



Universidad de las Ciencias Informáticas
Facultad 2

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

Plataforma de Gestión de Servicios Telemáticos en GNU/Linux.
Módulo de Directorio v2.0.

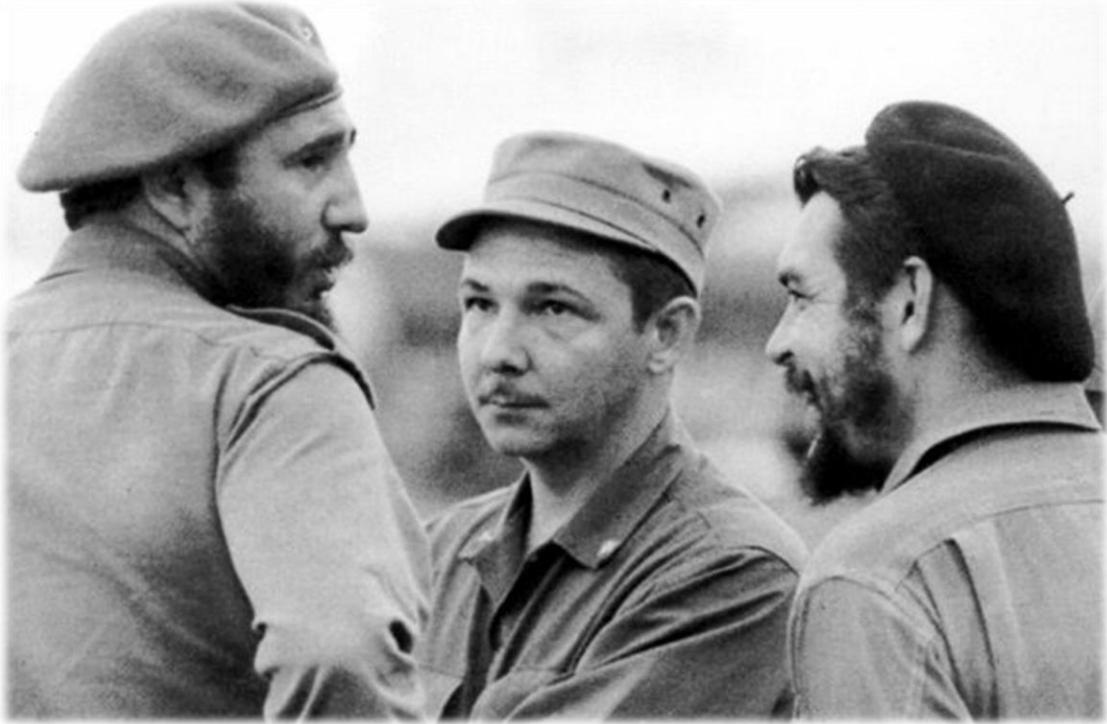
Autores:

Jenny De la Rosa Pasteur
Edgardo Zulueta Sarria

Tutor:

Ing. Sandy Noa Cabrera

Ciudad de La Habana, junio de 2010.
"Año 52 de la Revolución"



“El futuro de nuestra patria tiene que ser necesariamente un futuro de hombres de ciencia, tiene que ser un futuro de hombres de pensamiento...”

Comandante Fidel Castro Ruz

15 de enero de 1960



Declaración de Autoría

Declaramos que _____ y _____ somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) y a la Facultad 2 para que hagan el uso que estimen pertinente con dicho trabajo.

Para que así conste firmamos la presente a los ____ días del mes de junio de 2010

Firma del Autor
Jenny De la Rosa Pasteur

Firma del Autor
Edgardo Zulueta Sarria

Firma del Tutor
Ing. Sandy Noa Cabrera



Agradecimientos

Jenny

A mis abuelos Olga, Eroenio y Esther por alumbrar mi camino.

A mi madre, mi padre y mis hermanas Ericka, Amanda y Elis Andrea por darme tanta fuerza, valor y seguridad para llegar hasta aquí.

A mis tíos Carlín, Luisín y mi tía Danay por su apoyo incondicional en todo momento.

A Anglada por ser líder de proyecto, consejero, compañero y amigo incondicional.

A mi tutor Sandy, por enseñarme a ser más exigente conmigo misma.

A mi novio Yordany por ser mi guía, y compañero de la vida.

A Carmen Leisa por su amistad incondicional en los buenos y malos momentos.

A mis amigos y profesores del IPVCE: “Félix Varela” y de la Universidad.

A los tecleros y a Guillermo Cabrera.

A todas las personas que de una forma u otra me ayudaron.

Edgardo

A mi mamita linda Isabel, por haberme brindado todo su amor y comprensión.

A mi hermano, por ser siempre un ejemplo para mí.

A toda mi familia, por estar presente cada vez que los necesite.

A mi novia Yeni, por tanto amor, pasión y cariño.

Al proyecto Servicios Telemáticos y en especial a su líder Anglada, por convertirme en el profesional que soy hoy.

A todas mis amistades por permitirme compartir con ustedes durante todo este tiempo.



Dedicatoria

Jenny

A la Revolución, a Fidel y a Raúl. A mi familia, a mis hermanas.

Edgardo

A mi mamá Isabel.



Resumen

El presente trabajo tiene como objetivo añadir nuevas funcionalidades al Módulo de Directorio v1.0 desarrollado en el curso académico 2008-2009 en la Universidad de las Ciencias Informáticas (UCI), para mejorar la gestión de un servidor de directorio OpenLDAP. La herramienta Módulo de Directorio v1.0 es una aplicación de escritorio cómoda que permite administrar un servidor OpenLDAP de forma local y remota, y además cuenta con disímiles funcionalidades como: gestión de entradas, visualización de los puertos habilitados para brindar el servicio, gestión de archivos LDIF, conexiones cifradas, réplica de contenido, así como mostrar la estructura general del servidor.

Sin embargo, para la gestión más eficiente del servidor OpenLDAP se le puede adicionar a dicha herramienta una serie de funcionalidades tales como: gestión de esquemas, que permita la personalización de la información contenida en el servidor; gestión de atributos binarios, principalmente las imágenes asociadas a las entradas; realización de comunicación segura con certificados generados garantizando de esta manera la comunicación de forma segura entre el servidor y la herramienta. Otra funcionalidad que se le puede adicionar y resulta de gran importancia es la migración de un Active Directory de Windows Server hacia OpenLDAP aportando así una solución para la migración del servicio de directorio de un modo muy fácil hacia la plataforma del Software Libre, apoyando así la migración progresiva dispuesta por el Ministerio de la Informática y las Comunicaciones (MIC) desde abril del 2004.



Índice

Introducción 1

Capítulo 1: “Fundamentación teórica” 6

 1.1 Introducción 6

 1.2 Servicio de directorio 6

 1.3 LDAP 6

 1.4 Servidores de directorio 6

 1.5 Sistemas para la gestión de los servidores de red 9

 1.6 Lenguaje de modelado 10

 1.7 Metodología de desarrollo..... 11

 1.8 Modelación de funciones mediante IDEF0..... 15

 1.9 Lenguaje de programación y herramientas 17

 1.9.1 Python 18

 1.9.2 Herramienta Case..... 19

 1.9.2.1 Visual Paradigm 19

 1.9.3 Entorno de desarrollo integrado..... 19

 1.9.3.1 EasyEclipse 20

 1.9.4 Interfaces gráficas de usuario 20

 1.9.4.1 Qt Designer 20

 1.9.4.2 Librería Qt 21

 1.10 Conclusiones 23

Capítulo 2 “Características del sistema” 24

 2.2 Problema y situación problemática 24

 2.3 Información que se maneja..... 25

 2.4 Solución propuesta 25

 2.5 Modelo del Negocio 26

 2.6 Especificación de los requisitos de software 31

 2.6.1 Requisitos funcionales..... 31



2.6.2	Requerimientos no funcionales:.....	32
2.7	Modelo de Casos de Uso del Sistema	36
2.7.1	Actores del Sistema.....	36
2.7.2	Diagrama de paquetes.....	37
2.7.3	Diagrama de Casos de Uso del Sistema (DCUS)	37
2.7.4	Descripción de los CUS y prototipos de interfaces.....	40
2.8	Conclusiones	49
Capítulo 3	“Diseño del sistema”	50
3.1	Introducción	50
3.2	Modelo de Diseño	50
3.2.1	Diagrama de Clases del Diseño.....	53
3.3	Conclusiones	55
Capítulo 4	“Implementación”	56
4.1	Introducción	56
4.2	Diagrama de despliegue	56
4.3	Diagrama de componentes	56
4.4	Conclusiones	63
Conclusiones	64
Recomendaciones	65
Referencias Bibliográficas.....		66
Bibliografía.....		67
Anexos.....		69
Anexo 1: Esquemas del servidor de directorio OpenLDAP		69
Glosario de términos.....		90



Índice de Figuras

Figura 1: RUP en dos dimensiones..... 14

Figura 2: Componentes básicos de IDEF0..... 16

Figura 3: Jerarquía de procesos en IDEF0..... 17

Figura 4: Representación del proceso general “Administrar servidor de Directorio”26

Figura 5: Detalles del proceso general.....27

Figura 6: Detalles del proceso Redefinir esquemas29

Figura 7: Detalles del proceso Establecer conexión segura30

Figura 8: Diagrama de paquetes37

Figura 9: DCUS del paquete Autenticación38

Figura 10: DCUS del paquete Entradas38

Figura 11: DCUS del paquete Servidor39

Figura 12: Interfaz del CU Gestionar esquemas (objectclass).....42

Figura 13: Interfaz del CU Gestionar esquemas (attributetype).....43

Figura 14: Interfaz del CU Migrar servidor.....44

Figura 15: Interfaz del CU Gestionar atributos binarios46

Figura 16: Interfaz del CU Establecer comunicación segura47

Figura 17: Interfaz del CU Generar certificado49

Figura 18: Estructura general del sistema.....51

Figura 19: Estructura detallada del sistema52

Figura 20: Definición de los términos fundamentales53

Figura 21: Diagrama de Clases del Diseño: CU Gestionar Esquemas53

Figura 22: Diagrama de Clases del Diseño: CU Buscar Esquema54

Figura 23: Diagrama de Clases del Diseño: CU Migrar servidor54

Figura 24: Diagrama de Clases del Diseño: CU Gestionar atributos binarios.....54

Figura 25: Diagrama de Clases del Diseño: CU Buscar atributo binario.....55

Figura 26: Diagrama de Clases del Diseño: CU Generar certificado55

Figura 27: Diagrama de Clases del Diseño: CU Establecer comunicación segura55



Figura 28: Diagrama de Despliegue del sistema	56
Figura 29: Diagrama General de Componentes	58
Figura 30: Diagrama de Componentes: CU Migrar servidor	59
Figura 31: Diagrama de Componentes: CU Gestionar atributos binarios	60
Figura 32: Diagrama de Componentes: CU Gestionar esquemas	61
Figura 33: Diagrama de Componentes: CU Generar certificado	62
Figura 34: Diagrama de Componentes: CU Establecer comunicación segura	63



Índice de Tablas

Tabla 1: Actor del sistema.....	37
Tabla 2: Descripción CU Gestionar esquemas.....	41
Tabla 3: Descripción CU Migrar servidor.....	44
Tabla 4: Descripción CU Gestionar atributos binarios	45
Tabla 5: Descripción CU Establecer comunicación segura	47
Tabla 6: Descripción CU Buscar esquema.....	48
Tabla 7: Descripción CU Buscar atributo binario	48
Tabla 8: Descripción CU Generar certificado	48



Introducción

El gran número de descubrimientos científicos y avances tecnológicos del siglo XX han marcado la historia de la humanidad y la vida de los seres humanos. Sin embargo, es asombroso con la gran facilidad y rapidez que la sociedad los asimiló, acostumbrándose a estos cambios como si fueran algo natural y de rutina. Las computadoras y los softwares constituyen aspectos representativos de estos grandes avances que hoy forman parte de la vida cotidiana. Mucho se diferencian las primeras computadoras que utilizaban bulbos para procesar información, de gran tamaño y sumamente costosas, de las computadoras de hoy en día con cientos de microprocesadores vectoriales trabajando al mismo tiempo, grandes velocidades de procesamiento, más pequeñas y de menos costo, siendo así accesible por mayor cantidad de personas. De esta forma poco a poco se convirtieron en una herramienta importante para la comunicación, solución de problemas, procesamiento de información y su almacenamiento.

La tecnología, que representa un conjunto de teorías y de técnicas que permiten el aprovechamiento práctico del conocimiento científico, también se puso en función de la comunicación de las computadoras. Así para la década del 60 comienzan las redes con el establecimiento de la conmutación de paquetes, como método para fragmentar mensajes en partes llamadas *paquetes*, encaminarlos hacia su destino, y ensamblarlos una vez llegados allí. La primera red experimental de este tipo fue en Reino Unido, en los National Physics Laboratories; también en Francia se realizó otro experimento similar. En el año 1967 ya la ARPA o Agencia de Proyectos Avanzados de Investigación para la Defensa (agencia del Departamento de Defensa de los Estados Unidos) contaba con un plan completo para una nueva red donde no hubiera ningún punto central de mando y control, sino que, aunque cualquier punto de la red fuera destruido, podría ser restituida la comunicación encaminándola por otra ruta. La primera conexión de esta red que llamaron ARPANET, fue en la Universidad de California, Los Ángeles (UCLA), y posteriormente se instalaron otros en Santa Barbara, Stanford y Utah. Posteriormente en el año 1972 se enfatiza en la necesidad de ponerse de acuerdo en cuanto a protocolos, que es la forma de estar codificada la información y cómo se envolvía en los paquetes. De esta forma surgieron las redes y se extendieron por todo el mundo.

Hoy en día las redes permiten hacer múltiples actividades como puede ser compartir recursos, hacer conexiones remotas, comunicación mediante el correo electrónico y video, entretenimiento interactivo,



servicios de Internet, entre otros. A medida que van aumentando las redes también van aumentando los recursos y usuarios sobre ellas por lo que se hace necesario almacenar y organizar la información de los recursos de la red para su utilización y localización. Además es necesario controlar el acceso a dichos recursos por los usuarios de la red. En la década del 90 se dieron los primeros pasos al respecto surgiendo el concepto de servicio de directorio que representa una base de datos especializada en operaciones de consultas, que almacena información sobre los recursos presentes en la red (tales como usuarios, ordenadores, impresoras, etc.) y que pone dicha información a disposición del administrador del servidor para su gestión, y de los usuarios para realizar operaciones de consulta de la información no privada; surgiendo posteriormente las primeras implementaciones.

Existen en la actualidad múltiples variantes del servicio de directorio, unas pertenecen al Software Libre, y otras al Software Privativo, lo cual puede ser un elemento importante. Del lado del Software Libre se puede mencionar Fedora Directory Server, OpenLDAP, entre otros. En relación al Software Privativo se encuentran soluciones como Active Directory para Windows Server 2000, 2003 y 2008, Red Hat DS, entre otros. En el caso de las variantes del Software Libre brindan disímiles ventajas que vienen asociadas principalmente por las 4 libertades básicas que da a sus usuarios: la libertad de que el software se puede utilizar con cualquier fin, ya sea educativo, cultural, comercial, político o social, la libertad de estudiar su funcionamiento gracias a que se tiene acceso al código fuente, y adaptar el programa a las necesidades, suprimir partes o agregarle partes que se consideran importantes, la libertad de redistribuir copias del programa a una persona o a varias, y la libertad de hacer mejoras al programa, y de redistribuir dicho programa mejorado o de proponer la mejora en un lugar público. Gracias a estas libertades se pueden mencionar algunas ventajas como son: la corrección de errores es muy rápida al ser muchas las personas que tienen acceso al código fuente, cualquier persona que estudie y desarrolle a partir de estos códigos puede ofrecer soporte, adaptaciones de un software libre a cualquier lengua, o correcciones de errores ortográficos. Además brinda mayor seguridad puesto que ofrece garantía de continuidad ya que puede seguir siendo usado aún después de que haya desaparecido la persona que lo elaboró; y un ahorro significativo en costos de adquisición y de soporte.

En Cuba desde la década de los 90 cuando el proyecto GNU no llegaba aún a los diez años de creado, ya comenzaba el uso de software libre principalmente en servidores. También el gobierno cubano manifestó su interés por el Software Libre, considerándolo esencial para el desarrollo del país por las ventajas que



brindaba. Posteriormente, en el año 2002 fue lanzada una estrategia guiada por el Ministerio de la Informática y las Comunicaciones (MIC) para favorecer la inserción en el país de las tecnologías libres, reafirmando dos años después con el acuerdo 84 de abril de 2004 donde se disponía la migración progresiva hacia Software Libre.

Existen diferentes entidades impulsando dicha migración, entre las cuales vale destacar la Universidad de las Ciencias Informáticas (UCI) con diferentes proyectos productivos que tributan al mismo objetivo. Uno de esos proyectos productivos es “Servicios Telemáticos” que proporciona servicios de montaje, configuración y administración de los servicios telemáticos en GNU/Linux. A raíz de la problemática de que la administración de estos sistemas de servicios de directorio del lado de Software Libre en ocasiones puede ser realmente tedioso y complicado a través de terminales o consolas, en dicho proyecto productivo se realizó un Módulo de Directorio v1.0 en el curso académico 2008-2009 para la gestión del servicio de directorio con interfaces cómodas, de un modo más intuitivo y fácil para los administradores. En dicho módulo se utilizó para ofrecer servicio de directorio la variante de OpenLDAP, siendo consecuente con la estrategia del país.

Sin embargo, al Módulo de Directorio v1.0 se le pudieran aumentar una serie de funcionalidades que propiciarían una mejor gestión de este servicio. Por ejemplo en la actualidad la mayoría de las variantes de gestión del servicio de directorio no permiten redefinir esquemas -que representan la definición de clases de objetos y atributos asociados-, lo cual constituye una desventaja que puede impedir adecuarse a objetivos específicos de las organizaciones, pues el administrador no puede adicionarle a un objeto (que es un usuario o recurso) un nuevo atributo, ni agregar o eliminar nuevas clases de objetos. Además que no ofrece almacenamiento de atributos binarios como imágenes, no cuenta con la generación de certificados, y comunicación segura con el servidor; son aspectos que se convierten en puntos débiles para la gestión del servicio de directorio.

También se puede apreciar que una parte considerable de los servicios de directorio en Cuba están sobre plataformas Windows, lo cual no es consistente con la estrategia del país de utilizar Software Libre. Por esta razón es necesario progresivamente ir sustituyendo estos servicios de directorio por aquellos que estén en Software Libre, sin que esto conlleve a perder la información y por tanto almacenar nuevamente toda la información referente sobre los recursos presentes en la red.



De lo anterior surge el siguiente **problema científico**: ¿Cómo extender las capacidades del Módulo de Directorio v1.0 de la Plataforma de Gestión de Servicios Telemáticos en GNU/Linux?

Dado este problema científico se plantea como **objetivo general**: Desarrollar nuevas funcionalidades al Módulo de Directorio v1.0 de la Plataforma de Gestión de Servicios Telemáticos en GNU/Linux.

Como **objeto de estudio** se tiene los procesos relacionados con la gestión de los servicios telemáticos en redes. El **campo de acción** estará centrado en los procesos que garanticen la administración del servidor de directorio OpenLDAP.

Para cumplir el objetivo del presente trabajo se trazaron las siguientes **tareas de investigación**:

- Realización del estado del arte acerca de sistemas similares para el estudio de sus características.
- Selección de las tecnologías, herramientas y lenguaje de programación para el desarrollo del sistema.
- Investigación y desarrollo de la definición y uso de esquemas propios en el servidor OpenLDAP para la personalización de la información.
- Investigación y desarrollo de la gestión de atributos binarios para la identificación de los usuarios.
- Investigación y desarrollo de la generación de certificados para la comunicación segura desde el sistema empleando comunicación SSL.
- Investigación sobre el proceso de migración del servicio de directorio de Windows (Active Directory) a OpenLDAP para un mejor entendimiento de dicho proceso.
- Desarrollo de la funcionalidad migración del servicio de directorio de Windows (Active Directory) a OpenLDAP para la gestión del servicio de directorio desde la plataforma de Software Libre.
- Realización de la documentación de cada una de las funcionalidades para un mejor entendimiento y usabilidad.



Con el correcto cumplimiento de las tareas se espera obtener como **posible resultado**: Un Sistema integral para la gestión más eficiente del servicio de directorio que brinde la posibilidad de aplicarlo en Redes de Área Local (LAN).

El presente documento consta de 4 capítulos:

Capítulo 1: “Fundamentación Teórica”, permite encontrar 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 actual.

Capítulo 2: “Características del sistema”, se realiza el análisis de la situación problemática existente, la construcción de un modelo de negocio adecuado; la especificación de los requisitos que constituirán 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.

Capítulo 4: “Implementación”, muestra cómo será implementado el sistema en término de componentes, así como la estructura general de despliegue del mismo asociada a sus servidores de red.



Capítulo 1: “Fundamentación teórica”

1.1 Introducción

El presente capítulo recoge los conceptos fundamentales referentes al trabajo, tales como servicio de directorio, LDAP, entre otros; así como las tendencias actuales que siguen los servidores de directorio y herramientas para su administración. También se realiza la caracterización de las tecnologías y metodologías propuestas 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.

Los directorios están diseñados para dar una rápida respuesta a grandes volúmenes de búsquedas. Estos tienen la capacidad de replicar la información para incrementar la disponibilidad y la fiabilidad, al tiempo que reducen los tiempos de respuesta.(1)

1.3 LDAP

LDAP (Lightweight Directory Access Protocol), como el nombre sugiere, es un protocolo ligero de acceso a los servicios de directorio, específicamente al X.500 basado en servicios de directorio. LDAP corre sobre TCP / IP y otros servicios orientados a conexión de transferencia. LDAP es un protocolo estándar de IETF y se especifica en "Lightweight Directory Access Protocol (LDAP) Technical Specification Road Map" RFC4510.(1)

1.4 Servidores de directorio



Capítulo 1: “Fundamentación teórica”

Se conoce como servidor de directorio a la aplicación que contiene la implementación del protocolo LDAP. Además se encarga de atender las peticiones de los clientes y dar respuesta a las búsquedas y modificaciones sobre el árbol de directorio.

Como primer servidor de directorio a mencionar se tiene el **Active Directory** de **Windows Server 2003**, que constituye una variante de Software Privativo, que brinda la empresa Microsoft. Dispone de características como la escalabilidad (puede soportar un elevado número de objetos), seguridad (se puede controlar el acceso a cada objeto), replicación entre servidores (cualquier controlador de dominio puede procesar cambios del directorio, replicándose al resto de los controladores cuando existe una actualización), y flexibilidad. Además dispone de una interfaz familiar, haciéndolo muy fácil de usar y asistentes que facilitan la configuración de funciones específicas del servidor y de las tareas habituales de administración de servidores.

Recientemente fue liberado **Windows Server 2008**, que en esta versión el **Active Directory** aporta características como las copias de seguridad, recuperación de objetos eliminados, proceso mejorado para la unión de dominios, gestión de cuentas de usuario que se utilizan como identidad de los servicios, entre otros. Esta versión mejora considerablemente la herramienta Administrador del servidor que concentra en una misma interfaz la mayoría de las tareas de configuración, supervisión y administración, mediante la cual se puede examinar el estado de los servicios del sistema, efectuar copias de seguridad, administrar usuarios y grupos, editar directivas de seguridad, entre otros. Windows Server 2008 incluye también editores para herramientas fundamentales, como son las directivas de grupo que simplifica la creación, edición y asociación de las mismas a los distintos elementos del Directorio Activo, desde grupos concretos hasta dominios completos.

Lamentablemente para todas las distribuciones de Windows incluyendo la versión de servidores, es necesario comprar una licencia privativa por la cual hay que pagar altos precios. A pesar de ello, no permiten el acceso a su código fuente, así como la negación de soporte y mantenimiento, restricciones para futuras ventas de productos, ataduras tecnológicas, riesgo para la seguridad nacional, entre otros. Por todas las razones antes expuestas no constituye una opción para países pobres y en vías de desarrollo.



En el caso de **Fedora Directory Server** es un servidor de código abierto basado en LDAP, desarrollado por el proyecto Fedora. Dicho servidor brinda configuración de aplicaciones centralizada, información de grupos, así como información de control de acceso dentro de un sistema operativo independiente de la plataforma. Además contiene elementos significativos como la replicación entre varios servidores, escalabilidad, autenticación y transporte seguro (SSLv3, TLSv1, y SASL), soporte para LDAP v3, entre otros. Fedora Directory Server simplifica la administración de usuarios, eliminando la redundancia de datos y automatizando su mantenimiento. Este servidor se encuentra acoplado a la distribución Fedora de GNU/Linux por lo que se restringe la gestión de todos los servicios a una distribución específica.

Calculate Directory Server (CDS), es una implementación de LDAP que brinda la distribución Calculate Linux basada en Gentoo, como solución de código abierto. CDS ofrece también otros servicios como DNS, proxy, correo, ftp, Jabber y otros. Además CDS no es exigente con los recursos de hardware, tiene alta fiabilidad, no requiere de pago o licencia, trabaja en un entorno heterogéneo (Windows / Linux), permite la autorización centralizada de usuarios de la red, y dispone de herramientas para crear copias de seguridad de perfiles de usuario así como herramientas para la administración remota. La configuración y administración del servidor se realiza mediante el paquete Calculate–servidor. Sin embargo, CDS al igual que el servidor anteriormente mencionado se encuentra acoplado a una distribución específica.

Otra implementación libre del protocolo LDAP es **OpenLDAP** desarrollado por el proyecto del mismo nombre. Se distribuye bajo licencia GNU, que permite que el software se pueda usar de forma gratuita tanto de forma educativa como profesional. Además se dispone del código fuente para poder realizar modificaciones propias. Es un sistema flexible y escalable, está optimizado en lectura de registros, brinda la posibilidad de múltiples directorios independientes, la mayoría de aplicaciones y sistemas operativos disponen de soporte para LDAP, permite establecer criterios de búsqueda complejos, y garantiza la réplica de la base de datos a otro servidor. OpenLDAP funciona en sistemas operativos como: Apple Mac OS X, Linux: Debian, Red Hat, Suse, Fedora, Mandrake, y otros; FreeBSD, IBM AIX, Microsoft Windows 2000, XP, NetBSD, Solaris. Además OpenLDAP incluye un número de características importantes:

- 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), entre otras mejoras. Muchos de los cambios en el protocolo desde LDAPv2 han sido diseñados para hacer LDAP más seguro.



- 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.
- API de C actualizada: Mejora la forma en que los programadores se conectan para usar servidores de directorio LDAP.
- 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, herramientas mejoradas y mucho más.

Por todas las razones anteriores constituye una opción excelente para tomar como servidor de directorio para el sistema informático que se va a desarrollar.

1.5 Sistemas para la gestión de los servidores de red

Existe un grupo de herramientas que ayudan a la administración de los servidores. Una de las más utilizadas es **Webmin**, la cual se basa en la web para la administración de sistemas operativos Unix. Al ser una herramienta web posee todo un conjunto de bondades como por ejemplo no necesita ser instalada, es multiplataforma ya que solo depende de un navegador web y brinda la posibilidad de administrar ya sea una simple computadora o un servidor. Permite el uso de SSL y la administración de servidores LDAP. También brinda la posibilidad de configurar y administrar un conjunto de servidores como son el servidor web Apache, PHP, MySQL, DNS, Samba, DHCP, entre otros. Está escrito en Perl en su versión 5. Es muy fácil adicionar nuevas funcionalidades a Webmin debido a su diseño modular. Se encuentra bajo licencia BSD (Berkeley Software Distribution). La versión más reciente es Webmin versión 3.8.

phpLDAPadmin es otra herramienta para la administración de servidores LDAP. Está escrita en PHP, multiplataforma, bajo la licencia GPL y se encuentra en cerca de 10 idiomas. Además es una herramienta



basada en la web por lo que brinda al administrador la posibilidad de administrar el servidor LDAP desde cualquier lugar en la red. La versión más reciente es phpLDAPAdmin versión 1.2. phpLDAPAdmin dispone de una gran variedad de funcionalidades, entre las que se encuentran: vista jerárquica del árbol de directorio, copiar/eliminar recursivamente arboles enteros, ver y editar atributos como imágenes, visualizar los esquemas existentes en el servidor de directorio, realizar búsquedas simples y avanzadas, administrar las contraseñas de usuario, entre otros.

Recientemente también fue liberado **SIGATI** que es un sistema para la administración de un servidor OpenLDAP basado en la web. Es un software con licencia GPL, con interfaz amigable y brinda entre otras características: Seguridad (HTTPS, TLS: Transport Layer Security), autenticación SASL(Simple Authentication and Security Layer) para permitirle al administrador autenticarse solo una vez, administración de particiones (crear una partición inicial, crear sub-partición, transformar una entrada LDAP en una partición, eliminar una partición, control de acceso para la administración de particiones), administración de esquemas (visualización de esquemas, añadir / modificar esquemas, control de acceso a los esquemas), administración de “access control list” (concesión de acceso, eliminar el acceso, herencia de acceso, replicación de acceso, registro de fallos durante la propagación de acceso, reglas de control de acceso), entre otras.

No se puede dejar de mencionar la herramienta **Módulo de Directorio v1.0** de la Plataforma de Gestión de Servicios Telemáticos en GNU/Linux elaborada en la UCI en el curso académico 2008-2009. Dicha herramienta permite la administración del servidor de modo remoto, así como autenticar usuario, gestionar entrada, buscar entradas, obtener puertos, gestionar puertos de acceso, gestionar estado, certificar servidor, gestionar réplicas, gestionar configuraciones, y mostrar estructura general. Todas las funcionalidades anteriores permiten gestionar el servidor de directorio OpenLDAP. Con dicha herramienta se apuesta por la independencia tecnológica, intentando llevar las mejores prácticas y funcionalidades de otras herramientas a una totalmente desarrollada por Cuba.

1.6 Lenguaje de modelado

UML (Unified Modeling Language) permite a los creadores de sistemas generar diseños que capturen las ideas de una forma convencional y fácil de comprender para comunicarlas a otras personas. Es un lenguaje para visualizar, especificar, construir y documentar los artefactos que se crean durante el



proceso de desarrollo de un sistema que involucra una gran cantidad de software. No es una guía para realizar el análisis y diseño orientado a objetos, sino un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos. Es importante destacar que un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema. Tiene gran importancia hoy en día puesto que proporciona organización para manejar la complejidad de grandes sistemas informáticos, de manera que los analistas, clientes, desarrolladores y otras personas involucradas en el desarrollo del sistema lo comprendan. En resumen, UML es un lenguaje de representación visual que permite combinar diversos elementos gráficos y crear diagramas; y cuenta con reglas para combinar tales elementos.

En UML se identifican:

- Elementos: (abstracciones que constituyen los bloques básicos de construcción)
- Relaciones: (Relacionan los elementos)
- Diagramas: (Representación gráfica de un conjunto de elementos)

Entre los principales objetivos de UML se puede mencionar que es un lenguaje de modelado de propósito general que pueden usar todos los modeladores. No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática. Además UML pretende trabajar correctamente con todos, o al menos con la mayoría de los procesos de desarrollo existentes. También se puede mencionar que UML incluye todos los conceptos que se consideran necesarios para utilizar un proceso moderno iterativo, basado en construir una sólida arquitectura para resolver requisitos dirigidos por casos de uso.

Como objetivo final de UML es ser tan simple como fuera posible pero manteniendo la capacidad de modelar toda la gama de sistemas que se necesita construir. UML necesita ser lo suficientemente expresivo para manejar todos los conceptos que se originan en un sistema moderno, tales como la concurrencia y distribución, así como también los mecanismos de la ingeniería de software, tales como encapsulación y componentes. Debe ser un lenguaje universal, como cualquier lenguaje de programación de propósito general. (2)

1.7 Metodología de desarrollo



Capítulo 1: “Fundamentación teórica”

En el ciclo de vida del software se deben completar una serie de tareas para obtener un producto de software. Cada una de esas tareas puede ser abordada y resuelta de múltiples maneras, con distintas herramientas y utilizando distintas técnicas. Es necesario saber cuándo se puede dar por concluida una tarea, quién debe realizarla, qué tareas preceden o anteceden a una dada, qué documentación se utilizará para llevar a cabo esa tarea. Se está hablando de detalles organizativos, de un "estilo" de hacer las cosas. Viendo un poco más allá de un simple estilo, formalizando ese "estilo" añadiendo algo de rigurosidad y normas se obtiene una metodología.

Se entiende por metodología de desarrollo una colección de documentación formal referente a los procesos, las políticas y los procedimientos que intervienen en el desarrollo del software. La finalidad de una metodología de desarrollo es garantizar la eficacia y la eficiencia en el proceso de generación de software. Todos los integrantes de un equipo de desarrollo deben seguir un criterio común a la hora de realizar las tareas del ciclo de vida. Ese criterio, esa manera común es una metodología de desarrollo.

Una de las metodologías modernas es Rational Unified Process (Proceso Racional Unificado - RUP), la cual unifica los mejores elementos de metodologías anteriores, preparado para desarrollar grandes y complejos proyectos, orientado a objetos y utiliza el UML como lenguaje de representación visual.

Los principales elementos de RUP son:

Trabajadores (“quién”): Define el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que trabajan en conjunto.

Actividades (“cómo”): Es una tarea que tiene un propósito claro, es realizada por un trabajador y manipula elementos.

Artefactos (“qué”): Productos tangibles del proyecto que son producidos, modificados y usados por las actividades. Pueden ser modelos, elementos dentro del modelo, código fuente y ejecutables.

Flujo de actividades (“cuándo”): Secuencia de actividades realizadas por trabajadores y que produce un resultado de valor observable.



Capítulo 1: “Fundamentación teórica”

En RUP se han agrupado las actividades en grupos lógicos definiéndose 9 flujos de trabajo principales. Los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo, los cuales son:

- Modelación 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 (Testeo): 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 del proyecto: Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.
- 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.
- 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.

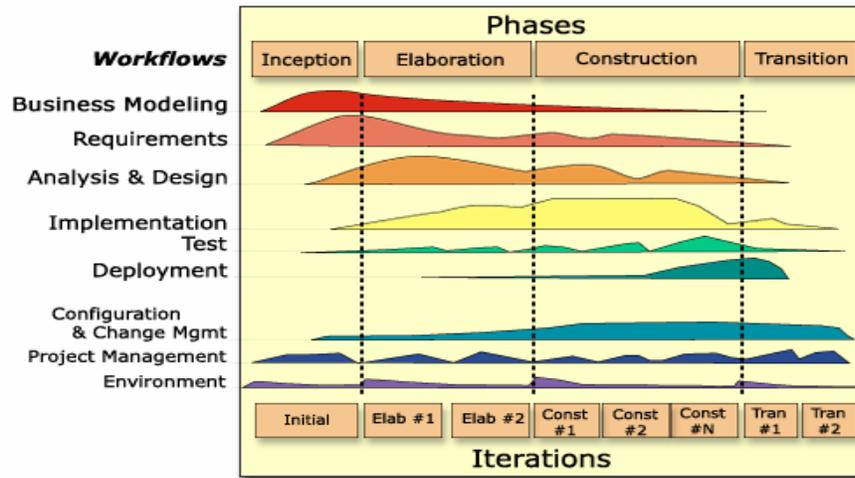


Figura 1: RUP en dos dimensiones

Las fases de RUP son:

- Conceptualización (Concepción o Inicio): Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.
- Elaboración: Se define la arquitectura del sistema. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.
- Construcción: Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene uno o varios *release* del producto que han pasado las pruebas. Se ponen estos *release* a consideración de un subconjunto de usuarios.
- Transición: El *release* ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.

El ciclo de vida de RUP se caracteriza por:



- Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.
- Centrado en la arquitectura: La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura.
- Iterativo e Incremental: RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros.

1.8 Modelación de funciones mediante IDEF0

IDEF0 (Integration Definition for Function Modeling) surge en la década de los 70 como una técnica de modelación. Permite modelar de una manera estructurada y jerárquica los procesos llevados a cabo en una empresa o sistema. Es utilizado para modelar nuevos sistemas o sistemas ya existentes. Utilizando esta técnica de modelado se pueden emplear dos notaciones diferentes: AS-IS y TO-BE. La notación AS-IS es empleada para representar los procesos del negocio de la manera en que se encuentran antes de la automatización de los mismos. Y la notación TO-BE es utilizada para representar la transformación que se le desea dar al proceso, después de identificarse las posibles mejoras y tomando en cuenta los objetivos perseguidos por la empresa.

Entre las ventajas se puede mencionar:

- Permite describir los procesos cronológicamente.
- Se describen los procesos orientado al cliente final.
- Es una notación simple.



- Permite incorporar en el flujo, los datos que entran y salen de las actividades, así como las reglas del negocio y los actores.
- Permite descubrir problemas existentes en la empresa que pueden ser arreglados.

IDEF0 es muy utilizado para la documentación, el análisis y la mejora de todo tipo de procesos. Existen cinco elementos en un modelo IDEF0:

Las flechas de las entradas, controles, salidas y mecanismos denominadas con el nombre de ICOMs (Input, Control, Output, Mechanism), y las actividades o procesos. Ver Figura 2



Figura 2: Componentes básicos de IDEF0

- La actividad (o proceso): Se representa por medio de cajas.
- Las entradas: Se representan mediante flechas que entran por la parte izquierda de una caja.
- Las salidas: Se representan por flechas saliendo por la parte derecha de las actividades.
- Los controles: Representados por flechas que entran por la parte superior de las cajas.
- Los mecanismos: Para poder llevar a cabo el proceso, que se representan mediante flechas entrantes a las cajas por su parte inferior.



El sistema es jerárquico. Cada diagrama representa una actividad necesaria para la tarea, en un grado de detalle específico. Las actividades se subdividen en diagramas que siguen en niveles inferiores hasta un grado de detalle necesario.

La estructura de la caja del hijo es una descomposición detallada de una caja padre, donde una caja hijo también puede ser una caja padre. La relación se indica por la presencia de un código corto escrito debajo de la esquina derecha inferior de la caja. Ver Figura 3.

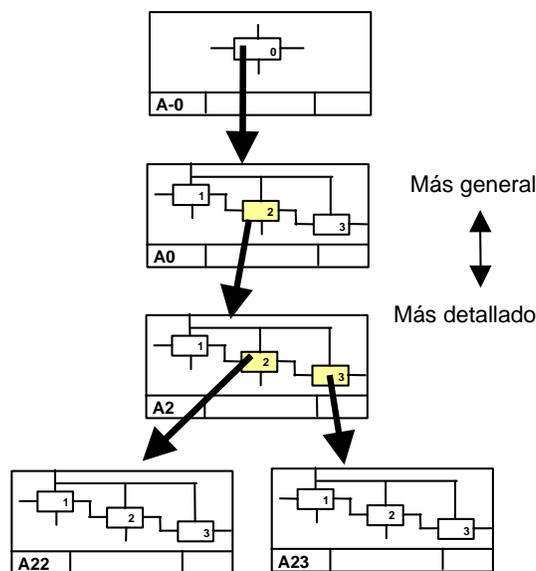


Figura 3: Jerarquía de procesos en IDEF0

Reglas para la numeración de las cajas

- El modelo contendrá un diagrama del proceso general A-0, que contiene solamente una caja.
- Los diagramas sucesivos tendrán por lo menos tres cajas y no más de seis cajas.
- Cada caja en un diagrama que no sea el A-0 será numerada en la esquina inferior derecha, en orden a partir de la 1 hasta el 6.

1.9 Lenguaje de programación y herramientas



1.9.1 Python

Python es un potente lenguaje de programación creado por Guido van Rossum en 1991. En la actualidad Python se desarrolla como un proyecto de código abierto. Python es un lenguaje de programación limpio y elegante que permite dividir el programa en módulos reutilizables. Incluye una gran colección de módulos estándar que se pueden utilizar como base de los programas que proporcionan funcionalidades para el manejo de ficheros, llamadas al sistema, sockets y hasta interfaces a librerías gráficas como (GTK, QT, TK). Es un lenguaje interpretado, característica que ahorra un tiempo considerable en el desarrollo del programa, al no ser necesario compilar ni enlazar. Se encuentra muy bien documentado y cuenta con una comunidad de desarrollo muy activa. Además posee tipado dinámico (puede tomar una variable diferentes tipos de datos durante la ejecución del programa) y soporta varios paradigmas, puesto que no trata de forzar a los programadores a adoptar un determinado modelo de paradigma o estilo. Utiliza la indentación para delimitar los bloques y es multiplataforma.

Python constituye la elección para dar soporte al desarrollo del sistema propuesto por ser un lenguaje:

- Simple: Python es un lenguaje simple. El pseudo-código natural de Python es una de sus grandes fortalezas ya que permite concentrarse en la solución del problema en lugar de la sintaxis del propio lenguaje.
- Sencillo de Aprender: Con Python es extremadamente sencillo de iniciarse en la programación ya que ofrece una sintaxis extraordinariamente simple.
- Libre y Fuente Abierta: Python es un ejemplo de un FLOSS (Free/Libre and Open Source Software - Gratuito/Libre y Software de Fuente Abierta). En términos simples, se puede distribuir libremente copias de este software, leer su código fuente, hacerle cambios y usar partes del mismo en nuevos programas libres.
- Lenguaje de Alto Nivel: Cuando se escriben programas en Python no hay que preocuparse por detalles de bajo nivel, como manejar la memoria empleada por tu programa.
- Portable: Debido a su naturaleza de ser Open Source, Python ha sido portado a diversas plataformas. Se puede usar Python sobre Linux, Windows, Macintosh, Solaris, OS/2, AROS,



AS/400, BeOS, OS/390, z/OS, Palm OS, QNX, VMS, Psion, Acorn RISC OS, VxWorks, PlayStation, Sharp Zaurus, Windows CE y PocketPC. Sin embargo, se debe ser lo suficientemente cuidadoso de evitar las características con dependencia de sistema (es decir librerías o módulos que funcionan sólo en un sistema en particular).

- Interpretado: El programa se ejecuta desde el código fuente. Internamente, Python convierte el código fuente en una forma intermedia llamada bytecodes, después los traduce en el lenguaje nativo de tu computadora y ejecuta. Todo esto hace el uso de Python mucho más sencillo.
- Orientado a Objetos: En lenguajes orientados a objetos, el programa es construido sobre objetos los cuales combinan datos y funcionalidad. Python ofrece una manera muy poderosa y simple de emplear programación orientada a objetos. (3)

1.9.2 Herramienta Case

1.9.2.1 Visual Paradigm

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases y generar documentación.

Visual Paradigm para UML (VP-UML) es una potente plataforma que ofrece disímiles ventajas, y la herramienta más fácil de usar de modelado visual de UML. VP-UML ofrece a los desarrolladores de software crear aplicaciones más rápidas, de calidad, mejor y más barato. Además facilita una excelente interoperabilidad con otras herramientas CASE.(4)

1.9.3 Entorno de desarrollo integrado

Un **entorno de desarrollo integrado** o **IDE** (*Integrated Development Environment*), es un programa informático que contiene un conjunto de herramientas para facilitar la programación. Los IDE brindan un marco de trabajo amigable para los desarrolladores en la mayoría de los lenguajes de programación tales como C++, C#, Java, Python, entre otros. Ofrecen por lo general un editor de textos especializado como



herramienta central, así como navegadores de archivos, asistentes de compilación, navegadores de objetos, lectores de documentación, entre otros.

1.9.3.1 EasyEclipse

La distribución EasyEclipse es un lanzamiento de Eclipse previamente empaquetado que proporciona un completo y funcional Entorno de Desarrollo Integrado (IDE). Cada distribución contiene todo lo necesario para empezar a desarrollar el código, sin descargas adicionales. Una vez que se haya instalado una distribución, puede complementarlo mediante la instalación de más plugins según sea necesario.

Precisamente, PyDev es un plugin que permite a los usuarios utilizar Eclipse para el desarrollo de aplicaciones en Python y Jython. Su distribución es bajo la licencia Eclipse Public License 1.0. Viene con muchos regalos, tales como el autocompletado de código, análisis de sintaxis resaltado, análisis de código, depuración, consola interactiva y muchos otros.

EasyEclipse es:

- Multiplataforma
- Gratuito y de código abierto
- Fácil de mantener, sin problemas de versión y de dependencia(5)

Por todas las razones antes expuestas constituye la elección para desarrollar el código Python del sistema.

1.9.4 Interfaces gráficas de usuario

1.9.4.1 Qt Designer

Qt Designer es una herramienta muy potente que permite diseñar de una forma muy sencilla y rápida ventanas de diálogo con las librerías Qt. Esta herramienta es una aplicación mediante la cual se puede realizar el diseño de aplicaciones GUI de forma gráfica, sencilla, rápida y muy intuitiva.



Las características de Qt Designer se mencionan a continuación:

- Ofrece una caja de componentes muy variada para interfaces de usuario.
- Posee un inspector de objetos para explorar los componentes usados en nuestro desarrollo.
- Posee un editor de propiedades, que permite cambiar las características de los objetos presentes en nuestra interfaz, características como el texto que presenta este en la interfaz, fuente, tamaño, el nombre del objeto, entre otros.
- Posee un editor de Señales (SIGNAL) y ranuras (SLOT), donde se puede asignar comportamientos predefinidos a ciertos objetos de nuestra interfaz.

Este generador de interfaces fue creado para trabajar en varias distribuciones Linux. No obstante, actualmente puede instalarse en otras plataformas como Windows y Mac OsX. El Qt Designer utiliza como base la librería gráfica de Qt, que ha sido transportada a diversas plataformas, lo que permite que el código generado por el Qt Designer pueda ser utilizado en diversas plataformas. Además, Qt funciona sólo o asociándose a algunos entornos de desarrollo integrado. Esta herramienta provee características muy poderosas como la pre-visualización de la interfaz, soporte para widgets y un editor de propiedades bastante poderoso.(6)

1.9.4.2 Librería Qt

Qt es un producto de la empresa noruega de software Trolltech AS, esta empresa se dedica a desarrollar librerías y herramientas de desarrollo de software, además es experta en servicios de consultoría. Qt son un conjunto de librerías multi-plataforma para el desarrollo del esqueleto de aplicaciones GUI, escritas en código C++. Qt además está completamente orientado a objetos.

Qt comenzó a distribuirse comercialmente en 1996 y desde entonces ha sido la base para numerosas aplicaciones incluyendo la popular interfaz gráfica para Linux llamada KDE, disponible en todas las grandes distribuciones de Linux.

Actualmente se pueden encontrar las siguientes distribuciones de Qt:



- Qt Enterprise Edition y Qt Profesional Edition, disponibles para el desarrollo de software con fines comerciales.
- Qt Free Edition Es la versión para Unix/X11 para el desarrollo de software gratuito y de código abierto. Se puede obtener gratis sujeto a los términos de la QPublic License y de la GNU General Public License. Para plataformas Windows también está disponible la versión Qt non comercial.
- QtEducationalEdition es una versión de la Qt Profesional Edition con licencia únicamente para fines educacionales.(7)

Se caracteriza por:

- Qt es una librería para la creación de interfaces gráficos. Se distribuye bajo una licencia libre GPL (o QPL) que permite incorporar las interfaces Qt en nuestras aplicaciones open-source.
- Se encuentra disponible para una gran número de plataformas: Linux, MacOS X, Solaris, HP-UX, UNIX con X11. Además, existe también una versión para sistemas empujados.
- Es orientado a objetos, lo que facilita el desarrollo de software. El lenguaje para el que se encuentra disponible es C++ aunque han aparecido bindings (una adaptación de una biblioteca para ser usada en un lenguaje de programación distinto al que ha sido escrito) a otros lenguajes como Python o Perl.
- Es una librería que se basa en los conceptos de widgets (objetos), Señales-Slots y Eventos.
- Las señales y los slots es el mecanismo para que unos widgets se comuniquen con otros.
- Los widgets pueden contener cualquier número de hijos. El widget "top-level" puede ser cualquiera, sea ventana, botón, etc.
- Algunos atributos como el texto de etiquetas, se modifican de modo similar al lenguaje HTML.
- Qt proporciona además otras funcionalidades:
 - Librerías básicas: Entrada/Salida, Manejo de Red, XML



- Plugins, librerías dinámicas (Imágenes, formatos, entre otros)
- Unicode, Internacionalización(8)

1.10 Conclusiones

Con los conceptos expuestos en el capítulo, y la revisión de las principales tendencias actuales de los servidores de directorio y las herramientas para su administración, en el mundo, Cuba y la UCI, se tiene una mejor comprensión del tema, garantizando que el sistema informático a desarrollar contenga las mejores prácticas, aportando así a la independencia tecnológica de Cuba. Además se realizó un análisis detallado de cada una de las elecciones tanto de lenguaje, IDE, metodología, y demás herramientas que se utilizarán en el desarrollo, siendo consecuente con las estrategias de la UCI y Cuba.



Capítulo 2 “Características del sistema”

2.1 Introducción

En el presente capítulo se describen de manera clara los procesos necesarios para realizar la gestión de un servidor de directorio OpenLDAP. Se utilizará la notación IDEF0 para el modelamiento del negocio ya que posibilita una mejor comprensión de los procesos a automatizar. Además son identificados los requisitos funcionales y no funcionales para definir casos de uso, así como los actores y sus relaciones dentro del sistema propuesto. Las diferentes funcionalidades del sistema se representan en los diagramas de paquete y de casos de uso.

2.2 Problema y situación problemática

Hoy en día existen un conjunto de herramientas que les permiten a los administradores de redes realizar la gestión de un servidor de directorio de manera cómoda, rápida e intuitiva. Dichas herramientas son en su mayoría soluciones de pago que no ofrecen su código fuente, con excesivas restricciones de licencias y a las cuales no se tiene derecho de asistencia y ayuda técnica en caso de necesitarla. En el caso de Cuba con el acuerdo 84 de abril de 2004 se disponía la migración progresiva hacia Software Libre.

En la UCI se desarrolló un Módulo de Directorio v1.0 para la gestión de un servidor OpenLDAP que contribuyera al desarrollo tecnológico, con sello propio y adecuado a las características de Cuba. A dicha aplicación le serán adicionadas una serie de funcionalidades para brindarles mayores prestaciones a los administradores de red.

Actualmente el Módulo de Directorio v1.0 no realiza la redefinición de esquemas -que representa la definición de clases de objetos y atributos asociados-, no permite al administrador gestionar atributos binarios como pueden ser imágenes, no genera sus propios certificados y no establece comunicación segura utilizando los certificados generados, y no brinda la posibilidad de migrar los datos de un Directorio Activo a un OpenLDAP. Si el administrador quisiera hacer alguna de estas acciones tendría que realizarlo de forma manual, por lo cual automatizar estas acciones sería de gran ayuda para los administradores que utilicen la herramienta.



2.3 Información que se maneja

Constituye de suma importancia conocer cómo se maneja la información del sistema en los múltiples procesos que se realizan. Durante los procesos de redefinición de esquemas, gestión de atributos binarios, y migración de un Directorio Activo a OpenLDAP se realizan modificaciones a los ficheros relativos a dichas acciones. Los ficheros de configuración constituyen un eslabón fundamental para el correcto funcionamiento del sistema. Por tal razón una incorrecta estructura del contenido, o barrado de alguna información puede ocasionar trabajo inadecuado del servidor, o incluso que el servicio se detenga. La modificación de dichos ficheros de configuración tiene que ser sumamente cuidadosa.

2.4 Solución propuesta

Para el mejoramiento de la herramienta de gestión del servidor de directorio OpenLDAP: Módulo de Directorio v1.0, se ha decidido adicionarle importantes funcionalidades tales como la gestión de atributos binarios, gestión de esquemas, migración de datos de un Directorio Activo a un OpenLDAP, generación de certificados y la posterior comunicación segura. Con esta propuesta se pretende brindarles a los administradores una herramienta mejorada que le facilite la gestión de un servidor OpenLDAP.

La gestión de esquemas permitirá al administrador adicionar, modificar, eliminar, y mostrar los esquemas existentes en el servidor, permitiendo la personalización de la información del servidor LDAP. Con la gestión de atributos binarios, se le podrá asignar una imagen a un usuario, para la identificación de los mismos. La generación y utilización de certificados para la comunicación segura constituye uno de los elementos más importantes puesto que garantiza la confidencialidad de la información entre el servidor y la herramienta de gestión. En el caso de la migración de datos de un Directorio Activo a un OpenLDAP se realizará de un modo muy sencillo con solo brindar los datos necesarios para dicha acción, facilitando la migración del servicio de directorio hacia la Plataforma de Software Libre.

Una herramienta como la que se propone no tiene limitantes de costo de la licencia, con soporte y mantenimiento para la comunidad cubana, portable a diferentes sistemas operativos, con interfaces cómodas e intuitivas sin dejar de ser profesional para los administradores, y con un número considerable de funcionalidades que facilita el trabajo de administración de un servidor OpenLDAP.



2.5 Modelo del Negocio

Para la modelación de los procesos del negocio se utiliza la notación IDEF0, ya que garantiza la identificación de los mismos de una manera más simple. A continuación se muestra los procesos del negocio.

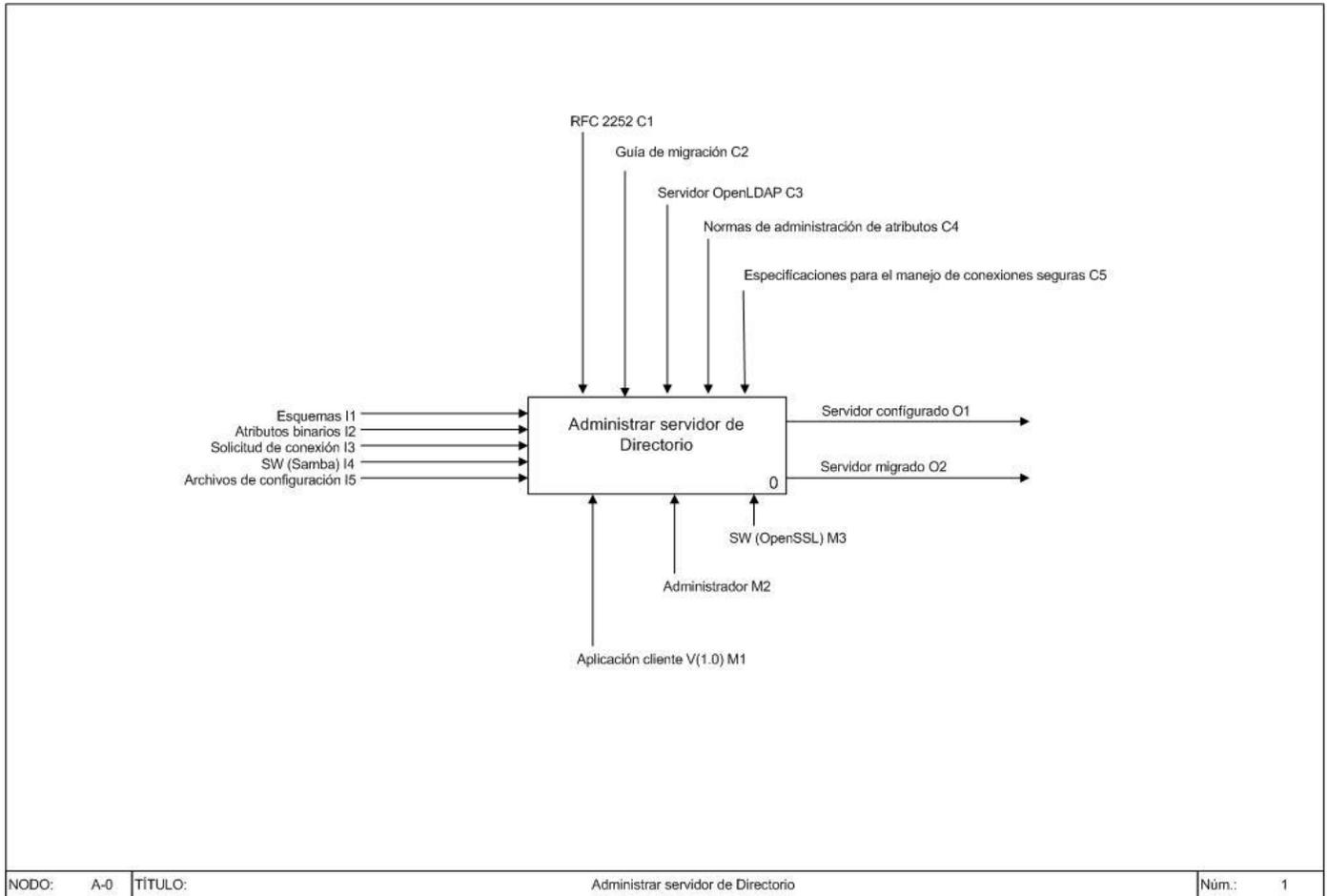


Figura 4: Representación del proceso general “Administrar servidor de Directorio”

Descripción

- **Administrar servidor de Directorio:** Este proceso consiste en extender las capacidades de administración del servicio de directorio LDAP, donde se pueden mencionar aspectos tales como



redefinir esquemas, gestionar atributos binarios, establecer conexión segura y migrar servicio de directorio. Cada uno de estos aspectos aportan mejoras a la administración del servidor de directorio.

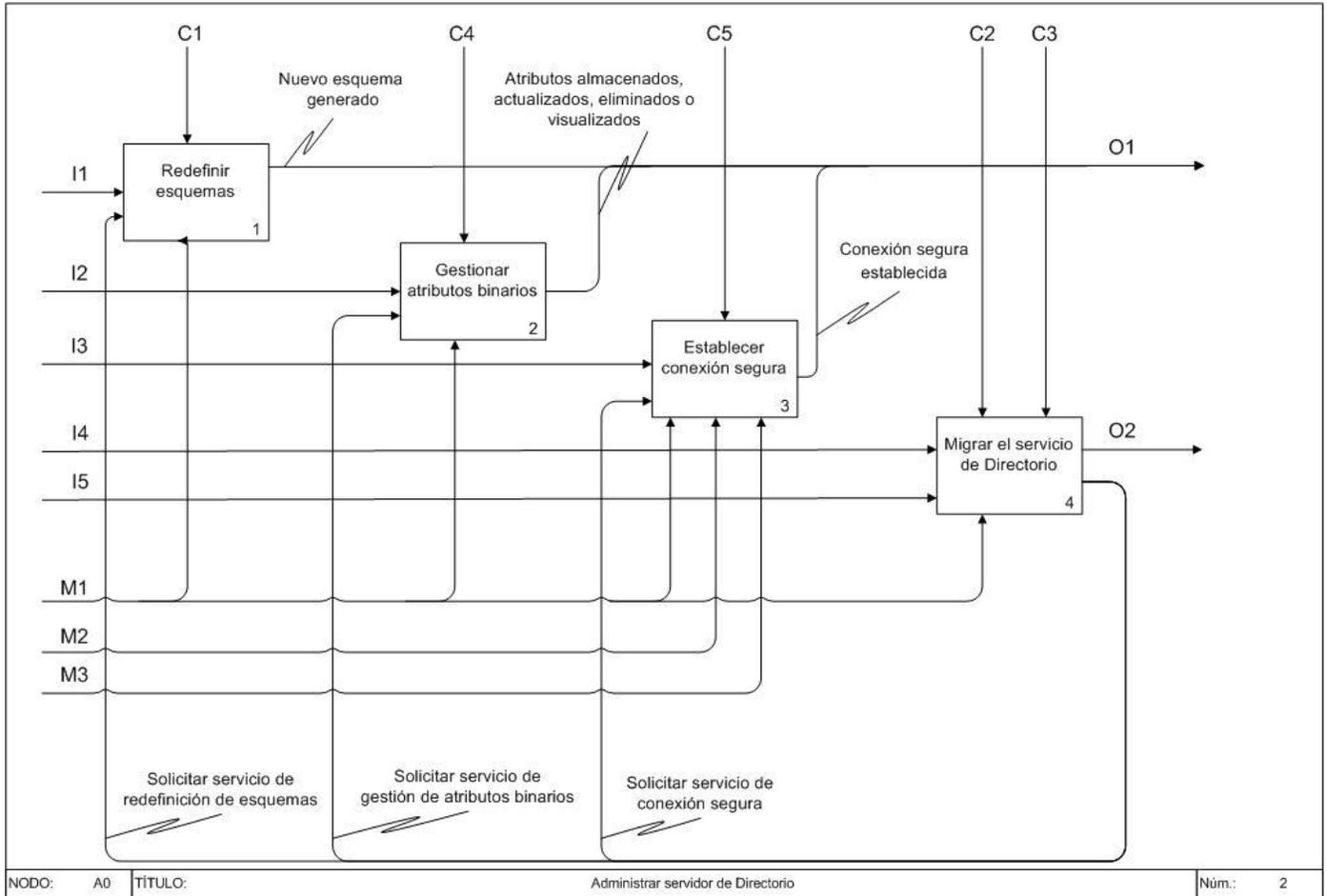


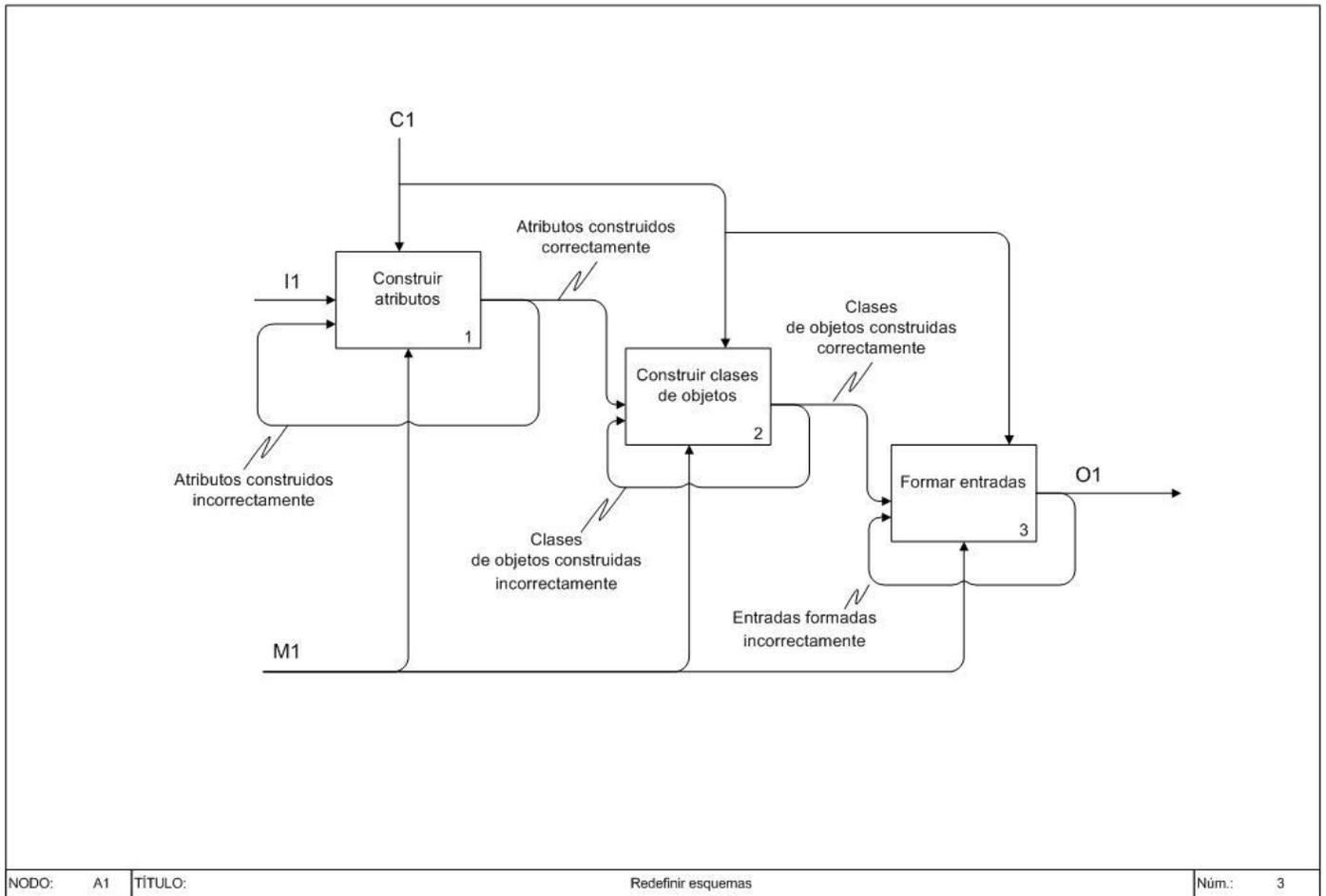
Figura 5: Detalles del proceso general

Descripción



Capítulo 2 “Características del sistema”

- **Redefinir esquemas:** Con este proceso se puede definir atributos, clases de objetos y entradas que permitirán adicionar nuevos objetos al directorio, que garantice la personalización de la información. Además se podrán modificar y eliminar dichos elementos.
- **Gestionar atributos binarios:** El proceso consiste en la gestión de los atributos binarios, tales como imágenes, para una mejor administración del servidor de directorio.
- **Establecer conexión segura:** Consiste en verificar si se puede establecer conexión segura, y seguir los pasos necesarios tales como obtener certificado, o generarlo, enviar certificado, para finalmente lograr la conexión segura para el intercambio de información.
- **Migrar el servicio de Directorio:** Cuando se tiene el servicio de directorio sobre plataforma de Windows y se trata de migrar este servicio hacia la plataforma del Software Libre, es necesario también migrar la información de un servidor hacia otro. De dicha migración de datos se trata este proceso.



NODO:	A1	TÍTULO:	Redefinir esquemas	Núm.:	3
-------	----	---------	--------------------	-------	---

Figura 6: Detalles del proceso Redefinir esquemas

Descripción

- **Construir atributo:** El proceso se trata de definir los atributos necesarios, que consiste en los nuevos datos que el administrador del directorio va a adicionar a un objeto, para lograr la personalización de la información.
- **Construir clases de objetos:** Con los atributos necesarios ya definidos, entonces se procede a la definición de las clases de objetos, en las cuales se definen los atributos que pueden usarse para cada una de las entradas.



- **Formar entradas:** Teniendo definido los atributos y las clases de objetos, es necesario adicionar nuevas entradas, para poder agregar un nuevo objeto en el directorio. Cada entrada en el directorio corresponde a un objeto abstracto o real (por ejemplo, una persona, un material, parámetros, y otros). Una entrada está conformada por un conjunto de clases de objetos.

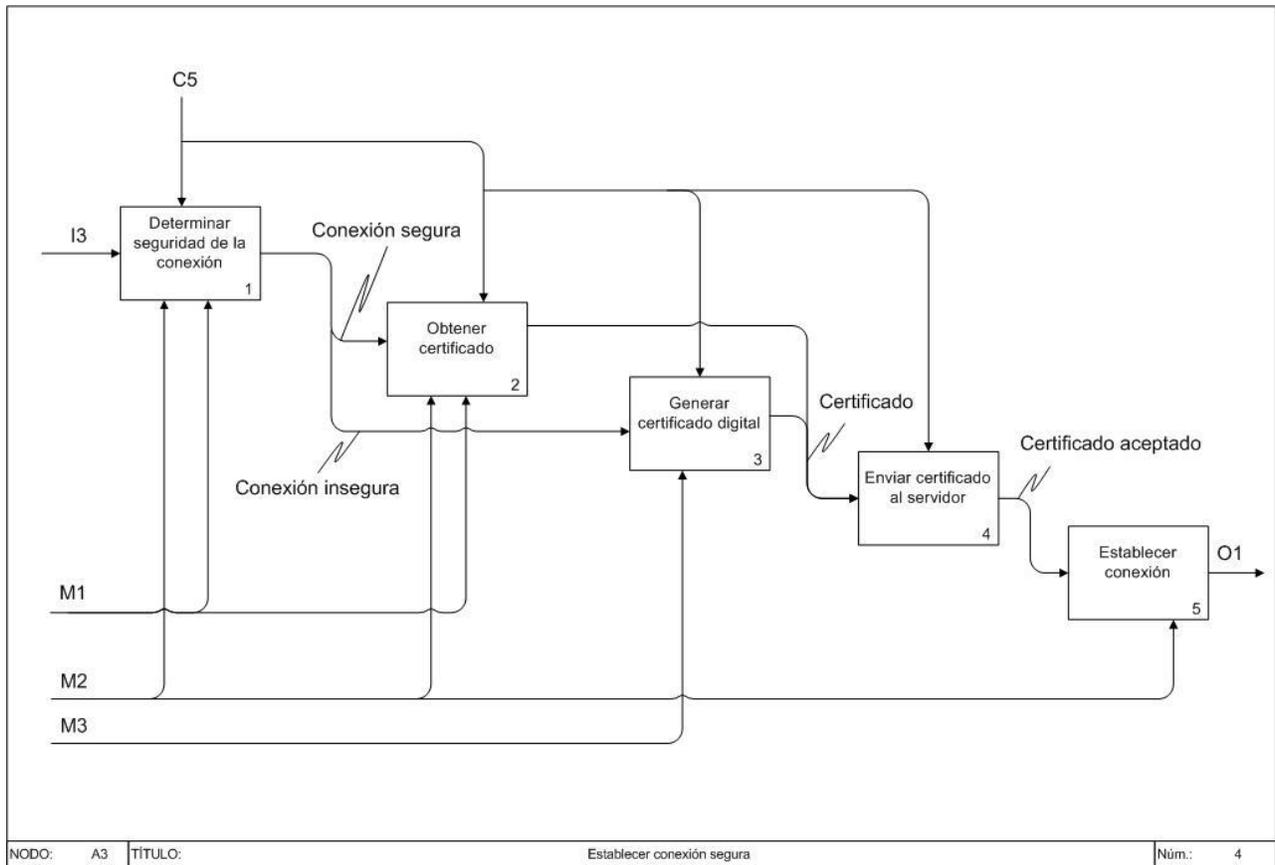


Figura 7: Detalles del proceso Establecer conexión segura

Descripción

- **Determinar seguridad de la conexión:** Con este proceso el cliente verifica si puede establecer una conexión segura con el servidor de directorio, verificando la existencia de un certificado para ello.



- **Obtener certificado:** En caso de que se pueda establecer una conexión segura, entonces se hace necesario obtener el certificado para poder intercambiar información de forma segura con el servidor.
- **Generar certificado digital:** Si no se puede establecer conexión segura porque no se cuenta con un certificado entonces se podría generar un certificado digital para el intercambio de la información.
- **Enviar certificado al servidor:** Contando con un certificado entonces se le envía al servidor de directorio, para su utilización en la conexión entre el cliente y el servidor.
- **Establecer conexión:** Cuando el cliente y el servidor cuentan con un certificado para la encriptación de la información, entonces se establece conexión segura entre ellos, siendo este proceso de gran importancia para la confidencialidad e integridad de la información.

2.6 Especificación de los requisitos de software

Posiblemente la etapa más compleja a lo largo de la producción de un software sea la de captura de requerimientos. Es en esta etapa en la cual se determina qué desea el cliente. Para lograr esto se hace necesario el establecimiento de una visión común entre el cliente y el equipo de desarrollo sobre los procesos que se llevan a cabo en el negocio y como estos se verán reflejados y mejorados en el futuro software. Esta tarea es generalmente complicada ya que la mayoría de los clientes no dominan las tecnologías informáticas.

Esta es una etapa de vital importancia puesto que de ella depende que los resultados sean los esperados y que el cliente quede satisfecho con el producto.

2.6.1 Requisitos funcionales

Los requisitos funcionales reflejan las capacidades con las que debe contar el sistema. Estos se mantienen invariables sin importar con que propiedades o cualidades se relacionen. A continuación serán tratados los requerimientos funcionales asociados al sistema:



R1. Gestionar esquemas

R1.1 Agregar esquema

R1.2 Eliminar esquema

R1.3 Modificar esquema

R1.4 Mostrar esquema

R2. Migrar servidor

R3. Gestionar atributos binarios

R3.1 Agregar atributo binario

R3.2 Eliminar atributo binario

R3.3 Actualizar atributo binario

R3.4 Mostrar atributo binario

R4. Establecer comunicación segura

R4.1 Establecer comunicación con certificado generado

R5. Buscar esquema

R6. Buscar atributo binario

R7. Generar certificado

2.6.2 Requerimientos no funcionales:



Capítulo 2 “Características del sistema”

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Estos, generalmente se encuentran muy ligados a requerimientos funcionales, pues una vez que se ha determinado lo que debe hacer el sistema se pasa a describir cómo debe de hacerlo.

En muchos casos los requerimientos no funcionales son fundamentales en el éxito de un producto. Estos generalmente están relacionados con cuán usable, seguro, conveniente y agradable debe de ser el sistema. El sistema propuesto consta de un modelo cliente-servidor por lo cual los requerimientos no funcionales serán abordados desde ambas perspectivas.

Requerimientos de software:

- Como sistema operativo se debe tener instalado cualquiera de la familia GNU/Linux.
- El proceso de instalación de servidores requiere que los ordenadores que van a hospedar el servidor contengan un compilador de paquetes apropiado.
- Los ordenadores para administrar el trabajo en los servidores podrán contener un sistema operativo GNU/Linux o Windows.

Requerimientos de hardware:

Estos requerimientos están en correlación con el tamaño de la red y a la cantidad de usuarios que recibirán el servicio.

Es necesario destacar que en el caso de los servidores se hace necesaria una buena estructura de hardware que les permita dar cumplimiento a su principal funcionalidad, atender las solicitudes de un grupo de clientes.

Si se tratase de una pequeña red, con servicio para algunas decenas de usuarios, podría emplearse:

- Microprocesador Pentium IV
- 512 MB de memoria RAM.
- 5 GB de espacio libre en el disco duro.



Capítulo 2 “Características del sistema”

Si el tamaño de la red aumentara y se tratase ya de algunos miles de usuarios, sería necesario aumentar la capacidad de almacenamiento y de procesamiento de los servidores.

En el caso de las estaciones de trabajo en las que se encontrará el cliente, los requisitos de hardware son muy bajos:

- Microprocesador Pentium II
- 256 MB de memoria RAM.

Restricciones en el diseño y la implementación:

Este tipo de requerimiento especifica o restringe la codificación o construcción de un sistema.

- Será empleado python 2.5 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.
- Como arquitectura general del sistema se ha determinado utilizar una arquitectura en capas debido a las ventajas que ello proporciona en términos de claridad, facilidad en la corrección de errores producidos durante el desarrollo, rápido desarrollo al permitir el trabajo paralelo en cada una de las capas definidas, entre otros factores.
- Sería realmente ineficiente si en cada ocasión que los desarrolladores de sistemas informáticos se proponen una nueva meta deban comenzar la elaboración de su producto desde la base. Debido a que existen casos donde se pueden reutilizar muchas funcionalidades. Con el objetivo de resolver la problemática anterior han sido creadas bibliotecas de clases o frameworks que faciliten el desarrollo y acorten los tiempos de producción, permitiendo a los desarrolladores concentrarse en aspectos de mayor nivel. El sistema propuesto se ha apoyado en un framework desarrollado en el



Capítulo 2 “Características del sistema”

proyecto anteriormente mencionado para resolver problemas puntuales asociados a la gestión de servidores.

Requerimientos de apariencia o interfaz externa:

Este tipo de requerimiento describe la apariencia del producto sin llegar a un alto nivel de detalle.

- El sistema deberá contener una interfaz agradable, sencilla y sugerente; presentará por defecto el color azul, sin sobrecargas de otros colores que puedan provocar rechazo a los usuarios finales. El azul es representativo del pensamiento, se caracteriza por su efecto calmante; proporciona descanso y relajación a la vista.
- No se presentarán excesivas imágenes o gráficos que puedan retardar el tiempo de respuesta de la aplicación, se hará uso fundamentalmente de pequeños íconos capaces de ilustrar las funcionalidades en combinación con textos claros.
- La navegación a través del sistema se realizará empleando un menú en la región superior que elimine el exceso de interfaces extras y ayude a una mejor orientación de los usuarios.

Requerimientos de seguridad:

Los requerimientos relacionados con la seguridad son uno de los más delicados en la construcción de cualquier sistema. Este hecho alcanza aun más relevancia si se tiene en cuenta que el sistema propuesto consta de un modelo cliente-servidor, el cual utiliza la red como medio fundamental de comunicación.

- La comunicación entre el cliente y el servidor se realizará haciendo uso del protocolo Idaps.
- El sistema brindará la posibilidad de generar sus propios certificados y utilizarlos para el establecimiento de la conexión entre el cliente y el servidor.

Requerimientos de usabilidad:

Describen los niveles apropiados de usabilidad, dados los usuarios finales del producto.

- La interfaz será fácil de usar para los diversos administradores que interactúen con ella.



- El sistema estará bien documentado, con el fin de lograr un mejor uso de los servicios que este ofrecerá, para ello se realizará una ayuda que explique paso a paso cada una de las funcionalidades del software.

Requerimientos de soporte:

Este tipo de requerimientos abarcan todas las acciones a tomar una vez que se ha terminado el desarrollo del software con motivos de asistir a los clientes de este, así como, lograr su mejoramiento progresivo y evolución en el tiempo.

- El sistema contará antes de su puesta en marcha con un período de pruebas que permitirá identificar y corregir errores cometidos durante el desarrollo.
- La portabilidad del sistema hará especial énfasis en extenderse a los sistemas operativos GNU/Linux y Windows, a excepción del ordenador que hospedara realmente la instalación del servidor, donde solamente podrá tratarse de GNU/Linux para ser consecuentes con la estrategia de migración que ha seguido el país.

Requerimientos legales:

- El Módulo de Directorio, así como el sistema de gestión de servidores en su versión 2.0 será propiedad exclusiva de la Universidad de las Ciencias Informáticas (UCI).

Requerimientos políticos y culturales:

- El sistema podrá ser utilizado en primera instancia en el territorio nacional y posteriormente en otros países que se ajusten a las normas de comercialización dispuestas por la UCI.
- El sistema constará inicialmente con el idioma español.

2.7 Modelo de Casos de Uso del Sistema

2.7.1 Actores del Sistema

El actor principal del Módulo de Directorio v2.0 es el administrador del servidor.



Actor	Justificación
Administrador	Es la persona encargada de realizar todos los procesos de gestión de información en el servidor de directorio. Tiene permisos para instalar el servidor, realizarle modificaciones a las configuraciones y modificar su estado.

Tabla 1: Actor del sistema

2.7.2 Diagrama de paquetes

Cuando el modelo es grande y/o las responsabilidades del modelo son distribuidas por partes resulta apropiado utilizar paquetes. Los paquetes, tienen el propósito de organizar en grupos funcionalidades semejantes. En el caso específico del Módulo de Directorio v2.0 los paquetes quedan de la siguiente forma:

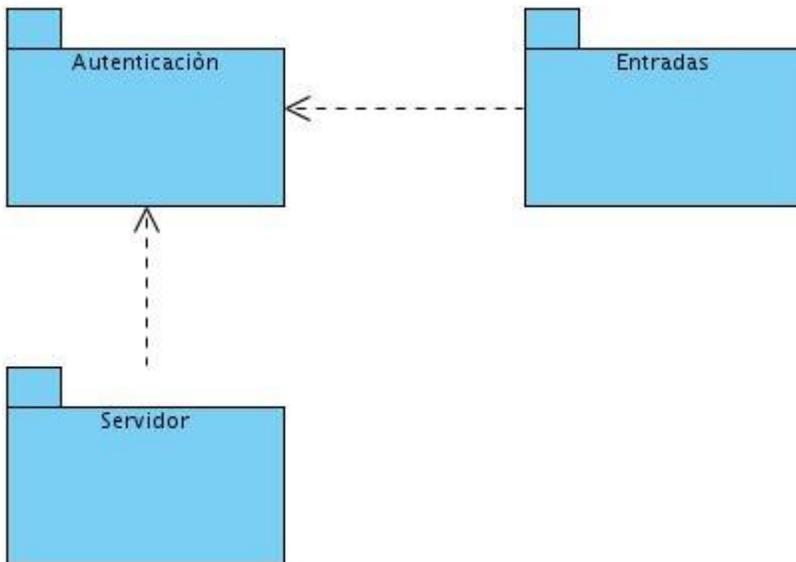


Figura 8: Diagrama de paquetes

2.7.3 Diagrama de Casos de Uso del Sistema (DCUS)



Un diagrama de casos de uso del sistema representa gráficamente a los procesos y su interacción con los actores. Para el Módulo de Directorio v2.0 queda de la siguiente forma:

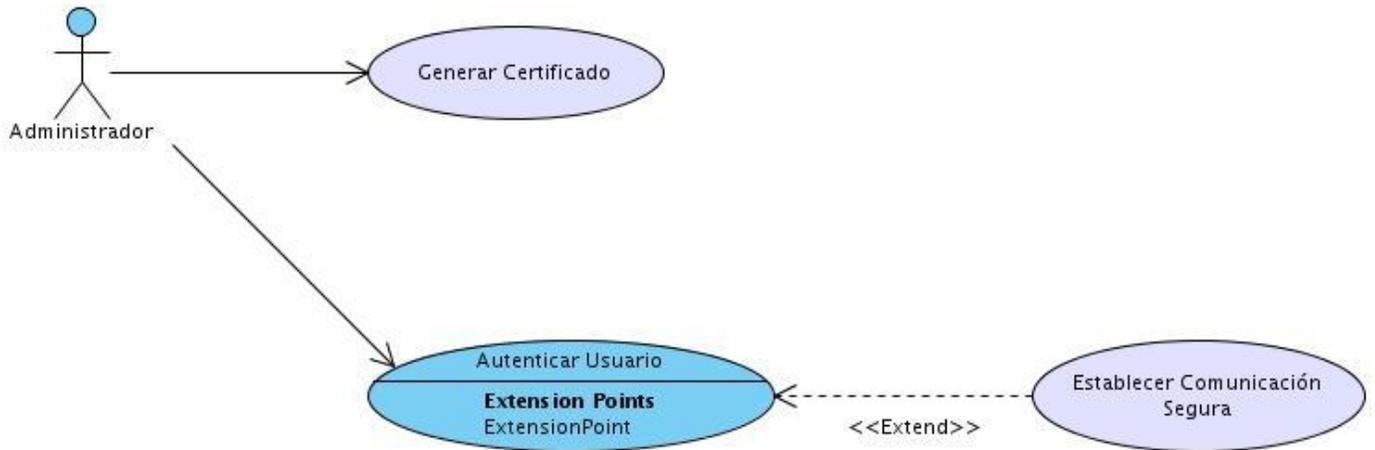


Figura 9: DCUS del paquete Autenticación

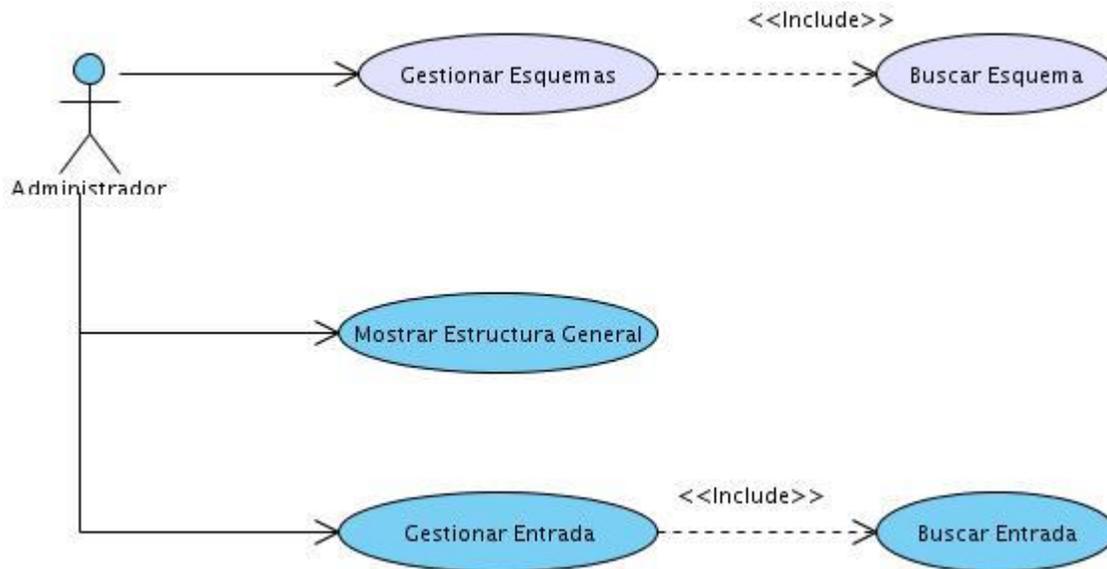


Figura 10: DCUS del paquete Entradas

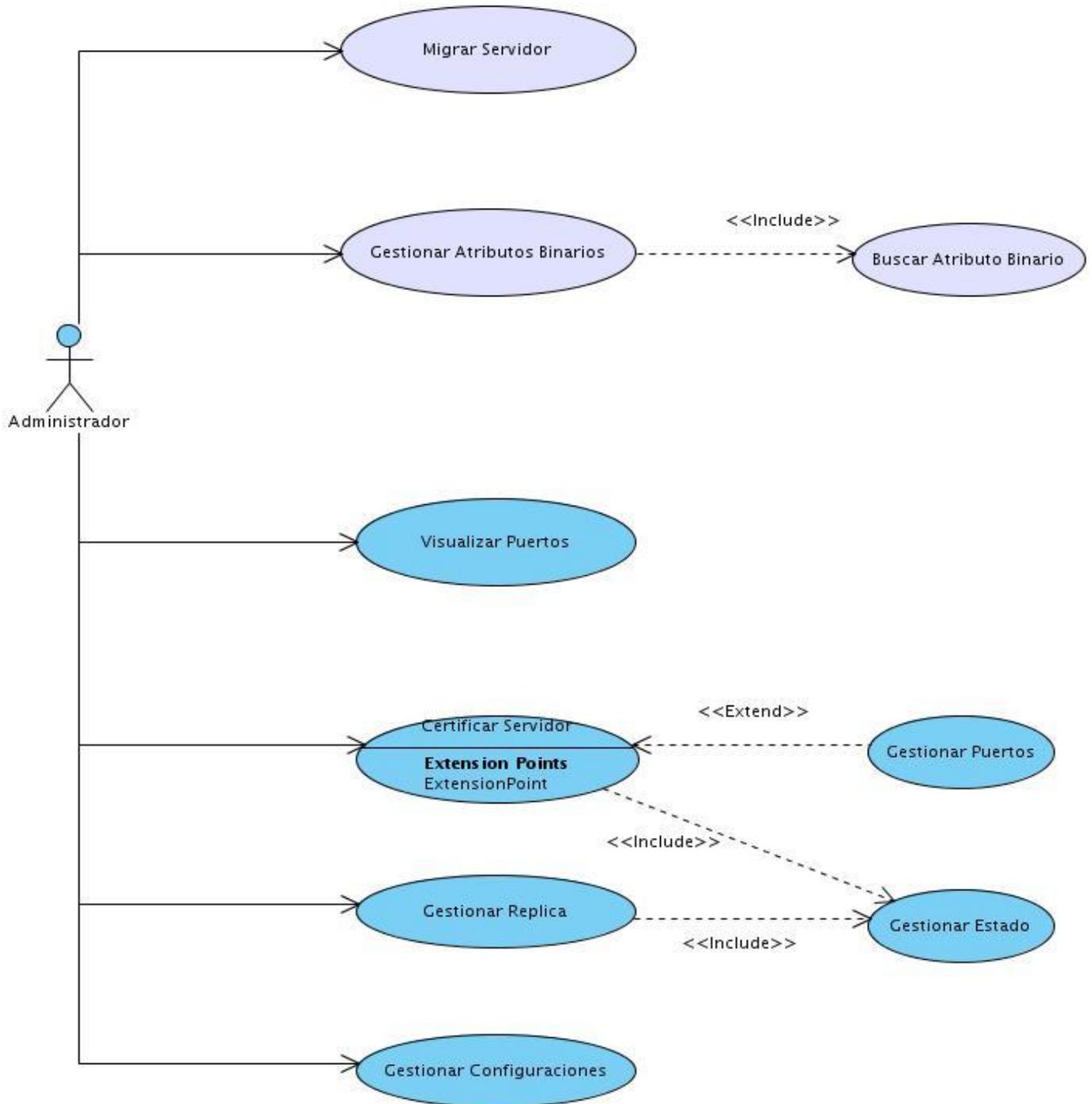


Figura 11: DCUS del paquete Servidor



2.7.4 Descripción de los CUS y prototipos de interfaces

Para entender la funcionalidad asociada a cada caso de uso no es suficiente con la representación gráfica del Diagrama de casos de uso, por lo que se realiza las descripciones de los casos de uso.

1. Gestionar esquemas
2. Migrar servidor
3. Gestionar atributos binarios
4. Establecer comunicación segura
5. Buscar esquema
6. Buscar atributo binario
7. Generar certificado

CU - 1	Gestionar esquemas
Actor	Administrador
Descripción	El CUS consiste en ofrecerle al administrador la posibilidad de agregar, eliminar, modificar o mostrar esquemas que permiten una mejor gestión del servidor y personalización de la información. Para las clases de objetos existe una interfaz específica acorde con los datos que componen a los mismos. En el caso de los atributos cuenta con otra interfaz con sus respectivos datos. Para agregar un esquema es necesario especificar el tipo de esquema, y se muestra la interfaz correspondiente. En la modificación, dado un esquema seleccionado se muestran sus datos, y se brinda la posibilidad de modificarlos. Si se



Capítulo 2 “Características del sistema”

	<p>desea eliminar un esquema es necesario seleccionarlo para ejecutar la acción. Además se muestra todos los esquemas existentes en el servidor. Los esquemas del servidor OpenLDAP se presentan en el Anexo 1.</p>
Referencia	R1

Tabla 2: Descripción CU Gestionar esquemas

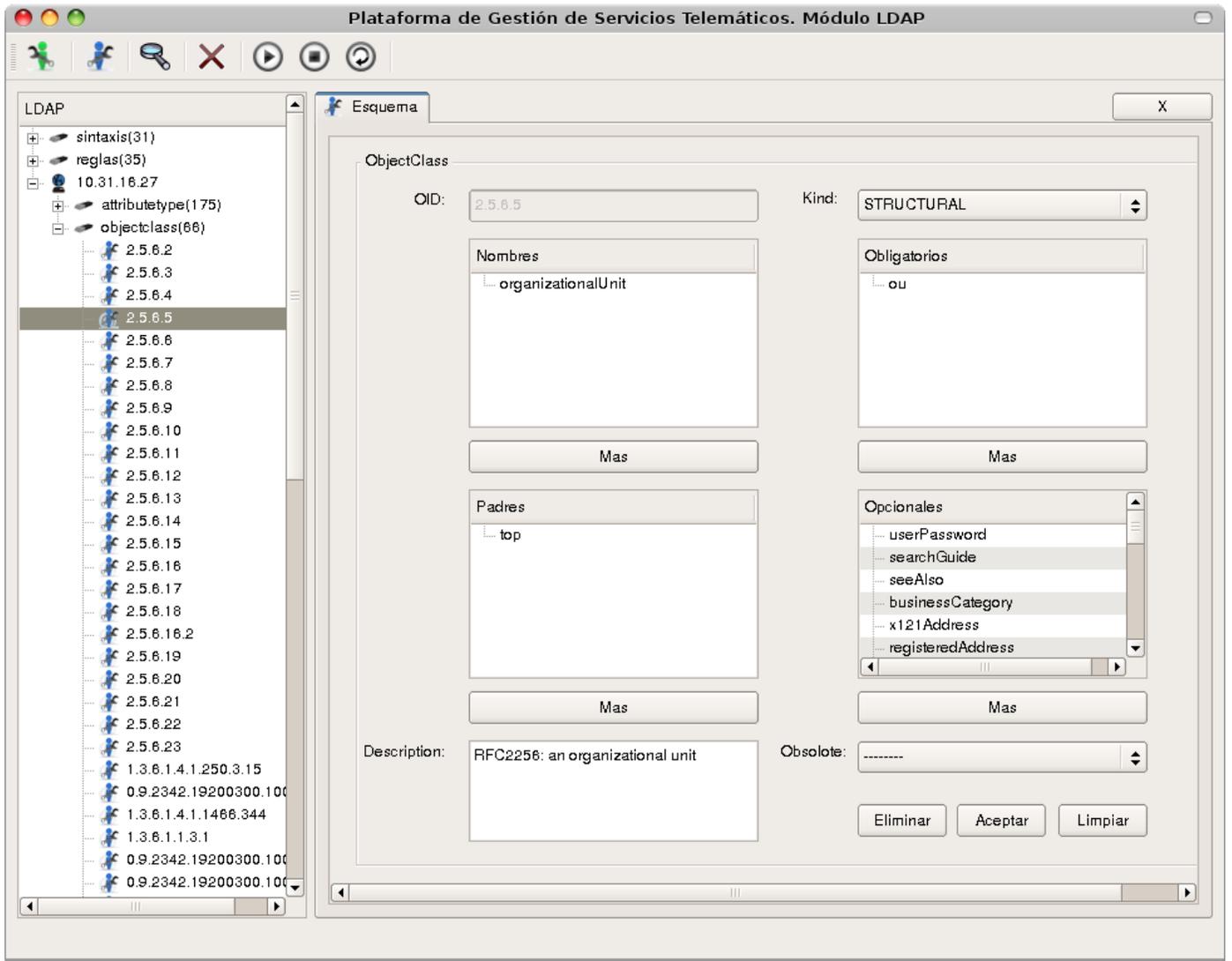


Figura 12: Interfaz del CU Gestionar esquemas (objectclass)

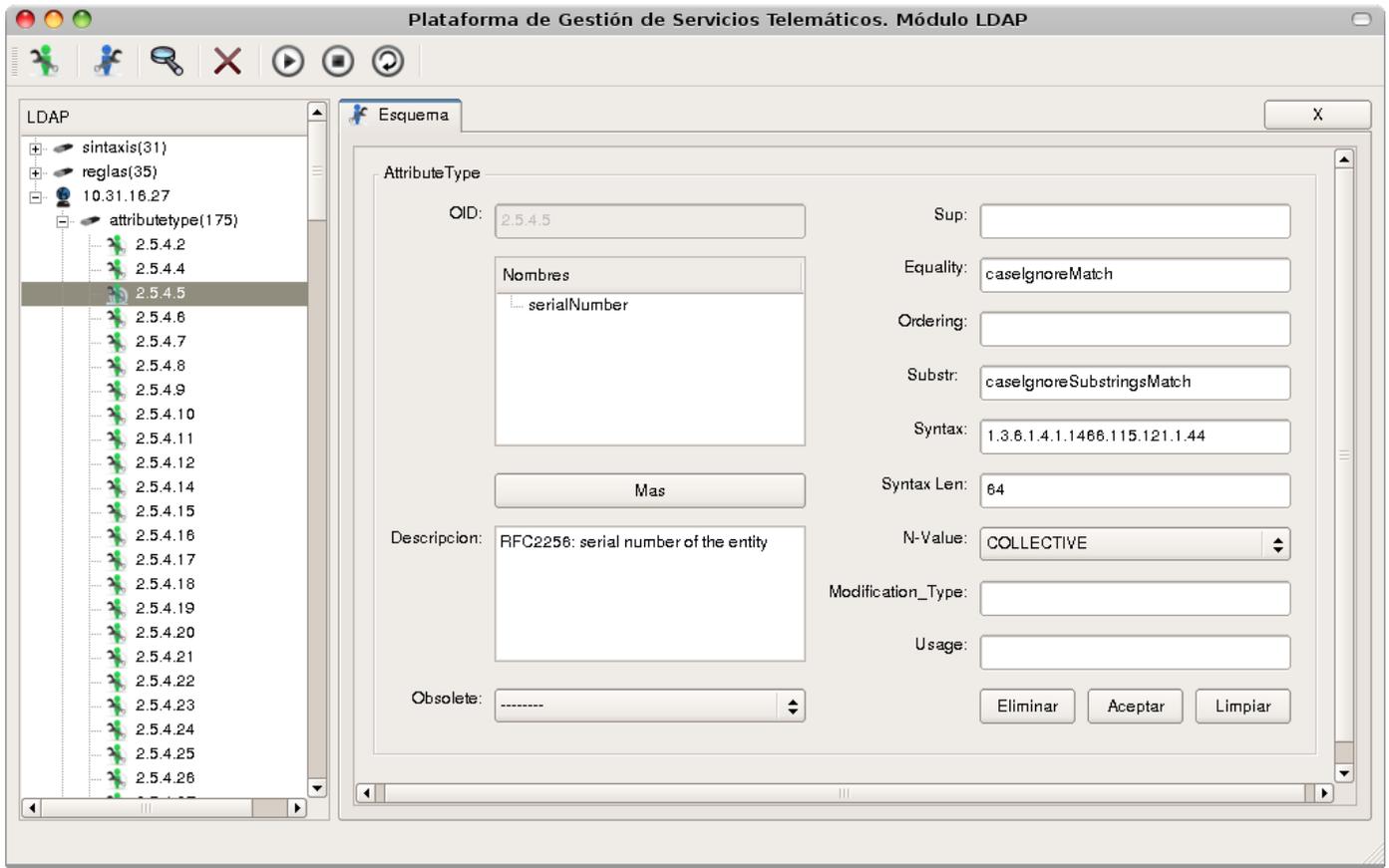


Figura 13: Interfaz del CU Gestionar esquemas (attributetype)

CU - 2	Migrar servidor
Actor	Administrador
Descripción	El CUS consiste en ofrecerle al administrador la posibilidad de migrar los datos del servidor Active Directory de la plataforma Windows al servidor OpenLDAP para garantizar la administración del servicio de directorio desde una plataforma Software Libre. El administrador debe llenar todos los



	campos que necesita el sistema para ejecutar la acción.
Referencia	R2

Tabla 3: Descripción CU Migrar servidor

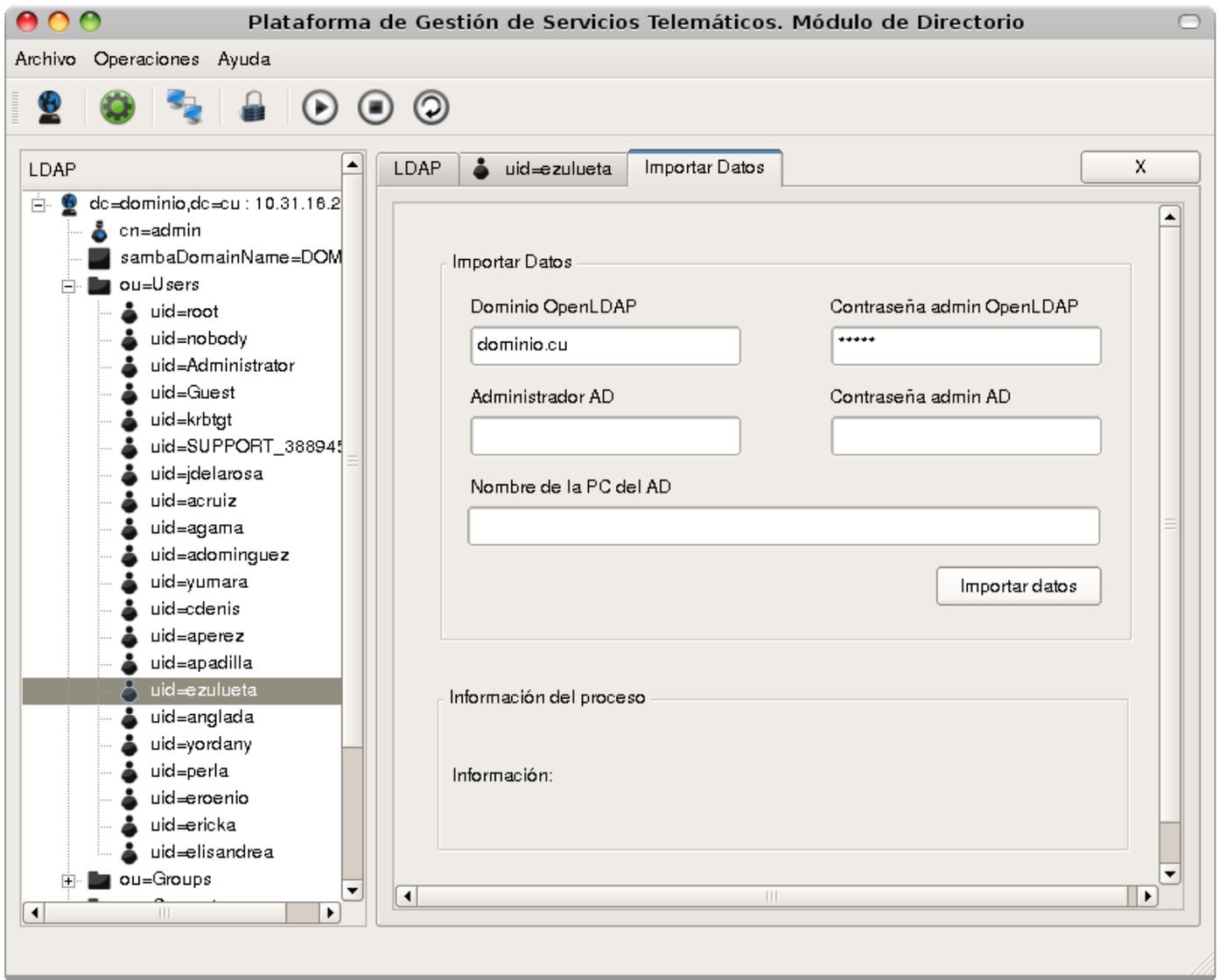


Figura 14: Interfaz del CU Migrar servidor



CU - 3	Gestionar atributos binarios
Actor	Administrador
Descripción	El CUS consiste en ofrecerle al administrador la posibilidad de agregar, eliminar o actualizar imágenes a los usuarios. Además se muestra la imagen correspondiente a un usuario.
Referencia	R3

Tabla 4: Descripción CU Gestionar atributos binarios

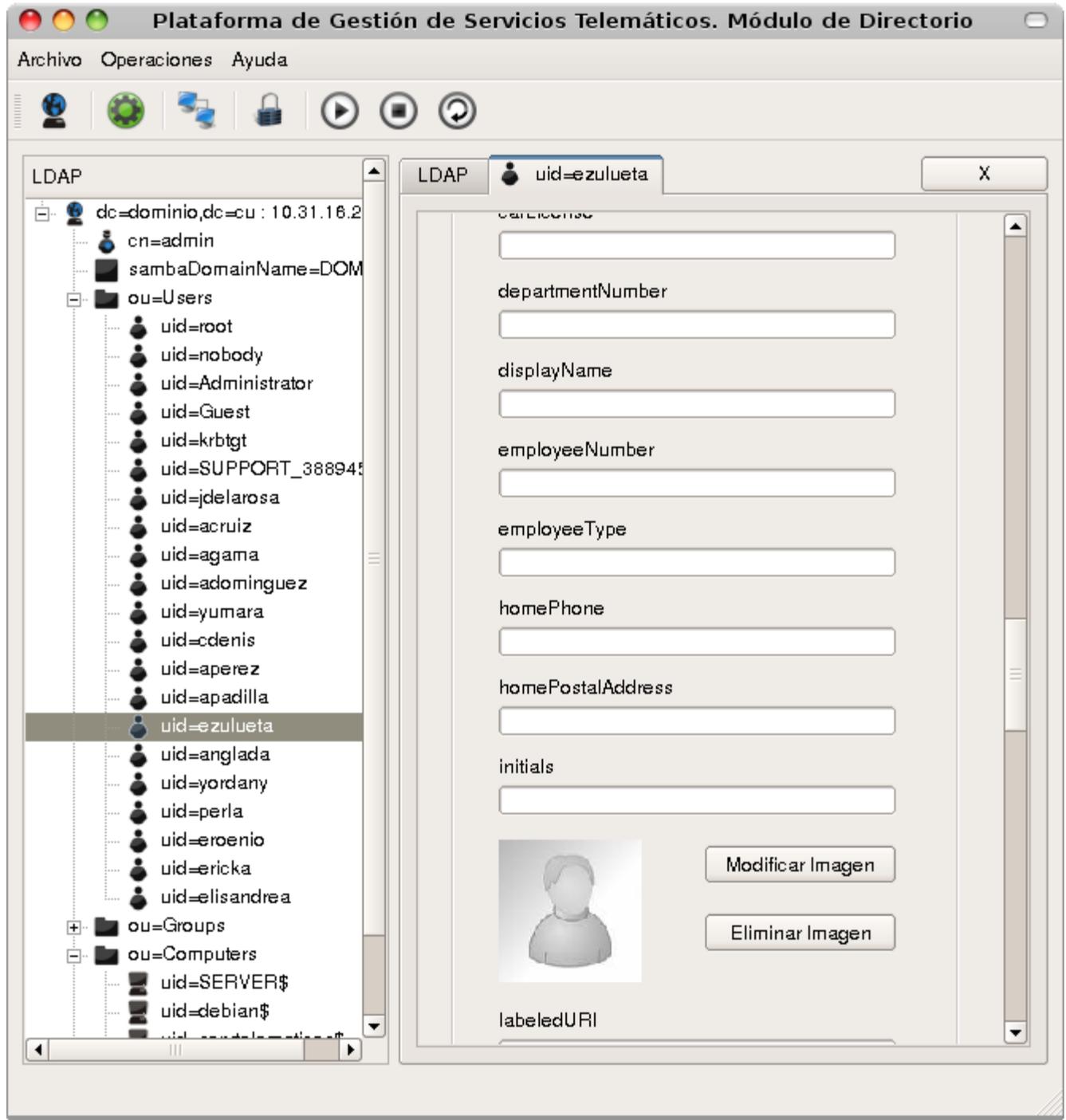


Figura 15: Interfaz del CU Gestionar atributos binarios



CU - 4	Establecer comunicación segura
Actor	Administrador
Descripción	El CUS consiste en ofrecerle al administrador la posibilidad de establecer una conexión segura utilizando un certificado generado.
Referencia	R4, R4.1

Tabla 5: Descripción CU Establecer comunicación segura



Figura 16: Interfaz del CU Establecer comunicación segura

CU - 5	Buscar esquema
Actor	
Descripción	Funcionalidad de apoyo al caso de uso



	Gestionar Esquemas.
Referencia	R5

Tabla 6: Descripción CU Buscar esquema

CU - 6	Buscar atributo binario
Actor	
Descripción	Funcionalidad de apoyo al caso de uso Gestionar atributos binarios.
Referencia	R6

Tabla 7: Descripción CU Buscar atributo binario

CU - 7	Generar certificado
Actor	Administrador
Descripción	El CUS consiste en ofrecerle al administrador la posibilidad de generar un nuevo certificado para establecer comunicación segura.
Referencia	R7

Tabla 8: Descripción CU Generar certificado



Figura 17: Interfaz del CU Generar certificado

2.8 Conclusiones

En el presente capítulo se realizó un estudio de los procesos del negocio que serán automatizados por la herramienta Módulo de Directorio v2.0. Además se identificaron los requisitos funcionales y no funcionales, y posteriormente se diseñaron los casos de uso que guiarán el desarrollo de la solución propuesta. También se describieron los casos de uso del sistema, y se mostraron los prototipos de interfaces.



Capítulo 3 “Diseño del sistema”

3.1 Introducción

Con el objetivo de crear una entrada apropiada y un punto de partida para actividades de implementación, capturando los requisitos, interfaces y clases, se realiza el diseño. Además constituye una actividad de suma importancia para descomponer los trabajos de implementación en partes más manejables. El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades, por lo que hay que prestarle gran atención a su realización.

En el presente capítulo se realizará el Diagrama de clases del Diseño de cada caso de uso.

3.2 Modelo de Diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Sirve de abstracción de la implementación y es utilizada como entrada fundamental de las actividades de implementación.

La arquitectura escogida va estrechamente ligada al diseño de un sistema. En este caso la Arquitectura en Capas cuenta con disímiles ventajas como organizar el modelo de diseño en capas, que pueden estar físicamente distribuidas, lo cual se refleja en que los componentes de una capa sólo pueden hacer referencia a componentes en capas inmediatamente inferiores. Este patrón es importante porque simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no son conscientes de ningún detalle o interfaz de las superiores. Además, ayuda a identificar qué puede reutilizarse.

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

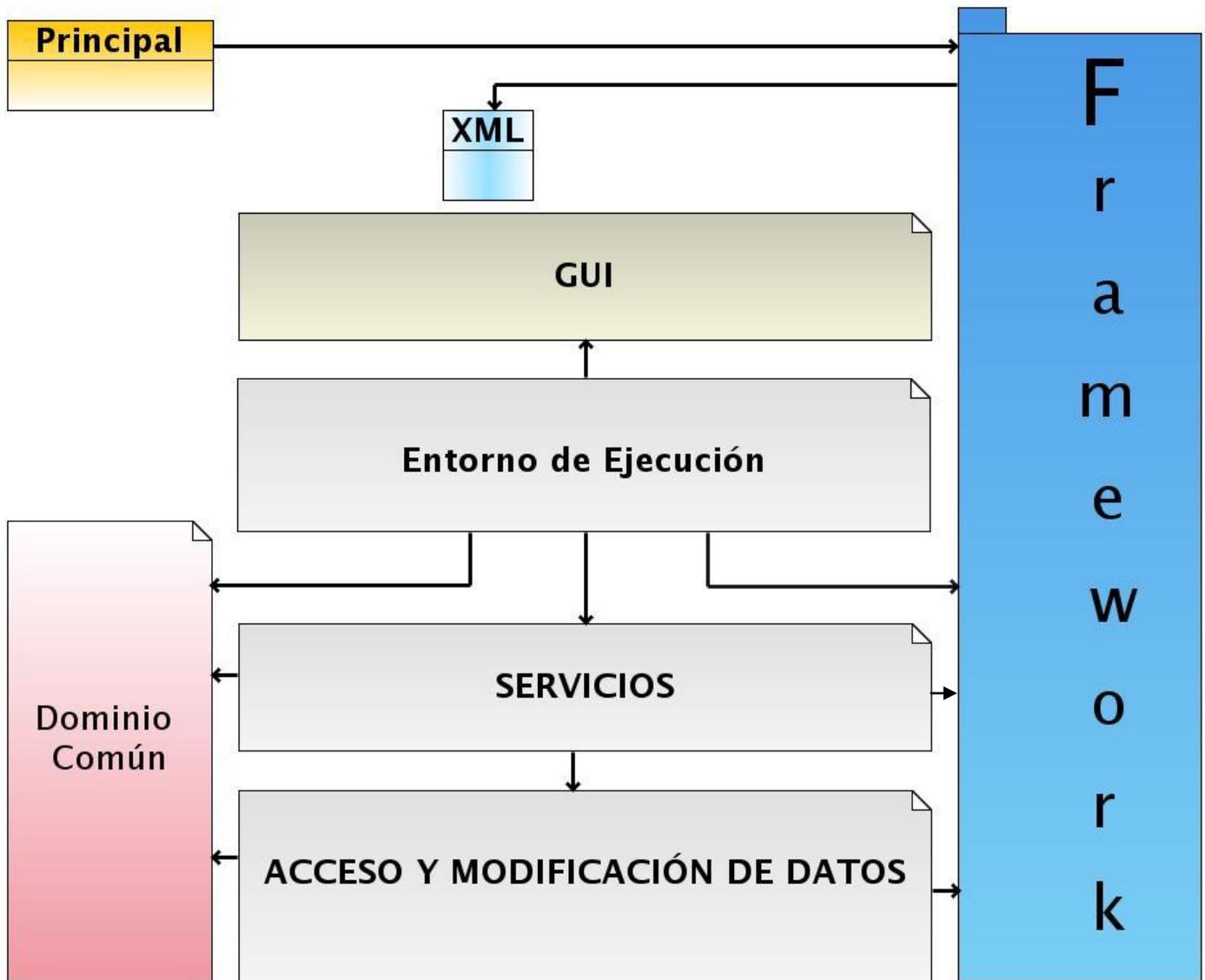


Figura 18: Estructura general del sistema

Existen cuatro capas principales las cuales son: GUI, Entorno de Ejecución, Servicios, y Acceso y Modificación de Datos. En el caso de la capa Entorno de Ejecución constituye una capa de abstracción entre GUI y Servicios. Es importante aclarar que se ha decidido construir un *framework* que se encargue de resolver problemas puntuales relacionados con la inicialización del sistema, la ejecución de funciones en paralelo, las conexiones remotas, entre otros aspectos; con el objetivo de poder reutilizar estas



funcionalidades en futuros desarrollos de software y darle al desarrollador la posibilidad de concentrarse en aspectos puramente de negocio que se encuentran a un nivel superior de abstracción. La configuración de la estructura general del sistema estará reflejada en un archivo XML que será utilizado por el *framework* para conocer que componentes serán empleados y donde residirá su implementación particular. Otro aspecto importante es el funcionamiento del componente *Dominio Común*, el cual contiene entidades o atributos que serán reutilizados por más de una clase en el sistema.

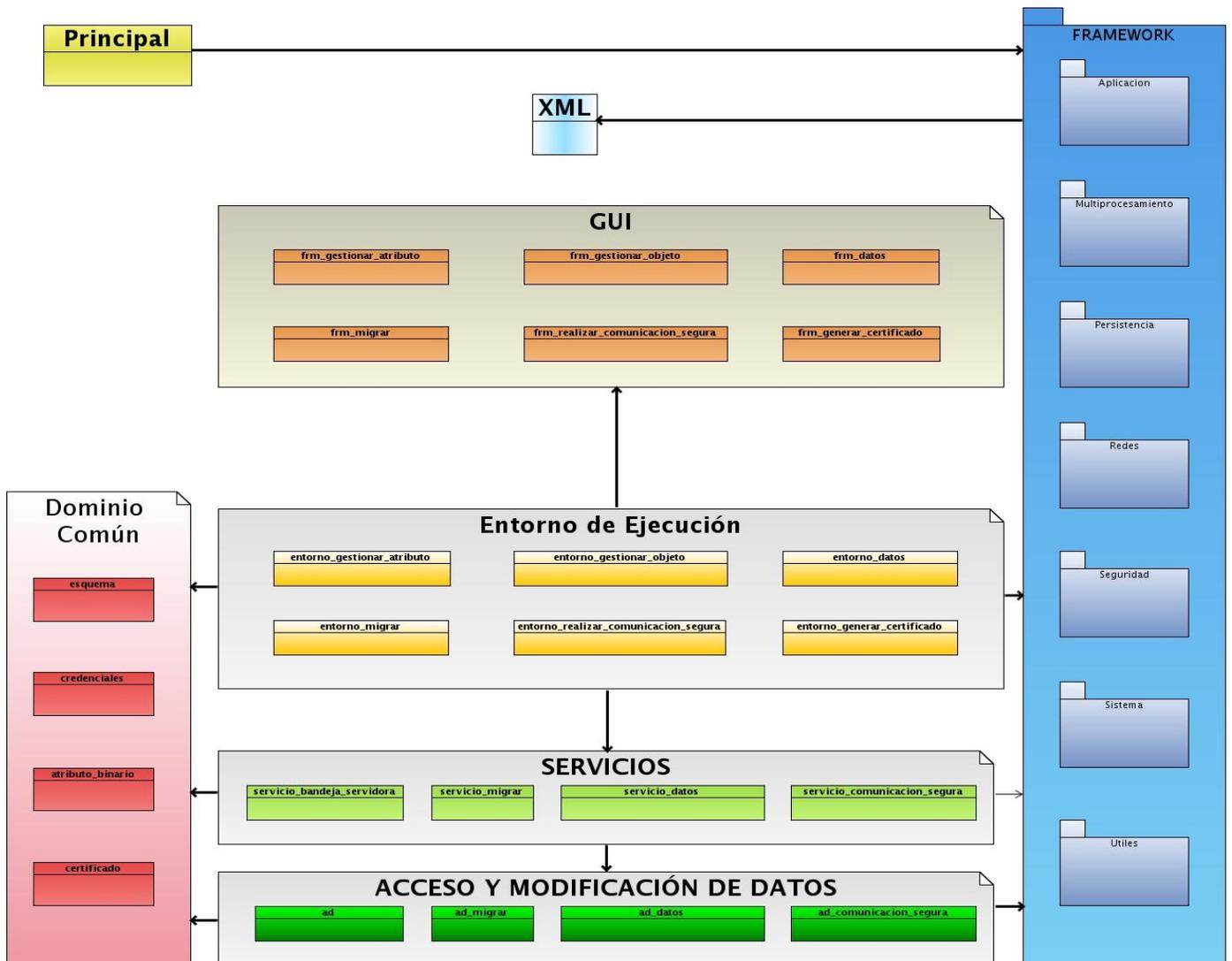


Figura 19: Estructura detallada del sistema

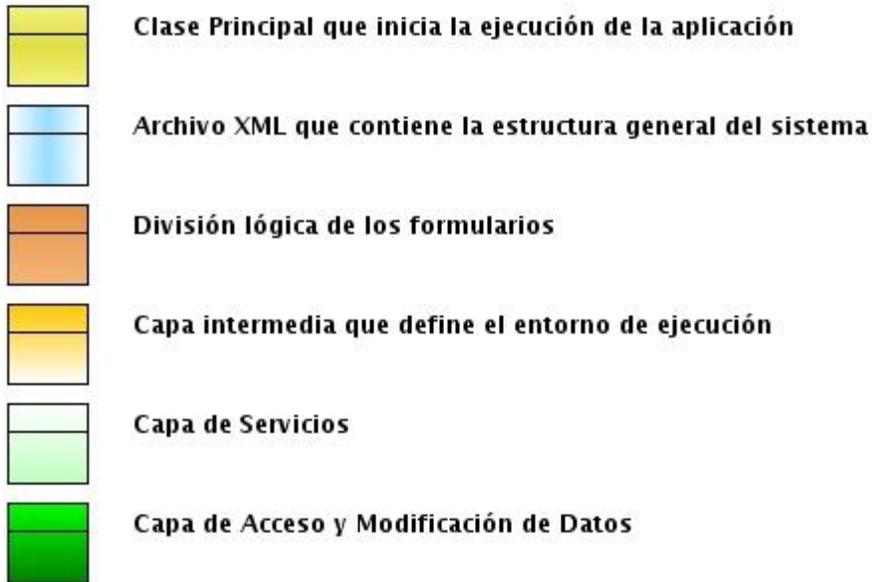


Figura 20: Definición de los términos fundamentales

3.2.1 Diagrama de Clases del Diseño

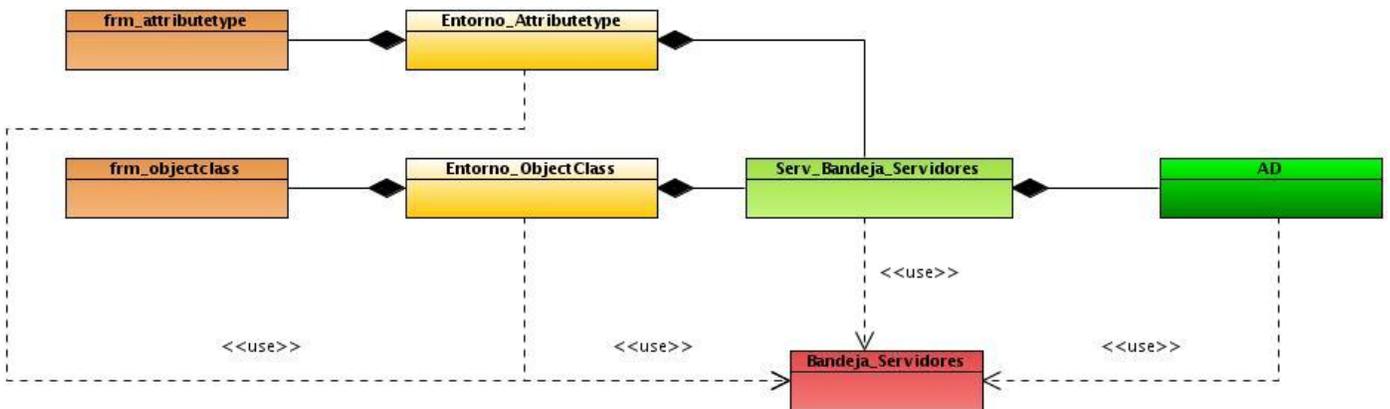


Figura 21: Diagrama de Clases del Diseño: CU Gestionar Esquemas

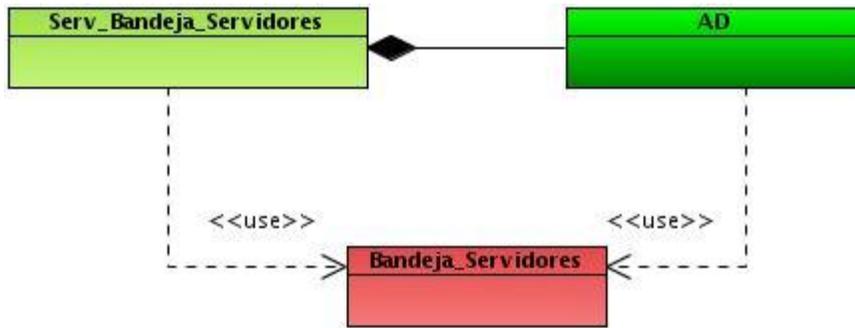


Figura 22: Diagrama de Clases del Diseño: CU Buscar Esquema

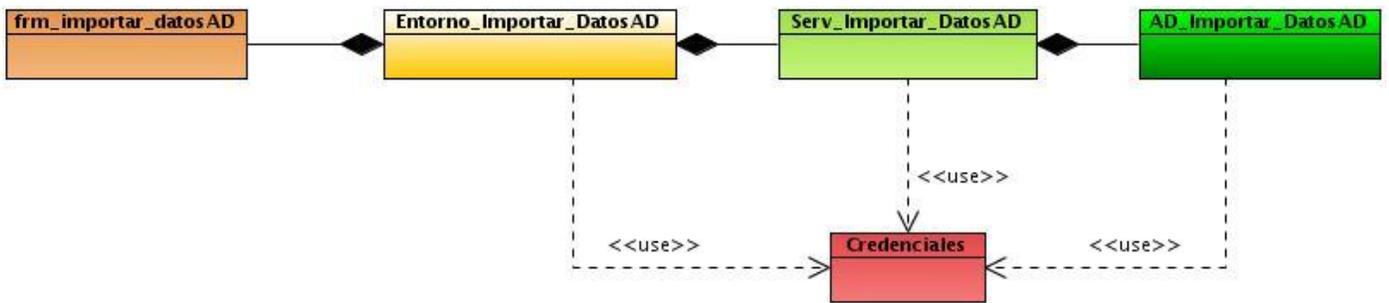


Figura 23: Diagrama de Clases del Diseño: CU Migrar servidor

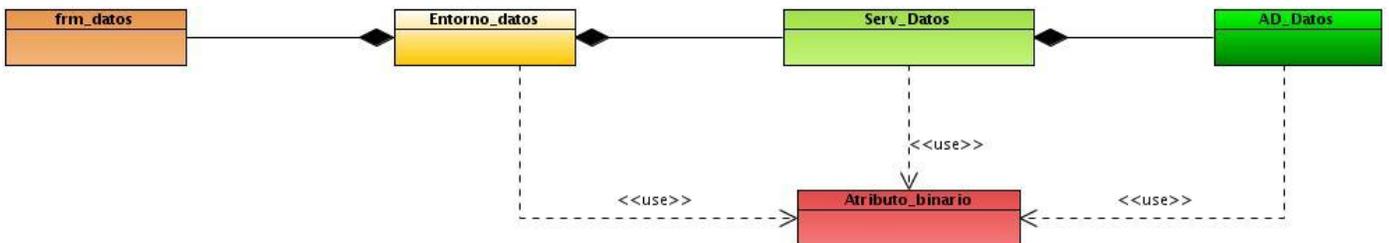


Figura 24: Diagrama de Clases del Diseño: CU Gestionar atributos binarios

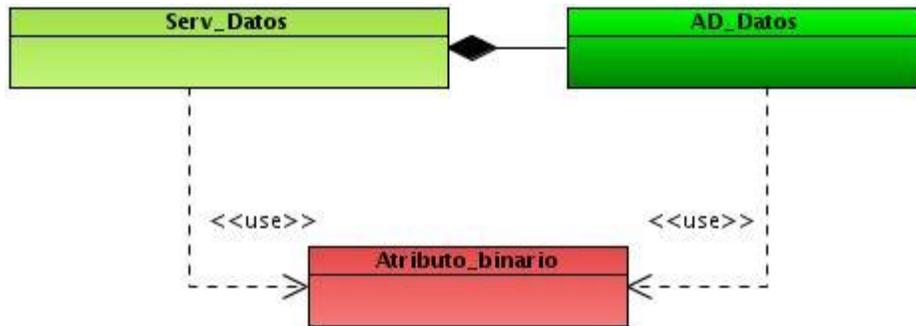


Figura 25: Diagrama de Clases del Diseño: CU Buscar atributo binario

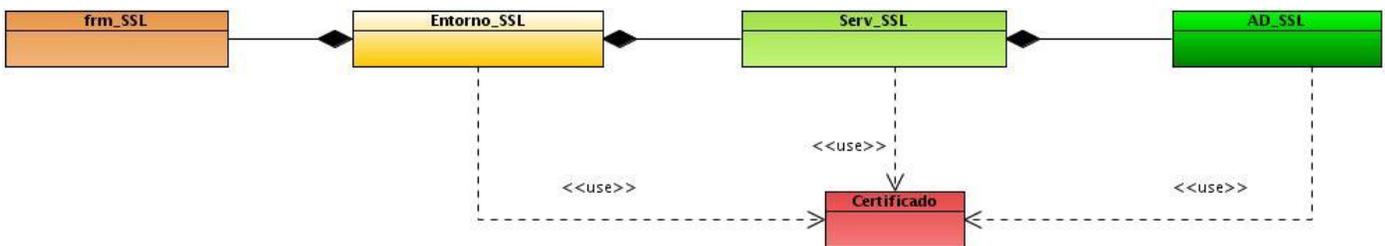


Figura 26: Diagrama de Clases del Diseño: CU Generar certificado



Figura 27: Diagrama de Clases del Diseño: CU Establecer comunicación segura

3.3 Conclusiones

En el capítulo se realizó una explicación de los principales elementos que componen la arquitectura escogida, así como el diagrama de clases del Diseño de cada caso de uso. Una buena realización del diseño abre las puertas para los pasos siguientes en la implementación, por lo que se prestó suma atención a esta fase.



Capítulo 4 “Implementación”

4.1 Introducción

El presente capítulo presenta los artefactos correspondientes a la implementación del sistema. El flujo de trabajo de implementación describe cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo éstos se organizan de acuerdo a los nodos específicos en el modelo de despliegue. Los diagramas de despliegue y componentes conforman lo que se conoce como un modelo de implementación al describir los componentes a construir y su organización y dependencia entre nodos físicos en los que funcionará la aplicación.

4.2 Diagrama de despliegue

El modelo de despliegue describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación. En general un nodo será una unidad de computación de algún tipo.

A continuación se muestra el diagrama de despliegue donde los principales elementos son:

- PC Cliente: ordenador desde donde se ejecutarán las operaciones de gestión sobre el servidor.
- Servidor de directorio: ordenador que contiene instalado el servidor de directorio OpenLDAP.



Figura 28: Diagrama de Despliegue del sistema

4.3 Diagrama de componentes

Un diagrama de componentes muestra las dependencias lógicas entre componentes del software.



- El paquete GUI contiene la definición e implementación de los formularios que se utilizan, y dentro se encuentra el paquete UI con los respectivos formularios.
- El paquete Entornos contiene los entornos correspondientes a los formularios que constituyen intermediarios entre GUI y Servicios.
- El paquete Servicios contiene los componentes del negocio de la aplicación.
- El paquete AD contiene componentes encargados de acceder y modificar los datos del servidor.
- El paquete “Dominio común” contiene un grupo de componentes que son reutilizados por la mayoría de las capas del sistema.

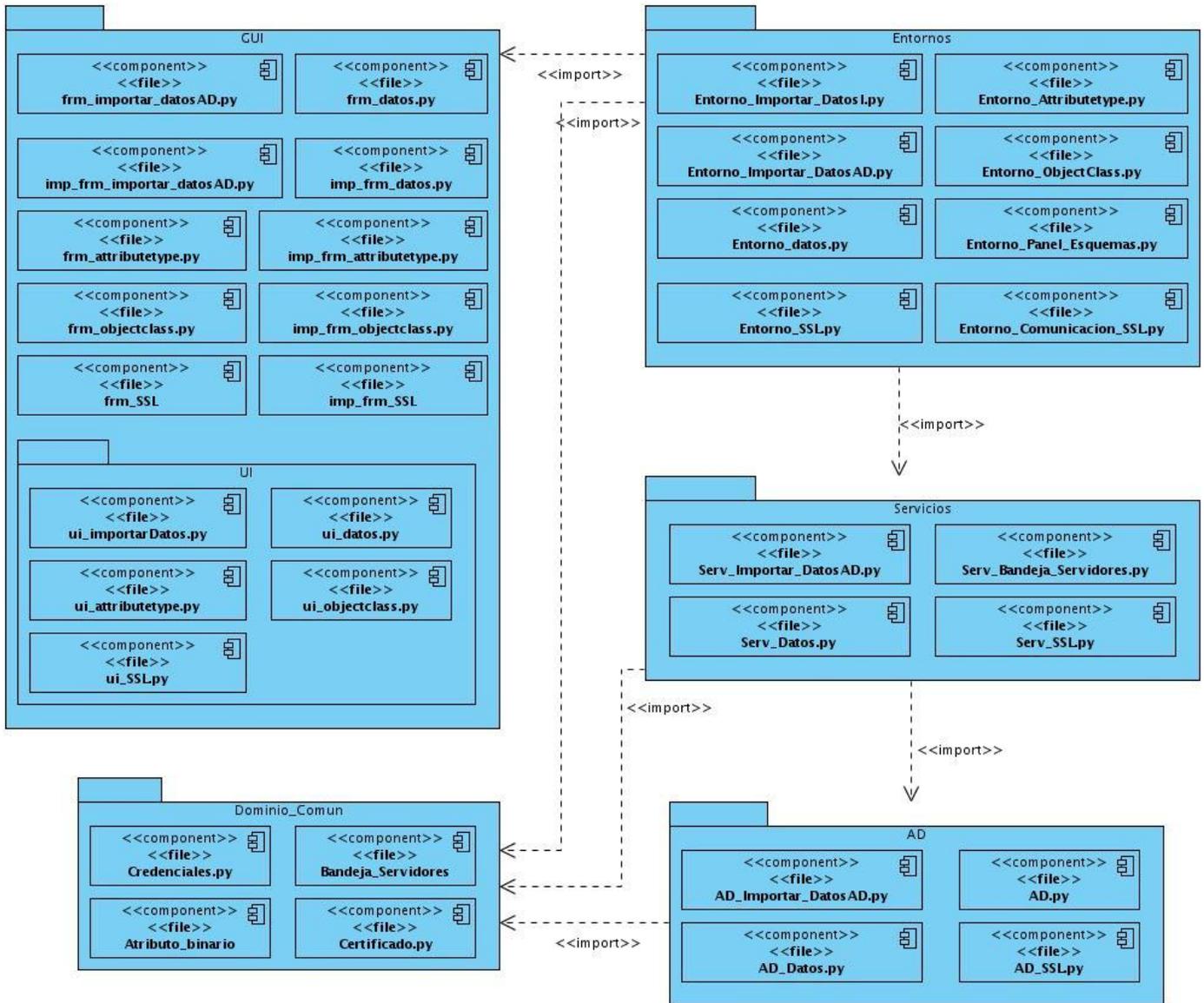


Figura 29: Diagrama General de Componentes

A continuación se muestran los diagramas de componentes correspondientes a cada caso de uso del Módulo de Directorio v2.0.

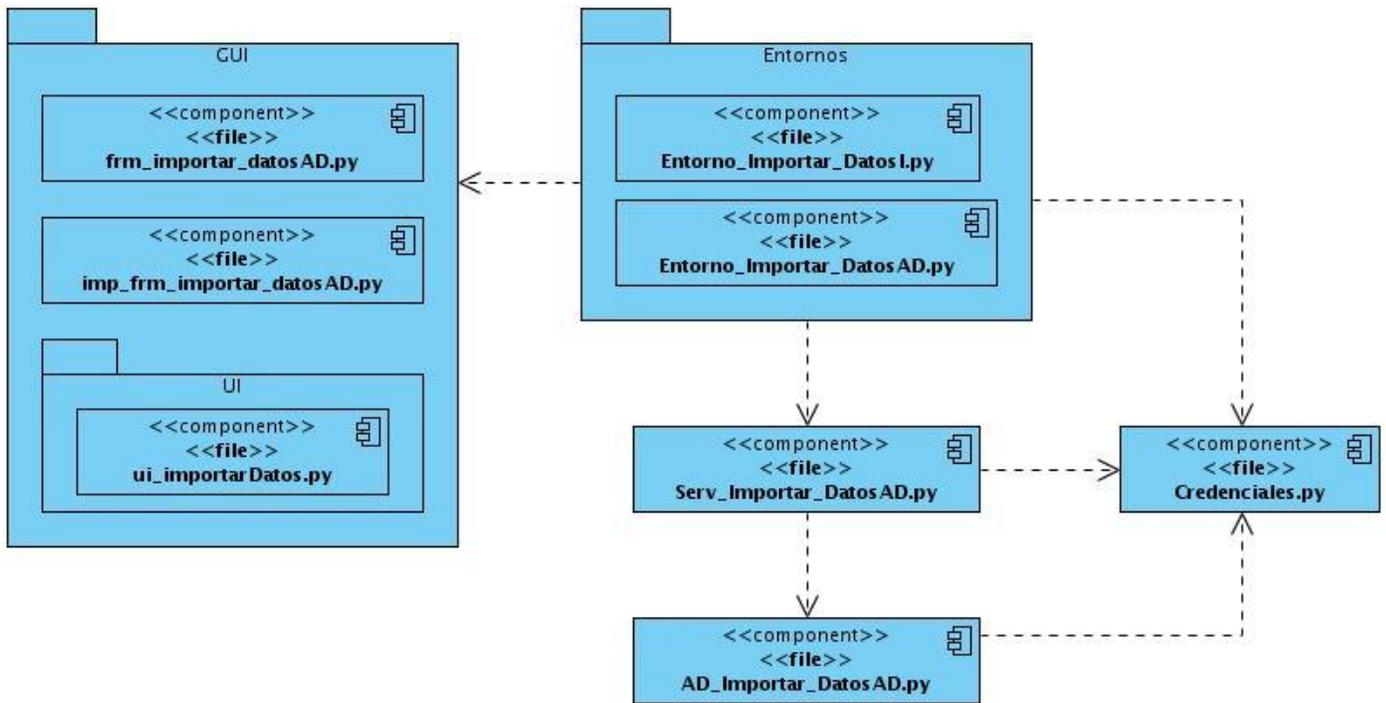


Figura 30: Diagrama de Componentes: CU Migrar servidor

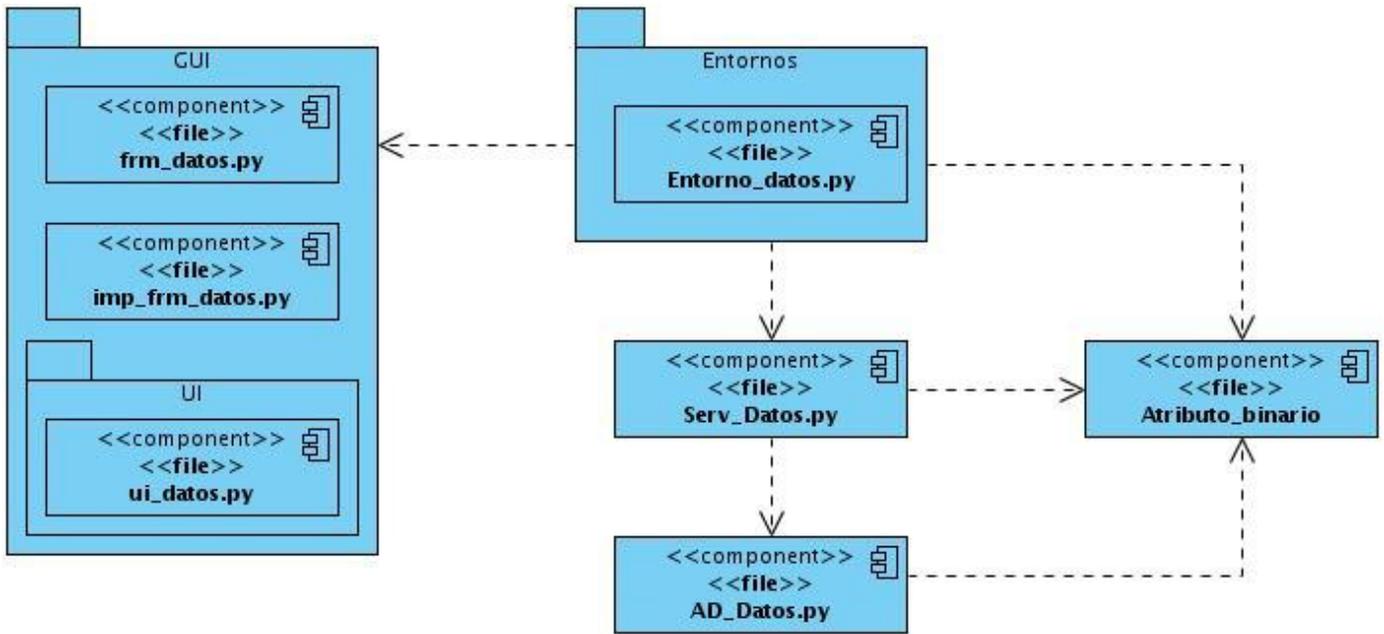


Figura 31: Diagrama de Componentes: CU Gestionar atributos binarios

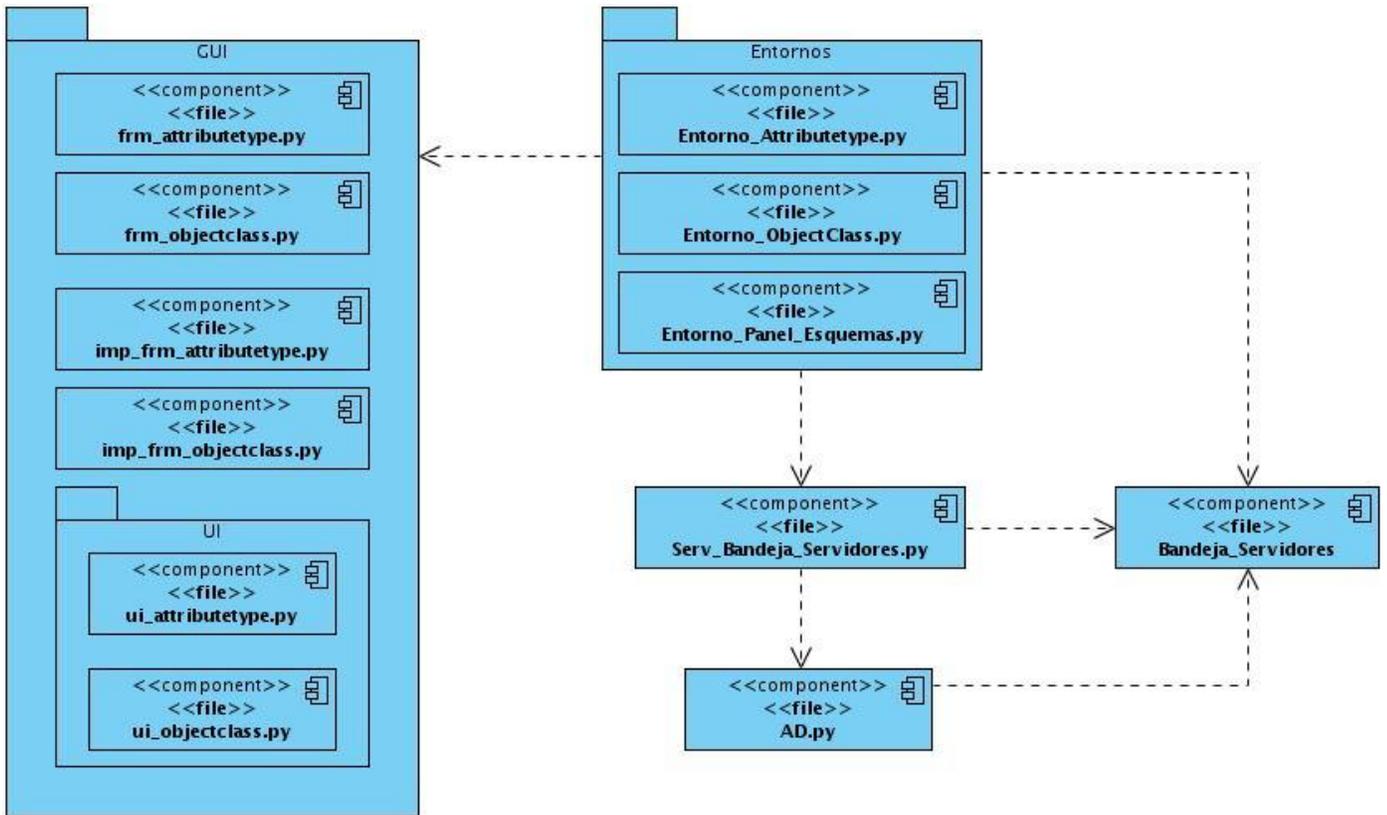


Figura 32: Diagrama de Componentes: CU Gestionar esquemas

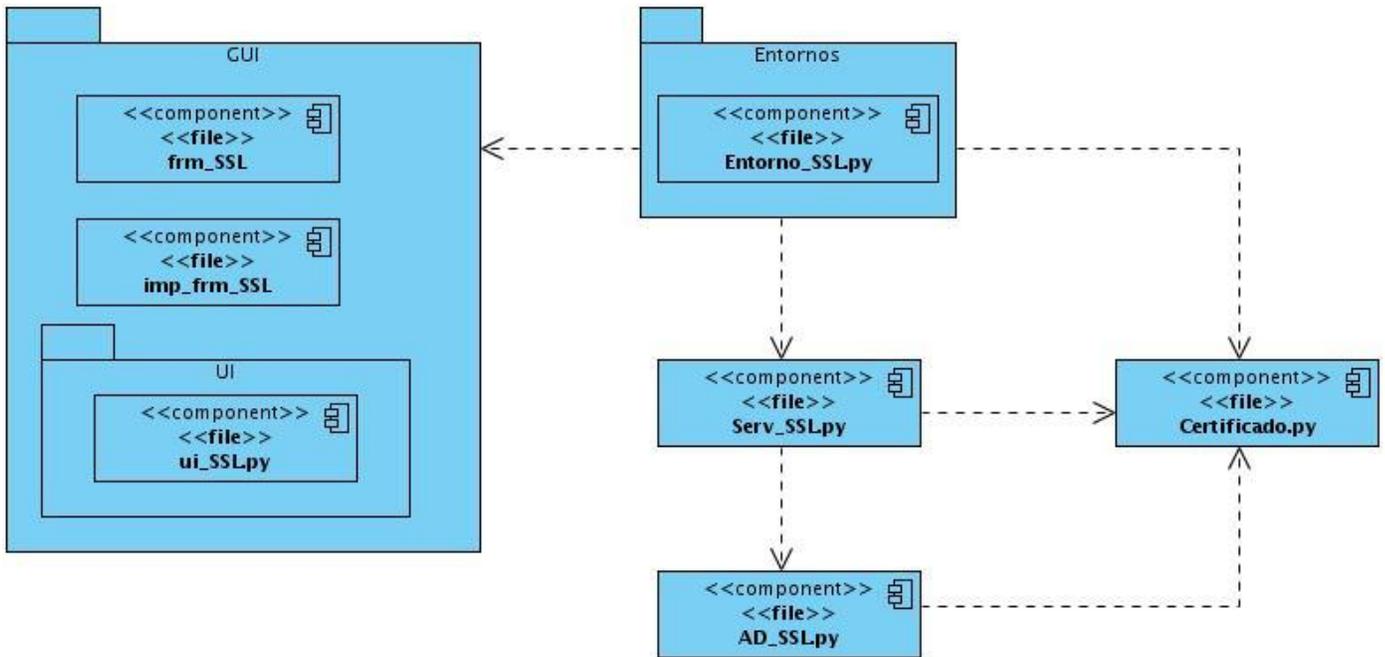


Figura 33: Diagrama de Componentes: CU Generar certificado

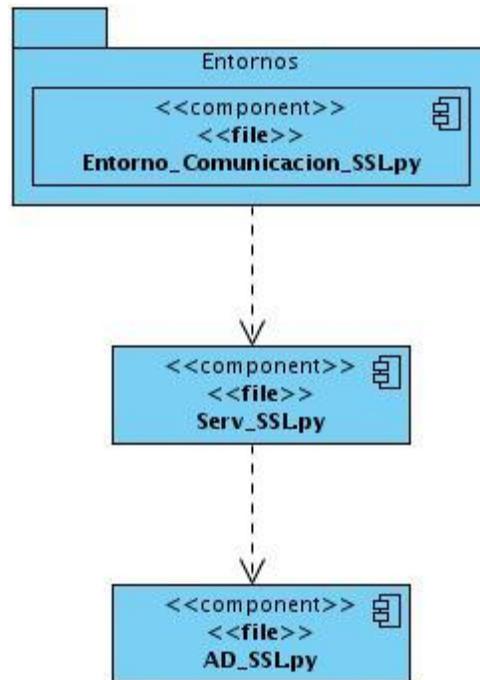


Figura 34: Diagrama de Componentes: CU Establecer comunicación segura

4.4 Conclusiones

En el presente capítulo se mostró los elementos correspondientes al modelo de implementación: diagrama de despliegue, y los diagramas de componentes correspondiente a cada caso de uso. De esta forma se completan los artefactos fundamentales del Módulo de Directorio v2.0.



Conclusiones

El servicio de directorio constituye actualmente uno de los principales servicios en una red de computadoras. Para la gestión de dicho servicio de una forma cómoda, fácil e intuitiva existen un conjunto de herramientas. Sin embargo, un gran número de herramientas son privativas, lo cual no constituye una opción viable para Cuba, por lo que desde abril del 2004 se dispuso la migración progresiva hacia Software Libre. El presente trabajo de diploma contribuye a dicho proceso de migración proporcionando al administrador cubano una herramienta que permite la gestión del servicio de directorio sobre la plataforma de Software Libre, específicamente un servidor OpenLDAP.

En el transcurso de este trabajo se realizó el estado del arte de sistemas similares para el estudio de sus características, así como la investigación y desarrollo de las nuevas funcionalidades del Módulo de Directorio, y su respectiva documentación, por lo que se consideran cumplidos los objetivos trazados al inicio. Se espera sea de gran ayuda en el proceso de migración del servicio de directorio en Cuba.



Recomendaciones

Se recomienda:

Seguir en la investigación de las políticas de grupo para su respectiva migración e incorporárselo a la herramienta Módulo de Directorio v2.0, así como un editor de políticas.

Extender la interfaz de administración a prototipos web y de consolas, reutilizando las funcionalidades, dándole al administrador más de una forma de realizar su labor de acuerdo a sus preferencias o condiciones específicas del entorno.

Incorporar a la herramienta la posibilidad de realizar búsquedas utilizando patrones complejos.



Referencias Bibliográficas

1. openldap.org. [Online] [Cited: 01 20, 2010.]
<http://www.openldap.org/doc/admin24/guide.html#What%20is%20LDAP>.
2. **James Rumbaugh, Ivar Jacobson, Grady Booch.** bibliodoc.uci.cu. [Online] [Cited: 01 22, 2010.]
<http://bibliodoc.uci.cu/pdf/reg03050.pdf>.
3. **Guarda, Alfonso de la.** dev.laptop.org. [Online] [Cited: 01 25, 2010.]
http://dev.laptop.org/~edsiper/byteofpython_spanish/ch01s02.html.
4. visual-paradigm.com. [Online] [Cited: 01 25, 2010.] <http://www.visual-paradigm.com/product/vpuml/> .
5. easyeclipse.org. [Online] [Cited: 01 26, 2010.] <http://www.easyeclipse.org/site/home/> .
6. di-mare.com. [Online] [Cited: 01 27, 2010.] <http://www.di-mare.com/adolfo/cursos/2007-1/pp-GenGUI.pdf>.
7. elai.upm.es. [Online] [Cited: 01 27, 2010.]
<http://www.elai.upm.es/spain/Investiga/GCII/personal/vcorte/informeqt.PDF>.
8. revistauxi.files.wordpress.com. [Online] [Cited: 01 29, 2010.]
http://revistauxi.files.wordpress.com/2009/01/uxi08_v02.pdf.



Bibliografía

1. openldap.org. [Online] [Cited: 01 20, 2010.]
<http://www.openldap.org/doc/admin24/guide.html#What%20is%20LDAP>.
2. **James Rumbaugh, Ivar Jacobson, Grady Booch.** bibliodoc.uci.cu. [Online] [Cited: 01 22, 2010.]
<http://bibliodoc.uci.cu/pdf/reg03050.pdf>.
3. **Guarda, Alfonso de la.** dev.laptop.org. [Online] [Cited: 01 25, 2010.]
http://dev.laptop.org/~edsiper/byteofpython_spanish/ch01s02.html.
4. visual-paradigm.com. [Online] [Cited: 01 25, 2010.] <http://www.visual-paradigm.com/product/vpuml/> .
5. easyeclipse.org. [Online] [Cited: 01 26, 2010.] <http://www.easyeclipse.org/site/home/> .
6. di-mare.com. [Online] [Cited: 01 27, 2010.] <http://www.di-mare.com/adolfo/cursos/2007-1/pp-GenGUI.pdf>.
7. elai.upm.es. [Online] [Cited: 01 27, 2010.]
<http://www.elai.upm.es/spain/Investiga/GCII/personal/vcorte/informeqt.PDF>.
8. revistauxi.files.wordpress.com. [Online] [Cited: 01 29, 2010.]
http://revistauxi.files.wordpress.com/2009/01/uxi08_v02.pdf.
9. **González, Sergio González.** www.ibiblio.org. [Online] 2004.
<http://www.ibiblio.org/pub/linux/docs/LuCaS/Tutoriales/doc-openldap-samba-cups-python/htmls/index.html>.
10. **Suárez, Jose Manuel.** www.goa.es. [Online] 11 3, 2004. http://www.goa.es/docs/curso_openldap.pdf.
11. **Hernández, Amaury Viera and Gómez, Leonardo de la Rosa.** *MigrateAD: Migración del Directorio Activo a plataforma libre.* Ciudad de la Habana : s.n., 2009.
12. **Cabrera, Sandy Noa.** *Plataforma de Gestión de Servicios Telemáticos en GNU/Linux. Módulo de Directorio.* Ciudad de la Habana : s.n., 2009.
13. www.zytrax.com. [Online] <http://www.zytrax.com/books/ldap/>.
14. **Hernández, Amaury Viera.** amauryvh.wordpress.com. [Online] 2007. <http://amauryvh.wordpress.com/>.
15. www.esdebian.org. [Online] <http://www.esdebian.org/wiki/controlador-primario-dominio-pdc-debian-lenny-50-mediante-samba-pamNSS-openldap>.



16. **Terpstra, John H.** <http://samba.org/samba/docs/man/Samba-Guide/ntmigration.html>. [Online] <http://samba.org/samba/docs/man/Samba-Guide/ntmigration.html>.
17. —. osr600doc.sco.com. [Online] http://osr600doc.sco.com/en/INT_sambaDocGuide/migration.html.
18. www.python.org. [Online] <http://www.python.org/doc/>.
19. **Duque, Raúl González.** *Python para todos*.
20. **Brittain, Jason and Darwin, Ian F.** *Python for Unix and Linux System Administration*. s.l. : O'Reilly Media, Inc., 2008. 978-0-596-51582-9.



Anexos

Anexo 1: Esquemas del servidor de directorio OpenLDAP

AttributeTypes	
OID	Nombre
1.3.6.1.4.1.4203.1.12.2.3.0.9	olcbackend
1.3.6.1.4.1.4203.1.12.2.3.0.8	olcauthzregexp
0.9.2342.19200300.100.1.4	info
0.9.2342.19200300.100.1.5	drink
0.9.2342.19200300.100.1.5	favouritedrink
0.9.2342.19200300.100.1.2	textencodedoraddress
1.3.6.1.4.1.4203.1.12.2.3.0.66	olcthreads
0.9.2342.19200300.100.1.1	uid
0.9.2342.19200300.100.1.1	userid
1.3.6.1.4.1.4203.1.12.2.3.0.1	olcaccess
1.3.6.1.1.1.1.10	shadowexpire
1.3.6.1.4.1.4203.1.12.2.3.0.3	olcargsfile
1.3.6.1.4.1.4203.1.12.2.3.0.2	olcallows
1.3.6.1.4.1.4203.1.12.2.3.0.5	olcattributeoptions
1.3.6.1.4.1.4203.1.12.2.3.0.4	olcattributetypes
1.3.6.1.4.1.4203.1.12.2.3.0.7	olcauthzpolicy



1.3.6.1.4.1.4203.1.12.2.3.0.6	olcauthidrewrite
2.5.4.8	st
2.5.4.8	stateorprovincename
1.3.6.1.4.1.4203.1.12.2.3.0.48	olcrestrict
1.3.6.1.4.1.4203.1.12.2.3.0.49	olcreverselookup
1.3.6.1.4.1.4203.1.12.2.3.0.46	olcreplogfile
1.3.6.1.4.1.4203.1.12.2.3.0.47	olcrequires
1.3.6.1.4.1.4203.1.12.2.3.0.44	olcreplicapidfile
1.3.6.1.4.1.4203.1.12.2.3.0.45	olcreplicationinterval
1.3.6.1.4.1.4203.1.12.2.3.0.43	olcreplicaargsfile
1.3.6.1.4.1.4203.1.12.2.3.0.40	olcreadonly
1.3.6.1.4.1.4203.1.12.2.3.0.41	olcreferral
0.9.2342.19200300.100.1.37	associateddomain
0.9.2342.19200300.100.1.50	singlelevelquality
0.9.2342.19200300.100.1.51	subtreeminimumquality
0.9.2342.19200300.100.1.25	dc
0.9.2342.19200300.100.1.25	domaincomponent
2.16.840.1.113730.3.1.40	usersmimecertificate
2.5.18.10	subschemasubentry
1.3.6.1.1.20	entrydn
1.3.6.1.1.1.1.12	memberuid
0.9.2342.19200300.100.1.30	soarecord



2.5.4.19	physicaldeliveryofficename
0.9.2342.19200300.100.1.31	cnamerecord
0.9.2342.19200300.100.1.6	roomnumber
1.2.840.113549.1.9.1	email
1.2.840.113549.1.9.1	emailaddress
1.2.840.113549.1.9.1	pkcs9email
1.3.6.1.4.1.4203.1.12.2.3.0.59	olcsecurity
1.3.6.1.4.1.4203.1.12.2.3.0.58	olcschemadn
0.9.2342.19200300.100.1.7	photo
1.3.6.1.4.1.4203.1.12.2.3.0.54	olcsaslrealm
1.3.6.1.4.1.4203.1.12.2.3.0.56	olcsaslsecprops
1.3.6.1.4.1.4203.1.12.2.3.0.51	olcrootdse
1.3.6.1.4.1.4203.1.12.2.3.0.53	olcsaslhost
1.3.6.1.1.5	vendorversion
1.3.6.1.1.4	vendorname
2.16.840.1.113730.3.1.3	employeenumber
2.5.4.21	telexnumber
2.5.4.20	telephonenumber
2.5.4.23	facsimiletelephonenumber
2.5.4.23	fax
2.5.4.22	teletextterminalidentifier
2.5.4.25	internationalisdnumber



2.5.4.24	x121address
2.5.4.27	destinationindicator
2.5.4.26	registeredaddress
1.3.6.1.1.1.1.19	iphostnumber
1.3.6.1.1.1.1.14	nisnetgrouptriple
0.9.2342.19200300.100.1.39	homepostaladdress
1.3.6.1.1.1.1.11	shadowflag
1.3.6.1.4.1.1466.101.119.4	dynamicsubtrees
1.3.6.1.1.1.1.17	ipprotocolnumber
1.3.6.1.1.1.1.13	membernismetgroup
1.3.6.1.4.1.1466.101.119.3	entryttl
1.3.6.1.4.1.4203.1.12.2.3.0.68	olciscacertificatefile
1.3.6.1.4.1.4203.1.12.2.3.0.69	olciscacertificatepath
1.3.6.1.4.1.4203.1.12.2.3.2.0.9	olcrootpw
1.3.6.1.4.1.4203.1.12.2.3.2.0.8	olcrootdn
2.16.840.1.113730.3.1.241	displayname
2.5.4.28	preferredeliverymethod
1.3.6.1.4.1.4203.1.12.2.3.0.60	olcsizelimit
1.3.6.1.4.1.4203.1.12.2.3.0.61	olcsockbufmaxincoming
1.3.6.1.4.1.4203.1.12.2.3.0.62	olcsockbufmaxincomingauth
1.3.6.1.4.1.4203.1.12.2.3.2.0.7	olcreplica
1.3.6.1.4.1.4203.1.12.2.3.2.0.6	olcmaxderefdpth



1.3.6.1.4.1.4203.1.12.2.3.2.0.5	olclimits
1.3.6.1.4.1.4203.1.12.2.3.2.0.4	olclastmod
0.9.2342.19200300.100.1.54	ditredirect
0.9.2342.19200300.100.1.55	audio
0.9.2342.19200300.100.1.56	documentpublisher
2.16.840.1.113730.3.1.34	ref
0.9.2342.19200300.100.1.8	userclass
0.9.2342.19200300.100.1.52	subtreemaximumquality
0.9.2342.19200300.100.1.53	personalsignature
2.5.4.54	dmdname
2.16.840.1.113730.3.1.39	preferredlanguage
0.9.2342.19200300.100.1.9	host
2.5.4.50	uniquemember
2.5.4.51	houseidentifier
2.5.4.52	supportedalgorithms
2.5.4.53	deltarevocationlist
0.9.2342.19200300.100.1.49	dsaquality
0.9.2342.19200300.100.1.48	buildingname
0.9.2342.19200300.100.1.60	jpegphoto
0.9.2342.19200300.100.1.43	co
0.9.2342.19200300.100.1.43	friendlycountryname
0.9.2342.19200300.100.1.42	pager



0.9.2342.19200300.100.1.42	pagertelephonenumber
0.9.2342.19200300.100.1.3	mail
0.9.2342.19200300.100.1.3	rfc822mailbox
0.9.2342.19200300.100.1.40	personaltitle
1.3.6.1.4.1.250.1.57	labeleduri
0.9.2342.19200300.100.1.46	janetmailbox
0.9.2342.19200300.100.1.45	organizationalstatus
0.9.2342.19200300.100.1.44	uniqueidentifier
0.9.2342.19200300.100.1.21	secretary
0.9.2342.19200300.100.1.20	homephone
0.9.2342.19200300.100.1.20	hometelephonenumber
1.3.6.1.4.1.4203.1.12.2.3.0.77	olctlsdhparamfile
1.3.6.1.4.1.4203.1.12.2.3.0.75	olctlsverifyclient
1.3.6.1.4.1.4203.1.12.2.3.0.74	olctlsrandfile
1.3.6.1.4.1.4203.1.12.2.3.0.73	olctlsrclcheck
1.3.6.1.4.1.4203.1.12.2.3.0.72	olctlsiphersuite
1.3.6.1.4.1.4203.1.12.2.3.0.71	olctlscertificatekeyfile
1.3.6.1.4.1.4203.1.12.2.3.0.70	olctlscertificatefile
2.5.4.9	street
2.5.4.9	streetaddress
1.3.6.1.4.1.4203.1.12.2.3.0.79	olconfigdir
1.3.6.1.4.1.4203.1.12.2.3.0.78	olconfigfile



2.5.21.8	matchingruleuse
2.5.21.9	structuralobjectclass
0.9.2342.19200300.100.1.27	mdrecord
2.5.4.49	distinguishedname
2.5.4.7	l
2.5.4.7	localityname
2.5.4.47	enhancedsearchguide
2.5.4.46	dnqualifier
2.5.4.45	x500uniqueidentifier
2.5.4.44	generationqualifier
2.5.21.4	matchingrules
2.5.21.5	attributetypes
2.5.4.41	name
2.5.4.40	crosscertificatepair
1.3.6.1.1.1.1.7	shadowmax
0.9.2342.19200300.100.1.10	manager
2.5.4.5	serialnumber
0.9.2342.19200300.100.1.38	associatedname
0.9.2342.19200300.100.1.11	documentidentifier
1.3.6.1.4.1.4203.1.12.2.3.0.82	olctlscrifile
1.3.6.1.4.1.4203.1.12.2.3.0.83	olcsortvals
1.3.6.1.4.1.4203.1.12.2.3.0.80	olctoolthreads



1.3.6.1.4.1.4203.1.12.2.3.0.81	olcserverid
0.9.2342.19200300.100.1.12	documenttitle
1.3.6.1.4.1.4203.1.12.2.3.0.84	olcindexintlen
1.3.6.1.4.1.4203.1.12.2.3.2.1.3	olcdbconfig
2.16.840.1.113730.3.1.216	userpkcs12
0.9.2342.19200300.100.1.22	othermailbox
0.9.2342.19200300.100.1.13	documentversion
2.5.18.9	hassubordinates
1.3.6.1.1.16.4	entryuuid
2.5.18.4	modifiersname
0.9.2342.19200300.100.1.14	documentauthor
1.3.6.1.4.1.4203.1.12.2.3.2.1.1	olcdbcachesize
2.5.18.1	createtimestamp
2.5.18.2	modifytimestamp
2.5.18.3	creatorsname
1.3.6.1.1.1.1.6	shadowmin
1.3.6.1.1.1.1.18	oncrpcnumber
1.3.6.1.4.1.4203.1.12.2.3.2.1.8	olcblockdetect
1.3.6.1.4.1.4203.1.12.2.3.2.1.9	olcdbsearchstack
1.3.6.1.4.1.4203.1.12.2.3.2.1.6	olcdblcacheidcachesize
1.3.6.1.4.1.4203.1.12.2.3.2.1.7	olcdblindex
1.3.6.1.4.1.4203.1.12.2.3.2.1.4	olcdbnosync



1.3.6.1.4.1.4203.1.12.2.3.2.1.5	olcldbdirtyread
1.3.6.1.4.1.4203.1.12.2.3.2.1.2	olcldbcheckpoint
2.5.4.3	cn
2.5.4.3	commonname
2.5.4.0	objectclass
2.5.4.1	aliasedobjectname
2.5.4.1	aliasedentryname
1.3.6.1.4.1.1466.101.120.16	ldapsyntaxes
1.3.6.1.4.1.1466.101.120.14	supportedsaslmmechanisms
1.3.6.1.4.1.1466.101.120.15	supportedldapversion
1.3.6.1.4.1.1466.101.120.13	supportedcontrol
2.5.4.6	c
2.5.4.6	countryname
2.5.4.10	o
2.5.4.10	organizationname
2.5.4.11	ou
2.5.4.11	organizationalunitname
2.5.4.12	title
2.5.4.13	description
2.5.4.14	searchguide
2.5.4.15	businesscategory
2.5.4.16	postaladdress



1.3.6.1.4.1.4203.1.12.2.3.2.0.18	olcmonitoring
1.3.6.1.4.1.4203.1.12.2.3.2.0.17	olchidden
1.3.6.1.4.1.4203.1.12.2.3.2.0.16	olcmirrormode
1.3.6.1.4.1.4203.1.12.2.3.2.0.15	olcsubordinate
0.9.2342.19200300.100.1.26	arecord
1.3.6.1.4.1.4203.1.12.2.3.2.0.13	olcupdateref
1.3.6.1.4.1.4203.1.12.2.3.2.0.12	olcupdatedn
1.3.6.1.4.1.4203.1.12.2.3.2.0.11	olcsyncrepl
1.3.6.1.4.1.4203.1.12.2.3.2.0.10	olcsuffix
1.3.6.1.1.1.1.1.20	ipnetworknumber
1.3.6.1.1.1.1.1.21	ipnetmasknumber
1.3.6.1.1.1.1.1.16	ipserviceprotocol
1.3.6.1.1.1.1.1.23	bootparameter
1.3.6.1.1.1.1.1.24	bootfile
1.3.6.1.1.1.1.1.26	nismapname
1.3.6.1.1.1.1.1.27	nismapentry
0.9.2342.19200300.100.1.41	mobile
0.9.2342.19200300.100.1.41	mobiletelephonenumber
1.3.6.1.4.1.4203.1.12.2.3.0.11	olcconnmaxpending
1.3.6.1.4.1.4203.1.12.2.3.0.10	olcconcurrency
1.3.6.1.4.1.4203.1.12.2.3.0.13	olcdatabase
1.3.6.1.4.1.4203.1.12.2.3.0.12	olcconnmaxpendingauth



1.3.6.1.4.1.4203.1.12.2.3.0.15	olcdisallows
1.3.6.1.4.1.4203.1.12.2.3.0.14	olcdefaultsearchbase
1.3.6.1.4.1.4203.1.12.2.3.0.17	olcgentlehub
1.3.6.1.4.1.4203.1.12.2.3.0.16	olciditcontentrules
1.3.6.1.4.1.4203.1.12.2.3.0.19	olcinclude
1.3.6.1.4.1.4203.1.12.2.3.0.18	olcidletimeout
2.5.4.65	pseudonym
1.3.6.1.4.1.4203.1.12.2.3.2.0.1	olcldbdirectory
2.5.4.4	sn
2.5.4.4	surname
1.3.6.1.1.1.1.4	loginshell
1.3.6.1.4.1.4203.1.12.2.3.2.1.10	olcldbshmkey
1.3.6.1.4.1.4203.1.12.2.3.2.1.11	olcdbcachefree
1.3.6.1.4.1.4203.1.12.2.3.2.1.12	olcdbdncachesize
1.3.6.1.4.1.4203.1.12.2.3.2.1.13	olcdbcryptfile
1.3.6.1.4.1.4203.1.12.2.3.2.1.14	olcdbcryptkey
0.9.2342.19200300.100.1.15	documentlocation
1.3.6.1.1.1.1.15	ipserviceport
0.9.2342.19200300.100.1.47	mailpreferenceoption
2.16.840.1.113730.3.1.4	employeetype
0.9.2342.19200300.100.1.29	nsrecord
2.16.840.1.113730.3.1.1	carlicense



2.16.840.1.113730.3.1.2	departmentnumber
0.9.2342.19200300.100.1.28	mxrecord
1.3.6.1.4.1.4203.1.12.2.3.0.67	olctimelimit
2.5.4.17	postalcode
2.5.4.18	postofficebox
1.3.6.1.4.1.4203.1.12.2.3.0.28	olcloglevel
2.5.4.2	knowledgeinformation
1.3.6.1.4.1.4203.1.12.2.3.0.26	olclocalssf
1.3.6.1.4.1.4203.1.12.2.3.0.27	olclogfile
1.3.6.1.4.1.4203.1.12.2.3.0.20	olcindexsubstrifminlen
1.3.6.1.4.1.4203.1.12.2.3.0.21	olcindexsubstrifmaxlen
1.3.6.1.4.1.4203.1.12.2.3.0.22	olcindexsubstranylen
1.3.6.1.4.1.4203.1.12.2.3.0.23	olcindexsubstranystep
2.5.4.36	usercertificate
2.5.4.37	cacertificate
2.5.4.34	seealso
2.5.4.35	userpassword
2.5.4.32	owner
2.5.4.33	roleoccupant
2.5.4.30	supportedapplicationcontext
2.5.4.31	member
2.5.4.29	presentationaddress



1.3.6.1.4.1.4203.1.3.5	supportedfeatures
2.5.4.48	protocolinformation
2.5.4.38	authorityrevocationlist
2.5.4.39	certificaterevocationlist
1.3.6.1.1.1.1.9	shadowinactive
1.3.6.1.1.1.1.8	shadowwarning
1.3.6.1.1.1.1.5	shadowlastchange
1.3.6.1.4.1.1466.101.120.5	namingcontexts
1.3.6.1.4.1.1466.101.120.6	altserver
1.3.6.1.4.1.1466.101.120.7	supportedextension
1.3.6.1.1.1.1.1	gidnumber
1.3.6.1.1.1.1.0	uidnumber
1.3.6.1.1.1.1.3	homedirectory
1.3.6.1.1.1.1.2	gecos
2.5.4.43	initials
1.3.6.1.1.1.1.22	macaddress
2.5.4.42	givenname
2.5.4.42	gn
1.3.6.1.4.1.4203.1.12.2.3.2.0.3	olcdbmode
1.3.6.1.4.1.4203.1.12.2.3.0.39	olcpluginlogfile
1.3.6.1.4.1.4203.1.12.2.3.0.38	olcplugin
2.5.21.6	objectclasses



1.3.6.1.4.1.4203.1.12.2.3.2.0.2	olcdbindex
1.3.6.1.4.1.4203.1.12.2.3.0.33	olcobjectidentifier
1.3.6.1.4.1.4203.1.12.2.3.0.32	olcobjectclasses
1.3.6.1.4.1.4203.1.12.2.3.0.31	olcmodulepath
1.3.6.1.4.1.4203.1.12.2.3.0.30	olcmoduleload
1.3.6.1.4.1.4203.1.12.2.3.0.37	olcpidfile
1.3.6.1.4.1.4203.1.12.2.3.0.36	olcpasswordhash
1.3.6.1.4.1.4203.1.12.2.3.0.35	olcpasswordcryptsaltformat
1.3.6.1.4.1.4203.1.12.2.3.0.34	olcoverlay
ObjectClass	
OID	Nombre
2.16.840.1.113730.3.2.2	inetorgperson
1.3.6.1.1.1.2.5	oncrpc
0.9.2342.19200300.100.4.22	qualitylabelleddata
2.16.840.1.113730.3.2.6	referral
0.9.2342.19200300.100.4.20	pilotorganization
0.9.2342.19200300.100.4.21	pilotdsa
0.9.2342.19200300.100.4.7	room
1.3.6.1.4.1.250.3.15	labeleduriobject
0.9.2342.19200300.100.4.5	account
0.9.2342.19200300.100.4.4	pilotperson



0.9.2342.19200300.100.4.4	newpilotperson
2.5.6.8	organizationalrole
2.5.6.9	groupofnames
2.5.6.4	organization
2.5.6.5	organizationalunit
2.5.6.6	person
2.5.6.7	organizationalperson
2.5.6.0	top
2.5.6.1	alias
2.5.6.2	country
2.5.6.3	locality
2.5.20.1	subschema
1.3.6.1.4.1.4203.1.4.1	openldaprootdse
1.3.6.1.4.1.4203.1.4.1	ldaprootdse
1.3.6.1.4.1.4203.1.12.2.4.2.2.1	olcl difconfig
2.5.6.23	deltacrl
2.5.6.22	pkica
2.5.6.21	pkuser
2.5.6.20	dmd
1.3.6.1.1.1.2.10	nisobject
1.3.6.1.4.1.1466.101.119.2	dynamicobject
1.3.6.1.1.1.2.12	bootabledevice



1.3.6.1.4.1.4203.1.12.2.4.2.1.2	olchdbconfig
2.5.6.18	usersecurityinformation
1.3.6.1.1.1.2.11	ieee802device
0.9.2342.19200300.100.4.17	domainrelatedobject
1.3.6.1.4.1.1466.101.120.111	extensibleobject
0.9.2342.19200300.100.4.15	dnsdomain
0.9.2342.19200300.100.4.14	rfc822localpart
0.9.2342.19200300.100.4.13	domain
2.5.6.10	residentialperson
1.3.6.1.1.3.1	uidobject
2.5.17.0	subentry
0.9.2342.19200300.100.4.19	simplesecurityobject
0.9.2342.19200300.100.4.18	friendlycountry
1.3.6.1.1.1.2.2	posixgroup
1.3.6.1.1.1.2.3	ipservice
1.3.6.1.1.1.2.0	posixaccount
1.3.6.1.1.1.2.1	shadowaccount
1.3.6.1.1.1.2.6	iphost
1.3.6.1.1.1.2.7	ipnetwork
1.3.6.1.1.1.2.4	ipprotocol
2.5.6.19	crldistributionpoint
1.3.6.1.1.1.2.8	nisnetgroup



1.3.6.1.1.1.2.9	nismap
1.3.6.1.4.1.1466.344	dcobject
0.9.2342.19200300.100.4.9	documentseries
0.9.2342.19200300.100.4.6	document
2.5.6.16.2	certificationauthority-v2
2.5.6.16	certificationauthority
2.5.6.17	groupofuniquenames
2.5.6.14	device
2.5.6.15	strongauthenticationuser
2.5.6.12	applicationentity
2.5.6.13	dsa
1.3.6.1.4.1.4203.1.12.2.4.0.8	olcmodulelist
2.5.6.11	applicationprocess
1.3.6.1.4.1.4203.1.12.2.4.0.6	olcincludefile
1.3.6.1.4.1.4203.1.12.2.4.0.7	olcfrontendconfig
1.3.6.1.4.1.4203.1.12.2.4.0.4	olcdatabaseconfig
1.3.6.1.4.1.4203.1.12.2.4.0.5	olcoverlayconfig
1.3.6.1.4.1.4203.1.12.2.4.0.2	olcschemaconfig
1.3.6.1.4.1.4203.1.12.2.4.0.3	olcbackendconfig
1.3.6.1.4.1.4203.1.12.2.4.0.0	olconfig
1.3.6.1.4.1.4203.1.12.2.4.0.1	olcglobal

Matching Rule



OID	Nombre
1.3.6.1.1.16.3	uuidorderingmatch
1.3.6.1.1.16.2	uuidmatch
2.5.13.20	telephonenumbermatch
2.5.13.21	telephonenumberstringsmatch
2.5.13.27	generalizedtimematch
2.5.13.28	generalizedtimeorderingmatch
2.5.13.29	integerfirstcomponentmatch
2.5.13.7	caseexactsubstringsmatch
1.2.36.79672281.1.13.3	rdnmatch
1.2.840.113556.1.4.803	integerbitandmatch
1.3.6.1.4.1.4203.1.2.1	caseexactia5substringsmatch
1.3.6.1.4.1.1466.109.114.1	caseexactia5match
1.2.840.113556.1.4.804	integerbitormatch
1.3.6.1.4.1.1466.109.114.3	caseignoreia5substringsmatch
1.3.6.1.4.1.1466.109.114.2	caseignoreia5match
2.5.13.17	octetstringmatch
2.5.13.16	bitstringmatch
2.5.13.15	integerorderingmatch
2.5.13.14	integermatch
2.5.13.13	booleanmatch
2.5.13.34	certificateexactmatch



2.5.13.10	numericstringsubstringmatch
2.5.13.19	octetstringsubstringmatch
2.5.13.18	octetstringorderingmatch
2.5.13.30	objectidentifierfirstcomponentmatch
2.5.13.3	caseignoreorderingmatch
2.5.13.2	caseignorematch
2.5.13.1	distinguishednamematch
2.5.13.0	objectidentifiermatch
2.5.13.23	uniquemembermatch
2.5.13.6	caseexactorderingmatch
2.5.13.5	caseexactmatch
2.5.13.4	caseignoresubstringmatch
2.5.13.9	numericstringorderingmatch
2.5.13.8	numericstringmatch
Syntax	
OID	Nombre
1.2.36.79672281.1.5.0	RDN
1.3.6.1.4.1.1466.115.121.1.40	Octet String
1.3.6.1.4.1.1466.115.121.1.41	Postal Address
1.3.6.1.4.1.1466.115.121.1.52	Telex Number
1.3.6.1.1.16.1	UUID



1.3.6.1.4.1.1466.115.121.1.50	Telephone Number
1.3.6.1.4.1.1466.115.121.1.28	JPEG
1.3.6.1.4.1.1466.115.121.1.49	Supported Algorithm
1.3.6.1.4.1.1466.115.121.1.44	Printable String
1.3.6.1.4.1.1466.115.121.1.45	SubtreeSpecification
1.3.6.1.4.1.1466.115.121.1.26	IA5 String
1.3.6.1.4.1.1466.115.121.1.27	Integer
1.3.6.1.4.1.1466.115.121.1.24	Generalized Time
1.3.6.1.1.1.0.0	RFC2307 NIS Netgroup Triple
1.3.6.1.1.1.0.1	RFC2307 Boot Parameter
1.3.6.1.4.1.1466.115.121.1.39	Other Mailbox
1.3.6.1.4.1.1466.115.121.1.38	OID
1.3.6.1.4.1.1466.115.121.1.15	Directory String
1.3.6.1.4.1.1466.115.121.1.14	Delivery Method
1.3.6.1.4.1.1466.115.121.1.12	Distinguished Name
1.3.6.1.4.1.1466.115.121.1.11	Country String
1.3.6.1.4.1.1466.115.121.1.10	Certificate Pair
1.3.6.1.4.1.1466.115.121.1.34	Name And Optional UID
1.3.6.1.4.1.1466.115.121.1.36	Numeric String
1.3.6.1.4.1.1466.115.121.1.22	Facsimile Telephone Number
1.3.6.1.4.1.1466.115.121.1.7	Boolean
1.3.6.1.4.1.1466.115.121.1.6	Bit String



1.3.6.1.4.1.1466.115.121.1.5	Binary
1.3.6.1.4.1.1466.115.121.1.4	Audio
1.3.6.1.4.1.1466.115.121.1.9	Certificate List
1.3.6.1.4.1.1466.115.121.1.8	Certificate



Glosario de términos

A

Apache: Es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP.

B

bytecodes: Código intermedio generalmente tratado como un fichero binario que contiene un programa ejecutable similar a un módulo objeto. Algunos sistemas, llamados traductores dinámicos traducen el bytecode a código máquina inmediatamente antes de su ejecución para mejorar la velocidad de ejecución.

D

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.

G

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.

GTK: Biblioteca constituida por objetos y funciones que permiten crear interfaces gráficas de usuario. Maneja widgets como ventanas, botones, menús, etiquetas, deslizadores, pestañas, entre otros.

J



Jabber: Mensajería instantánea basada en XMPP, un protocolo extensible, abierto y estándar basado en XML para el intercambio en tiempo real de mensajes y presencia entre dos puntos en una red.

L

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.

M

MySQL: Sistema de gestión de base de datos relacional, multiplataforma y bajo licencia GNU GPL.

O

Open Source: Término con el que se conoce al software distribuido y desarrollado libremente. Sin embargo, mucho de este software no da a sus usuarios la libertad de distribuir sus modificaciones, restringe el uso comercial, o en general restringe los derechos de los usuarios.

P

Perl: Practical Extraction and Report Language es un lenguaje de programación para la administración de tareas propias de sistemas UNIX.

PHP: 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.

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

Q

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

R



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

S

Samba: Es una implementación libre del protocolo de archivos compartidos de Microsoft Windows para sistemas de tipo UNIX. También permite validar usuarios haciendo de Controlador Principal de Dominio (PDC).

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.

T

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.

W

Widgets: es un componente gráfico, o control, con el cual el usuario interactúa, como por ejemplo, una ventana, una barra de tareas o una caja de texto.