

ESCUELA POLITÉCNICA DEL EJÉRCITO

DPTO. DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

**DESARROLLO, IMPLEMENTACIÓN E INTEGRACIÓN
DEL MÓDULO DE CONTROL DE ASISTENCIA DEL
PERSONAL DOCENTE DE LA ESPE, ENTRE EL
SUBSISTEMA ACADÉMICO Y ROLES DE PAGO VÍA
WEB.**

Previa a la obtención del Título de:

INGENIERO EN SISTEMAS E INFORMÁTICA

POR: FRANCISCO JAVIER NARANJO MERIZALDE

SANGOLQUI, 13 de Febrero de 2009

CERTIFICACIÓN

Certifico que el presente trabajo fue realizado en su totalidad por el Sr. FRANCISCO JAVIER NARANJO MERIZALDE como requerimiento parcial a la obtención del título de INGENIERO EN SISTEMAS E INFORMÁTICA.

Fecha

ING JAIME MAILA

AGRADECIMIENTOS

A Dios, ya que todas las metas e ilusiones cumplidas en nuestra vida se las debemos a Él.

A mis padres, Marcelo y Grace, por el enorme esfuerzo y apoyo recibido de ellos a lo largo de todos estos años, por el ejemplo, el sacrificio y la paciencia en mi educación, por el inmenso cariño entregado, les doy las gracias por ser unos padres excelentes y permitirme así alcanzar mis objetivos.

A mi hermano, Roberto, por ser mi mejor amigo y compañero de la vida, por apoyarme en cada momento y ayudarme a superar muchísimos problemas, por hacer de cada día un momento alegre e inolvidable al compartirlo generosamente junto a mí.

A mi abuelita, Emma, por haber sido partícipe de mi formación desde muy pequeño, por la gran dedicación mostrada para educarme y permitirme crecer como persona y por acompañarme hasta la culminación de mi carrera.

A mis abuelitos, Edmundo y Maria Antonieta, por compartir su experiencia, sus lecciones y enseñanzas y facilitar así el cumplimiento de mis metas.

A mi hermana, María Fernanda, por el enorme cariño y el gran ejemplo de tenacidad brindado, el mismo que siempre me ayuda a continuar y sobrellevar los obstáculos.

A mis tíos: Maria Yolanda, José Alberto, Patricio y Nelson, por educarme siempre y brindarme muchísimo cariño a lo largo de mi vida, por ayudarme a ser una mejor persona y con su ejemplo darme fuerzas en los momentos más difíciles.

A Natalia, por la ayuda, apoyo y cariño recibido en la etapa final de mi vida profesional, por el trabajo realizado que facilitó enormemente el desarrollo de este proyecto.

A toda mi familia, por permitirme aprender de cada uno de ellos y darme ánimos e incentivar me a siempre seguir adelante.

A mis amigos, por compartir conmigo todo este tiempo, por los buenos consejos, lealtad y todo el apoyo demostrado.

Al Colegio Intisana y a la Escuela Politécnica del Ejército por brindarme una educación de calidad basada en valores y hacer de mi un mejor profesional.

A los directores de este proyecto de grado, Jaime Mayla y Pilar Bucheli, por el esfuerzo y dedicación puestos en el desarrollo del mismo.

Francisco Javier Naranjo Merizalde

ÍNDICE DE CONTENIDOS

Resumen	1
CAPITULO 1	2
Introducción	2
1.1- Planteamiento del Problema.....	2
1.2- Justificación	3
1.3- Objetivos	3
1.3.1- Objetivo General.....	3
1.3.2- Objetivos Específicos	4
1.4- Alcance.....	4
CAPITULO 2	6
Marco Teórico	6
2.1- Aplicación informática.....	6
2.1.1- Software Empresarial	6
2.1.2- Software de Infraestructura Empresarial	7
2.1.3- Software de Acceso a Contenido	7
2.1.4- Software Educativo.....	7
2.1.5- Software de Simulación.....	8
2.1.6- Software de Desarrollo de Contenidos	8
2.1.7- Software de Ingeniería	8
2.1.8- Software de Desarrollo de Sistemas	8
2.2- Aplicación Web.....	9
2.2.1- Historia	10
2.2.2- Estructura	11
2.2.3- Características de una Aplicación Web	13
2.3- Ingeniería de Software	14
2.3.1- Proceso de Desarrollo de Software	15
2.3.2- Metodología de Desarrollo de Software.	17
2.4- Programación Orientada a Objetos	23
2.5- Lenguaje de marcado de hipertexto (HTML)	27
2.5.1- Historia	27
2.5.2- Marcado HTML.....	28
2.6- Tecnología JSP	31
2.7- Lenguaje Unificado de Modelado (UML)	35
2.7.1- Métodos.....	35
2.7.2- Modelamiento.....	36

2.7.3- Diagramas	36
2.7.3.1- Diagramas de Estructura.....	37
2.7.3.2- Diagramas de Comportamiento.....	39
2.7.3.3- Diagramas de Interacción.....	41
CAPITULO 3	44
Análisis de Requerimientos	44
3.1- Ámbito del proyecto	44
3.2- Descripción General	45
3.3- Requerimientos Funcionales	46
3.3.1- CU001-Autenticación de Usuario en el sistema	50
3.3.2- CU002-Cerrar Sesión	51
3.3.3- CU003-Ir a Web Principal.....	52
3.3.4- CU004-Elegir Rol.....	53
3.3.5- CU005-Ver Reporte Personal.....	54
3.3.6- CU006-Ver Reporte Docente (Administrador)	55
3.3.7- CU007-Ver Reporte Docente (Coordinador)	56
3.3.8- CU008- Ver Reporte Docente (Director Departamento)	58
3.3.9- CU009-Ingresa Reporte Docente (Coordinador)	59
3.3.10- CU010- Ingresa Reporte Docente (Director Departamento)	62
3.4- Requerimientos de interfaces externas	66
3.5- Requerimientos No Funcionales.....	66
CAPITULO 4	68
Diseño del sistema	68
4.1- Diseño de la Estructura del Sistema.....	68
4.2- Definición de Metodología	69
4.3- Diseño de la Arquitectura de la Aplicación	69
4.4- Definición de la plataforma de desarrollo	70
4.5- Servidor de Aplicaciones JAVA	71
4.6- Motor de Base de Datos	71
4.7- Reconocimiento de Objetos.....	72
4.7.1- Entidad de Datos-Calendario	72
4.7.2- Entidad de Datos-CaracterísticasHoras.....	72
4.7.3- Entidad de Datos-Asigna_Coordinador	73
4.7.4- Entidad de Datos-DirectorDepartamento.....	74
4.7.5- Entidad de Datos-DocenteXDepartamento	75
4.7.6- Entidad de Datos-Horario	76
4.7.7- Entidad de Datos-HoraDictada.....	76

4.7.8- Entidad de Datos-MateriaDocente	77
4.7.9- Entidad de Datos-ReporteGenerado	78
4.7.10- Entidad de Datos-TipoDocente	79
4.8- Clases Utilitarias.....	80
4.9- Diseño de Base de Datos.....	82
4.10- Codificación del Sistema	89
4.11- Modelo de Componentes	90
CAPITULO 5	92
Implementación y Pruebas	92
5.1- Implementación	92
5.2- Pruebas	100
5.3- Manual de Usuario	101
5.3.1- Usuario General	101
5.3.2- Usuario Docente.....	104
5.3.3- Usuario Administrador del Proceso de Control de Asistencia	106
5.3.4- Usuario Coordinador de Carrera	111
5.3.5- Usuario Director de Departamento.....	117
CAPITULO 6	123
Conclusiones y Recomendaciones	123
6.1- Conclusiones.....	123
6.2- Recomendaciones.....	126
BIBLIOGRAFIA	129

LISTA DE FIGURAS

Figura 2.1. Estructura de una aplicación WEB	12
Figura 2.2. Proceso de Desarrollo de Software	15
Figura 2.3. Ciclo de Vida del Software	18
Figura 2.4. Ejemplo de Clase	23
Figura 2.5. Ejemplo de Método	24
Figura 2.6. Ejemplo de Objeto	25
Figura 2.7. Ejemplo de Herencia	26
Figura 2.8. Ejemplo de Código HTML	28
Figura 2.9. Ejemplo de Página HTML	28
Figura 2.10. Ejemplo de uso conjunto de elementos HTML	30
Figura 2.11. Interfaz Ejemplo de uso conjunto de elementos HTML	31
Figura 2.12. Ejemplo de JSP	33
Figura 2.13. Arquitectura basada en Portlets	34
Figura 2.14. Diagrama de Estructura	37
Figura 2.15. Diagrama de Componentes	37
Figura 2.16. Diagrama Detallado de Estructura	38
Figura 2.17. Diagrama de Despliegue	38
Figura 2.18. Diagrama de Objetos	39
Figura 2.19. Diagrama de Paquetes	39
Figura 2.20. Diagrama de Actividad	40
Figura 2.21. Diagrama de Estado	40
Figura 2.22. Diagrama de Caso de Uso	41
Figura 2.23. Diagrama de Comunicación	41
Figura 2.24. Diagrama de Interacción	42
Figura 2.25. Diagrama de Secuencia	42
Figura 2.26. Diagrama de Tiempo	43
Figura 3.1. Diseño General del Sistema	45
Figura 3.2. Autenticación de Usuario en el sistema	50
Figura 3.3. Cerrar Sesión	51
Figura 3.4. Ir a Web Principal	52
Figura 3.5. Elegir Rol	53
Figura 3.6. Ver Reporte Personal	54
Figura 3.7. Ver Reporte Docente (Administrador)	55
Figura 3.8. Ver Reporte Docente (Coordinador)	56

Figura 3.9. Ver Reporte Docente (Director Departamento)	58
Figura 3.10. Ingresar Reporte Docente (Coordinador)	59
Figura 3.11. Ingresar Reporte Docente (Director Departamento).....	62
Figura 4.1. Estructura del Sistema	68
Figura 4.2. Arquitectura del Sistema	70
Figura 4.3. Entidad de Datos–Calendario	72
Figura 4.4. Entidad de Datos-CaracteristicasHoras	73
Figura 4.5. Entidad de Datos-Asigna_Coordinador	74
Figura 4.6. Entidad de Datos-DirectorDepartamento	75
Figura 4.7. Entidad de Datos-DocenteXDepartamento	75
Figura 4.8. Entidad de Datos-Horario.....	76
Figura 4.9. Entidad de Datos-HoraDictada.....	77
Figura 4.10. Entidad de Datos-MateriaDocente	78
Figura 4.11. Entidad de Datos-ReporteGenerado	78
Figura 4.12. Entidad de Datos-TipoDocente	79
Figura 4.13. Tabla Calendario	82
Figura 4.14. Tabla Coordinador.....	82
Figura 4.15. Tabla Caracteristicas_Horas	83
Figura 4.16. Tabla Director_Departamento	84
Figura 4.17. Tabla Horas_Docente	84
Figura 4.18. Tabla Perfiles	85
Figura 4.19. Tabla PerfilesXUsuario.....	85
Figura 4.20. Tabla Reportes_Generados	86
Figura 4.21. Tabla Servicios.....	86
Figura 4.22. Tabla ServiciosXPerfil	87
Figura 4.23. Tabla Usuarios	87
Figura 4.24. Tabla Tipo_Docente.....	88
Figura 4.25. Diagrama de Base de Datos	89
Figura 4.26. Diagrama de Componentes-Web.....	90
Figura 4.27. Diagrama de Componentes-Clases	91
Figura 4.28. Diagrama de Despliegue.....	91
Figura 5.1. Barra de Herramientas Eclipse IDE SDK.	94
Figura 5.2. Ventana de Configuración Eclipse (JAVA).	95
Figura 5.3. Ventana de Configuración Eclipse (Tomcat).	96
Figura 5.4. Ventana de Nuevo Proyecto.	97
Figura 5.5. Configuración de Proyecto JAVA.	98
Figura 5.6. Proyecto JAVA importado en Espacio de Trabajo.....	99

Figura 5.7. Pantalla de Inicio de Sesión.....	102
Figura 5.8. Pantalla de Error en Inicio de Sesión.....	102
Figura 5.9. Pantalla de Bienvenida.....	102
Figura 5.10. Lista de Roles para el usuario.....	103
Figura 5.11. Servicios para un rol determinado.....	103
Figura 5.12. Cerrar Sesión ir a Página Principal	104
Figura 5.13. Servicios para el docente.....	104
Figura 5.14. Escoger periodo de asistencia.....	104
Figura 5.15. Modalidades de Reportes.....	105
Figura 5.16. Reporte de Docente.....	105
Figura 5.17. Reporte Resumido.....	106
Figura 5.18. Servicios para el administrador.....	106
Figura 5.19. Escoger periodo de asistencia.....	107
Figura 5.20. Escoger docente.....	107
Figura 5.21. Modalidades de Reportes.....	107
Figura 5.22. Identificar Usuario.....	108
Figura 5.23. Asignación de Roles.....	109
Figura 5.24. Confirmación de Asignación.....	109
Figura 5.25. Ingreso de Calendarios.....	110
Figura 5.26. Confirmación Calendarios.....	110
Figura 5.27. Ingreso Coordinador.....	111
Figura 5.28. Asignación realizada.....	111
Figura 5.29. Selección de Carrera.....	111
Figura 5.30. Servicios de Ingreso de Reportes.....	112
Figura 5.31. Informe de Reportes Generados.....	113
Figura 5.32. Escoger Docente.....	113
Figura 5.33. Escoger Materia.....	114
Figura 5.34. Escoger Día de Clases.....	114
Figura 5.35. Ingreso de Características.....	115
Figura 5.36. Confirmación de Éxito.....	115
Figura 5.37. Interfaz Dinámica de Ingreso de Reportes.....	116
Figura 5.38. Selección de Carrera.....	116
Figura 5.39. Selección de Docente.....	117
Figura 5.40. Selección de Departamento.....	117
Figura 5.41. Selección de Docente.....	118
Figura 5.42. Selección de Carrera.....	118
Figura 5.43. Servicios de Ingreso de Reportes.....	119

Figura 5.44. Selección de Docente.	119
Figura 5.45. Lista Días de Asistencia.	120
Figura 5.46. Ingreso Características Administrativas.	120
Figura 5.47. Confirmación de Éxito.	121
Figura 5.48. Lista Días de Clase Semanal.	121
Figura 5.49. Ingreso Horas Adicionales.	121
Figura 5.50. Confirmación de Éxito.	122
Figura 5.51. Aumentar o Quitar Horas.	122

LISTA DE TABLAS

Tabla 2.1. Niveles de Madurez en el Desarrollo de Software	16
Tabla 2.2. Estructura de Elemento HTML	29
Tabla 2.3. Principales Elementos HTML	29
Tabla 3.1. Autenticación de Usuario en el sistema.....	50
Tabla 3.2. Cerrar Sesión	51
Tabla 3.3. Ir a Web Principal	52
Tabla 3.4. Elegir Rol.....	53
Tabla 3.5. Ver Reporte Personal	54
Tabla 3.6. Ver Reporte Docente (Administrador)	55
Tabla 3.7. Ver Reporte Docente (Coordinador).....	57
Tabla 3.8. Ver Reporte Docente (Director Departamento)	58
Tabla 3.9. Ingresar Reporte Docente (Coordinador)	60
Tabla 3.10. Generar Reporte de Docentes	61
Tabla 3.11. Ingresar Características Horas Clase.....	61
Tabla 3.12. Ingresar Reporte Docente (Director Departamento).....	63
Tabla 3.13. Ingresar Características Administrativas	64
Tabla 3.14. Ingresar Horas Adicionales	64
Tabla 3.15. Aumentar o Quitar Horas.....	65
Tabla 4.1. Entidad de Datos-Calendarario	72
Tabla 4.2. Entidad de Datos-CaracteristicasHoras.....	72
Tabla 4.3. Entidad de Datos-Asigna_Coordinador	73
Tabla 4.4. Entidad de Datos-DirectorDepartamento.....	74
Tabla 4.5. Entidad de Datos-DocenteXDepartamento	75
Tabla 4.6. Entidad de Datos-Horario	76
Tabla 4.7. Entidad de Datos-HoraDictada.....	76
Tabla 4.8. Entidad de Datos-MateriaDocente	77
Tabla 4.9. Entidad de Datos-ReporteGenerado	78
Tabla 4.10. Entidad de Datos-TipoDocente.....	79

Resumen

El presente proyecto de Grado tiene como objetivo implementar un Sistema de Software Informático orientado a la web, para la administración del área de Recursos Humanos, este programa incluye la gestión de Control de Asistencia para el Personal Docente de la Escuela Politécnica del Ejército.

Para la implementación de este programa, fue necesario el diseño de un módulo que permita satisfacer las necesidades antes mencionadas e integrarlo al actual Sistema Académico de la Institución.

El Sistema de Software Informático cumplirá con los requerimientos necesarios, logrando así perfeccionar los procedimientos de funcionamiento en las áreas de administración y gestión del talento humano de la institución.

El proyecto se realizó con el apoyo de dos dependencias de la ESPE. La primera, la Unidad de Administración de Recursos Humanos, los mismos que definieron y establecieron los lineamientos y requerimientos en base a los cuales el software fue diseñado y la unidad de Tecnologías de Información que proveyó los medios necesarios para el desarrollo de este proyecto, es justamente el trabajo conjunto de estas áreas el que logró que el sistema realizado se acople de mejor manera al Sistema Integrado de Información de la institución haciendo de éste una herramienta acorde a las necesidades de agilidad en el funcionamiento de la institución y a la vanguardia de las exigencias de la comunidad a la que está dirigido el servicio que presta este establecimiento.

CAPITULO 1

Introducción

La Escuela Politécnica del Ejército está ubicada en la ciudad de Sangolquí del cantón Rumiñahui de la provincia de Pichincha. Esta institución brinda servicios a la comunidad y enfoca su campo de acción hacia la docencia, la investigación y la extensión.

Con el aumento de personal docente, la institución se ha visto en la necesidad de implementar un sistema de software informático orientado a la web, para la administración del área de Recursos Humanos, que incluye la gestión de control de asistencia para el personal docente.

El presente proyecto de grado, busca desarrollar e implementar el módulo que permita satisfacer las necesidades antes mencionadas e integrarlo al actual sistema académico de la Institución. El sistema de software informático cumplirá con los requerimientos necesarios para perfeccionar los procedimientos de funcionamiento en las áreas de administración y gestión del talento humano de la institución.

1.1- Planteamiento del Problema

Durante los últimos años, la cantidad de personal docente en la Escuela Politécnica del Ejército, ha sufrido un gran incremento, y contrariamente, el software utilizado en la institución no fue reemplazado para satisfacer esta demanda. Efectos negativos como la falta de control del personal docente y administrativo, y la mala utilización de recursos, son resultado del uso de un Software no especializado (Ej.: Microsoft Office Excel TM©) en el área de recursos humanos. Esta realidad hace de la implementación de un sistema especializado, como el que se plantea en este proyecto de grado, una

necesidad para estar acorde a las necesidades de agilidad en el funcionamiento de la institución y a la vanguardia de las exigencias de la comunidad a la que está dirigido el servicio que presta este establecimiento.

1.2- Justificación

La aplicación e implementación del presente proyecto representará para la Escuela Politécnica del Ejército un avance tecnológico que permitirá perfeccionar los procedimientos de funcionamiento en las áreas de administración y gestión del talento humano de la institución. Así como también permitirá un ahorro significativo de recursos y una perspectiva de crecimiento a futuro coherente con la planificación institucional y por ende con la visión a largo plazo de la institución.

El proyecto de tesis, contará con el auspicio de la unidad de TIC's y el apoyo directo de la unidad de talento humano-docente, unidades que serán beneficiarias, por que existirá una integración entre el subsistema académico y de roles de pago del personal docente. Las unidades antes mencionadas se comprometen a dar todas las facilidades para que este proyecto termine con éxito y en el tiempo establecido.

1.3- Objetivos

1.3.1- Objetivo General

Desarrollar e implementar el módulo de control de asistencia del personal docente de la ESPE, mediante la integración del subsistema académico y roles de pago vía Web, para mejorar el intercambio de información de manera eficaz y eficiente.

1.3.2- Objetivos Específicos

- Analizar los requerimientos del módulo de control de asistencia de docentes.
- Realizar la integración con el sistema académico de la ESPE.
- Realizar las pruebas del módulo desarrollado.
- Implementar el módulo desarrollado en el sistema de software informático de la Escuela Politécnica del Ejército.

1.4- Alcance

El presente proyecto comprenderá el análisis, diseño e implementación del módulo de software informático orientado a la web para la administración del control de asistencia del personal docente de la Escuela Politécnica del Ejército.

En el proyecto se realizará el levantamiento de requerimientos del sistema informático demandado por la institución.

Utilizando los requerimientos obtenidos se procederá con el estudio y diseño del sistema informático a desarrollar.

Posteriormente se desarrollará el módulo que cumplirá los requerimientos obtenidos.

Por último, se realizarán las pruebas, correcciones e implementación del módulo desarrollado.

El desarrollo del módulo comprenderá los siguientes ítems:

- Ingreso y modificación de Asistencia por parte de los señores Coordinadores de Carrera.

- Ingreso y modificación de Asistencia por parte de los señores Directores de Departamento.
- Generación de reportes de control de asistencia del personal docente, por departamentos y carreras.
- Reportes para Administradores, Coordinadores, Directores y Docentes.

CAPITULO 2

Marco Teórico

2.1- Aplicación informática

Una aplicación informática es un sistema basado en computadora diseñado con el fin de facilitar la realización de una tarea específica a los usuarios de la misma; es una subclase de software que busca potenciar la capacidad de procesamiento de un computador para ponerla a disposición de la automatización de procesos manuales.

Dentro de las ciencias de la computación, una aplicación informática es un programa diseñado para ayudar a las personas a realizar cierto tipo de trabajo; es importante diferenciarla de un sistema operativo el mismo que se encarga de administrar y operar cada elemento del hardware de un computador y es la base sobre el cual se ejecutan las aplicaciones informáticas. Dependiendo del trabajo que vayan a desempeñar estas pueden manipular texto, números, gráficos, audio, video o la combinación de estos.

Existen varios subtipos de aplicación informática:

2.1.1- Software Empresarial

Esta destinado a satisfacer las necesidades de los procesos de negocio y el tráfico de información de una organización. Ejemplos prácticos de estos son las aplicaciones Financieras, el software CRM¹ y todo tipo de aplicaciones que faciliten el trabajo en una institución o sus departamentos.

¹ Customer Relationship Management

2.1.2- Software de Infraestructura Empresarial

Este tipo de software proporciona el soporte necesario para facilitar la operación de los Sistemas Empresariales. Sus ejemplos más comunes son Bases de Datos y Servidores de Seguridad y Correo Electrónico.

2.1.3- Software de Acceso a Contenido

Este tipo de software se emplea principalmente para acceder a un sin número de contenidos que no admiten edición. Responde principalmente a la necesidad de los usuarios por acceder a información digital y medios de entretenimiento. Ejemplos de este tipo de software son: exploradores de Internet, reproductores de archivos multimedia (sonido, video) y juegos.

2.1.4- Software Educativo

Está íntimamente ligado al software de acceso a contenidos, sin embargo este tipo de software se encarga de monitorear y evaluar al usuario haciendo del aprendizaje un procedimiento altamente dinámico e interactivo. Actualmente incluso algunos sistemas operativos con Ubuntu han ingresado a este mercado con versiones del producto con un alto enfoque hacia el aspecto educativo como es el caso de Edubuntu.

2.1.5- Software de Simulación

Es software diseñado con el objetivo específico de simular sistemas físicos y abstractos tanto como para investigación, entrenamiento o entretenimiento.

2.1.6- Software de Desarrollo de Contenidos

Este software se utiliza para generar contenidos multimedia de todo tipo con diferentes propósitos, se incluyen entre estos editores gráficos, editores de audio y video, procesadores de contenido como DreamWeaver o HTMLPro.

2.1.7- Software de Ingeniería

Es un tipo de software dedicado a las ramas de ingeniería, proporcionando herramientas de diseño y análisis que facilitan enormemente las labores de un proyecto de ingeniería. Un ejemplo claro de este tipo de software es Auto CAD.

2.1.8- Software de Desarrollo de Sistemas

Software dedicado al desarrollo de sistemas informáticos, típicamente enmarcados en un ambiente de desarrollo integrado (IDE²). Proporcionan a los desarrolladores las herramientas necesarias para realizar cada una de las fases que comprenden el desarrollo de software.

² Integrated Development Environments

2.2- Aplicación Web

Una aplicación Web es una aplicación informática que es accedida a través de un explorador web, dicha aplicación puede encontrarse ya sea en la red pública (Internet) o pertenecer a una red privada (Intranet).

Es precisamente la difusión de este tipo de aplicación la que actualmente ha hecho que estas aplicaciones alcancen enormes niveles de aceptación en el mercado, ya que cualquier cambio en la misma no requiere realizar ninguna modificación ni instalación en los clientes. Otro gran punto a favor de estas es sin duda la accesibilidad que presentan ya que no necesitamos necesariamente de equipos informáticos para acceder a estas, basta con un dispositivo, ya sea estático o móvil, que cuente con acceso a la red donde se encuentre la aplicación para poder utilizarla.

A pesar de todas las cualidades que presentan las aplicaciones web, como era de esperarse presentan también grandes problemas. Uno es la gran necesidad de seguridad ya que al ser el Internet una red pública los ataques e intentos de violaciones de este tipo de sistemas se dan con mayor frecuencia, el otro es la enorme variedad de exploradores web que existen, los cuales no siguen claramente un estándar al momento de interpretar los contenidos mediáticos ni la programación establecida del lado del cliente lo que puede llegar a complicar el desarrollo de aplicaciones y incluso llegar a hacer inservibles las aplicaciones dependiendo del explorador que se utilice para acceder esta.

2.2.1- Historia

En el pasado las aplicaciones se desarrollaban bajo el modelo cliente servidor, en donde la lógica principal de la aplicación se encontraba del servidor y existían múltiples clientes que permitían al usuario acceder a las funcionalidades que ofrecía el servidor, sin embargo, el mantenimiento de este tipo de aplicaciones demandaba gran esfuerzo ya que cualquier cambio en el servidor significaba cambios en cada uno de los clientes elevando como es obvio el costo de soporte e influyendo negativamente en la productividad.

Luego por la influencia y la demanda de una sociedad globalizada y altamente dinámica se requirió del avance de la tecnología hacia aplicaciones que controlen de mejor manera estos inconvenientes, es así que las aplicaciones web toman un gran empuje, y las paginas HTML³ de la Internet dejan de ser simples medios de información visuales para pasar a convertirse en la interfaz gráfica del lado del cliente de aplicaciones enormemente útiles y con una característica en común; con las aplicaciones web se había logrado descifrar el dilema de la aplicación del lado del cliente reduciéndola a un simple pero muy potente explorador web.

Hoy en día existen muchas variedades de exploradores web, y la tecnología ha seguido evolucionado para traernos sitios web cada vez más dinámicos, con mejores interfaces front-end y con increíbles funcionalidades.

³ Hyper Text Markup Language

La tendencia de este grupo de aplicaciones es la utilización de portales en donde este se encarga de enmarcar un sin número de aplicaciones utilizadas por un usuario y de ponerlas a su disposición, las características principales de un portal web sin lugar a dudas son el alto nivel de seguridad que proporciona así como también que permite la personalización de su contenido para cada usuario.

2.2.2- Estructura

La estructura típica de una aplicación web está dada por tres capas:

- **Presentación**

La capa de presentación es el front-end al usuario final, en el caso de la aplicación web es el explorador, en este se puede incluir algo de la lógica del negocio y es la que se encarga de presentar interfaces gráficas altamente dinámicas y atractivas. Las tecnologías más utilizadas actualmente para esta capa son JavaScript, Ajax, Dojo.

- **Lógica de Aplicación**

Esta capa es la encargada de proporcionar toda la lógica sobre la cual trabajará la capa front-end, es decir, es en esta capa donde se incluirá la lógica del negocio del sistema a implementar.

Las tecnologías que se utilizan derivan principalmente de conocidos lenguajes de programación que se extendieron para poder manejar tecnología web dinámica, algunos ejemplos son: ASP, ASP.NET, CGI, ColdFusion, JSP/Java, PHP, embPerl, Python, or Ruby on Rails.

- **Almacenamiento**

Como en toda aplicación informática la necesidad de almacenar la información resultante de los procesos realizados en el sistema es esencial, por lo que la tercera capa esta compuesta del motor de base de datos o cualquier otro medio de almacenamiento en donde se depositará la información obtenida y donde se recuperará la información necesario para el funcionamiento de la aplicación. Algunos ejemplos de agentes utilizados en esta capa son Microsoft SQL Server, Base de Datos Oracle, MySql, Microsoft Access.

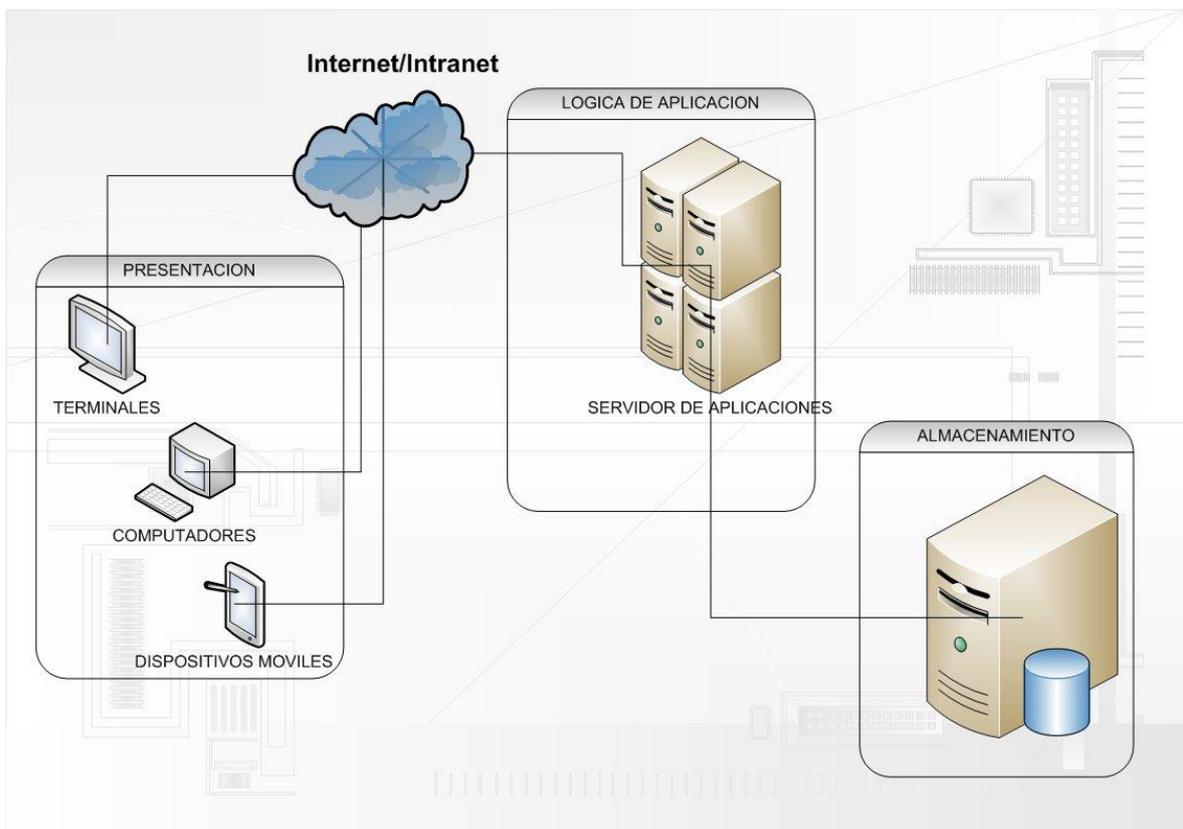


Figura 2.1. Estructura de una aplicación WEB

2.2.3- Características de una Aplicación Web

Una aplicación web podría ser como un conjunto de páginas HTML o en la actualidad XHTML, que tienen relación entre sí.

Los primeros sitios Web que se desarrollaron eran páginas llenas de texto y en su gran mayoría proporcionaban información básica de las diferentes empresas. Sin embargo, con el crecimiento a nivel mundial de la demanda de este tipo de aplicaciones la tecnología se vio obligada a tomar un cambio de dirección.

A la hora de diseñar aplicaciones web es necesario recordar que cumplir con las expectativas del usuario es de vital importancia por lo que el sitio Web debe ser fácil de usar y cumplir de manera efectiva con las necesidades de la persona que los utiliza.

Existen numerosos consejos a la hora de diseñar aplicaciones web, sin embargo, la W3C⁴ que es una organización dedicada a la elaboración de estándares, guías, software y herramientas, cuenta actualmente con una considerable cantidad de información acerca de reglas y documentación de gran cantidad de elementos que intervienen en la elaboración de un sitio web las mismas que permiten tener un modelo base a la hora de diseñar estas aplicaciones, entre las muchos temas que abarca la W3C están:

- ✓ Diseño de páginas HTML (Funcionalidad, Accesibilidad,...etc.)
- ✓ Codificación documentos XML
- ✓ Diseño de Web Services
- ✓ Nomenclatura de documentos WSDL

⁴ The World Wide Web Consortium

- ✓ Estándares JavaScript
- ✓ Reglas del protocolo SOAP⁵
- ✓ Arquitectura REST⁶
- ✓ Arquitectura SOA⁷

A pesar de que no constituyen un estándar, es recomendable apegarse a estos lineamientos para evitar incompatibilidades a la hora de integrar una aplicación web y lograr así un producto de muy alta calidad.

2.3- Ingeniería de Software

Es una rama de la ingeniería que se dedica a la aplicación de procedimientos y técnicas para el desarrollo de software. La utilización de estas técnicas asegurará que el producto final cumpla principalmente con los requerimientos para los cuales fue diseñado.

Según Fritz Bauer ingeniería de software: *“es el establecimiento y uso de principios robustos de la ingeniería a fin de obtener económicamente software que sea fiable y que funcione eficientemente sobre máquinas reales.”*⁸

Al ser esta una aplicación de procedimientos es necesario que siga un proceso de desarrollo, en donde se establecen bases generales acerca del trabajo a ser realizado para la elaboración del software.

⁵ Simple Object Access Protocol

⁶ Representational state transfer

⁷ Service-oriented architecture

⁸ Roger S. Pressman (2002). Ingeniería de Software, un enfoque práctico, Quinta Edición, Madrid, España.

2.3.1- Proceso de Desarrollo de Software

El proceso de desarrollo se caracteriza inicialmente por establecer un marco inicial del trabajo, en donde se definen tareas y procesos comunes a todo desarrollo de software. Este marco inicial nos sirve de base para establecer los diferentes puntos en particular que puede tener un proyecto. A continuación se define un conjunto de tareas las mismas que van a ser una colección de actividades de ingeniería, productos, entregas, estándares de calidad y métricas. Finalmente todo el proceso de ingeniería se ve englobado por las llamadas actividades de protección que son las que ayudan a minimizar los impactos y riesgos de la elaboración de aplicaciones y se ven aplicadas a lo largo de todo el proceso de software.



Figura 2.2. Proceso de Desarrollo de Software

Una de las últimas tendencias en cuanto al proceso de desarrollo es la de clasificar al mismo en base a la madurez de los procedimientos utilizados y a diversas métricas que evalúen los pasos que aparecen durante todas las etapas de desarrollo.

El SEI⁹ ha desarrollado un modelo que se basa en la evaluación del conjunto de funciones de ingeniería del software que deberían estar presentes conforme las organizaciones alcanzan niveles de madurez en el proceso, para determinar el estado de las mismas el SEI utiliza un cuestionario de evaluación y un esquema de cinco grados.

Tabla 2.1. Niveles de Madurez en el Desarrollo de Software

NIVELES DE MADUREZ EN EL DESARROLLO DE SOFTWARE	
NIVEL	DETALLES
UNO - INICIAL	El proceso de software se evalúa y diseña según el caso, se definen un número limitado de procesos y el éxito depende de esfuerzos individuales.
DOS - REPETIBLE	Se establecen procesos necesarios para la gestión del proyecto, para lograr éxito se repiten situaciones favorables anteriores.
TRES – DEFINIDO	Aparecen los procesos de gestión e ingeniería los mismos que son documentados, cada proyecto cuanta con documentación particular y definida.
CUATRO – GESTIONADO	Se utilizan medidas cuantificables y detalladas para evaluar los procesos definidos en el nivel 3.
CINCO	Se utiliza la evaluación y retroalimentación para implementar mejoras continuas al proceso de software, cumpliendo este con todos los estándares y procedimientos definidos en niveles anteriores.

En definitiva cuando se trabaja para diseñar un producto o sistema, es muy importante guiarse de un plan o procedimiento, una analogía altamente utilizada a largo de los años es la de comparar a la ingeniería de software con la elaboración de los planos de una edificación sin los cuales sería muy difícil lograr un producto de calidad.

⁹ Software Engineering Institute

2.3.2- Metodología de Desarrollo de Software.

La metodología de desarrollo de software es un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software que basándose en la ingeniería de software nos ayudan a construir aplicaciones de calidad.

En la metodología se van indicando paso a paso las tareas y actividades que se debe realizar a para lograr el producto informático deseado indicando además otros factores que intervienen en el desarrollo de un proyecto como son los roles de las personas, las métricas y el control de riesgos.

Las técnicas definen como se deben realizar las actividades del desarrollo identificadas mediante la metodología, combinan el empleo de un sin numero de técnicas como son representaciones gráficas y hasta procedimientos audiovisuales. Es necesario considerar que un técnica en particular puede ser utilizada en muchas actividades de la metodología de desarrollo de software.

En los últimos años han aparecido un sin numero de metodologías de desarrollo, las mismas que se han ido adaptando al crecimiento acelerado del mercado y a las múltiples y diversas exigencias que presumen las diferentes necesidades de software en la sociedad, de tal manera, que prácticamente en los últimos años son las metodologías las que se han adaptado a los requerimientos de los usuarios y no viceversa como inicialmente fue pensado.

Sin embargo, existe una base inicial el llamado “Ciclo de Vida del Software” en donde se definen una serie de pasos estandarizados que las metodologías deberían cumplir.

El ciclo de vida del software o el modelo lineal secuencial sugiere un esquema lineal para el desarrollo que comienza en un nivel de sistemas y progresa con el análisis, diseño, codificación, pruebas y mantenimiento.

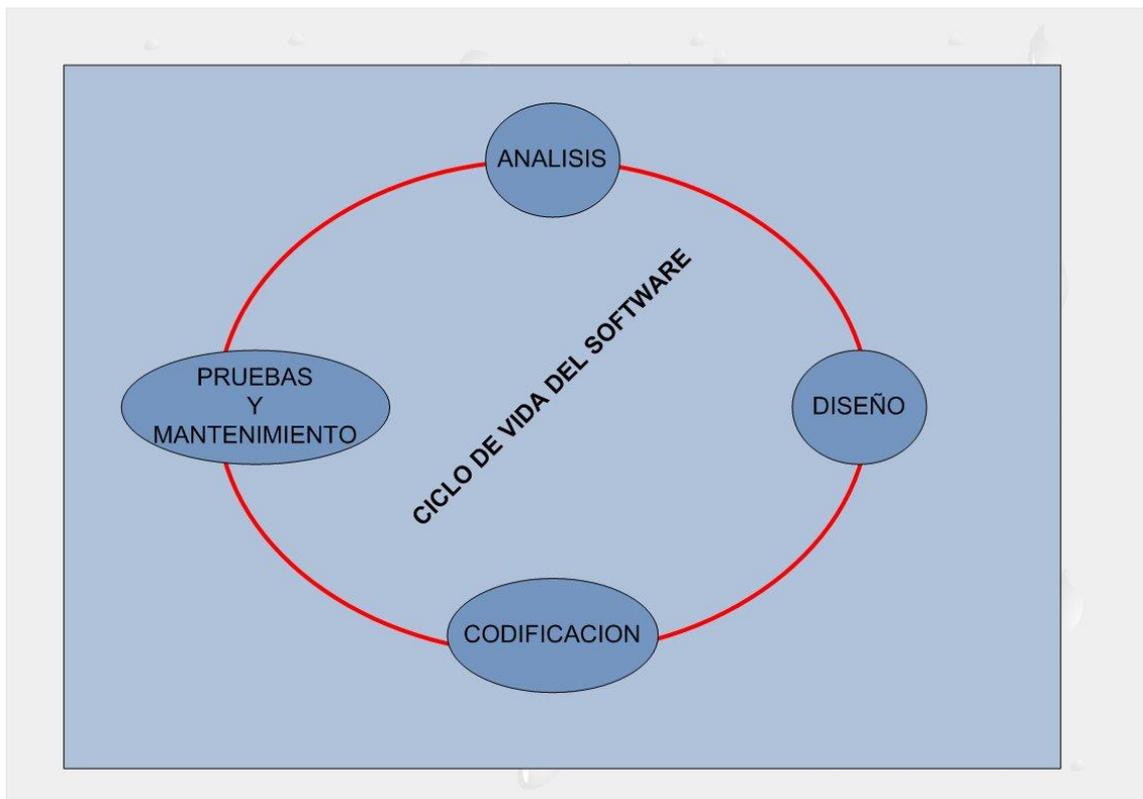


Figura 2.3. Ciclo de Vida del Software

✓ **Análisis de los requisitos del software.**

Lo primero que debemos tomar en cuenta es que el software casi siempre forma parte de un sistema mas grande, como es obvio, siempre forma o va a formar parte de una organización, por lo que el trabajo inicia al recopilar información acerca de la empresa y el medio en el cual el software va a integrarse, por sobre todas las cosas es de vital importancia estar al tanto de los procesos que este va a automatizar. Después el proceso se enfoca a recopilar y reunir los requisitos específicos del software.

Los requerimientos de usuario son el conjunto de resultados que el cliente espera obtener al utilizar el sistema, estos deben mostrar todo lo que el sistema puede hacer además de las restricciones de funcionalidad del mismo.

Existen un sin numero de requerimientos y al no poseer similares características es recomendable organizarlos y clasificalos para tener una visión mas clara de estos.

Por un lado están los requerimientos funcionales que son los que definen el comportamiento del software internamente, como son el manejo de la información, los datos, cálculos y demás detalles técnicos que se involucren en el desarrollo del sistema, por otro lado se encuentran los requerimientos no funcionales que especifican métricas para poder juzgar al software y comportamientos esperados en cuento a su

funcionamiento en forma global como por ejemplo: Precio, Calidad, Disponibilidad, Mantenimiento, Usabilidad, etc.

✓ **Diseño de Software.**

El diseño de software es el paso en donde en base a los requerimientos obtenidos en la primera fase se procede a diagramar y estructurar la arquitectura del sistema a ser realizado. Se consideran algunos puntos como la estructura de datos, que es la definición de los objetos de datos que van a intervenir en el flujo de información que formará parte del software. La arquitectura del software que se centra en definir la estructura de tanto de software como de hardware que se necesitarán para el desarrollo, en este paso se definen cosas tales como la tecnología a ser empleada y los recursos físicos que se emplearán. El diseño de las interfaces de usuario es también una parte fundamental de esta etapa por que es donde se definirá como se mostrará el sistema hacia el usuario así como la manera en que este último podrá interactuar con el sistema.

Por ultimo esta etapa concluye con la especificación procedimental acerca de la codificación misma del software, es aquí en donde se indica como se procederá con la codificación del sistema de tal manera de que el código sea un fiel reflejo de los requerimientos, responda correctamente a la arquitectura, maneje de manera adecuada la información y satisfaga las pautas acerca de las interfaces.

✓ **Codificación de Software**

En este paso es donde se debe traducir los resultados del diseño en lenguaje legible para la máquina. Este es un paso a menudo exagerado en el desarrollo de software ya que se le dedica el mayor porcentaje de tiempo en un proyecto, sin embargo, si el diseño se lleva a cabo de manera detallada y este a su vez fue basado en requerimientos plenamente definidos la generación del código se debería realizar mecánicamente.

A pesar de esto y como es de conocimiento de cualquier persona que haya realizado software no es tan sencillo como la teoría nos lo presenta, la codificación requiere de un alto conocimiento técnico y a menudo se vemos envuelto por situaciones que requieren de investigación.

La codificación del sistema se hace en base a un lenguaje de programación definido previamente en base al diseño en donde se escogerá el mas adecuado acorde a la funcionalidad que se requiera lograr, ejemplos comunes son: C++,Java, C#, Perl, etc.

Es necesario recalcar que así las anteriores fases estén perfectamente bien definidas, no se podrá alcanzar la calidad deseada al menos de que la codificación sea muy bien realizada y basada en estándares y procedimientos propios de la etapa de desarrollo de el código.

✓ Pruebas y Mantenimiento

Después de generado el código se deben realizar las pruebas del sistema, estas se apegan a los procesos lógicos internos de la aplicación asegurando que cada una de las líneas de código genere los resultados esperados y que las entradas y las salidas de información respondan adecuadamente a cada uno de las interacciones del usuario. Durante el desarrollo de estas pruebas es posible que se encuentren errores en la funcionalidad del mismo y deberán ser corregidos adecuadamente asegurando siempre minimizar el impacto en realizarlo.

Una vez finalizadas las pruebas finales de funcionalidad y corregidos los errores, el sistema queda listo para pasar a pruebas UAT¹⁰ en donde se revisan las interfaces y requerimientos menores del sistema, durante el desarrollo de estas pruebas es común encontrar cambios para las interfaces, mensajes que se muestran y sobre todo aspectos de diseño gráfico y usabilidad.

Luego de estas pruebas el software se encuentra listo para pasar a producción, termino utilizado para definir la implementación final del software en el hardware definitivo.

Posteriormente empieza la fase de mantenimiento en donde como era de esperarse aparecerán cambios, ya no producidos por errores en el desarrollo si no por la variación en aspectos del negocio mismo, por lo que muchas de las veces será necesario

¹⁰ User Acceptance Testing

realizar las anteriores etapas de desarrollo para poder implantarlos.

2.4- Programación Orientada a Objetos

La programación orientada a objetos es un paradigma de programación que usa “objetos”, su principal característica es que divide al mundo en dos partes bien definidas: datos y operaciones sobre datos.

Considera a los datos como una unidad estática e inmutable que solo puede ser cambiada a través de sus propias operaciones.

A continuación se definirán algunos de los principales conceptos de programación orientada a objetos y se ejemplificará su significado:

✓ Clase

Define las características abstractas de una cosa, así como su comportamiento, es como decir los planos o la estructura de las propiedades que definen a un objeto de forma global.

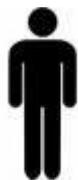
Supongamos la clase que represente a una persona para el padrón electoral de un país, con atributos básicos: como nombre, apellido y cedula.



Atributos
nombre
apellido
cedula

Figura 2.4. Ejemplo de Clase

Ahora bien la clase persona presentará algunas acciones propias que se consideran los métodos de esta clase, los mismos que serán útiles para la interacción de esta clase con el entorno. Por ejemplo para este caso un método que podría implementar esta clase podría ser el que ejecuta la acción del voto.



Atributos
nombre
apellido
cedula
Métodos
Votar()

Figura 2.5. Ejemplo de Método

Ahora después de definir los atributos y métodos de esta clase “Persona” hemos definido abstractamente a una persona con las propiedades y capacidades necesarias para interactuar con el ambiente.

✓ **Objeto**

Es una instancia particular de una clase, es decir, es cuando los atributos de una clase toman valores específicos y definen a un objeto en particular.

En una implementación de un sistema pueden existir tantas instancias de una clase como sea necesario.



Atributos
nombre="Francisco"
apellido="Naranjo"
cedula="1714884770"
Métodos
Votar()

Figura 2.6. Ejemplo de Objeto

Ahora el objeto que representa a la clase "Persona" es único y define a una instancia en particular de esta clase.

✓ **Método**

Esta definido por una habilidad que tiene un objeto, en el ejemplo que se esta usando la persona puede ejercer el voto por lo que posee un método que le permite realizar dicha acción.

✓ **Paso de Mensajes**

El paso de mensajes es una característica que presenta la POO¹¹ en donde las clases pueden comunicarse entre si, es decir, una clase le pasa una mensaje a otra para hacer que esta ejecute uno de sus métodos y espera una respuesta.

✓ **Herencia**

Las subclases o clases heredadas son clases más especializadas, que heredan propiedades y métodos de las clases a las que pertenecen. Por ejemplo en el ejemplo de la clase persona podría existir una clase

¹¹ Programación Orientada a Objetos

heredada llamada “Veedor”, que es otro tipo de persona que interviene en un proceso de elecciones sin embargo para este tipo en particular se necesitan otras propiedades y métodos.

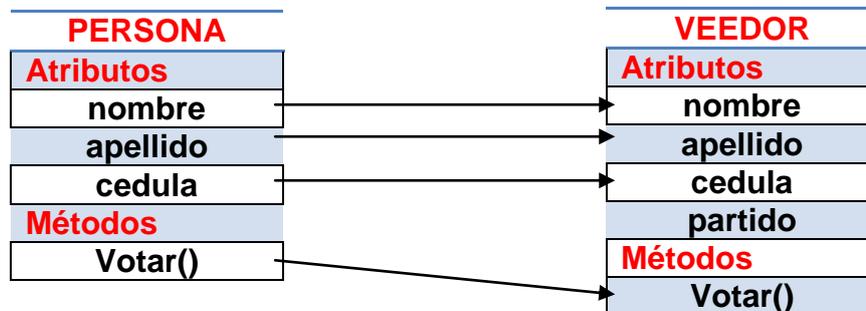


Figura 2.7. Ejemplo de Herencia

En este caso la clase veedor hereda de la clase persona todos sus atributos y métodos pero agrega además atributos propios de la misma, en caso de ser necesario también podría agregar métodos.

✓ Encapsulamiento

Es la forma en que se organizan los atributos y métodos de una clase de tal manera que protegen la implementación de los mismos, previniendo el acceso a los datos por cualquier otro motivo excepto el original. El encapsulamiento por lo tanto garantiza la integridad de los datos que pertenecen al objeto.

El usuario de una clase en particular no necesita saber como los datos de la misma están estructurados por lo que se previene que el usuario modifique los datos directamente, en su lugar debe usar los métodos definidos para realizar operaciones permitidas sobre los datos.

El encapsulamiento define el nivel de acceso a los elementos de una clase, existen tres niveles de acceso:

public: Funciones de otras clases o métodos de la misma tiene total acceso a los datos de la clase. Este es el nivel de protección mas bajo.

protected: El acceso a los datos esta restringido y permite acceso únicamente a clases heredadas.

private: El acceso a los datos se encuentra completamente restringido. Este es el nivel de protección más alto.

✓ **Polimorfismo**

Es la capacidad que tienen los objetos a responder a los parámetros que interactúan con los mismos, existen varios tipo de polimorfismo como son el aplicado a tipos de datos y el aplicado a métodos. El uso del mismo es gran utilidad puesto que nos permite dar a una función o dato especifico gran funcionalidad y así evitarnos la re codificación.

2.5- Lenguaje de marcado de hipertexto (HTML)

HTML ó HyperText Markup Language, Lenguaje De Marcado de Hipertexto es el lenguaje de marcado que predomina en la codificación de páginas web. Es principalmente la manera de describir la posición de elementos de texto en un documento, adicionalmente el documento puede describir el posicionamiento de elementos de media tales como imágenes y elementos para entrada de información.

2.5.1- Historia

La primera publicación existente acerca de HTML fue un documento llamado HTML Tags, fue mencionada por primera vez en la internet por Berners-Lee en los años 1991. El documento contenía muchas de las especificaciones de los tags que se usan en la actualidad.

Sin embargo la primera especificación estándar acerca de los documentos HTML no aparecería sino hasta el año 1995 con la publicación de HTML 2.0.

Posteriormente en el año 2000 HTML se convirtió en un estándar internacional ISO/IEC 15445:2000.

2.5.2- Marcado HTML

El marcado o disposición de los elementos en HTML consiste en varios elementos y sus atributos, además de tipos de dato basados en caracteres, referencias a caracteres y entidades.

A continuación se muestra un ejemplo de código HTML:

```
3
4 <html>
5   <head>
6     <title>Hello HTML</title>
7   </head>
8   <body>
9     <p>Hello World!</p>
10  </body>
11 </html>
12
```

Figura 2.8. Ejemplo de Código HTML

Esta simple combinación de tags son interpretados por el explorador, el mismo que los convierte en una pagina web visible para el usuario, el código anterior produciría una interfaz grafica como la siguiente:



Figura 2.9. Ejemplo de Página HTML

✓ Elementos HTML

Los elementos HTML son aquellos que definen la estructura en un documento HTML, los mismos poseen un grupo de atributos que definen varias de sus características, estas que pueden ser por ejemplo: tamaño, color, comportamiento, etc.

Los elementos HTML están constituidos principalmente de:

- ✓ Un tag inicial que indica el inicio del elemento.
- ✓ Un grupo de atributos que definen las características del elemento.
- ✓ El contenido del elemento.
- ✓ Un tag de finalización.

Tabla 2.2. Estructura de Elemento HTML

INICIO TAG	ATRIBUTO	VALOR	CONTENIDO	FIN TAG
<p>	class	title	Francisco Naranjo	</p>

Existen un sin número de elementos HTML que combinados conforman un documento HTML, los elementos se clasifican según su funcionalidad y se usan a lo largo del documento para lograr la usabilidad que se requiere a la hora de ser interpretado por el explorador.

A continuación se lista algunos de los más importantes:

Tabla 2.3. Principales Elementos HTML

Tag	Descripción
a	Designa el inicio y destino de un hipervínculo de texto.
abbr	Inserta abreviaciones.
applet	Inserta contenido ejecutable.
b	Especifica que el texto irá en negritas.
big	Indica que el texto se mostrará de mayor tamaño.
body	Indica el inicio y el fin de un documento de contenido.
br	Inserta un salto de línea.

center	Centra el contenido.
div	Especifica un contenedor para grupos de elementos HTML.
fieldset	Dibuja un rectángulo que contiene a un grupo de elementos HTML.
font	Especifica el formato de la fuente.
form	Indica una agrupación de elementos que pertenecen a un formulario.
head	Coloca información adicional al documento HTML.
html	Identifica todo el contenido del documento HTML.
i	Especifica que el texto ira en cursiva.
img	Inserta una imagen o video en el documento
input	Crea una variedad de controles para ingreso de información (button,checkbox,file,hidden,text,password,radio).
option	Inserta un combo box en la página.
p	Denota un párrafo.
small	Indica que el texto se mostrará de mayor menor.
span	Indica un contenedor de texto para una línea.
table	Especifica que el contenido se organizara en columnas y filas.
tr	Especifica la fila de una tabla.
td	Especifica una celda de la tabla.
title	Coloca el título del documento
u	Especifica que el texto ira en subrayado.

El uso de estos elementos en conjunto se ejemplifica en el siguiente código:

```

4 <html>
5 <head>
6 <title>Ejemplo de Pagina Html</title>
7 </head>
8 <body>
9 <p>Ingrese sus datos:</p>
10 <table>
11 <tr>
12 <td>Nombre:</td>
13 <td><input type="text" ></td>
14 </tr>
15 <tr>
16 <td>Sexo:</td>
17 <td>
18 <select name="sexo">
19 <option value="m">Masculino</option>
20 <option value="f">Femenino</option>
21 </select></td>
22 </tr>
23 </table>
24 </body>
25 </html>

```

Figura 2.10. Ejemplo de uso conjunto de elementos HTML

El código anterior produciría una interfaz grafica como la siguiente:



Ingrese sus datos:

Nombre:

Sexo:

Figura 2.11. Interfaz Ejemplo de uso conjunto de elementos HTML

2.6- Tecnología JSP

JSP ó Java Server Pages es una tecnología basada en el lenguaje de programación Java, su principal funcionalidad es la de permitir la creación de contenidos web dinámicos ya sean estos HTML ó XML.

Las paginas JSP son compiladas en servlets de java, que funcionan como clases que exponen dos interfaces de interacción con los métodos POST y GET. Las paginas JSP son consideradas como un abstracción de un servlet, las mismas que al ser compiladas se ven traducidas en código mediante las múltiples directrices que presentan para indicar al compilador como generar el servlet resultante.

Una página JSP puede componerse por los siguientes elementos:

- ✓ Contenido estático como HTML.
- ✓ Directrices e inclusiones JSP.
- ✓ Elementos de Acción y Variables.
- ✓ Librerías.

Esencialmente un pagina JSP es una pagina HTML normal con contenido estático pero con la particularidad que incluye código java en su interior. Una ventaja de esta tecnología es justamente el hecho de que aprovecha toda la funcionalidad y robustez del lenguaje Java y la usa para generar paginas web con alto contenido de dinamismo.

No existe ninguna restricción a la hora de utilizar el lenguaje Java en el diseño de paginas JSP, es decir, todas las clases, métodos y herramientas que nos ofrece Java pueden ser usados en el contenido de las paginas.

Para entender el funcionamiento de la tecnología JSP se usará el siguiente ejemplo:

Suponga que necesitamos realizar una página web, que cada vez que sea visitada muestre la fecha actual; hay mucha gente que puede decir que resultados como este se pueden lograr a través del uso de tecnología del lado del cliente como javascript, pero supongamos que esta opción sea deshabilitada o que simplemente el explorador no lo soporte.

La manera más sencilla de lograrlo es utilizando tecnología que interactúe con el servidor y que provea de esta manera contenido dinámico a la página web.

```
<%@page import="java.util.Date"%>
<html>
<head>

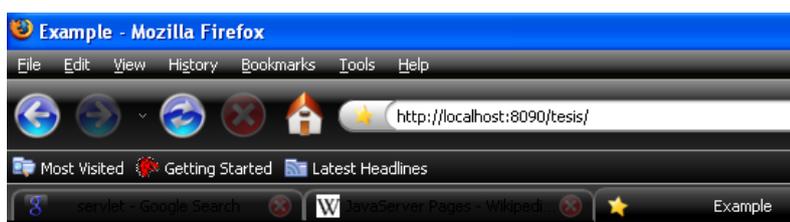
<title>Example</title>
</head>
<body>
<%Date fecha=new Date();%>
La fecha de hoy es:<%=fecha.toString() %>
</body>
</html>
```

En el ejemplo anterior se muestra código HTML simple, el mismo que define como ya se explicó anteriormente la estructura del documento HTML, sin

embargo en esta ocasión se ha agregado contenido dinámico utilizando tecnología JSP.

El contenido JSP en una página se identifica por los siguientes tags: `<%.... %>`, dentro de los cuales podemos incluir el código Java que será procesado por el servidor y devuelto al explorador en el momento en que la página se carga. En el ejemplo anterior se utiliza la clase Date, muy común en el lenguaje de programación Java, para obtener la fecha actual y mostrarla como parte del contenido de la pagina JSP.

El resultado obtenido al desplegar la página en un servidor que sea capaz de procesar contenido dinámico, en este caso JSP, es el siguiente:



La fecha de hoy es: Mon Nov 10 00:25:38 COT 2008

Figura 2.12. Ejemplo de JSP

En los últimos años y como respuesta a la gran dinámica exigida por el mercado de diseño de aplicaciones web la tecnología JSP ha alcanzado altos niveles de desarrollo, esto gracias al gran avance del lenguaje de programación JAVA que con su arquitectura J2EE¹² ha permitido la aparición de nuevas e innovadoras tendencias de desarrollo como son los EJB'S.¹³

¹² Java 2 Enterprise Edition

¹³ Enterprise Java Beans

Esto como es obvio ha dado cabida al desarrollo de nuevos y mas potentes servidores de aplicaciones que son utilizados para el despliegue de estas aplicaciones.

El estado del arte en cuenta al desarrollo de aplicaciones web esta enfocado a la arquitectura orientada a servicios (SOA¹⁴) en donde la reutilización de componentes y los servicios web son las técnicas más utilizadas.

En cuanto al desarrollo de la tecnología JSP, la inclusión en el mercado de los llamados Porlets es una completa revolución. Los Porlets son interfaces de usuario que poseen alta funcionalidad que se despliegan en un portal de aplicaciones, están construidos con tecnología JSP y utilizan APIS especiales que permiten que su contenido sea añadido a páginas de portales web, pero su condición mas importante son las características plug and play que presentan, es decir, pueden ser incluidos o no en un portal y están disponibles en el momento que se desea además de que son altamente configurables y personalizables. Un ejemplo del funcionamiento de esta nueva tecnología es la página de la compañía Google: www.igoogle.com.



Figura 2.13. Arquitectura basada en Portlets

¹⁴ Service Orientated Architecture

2.7- Lenguaje Unificado de Modelado (UML)

UML ó Unified Modeling Language, es un lenguaje estándar de modelamiento para la ingeniería de software. Es un lenguaje gráfico que nos permite visualizar, especificar y construir los diferentes elementos que intervienen en una solución de arquitectura de software.

Una de las principales ventajas es que nos permite utilizar un método estándar para la construcción de software, que puede incluir desde un diseño de componentes o clases hasta las mismas líneas de código de la aplicación.

UML combina las bondades de los conceptos del modelamiento tradicional como las relaciones entre entidades, el flujo de negocio, el modelado de objetos y el modelado de componentes. Puede ser usado a lo largo de todo el ciclo de vida del software así como también en un sin número de tecnologías de desarrollo.

2.7.1- Métodos

UML no es un método por sí mismo, sin embargo, fue diseñado para que trabaje conjuntamente con los principales métodos de desarrollo de software orientado a objetos. A partir de esto y como consecuencia de la evolución de UML surgieron nuevos métodos y algunos de los ya existentes tuvieron que adaptarse para poder tomar ventaja de las nuevas funcionalidades que UML presentó. Uno de los más comunes es el RUP desarrollado por IBM.

2.7.2- Modelamiento

Los diagramas UML representan tres diferentes vistas del diseño de un sistema:

- ✓ **Requerimientos Funcionales**

Se centran en la descripción del sistema en cuanto a su funcionalidad desde el punto de vista del usuario

- ✓ **Arquitectura**

Evalúa las estructuras estáticas de un sistema como son objetos, atributos, operaciones y relaciones.

- ✓ **Comportamiento Dinámico**

Se enfoca en el comportamiento dinámico de un sistema, analiza las colaboraciones entre los objetos y los cambios a las estructuras internas de los objetos.

2.7.3- Diagramas

Actualmente UML en su versión 2.0 presenta 13 diagramas que se dividen en tres categorías principales:

- ✓ Seis se utilizan para representar la estructura estática de la aplicación.
- ✓ Tres se enfocan en los aspectos generales de comportamiento.
- ✓ Cuatro representan las interacciones entre los elementos de la aplicación.

2.7.3.1- Diagramas de Estructura

✓ Diagrama de Clases

Describe la estructura de un sistema mediante la descripción de las clases que se usan en el sistema, sus atributos y las relaciones entre estas.

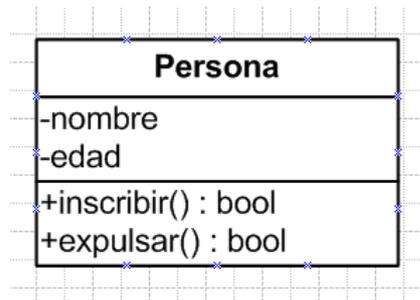


Figura 2.14. Diagrama de Estructura

✓ Diagrama de Componentes

Describe como un sistema de software se encuentra dividido en módulos o componentes y las interacciones entre los mismos.

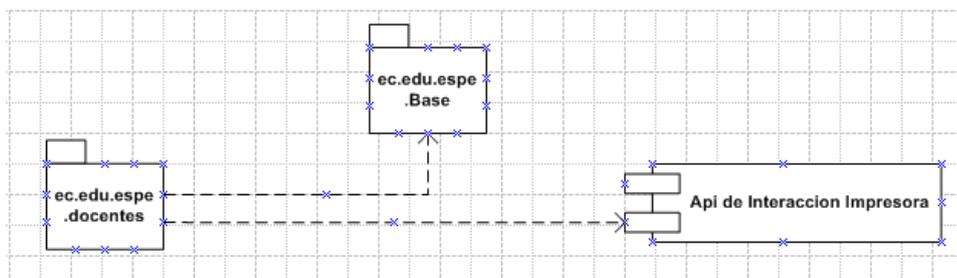


Figura 2.15. Diagrama de Componentes

✓ Diagrama Detallado de Estructura

Muestra la estructura interna de una clase y todas las posibles interacciones que esta puede realizar.

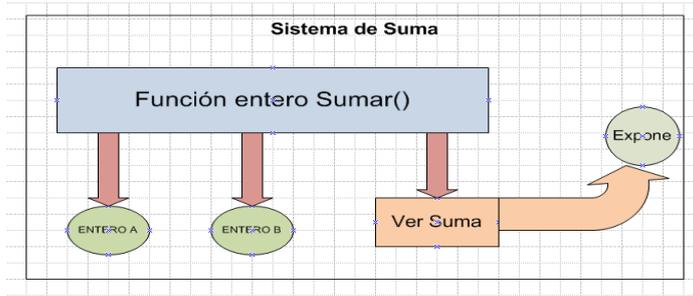


Figura 2.16. Diagrama Detallado de Estructura

✓ Diagrama de Despliegue

Se usa para describir los componentes de hardware que van a ser utilizados para la implementación del software, es fundamental en este describir como el software y el hardware van a estar relacionados.

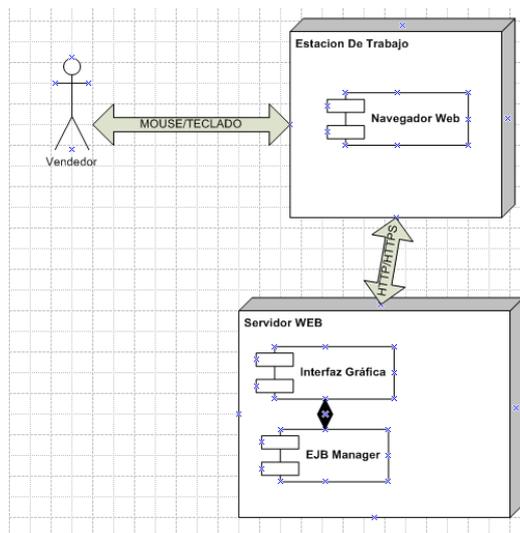


Figura 2.17. Diagrama de Despliegue

✓ Diagrama de Objetos

Muestra el estado de una estructura de datos en un momento específico de ejecución de la aplicación.

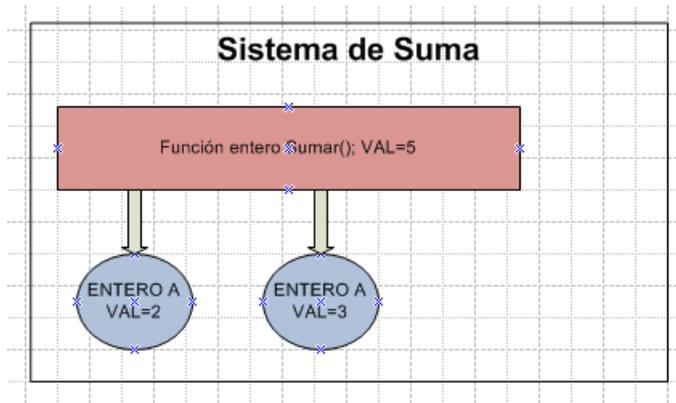


Figura 2.18. Diagrama de Objetos

✓ Diagrama de Paquetes

Muestra la división lógica de una aplicación y las interacciones entre las partes.

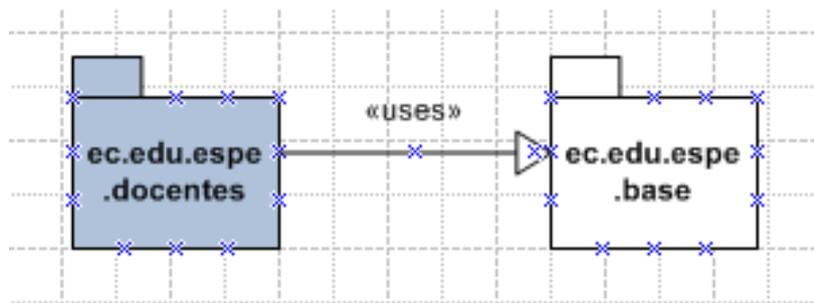


Figura 2.19. Diagrama de Paquetes

2.7.3.2- Diagramas de Comportamiento

✓ Diagrama de Actividad

Representa el flujo del negocio de manera secuencial utilizando los objetos que intervienen en el mismo.

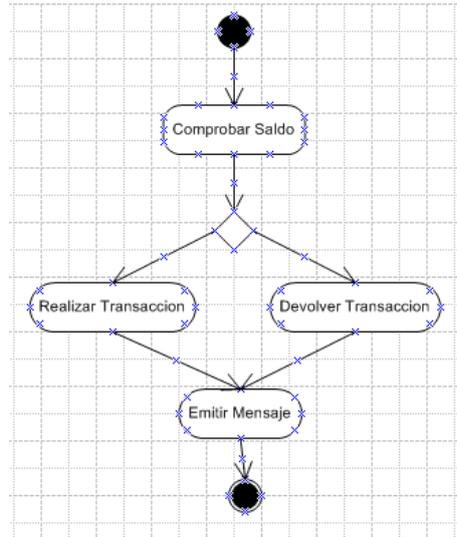


Figura 2.20. Diagrama de Actividad

✓ Diagrama de Estado

Usado para describir un sistema o un modulo específico del mismo, en este se centra principalmente en el proceso de negocio.

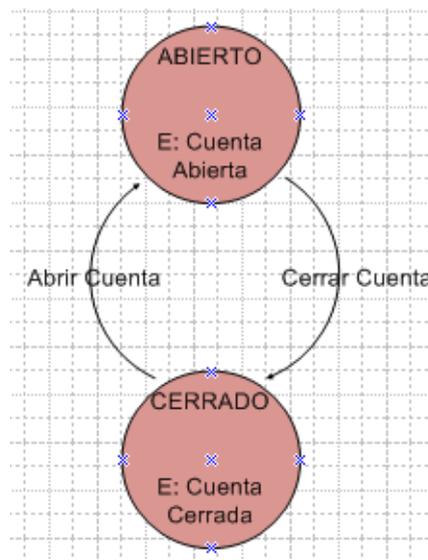


Figura 2.21. Diagrama de Estado

✓ Diagrama de Casos de Uso

Es usado para describir la funcionalidad del sistema en términos de sus actores, se entiende por actor a toda persona o sistema que interactúa con la aplicación descrita.

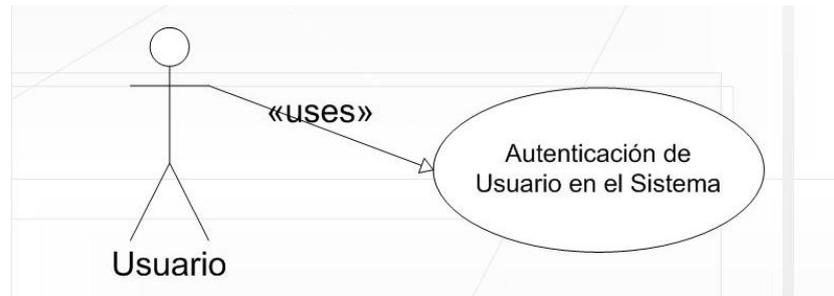


Figura 2.22. Diagrama de Caso de Uso

2.7.3.3- Diagramas de Interacción

✓ Diagrama de Comunicación

Muestra al sistema en términos de los mensajes que se intercambian entre los objetos del mismo.



Figura 2.23. Diagrama de Comunicación

✓ Diagrama de Modelo de Interacción

Es un tipo de diagrama de actividad en donde los nodos representan a diagramas de interacción.

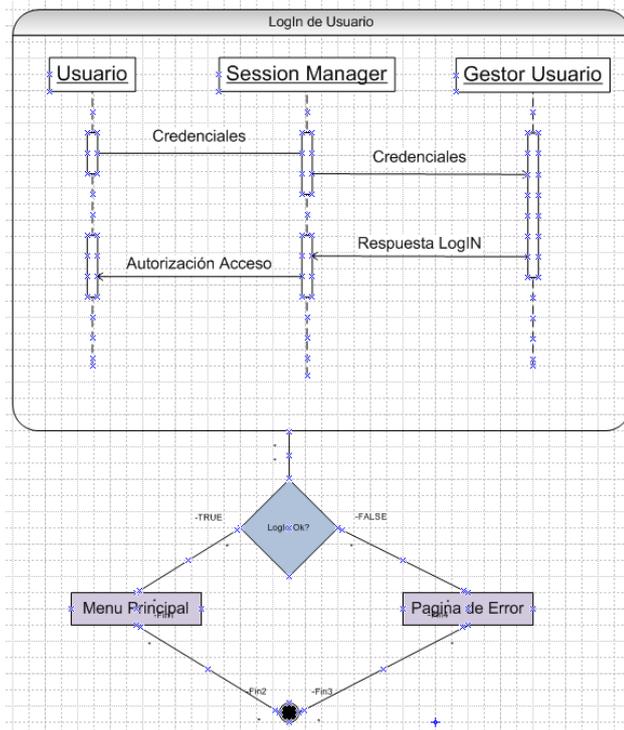


Figura 2.24. Diagrama de Interacción

✓ Diagrama de Secuencia

Muestra la relación entre objetos y flujo de negocio en base a mensajes secuenciales enviados entre los mismos.

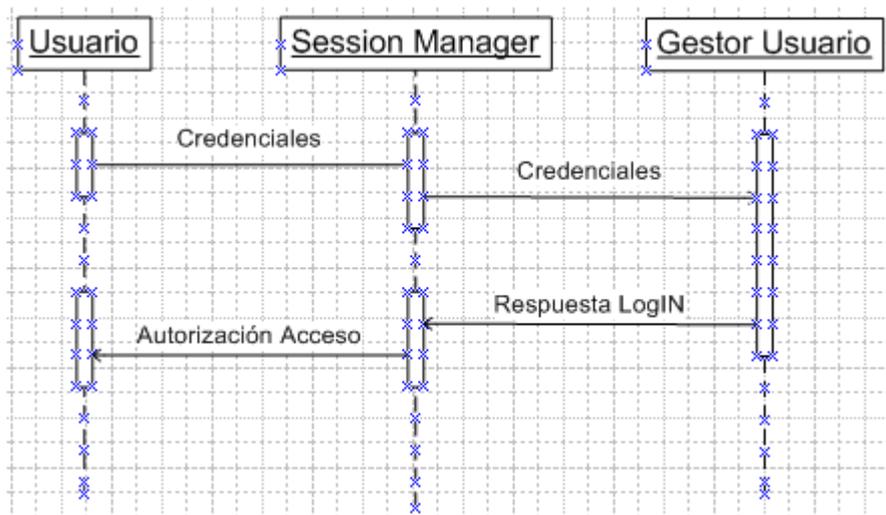


Figura 2.25. Diagrama de Secuencia

✓ Diagrama de Tiempo

Es un tipo de diagrama en donde se representa principalmente el tiempo que se requiere para cumplir con determinada operación o interacción entre elementos.

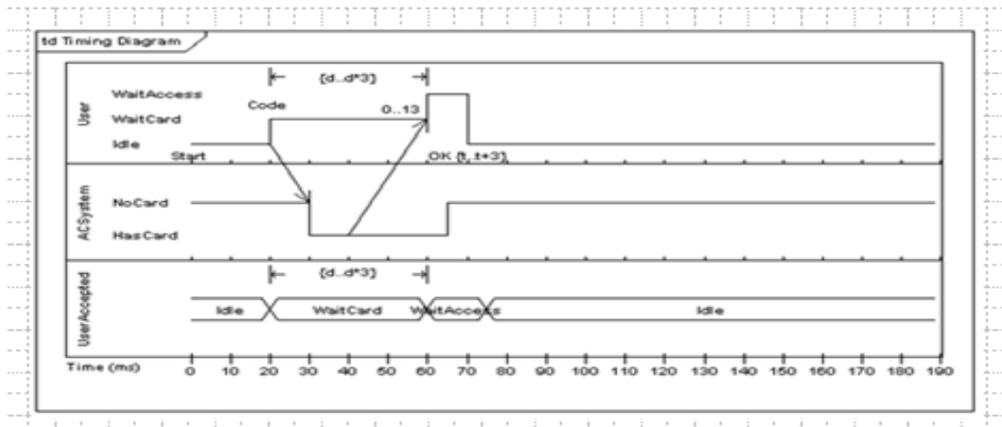


Figura 2.26. Diagrama de Tiempo¹⁵

¹⁵ http://www.sparxsystems.com.ar/resources/tutorial/uml2_timingdiagram.htm

CAPITULO 3

Análisis de Requerimientos

El propósito de este capítulo es el de proporcionar una descripción de la aplicación web a ser realizada. Explicará el propósito y funcionalidades del sistema, que es lo que sistema realizará y las directrices bajo las cuales debe operar y como debe responder a estímulos externos que interactúan con el mismo. Servirá como base tanto a los usuarios finales como al desarrollador para tener una idea más clara del software y sus funcionalidades.

3.1- Ámbito del proyecto

Este sistema de software será un sistema orientado a la Web para el manejo de recursos humanos docentes de la Escuela Politécnica del Ejército. El sistema estará diseñado para maximizar la productividad proveyendo herramientas que asistirán en la automatización de los procesos de control del personal docente de la institución que de otra manera se realizarían manualmente o mediante software no especializado.

Mediante la optimización de estos procesos el sistema cumplirá con los requerimientos del usuario haciendo del proceso un procedimiento fácil de entender y utilizar.

Específicamente el sistema estará diseñado para permitir el control de asistencia de los señores docentes por parte de los coordinadores de carrera y directores de departamento, además de proporcionar herramientas de consulta tanto como para un administrador global del proceso como para los docentes. El sistema facilitará la comunicación entre sus actores haciendo del proceso de control de asistencia un proceso más participativo y coordinado entre los diferentes entes que participan en su ejecución.

3.2- Descripción General

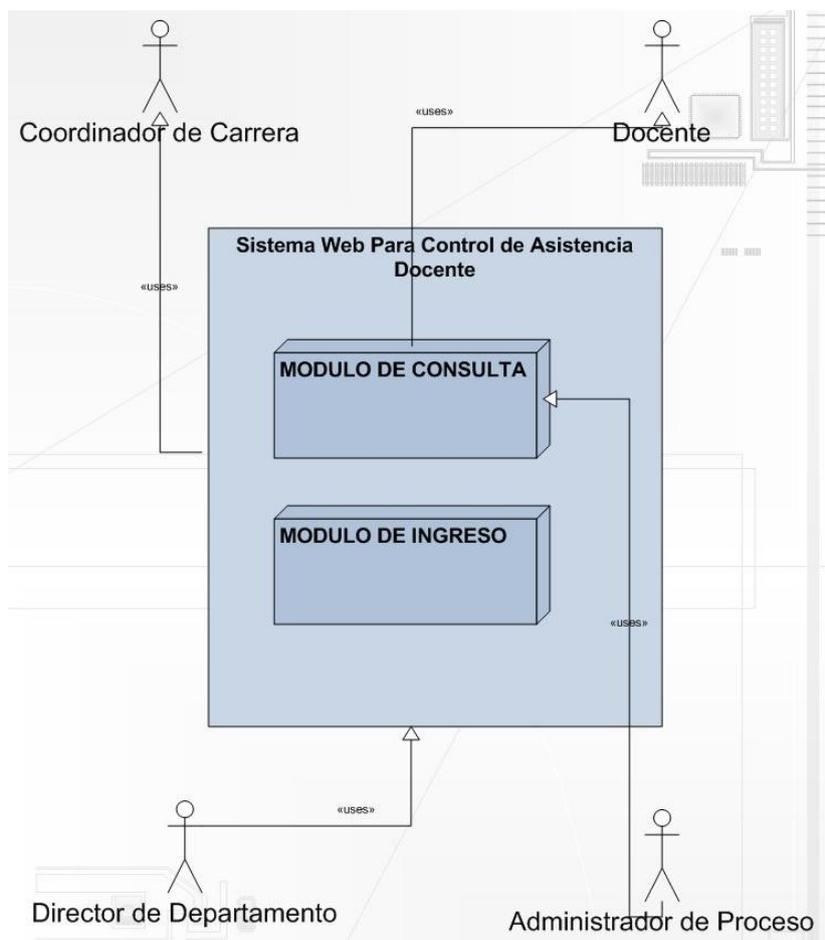


Figura 3.1. Diseño General del Sistema

El sistema de control docente tiene cuatro actores activos. El coordinador de carrera, el Director de Departamento, el Administrador del proceso de control de asistencia y el docente. Todos los actores acceden al sistema mediante el explorador web y utilizando una interfaz de autenticación que válida la identidad del usuario mediante el uso de su número de identificación y una contraseña.

Una vez dentro del sistema cada uno de los usuarios accederá a las diferentes opciones dependiendo del rol que estos ejecuten.

3.3- Requerimientos Funcionales

Los requerimientos funcionales fueron obtenidos en base a la entrevista directa con los usuarios que forman parte activa del sistema, en estas entrevistas se recopiló la información necesaria para poder determinar los diferentes casos de uso en base a las funcionalidades que debe tener el software.

✓ **Descripción del Proceso**

El control de asistencia de docentes es el proceso mediante el cual se puede evidenciar las horas de clase dictadas, las horas de trabajo administrativo, las horas de trabajo bajo comisión y cualquier otra información que nos permita obtener un reporte acerca de la actividad laboral de un docente en un periodo de tiempo determinado como son horas extra, horas adicionales, permisos, faltas, etc. Este periodo de tiempo comprende la actividad laboral desde el día veinte de un mes hasta el día diecinueve del mes subsiguiente. Durante este periodo de tiempo se registran todas las variables descritas anteriormente y se genera un reporte, el mismo que puede ser accedido tanto por autoridades como los coordinadores de carrera, directores de departamento y administradores de este proceso con el fin de controlar y gestionar la actividad laboral de un docente, como por lo docentes para revisar su actividad laboral.

Dentro del ámbito de trabajo de la institución se han identificado tres tipos de docentes cada uno con características particulares en cuanto a la forma en que deben ser procesados por el sistema informático.

En primer lugar se encuentra el docente hora clase, el mismo que dicta materias en horarios predefinidos pero no debe cumplir con labores administrativas, las particularidades acerca de su actividad laboral son: Faltas Justificadas, Faltas Injustificadas, Atrasos, Comisiones y Permisos.

Por otro lado esta el docente tiempo completo, el que además de dictar materias en horarios predefinidos debe cumplir con labores administrativas durante las horas de la jornada laboral diaria que no se encuentre dictando clases hasta completar las ocho horas diarias, las particularidades acerca de su actividad laboral son: Faltas Justificadas, Faltas Injustificadas, Atrasos, Comisiones y Permisos, las mismas variables se repiten para el caso de las horas administrativas que labora.

Además esta el docente dedicación exclusiva el mismo que presenta las mismas particularidades del docente tiempo completo, con excepción de que este debe laborar un total de doce horas en su jornada laboral diaria.

A continuación se detallarán las diferentes funcionalidades a las que cada uno de los actores en este sistema de software podrá acceder.

✓ **Administrador del Proceso**

El administrador del proceso es el rol que permitirá al usuario tener acceso a toda la información acerca del control de asistencia docente de la institución, este podrá revisar información histórica acerca de reportes generados para cualquier docente. Este usuario no podrá modificar ni alterar los reportes de asistencia por lo que su interacción con el sistema es estrictamente de lectura y consulta.

La información de los reportes generados es de vital importancia para este rol ya que permitirá al mismo poder vincular esta información con el sistema de roles de pago de la institución.

Dependiendo del tipo de docente del que se requiera la información los reportes varían, en el caso del docente hora clase la información es la siguiente: Total Horas Dictadas, Total Faltas Justificadas, Total Faltas Injustificadas, Total Atrasos, Total Comisiones y Total Permisos. En el caso de los otros dos tipos de docentes la información es similar pero incluye lo referente a las particularidades en las horas administrativas, horas extra y horas adicionales.

Para cualquier tipo de docente el reporte presenta información detallada clasificada diariamente acerca de la actividad laboral de los docentes, además de tener la opción de visualizar un reporte resumido.

✓ **Coordinador de Carrera**

El coordinar de carrera es el rol que permite al usuario ingresar las novedades acerca de las horas de clase dictadas por un docente en una carrera determinada. Este estará capacitado a registrar características acerca de una hora de clase en particular y podrá generar los reportes para la carrera en la cual es coordinador. Las características que este puede registrar son Faltas Justificadas, Faltas Injustificadas, Atrasos, Comisiones y Permisos, todas estas con excepción de los atrasos tendrán un valor igual al total de horas dictadas para la asignatura seleccionada. Este rol también tiene la opción de revisar los reportes generados de todos los docentes que dictan clases en esa carrera.

✓ **Director de Departamento**

El director de departamento es el rol que permite al usuario ingresar las características de las horas administrativas de un determinado docente perteneciente al departamento así como también las horas extra o adicionales. Las características de las horas administrativas son: Faltas Justificadas, Faltas Injustificadas, Atrasos, Comisiones y Permisos. Como en los otros casos también tiene la opción de revisar los reportes generados de todos los docentes de ese departamento.

✓ **Docente**

El docente es el rol que permite al usuario visualizar el reporte personal que se ha generado para su actividad laboral en un periodo de control de asistencia determinado, este tendrá la opción de tener información histórica acerca de sus reportes.

En esta sección se especificarán los casos de uso para cada uno de los usuarios del sistema.

3.3.1- CU001-Autenticación de Usuario en el sistema

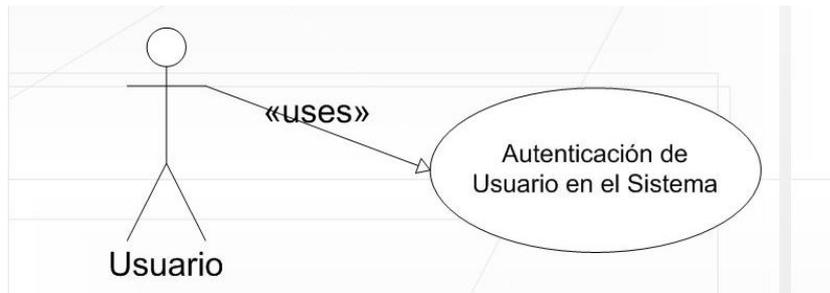


Figura 3.2. Autenticación de Usuario en el sistema

Descripción

El usuario se identifica en el sistema para que este le asigne los roles designados y se muestra la pantalla inicial.

Descripción de actividades

- ✓ El usuario ingresa su identificación y contraseña, las mismas que son verificadas por el sistema para a continuación ser direccionados a la pagina principal.

Tabla 3.1. Autenticación de Usuario en el sistema

Caso de Uso	Autenticación de Usuario en el sistema
Inicio	El usuario ingresa su identificación y las credenciales son analizadas por el sistema
Precondición	El usuario debe encontrarse en la pantalla inicial y debe ingresar su identificación y contraseña
Camino Básico	1. El usuario ingresa su identificación y contraseña. 2. El sistema recoge las credenciales y

	<p>comprueba su veracidad basándose en la base de datos de usuarios del sistema.</p> <ol style="list-style-type: none"> 3. El sistema recupera los roles que han sido asignados para estas credenciales. 4. Con esta información se construye la pantalla inicial y el menú de roles para el usuario. 5. El sistema muestra el menú principal de la aplicación.
Camino Alternativo	<ul style="list-style-type: none"> • En el paso 2, si las credenciales no están completas o presentan algún error el sistema muestra un error y no se continúa con el flujo. • En el paso 2, si las credenciales no pertenecen a ningún usuario registrado. <ol style="list-style-type: none"> 1. El sistema muestra una pantalla donde indica que ocurrió un error en la autenticación del usuario.

3.3.2- CU002-Cerrar Sesión

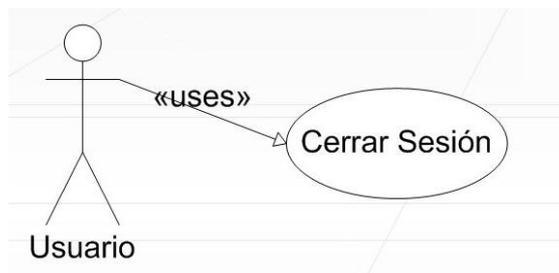


Figura 3.3. Cerrar Sesión

Descripción

El usuario termina con su trabajo en el sistema y cierra sesión.

Descripción de actividades

- ✓ El usuario pulsa sobre el link cerrar sesión y finaliza su trabajo

Tabla 3.2. Cerrar Sesión

Caso de Uso	Cerrar Sesión
Inicio	El usuario pulsa sobre el link cerrar sesión
Precondición	El usuario debe encontrarse autenticado en el sistema.

Camino Básico	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el link para finalizar con su sesión actual. 2. El sistema recibe la petición y elimina todas las variables de sesión existentes. 3. El sistema cierra la ventana actual en el explorador web.
----------------------	--

3.3.3- CU003-Ir a Web Principal

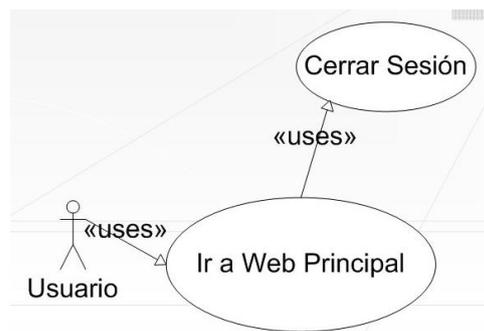


Figura 3.4. Ir a Web Principal

Descripción

El usuario abandona el sistema y es re direccionado al sitio web principal de la Escuela Politécnica del Ejército.

Descripción de actividades

- ✓ El usuario pulsa sobre el link Sitio ESPE y es direccionado hacia la página principal de la Escuela Politécnica del Ejército.

Tabla 3.3. Ir a Web Principal

Caso de Uso	Ir a Web Principal
Inicio	El usuario pulsa sobre el link Sitio ESPE
Precondición	El usuario debe encontrarse autenticado en el sistema.
Camino Básico	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el link para dirigirse al sitio web principal de la ESPE. 2. El sistema recibe la petición y elimina todas las variables de sesión existentes. 3. El sistema re direcciona al explorador hacia el portal principal de la ESPE.

3.3.4- CU004-Elegir Rol

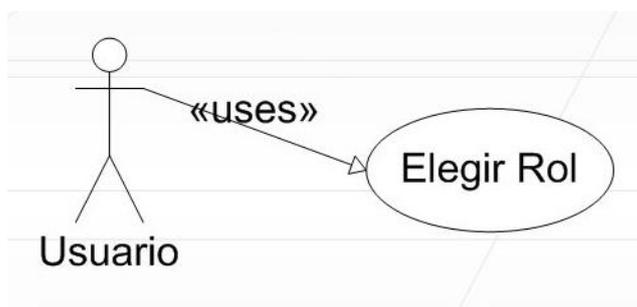


Figura 3.5. Elegir Rol

Descripción

El usuario elige un rol asignado y el sistema muestra los diferentes servicios que este rol presenta.

Descripción de actividades

- ✓ El usuario pulsa sobre el nombre del rol al que desea acceder y el sistema le muestra todos los servicios que se encuentran disponibles para dicho rol.

Tabla 3.4. Elegir Rol

Caso de Uso	Elegir Rol
Inicio	El usuario pulsa sobre el link del rol que desea, escogiéndolo de la lista de roles a los cuales pertenece.
Precondición	El usuario debe encontrarse autenticado en el sistema y debe tener asignado por lo menos a un rol.
Camino Básico	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el link del rol al cual desea acceder. 2. El sistema recibe la petición y obtiene los servicios que se encuentran disponibles para dicho rol. 3. El sistema presenta un menú con los servicios disponibles para que el usuario pueda utilizarlos.
Camino Alternativo	<ul style="list-style-type: none"> • En el paso 2, si el sistema no encuentra servicios disponibles para el rol muestra un mensaje indicando esta condición.

3.3.5- CU005-Ver Reporte Personal

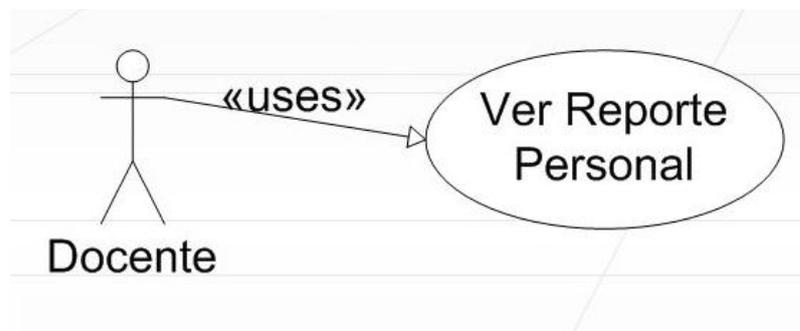


Figura 3.6. Ver Reporte Personal

Descripción

El docente de la Escuela Politécnica del Ejército consulta la información acerca del reporte de asistencia a clases para las materias que dicta en el periodo actual.

Descripción de actividades

- ✓ El docente accede al perfil designado para el mismo y utiliza el servicio que le permite conocer el reporte de asistencia a clases generado para las materias que dicta en la institución.

Tabla 3.5. Ver Reporte Personal

Caso de Uso	Ver Reporte Personal
Inicio	El usuario pulsa sobre el link del servicio Ver Reporte Personal.
Precondición	El usuario debe encontrarse autenticado en el sistema y debe tener asignado el rol de docente.
Camino Básico	<ol style="list-style-type: none">1. El usuario pulsa sobre el link del servicio Ver Reporte Personal.2. El sistema muestra una lista de todos los periodos de asistencia disponibles.3. El usuario escoge el periodo a su elección.4. El sistema muestra las modalidades en donde existen reportes disponibles.5. El usuario escoge la modalidad a su elección.

	6. El sistema muestra el reporte general de asistencia para las asignaturas que el docente se encuentra dictando. 7. Adicionalmente se tiene la opción de visualizar un reporte resumido acerca de la asistencia del docente.
Camino Alternativo	<ul style="list-style-type: none"> En el paso 4, si el sistema no encuentra reportes disponibles muestra un mensaje indicando dicha situación.

3.3.6- CU006-Ver Reporte Docente (Administrador)



Figura 3.7. Ver Reporte Docente (Administrador)

Descripción

El administrador del proceso de control de asistencia docente de la Escuela Politécnica del Ejército consulta la información de cualquier docente y visualiza el reporte de asistencia a clases para las materias que dicta en el periodo actual.

Descripción de actividades

- ✓ El administrador del proceso accede al perfil designado para el mismo y utiliza el servicio que le permite conocer el reporte de asistencia para cualquier docente de la institución.

Tabla 3.6. Ver Reporte Docente (Administrador)

Caso de Uso	Ver Reporte Docente (Administrador)
Inicio	El usuario pulsa sobre el link del servicio Ver Reporte Docente.
Precondición	El usuario debe encontrarse autenticado en el sistema y debe tener asignado el rol

	de administrador.
Camino Básico	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el link del servicio Ver Reporte Docente. 2. El sistema muestra una lista de todos los periodos de asistencia disponibles. 3. El usuario escoge el periodo a su elección. 4. El sistema muestra una lista de todos los docentes disponibles. 5. El usuario escoge el docente de su elección. 6. El sistema muestra las modalidades en donde existen reportes disponibles. 7. El usuario escoge la modalidad a su elección. 8. El sistema muestra el reporte general de asistencia para las asignaturas que el docente se encuentra dictando. 9. Adicionalmente se tiene la opción de visualizar un reporte resumido acerca de la asistencia del docente.
Camino Alternativo	<ul style="list-style-type: none"> • En el paso 6, si el sistema no encuentra reportes disponibles muestra un mensaje indicando dicha situación.

3.3.7- CU007-Ver Reporte Docente (Coordinador)

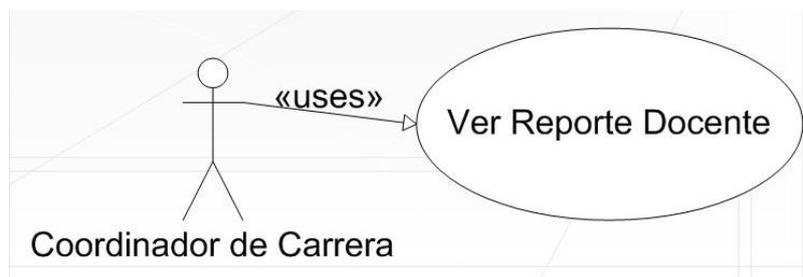


Figura 3.8. Ver Reporte Docente (Coordinador)

Descripción

El coordinador de carrera consulta la información de cualquier docente perteneciente a la carrera de la cual es coordinador y visualiza el reporte de asistencia a clases para las materias que dicta en el periodo actual.

Descripción de actividades

- ✓ El coordinador de carrera accede al perfil designado para el mismo y utiliza el servicio que le permite conocer el reporte de asistencia para cualquier docente que dicte clases en la carrera de la cual es coordinador.

Tabla 3.7. Ver Reporte Docente (Coordinador)

Caso de Uso	Ver Reporte Docente (Coordinador)
Inicio	El usuario pulsa sobre el link del servicio Ver Reporte Docente.
Precondición	El usuario debe encontrarse autenticado en el sistema y debe tener asignado el rol de coordinador de carrera.
Camino Básico	<ol style="list-style-type: none">1. El usuario pulsa sobre el link del servicio Ver Reporte Docente.2. El sistema le muestra una lista de todas las carreras de las cuales ha sido designado coordinador.3. El usuario elige la carrera a su elección.4. El sistema muestra una lista de todos los docentes que dictan clase en la carrera seleccionada.5. El usuario escoge el docente de su elección.6. El sistema muestra el reporte general de asistencia para las asignaturas que el docente se encuentra dictando en esa carrera y en el periodo de control de asistencia actual.7. Adicionalmente se tiene la opción de visualizar un reporte resumido acerca de la asistencia del docente.
Camino Alternativo	<ul style="list-style-type: none">• En el paso 5, si el sistema no encuentra reportes disponibles muestra un mensaje indicando dicha situación.

3.3.8- CU008- Ver Reporte Docente (Director Departamento)

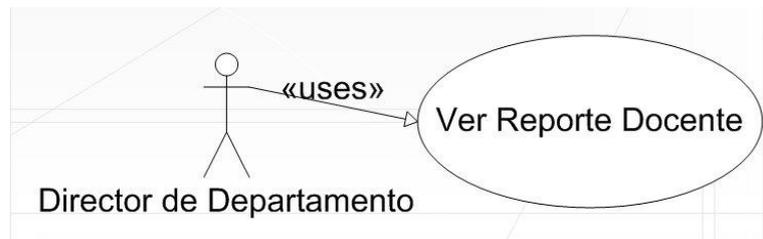


Figura 3.9. Ver Reporte Docente (Director Departamento)

Descripción

El director de departamento consulta la información de cualquier docente perteneciente al departamento del cual es director y visualiza el reporte de asistencia a clases para las materias que dicta en el periodo actual.

Descripción de actividades

- ✓ El director de departamento accede al perfil designado para el mismo y utiliza el servicio que le permite conocer el reporte de asistencia para cualquier docente que dicte clases en el departamento del cual es director.

Tabla 3.8. Ver Reporte Docente (Director Departamento)

Caso de Uso	Ver Reporte Docente (Director Departamento)
Inicio	El usuario pulsa sobre el link del servicio Ver Reporte Docente.
Precondición	El usuario debe encontrarse autenticado en el sistema y debe tener asignado el rol de director de departamento.
Camino Básico	<ol style="list-style-type: none">1. El usuario pulsa sobre el link del servicio Ver Reporte Docente.2. El sistema le muestra una lista de todas las carreras de las cuales ha sido designado director.3. El usuario elige la carrera a su elección.4. El sistema muestra una lista de todos los docentes que pertenecen al departamento seleccionado.

	<ol style="list-style-type: none"> 5. El usuario escoge el docente de su elección. 6. El sistema muestra el reporte general de asistencia para las asignaturas que el docente se encuentra dictando en el periodo de control de asistencia actual. 7. Adicionalmente se tiene la opción de visualizar un reporte resumido acerca de la asistencia del docente.
Camino Alternativo	<ul style="list-style-type: none"> • En el paso 5, si el sistema no encuentra reportes disponibles muestra un mensaje indicando dicha situación.

3.3.9- CU009-Ingresar Reporte Docente (Coordinador)

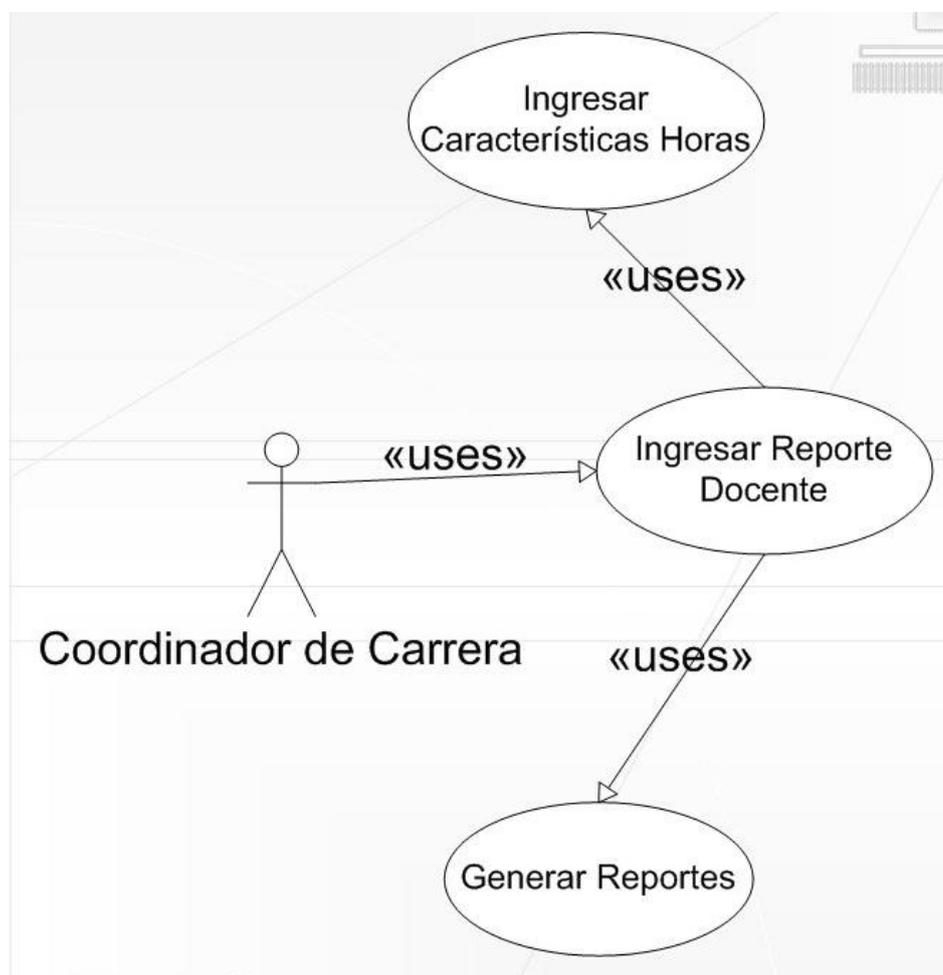


Figura 3.10. Ingresar Reporte Docente (Coordinador)

Descripción

Este es un caso de uso compuesto en donde el coordinador de carrera ingresa un reporte para un docente, dentro del mismo tiene la opción ya sea de generar el reporte para la carrera de la cual es coordinador como de ingresar las características de una hora de clase dictada por un docente en particular.

Descripción de actividades

✓ El coordinador de carrera accede al perfil designado para el mismo y utiliza el servicio que le permite ya sea generar el reporte para cada uno de los docentes de la carrera o ingresar características en una hora de clase específica. Al ser un caso de uso compuesto se describirán todas sus variantes.

Tabla 3.9. Ingresar Reporte Docente (Coordinador)

Caso de Uso	Ingresar Reporte Docente (Coordinador)
Inicio	El usuario pulsa sobre el link del servicio Ingresar Reporte Docente.
Precondición	El usuario debe encontrarse autenticado en el sistema y debe tener asignado el rol de coordinador.
Camino Básico	<ol style="list-style-type: none">1. El usuario pulsa sobre el link del servicio Ver Reporte Docente.2. El sistema le muestra una lista de todas las carreras de las cuales ha sido designado coordinador.3. El usuario elige la carrera a su elección.4. El sistema muestra un menú en donde le permite al usuario generar reportes o ingresar características.5. Si elige generar reporte va al caso de uso A.6. Si elige ingresar características va al caso de uso B.

Tabla 3.10. Generar Reporte de Docentes

Caso de Uso	A – Generar Reporte de Docentes
Inicio	El usuario pulsa sobre el link del servicio Generar Reporte de Docentes
Precondición	El usuario debe encontrarse autenticado en el sistema y debe tener asignado el rol de coordinador.
Camino Básico	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el link del servicio Generar Reporte de Docentes. 2. El sistema verifica que el periodo actual de ingreso de reportes este activo. 3. El sistema procesa la petición y genera el reporte para todos los docentes de esa carrera, al final muestra una lista a manera de resumen del reporte generado.
Camino Alternativo	<ul style="list-style-type: none"> • En el paso 2, si el sistema no encuentra periodos activos muestra un mensaje indicando dicha situación.

Tabla 3.11. Ingresar Características Horas Clase

Caso de Uso	B – Ingresar Características Horas Clase
Inicio	El usuario pulsa sobre el link del servicio Ingresar Características Horas Clase
Precondición	El usuario debe encontrarse autenticado en el sistema y debe tener asignado el rol de coordinador.
Camino Básico	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el link del servicio Ingresar Características Horas Clase. 2. El sistema muestra todos los docentes que dictan clases en esa carrera. 3. El usuario escoge el docente de su elección. 4. El sistema muestra todas las asignaturas que el docente seleccionado dicta actualmente. 5. El usuario escoge la asignatura deseada. 6. El sistema presenta todos los días en donde la asignatura debe ser dictada dentro del periodo de control actual, se debe implementar un método para conocer si el día presenta novedades. 7. El usuario escoge el día sobre el cual quiere ingresar novedades. 8. El sistema presenta una interfaz en donde se podrá escoger la características a ingresar y se

	<p>colocarán valores y observaciones de las mismas.</p> <p>9. El usuario ingresa las características y guarda la información.</p> <p>10. El sistema realiza la operación y muestra un mensaje de éxito.</p>
Camino Alternativo	<ul style="list-style-type: none"> • En el paso 5, si el sistema no encuentra reportes generados muestra un mensaje indicando dicha situación y no continua. • En el paso 9, si las características o valores ingresados son incorrectas o incompletos el sistema muestra un mensaje y no continua hasta que la situación sea corregida. • En el paso 10, en el caso de que existe algún error al momento de guardar la información el sistema muestra un error indicando el fallo.

3.3.10- CU010- Ingresar Reporte Docente (Director Departamento)

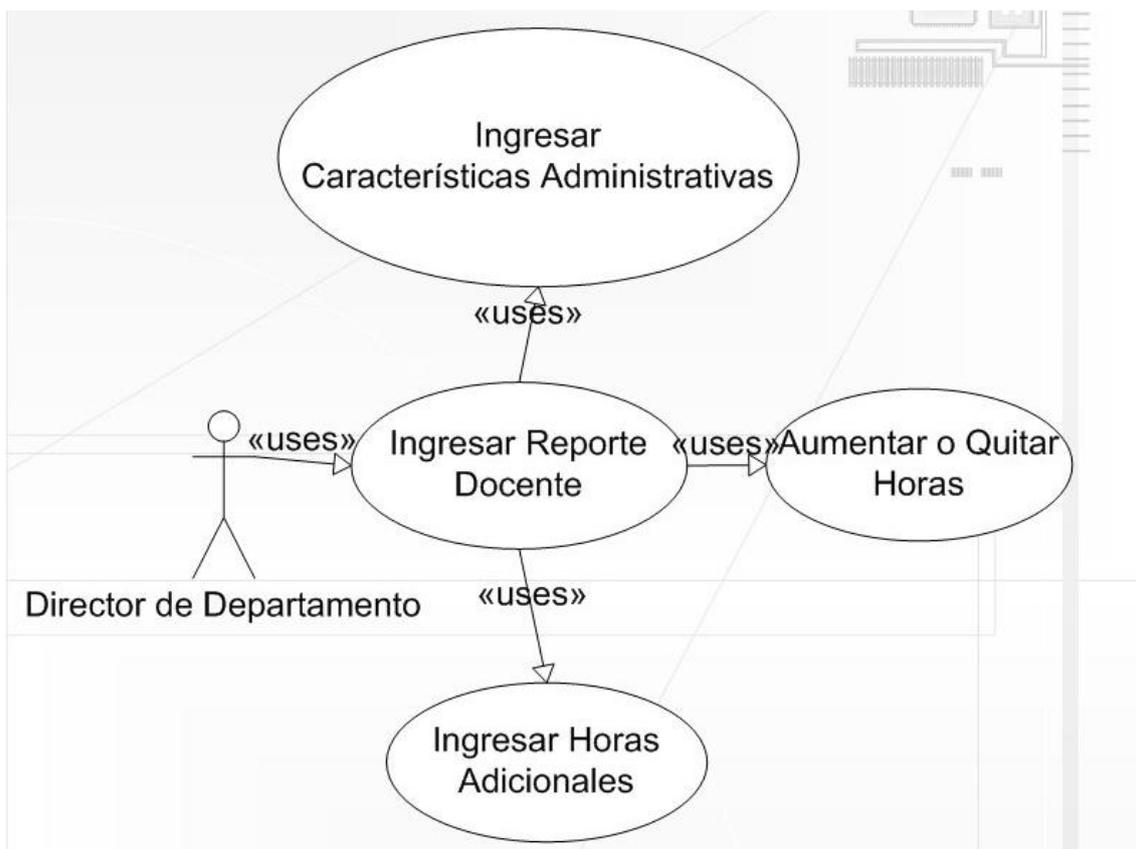


Figura 3.11. Ingresar Reporte Docente (Director Departamento)

Descripción

Este es un caso de uso compuesto en donde el coordinador de carrera ingresa un reporte para un docente, dentro del mismo tiene las siguientes opciones: ingresar características administrativas para un docente en particular, marcar horas como adicionales o extras y aumentar o quitar un número determinado de horas del historial laboral de un docente en el periodo de control de asistencia actual.

Descripción de actividades

✓ El coordinador de carrera accede al perfil designado para el mismo y utiliza el servicio que le permite ya sea generar el reporte para cada uno de los docentes de la carrera o ingresar características en una hora de clase específica. Al ser un caso de uso compuesto se describirán todas sus variantes.

Tabla 3.12. Ingresar Reporte Docente (Director Departamento)

Caso de Uso	Ingresar Reporte Docente (Director Departamento)
Inicio	El usuario pulsa sobre el link del servicio Ingresar Reporte Docente.
Precondición	El usuario debe encontrarse autenticado en el sistema y debe tener asignado el rol de director de departamento.
Camino Básico	<ol style="list-style-type: none">1. El usuario pulsa sobre el link del servicio Ver Reporte Docente.2. El sistema le muestra una lista de todas las carreras de las cuales ha sido designado como director.3. El usuario elige la carrera a su elección.4. El sistema muestra un menú en donde le permite al usuario ingresar características administrativas, ingresar horas adicionales ó aumentar ó quitar horas.5. Si elige ingresar características administrativas va al caso de uso A.6. Si elige ingresar horas adicionales va al caso de uso B.7. Si elige aumentar ó quitar horas. va al caso de uso C.

Tabla 3.13. Ingresar Características Administrativas

Caso de Uso	A – Ingresar Características Administrativas
Inicio	El usuario pulsa sobre el link del servicio Ingresar Características Administrativas
Precondición	El usuario debe encontrarse autenticado en el sistema y debe tener asignado el rol de director de departamento.
Camino Básico	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el link del servicio Ingresar Características Administrativas. 2. El sistema verifica que el periodo actual de ingreso de reportes este activo 3. El sistema muestra todos los docentes que dictan clases en ese departamento. 4. El usuario escoge el docente de su elección. 5. El sistema muestra todos los días correspondientes al periodo de ingreso de control de asistencia para los docentes. 6. El usuario escoge el día deseado. 7. El usuario escoge el día sobre el cual quiere ingresar novedades. 8. El sistema presenta una interfaz en donde se podrá escoger la características a ingresar y se colocarán valores y observaciones de las mismas. 9. El usuario ingresa las características y guarda la información. 10. El sistema realiza la operación y muestra un mensaje de éxito.
Camino Alternativo	<ul style="list-style-type: none"> • En el paso 2, si el sistema no encuentra periodos activos muestra un mensaje indicando dicha situación. • En el paso 9, si las características o valores ingresados son incorrectas o incompletos el sistema muestra un mensaje y no continua hasta que la situación sea corregida. • En el paso 10, en el caso de que existe algún error al momento de guardar la información el sistema muestra un error indicando el fallo.

Tabla 3.14. Ingresar Horas Adicionales

Caso de Uso	B – Ingresar Horas Adicionales
Inicio	El usuario pulsa sobre el link del servicio Ingresar Horas Adicionales
Precondición	El usuario debe encontrarse autenticado en el sistema y debe tener asignado el rol de director de departamento.

Camino Básico	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el link de Ingresar Horas Adicionales 2. El sistema muestra todos los docentes que pertenecen a ese departamento. 3. El usuario escoge el docente de su elección. 4. El sistema muestra todas las asignaturas que el docente seleccionado dicta actualmente. 5. El usuario escoge la asignatura deseada. 6. El sistema presenta todos los días en donde la asignatura debe ser dictada dentro del periodo de control actual. 7. El usuario escoge el día sobre el cual quiere ingresar novedades. 8. El sistema presenta una interfaz en donde se podrá escoger la características a ingresar y se colocarán valores y observaciones de las mismas. 9. El usuario ingresa las características y guarda la información. 10. El sistema realiza la operación y muestra un mensaje de éxito.
Camino Alternativo	<ul style="list-style-type: none"> • En el paso 5, si el sistema no encuentra reportes generados muestra un mensaje indicando dicha situación y no continua. • En el paso 9, si las características o valores ingresados son incorrectas o incompletos el sistema muestra un mensaje y no continua hasta que la situación sea corregida. • En el paso 10, en el caso de que existe algún error al momento de guardar la información el sistema muestra un error indicando el fallo.

Tabla 3.15. Aumentar o Quitar Horas

Caso de Uso	B – Aumentar o Quitar Horas
Inicio	El usuario pulsa sobre el link del servicio Aumentar o Quitar Horas
Precondición	El usuario debe encontrarse autenticado en el sistema y debe tener asignado el rol de director de departamento.
Camino Básico	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el link del servicio Aumentar o Quitar Horas 2. El sistema muestra todos los docentes que pertenecen a ese departamento. 3. El usuario escoge el docente de su elección. 4. El sistema presenta una interfaz en donde se podrá escoger la características a ingresar y se colocarán valores y observaciones de las mismas.

	<p>5. El usuario ingresa las características y guarda la información.</p> <p>6. El sistema realiza la operación y muestra un mensaje de éxito.</p>
Camino Alternativo	<ul style="list-style-type: none"> • En el paso 5, si las características o valores ingresados son incorrectas o incompletos el sistema muestra un mensaje y no continua hasta que la situación sea corregida. • En el paso 6, en el caso de que existe algún error al momento de guardar la información el sistema muestra un error indicando el fallo.

3.4- Requerimientos de interfaces externas

El único link hacia una interfaz externa que presenta el sistema es el que permite la comunicación entre el sistema de control docente y la base de datos del sistema académico de la Escuela Politécnica del Ejército. La necesidad de este enlace surge del requerimiento de obtener la información de materias y horarios asignados a cada uno de los docentes de la institución. Este enlace se implementará mediante la utilización de un JDBC el mismo que estará direccionado a dicho repositorio de información.

3.5- Requerimientos No Funcionales

El sistema presentará los siguientes requerimientos no funcionales:

- ✓ El desarrollo de la aplicación deberá apegarse a los estándares de desarrollo establecidos por la Escuela Politécnica del Ejército.
- ✓ El sistema deberá ser fácil de usar, haciendo de cada uno de sus funcionalidades, operaciones transparentes y sencillas de cara al usuario.

- ✓ El sistema será capaz de funcionar adecuadamente independientemente del sistema operativo o explorador web en el que se instale o se use respectivamente
- ✓ El sistema deberá implementar un control de errores para la ejecución de todos los procesos de la aplicación de tal manera de que al ocurrir una anomalía estos errores sean controlados adecuadamente y no se produzca el fallo total del sistema.
- ✓ El sistema basará su funcionamiento en la ejecución de funciones atómicas en donde el proceso debe ser culminado por completo para que realice cambios en la información.
- ✓ El sistema deberá ser lo suficientemente flexible para poder adaptarse a los cambios en los elementos del negocio sin que se requiera una modificación de consideración.
- ✓ La integridad y confidencialidad de los datos manejados por el sistema deberán estar garantizadas, se deberá implementar métodos que colaboran con dichas tareas como la encriptación de contraseñas.

CAPITULO 4

Diseño del sistema

4.1- Diseño de la Estructura del Sistema

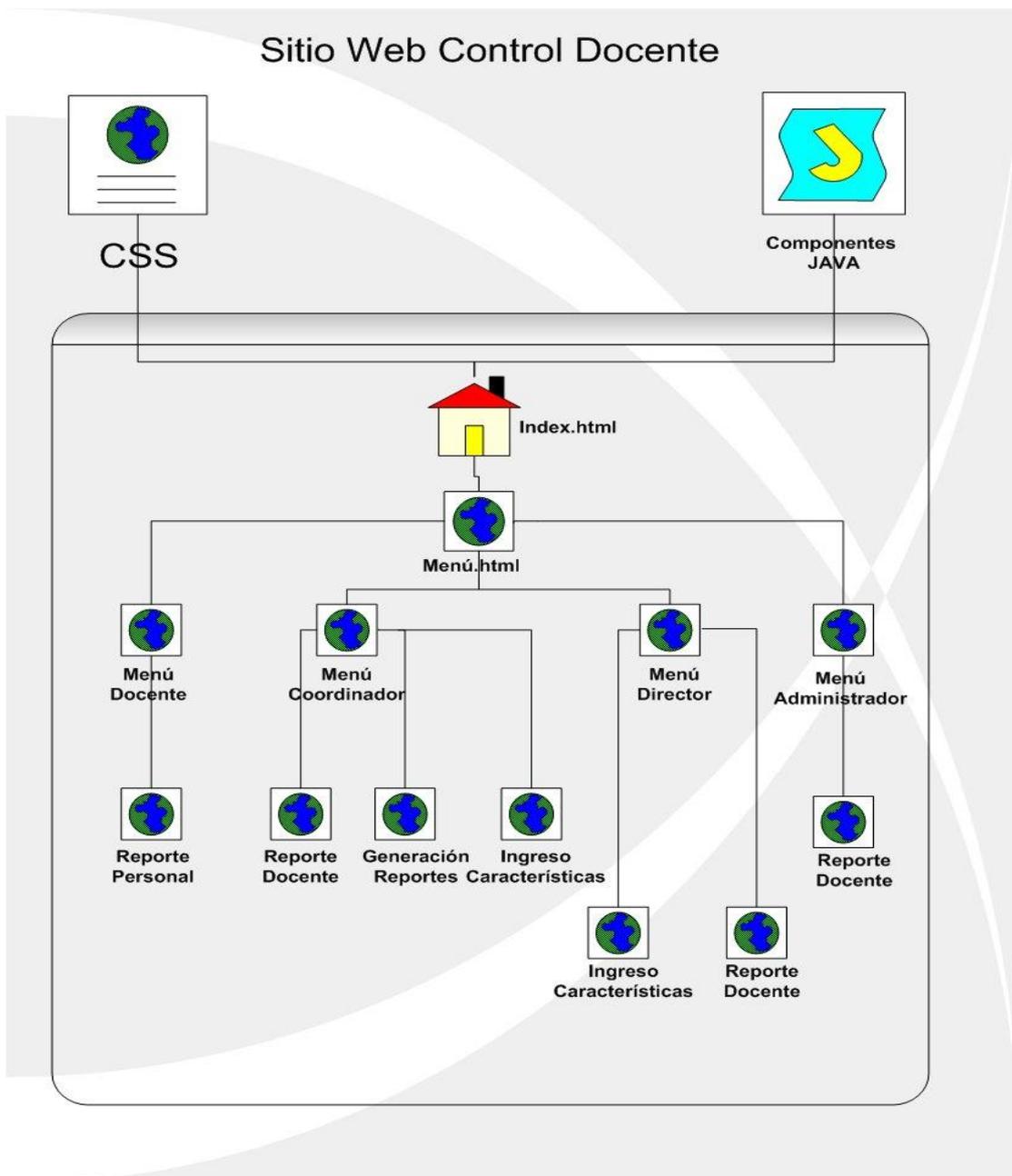


Figura 4.1. Estructura del Sistema

4.2- Definición de Metodología

La implementación del sistema se realizará utilizando metodología orientada a objetos con el soporte de la notación UML para la definición de algunos aspectos acerca de la funcionalidad del sistema.

En cuanto a la metodología de desarrollo de software se ha escogido el desarrollo por prototipos en donde se irán desarrollando componentes del sistema en cada iteración los mismos que serán agregados a la versión anterior logrando con cada iteración un producto mas completo y con mayor funcionalidad. Se ha elegido esta metodología puesto que es la recomendada al no tener plena certeza de que los requerimientos de software están completamente definidos desde el inicio del proyecto, por lo que el desarrollo por prototipos ayudará a que el usuario revise versiones parciales del proyecto pudiendo definir en le camino algunos de los requerimientos que se han pasado por alto. El diseño rápido lleva a la construcción de un prototipo. El prototipo lo evalúa el usuario y se utiliza para refinar los requisitos del software a desarrollar. La iteración ocurre cuando el prototipo se pone a punto para satisfacer las necesidades del cliente, permitiendo al mismo tiempo que el desarrollador comprenda mejor lo que se necesita hacer.

4.3- Diseño de la Arquitectura de la Aplicación

La arquitectura que se va a utilizar para el desarrollo del sistema es una arquitectura separada en 3 capas en donde se puede separar claramente la capa de presentación o FrontEnd, la capa de reglas de negocio y la capa de persistencia de datos (Base de Datos).

La utilización de capas como diseño de arquitectura permite obtener grandes ventajas en el producto final, como son la escalabilidad, la tolerancia a fallos, la facilidad en el mantenimiento y la optimización del pool de conexiones con la base de datos.

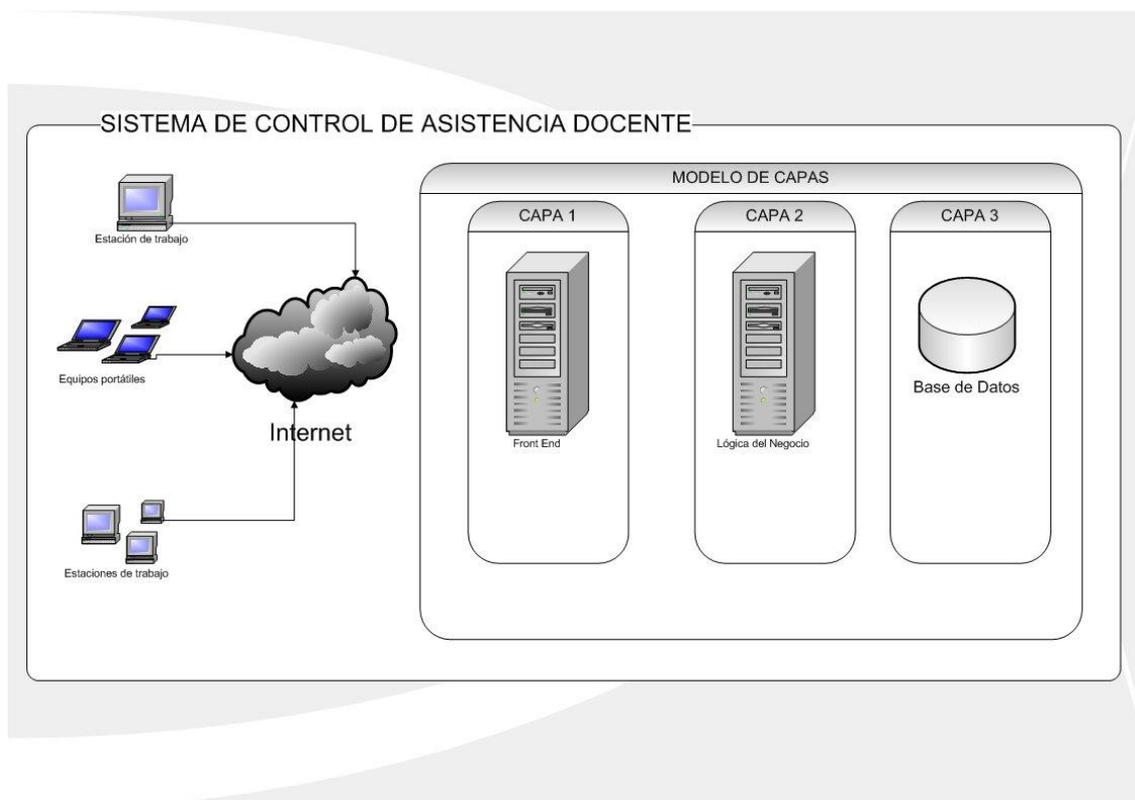


Figura 4.2. Arquitectura del Sistema

4.4- Definición de la plataforma de desarrollo

La plataforma de desarrollo que se utilizará para la realización del presente proyecto será J2EE¹⁶, apeándose a los estándares de desarrollo de la Escuela Politécnica del Ejército. Al ser este un proyecto web la implementación de la solución se hará en páginas java de servidor (JSP¹⁷). En

¹⁶ Java 2 Enterprise Edition

¹⁷ Java Server Pages

cuanto al IDE¹⁸ que va a ser empleado se ha escogido a Eclipse SDK¹⁹ 3.2 por la enorme funcionalidad que presenta al trabajar en proyectos que utilizan la plataforma JAVA.

4.5- Servidor de Aplicaciones JAVA

En la actualidad existen un sin número de servidores de aplicaciones para la plataforma J2EE; para la implementación y despliegue de esta aplicación se ha seleccionado el servidor Tomcat por ser el servidor oficial en la implementación de la tecnología de Java Servlets y Java Server Pages (JSP). Tomcat es desarrollado en un ambiente flexible y participativo, su principal funcionalidad es la de contener servlets en un entorno JSP, los mismos que son procesados e invocados conforme el usuario los requiera.

Tomcat sin lugar a dudas es la solución de servidor más adecuada para albergar la aplicación a ser desarrollada puesto que como ya se ha dicho presenta un alto grado de integración con aplicaciones J2EE.

4.6- Motor de Base de Datos

Para la implementación del sistema de base de datos se ha escogido la plataforma Oracle 10G, actualmente gran cantidad de información y aplicaciones dentro del sistema integrado de información de la Escuela Politécnica del Ejército utilizan este sistema de administración de base de datos relacionales como su repositorio de información por lo que se pensó adecuado utilizar dicha plataforma para la base de datos del software a desarrollarse además de cumplir con la premisa de mantener los estándares de desarrollo de la institución.

¹⁸ Integrated Development Environment

¹⁹ Software Development Kit

4.7- Reconocimiento de Objetos

4.7.1- Entidad de Datos-Calendario

La entidad de datos calendario será utilizada para manejar los periodos de control de asistencia existentes en el sistema.

Ej.

Tabla 4.1. Entidad de Datos-Calendario

fechaInicio	fechaFin	descripción	periodo	flagActivo
20/10/2008	19/11/2008	Octubre – Noviembre 2008	1	1

Calendario
-fechaInicio : Date
-fechaFin : Date
-descripcion : String
-periodo : Integer
-flagActivo : Integer
+getFechaInicio() : Date
+setFechaInicio(entrada fecha : Date)
+getFechaFin() : Date
+setFechaFin(entrada fecha : Date)
+getDescripcion() : String
+setDescripcion(entrada descripcion : String)
+getPeriodo() : Integer
+setPeriodo(entrada periodo : Integer)
+getFlagActivo() : Integer
+setFlagActivo(entrada flagActivo : Integer)

Figura 4.3. Entidad de Datos–Calendario

4.7.2- Entidad de Datos-CaracteristicasHoras

Esta entidad de datos sirve para manejar la información acerca de las diferentes propiedades que puede tener una hora de clase o administrativa en particular.

Ej.

Tabla 4.2. Entidad de Datos-CaracteristicasHoras

Código	Descripción	Porcentaje	tipoDocente	Clave
001	Atraso a Clases	5	Tiempo Completo	ACTC



Figura 4.4. Entidad de Datos- CaracteristicasHoras

4.7.3- Entidad de Datos-Asigna_Coordinador

La entidad de datos Asigna_Coordinador sirve para manejar la información acerca de los coordinadores de carrea en el sistema.

Ej.

Tabla 4.3. Entidad de Datos-Asigna_Coordinador

Cedula	Localidad	Modalidad	Facultad	Especialidad	Usuario	Flagactivo	descripción
1714884770	1	1	SIS	1	321	1	Coordinador SIS

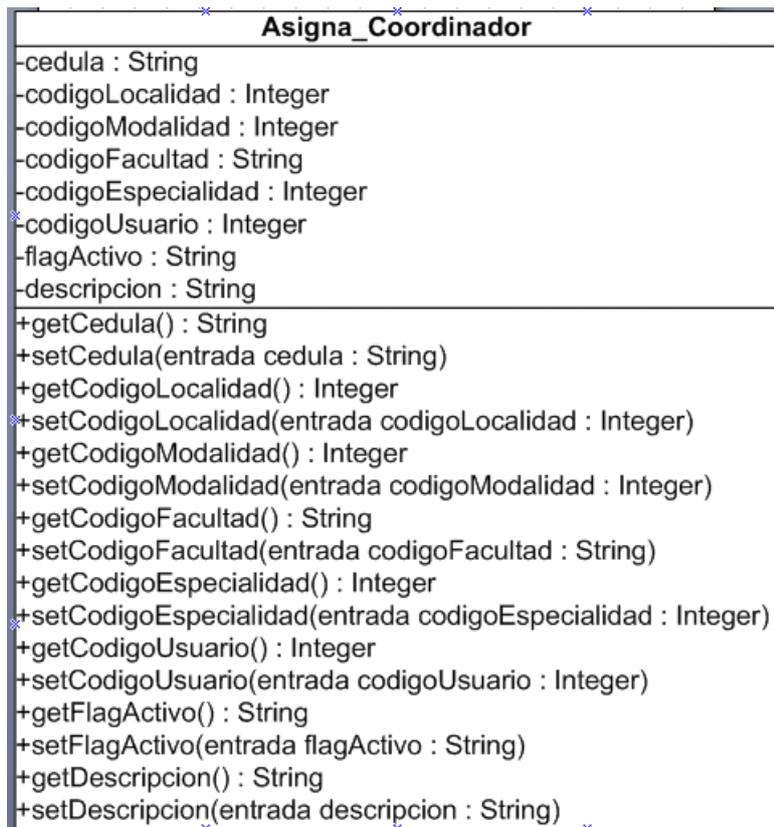


Figura 4.5. Entidad de Datos-Asigna_Coordinador

4.7.4- Entidad de Datos-DirectorDepartamento

Entidad utilizada para manejar la información del director de departamento en la aplicación.

Ej.

Tabla 4.4. Entidad de Datos-DirectorDepartamento

Cedula	Nombre	Apellido	departamento
1714884770	Francisco	Naranjo	SIS

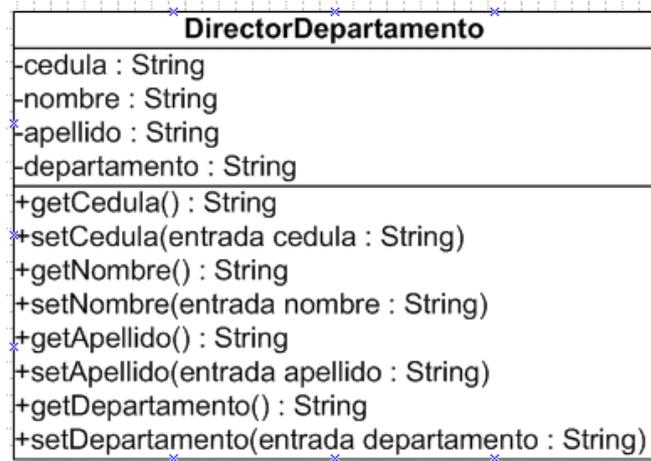


Figura 4.6. Entidad de Datos-DirectorDepartamento

4.7.5- Entidad de Datos-DocenteXDepartamento

Entidad utilizada para la instancia de un docente dentro de la aplicación.

Ej.

Tabla 4.5. Entidad de Datos-DocenteXDepartamento

CedulaDocente	NombreDocente
1714884770	Francisco Naranjo



Figura 4.7. Entidad de Datos-DocenteXDepartamento

4.7.6- Entidad de Datos-Horario

Entidad utilizada para representar el horario en el que se dicta una hora de clase en particular.

Ej.

Tabla 4.6. Entidad de Datos-Horario

horalnicio	horaFin	NombreDia
19:30	21:30	Jueves

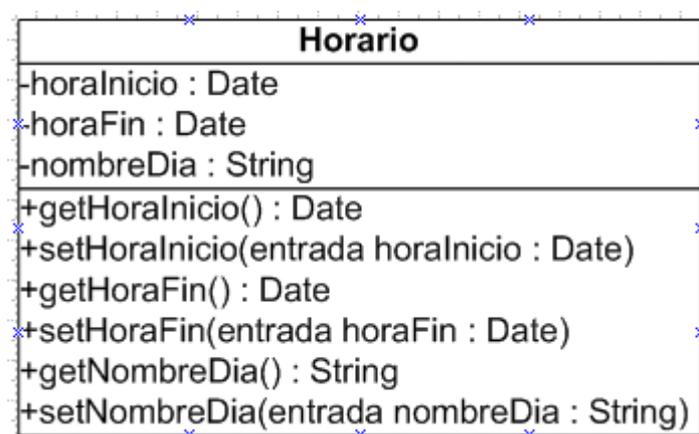


Figura 4.8. Entidad de Datos-Horario

4.7.7- Entidad de Datos-HoraDictada

Esta entidad sirve para llevar el registro de las horas dictadas así como de sus características.

Ej.

Tabla 4.7. Entidad de Datos-HoraDictada

ced	facultad	mat	Paralelo	Fecha	Num	periodo	obs	flagRevisada	Hora	docente
17148840	SIS	111	1	0/02/02	2	1	-	1	horaClase	TC

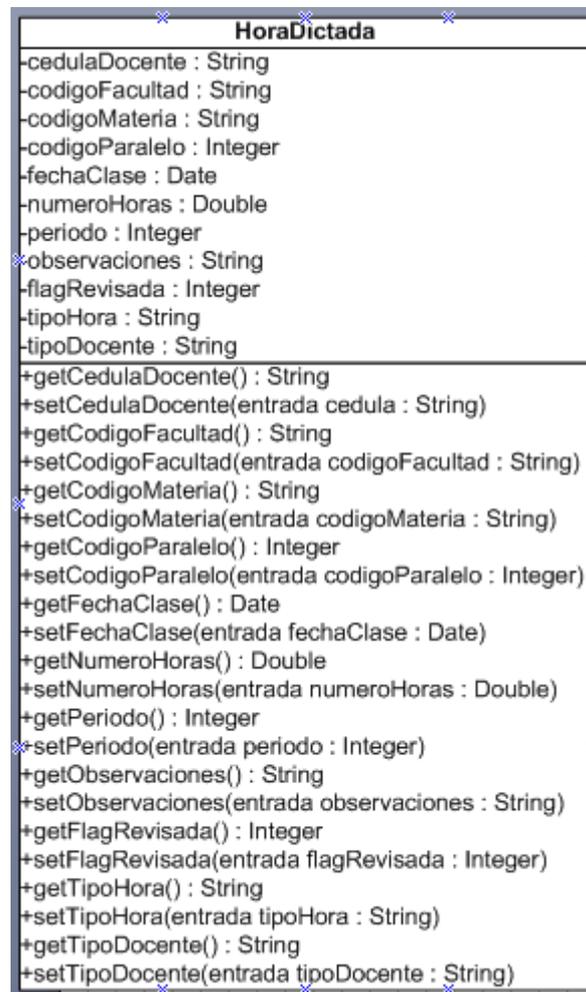


Figura 4.9. Entidad de Datos-HoraDictada

4.7.8- Entidad de Datos-MateriaDocente

Entidad para gestionar las materias que dicta un docente en la ejecución de la aplicación.

Ej.

Tabla 4.8. Entidad de Datos-MateriaDocente

descripcionMateria	codigoMateria	codigoParalelo	codFacultad
Algebra1	123456789	2	SIS

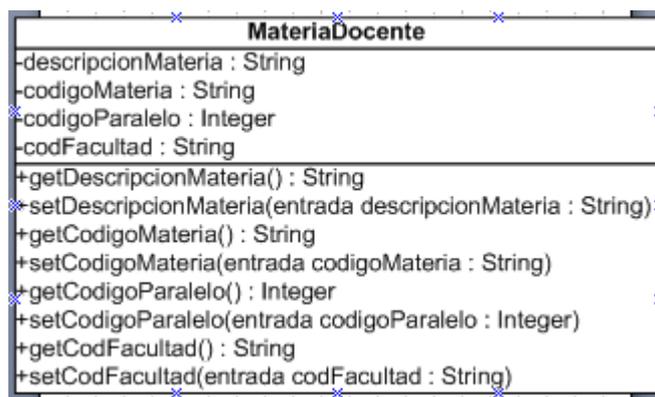


Figura 4.10. Entidad de Datos-MateriaDocente

4.7.9- Entidad de Datos-ReporteGenerado

Esta entidad permite manejar la información de los reportes de asistencia generados.

Ej.

Tabla 4.9. Entidad de Datos-ReporteGenerado

Periodo	codCarrera
1	SIS

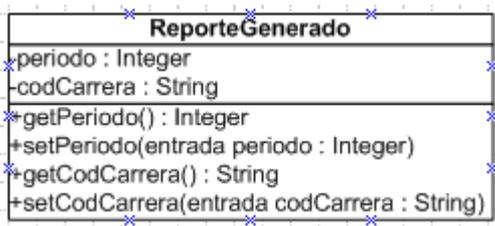


Figura 4.11. Entidad de Datos-ReporteGenerado

4.7.10- Entidad de Datos-TipoDocente

Esta entidad permite manejar la información acerca de los docentes y su categorización dentro del sistema.

Ej.

Tabla 4.10. Entidad de Datos-TipoDocente

Cedula	Nombre	tipoDocente	valorHora	departamento
1714884770	Francisco Naranjo	Tiempo Completo	5.0	SIS

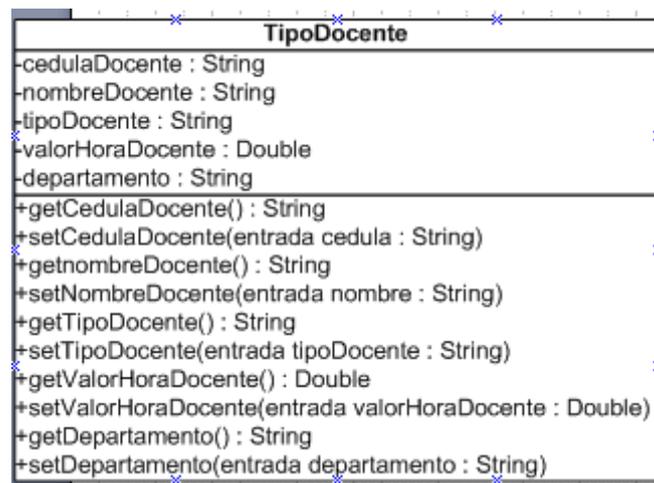


Figura 4.12. Entidad de Datos-TipoDocente

4.8- Clases Utilitarias

Las clases utilitarias no representan persistencia, es decir, no representan ningún objeto de la lógica del negocio, sin embargo, son de gran utilidad ya que prestan servicios globales utilizados por la gran mayoría de clases dentro del sistema.

- ✓ La clase **DatabaseJNDI** es la que obtiene conexiones hacia la base de datos, y nos permite mediante las mismas ejecutar sentencias de consulta y acción con la base de datos. Su estructura se encuentra conformada por un atributo del tipo DataSource el mismo que es inicializado con el nombre de un contexto determinado ubicado en el archivo de configuración del servidor, y por dos funciones que permiten obtener y cerrar el objeto de conexión(`java.sql.Connection`) respectivamente.

- ✓ La clase **Gestor** que es la que nos permite ejecutar todo tipo de sentencias contra el motor de base de datos, las sentencias incluyen comandos: `select`, `insert`, `delete`, `update`. La particularidad de esta clase es la independencia y nivel de abstracción que presenta, de tal manera que es utilizada como clase padre para todas las clases que representen interacción entre clases de persistencia y el repositorio de información.

- ✓ La clase **Clave** y **HexUtil** son componentes que nos ayudan a ejecutar los algoritmos que permiten la encriptación de las contraseñas para proveer seguridad al sistema.

- ✓ La clase **Fechas** es una clase que nos permite utilizar todo tipo de funciones de gran utilidad para el manejo de fechas, como es de conocimiento general el manejo de los valores de fechas siempre presenta una complicación debido a la diferencia de criterios en cuenta a la definición del tipo de dato dependiendo de la configuración regional del software que las maneje, por lo que es de vital importancia disponer de un componente que nos facilite su manipulación.

- ✓ La clase **SessionManager** permite manejar la seguridad en el sistema, es la que se encarga de validar la autenticidad del usuario que desea ingresar en el sistema.

- ✓ La clase **ProfileManager** se encarga del manejo de los diferentes roles que pueden tener un usuario una vez que este se ha autenticado de manera correcta en el sistema.

4.9- Diseño de Base de Datos

La base de datos como ya se indicó será implementada en la plataforma Oracle 10G, a continuación se presenta las tablas necesarias para el funcionamiento del sistema y su correspondiente script de creación.

✓ Tabla Calendario

Column Name	Data Type	Nullable	Default	Primary Key
COD_CALEDARIO	NUMBER	No	-	1
DESCRIPCION	VARCHAR2(30)	No	-	-
FEC_INICIO	DATE	No	-	-
FEC_FIN	DATE	No	-	-
PERIODO_HORA	NUMBER	Yes	-	-
FLG_ACTIVO	NUMBER	Yes	-	-
				1 - 6

Figura 4.13. Tabla Calendario

```
CREATE TABLE CALEDARIO
(
  COD_CALEDARIO NUMBER          NOT NULL,
  DESCRIPCION  VARCHAR2 (100)  NOT NULL,
  FEC_INICIO   DATE             NOT NULL,
  FEC_FIN      DATE             NOT NULL,
  PERIODO_HORA NUMBER,
  FLG_ACTIVO   NUMBER,
  CONSTRAINT CALEDARIO_PK PRIMARY KEY (COD_CALEDARIO)
)
```

✓ Tabla Coordinador

Column Name	Data Type	Nullable	Default	Primary Key
COD_LOCALIDAD	NUMBER	No	-	-
COD_MODALIDAD	NUMBER	No	-	-
COD_FACULTAD	CHAR(3)	No	-	-
COD_ESPECIALIDAD	NUMBER	No	-	-
CED_DOCENTE	NVARCHAR2(10)	No	-	-
FLG_ACTIVO	CHAR(1)	Yes	-	-
COD_USUARIO	NUMBER	Yes	-	-
DESCRIPCION	NVARCHAR2(40)	No	-	-
				1 - 8

Figura 4.14. Tabla Coordinador

```

CREATE TABLE COORDINADOR
(
  COD_LOCALIDAD    NUMBER                NOT NULL,
  COD_MODALIDAD    NUMBER                NOT NULL,
  COD_FACULTAD     CHAR (3)              NOT NULL,
  COD_ESPECIALIDAD NUMBER                NOT NULL,
  CED_DOCENTE      VARCHAR2 (10)         NOT NULL,
  FLG_ACTIVO       CHAR (1),
  COD_USUARIO      NUMBER,
  DESCRIPCION      VARCHAR2 (40)         NOT NULL,
  CONSTRAINT COORDINADOR_PK PRIMARY KEY (COD_LOCALIDAD,
  COD_MODALIDAD,COD_FACULTAD,COD_ESPECIALIDAD,CED_DOCENTE
)
)

```

✓ **Tabla Caracteristicas_Horas**

Column Name	Data Type	Nullable	Default	Primary Key
CODIGO	VARCHAR2(3)	No	-	1
DESCRIPCION	VARCHAR2(30)	No	-	-
PORCENTAJE	FLOAT	No	-	-
TIPO_DOCENTE	VARCHAR2(3)	No	-	-
CLAVE	VARCHAR2(50)	No	-	-
				1 - 5

Figura 4.15. Tabla Caracteristicas_Horas

```

CREATE TABLE CARACTERISTICAS_HORAS
(
  CODIGO          VARCHAR2(3)           NOT NULL,
  DESCRIPCION     VARCHAR2(30)          NOT NULL,
  PORCENTAJE     FLOAT                  NOT NULL,
  TIPO_DOCENTE   VARCHAR2(3)           NOT NULL,
  CLAVE          VARCHAR2(50)           NOT NULL,
  CONSTRAINT CARACTERISTICAS_HORAS_PK PRIMARY KEY
(CODIGO)
)

```

✓ **Tabla Director_Departamento**

Column Name	Data Type	Nullable	Default	Primary Key
CEDULA	VARCHAR2(10)	No	-	1
DEPARTAMENTO	VARCHAR2(30)	No	-	2
NOMBRE	VARCHAR2(30)	No	-	-
APELLIDO	VARCHAR2(30)	No	-	-
				1 - 4

Figura 4.16. Tabla Director_Departamento

```
CREATE TABLE DIRECTOR_DEPARTAMENTO
(
  CEDULA          VARCHAR2(10)          NOT NULL,
  NOMBRE          VARCHAR2(30)          NOT NULL,
  APELLIDO        VARCHAR2(30)          NOT NULL,
  DEPARTAMENTO    VARCHAR2(30)          NOT NULL,
  CONSTRAINT DIRECTOR_DEPARTAMENTO_PK PRIMARY KEY
(CEDULA,DEPARTAMENTO)
)
```

✓ **Tabla Horas_Docente**

Column Name	Data Type	Nullable	Default	Primary Key
CED_DOCENTE	NVARCHAR2(10)	No	-	-
COD_MATERIA	NVARCHAR2(11)	No	-	-
COD_PARALELO	NUMBER	Yes	-	-
FECHA_CLASE	DATE	Yes	-	-
NUMERO_HORAS	NUMBER	No	-	-
COD_FACULTAD	NVARCHAR2(3)	No	-	-
PERIODO_HORA	NUMBER	Yes	-	-
OBSERVACIONES	NVARCHAR2(300)	Yes	-	-
FLG_REVISADO	NUMBER	Yes	-	-
TIPO_HORA	NVARCHAR2(50)	Yes	-	-
TIPO_DOCENTE	NVARCHAR2(3)	Yes	-	-
				1 - 11

Figura 4.17. Tabla Horas_Docente

```

CREATE TABLE HORAS_DOCENTE
(
  CED_DOCENTE  VARCHAR2 (10)          NOT NULL,
  COD_MATERIA  VARCHAR2 (11)          NOT NULL,
  COD_PARALELO NUMBER,
  FECHA_CLASE  DATE,
  NUMERO_HORAS NUMBER                NOT NULL,
  COD_FACULTAD VARCHAR2 (3)          NOT NULL,
  PERIODO_HORA NUMBER,
  OBSERVACIONES VARCHAR2 (300),
  FLG_REVISADO NUMBER                NOT NULL,
  TIPO_HORA    VARCHAR2 (50)          NOT NULL,
  TIPO_DOCENTE VARCHAR2 (3)          NOT NULL,
)

```

✓ **Tabla Perfiles**

Column Name	Data Type	Nullable	Default	Primary Key
COD_PERFIL	NUMBER(1,0)	No	-	1
NOM_PERFIL	VARCHAR2(4000)	No	-	-
URLIMAGEN_PERFIL	VARCHAR2(4000)	No	-	-
				1 - 3

Figura 4.18. Tabla Perfiles

```

CREATE TABLE PERFILES
(
  COD_PERFIL    NUMBER(1)              NOT NULL,
  NOM_PERFIL    VARCHAR2(100)          NOT NULL,
  URLIMAGEN_PERFIL VARCHAR2(60)        NOT NULL,
  CONSTRAINT PERFILES_PK PRIMARY KEY ( COD_PERFIL )
)

```

✓ **Tabla PerfilesXUsuario**

Column Name	Data Type	Nullable	Default	Primary Key
CED_USUARIO	VARCHAR2(4000)	No	-	1
COD_PERFIL	NUMBER(1,0)	No	-	2
				1 - 2

Figura 4.19. Tabla PerfilesXUsuario

```

CREATE TABLE PERFILESXUSUARIO
(
CED_USUARIO VARCHAR2(15)          NOT NULL,
COD_PERFIL  NUMBER(1)              NOT NULL,
CONSTRAINT PERFILESXUSUARIO_CON PRIMARY KEY (CED_USUARIO,
COD_PERFIL),
CONSTRAINT PERFILESXUSUARIO_FK FOREIGN KEY (CED_USUARIO)
REFERENCES USUARIOS (CED_USUARIO)),
CONSTRAINT PERFILESXUSUARIO_FK2 FOREIGN KEY (COD_PERFIL)
REFERENCES PERFILES (COD_PERFIL)
)

```

✓ **Tabla Reportes_Generados**

Column Name	Data Type	Nullable	Default	Primary Key
PERIODO	NUMBER	No	-	1
COD_CARRERA	VARCHAR2(5)	No	-	2
				1 - 2

Figura 4.20. Tabla Reportes_Generados

```

CREATE TABLE REPORTES_GENERADOS
(
PERIODO          NUMBER          NOT NULL,
COD_CARRERA     VARCHAR2(5)     NOT NULL,
CONSTRAINT DIRECTOR_DEPARTAMENTO_PK PRIMARY KEY
(PERIODO,COD_CARRERA)
)

```

✓ **Tabla Servicios**

Column Name	Data Type	Nullable	Default	Primary Key
COD_SERVICIO	NUMBER	No	-	1
NOM_SERVICIO	VARCHAR2(4000)	No	-	-
URL_SERVICIO	VARCHAR2(4000)	No	-	-
FLAG_ACTIVO	NUMBER	Yes	-	-
				1 - 4

Figura 4.21. Tabla Servicios

```

CREATE TABLE SERVICIOS
(
  COD_SERVICIO NUMBER          NOT NULL,
  NOM_SERVICIO VARCHAR2(100)  NOT NULL,
  URL_SERVICIO VARCHAR2(100)  NOT NULL,
  FLAG_ACTIVO  NUMBER,
  CONSTRAINT SERVICIOS_CON PRIMARY KEY (COD_SERVICIO)
)

```

✓ **Tabla ServiciosXPerfil**

Column Name	Data Type	Nullable	Default	Primary Key
COD_SERVICIOSXPERFIL	NUMBER(1,0)	No	-	1
COD_PERFIL	NUMBER(1,0)	No	-	-
COD_SERVICIO	NUMBER(1,0)	No	-	-
				1 - 3

Figura 4.22. Tabla ServiciosXPerfil

```

CREATE TABLE SERVICIOSXPERFIL
(
  COD_SERVICIOSXPERFIL NUMBER(1)          NOT NULL,
  COD_PERFIL           NUMBER(1)          NOT NULL,
  COD_SERVICIO         NUMBER(1)          NOT NULL,
  CONSTRAINT SERVICIOSXPERFIL_PK PRIMARY KEY
(COD_SERVICIOSXPERFIL),
  CONSTRAINT SERVICIOSXPERFIL_FK FOREIGN KEY (COD_PERFIL)
REFERENCES PERFILES (COD_PERFIL)),
  CONSTRAINT SERVICIOSXPERFIL_FK2 FOREIGN KEY (COD_SERVICIO)
REFERENCES SERVICIOS (COD_SERVICIO))
)

```

✓ **Tabla Usuarios**

Column Name	Data Type	Nullable	Default	Primary Key
CED_USUARIO	VARCHAR2(4000)	No	-	1
NOM_USUARIO	VARCHAR2(4000)	No	-	-
APE_USUARIO	VARCHAR2(4000)	No	-	-
PAS_MD5	VARCHAR2(4000)	No	-	-
				1 - 4

Figura 4.23. Tabla Usuarios

```

CREATE TABLE USUARIOS
(
  CED_USUARIO VARCHAR2(15)          NOT NULL,
  NOM_USUARIO VARCHAR2(30)         NOT NULL,
  APE_USUARIO VARCHAR2(30)         NOT NULL,
  PAS_MD5     VARCHAR2(100)        NOT NULL,
  CONSTRAINT USUARIOS_PK PRIMARY KEY (CED_USUARIO)
)

```

✓ **Tabla Tipo_Docente**

Column Name	Data Type	Nullable	Default	Primary Key
CED_DOCENTE	VARCHAR2(10)	No	-	1
MODALIDAD	VARCHAR2(4)	No	-	2
NOM_DOCENTE	VARCHAR2(30)	No	-	-
TIP_DOCENTE	VARCHAR2(3)	No	-	-
VALOR_HORA	FLOAT	No	-	-
DEPARTAMENTO	VARCHAR2(10)	No	-	-
				1 - 6

Figura 4.24. Tabla Tipo_Docente

```

CREATE TABLE TIPO_DOCENTE
(
  CED_DOCENTE VARCHAR2(10)          NOT NULL,
  NOM_DOCENTE VARCHAR2(30)         NOT NULL,
  TIP_DOCENTE VARCHAR2(3)          NOT NULL,
  VALOR_HORA  FLOAT                NOT NULL,
  DEPARTAMENTO VARCHAR2(10)         NOT NULL,
  MODALIDAD   VARCHAR2(4)          NOT NULL,
  CONSTRAINT TIPO_DOCENTE_PK PRIMARY KEY
(CED_DOCENTE,MODALIDAD)
)

```

✓ **Diagrama de Base de Datos**

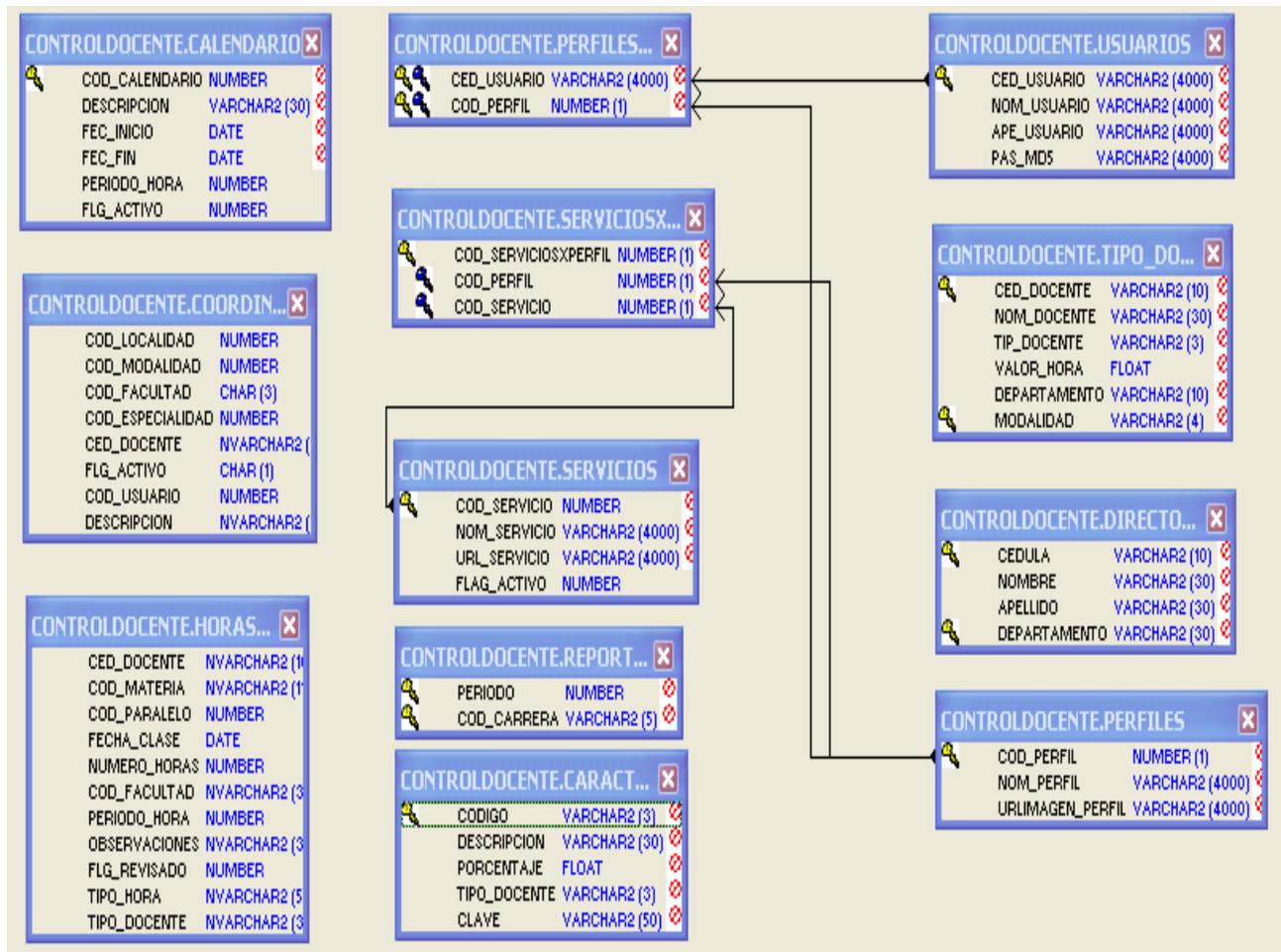


Figura 4.25. Diagrama de Base de Datos

4.10- Codificación del Sistema

Dada la extensión de la codificación del sistema esta se encontrará en un medio físico de almacenamiento adjunto al presente proyecto. En este se encontrará el código fuente de cada una las versiones del sistema junto con los archivos de configuración del servidor y script de base de datos necesarios para el correcto funcionamiento de la aplicación.

4.11- Modelo de Componentes

El sistema se ha dividido en dos grandes componentes, el uno hace referencia a las páginas de interfaz de usuario y el otro a las clases JAVA necesarias para la implementación del sistema. Cada uno de estos componentes se subdivide en grupos dependiendo del requerimiento o funcionalidad que vayan a manejar. Como se explicó anteriormente estos componentes se desplegarán en un servidor Tomcat y tendrán dos conexiones hacia repositorios de información: la base de datos del sistema (Oracle 10G) y la base de datos del sistema académico de la institución.

✓ Diagrama de Componentes

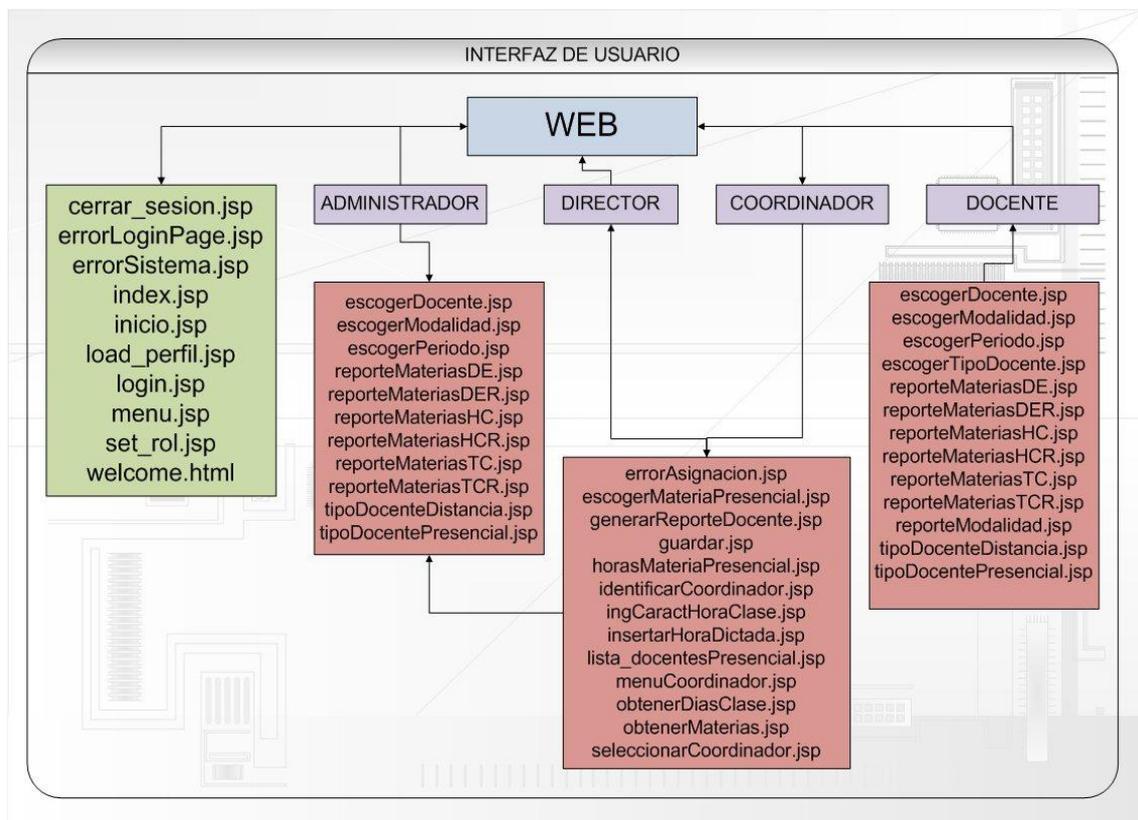


Figura 4.26. Diagrama de Componentes-Web

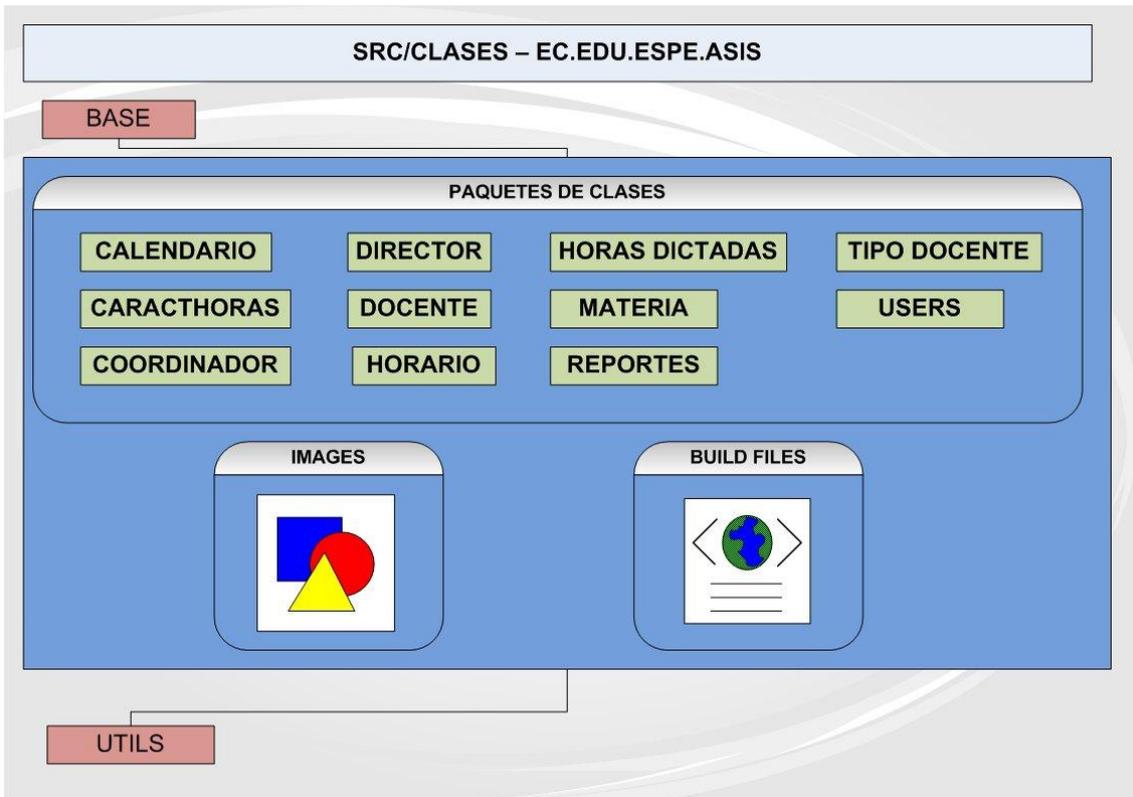


Figura 4.27. Diagrama de Componentes-Clases

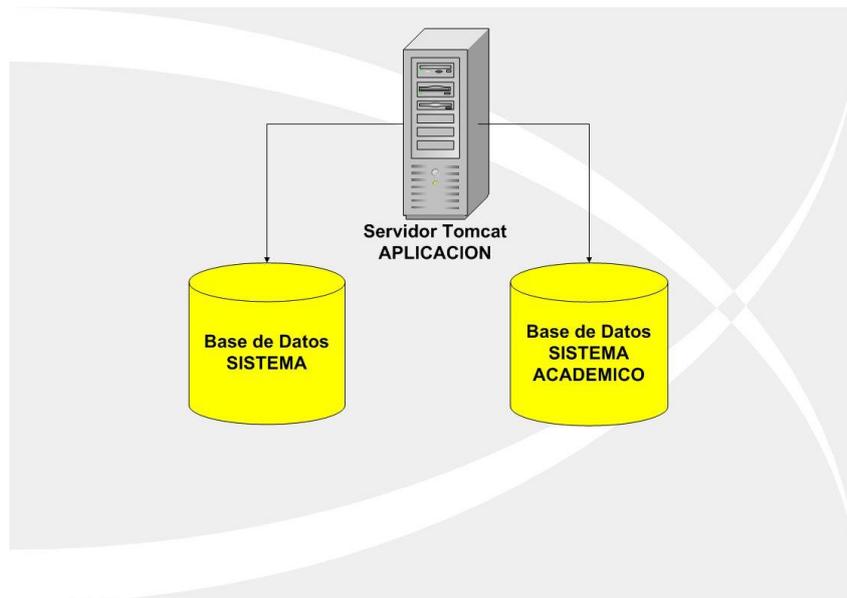


Figura 4.28. Diagrama de Despliegue

CAPITULO 5

Implementación y Pruebas

5.1- Implementación

Para la implementación de este sistema es necesario realizar los siguientes pasos:

- **Instalación de JRE²⁰ en el Sistema Operativo.**

El conjunto de paquetes envueltos en JRE son las clases y librerías que permiten la ejecución a las aplicaciones basadas en el lenguaje JAVA. Por lo que es necesaria su instalación en el sistema operativo base en donde vaya a ser desplegado el sistema del presente proyecto. Tanto el servidor Tomcat como el IDE de desarrollo necesario para compilar el sistema, son aplicaciones que dependen de esta librería por lo que su instalación debe ser lo primero que se haga.

Existen varias versiones JRE en el mercado, la última es la 1.6, la versión que se utilizó para el desarrollo de este proyecto es la 1.4.2 ya que es la misma que actualmente se usa en los sistemas JAVA de la institución. El instalador es parte del kit de desarrollo de JAVA (JSDK²¹) que se puede conseguir en el sitio web de su fabricante: <http://www.java.sun.com>. El instalador de la versión de JSDK utilizada en el desarrollo del presente proyecto se incluye en el medio de almacenamiento adjunto a este documento. Una vez instalado es necesario declarar las variables de entorno en el sistema operativo para que este pueda utilizar las librerías. Dicha variable de entorno

²⁰ JAVA Runtime Environment

²¹ JAVA Software Development Kit

debe llevar el nombre de JAVA_HOME y deberá apuntar al directorio de instalación utilizado para el JSDK.

- **Instalación del Servidor Tomcat**

El servidor Tomcat utilizado para el desarrollo de este proyecto se adjunta como un archivo comprimido con el nombre de tomcat5.zip, para su utilización solamente es necesario descomprimir este archivo. Se recomienda descomprimirlo en la raíz para evitar problemas en el tamaño de las rutas de acceso hacia la aplicación.

Una vez descomprimido se observarán varias carpetas, que incluyen archivos en donde se podrán realizar configuraciones para el funcionamiento del servidor además se encuentran los archivos necesarios para su ejecución. La carpeta conf es la que contiene los archivos de configuración del servidor, los dos más importantes son:

- tomcat-users.xml que es donde se definen los usuarios del servidor, los que se utilizarán para su administración y monitoreo.
- server.xml que es el que lleva la configuración de los parámetros de ejecución del servidor en este se definen parámetros como el puerto en donde va a operar y los contextos necesarios para realizar las búsquedas por JNDI de la configuración de acceso a las bases de datos. La configuración completa necesaria de este archivo para la ejecución de la aplicación se incluye en el medio de almacenamiento adjunto.

La carpeta bin es la que contiene los archivos binarios que ejecutan acciones sobre el servidor, están por ejemplo las acciones startup.bat y shutdown.bat que inician o detienen al servidor respectivamente.

- **Instalación de IDE, compilación y despliegue en el servidor.**

El IDE utilizado para el desarrollo de este proyecto es Eclipse SDK Versión 3.2.0, este se incluye en el archivo comprimido eclipse.zip, al igual que con el servidor Tomcat solamente es necesario descomprimirlo para poder utilizarlo.

Una vez descomprimido encontraremos el archivo ejecutable eclipse.exe para acceder a la aplicación.

Eclipse necesita ser configurado adecuadamente para que se integre al entorno de desarrollo utilizado.

- **Configuración JAVA.**

Es necesario configurar al IDE con la versión de JAVA instalada previamente para que este pueda realizar sus funciones de compilación.

Para esto debemos:

1.- Ingresar a las opciones de configuración, escogemos el menú Window y a continuación seleccionamos la opciones Preferences...



Figura 5.1. Barra de Herramientas Eclipse IDE SDK.

2.- A continuación seleccionamos la opción java y dentro de esta seleccionamos la opción Intalled JREs.

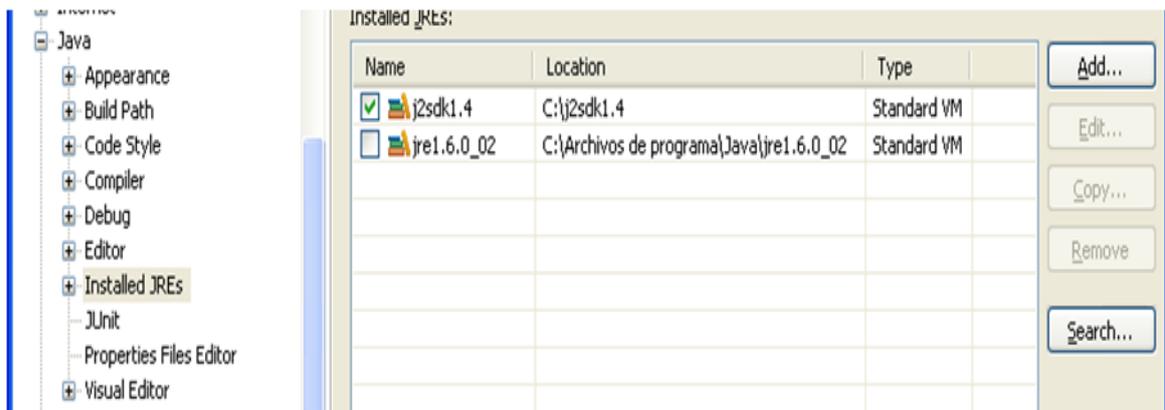


Figura 5.2. Ventana de Configuración Eclipse (JAVA).

Mediante la utilización de la opción add seleccionamos la ubicación de instalación del JRE instalado previamente.

- **Configuración Tomcat**

La versión de Eclipse utilizada presenta un plug-in que permite trabajar directamente con el servidor Tomcat instalado, para esto es necesario indicarle al IDE la ubicación y versión de dicho servidor.

1.- Ingresamos al menú de configuraciones como el paso anterior utilizando el menú Window.

2.- A continuación seleccionamos la opción Tomcat y dentro de esta colocamos los valores correspondientes de acuerdo con la instalación anterior del servidor Tomcat.

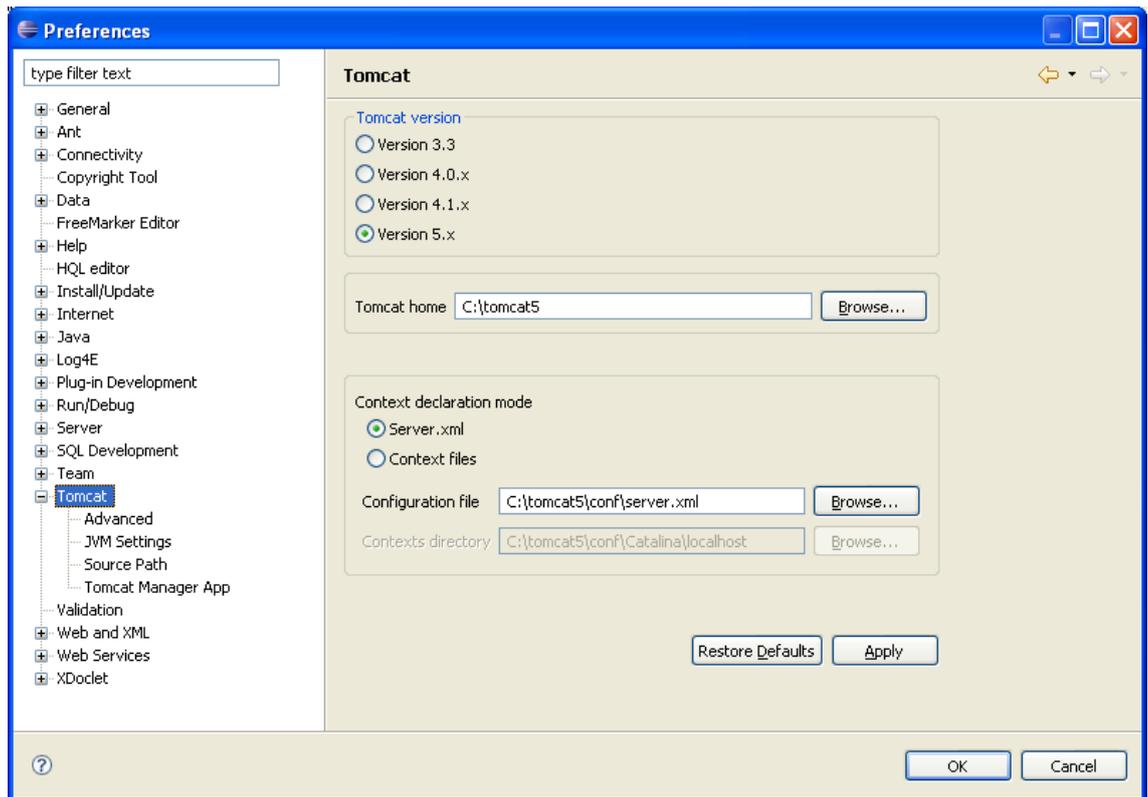


Figura 5.3. Ventana de Configuración Eclipse (Tomcat).

- **Compilación y Despliegue**

Para realizar las tareas de compilación y despliegue de la aplicación en el servidor, primero es necesario importar las fuentes del sistema en nuestro espacio de trabajo en el IDE.

- 1.- En la ventana de Package Explorer de Eclipse pulsamos click derecho y seleccionamos la opción new y posteriormente escogemos la opción Project.
- 2.- Se desplegará la ventana en donde escogeremos el tipo de proyecto que vamos a realizar, escogemos la opción Java Project.

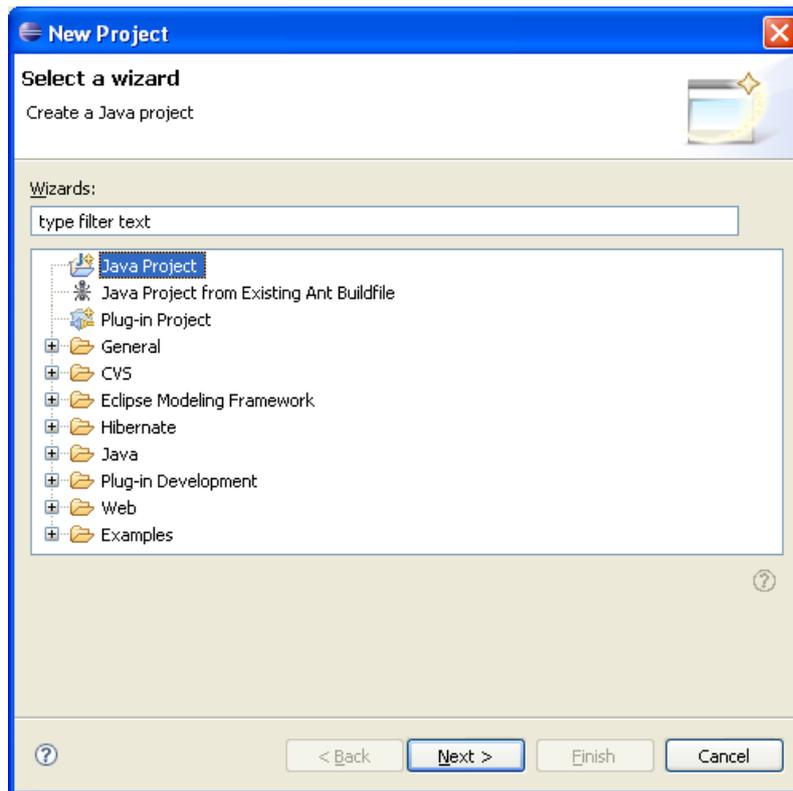


Figura 5.4. Ventana de Nuevo Proyecto.

3.- Damos click sobre el botón next y se desplegará la ventana sobre las opciones de configuración del proyecto, en esta colocamos el nombre del proyecto “**controldocente**” y seleccionamos la opción “**Create project from existing source**”, en la ruta al directorio de las fuentes colocamos la dirección en donde se encuentre la carpeta que contiene las fuentes de la aplicación las mismas que se encuentran en el medio de almacenamiento adjunto, para la importación de estas al sitio de trabajo del IDE es necesario seleccionar la ultima versión de la aplicación y descomprimirla.

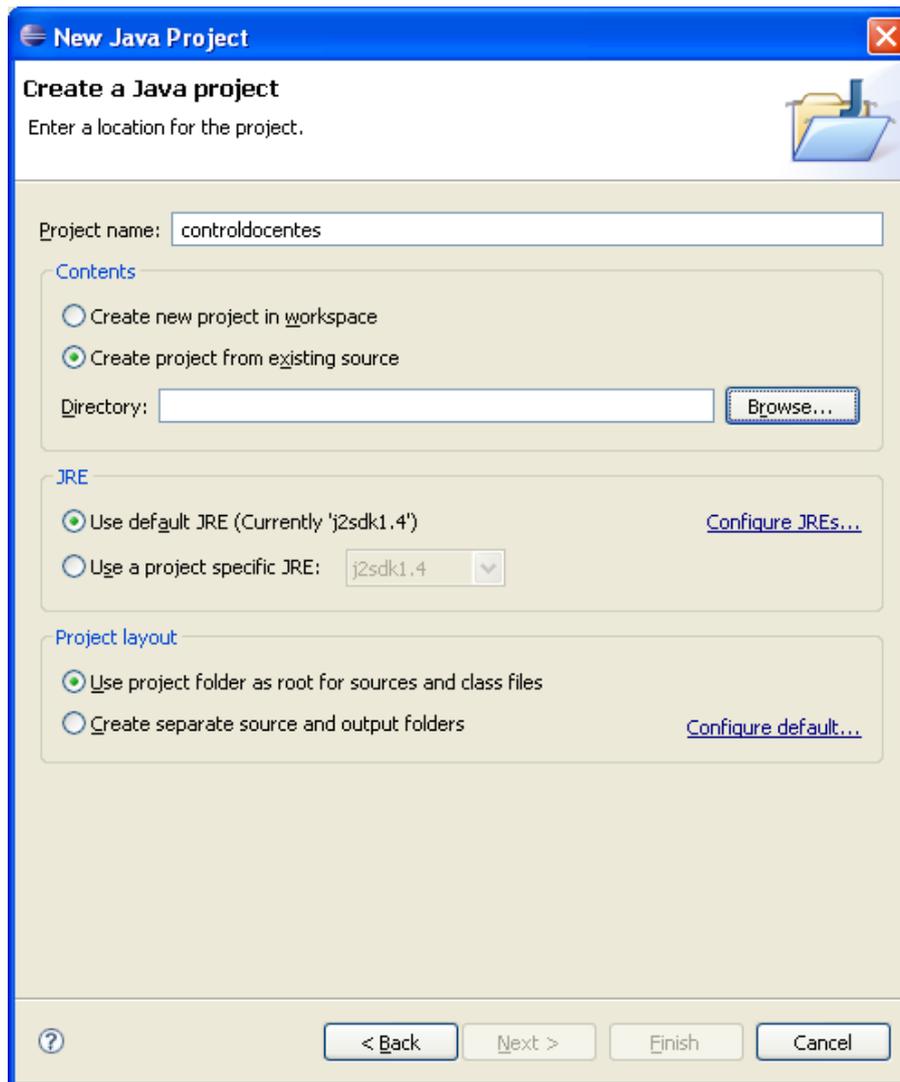


Figura 5.5. Configuración de Proyecto JAVA.

4.- Damos click sobre la opción next y aceptamos todas las políticas y configuraciones sugeridas por defecto, posteriormente las fuentes se importarán a nuestro espacio de trabajo y estarán listas para ser modificadas, compiladas y desplegadas.

Para realizar las tareas de compilación y despliegue el proyecto contiene dos archivos build.

- build.properties que contiene las opciones de despliegue que deberán contener la configuración del servidor Tomcat.

- build.xml que es el archivo ejecutable (ant) que realiza dichas acciones; para realizar el despliegue damos click derecho sobre este archivo y lo ejecutamos con la opción Ant Build. Las fuentes compiladas de la aplicación se colocan automáticamente en la carpeta de despliegue del servidor Tomcat.

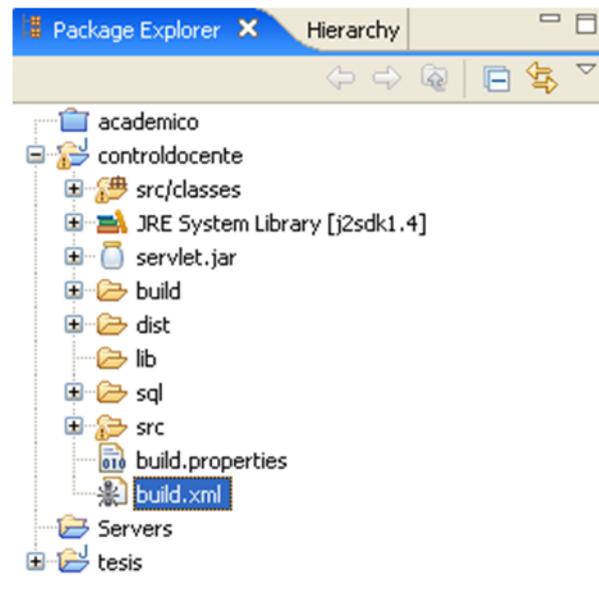


Figura 5.6. Proyecto JAVA importado en Espacio de Trabajo.

- **Base de Datos**

Para la instalación y configuración de la base de datos es necesario instalar y crear la base de datos necesaria para la ejecución del sistema. El servidor y el cliente de la base de dato Oracle 10 Express Edition esta incluido en el medio de almacenamiento adjunto así como también el script necesario para la creación de la tablas.

5.2- Pruebas

La metodología de desarrollo de software conceptualiza a la etapa de pruebas como el proceso que permite comprobar que el software realice correctamente las tareas indicadas en la especificación. Algo muy interesante es que recomienda como una técnica de prueba el probar por separado cada módulo del software, y luego probarlo de forma integral, para así llegar al objetivo, condición que se logra más fácilmente en la metodología de desarrollo por prototipos utilizada en el presente desarrollo.

Se considera una buena práctica el que las pruebas sean efectuadas por alguien distinto al desarrollador que la programó, idealmente un área de pruebas; sin perjuicio de lo anterior el programador debe hacer sus propias pruebas, este aspecto también está cubierto puesto que las revisiones de los prototipos se realizaron con los usuarios finales de la aplicación.

El procedimiento de pruebas del sistema se ha realizado de forma continua a lo largo del desarrollo del proyecto, al utilizarse la metodología de desarrollo por prototipos fue posible evaluar el desempeño y el cumplimiento de requerimientos en cada una de las revisiones periódicas de las versiones de la aplicación. Dichas revisiones se realizaron con algunos señores coordinadores y directores de departamento que como usuarios finales validaban de mejor manera la ejecución del software, además de ser un escenario propicio para la identificación de nuevos requerimientos y corrección de errores.

Otro punto a favor de la realización de las pruebas es que el director del presente proyecto de desarrollo de software el Ing. Jaime Mayla desempeña el rol del administrador del proceso de control de asistencia docente lo que ayudó

de manera significativa en el desarrollo de las pruebas e identificación de requerimientos.

Después de la entrega de la versión final de la aplicación se logró determinar que esta cumple con los requerimientos de funcionalidad especificados y que se encuentra lista para ser implementada en producción. Durante los primeros meses se realizarán pruebas piloto del software en donde se incluirán algunos departamentos de la institución para luego incluir a la totalidad de departamentos de la Escuela Politécnica del Ejército.

5.3- Manual de Usuario

5.3.1- Usuario General

Todos los usuarios independientemente del rol que desempeñen pueden realizar cuatro acciones básicas: autenticarse en el sistema, manejo de roles, cerrar sesión y dirigirse a la página principal de la institución.

- **Autenticación en el sistema.**

Al ingresar al sitio web para el control de asistencia docente la primera página que se desplegará es la de inicio de sesión, en esta el usuario deberá ingresar su número de identificación y su contraseña para acceder al sitio. En caso de que los datos no sean ingresados el sistema advertirá de la falta de información y no procederá a realizar la verificación, en caso de tener los datos completos el sistema procederá a verificar la veracidad de las credenciales ingresadas. Dependiendo de esta verificación se mostrará la página de bienvenida o la página de error de inicio de sesión.



Figura 5.7. Pantalla de Inicio de Sesión.

Error al autenticarse

El usuario o password ingresados son incorrectos.

Por favor intente nuevamente o comuníquese con el Administrador del sistema.

Comentarios y sugerencias: sisinfor@espe.edu.ec

[\[Sitio ESPE\]](#) [\[Cerrar Sesión\]](#)

©2002 Todos los contenidos son propiedad de la Escuela Politécnica del Ejército - ESPE

Optimizado para resolución de 800x600

Figura 5.8. Pantalla de Error en Inicio de Sesión.

[MENU PRINCIPAL](#)

Bienvenido

La Dirección de Organización y Sistemas ha diseñado este sitio pensando en sus necesidades - Esperamos que todas las opciones presentadas sean de su agrado y utilidad

Figura 5.9. Pantalla de Bienvenida.

- **Manejo de Roles**

Al ingresar al sistema se mostrará en la parte izquierda todos los roles que podrá desempeñar ese usuario, para acceder a los diferentes servicios que presenta cada uno de los roles es necesario dar click sobre el icono de cada rol y a continuación se desplegarán los servicios disponibles.



Figura 5.10. Lista de Roles para el usuario.

Bienvenido Sr./Srta. Francisco Naranjo

Seleccione una de las siguientes opciones:

- [Ingresar Reporte Docente](#)
- [Ver Reporte Docente](#)

Figura 5.11. Servicios para un rol determinado.

- **Cerrar Sesión e ir a Página Principal**

Todas las páginas del sistema presentan en la parte inferior dos links que permiten a los usuarios en cualquier momento finalizar su sesión o dirigirse a la página principal de la Escuela Politécnica del Ejército. Para realizar dichas acciones es necesario únicamente pulsar sobre estos links.

Figura 5.12. Cerrar Sesión ir a Página Principal

5.3.2- Usuario Docente

- **Ver Reporte Personal**

El usuario docente ingresa al sistema y al dar click sobre su rol se le presentará la pantalla con el servicio para ver su reporte personal.

MENU PRINCIPAL > Docente

Bienvenido Sr./Srta. Francisco Naranjo

Seleccione una de las siguientes opciones:

Ver Reporte Personal

Comentarios y sugerencias: sisinfor@espe.edu.ec

[\[Sitio ESPE\]](#) [\[Cerrar Sesión\]](#)

©2002 Todos los contenidos son propiedad de la Escuela Politécnica del Ejército - ESPE

Optimizado para resolución de 800x600

Figura 5.13. Servicios para el docente.

Al acceder a este servicio el usuario tendrá que elegir el periodo de asistencia del cual desea obtener la información.

Por Favor escoja el periodo:

NOVIEMBRE - DICIEMBRE
NOVIEMBRE - DICIEMBRE
OCTUBRE - NOVIEMBRE 2008
SEPTIEMBRE - OCTUBRE 2008
AGOSTO - SEPTIEMBRE 2008
JULIO - AGOSTO 2008
JUNIO - JULIO 2008
MAYO - JUNIO 2008
ABRIL - MAYO 2008
MARZO - ABRIL 2008

Comentarios y sugerencias: sisinfor@espe.edu.ec

©2002 Todos los contenidos son propiedad de la Escuela Politécnica del Ejército - ESPE

Optimizado para resolución de 800x600

Figura 5.14. Escoger periodo de asistencia.

Una vez escogido el periodo de asistencia se deberá seleccionar la modalidad de la cual se quiere obtener el reporte.

Escoja el reporte:

- * Presencial
- * Distancia

Comentarios y sugerencias: sisinfor@espe.edu.ec

[\[Sitio ESPE\]](#) [\[Cerrar Sesión\]](#)

©2002 Todos los contenidos son propiedad de la Escuela Politécnica del Ejército - ESPE

Optimizado para resolución de 800x600

Figura 5.15. Modalidades de Reportes.

Al seleccionar la modalidad se presentará un reporte de las asignaturas que dicta el docente, en esta pantalla existe la opción de ver un reporte resumido del periodo de asistencia.

MENU PRINCIPAL > Docente > Ver Reporte Personal

Reporte Resumido

Reporte Docente Dedicación Exclusiva																															
Periodo	NOVIEMBRE - DICIEMBRE																														
Docente	Luis Dias																														
Dias	20	21	22	23	24	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	
Materia	J	V	S	D	L	M	M	J	V	S	D	L	M	M	J	V	S	D	L	M	M	J	V	S	D	L	M	M	J	V	
Biología	0	2.0	0	0	2.0	2.0	0	0	2.0	0	0	2.0	2.0	0	0	2.0	0	0	2.0	0	0	2.0	0	0	2.0	0	0	2.0	0	0	2.0
Dentro Horario	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Fuera Horario	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Permiso Clase	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Faltas Injustificadas	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Faltas Justificadas	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Comision	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Figura 5.16. Reporte de Docente.

Reporte Docente Dedicación Exclusiva	
Periodo	NOVIEMBRE - DICIEMBRE
Docente	Luis Dias
Total Permiso Adm.	0
Total Injustificadas Adm.	0
Total Justificadas Adm.	0
Total Comision Adm.	0
Total Atraso Adm.	0
Total Horas Dictadas	34
Total Extra Dentro Horario	0
Total Extra Fuera Horario	0
Total Horas Administrativas	230
Total Permiso Clase	0
Total Faltas Injustificadas	0
Total Faltas Justificadas	0
Total Horas Comision	0
Total Atrasos Clase	0
Total Horas(+/-)	0

Figura 5.17. Reporte Resumido.

5.3.3- Usuario Administrador del Proceso de Control de Asistencia

- **Ver Reporte Docente**

El usuario administrador ingresa al sistema y al dar click sobre su rol se le presentará la pantalla con el servicio para ver reportes de los docentes.

Bienvenido Sr./Srta. Francisco Naranjo

Seleccione una de las siguientes opciones:

- [Ver Reporte Docente](#)

Comentarios y sugerencias: sisinfor@espe.edu.ec

[\[Sitio ESPE\]](#) [\[Cerrar Sesión\]](#)

©2002 Todos los contenidos son propiedad de la Escuela Politécnica del Ejército - ESPE

Optimizado para resolución de 800x600

Figura 5.18. Servicios para el administrador.

Al acceder a este servicio el usuario tendrá que elegir el periodo de asistencia del cual desea obtener la información.



Figura 5.19. Escoger periodo de asistencia.

A continuación deberá elegir el nombre del docente del cual quiere consultar la información.



Figura 5.20. Escoger docente.

Una vez escogido el docente se deberá seleccionar la modalidad de la cual se quiere obtener el reporte

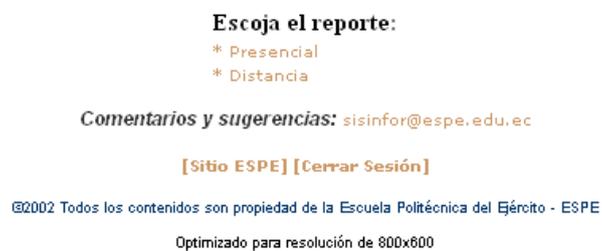


Figura 5.21. Modalidades de Reportes.

Al seleccionar la modalidad se presentará un reporte de las asignaturas que dicta el docente, en esta pantalla existe la opción de ver un reporte resumido del periodo de asistencia.

- **Asignación de Perfiles de Usuario**

El usuario administrador tiene la opción de asignar los perfiles correspondientes a los usuarios del sistema, al escoger este servicio se le presentará una interfaz en donde deberá ingresar el número de identificación del usuario deseado.

Busqueda de Usuarios

Identificación:	
-----------------	--

Aceptar

Figura 5.22. Identificar Usuario.

Una vez pulsado el botón, se visualizará una pantalla en donde se podrá confirmar la identidad del usuario seleccionado y se podrá asignar los roles deseados al mismo.

La interfaz contiene dos listas en donde se detallan tanto los roles disponibles dentro del sistema como los roles asignados a este usuario. Luego de haber asignado los roles deseados se continúa y se recibe el mensaje de confirmación, es necesario recalcar que se debe asignar al menos un rol para poder realizar la operación.

Identificación:	1714884770
Nombre:	FRANCISCO JAVIER
Apellido:	NARANJO MERIZALDE

Perfiles de Usuario

Disponibles	Asignados
Coordinador Docente	Administrador Director
<input type="button" value="-->"/> <input type="button" value="<--"/>	
<input type="button" value="Aceptar"/>	

Figura 5.23. Asignación de Roles.

Actualización de Usuario Correcta

Comentarios y sugerencias: sisinfor@espe.edu.ec

[\[Sitio ESPE\]](#) [\[Cerrar Sesión\]](#)

Figura 5.24. Confirmación de Asignación.

- **Ingreso de Calendario**

El administrador podrá ingresar nuevos calendarios para los períodos de control de asistencia docente, al hacer uso del servicio se accederá a una pantalla en donde deberá ingresar las fechas de inicio y fin del periodo, es necesario recordar que dichas fechas deben cumplir con ciertas características para que el ingreso de calendario se haga de manera correcta, el formato debe ser el especificado (dd/mm/aaaa), la fecha de fin no puede

ser menor a la de inicio y la fecha de inicio deberá ser mayor a las fechas de cualquier calendario existente.

Ingreso de Calendario

Fecha Inicio(dd/mm/aaaa)	<input type="text"/>
Fecha Fin(dd/mm/aaaa)	<input type="text"/>

Figura 5.25. Ingreso de Calendarios.

Si los datos cumplen con las consideraciones antes mencionadas se mostrará un mensaje confirmando el éxito de la operación.

Calendario insertado correctamente.

Comentarios y sugerencias: sisinfor@espe.edu.ec

[\[Sitio ESPE\]](#) [\[Cerrar Sesión\]](#)

Figura 5.26. Confirmación Calendarios.

- **Asignar Coordinador**

El usuario administrador posee un servicio que le permite asignar como coordinador de una carrera específica a un usuario en particular, al acceder a este servicio se presenta una interfaz en donde se deberá ingresar la identificación de la persona a ser asignada y la carrera de la cual es coordinador.

Una vez ingresado se presentará un mensaje confirmando la asignación realizada.

Favor Ingrese Coordinador

Cedula:

Carrera:

Figura 5.27. Ingreso Coordinador.

Coordinador Asignado Correctamente

Comentarios y sugerencias: sisinfor@espe.edu.ec

[\[Sitio ESPE\]](#) [\[Cerrar Sesión\]](#)

Figura 5.28. Asignación realizada.

5.3.4- Usuario Coordinador de Carrera

- **Ingresar Reporte Docente**

El usuario coordinador de carrera al seleccionar su rol, se le presentarán dos servicios, uno de estos es el de ingresar reporte docentes, una vez seleccionado este se tendrá que elegir la carrera de la cual desea ingresar la información.

Por Favor escoja el Departamento:

Comentarios y sugerencias: sisinfor@espe.edu.ec

[\[Sitio ESPE\]](#) [\[Cerrar Sesión\]](#)

©2002 Todos los contenidos son propiedad de la Escuela Politécnica del Ejército - ESPE
Optimizado para resolución de 800x600

Figura 5.29. Selección de Carrera.

Una vez seleccionada la carrera aparecerán las siguientes opciones:

1. Ingresar Características Horas Clase.
2. Ingresar Características.
3. Generar Reporte de Docentes.

Cada una de estas opciones presenta interfaces diferentes y serán explicadas a continuación.

Favor escoja la opcion a realizar

- [Ingresar Características Horas Clase](#)
- [Ingresar Características](#)
- [Generar reporte de docentes](#)

Comentarios y sugerencias: sisinfor@espe.edu.ec

[\[Sitio ESPE\]](#) [\[Cerrar Sesión\]](#)

©2002 Todos los contenidos son propiedad de la Escuela Politécnica del Ejército - ESPE

Optimizado para resolución de 800x600

Figura 5.30. Servicios de Ingreso de Reportes.

El servicio de Generar reporte de docentes ejecutará un proceso que genera el reporte general de asistencia de los docentes de esa carrera basándose en los horarios de clases del sistema académico de la Escuela Politécnica del Ejército, después de cumplir esta tarea presentará un informe de los reportes generados.

Informe de reporte Generado			
Docente	Luis Dias	Cedula	9999999999
Biologia			
Calculo			
Docente	Natalia Saenz	Cedula	1718869330
Calculo			

Comentarios y sugerencias: sisinfor@espe.edu.ec

[\[Sitio ESPE\]](#) [\[Cerrar Sesión\]](#)

©2002 Todos los contenidos son propiedad de la Escuela Politécnica del Ejército - ESPE

Optimizado para resolución de 800x600

Figura 5.31. Informe de Reportes Generados.

Al ejecutar el servicio de Ingresar Características Horas Clase se deberá escoger el docente del cual se desea ingresar alguna novedad.

Favor escoja un docente:

Carlos Dias	▼
Carlos Dias	
Luis Dias	
Natalia Saenz	

Comentarios y sugerencias: sisinfor@espe.edu.ec

[\[Sitio ESPE\]](#) [\[Cerrar Sesión\]](#)

©2002 Todos los contenidos son propiedad de la Escuela Politécnica del Ejército - ESPE

Optimizado para resolución de 800x600

Figura 5.32. Escoger Docente.

Después de seleccionar el docente se deberá elegir la materia de la cual se va a ingresar el reporte.

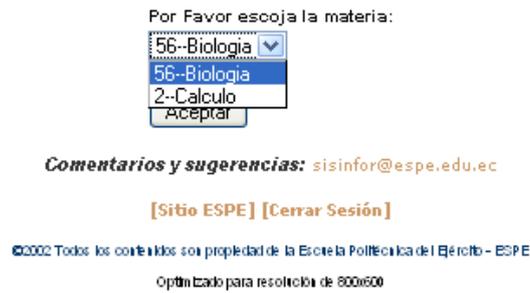


Figura 5.33. Escoger Materia.

Posteriormente se deberá elegir el día en el cual se presentó alguna novedad para poder ser ingresada.

Materia:	Calculo
Docente:	Luis Dias
Escoja el día que desea modificar	
Fecha de Clases	Revisado
Miercoles Nov 26 2008	✘
Miercoles Dec 03 2008	✘
Miercoles Dec 10 2008	✘
Miercoles Dec 17 2008	✘

Figura 5.34. Escoger Día de Clases.

Se mostrará una interfaz en donde se podrá ingresar la novedad existente en la hora de clases y al hacer click en Aceptar se visualizará un mensaje confirmando el éxito de la operación realizada.

Materia:	Calculo
Docente:	Luis Dias
Fecha:	Miercoles Dec 03 2008
Hora Inicio:	07:15:00
Hora Fin:	09:15:00

*** Características Horas**

- Permiso Clase
- Faltas Injust. Clase
- Faltas Just. Clase
- Comision Clase
- Atrasos Clase

Observaciones:

* # de Horas:

Figura 5.35. Ingreso de Características.

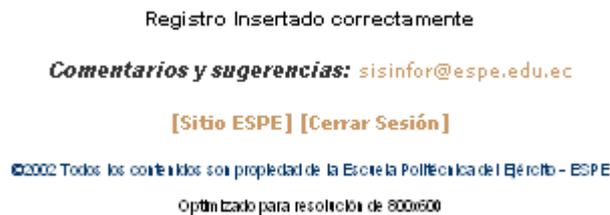


Figura 5.36. Confirmación de Éxito.

El servicio de Ingreso de características cumple la misma función del anterior sin embargo la manera de interactuar con el usuario es altamente dinámica y permite agilizar el proceso de ingreso de reportes.

En este existe una única pantalla en donde se escogerá el docente, la materia y se ingresarán las novedades correspondientes a cada día de clases.

Por Favor escoja el docente:
Luis Días

Por favor escoja la materia:
Biología
Calculo



Días de Clase	Inicio	Fin	Permiso	Falta Just.	Falta Intj.	Comisión	Atraso	Observaciones	# Horas
Miercoles Nov 26 2008	07:15:00	09:15:00	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		2
Miercoles Dec 03 2008	07:15:00	09:15:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		2
Miercoles Dec 10 2008	07:15:00	09:15:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		2
Miercoles Dec 17 2008	07:15:00	09:15:00	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		2

Figura 5.37. Interfaz Dinámica de Ingreso de Reportes.

- **Ver Reporte Docentes**

Al igual que el administrador el coordinador dispone de un servicio mediante el cual podrá consultar los reportes de los docentes que pertenecen a la carrera.

Al ingresar al servicio el usuario deberá seleccionar la carrera de la cual requiere información.

Por Favor escoja el Departamento:

SISTEMAS-PRESENCIAL	▼
SISTEMAS-PRESENCIAL	
ADMINISTRATIVAS-PRESENCIAL	
SEGURIDADES-MED	

Comentarios y sugerencias: sisinfor@espe.edu.ec

[\[Sitio ESPE\]](#) [\[Cerrar Sesión\]](#)

©2002 Todos los contenidos son propiedad de la Escuela Politécnica del Ejército - ESPE
Optimizado para resolución de 800x600

Figura 5.38. Selección de Carrera.

Después deberá seleccionar el docente del cual quiere visualizar el reporte.



Figura 5.39. Selección de Docente.

Al igual que para el administrador y el docente, se visualizará el reporte de asistencia correspondiente con la opción de ver la versión resumida del mismo.

5.3.5- Usuario Director de Departamento

- **Ver Reporte Docente**

Los pasos que le permiten al director visualizar el reporte de asistencia de los docentes de el departamento del cual es director inician escogiendo el departamento.

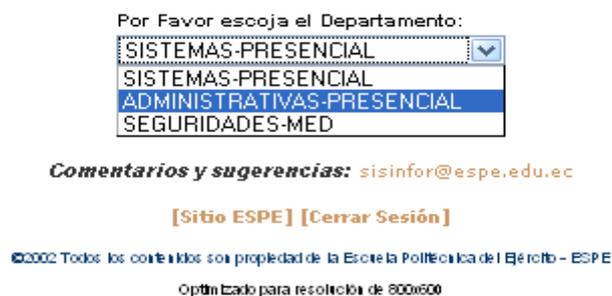


Figura 5.40. Selección de Departamento.

Después deberá seleccionar el docente del cual quiere visualizar el reporte.



Figura 5.41. Selección de Docente.

Al igual que para los otros casos, se visualizará el reporte de asistencia correspondiente con la opción de ver la versión resumida del mismo.

- **Ingresar Reporte Docente**

El usuario director de departamento al seleccionar su rol, se le presentarán dos servicios, uno de estos es el de ingresar reporte docentes, una vez seleccionado este se tendrá que elegir la carrera de la cual desea ingresar la información.

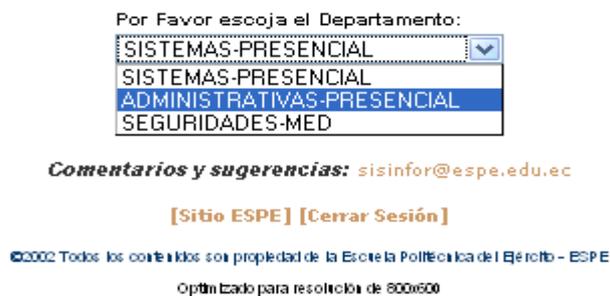


Figura 5.42. Selección de Carrera.

Una vez seleccionada la carrera aparecerán las siguientes opciones:

1. Ingresar Características Administrativas.
2. Ingresar Horas Adicionales.
3. Aumentar o Quitar Horas.

Cada una de estas opciones presenta interfaces diferentes y serán explicadas a continuación.



Figura 5.43. Servicios de Ingreso de Reportes.

Al pulsar sobre el servicio de Ingreso de Características Administrativas en primer lugar se deberá elegir el docente.

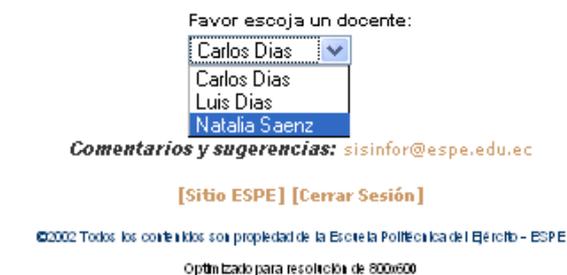


Figura 5.44. Selección de Docente.

A continuación se deberá elegir el día del calendario de asistencia en el cual se desea agregar alguna novedad.

Docente:	Luis Dias
Escoja el día que desea modificar	
Fecha de Clases	
Jueves Nov 20 2008	
Viernes Nov 21 2008	
Lunes Nov 24 2008	
Martes Nov 25 2008	
Miercoles Nov 26 2008	
Jueves Nov 27 2008	
Viernes Nov 28 2008	
Lunes Dec 01 2008	
Martes Dec 02 2008	
Miercoles Dec 03 2008	
Jueves Dec 04 2008	
Viernes Dec 05 2008	
Lunes Dec 08 2008	
Martes Dec 09 2008	

Figura 5.45. Lista Días de Asistencia.

Después se mostrará una interfaz en donde se podrán ingresar las características correspondientes al día seleccionado, una vez finalizado se pulsa el botón aceptar y se obtiene un mensaje confirmando el éxito de la operación realizada.

Docente:	Luis Dias
Fecha:	Jueves Nov 20 2008
Características Horas	
Permiso Administrativo	<input type="radio"/>
Faltas Injust. Administrativo	<input type="radio"/>
Faltas Just. Administrativo	<input checked="" type="radio"/>
Comision Administrativo	<input type="radio"/>
Atrasos Administrativo	<input type="radio"/>
Observaciones:	<input type="text"/>
* # de Horas:	<input type="text"/>
<input type="button" value="Aceptar"/>	

Figura 5.46. Ingreso Características Administrativas.

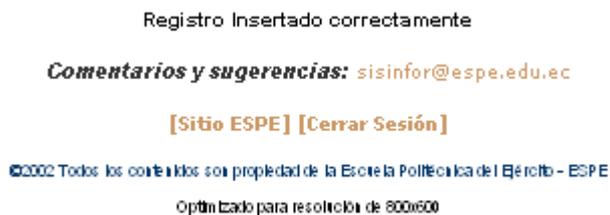


Figura 5.47. Confirmación de Éxito.

El servicio de ingreso de horas adicionales comienza por escoger el docente del cual se desea ingresar dicha característica. Una vez escogido este se procede a escoger la materia y se presenta la interfaz en donde de escogerá el día de clases para el cual se requiera esta particularidad.

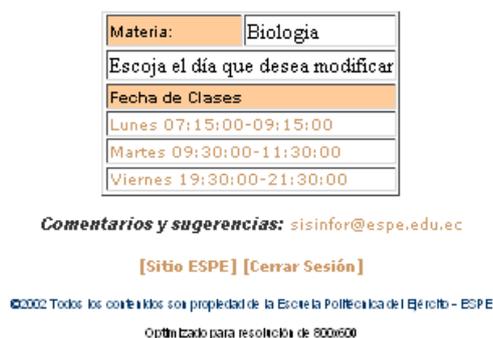


Figura 5.48. Lista Días de Clase Semanal.

Una vez escogido el día se presenta la interfaz que permite ingresar las propiedades correspondientes.

Materia:	Biología
Docente:	Luis Dias
Fecha:	Martes
Hora Inicio:	09:30:00
Hora Fin:	11:30:00

Horas Extra

Hora Extra Dentro de Horario

Hora Extra Fuera de Horario

de Horas Extra

Observaciones:

Figura 5.49. Ingreso Horas Adicionales.

Después de registrar las propiedades se pulsa el botón aceptar y se muestra un mensaje confirmando el éxito de la operación.

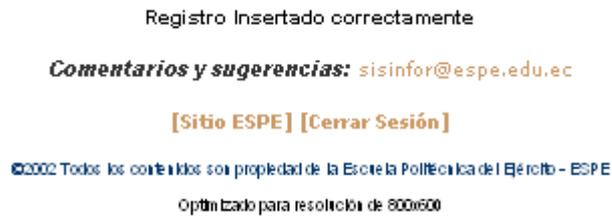


Figura 5.50. Confirmación de Éxito.

Por ultimo esta el servicio de Aumentar o Quitar Horas, en donde en un inicio se escoge el docente y después se muestra la interfaz que permite ingresar los valores correspondientes. Después de ingresar los mismos se procede con el proceso y el sistema muestra el mensaje confirmando el éxito de la operación.

Docente:	Luis Dias
Elija la opción que desea:	
Añadir	<input type="radio"/>
Disminuir	<input type="radio"/>
# Horas	<input type="text"/>
Observaciones	<input type="text"/>
<input type="button" value="Aceptar"/>	

Figura 5.51. Aumentar o Quitar Horas.

CAPITULO 6

Conclusiones y Recomendaciones

6.1- Conclusiones

El presente proyecto de grado buscó el desarrollo e implementación de un módulo que permita la automatización del proceso de control de asistencia del personal docente de la Escuela Politécnica del Ejercito mediante la elaboración de un sistema orientado a la web que se integraría al actual sistema académico de la institución y contribuiría a mejorar el intercambio de información haciendo de este un proceso mas eficiente. Para lograr el desarrollo de esta aplicación fue necesario elaborar un proyecto de desarrollo que comprendió cada una de las etapas del ciclo de vida del software.

La obtención de requerimientos se realizó mediante entrevistas con los usuarios directos del sistema informático, debido a que los requerimientos no estaban completamente definidos desde un inicio se aplicó la metodología de desarrollo por prototipos la misma que ayudó a ir identificando requerimientos periódicamente. El análisis de requerimientos permitió explicar el propósito y funcionalidades del sistema, que es lo que sistema realizará y las directrices bajo las cuales debía operar y como debía responder a estímulos externos que interactúan con el mismo además de proporcionar la base tanto a los usuarios finales como al desarrollador para tener una idea más clara del software y sus funcionalidades.

Como resultado de este análisis se elaboró y diseño los componentes que debían pertenecer al sistema así como se definió las funcionalidades que debía tener el mismo, se especificó la metodología que debía ser utilizada, se

estableció la arquitectura y el lenguaje que mejor se adaptarían a los requerimientos obtenidos con la finalidad de que el proceso de desarrollo y codificación se faciliten y se obtenga como resultado un software de mejor calidad.

Uno de los aspectos más relevantes del presente sistema informático es la integración del mismo con el actual sistema académico de la institución, la información generada y manejada por el sistema académico como son las materias, horarios y docentes es de mucha importancia para el funcionamiento del sistema de control docente desarrollado en este proyecto, dicha integración se realizó exitosamente mediante la utilización de enlaces hacia los repositorios de información del sistema académico. Es justamente esta funcionalidad la que permite al sistema mejorar el intercambio de información entre las dependencias de la Escuela Politécnica del Ejército logrando así a ser de este proceso un procedimiento más dinámico y eficiente.

El proceso de pruebas de la aplicación se fue realizando de forma continua a lo largo del desarrollo del proyecto, como fue expuesto anteriormente la utilización de la metodología de desarrollo por prototipos permitió evaluar el desempeño y el cumplimiento de requerimientos en cada una de las revisiones periódicas de las versiones de la aplicación, estas revisiones contaron con la valiosa ayuda de los señores coordinadores de carrera y directores de departamento que al ser usuarios finales de la aplicación validaban de mejor manera la ejecución del software logrando así un producto final de mayor calidad y mas alineado a cumplir con los requerimientos del usuario.

La implementación del modulo desarrollado con el actual sistema de tecnología de información de la institución fue llevado acabo en trabajo conjunto con la unidad de TIC's y la unidad de talento humano-docente, el hecho de que el director y codirector de este proyecto de grado pertenezcan a las mencionadas áreas facilitó la elaboración de esta tarea y jugo un papel de altísima importancia en el éxito de todo el desarrollo.

Por último, se concluye que la elaboración de un proyecto completo de ingeniería de software ayuda significativamente a entender y visualizar de mejor manera el proceso macro de desarrollo de software, es justamente al realizar proyectos de esta naturaleza que se evidencia la importancia de la utilización de una metodología y se logra entender la gran trascendencia de cada uno de los pasos en el desarrollo de aplicaciones informáticas. Sin embargo luego de haber concluido el proyecto hace falta resaltar que la etapa que sin lugar a dudas es la base de la cual depende el éxito a fracaso de un proyecto de software es la obtención y análisis de requerimientos, puesto que la tecnología o técnicas utilizadas para solventar estas funcionalidades a pesar de ser de gran importancia y de ser utilizada de la mejor manera posible, si se basan en funcionalidades o requerimientos mal definidos jamás lograrán obtener un software de calidad y peor aun uno que cumpla con las expectativas del usuario.

6.2- Recomendaciones

Se recomienda que para garantizar un correcto funcionamiento del sistema realizado los datos que este necesita obtener del sistema académico de la institución posean información completa y fiable, puesto que si la información entregada por el sistema académico esta incompleta o presenta errores los resultados que presente el software de control de asistencia para los docentes no serán los esperados.

Al ser este un sistema para el control de la actividad laboral de los docentes dentro de al Escuela Politécnica del Ejército, se recomienda además que el funcionamiento del mismo se encuentre respaldado por las respectivas normas y reglamentos que avalen los resultados que este entregue.

La información que genere este sistema podrá ser usada para la elaboración de futuros proyectos, una opción interesante sería la de diseñar un proyecto de Data Ware Housing que permita llevar un control estadístico y más detallado de la actividad laboral de los docentes.

Luego de la elaboración del presente proyecto se han identificado ciertas mejoras o recomendaciones para el manejo global de la infraestructura de tecnología de información que se utiliza en la Escuela Politécnica del Ejército.

Se identificó que la información acerca de los docentes se encuentra distribuida y duplicada tanto en la base de datos del sistema académico como en la base de datos del sistema administrativo, esto como es obvio causa una falta de sincronización entre la información que existe en ambos repositorios de información, una posible solución a esta problemática sería la de unificar la información en un único repositorio, otra alternativa es la de implementar

diferentes métodos como la utilización de store procedures o triggers para garantizar la sincronización de información entre dichos repositorios de almacenamiento.

Se evidenció que la definición de los procesos de negocio de la institución no se encuentra plenamente identificada ni documentada, es por esto que la realización de un proyecto global de ingeniería de procesos de negocio sería fundamental, un proyecto de estas características facilitaría la automatización de los procesos en el futuro así como también ayudaría a la optimización de los mismos. La implementación de proyecto de estas características sería una estrategia importante para la consecución de metas y objetivos planteados a futuro por la Escuela Politécnica del Ejército.

Actualmente la tendencia de desarrollo está enfocada a la reutilización de componentes e integración de sistemas de información, la arquitectura SOA (Service Orientated Architecture) se basa en la definición de servicios que satisfagan las necesidades del negocio para implementarlos mediante la utilización de servicios web, estos servicios pasan a formar parte de un servidor de aplicaciones y pueden ser utilizados por cualquier aplicación, un ejemplo sería un servicio que obtenga la información de un usuario basándose en el número de identificación. Por lo que se aconseja la realización de un proyecto que permita identificar los servicios que requiere el negocio para gestionar adecuadamente el sistema integrado de información de la institución e implementar una solución SOA que permita obtener un conjunto de funcionalidades mucho más flexibles, independientes y reutilizables.

En los últimos años la Escuela Politécnica del Ejército ha realizado modificaciones en cuanto a la estructuración de la misma, la inclusión de la división por departamentos es un ejemplo claro de la tendencia y enfoque al que se está direccionando la institución. Al ser esta una organización como es obvio su funcionamiento está basado en procesos de negocio. Actualmente existen nuevas tecnologías enfocadas a la administración de procesos de negocio, el ejemplo más claro de esto es la tecnología BPM (Business Process Management), la misma que se encarga del modelado, implementación y monitoreo de los procesos de negocio de una institución. La implementación de un proyecto de BPM en la institución sería un proyecto altamente ambicioso y requeriría de esfuerzos muy grandes, por lo que se recomienda el análisis pertinente para contemplar la posibilidad de realizarlo ya que los beneficios obtenidos sin lugar a dudas colocarían a la organización en una posición de vanguardia y dotada de la última tecnología.

En cuanto al Departamento de Ciencias de la computación sería de gran utilidad incluir en su plan de estudios a la arquitectura orientada a servicios en donde se haga énfasis en conceptos como XML²² y las maneras de manejarlo, servicios web, ejb's, servidores de aplicaciones y protocolo de comunicaciones SOAP, este tipo de conocimiento será de gran ayuda al momento de implementar nuevas soluciones de aplicaciones informáticas y permitirá a los estudiantes estar a la vanguardia del conocimiento en lo que respecta al desarrollo de software.

²² Extensible Markup Language

BIBLIOGRAFIA

Libros

Roger S. Pressman. (2002). Ingeniería de Software, un enfoque práctico. Quinta Edición. Madrid. España.

Martin Fowler. (2000). UML Distilled. Segunda edición. Estados Unidos. Addison Wesley.

Martin Robert Cecil. (2003). UML for Java Programmers. Segunda edición. Estados Unidos. Valley Stream.

Duffey Goyal. (2001). Professional JSP Site Design. Primera Edición. Estados Unidos. Wrox

Penker Magnus; Hans-Erik Eriksson (2000). Business Modeling with UML. Tercera Edición. Estados Unidos. Cambridge University Press.

Netscape Communications Corporation (1999). Client Side Javascript reference. Cuarta Edición. Estados Unidos. Mountain View.

Web

<http://jakarta.apache.org/tomcat/index.html> Servidor de Aplicaciones Jakarta Tomcat

http://w3.tap.ibm.com/medialibrary/media_set_view?id=5616 Business Process Management

<http://www.java.sun.com/products/jsp/>

JSP Technology Fundamentals

<https://www-01.ibm.com/software/ucd/designconcepts.html>
Concepts

Design

www.mc-store.com/5213.html
Development

An Introduction to Web Application

www-01.ibm.com/software/webservers/appserv/was/
WebSphere Application Server

IBM -

Nombre: Francisco Javier Naranjo Merizalde

Lugar y Fecha de Nacimiento: Quito, 6 de Diciembre de 1983

ESTUDIOS REALIZADOS

- **Primarios:** Colegio Intisana 1990-1996.
Obtención de diploma de honor por rendimiento académico.
- **Secundarios:** Colegio Intisana 1996-2002.
Título: Bachiller en ciencias, especialización Físico-Matemáticas.
Obtención de diploma de honor por rendimiento académico.
- **Superiores: Escuela Politécnica del Ejército**
Departamento Ciencias de la Computación Feb 2003-Feb 2008
Beca por rendimiento académico.

HOJA DE LEGALIZACION DE FIRMAS

ELABORADA(O) POR

Francisco Javier Naranjo Merizalde

COORDINADOR DE LA CARRERA

Ing. Danilo Martínez.

Lugar y fecha: _____