

**Universidad de las Ciencias Informáticas**  
**“Facultad 6”**



**“Sistema informático para la gestión de los servicios telemáticos en el grupo de redes de la Universidad de las Ciencias Informáticas”**

Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas

**Autor(es):**

Yordan Laza Díaz

Yoandy Silva Vázquez

**Tutor(es):**

Ing. Yusdenis Sánchez Perodín

Ing. Rolando Toledo Fernández

“Ciudad de la Habana, mayo de 2010”

## DECLARACIÓN DE AUTORÍA

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

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

**Yordan Laza Díaz**

\_\_\_\_\_  
Firma del Autor

**Yoandy Silva Vázquez**

\_\_\_\_\_  
Firma del Autor

Ing. Yusdenis Sánchez Perodín

\_\_\_\_\_  
Firma del Tutor

Ing. Rolando Toledo Fernández

\_\_\_\_\_  
Firma del Tutor

## DATOS DE CONTACTO

### TUTORES:

Ing. Yusdenis Sánchez Perodín  
Universidad de las Ciencias Informáticas, Habana, Cuba.  
Email: [ysanchezp@uci.cu](mailto:ysanchezp@uci.cu)

Ing. Rolando Toledo Fernández  
Universidad de las Ciencias Informáticas, Habana, Cuba.  
Email: [rtoledo@uci.cu](mailto:rtoledo@uci.cu)

## AGRADECIMIENTOS

*Agradecemos a todas aquellas personas que nos han acompañado a lo largo de nuestras vidas y que de una forma u otra han contribuido a nuestra formación profesional y a la realización de este trabajo. En especial a Yusdenis Sánchez y Rolando Toledo, por sus imprescindibles aportes, que han hecho posible la consumación de este sueño. A todos, gracias, ustedes forman parte de este logro.*

## DEDICATORIA

*A las personas que más amo en este mundo: mi familia.*

*A mi mamá, a mi abuela por saberme educar y guiar hacia caminos buenos y decisivos en mi vida, por poner en mis manos mi destino y confiar en mí, por ser tan comprensivas y estar presente en momentos tan difíciles en mi carrera dándome apoyo y fuerzas para seguir adelante y por confiar siempre en que yo podía hacer mi sueño realidad.*

*A mi padre que siempre se ha enorgullecido de mí, a mi hermana, abuelo, a mis tías queridas, a mis mejores amigos (as) de la brigada #2 del Grupo de Asistencia Técnica.*

*A mi gran amor Yanet quien más ha influido en mi formación profesional por estar presente en mi vida universitaria desde el comienzo hasta el final, queriéndome, ayudándome, dándome apoyo, gracias te doy por aparecer y quiero que sepas que desde que eso sucedió mi vida cambió por completo.*

Yordan

*A mi madre, que le debo todo lo que soy y he alcanzado en mi vida.*

*A mi abuela, mi segunda madre, que siempre ha estado ahí para apoyarme y darme su cariño.*

*A mi padre, por haberme guiado desde pequeño por el camino correcto de la vida.*

*A mi novia, por darme fuerza en todo momento cuando más la he necesitado.*

*A mi familia sin que se me quede uno, esto es una victoria de ustedes también.*

Yoandy

## RESUMEN

La vertiginosa evolución de las tecnologías de la información ha traído consigo el consecuente desarrollo de las redes telemáticas existentes, así como el crecimiento acelerado de los avances y aplicaciones de los servicios telemáticos. De igual modo la automatización de la gestión de los servicios telemáticos ha pasado a ser una necesidad para las instituciones u organizaciones que se dedican a la prestación de los mismos, debido a las nuevas necesidades ciudadanas y la expansión global de los servicios.

En la Universidad de las Ciencias Informáticas no existe una solución ágil para llevar a cabo el proceso de gestión de las solicitudes de los servicios telemáticos efectuadas por los líderes de los proyectos productivos. Sin embargo, el mismo es llevado a cabo de forma manual y tras el aumento considerable del número de proyectos con que cuenta dicha institución la información a controlar ha crecido notablemente, lo que trae consigo que se convierta en un proceso poco eficiente. Razón por la cual surge la presente investigación con el objetivo fundamental de diseñar e implementar un sistema informático para la gestión de los servicios telemáticos en la dirección de redes de la Universidad de las Ciencias Informáticas (UCI). Este sistema representa, sin dudas, una mejora notable en el manejo de toda la información referente a la gestión de los servicios telemáticos, específicamente a los servicios de asignación de dirección IP y apertura de puertos lógicos, ofreciendo un control riguroso y actualización inmediata de datos, sustituyendo todo el trabajo manual por tecnología.

## PALABRAS CLAVES

Asignación de IP, Apertura de puertos, Servicios telemáticos.

TABLA DE CONTENIDOS	
AGRADECIMIENTOS.....	I
DEDICATORIA.....	II
RESUMEN.....	III
INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	5
1.1 SERVICIOS TELEMÁTICOS EN LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS.....	5
1.2 PROCESO DE SOLICITUDES DE SERVICIOS TELEMÁTICOS EN LA UCI.....	7
1.3 INFORMATIZACIÓN DEL PROCESO DE GESTIÓN DE SOLICITUDES.....	8
1.3.1 SERVICEDESK PLUS.....	8
1.3.2 ARTOLOGIK HELPDESK.....	9
1.3.3 NUMARA FOOTPRINTS.....	10
1.3.4 INTRALITIA HELP DESK.....	10
1.4 HERRAMIENTAS Y TECNOLOGÍAS DE SOPORTE AL DESARROLLO.....	11
1.4.1 NORMAS PARA EL DESARROLLO DE SOFTWARE EN LA UCI.....	11
1.4.2 METODOLOGÍA PARA EL DESARROLLO DE SOFTWARE.....	12
1.4.3 PROGRAMACIÓN EXTREMA.....	13
1.4.4 PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE.....	14
1.4.5 UML COMO LENGUAJE DE MODELADO.....	17
1.4.6 HERRAMIENTA CASE VISUAL PARADIGM.....	18
1.4.7 ARQUITECTURA DEL SISTEMA.....	18
1.4.8 LENGUAJES DE PROGRAMACIÓN.....	20
1.4.9 SISTEMA GESTOR DE BASE DE DATOS (SGBD).....	21
1.4.10 SERVIDOR WEB.....	23
1.4.11 FRAMEWORK DE DESARROLLO.....	24
1.4.12 ENTORNO DE DESARROLLO INTEGRADO.....	26
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA.....	28

2.1	MODELO DEL NEGOCIO .....	28
2.1.1	REGLAS DEL NEGOCIO .....	29
2.1.2	ACTORES DEL NEGOCIO.....	29
2.1.3	TRABAJADORES DEL NEGOCIO .....	30
2.1.4	DIAGRAMA DE CASOS DE USO DEL NEGOCIO.....	31
2.1.5	DESCRIPCIÓN DE LOS CASOS DE USO DEL NEGOCIO.....	32
2.1.6	DIAGRAMAS DE ACTIVIDADES .....	36
2.1.7	MODELO DE OBJETO .....	38
2.2	ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE .....	38
2.2.1	REQUERIMIENTOS FUNCIONALES.....	38
2.2.2	REQUERIMIENTOS NO FUNCIONALES .....	42
2.3	MODELO DE CASOS DE USO DEL SISTEMA .....	44
2.3.1	DEFINICIÓN DE LOS ACTORES DEL SISTEMA .....	45
2.3.2	PATRONES DE CASO DE USO .....	46
2.3.3	DIAGRAMA DE CASOS DE USO DEL SISTEMA.....	47
2.3.4	DESCRIPCIÓN TEXTUAL DE LOS CASOS DE USO DEL SISTEMA.....	48
CAPÍTULO 3: DISEÑO DEL SISTEMA.....		49
3.1	ESTILO ARQUITECTÓNICO UTILIZADO .....	49
3.2	PRINCIPALES PATRONES DE DISEÑO UTILIZADOS .....	50
3.3	DIAGRAMAS DE CLASES DEL DISEÑO .....	51
3.4	DIAGRAMAS DE INTERACCIÓN .....	53
3.5	DISEÑO DE LA BASE DE DATOS.....	55
3.5.1	DIAGRAMA DE CLASES PERSISTENTES .....	55
3.5.2	DIAGRAMA MODELO RELACIONAL .....	56
3.5.3	DESCRIPCIÓN DE LAS TABLAS .....	<b>¡Error! Marcador no definido.</b>
3.6	DIAGRAMA DE DESPLIEGUE .....	57
3.6.1	DESCRIPCIÓN DE LOS NODOS.....	58
3.6.2	DESCRIPCIÓN DE LAS INTERFACES DE COMUNICACIÓN .....	59
3.7	PRINCIPIOS DE DISEÑO UTILIZADOS .....	60

3.8 ESTÁNDARES EN LA INTERFAZ DE LA APLICACIÓN .....	60
3.9 TRATAMIENTO DE ERRORES .....	61
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA.....	63
4.1 DIAGRAMA DE COMPONENTES. ....	63
4.2 MODELO DE PRUEBAS.....	64
4.2.1 CASO DE USO: Crear Solicitud de Servicio Telemático .....	64
CONCLUSIONES GENERALES.....	68
RECOMENDACIONES.....	69
REFERENCIA BIBLIOGRÁFICA .....	70

## INTRODUCCIÓN

La información es, sin lugar a dudas, la materia prima imprescindible para crear conocimientos que permitan afrontar las diferentes situaciones que se van presentando en el quehacer cotidiano de la sociedad, el cual gracias al auge de las Tecnologías de la Informática y las Comunicaciones (TIC) tienen una mayor posibilidad de enriquecerse.

De todos los elementos que integran las TIC, el que más ha revolucionado la forma en que la sociedad aprecia el mundo es Internet, que no sólo abre las puertas de una nueva era sino que proporciona una conexión con un mundo en el que se puede hacer lo que años atrás era inconcebible.

El Internet no es simplemente una vía mediante la cual se puede obtener la información necesaria para desarrollarse física y mentalmente, también da la posibilidad de sumergirse en un mundo sin distancias, en el cual se pueda recrear y comerciar además de poner en las manos de la humanidad una potente herramienta para realizar sus investigaciones. Una de las facilidades que propicia el uso de Internet es la automatización de los disímiles procesos que se llevan a cabo en las esferas de la sociedad.

Muchos son los sistemas que han emergido en el mundo para lograr una correcta gestión de estos importantes procesos. Cuba, a pesar de estar bloqueada, también marcha al ritmo de esta revolución tecnológica. Para lograr llevar esta nueva era hasta el rincón más intrincado de la isla, se están desarrollando varios programas como parte de la Batalla de Ideas destinados a dotar al país de profesionales preparados.

La Universidad de las Ciencias Informáticas (UCI), fundada en el año 2002 como primera universidad de la batalla de ideas es indiscutiblemente el ejemplo más destacado de esta iniciativa, la cual se ha adentrado en este mundo del desarrollo de software y cuenta hoy con varias facultades especializadas con diferentes perfiles de desarrollo. Con vista a la organización de estos planes surge la Infraestructura Productiva compuesta por un grupo de direcciones que atienden los diferentes proyectos que lleva a cabo la universidad. Entre las direcciones que pertenecen a esta infraestructura se encuentra la Dirección de Informatización encargada de convertir la UCI en un prototipo para la informatización de la sociedad cubana, basado en el funcionamiento armónico de las tecnologías y los servicios informáticos, propiciando

el uso ordenado y masivo de las TIC. Esta dirección organiza, coordina, chequea y diseña la informatización de todos los procesos internos en cada una de las áreas que rigen la vida de la universidad, desde la perspectiva de una Ciudad Digital. Además propone la arquitectura base para la informatización de la UCI y garantiza el soporte a los productos terminados.

En la actualidad existe deficiencia en la automatización de los procesos asociados a ciertas áreas de la universidad provocando retrasos en las prestaciones de los servicios solicitados. La Dirección de Gestión Tecnológica (DGT) es una de las áreas más afectadas por este fenómeno. La cual tiene la misión de garantizar una alta disponibilidad tecnológica que proporcione los procesos fundamentales de la universidad en un escenario ideal para el alcance de sus objetivos.

Para lograr dicha disponibilidad la DGT cuenta con varios grupos donde se encuentran los especialistas de redes que son los encargados de investigar las causas subyacentes a toda alteración, real o potencial, del servicio de las Tecnologías Informáticas (TI) los cuales determinan posibles soluciones a las mismas y son los que proponen las peticiones de cambios necesarios para elevar la calidad de los servicios.

En la actualidad la Universidad cuenta con disímiles proyectos productivos en los que se desarrollan importantes aplicaciones informáticas y sus líderes no cuentan con una vía rápida y eficiente para la solicitud de los servicios necesarios para publicar dichas aplicaciones. Cuando desean solicitar un nuevo servicio deben llenar una planilla con los datos referentes a su proyecto, explicar detalladamente el servicio que solicitan así como sus datos personales. Esta planilla se debe enviar a la DGT y Dirección de Seguridad Informática (DSI) para su posterior aprobación donde en la mayoría de los casos el usuario desconoce el seguimiento de la misma, lo cual limita la interactividad de los datos y propicia demoras en la entrega de la información.

El registro y revisión de todo el volumen de información acumulada al respecto, se realiza de forma manual, lo cual propicia demora en la respuesta a las necesidades de los líderes de proyectos. También existe la posibilidad de que ocurra el deterioro o la pérdida de las planillas lo que provoca que a la hora de realizar búsquedas ante alguna reclamación por parte de los usuarios dicho proceso se ralentice. No existe un control estricto y ágil sobre los diferentes puertos que están abiertos en la universidad y las aplicaciones que hacen uso de ellos, propiciando vulnerabilidades en la seguridad de la institución frente

a posibles ataques piratas, además de no garantizar un almacenamiento confiable y restringido de la información reflejada en las planillas de solicitudes para personas no autorizadas.

Las diversas problemáticas planteadas anteriormente conllevan a determinar como **Problema Científico** ¿Cómo mejorar el proceso de gestión de los servicios telemáticos en el grupo de redes de la Universidad de las Ciencias Informáticas? Definiendo como **Objeto de Estudio** los servicios telemáticos en la Universidad de las Ciencias Informáticas, enmarcando el **Campo de Acción** el proceso de gestión de los servicios telemáticos en el grupo de redes de la Universidad de las ciencias informáticas.

El **Objetivo General** de la investigación se define como: Desarrollar un sistema informático para la gestión de los servicios telemáticos en el grupo redes de la Universidad de las Ciencias Informáticas. Dicho objetivo ha sido desglosado en los siguientes **Objetivos Específicos**:

- ✓ Definir los principales elementos teóricos-metodológicos para el análisis del problema a resolver.
- ✓ Identificar funcionalidades del sistema informático para la gestión de los servicios telemáticos en la dirección de redes de la Universidad de las Ciencias Informáticas.
- ✓ Realizar el diseño del sistema.
- ✓ Implementar el sistema.

Para dar cumplimiento a los objetivos propuestos se plantean las siguientes **Tareas**:

- ✓ Análisis de los elementos fundamentales, tecnologías y herramientas informáticas para el desarrollo de software.
- ✓ Modelación del negocio.
- ✓ Identificación de los requisitos funcionales.
- ✓ Identificación de los requisitos no funcionales.
- ✓ Elaboración de los diagramas de clases de diseño.
- ✓ Elaboración de los diagramas de base de datos.
- ✓ Elaboración de los diagramas de componentes.
- ✓ Implementación de los componentes.
- ✓ Realización de pruebas funcionales.

Para una mejor comprensión del Trabajo de diploma, el contenido ha sido desglosado en 4 capítulos, conclusiones, recomendaciones, referencias bibliográficas, los anexos, mediante los cuales se percibirá con mayor facilidad la investigación realizada y por últimos un glosario de términos.

**CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA:** Se describen los conceptos fundamentales asociados al dominio del problema, se expone un estado del arte del tema tratado, tanto a nivel nacional como internacional, se hace un análisis crítico entre las soluciones ya existentes y la propuesta, explica y justifica, además, las tendencias, tecnologías y herramientas en las que se apoya la solución al problema.

**CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA:** Se describe el flujo actual de los procesos a través del Modelo del negocio, el cual propicia el análisis necesario para definir las funcionalidades del sistema a desarrollar. Estas se describen detalladamente mediante la especificación de requerimientos, descripción de casos de uso y representación gráfica.

**CAPÍTULO 3 DISEÑO DEL SISTEMA:** En este capítulo se describen los patrones utilizados en el diseño de la aplicación, se define la estructura y los elementos del diseño, se muestran los diagramas de clases de diseño por cada caso de uso del sistema, el modelo de datos y el diagrama de despliegue.

**CAPÍTULO 4 IMPLEMENTACIÓN Y PRUEBA:** En este capítulo se describe como fue implementada la herramienta en términos de componentes. Se desarrollarán varias pruebas funcionales usando el método de caja negra para garantizar que se han cumplido los requisitos funcionales.

# CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

## INTRODUCCION

Este capítulo tiene como principal objetivo transmitir los aspectos teóricos fundamentales con los que se contará para el desarrollo del presente trabajo. Se brinda una visión general de los aspectos relacionados con los servicios telemáticos, los conceptos necesarios para la comprensión del problema planteado y la solución propuesta. Finalmente se da una descripción de las herramientas, tecnologías y metodologías a utilizar en el desarrollo de la solución propuesta.

### 1.1 SERVICIOS TELEMÁTICOS EN LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Un servicio es el conjunto de prestaciones que un ordenador o sistema informático facilita a otros sistemas informáticos, o a otros ordenadores que están conectados a este (1).

La Universidad de las Ciencias Informáticas (UCI) cuenta hoy con varias facultades, especializadas en diferentes perfiles de desarrollo con una amplia gama de información y servicios telemáticos que son: Correo Electrónico, Internet, Mensajería Instantánea, Alojamiento Web, Apertura de Puertos, Acceso Remoto y Búsqueda de Software; estos constituyen algunos de los servicios que disfrutan los profesores, trabajadores y estudiantes de la Universidad, los cuales se solicitan en dependencia de las labores que desempeñan y las necesidades de utilizarlos en su actividad cotidiana.

A continuación se da una breve descripción de los servicios telemáticos existentes en la UCI.

**Servicio de Correo Electrónico:** Es un servicio que permite el intercambio de mensajes a través de sistema de comunicación electrónica. Se utiliza principalmente para denominar al sistema que brinda este servicio vía internet mediante el protocolo SMTP (Simple Mail Transfer Protocol). Los mensajes de correo electrónico posibilitan el envío, además de texto, de cualquier tipo de documento digital (2).

**Servicio de Alojamiento Web:** Es el servicio que provee a los usuarios de internet de un sistema para poder almacenar información, imágenes, videos o cualquier contenido accesible vía web. Por lo general los Web Host son compañías que proporcionan espacio de un servidor a sus clientes.

**Servicio de Apertura de Puertos Lógicos:** En un ordenador, un puerto es la puerta lógica mediante la cual se realiza la comunicación entre el cliente y el servidor de cualquier servicio. A través de este fluye la información, entrando, saliendo o en ambas direcciones. Por ejemplo, un servidor web utilizará un puerto por el que se conectaran todos los clientes, el puerto estándar asignado a este servicio es el 80 y por lo tanto no hay que ponerlo en el navegador de internet, pero pudiera ser otro y en ese caso se colocaría después de la URL seguido del símbolo de dos puntos.

Cada servicio corriendo en un servidor escucha en un número de puerto particular. La mayoría de estos servicios tienen números de puerto estándares pero también pueden escuchar por puertos no estándares en cuyo caso dicho puerto se debe especificar para su correcto funcionamiento. En algunas instituciones, por motivos de seguridad informática varios puertos se encuentran cerrados previendo posibles ataques; cuando se desee instalar aplicaciones que usan algunos de los puertos cerrados se debe proceder a abrirlos siguiendo ciertas políticas que deben estar definidas en la institución o empresa donde se realiza dicha apertura.

**Servicio de Asignación de Dirección IP:** Se basa en asignar una dirección IP la cual puede ser estática o dinámica a cada estación de trabajo, impresora o servidor de una red. Los servidores deben recibir una dirección IP estática de modo que las estaciones de trabajo y otros dispositivos siempre sepan como acceder a los servicios requeridos.

**Servicio de Proxy:** Hace referencia a un programa o dispositivo que realiza una acción en representación de otro. Su finalidad más habitual es la de servidor proxy, que sirve para permitir el acceso a Internet a todos los equipos de una organización cuando sólo se puede disponer de un único equipo conectado, esto es, una única dirección IP.

**Servicio de Administración Remota:** Se considera Administración Remota a la funcionalidad de algunos programas que permiten realizar ciertos tipos de acciones desde un equipo local y que las mismas se ejecuten en otro equipo remoto (3).

**Servicio de Ficheros:** Son los sistemas que comparten información en forma de ficheros a través de una intranet. Proporcionan acceso a ficheros almacenados en un servidor. Prestaciones y una fiabilidad similares (o mejores) que en almacenamiento local con la ventaja que se reducen discos locales en los

clientes, los costes de mantenimiento de información y facilita la implementación de otros servicios (nombramiento, autenticación o Impresión).

**Servicio de Controlador de Dominio:** Es un servidor que se encarga de la seguridad de un dominio, es decir, administra toda la información correspondiente a usuarios y recursos de su dominio. Todo dominio necesita al menos un controlador (4).

**Servicio de Mensajería Instantánea:** La mensajería instantánea es una tecnología que permite enviar y recibir mensajes de forma casi instantánea a través de una conexión a Internet (5).

## 1.2 PROCESO DE SOLICITUDES DE SERVICIOS TELEMÁTICOS EN LA UCI

De los servicios anteriormente descrito, la asignación de dirección IP, la apertura de puertos lógicos, el alojamiento web y la administración remota son servicios que no se le brinda a toda la comunidad universitaria, la persona o grupo de proyecto que desee disfrutar de estos debe realizar su solicitud siguiendo el proceso establecido por la dirección de seguridad informática de la Universidad.

Específicamente a la dirección de redes de la universidad, área para la cual se realiza la presente investigación, le corresponde gestionar todo el proceso relacionado con la asignación de direcciones IP y la apertura de puertos lógicos; servicios que actualmente se le brinda a determinados proyectos productivos que así lo requieran. Actualmente en la Universidad la solicitud de los servicios telemáticos se realiza siguiendo esquemas tradicionales en los cuales la persona o proyecto interesado debe acudir a la dirección de redes y llenar los documentos necesarios, luego esperar a que le sea aprobada la solicitud y se le comunique a través de correo electrónico o vía telefónica.

A medida que ha aumentado el número de proyectos interesados en hacer uso de los servicios que brinda la dirección de redes, la gestión de las solicitudes y el seguimiento de las mismas se hace más engorrosa. Si los directivos necesitan conocer algún reporte estadístico al respecto se deben contabilizar todas las planillas existentes así como el estado actual de cada una, en caso que algún usuario reclame algo al respecto de una solicitud hecha por él hay que acudir también a la revisión de las planillas.

### 1.3 INFORMATIZACIÓN DEL PROCESO DE GESTIÓN DE SOLICITUDES

Si el proceso descrito anteriormente se realizara a través de alguna herramienta informática se disminuiría la probabilidad de errores a cometer, se agilizaría el tiempo de respuesta a los usuarios, se podría realizar reportes estadísticos actualizados con un elevado nivel de precisión y aumentaría el nivel de control de los servicios brindados.

En revisiones bibliográficas realizadas no se encontraron software dedicados completamente a la gestión de solicitudes, en cambio fueron estudiadas algunas aplicaciones informáticas que pueden ser utilizadas para tal fin por contener algunos módulos que posibilitan dicha tarea. A continuación se describirán estas aplicaciones.

#### 1.3.1 SERVICEDESK PLUS

ServiceDesk Plus es una solución de servicios técnicos y gestión de inventario. Ofrece un paquete integrado con la gestión de incidencias, inventario automático, gestión de activos, compras, gestión de contratos, portal de autoservicio para usuarios finales y gestión de conocimientos.

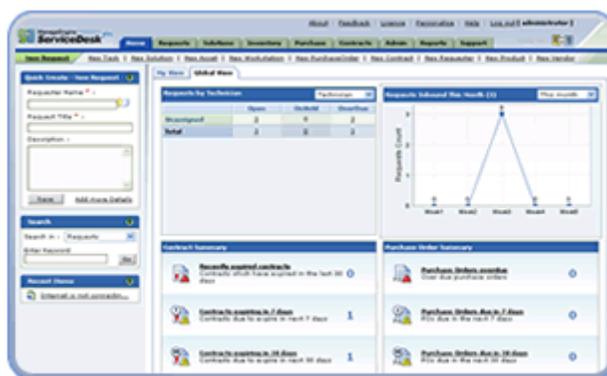


Figura 1.1 SERVICEDESK PLUS

Esta aplicación tiene un módulo llamado Gestión de Contratos que puede ser adaptado al proceso de gestión de solicitudes de servicios que se lleva a cabo en la dirección de redes de la universidad, tendría como principal ventaja que se haría uso de una aplicación ya probada y que presenta otros módulos que

pueden ser utilizados para otros fines como es el caso del portal de servicios, la base de conocimiento y la gestión de activos e inventarios (6).

Este sistema tiene como principal desventaja que no es un software libre, es un software para el cual se debe pagar una licencia, en el caso de la universidad debería pagar una licencia comercial que debe renovar cada un año el costo de la misma estaría actualmente sobre los 30 000 dólares, pero es un monto que puede variar cada vez que se renueve el contrato.

### 1.3.2 ARTOLOGIK HELPDESK

Artologik HelpDesk, Software de gestión de incidencias: Este es un sistema de soporte basado en Web para gestionar fácilmente las incidencias tanto internas como externas, enviando las solicitudes de los usuarios al personal de soporte especializado.

Presenta un módulo complementario llamado HD-Advanced, para aquellos clientes que necesitan un sistema de gestión de soporte más complejo, este permite crear un sistema para delegar solicitudes en diferentes niveles clasificando las organizaciones de clientes por grupos de organizaciones vinculadas a grupos de soporte, y estableciendo una forma de trabajo más estructurada, para una mejor provisión del servicio para sus clientes y una reducción del volumen de trabajo para el personal de soporte (7).

Dicho módulo podría ser utilizado para la gestión de solicitudes que se le realiza a la dirección de redes de la universidad pero presenta las siguientes desventajas

Debe ser instalado sobre sistemas operativos Windows, ya sea NT, XP, 2000, 2003 o 2008 server.

El gestor de bases de datos que utiliza es MS SQL server.

Es un software privativo y dadas las características de la UCI se debe pagar una licencia Enterprise cotizando alrededor de 3000 Euros.

### 1.3.3 NUMARA FOOTPRINTS

Numara FootPrints ofrece un paquete integrado con la gestión de solicitudes, eventos, configuración y servicios de activos, cambios, conocimientos, nivel de servicio, reportes de servicios, entre otros.

Construida en una plataforma de gestión de servicio abierta y flexible. Numara FootPrints trabaja con navegadores web que operen en cualquier plataforma, incluyendo todos los sistemas de Microsoft Windows, Apple Mac, UNIX, y Linux. Soporta SSL (capa de conexión segura) que es un protocolo de Internet (visto como HTTPS: en un URL) diseñado para encriptar la información que pasa entre el usuario y el sistema o con una fuente segura (8).

A pesar de integrar un grupo de módulos que fortalecerían la gestión de solicitudes tiene como principales desventajas que sólo se puede desplegar en servidores que tengan instalado los sistemas operativos Windows Server 2000, 2003 o 2008; también se puede destacar que no es un software libre, se necesita pagar una licencia comercial la cual encarece por el pago de la licencia de el sistema operativo de los servidores y del gestor de bases de datos, pues aconsejan el uso de MS SQL Server u Oracle.

### 1.3.4 INTRALITIA HELP DESK

INTRALITIA HELP DESK - Control y administración del soporte a usuarios

Es una herramienta para facilitar la administración de servicios de soporte en el marco de trabajo ITIL (Information Technology Infrastructure Library)

El módulo Help Desk de intralitia es una herramienta que ofrece una solución sencilla, rápida y económica para controlar el área de servicio, que le permite:

- ✓ Administrar solicitudes desde su generación hasta su solución.
- ✓ Crear tickets y asignarlos a técnicos de forma automática.
- ✓ Conocer en qué estado del proceso de solución se encuentra cada ticket y asegurarse que ninguno quede sin seguimiento.

- ✓ Mejorar la respuesta al usuario facilitándole el registro de solicitudes y las consultas de su avance a través de una interface Web, 24 horas los 7 días a la semana.
- ✓ Realizar el seguimiento y asegurar el cumplimiento de los procesos de soporte.
- ✓ Reportar y medir la efectividad del área de soporte (9).

Esta herramienta no puede ser adoptada como solución a la problemática abordada en la presente investigación, pues al igual que la anterior, requiere ser instalada en servidores con sistemas operativos Windows Server, y para utilizarla cualquier organización debe pagar una licencia comercial cuyo valor está en dependencia de la versión y número de módulos que desee utilizar.

## 1.4 HERRAMIENTAS Y TECNOLOGÍAS DE SOPORTE AL DESARROLLO

### 1.4.1 NORMAS PARA EL DESARROLLO DE SOFTWARE EN LA UCI

La Dirección de Informatización de la Universidad de las Ciencias Informáticas (UCI), perteneciente a la Infraestructura Productiva (IP), tiene como misión: dirigir, organizar, coordinar, chequear, diseñar y definir la Informatización de todos los procesos internos en cada una de las áreas que rigen la vida de la universidad, desde la perspectiva de una Ciudad Digital, logrando una total integración de todas las entidades, flujos y/o procesos, basado en el funcionamiento armónico de la tecnología y los servicios informáticos. Propiciando además el uso ordenado y masivo de las tecnologías con que cuenta esta importante institución. Convirtiéndose en el prototipo para la Informatización de la Sociedad Cubana (10).

El proceso de informatización de la Universidad de las Ciencias Informáticas se rige en todo momento por los lineamientos arquitectónicos, los cuales tienen como objetivo estandarizar las arquitecturas de los sistemas desarrollados para la informatización de la universidad, y se encuentran disponibles de forma online en la dirección Web de la UDDI, (11) mediante un documento que establece las pautas en las líneas bases de la arquitectura para el desarrollo de sistemas informáticos en la UCI.

Dichos lineamientos son:

- ✓ Las aplicaciones Web como estándar para la Gestión de Servicios en la UCI.

- ✓ Se seguirá el patrón Modelo-Vista-Controlador (MVC) para todas las aplicaciones de la Universidad, manteniendo una clara separación de la lógica de negocios, presentación y acceso a datos. Permitiendo flexibilidad y facilidad a la hora de hacer futuros cambios.
- ✓ Como tecnología aprobada se usa el lenguaje de programación PHP para la mayoría de los sitios web y portales por ser más liviano y fácil de utilizar.
- ✓ Para la confección de sitios en PHP de manera más rápida se usa el framework Codeigniter.
- ✓ Se utilizará PostgreSQL como gestor de Bases de datos.

#### 1.4.2 METODOLOGÍA PARA EL DESARROLLO DE SOFTWARE

Una Metodología de desarrollo de software es un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software. En ella se va indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando además qué personas deben participar en el desarrollo de las actividades y qué papel deben tener (12).

Por otro lado dichas metodologías detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla. Muchas veces el proceso de desarrollo de software resulta riesgoso y se convierte en una tarea difícil hallar el modo de controlar su curso de principio a fin. El problema principal radica en cómo coordinar todas las actividades que comprende el desarrollo de un proyecto de software, sobre todo si se trata de un proyecto de gran envergadura. De modo que se torna imprescindible contar con una forma organizada y adecuadamente estructurada para trabajar.

El uso de una metodología de desarrollo de software acertada, permite obtener desarrolladores satisfechos con el software de calidad que ha sido fruto de su trabajo.

Las metodologías para el desarrollo de software pueden ser clasificadas como (13):

- ✓ Metodologías Ágiles
- ✓ Metodologías Tradicionales

## Metodologías Ágiles

Las Metodologías Ágiles basan su fundamento en la adaptabilidad de los procesos de desarrollo, promueve la formalización de procesos adaptables y tienen como principal característica la habilidad de responder al cambio. Las cuales son sin duda uno de los temas recientes en ingeniería de software que están acaparando gran interés. Prueba de ello es que se están haciendo un espacio destacado en la mayoría de conferencias y workshops celebrados en los últimos años. Es tal su impacto que actualmente existen 4 conferencias internacionales de alto nivel y específicas sobre el tema (14).

El ejemplo más destacado y exitoso de la metodología ágil, es la programación extrema o eXtreme Programming (XP) como enfoque de la ingeniería de software.

### 1.4.3 PROGRAMACIÓN EXTREMA

Programación Extrema (XP por sus siglas en inglés) es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico (15).

XP no es recomendable aplicarla al desarrollo de la herramienta debido a que la misma exige la integración total del cliente al equipo de trabajo para lograr el éxito; condición con la cual no se cuenta, ya que el cliente es un agente externo al equipo de trabajo, cuenta con la desventaja de generar poca documentación en comparación con el Proceso Unificado de Desarrollo de Software (RUP por sus siglas en inglés), además XP fue inicialmente creada para el desarrollo de aplicaciones donde el cliente no sabe muy bien lo que quiere, lo que provoca un cambio constante en los requisitos que debe cumplir la aplicación.

## Metodologías Tradicionales

Por su parte las Metodologías Tradicionales se caracterizan por exponer procesos basados en planeación exhaustiva. Esta planeación se realiza esperando que el resultado de cada proceso sea determinante y predecible. Se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán.

Están caracterizadas por:

- ✓ Ser rígidas y dirigidas por la documentación que se genera en cada una de las actividades desarrolladas.
- ✓ Ofrecer cierta resistencia a los cambios.
- ✓ Constituir un proceso mucho más controlado, con numerosas políticas y normas.
- ✓ Más artefactos.
- ✓ Más roles.
- ✓ La arquitectura del software es esencial y se expresa mediante modelos.

El ejemplo más representativo e importante de este tipo de metodologías es el Proceso Unificado de Desarrollo Software o Proceso Unificado de Rational ya que los dos nombres suelen utilizarse para referirse a un mismo concepto.

### 1.4.4 PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE

RUP es un proceso de desarrollo de software, cuyos modelos y artefactos se expresan en el Lenguaje Unificado de Modelado. RUP es un marco de trabajo de proceso, que puede ser adaptado a proyectos de mayor o menor complejidad, aplicable a diferentes esferas y ajustable a las necesidades de cada organización (16).

En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo. La Figura 1.2 representa el proceso en el que se grafican los flujos de trabajo y las fases y muestra la dinámica expresada en iteraciones y puntos de control (17).

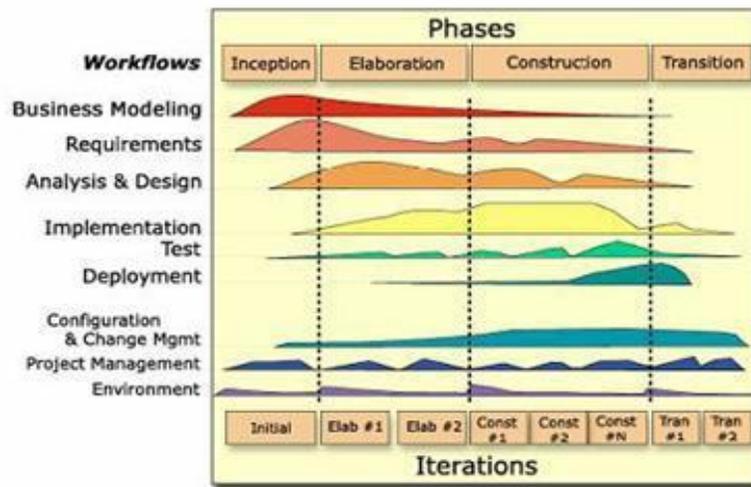


Figura 1.2 RUP en Dos Dimensiones

El ciclo de vida de RUP se caracteriza por estar (17):

- ✓ Dirigido por los casos de uso

Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.

- ✓ Centrado en la arquitectura

La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo,

desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura. El modelo de arquitectura se representa a través de vistas en las que se incluyen los diagramas de UML.

- ✓ Iterativo e incremental

RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Por ejemplo, una iteración de elaboración centra su atención en el análisis y diseño, aunque refina los requerimientos y obtiene un producto con un determinado nivel, pero que irá creciendo incrementalmente en cada iteración.

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

Fases definidas por RUP:

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

**Elaboración:** Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.

**Construcción:** Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene 1 o varios reléase del producto que han pasado las pruebas. Se ponen estos reléase a consideración de un subconjunto de usuarios.

**Transición:** El reléase ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.

El Proceso Unificado de Desarrollo de Software (RUP) fue la metodología utilizada para planificar y controlar la propuesta de este trabajo, para el análisis, diseño, implementación y documentación de sistemas orientados a objetos, por todas las características y facilidades planteadas anteriormente, que aportan organización a todo el proceso de desarrollo de software y garantizan el cumplimiento de las metas trazadas.

#### 1.4.5 UML COMO LENGUAJE DE MODELADO

UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema en los que el software juega un papel importante. La estandarización de un lenguaje de modelado es invaluable, ya que es la parte principal de comunicación. Si se quiere discutir un diseño con alguien más, ambos deben conocer el lenguaje de modelado y no así el proceso que se siguió para obtenerlo (18).

El lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad es UML. Básicamente permite a los desarrolladores visualizar los resultados de su trabajo en esquemas o diagramas estandarizados. Este lenguaje de modelado puede ser empleado de variadas formas para dar soporte a una metodología de desarrollo de software (como por ejemplo RUP), pero no define en qué metodología aplicar. Dispone de un gran número de diagramas, los cuales muestran diferentes aspectos de las entidades representadas. Su unificación le permite que sea interpretado por cualquier analista en cualquier parte del mundo y el análisis y diseño que esté notado en UML podrá ser implementado en cualquier lenguaje ya que no depende de una herramienta en específico.

Se ha convertido en el estándar de la industria, debido a que ha sido impulsado por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh, quienes fueron contratados por la empresa Rational Software Co. para crear una notación unificada en la cual basar la construcción de sus herramientas CASE. Esta notación incorpora las principales ventajas de cada uno de los métodos particulares en los que se basa (principalmente Booch, OMT y OOSE). Con UML se fusiona la notación de estas técnicas, para formar una herramienta compartida entre todos los ingenieros software que trabajan en el desarrollo orientado a objetos (19).

#### 1.4.6 HERRAMIENTA CASE VISUAL PARADIGM

Las Herramientas CASE son aplicaciones informáticas usadas por analistas, ingenieros de software y desarrolladores destinadas a aumentar la productividad en el desarrollo de software. Entre ellas una de las más destacadas es Visual Paradigm por todas las características que ofrece.

Visual Paradigm es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software, permitiendo el modelado de procesos de negocios utilizando UML. Es considerado como muy completo y fácil de usar. Permite dibujar todos los tipos de diagramas de clases, posee soporte multiplataforma, capacidades de ingeniería directa e inversa, generar código desde diagramas y generar documentación, cuenta con licencia gratuita y comercial. Además ofrece diseño centrado en casos de uso, permite la exportación e importación de imágenes, posibilitando así la asignación de nuevos estereotipos.

Se ha decidido usar Visual Paradigm como herramienta CASE para la realización de los artefactos de la ingeniería de software previa a la implementación del sistema por todas las ventajas que ofrece, teniendo en cuenta sus potencialidades para el modelado visual y soporte UML.

#### 1.4.7 ARQUITECTURA DEL SISTEMA

Una arquitectura es un entramado de componentes funcionales que aprovechando diferentes estándares, convenciones, reglas y procesos, permite integrar una amplia gama de productos y servicios informáticos, de manera que pueden ser utilizados eficazmente (20).

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

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

## Arquitectura Cliente/Servidor

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

Siempre que un cliente requiere un servicio lo solicita al servidor correspondiente y este le responde proporcionándolo. Normalmente, pero no necesariamente, el cliente y el servidor están ubicados en distintos procesadores. Los clientes se suelen situar en ordenadores personales y/o estaciones de trabajo y los servidores en procesadores departamentales o de grupo.

Tanto los programas que cumplen con las funciones de cliente y servidor, como las computadoras donde son ejecutados, son identificados con los términos cliente y servidor. El programa cliente cumple varias funciones al mismo tiempo:

- ✓ Gestiona la comunicación con el servidor.
- ✓ Solicita un servicio.
- ✓ Recibe los datos enviados por el servidor y realiza su validación.
- ✓ Maneja la interfaz con el usuario.
- ✓ Presenta los datos en el formato adecuado, brinda las herramientas y comandos para que el usuario utilice las prestaciones del servidor de manera sencilla.

El programa servidor, por su parte:

- ✓ Acepta peticiones recibidas a través de la red.
- ✓ Realiza el servicio y regresa el resultado al cliente.
- ✓ Gestiona los periféricos compartidos.
- ✓ Controla accesos concurrentes a bases de datos compartidas.

Este modelo posibilita que, a partir de una interfaz de usuario homogénea para acceder al sistema, los usuarios se mantengan al margen de los cambios o actualizaciones que se realicen al mismo y de la localización de la información. Por otra parte, este modelo elimina la necesidad de transportar grandes

volúmenes de información a través de la red y hacia las estaciones de trabajo para su procesamiento, facilitando el soporte de un mayor número de usuarios, puesto que los servidores procesan las peticiones y luego envían sólo los datos requeridos a la computadora cliente.

#### 1.4.8 LENGUAJES DE PROGRAMACIÓN

Un lenguaje de programación es aquel elemento dentro de la informática que permite desarrollar programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis, y está a disposición del programador para que este pueda comunicarse con los dispositivos hardware y software existentes.

Actualmente existen diferentes lenguajes de programación para desarrollar aplicaciones Web, tanto de lado del servidor como de lado del cliente, estos han ido surgiendo debido a las tendencias y necesidades de las plataformas, entre los que destacan: **HTML** (HyperText Markup Language), **PHP** (Personal Home Page), **Java Script**, **ASP** (Active Server Pages), **JSP** (Java Server Pages), entre otros.

Siguiendo las pautas establecidas por la dirección de informatización de la Universidad se selecciona como lenguaje de programación de lado del servidor PHP.

##### Lenguajes del lado del cliente

Los lenguajes del lado cliente son aquellos que pueden ser directamente "digeridos" por el navegador y no necesitan un pre tratamiento (22).

HTML (HyperText Markup Language) es un lenguaje de composición de documentos y especificación de ligas de hipertexto que define la sintaxis y coloca instrucciones especiales que no muestra el navegador, aunque si le indica como desplegar el contenido del documento, incluyendo texto, imágenes y otros medios soportados. HTML también le indica cómo hacer un documento interactivo a través de ligas especiales de hipertexto, las cuales conectan diferentes documentos -ya sea en su computadora o en otras- así como otros recursos de Internet, como FTP y Gopher (23).

El HTML se ha convertido en uno de los formatos más populares y fáciles de aprender que existen para la elaboración de documentos para la Web, además, su surgimiento constituyó la base o componente más

importante para los mismos. Es admitido por todos los exploradores, posee archivos pequeños y despliegue rápido. No es un Lenguaje de Programación, sin embargo, permite incluirle código en lenguajes de programación, bajo ciertos criterios, extendiendo así su capacidad y funcionalidad.

Java Script es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas. Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario. Técnicamente, es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. El lenguaje Java Script se inserta en documentos HTML (HyperText Markup Language), de forma que su código queda reflejado en la propia página y no es llamado o cargado de ninguna fuente externa. En otras palabras, los programas escritos con Java Script se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios. A pesar de su nombre, Java Script no guarda ninguna relación directa con el lenguaje de programación Java. Legalmente, Java Script es una marca registrada de la empresa Sun Microsystems (24).

Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado ya que permite de forma eficiente validar formularios, además de ser un lenguaje simple, de fácil aprendizaje y dinámico, ya que lo que hace es responder a eventos locales, producidos por el usuario, y reaccionar ante ellos en tiempo real.

#### 1.4.9 SISTEMA GESTOR DE BASE DE DATOS (SGBD)

Un Sistema de gestión de base de datos (SGBD) es un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan (25).

A la vez los SGBD permiten crear y mantener una Base de datos, asegurando su integridad, confidencialidad y seguridad. Por tanto debe permitir (16):

- ✓ Definir una BD: especificar tipos, estructuras y restricciones de datos.
- ✓ Construir la BD: guardar los datos en algún medio controlado por el mismo SGBD.
- ✓ Manipular la BD: realizar consultas, actualizarla, generar informes.

Los cuales están compuestos por un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de consulta, permitiendo definir los datos a distintos niveles de abstracción y manipular dichos datos garantizando la seguridad e integridad de los mismos. Tiene como propósito manejar de manera clara, sencilla y ordenada; un conjunto de datos que posteriormente se convertirán en información. Algunos de los SGBD comúnmente utilizados a nivel mundial son: Oracle, MySQL, PostgreSQL, entre otros.

Existen varias características que deben cumplir los SGBD, dentro de los cuales se consideran más importantes los siguientes:

- ✓ Seguridad: Los SGBD deben disponer de un sistema de permisos a usuarios y grupos de usuarios, que permitan restringir los accesos no autorizados.
- ✓ Integridad: Se debe proteger la información almacenada ante fallos de hardware, datos introducidos por descuido o cualquier situación que pueda alterar la integridad de los datos almacenados.
- ✓ Tiempo de respuesta: Se debe minimizar el tiempo que el SGBD tarda en proporcionar la información solicitada y en guardar los cambios efectuados.
- ✓ Abstracción de la información: Los usuarios deben ser ajenos a detalles sobre la localización física donde han sido almacenados los datos.
- ✓ Manejo de Transacciones: Están provistos de mecanismos para programar las modificaciones de los datos de una forma más simple que las aplicaciones que lo usan.
- ✓ Independencia: La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.

## POSTGRESQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado (26). Permite mantener la integridad de los datos y puede manejar múltiples conexiones concurrentes de los clientes, característica muy significativa de su funcionamiento. Cuenta con

una arquitectura que se caracteriza por su confiabilidad y escalabilidad y permite una manipulación potente, flexible y eficiente de la información. Posee una interfaz amigable y confiable que guía a los usuarios con menor experiencia a través del complejo proceso de creación haciendo todo más veloz y dinámico. Es muy portable, por lo que se puede ejecutar en la gran mayoría de sistemas operativos existentes en la actualidad y funciona perfectamente con grandes cantidades de datos.

#### 1.4.10 SERVIDOR WEB

Un servidor web es un programa que se encarga de mantenerse a la espera de peticiones HTTP llevadas a cabo por un cliente HTTP que se suele conocer como navegador. El navegador realiza una petición al servidor y éste le responde con el contenido que el cliente solicita. El servidor responde al cliente enviando el código HTML de la página; el cliente, una vez recibido el código, lo interpreta y lo muestra en pantalla. El cliente es el encargado de interpretar el código HTML, es decir, de mostrar las fuentes, los colores y la disposición de los textos y objetos de la página; el servidor tan sólo se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma. Algunos servidores web importantes son IIS (Internet Information Services) y Apache.

#### APACHE

Apache es uno de los servidores web más populares del mercado y considerado el más utilizado de todo el mundo. El recuento ha sido llevado a cabo por Netcraft, una empresa que recorre la Red preguntando a los distintos servidores sobre el programa que utilizan. Apache ocupa el primer lugar, con un 57,2% de los servidores; seguido de Microsoft, con un 22,9%; y de Netscape, con un 4,9% (27).

Desde su origen ha evolucionado hasta convertirse en uno de los mejores servidores en términos de eficiencia, funcionalidad y velocidad. Su robustez y estabilidad hacen que este sea confiable y se caracterice por una gran calidad profesional. La popularidad de este software libre, reconocido en muchos ámbitos empresariales y tecnológicos, se debe principalmente a las siguientes razones:

- ✓ Funciona en múltiples plataformas, lo que lo hace prácticamente universal.
- ✓ Es gratuito, distribuido bajo la licencia Apache Software License, lo que permite la modificación y adaptación de su código.

- ✓ Es un servidor altamente configurable. Sus capacidades son muy sencillas de ampliar.
- ✓ Te permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.
- ✓ Es extremadamente popular, por lo que resulta muy fácil conseguir documentación o ayuda para su uso.

Para la publicación de la aplicación informática, se ha seleccionado el servidor Web Apache por ser el más factible de acuerdo a sus características antes mencionadas, fundamentalmente gracias a su presentación como software Libre y su potente seguridad.

#### 1.4.11 FRAMEWORK DE DESARROLLO

Un framework no es más que una herramienta para simplificar y acelerar el proceso de desarrollo de una aplicación, reutilizar código ya existente y promover buenas prácticas como el uso de patrones automatizando tareas comunes mediante el encapsulado de operaciones complejas en instrucciones sencillas. Un framework Web, por tanto, se define como un conjunto de componentes (por ejemplo clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de aplicaciones Web.

#### CODEIGNITER

CodeIgniter es un conjunto de herramientas para personas que construyen su aplicación web usando PHP. Su objetivo es permitirle desarrollar proyectos mucho más rápido de lo que podría si lo escribiera desde cero, proveyéndole un rico juego de librerías para tareas comúnmente necesarias, así como una interface simple y estructura lógica para acceder a esas librerías. CodeIgniter le permite creativamente enfocarse en su proyecto minimizando la cantidad de código necesaria para una tarea dada.

El uso de CodeIgniter se basa en el patrón Modelo-Vista-Controlador, separando la vista, del negocio, y del controlador. Es realmente rápido, liviano, el núcleo del sistema sólo requiere unas pocas pequeñas librerías. Esto es en duro contraste a muchos entornos de trabajo que requieren significativamente más recursos, se encuentra libremente bajo una licencia open source Apache/BSD-style (28).

CodeIgniter tiene buena documentación en línea, además de una comunidad de usuarios muy activos, presenta muy buen rendimiento en comparación con otros framework donde la curva de aprendizaje es sumamente corta. El manejo y abstracción de base de datos es realmente bueno, e implementa el patrón Active Record. La estructura de archivos es muy clara y ofrece gran flexibilidad en caso que necesite cambiarse por algún motivo. Los mecanismos de ruteo de CodeIgniter favorecen a éste y otras aplicaciones. Es un framework con gran flexibilidad. No fuerza al usuario a utilizar una Base de Datos para un desarrollo. Además su separación de código PHP y código HTML no está basada en un template en particular, puede usarse incluso sin este. A diferencia de otros framework que incluyen un sistema de validación de usuarios por defecto, CodeIgniter no lo tiene. Ofrece el cache en el propio server de las páginas generadas, para aumentar aún más el rendimiento. Tiene sus limitaciones como todo framework, pero de lejos es el más sencillo, mejor documentado y altamente flexible.

El siguiente gráfico ilustra como fluyen los datos a través del sistema

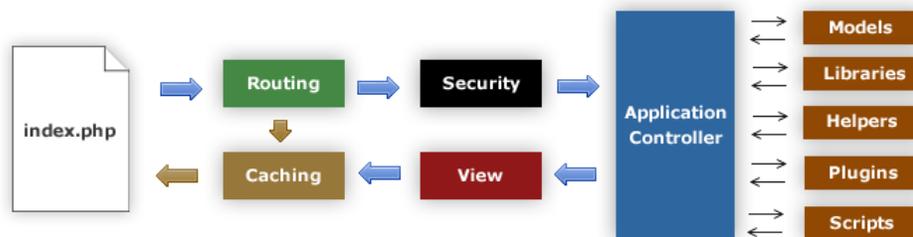


Figura 1.3 Diagrama de flujo de la aplicación

1. El index.php sirve como controlador frontal, inicializando los recursos básicos necesarios para correr CodeIgniter.
2. El Router examina la petición HTTP para determinar que debe ser hecho con él.
3. Si un archivo de caché existe, es enviado directamente al explorador, sobrepasando el sistema de ejecución normal.
4. Seguridad. Antes que el controlador sea cargado, la petición HTTP y cualquier dato suministrado por el usuario es filtrado por seguridad.
5. El controlador carga los modelos, librerías, plugins, asistentes y cualquier otro recurso necesario para procesar la petición específica.

6. La Vista finalizada es presentada entonces enviada al explorador web para ser vista. Si el cacheo está habilitado, la vista es cacheada primero para que las peticiones subsecuentes puedan ser servidas.

#### 1.4.12 ENTORNO DE DESARROLLO INTEGRADO

##### ZEND STUDIO

Se utilizó Zend Studio por ser uno de los ambientes de desarrollo integrado (IDE por sus siglas en inglés) muy completo para la programación en el lenguaje PHP, teniendo en cuenta que permite la codificación y depuración de aplicaciones escritas en este lenguaje. El programa, además de servir de editor de texto para paginas PHP agrupa todos los componentes necesario para un ciclo de desarrollo de aplicaciones proporcionando una serie de ayudas que pasan desde la creación y gestión de proyectos hasta la depuración del código, lo que acelera los ciclos de desarrollo y simplifica los proyectos complejos.

Al contrario que otros IDE para PHP, el diseño cliente/servidor de Zend Studio permite correr múltiples clientes para cada uno de los programadores de un gran proyecto desde un único servidor. Además, los clientes y el servidor de Zend Studio pueden ser ejecutados tanto bajo Linux como Windows. Incluso Zend Studio permite mantener la documentación del software autogenerándola y actualizando en cada nueva funcionalidad que se pretenda agregar.

Los expertos en PHP consideran a Zend Studio como el entorno IDE más maduro y con más características útiles. Si se desea aumentar la productividad en los desarrollos con PHP, basados en estas afirmaciones es que se selecciona este IDE para el desarrollo de la aplicación informática. Todas las opciones que dispone están pensadas con acierto por personas que conocen como nadie la tecnología.

##### CONCLUSIONES

En este capítulo ha sido abordado el estado del arte relacionado con los sistemas de gestión de solicitudes en el ámbito internacional y nacional. De estos sistemas estudiados se ha obtenido un conjunto de experiencias que han servido de apoyo a la investigación, los cuales presentan arquitecturas similares al sistema a diseñar, aportando un gran conocimiento y así colaborar de forma óptima con la arquitectura a escoger. Por otra parte se describieron las tecnologías, metodologías y herramientas que se utilizarán

para el desarrollo del sistema, es válido señalar que no se realiza un estudio comparativo entre estas, ya que la investigación se ajusta a los lineamientos arquitectónicos de la dirección de informatización en cuanto al uso de determinadas normas y tecnologías para el desarrollo de las aplicaciones informáticas a usar en el ámbito de la Universidad.

## CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA

### INTRODUCCIÓN

En el presente capítulo se describen los procesos de negocio relacionado con las solicitudes de servicios telemáticos llevada a cabo por los líderes de proyectos a favor de sus proyectos productivos, así como definir las funcionalidades que debe permitir el sistema a desarrollar. Para ello se identifican y analizan las necesidades del usuario definiendo los aspectos que serán objeto de automatización y brindando una propuesta de solución que incluye algunas reglas dentro del negocio. A su vez, quedarán identificados los actores y trabajadores del negocio, además de que se detallarán los procesos con los que interactúan a través de la representación de los casos de uso del negocio y su descripción textual. Se incluye del mismo modo el diagrama de actividades correspondiente a cada caso de uso, para una mejor comprensión de los procesos que tienen lugar en el ámbito del negocio y mediante un diagrama de clases quedará plasmada la interacción de los trabajadores con las distintas entidades. Seguidamente se detallarán los requisitos funcionales y no funcionales a considerar para la construcción del sistema que posibilitarán la correcta puesta en funcionamiento del Software una vez desarrollado. Por último se presenta el diagrama de casos de uso del sistema y luego cada uno por separado acompañado de su descripción textual.

### 2.1 MODELO DEL NEGOCIO

Con el fin de obtener una visión clara de los procesos que intervienen en el negocio así como lograr un entendimiento entre los clientes y desarrolladores se realiza la modelación del negocio. El modelo del negocio describe los procesos de negocio, en términos de casos de usos del negocio, identificando quiénes participan y las actividades que requieren automatización.

Los objetivos del modelado del negocio son:

- ✓ Comprender la estructura y la dinámica de la organización en la cual se va a implantar un sistema.
- ✓ Comprender los problemas actuales de la organización e identificar las mejoras potenciales.
- ✓ Asegurar que los consumidores, usuarios finales y desarrolladores tengan un entendimiento común de la organización.

- ✓ Derivar los requerimientos del sistema que va a soportar la organización.

Para lograr esos propósitos, el proceso de modelado permite obtener una visión de la organización para definir los procesos, roles y responsabilidades de la organización en los modelos de casos de uso del negocio y de objetos.

### 2.1.1 REGLAS DEL NEGOCIO

Las reglas de negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio (29).

- ✓ Para realizar una solicitud de servicio telemáticos de apertura de puerto el solicitante debe cumplir el rol de líder de proyecto.
- ✓ El solicitante especificará la fecha hasta cuándo tendrá validez la solicitud, de lo contrario se validará la solicitud por 6 meses. Pasado ese intervalo de tiempo se perderá el derecho sobre el servicio solicitado.
- ✓ El solicitante debe estar de acuerdo con el código de ética propuesto por la dirección de Seguridad Informática y Redes de la Universidad de las Ciencias Informáticas.
- ✓ El solicitante debe informar al grupo de redes la cancelación del servicio antes de la fecha de culminación del mismo.

### 2.1.2 ACTORES DEL NEGOCIO

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados (29).

A continuación se muestra los actores del negocio y su correspondiente justificación:

Tabla 2.1 Descripción de los actores del negocio

Actores del negocio	Justificación
Líder de Proyecto	Es aquel rol que se desempeña como representante de un proyecto convirtiéndose en usuario de los servicios del grupo de Redes. Además es el que inicia todas las acciones que dan comienzo a los procesos de negocio analizados y al mismo tiempo se beneficia de dichos procesos.

### 2.1.3 TRABAJADORES DEL NEGOCIO

Un trabajador del negocio es una abstracción de una persona (o un grupo de personas), una máquina o un sistema automatizado, que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores y manipulando entidades del negocio (29), representando estas generalmente, a los objetos que se toman, inspeccionan, producen o utilizan durante la realización de los casos de uso del negocio.

A continuación se muestra los trabajadores del negocio y su correspondiente justificación:

Tabla 2.2 Descripción de los trabajadores del negocio

Trabajadores del negocio	Justificación
Grupo de Redes	Es aquel rol que habilita el servicio solicitado luego de haber sido aprobada la solicitud por los directivos responsabilizados de esta actividad.

Director del Área Tecnológica	Es aquella persona encargada de estudiar las solicitudes de los servicios telemáticos recibidas y determinar su posible aprobación o denegación del servicio solicitado.
Director de Seguridad Informática	Es aquella persona encargada de estudiar las solicitudes de los servicios telemáticos recibidas y determinar su posible aprobación o denegación del servicio solicitado.

#### 2.1.4 DIAGRAMA DE CASOS DE USO DEL NEGOCIO

Un diagrama de casos de uso del negocio representa gráficamente a los procesos del negocio y su interacción con los actores del negocio. Se pueden agrupar casos de uso del negocio para particional el diagrama en sub-diagramas más pequeños (29), de esta forma se hace más claro el modelo de casos de uso del negocio.

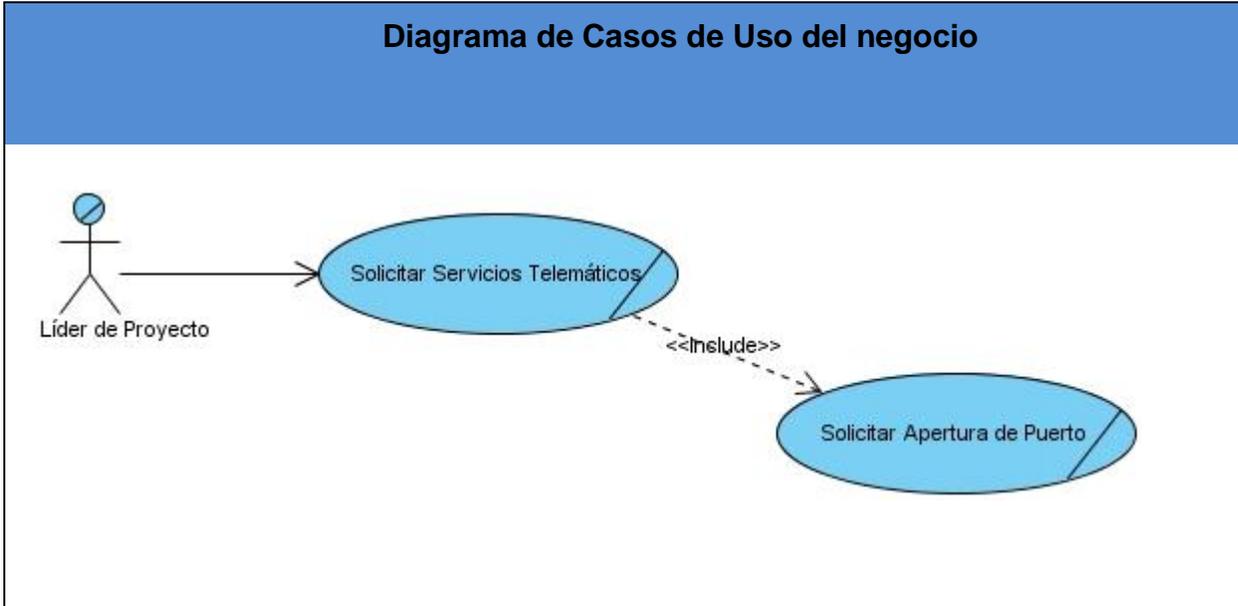


Figura 2.1 Diagrama de Casos de Uso del Negocio

### 2.1.5 DESCRIPCIÓN DE LOS CASOS DE USO DEL NEGOCIO

La Realización de un Caso de Uso de negocio muestra como colaboran los trabajadores y entidades del negocio para ejecutar el proceso. Se considera que con una descripción textual y los diagramas de actividad y diagrama de clases del Modelo de Objeto, es suficiente para describir completamente el proceso de negocio.

Tabla 2.3 Especificación del caso de uso de negocio: “Solicitar Servicios Telemáticos”

<b>Caso de Uso:</b>	<b>Solicitar Servicios Telemáticos</b>
<b>Actores:</b>	Líder de Proyecto (inicia)
<b>Trabajadores:</b>	Grupo de Redes, Director del Área Tecnológica, Director de Seguridad Informática.
<b>Resumen:</b>	El CU se inicia cuando el Líder de Proyecto solicita un servicio telemático, el Grupo de Redes le informa de los pasos a seguir para la activación del servicio. El caso de uso termina cuando el grupo de redes le informa al líder de proyecto la activación del servicio.

<b>Precondiciones:</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>
1. El Líder de Proyecto Solicita servicio.	2. El grupo de redes verifica la existencia de servicios.
	3. Si no existe servicio IP, el grupo de redes entregan planillas IP y Apertura de Puerto.
4. El Líder de Proyecto recibe planillas IP y Apertura de Puerto.	
5. El Líder de Proyecto llena planillas IP y Apertura de Puerto con sus datos personales y referentes al proyecto.	
6. El Líder de Proyecto entrega planillas IP y Apertura de Puerto.	7. La dirección tecnológica recibe planillas IP y Apertura de Puerto.
	8. La dirección tecnológica revisa planillas IP y Apertura de Puerto.
	9. Si la dirección tecnológica aprueba el servicio, esta procede a firmar las planillas IP y Apertura de Puerto.
	10. Entrega planillas IP y Apertura de Puerto firmadas.
11. El Líder de Proyecto recibe planillas IP y Apertura de Puerto respectivamente, firmadas.	
12. El Líder de Proyecto entrega planillas IP y Apertura de Puerto.	13. La dirección de Seguridad Informática recibe las planillas IP y Apertura de Puerto.
	14. La dirección de Seguridad Informática revisa las planillas IP y Apertura de Puerto.
	15. Si la dirección de Seguridad Informática aprueba el servicio, procede a firmar las planillas IP y Apertura de Puerto.

	16. La dirección de Seguridad Informática entrega planillas IP y Apertura de Puerto firmadas.
17. El Líder de Proyecto recibe planillas IP y Apertura de Puerto firmadas.	
18. El Líder de Proyecto entrega planillas IP y Apertura de Puerto firmadas.	19. El Grupo de Redes recibe planillas IP y Apertura de Puerto firmadas.
	20. Realiza servicio IP y Apertura de Puerto
	21. Informa la activación de los servicios solicitados
22. El Líder de Proyecto recibe información del servicio.	23. Culmina el CU.
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	3.1 Si existe dirección IP, ir CU Solicitar Apertura de puerto.
	9.1 Si el servicio no es aprobado por la dirección tecnológica, esta informa la no aprobación del servicio.
	15.1 Si la dirección de Seguridad Informática no aprueba el servicio, informa la no aprobación.
<b>Prioridad</b>	<b>Crítico</b>
<b>Mejoras</b>	<b>Automatizar el servicio</b>

Tabla 2.4 Especificación del caso de uso de negocio: “Realizar Apertura de Puertos”

<b>Caso de Uso:</b>	Realizar Apertura de puertos
<b>Actores:</b>	Líder de Proyecto
<b>Trabajadores:</b>	Grupo de Redes, Director del Área Tecnológica, Director de Seguridad Informática.
<b>Resumen:</b>	El CU se inicia cuando el Grupo de Redes le entrega la planilla AP y le

	informa de los pasos a seguir para la activación del servicio. El caso de uso termina cuando el grupo de redes le informa al líder de proyecto la activación del servicio.
<b>Precondiciones:</b>	
<b>Flujo Normal de Eventos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>
	1. El grupo de redes entregan planillas Apertura de Puerto.
2. Recibe planilla Apertura de Puerto.	
3. Llena planilla Apertura de Puerto.	
4. Entrega planilla Apertura de Puerto.	5. La dirección tecnológica recibe planilla Apertura de Puerto.
	6. La dirección tecnológica revisa planilla Apertura de Puerto.
	7. Si la dirección tecnológica aprueba el servicio, esta procede a firmar la planilla Apertura de Puerto.
	8. Entrega planilla Apertura de Puerto firmadas.
9. Recibe planilla Apertura de Puerto respectivamente, firmadas.	
10. Entrega planilla Apertura de Puerto.	11. La dirección de Seguridad Informática recibe la planilla Apertura de Puerto.
	12. La dirección de Seguridad Informática revisa la planilla Apertura de Puerto.
	13. Si la dirección de Seguridad Informática aprueba el servicio, procede a firmar la planilla Apertura de Puerto.
	14. La dirección de Seguridad Informática entrega planilla Apertura de Puerto firmada.
15. Recibe planilla Apertura de Puerto firmada.	

16. Entrega planilla Apertura de Puerto firmada.	17. El Grupo de Redes recibe planilla Apertura de Puerto firmadas.
	18. Realiza servicio Apertura de Puerto
	19. Informa que la activación de los servicios solicitados.
20. Recibe información.	21. Culmina el CU.
<b>Flujos Alternos</b>	
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>
	7.1 Si el servicio no es aprobado por la dirección tecnológica, esta informa la no aprobación del servicio.
	13.1 Si la dirección de Seguridad Informática no aprueba el servicio, informa la no aprobación.
<b>Prioridad</b>	<b>Crítico</b>
<b>Mejoras</b>	<b>Automatizar el servicio</b>

### 2.1.6 DIAGRAMAS DE ACTIVIDADES

Un Diagrama de Actividades permite describir gráficamente la estructura de los flujos básicos o alternativos de los casos de uso del negocio. Consiste en un grafo que contiene estados en que puede encontrarse una actividad, los mismos representan la ejecución de una sentencia de un procedimiento. Se examinan los roles específicos que juegan los trabajadores del negocio y las actividades que realizan.

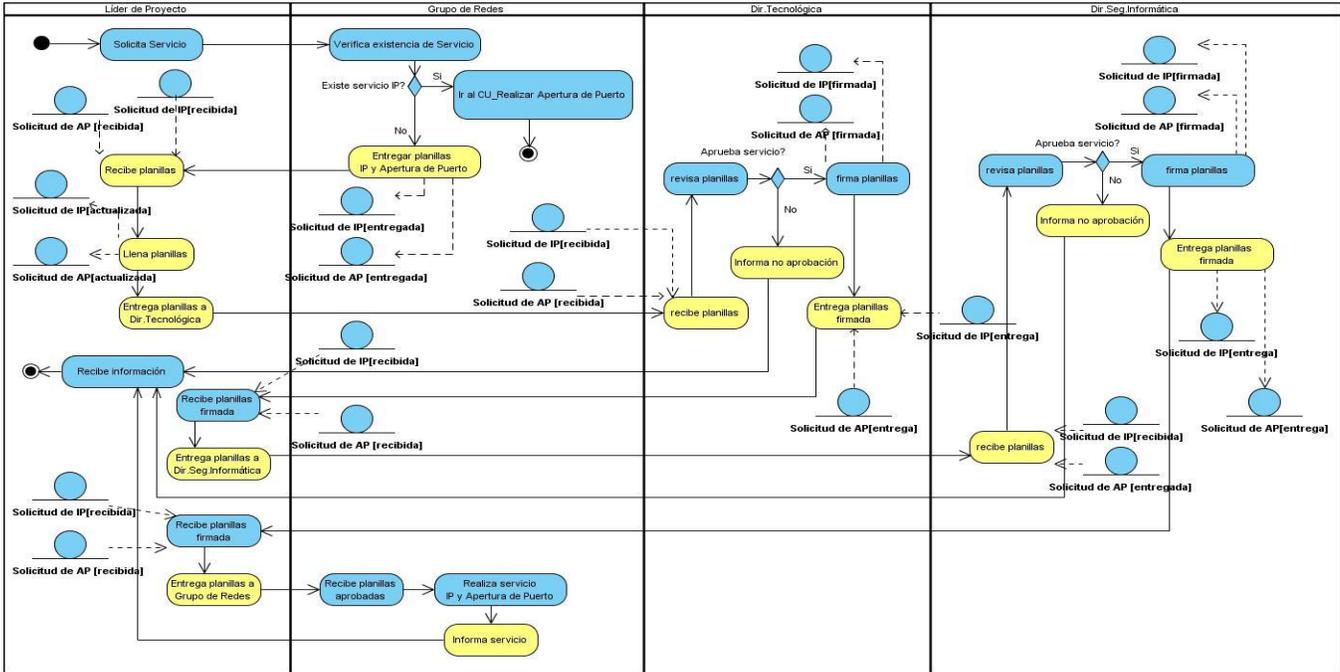


Figura 2.2 Diagrama de actividad del caso de uso del negocio “Solicitar Servicio Telemático”

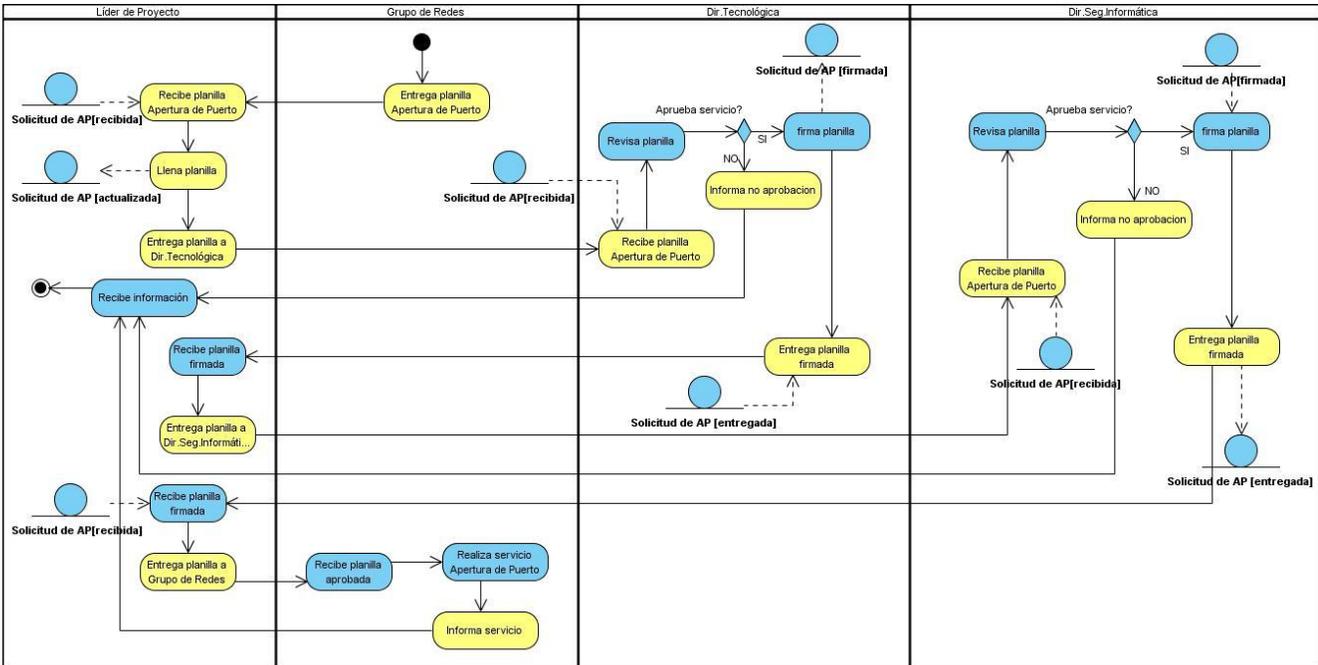


Figura 2.3 Diagrama de actividad del caso de uso del negocio “Realizar apertura de puertos”

## 2.1.7 MODELO DE OBJETO

El Modelo de objeto es el diagrama de clases mediante el cual se muestra la participación de los trabajadores y entidades del negocio y la relación entre ellos.

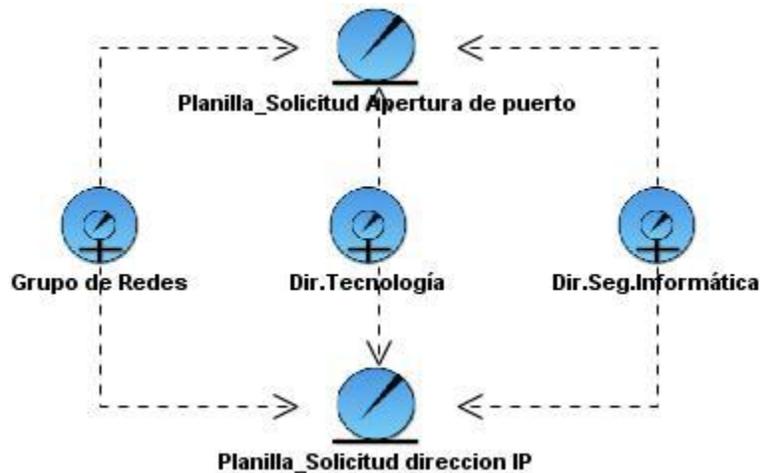


Figura 2.4 Modelo de objeto

## 2.2 ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE

Los objetivos del flujo de trabajo de requerimientos son definir el ámbito del sistema, definir una interfaz de usuarios para el sistema enfocada a las necesidades y metas del usuario, establecer y mantener un acuerdo entre clientes y otros involucrados sobre lo que el sistema debería hacer, proveer a los desarrolladores un mejor entendimiento de los requerimientos del sistema. Todos estos objetivos encaminados hacia un propósito principal que es guiar el desarrollo hacia un sistema correcto. Los requerimientos deben ser especificados por escrito, descritos como una característica del sistema a construir y posibles de probar o verificar.

### 2.2.1 REQUERIMIENTOS FUNCIONALES

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir, los mismos se mantienen invariables sin importar con que propiedades o cualidades se relacionen (30).

### **CUS 1 Autenticar Usuario**

### **CUS 2 Gestionar Usuario**

RF 2.1 Adicionar usuario.

RF 2.1.1 Asignar Rol.

RF 2.2 Mostrar usuario.

RF 2.3 Modificar Rol de Usuario.

RF 2.4 Eliminar usuario.

### **CUS 3 Gestionar Rol**

RF 3.1 Adicionar Rol

RF 3.2 Mostrar Rol

RF 3.3 Modificar Rol

RF 3.4 Eliminar Rol

### **CUS 4 Crear solicitud de servicio telemático**

RF 4.1 Crear solicitud de IP.

RF 4.2 Mostrar solicitud de IP.

RF 4.3 Crear solicitud de puerto.

RF 4.4 Mostrar solicitud de puerto.

### **CUS 5 Modificar solicitud de servicio telemático.**

RF 5.1 Mostrar solicitud de IP.

RF 5.2 Modificar solicitud de IP.

RF 5.3 Mostrar solicitud de Puerto.

RF 5.4 Modificar solicitud de Puerto.

### **CUS 6 Cancelar solicitud de servicio telemático.**

RF 6.1 Mostrar solicitud de IP.

RF 6.2 Cancelar solicitud de IP.

RF 6.3 Mostrar solicitud de Puerto.

RF 6.4 Cancelar solicitud de Puerto.

### **CUS 7 Gestionar Subredes**

RF 7.1 Agregar Subred.

RF 7.2 Mostrar subred.

RF 7.3 Modificar subred.

RF 7.4 Eliminar subred.

### **CUS 8 Gestionar protocolos**

RF 8.1 Agregar protocolo.

RF 8.2 Mostrar protocolo.

RF 8.3 Modificar protocolo.

RF 8.4 Eliminar protocolo.

### **CUS 9 Gestionar puertos**

RF 9.1 Agregar puertos.

RF 9.2 Mostrar puertos.

RF 9.3 Modificar puertos.

RF 9.4 Eliminar puertos.

### **CUS 10 Gestionar Aplicaciones**

RF 10.1 Agregar aplicación.

RF 10.2 Mostrar aplicación.

RF 10.3 Modificar aplicación.

RF 10.4 Eliminar Aplicación.

**CUS 11 Realizar reportes de Auditoría**

RF 11.1 Mostrar reporte de auditoría.

RF 11.2 Exportar reporte de auditoría.

**CUS 12 Mostrar servicios a finalizar**

**CUS 13 Mostrar planillas de solicitudes**

RF 13.1 Mostrar planillas de IP.

RF 13.2 Mostrar planillas de Puerto.

**CUS 14 Aprobar o rechazar solicitud de servicio telemático.**

RF 14.1 Mostrar solicitud de IP.

RF 14.2 Aprobar solicitud de IP.

RF 14.3 Rechazar solicitud de IP.

RF 14.4 Mostrar solicitud de Puerto.

RF 14.5 Aprobar solicitud de Puerto.

RF 14.6 Rechazar solicitud de Puerto.

**CUS 15 Crear o Rechazar solicitud de Servicio telemático**

RF 15.1 Mostrar solicitud de IP.

RF 15.2 Crear solicitud de IP.

RF 15.3 Rechazar solicitud de IP.

RF 15.4. Mostrar solicitud de Puerto.

RF 15.5 Crear solicitud de Puerto.

RF 15.6 Rechazar solicitud de Puerto.

## 2.2.2 REQUERIMIENTOS NO FUNCIONALES

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido y confiable (30), los mismos son fundamentales para el éxito del producto e indican cómo ha de comportarse el sistema.

Usabilidad:

- ✓ El futuro usuario del sistema debe poseer conocimientos básicos de computación.
- ✓ Diseño adaptable a diferentes resoluciones de pantalla.
- ✓ La navegación por el sitio no debe ser compleja.
- ✓ Todas las funcionalidades deben ser rápidamente accesibles por el usuario.

Apariencia o interfaz externa:

- ✓ Interfaces ligeras que posibiliten respuestas rápidas al usuario.
- ✓ Diseño sencillo y claro, con reconocimiento visual a través de elementos visibles que identifiquen cada una de sus funcionalidades.

Seguridad:

- ✓ La información manejada por el sistema deberá ser objeto de cuidadosa protección contra la corrupción y estados inconsistentes.
- ✓ El sistema debe estar integrado al LDAP UCI para que sus usuarios sean los registrados en el dominio UCI.
- ✓ El sistema se encarga de controlar los diferentes niveles de acceso y roles de usuarios, garantiza que la información sea vista únicamente por quien tiene derecho a verla.
- ✓ Las notificaciones de correos serán mediante el servidor de correos de la UCI.

#### Disponibilidad:

- ✓ El sistema deberá ser posible acceder las 24 horas del día para todos los usuarios autorizados.

#### Rendimiento:

- ✓ Como aplicación de tiempo real, debe tener alto grado de velocidad de procesamiento o cálculo, tiempo de respuesta y de recuperación, y disponibilidad.
- ✓ La eficiencia del producto estará determinada en gran medida por el aprovechamiento de los recursos que se disponen en el modelo Cliente/Servidor, y la velocidad de las consultas a la Base de Datos.

#### Restricciones en el diseño y la implementación:

- ✓ Se implementará el sistema usando el lenguaje PHP.
- ✓ Se utiliza el framework CodeIgniter que hará uso del patrón arquitectónico Modelo-Vista-Controlador.
- ✓ Se hará uso del sistema gestor de base de datos PostgreSQL.

#### Portabilidad:

- ✓ El sistema desarrollado es compatible con los sistemas operativos, Linux y Windows, sin necesidad de cambios significativos.

#### Soporte:

- ✓ La aplicación debe permitir agregar nuevas funcionalidades sin alterar el comportamiento de la misma.
- ✓ Se requiere que el producto reciba mantenimiento ante cualquier fallo que ocurra.

#### Confidencialidad:

- ✓ Toda la información guardada referente a las solicitudes será confidencial, evitando de esta forma su divulgación y que sea consultada por personal no autorizado.

#### Software:

- ✓ Los clientes tendrán acceso a la aplicación a través de los navegadores Internet Explorer 6 y Mozilla Firefox 3.5.
- ✓ El servidor local deberá utilizar, servidor Web Apache 2.2, SGBD PostgreSQL 8.0 y lenguaje de programación PHP 5.1.

#### Hardware:

##### Requerimientos mínimos para el servidor:

- ✓ Procesador de tipo Pentium 4 con velocidad de 2 GHz.
- ✓ Memoria RAM de 1GB.
- ✓ Disco duro de 40 GB.
- ✓ Tarjeta de red.

##### Requerimientos mínimos para la estación de trabajo del usuario:

- ✓ Clientes ligeros.
- ✓ Estaciones de trabajo de tipo Laptop o Destok

## 2.3 MODELO DE CASOS DE USO DEL SISTEMA

El modelo de casos de uso permite que los desarrolladores de software y los clientes lleguen a un acuerdo sobre los requerimientos, es decir, sobre las condiciones y posibilidades que debe cumplir el sistema. El mismo contiene actores, casos de uso y sus relaciones, constituye una entrada para el análisis, el diseño y las pruebas.

### 2.3.1 DEFINICIÓN DE LOS ACTORES DEL SISTEMA

Los actores del sistema intercambian información con dicho sistema, pueden ser un recipiente pasivo de información, pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado (30).

Tabla 2.5 Descripción de los Actores del Sistema

ACTORES DEL SISTEMA	JUSTIFICACIÓN
Solicitante	Es el rol usuario del sistema que podrá hacer solicitudes de servicios telemáticos de aperturas de puertos y direccionamiento IP como también modificar planillas de solicitudes echas anteriormente por él, cancelar servicios que en algún momento solicitó.
Usuario	Es aquel rol autorizado a entrar al sistema y según sus permisos, podrá llevar a cabo acciones dentro del mismo.
Directivo	Es el rol usuario del sistema autorizado a realizar acciones sobre las solicitudes, aprobarlas (firmarlas) o rechazarlas, y en caso de rechazarlas puede elaborar un comentario del porqué del rechazo. También puede realizar búsquedas en el sistema para consultar cualquier tipo de estadísticas referentes a las solicitudes, así como realizar reportes de auditoría en el sistema.
Trabajador de Tecnología	Representa un actor generalizado que puede comportarse como Directivo o Administrador el cual puede mostrar planillas de solicitudes y reportes de Auditoria.
Administrador	Es el rol que configura y controla el comportamiento del sistema. Encargado de adicionar, modificar, eliminar usuarios

	y servicios del sistema así como asignarles un rol a los usuarios. Además puede utilizar el sistema para buscar información, gestionarla, mostrar reportes, etc.
--	------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 2.3.2 PATRONES DE CASO DE USO

Los patrones de caso de uso son una pareja de problema / solución con un nombre, que estandariza principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades (31).

Dado un contexto y un problema a resolver, estos han mostrado ser la solución adoptada en la comunidad del desarrollo de software. Se presentan a modo de herramientas que permiten resolver los problemas que se les plantean a los desarrolladores de una forma ágil y sistemática. Estos patrones se enfocan hacia el diseño y las técnicas utilizadas en modelos de alta calidad, y no en como modelar usos específicos.

En el presente trabajo, dadas las características del mismo, se usaron los patrones de caso de uso siguiente: CRUD completo y Múltiples Actores.

- ✓ CRUD (Crear, Leer, Actualizar, Eliminar)

Este patrón se basa en la fusión de casos de uso simples para formar una unidad conceptual.

Completo:

Este patrón consta de un caso de uso, llamado Información CRUD o Gestionar información modela todas las operaciones que pueden ser realizadas sobre una parte de la información de un tipo específico, tales como creación, lectura, actualización y eliminación. Suele ser utilizado cuando todos los flujos contribuyen al mismo valor del negocio, y estos a su vez son cortos y simples.

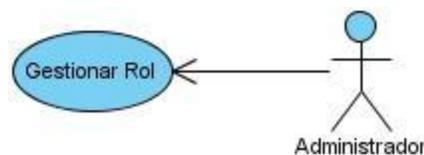


Figura 2.5 CRUD Completo

## ✓ MULTIPLES ACTORES

Roles comunes:

Puede suceder que los dos actores jueguen el mismo rol sobre el CU. Este rol es representado por otro actor, heredado por los actores que comparten este rol. Es aplicable cuando, desde el punto de vista del caso de uso, solo exista una entidad externa interactuando con cada una de las instancias del caso de uso.

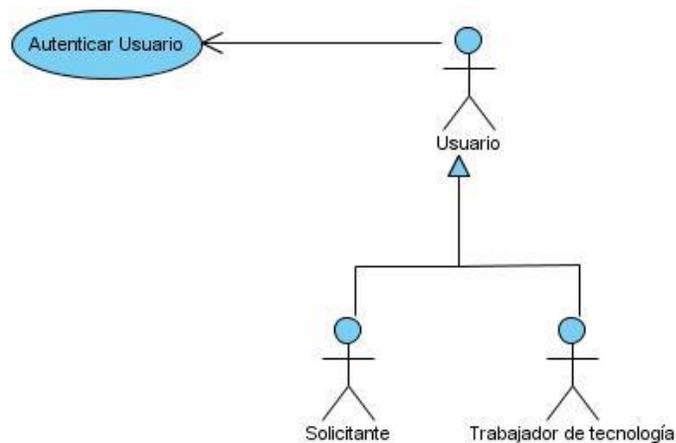


Figura 2.6 Roles Comunes

### 2.3.3 DIAGRAMA DE CASOS DE USO DEL SISTEMA

Un diagrama de casos de uso del sistema representa gráficamente a los procesos y su interacción con los actores (30).

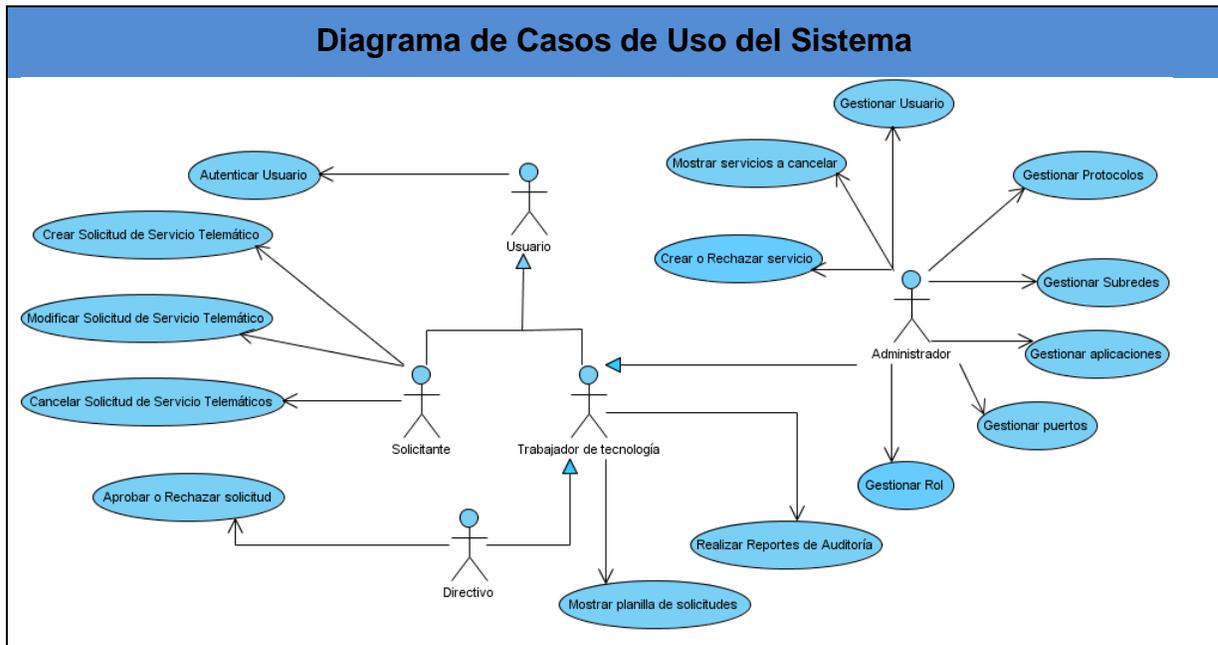


Figura 2.7 Diagrama de Casos de Uso del sistema

### 2.3.4 DESCRIPCIÓN TEXTUAL DE LOS CASOS DE USO DEL SISTEMA

Para entender la funcionalidad asociada a cada caso de uso no es suficiente con la representación gráfica del diagrama de casos de uso, también se lleva a cabo la descripción textual de cada caso de uso donde se especifican todas las acciones necesarias para la realización del mismo. Las descripciones textuales de las tablas se pueden encontrar en el Anexo #1.

### CONCLUSIONES

En este capítulo se han modelado en términos de casos de uso los procesos que se llevan a cabo en el negocio existente, identificando además los actores y trabajadores que intervienen en los mismos y describiendo las actividades que los componen. Se especifican los requerimientos funcionales y no funcionales, siendo modelados gráficamente mediante un Diagrama de Casos de Uso del Sistema. Se identificaron y escribieron los actores que interactuarán con las funcionalidades previstas para cada uno de estos casos de usos, al igual se realizó una descripción textual de cada caso de uso. Con el desarrollo de este capítulo quedaron sentadas las bases que darán paso al diseño del sistema.

## CAPÍTULO 3: DISEÑO DEL SISTEMA

### INTRODUCCIÓN

En el presente capítulo se plantea la propuesta de solución del diseño. Esta fase es muy importante para crear una buena arquitectura del sistema que sirve de plano para el período de implementación. Se pretende dar una descripción del estilo arquitectónico utilizado para el desarrollo de la aplicación, donde también se describirán y definen los diferentes patrones de diseño empleados. Se elaboran los diagramas de clases de diseño y los diagramas de secuencia. Además se expondrá el diseño de la base de datos mediante el modelo físico y lógico de datos. Por último, se incluye el modelo de implementación a través de los diagramas de despliegue y componentes.

### 3.1 ESTILO ARQUITECTÓNICO UTILIZADO

Para el desarrollo del sistema se utilizó el patrón arquitectónico Modelo Vista Controlador (Model-View-Controller - MVC), clásico patrón de diseño utilizado para diseñar aplicaciones con sofisticadas interfaces donde la lógica de interfaz de usuario cambia con más frecuencia que los almacenes de datos y la lógica de negocio, dicho en otras palabras para el diseño de herramientas que necesitan la habilidad de mantener múltiples vistas para los mismo datos, tales como editores gráficos.

Este estilo se basa principalmente en separar en tres capas el diseño de las aplicaciones, el modelo de datos, la presentación de los mismos y las acciones de los usuarios (Figura 3.1), utilizando la capa que gestiona las acciones (controlador) como administradora de los posibles eventos.

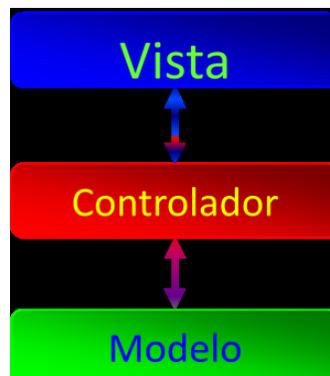


Figura 3.1 Modelo - Vista - Controlador

**Modelo:** Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada.

**Vista:** Muestra la información al usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.

**Controlador:** Reciben las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón, pulsaciones de teclas, etc. Los eventos son traducidos a solicitudes de servicio ("service requests") para el modelo o la vista.

### 3.2 PRINCIPALES PATRONES DE DISEÑO UTILIZADOS

“El patrón es una pareja de problema/solución con un nombre y que es aplicable a otros contextos, con una sugerencia sobre la manera de usarlo en situaciones nuevas.” (32).

En el diseño de la aplicación se utilizaron los patrones GRASP; los cuales son Patrones Generales de Software para asignación de responsabilidades.

Entre los patrones GRASP se encuentran los siguientes:

**Experto:** Es el principio básico de asignación de responsabilidades. Indica que la responsabilidad de la creación de un objeto debe recaer sobre la clase que conoce toda la información necesaria para crearlo; lo cual permite que se conserve el encapsulamiento, soportando un bajo acoplamiento y una alta cohesión.

**Creador:** Ayuda a identificar quien debe ser el responsable de la creación de nuevos objetos o clases. La nueva instancia deberá ser creada por la clase que tiene la información necesaria para realizar la creación del objeto, usa directamente las instancias creadas del objeto, o almacena o maneja varias instancias de la clase. Este patrón brinda soporte de bajo acoplamiento, lo cual supone menos dependencias entre clases y posibilidades.

**Alta Cohesión:** Este patrón propone asignar la responsabilidad de manera que la complejidad se mantenga dentro de límites manejables asumiendo solamente la responsabilidad que deben manejar, evadiendo un trabajo excesivo. Esto se evidencia desde el punto de vista que cada una de las clases

implementadas tiene responsabilidades específicas y no están sobrecargadas con demasiadas de ellas. Su utilización mejora la claridad y facilidad con que se entiende el diseño, simplifica el mantenimiento y las mejoras de funcionalidad, generan un bajo acoplamiento, soporta mayor capacidad de reutilización.

**Bajo Acoplamiento:** Es la medida de la fuerza con que una clase está conectada a otra clase. De tal forma que en caso de producirse una modificación en alguna de ellas, tenga la mínima repercusión posible en el resto de las clases. Mientras más bajo sea el acoplamiento más independiente será una clase de otra, potenciando la reutilización, y disminuyendo la dependencia entre las clases.

**Controlador:** Este patrón sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado. La utilización del CodeIgniter para la realización del sistema contribuye a la aplicación de este patrón ya que define un controlador frontal mediante el cual se accede a toda la aplicación.

Al utilizar CodeIgniter como framework se incluyen los patrones de diseño con los que trabaja, uno de ellos es el patrón Decorador perteneciente al patrón de diseño GOF donde el layout es quien almacena el código HTML común para todas las páginas de la aplicación, el cual integra dentro de sí el contenido de la plantilla o sea, el layout decora la plantilla.

### 3.3 DIAGRAMAS DE CLASES DEL DISEÑO

En las aplicaciones Web, donde es más importante la modelación de la lógica y el estado del negocio que los detalles de presentación, un diagrama de clases presenta las páginas, los enlaces entre estas (donde cada página lógica puede ser representada como una clase), todo el código que irán creando las páginas, así como el contenido dinámico de estas una vez que estén en el navegador del cliente.

El diagrama de clases Web, fue definido, a partir de los diferentes casos de uso del sistema y empleando las extensiones de UML para Web. A continuación se muestra el diagrama de clases del diseño Generalizado para los casos de uso “Crear solicitud de servicio telemático”, “Modificar solicitud de servicio telemático” y “Cancelar solicitud de servicio telemático”. El resto de los diagramas se pueden encontrar en el Anexo #2.

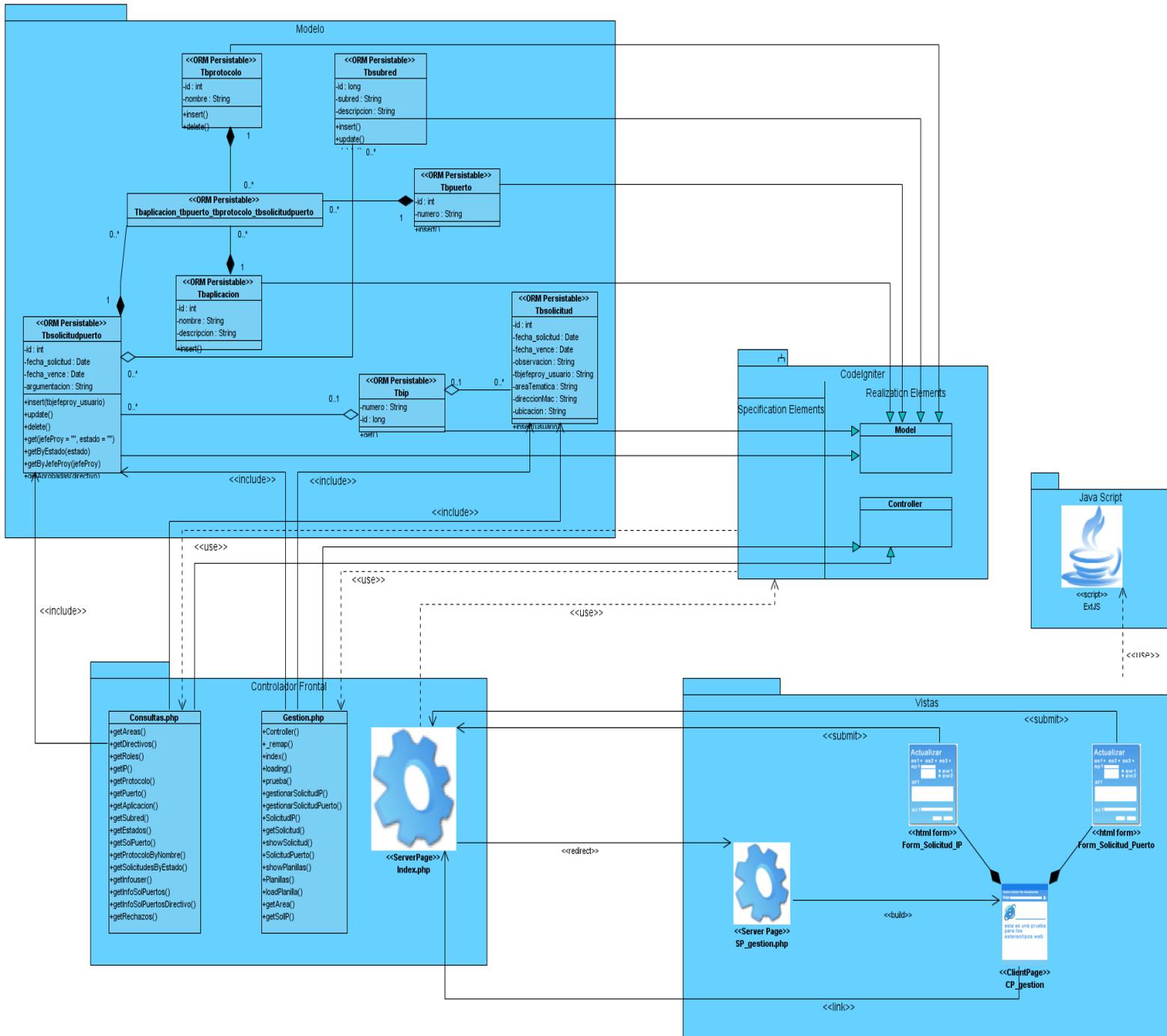


Figura 3.2 Diagrama de clases del Diseño “Gestionar solicitud de servicios telemáticos”.

### 3.4 DIAGRAMAS DE INTERACCIÓN

La modelación de los aspectos dinámicos del sistema, es principio distintivo de los diagramas de interacción. Se trata de diagramas que permiten representar la interacción que tiene lugar entre objetos mediante transferencia de mensajes entre los mismos. Son diagramas de interacción el diagrama de secuencia y el de colaboración.

#### DIAGRAMAS DE SECUENCIA

En el diseño, es preferible realizar los diagramas de secuencia ya que el centro de atención principal es encontrar secuencias de interacciones detalladas y ordenadas en el tiempo. Los Diagramas de secuencia muestran las interacciones entre objetos mediante transferencia de mensajes entre los mismos. A continuación se muestra el diagrama de secuencia para el caso de uso “Crear solicitud de Servicio Telemático” en el escenario “Crear solicitud de Puerto” el resto de los diagramas se pueden encontrar en el Anexo #3.

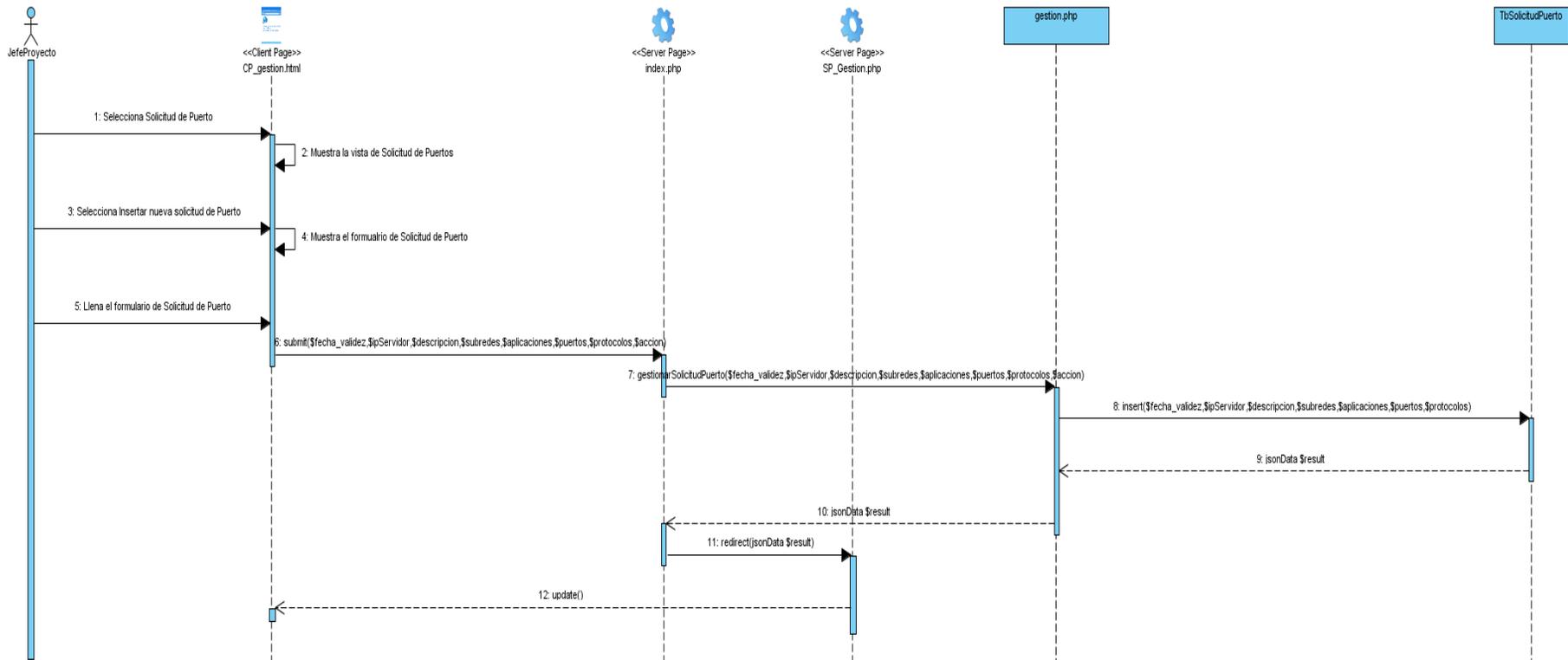


Figura 3.3 Escenario “Crear Solicitud de Puerto”





### 3.6 DIAGRAMA DE DESPLIEGUE

El diagrama de despliegue permite apreciar de forma visual como se encuentran distribuidos físicamente los componentes Hardware y Software de la aplicación, es decir, la configuración de los elementos de procesamientos en tiempo de ejecución y los componentes software (procesos y objetos que se ejecutan en ellos).

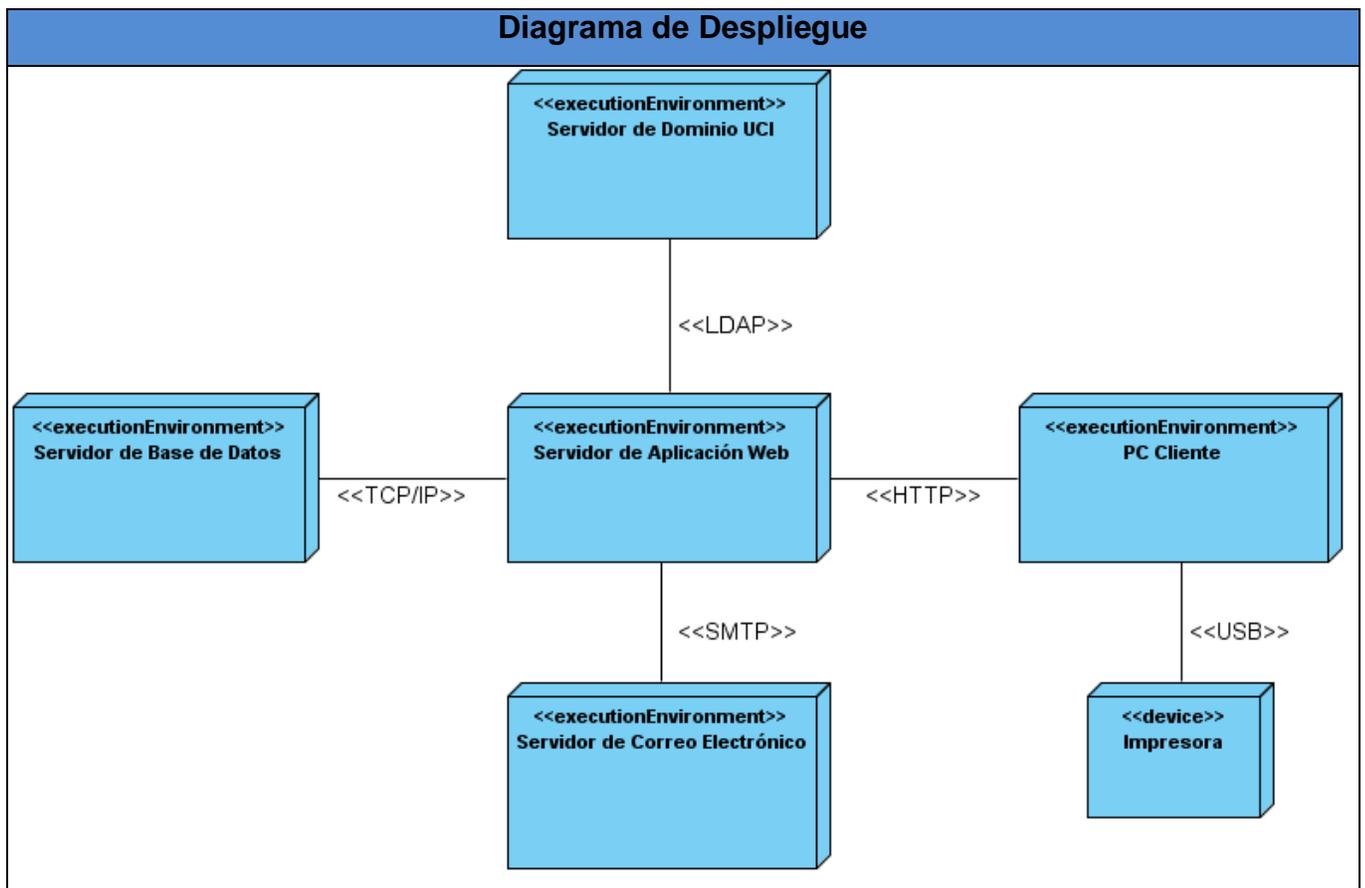


Figura 3.6 Diagrama de Despliegue

### 3.6.1 DESCRIPCIÓN DE LOS NODOS

**Nombre del Nodo Procesador:** Servidor de Dominio UCI

Este nodo procesador es externo al sistema y es usado por el mismo para la autenticación de los usuarios contra el LDAP de la UCI.

**Nombre del Nodo Procesador:** Servidor de Correo Electrónico

Este nodo procesador es externo al sistema y es usado por el mismo para el envío de correo electrónico mediante la aplicación.

**Nombre del Nodo Procesador:** PC Cliente

Este nodo procesador es físico del sistema, y hace referencia a cualquier PC cliente que se encuentre dentro de la UCI mediante la cual se puede acceder vía Web a la aplicación. Los requerimientos de Hardware para esta PC son mínimos ya que el procesamiento de toda la información se realiza en el servidor Web.

**Nombre del Nodo Procesador:** Servidor de Base de Datos

Este nodo procesador es físico del sistema, y en él se despliega la base de datos la cual contiene todos los datos necesarios para el correcto funcionamiento del sistema. Es usado como SGBD a PostgreSQL v 8.3.

**Nombre del Nodo Procesador:** Servidor de Aplicación Web

Este nodo procesador es físico del sistema, y en él se hospeda la aplicación Web. Dicho servidor Web deberá contar con una alta disponibilidad y un rendimiento adecuado que facilite una respuesta rápida frente a las demandas de los clientes.

**Nombre del Nodo Dispositivo:** Impresora

Este nodo Dispositivo hará referencia a cualquier impresora de la UCI compatible con Windows o Linux.

### 3.6.2 DESCRIPCIÓN DE LAS INTERFACES DE COMUNICACIÓN

#### <<HTTP>>

Protocolo de Transferencia de Hipertextos (protocolo estándar básico de la arquitectura Web). Utilizado para el intercambio de páginas Web (HTML). Se ha usado para la comunicación establecida entre el servidor de aplicación Web y las PC clientes de la UCI.

#### <<TCP/IP>>

El TCP/IP es la base de Internet, y sirve para enlazar computadoras que utilizan diferentes sistemas operativos, incluyendo PC, minicomputadoras y computadoras centrales sobre redes de área local (LAN) y área extensa (WAN). Se ha usado para establecer la comunicación entre el servidor de aplicación Web y el servidor de Base de Datos.

#### <<USB>>

Bus Serie Universal: En ordenadores, un bus es un subsistema que transfiere datos o electricidad entre componentes dentro de un ordenador o entre ordenadores. Dicho de otra forma, es una interfaz de hardware que permite conectar periféricos de baja velocidad, como el teclado, el ratón o mouse, la impresora o cámaras digitales, a los ordenadores o computadoras. Se ha usado para establecer la comunicación entre la PC cliente y cualquier Impresora conectada a ella.

## <<SMTP>>

Protocolo Simple de Transferencia de Correo, es un protocolo de red de la capa de aplicación basado en texto utilizado para el intercambio de mensajes de correo electrónico entre computadoras u otros dispositivos.

## <<LDAP>>

Protocolo Ligero de Acceso a Directorios, es un protocolo de red a nivel de aplicación que permite el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red.

### 3.7 PRINCIPIOS DE DISEÑO UTILIZADOS

El diseño de una aplicación sea cual sea el objetivo del mismo tiene que basarse en el usuario por lo tanto requiere de una atención especial ya que éste será el que modele la interacción entre la aplicación y el usuario, y por tanto posibilitará la consecución de los objetivos perseguidos por el mismo, por lo que se debe trabajar en pro de su facilidad de uso y a la vez de una apariencia agradable, siendo conveniente además que sea técnica y visualmente coherente, bien estructurado y de apariencia profesional. Los usuarios que utilizarán el sistema deberán diferenciarse en cuanto al uso de las nuevas tecnologías por lo que se deberá desarrollar la misma en base al nivel medio que en algunos prevalece. Por todo lo anteriormente expuesto se deberán considerar en el diseño de la aplicación las siguientes características:

- ✓ Diseñar para 1024 X 768 pixeles.
- ✓ Los elementos que se presenten en cada una de las pantallas no deberán ser numerosos.
- ✓ El sistema solo deberá mostrar al usuario aquellas opciones a las que dado su rol tenga permiso.
- ✓ Debe solo verse lo que el usuario puede usar en ese momento y no otras opciones.

### 3.8 ESTÁNDARES EN LA INTERFAZ DE LA APLICACIÓN

La calidad de la interfaz de usuario puede ser uno de los motivos que conduzca a un sistema al éxito o al fracaso, es por eso que uno de los aspectos más relevantes de la usabilidad de un sistema es la consistencia de su interfaz de usuario. La misma deberá presentar una interfaz sencilla y de fácil uso,

estableciéndose un mismo patrón para el diseño de todas las páginas. Teniéndose especial interés en no sobrecargar las mismas con banners y controles innecesarios. Para el diseño de la interfaz de la aplicación se utilizaron colores azul claro, gris y blanco, estándares y acordes a las normas de diseño de la Universidad. Además una barra de menú con diferentes opciones se deberá encontrar en la parte superior de la aplicación de forma horizontal, donde pudieran incluirse otros servicios a los que el usuario tendría acceso. Los colores, fundamentalmente blanco y azul en varias tonalidades y el color rojo propuesto para los mensajes de error emitidos, con el objetivo de que sean observados sin dificultad. Poniendo en práctica los aspectos anteriormente descritos se garantizara una rápida y fácil adaptación por parte de los usuarios para con el despliegue del sistema. En general se realizan múltiples operaciones en cada página, de forma que el usuario no tenga que moverse tanto dentro de la aplicación, para completar una operación. Por ejemplo se pueden hacer la modificación, eliminación en una misma página.

### 3.9 TRATAMIENTO DE ERRORES

Identificar y controlar los posibles errores que se pueden presentar a la hora de interactuar con el software es de vital importancia para así garantizar un correcto funcionamiento del sistema. El sistema se deberá proponer prevenir al máximo los posibles errores, sobre todo los que ocurren por parte del usuario, para ello solo se le brindaran las opciones a las que tiene acceso. Se deberá insistir en que el usuario realice la menor cantidad posible de entrada de datos aprovechando al máximo los componentes visuales de selección. Al introducir información en un formulario se deberán verificar los campos obligatorios, la veracidad y tipo de los datos. En caso de error se deberá emitir un mensaje de alerta en color rojo para que resalte fácilmente a la vista, especificando claramente en qué consiste para su fácil comprensión por el usuario. Se deberán tratar estos errores de forma tal que las interacciones con la base de datos se realicen de forma correcta. Mediante la validación en el lado del cliente, se garantizara que los datos suministrados por los usuarios, se almacenen íntegros y no existan inconsistencias. Se propone utilizar el lenguaje Java Script para la implementación de las funciones encargadas del control y validación de datos. Se mostraran mensajes de confirmación luego de llevadas a cabo las distintas acciones dentro del sistema.

## CONCLUSIONES

En el desarrollo de este capítulo, se muestran los resultados obtenidos en el flujo de trabajo de Diseño, realizando el modelado de la solución propuesta dando una breve explicación de los artefactos que se construyeron durante el desarrollo en la búsqueda de una solución final óptima y que satisfaga las necesidades del usuario. Para ello se desarrollaron los diagramas de clases del diseño en conjunto con sus diagramas de interacción según corresponden, además de los diagramas de clases persistentes y de entidad relación que necesita la base de datos para la futura implementación del sistema. También se cumplió con la descripción de la distribución física del sistema, en términos de como se distribuye la funcionalidad entre los nodos de cómputo, a través del diagrama de despliegue.

## CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA

### INTRODUCCION

En este capítulo se describe como los elementos del modelo del diseño se implementan en términos de componentes. Se desarrollaran varias pruebas de aceptación usando el método de caja negra para garantizar que se han cumplido los requisitos funcionales.

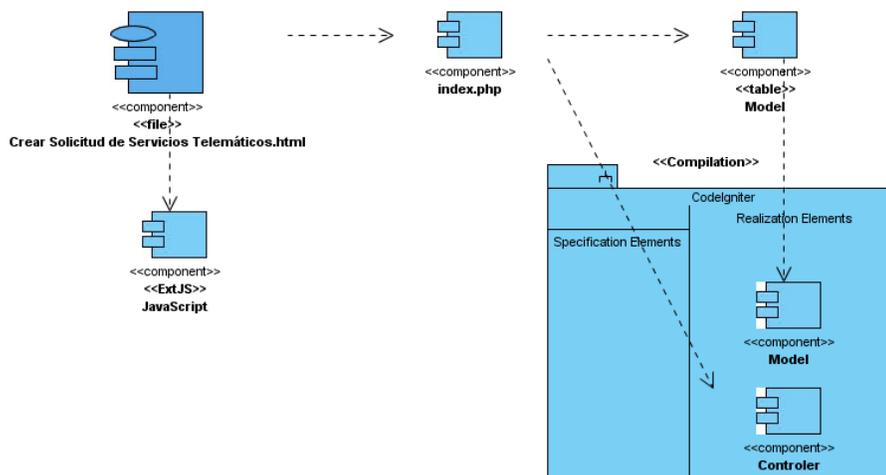
#### 4.1 DIAGRAMA DE COMPONENTES.

En el flujo de trabajo de diseño se propone crear un plano del modelo de implementación. El flujo de trabajo de implementación describe como los elementos del modelo del diseño se implementan en términos de componentes y como estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue.

Los diagramas de componentes son usados para estructurar el modelo de implementación en términos de subsistemas de implementación y mostrar las relaciones entre los elementos de implementación.

Se representa como un grafo de componentes software unidos por medio de relaciones de dependencia (compilación, ejecución), pudiendo mostrarse las interfaces que estos soporten.

A continuación se muestra el diagrama de componente para el caso de uso “Crear solicitud de Servicio Telemático”. El resto de los diagramas se pueden encontrar en el Anexo #4.



## 4.2 MODELO DE PRUEBAS

Como todo proceso de desarrollo de cualquier producto de la industria, es preciso luego de haberlo construido, realizarle algunas pruebas a cada uno de sus elementos obteniendo de esta forma una medida de cumplimiento del proceso de su elaboración. Ya que no se puede asegurar que un producto es ciento por ciento fiable ni que cumplirán al máximo con las expectativas de los clientes. El método con el que se cuenta para garantizar la calidad y el buen funcionamiento de un producto es la realización de pruebas. Las pruebas no confirman la ausencia de errores en el software, sólo brindan una medida de como responderá el mismo ante algunas situaciones determinadas.

A la herramienta desarrollada se le aplicaron pruebas de aceptación con el objetivo de verificar las funcionalidades del sistema. Para esto se utilizó el método de caja negra.

Prueba de caja negra: Se refiere a las pruebas que se llevan a cabo sobre la interfaz del software, por lo que los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, así como que la integridad de la información externa se mantiene. Esta prueba examina algunos aspectos del modelo fundamentalmente del sistema sin tener mucho en cuenta la estructura interna del software.

Se definieron casos de prueba para todos los casos de uso, a continuación se dan a conocer las pruebas más significativas que se realizaron a los casos de uso que se consideran más importantes.

### 4.2.1 CASO DE USO: Crear Solicitud de Servicio Telemático

Tabla 4.7 Caso de prueba “Crear solicitud de puerto”

<b>Caso de Uso</b>	Crear Solicitud de Servicio Telemático
<b>Caso de prueba</b>	Crear solicitud de puerto
<b>Nombre de la Persona que realiza la prueba:</b> Yoandy	
<b>Descripción de la prueba:</b> Se comprueba si se logra crear una solicitud de puerto	

<p><b>Condiciones de ejecución:</b></p> <ol style="list-style-type: none"> <li>1. Ser usuario con el rol de Jefe de Proyecto.</li> <li>2. Estar autenticado en el sistema</li> <li>3. El usuario debe tener asignada una dirección IP</li> </ol>
<p><b>Entrada / Pasos de ejecución:</b></p> <ol style="list-style-type: none"> <li>1. Seleccionar la opción “Nueva Solicitud”</li> <li>2. llenar el formulario correspondiente a la nueva solicitud de apertura de puerto</li> <li>3. Seleccionar la opción guardar</li> </ol>
<p><b>Resultado Esperado:</b> Se crea en el sistema una nueva solicitud de Apertura de Puerto con el estado de espera</p>
<p><b>Evaluación de la prueba:</b> Satisfactoria</p>

## CASO DE USO: Modificar Solicitud de Servicio Telemático

Tabla 4.8 Caso de prueba “Modificar solicitud de puerto”

<b>Caso de Uso</b>	Modificar Solicitud de Servicio Telemático
<b>Caso de prueba</b>	Modificar solicitud de puerto
<b>Nombre de la Persona que realiza la prueba:</b> Yordan	
<b>Descripción de la prueba:</b> Se comprueba si se logra actualizar los datos de una solicitud de Puerto la cual fue rechaza o está en estado habilitada	
<p><b>Condiciones de ejecución:</b></p> <ol style="list-style-type: none"> <li>1. Ser usuario con el rol de Jefe de Proyecto.</li> <li>2. Estar autenticado en el sistema</li> <li>3. Debe existir en el sistema una solicitud de puerto con el estado de “Rechazada” o con el estado “Habilitada”</li> </ol>	
<p><b>Entrada / Pasos de ejecución:</b></p> <ol style="list-style-type: none"> <li>1. Seleccionar una solicitud de puerto</li> <li>2. Seleccionar la opción “modificar” en la sección rechazadas puerto o seleccionar la opción</li> </ol>	

<p>“renovar” en la sección habilitada puerto.</p> <p>3. Modificar los datos de la solicitud</p> <p>4. Seleccionar la opción “Guardar”</p>
<p><b>Resultado Esperado:</b> Se actualiza los datos de la solicitud de puerto seleccionada y se le cambia el estado.</p>
<p><b>Evaluación de la prueba: Satisfactoria</b></p>

### CASO DE USO: Cancelar Solicitud de Servicio Telemático

Tabla 4.9 Caso de prueba “Cancelar solicitud de puerto”

<b>Caso de Uso</b>	Cancelar Solicitud de Servicio Telemático
<b>Caso de prueba</b>	Cancelar solicitud de puerto
<b>Nombre de la Persona que realiza la prueba:</b> Yoandy	
<b>Descripción de la prueba:</b> Se comprueba si se logra cancelar una solicitud que fue rechazada o finalizar el servicio que está habilitado	
<p><b>Condiciones de ejecución:</b></p> <ol style="list-style-type: none"> <li>1. Ser usuario con el rol de Jefe de Proyecto.</li> <li>2. Estar autenticado en el sistema</li> <li>3. Debe existir en el sistema una solicitud de puerto con el estado de “Rechazada” o con el estado “Habilitada”</li> </ol>	
<p><b>Entrada / Pasos de ejecución:</b></p> <ol style="list-style-type: none"> <li>1. Seleccionar una solicitud de puerto</li> <li>2. Seleccionar la opción “Cancelar” en la sección rechazadas puerto o seleccionar la opción “Finalizar servicio” en la secciona habilitadas puerto</li> </ol>	
<p><b>Resultado Esperado:</b> Se cancela la solicitud y se le cambia el estado, si estaba con estado “rechazada” pasa al estado “cancelado” y si estaba con el estado habilitada pasa al estado “pendientes a cancelar”</p>	
<p><b>Evaluación de la prueba: Satisfactoria</b></p>	

## CONCLUSIONES

Mediante la realización del presente capítulo se describió como fue implementada la herramienta en términos de componentes. Se desarrollaron varias pruebas de aceptación usando el método de caja negra para garantizar que se han cumplido los requisitos funcionales.

## CONCLUSIONES GENERALES

- Se realizó un estudio de los servicios telemáticos ofrecidos en el grupo de redes de la Universidad y se definió como vía para contribuir a su mejora, el desarrollo de una aplicación informática para la gestión de solicitudes.
- Se realizó un análisis del proceso actual de gestión de solicitudes llevado a cabo en el grupo de redes logrando identificarse los requisitos de la aplicación.
- Se desarrolló la aplicación para la gestión de solicitudes de servicios telemáticos ofrecidos por el grupo de redes de la Universidad.
- Se realizaron pruebas funcionales al sistema para certificar que la aplicación obtenida cumple con las necesidades expresadas por el cliente.

## RECOMENDACIONES

- Potenciar el módulo de seguridad del sistema añadiéndole un módulo visual para mostrar las trazas.
- Automatizar el proceso de apertura de puerto a partir de la aplicación desarrollada, para una vez creada la solicitud, esta habilite los puertos correspondientes mediante log, logrando que la intervención del personal sea la mínima en el proceso que se lleva a cabo.
- Incluir el sistema Patsi o Reporteador dinámico para mejorar los reportes que puede dar salida el sistema, esto haría que el sistema tuviera nuevos reportes que pudieran ser configurables por el administrador.
- Continuar la investigación con el objetivo de identificar nuevas funcionalidades que contribuyan con el perfeccionamiento del sistema.

## REFERENCIA BIBLIOGRÁFICA

1. DIP Solutions. [En línea] [Citado el: 06 de Noviembre de 2009.] Disponible en: <http://www.dipsolutions.com.ar/glosario.htm>.
2. Definicion de Rigidez. [En línea] [Citado el: 15 de Noviembre de 2009.] Disponible en: <http://definicion.de/correo-electronico/>.
3. Alegsa.com. [En línea] [Citado el: 15 de Noviembre de 2009.] Disponible en: <http://www.alegsa.com.ar/Dic/escritorio%20remoto.php>.
4. windows 2003. Controlador de dominio. [En línea] [Citado el: 18 de Noviembre de 2009.] Disponible en: <http://vnanock.wordpress.com/2007/07/29/windows-2003-controlador-de-dominio/>.
5. Qué es la mensajería instantánea. [En línea] [Citado el: 03 de Noviembre de 2009.] Disponible en: [es.tech-faq.com/instant-messaging.shtml](http://es.tech-faq.com/instant-messaging.shtml).
6. Manage Engine ServiceDesk Plus. [En línea] [Citado el: 2 de Enero de 2010.] Disponible en: <http://www.manageengine.com/products/service-desk/spanish/index.html>.
7. *HelpDesk - Gestión de Soporte Web*. [En línea] [Citado el: 26 de Diciembre de 2009.] Disponible en: <http://www.artologik.com/es/HelpDesk.aspx>.
8. NUMARA Software. [En línea] [Citado el: 19 de Diciembre de 2009.] Disponible en: <http://www.numarasoftware.com/default.aspx>.
9. Intralitia. [En línea] [Citado el: 6 de Enero de 2010.] Disponible en: <http://www.intralitia.com/index.htm>.
10. *Sitio Web de la Dirección de Informatización*. [En línea] [Citado el: 8 de Noviembre de 2009.] Disponible en: <http://informatizacion.uci.cu/>.
11. *Directorio de Servicios Web UCI*. [En línea] Mayo de 2007. [Citado el: 5 de Diciembre de 2009.] Disponible en: <http://uddi.uci.cu/files/arquitectura.2007.5.9.pdf>.
12. **Brito Acuña, Kareenny**. <http://www.eumed.net>. [En línea] [Citado el: 28 de Noviembre de 2009.] Disponible en: [http://www.eumed.net/libros/2009c/584/Fundamentacion de las metodologías de desarrollo de software seleccionadas.htm](http://www.eumed.net/libros/2009c/584/Fundamentacion%20de%20las%20metodologias%20de%20desarrollo%20de%20software%20seleccionadas.htm).
13. **G. Figueroa, Roberth, J. Solís, Camilo y A. Cabrera, Armando**. *METODOLOGÍAS TRADICIONALES VS. METODOLOGÍAS ÁGILES*. s.l.: Universidad Técnica Particular de Loja, Escuela de Ciencias en Computación, 2008.
14. **Letelier, Patricio y Penadés, Maria del Carmen**. *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP 26, 2005, Vol. V. ISSN 1666-1680*.

15. **Letelier Torres, Patricio y Sánchez López, Emilio A.** Metodologías Ágiles en el Desarrollo de Software. Alicante : ISSI, 2003.
16. **Martínez Sousa, Dayren y Zamora Rodríguez, Yunieski.** *SEGIPP. Sistema para la Selección de Estudiantes y Gestión de Información referente a Polos y Proyectos Productivos en la Facultad 5. Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas.* [En línea] Julio de 2008. Disponible en: [http://bibliodoc.uci.cu/TD/TD\\_1550\\_08.pdf](http://bibliodoc.uci.cu/TD/TD_1550_08.pdf).
17. *Introducción a la Ingeniería de Software. Conferencia 1 de Ingeniería del Software I.* Curso 2007-2008, UCI.
18. **James Rumbaugh, Ivar Jacobson y Grady Booch.** *The Unified Modeling Language.* s.l. : Addison Wesley Longman, Inc., 1999. 0-201-30998-X.
19. Portal para desarrolladores .NET Framework. Desarrollo Orientado a Objetos con UML. *Sitio Web Clikear.com.* [En línea] [Citado el: 26 de 10 de 2009.] Disponible en: <http://www.clikear.com/manuales/uml/introduccion.asp>.
20. UNIVERSIDAD TECNICA PARTICULAR DE LOJA. [En línea] [Citado el: 15 de Diciembre de 2009.] Disponible en: <http://blogs.utpl.edu.ec/disenowebymultimedia/2009/07/03/la-arquitectura-cliente-servidor/>.
21. <http://www.csi.map.es/>. [En línea] [Citado el: 25 de noviembre de 2009.] Disponible en: <http://www.csi.map.es/csi/silice/Global71.html>.
22. **Alvarez, Rubén.** *www.desarrolloweb.com.* [En línea] [Citado el: 25 de Noviembre de 2008.] Disponible en: <http://www.desarrolloweb.com/articulos/239.php>.
23. **Musciano, Chuck y Kenedy, Bill.** *HTML la Guía Completa.* s.l. : McGRAW-HILL, 1999. ISBN 970-10-2141-X.
24. **Eguíluz Pérez, Javier.** *Introducción a Java Script.* 2006.
25. **Delgado, Olivia.** *Sistemas Gestores de Bases de Datos.* Madrid : s.n., 2005-2006.
26. *postgresql-es.org.* [En línea] [Citado el: 25 de Noviembre de 2009.] Disponible en: [http://www.postgresql-es.org/sobre\\_postgresql](http://www.postgresql-es.org/sobre_postgresql).
27. Apache, el servidor más usado en la Red. *Periodico elmundo.es.* [En línea] [Citado el: 25 de Noviembre de 2009.] Disponible en: <http://www.elmundo.es/navegante/99/mayo/15/apache.html>.
28. Codeigniter corp. . *Codeigniter. Manual disponible en Inglés, traducido al español.* [En línea] [Citado el: 25 de Noviembre de 2009.] Disponible en: <http://www.codeigniter.com>.

29. Conferencia 5. Fase de Inicio. Modelo del Negocio. . *Eva.uci.cu*. [En línea] [Citado el: 25 de Noviembre de 2009.] Disponible en: <http://eva.uci.cu/course/view.php?id=102>.
30. Conferencia 6. Fase de Inicio. Disciplina de Requisitos. *Eva.uci.cu*. [En línea] [Citado el: 25 de Noviembre de 2009.] Disponible en: <http://eva.uci.cu/course/view.php?id=102>.
31. Teleformación. *Conferencia #4 Flujo de Trabajo Requisitos*. [En línea] [Citado el: 01 de Diciembre de 2009.] Disponible en: <http://teleformacion.uci.cu>.
32. **Roig Morejón, Yaniris y Borges Ramírez, Ygraine**. *Análisis y Diseño del Módulo Gestión de Tesis. Trabajo de diploma para optar por el título de: Ingeniero en Ciencias Informáticas*. [En línea] Junio 2007.
33. *maestrosdelweb*. [En línea] 2 de 11 de 2007. [Citado el: 25 de Noviembre de 2008.] Disponible en: <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.
34. **Alvarez, Miguel Angel**. *Desarrolloweb.com*. [En línea] 09 de 5 de 01. [Citado el: 25 de Noviembre de 2008.] Disponible en: <http://www.desarrolloweb.com/articulos/392.php>.