



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE INGENIERÍA

**DIRECTORIO DE PERSONAS E INSTITUCIONES,
INSTITUTO MEXICANO
DE TECNOLOGÍA DEL AGUA**

INFORME DE ACTIVIDADES

**QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN**

PRESENTA:

CARLOS ROBERTO BOOL RUIZ

DIRECTOR DE TESIS:

ING. ALBERTO GONZÁLEZ GUÍZAR

CIUDAD UNIVERSITARIA AGOSTO 2010



*A mi madre, mi padre, mi abuelita y mi hermano
por su amor y el apoyo que me brindan incondicionalmente todos los días.*

*A la Universidad y en especial a la Facultad de Ingeniería
por la enseñanza y valores que aprendí en sus aulas.*

ÍNDICE

OBJETIVO.....	1
ANTECEDENTES DEL TEMA.....	2
DEFINICIÓN DEL PROBLEMA O CONTEXTO DE LA PARTICIPACIÓN PROFESIONAL	4
MARCO TEÓRICO.....	5
Software.....	5
Proceso de Software.....	5
Comunicación.....	6
Planeación.....	6
Modelado.....	7
Construcción.....	8
ANÁLISIS Y METODOLOGÍA EMPLEADA.....	11
Visión General del Sistema.....	12
Requerimientos Significativos.....	14
Panorama de la Solución.....	23
Patrón Arquitectónico Modelo Vista Controlador.....	24
Tecnología.....	25
Arquitectura.....	27
Vista Conceptual.....	28
Catálogo de Elementos.....	29
Vista de Módulos.....	32
Vista de Datos.....	37
Vista de Código.....	40
Vista de Ejecución.....	42
PARTICIPACIÓN PROFESIONAL.....	43
Fase de Inicio.....	43
Fase de Requerimientos.....	43
Fase de Análisis y Diseño.....	44
Fase de Construcción.....	44
Fase de Integración.....	45
Fase de Pruebas.....	45
Fase de Cierre.....	45
RESULTADOS Y APORTACIONES.....	47
CONCLUSIONES.....	49
REFERENCIAS	50

OBJETIVO

Desarrollar un sistema de información para la administración y consulta del directorio de los clientes, profesores, investigadores, proveedores, asesores, empleados, autores, dictaminadores y suscriptores de la Revista de Ingeniería (stakeholders) que permitiera centralizar y controlar el acceso a la información. Así como el envío de correo electrónico a los stakeholders seleccionados mediante la aplicación.

ANTECEDENTES DEL TEMA

Las Tecnologías de Información y Comunicación (TIC) se han convertido en elementos fundamentales para incrementar la efectividad operativa de las organizaciones, crear nuevas formas de hacer negocio, y en varias industrias permite impulsar de manera significativa sus ventajas competitivas. Para hacer frente a esta evolución vertiginosa, las organizaciones deben considerar una alineación entre sus estrategias y objetivos de negocio y las TIC.

En este orden de ideas, la Universidad Nacional Autónoma de México (UNAM) cuenta dentro de su estructura con la Dirección General de Servicios de Cómputo Académico (DGSCA), entidad encargada de la operación de los sistemas centrales de cómputo académico y de las telecomunicaciones de la institución, así como con el Centro Regional de Investigaciones Multidisciplinarias (CRIM), que tiene como objetivo primordial realizar investigaciones sobre los problemas sociales de México y sus regiones, mediante estrategias de investigación que incorporen las perspectivas de las ciencias sociales y de otras disciplinas

La UNAM responde de forma eficiente a su compromiso en la formación de recursos humanos en TIC, prospección y asimilación de estas tecnologías en beneficio de la Universidad y de la sociedad en general, ofreciendo diversos servicios a través de sus diferentes áreas, entre éstos se encuentran:

- Asesoría en el uso de las tecnologías aplicadas a la educación.
- Asesorías en Seguridad Informática.
- Asesorías en Tecnologías de la Información.
- Creación y mejora de Portales, Sitios Web y Sistemas Seguros en Internet.
- Desarrollo de cursos y materiales didácticos, y capacitación en tecnologías para robustecer la enseñanza presencial, continua y a distancia.
- Educación en Cómputo, proporcionando cursos básicos, avanzados y de alta especialización en diferentes técnicas, lenguajes y programas.
- Ingeniería de Software.
- Proyectos integrales de publicaciones digitales.

Por su parte, el Instituto Mexicano de Tecnología del Agua (IMTA) es un organismo público descentralizado del gobierno federal, con personalidad jurídica y patrimonio propios, coordinado sectorialmente por la Secretaría de Medio Ambiente y Recursos Naturales (SEMARNAT), cuya misión es “producir y diseminar conocimiento y tecnología para el aprovechamiento sustentable de los recursos hídricos de México y su gestión integral”.

En respuesta a los requerimientos que expresó el IMTA a la UNAM y su interés de colaborar conjuntamente en la realización de un Directorio de Personas e Instituciones, se presentó a su consideración, un esquema de colaboración entre ambas instituciones que permitiera optimizar los recursos y crear una relación de valor a largo plazo para ambas partes.

Bajo el importante reto de generar una solución de valor para el IMTA, la UNAM desarrolló una propuesta para el Sistema Informático del Directorio de Personas e Instituciones, la cual fue recibida, revisada y aprobada por el Instituto dando así inicio al desarrollo del sistema.

Al final del desarrollo la UNAM, a través de la DGSCA, hizo entrega del código fuente del sistema junto con la implantación del mismo en los servidores del Instituto. Además se capacitó al personal de sistemas acerca de los detalles técnicos empleados para el diseño,

desarrollo e implantación del sistema con el objetivo de que el instituto pueda dar el mantenimiento necesario al sistema. Adicionalmente se capacitó a los responsables de las diferentes áreas del instituto acerca del uso del sistema junto con la entrega de los manuales y documentación correspondiente.

DEFINICIÓN DEL PROBLEMA O CONTEXTO DE LA PARTICIPACIÓN PROFESIONAL

El proyecto consistió en el desarrollo de un sistema que cumpliera con las necesidades y prioridades del Instituto Mexicano de Tecnología del Agua (IMTA). Dentro del alcance propuesto se realizaron las siguientes actividades:

- Levantamiento de requerimientos y su documentación.
- Análisis de la información contenida en los repositorios del instituto.
- Análisis, diseño y desarrollo de la aplicación del Directorio de Personas e Instituciones.
 - Módulo de autenticación.
 - Módulo de administración de usuarios, grupos, roles y permisos.
 - Módulo de consultas por diferentes criterios.
 - Envío de correos.
 - Módulo de impresión personalizada.
 - Módulo de exportación de la información.
- Elaboración de los manuales de usuario y técnico.
- Implantación del sistema en el Instituto

MARCO TEÓRICO

En la actualidad, el software de computadora es la tecnología individual más importante en el ámbito mundial. Nadie en la década de 1950 podría haber predicho que el software se convertiría en una tecnología indispensable en los negocios, la ciencia y la ingeniería; tampoco que el software permitiría la creación de nuevas tecnologías, la expansión de tecnologías existentes, el fin de tecnologías antiguas; que una compañía de software se volvería muy grande y más influyente que la mayoría de las compañías de la era industrial; que una gran red construida con software llamada internet cubriría y cambiaría todo, desde la investigación hasta las compras de los consumidores y los hábitos diarios de los jóvenes y no tan jóvenes.

Software

El software de computadora es el producto que los ingenieros de software construyen y después mantienen en el largo plazo. Incluye los programas que se ejecutan dentro de una computadora de cualquier tamaño y arquitectura, el contenido que se presenta conforme los programas que se ejecutan y los documentos que engloban todas las formas de medios electrónicos.

El software de computadora se construye de la misma forma que cualquier producto de éxito: mediante la aplicación de un proceso que conduzca a un resultado de alta calidad que satisfaga las necesidades de la gente que usará el producto. Se aplica un enfoque de ingeniería de software.

Proceso de Software

Desde un punto de vista técnico, un proceso de software se define como un marco de trabajo para las tareas que se requieren en la construcción de software de alta calidad.

El proceso define un marco de trabajo que debe establecerse para la entrega efectiva del sistema. El proceso de software forma la base para el control de la gestión de proyectos de software y establece el contexto en el cual se aplican los métodos técnicos, se generan los productos de trabajo (modelos, documentos, datos, reportes, formatos, etc.), se establecen los fundamentos, se asegura la calidad y el cambio se maneja de manera apropiada.

El siguiente marco de trabajo genérico se puede aplicar en la inmensa mayoría de los proyectos de software:

- *Comunicación.* Esta actividad implica una intensa colaboración y comunicación con los clientes; además, abarca la investigación de requerimientos y otras actividades relacionadas.
- *Planeación.* Esta actividad establece un plan de trabajo. Describe las tareas técnicas que deben realizarse, los riesgos probables, los recursos que serán requeridos, los productos de trabajo y un programa de trabajo.

- *Modelado.* Esta actividad abarca la creación de modelos que permiten al desarrollador y al cliente entender mejor los requerimientos del software y el diseño para lograr satisfacerlos.
- *Construcción.* Esta actividad combina la generación del código y la realización de pruebas necesarias para descubrir errores en el código.
- *Despliegue.* El software (como una entidad completa o un incremento completado de manera parcial) se entrega al cliente, quien evalúa el producto recibido y proporciona información basada en su evaluación.

Comunicación

La administración de requerimientos, de acuerdo a CMM, abarca actividades como la recopilación, documentación, validación y control de los requerimientos y sus cambios. UML, como estándar integrador de las buenas prácticas de desarrollo ofrece en este sentido los casos de uso, como una técnica excelente para administrar los requerimientos de un proyecto.

Para desarrollar el mejor sistema posible se debe realizar un trabajo serio para identificar, en primer lugar, cuál es el valor que el sistema debe proporcionar al negocio, para lo cual habrá que preguntárselo a las personas que obtendrán alguna clase de beneficio cuando se ponga en operación. Una buena parte de estas personas probablemente vayan a ser usuarios del sistema; en UML se les conoce como Actores.

Una vez identificados los usuarios del sistema habrá que preguntarles en que situaciones valdrá la pena para ellos utilizarlo. La lista identificada de aquellas situaciones no debe de tener pequeñas funciones, sino flujos completos que le proporcionen suficiente valor tanto a ellos como al negocio, de manera que valga la pena usar el sistema en dichas situaciones. Ejemplo: un vendedor en cierto sistema de ventas querrá registrar venta, registrar pedido, consultar comisiones, administrar prospectos, generar factura y administrar clientes. Todas ellas son ejemplos de situaciones u objetivos importantes que podría necesitar un vendedor para llevar a cabo su trabajo, conocidas y modeladas como casos de uso en UML.

Normalmente los casos de uso son iniciados por algún actor. Lo inicia con algún evento, que podría ser tan simple como elegir una opción en el sistema, y continúa como una serie de eventos o interacciones entre actores y sistema, hasta que el objetivo del caso de uso se cumple. Lo que se logra así es una especificación de requerimientos funcionales mediante un análisis top-down (de lo general a lo particular). Esto nos lleva a identificar requerimientos realmente relevantes para alcanzar la misión del sistema.

Planeación

El objetivo de la planeación del proyecto de software es proporcionar un marco de trabajo que permita al administrador hacer estimaciones razonables de recursos costos y planificación temporal. Estas estimaciones se hacen dentro de un marco de tiempo limitado al comienzo de un proyecto de software, y deberían actualizarse regularmente medida que progresa el proyecto. Además las estimaciones deberían definir los escenarios del mejor caso, y peor caso, de modo que los resultados del proyecto pueden limitarse.

Modelado

El modelado visual es una práctica muy común en el desarrollo de software moderno. A través del modelado se puede visualizar cómo se quiere que sea un sistema antes de construirlo. Los modelos nos permiten especificar la estructura y comportamiento de un sistema y comunicarla al resto de los miembros del equipo.

El Lenguaje de Modelado Unificado (UML) es el lenguaje estándar para modelar sistemas de información, fue creado por Grady Booch, Ivar Jacobson y James Rumbaugh de Rational Software a partir de notaciones y técnicas que cada uno había creado anteriormente. En 1996 se publicó la versión 1.0 y se donó a la Object Management Group (OMG) para que administrara y coordinara su evolución. La manera más sencilla de definir UML, es como un lenguaje que define nueve diferentes tipos de diagramas. Estos diagramas se dividen en dos categorías: estáticos y dinámicos:

Diagramas Estáticos

Los diagramas estáticos son aquellos que permiten modelar la estructura de un sistema. Éstos son: casos de uso, clases, objetos, componentes y distribución.

- *Casos de uso.* Este diagrama se utiliza para representar la funcionalidad que un sistema ofrece a sus actores. Un actor es cualquier agente externo que interactúa con un sistema, como un humano, un dispositivo u otro sistema.
- *Objetos.* El diagrama de objetos es como el de clases, sólo que representa objetos en lugar de clases. Sucede que hay casos en los que las clases tienen relaciones complicadas, y a veces es útil apoyarse en diagramas de objetos para ilustrar casos específicos. O sea que los diagramas de objetos se utilizan para ejemplificar los diagramas de clases.
- *Componentes.* En éstos se presenta la organización de un sistema en componentes, y las dependencias entre éstos. Mientras que una clase es algo imaginario, un componente ya es algo más real, como un programa o un archivo.
- *Distribución.* Los diagramas de distribución o despliegue representan la arquitectura física de un sistema. Esto es, sus nodos (servidores, clientes, dispositivos, etc.) y conexiones.

Diagramas Dinámicos

Los diagramas dinámicos se enfocan en modelar el comportamiento. Éstos son: colaboración, secuencia, estado y actividad.

- *Colaboración.* Estos diagramas muestran la interacción (en forma de mensajes intercambiados) que se da entre objetos para ejecutar algún escenario específico.
- *Secuencia.* Aunque en esencia los diagramas de colaboración representan la misma información que los de secuencia, estos últimos reflejan el orden de las operaciones

y también pueden mostrar información adicional como la creación y destrucción de objetos.

- *Estado*. Estos diagramas especifican la secuencia de estados por los cuales puede pasar un objeto durante su ciclo de vida y en respuesta a eventos.
- *Actividad*. Los diagramas de actividad se utilizan para modelos flujos de trabajo.

Elementos Adicionales

Además de los diagramas y sus elementos, UML define algunos mecanismos auxiliares para organización y extensión. Los más comunes son:

- Paquetes. Un paquete es un mecanismo para agrupar y organizar los elementos de un modelo.
- Notas. Permiten incluir comentarios sobre uno o más elementos.
- Estereotipos. Este es un mecanismo de extensión, que permite crear nuevos tipos de elementos a partir de elementos existentes. Estos nuevos tipos pueden estar personalizados o restringidos a algún dominio específico.

Construcción

Una realidad que no se puede ocultar es el hecho de que el desarrollo y la codificación de software gozan de complejidad. A pesar de su relativa sencillez, resulta difícil entender todo ese grupo de tecnologías existentes para el desarrollo de sistemas. Por ello, se han diseñado estrategias basadas en capas conceptuales cuyo objetivo es el designar un orden lógico a cada elemento haciendo más claro el desarrollo.

Una de las más populares y difundidas en el mundo del software utiliza cinco capas: Presentación, Aplicación, Servicios, Dominio y Persistencia. Algunas de estas capas pueden no estar presentes en un sistema. Dependiendo de las características particulares de éste (el dominio del problema), de los marcos de trabajo que se empleen y de la estrategia con que es abordado el desarrollo, alguna de estas capas no serán codificadas.

Presentación

Compuesta por todos los elementos relacionados con la interfaz de usuario, tales como botones, ventanas, campos de textos, etiquetas, etc., si hablamos de clientes Win32, AWT, SWT o Swing; HTML, páginas dinámicas, JavaScript, aplicaciones Macromedia Flash si se tiene a un navegador Web por cliente; Xlets o Midlets, y sus correspondientes controles de interfaz de usuario si se trata de una aplicación inalámbrica. Ésta es la capa de interfaz de usuario, el rostro visible y, en cierta forma, estético del sistema. Sin lugar a dudas, la parte más importante para el usuario final del sistema.

Aplicación

Responsable del control de flujo de trabajo del lado del cliente y de componentes de interfaz de usuario reutilizables. Éstos se caracterizan por presentar un acoplamiento con la capa inmediata siguiente (servicios). Para el caso del control del flujo de trabajo, se puede emplear sesiones o algún marco de trabajo. En el caso de componentes, por ejemplo, se puede tener un control heredado de un "Combo Box" que despliega los nombres de clientes que están almacenados en la base de datos, y estos son obtenidos a través de un método de un componente de la capa de servicios. Dependiendo del sistema que estemos construyendo, puede ser que no necesitemos administrar el flujo de control ni se empleen componentes, por tanto, esta capa no estaría presente en tal situación.

Servicios

Si la capa de presentación es la más importante para el usuario final, la de servicios lo es para los desarrolladores. Sin lugar a dudas, la más importante e indispensable de un software, el Application Programming Interface (API) del sistema. Es el punto de entrada a la lógica de negocios, la cual se encuentra encapsulada a través de un conjunto de componentes de servicio.

Esta capa requiere un gran esfuerzo para las pruebas unitarias, ya que es imperativo asegurarse de que estos componentes respondan adecuadamente a las diferentes condiciones posibles de ejecución. De tenerlos, es aquí donde residirían los servicios web (web services) del sistema.

Dominio

Si se desean los beneficios de un diseño orientado a objetos (el ser reutilizable y de fácil mantenimiento), esta capa debe estar presente dentro del sistema que se desee desarrollar. En ella está contenido nuestro modelo conceptual, mejor conocido como el modelo de dominio, de allí el nombre de la capa.

Este modelo consiste en todas aquellas entidades (facturas, productos, órdenes de compra, etc.) y actores (clientes, bancos, empleados, etc.) que componen y/o interactúan en los procesos del negocio, y son creadas a raíz de la abstracción de sus atributos significativos para el sistema. Este modelo típicamente es ilustrado con diagramas de estructura, como el diagrama de clases de UML. Una característica de estas entidades es su capacidad para asociarse con otras entidades o, incluso con sí mismas. Es posible que un sistema no cuente con capa de dominio; en estos casos la capa de servicios estaría codificada de tal manera que dentro de los mismos métodos de sus objetos se accederían directamente los datos en donde estos se almacenan. Esta opción tiene la desventaja de que es de difícil mantenimiento y poco reutilizable, pero puede ser factible para sistemas de vida corta o para aquellos que no tendrán cambios significativos a través del tiempo, incluso para prototipos.

Persistencia

Al hablar de persistencia hablamos de almacenamiento de información en un medio no volátil (bases de datos, archivos de texto plano, etc). Los elementos de esta capa típicamente funcionan como objetos de acceso a datos a Data Access Objects (DAOs), proveyendo métodos para leer y almacenar los datos de los objetos de negocio. Es por eso que la existencia de esta capa típicamente está condicionada a la existencia de la capa de dominio presente, no hay razón para hacerlo persistente.

Incluso con la presencia de un modelo de dominio, podría no ser necesario codificar esta capa cuando se empleen marcos de trabajo tales como persistencia manejada por el contenedor. Al codificar las entidades en la capa de dominio, estos marcos de trabajo se encargarán de la persistencia de cada una de ellas, liberando a los programadores de esta responsabilidad.

La persistencia, como caso clásico, es implementada por los programadores mediante la codificación de clases con sentencias SQL cuyo objeto es crear, modificar, eliminar y consultar los registros de una tabla.

Orden de Desarrollo

Ya descritas las cinco capas. Ahora hablemos sobre como ayudan a definir el orden en que se debería desarrollar un sistema. Sucede que estas capas presentan una característica adicional: cada una de ellas depende de la capa inmediata inferior. Ya sea en tiempo de compilación o de ejecución, los tipos en las capas superiores dependerán de los tipos de las capas inferiores. Por ejemplo, volviendo al ejemplo del componente que muestra los nombres de los clientes, para que la capa de aplicación, en donde se encuentra codificado el componente, muestre esta información necesita invocar un método que se encuentra en la capa de servicios. Es esta dependencia la que marca el orden del desarrollo.

Lo primero que debe ser codificado es la capa de dominio, seguida por las de persistencia, servicios y por las de interfaz de usuario (aplicación y presentación). La de dominio es la primera debido a que aquí residen las entidades que deben estar listas y compiladas para poder ser utilizadas por la capa de servicios, que es la que contiene la lógica de negocio; adicionalmente, estas entidades típicamente son sencillas de implementar. Una vez lista la capa de dominio, ésta desencadena ordenadamente el desarrollo de las otras capas. Es recomendable tener en operación la capa de servicios lo más pronto posible para lograr el desarrollo de la interfaz de usuario en paralelo con las demás capas.

Una importante observación, asumiendo que se está creando un sistema bajo un proceso de desarrollo iterativo e incremental: no es necesario desarrollar totalmente y con lujo de detalle cada capa. Sólo deben ser definidas aquellas piezas de código directamente relacionadas con el o los casos de uso que se estén llevando a cabo en la iteración actual.

ANÁLISIS Y METODOLOGÍA EMPLEADA

La metodología empleada para el desarrollo del directorio de personas e instituciones del IMTA tomó como base el proceso de Desarrollo y Mantenimiento de Software definido previamente por la DGSCA en su intento de implantar el Modelo de Procesos de Software (Moprosoft). En la Figura 1 presentó el diagrama que representa las diferentes fases, actividades, actores y entregables definidas en el proceso.

A continuación explicó los detalles del sistema, los cuales muestran el análisis y diseño realizados durante el desarrollo del sistema con base en la información recabada durante la etapa de levantamiento de requerimientos.

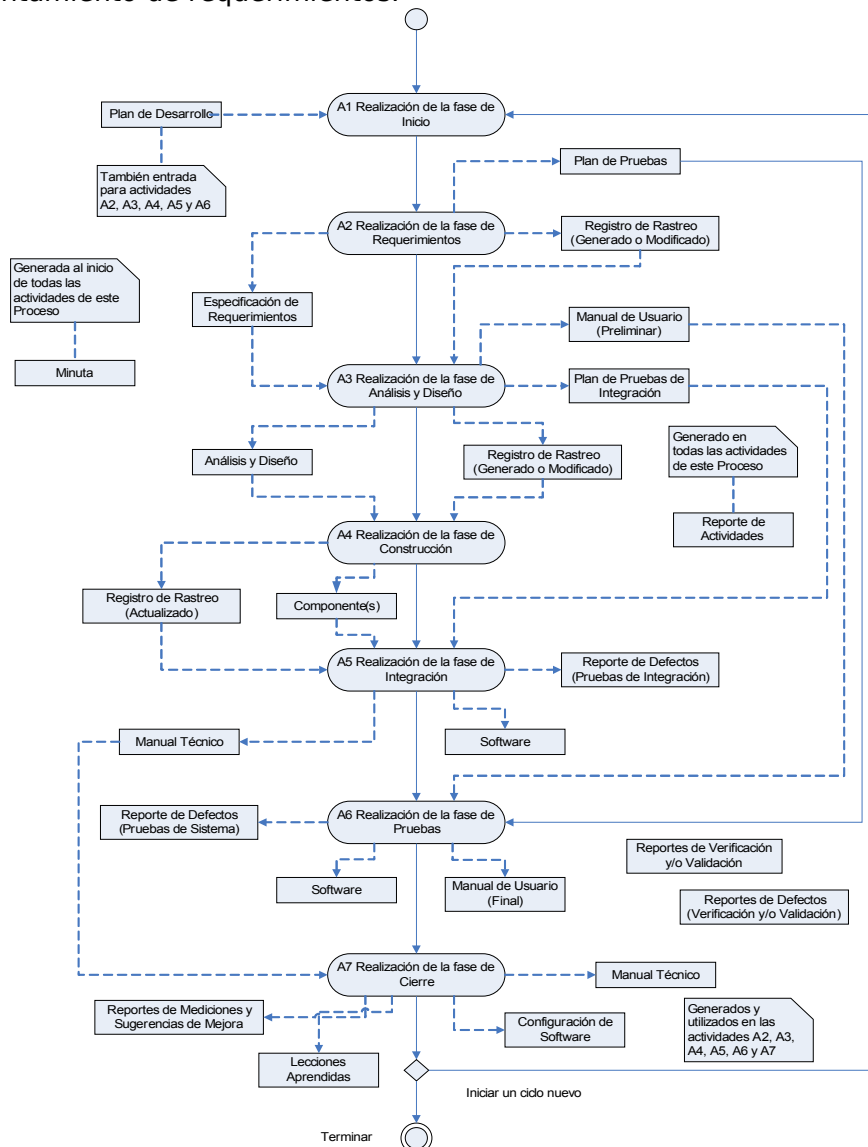


Figura 1. Diagrama del Modelo de Procesos de Software

Visión General del Sistema

El sistema cuenta con tres actores o roles, los cuales se describen a continuación:

ACTOR	DESCRIPCIÓN
Administrador (AD)	Responsable en el sistema de administrar las ramas, roles y usuarios, así como de la exportación del contenido del directorio.
Responsable de Rama (RR)	Responsable en el sistema de ingresar, modificar y eliminar los contactos que conforman una rama (Stakeholders).
Usuario (US)	Usuario en el sistema capaz de consultar el directorio, realizar búsquedas, enviar correos electrónicos e imprimir documentos personalizados.

Tabla 1. Actores del sistema

Las principales acciones que el sistema realiza son:

No.	ACTOR	NOMBRE DEL MÓDULO	DESCRIPCIÓN DEL MÓDULO	NOMBRE DEL COMPONENTE	DESCRIPCIÓN DEL COMPONENTE
1	AD RR US	Autenticación	Permite únicamente el acceso a los usuarios autorizados (Administrador, Responsable de Rama y Usuario) al sistema mediante la verificación de su <i>clave de usuario y contraseña</i> , elementos de autenticación requeridos	Autenticación	Verifica que el usuario sea válido para poder acceder al sistema

No.	ACTOR	NOMBRE DEL MÓDULO	DESCRIPCIÓN DEL MÓDULO	NOMBRE DEL COMPONENTE	DESCRIPCIÓN DEL COMPONENTE
2	AD	Administración	Permite la administración de los diferentes elementos que componen el directorio	Administración de Usuarios	Permite agregar, modificar y eliminar usuarios en el sistema
				Administración de Ramas	Permite agregar, modificar y eliminar ramas en el directorio
3	AD RR	Administración	Permite la administración de los diferentes elementos que componen el directorio	Administración de Contactos	Permite agregar, modificar y eliminar contactos en el directorio
4	AD	Exportación	Permite la exportación del contenido del directorio en otro formato	Exportación de Contactos	Permite la exportación del contenido del directorio en el formato de Excel.
5	AD RR US	Consulta	Permite consultar el directorio, hacer búsquedas sobre el mismo, así como enviar correos electrónicos e imprimir documentos personalizados para un conjunto de contactos	Consulta y Búsqueda de Contactos	Permite consultar y realizar búsquedas sobre el directorio
				Envío de Correo Electrónico	Permite el envío de correo electrónico a un grupo de contactos previamente seleccionado
				Impresión de Documentos Personalizados	Permite la impresión de documentos personalizados a partir de una plantilla o machote y un conjunto de contactos.

Tabla 2. Acciones del sistema

Requerimientos Significativos

Esta sección describo los requerimientos funcionales y requerimientos de atributos de calidad en los que se basa la arquitectura de software. Incluye cualquier escenario que exprese el manejo de la funcionalidad y los objetivos de calidad:

CLAVE DEL ESCENARIO	NOMBRE DEL ESCENARIO	RECURSOS DEL ESTIMULO	ESTÍMULO	ENTORNO	RESPUESTA	RESPUESTA MEDIBLE
SCD-1	Autenticación de Administrador	Administrador	Interfaz de acceso, proporcionando: <ul style="list-style-type: none"> ○ Clave de usuario ○ Contraseña 	Acceso al sistema	Abre la sesión, permite el acceso o en su defecto restringe el mismo	Accede a todos los módulos del sistema
SCD-2	Autenticación de Responsable de Rama	Responsable de Rama	Interfaz de acceso, proporcionando: <ul style="list-style-type: none"> ○ Clave de usuario ○ Contraseña 	Acceso al sistema	Abre la sesión, permite el acceso o en su defecto restringe el mismo	Accede y realiza operaciones en los siguientes módulos: <ul style="list-style-type: none"> •Administración de Contactos •Consulta y Búsqueda de Contactos •Envío de Correo Electrónico •Impresión de Documentos Personalizados
SCD-3	Autenticación de Usuario	Usuario	Interfaz de acceso, proporcionando: <ul style="list-style-type: none"> ○ Clave de usuario ○ Contraseña 	Acceso al sistema	Abre la sesión, permite el acceso o en su defecto restringe el mismo	Accede y realiza operaciones en los siguientes módulos: <ul style="list-style-type: none"> •Consulta y Búsqueda de Contactos •Envío de Correo Electrónico •Impresión de Documentos Personalizados
SCD-4	Modificación de Contraseña	Administrador Responsable de Rama Usuario	En la interfaz de modificación de contraseña, se proporcionan los siguientes datos: <ul style="list-style-type: none"> ○ Contraseña nueva ○ Confirmación de 	Acceso al sistema	Actualiza la contraseña actual del usuario	Acceso al sistema con la nueva contraseña del usuario

CLAVE DEL ESCENARIO	NOMBRE DEL ESCENARIO	RECURSOS DEL ESTIMULO	ESTÍMULO	ENTORNO	RESPUESTA	RESPUESTA MEDIBLE
			contraseña			
SCD-5	Cerrar sesión	Administrador Responsable de Rama Usuario	En la interfaz donde se encuentre el usuario, elige la opción de cerrar sesión	Estando en sesión, es decir, una vez que se ha autenticado el usuario	Muestra interfaz de acceso de usuario	No debe permitir realizar operaciones o visualizar el resto de las interfaces del sistema
SCD-6	Alta de usuarios	Administrador	<p>Crear una cuenta de acceso al sistema, ingresando los siguientes datos:</p> <ul style="list-style-type: none"> ○ Clave de usuario ○ Nombre(s) ○ Primer apellido ○ Segundo apellido ○ Contraseña ○ Confirmación de contraseña ○ Rol ○ Rama(s) asignada(s) ○ Título o Grado ○ Dirección(es) ○ Correo electrónico(s) ○ Sitio Web ○ Teléfono(s) ○ Fax(es) ○ Ciudad ○ Estado ○ Región ○ País ○ Institución u organización ○ Cargo 	Administración de Usuarios	Confirmación de operación realizada	Acceso al sistema con la nueva cuenta

CLAVE DEL ESCENARIO	NOMBRE DEL ESCENARIO	RECURSOS DEL ESTIMULO	ESTÍMULO	ENTORNO	RESPUESTA	RESPUESTA MEDIBLE
			<ul style="list-style-type: none"> ○ Área ○ RFC ○ Domicilio fiscal ○ Razón social ○ Asociación(es) gremial(es) ○ Área de conocimiento ○ Área de especialidad ○ Subespecialidades ○ Sector ○ Subsector 			
SCD-7	Modificación de datos de usuario	Administrador	<p>Cambio de uno o varios de los siguientes datos:</p> <ul style="list-style-type: none"> ○ Nombre(s) ○ Primer apellido ○ Segundo apellido ○ Contraseña ○ Rol ○ Rama(s) asignada(s) ○ Título o Grado ○ Dirección(es) ○ Correo electrónico(s) ○ Sitio Web ○ Teléfono(s) ○ Fax(es) ○ Ciudad ○ Estado ○ Región ○ País 	Administración de Usuarios	Confirmación de operación realizada	Consulta de la cuenta actualizada

CLAVE DEL ESCENARIO	NOMBRE DEL ESCENARIO	RECURSOS DEL ESTIMULO	ESTÍMULO	ENTORNO	RESPUESTA	RESPUESTA MEDIBLE
			<ul style="list-style-type: none"> ○ Institución u organización ○ Cargo ○ Área ○ RFC ○ Domicilio fiscal ○ Razón social ○ Asociación(es) gremial(es) ○ Área de conocimiento ○ Área de especialidad ○ Subespecialidades ○ Sector ○ Subsector 			
SCD-8	Eliminación de un usuario	Administrador	El administrador selecciona la cuenta de usuario a eliminar y acepta continuar con el proceso	Administración de Usuarios	Confirmación de operación realizada	El usuario no aparece en la lista de usuarios del sistema. No se puede acceder al sistema con la clave de usuario y contraseña que se eliminaron
SCD-9	Alta de contactos	Administrador Responsable de Rama	<p>Crear una ficha de contacto, ingresando los siguientes datos:</p> <ul style="list-style-type: none"> ○ Nombre(s) ○ Primer Apellido ○ Segundo Apellido ○ Título o Grado ○ Dirección(es) 	Administración de Contactos	Confirmación de operación realizada	Consulta de la nueva ficha de contacto

CLAVE DEL ESCENARIO	NOMBRE DEL ESCENARIO	RECURSOS DEL ESTIMULO	ESTÍMULO	ENTORNO	RESPUESTA	RESPUESTA MEDIBLE
			<ul style="list-style-type: none"> ○ Correo electrónico(s) ○ Sitio Web ○ Teléfono(s) ○ Fax(es) ○ Ciudad ○ Estado ○ Región ○ País ○ Institución u organización ○ Cargo ○ Área ○ RFC ○ Domicilio fiscal ○ Razón social ○ Asociación(es) gremial(es) ○ Área de conocimiento ○ Área de especialidad ○ Subespecialidades ○ Sector ○ Subsector 			
SCD-10	Modificación de datos de contacto	Administrador Responsable de Rama	<p>Cambio de uno o varios de los siguientes datos:</p> <ul style="list-style-type: none"> ○ Nombre(s) ○ Primer Apellido ○ Segundo Apellido ○ Título o Grado ○ Dirección(es) ○ Correo 	Administración de Contactos	Confirmación de operación realizada	Consulta de la ficha de contacto actualizada

CLAVE DEL ESCENARIO	NOMBRE DEL ESCENARIO	RECURSOS DEL ESTIMULO	ESTÍMULO	ENTORNO	RESPUESTA	RESPUESTA MEDIBLE
			electrónico(s) ○ Sitio Web ○ Teléfono(s) ○ Fax(es) ○ Ciudad ○ Estado ○ Región ○ País ○ Institución u organización ○ Cargo ○ Área ○ RFC ○ Domicilio fiscal ○ Razón social ○ Asociación(es) gremial(es) ○ Área de conocimiento ○ Área de especialidad ○ Subespecialidades ○ Sector ○ Subsector			
SCD-11	Eliminación de un contacto	Administrador Responsable de Rama	El administrador selecciona la ficha de contacto a eliminar y acepta continuar con el proceso	Administración de Contactos	Confirmación de operación realizada	No se puede consultar la ficha de contacto eliminada
SCD-12	Alta de ramas	Administrador	Crear una rama del directorio, ingresando los siguientes datos: ○ Nombre ○ Responsable	Administración de Ramas	Confirmación de operación realizada	Consulta de la nueva rama Se pueden realizar operaciones de administración de contactos en la

CLAVE DEL ESCENARIO	NOMBRE DEL ESCENARIO	RECURSOS DEL ESTIMULO	ESTÍMULO	ENTORNO	RESPUESTA	RESPUESTA MEDIBLE
						nueva rama
SCD-13	Modificación de datos de una rama	Administrador	Cambio de uno o varios de los siguientes datos: <ul style="list-style-type: none"> ○ Nombre ○ Responsable 	Administración de Ramas	Confirmación de operación realizada	Consulta de la rama actualizada
SCD-14	Eliminación de una rama	Administrador	El administrador o responsable de rama selecciona la rama a eliminar y acepta continuar con el proceso	Administración de Ramas	Confirmación de operación realizada	No se puede consultar la rama eliminada ni los contactos que contenía No se pueden realizar operaciones de administración de contactos en la rama eliminada
SCD-15	Exportación de contactos	Administrador	El administrador ingresa el nombre del archivo junto con la ruta absoluta donde éste ubicará.	Exportación	Confirmación de operación realizada	El archivo fue creado y contiene la información de los contactos del directorio
SCD-16	Exportación de una rama	Administrador	El administrador selecciona la rama a exportar e ingresa el nombre del archivo junto con la ruta absoluta donde éste ubicará	Exportación	Confirmación de operación realizada	El archivo fue creado y contiene la información de los contactos de la rama seleccionada
SCD-17	Consulta de contactos	Administrador Responsable de Rama Usuario	El usuario ingresa a la interfaz de consulta de contactos	Consulta	Listado de los contactos a los cuales tiene acceso el usuario por rama	Se encuentran todas las fichas de contacto registradas y actualizadas

CLAVE DEL ESCENARIO	NOMBRE DEL ESCENARIO	RECURSOS DEL ESTIMULO	ESTÍMULO	ENTORNO	RESPUESTA	RESPUESTA MEDIBLE
SCD-18	Búsqueda de contactos	Administrador Responsable de Rama Usuario	<p>En la interfaz de consulta, proporcionando y seleccionando si debe cumplir con uno o varios de los siguientes datos:</p> <ul style="list-style-type: none"> ○ Nombre(s) ○ Primer Apellido ○ Segundo Apellido ○ Título o Grado ○ Correo electrónico(s) ○ Ciudad ○ Estado ○ Región ○ País ○ Institución u organización ○ Asociación(es) gremial(es) ○ Área de conocimiento ○ Área de especialidad ○ Subespecialidades ○ Sector ○ Subsector 	Consulta	Listado de todos los contactos que concuerden con el criterio de búsqueda	Se encuentran todas las fichas de contacto correspondientes al criterio de búsqueda
SCD-19	Envío de correo electrónico	Administrador Responsable de Rama Usuario	<p>Seleccionados uno o varios contactos, a partir de una consulta o búsqueda, se proporcionan los siguientes datos:</p> <ul style="list-style-type: none"> ○ Asunto ○ Cuerpo del mensaje 	Consulta	Envío de los correos electrónicos a los contactos seleccionados	Se recibieron los correos electrónicos por los contactos correspondientes

CLAVE DEL ESCENARIO	NOMBRE DEL ESCENARIO	RECURSOS DEL ESTIMULO	ESTÍMULO	ENTORNO	RESPUESTA	RESPUESTA MEDIBLE
SCD-20	Impresión de documentos personaliza-dos	Administrador Responsable de Rama Usuario	Seleccionados uno o varios contactos, a partir de una consulta o búsqueda, se escoge la plantilla y se ingresan los datos correspondientes	Consulta	Impresión de los documentos personalizados	Visualización de los documentos impresos con la información correspondientes

Tabla 3. Requerimientos significativos

Panorama de la Solución

En este apartado describo el diseño de la arquitectura de software que se apega más al contexto de desarrollo y operación del sistema.

Esta sección provee el razonamiento de las decisiones principales de diseño que contempla la arquitectura del sistema. Describe cualquier enfoque de diseño, incluyendo el uso de estilos de arquitectura o patrones de diseño. De la misma manera, se justifica la selección del estilo o patrón de arquitectura. Por otra parte, describe cualquier alternativa significativa que fue seriamente considerada y el por qué finalmente fue rechazada.

Este diseño se encuentra basado en los siguientes estilos arquitectónicos:

- *Estilo basado en llamada y retorno.* Especifican un orden para el cómputo de la información a manejar en el sistema, es decir los componentes tienen una secuencia para relacionarse, enviar una petición y esperar una respuesta. De esta manera, un componente puede quedarse ocioso para el sistema, mientras espera la respuesta de otros componentes.

Este estilo arquitectónico es utilizado para agrupar los componentes de la arquitectura de software del Directorio de Personas e Instituciones del Instituto Mexicano de Tecnología del Agua de acuerdo a cierta funcionalidad para comunicarlos por medio de una llamada y retorno.

Del estilo basado en llamada y retorno, el equipo de trabajo determinó específicamente la combinación de los siguientes estilos arquitectónicos:

- *Estilo organizado en capas.* Define que las tareas que realizan ciertos tipos de componentes, de acuerdo con su objetivo, pueden ser divididas entre aplicaciones específicas, las cuales se conocerán como capas. Este estilo arquitectónico es utilizado por la arquitectura de software del Directorio de Personas e Instituciones del Instituto Mexicano de Tecnología del Agua para organizar los componentes a través de capas bien definidas de acuerdo a un alcance identificado.
- *Estilo orientado a objetos.* Explica que la definición de componentes debe estar basada en una filosofía de reutilización de funcionalidades que ofrecen dichos componentes de acuerdo a elementos y tareas en común que asignemos a los mismos. Este estilo arquitectónico es utilizado por la arquitectura de software del Directorio de Personas e Instituciones del Instituto Mexicano de Tecnología del Agua para crear componentes con funcionalidad en común y que son reutilizables con el fin de comunicarse con otros componentes que ayuden a ejecutar un requerimiento del sistema en específico.
- *Estilo centrado en datos.* Atiende el problema del almacenamiento, representación, administración y recuperación de una gran cantidad de datos relacionados. Específicamente, el estilo arquitectónico de este tipo que utilizará la arquitectura de software del Directorio de Personas e Instituciones del Instituto Mexicano de Tecnología del Agua es:
 - *Estilo repositorio / base de datos jerárquica.* Define que el orden de la ejecución de un componente está determinado por una cadena de solicitudes para acceder a los datos. Este estilo arquitectónico es utilizado por la arquitectura del software del Directorio de Personas e Instituciones del Instituto Mexicano de Tecnología del Agua para almacenar y gestionar los datos que se generen a través de la misma.

Patrón Arquitectónico Modelo Vista Controlador

La arquitectura de software del directorio de Personas e Instituciones hace uso del patrón arquitectónico Modelo Vista Controlador (model-view-controller, MVC). Este patrón arquitectónico separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos de forma que las modificaciones al componente de la vista pueden ser hechas con un mínimo impacto en el componente del modelo de datos. Esto es útil ya que los modelos típicamente tienen cierto grado de estabilidad (dependiendo de la estabilidad del dominio del problema que está siendo modelado), donde el código de la interfaz de usuario sea más robusto, debido a que el desarrollador está menos propenso a "romper" el modelo mientras trabaja de nuevo en la vista.

En términos generales, implementar una arquitectura haciendo uso del patrón MVC implica definir tres clases de módulos.

Modelo. Es la representación específica del dominio de la información sobre la cual funciona el sistema, es decir, la lógica del negocio.

Vista. Presenta el modelo en un formato adecuado para interactuar, usualmente un elemento de interfaz de usuario.

Controlador. Responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

Es común pensar que una aplicación tiene tres capas principales: presentación, dominio o lógica de negocio, y acceso a datos. En MVC, la capa de presentación está partida en controlador y vista. La principal separación es entre presentación y dominio; la separación entre V/C es menos clara.

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente:

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, pulsa un botón, enlace)
2. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario. Los controladores complejos están a menudo estructurados usando un patrón que encapsula las acciones y simplifica su extensión.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo. El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, otro patrón llamado observador puede ser utilizado para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice.

- La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

El patrón arquitectónico MVC es utilizado por la arquitectura de software del Directorio de Personas e Instituciones del Instituto Mexicano de Tecnología del Agua para asignar responsabilidades a las capas definidas en las mismas y agrupar los componentes de acuerdo a una funcionalidad específica.

Tecnología

Teniendo en cuenta la arquitectura de software anteriormente definida, se requirió de la selección de diversas tecnologías. Para ello, se empleó una tabla en donde se ponderaron las ventajas de las diversas opciones disponibles en cada rubro. La opción con puntaje más alto fue la seleccionada. A continuación se muestran las tablas de ponderación, en las cuales se marca en azul la opción seleccionada:

TECNOLOGIA	SISTEMA OPERATIVO	FRAMEWORK	EXPERTOS EN SISTEMAS OPERATIVOS	EXPERTOS EN LA TECNOLOGIA	CURVA DE APRENDIZAJE	MODULARIDAD DE COMPONENTES	SOFTWARE LIBRE O PROPIETARIO	SEGURIDAD	TOTAL
Importancia	5	5	5	5	4	5	5	3	
Java	5	5	5	5	3	5	5	5	177
PHP	5	3	5	5	5	4	5	4	167

Tabla 4. Lenguaje de programación

TECNOLOGIA	SISTEMA OPERATIVO	FRAMEWORKS	EXPERTOS EN SISTEMAS OPERATIVOS	EXPERTOS EN LA TECNOLOGIA	CURVA DE APRENDIZAJE	MODULARIDAD DE COMPONENTES	SOFTWARE LIBRE O PROPIETARIO	SEGURIDAD	TOTAL
Importancia	0	0	0	5	5	5	5	3	
Struts	0	0	0	4	4	5	5	5	105
Java Server Faces	0	0	0	3	3	5	5	5	95
Spring	0	0	0	1	1	5	5	5	75

Tabla 5. Framework

TECNOLOGÍA	SISTEMA OPERATIVO	FRAMEWORKS	EXPERTOS EN SISTEMAS OPERATIVOS	EXPERTOS EN LA TECNOLOGÍA	CURVA DE APRENDIZAJE	MODULARIDAD DE COMPONENTES	SOFTWARE LIBRE O PROPIETARIO	SEGURIDAD	TOTAL
Importancia	0	0	0	5	5	0	5	3	
Tomcat	0	0	0	5	4	0	5	5	85
Java EE Application Server	0	0	0	1	1	0	5	5	50
JBoss Application Server	0	0	0	1	1	0	5	5	50

Tabla 6. Servidor de aplicaciones

TECNOLOGÍA	FACILIDADES DE ADMINISTRACIÓN	FRAMEWORKS	DESEMPEÑO	EXPERTOS EN LA TECNOLOGÍA	CURVA DE APRENDIZAJE	MODULARIDAD DE COMPONENTES	SOFTWARE LIBRE O PROPIETARIO	SEGURIDAD	TOTAL
Importancia	5	5	3	5	4	5	5	3	
OpenLdap	3	5	3	1	3	5	5	5	131
Fedora Directory Server	5	5	5	1	3	5	5	5	147

Tabla 7. Base de datos

TECNOLOGÍA	SISTEMA OPERATIVO	FRAMEWORKS	EXPERTOS EN SISTEMAS OPERATIVOS	EXPERTOS EN LA TECNOLOGÍA	CURVA DE APRENDIZAJE	MODULARIDAD DE COMPONENTES	SOFTWARE LIBRE O PROPIETARIO	SEGURIDAD	TOTAL
Importancia	0	0	0	5	4	5	5	3	
JNDI	0	0	0	3	5	5	5	5	100
LDAPJDK (sun)	0	0	0	1	4	5	5	5	86
JLDAP (OpenLdap)	0	0	0	1	4	5	5	5	86
Clases del SAI	0	0	0	3	3	5	5	5	92

Tabla 8. API de integración

Estas decisiones se pueden agrupar en la siguiente matriz organizada en capas horizontales y verticales:

	CLIENTE	VISTA	CONTROLADOR	MODELO	INTEGRACIÓN	DATOS
COMPONENTES	No aplica	JSP	Filters, Interceptors, archivos xml.	Actions.	POJOs	
APIs	No aplica	Struts 2.0.9 Servlet API 2.4 JSP API 2.0	Struts 2.0.9 Servlet API 2.4	Struts 2.0.9	JNDI	
MIDDLEWARE	Navegador: Internet Explorer 7 o posterior, Firefox 2 o posterior, Safari 3 o posterior. Acrobat reader 8 o posterior.	Tomcat 6.0.13 Java 6v2	Tomcat 6.0.13 Java 6v2	Tomcat 6.0.13 Java 6v2	Tomcat 6.0.13 Java 6v2	Fedora Directory Server 1.0.4
SISTEMA OPERATIVO	Cualquiera que soporte los navegadores mencionados	Fedora 6	Fedora 6	Fedora 6	Fedora 6	Fedora 6
HARDWARE	Cualquiera que soporte los navegadores mencionados					

Tabla 9. Tecnología utilizada

Arquitectura

El estilo arquitectónico descrito establece la forma que siguió el diseño de la aplicación planteando los escenarios o acciones entre las capas específicas. Considerando que el sistema es orientado a Web y que debe seguir el estilo cliente-servidor, la descripción de las capas se definió de la siguiente manera: El usuario proporciona la información (Cliente) del formulario en el navegador (el usuario pulsa un botón, enlace). Al enviar la información y ser atendida por el servidor Web, es recibida por el Controlador, el cual realiza una petición de operación correspondiente a la acción solicitada por el usuario, tomando la información recibida. La relación con otros componentes que definen la lógica del negocio dependerá de la operación, de la información recibida para la obtención de resultados y de los datos almacenados en la base de datos para entonces dirigir al cliente Web a la siguiente página correspondiente que le presentará al usuario.

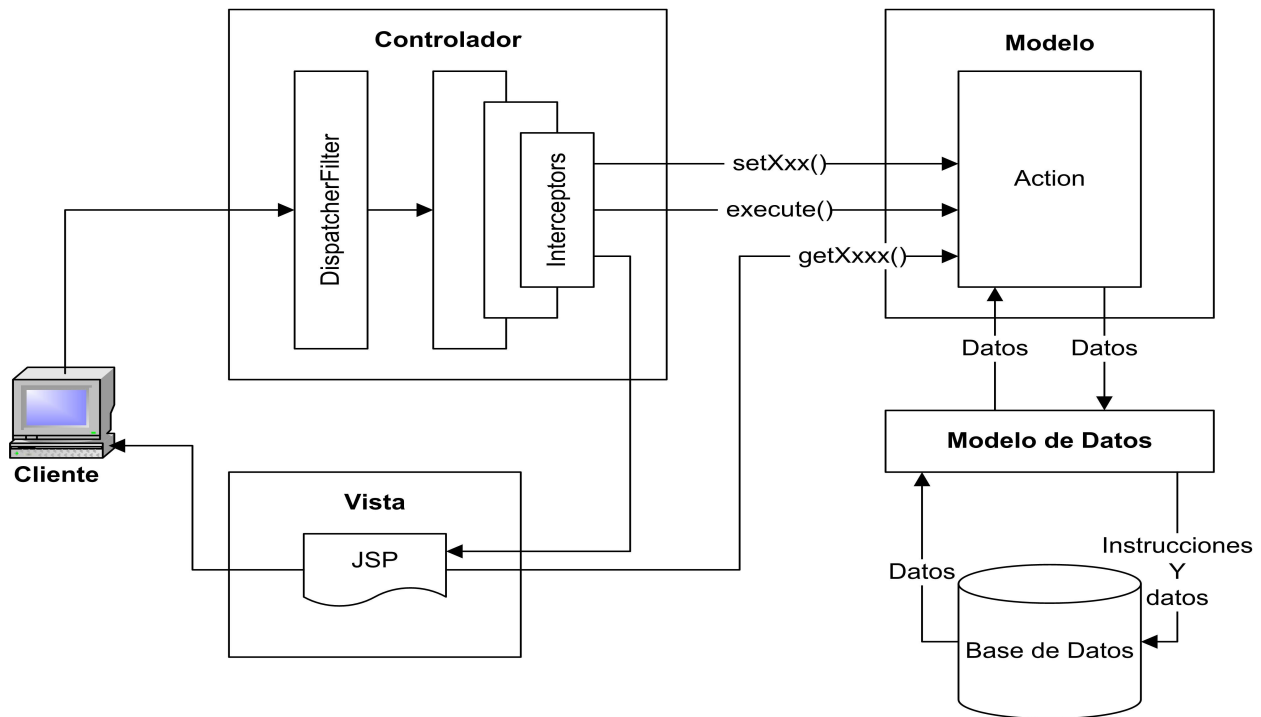


Figura 2. Arquitectura general del sistema.

Vista Conceptual

La vista conceptual describe el sistema en términos de componentes específicos de dominio y relaciones, es usada para definir los requerimientos funcionales así como la visión que el administrador, responsable de rama y usuario tienen de la aplicación y refleja las reglas de negocio que la arquitectura debe cubrir.

La vista conceptual presentada en este documento forma en su conjunto una especificación completa del sistema y representa los límites de abstracción entre los componentes que interactúan para satisfacer los requerimientos del sistema.

Los componentes que integran el sistema están organizados en capas, donde cada una de ellas tiene una funcionalidad específica y puede invocar operaciones de la capa subsecuente. Las capas que componen el sistema son:

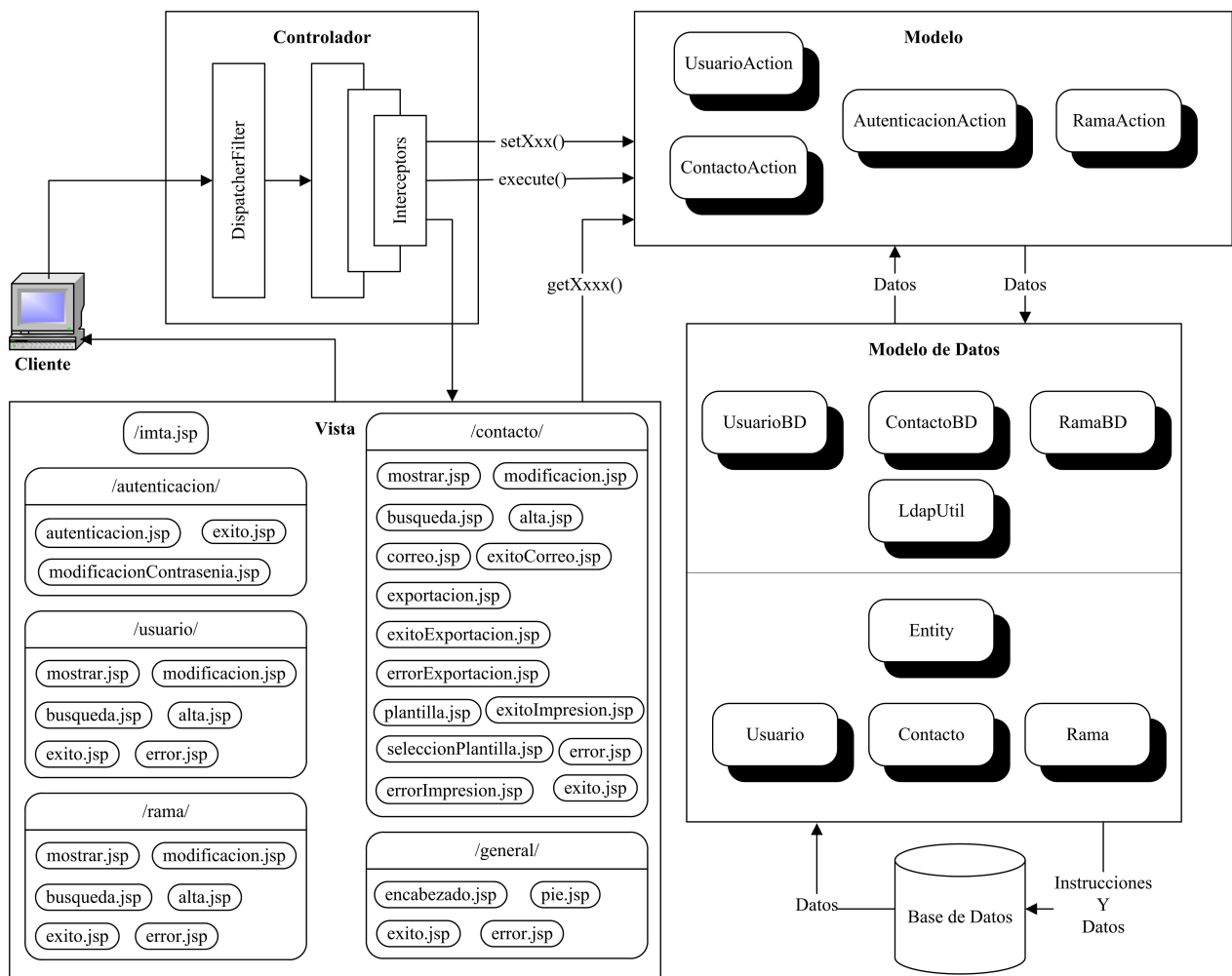


Figura 3. Vista conceptual del sistema.

Catálogo de Elementos

Capa de Vista

Corresponde a la parte del sistema que tiene como función interactuar con el usuario: presenta la información, recibe los datos de entrada, valida los datos si es necesario y atiende las peticiones que se realicen en la capa de Control.

Para lograr lo anterior se utilizó una serie de páginas Web dinámicas donde la secuencia de páginas, en general variables, depende de las acciones del usuario.

COMPONENTE	ELEMENTO
Principal	imta.jsp
General	encabezado.jsp encabezado_inicio.jsp pie.jsp paginadorForma.jsp paginadorLeyenda.jsp paginadorMenuNavegacion.jsp

autenticacion	autenticacion.jsp modificacionContrasenia.jsp
usuario	mostrar.jsp detalle.jsp alta.jsp modificacion.jsp busqueda.jsp exito.jsp error.jsp
rama	mostrar.jsp alta.jsp modificacion.jsp busqueda.jsp exito.jsp error.jsp exitoAgregar.jsp exitoModificar.jsp exitoEliminar.jsp
contacto	mostrar.jsp mostrarAdmin.jsp detalle.jsp detalleAdmin.jsp alta.jsp exitoAlta.jsp modificacion.jsp exitoModificacion.jsp eliminacion.jsp exitoBaja.jsp errorBaja.jsp busqueda.jsp busquedaAdmin.jsp correo.jsp exitoCorreo.jsp seleccionPlantilla.jsp exitoImpresion.jsp
exportación	imta.jsp input.jsp exito.jsp error.jsp

Tabla 10. Componentes de la capa de vista.

Capa de Control

Está encargada de controlar el flujo de todo el sistema. Tiene por objetivo atender las peticiones efectuadas por el usuario y de acuerdo a la solicitud enviada delegará a la correspondiente subcapa o componente(s) la realización de la operación. Funge el rol de intermediario entre la capa de vista y la de modelo, por lo que así como recibe las peticiones por parte de la vista, también le envía la respuesta (confirmación de operación, errores o datos) a la del modelo. Esta capa esta implementada en el framework elegido.

Capa de Modelo

Aplica las reglas de negocio especificadas para las funcionalidades del sistema. Tiene por objetivo llevar a cabo la lógica del negocio de acuerdo a las peticiones enviadas por parte de la capa de control, realiza el proceso de los datos, envía la información a la capa de modelo de datos para hacer persistentes los datos y envía la respuesta (éxito de operación, errores o datos) a la capa control.

COMPONENTE	ELEMENTO
General	CorreoElectronico Paginador.java EstatusError.java EstatusSeleccion.java Plantilla.java Rol.java Validación.java XMLUtil.java
autenticacion	AutenticacionAction.java AutenticacionAutenticarAction.java AutenticacionModificarAction.java
usuario	UsuarioAction.java
rama	RamaAction.java
contacto	ContactoAction.java
exportacion	ExportacionAction.java

Tabla 11. Componentes de la capa de modelo.

Capa de Modelo de Datos

Está encargada de hacer persistentes los datos del sistema, recibe los datos de la capa de modelo y los prepara de acuerdo a las especificaciones establecidas en la Base de Datos. Esta capa contiene una capa interna:

COMPONENTE	ELEMENTO
General	LdapUtil.java UidNumber.java Entity.java
usuario	UsuarioBD.java Usuario.java
rama	RamaBD.java Rama.java
contacto	ContactoBD.java Contacto.java
Exportacion	ExportarBD.java

Tabla 12. Componentes de la capa de modelo de datos.

Capa de BD

Tiene por objetivo realizar las operaciones en la Base de Datos, manejando los datos ya preparados, para realizar dichas operaciones en la misma. Esta capa esta implementada por el manejador elegido.

Vista de Módulos

Esta vista de módulos muestra cómo los componentes de la arquitectura de software y la comunicación a través de ellos son mapeados en módulos o subsistemas los cuales son organizados dentro de capas, mismas que se describieron en la vista conceptual, dichas capas son: vista, control, modelo y modelo de datos.

Esta vista contempla los siguientes diagramas:

- Diagrama de paquetes
- Diagramas de clases
- Diagramas de navegación

Diagrama de Paquetes

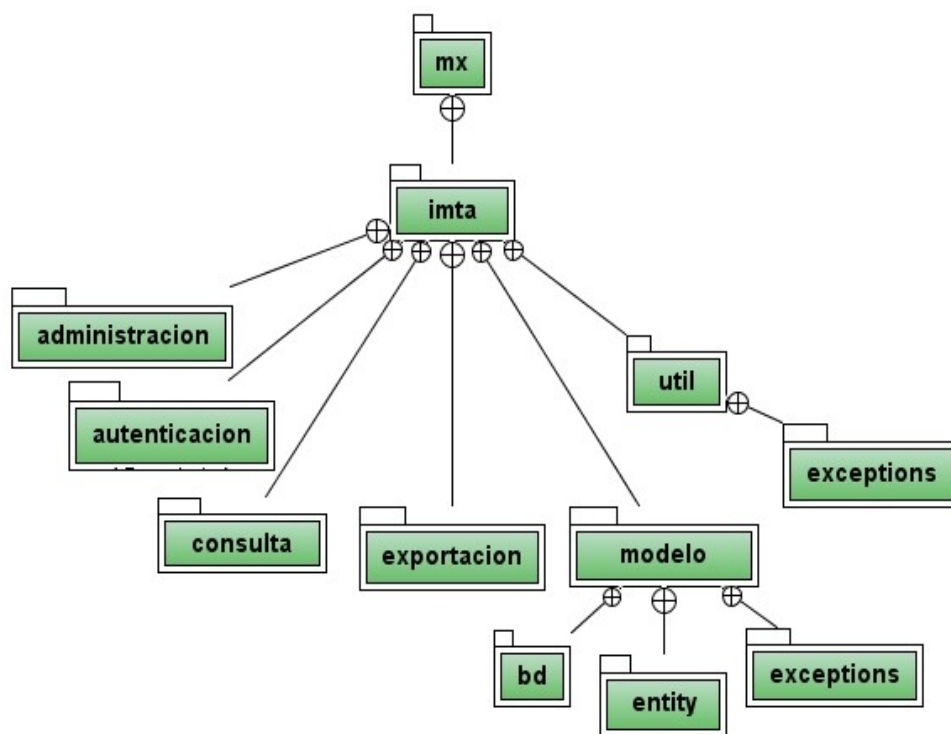


Figura 4. Diagrama de paquetes.

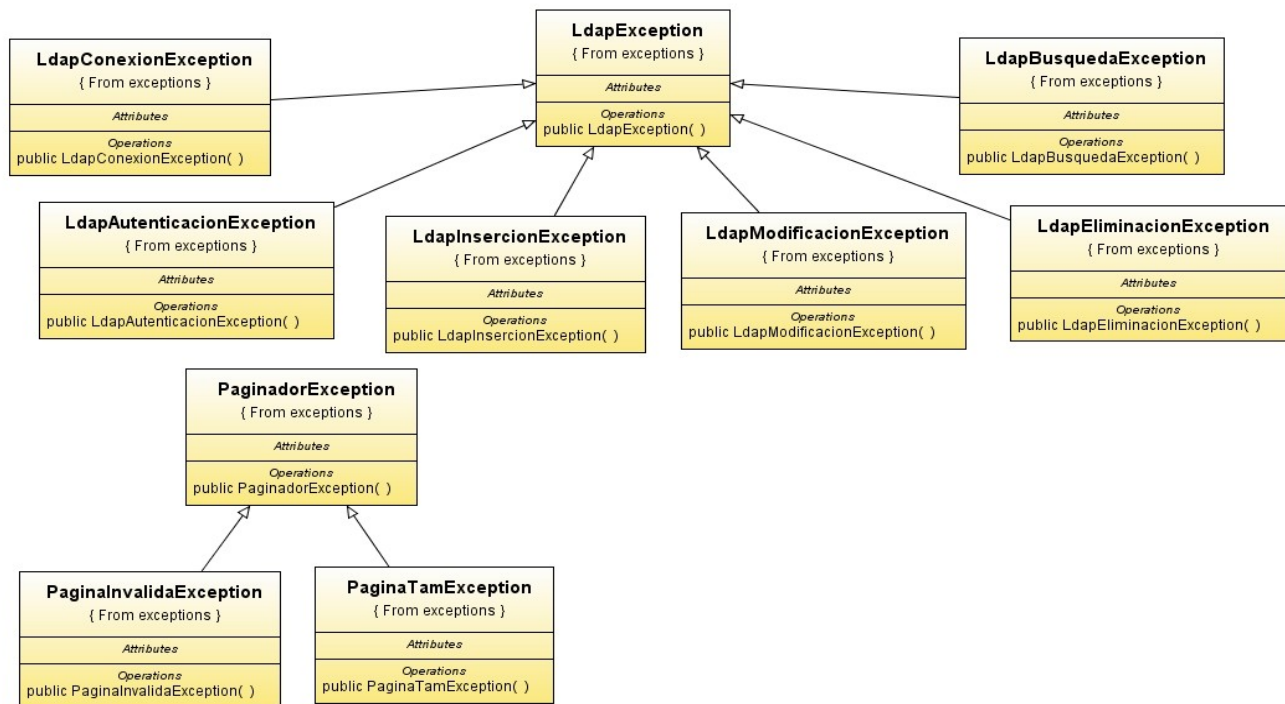


Figura 6. Clases de excepción.

Diagramas de navegación

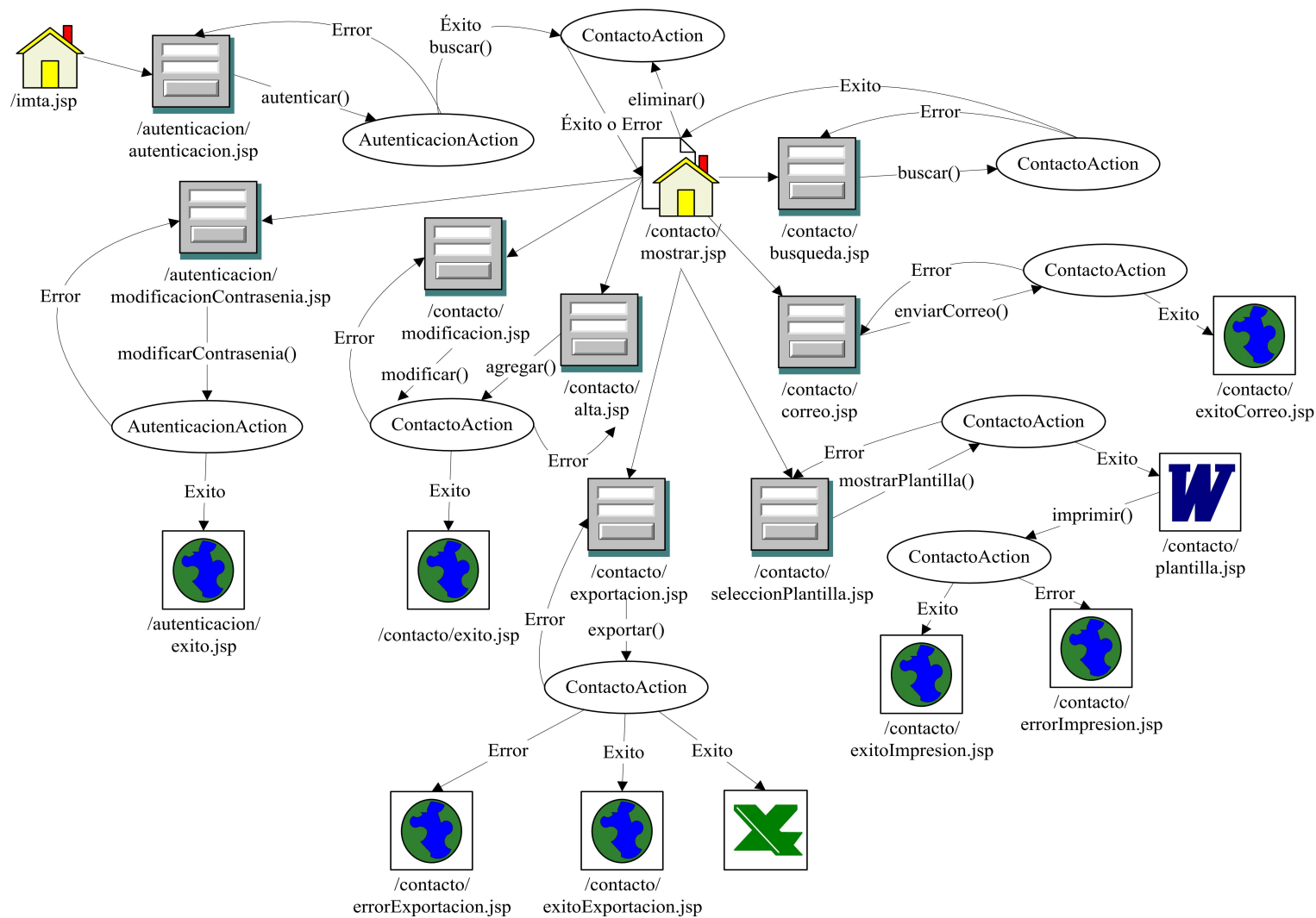


Figura 7. Diagrama de navegación del módulo de autenticación y del módulo de administración y consulta de contactos.

Vista de Datos

Esta vista contiene la información de los datos necesarios para llevar a cabo el registro de la información del sistema:

- Usuarios.
- Contactos.
- Ramas.

Dichos datos se almacenan bajo un diseño de base de datos jerárquica como se muestra en las Figuras 9 y 10.

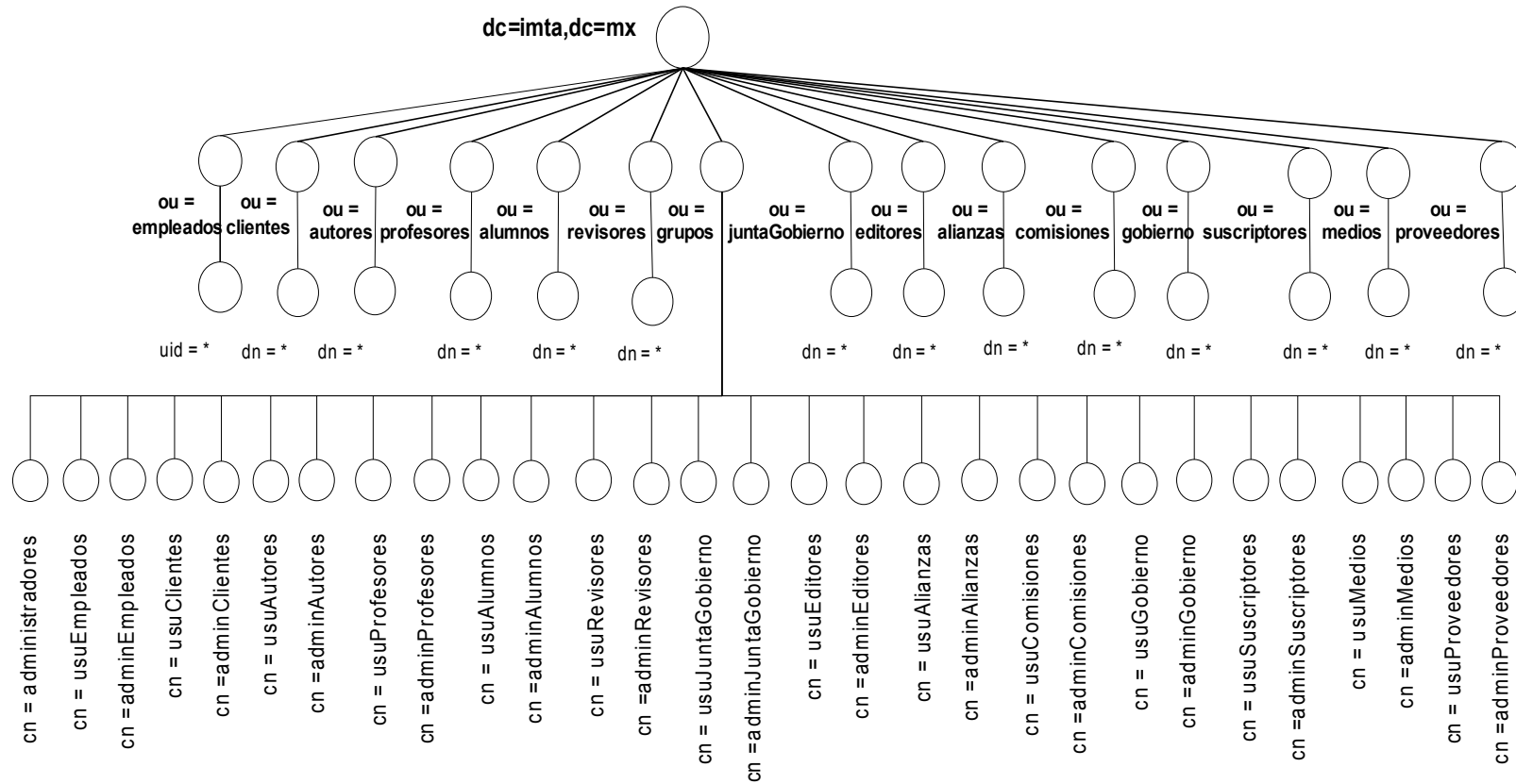


Figura 9. Estructura de almacenamiento del directorio.

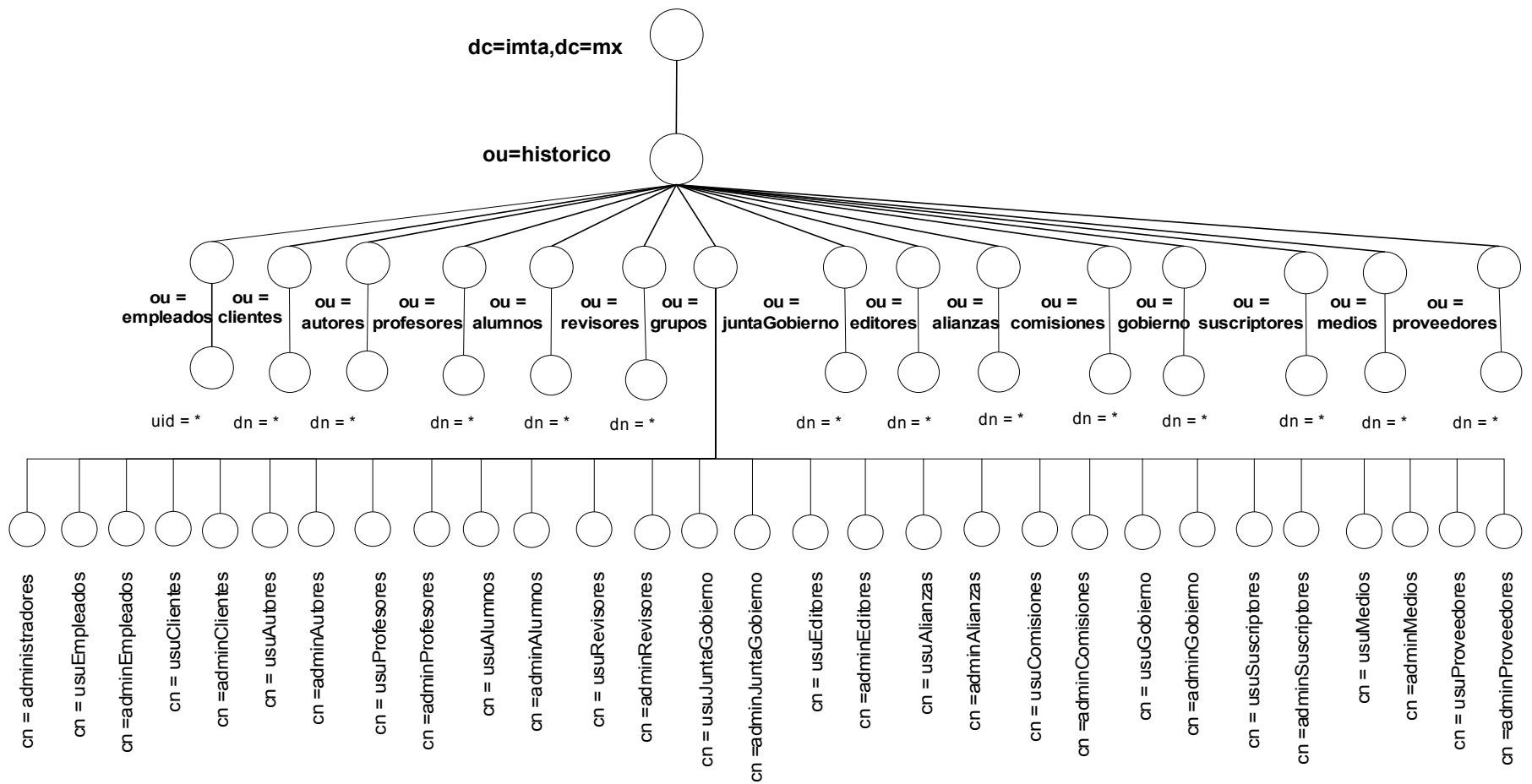


Figura 10. Estructura de almacenamiento del directorio histórico.

El modelo de nombrado muestra la inromación que contiene cada uno de los elementos almacenados en las ramas del directorio. Adicionalmente se muestra la rama de grupos en la cual se almacenan los usuarios del sistema junto con sus permisos correspondientes.

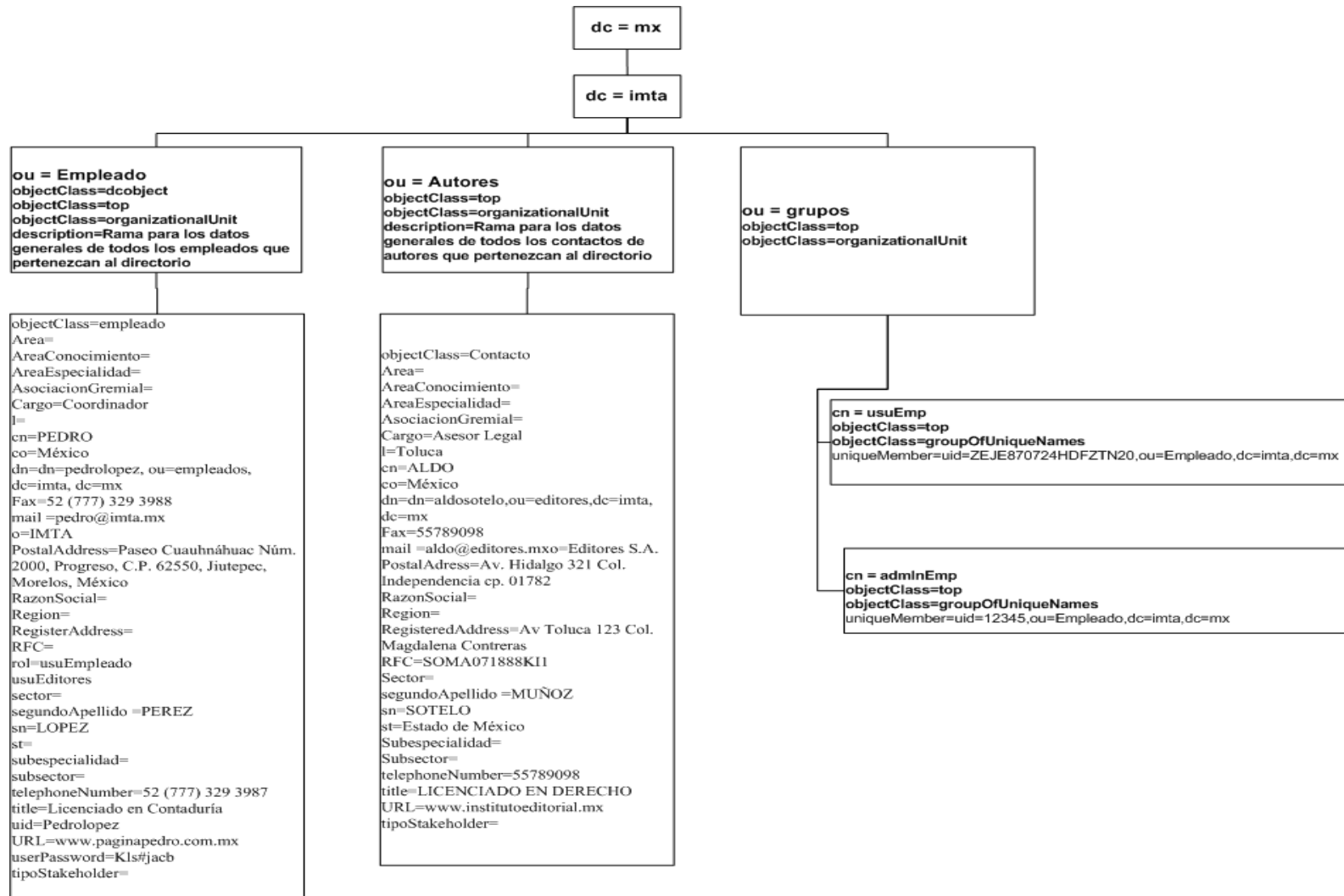


Figura 11. Modelo nombrado.

Vista de Código

Permite mostrar las asignaciones de módulos del sistema al entorno de desarrollo e implementación (librerías, archivos y directorios). Esta vista contiene uno o varios directorios que representarán el esqueleto físico del sistema.

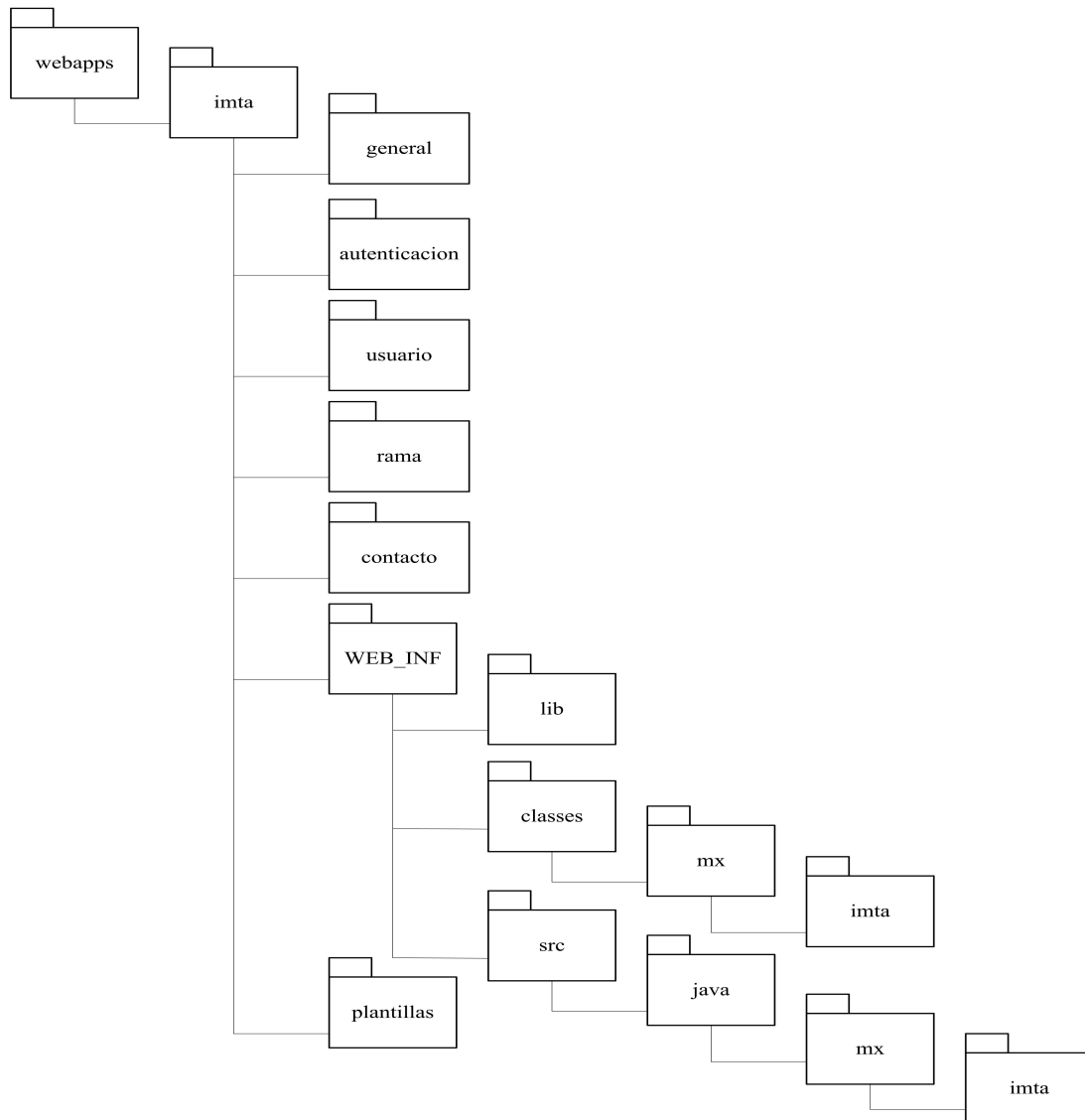


Figura 12. Vista de Código.

DIRECTORIOS	DESCRIPCIÓN
PLANTILLAS	Contiene las plantillas que se emplean en la impresión de documentos personalizados.
PLANTILLAS/DOCS	Contiene los documentos generados por la aplicación.
GENERAL	Contiene archivos que se usan de forma general en la capa de vista.
CSS	Contiene las hojas de estilo.
JS	Contiene las funciones de java script.
IMG	Contiene las imágenes del sitio.
AUTENTICACION	Contiene los archivos de la capa de vista del módulo de autenticación.
USUARIO	Contiene los archivos de la capa de vista del módulo de administración usuarios.
RAMA	Contiene los archivos de la capa de vista del módulo de administración de ramas.
CONTACTO	Contiene los archivos de la capa de vista del módulo de administración y consulta de contactos.
EXPORTACION	Contiene los archivos de la capa de vista del módulo de exportación.
WEB-INF	Contiene las librerías, clases y archivos de configuración necesarios para el funcionamiento del sistema.
WEB-INF/LIB	Contiene archivos jar que incluyen clases y recursos requeridos por la aplicación.
WEB-INF/CLASSES	Contiene las clases y archivos xml del sistema.
WEB-INF/SRC/JAVA	Contiene el código fuente de las clases y archivos xml del sistema.
META-INF	Contiene archivos de configuración generales.

Tabla 13. Directorios instalados en el servidor.

En el documento anexo se listan y muestran todos los archivos que conforman la vista de código.

Vista de Ejecución

Esta vista presenta la infraestructura necesaria para implementar el Sistema de Directorio de Personas e Instituciones del Instituto Mexicano de Tecnología del Agua; se indican los nodos principales, los componentes que los conforman y los canales de comunicación entre ellos.

La vista de ejecución está conformada por un diagrama que permite identificar claramente la distribución de los componentes de hardware que intervienen en la ejecución del Directorio de Personas e Instituciones del Instituto Mexicano de Tecnología del Agua.

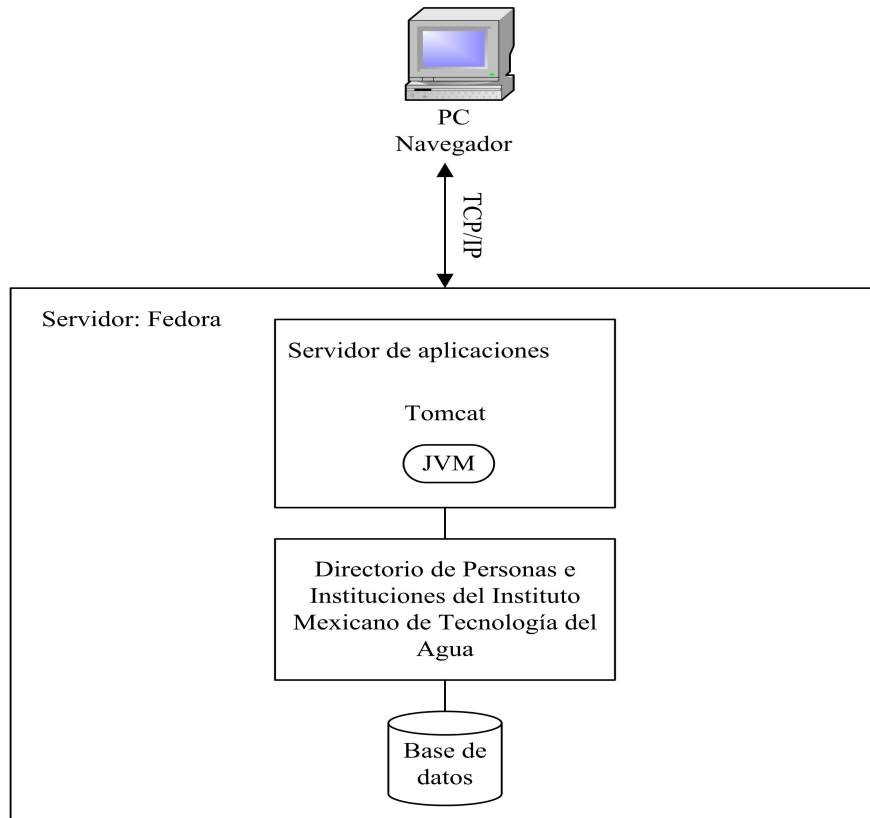


Figura 13. Diagrama de ejecución.

PARTICIPACIÓN PROFESIONAL

A lo largo del desarrollo del Directorio de Personas e Instituciones participé durante todo el proceso de Desarrollo y Mantenimiento de Software desempeñando los siguientes roles:

- Responsable de Desarrollo y Mantenimiento
- Analista
- Diseñador
- Desarrollador

Las actividades que realicé en cada una de las etapas del proceso están apegadas al Modelo de Procesos de Software representado en la Figura 1 y los documentos que menciono corresponden a los entregables y entradas definidas en dicho proceso. Las actividades organizadas por fase son las siguientes:

Fase de Inicio

En una junta de trabajo revisé con los miembros del equipo de trabajo el Plan de Desarrollo, entregado por el Administrador del Proyecto, para lograr un entendimiento común y generé la minuta correspondiente.

Elaboré el Reporte de Actividades registrando las actividades realizadas y lo agregué al repositorio documental.

Fase de Requerimientos

Distribuí las tareas a los miembros del equipo de trabajo según su rol con base en el Plan de Desarrollo.

Obtuve y recopilé los requisitos (funcionales y no funcionales) del sistema. Para ello, me trasladé a las instalaciones del instituto, ubicadas en la ciudad de Cuernavaca, y durante 3 días consecutivos sostuve entrevistas con los responsables y/o empleados de las diferentes áreas que conforman el instituto. De cada una de las entrevistas revisé los repositorios y mecanismos que cada área tenía implementados para llevar a cabo la administración de sus contactos y generé las minutas correspondientes quedando asentados los datos recabados junto con los usos dados por las áreas a sus contactos..

Con base a la información recabada durante las entrevistas documenté la Especificación de Requerimientos y la entregué para su revisión, en primer instancia por el Administrador del Proyecto y el Revisor y posteriormente por el cliente.

Corregí los defectos encontrados en la Especificación de Requerimientos con base en los Reportes de Verificación y de Defectos correspondientes hasta su aprobación por el cliente.

Generé el Registro de Rastreo con la información correspondientes a los requerimientos.

Incorporé la Especificación de Requerimientos como línea base a la Configuración de Software.

Elaboré el Reporte de Actividades registrando las actividades realizadas.

Fase de Análisis y Diseño

Distribuí las tareas a los miembros del equipo de trabajo según su rol y de acuerdo al Plan de Desarrollo.

Analicé la Especificación de Requerimientos para generar la descripción de la estructura interna del sistema y su descomposición en subsistemas, y éstos a su vez en componentes definiendo las interfaces entre ellos.

Creé el documento de Análisis y Diseño, indicando:

- La descripción de la arquitectura de software del sistema.
- El detalle de la apariencia
- El comportamiento de la interfaz
- El modelado de la implementación
- La especificación de los componentes de software

Generé el Registro de Rastreo ingresando los requerimientos del sistema junto con los diferentes elementos planteados en el análisis y diseño indicando las relaciones existentes entre los mismos en una matriz de rastreo.

Corregí los defectos encontrados en el documento de Análisis y Diseño con base en el Reporte de Verificación y el Reporte de Defectos correspondientes.

Elaboré el Plan de Pruebas de Integración en conjunto con el tester asignado.

Incorporé el documento de Análisis y Diseño, el Registro de Rastreo, el Plan de Pruebas de Integración como líneas base a la Configuración de Software.

Elaboré el Reporte de Actividades registrando las actividades realizadas.

Fase de Construcción

Distribuí las tareas a los miembros del equipo de trabajo según su rol con base en el Plan de Desarrollo.

Particé en la creación del entorno de desarrollo necesario para desarrollar el sistema. Dicho entorno definía la estructura básica del proyecto y contenía todos los elementos definidos en el diseño junto con las librerías y archivos de configuración requeridos con base a los frameworks elegidos.

Construí los siguientes Componentes de Software:

- Módulo de Administración de Ramas
- Módulo de Administración de Contactos
- Módulo de Exportación de Contactos

para lo cual,

- Implementé los componentes con base en el documento de Análisis y Diseño
- Aplicé el diseño de interfaz de usuario
- Definí y apliqué pruebas unitarias para verificar el comportamiento de los componentes

- Corregí los defectos encontrados en las pruebas unitarias

Además apoyé en la creación y definición del modelo de datos.

Actualicé el Registro de Rastreo actualizando el estado de los componentes construidos.

Corregí los defectos encontrados en el Registro de Rastreo con base en el Reporte de Verificación y el Reporte de Defectos correspondientes.

Incorporé los componetes y el Registro de Rastreo como líneas base a la Configuración de Software.

Verifiqué el Manual Técnico realizado por el Renposable de Manuales.

Elaboré el Reporte de Actividades registrando las actividades realizadas.

Fase de Integración

Distribuí las actividades a los miembros del equipo de trabajo según su rol y de acuerdo al Plan de Desarrollo.

Realicé la intregación de los componentes que me fueron asignados

Actualicé el Registro de Rastreo incorporando los componentes integrados.

Elaboré el Reporte de Actividades registrando las actividades realizadas.

Fase de Pruebas

Distribuí las tareas a los miembros del equipo de trabajo según su rol y de acuerdo al Plan de Desarrollo.

Corregí los defectos encontrados en las pruebas de concepto y de integración con base al Reporte de Defectos.

Incorporé el Software, los Reportes de Defectos, y el Registro de Rastreo como líneas base a la Configuración de Software.

Elaboré el Reporte de Actividades registrando las actividades realizadas.

Fase de Cierre

Incorporé el Manual Técnico como línea base a la Configuración de Software.

Revisé que la Configuración de Software contuviera el total de productos definidos en el Plan de Desarrollo.

Participé en la reunión efectuda para identificar las lecciones aprendidas y las integré a la Base de Conocimiento.

Generé el Reporte de Mediciones, Observaciones y Sugerencias de Mejora.

Elaboré el Reporte de Actividades registrando las actividades realizadas.

RESULTADOS Y APORTACIONES

Durante la fase de cierre se realizó un cuestionario a todos los integrantes del equipo de trabajo para evaluar el desempeño de los mismos durante el proyecto. El cuestionario evaluó los siguientes aspectos:

- Calidad en el trabajo
- Colaboración y trabajo en equipo
- Comunicación
- Logro de objetivos
- Nivel de experiencia en el rol
- Orientación a resultados
- Orientación al cliente

Durante la evaluación únicamente se calificó mi desempeño en el rol de Desarrollador y para ello se entrevistó a dos compañeros del mismo rol y a mi jefe, en este caso el Administrador o Líder del proyecto. Los resultados que obtuve son los siguientes:

COMPETENCIA	GRADO OBTENIDO
CALIDAD EN EL TRABAJO	83.33%
COLABORACIÓN Y TRABAJO EN EQUIPO	91.67%
COMUNICACIÓN	75.00%
LOGRO DE OBJETIVOS	83.33%
NIVEL DE EXPERIENCIA EN EL ROL	83.33%
ORIENTACIÓN A RESULTADOS	91.67%
ORIENTACIÓN AL CLIENTE	83.33%
PROMEDIO GENERAL	84.52%

Tabla 14. Desempeño durante el proyecto.

Una de las aportaciones más importantes que realice fue una propuesta para resolver el problema de la información duplicada. Dicha propuesta consistió en el uso de referrals, es decir, el manejo de “ligas”, para interconectar los nodos en un directorio. Por lo tanto, me fue asignado el desarrollo de un módulo para el manejo de las conexiones con el LDAP así como la lógica para el manejo de los referrals en cada una de las operaciones permitidas en el directorio.

Algunas aportaciones adicionales fueron:

- Conocimiento amplio del lenguaje Java
- Experiencia en el desarrollo de aplicaciones Web
- Capacitación recibida respecto del proceso de Desarrollo y Mantenimiento de Software

Entre los resultados que tuvimos se encuentran:

- Desarrollo dentro de los tiempos establecidos al inicio del proyecto
- Alta nivel de satisfacción del cliente tanto con el software entregado así como con los diversos entregables generados en las diferentes etapas del proceso.
- Disminuir el rechazo al cambio por parte de los integrantes de la Subdirección de Sistemas referente a la implantación del modelo de proceso de software.
- Documentación de todo el proceso de desarrollo integrado por los siguientes entregables:
 - Código fuente de la aplicación versionado y comentado
 - Especificación de Requerimientos

- Documento de Análisis y Diseño
- Reporte de Actividades de cada uno de los miembros del equipo de trabajo
- Reportes de Pruebas
- Matriz de rastreo
- Minuta de las reuniones con el cliente así como con el equipo de trabajo
- Plan de Desarrollo
- Reporte de Mediciones, Observaciones y Sugerencias de Mejora
- Manuales técnico y de usuario
- Software de mayor calidad sustentado en un programa de mejora y un proceso de desarrollo definido.
- Mayor experiencia de los miembros del equipo de trabajo en el desarrollo de aplicaciones Web.

CONCLUSIONES

El empleo de una metodología de procesos facilitó en gran medida el desarrollo del proyecto debido a que se realizaron de manera sistemática las actividades de análisis, diseño, construcción, integración y pruebas de productos de software cumpliendo con los requerimientos especificados por el cliente.

Igualmente se logró que los productos de salida fueran consistentes con los productos de entrada en cada fase del ciclo de desarrollo mediante las actividades de verificación, validación o prueba. Así como se llevaron a cabo las actividades de las fases de un ciclo mediante el cumplimiento del Modelo de Procesos de Software (ver Figura 1).

El apego al framework de Struts eficientó la integración de los diferentes módulos que componen la aplicación. De igual forma obligó a los desarrolladoras a homologar la codificación de los módulos que les fueron asignados.

Principalmente, el contar con la verificación y validación de los requerimientos por parte del cliente nos evitó el tener que rediseñar la aplicación en fases posteriores al levantamiento de requerimientos ahorrando tiempo y recursos a la DGSCA.

La creación de un análisis y diseño es fundamental para poder tener un rápido desarrollo, una fácil integración y un buen entendimiento con el grupo de desarrolladores.

REFERENCIAS

Bibliografía

- Modelo de Procesos para la Industria de Software
Oktaba, Hanna
Alquicira Esquivel, Claudia
Bastida Martínez, Brenda
Versión 1.3, 2005
Secretaría de Economía, México
- Tecnologías de Servidor con Java: Servlets, JavaBeans, JSP
Esteban, Ángel
Grupo EIDOS Consultoría y Documentación Informática, S.L., 2000, México
- Documentación del Directorio de Personas e Instituciones
UNAM-DGSCA, 2007
IMTA_Acta de Proyecto
IMTA_Documento de Diseño y Arquitectura de SW.
IMTA_Especificación de Requerimientos
IMTA_Manual Técnico
Convenio IMTA-CRIM
IMTA_Reporte de Evaluación de Competencias

Mesografía

- Introducción a la Arquitectura de Software versión 1.0
Carlos Billy Reynoso
Marzo 2004
Universidad de Buenos Aires – MSDN en español
http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/intro.asp
- Struts Framework Documentation
<http://struts.apache.org/1.3.10/index.html>
- Unified Modeling Language
<http://www.uml.org/>