciona la fecha, sexo deseado, acepta la operación y retorna al paso 6.	
Curso Alterno CA5	
	4. Muestra los controles para seleccionar el rango de fecha y el color de la piel.
5. Selecciona la fecha, el color de la piel deseado, acepta la operación y retorna al paso 6.	
Curso Alterno CA6	
	4. Muestra los controles para seleccionar el rango de fecha y el municipio donde radica.
5. Selecciona la fecha, municipio, acepta la operación y retorna al paso 6.	
Curso Alterno CA7	
	4. Muestra los controles para seleccionar el rango de fecha y la categoría docente.
5. Selecciona la fecha, categoría docente deseada, acepta la operación y retorna al paso 6.	
Curso Alterno CA8	
	4. Muestra los controles para seleccionar el rango de fecha y la antigüedad en la profesión.
5. Selecciona la fecha, los años de antigüedad en la profesión, acepta la operación y retorna al paso 6.	
Curso Alterno CA9	
	4. Muestra los controles para seleccionar el rango de fecha y si es docente.

5. Selecciona la fecha y si es docente, acepta la operación y retorna al paso 6.	
Curso Alterno CA10	
	4. Muestra los controles para seleccionar el rango de fecha y si es becado en la universidad.
5. Selecciona la fecha y si es becado, acepta la operación y retorna al paso 6.	
Poscondiciones	Se muestra la información solicitada por el usuario.
Requisitos Especiales	
Prioridad	Secundario
Interfaz	Ver Anexo 9

Tabla 6: Descripción Caso de Uso Buscar Modificaciones

Caso de Uso	Buscar Modificaciones
Actores	Especialista
Propósito	Mostrar las modificaciones solicitadas.
Resumen	El caso de uso se inicia cuando un usuario selecciona la opción de observar modificaciones y termina con la visualización de la información que desea observar.
Referencias	
Precondiciones	Estar autenticado
Curso Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. Selecciona opción de mostrar modificaciones.	2. Muestra el listado de las diferentes búsquedas que puede realizar.

<p>3. Selecciona la opción de búsqueda de modificaciones generales. Si selecciona la opción de búsqueda por dirección ver CA1, Si selecciona la opción de búsqueda por áreas ver CA2, Si selecciona la opción de búsqueda por cargo ver CA3, Si selecciona la opción de búsqueda por sexo ver CA4, Si selecciona la opción de búsqueda por color de la piel ver CA5, Si selecciona la opción de búsqueda por municipio ver CA6, Si selecciona la opción de búsqueda por categoría docente ver CA7, Si selecciona la opción de búsqueda por antigüedad ver CA8, Si selecciona la opción de búsqueda del personal docente ver CA9, Si selecciona la opción de búsqueda si es becado CA10,</p>	<p>4. Muestra los controles para seleccionar el rango de fecha.</p>
<p>5. Selecciona la fecha deseada y acepta la operación.</p>	<p>6. Muestra la información solicita por el usuario.</p>
<p>Curso Alterno CA1</p>	
	<p>4. Muestra los controles para seleccionar el rango de fecha y el tipo de dirección de trabajo.</p>
<p>5. Selecciona la fecha, dirección deseada y acepta la operación.</p>	<p>6. Muestra la información solicita por el usuario.</p>
<p>Curso Alterno CA2</p>	
	<p>4. Muestra los controles para seleccionar el rango de fecha y el tipo de área de trabajo.</p>
<p>5. Selecciona la fecha, área deseada, acepta la operación y retorna al paso 6.</p>	
<p>Curso Alterno CA3</p>	
	<p>4. Muestra los controles para seleccionar el rango de fecha y el tipo de cargo.</p>
<p>5. Selecciona la fecha, cargo deseado, acepta la operación. Luego paso 6.</p>	
<p>Curso Alterno CA4</p>	
	<p>4. Muestra los controles para seleccionar el rango de fecha y el tipo de sexo.</p>
<p>5. Selecciona la fecha, sexo deseado, acepta la operación y retorna al paso 6.</p>	
<p>Curso Alterno CA5</p>	

	4. Muestra los controles para seleccionar el rango de fecha y el color de la piel.
5. Selecciona la fecha, el color de la piel deseado, acepta la operación y retorna al paso 6.	
Curso Alterno CA6	
	4. Muestra los controles para seleccionar el rango de fecha y el municipio donde radica.
5. Selecciona la fecha, municipio, acepta la operación y retorna al paso 6.	
Curso Alterno CA7	
	4. Muestra los controles para seleccionar el rango de fecha y la categoría docente.
5. Selecciona la fecha, categoría docente deseada, acepta la operación y retorna al paso 6.	
Curso Alterno CA8	
	4. Muestra los controles para seleccionar el rango de fecha y la antigüedad en la profesión.
5. Selecciona la fecha, los años de antigüedad en la profesión, acepta la operación y retorna al paso 6.	
Curso Alterno CA9	
	4. Muestra los controles para seleccionar el rango de fecha y si es docente.
5. Selecciona la fecha y si es docente, acepta la operación y retorna al paso 6.	
Curso Alterno CA10	
	4. Muestra los controles para seleccionar el rango de fecha y si es becado en la universidad.
5. Selecciona la fecha y si es becado, acepta la operación y retorna al paso 6.	
Poscondiciones	Se muestra la información solicitada por el usuario.
Requisitos Especiales	
Prioridad	Secundario
Interfaz	Ver Anexo 10

Tabla 7: Descripción Caso de Uso Buscar Errores

Caso de Uso	Buscar Errores	
Actores	Especialista	
Propósito	Mostrar los errores solicitados.	
Resumen	El caso de uso se inicia cuando un usuario selecciona la opción de observar errores y termina con la visualización de la información que desea observar.	
Referencias		
Precondiciones	Estar autenticado	
Curso Normal de Eventos		
	Acción del Actor	Respuesta del Sistema
	1. Selecciona opción de mostrar errores.	2. Muestra el listado de las diferentes búsquedas que puede realizar.
	3. Selecciona la opción de búsqueda de errores generales.	4. Muestra los controles para seleccionar el rango de fecha.
	5. Selecciona la fecha deseada y acepta la operación.	6. Muestra la información solicita por el usuario.
Poscondiciones	Se muestra la información solicitada por el usuario.	
Requisitos Especiales		
Prioridad	Secundario	
Interfaz	Ver Anexo 11	

Tabla 8: Descripción Caso de Uso Buscar Reportes

Caso de Uso	Buscar Reportes	
Actores	Especialista	
Propósito	Mostrar los reportes solicitados.	
Resumen	El caso de uso se inicia cuando un usuario selecciona la opción de observar reportes y termina con la visualización de la información que desea observar.	
Referencias		
Precondiciones	Estar autenticado	
Curso Normal de Eventos		
	Acción del Actor	Respuesta del Sistema
	1. Selecciona opción de mostrar reportes.	2. Muestra el listado de las diferentes búsquedas que puede realizar.

3. Selecciona la opción de búsqueda de reportes generales.	4. Muestra los controles para seleccionar el rango de fecha.
5. Selecciona la fecha deseada y acepta la operación.	6. Muestra la información solicitada por el usuario.
Poscondiciones	Se muestra la información solicitada por el usuario.
Requisitos Especiales	
Prioridad	Secundario
Interfaz	Ver Anexo 12

3.6 Diagramas de Secuencia

Con la idea de dar una visión gráfica de las interacciones de los actores con el sistema, se utilizan los diagramas de secuencia (DSS), los cuales muestran qué hace el sistema ante el medio, sin explicar el cómo. (ÁLVAREZ and HERNÁNDEZ 2000)

Forma parte del modelado dinámico del sistema. Se modelan las llamadas entre clases desde un punto concreto. Es útil para observar la vida de los objetos, identificar llamadas a realizar o posibles errores del modelado estático, que imposibiliten el flujo de información o de invocaciones entre los componentes. Muestra cómo los objetos se comunican unos con otros en una secuencia de tiempo, qué sucede en cada momento, y para ello contienen objetos con sus ciclos de vida y los mensajes que se envían entre ellos ordenados secuencialmente. (ÁLVAREZ and HERNÁNDEZ 2000)

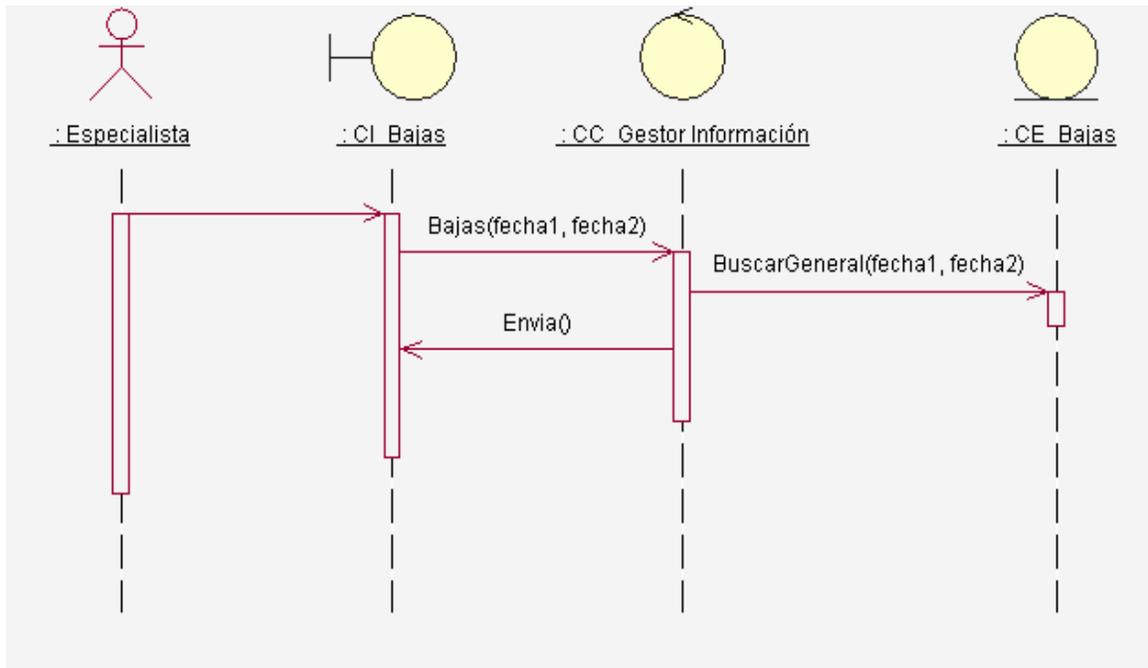


Figura 3: Caso de Uso Buscar Bajas, Escenario Buscar General

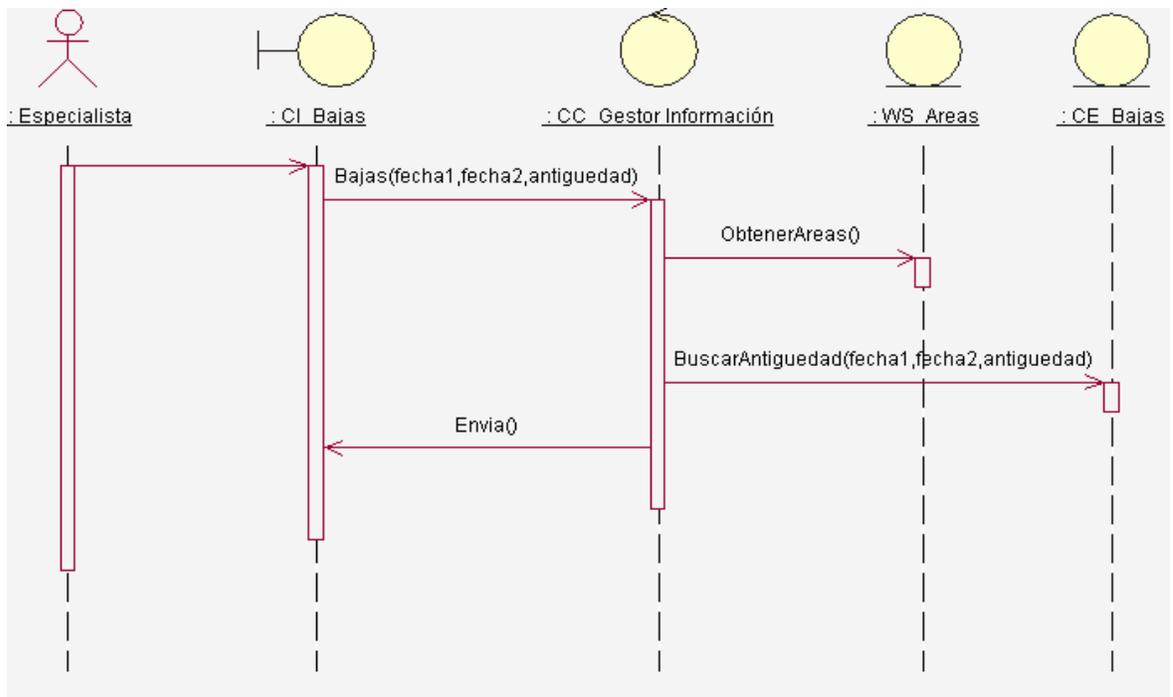


Figura 4: Caso de Uso Buscar Bajas, Escenario Buscar Áreas

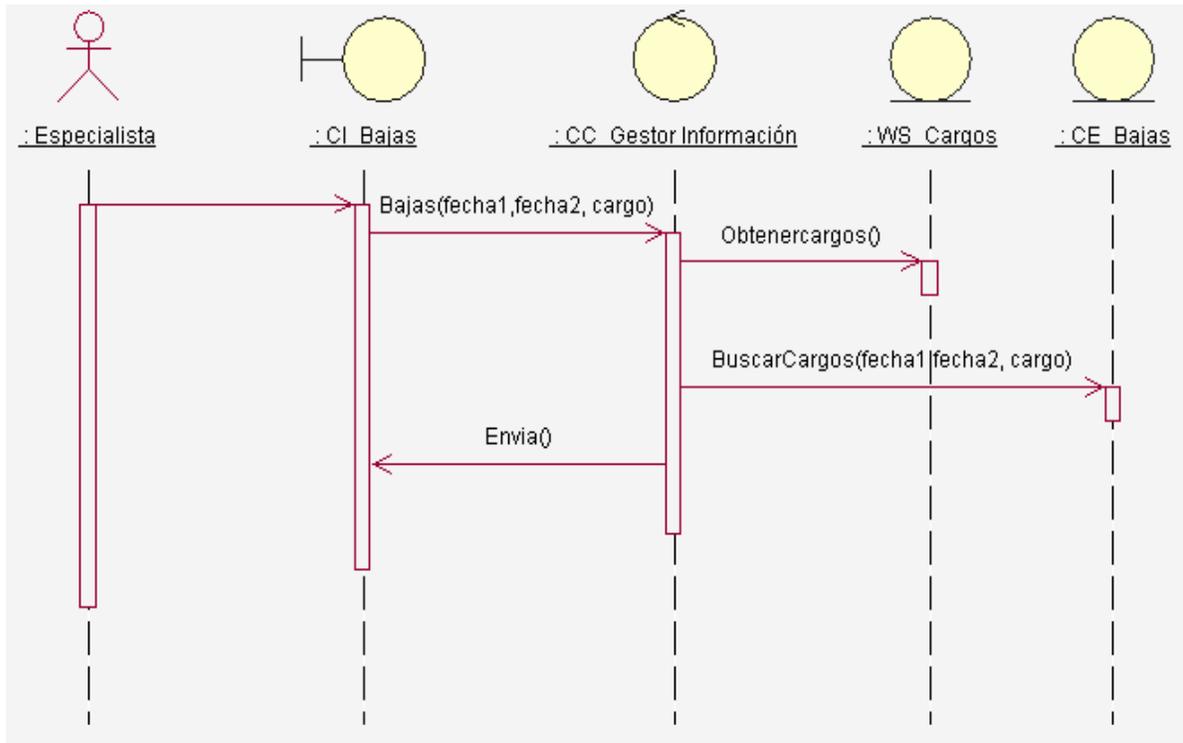


Figura 5: Caso de Uso Buscar Bajas, Escenario Buscar Cargo

Conclusiones

En este capítulo se dio paso a desarrollar la propuesta de solución planteada, obteniéndose a partir de un análisis las funcionalidades que debe poseer el sistema las cuales se representaron mediante un Diagrama de Casos de Uso describiéndose, paso a paso todas las acciones de los actores del sistema con los casos de uso con los que interactúan. Teniendo en cuenta todas estas características podemos pasar a construir el sistema poniendo en práctica el cumplimiento de los requisitos tanto funcionales como no funcionales planteados en este capítulo.

Capítulo 4

Construcción de la solución propuesta

En el desarrollo de un proyecto intervienen diferentes etapas que se interrelacionan y complementan, con la finalidad de alcanzar los objetivos iniciales. Estas etapas pueden variar dependiendo del ciclo de vida que se haya decidido para el sistema, pero las principales fases de un proyecto son el análisis de requerimientos, la especificación, el diseño, el desarrollo o implementación, y por último las pruebas.

En este capítulo se pone en marcha la construcción de la solución propuesta en los capítulos anteriores, se definen las clases, las cuales darán paso a la definición de las clases persistentes, para confeccionar la base de datos. Se describe la arquitectura del sistema y la configuración sobre la cual se realizará la implementación. Se define el modelo de despliegue originado por la selección de los artefactos más importantes para el sistema y donde se precisan los componentes que conforman la estructura física de la aplicación.

4.1 Diagrama de clases.

Este sistema se ha modelado sobre la base de la arquitectura de 3 capas (Interfaz, Aplicación, Acceso a Datos), dentro de las capas pueden existir los paquetes, esto se realizan teniendo en cuenta el nivel de complejidad que encierra cada sistema, ya que la utilización de los paquetes, garantiza un mayor entendimiento, independencia, legibilidad y organización a la hora de modelar y entender el sistema. Para este sistema la utilización de los paquetes será de forma sencilla teniendo en cuenta que la aplicación no es muy compleja.

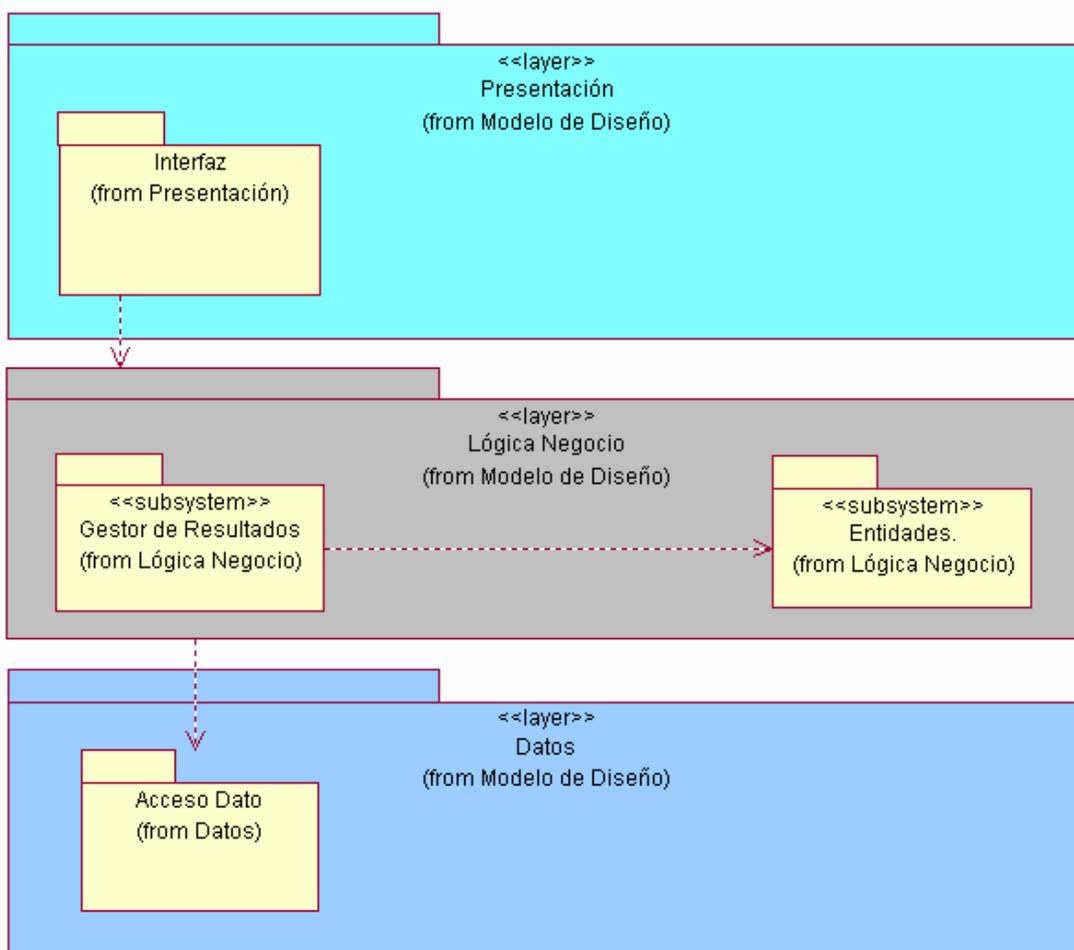


Figura 6: Arquitectura en Capas

Capa de presentación: Esta capa es la encargada de interactuar con el usuario, es lo que normalmente se conoce como interfaz de usuario. Contiene un paquete llamado interfaz donde se agrupan las páginas que interactúan directamente con los usuarios.

Capa aplicación: Aquí se localiza la Lógica del Negocio, esta capa recibe la petición del usuario a través de la capa de presentación y se encarga de darle curso a dichas peticiones, esta capa contiene todas las clases que se encargan de darle el principal funcionamiento al sistema, se realizan las validaciones, búsquedas, etc. Esta capa contiene los subsistemas gestión de resultados y entidades (Ver Anexo 5), que contienen clases controladoras.

Capa de datos: Esta es la capa mediante la cual se accede a los repositorios de información o sea a la Base de Datos. El paquete Acceso a Datos contiene las clases que posibilitan el acceso a la Base de Datos y manipular la persistencia de las Entidades.

A continuación se muestran los diagramas de clases correspondientes a cada paquete.

Capa: Interfaz de Usuario

Subpaquete: Interfaz

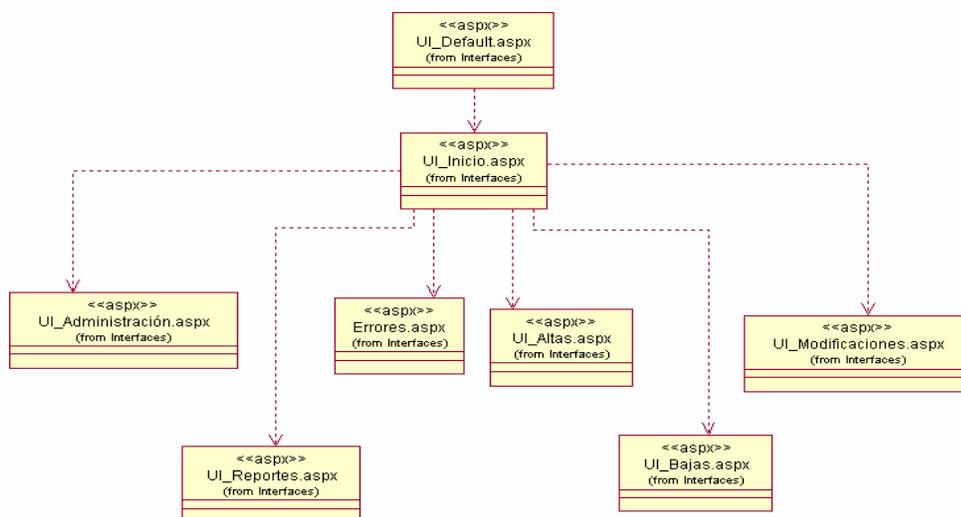


Figura 7: Subpaquete Interfaz Usuario

Capa: Lógica del Negocio

Subsistema: Gestor de Resultados

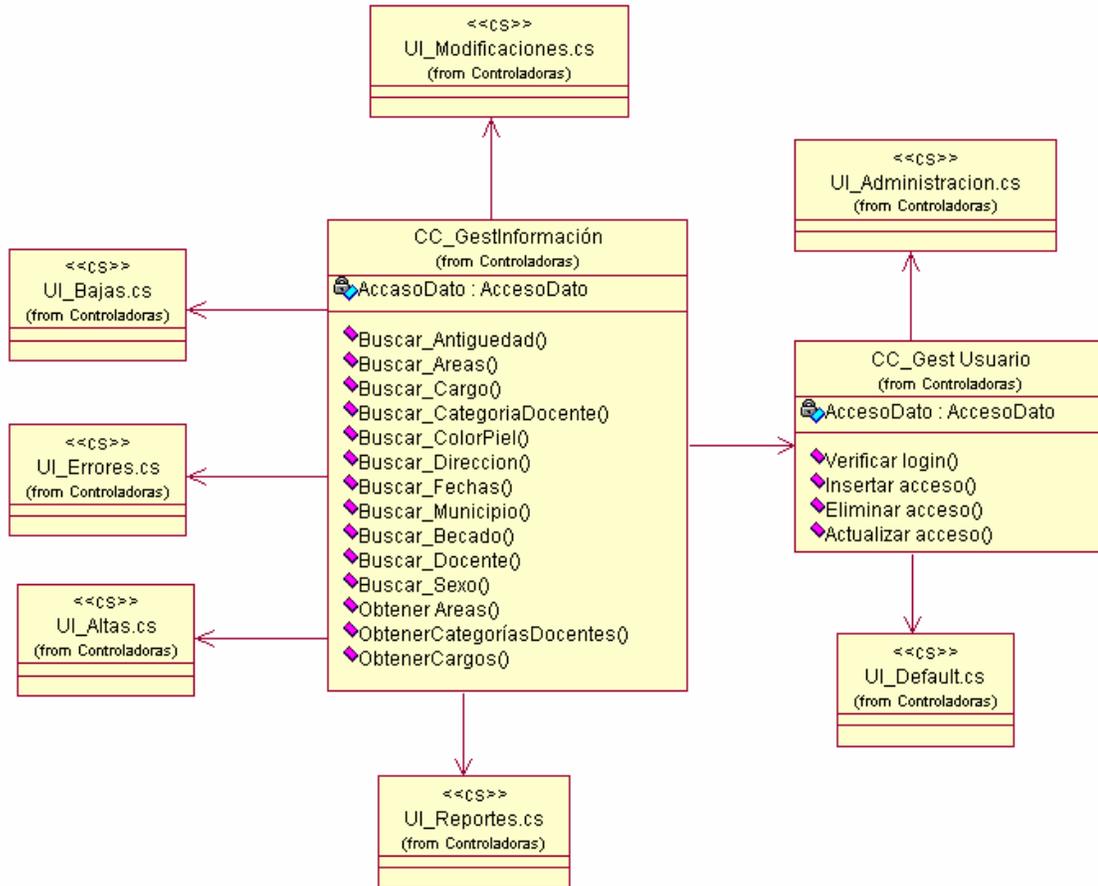


Figura 8: Subsistema Gestor de Resultados

Capa: Lógica del Negocio

Subsistema: Entidades

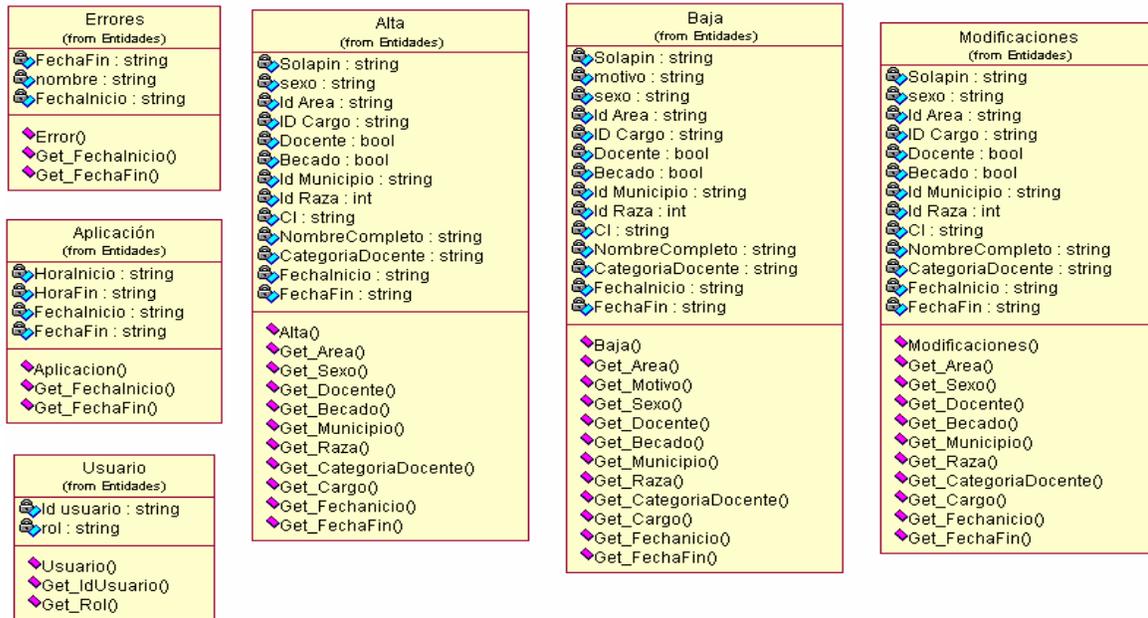


Figura 9: Subsistema Entidades

Capa: Datos

Subpaquete: Acceso a Datos

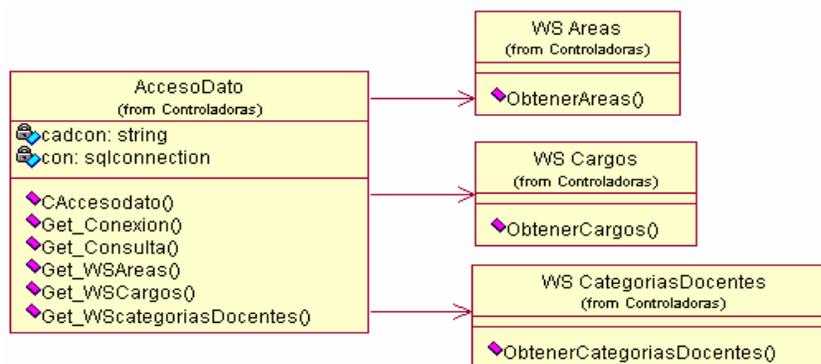


Figura 10: Subpaquete Acceso a Datos

4.2 Diagramas de Clases Diseño Web

Diagrama Clases Diseño Web CU Autenticar

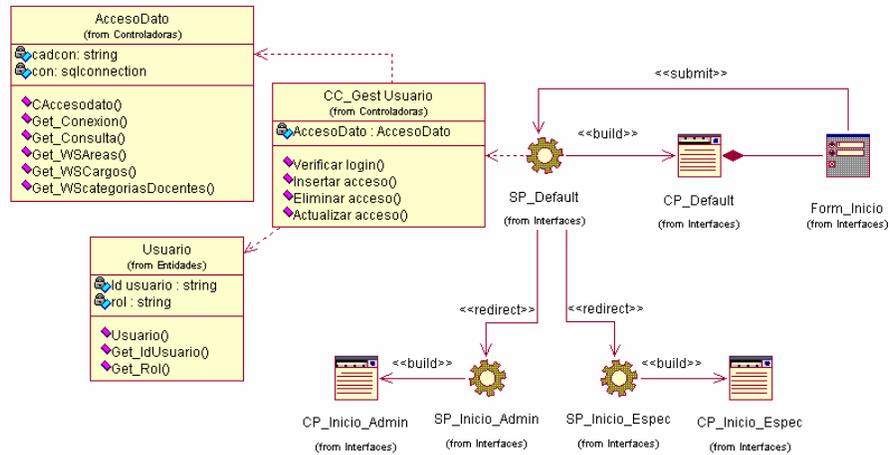


Figura 11: Diagrama Clases Diseño Web Caso de Uso Autenticar

Diagrama Clases Diseño Web CU Buscar Altas

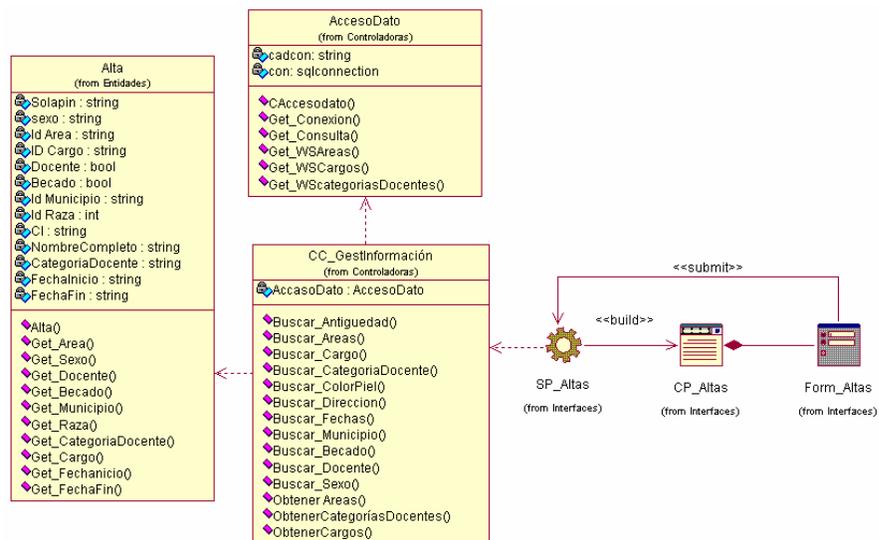


Figura 12: Diagrama Clases Diseño Web Caso de Uso Buscar Altas

Diagrama Clases Diseño Web CU Buscar Bajas

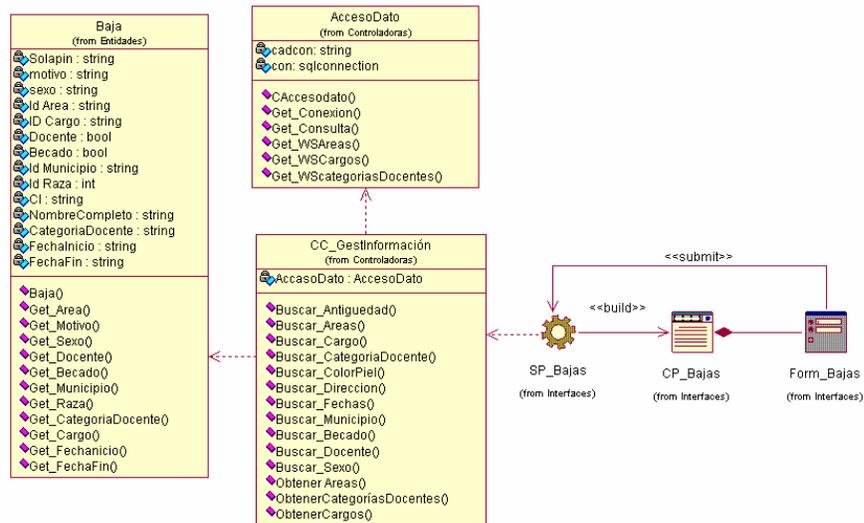


Figura 13: Diagrama Clases Diseño Web Caso de Uso Buscar Bajas

Diagrama Clases Diseño Web CU Buscar Modificaciones

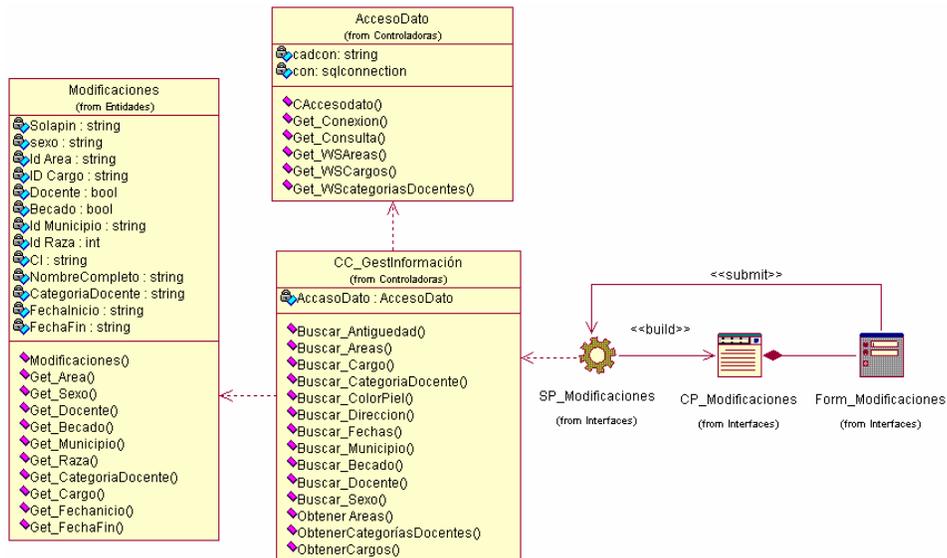


Figura 14: Diagrama Clases Diseño Web Caso de Uso Buscar Modificaciones

Diagrama Clases Diseño Web CU Buscar Errores

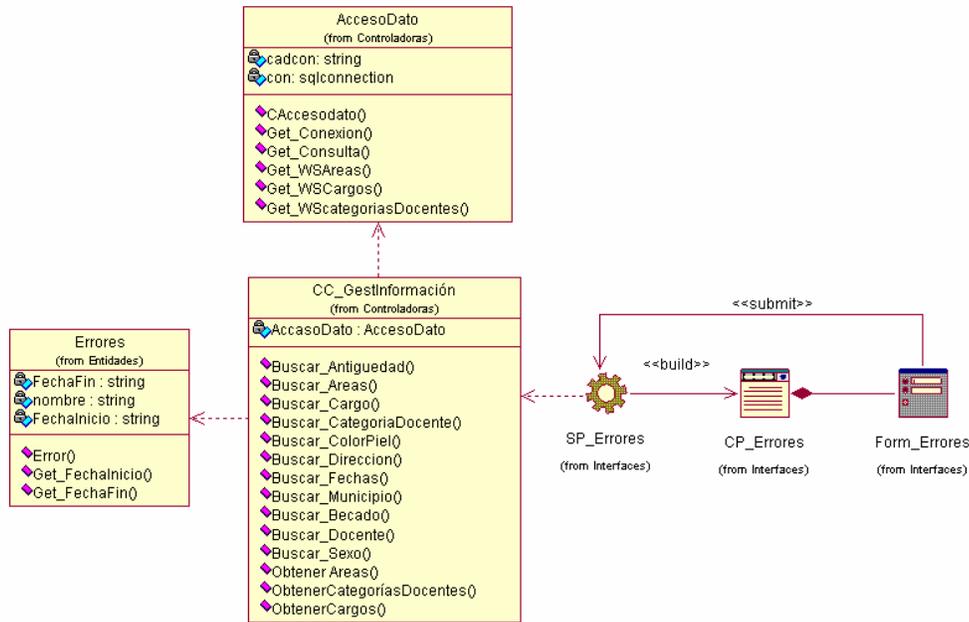


Figura 15: Clases Diseño Web Caso de Uso Buscar Errores

La aplicación Web publicará los métodos implementados en un servicio Web, este tendrá una arquitectura de tres capas al igual que la aplicación, por lo que tendrá un diseño similar solo que se diferencia en la capa de presentación que no constará con las interfaces de la aplicación sino con la interfaz (única) que contiene el servicio Web.

4.3 Principios de Diseño

4.3.1 Estándares de la interfaz de la aplicación

La calidad de la aplicación es un factor muy importante que hay que tener en cuenta pues influye en la satisfacción del usuario o sea que es preciso tener en cuenta la calidad de los contenidos y del diseño de

la aplicación. La importancia del diseño de la aplicación se basa en el método de interacción entre los usuarios y el sistema.

Para el diseño de la aplicación Web se tuvo en cuenta una serie de aspectos como: colores claros (azul y blanco), operaciones que el usuario puede realizar se situaron en la parte superior e inferior de la aplicación logrando así que estén accesibles y fáciles de localizar, se utilizó plantilla para lograr la uniformidad entre las páginas.

4.3.2 Tratamiento de excepciones

El tratamiento de excepciones es un proceso sumamente importante en el desarrollo de cualquier aplicación. El tratamiento de excepciones o errores es un detalle considerado en cada acción ejecutada por el sistema. Se evitan de esta manera operaciones innecesarias y aumenta el tiempo útil disponible por el usuario.

En la entrada de datos se valida que estos sean correctos, emitiendo un mensaje de alerta en caso contrario. Las excepciones que puedan ocurrir en tiempo de ejecución se tratan haciendo uso del mecanismo de tratamiento de excepciones que brinda .Net.

4.3.3 Estándares de Codificación

Los estándares de codificación permiten una mayor comunicación entre los desarrolladores a nivel mundial. Para obtener un producto final óptimo y comprensible fácilmente por cualquier desarrollador que le de continuidad se utilizó la programación orientada a objetos. Durante el desarrollo estos estándares ayudan a crear un código de alta calidad y a realizar considerablemente la capacidad de mantenimiento a largo plazo del producto final.

Entre otras ventajas de programar usando estándares de codificación están:

- Reducir la posibilidad de producir errores.

- Hacer el código más uniforme y fácil de entender.
- Hacer el software más fácil de entender.

En esta aplicación se ha utilizado el estándar de codificación “Camel Case” que principalmente tiene que ver con la capitalización de los caracteres. Así mismo, se ha seguido el estilo de codificación propuesto por Microsoft para programar con C#.

4.3.4 Declaraciones

En el momento de nombrar las clases, variables y funciones se ha tenido especial cuidado, en el caso de las variables se le antepone un prefijo al nombre indicando el tipo de variable que es, la siguiente lista muestra ejemplo de algunas declaraciones de tipos de variables.

- Campos de edición: **Txt**Nombre
- Campos de selección: **Lbx**Nombre o **Rb**Nombre
- Botones de acción: **Btn**Nombre
- DropDownList: **DI**Nombre
- Datasets: **Ds**Nombre
- Conexión: **Conn**Nombre
- GridView: **Gv**Nombre
- MultiView: **Mv**Nombre
- View: **V**Nombre
- Calendar: **Cal**Nombre

4.3.5 Diseño de la BD

Para diseñar la base de datos del sistema, se utilizó el diagrama de clases persistentes y el modelo de datos, que están basados en la modelación de las clases del epígrafe anterior. Algunas de las clases representaban los datos que se obtienen y almacenan durante los procesos de la aplicación, estos son lo

que pueden modelarse a través de un diagrama de clases persistentes, lo que permitirá ver la relación entre los datos, y completará la modelación de la lógica de negocio de la aplicación.

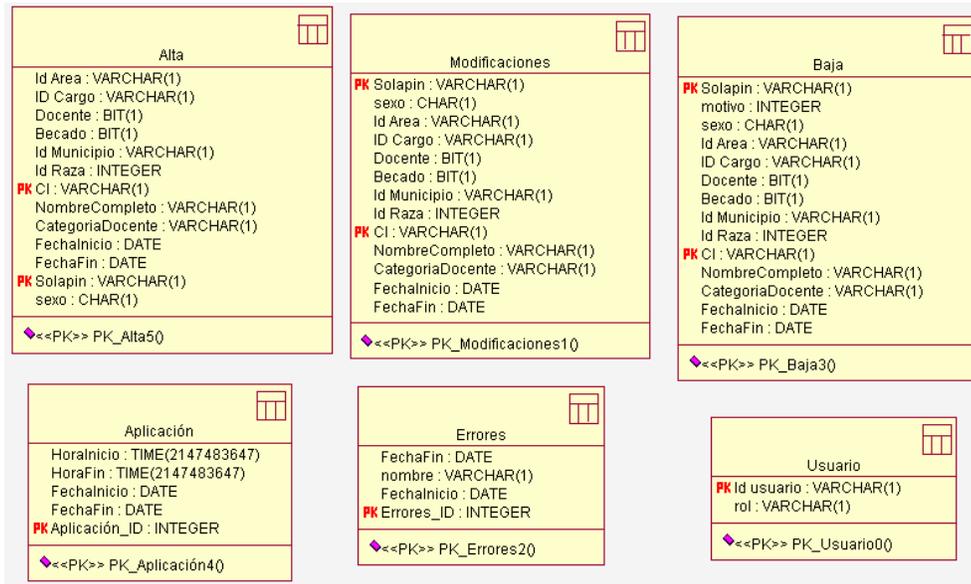


Figura 16: Modelo de Datos

4.4 Generalidades de la Implementación

La implementación es una fase generalmente muy extensa y requiere de un equipo de desarrolladores para llevarse a cabo. En la fase de desarrollo se implementa el sistema, es decir, se crea el código correspondiente al resultado de la fase de diseño, siguiendo los patrones y la arquitectura escogida.

4.4.1 Modelo de Despliegue

EL modelo de despliegue constituye la representación física del sistema en términos de la distribución de la funcionalidad de los nodos de cómputo. Muestra las relaciones entre el hardware y el software en el sistema final. Se representa como un grafo de nodos unidos por conexiones de comunicación.

Es la manera mediante la cual el sistema interactúa con el usuario, partiendo de una estación de trabajo cliente, la cual le hace la petición necesaria en cada caso al servidor. Es un modelo de objeto fundamental ya que constituye una entrada para el modelo de diseño e implementación.

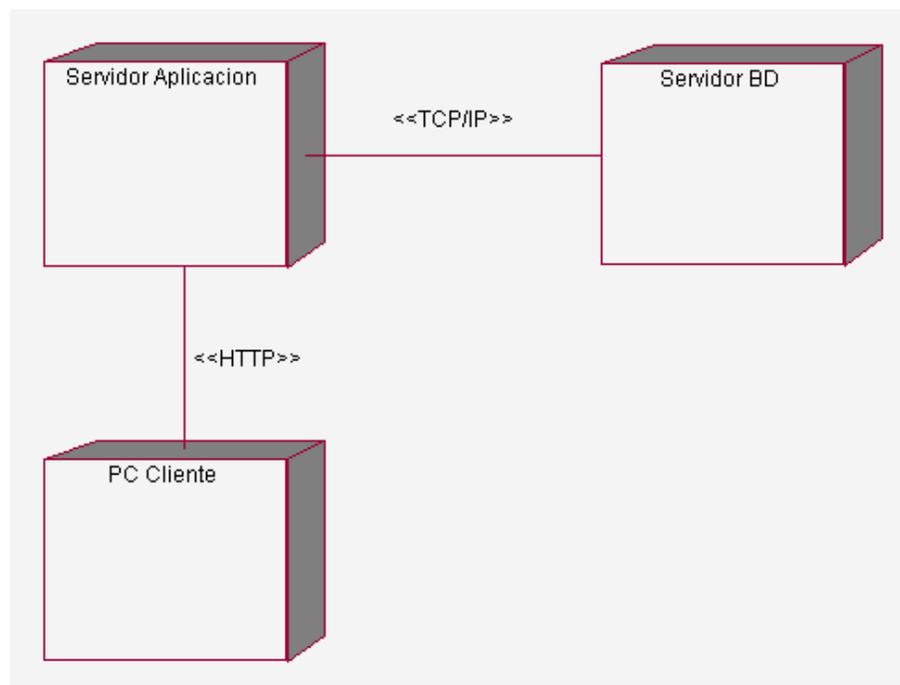


Figura 17: Modelo de Despliegue

4.4.2 Modelo de Componentes

Se muestran los componentes por casos de usos con sus relaciones con lo demás, los cuales para poder acceder a la base de datos de la aplicación tiene que pasar por el acceso a datos.cs.

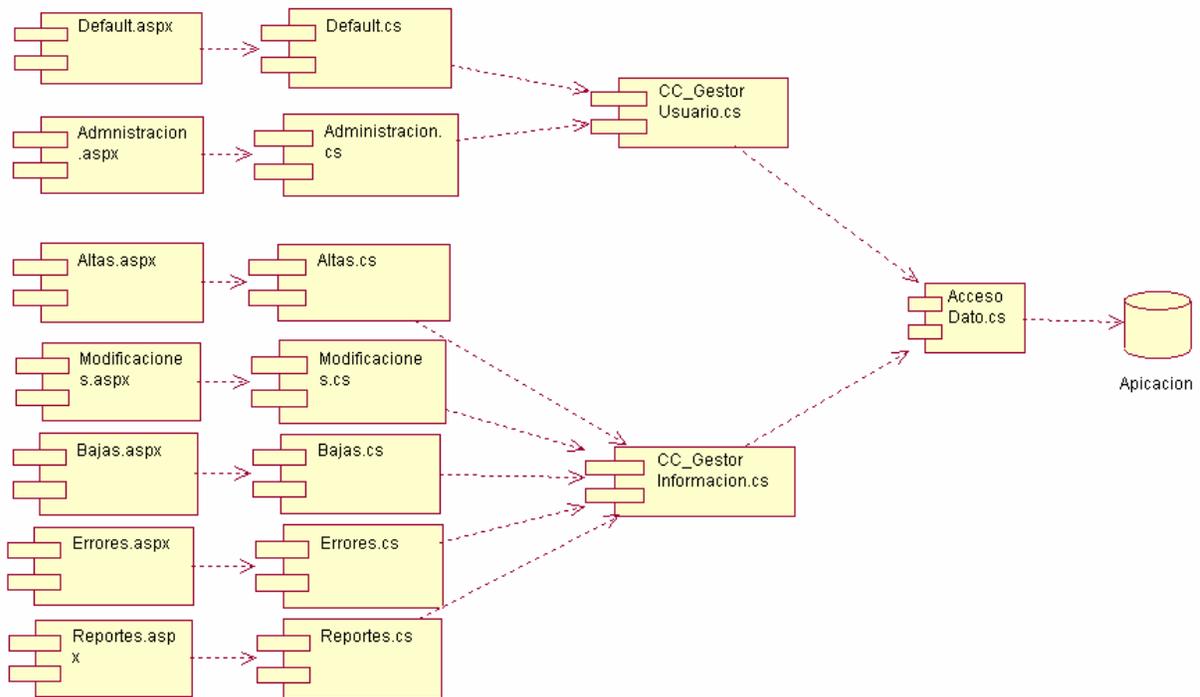


Figura 18: Modelo de Componentes

4.4.3 Prueba del sistema propuesto

En el proceso de prueba de este sistema es importante para contar con utilidades que ayuden a conocer el funcionamiento real de la aplicación.

La prueba de **caja negra** se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene.

Para la prueba de caja negra de este sistema se selecciona el caso de uso Autenticar Usuario. Las posibles variantes de este caso de uso son:

- Usuario y contraseña correcta.
- Usuario incorrecto y contraseña correcta.
- Usuario correcto y contraseña incorrecta.
- Usuario incorrecto y contraseña incorrecta.

Interfaz



Figura 19: Prueba Caja Negra

Tabla 9: Caso Prueba Usuario y contraseña correcta

Caso de Prueba	Usuario y contraseña correcta.
Caso de uso	Autenticar Usuario.
Caso de prueba	Comprobar que cuando el usuario va a entrar al sistema con su usuario y contraseña correcta puede acceder.
Entrada	Se accederá al sistema con un usuario Administrador y una contraseña administrador* .
Resultado	Como el usuario que existe en la base de datos es administrador y la contraseña administrador*, los datos entrados son válidos.

Tabla 10: Usuario incorrecto y contraseña correcta

Caso de Prueba	Usuario incorrecto y contraseña correcta.
Caso de uso	Autenticar Usuario.
Caso de prueba	Comprobar que cuando el usuario no va a entrar al sistema con un usuario incorrecto y contraseña correcta.

Entrada	Se accederá al sistema con un usuario Administrator12 y una contraseña administrador* .
Resultado	Como el usuario que existe en la base de datos es administrador y la contraseña no es la introducida, los datos entrados no son válidos y el sistema muestra que no es usuario de este sistema.

Tabla 11: Usuario incorrecto y contraseña incorrecta

Caso de Prueba	Usuario incorrecto y contraseña incorrecta.
Caso de uso	Autenticar Usuario.
Caso de prueba	Comprobar que cuando el usuario va a entrar al sistema con su usuario y contraseña incorrecta no puede acceder al sistema.
Entrada	Se accederá al sistema con un usuario Admin y una contraseña admin .
Resultado	Como el usuario que existe en la base de datos es administrador y la contraseña administrador* , los datos entrados no son válidos.

Tabla 12: Usuario correcto y contraseña incorrecta

Caso de Prueba	Usuario correcto y contraseña incorrecta.
Caso de uso	Autenticar Usuario.
Caso de prueba	Comprobar que cuando el usuario va a entrar al sistema con su usuario y contraseña incorrecta no puede acceder al sistema.
Entrada	Se accederá al sistema con un usuario Administrador y una contraseña admin .
Resultado	Como el usuario que existe en la base de datos es administrador y la contraseña administrador* , los datos entrados no son válidos.

Con la prueba de la **caja blanca** del software se comprueban los caminos lógicos del software, proponiendo casos de prueba que se ejerciten, conjuntos específicos de condiciones y/o bucles. Se puede examinar el estado del programa en varios puntos para determinar si el estado real coincide con el esperado o mencionado, consiste en correr uno de los métodos del código para calcular la complejidad ciclomática, por tanto es quien define el número de caminos independientes (siendo esta última una métrica de software que proporciona una medición cuantitativa de la complejidad lógica del programa) y ofrece también el límite de pruebas necesarias para asegurar que se ejecute cada sentencia como mínimo una vez.

```

public List<area> Get_Areas()
{
    Trabajadores Ws = new cu.uci.trabajadores.Trabajadores();           1
    area[] a = Ws.ObtenerAreas();                                       1
    List<area> a1 = new List<area>();                                     1
    for (int i = 0; i < a1.Length; i++)                                  2
    {
        DIAreas.Items.Add(a1.Seleccionar_Usuario()[i]);                 3
    }                                                                    4

    return a1;                                                         5
}                                                                       6

```

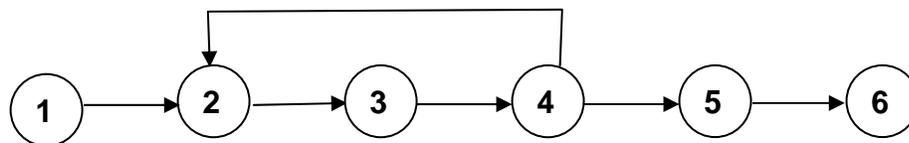


Figura 20: Prueba de caja blanca. Cálculo de la complejidad ciclomática.

Complejidad ciclomática $V(G) = \text{Cantidad de Aristas } [A] - \text{Cantidad de nodos } [N] + 2$.

$$V(G) = A - N + 2$$

$$V(G) = 6 - 6 + 2$$

$$V(G) = 2$$

Conclusiones

En el capítulo se ha planteado el modelo de construcción propuesto a través de las descripciones de las clases, diagramas de clases, diagrama de clases persistentes y demás elementos necesarios para la implementación de la aplicación. Se explicó cómo está estructurada la aplicación físicamente, mediante los modelos de despliegue y de componentes.

Capítulo 5

Estudio de la factibilidad del proyecto

En este capítulo se formula un estudio de la factibilidad de la Aplicación Web desarrollada teniendo en cuenta la complejidad que presenta la evaluación de un proyecto, pues no solo se basa en el estudio del problema a resolver, los medios informáticos para su implementación sino que impone la utilización de una metodología que establezca una disciplina de trabajo que permita el planeamiento y control del proyecto, facilite la asignación de tareas, mejorar las estimaciones de los costos, los gastos económicos en los cuales se incurrirán, el tiempo de desarrollo de dicho proyecto y permita garantizar la reducción de los riesgos.

Para los estudios de factibilidad se aplicará la técnica Cálculo de Puntos de Casos de Usos sin ajustar para determinar el costo y tiempo de desarrollo, si es factible o no el desarrollo del sistema y los beneficios tangible e intangibles que este proporciona.

5.1 Planificación

Para la realización de los proyectos, se necesita diseñar planes estratégicos, definir los objetivos y las metas que se deseen alcanzar, es por ello que la planificación constituye un paso imprescindible para alcanzar la máxima calidad en el desarrollo de un proyecto. La planificación observa la cadena de consecuencias, causas y efectos que origina la puesta en marcha del desarrollo de una aplicación, consiste en la identificación sistemática de las oportunidades, las cuales combinados con otros datos importantes como: estimar los resultados del proyecto, los valores de costo, tiempo y recursos requeridos estos proporcionan la base de la determinación que una empresa tome mejores decisiones en el presente para explotar las oportunidades y evitar los riesgos. Es preciso planificar de ante mano el trabajo, los recursos humanos y la tecnología a emplear en el desarrollo de un proyecto.

5.1.1 Objetivos de la planificación

El objetivo primordial de la planificación es, realizar el procesamiento de la información que conduzca a estimaciones razonables, para ello se necesita de:

- Experiencia y sentido común alcanzado en proyectos anteriores.
- Diferentes técnicas de estimación para estimar los costos, esfuerzos y tiempo necesarios para obtener el software.

Existen 3 variantes para la estimación:

1. Análisis de Puntos de Casos de Uso
2. Análisis de Puntos de Función y COCOMO II
3. Las estimaciones por líneas de código (LDC)

Se alega que no es factible realizar la estimación para la aplicación Web, del sistema migración de datos del sistema ASSETS al sistema trabajadores, de la UCI con la variante: Análisis de puntos de función y COCOMO II a pesar de constituir una técnica extremadamente eficiente, debido a los aspectos

relacionados con puntos de función, entradas externas, salidas externas, peticiones, ficheros internos etc. Por lo que se considera que no es pertinente la utilización de esta variante. Sin embargo se utilizará para este trabajo la variante: Análisis de Puntos de Casos de Uso en el que se garantiza la posibilidad de predecir el tamaño de un sistema a partir de las características y sus requisitos, expresados en los casos de uso. Además se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a una serie de factores que lo afectan, para finalmente, definir el tiempo total estimado para el proyecto a partir de esos factores antes mencionados.

5.2 Estimación Costos

Cálculo de Puntos de Casos de Usos sin ajustar

Se calcula a partir de la siguiente ecuación:

$$UUCP = UAW + UUCW$$

Donde:

UUCP: Puntos de Casos de Uso sin ajustar.

UAW: Factor de Peso de los Actores sin ajustar.

UUCW: Factor de Peso de los Casos de Uso sin ajustar.

Factor de Peso de los Actores sin ajustar (UAW)

Este valor se calcula mediante un análisis de la cantidad de Actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Actores se establece teniendo en cuenta en primer lugar si se trata de una persona o de otro sistema, y en segundo lugar, la forma en la que el actor interactúa con el sistema.

Tabla 13: Complejidad de los actores

Tipo de actor	Descripción	Factor de peso	Actores	Total
Simple	Sistema con sistema a través de interfaz de programación.	1	0	0
Medio	Sistema con sistema mediante protocolo de interfaz basada en texto.	2	0	0
Complejo	Persona que interactúa con el sistema mediante interfaz gráfica.	3	2	6

$$UAW = \Sigma (\text{Factor} * \text{Actores})$$

$$UAW = 6$$

Factor de Peso de los Casos de Uso sin ajustar (UUCW)

Este valor se calcula mediante un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Casos de Uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una transacción se entiende como una secuencia de actividades atómica, es decir, se efectúa la secuencia de actividades completa, o no se efectúa ninguna de las actividades de la secuencia.

Tabla 14: Complejidad de los Casos de Usos

Tipo de CU	Descripción	Peso	Cantidad de CU	Total
Simple	El caso de uso tiene de 1 a 3 transacciones.	5	2	10
Medio	El caso de uso tiene de 4 a 7 transacciones.	10	0	0
Complejo	El caso de uso tiene más de 8 transacciones.	15	5	75

$$UUCW = \Sigma (\text{Factor} * \text{CantCU})$$

$$UUCW = 85$$

Luego: $UUCP = UAW + UUCW$

$$UUCP = 6 + 85$$

UUCP = 91

Cálculo de Puntos de Casos de Uso ajustados

$$UCP = UUCP \times TCF \times EF$$

Donde:

- UCP:** Puntos de Casos de Uso ajustados
- UUCP:** Puntos de Casos de Uso sin ajustar
- TCF:** Factor de complejidad técnica
- EF:** Factor de ambiente

Factor de complejidad técnica (TCF)

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante.

Tabla 15: Factor de Complejidad Técnica

Factor	Descripción	Peso	Valor asignado	Total
T1	Sistema distribuido	2	0	0
T2	Tiempo de respuesta	1	4	4
T3	Eficiencia del usuario final	1	4	4
T4	Funcionamiento Interno complejo	1	3	3
T5	El código debe ser reutilizable	1	3	3
T6	Facilidad de instalación	0.5	0	0
T7	Facilidad de uso	0.5	5	2.5
T8	Portabilidad	2	4	8
T9	Facilidad de cambio	1	4	4
T10	Concurrencia	1	4	4
T11	Incluye objetivos especiales de seguridad	1	4	4
T12	Provee acceso directo a terceras partes	1	1	1

T13	Se requieren facilidades especiales de entrenamiento de usuarios	1	1	1
-----	--	---	---	---

El Factor de complejidad técnica se calcula mediante la siguiente ecuación:

$$TCF = 0.6 + 0.01 \times \sum (\text{Peso}_i \times \text{Valor asignado}_i)$$

$$\sum (\text{Peso}_i \times \text{Valor}_i) = 38.5$$

$$TCF = 0.6 + 0.01 * \sum (\text{Peso}_i * \text{Valor}_i)$$

$$TCF = 0,985$$

Factor de ambiente (EF)

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del Factor de ambiente. El cálculo del mismo es similar al cálculo del Factor de complejidad técnica, es decir, se trata de un conjunto de factores que se cuantifican con valores de 0 a 5.

Tabla 16: Factor Ambiente

Factor	Descripción	Peso	Valor asignado	Total(Peso*Valor)
E1	Familiaridad con el modelo de proyecto utilizado	1.5	4	6
E2	Experiencia en la aplicación	0.5	3	1.5
E3	Experiencia en la orientación a objetos.	1	3	3
E4	Capacidad del analista líder.	0.5	3	1.5
E5	Motivación.	1	4	4
E6	Estabilidad de requerimientos	2	4	8
E7	Personal Part-Time	-1	1	-1
E8	Dificultad del lenguaje de programación	-1	1	-1

Para los factores E1 al E4, un valor asignado de 0 significa sin experiencia, 3 experiencia media y 5 amplia experiencia (experto).

- Para el factor E5, 0 significa sin motivación para el proyecto, 3 motivación media y 5 alta motivación.
- Para el factor E6, 0 significa requerimientos extremadamente inestables, 3 estabilidad media y 5 requerimientos estables sin posibilidad de cambios.
- Para el factor E7, 0 significa que no hay personal part-time (es decir todos son full-time), 3 significa mitad y mitad, y 5 significa que todo el personal es part-time (nadie es full-time).
- Para el factor E8, 0 significa que el lenguaje de programación es fácil de usar, 3 medio y 5 que el lenguaje es extremadamente difícil.

El Factor de ambiente se calcula mediante la siguiente ecuación:

$$EF = 1.4 - 0.03 \times \sum (\text{Peso}_i \times \text{Valor}_i)$$

$$\sum (\text{Peso}_i \times \text{Valor}_i) = 22$$

$$EF = 1.4 - 0.03 \times \sum (\text{Peso}_i \times \text{Valor}_i)$$

$$EF = 0.74$$

Luego: $UCP = UUCP \times TCF \times EF$

$$UCP = 66.3299$$

De los Puntos de Casos de Uso a la estimación del esfuerzo

Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por debajo del valor medio (3), para los factores E1 a E6.

Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por encima del valor medio (3), para los factores E7 y E8.

- Si el total es 2 o menos, se utiliza el **factor de conversión 20 horas-hombre/Punto de Casos de Uso**, es decir, un Punto de Caso de Uso toma 20 horas-hombre.

- Si el total es 3 o 4, se utiliza el factor de conversión **28 horas-hombre/Punto de Casos de Uso**, es decir, un Punto de Caso de Uso toma 28 horas-hombre.
- Si el total es mayor o igual que 5, se recomienda efectuar cambios en el proyecto, ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

CF = 20

El esfuerzo en horas-hombre viene dado por:

$$E = UCP \times CF$$

Donde:

- E:** esfuerzo estimado en horas-hombre
- UCP:** Puntos de Casos de Uso ajustados
- CF:** factor de conversión

$$E = UCP * CF$$

$$E = 1326.598 \text{ horas-hombres}$$

Tabla 17: Estimación del Esfuerzo

Actividad	Porcentaje %	Horas-Hombres
Análisis	10	331.6495
Diseño	20	633.299
Implementación	40	1326.598
Pruebas	15	497.47425
Sobrecarga (otras actividades)	15	497.47425
Total	100	3316.495

Como el valor de esfuerzo calculado representa el esfuerzo del Flujo de Trabajo Implementación, por comparación se obtienen el resto de los esfuerzos y la suma de ellos es el esfuerzo total (ET).

Para el desarrollo de este proyecto se trabaja a un ritmo de 8 horas por día, durante 22 días al mes por lo que un mes de trabajo implica 176 horas.

Si $ET = 3316.495$ horas-hombre para un mes de trabajo que son 176 horas, el proyecto necesitaría un $ET = 18.84$ mes-hombre

En el caso de la herramienta Web que se propone, el equipo de desarrollo de software es de 2 personas, entonces el problema analizado puede terminarse en aproximadamente 9 meses (9.42 meses).

5.3 Beneficios tangibles e intangibles

5.3.1 Beneficios tangibles

Teniendo en cuenta que la aplicación Web y el Servicio Web no son construidos con el fin de comercializarlos sino para resolver un problema existente en el Departamento de Informatización de la Universidad de las Ciencias Informáticas (UCI), no se puede afirmar que existen beneficios económicos, sociales, políticos ni resultados palpables y cuantificables.

Los beneficios tangibles de los sistemas no están orientados a cifras en cuanto a materiales, sino más en concreto a las variables que definen el control eficiente de la calidad en el Departamento de Informatización al tener un control eficiente de la información del personal que labora en la Universidad y un aprovechamiento óptimo de los del Servicio Web por otras aplicaciones.

5.3.2 Beneficios intangibles

Como beneficios intangibles asociados a la puesta en marcha de la aplicación Web para el sistema de migración de datos del sistema ASSETS al sistema trabajadores se mencionan los siguientes:

- Aumento de la factibilidad y confiabilidad de los datos del personal de la Universidad.
- Aumento de la rapidez en la cual se muestra la información solicitada.
- Posibilidad de registros de información según una fecha determinada.
- Aumento de un eficiente desarrollo del proceso de gestión y control de los datos el sistema de migración.
- Posibilidad de evaluar y llevar un seguimiento de los resultados obtenidos diariamente en la ejecución de la aplicación.
- Posibilidad de obtener reportes diarios de los sucesos ocurridos.
- Posibilidad de mantener actualizado la información del personal de la Universidad.
- Aumento de la confiabilidad para el uso de los servicios brindados por la Universidad (obtención del solapín, acceso al comedor, acceso a lo ómnibus de transporte, acceso al centro).
- Disminución de la entrada del personal ajeno al centro.
- Disminución del mantenimiento de la aplicación, al presentar errores desconocidos.
- Posibilidad de brindar servicios de alto valor agregado a la comunidad universitaria a partir de la información adquirida, gestionada o generada por la Aplicación Web.

Como beneficios intangibles asociados a la puesta en marcha del Servicio Web para la integración del sistema a la arquitectura SOA de la UCI mencionan los siguientes:

- Posibilidad de brindar servicios de alto valor agregado a la comunidad universitaria a partir de la información de los métodos de la aplicación.
- Posibilidad de encapsular funciones específicas y hacer que estas funciones puedan ser utilizadas para el desarrollo y uso de otras aplicaciones y servidores.
- Posibilidad de acceder a esta información desde cualquier sitio Web en la Universidad.
- Posibilidad de conectarse al Servicio Web las personas que lo deseen.
- Aumento de la interoperabilidad del sistema.
- Garantiza la no dependencia al lenguaje de programación, plataforma de implantación, formato de presentación o sistema operativo.
- Garantiza su fácil publicación, localización e invocación mediante los protocolos Web estándar.

- Garantiza un escaso acoplamiento, los usuarios no necesitan conocer nada acerca del funcionamiento del mismo, solo como acceder a el.
- Posibilidad de emplear múltiples modos de invocación.
- Posibilidad de emplear múltiples estilos de comunicación.
- No dependencia del modo de transporte de la información sobre los protocolos de transporte.
- Posibilidad de ensamblar este servicio con aplicaciones Web usando componentes de otros servidores.

5.4 Análisis de costos y beneficios

El sistema como producto terminado no ha representado ningún costo para el Departamento de Informatización de la UCI debido a que el centro posee la infraestructura informática necesaria para la ejecución de la aplicación por lo que no se incurrió en gastos de equipamiento.

Con su implantación se puede realizar un riguroso control de los datos del personal del centro en un intervalo de tiempo corto y se puede garantizar un control eficiente de los servicios ofrecidos por la institución así como la interoperabilidad entre varias aplicaciones.

Conclusiones

Con el correspondiente desarrollo de este capítulo se aprecia que el estudio de la factibilidad del proyecto es un paso imprescindible en el desarrollo de cualquier software, pues de una forma u otra se exponen los aspectos que influyen en la ejecución o no del proyecto. Se ha profundizado en los costos, los recursos y materiales, tiempo de ejecución necesarios para la culminación del proyecto y también los beneficios que aporta el mismo.

CONCLUSIONES

Se realizó una exhaustiva búsqueda bibliográfica acerca de las últimas tendencias y tecnologías que a nivel internacional se están utilizando en el mundo de la informática para seleccionar las respuestas que se darían al problema.

Se seleccionó RUP como metodología de desarrollo de software a seguir para la construcción de la solución propuesta. Siguiendo este proceso, se diseñó e implementó la aplicación Web con la cual se propone aumentar la eficiencia y confiabilidad del control de la información de los trabajadores de la Universidad.

Por tanto se arriba a las siguientes conclusiones:

- La aplicación Web favorece la gestión de la información del personal que labora en la Universidad, garantizando los niveles de seguridad de la información., mayor agilidad en los resultados, por lo que reduce los errores, además posee una interfaz sencilla fácil de utilizar.
- El objetivo general planteado durante la investigación fue cumplido, comprobándose la idea a defender como respuesta del problema a resolver.
- El uso de los métodos de investigación científica proporcionó un estudio detallado del objeto de estudio.
- Las tareas investigativas posibilitaron un mejor desarrollo del trabajo.
- La aplicación Web fue avalada por el Departamento de Informatización de la UCI.

RECOMENDACIONES

- Agregar ayuda del sistema.
- Integrar el servicio Web a la arquitectura SOA de la UCI.
- Agregar otras funcionalidades como emitir una alarma en caso de ocurrir algún error en el sistema de migración de datos así como poder buscar a las personas que han causado baja por el motivo de la baja mediante el nombre y otras características. También poder buscar las altas y las modificaciones dado el nombre de la persona y otras características.
- Darle mantenimiento a la aplicación.
- Mejorar los reportes.
- Para próximas versiones migrar la aplicación a tecnologías libres.

REFERENCIAS BIBLIOGRÁFICAS

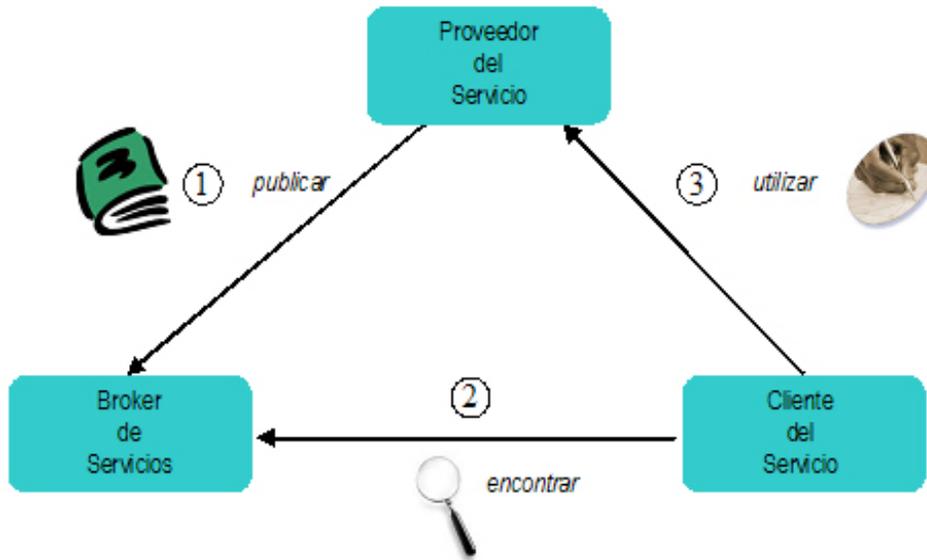
- ALFIN <http://www.mariapinto.es/alfineees/glosario.htm>, 2006.
- ÁLVAREZ, S. and A. HERNÁNDEZ. *Metodología para el desarrollo de aplicaciones con tecnología Orientada a Objetos utilizando notación UML*. La Habana, 2000. p.
- ARSYS <http://www.arsys.es/soporte/programacion> 2006.
- CORREA, S. C. *Asegure una migración exitosa de bases de datos*, 2006. 1.
- EVANS, G. *A simplified approach to RUP*, 2003.
- FERGUSON, J.; B. PATTERSON, et al. *La Biblia C#*. España, Anaya Multimedia, 2003. 805 p. 84-415-1484-4
- FERNÁNDEZ, I. and M. SUÁREZ. *Estudio sobre la unificación de conceptos en Ciencias de la información, Bibliotecología y Archivología: Una propuesta fundamentada*. La Habana, 1996. 166. p.
- GARCIA, R. M. M. <http://docencia.uci.cu/sbd/cgi-bin/admin/coments/files/FOLLETONUEVO.rtf>. 1999. p.
- *GLOSARIO DE TÉRMINOS SOBRE ADMINISTRACIÓN PÚBLICA*. 2006.
- GUTIÉRREZ, G. A. *J2EE Una plataforma para el cómputo empresarial.*, 2003.
- HERNÁNDEZ, M. L. *Programación en Java Script Segunda Edición* 2005. p.
- LARMAN, C. *UML y Patrones*. Habana, Editorial Félix Varela, 1999. p.
- MOLINA, J. G. and M. J. O. IBÁÑEZ. *Un Proceso basado en UML para aplicaciones Web*, 2001.
- PAZ, A. C. *Fuentes de información: Aspectos teóricos*. La Habana, Universidad de la Habana, 1994. 104. p.
- SANCHEZ, M. A. M. *Metodologías de Desarrollo de Software* 2006.
- SCHWABER, K. and M. BEEDLE. *Agile Software Development with Scrum*, 2005.
- SECO, J. A. *El lenguaje de programación C#*, 2001.
- TOLEDANO, M. D. D. *Servicio Web. Introducción y Escenarios para su Uso*, 2003.
- WIKIPEDIA *Aplicación Web*, 2007.
- WOODMAN, L. *Information management from strategies to action*. London, ASLIB, 1985. 95-114 p.

BIBLIOGRAFÍA

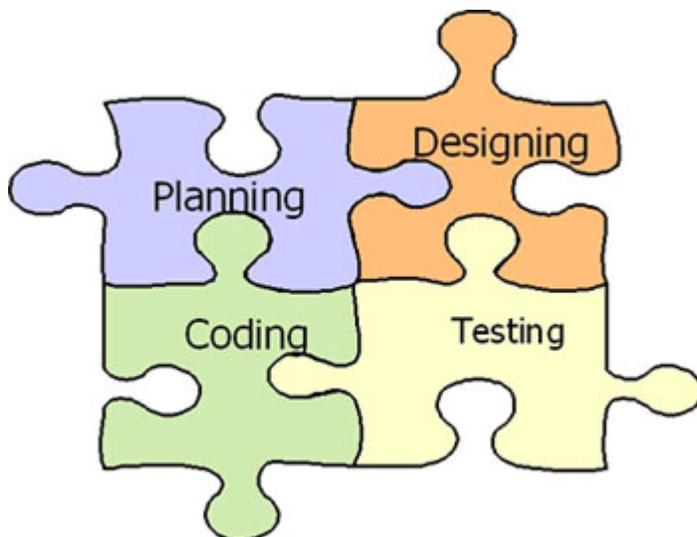
- ARCHER, T. *A fondo C#* McGRAW-HILL, 2001. 381 p. 0-7356-1288-9
- ARORA, G.; B. AIASWAMY, *et al.* *Programación C#*. Madrid. España, 2002. 770 p. 84-4 15-1420-8
- BOGGS, W. and M. BOGGS. *UML with Rational Rose 2002*. SYBEX, 2002. 720 p.
- CERAMI, E. *Web Service*. O'Reilly, 2002. 304 p. 0-596-00224-6
- FERGUSON, J.; B. PATTERSON, *et al.* *La Biblia C#*. España, Anaya Multimedia, 2003. 805 p. 84-415-1484-4
- GILFILLAN, I. *La Biblia MySQL*. España, Anaya Multimedia, 2003. 850 p.
- HANSEN, G. W. and J. V. HANSEN. *Diseño y Administración de Bases de Datos*. Segunda Edición. España, Prentice Hall, 2002. 576 p.
- JACOBSON, I.; G. BOOCH, *et al.* *El Proceso Unificado de Desarrollo de Software*. Madrid, Pearson Educación, 2000. 464 p. 0-201-57169-2
- LARMAN, C. *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. México, Prentice Hall, 1999. 536 p. 970-17-0261-1
- MACDONALD, M. *ASP. Net. Manual de Referencia*. España, McGraw-Hill/Interamericana, 2002. p. 0-07-2195 13-4
- OJEDA, F. C. *Visual C# .Net*. Madrid. España, Anaya Multimedia, 2002. p. 84-415- 1392-9
- PARIHAR, M. *ASP. Net*. Anaya Multimedia, 2002. 970 p.
- PRESSMAN, R. S. *Ingeniería de Software. Un enfoque práctico*. Madrid, 1997. 620 p.
- RUMBAUGH, J.; I. JACOBSON, *et al.* *El Lenguaje Unificado del Modelado. Manual de Referencia*. California, Adison Wesley, 1999. 528 p.
- SCHMULLER, J. *Aprendiendo UML en 24 horas*. México, Paerson Educación, 2000. 448 p. 968-444-463-X

ANEXOS

Anexo 1: Modelo de acceso a Servicios Web



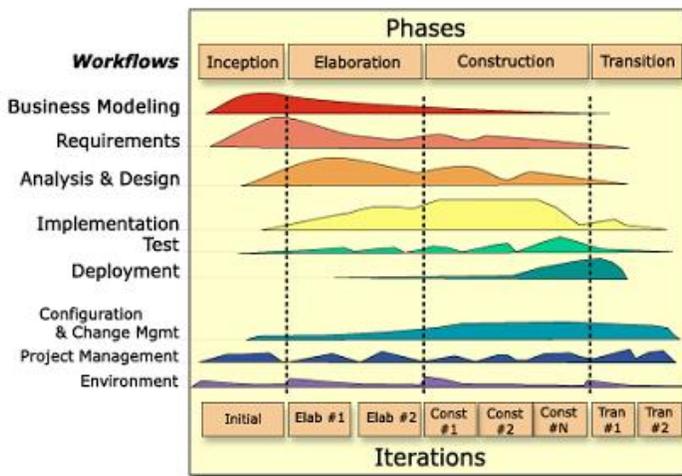
Anexo 2: Metodología Extreme Programming (XP)



Anexo 3: Metodología Microsoft Solution Framework (MSF)



Anexo 4: Metodología Rational Unified Process (RUP)



Anexo 5: Descripción de las entidades de la Base de Datos

Municipio		
Contiene información de los municipios		
Atributo	Tipo	Descripción
Id_Municipio	int	Id del municipio es la llave de la tabla
Id_Provincia	int	Id de la Provincia
Nombre municipio	nvarchar(255)	Nombre de municipio

Provincia		
Contiene información de las Provincia		
Atributo	Tipo	Descripción
Id_Provincia	int	Id dela Provincia es la llave de la tabla
Nombre Provincia	nvarchar(255)	Nombre de municipio

Modificaciones		
Contiene información relacionada con las modificaciones del personal de la Universidad.		
Atributo	Tipo	Descripción
Solapín	varchar(10)	Número del solapín.
Sexo	char(1)	Sexo de la persona.
Id_Area	varchar(10)	El id del área en la que se encuentra el trabajador.
Id_Cargo	varchar(10)	El id del cargo que desempeña el trabajador.
Docente	Bit	Si es docente o no.
Becado	Bit	Si es becado o no.
Id_Municipio	varchar(10)	El id del municipio que proviene el trabajador.
Id_Raza	Integer	El id de la raza.
CI	varchar(11)	Carné de identidad
Fech_Modif	Date time	La fecha en que se realizo algún cambio a sus datos.
Antigüedad	Integer	La antigüedad en la profesión.
Id_Catdocente	varchar(10)	El id de la categoría docente que tiene otorgada.

Altas		
Contiene información relacionada con las altas del personal de la Universidad.		
Atributo	Tipo	Descripción
Solapín	varchar(10)	Número del solapín.

Sexo	char(1)	Sexo de la persona.
Id_Area	varchar(10)	El id del área en la que va a ingresar.
Id_Cargo	varchar(10)	El id del cargo que desempeña el trabajador.
Docente	Bit	Si es docente o no.
Becado	Bit	Si es becado o no.
Id_Municipio	varchar(10)	El id del municipio que proviene el trabajador.
Id_Raza	Integer	El id de la raza.
CI	varchar(11)	Carné de identidad
Fech_Altas	Date time	La fecha en que ingreso a la universidad
Antigüedad	Integer	La antigüedad en la profesión.
Id_CatDocente	varchar(10)	El id de la categoría docente que tiene otorgada.

Bajas		
Contiene información relacionada con las bajas del personal de la Universidad.		
Atributo	Tipo	Descripción
Solapín	varchar(10)	Número del solapín.
Sexo	char(1)	Sexo de la persona.
Id_Area	varchar(10)	El id del área en que se encontraba trabajando.
Id_Cargo	varchar(10)	El id del cargo que desempeñaba el trabajador.
Docente	Bit	Si es docente o no.
Becado	Bit	Si es becado o no.
Id_Municipio	varchar(10)	El id del municipio que proviene el trabajador.
Id_Raza	Integer	El id de la raza.
CI	varchar(11)	Carné de identidad
Fech_Bajas	Date time	La fecha en que causo baja del centro
Antigüedad	Integer	La antigüedad en la profesión.
Id_CatDocente	varchar(10)	El id de la categoría docente que tiene otorgada.
Motivo_Baja	Integer	El motivo de la baja.

Errores		
Contiene información sobre los errores ocurridos en la ejecución del sistema de migración de datos.		
Atributo	Tipo	Descripción
Error	varchar(255)	El tipo de error ocurrido.
Fecha_error	Date time	Fecha en la que ocurrió el error.

Aplicación		
Contiene información relacionada con el funcionamiento de la aplicación.		
Atributo	Tipo	Descripción
Hora_Inicio	Date time	Hora en que se inició la ejecución de la aplicación.
Hora_Fin	Date time	Hora en que terminó la ejecución de la aplicación.
Fecha_Ejecucion	Date time	Día, Mes y Año en que comenzó la ejecución.

Usuario		
Contiene información relacionada con los usuarios del sistema.		
Atributo	Tipo	Descripción
Usuario	varchar(50)	Usuario con el que será registrado
contraseña	varchar(50)	Contraseña para acceder a la aplicación
Rol	varchar(50)	Rol que será asignado en la aplicación
Correo	varchar(50)	Correo Electrónico

Anexo 6: Autenticar Usuario



The screenshot shows the login interface for the 'Sistema de migración de Datos'. At the top, there is a blue header with the title 'Sistema de migración de Datos' and the UCI logo. Below the header, on the left, is a vertical sidebar with a photo of a man at a desk. The main content area features a login form with the following elements:

- Autenticarse** (header)
- Usuario:
- Contraseña:
- Entrar (button)

To the right of the form is a graphic of two computers connected by a stream of binary code. Below the graphic, the text reads: 'Sistema Migración de datos del Sistema ASSETS al Sistema Trabajadores de la Universidad de las Ciencias Informáticas (UCI)'.

Anexo 7: Administrar usuarios



The screenshot shows the user administration interface. At the top, there is a blue header with the title 'Sistema de migración de Datos' and the UCI logo. Below the header, there is a navigation menu with the following items: Administrar, Reporte, Bajas, Altas, Modificaciones, Errores. Below the header, on the left, is a vertical sidebar with a photo of a man at a desk. The main content area features a dropdown menu titled 'Administrar' with the following options: Insertar, Cambiar Rol, and Eliminar. Below the sidebar, there is a link 'Salir' and the text 'Bienvenido: administrador'.

Insertar usuarios



Administrar
Insertar

Seleccione el rol

- administrador
- especialista

Sign Up for Your New Account

Usuario:

Contraseña:

Confirmar contraseña:

Correo electrónico:

Pregunta de seguridad:

Respuesta de seguridad:

Crear Usuario



[Salir](#)
Bienvenido: administrador

Eliminar usuarios



Administrar
Eliminar

Usuarios



[Salir](#)
Bienvenido: administrador

Cambiar rol a usuarios

The screenshot shows the header of the 'Sistema de migración de Datos' with navigation links: Administrar, Reporte, Bajas, Altas, Modificaciones, Errores, and the UCI logo. The main content area is titled 'Administrar' and features a dropdown menu with 'Cambiar Rol' selected. Below this, there are two dropdown menus labeled 'Usuarios' and 'Roles', both with 'Seleccione' as the current selection. There are two text input fields: 'Usuario a Actualizar' and 'Rol Asignado'. An 'Actualizar' button is positioned below these fields. On the left side, there is a profile picture of a man, a 'Salir' link, and the text 'Bienvenido: administrador'.

Anexo 8: Buscar bajas

The screenshot shows the same header as the previous page. The main content area is titled 'Bajas' and features a dropdown menu that is open, displaying a list of categories: General, Antigüedad, Áreas, Cargos, Sexo, Raza, Municipio, Categoría Docente, Docentes, and Becados. On the left side, there is a profile picture of a man, a 'Salir' link, and the text 'Bienvenido: administrador'.

Anexo 9: Buscar altas

The screenshot displays the header of the 'Sistema de migración de Datos' web application. The header includes a navigation menu with the following items: Administrar, Reporte, Bajas, Altas, Modificaciones, and Errores. The UCI logo is visible in the top right corner. Below the header, there is a user profile section on the left featuring a photo of a man at a desk, a 'Salir' link, and the text 'Bienvenido: administrador'. To the right of the profile is a dropdown menu titled 'Altas' with a list of search categories: General, Antigüedad, Áreas, Cargos, Sexo, Raza, Municipio, Categoría Docente, Docentes, and Becados.

Anexo 10: Buscar modificaciones

The screenshot displays the header of the 'Sistema de migración de Datos' web application. The header includes a navigation menu with the following items: Administrar, Reporte, Bajas, Altas, Modificaciones, and Errores. The UCI logo is visible in the top right corner. Below the header, there is a user profile section on the left featuring a photo of a man at a desk, a 'Salir' link, and the text 'Bienvenido: administrador'. To the right of the profile is a dropdown menu titled 'Modificaciones' with a list of search categories: General, Antigüedad, Áreas, Cargos, Sexo, Raza, Municipio, Categoría Docente, Docentes, and Becados.

Anexo11: Buscar errores

The screenshot shows the header of the 'Sistema de migración de Datos' application. The header includes a navigation menu with the following items: Administrar, Reporte, Bajas, Altas, Modificaciones, and Errores. The UCI logo is visible on the right side of the header. Below the header, there is a sidebar on the left containing a profile picture of a man, a 'Salir' link, and the text 'Bienvenido: administrador'. The main content area is titled 'Errores' and features a dropdown menu with 'General' selected. The page has a blue gradient background.

Anexo12: Buscar reportes

The screenshot shows the header of the 'Sistema de migración de Datos' application. The header includes a navigation menu with the following items: Administrar, Reporte, Bajas, Altas, Modificaciones, and Errores. The UCI logo is visible on the right side of the header. Below the header, there is a sidebar on the left containing a profile picture of a man, a 'Salir' link, and the text 'Bienvenido: administrador'. The main content area is titled 'Reporte' and features a dropdown menu with 'General' selected. The page has a blue gradient background.

GLOSARIO

Actor: Alguien o algo, fuera del sistema o negocio que interactúa con el sistema o negocio.

Artefactos: Una parte de la información que es producida, modificada, o usada por un proceso. Define un área de responsabilidad y está sujeta al control de versión. Un artefacto puede ser un modelo, un elemento del modelo, o un documento. Un documento puede adjuntar otros documentos. Una parte de la información que es usada o producida por un proceso de desarrollo del software. Un artefacto puede ser un modelo, una descripción, o un software.

Browser: El navegador puede considerarse como una interfaz de usuario universal. Dentro de sus funciones están la petición de las páginas Web, la representación adecuada de sus contenidos y la gestión de los posibles errores que se puedan producir.

Base de Datos (BD): Conjunto de datos interrelacionados, almacenados con carácter más o menos permanente en la computadora, puede ser considerada una colección de datos variables en el tiempo.

CORBA: (Common Object Request Broker Architecture). Especificación de mensajes basada en objetos.

CGI: (Common Gateway Interface). Interface Común de Pasarela. Interface de intercambio de datos estándar en WWW a través del cual se organiza el envío de recepción de datos entre visualizadores y programas residentes en servidores WWW.

Clave foránea: Sirve para relacionar lógicamente varias tablas. Es una columna o combinación de columnas de una tabla cuyos valores coinciden con los valores de la clave primaria de otra tabla.

Clave primaria: Columna o columnas utilizadas para identificar de manera única las filas de una tabla.

Cliente: Una persona u organización, interna o externa a la organización productora que toma responsabilidad financiera por el sistema. El cliente es el último destinatario del producto desarrollado y sus artefactos.

CLR: “Common Language Runtime” es el núcleo de la plataforma .Net. Es el motor encargado de gestionar la ejecución de las aplicaciones para ella desarrolladas y a las que ofrece numerosos servicios que simplifican su desarrollo y favorecen su fiabilidad y seguridad.

Dirección IP: es un número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo IP (Internet Protocol), que corresponde al nivel de red o nivel 3 del modelo de referencia OSI. Dicho número no se ha de confundir con la dirección MAC que es un número hexadecimal fijo que es asignado a la tarjeta o dispositivo de red por el fabricante, mientras que la dirección IP se puede cambiar.

Encriptación (encryption): Acto de proteger un archivo legible.

Firewall: Una barrera de seguridad que ayuda a limitar los accesos no autorizados a un sistema de computadoras.

FTP: Protocolo de transferencia de archivos (Transferer File Protocol); una protocolo para transferir archivos hacia y desde computadoras remotas.

GUI: Interfaz Gráfica de Usuario (Graphical User Interface); una pantalla gráfica que permite al usuario interactuar con la computadora.

HTTP: (Hyper Text Transfer Protocol HTTP es un protocolo del nivel de aplicación para sistemas de información multimedia distribuidos. Es un protocolo no orientado a estado que puede ser utilizado para más propósitos que para manejar ficheros HTML).

Hipervínculo (hyperlink): Un elemento de HTML que, cuando se hace clic en él, permite que las personas pasen a otros documentos, imágenes, sonidos o películas.

HTML (HyperText Markup Language): Lenguaje de marcado de Hipertexto. Es el lenguaje estándar para describir el contenido y la apariencia de las páginas en el WWW.

Internet: Un sistema de redes formado por computadoras y usuarios con alcance mundial.

Interfaz: Una colección de operaciones que se usan para especificar el servicio de una clase o de un componente. Un juego nombrado de operaciones que caracterizan la conducta de un elemento. La Interfaz hombre-máquina es un canal comunicativo entre el usuario y el ordenador.

Java: Lenguaje de programación, similar al C++, que extiende las capacidades de Web.

Navegar: Usar Internet o World Wide Web.

.Net Framework: El .Net Framework de Microsoft es un ambiente para desarrollar, implementar y ejecutar servicios Web XML y otras aplicaciones. Consiste de tres partes principales: el tiempo de ejecución del lenguaje común, los tipos de Framework, y ASP.Net. Como una infraestructura complementaria, el .Net Compact Framework de Microsoft, es un conjunto de interfases de programación que permiten a los desarrolladores apuntar a dispositivos móviles.

PHP: (Preprocessed Hypertext Pages).

Proceso: Secuencia de actividades invocadas para producir un producto de software.

Rol: Papel, cometido o función que tiene o desempeña, que interpreta un actor.

Servidor Web: El servidor Web es un programa que corre sobre el servidor que escucha las peticiones HTTP que le llegan y las satisface. Dependiendo del tipo de la petición, el servidor Web buscará una

página Web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición.

SOAP (Single Object Access Protocol): Es un protocolo de mensaje liviano basado en XML, usado para codificar los mensajes de Servicio Webs antes de enviarlos por la red. Los mensajes SOAP son independientes de cualquier sistema operativo y protocolo, y pueden ser transportados usando una variedad de protocolos de internet, incluyendo HTTP, SMTP y MIME. Permite que programas que corren en diferentes sistemas operativos se comuniquen.

SDL (Servicio Web Description Language): Es un lenguaje en formato XML que define las operaciones que proporciona un servicio, desarrollado conjuntamente por Microsoft e IBM.

Servidor: Es un computador o un software que provee una clase especial de servicio a los software clientes que están corriendo en otros computadores y que lo accesan para realizar una función determinada. Un computador funcionando como servidor puede tener operando varios software servidores para prestar servicios, por ejemplo: servidor de WWW, servidor de FTP, de Mail, etc.

SGBD: Es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez.

Software: Palabra en inglés utilizada para indicar a los programas de computadoras, a las aplicaciones.

TCP/IP: "Transmission Control Protocol/Internet Protocol" Es el conjunto de protocolos que definen a Internet. Originalmente diseñado para el sistema operativo UNIX, hoy en día existe software TCP/IP disponible para la mayoría de los sistemas operativos. Para poder utilizar la Internet, su computador debe tener software TCP/IP.

UDDI (Universal Description Discovery and Integration): Es un directorio universal de Servicios Web basado en XML, que permite publicar, localizar y utilizar servicios Web.

URL: Localizador Uniforme de Recursos (Uniform Resource Locator); el medio para localizar una página de inicio de Web.

UML: “Unified Modeling Language” Lenguaje gráfico que brinda un vocabulario y reglas para especificar, construir, visualizar y documentar los artefactos de un sistema utilizando el enfoque orientado a objetos.

Servicio Web: Permiten publicar componentes de negocios(Servicio Web) en un servidor web.

XML (Extensible Markup Language): Es un metalenguaje de marcas que permite definir cómo es la información que se transmite. Esto permite una comunicación de datos entre distintos sistemas. Es la base de los Servicios Web, y a pesar de su sencillez aparente, está transformando completamente la creación y el uso de software. Es la solución a un problema de comunicación entre programas de ordenador, pues la información generalmente queda fuertemente ligada al programa con el cual fue creada, y es así como se pierde mucho tiempo en pasar de un formato de definición a otro. El contenido almacenado en un documento XML se puede transferir fácilmente a través de la red.