

Universidad de las Ciencias Informáticas “Facultad 2”



Título: Plataforma de Gestión de Servicios Telemáticos en
GNU/Linux. Módulo de Directorio.
Configuraciones avanzadas.

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

Autor: Yarisel Ortega Díaz

Tutor: Ing. Ramón Alexander Anglada Martínez

Co-tutor: Ing. Jenny De la Rosa Pasteur

Ciudad de la Habana, Junio del 2011

”Año 53 de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaro que soy la única autora de este trabajo y autorizo al Centro de Telemática de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Autora:

Yarisel Ortega Díaz

Co tutora:

Ing. Jenny de la Rosa Pasteur

Datos de Contacto

Tutor: Ing. Ramón Alexander Anglada Martínez

Profesor de las asignaturas de Programación II y IV. Profesor Instructor. Trabajó durante dos años en el proyecto Servicios Telemáticos. Se desempeñó como Líder de proyecto y Jefe de Dpto. en el centro TLM. Actual Jefe de Despliegue del Proyecto 171.

Co tutor: Jenny De La Rosa Pasteur

Trabaja en el proyecto Servicios Telemáticos. Participó en la Jornada Científica Juvenil ICIMAF 2011y en Semana Tecnológica 2010.

Agradecimientos

A mi familia por apoyarme en mis decisiones.

A Anyu por ser parte de mi formación.

A Nilcy y Pachele por apoyarme y quererme como otra hija.

Al grupo de Software Libre y todos los que ayudaron en la realización de este trabajo.

Dedicatoria

Dedico este trabajo a mis padres que siempre me han apoyado.

A mi hermana y mi sobrina que me cuidan y están pendientes de mí.

A mis amigas que han estado presentes en los momentos que más las necesite.

A mi novio por estar a mi lado, apoyarme en todo y ayudarme a pasar por este momento.

Resumen

En la Universidad de las Ciencias Informáticas se desarrolla una Plataforma de Gestión de Servicios Telemáticos en GNU/Linux. Entre los servicios telemáticos que se gestionan en esta plataforma se encuentran el servicio de nombre de dominio (DNS), configuración dinámica de host (DHCP) y el servicio de directorio; este último se gestiona a través de uno de los módulos de la plataforma. Ya en años anteriores se llegó a la versión dos con un conjunto de funcionalidades implementadas entre las que se encuentran la gestión de entradas y esquemas, también brinda la posibilidad de migrar los datos de un Directorio Activo a un servidor OpenLDAP e importar o exportar ficheros en formato LDIF (Formato de Intercambio de Datos del protocolo de directorio, LDAP), pero carece de la búsqueda avanzada que permite filtrar por atributo el resultado de la búsqueda, la gestión de plantillas para hacer más amigable la creación de entradas al OpenLDAP y la integración del servidor de directorio con un servidor DNS.

En el presente documento se describen dichas funcionalidades utilizando RUP como metodología de desarrollo y se define una arquitectura en capa como base del sistema implementándose cuatro capas. Se realiza una descripción de las herramientas y tecnologías utilizadas para el desarrollo de las funcionalidades además se aplican las pruebas definidas determinándose el correcto funcionamiento del sistema.

PALABRAS CLAVE

OpenLDAP, servidor de directorio, LDAP, plantilla, DNS

INDICE DE CONTENIDOS	
INDICE DE FIGURAS	VII
INDICE DE TABLAS	VIII
INTRODUCCIÓN	9
CAPÍTULO 1: “FUNDAMENTACIÓN TEÓRICA”	14
1.1 Introducción	14
1.2 Servicio de directorio.....	14
1.3 LDAP	14
1.3.1 Características del protocolo LDAP	14
1.3.1 Estructura LDAP.....	16
1.4 Servidor de directorio	17
1.4.1 Fedora Directory Server	17
1.4.2 Active Directory	17
1.4.3 OpenLDAP	17
1.5 Sistemas para la gestión de servidores de directorio.....	18
1.6 Lenguajes, Metodología y Herramientas	20
1.6.1 Lenguaje de Modelado. Lenguaje Unificado de Modelado	20
1.6.2 IDEF0.....	21
1.7 Metodología de desarrollo.....	22
1.8 Lenguaje de programación y herramientas	25
1.8.1 Python.....	25
1.9 Herramienta CASE.....	26
1.9.1 Visual Paradigm	26
1.10 Entorno de desarrollo integrado	26
1.10.1 Eclipse.....	26
1.11 Interfaces Gráficas de Usuarios	27
1.11.1 Qt Designer	27
1.12 Conclusiones.....	27

CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	28
2.1 Introducción	28
2.2 Problema a resolver	28
2.3 Información que se maneja	29
2.4 Solución Propuesta	29
2.5 Modelo de Procesos	30
2.5.1 Diagrama A-0	30
2.5.2 Diagrama A0	32
2.5.3 Descripción de los procesos del negocio.....	32
2.6 Especificación de los requisitos de software.	36
2.6.1 Requerimientos Funcionales	36
2.6.2 Requerimientos no funcionales	37
2.7 Definición de los casos de uso	39
2.7.1 Actores del Sistema.....	39
 CAPÍTULO 3: DISEÑO DEL SISTEMA.....	 52
3.1 Introducción	52
3.2 Modelo de diseño.....	52
3.2.1 Diagramas de clases del diseño.....	54
3.3 Patrones de diseño	56
3.4 Conclusiones	58
 CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS.....	 59
4.1 Introducción	59
4.2 Diagramas de componentes.....	59
4.3 Diagrama de despliegue	60
4.4 Prueba	61
4.4.1 Niveles de Prueba	61
4.4.2 Tipos de Prueba	61

4.6 Conclusiones	65
CONCLUSIONES	66
RECOMENDACIONES	67
REFERENCIAS BIBLIOGRÁFICAS	68
BIBLIOGRAFÍA	69
GLOSARIO DE TÉRMINOS	71
ANEXOS.....	¡ERROR! MARCADOR NO DEFINIDO.

INDICE DE FIGURAS

Figura 1: Estructura de un directorio LDAP.....	16
Figura 2: Elementos gráficos para el modelado con IDEF0.....	21
Figura 3: Fases y flujo de la metodología RUP.	24
Figura 4: Representación del proceso general “Administrar servidor de directorio”.....	30
Figura 5: Detalles del proceso general.....	32
Figura 6: Diagrama de paquetes.....	40
Figura 7: Diagrama de Caso de Uso del Sistema-Paquete Autenticación.	41
Figura 8: Diagrama de Casos de Uso del Sistema-Paquete Entradas.	41
Figura 9: Diagrama de Casos de Uso del Sistema-Paquete Servidor.	42
Figura 10: Formulario definir plantilla.	45
Figura 11: Formulario modificar plantilla.	46
Figura 12: Formulario eliminar plantilla.	47
Figura 13: Formulario búsqueda avanzada.....	49
Figura 14: Árbol del servidor de directorio.....	51
Figura 15: Estructura general del sistema.....	53
Figura 16: Diagrama de Clases del Diseño: CU Gestionar plantilla.....	55
Figura 17: Diagrama de Clases del Diseño: CU Realizar búsqueda avanzada.....	55
Figura 18: Diagrama de Clases del Diseño: CU Buscar en el árbol por niveles.....	56
Figura 19: Implementación del patrón Singleton.....	56
Figura 20: Uso del patrón Singleton.....	57
Figura 21: Envío de eventos en el uso del patrón Observer.....	58
Figura 22: Suscripción de eventos.....	58
Figura 23: Uso del patrón IoC.....	58
Figura 24: Diagrama de componente: CU Realizar búsqueda avanzadas.....	60
Figura 25: CU Gestionar Plantillas.....	60
Figura 26: Diagrama de despliegue.....	61

INDICE DE TABLAS

Tabla 1: Descripción del proceso Administrar servidor de directorio.	30
Tabla 2: Descripción del proceso Configurar servidor de directorio para integrarlo con servidor DNS.	32
Tabla 3: Descripción del proceso Definir plantilla.	33
Tabla 4: Descripción del proceso Realizar búsqueda avanzada.	34
Tabla 5: Mejorar lectura de las entradas.	35
Tabla 6: Actores del sistema.	39
Tabla 7: Descripción CU Configurar servidor de directorio con DNS.	43
Tabla 8: Descripción CU Gestionar plantilla.	43
Tabla 9: Descripción CU Realizar búsqueda avanzada.	47
Tabla 10: Descripción CU Buscar en el árbol por niveles.	49
Tabla 11: Caso de prueba Búsqueda avanzada.	63
Tabla 12: Descripción de las variables.	64
Tabla 13: No Conformidades del proceso de liberación interno.	65

Introducción

En las tres últimas décadas, la humanidad ha sido testigo de un vertiginoso avance en las tecnologías, la informática y las telecomunicaciones. Con ello también han aumentado los recursos en la red dígase, impresoras, ordenadores y usuarios que acceden a estos recursos por lo que ha surgido la necesidad de gestionar y controlar estos recursos. Al mismo tiempo fueron surgiendo softwares especializados en la organización y almacenamiento de información, hasta existir numerosos tipos de base de datos ya sean estáticas, dinámicas y bibliográficas. Pero ninguna permitía gestionar los recursos en la red.

Una de las soluciones que se encontró para la gestión de recursos fue precisamente los servidores de directorio los cuales representan contenedores de información descriptiva basada en atributos y apoyado en sofisticadas capacidades de filtrado. Los directorios por lo general no admiten transacciones complicadas como las que se encuentran en los sistemas de bases de datos diseñados para manejar grandes y complejos volúmenes de actualizaciones. Las actualizaciones de los directorios son normalmente cambios simples, o todo o nada, siempre y cuando estén permitidos (1), es decir está específicamente diseñada para la búsqueda y navegación dentro de los directorios.

Existen diferentes estándares que especifican servicios de directorio, uno de los más conocidos es X.500; el cual fue desarrollado conjuntamente por ISO/IEC¹ e ITU-T², para la creación y mantenimiento de servicios de directorio de forma distribuida. Este estándar define de forma nativa un protocolo de acceso denominado Protocolo de Acceso a Directorio, DAP por sus siglas en inglés *Directory Access Protocol*, que resulta muy complejo y está definido sobre la capa completa de niveles OSI³ por lo cual se dice que es computacionalmente pesado. Como alternativa a DAP para acceder a directorios de tipo X.500 surge el Protocolo Ligero de Acceso a Directorios, LDAP por sus siglas en inglés (*Lightweight Directory Access Protocol*) que ofrece un protocolo ligero casi equivalente, pero mucho más sencillo y eficiente, diseñado para operar directamente sobre TCP/IP. Actualmente, la mayoría de los servidores de directorio X.500 incorporan LDAP como uno de sus protocolos de acceso. (13)

¹ Organismo internacional de normalización

² Antes llamado CCITT

³ Open System Interconnection

El protocolo LDAP fue creado originalmente por Tim Howes de la University of Michigan Steve Kille de ISODE, y Wengyik Yeong de Performance Systems International en 1993. En las primeras etapas de ingeniería de LDAP, éste era conocido como Lightweight Directory Browsing Protocol, o LDBP. Posteriormente fue renombrado dado que el ámbito del protocolo había sido expandido para incluir no sólo navegación en el directorio y funciones de búsqueda, sino también funciones de actualización de directorio. (3)

Implementaciones de servidores de directorio en el mundo existen varias, algunos se encuentran bajo licencias de Software Privativo como es el caso del Active Directory que se encuentra acoplado a las versiones del sistema Windows Server 2000, 2003, 2008 y superiores y es precisamente uno de los más usados. También existen otros softwares como Fedora Directory Server que se encuentra en el sistema operativo Fedora y OpenLDAP que es un software desacoplado; es decir que se puede usar en varios sistemas operativos. Estos softwares se encuentran bajo licencia de Software Libre lo que le brinda libertades a sus usuarios, entre las que se encuentra la libertad de usar el programa, con cualquier propósito; la libertad de estudiar cómo funciona el programa, y adaptarlo a sus necesidades; le permite distribuir copias y la libertad de mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie.

Para la administración de estos servidores existen herramientas que ayudan a configurarlo y administrarlo pero en el caso del servidor OpenLDAP no cuenta con una herramienta propia para su efectiva administración, siendo administrado por consola, a través de líneas de comandos lo cual dificulta en extremo la administración del servicio. Existen variadas herramientas individuales que interactúan directamente con el servidor, como es el caso de Gosa y phpLDAPadmin que realizan su configuración y administración, pero no cuentan con funcionalidades como la creación de nuevas plantillas y Gosa no muestra el árbol de directorio lo cual es necesario para facilitar el trabajo del administrador de red entre otras funcionalidades importantes para la administración del servicio de directorio.

Por esta razón en el Proyecto Servicios Telemáticos de la Facultad 2 de la Universidad de Ciencias Informáticas se está desarrollando una plataforma para la administración del servicio de nombre de dominio (DNS), el servicio de configuración dinámica de host (DHCP) y servicio de directorio, con nuevas funcionalidades orientadas a las necesidades del administrador de red. Particularmente a la aplicación correspondiente al Módulo de Directorio, que se encuentra en su segunda versión, es necesario añadirle

nuevas funcionalidades, como son: la capacidad para la gestión de plantillas personalizando así las entradas, realizar búsquedas avanzadas filtrando los resultados por atributos, optimización de lecturas del árbol de directorio y la integración con servidor DNS.

Por lo que se plantea el siguiente **problema científico** ¿Cómo mejorar el Módulo de Directorio de la Plataforma de Gestión de Servicios Telemáticos en GNU/Linux? Se define como **objetivo general** el desarrollo de nuevas funcionalidades al Módulo de Directorio v2.0 de la Plataforma de Gestión de Servicios Telemáticos en GNU/Linux. Definiendo como **objeto de estudio** los procesos relativos a la gestión de servicios telemáticos en redes. El **campo de acción** estará dirigido a los procesos de administración del servidor de directorio OpenLDAP.

Como **idea a defender** se plantea que con el desarrollo de nuevas funcionalidades al Módulo de Directorio V2.0 se obtendrá una herramienta que brindará mayor facilidad para la administración del servidor de directorio OpenLDAP.

Los objetivos específicos que se plantearon son:

- Realizar el marco teórico de la investigación.
- Elaborar el modelo de negocio, modelo de sistema y modelo de diseño para la implementación de las nuevas funcionalidades.
- Desarrollar las nuevas funcionalidades para el Módulo de Directorio v2.0 de la Plataforma de Gestión de Servicios Telemáticos en GNU/Linux que permita mejorar la administración del servidor OpenLDAP.
- Elaborar y realizar las pruebas al sistema para verificar el correcto funcionamiento de las funcionalidades añadidas.

En correspondencia con el objetivo planteado y la problemática existente se ha decidido enfocar las tareas investigativas a:

- Estudio de las tendencias y herramientas actuales referentes a la administración de servidores de directorio.

- Selección de metodología, tecnologías, lenguajes y herramientas para el desarrollo de la aplicación.
- Definición del modelo de negocio para un mejor entendimiento de las funcionalidades a añadir.
- Análisis de los requisitos de software correspondientes al Módulo de Directorio para definir los casos de uso del sistema.
- Definición de patrones de diseño y arquitectura para la posible solución.
- Realización y documentación del diseño de cada una de las capas.
- Implementación de las clases definidas en el diseño.
- Realización de pruebas a la solución propuesta.

Métodos de la Investigación

Para el desarrollo del trabajo de diploma se utilizó el método analítico-sintético para descomponer el proceso de gestión de plantillas. Con el fin de esclarecer las principales características del problema a resolver para así identificar las prácticas y soluciones más óptimas a incluir.

El método histórico-lógico se empleó en el análisis del estado actual del proceso de gestión de plantillas para personalizar las entradas. Siendo contempladas las soluciones existentes, así como las tendencias más novedosas, revolucionarias o conservadoras que pudieran contener soluciones a la problemática o pudieran tributar a la resolución de la misma.

El método empírico de observación fue utilizado en el entendimiento de la problemática, permitiendo identificar las características, procesos y comportamiento del campo, así como las necesidades del usuario final, a partir de las cuales se puede definir las modificaciones a aplicar en el Módulo de Directorio.

Estructura del documento

El presente documento está estructurado en 4 capítulos.

Capítulo 1: Fundamentación teórica

Contiene los principales conceptos que se manejan a lo largo del trabajo; estado del arte del tema tratado tanto internacional, nacional como en la Universidad, así como las tendencias, técnicas, tecnologías, metodologías y software usados en la actualidad para la solución del problema planteado.

Capítulo 2: Características del sistema

Contiene el análisis de la situación problemática existente, la construcción de un modelo de negocio adecuado; la especificación de los requisitos que constituirá las bases de la propuesta del sistema, así como la elaboración de los diagramas de casos de uso del sistema, descripción y prototipos de interfaces.

Capítulo 3: Diseño del sistema

Representa la base de la futura implementación del sistema; donde se realiza la elaboración de diagramas de clases del Diseño, se define la arquitectura y los patrones de diseño a utilizar en la implementación del Módulo de Directorio.

Capítulo 4: Implementación y Prueba

Muestra cómo será implementado el sistema en términos de componentes, la estructura general de despliegue del sistema asociada a sus servidores de red y los casos de prueba realizados, así como el resultado de la aplicación de estas pruebas.

Capítulo 1: “Fundamentación teórica”

1.1 Introducción

Este capítulo trata sobre los principales conceptos referente al trabajo, así como las respectivas herramientas que se usan a nivel internacional para la administración de directorios, la metodología que se van a utilizar y las herramientas que se van a usar para el desarrollo del sistema.

1.2 Servicio de directorio

Los directorios representan contenedores de información descriptiva basada en atributos, apoyado en sofisticadas capacidades de filtrado. Los directorios por lo general no admiten transacciones complicadas como las que se encuentran en los sistemas de bases de datos diseñados para manejar grandes y complejos volúmenes de actualizaciones. Las actualizaciones de los directorios son normalmente cambios simples, o todo o nada, siempre y cuando estén permitidos. Es decir es una base de datos específicamente diseñada para la búsqueda y navegación. (2)

1.3 LDAP

LDAP por sus siglas en inglés Lightweight Directory Access Protocol, es un protocolo ligero de acceso a los servicios de directorio, basado específicamente los estándares X.500. LDAP utiliza como protocolo de transporte TCP/IP y otros servicios orientado a la conexión de transferencia. Es un protocolo estándar del Grupo Especial sobre Ingenierías de Internet (IETF) y está descrito en la especificación técnica del LDAP: RFC4510. (2)

Actualmente hay dos versiones del protocolo en uso: una es LDAPv2 que se describe en el RFC 1777 y la LDAPv3 descrito en el RFC 2251.

1.3.1 Características del protocolo LDAP

EL protocolo LDAP cuenta con varias características entre las que se encuentran:

- Es independiente del proveedor y funciona con X.500, pero no lo requiere.
- Soporta remisiones o referrals que permiten implementar bases de datos distribuidas.

- Los elementos de LDAP son cadenas de caracteres.
- Soporta operaciones como:
 - ✓ Buscar entradas en un directorio.
 - ✓ Adicionar entradas a un directorio.
 - ✓ Eliminar entradas de un directorio.
 - ✓ Modificar una entrada de un directorio.
 - ✓ Iniciar una nueva sesión en el servidor LDAP, autenticándose y especificando la versión del protocolo; esta operación es llamada bind.
 - ✓ Cancelar la operación previa enviada al servidor.
 - ✓ Comparar las entradas en un directorio dado el valor de un atributo.
 - ✓ Realizar operaciones extendidas.
 - ✓ Cambiar el nombre de una entrada del directorio.
 - ✓ Finalizar una sesión en el servidor LDAP.
- Los elementos del protocolo de LDAP son codificados para el intercambio con Reglas de Codificación Básicas⁴. Sin embargo, debido a los altos costos involucrados en el uso de ciertos elementos de las reglas, se colocan las siguientes restricciones adicionales en la codificación de los elementos del protocolo LDAP:
 - ✓ Sólo se utilizará la forma definitiva de codificación de longitud.
 - ✓ Las cadenas de bits y cadenas de octeto deben ser codificadas en la forma primitiva solamente. (3)
- El protocolo LDAPv3 ofrece mecanismos de cifrado SSL, y autenticación para permitir el acceso seguro a la información almacenada en el directorio.
- Funciona por defecto en el puerto TCP 389 y cuando usa SSL se conecta por el puerto 636.

⁴ Formatos de codificación definidos como parte del estándar ASN.1. cuyas siglas son BER.

1.3.1 Estructura LDAP

El protocolo LDAP organiza la información de forma jerárquica usando directorios que pueden contener gran variedad de información estructurando esta en forma de árbol, denominada DIT⁵ es decir árbol de información de directorio. La información contenida en este árbol se le llama entrada y consiste en un conjunto de pares clave/valor llamado atributos que contienen información acerca del objeto que representa la entrada. Algunos atributos representan información de usuario y a estos se les llama atributos de usuario. Otros atributos representan la información operativa y / o administrativa y son llamados atributos operativos. (4).

Cada entrada tiene un nombre completo relativo, RDN⁶ que se conforma por la ruta de la entrada en relación con sus entradas superiores, y posee como identificador un nombre distinguido DN por sus siglas en inglés que se conforma concatenando el RDN con el DN de la raíz del árbol; un ejemplo es siguiendo el árbol de la figura 1 el DN del estudiante yodiaz sería cn=yodiaz,ou=estudiantes,dc=uci,dc=cu.

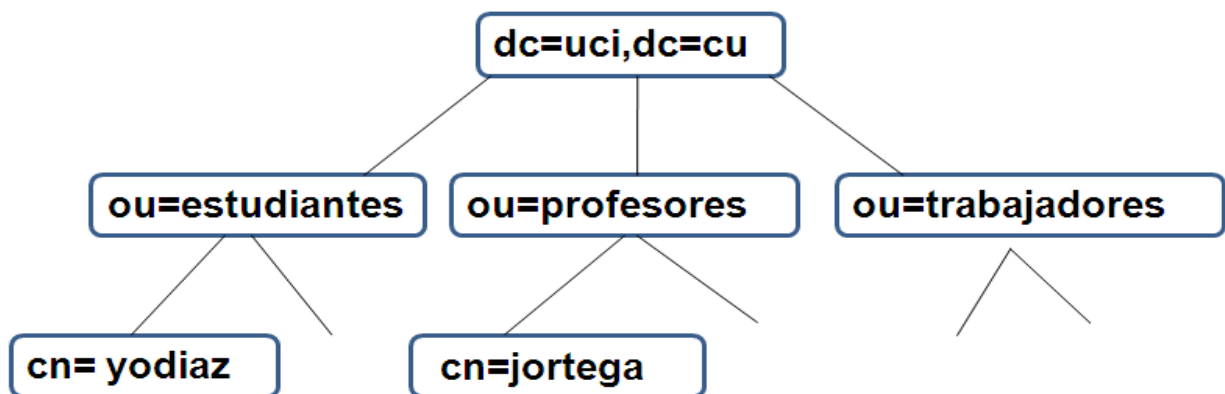


Figura 1: Estructura de un directorio LDAP.

Para determinar qué tipo de objeto ha de guardarse en el árbol de directorio se utilizan los esquemas, entiéndase por esquema clase de objetos y atributos asociados El tipo de objeto se determina mediante la clase de objeto. La clase de objeto especifica atributos obligatorios y opcionales que constituyen sus valores.

⁵ Directory Information Tree

⁶ Relative Distinguished Name

1.4 Servidor de directorio

Se conoce como servidor de directorio a la aplicación que contiene la implementación del protocolo LDAP. Permite atender las peticiones de los clientes, dar respuesta a las búsquedas, y realizar modificaciones sobre el árbol de directorio; así como almacenar y organizar la información de los usuarios y recursos de una red.

En las siguientes secciones se hablará de algunos servidores de directorios usados en el mundo.

1.4.1 Fedora Directory Server

Es un servidor de código abierto, desarrollado por Red Hat y la comunidad de Fedora. Entre sus características se encuentra la capacidad de replicación multimaster, además de compatibilidad con el Active Directory de Microsoft. Permite administrar usuarios y grupos de forma gráfica y en consola, así y cuenta con buena documentación. Este servidor está integrado a la distribución Fedora de GNU/Linux lo cual restringe su uso.

1.4.2 Active Directory

Es el servidor de directorio desarrollado por Microsoft que viene acoplado a Windows Server 2003 y a Windows Server 2008. Entre sus características se encuentran que posee buena escalabilidad, seguridad, flexibilidad y replicación entre servidores. Además permite la administración de grupos con una gran variedad de políticas. Este servidor es de software privativo y hay que pagar una licencia para su uso por lo que no es la mejor opción para países subdesarrollados.

1.4.3 OpenLDAP

Es un servidor de directorio que se acopla a varios sistemas operativos entre los que se encuentra Apple Mac OS X, Linux: Debian, Red Hat, Suse, Fedora, Mandrake, FreeBSD, Microsoft Windows 2000, XP entre otros, permite la replicación entre servidores. Fue desarrollado por el proyecto de mismo nombre y se distribuye bajo la licencia GNU, se dispone de su código fuente lo que permite realizar modificaciones propias. Entre las principales características encontramos:

Soporte LDAPv3: OpenLDAP soporta la Capa de autenticación y seguridad (SASL), la Seguridad de la capa de transporte (TLS) y la Capa de conexión segura (SSL).

Soporte IPv6: OpenLDAP soporta la próxima generación del protocolo de Internet versión 6.

LDAP sobre IPC: OpenLDAP se puede comunicar dentro de un sistema usando comunicación interproceso (IPC). Esto mejora la seguridad al eliminar la necesidad de comunicarse a través de la red.

Soporte LDIFv1: Provee compatibilidad completa con el formato de intercambio de datos, Data Interchange Format (LDIF) versión 1.

Servidor Stand-Alone mejorado: Incluye un sistema de control de acceso actualizado, conjunto de hilos y herramientas mejoradas. (2)

1.5 Sistemas para la gestión de servidores de directorio

Existen un grupo de herramientas que les facilitan la administración de servidores de directorio de forma gráfica a los administradores de red tanto privativa como de código abierto las que se relacionan en las siguientes secciones.

1.5.1 phpLDAPAdmin

PhpLDAPAdmin está desarrollada en el lenguaje de programación PHP, permite ver los esquemas LDAP, realizar búsquedas, gestionar entradas LDAP. Además se puede ver y navegar por toda la estructura de directorio (árbol jerárquico) y exportar e importar ficheros en el formato estándar LDIF, una de sus principales deficiencias es que no es compatible con versiones superiores al OpenLDAP 2.4.11, ni con PHP 5 lo que puede provocar error en la administración del servidor de OpenLDAP, por lo que no se recomienda su uso para la versión 2.4.23.

1.5.2 Softerra LDAP Administrator

Es una herramienta de gestión en cuyas características se puede apreciar que cuenta con un funcionamiento rápido con grandes cantidades de datos, permite la gestión de grupos, búsqueda avanzada, cuenta con autenticación SASL y posee certificados SSL. Está diseñado para el sistema operativo Windows, por lo que hay que pagar una licencia para su uso, y Cuba desde abril del 2004 se ha trazado como estrategia migrar a Software Libre por lo que no se recomienda el uso de esta herramienta. (14)

1.5.3 OpenDS

Es un software un tanto menos conocido que surgió como un proyecto interno de la empresa de software SUN, basado en los estándares LDAPv3 y DSMLv2, está desarrollado en java y es necesario un entorno de ejecución como Java Runtime Enviroment (JRE) para su funcionamiento, además por la cantidad de RAM que consume el JRE tiene como requisitos mínimos para su correcta ejecución en trabajo con pequeño volumen de información 2GB de RAM (6).

1.5.4 Gosa

Es una herramienta de administración web para la gestión de cuentas muy usada sobre todo en países Europeos. Está escrita en PHP y bajo la licencia GNU GPL. Puede administrar usuarios, grupos, listas de distribución de correo, clientes ligeros, y los faxes. Los usuarios pueden obtener información sobre ellos mismos, dígame cambiar sus contraseñas, y ver las estadísticas de fax. Asimismo, los usuarios pueden configurar sus cuentas de correo propio, pero estas configuraciones son limitadas, pues solo se puede usar para el protocolo POP3 ⁷de correo.

1.5.5 Red Hat Directory Server

Una de las herramientas más usadas precisamente por estar acoplada al sistema operativo Red Hat es el Red Hat Directory Server, la cual centraliza configuraciones de aplicaciones, así como perfiles de usuario, información de grupo, políticas e información de control de acceso dentro de un sistema operativo independiente de la plataforma. Entre sus principales características se encuentra la eliminación de la redundancia de datos y automatización del mantenimiento de estos. Pero esta herramienta solo administra el servidor de directorio que se encuentra acoplado al sistema operativo Red Hat y no se puede administrar de forma remota por lo que es necesario estar sentado en el servidor lo cual limita su uso, además se encuentra bajo licencia privativa que no es recomendable usar en Cuba por el proceso de migración que se está llevando a cabo.

En la Universidad de las Ciencias Informáticas se desarrolla una Plataforma de Gestión de Servicios Telemáticos en GNU/Linux al que pertenece el Módulo de Directorio que gestiona el servidor de directorio OpenLDAP. Actualmente se encuentra en su segunda versión desarrollada en el curso académico 2009-2010. Dicha herramienta permite redefinir y eliminar esquemas, gestionar entradas, generar certificados,

⁷ Post Office Protocol

exportar e importar ficheros en formato LDIF, administrar el servidor de forma remota, además establece comunicación segura con el servidor de directorio. Se puede decir que el sistema engloba las mejores funcionalidades de otras herramientas en una totalmente desarrollada por Cuba y es lo que se seguirá mejorando con este trabajo.

1.6 Lenguajes, Metodología y Herramientas

1.6.1 Lenguaje de Modelado. Lenguaje Unificado de Modelado

El Lenguaje Unificado de Modelado UML es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG⁸. Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener, y controlar la información sobre tales sistemas (7). Se puede aplicar en el desarrollo de software entregando gran variedad de formas para dar soporte a una metodología de desarrollo de software. Aunque no especifica en sí mismo qué metodología o proceso usar, está pensado para ser útil en un proceso de desarrollo iterativo (8).

Como parte del modelo gráfico de UML se identifican:

Los elementos: Abstracciones que constituyen los bloques básicos de construcción.

Las relaciones: Conexión semántica entre elementos de un modelo.

Los diagramas: Es la representación gráfica de un conjunto de elementos y sus relaciones. Visualizan un sistema desde diferentes perspectivas.

Como cualquier lenguaje, UML está compuesto además por reglas utilizadas para combinar los elementos anteriormente citados. Éstas establecen cómo llamar a los elementos, relaciones y diagramas, cómo se pueden ver y utilizar esos nombres por otros y cómo se relacionan apropiada y consistentemente unos elementos con otros. (7)

Puede decirse además que UML es un lenguaje consolidado, estándar de facto y fácil de aprender permitiendo la comunicación fluida entre los diversos actores acerca del modelo. Por otra parte es necesario recalcar que UML no ha sido diseñado para modelar procesos del negocio y que al presentar un enfoque orientado a objetos podría contradecir un enfoque orientado al negocio, este lenguaje suele estar

⁸ Object Management Group

orientado a los arquitectos de sistemas y diseñadores de software y pensado para un público eminentemente técnico. (8)

Desde la primera versión del módulo se ha utilizado UML como lenguaje de modelado por las características antes mencionadas. Siendo ésta la tercera versión también se usa el mismo para mantener uniformidad en la documentación generada.

1.6.2 IDEF0

IDEF0 es un método de modelación concebido para representar de manera estructurada y jerárquica las actividades que conforman un sistema o empresa y los objetos o datos que soportan la interacción de esas actividades. Un modelo IDEF0 se compone de una serie jerárquica de diagramas que permiten mediante niveles de detalle, describir las funciones especificadas en el nivel superior. En las vistas superiores del modelo la interacción entre las actividades representadas permite visualizar los procesos fundamentales que sustentan la organización. Los elementos gráficos utilizados para la construcción de los diagramas IDEF0 son cuadros y flechas (9). La semántica de utilización de estos elementos gráficos se representa en la figura 2:

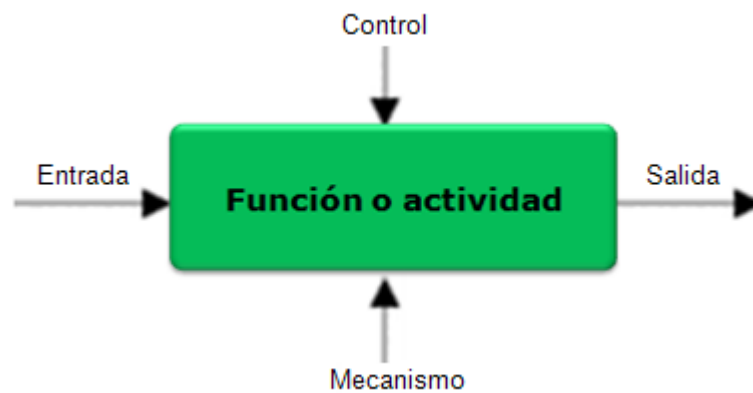


Figura 2: Elementos gráficos para el modelado con IDEF0.

Actividad: Se representa con un cuadro, indica una función, proceso o transformación.

Entrada: Designan la materia o información que es transformada o consumida por la actividad.

Salida: Todo aquello que es producido por la actividad o proceso.

Control: Objetos que regulan cómo, cuándo y si una actividad se ejecuta o no.

Mecanismos: Todos aquellos recursos que son necesarios para llevar a cabo un proceso (personas, herramientas, software, información).

Se deben evitar en el modelo aquellas actividades que no aportan información de valor para la siguiente actividad de la cadena. La cadena de actividades debe ser ininterrumpida, y se considerará el flujo cronológico de actividades sin importar la duración de tiempo físico que media entre una actividad y la otra. (10)

Por su parte, IDEF0 es una notación simple que cualquier empleado puede usar para describir qué hace en el negocio y que permite representar el proceso cronológicamente. A través del modelado con IDEF0 se involucra a los empleados de la organización en la modelación del negocio lo que permite ahorrar tiempo simultaneando el trabajo en varias áreas, así como obtener un modelo más fiel ya que ha sido elaborado por sus protagonistas incorporándose en el flujo, los datos que entran y salen de las actividades, así como las reglas del negocio y los actores, todo en la misma vista con la observación de que estos diagramas son estáticos por lo que no permiten representar el modelo desde la perspectiva de su comportamiento. IDEF0 permite descomponer una actividad como un proceso a su vez y lo que hace que sea más entendible los diagramas, por lo que es muy usado para modelar procesos complejos o muy grandes y esta es precisamente la razón por lo que se ha decidido utilizar en la modelación del negocio del Módulo de Directorio.

1.7 Metodología de desarrollo

En un proyecto de desarrollo de software la metodología define quién debe hacer qué, cuándo y cómo debe hacerlo (7). Escoger la metodología apropiada para desarrollar un software es una tarea difícil y de suma importancia pues ello juega un papel importante, el tiempo de desarrollo, la cantidad de personas que desarrollarán y los requerimientos del cliente, pues se necesita definir una metodología que se adapte más al sistema que se quiere desarrollar. Otro factor que influye en seleccionar la metodología apropiada está en escoger entre el desarrollo con enfoque ágil y tradicional o pesado. Los métodos pesados o rígidos son los que 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. Los métodos ágiles o ligeros que son los que intentan evitar los tortuosos y burocráticos caminos de las metodologías tradicionales enfocándose en las personas y los resultados y la adaptabilidad para dar respuesta pronta y eficaz a los inevitables cambios. Para el desarrollo del Módulo de Directorio se

decidió utilizar la metodología pesada RUP pues no se cuenta con un cliente específico y es necesario documentar bien todo el proceso de desarrollo para un mejor entendimiento en versiones posteriores.

La metodología RUP por sus siglas en inglés *Rational Unified Process* es una forma disciplinada de asignar tareas y responsabilidades en una organización de desarrollo con el objetivo de asegurar la producción de software de calidad. Constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Está definido por 9 flujos de trabajo, los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo, y en 4 fases como se define a continuación:

Flujos de trabajo

- **Modelamiento del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.
- **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.
- **Análisis y diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.
- **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán y la ubicación en ellos de los componentes y la estructura de capas de la aplicación.
- **Prueba:** Busca los defectos a lo largo del ciclo de vida.
- **Instalación:** Produce release del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.
- **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.
- **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

Fases

- **Inicio:** El objetivo en esta etapa es determinar la visión del proyecto.
- **Elaboración:** En esta etapa el objetivo es determinar la arquitectura óptima.
- **Construcción:** En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
- **Transición:** El objetivo es llegar a obtener el release del proyecto. (4)

En cada fase se ejecutarán una o varias iteraciones como muestra la siguiente figura:

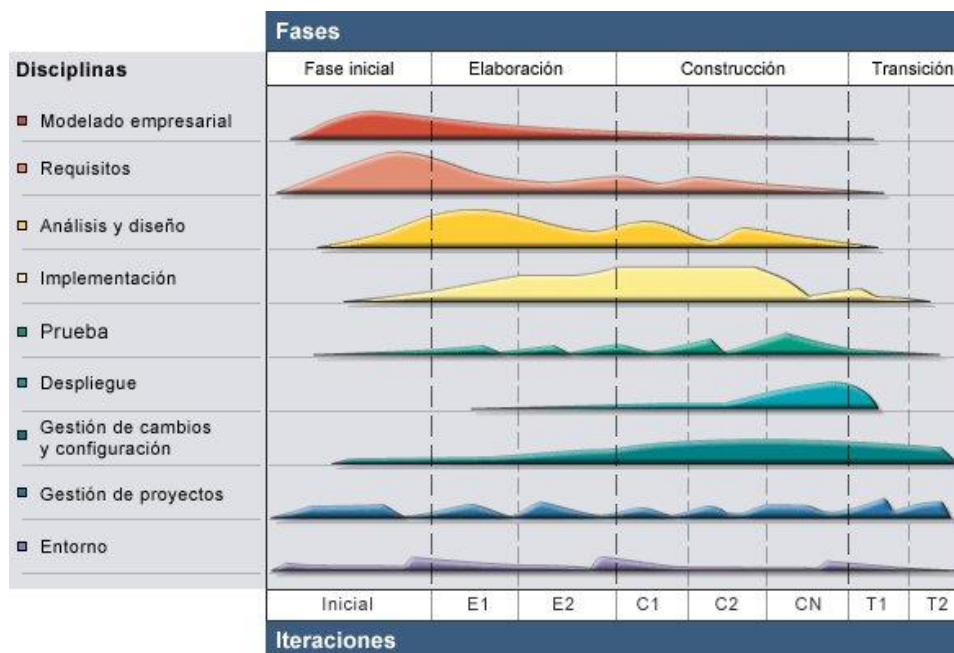


Figura 3: Fases y flujo de la metodología RUP.

El proceso de desarrollo de software se caracteriza por estar dirigido por casos de uso, centrado en la arquitectura y por ser iterativo e incremental.

Dirigido por Casos de Uso: Un caso de uso es una secuencia de acciones relacionadas lógicamente que debe realizar el sistema con las que se da cumplimiento a un objetivo y produce resultados de valor al usuario. Estos casos de uso se obtienen cuando se modela el negocio y se representan a través de los requerimientos. RUP se basa en estos casos de uso para describir lo que se espera del software. A partir de ellos se realizan todos los artefactos de cada iteración, es decir, guían las actividades de las mismas.

La idea es que cada iteración tome un conjunto de casos de uso o escenarios y desarrolle todo el camino a través de las distintas disciplinas: diseño, implementación, prueba, etc.

Centrado en la arquitectura: La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, da una perspectiva del sistema completo y describe los elementos más importantes. Para que los desarrolladores obtengan una visión común del software es muy importante la arquitectura, ésta se ve condicionada por los casos de uso por lo que cuando más se conozca la arquitectura mejor se hará una captura de requisitos para desarrollar los casos de uso. RUP plantea que no existe un modelo único que cubra todos los aspectos del sistema por lo que existen diferentes maneras de definir la arquitectura de software de un sistema. Además de lo anterior planteado RUP establece que la arquitectura debe empezarse a definir desde los inicios del proyecto y que deben realizarse refinamientos sucesivos de la arquitectura para comprender, desarrollar y producir el sistema económicamente.

Iterativo e incremental: RUP propone que el sistema se desarrolle en pequeños pasos, cada fase se divide en iteraciones. Cada iteración está compuesta por actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Las iteraciones resultan ser un incremento y permiten verificar si se han cumplido los objetivos establecidos en función de la evaluaciones de las iteraciones precedentes, así el proyecto irá avanzando sin correr el riesgo de que existan problemas al final.

Como se ve RUP se basa en los casos de uso para describir lo que se espera del software y está orientado a la arquitectura del sistema, haciendo uso de artefactos en cada iteración convirtiéndola en una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software y se basa en el Lenguaje Unificado de Modelado (UML) como herramienta principal.

1.8 Lenguaje de programación y herramientas

1.8.1 Python

Python es un lenguaje interpretado de programación, multiplataforma, muy flexible, su implementación está bajo la licencia *Python Software Foundation License*, la cual es de código abierto. Se ha decidido utilizar para el desarrollo del software, principalmente por poseer una sintaxis muy clara y legible; por ser orientado a objeto lo que permite crear programas con componentes reutilizables. También por su

modularidad completa con soporte de paquetes jerárquico e incluir una gran colección de módulos estándar que proporcionan funcionalidades para el manejo de ficheros, además de la existencia de módulos de terceros para disímiles tareas. Python entre sus características están el tipado dinámico, es decir, una variable puede tomar diferentes tipos de datos durante la ejecución del programa y utiliza la indentación para delimitar los bloques.

1.9 Herramienta CASE

1.9.1 Visual Paradigm

Visual Paradigm es una herramienta CASE multiplataforma que utiliza UML como lenguaje de modelado y facilita enormemente el desarrollo de software. Se ha decidido utilizar por la eficiencia y facilidad de uso, soporta el ciclo de vida completo del desarrollo de software dígase análisis y diseño orientado a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, también permite realizar ingeniería inversa, es decir, del código generar diagramas y está soportado para varios lenguajes de programación entre ellos Python. (11)

1.10 Entorno de desarrollo integrado

Un entorno de desarrollo integrado o IDE (*Integrated Development Environment*) por sus siglas en inglés es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica en algunos casos. Pueden ser aplicaciones solas o ser parte de otras. Facilitan el desarrollo de un software pues provee un marco de trabajo amigable para la mayoría de los lenguajes de programación. Es posible que un mismo IDE pueda funcionar con varios lenguajes de programación.

1.10.1 Eclipse

El entorno de desarrollo integrado que se decidió utilizar es el Eclipse, el cual se encuentra bajo la licencia *Eclipse Public License*, es de código abierto y entre sus características están el resaltado de sintaxis, la compilación en tiempo real y permite la posibilidad de añadirle módulos o plugins, dando la opción de incluirle el plugin pydev que se utiliza para el desarrollo en el lenguaje Python, cuenta con completamiento de código lo que es útil en el desarrollo del sistema. Además que permite trazar el código dando la

posibilidad de ver qué valores van tomando las variables y del comportamiento de clases y listas que se utilizan en el desarrollo del sistema.

1.11 Interfaces Gráficas de Usuarios

1.11.1 Qt Designer

Qt Designer es la herramienta de Qt para el diseño y la creación gráfica de interfaces de usuario (GUI) de los componentes de Qt, es multiplataforma y entre sus características principales cuenta con extensa documentación de sus componentes, permite incluirle eventos a los botones lo que ahorra tiempo al programador. También brinda la posibilidad de validar la entrada de texto en algunos componentes como el LineEdit, lo cual es de mucha ayudada en la aplicación sobre todo para validar los criterios de la búsqueda. Permite además ejecutar la interfaz dando una vista previa de la misma y permitiendo determinar errores en la ubicación de los componentes y con el comando `pyuic4` se puede generar el código Python de las interfaces. Por las características anteriormente descritas es que se decide utilizar Qt Designer para el diseño de las interfaces del Módulo de Directorio.

1.12 Conclusiones

En el presente capítulo se definieron los conceptos relacionados con el servidor de directorio, y se plantearon las características y estructura del protocolo LDAP para un mejor entendimiento del funcionamiento de los servidores de directorio. Después de un análisis de las implementaciones de servidores existentes se escoge OpenLDAP como servidor de directorio a administrar. Se realizó un estudio de las herramientas que se utilizan en el mundo para la administración de este servidor; y se propusieron las herramientas y metodología a utilizar para el desarrollo de la solución a la problemática planteada.

Capítulo 2: Características del sistema

2.1 Introducción

En el presente capítulo se describen los procesos asociados a la gestión de un servidor de directorio OpenLDAP en sistemas operativos GNU/Linux. Se muestra el modelamiento de negocio utilizando la notación IDEF0. Se expresa una propuesta del sistema en términos de calidad asociada a la usabilidad, el soporte, la portabilidad y otros aspectos. Además son representados las características o requisitos funcionales del sistema a desarrollar

2.2 Problema a resolver

Actualmente el Módulo de Directorio v2.0 al iniciarse muestra el árbol de directorio del servidor al que el usuario se autenticó y en este es posible seleccionar una de las entradas existentes y modificarla o eliminarla, además que permite crear nuevas de los tipos predeterminados, y da la facilidad de gestionar esquemas. También el usuario puede modificar, adicionar o eliminar atributos binarios como pueden ser imágenes en las entradas que lo permitan. La aplicación permite generar su propio certificado y establece comunicación segura utilizando estos certificados, también brinda la posibilidad de migrar los datos contenidos en un Directorio Activo a un servidor OpenLDAP e importar o exportar ficheros en formato LDIF.

El módulo carece de funcionalidades que permitan al administrador de red gestionar las plantillas para personalizar las entradas en el servidor de directorio de la empresa, además de poder realizar búsquedas avanzadas en el árbol de directorio con diferentes criterios de búsqueda. También es necesario optimizar la lectura de las entradas del árbol a la hora de mostrarlas para evitar que la aplicación se tarde en cargar las entradas en un servidor que posea una gran cantidad de usuarios. Hasta el momento la aplicación no permite la integración con otros servicios telemáticos como puede ser el servidor DNS por lo que es necesario darle esta opción al administrador de la red. La automatización de los procesos antes mencionados le facilitará e optimizará el trabajo a los administradores de redes.

2.3 Información que se maneja

Es importante tener en cuenta la información con la que trabaja la aplicación para la integración del servidor de directorio con un servidor DNS donde se incluye un nuevo esquema al servidor y si este no tiene el formato correcto puede que las entradas donde se especifiquen atributos u objetos de clase del nuevo esquema no funcionen correctamente. También es necesario tener conocimiento de los esquemas y atributos del servidor de directorio para crear nuevas plantillas, pues las plantillas se forman con esquemas y permiten crear entradas al servidor de directorio donde se pueden especificar desde los datos de las personas de la empresa que utiliza el software hasta de los recursos con que esta cuenta.

Para la creación de plantillas es necesario además la modificación y creación de ficheros en el servidor por lo que se debe considerar que los ficheros de configuración son un eslabón esencial para el correcto funcionamiento del servidor de directorio pues constituyen el núcleo del sistema, lo que una incorrecta estructura del contenido puede ocasionar un mal funcionamiento del servidor e incluso la interrupción del servicio afectando así a los usuarios del mismo.

2.4 Solución Propuesta

Para darle solución a la problemática anteriormente mencionada se ha decidido añadirle nuevas funcionalidades al Módulo de Directorio v2.0 tales como integración del servidor de directorio con un servidor de DNS, optimizar la lectura de las entrada cuando inicia la aplicación para mejorar el tiempo que demora en mostrar el árbol del directorio. También permitirle al administrador realizar búsquedas avanzadas en el árbol de directorio de acuerdo a filtros o criterios de búsqueda que necesite, además de poder definir plantillas para personalizar entradas personalizadas de acuerdo a las características de la empresa donde se utilice la aplicación.

Una herramienta como la que se propone con interfaz cómoda y con nuevas funcionalidades, facilitará el trabajo de administración de un servidor OpenLDAP y no tendrá costos de licencia lo que favorecerá a la comunidad de administradores cubanos.

2.5 Modelo de Procesos

2.5.1 Diagrama A-0

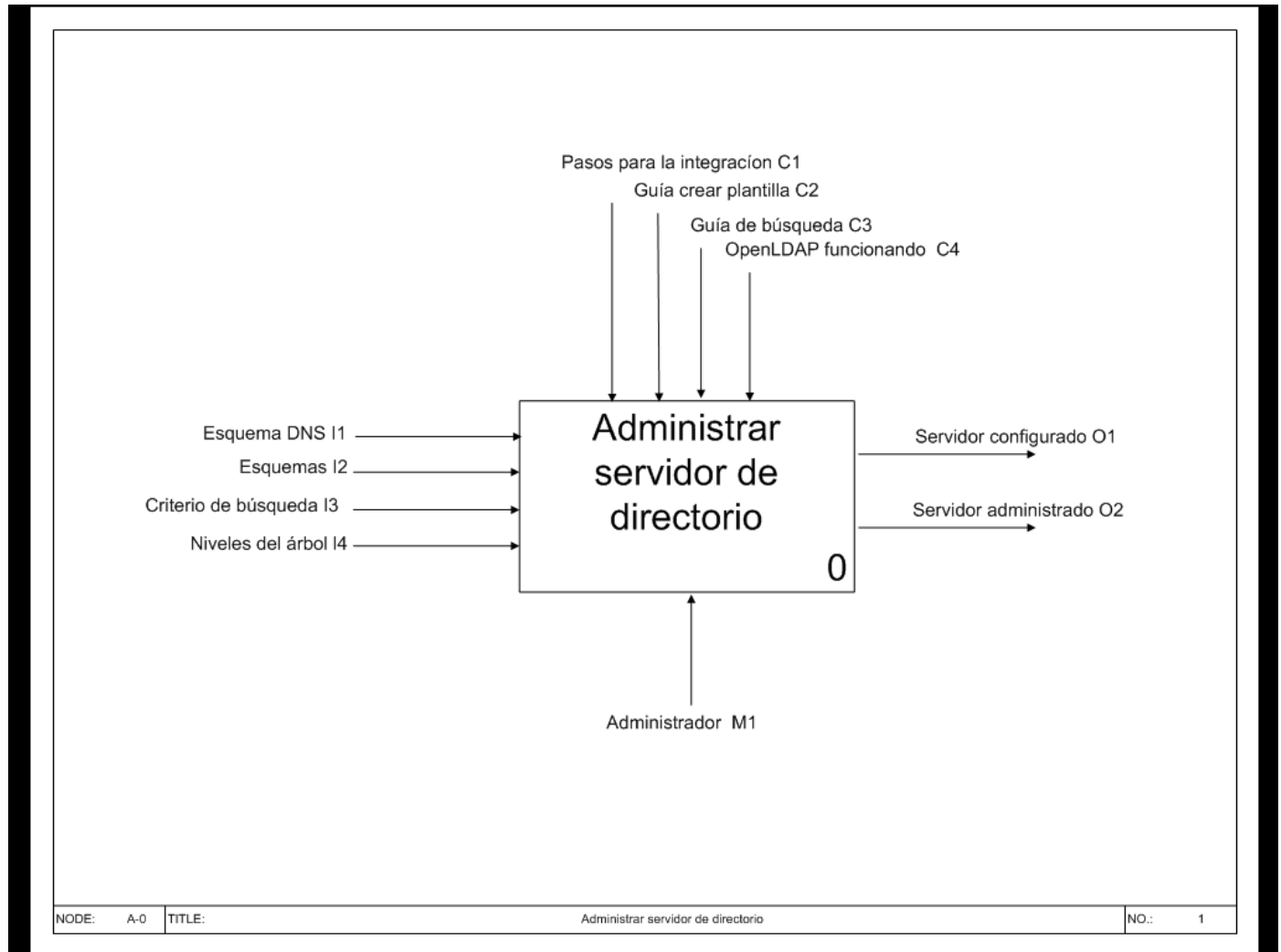


Figura 4: Representación del proceso general "Administrar servidor de directorio".

Tabla 1: Descripción del proceso Administrar servidor de directorio.

Ficha de Proceso	
Proceso	Administrar servidor de directorio
Notación:	A-0

Entradas:	I1 I2 I3 I4	I1: Esquema DNS I2: Esquemas I3: Criterio de Búsqueda I4: Niveles del árbol
Controles:	C1 C2 C3 C4	C1: Pasos para la integración C2: Guía crear plantilla C3: Guía de búsqueda C4: OpenLDAP funcionando
Salidas:	O1 O2	O1: Servidor configurado O2: Servidor administrado
Mecanismos:	M1	M1: Administrador
Descripción del Proceso		
Este proceso consiste en extender las capacidades de administración y configuración del servicio de directorio, entre las que se encuentran la creación de plantillas, la realización de búsqueda, configuración del servidor de directorio para integrar con un servidor DNS y mejorar la lectura de las entradas del servidor OpenLDAP.		

2.5.2 Diagrama A0

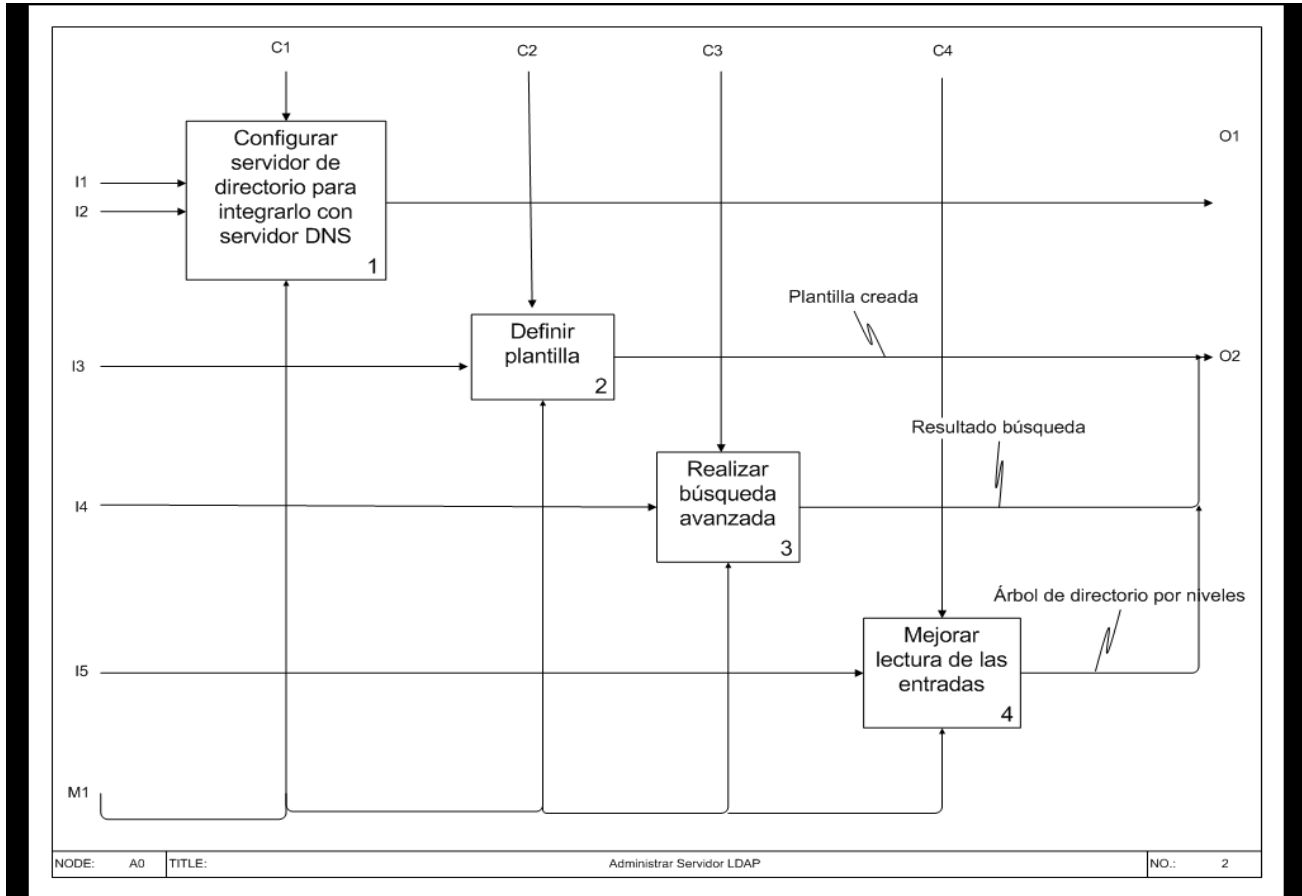


Figura 5: Detalles del proceso general.

2.5.3 Descripción de los procesos del negocio

Tabla 2: Descripción del proceso Configurar servidor de directorio para integrarlo con servidor DNS.

Ficha de Proceso	
Proceso	Configurar servidor de directorio para integrarlo con servidor DNS
Notación:	A1
Entradas:	I1 I1: Esquema DNS consiste en un esquema que tiene como atributos

		las zonas y características que tienen los ficheros del servidor de DNS.
Controles:	C1	C1: Pasos para la integración: consisten en incluirle al servidor OpenLDAP el esquema DNS así como realizar las configuraciones necesarias como asignación de permisos del grupo OpenLDAP a los ficheros modificados.
Salidas:	O1	O1: Servidor configurado.
Mecanismos:	M1	M1: Administrador.
Descripción del Proceso		
Se busca y añade al servidor de directorio el esquema con los atributos y clases de objeto referente a las zonas y características del servidor DNS. Se cambian los permisos al grupo OpenLDAP a todos los ficheros modificados, quedando así integrado el servidor, en este proceso se definen las zonas del DNS en una entrada en el LDAP.		

Tabla 3: Descripción del proceso Definir plantilla.

Ficha de Proceso		
Proceso	Definir plantilla	
Notación:	A2	
Entradas:	I2	I2: Esquemas: son los esquemas que tiene el servidor incluido.
Controles:	C2	C2: Guía crear plantilla: conjunto de pasos a seguir para crear una plantilla entre los que se encuentran buscar los esquemas

		del OpenLDAP y crear el fichero con las nuevas plantillas.
Salidas:	O2	O2: Servidor administrado: específicamente se obtiene una nueva plantilla.
Mecanismos:	M1	M1: Administrador: persona que va a trabajar con el sistema para administrar el servidor de directorio.
Descripción del Proceso		
Dado los esquemas que se encuentren en el servidor de directorio al mostrarlos el administrador puede escoger un esquema o varios y se le asigna un nombre creando así una nueva plantilla.		

Tabla 4: Descripción del proceso Realizar búsqueda avanzada.

Ficha de Proceso		
Proceso	Realizar búsqueda avanzada	
Notación:	A3	
Entradas:	I3	I3: Criterio de búsqueda se define como criterio de búsqueda, base, nombre de la entrada, atributos del OpenLDAP y filtros descritos en la RFC 1558.
Controles:	C3	C3: Guía de búsqueda: criterio y pasos a tener en cuenta para buscar.
Salidas:	O2	O2: Servidor administrado, específicamente se obtiene los

		resultados de la búsqueda.
Mecanismos:	M1	M1: Administrador.
Descripción del Proceso		
Realizar una búsqueda por el árbol de directorio de acuerdo a los criterios de búsqueda establecidos por el administrador que pueden ser criterios además definidos mediante filtro con una sintaxis determinada.		

Tabla 5: Mejorar lectura de las entradas.

Ficha de Proceso		
Proceso	Mejorar lectura de las entradas	
Notación:	A4	
Entradas:	I5	I5: Niveles del árbol.
Controles:	C4	C4: OpenLDAP funcionando, es decir, que tenga definido la base y al menos una entrada que por defecto son los datos del usuario administrador.
Salidas:	O2:	O2: Servidor administrado: específicamente se muestra el árbol de directorio.
Mecanismos:	M1	M1: Administrador.
Descripción del Proceso		
El proceso consiste en mostrar el panel e ir leyendo el árbol de directorio por niveles e ir mostrándolo, de esta forma se realiza el proceso de mostrar el árbol de forma más rápida.		

2.6 Especificación de los requisitos de software.

2.6.1 Requerimientos Funcionales

Son capacidades o condiciones que el sistema debe cumplir y se definen a partir de los casos de uso del negocio donde se obtienen las actividades que serán objeto de automatización. Para el Módulo de Directorio se determinaron los siguientes requisitos funcionales:

RF1 Configurar servidor de directorio para integrar con servidor DNS.

Permitirá preparar el servidor de directorio configurando este para que sea integrado con un servidor DNS.

RF2. Buscar esquema DNS.

Se busca el esquema DNS en el servidor OpenLDAP.

RF3 Buscar esquemas.

Se buscan los esquemas existentes en el servidor de directorio.

RF4 Mostrar esquemas.

Se muestran los esquemas existentes en el servidor de directorio.

RF 5 Definir plantilla.

Permitirá definir una nueva plantilla, llenando los siguientes datos.

- Nombre.
- Esquemas donde se puede seleccionar uno o varios de los mostrados.

RF6 Modificar plantilla.

Permitirá modificar una plantilla, cambiando el nombre o añadiéndole o quitándole esquemas.

RF7 Eliminar plantilla.

Permitirá eliminar una plantilla.

RF 8 Mostrar árbol de plantillas.

RF 9 Buscar atributos.

Buscar los atributos existentes en el servidor de directorio.

RF 10 Mostrar atributos.

Mostrar todos los atributos existentes en el servidor de directorio.

RF11 Realizar búsqueda avanzada.

Permitirá realizar una búsqueda en el árbol de directorio mediante los siguientes campos:

- Nombre.
- Atributos del servidor de directorio.
- Niveles de búsqueda que pueden ser:
 - ✓ Subárbol completo.
 - ✓ Un nivel bajo la base.
 - ✓ En la base.
- Filtros definidos en el RFC 1558.

RF12 Optimizar lectura de las entradas.

Permitirá realizar la búsqueda de entradas por los niveles del árbol de forma más rápida y eficiente que en la versión del Módulo de Directorio 2.0.

RF 13 Buscar entradas.

Permitirá buscar entradas existentes en el servidor de directorio.

2.6.2 Requerimientos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Se debe pensar en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. Normalmente están vinculados a requisitos funcionales, es decir una vez se conozca lo que el sistema debe hacer se puede determinar cómo ha de comportarse, qué cualidades debe tener o cuán rápido debe ser. Existen múltiples categorías para clasificar a los requisitos no funcionales, siendo las siguientes representativas de un conjunto de aspectos que se deben tener en cuenta.

Requisitos de Software.

Para lograr el funcionamiento del sistema es necesario tener instalado Python 2.6 y se ha probado en Sistema Operativo Debian 5 y funciona correctamente además en las versiones de Ubuntu 10.4 y 10.10.

Requisitos de Hardware

El sistema fue probado bajo las siguientes condiciones:

- Microprocesador Core 2 Duo.
- 1 GB de memoria RAM.
- 5 GB de espacio libre en el disco duro.

Por lo que se garantiza que para estos requerimientos el Módulo de Directorio funciona correctamente.

Restricciones de diseño e implementación

- Será empleado Python como lenguaje de programación para la construcción del sistema.
- La codificación estará regida por la guía de estilo propuesta por Guido van Rossum creador del lenguaje de programación Python.
- Las herramientas utilizadas están divididas en:
 - Herramientas de diseño: Qt Designer en su versión 4.2.1 para elaborar las interfaces de usuario.
 - Herramientas de codificación: Eclipse con Pydev. Pydev es un plugin o complemento para el entorno de desarrollo Eclipse que permite la programación en Python.

Requisitos de seguridad

- **Confidencialidad**

Define que la información manejada por el sistema está protegida de acceso no autorizado y su incorrecta divulgación. En el módulo de Directorio esto se garantizará autenticando el sistema contra el servidor de directorio que es donde se establece el nivel de acceso y las políticas para un usuario determinado, y al sistema sólo podrán autenticarse usuarios con privilegios de administrador en el servidor de directorio OpenLDAP.

- **Integridad**

Define que la información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes, de la misma forma será considerada igual a la fuente o autoridad de los datos.

En el Módulo de Directorio será necesario autenticarse para configurar, adicionar o modificar los ficheros de configuración del servidor de directorio y esto se realizará a través

del protocolo SSH⁹ el cual establece un canal seguro de comunicación entre la PC gestora donde se encontrará el Módulo de Directorio y el servidor de directorio OpenLDAP garantizándose así la protección de los datos con los que se maneja en la aplicación.

- **Disponibilidad:**

Se garantizará el acceso a la información por los usuarios autorizados y que los dispositivos o mecanismos utilizados para lograr la seguridad no impedirán obtener los datos deseados en un momento dado.

Esto se garantizará en el Módulo de Directorio siempre y cuando el servidor de directorio funcione correctamente y halla red entre la PC gestora y el servidor pues los datos con los que se trabaja se encuentran en el servidor OpenLDAP y el sistema lo que hace es solicitar estos datos de acuerdo a las peticiones del usuario autenticado, dígame clase de objeto, atributos así como los datos de las entradas existentes en el servidor.

2.7 Definición de los casos de uso

2.7.1 Actores del Sistema

Actor	Descripción
Administrador	Es el que puede realizar configuraciones y administrar el servidor de directorio.
Módulo integrador	Es el que realiza la integración de los Módulos de Directorio y DNS.

Tabla 6: Actores del sistema

2.7.2 Diagrama de paquetes

Un diagrama de paquetes muestra cómo un sistema está dividido en agrupaciones lógicas y ellos suministran una descomposición de la jerarquía lógica de un sistema. Para minimizar el acoplamiento entre paquetes es que se decide dividir el Módulo de Directorio en 3 paquetes agrupando los casos de uso por funcionalidades donde: el paquete de autenticación agrupa los casos de uso referentes a la autenticación del sistema, el paquete de entradas que engloba los casos de uso referentes a las entradas realizadas al servidor de directorio y el de servidor que contiene los casos de uso referente a las

⁹ Secure SHell

configuraciones que se realizan en el servidor de directorio.

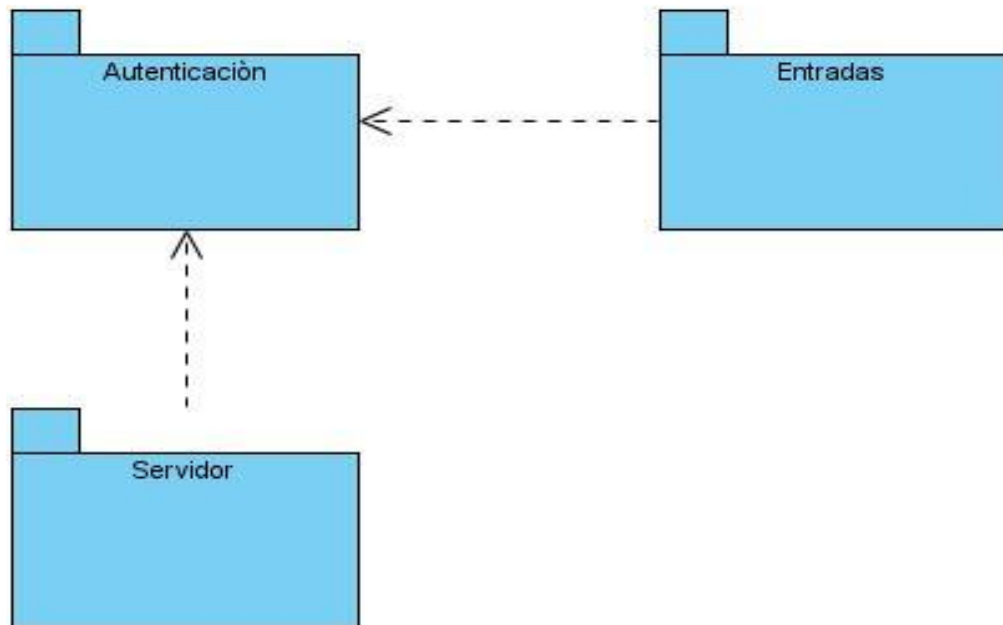


Figura 6: Diagrama de paquetes.

2.7.3 Diagrama de Casos de Uso de Sistema (DCUS)

Un diagrama de casos de uso del sistema representa gráficamente a los procesos y su interacción con los actores. A continuación se muestran los diagramas de casos de uso agrupado por paquetes donde los casos de uso de color azul hacen referencia a los casos de uso implementado en la primera versión del módulo, los violetas a la segunda versión y los de color verde son los casos de uso definidos para este trabajo, que son el resultado de agrupar los requisitos funcionales.

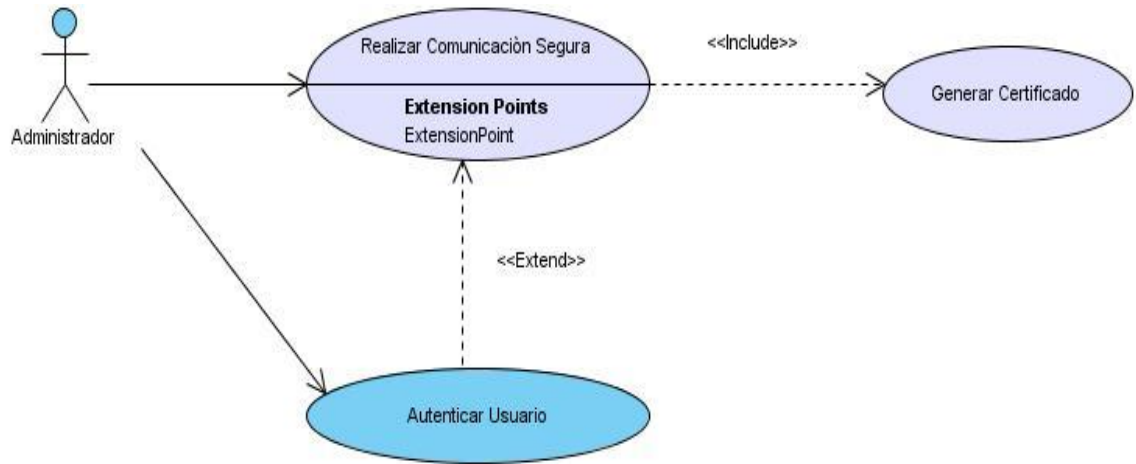


Figura 7: Diagrama de Caso de Uso del Sistema-Paquete Autenticación.

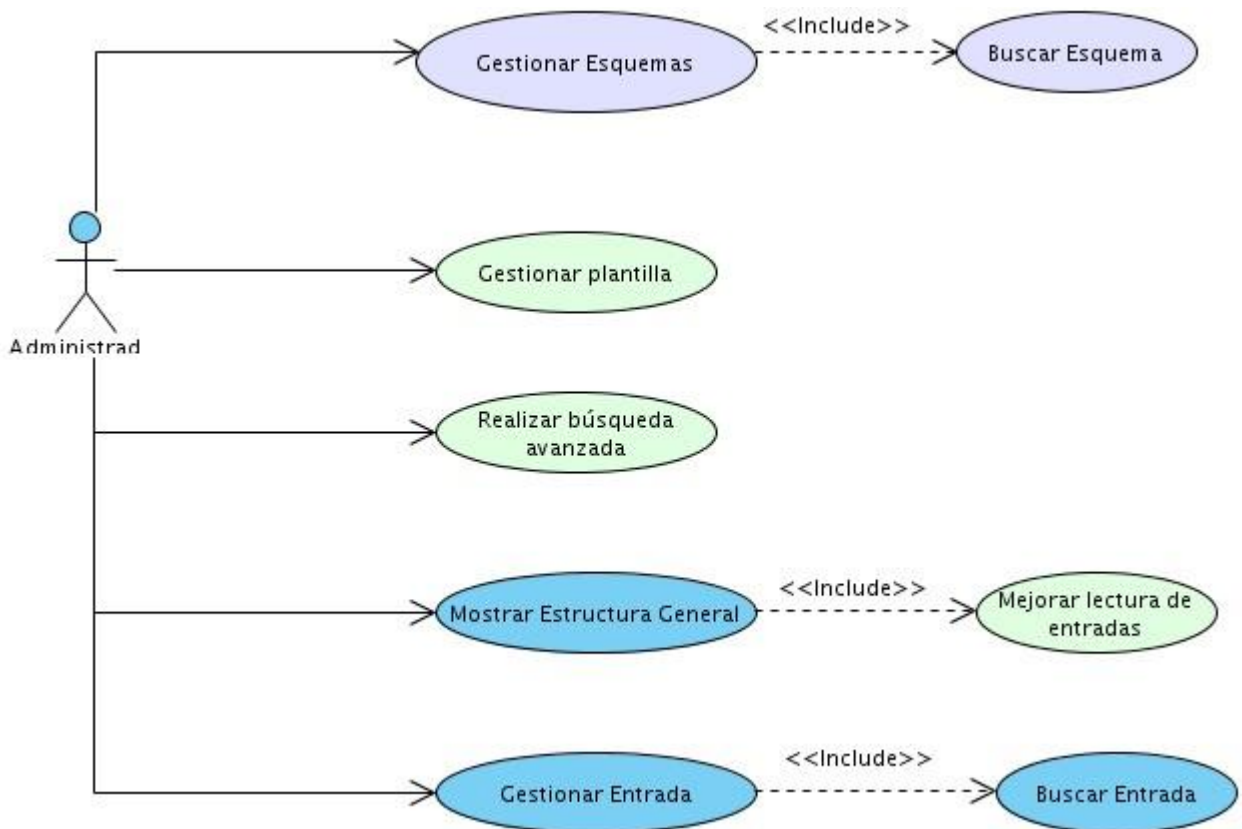


Figura 8: Diagrama de Casos de Uso del Sistema-Paquete Entradas.

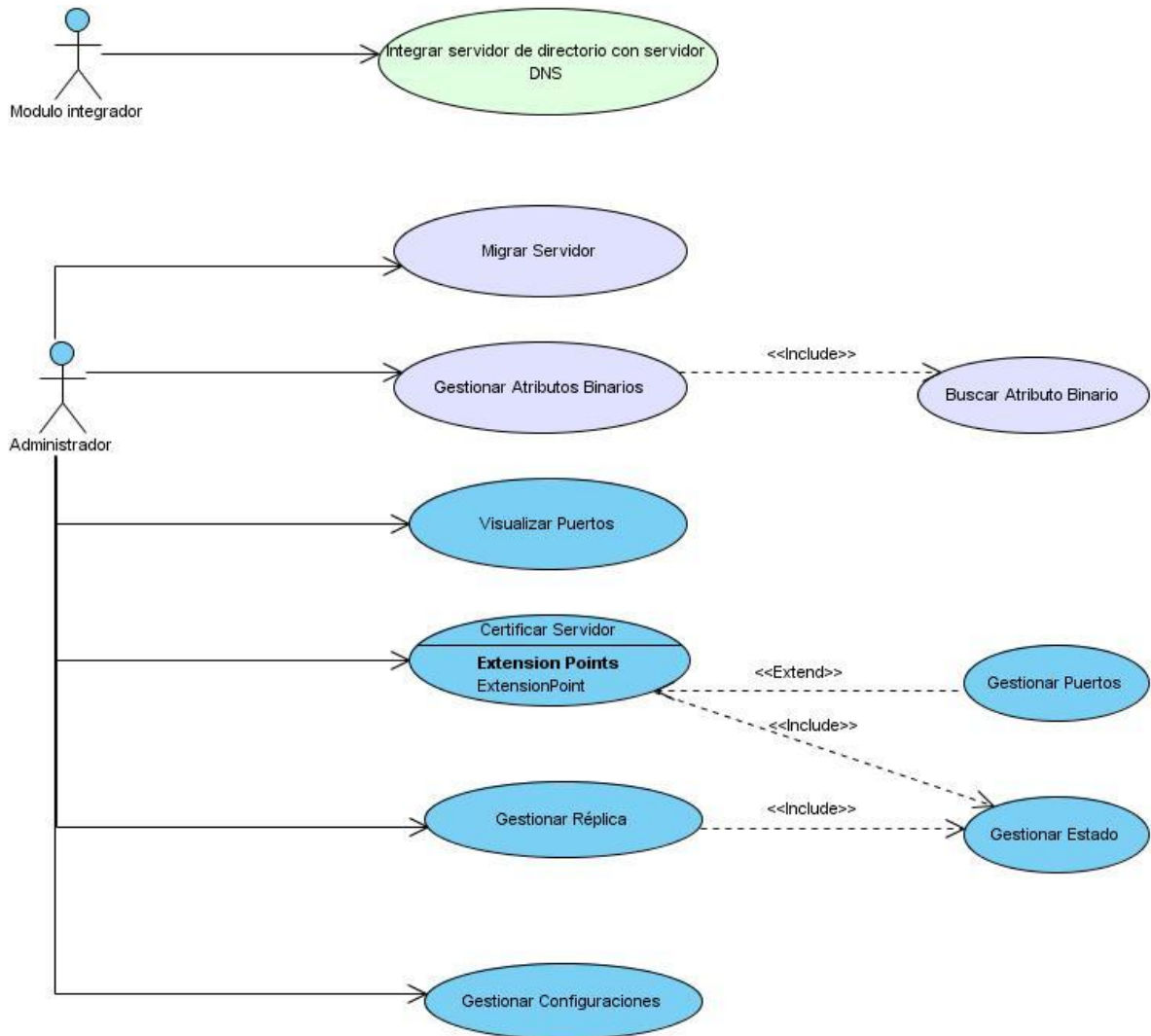


Figura 9: Diagrama de Casos de Uso del Sistema-Paquete Servidor.

2.7.4 Descripción de los CUS y prototipos de interfaz

Para un mejor entendimiento de los casos de uso del sistema definidos, se realiza una descripción general de los siguientes casos de uso.

1. Integrar servidor de directorio con servidor DNS.
2. Gestionar plantilla.
3. Realizar búsqueda avanzada.
4. Mejorar lectura de entradas.

Tabla 7: Descripción CU Configurar servidor de directorio con DNS.

Caso de Uso:	Integrar servidor de directorio con servidor DNS	
Actores:	Módulo integrador	
Resumen:	El caso de uso de inicia cuando el módulo integrador desea integrar el servidor de directorio con un DNS.	
Precondiciones:	El módulo integrador se encuentre autenticado en modo SSH.	
Referencias	RF 1, RF 2	
Prioridad	5	
Flujo Normal de Eventos		
Sección “Configurar servidor de directorio con DNS”		
Acción del Actor	Respuesta del Sistema	
1. Inicia cuando el módulo integrador selecciona la opción de integrar el servidor de DNS con el servidor de directorio.	2. El sistema crea el esquema DNS en la carpeta de esquemas del servidor de directorio.	
Prototipo de Interfaz		
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
	2.1 El sistema muestra un mensaje de error “No se realizó la integración”.	
Prototipo de Interfaz		
No posee		
Poscondiciones		

Tabla 8: Descripción CU Gestionar plantilla.

Caso de Uso:	Gestionar plantilla
Actores:	Administrador
Resumen:	El caso de uso de inicia cuando el administrador desea: definir una plantilla, modificar una plantilla o eliminar una plantilla.
Precondiciones:	El administrador se encuentre autenticado modo SSH.

Referencias	RF 3,RF 4,RF 5, RF 6, RF 7,RF 8	
Prioridad		
Flujo Normal de Eventos		
Sección “Definir plantilla”		
Acción del Actor	Respuesta del Sistema	
1. El administrador selecciona la opción Crear nuevo objeto en el árbol de plantillas.	2. El sistema muestra un formulario con los campos necesarios: <ul style="list-style-type: none"> • Nombre. • Se muestran los Esquemas del servidor de directorio. 	
3. El administrador llena los datos del nombre y selecciona los esquemas que desee y acepta.	4. El sistema crea una nueva plantilla en el servidor de directorio.	
	5. El sistema muestra un mensaje “Nueva plantilla creada correctamente”.	
	6. Se muestra la nueva plantilla en el árbol de plantilla.	
Flujos Alternos		
Acción del Actor	Respuesta del Sistema	
	4.1 El sistema muestra un mensaje “Plantilla existente”.	
3.1 El administrador deja el nombre o no especifica ningún esquema y acepta.	4.2 El sistema muestra un mensaje “Llene todos los campos”.	
Prototipo de Interfaz		

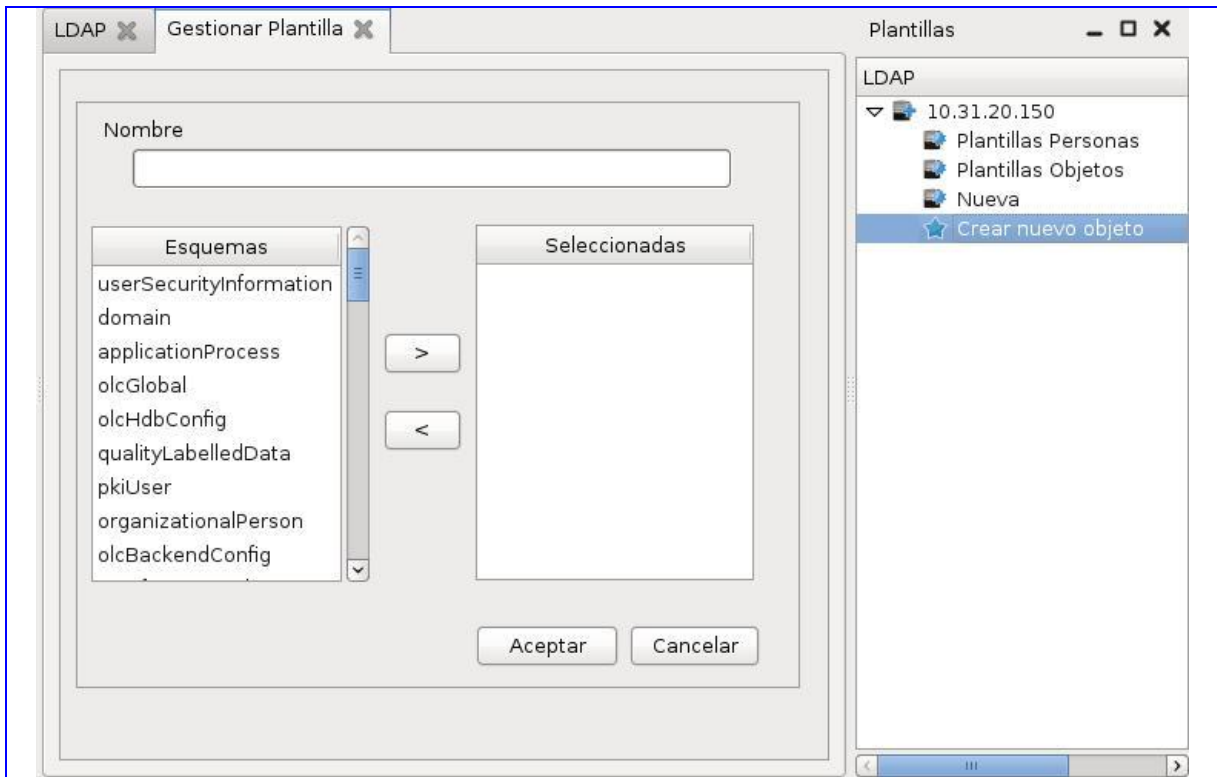


Figura 10: Formulario definir plantilla.

Flujo Normal de Eventos

Sección "Modificar plantilla"

Acción del Actor	Respuesta del Sistema
1. El administrador selecciona una plantilla del árbol de plantilla.	2. El sistema muestra los datos de la plantilla seleccionada.
3. El administrador modifica el nombre o adiciona o elimina esquemas a la plantilla y presiona el botón modificar.	4. El sistema inserta la plantilla modificada en el fichero del servidor de directorio.
	5. El sistema muestra un mensaje "Plantilla modificada correctamente".

Flujos Alternos

Acción del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje "Plantilla

existente”.

4.2 El sistema muestra la plantilla para modificarla o eliminarla.

Prototipo de Interfaz

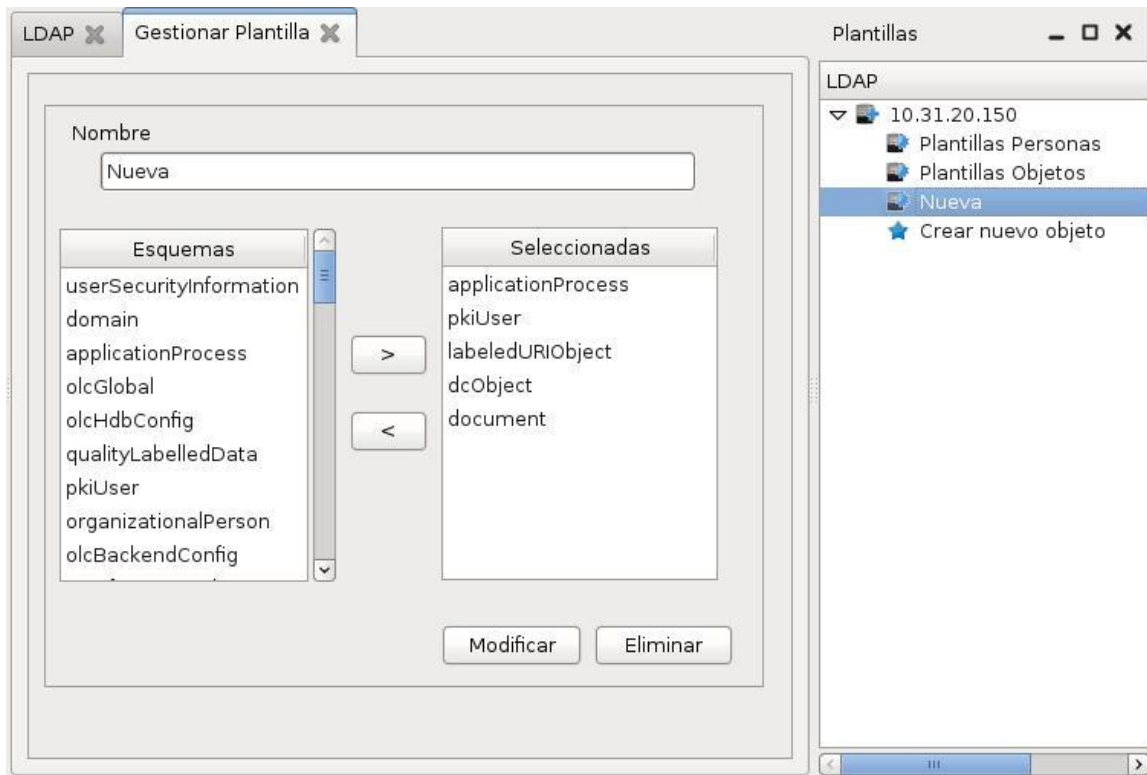


Figura 11: Formulario modificar plantilla.

Flujo Normal de Eventos

Sección “Eliminar plantilla”

Acción del Actor	Respuesta del Sistema
1. El administrador selecciona una plantilla del árbol de plantilla.	2. El sistema muestra los datos de la plantilla seleccionada.
	3. El sistema elimina la plantilla del fichero del servidor de directorio.
	4. El sistema muestra un mensaje “La plantilla se eliminó correctamente”.

Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	4.1 El sistema muestra un mensaje “No se pudo eliminar la plantilla seleccionada”.

Prototipo de Interfaz

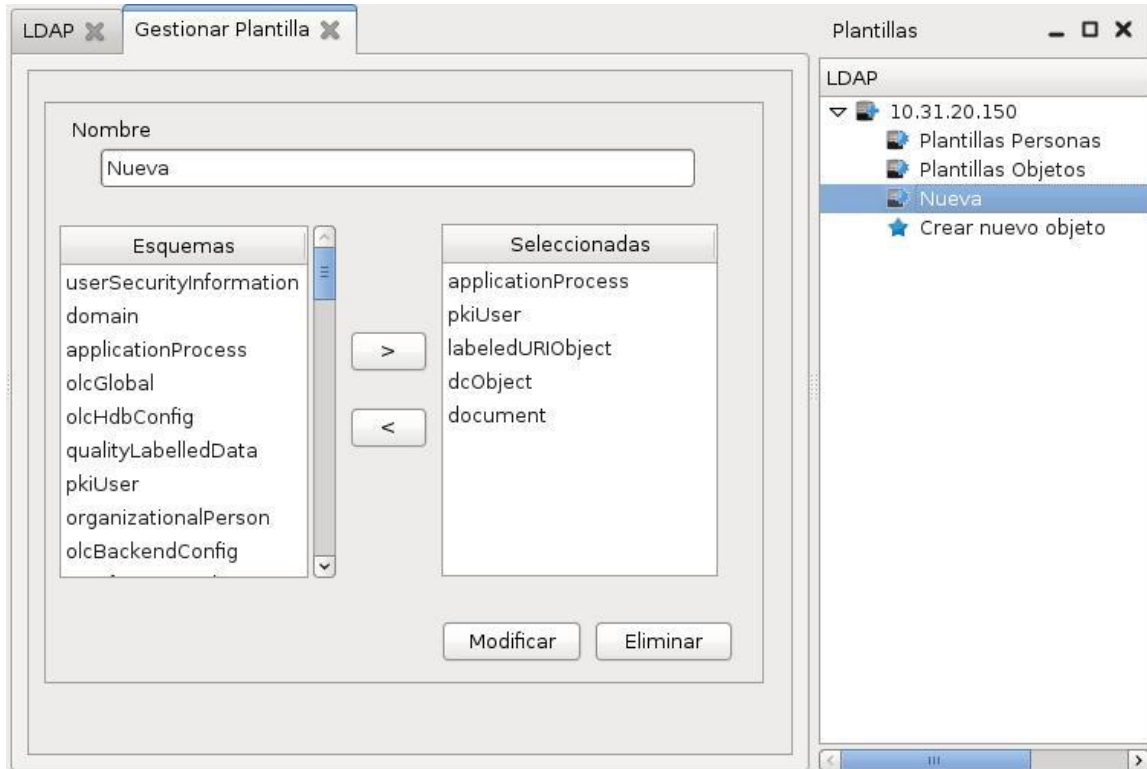


Figura 12: Formulario eliminar plantilla.

Tabla 9: Descripción CU Realizar búsqueda avanzada.

Caso de Uso:	Realizar búsqueda avanzada
Actores:	Administrador
Resumen:	El caso de uso de inicia cuando el administrador desea realizar una búsqueda en el árbol de directorio.

Precondiciones:	Que se encuentre instalado el servidor de directorio.
Referencias	RF 9, RF 10, RF 11
Prioridad	
Flujo Normal de Eventos	
Sección “Realizar búsqueda avanzada”	
Acción del Actor	Respuesta del Sistema
1. El administrador selecciona una rama del árbol y selecciona la opción búsqueda avanzada del menú.	2. El sistema muestra la interfaz para realizar la búsqueda con los siguientes campos: <ul style="list-style-type: none"> • Base. • Nombre. • Atributos. • Nivel de búsqueda. • Filtro.
3. El administrador selecciona el botón editar.	4. El sistema muestra los atributos del servidor de directorio.
5. El administrador selecciona uno o varios atributos.	6. El sistema muestra la interfaz para realizar la búsqueda con los atributos seleccionados.
7. El administrador llena el resto de los campos dígame nombre y filtros.	8. El sistema muestra los resultados de la búsqueda.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
	2.1 El sistema muestra un mensaje “Seleccione una rama del árbol”.
	8.1 El sistema muestra un mensaje “Filtros incorrectos”.
	8.2 El sistema muestra un mensaje “No se encontraron resultado para los criterios definidos”.

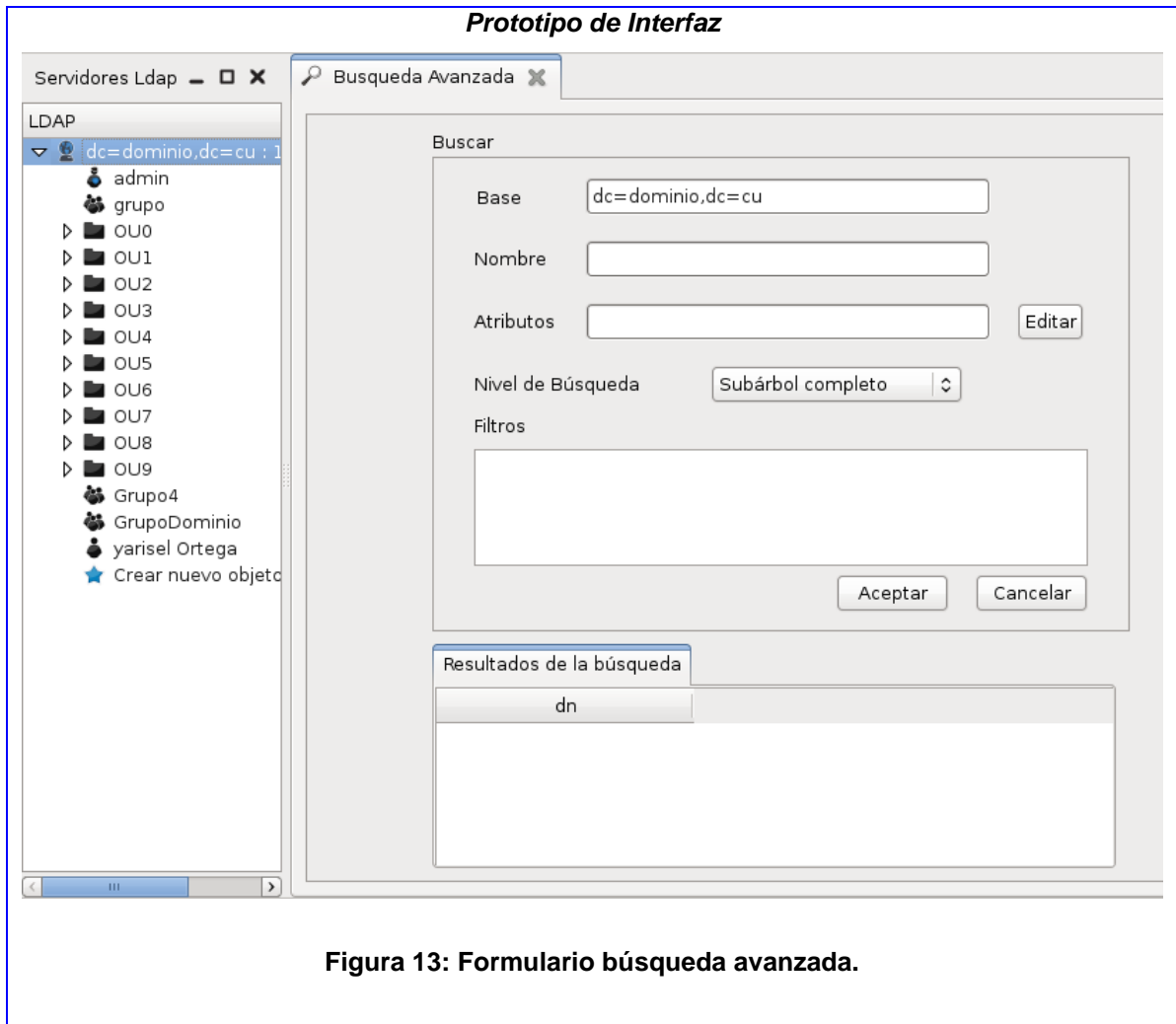


Tabla 10: Descripción CU Buscar en el árbol por niveles.

Caso de Uso:	Buscar en el árbol por niveles
Actores:	Administrador
Resumen:	El caso de uso de inicia cuando el administrador inicia el sistema.
Precondiciones:	Que se encuentre instalado el servidor de directorio.
Referencias	RF 32
Prioridad	
Flujo Normal de Eventos	

Sección “Buscar en el árbol por niveles.”	
Acción del Actor	Respuesta del Sistema
1. El administrador se autentica en el sistema llenando los campos: <ul style="list-style-type: none"> • IP. • Usuario. • Contraseña. • Dominio. 	2. Muestra el panel y carga el árbol por niveles.
Flujos Alternos	
Acción del Actor	Respuesta del Sistema
1.1 El administrador llena incorrectamente uno de los campos requeridos.	2.1 Muestra un mensaje de error: “Credenciales inválidas”.
Prototipo de Interfaz	

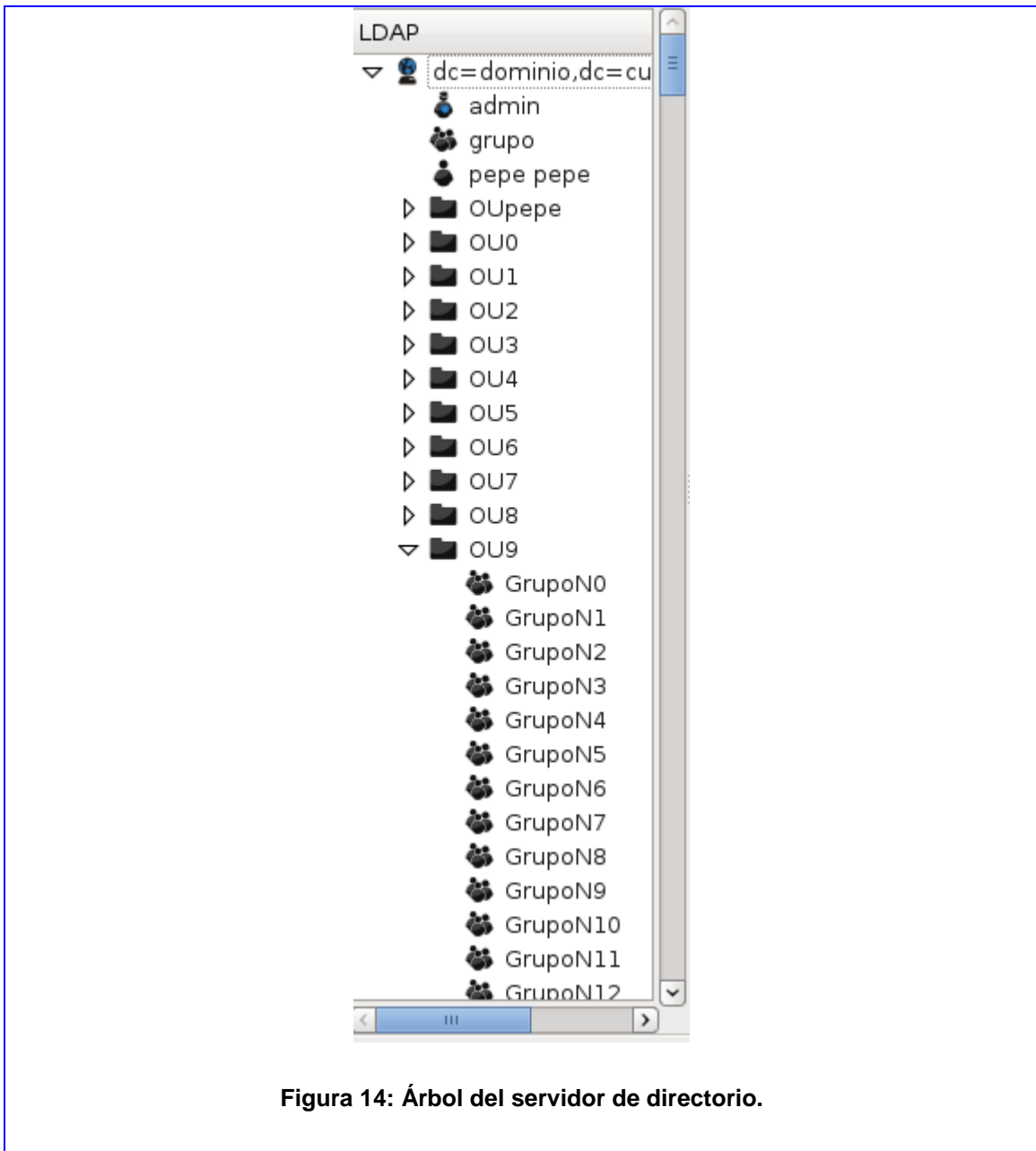


Figura 14: Árbol del servidor de directorio.

2.7 Conclusiones

En este capítulo se mostraron los procesos del negocio definidos que serán automatizados por la herramienta Módulo de Directorio v3.0. Además se identificaron los requisitos funcionales y no funcionales, y a partir de estos se diseñaron y describieron los casos de usos, los cuales guiarán el desarrollo de la solución propuesta.

Capítulo 3: Diseño del sistema

3.1 Introducción

Para dar paso a la implementación es necesario definir un conjunto de elementos entre los que se encuentra la arquitectura y patrones de arquitectura a utilizar, además se realizan los diagramas de clase del diseño y se definen los patrones de diseño, todo esto se describe en el presente capítulo.

3.2 Modelo de diseño

El modelo de diseño es una abstracción de la implementación del sistema. Se utiliza para concebir y documentar el diseño del sistema de software. Además se utiliza como entrada esencial para actividades en implementación y prueba.

La arquitectura que se definió en la aplicación es la arquitectura en capas que entre sus ventajas cuenta con que las capas pueden estar físicamente distribuidas, lo cual quiere decir que los componentes de una capa sólo pueden hacer referencia a componentes en capas inmediatamente inferiores. Esta arquitectura reduce las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores, lo cual es muy cómodo a la hora de realizar modificaciones y permite la reutilización de código.

A continuación se muestra la estructura general del sistema en términos de capas y componentes:

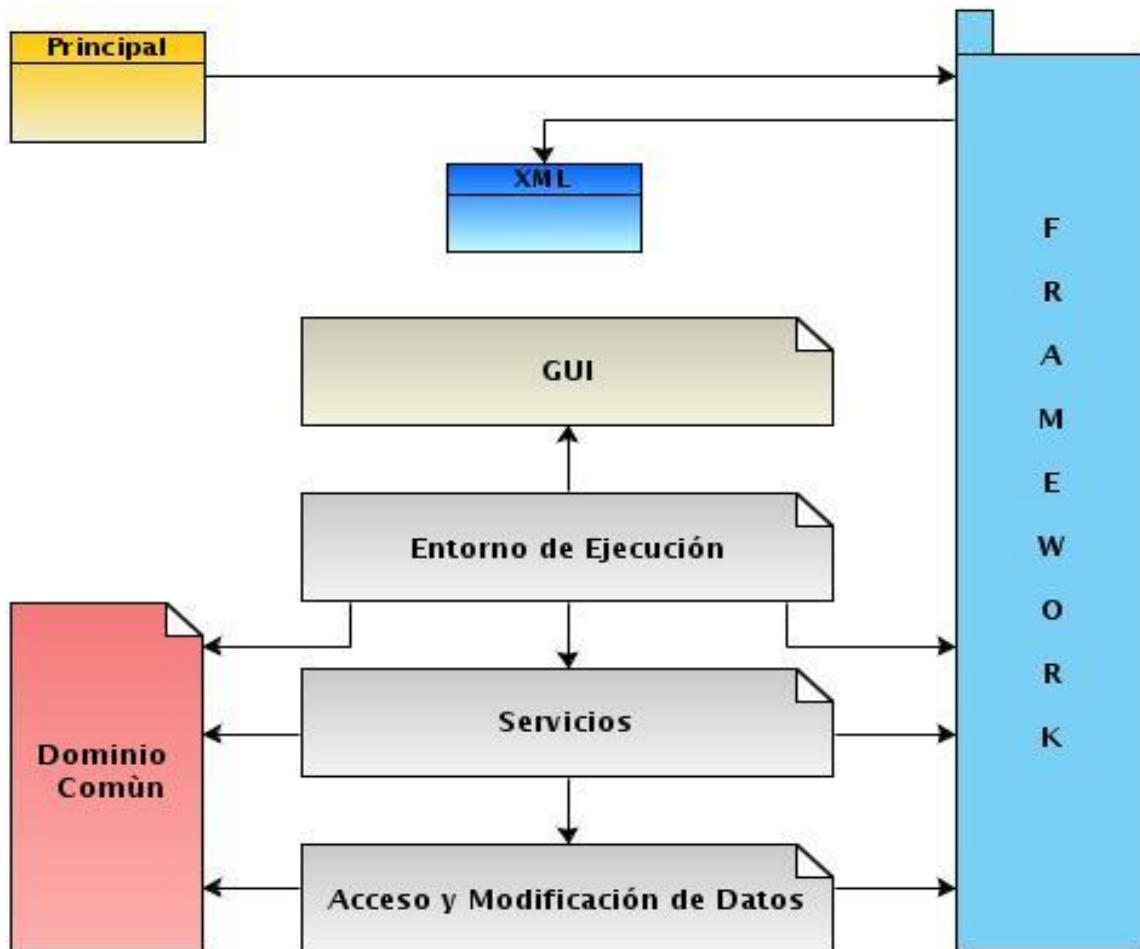


Figura 15: Estructura general del sistema.

El patrón de arquitectura utilizado es el patrón n-capas, que se utilizó además en la implementación de un marco de trabajo o framework que se encarga de resolver problemas como la inicialización del sistema, la ejecución de funciones en paralelo, las conexiones remotas, entre otros aspectos; proporcionándole al desarrollador la posibilidad de concentrarse en aspectos sólo de su negocio.

EL sistema cuenta con 4 capas principales entre las que están: GUI donde estarán los formularios, las interfaces que guían el comportamiento y también los recursos; otra de las capas es entorno de ejecución que es una capa de abstracción entre los formularios y la capa de servicio en esta última es donde se encuentra el negocio de la aplicación, y por último la capa acceso y modificación de los datos que es la encargada como su nombre indica de acceder y modificar los datos en el servidor de directorio ya sea

datos de configuración como datos de entradas al OpenLDAP, se encuentra además el paquete dominio común que contiene clases con información necesaria para más de una capa dígame información del servidor ejemplo los valores IP y credenciales, pues la aplicación soporta más de un servidor autenticado.

También está la clase principal que es la encargada de iniciar el sistema ejecutando el framework que carga a su vez las configuraciones existentes en el componente XML, donde está la configuración referente al sistema, datos del menú y el panel inicial del Módulo de Directorio.

3.2.1 Diagramas de clases del diseño

A continuación se muestran los diagramas de clases del diseño para cada caso de uso:

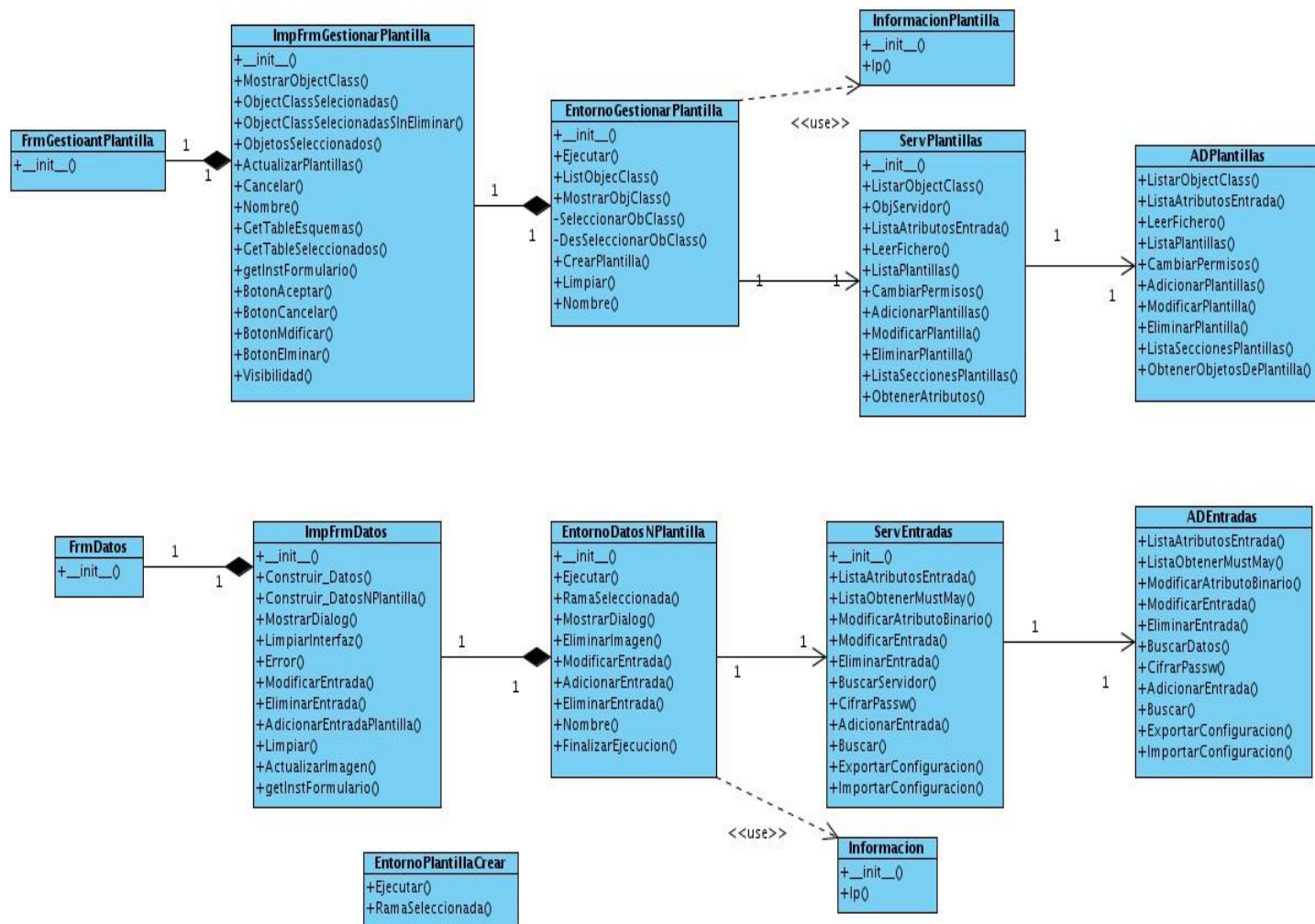


Figura 16: Diagrama de Clases del Diseño: CU Gestionar plantilla.

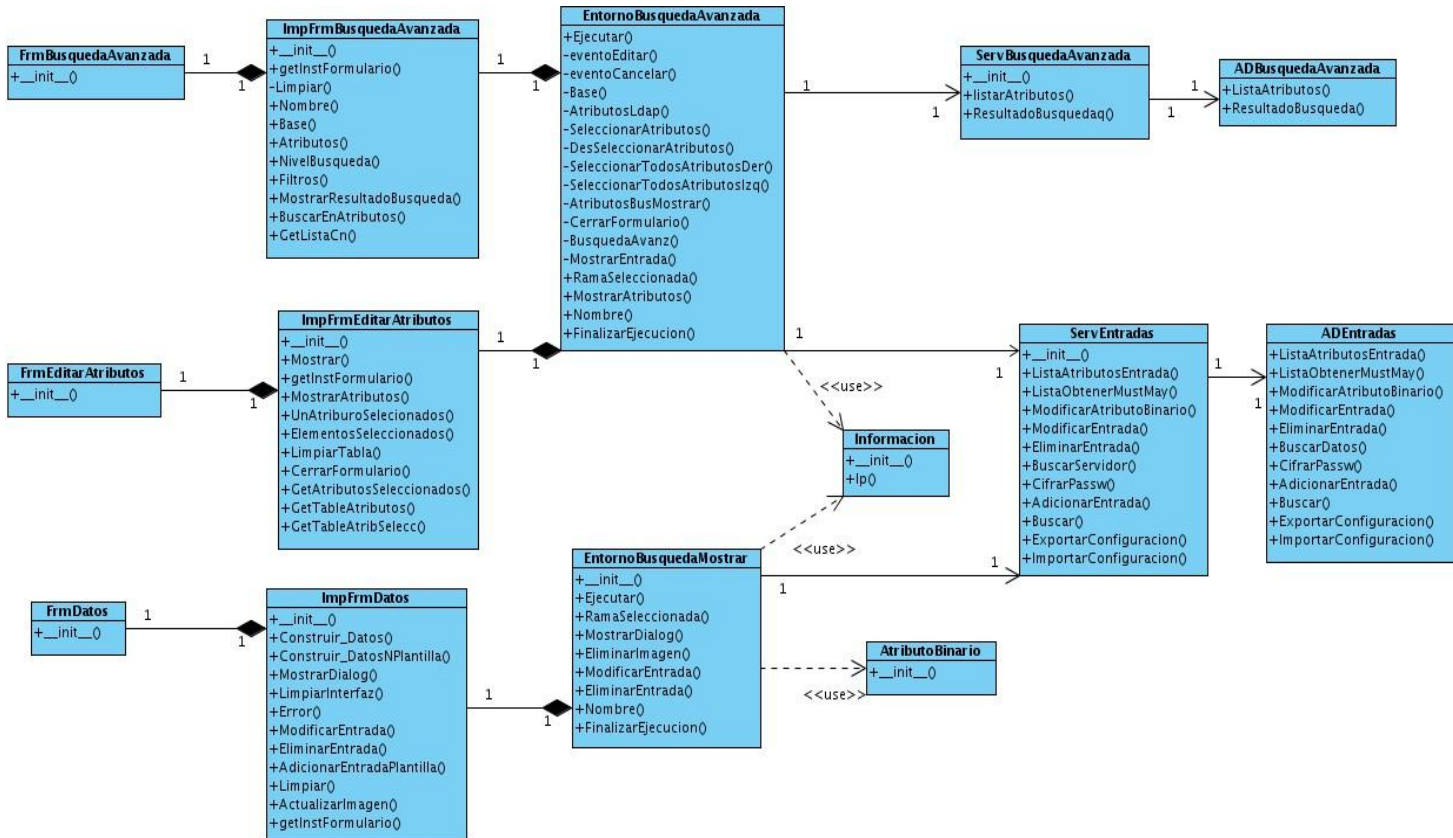


Figura 17: Diagrama de Clases del Diseño: CU Realizar búsqueda avanzada.

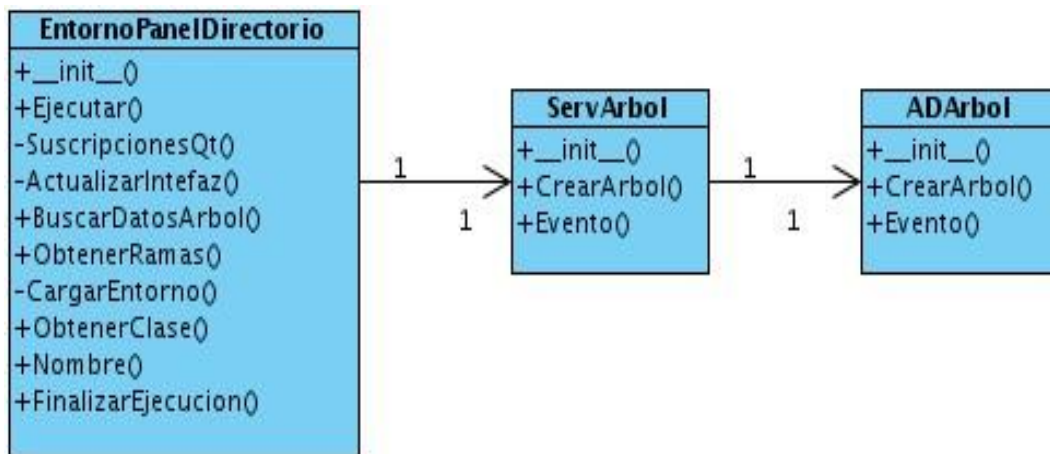


Figura 18: Diagrama de Clases del Diseño: CU Buscar en el árbol por niveles.

3.3 Patrones de diseño

Los patrones de diseño brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. Se debe tener presente los siguientes elementos de un patrón: nombre, el problema, una descripción de cuándo aplicar un patrón, la solución que es una descripción abstracta del problema y las consecuencias: costos y beneficios. Existen varios patrones clasificados en patrones de creación que se utilizan para la inicialización y creación de objetos, los patrones estructurales que se encargan de separar la interfaz de la implementación y se ocupan de cómo las clases y objetos se agrupan, para formar estructuras más grandes. Además están los patrones de comportamiento que describen la comunicación entre los objetos y las clases. (12)

Para el desarrollo de la aplicación se decidió utilizar el patrón de creación Singleton el cual garantiza que una clase tenga una sola instancia y proporciona un punto global a ella, lo cual es necesario para el sistema pues se hacen varias llamadas a una misma clase, además que permite el acceso controlado a esta única instancia.

La implementación se realiza en la clase Aplicación como se muestra a continuación:

```
class Aplicacion( object ):

    __instancia = None
    __componentes = None

    def __new__( cls, *args, **kargs ):
        """Implementacion del patron Singleton"""
        if cls.__instancia is None:
            cls.__instancia = object.__new__( cls, *args, **kargs )
        return cls.__instancia
```

Figura 19: Implementación del patrón Singleton.

Y se usa por ejemplo en la clase principal que es la encargada de mandar a ejecutar el framework que es el que carga las configuraciones del panel y las interfaces del módulo.

```
principal ✖
from Framework.Aplicacion.Aplicacion import Aplicacion

app = Aplicacion()
app.IniciarAplicacion()
```

Figura 20: Uso del patrón Singleton.

También se utiliza el patrón de comportamiento Observer cuyo objetivo es desacoplar la clase de los objetos clientes, aumentando la modularidad del lenguaje, así como evitar bucles de actualización, define una dependencia de uno a muchos entre objetos de forma que cuando un objeto cambia su estado se notifica y actualizan automáticamente todos los objetos, esto es importante en la ejecución de los eventos, dígame eventos QT o eventos que fueron necesarios utilizar para desarrollar casos de uso como el cargar el árbol de directorio y en la gestión de esquemas.

Un ejemplo de cómo se utiliza en el Módulo de Directorio haciendo uso de la clase Eventos del framework donde está implementado el patrón Observer y entre los casos de uso en el que se evidencia su aplicación se encuentra el Gestionar Plantilla donde en la implementación del formulario se lanzan los eventos como se muestra a continuación:

```
class ImpFrmGestionarPlantilla(IFrmGestionarPlantilla, IFormulario):

    def __init__(self, parentFrm=None):
        self.__frm = FrmGestioantPlantilla(parentFrm)
        self.eventoAceptar = Evento()
        self.eventoCancelar = Evento()
        self.eventoMostrar = Evento()
        self.eventoSeleccionar = Evento()
        self.eventoDesSeleccionar = Evento()
        self.eventoModificar = Evento()
        self.eventoEliminar = Evento()
        self.__frm.connect(self.__frm.pushButtonAceptar, QtCore.SIGNAL("clicked()"), self.eventoAceptar)
        self.__frm.connect(self.__frm.pushButtonCancelar, QtCore.SIGNAL("clicked()"), self.eventoCancelar)
        self.__frm.connect(self.__frm.pushButtonDerech, QtCore.SIGNAL("clicked()"), self.eventoSeleccionar)
        self.__frm.connect(self.__frm.pushButtonIzq, QtCore.SIGNAL("clicked()"), self.eventoDesSeleccionar)
        self.__frm.connect(self.__frm.pushButtonModificar, QtCore.SIGNAL("clicked()"), self.eventoModificar)
        self.__frm.connect(self.__frm.pushButtonEliminar, QtCore.SIGNAL("clicked()"), self.eventoEliminar)
```

Figura 21: Envío de eventos en el uso del patrón Observer.

Y en el entorno se subscribe al evento a la espera de que sea lanzado como se muestra a continuación:

```
self.guipantilla.eventoSeleccionar +=self.__SeleccionarObClass
self.guipantilla.eventoDesSeleccionar +=self.__DesSeleccionarObClass
self.guipantilla.eventoAceptar +=self.CrearPlantilla
self.guipantilla.eventoCancelar += self.Limpiar
```

Figura 22: Suscripción de eventos.

Además se decide utilizar el patrón IoC o inversión del control el cual permite que en lugar de que una clase tenga el control sobre la creación de objetos, le pide a otra que lo haga, este es usado en las clases que ejecutan los archivos .xml que se encuentran en el framework utilizado.

Un ejemplo de cómo se usa es que el método Instancia de la clase Aplicación es la que se encarga de crear objetos como se muestra a continuación se crean instancias de las clase ServArbol que en el.xml de configuraciones se especifica con el nombre iservarbol.

```
def Ejecutar( self ):
    self.__app = Aplicacion()
    self.__panel = self.__app.Instancia( "Entorno.Panel.Qt" )
    self.__servicio = self.__app.Instancia("serv.iservarbol")
    self.__serviServ = self.__app.Instancia( "serv.iserv" )
```

Figura 23: Uso del patrón IoC.

3.4 Conclusiones

En el presente capítulo se explicó la arquitectura que se va a usar en el desarrollo del sistema para lograr desacoplar las capas, se mostraron además los diagramas de clases diseño de los casos de uso, y los patrones de diseño utilizados en la implementación de sistema.

Capítulo 4: Implementación y Pruebas

4.1 Introducción

En el presente capítulo se desarrolla el flujo de trabajo correspondiente a la implementación, se modelan los diagramas de componentes por cada caso de uso y el diagrama de paquete y se definen las pruebas que se le van a realizar al sistema para verificar que las funcionalidades estén correctamente implementadas.

4.2 Diagramas de componentes

Los diagramas de componentes se utilizan para modelar la vista estática de un sistema. Muestran la organización y las dependencias entre un conjunto de componentes, donde un componente se corresponde con una o varias clases, interfaces o colaboraciones. Los componentes pueden realizarse por partes y cada diagrama describe un apartado del sistema, el siguiente diagrama está agrupado por caso de uso, y los componentes son las clases referentes a ese caso de uso.

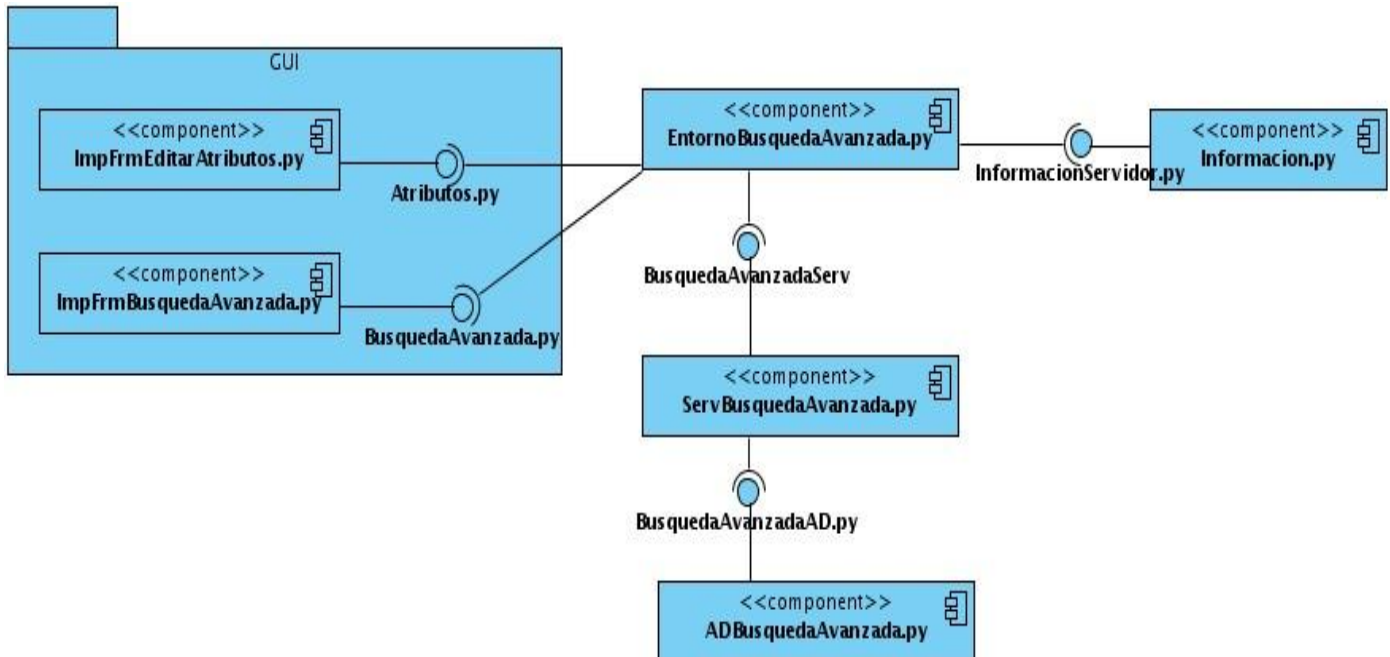


Figura 24: Diagrama de componente: CU Realizar búsqueda avanzadas.

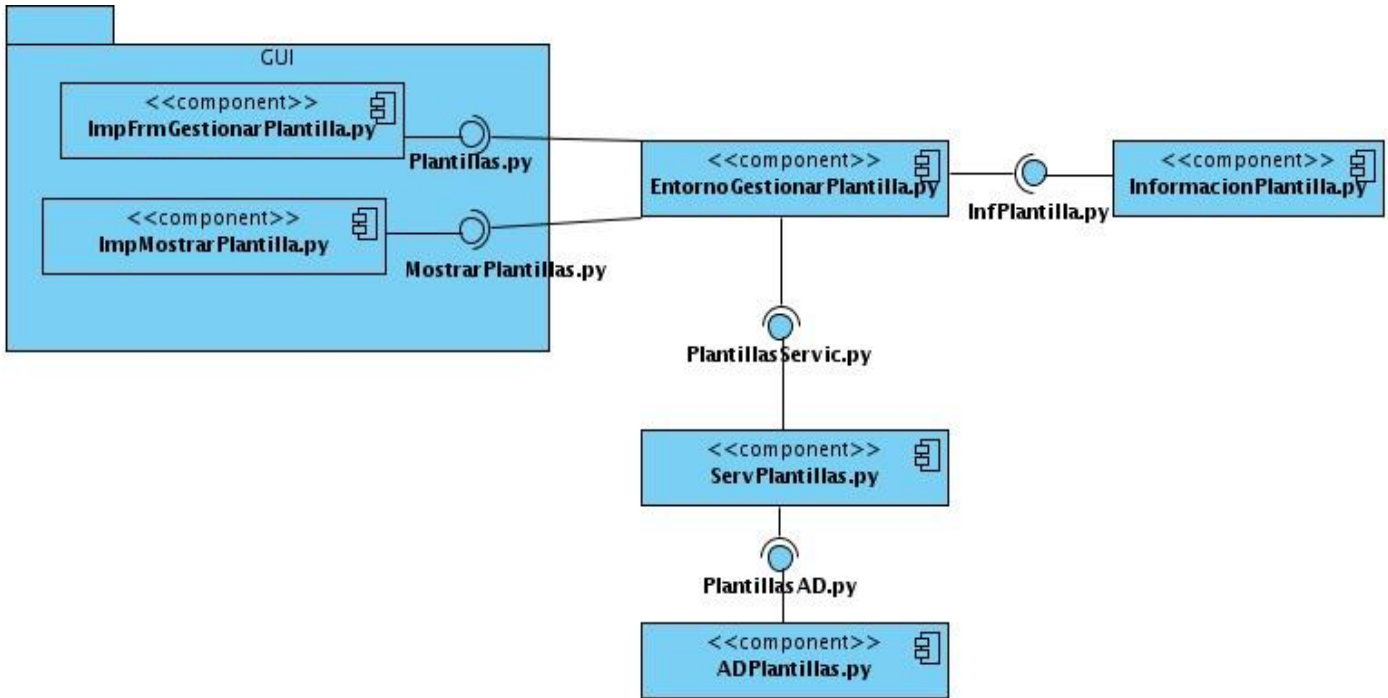


Figura 25: CU Gestionar Plantillas

4.3 Diagrama de despliegue

El diagrama de despliegue indica la situación física de los componentes lógicos desarrollados (4) y muestra los nodos y sus relaciones donde un nodo es un conjunto de componentes.

El siguiente diagrama cuenta con dos nodos uno representa la PC gestora que es donde se encontrará instalado el Módulo de Directorio y el otro representa el servidor de directorio, es decir, donde estará el servidor OpenLDAP instalado en su versión 2.4.23. La interacción entre estos nodos se realiza a través de TCP/IP, sobre el que operarán LDAP y SSH que son los protocolos usados por el sistema para la gestión del servidor de directorio.



Figura 26: Diagrama de despliegue.

4.4 Prueba

Las pruebas consisten en una actividad en la cual un sistema o componente es ejecutado bajo unas condiciones o requerimientos específicos, los resultados son observados y registrados y se realiza una evaluación de algún aspecto del sistema o componente valorando su calidad.

La prueba de software es un elemento crítico para garantizar la calidad del software y representa además una revisión final de las especificaciones del diseño y de la implementación. (7)

Las pruebas tienen dos objetivos fundamentales:

- Demostrar al desarrollador y al cliente que el software satisface sus requerimientos.
- Descubrir defectos en el software en que el comportamiento de este es incorrecto, no deseable o no cumple su especificación.

4.4.1 Niveles de Prueba

La prueba es aplicada para diferentes tipos de objetivos, en diferentes escenarios o niveles de trabajo. Existen varios tipos niveles entre el que se encuentra:

Prueba de integración

Es ejecutada para asegurar que los componentes en el modelo de implementación operen correctamente cuando son combinados para ejecutar un caso de uso. Se prueba un paquete o un conjunto de paquetes del modelo de implementación. Estas pruebas descubren errores en las especificaciones de las interfaces de los paquetes. (7)

4.4.2 Tipos de Prueba

Cada tipo de prueba tiene un objetivo específico y una técnica de soporte. Cada técnica se centra en la prueba de uno o varios atributos o características del destino de la prueba.

De los tipos de prueba existentes se escogió el tipo de prueba de funcionalidad que engloba:

- **Función** que consisten en pruebas fijando su atención en la validación de las funciones, métodos, servicios y caso de uso.

- **Pruebas de Seguridad** asegura que los datos o el sistema solamente es accedido por los actores deseados.

Tipo de prueba Funcionalidad. Función

Tiene como objetivo: Asegurar el trabajo apropiado de los requisitos funcionales, incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados.

Y como metas:

- ✓ Verificar el procesamiento, recuperación e implementación adecuada de las reglas del negocio.
- ✓ Verificar la apropiada aceptación de datos.

Como método de prueba a utilizar se escogió el de Caja Negra, que consiste en 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 las entradas se aceptan de forma adecuada y se produce una salida correcta, y que la integridad de la información externa se mantiene. Esta prueba examina algunos aspectos del modelo fundamentalmente del sistema sin considerar la estructura interna del software.

La elaboración de los casos de prueba se realizó a partir de la técnica de Particiones de Equivalencia. Esta técnica se basa en una evaluación de las clases de equivalencia para una condición de entrada. Una clase de equivalencia representa un conjunto de estados válidos o inválidos para condiciones de entrada. Regularmente, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición lógica. (7)

Se diseñaron casos de prueba como se muestra a continuación:

Caso de Prueba Búsqueda Avanzada

Escenario	Descripción	Base	Nombre	Atributos	Filtro	Respuesta del sistema	Flujo central
EC 1.1 Selecciona opción búsqueda avanzada.	El usuario selecciona la opción búsqueda avanzada para	V	N/A	N/A	N/A	El sistema muestra la interfaz para realizar la búsqueda.	Selecciona rama del árbol/En el menú selecciona la

	realizar una búsqueda.						opción búsqueda avanzada.
EC 1.2	El usuario llena los datos para realizar una búsqueda.	V	V	V	V	Consulta los datos y muestra una lista con los resultados.	Selecciona rama del árbol/búsqueda avanzada.
Introduce los datos y selecciona la opción buscar.		V	N/A	N/A	N/A		
EC 1.3	Se selecciona la opción para realizar búsqueda avanzada sin marcar ninguna rama.	I	V	V	V	Muestra un mensaje de alerta: "Debe seleccionar una rama".	Opción del menú búsqueda avanzada.
EC 1.4	El usuario llena los datos para realizar una búsqueda.	V	I	V	V	Muestra un mensaje de alerta: "Sintaxis incorrecta".	Selecciona rama del árbol/búsqueda avanzada.
Introduce incorrectamente alguno de los datos y selecciona la opción buscar.							

Tabla 11: Caso de prueba Búsqueda avanzada.

Descripción de las variables

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Base	Campo de selección	No	Selección simple
2	Nombre	Campo de texto	Sí	Sólo letras y números
3	Atributos	Campo de selección	Sí	Selección múltiple
4	Filtro	Campo de texto	Sí	Caracteres alfanuméricos y caracteres especiales.

Tabla 12: Descripción de las variables.

Después de realizadas las pruebas se llena la plantilla de No Conformidades que recoge los errores que son detectados durante la revisión de la documentación del sistema. Se elabora un documento por cada revisión que se haga y se controlan a través de versiones según se vayan eliminando los errores, hasta que finalmente se hayan erradicado todos los defectos que posea el elemento que se prueba. Además de estar inmerso en la planilla de diseño de Casos de Prueba, estas No Conformidades se van registrando en un documento aparte para luego enviarlo al equipo de desarrolladores.

En la siguiente tabla se expone un resumen de los resultados de las No Conformidades obtenidas, distinguiendo de ellas cuáles proceden, cuáles no proceden y las que ya fueron resueltas.

Módulo	Total de No Conformidades	# de No Conformidades que proceden	# de No Conformidades que no proceden	# de Conformidades resueltas
--------	---------------------------	------------------------------------	---------------------------------------	------------------------------

Módulo de Directorio	6	6	0	6
----------------------	---	---	---	---

Tabla 13: No Conformidades del proceso de liberación interno.

Tipo de prueba Funcionalidad. Seguridad

Objetivo:

- Nivel de Seguridad del Sistema: Verificar que sólo los actores con acceso al sistema puedan acceder a este.

Garantiza:

- Que sólo aquellos usuarios autorizados a acceder al sistema son capaces de ejecutar las funciones del sistema.

Técnica:

- Identificar cada tipo de usuario y las funciones a las que se debe autorizar.
- Crear pruebas para varios tipos de usuario definidos en el servidor OpenLDAP y verificar cada permiso, autenticando cada tipo de usuario al sistema.
- Modificar tipos de usuarios y volver a ejecutar las pruebas.

Criterio de Completitud:

Para cada tipo de usuario conocido, las funciones, datos apropiados y todas las transacciones funcionan como se esperaba, pues para acceder al sistema sólo se puede realizar con usuarios de privilegio *root* o administrador definidos en el servidor de directorio OpenLDAP, probándose que sólo estos son los que acceden al sistema.

4.6 Conclusiones

En el presente capítulo se obtuvieron como artefactos los diagramas de componentes y el diagrama de despliegue que forman parte de la fase de implementación. Se describieron las pruebas diseñadas y se elaboraron los casos de prueba, aplicándose a los casos de uso, donde todas las no conformidades encontradas fueron solucionadas.

Conclusiones

Los objetivos propuestos para este trabajo fueron plenamente cumplidos a través de las actividades realizadas.

- Se realizó un estudio de las herramientas existentes en el mundo lo que permitió identificar la necesidad de añadir nuevas funcionalidades al Módulo de Directorio v2.0, para facilitar la administración del servidor de directorio OpenLDAP.
- Se describió el negocio a partir de lo cual se especificaron los requisitos funcionales y no funcionales que fueron capturados en casos de uso, obteniéndose el Modelo de Casos de Uso del Sistema.
- Se refinó el diseño del Módulo de Directorio, en particular los Diagramas de Clases del Diseño para cada uno de los casos de uso identificados para esta versión, resultando un Modelo del diseño refinado.
- Se implementaron todas las funcionalidades definidas para el Módulo de Directorio, atendiendo a la arquitectura definida, obteniéndose una aplicación funcional.
- Se validó y verificó la calidad de la solución propuesta con la realización de pruebas, demostrándose que cada una de las funcionalidades responde apropiadamente a los requisitos funcionales descritos.

Recomendaciones

Se recomienda:

- Realizar la migración de políticas de grupo del Directorio Activo de Windows Server 2003 a servidor de directorio OpenLDAP.
- Realizar la gestión de las políticas de grupo.
- Realizar un editor de políticas de grupo en la integración de Samba con OpenLDAP.
- Desarrollar una versión para la web del Módulo de Directorio.

Referencias Bibliográficas

1. OpenLDAP. [Online] <http://www.openldap.org/doc/admin24/intro.html>.
2. **OpenLDAP, Desarrolladores. 2011.** OpenLDAP Software 2.4 Administrator's Guide. 2011.
3. Study Java. Study Java. [Online] <http://studyjava.org/general/ldap-%E2%80%93-lightweight-directory-access-protocol> 2011 .
4. **Junio del 2006.** RFC 4512 Lightweight Directory Access Protocol (LDAP) Directory Information Models. Junio del 2006.
5. **Softerra LDAP administrator.** (2011). Retrieved from <http://www.ldapadministrator.com/>
6. **2011.** OpenDS. [Online] febrero 16, 2011. <https://www.openss.org/wiki/page/OpenDSSystemRequirements>.
7. **Jacobson, I. Booch, G. y Rumbaugh, J. 2000.** El Proceso Unificado de Desarrollo de software. 2000.
8. **Presman. 2000.** Ingeniería de Software un enfoque Práctico. 2000.
9. **Ing. Yisel Alonso Riverón, Ing. Yaneisy Cruz Navarro y Ing. Yordanis Tornés Medina. 2008.** IDEF. Una alternativa para modelamiento de negocio con RUP . [Online] 03 10, 2008. <http://www.gestiopolis.com/administracion-estrategia/rup-tecnologia-aplicada-al-modelo-de-negocios.htm>.
10. **2005.** Rodríguez, A., Fernández, E., & Piattini, M. Hacia la obtención de clases de análisis y casos de uso desde modelos de procesos del negocio. [Online] 2005. http://www.eici.ucm.cl/Academicos/R_Villarroel/descargas/calidad.produccion/Clases.de.Analisis.d desde.modelos.proc.negocios.pdf.
11. Visual Paradigm. Visual Paradigm. [Online] <http://www.visual-paradigm.com/>.
12. **Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. 1994.** Design Patterns.Elements of Reusable Object-Oriented Software. 1994.
13. **Julio 1993.** RFC 1487 X.500 Lightweight Directory Access Protocol. Julio 1993.
14. CubaSi. (2006, 6 15). Retrieved 2011, from <http://www.cubasi.cu/desktopdefault.aspx?spk=160&clk=120217&lk=1&ck=62573&spka=35>

Bibliografía

- 2003.** [Online] enero 15, 2003. http://www.rhernando.net/modules/tutorials/doc/redes/serv_dir.html.
- Aranzadi. Glosarios de Término. [Online] <http://www.aranzadi.es/index.php/informacion-juridica/informacion-interes/glosario-de-terminos-sobre-internet-y-spam>.
- Brittain, Jason and Darwin, Ian F.** Python for Unix and Linux System Administration. s.l. : O'Reilly Media, Inc.
- Duque, Raúl González.** Python para todos.
- 2010.** easyeclipse.org. [Online] 2010. <http://www.easyeclipse.org/site/home/>.
- Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides. 1994.** Design Patterns.Elements of Reusable Object-Oriented Software. 1994.
- Hodges, J. and R. Morgan. 2002.** RFC3377 "Lightweight Directory Access Protocol (v3): Technical Specification". 2002.
- IDEF0. [Online] <http://www.idef.com/IDEF0.htm>.
- Ing. Yisel Alonso Riverón, Ing. Yaneisy Cruz Navarro y Ing. Yordanis Tornés Medina. 2008.** IDEF. Una alternativa para modelamiento de negocio con RUP . [Online] 03 10, 2008. <http://www.gestiopolis.com/administracion-estrategia/rup-tecnologia-aplicada-al-modelo-de-negocios.htm>.
- Jacobson, I. Booch, G. y Rumbaugh, J. 2000.** El Proceso Unificado de Desarrollo de software. 2000.
- 2011.** OpenDS. [Online] febrero 16, 2011. <https://www.opensds.org/wiki/page/OpenDSSystemRequirements>.
- OpenLDAP. [Online] <http://www.openldap.org/doc/admin24/intro.html>.
- OpenLDAP, Desarrolladores. 2011.** OpenLDAP Software 2.4 Administrator's Guide. 2011.
- Presman. 2000.** Ingeniería de Software un enfoque Práctico. 2000.
- Python. [Online] <http://www.python.org/doc/>.
- 2010.** Red Hat. [Online] 2010. <http://www.es.redhat.com/>.
- Julio 1993.** RFC 1487 X.500 Lightweight Directory Access Protocol. Julio 1993.
- Junio del 2006.** RFC 4512 Lightweight Directory Access Protocol (LDAP) Directory Information Models. Junio del 2006.
- 2005.** Rodríguez, A., Fernández, E., & Piattini, M. Hacia la obtención de clases de análisis y casos de uso desde modelos de procesos del negocio. [Online] 2005.

http://www.eici.ucm.cl/Academicos/R_Villarroel/descargas/calidad.produccion/Clases.de.Analisis.desde.modelos.proc.negocios.pdf.

Rodríguez, Medardo. [Online] <http://www.mastermagazine.info/articulo/12671.php>.

Study Java. Study Java. [Online] <http://studyjava.org/general/ldap-%E2%80%93-lightweight-directory-access-protocol> .

2011. Tecnohackers. [Online] marzo 3, 2011. <http://www.tecnohackers.net/redes/protocolo-ldap/> .

Visual Paradigm. Visual Paradigm. [Online] <http://www.visual-paradigm.com/>.

Glosario de Términos

BER: Reglas de Codificación Básicas son las reglas definidas originalmente en el estándar ASN.1 para codificar información abstracta en un flujo de bits único, esto es, que pueda ser interpretado en cualquier máquina de la misma manera.

DAP: Directory Access Protocol es un estándar dentro de las redes de ordenadores que ha sido promulgado por la ITU-T y por la ISO para el acceso de un servicio de directorio X.500.

DHCP: Protocolo de configuración de host dinámico es un protocolo que permite configurar dinámicamente los parámetros de red necesarios para que un sistema pueda comunicarse.

DNS: Domain Name System es una base de datos distribuida, con información que se usa para traducir los nombres de dominio, fáciles de recordar y usar por las personas, en números de IP que es la forma en la que las máquinas pueden encontrarse en Internet.

GPL: GNU General Public License, licencia creada por la Free Software Foundation a mediados de los años 80, orientada principalmente a proteger la libre distribución, modificación y uso del software. Su propósito es declarar que el software cubierto por esta licencia es Software Libre.

LDIF: LDAP Data Interchange Format, define el formato estándar para el intercambio de datos en texto plano de un servidor que implementa el protocolo LDAP.

OMG: Object Management Group en español Grupo de Gestión de Objetos es un consorcio dedicado al cuidado y el establecimiento de diversos estándares de tecnologías orientadas a objetos, tales como UML, XMI, CORBA. Es una organización sin ánimo de lucro que promueve el uso de tecnología orientada a objetos mediante guías y especificaciones para las mismas.

OSI: Open System Interconnection es un modelo de interconexión de sistemas abierto creado por la Organización Internacional para la Estandarización y es marco de referencia para la definición de arquitecturas de interconexión de sistemas de comunicaciones.

PHP: Hypertext Pre-processor es un lenguaje de programación interpretado generalmente usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web.

Plugins: Aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.

Qt: Biblioteca multiplataforma para desarrollar interfaces gráficas de usuario desarrollada por la compañía noruega Trolltech.

Replicación multimaster: es un método de replicación de base de datos donde se habilitan los datos para ser distribuidos en un grupo de computadoras, y actualizado por cualquier miembro del grupo.

Release: Refiere a un producto final, preparado para lanzarse como versión definitiva a menos que aparezcan errores que lo impidan.

SASL: Simple Authentication and Security Layer es un framework para autenticación y autorización en protocolos de internet. Separa los mecanismos de autenticación de los protocolos de la aplicación permitiendo, en teoría, a cualquier protocolo de aplicación que use SASL usar cualquier mecanismo de autenticación soportado por SASL.

SSH: Secure SHell, en español intérprete de órdenes seguro, es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red.

SSL: Secure Sockets Layer o Protocolo de Capa de Conexión Segura, proporciona autenticación y privacidad de la información mediante comunicaciones seguras por una red, comúnmente Internet con el uso de criptografía.

TCP/IP: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP) representan los dos protocolos más importantes que componen Internet. TCP/IP es la base de Internet, y sirve para enlazar computadoras que utilizan diferentes sistemas operativos.

UIT-T: Sector de Normalización de las Telecomunicaciones de la UIT (UIT-T por sus siglas en inglés) es el órgano permanente de la Unión Internacional de Telecomunicaciones (UIT) que estudia los aspectos técnicos, de explotación y tarifarios, y publica normativa sobre los mismos, con vista a la normalización de las telecomunicaciones a nivel mundial.

X.500: Es un conjunto de estándares de redes de ordenadores de la ITU-T sobre servicios de directorio.