

República de Cuba



*Universidad de las Ciencias Informáticas
Facultad 2*

*Título: Plataforma de Gestión de Servicios Telemáticos
en GNU/Linux,
Módulo de Directorio*

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

*Autor: Sandy Noa Cabrera
Tutor: Ing. Ramón Alexander Anglada Martínez*

*Ciudad de la Habana
Junio de 2009*



*“El escalón más alto que puede alcanzar
la especie humana es ser revolucionario”*

Che



Agradecimientos

Agradezco a mis padres por haberme guiado en la vida, por tanto sacrificio, por confiar siempre en mí.

A mi novia Mayret, por haber estado tanto tiempo a mi lado, por tanto amor, por tantos y tan buenos momentos, por tu sinceridad, por ser siempre la misma; auténtica y diferente. A ti mi AFU, te agradezco por esta tesis.

A Maite, mi hermana, por haber cuidado de mami mientras yo no estaba, por haber confiado en mí durante todos estos años.

A mi abuela Isabel por acompañarme cada día de mi vida.

A mi abuelo Rubén por haber sido su hombre de confianza.

A mi familia toda, por darme todo lo que siempre pude necesitar: afecto y confianza.

A Norlen, Kenny, Yovannis y una larga lista de amigos, por su apoyo incondicional, por haberme hecho sentir orgulloso de estar rodeado de hermanos. A ustedes, los que nunca cambian, muchas gracias.

Al tutor que se convirtió en compañero de tesis, al compañero de tesis que se convirtió en amigo, al amigo que se convirtió en hermano, a Anglada.



Dedicatoria

A mis padres, mi hermana y mi familia toda por haber sacrificado tanto.

A mi novia, por su constancia, apoyo, dedicación y amor;

sin ti no hubiera podido lograr tantas cosas buenas.

A mis grandes amigos, los que siempre estaban cuando más los necesite.



Resumen

El presente trabajo tiene por objetivo la descripción de una propuesta que mejore la gestión del servidor de Directorio OpenLDAP para lograr establecer un control de eficiencia sobre los usuarios y recursos de las redes informáticas, entre ellas las auditadas por la Oficina para la Seguridad en Redes Informáticas (OSRI) adjunta al Ministerio de la Informática y las Comunicaciones. Actualmente muchas entidades nacionales realizan este proceso de gestión empleando soluciones de pago como Microsoft Windows Server, lo cual limita sustancialmente en cuanto a posibles mejoras a aplicar, soporte, mantenimiento, entre otros factores. La propuesta realizada es consecuente con la estrategia definida por el Consejo de Ministros de comenzar a migrar gradualmente todos los servicios de red nacionales a Software Libre.

Teniendo en cuenta esta necesidad, se ha modelado e implementado una aplicación de escritorio que permitirá administrar cómoda y profesionalmente el servidor OpenLDAP ya sea de manera local o remota. El sistema permitirá realizar operaciones con las entidades o entradas pertenecientes al servidor, como agregaciones, modificaciones, eliminaciones y búsquedas. Posibilitando además realizar una gestión sobre los archivos LDIF empleados para el intercambio de información entre servidores, dicha gestión incluirá la posibilidad de exportar parcial o totalmente el Directorio, así como importar las configuraciones residentes en archivos previamente generados. Podrán visualizarse los puertos habilitados para brindar el servicio, el protocolo empleado, entre otros valores de interés. El sistema permitirá asociarle un certificado al servidor garantizando las conexiones cifradas de terceras aplicaciones. Serán incluidas opciones para administrar modos de réplicas del contenido, ya sean másteres o esclavos. En términos de portabilidad será posible extender el sistema de gestión a otras plataformas además de Unix como es el caso de Windows. Un sistema de este tipo, dadas las condiciones actuales cubanas, es sin lugar a dudas un recurso importante para los administradores de redes.

El análisis de factibilidad realizado demuestra que es totalmente viable acometer el proyecto y que la Universidad de las Ciencias Informáticas está dando pasos sólidos en correspondencia con la visión que ha alcanzado el país.

Palabras claves: servidor de Directorio, OpenLDAP, redes, Software Libre, migrar, gestión, administradores.



Summary

The present investigation has for objective the description of a proposal that improves the administration of Directory's server OpenLDAP to establish a control of efficiency on the users and resources of the networks, among them the entities checked by the Office for the Security in Computer Nets (OSCN) enclosed to the Computer Science's Ministry and the Communications. Actually many national entities carry out this administration process using payment solutions as Microsoft Windows Server, that limits substantially as for possible improvements to apply, support, maintenance, among other factors. The carried out proposal is consequent with the strategy defined by the Council of Secretaries of beginning to transform gradually all the national net services to Free Software.

Keeping in mind this necessity, it has been modeled and implemented a desktop application that will allow administering comfortable and professionally OpenLDAP server, either in a local or remote way. The system will allow carrying out operations with the entities or entrances belonging to the server, as aggregations, modifications, eliminations and searches. Also facilitating to carry out an administration on the files LDIF used for the exchange information among servers, this administration will include the possibility to export partial or totally the Directory, as well as to import the configurations residents in previously generated files. The system will be able to visualize the ports enabled to offer the service, the used protocol, among other values of interest. It will allow associating him a certificate to the server guaranteeing ciphers connections of third applications. Options will be included to administer ways of replicas of the content, like masters or slaves. In portability terms it will be possible to extend the administration system to other platforms besides UNIX like Windows. A system of this type, given the Cuban current conditions, it is an important resource for the administrators of nets.

The analysis of feasibility demonstrates that it is completely viable to eject the project and that the University of the Computer Sciences is taking solid steps in correspondence with the vision that has reached the country.



Índice

INTRODUCCIÓN	1
CAPÍTULO 1 “Fundamentación Teórica”	6
1.1 Introducción	6
1.2 Términos utilizados	6
1.3 Sistemas para la gestión de los servidores de red	6
1.3.1 Soluciones acopladas a sistemas operativos.....	7
1.3.2 Soluciones independientes.....	10
1.4 Lenguaje de modelado.....	10
1.5 Metodología de desarrollo	11
1.6 Modelación de funciones mediante IDEF0	14
1.8 Lenguaje de programación y herramientas	17
1.8.1 Elección de un lenguaje adecuado	17
1.8.1.1 Python	17
1.8.2 Herramientas Case.....	18
1.8.2.1 Visual Paradigm	18
1.8.3 Entorno de Desarrollo Integrado.....	18
1.8.3.1 Alternativas aplicables	19
1.8.3.2 EasyEclipse	19
1.8.4 Interfaces gráficas de usuario.....	20
1.8.4.1 TkInter	20
1.8.4.2 wxPython.....	20
1.8.4.3 PyGTK.....	21
1.8.4.4 PyQt	21
1.8.4.5 Qt Designer	22
1.9 Conclusiones	22
CAPÍTULO 2 “Características del sistema”	23
2.1 Introducción	23
2.2 Problema y situación problemática.....	23



2.2.1 Descripción de los procesos de negocio.....	26
2.3 Objeto de automatización.....	26
2.4 Información que se maneja	26
2.5 Solución propuesta	27
2.6 Modelo de negocio.....	27
2.6.1 Representación de los procesos con IDEF0.....	28
2.6.2 Descripción detallada de los procesos de negocio	30
2.7 Especificación de los requisitos de software	31
2.7.1 Requerimientos funcionales	31
2.7.1.2 Requerimientos funcionales del módulo Directorio	31
2.7.1 Requerimientos no funcionales	33
2.8 Modelo de Casos de Uso del Sistema.....	37
2.8.1 Actores del Sistema.....	37
2.8.2 Casos de Uso del Sistema (CUS).....	38
2.8.2.2 CUS del módulo “Directorio”	38
2.8.3 Diagrama de paquetes	42
2.8.4 Diagrama de Casos de Uso del Sistema (DCUS)	43
2.8.4.1 DCUS del módulo: “Directorio”	44
2.8.5 Descripción detallada de los CUS y prototipos de interfaces	45
2.9 Conclusiones	63
CAPÍTULO 3 “Análisis y Diseño del Sistema”	64
3.1 Introducción	64
3.2 Modelo de Análisis	64
3.2.1 Diagrama de Clases del Análisis	64
3.3 Modelo de Diseño	68
3.4 Conclusiones	79
CAPÍTULO 4 “Implementación”	80
4.1 Introducción	80
4.2 Diagrama de despliegue	80
4.3 Diagrama de componentes	81



4.4 Modelo de Prueba	89
4.5 Conclusiones	99
CAPÍTULO 5 “Estudio de Factibilidad”	100
5.1 Introducción	100
5.2 Planificación	100
5.2.1 Puntos de Casos de Uso	100
5.2.1.1 Cálculo de Puntos de Casos de Uso sin ajustar	101
5.2.1.1 Cálculo de Puntos de Casos de Uso ajustados	102
5.2.2 Estimación del esfuerzo	105
5.3 Beneficios tangibles e intangibles.....	107
5.4 Análisis de costos y beneficios.....	108
5.5 Conclusiones	108
CONCLUSIONES	109
RECOMENDACIONES	110
ANEXOS.....	111
Anexo 1: Entradas del servidor de directorio OpenLDAP	111
Anexo 2: Tipos de clases y atributos en el servidor OpenLDAP	113
Bibliografía.....	138
Glosario de términos.....	140



Figura 1: RUP en dos dimensiones.....	14
Figura 2: Componentes básicos de IDEF0.....	15
Figura 3: Jerarquía de procesos en IDEF0.....	16
Figura 4: Representación del proceso base empleando as-is	25
Figura 5: Detalles del proceso base empleando as-is	25
Figura 6: Representación del proceso base empleando to-be.....	29
Figura 7: Detalles del proceso base empleando to-be.....	29
Figura 8: Diagrama de paquetes del módulo Directorio.....	43
Figura 9: DCUS del paquete de Autenticación	44
Figura 10: DCUS del paquete de Entradas	44
Figura 11: DCUS del paquete de Servidor	45
Figura 12: Interfaz del CU Autenticar Usuario	47
Figura 13: Interfaz del CU Gestionar entradas, sección de Adición. Parte I	51
Figura 14: Interfaz del CU Gestionar entradas, sección de Adición. Parte II	52
Figura 15: Interfaz del CU Gestionar entradas, sección de Eliminación	53
Figura 16: Interfaz general del sistema	63
Figura 17: Diagrama de Clases del Análisis: CU Autenticar usuario	64
Figura 18: Diagrama de Clases del Análisis: CU Gestionar entrada.....	65
Figura 19: Diagrama de Clases del Análisis: CU Mostrar estructura general	65
Figura 20: Diagrama de Clases del Análisis: CU Gestionar puerto	66
Figura 21: Diagrama de Clases del Análisis: CU Certificar servidor	66
Figura 22: Diagrama de Clases del Análisis: CU Gestionar réplica	67
Figura 23: Diagrama de Clases del Análisis: CU Gestionar configuración.....	67
Figura 24: Estructura general del sistema.....	69
Figura 25: Estructura detallada del sistema	71
Figura 26: Definición de los términos fundamentales	71
Figura 27: Diagrama de Clases del Diseño: CU Autenticar Usuario	72
Figura 28: Diagrama de Clases del Diseño: CU Gestionar Entrada	73
Figura 29: Diagrama de Clases del Diseño: CU Mostrar Estructura General	74
Figura 30: Diagrama de Clases del Diseño: CU Gestionar Puerto	75



Figura 31: Diagrama de Clases del Diseño: CU Construir Certificado.....	76
Figura 32: Diagrama de Clases del Diseño: CU Gestionar Réplica.....	77
Figura 33: Diagrama de Clases del Diseño: CU Gestionar Configuración.....	78
Figura 34: Diagrama de Despliegue del sistema.....	80
Figura 35: Diagrama General de Componentes.....	82
Figura 36: Diagrama de Componentes: CU Autenticar usuario.....	83
Figura 37: Diagrama de Componentes: CU Gestionar entrada.....	84
Figura 38: Diagrama de Componentes: CU Mostrar estructura general.....	85
Figura 39: Diagrama de Componentes: CU Obtener puertos.....	86
Figura 40: Diagrama de Componentes: CU Gestionar réplicas.....	87
Figura 41: Diagrama de Componentes: CU Gestionar configuraciones.....	88
Figura 42: Diagrama de Componentes: CU Certificar servidor.....	89



Tabla 1: Actor del sistema.....	38
Tabla 2: Descripción CU Autenticar usuario.....	39
Tabla 3: Descripción CU Gestionar entrada.....	39
Tabla 4: Descripción CU Buscar entradas.....	39
Tabla 5: Descripción CU Obtener puertos.....	40
Tabla 6: Descripción CU Gestionar puertos de acceso.....	40
Tabla 7: Descripción CU Gestionar estado.....	41
Tabla 8: Descripción CU Certificar servidor.....	41
Tabla 9: Descripción CU Gestionar réplicas.....	42
Tabla 10: Descripción CU Gestionar configuraciones.....	42
Tabla 11: Descripción CU Mostrar estructura general.....	42
Tabla 12: Descripción detallada CU Autenticar usuario.....	47
Tabla 13: Descripción detallada CU Gestionar entrada.....	50
Tabla 14: Descripción detallada CU Buscar entrada.....	54
Tabla 15: Descripción detallada CU Obtener puertos.....	55
Tabla 16: Descripción detallada CU Gestionar puertos.....	56
Tabla 17: Descripción detallada CU Gestionar estado.....	57
Tabla 18: Descripción detallada CU Certificar servidor.....	58
Tabla 19: Descripción detallada CU Gestionar réplicas.....	60
Tabla 20: Descripción detallada CU Gestionar configuraciones.....	61
Tabla 21: Descripción detallada CU Mostrar estructura general.....	62
Tabla 22: Descripción del caso de prueba Autenticar usuario.....	91
Tabla 23: Descripción del caso de prueba Gestionar entradas.....	93
Tabla 24: Descripción del caso de prueba Mostrar estructura general.....	94
Tabla 25: Descripción del caso de prueba Gestionar configuración.....	95
Tabla 26: Descripción del caso de prueba Certificar servidor.....	97
Tabla 27: Descripción del caso de prueba Gestionar puertos.....	98
Tabla 28: Descripción del caso de prueba Gestionar réplicas.....	99
Tabla 29: Puntos de Casos de Uso sin ajustar.....	101
Tabla 30: Factor de Peso de los Actores sin ajustar.....	102



Tabla 31: Factor de peso de los Casos de Uso sin ajustar.....	102
Tabla 32: Cálculo de Puntos de Casos de Uso ajustados	103
Tabla 33: Valores del Factor de complejidad técnica	104
Tabla 34: Valores del Factor de Ambiente	104
Tabla 35: Cálculo del esfuerzo.....	105
Tabla 36: Esfuerzo del proyecto.....	106
Tabla 37: Entradas en el servidor de Directorio: OpenLDAP.....	112
Tabla 38: Tipos de clases y atributos del servidor OpenLDAP	137



INTRODUCCIÓN

La actualidad informática está caracterizada hoy por un auge en el desarrollo de las condiciones de hardware que dan soporte a los servicios tradicionales que operan sobre la red, sin embargo el tema “software” no ha quedado para nada rezagado. Aunque inicialmente no estaba considerado como un producto, sino como un añadido que se brindaba integrado a las primeras computadoras; la realidad es que ha mostrado un vertiginoso crecimiento desde aquellos primeros momentos en los años '60 y '70 del siglo pasado. A pesar de que la integración entre el software y el hardware posibilitan una gestión de la información nunca antes experimentada, que materializa sus bases en todos los sectores de la sociedad, desafortunadamente no prima en sus desarrolladores la misma filosofía de trabajo en cuanto a la construcción de bienes informáticos.

Grandes compañías gobiernan con sus productos la mayor parte del mercado mundial, ocasionando una afectación directa sobre los países más pobres, que se ven obligados a pagar elevados precios por concepto de patentes privativas; que en muchos casos no garantizan la calidad de la inversión. Uno de los principales iconos del mundo empresarial de la producción de software y servicios es la conocida empresa Microsoft, que se ha enriquecido apoyándose en licencias privativas para su trabajo, sin embargo todo su capital contrasta con los niveles de acceso al software de este tipo, los cuales son expresión solamente de una élite de poder en el mundo; provocando de esta forma la piratería no controlada, el incremento de ataques informáticos e incluso el abuso de patentes en el mejor de los casos. Sin ánimo de permitir que se estableciera un mundo unipolar, a pesar de que algunas estadísticas reflejen que nueve de cada diez computadoras de escritorio en el mundo utilizan los sistemas operativos de la familia de Windows, en 1984 Richard Stallman comenzó a trabajar en el proyecto GNU que perseguía la idea de crear un sistema operativo totalmente libre y que según el propio Stallman permitiera recuperar el espíritu de cooperación que prevaleció en los tiempos iniciales de la comunidad de usuarios de computadoras. Un año más tarde creó la Fundación para el Software Libre o Free Software Foundation (FSF) con el objetivo de eliminar las restricciones sobre la copia, redistribución y modificación de programas; además se introdujo una definición para “free software” y el concepto de “copyleft”, el cual desarrolló para darle a los usuarios libertad y restringir las posibilidades de apropiación del software; todo ello amparado bajo la Licencia



Pública General de GNU (GNU GPL), la cual posibilita la modificación y redistribución del software, pero únicamente bajo esa misma licencia.

El software se considera libre si en su conjunto es capaz de garantizar las siguientes libertades: usar el programa con cualquier propósito, estudiar cómo funciona el programa, y adaptarlo a las necesidades propias, distribuir copias, con lo cual se puede ayudar a otras personas y finalmente mejorar el programa y hacer públicas las mejoras a los demás, de modo que toda la comunidad se beneficie. La mayor parte del Software Libre es desarrollado por equipos internacionales que cooperan a través de la asociación voluntaria; dichos equipos están constituidos por individuos con una gran variedad de motivaciones. Todo este movimiento ha generado diversas posturas en relación al sistema económico capitalista; debido a que algunos lo consideran un competidor del capitalismo e incluso hay quienes llegan a catalogarlo como una economía de regalo; donde vale más quien más aporta.

Cuba sin lugar a dudas ha optado por el uso del Software Libre para su desarrollo, respaldado en el Acuerdo 84, de abril de 2004 del Consejo de Ministros, donde se ha dispuesto la organización de un programa para cambiar progresivamente los sistemas de los órganos y organismos del Estado y el Gobierno hacia la plataforma de Software Libre. Esta decisión abre la posibilidad de migrar en el futuro todos los sistemas públicos a una plataforma libre.

Este nuevo camino, que cada día se abre paso, resulta sustentable frente a otras tendencias de desarrollo de software, alcanzando su máxima expresión en el ámbito político, económico, tecnológico y social. En el ámbito político significa la no utilización de productos informáticos que demanden la autorización de sus propietarios para su utilización, representa una alternativa para los países pobres, además es desarrollado de forma cooperativa; mostrando su carácter público y sus objetivos de beneficiar a toda la comunidad. En el ámbito económico es importante destacar que su puesta en marcha no implica gastos monetarios asociados al costo del producto. Tecnológicamente no se requiere aplicar innovaciones adicionales por concepto de cambio de plataforma de software, lo que significa que es perfectamente aplicable al soporte de hardware con que contamos hoy. La sociedad agradecida experimentará la igualdad en el acceso al conocimiento empleando el menor tiempo y recursos posibles.

Varias de las principales entidades del país que impulsan sustancialmente el desarrollo y la innovación



como el Ministerio de la Informática y las Comunicaciones (MIC), la Oficina de Seguridad para Redes Informáticas (OSRI), adscrita al MIC y la UCI; se han trazado la misión de lograr sustentar sus servicios de red con Software Libre.

Generalmente para la mayoría de los profesionales de la informática y las telecomunicaciones de todo el mundo, administrar una pequeña infraestructura red no representa un problema significativo. Añadir nuevos servidores, periféricos o usuarios puede resultar muy sencillo, pero todo ello se complica en la medida que crecen las redes. En muchos casos se arriba a un punto en el cual la mayoría de las instituciones notan que les resulta imprescindible adoptar una estrategia de gestión de servidores más sólida que permita controlar entornos de mayor envergadura. Además, el desarrollo actual demuestra que en muchos casos redes como las de Windows y Linux necesitan inevitablemente interactuar para poder alcanzar un mayor y mejor aprovechamiento de los recursos con los cuales contamos. A medida que crece el negocio, se comienzan a utilizar más aplicaciones, ocasionando que inevitablemente se necesiten más servidores, más mantenimiento y más capacitación al personal implicado. En este marco resulta de gran apoyo un sistema para la gestión de los servicios telemáticos, el cual permita integrarse de forma inmediata a las infraestructuras de gestión, garantizando además un uso profesional en instancias concretas locales o remotas; basado en potentes herramientas que permitan simplificar el trabajo.

Actualmente el montaje y configuración de los servidores de Software Libre que ofrecen los servicios básicos de las redes, aunque utilizan potentes herramientas, se maneja en algunos casos a través de terminales o consolas, lo cual disminuye el número de usuarios que pueden realizar tareas administrativas; en otros casos se hace uso de herramientas de pago, que aunque pudieran resolver el problema limita toda la libertad que genera el uso de Software Libre.

La problemática se intensifica para los administradores de redes cuando se debaten en la disyuntiva de emplear soluciones basadas en Microsoft Windows o en plataformas Unix. Las soluciones que ofrece Windows presentan por lo general interfaces cómodas para la administración de los servidores a diferencia de muchas de las soluciones que podrían encontrarse en GNU/Linux. Sin embargo, más del 50% de los ordenadores que brindan servicios telemáticos en redes operan sobre plataformas Unix, lo cual demuestra su fortaleza en este tipo de tareas. Si se emplea una solución basada en Windows aunque



con herramientas muy eficientes, se estaría negando la posibilidad de emplear otra herramienta libre con potencialidades quizás superiores, se limitarían los accesos a los fuentes de la misma en caso de requerirse alguna mejora o mantenimiento, se privaría al país de los derechos de soporte y ayuda, por solo mencionar algunos. Si la elección está dirigida a una solución basada en Software Libre con servidores realmente potentes y acceso a los fuentes, quizás no se cuente con una herramienta integrada, sencilla e intuitiva para los administradores; utilizar una terminal o consola limitaría las labores de gestión, limitando el trabajo administrativo a un reducido grupo de personas. Uno de los propósitos del presente trabajo está sustentado en tomas de ambos sus mejores prácticas: interfaces robustas con servidores potentes.

El presente documento expresa la necesidad de lograr un producto con sello propio que permita administrar el servicio de directorio. Esta situación conlleva a la creación de un sistema informático que automatice la gestión de dicho servicio de forma profesional.

Con el objetivo de resolver los requerimientos impuestos y la problemática existente se ha identificado como **problema científico**: ¿Cómo mejorar la administración del servicio telemático de directorio en sistemas operativos GNU/Linux? Definiendo como **objeto de estudio** los procesos relativos a la gestión de servicios telemáticos en redes. El **campo de acción** estará dirigido a los procesos que garanticen la administración del servidor de Directorio OpenLDAP. El **objetivo general** es desarrollar una aplicación de escritorio para sistemas operativos GNU/Linux, que brinde la posibilidad de administrar el servicio de Directorio, de manera profesional, ya sea local o remotamente. En correspondencia con el objetivo planteado y la problemática existente se ha decidido enfocar las **tareas investigativas** a:

- Analizar el estado del arte relativo a la existencia en el mundo de sistemas similares, ya sean propietarios o no.
- Identificar los niveles de calidad que brindan los servidores de redes en los principales sistemas operativos que existen.
- Identificar los requisitos de un producto que garanticen la gestión de los servicios tratados.
- Implementar un mecanismo que gestione la administración del servidor de Directorio; incluso de manera remota.



El presente documento consta de 5 capítulos:

Capítulo 1: “Fundamentación Teórica”, permite encontrar los principales conceptos que se manejan a lo largo del trabajo; entre ellos el estado del arte del tema tratado, a nivel internacional, nacional y de 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”, hace especial énfasis en el análisis de la situación problemática existente, la construcción de un modelo de negocio adecuado; así como la definición de los procesos que serán objeto de automatización, dando paso a la especificación de los requisitos que constituirán las bases de la propuesta del sistema.

Capítulo 3: “Análisis y Diseño del Sistema”, representa la base para entender la futura implementación del sistema; esta sección asegura la elaboración de diagramas de clases e interacción, descripciones, entre otros elementos.

Capítulo 4: “Implementación y Prueba”, muestra como será implementado el sistema en término de componentes, contiene además; la estructura general de despliegue del mismo asociada a sus servidores de red, así como el modelo de pruebas realizadas durante todo el ciclo de vida del software.

Capítulo 5: “Estudio de factibilidad”, expresa la estimación en cuanto a tiempo y esfuerzo se refiere, así como los costos y beneficios asociados a la producción del software; dando una visión general de factibilidad.



CAPÍTULO 1 “Fundamentación Teórica”

1.1 Introducción

El presente capítulo recoge un análisis crítico referente a las tendencias y técnicas actuales para el desarrollo del tema tratado, tanto en Cuba como en el resto del mundo. Se hace especial énfasis en lograr aprehender las buenas prácticas que caracterizan soluciones actuales líderes del mercado. Además constituye el espacio idóneo para reflejar las tecnologías y metodologías propuestas para el sistema.

1.2 Términos utilizados

YaST y AutoYaST: constituyen un conjunto único de herramientas de gestión, configuración e implantación acopladas al sistema operativo SUSE Linux Enterprise Server 10, las cuales aligeran las tareas de gestión de sistemas y servidores Linux.

Novell: compañía de origen estadounidense dedicada al software, específicamente en el área de sistemas operativos de redes.

Streaming: Servidor de red dedicado a la transmisión de multimedia en tiempo real.

Licencia CAL: Tipo de licencia privativa que garantiza el acceso de los clientes al software.

Licencia BSD: Licencia de software otorgada principalmente para los sistemas BSD (*Berkeley Software Distribution*). Pertenece al grupo de licencias de Software Libre.

1.3 Sistemas para la gestión de los servidores de red

El objetivo fundamental de esta sección es analizar sistemas para la gestión de servidores en Cuba y el resto del mundo, de esta manera logramos identificar sus principales deficiencias; así como sus potencialidades distintivas y buenas prácticas, para contribuir con ello al refinamiento del producto.



1.3.1 Soluciones acopladas a sistemas operativos

Como primer elemento estudiado tenemos a los sistemas operativos de la familia **SUSE Linux Enterprise** con herramientas para la gestión de servidores Linux que permiten entre otros factores:

- Aligerar las tareas de mantenimiento de sistemas con YaST y AutoYaST
- Gestionar las suscripciones y los derechos de asistencia técnica desde el Centro de servicios al cliente de Novell.

Resulta importante destacar que el paquete integrado de herramientas de interfaz gráfica de usuario YaST, incluido en SUSE Linux Enterprise Server, simplifica las tareas de gestión de servidores. YaST cuenta, además, con una consola de texto con las mismas prestaciones que la interfaz gráfica de usuario, lo que facilita el acceso remoto. Su diseño estructurado en módulos simplifica el uso en los principales ámbitos de gestión de servidores, como la instalación de sistemas, la configuración de red y del hardware, la selección del software y los paquetes, y la configuración de servicios. Por lo que se considerará útil tanto para añadir usuarios como para configurar aplicaciones, como DNS, servidores Web Apache, archivos compartidos con Samba, entre otros.

Por otro lado, también cuenta con la herramienta AutoYaST, la misma constituye una extensión de YaST que automatiza la instalación para un número elevado de equipos. AutoYast permite ahorrar tiempo, ya que pueden llevarse a cabo varias instalaciones al mismo tiempo y sin la intervención del usuario. Se trata de una solución de gestión de servidores Linux adaptada al tamaño y a las necesidades de cada empresa.

SUSE Linux Enterprise Server 10 se integra perfectamente con el Centro de servicios a clientes de Novell, por lo que los clientes y los socios pueden gestionar fácilmente sus suscripciones y derechos de asistencia técnica. Esto garantiza el acceso constante a las actualizaciones de software y a los parches de seguridad. (1)

Sin embargo SUSE Linux Enterprise Server 10 es un sistema operativo Linux de pago, que contiene una inevitable base común entre sus servicios de gestión y su sistema operativo, lo cual provoca restringir los beneficios de la gestión de servidores a una distribución particular. Además aunque existe una fuerte



Capítulo 1: Fundamentación Teórica

alianza con la compañía Novell, Cuba no figura entre la lista de países con derechos a recibir asistencia técnica.

Un segundo elemento a tener en cuenta es el sistema operativo **Red Hat Enterprise Linux** el cual se encuentra actualmente disponible en dos versiones para servidores. Una primera versión considerada básica denominada Red Hat Enterprise Linux Server la cual está elaborada para implantaciones de pequeño tamaño, y una segunda versión llamada Red Hat Enterprise Linux Advanced Platform. Ambas versiones comparten la misma base tecnológica, e incorporan un amplio conjunto de aplicaciones de servidor de código abierto. Además incluye actualizaciones periódicas cada 18 meses, a pesar de ello constituye un sistema operativo de pago, lo cual no representa la mejor alternativa para países pobres o en vías de desarrollo. (2)

No solamente existen soluciones para la gestión de servidores en entornos Linux, sino que además, Windows ha realizado sus propias implementaciones con soluciones como **Windows Server 2003** capaz de manejar una amplia gama de funciones, tanto de manera centralizada como distribuida. Por solo mencionar algunas de ellas tenemos:

- Servidor de archivos e impresión.
- Servidor Web y aplicaciones Web.
- Servidor de correo.
- Servidor de acceso remoto.
- Servicio de directorio.
- Servidor de dominio.
- Servidor DHCP.
- Servidor de Streaming.

Si quisiéramos compararlo con otros sistemas operativos de la familia de Windows, realmente encontraríamos que ofrece ventajas sustanciales:

- Como servidor de ficheros es de un 100% a un 139% más rápido que Windows 2000 Server y un 200% más que Windows NT Server 4.0.



Capítulo 1: Fundamentación Teórica

- Como servidor de impresión, es un 135% más eficiente que Windows NT Server 4.0.
- Como servidor web es de un 100% a un 165% más rápido que Windows 2000 Server. (3)

Para el 2008 la empresa Microsoft productora de la solución anterior desarrolló su versión de **Windows Server 2008** especialmente diseñado para la publicación Web, virtualización de varios sistemas operativos e incremento de su seguridad. (4)

Desafortunadamente los sistemas operativos de la familia de Windows que se dedican a la gestión de servidores de red, aunque ofrecen servicios de muy buena calidad, son soluciones de pago para las cuales es necesario adquirir licencias privativas.

En el primer caso: Windows Server 2003 exige que:

- Cada copia instalada del software de servidor requiere la compra de una licencia de servidor de Windows.
- Se requiere una Licencia de Acceso para Cliente de Windows (CAL de Windows) para poder acceder al uso del software del servidor.

En el segundo caso: Windows Server 2008 con una disposición bastante similar a la anterior define una licencia de servidor para cada una de las instancias que operan con el software de servidor, así como una CAL de Windows Server.

Son estas solo algunas de las disposiciones que establecen las licencias de venta de las soluciones tratadas, las cuales además del cobro del producto no garantizan el acceso a su código fuente; siendo este uno de los puntos claves que hacen rechazar la propuesta de utilización de alguno de estos dos últimos sistemas.

Las soluciones anteriores ya sean para entornos GNU/Linux o Windows, aunque no son las únicas, a pesar de presentar limitantes en uno u otro sentido, implementan muy buenas prácticas que son necesarias para alcanzar niveles de calidad que se encuentren a la altura que exige la comunidad internacional.



1.3.2 Soluciones independientes

No solamente existen tendencias actuales para la gestión de los servidores de redes empleando soluciones fuertemente acopladas a un sistema operativo, sino que además existen propuestas totalmente independientes, y he aquí una de sus principales ventajas, que garantizan un servicio de calidad a sus usuarios; es el caso de **Webmin** el cual representa una herramienta para la configuración de sistemas accesible vía Web desde GNU/Linux y otros sistemas. Permite configurar características propias de los sistemas operativos, tales como servicios, usuarios, cuotas de espacio, archivos, comandos de arranque y apagado, entre otros; así como la instalación y control de muchas aplicaciones, entre las que destacan los servidores: DNS, DHCP, Jabber, Web, Correo, Proxy, FTP, SSH y LDAP. Su control se extiende no solo a los servidores locales, sino también a los remotos y aunque es una aplicación idealmente construida para la Web, no le es imprescindible ejecutarse en un ordenador con interfaz gráfica; pues pudiera emplearse navegadores de consola para garantizar su funcionamiento. Es una aplicación totalmente modular, escrita en el lenguaje Perl, lo cual permite adicionar o quitar variadas funcionalidades. Su licencia inicial, la BSD o Berkeley Software Distribution (en español, Distribución de Software Berkeley) no es exactamente GPL, pero Webmin en su versión “free” es de libre distribución, siempre y cuando no se modifique, pues, en ese caso habrá que enviar las modificaciones realizadas a su autor. También existe una versión comercial que incluye módulos comerciales.

Como parte de las soluciones independientes para gestionar servidores existe un subgrupo de herramientas que ejercen su gestión sobre un servidor en particular, es el caso de **phpLDAPadmin** el cual permite la administración de servidores de Directorio LDAP, escrito totalmente en PHP y basado en una interfaz Web para su gestión. Posee una vista jerárquica basada en árbol en donde se puede navegar por toda la estructura del Directorio. Permite visualizar los esquemas LDAP, realizar búsquedas, crear, borrar, copiar y editar entradas LDAP, incluso copiar entradas entre servidores LDAP.

1.4 Lenguaje de modelado

El **UML** (Lenguaje Unificado de Modelado) es una de las herramientas más atractivas en el mundo actual del desarrollo de sistemas; debido a que asegura la generación de diseños que capturen las ideas en una



forma convencional y fácil de comprender para comunicarlas a otras personas. Constituye el lenguaje de modelado más conocido y utilizado en la actualidad. Aunque no es una guía para realizar el análisis y diseño orientado a objetos, debido a que no es un proceso, UML es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos. Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas que permiten visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. Los diagramas más comunes de UML son clasificados en:

- Diagramas de estructura estática.
- Diagramas de comportamiento.
- Diagramas de implementación.

El objetivo principal de emplear UML radica en ser un lenguaje de modelado de propósito general el cual pueden utilizar todos los modeladores; no posee propietario y está basado en el común acuerdo de gran parte de la comunidad informática. Además pretende trabajar correctamente con la mayoría de los procesos de desarrollo existentes pues incluye conceptos necesarios para utilizar un proceso moderno iterativo, basado en construir una sólida arquitectura para resolver requisitos dirigidos por casos de uso. Conjuga de manera sorprendente sus niveles de simplicidad con la capacidad de modelar toda la gama de sistemas que se necesita construir. (5)

1.5 Metodología de desarrollo

La tendencia actual de la producción de software está dirigida a la construcción de sistemas más grandes y más complejos, debido en parte, al hecho de que los ordenadores son más potentes cada año, y los usuarios, por tanto, esperan más de ellos. Se requiere de un software que esté mejor adaptado a las necesidades y sea más rápido de desarrollar; pues el tiempo que este demora en salir al mercado es otro factor a tener en cuenta. Sin embargo alcanzar todas estas ventajas resulta un tanto difícil, muchos desarrolladores trabajan con los mismos métodos de hace décadas atrás, lo cual, a menos que renueven sus métodos, representa un problema para cumplir con el objetivo de desarrollar software de alta complejidad. Esta situación tiene una base común, radica en la dificultad que afrontan los desarrolladores



Capítulo 1: Fundamentación Teórica

para coordinar sus esfuerzos e integrar múltiples fases del desarrollo, se impone la necesidad de un método o proceso común que:

- Proporcione una guía para ordenar las actividades de un equipo.
- Dirija las tareas de cada desarrollador por separado y del equipo como un todo.
- Especifique los artefactos que deben desarrollarse.
- Ofrezca criterios para el control y la medición de los productos y actividades del proyecto. (6)

La línea divisoria que existe generalmente entre proyectos realmente productivos y otros que fracasan es precisamente la presencia de un proceso bien definido y gestionado. El *Proceso Unificado de Desarrollo* o *Rational Unified Process* (RUP) es una solución a este problema en el área de software, además ha sido resultado de más de 30 años de experiencia. Constituye el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software; representa además un marco de trabajo genérico que permite adaptarse a gran variedad de sistemas, para diferentes áreas de aplicación, diferentes organizaciones, diferentes tamaños de proyecto, entre otros.

El Proceso Unificado emplea el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema de software, por lo que constituye parte esencial de RUP.

El ciclo de vida de RUP se caracteriza por ser:

- Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de este momento los casos de uso guían el proceso de desarrollo, pues los modelos que se obtienen en 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 en la que el equipo del proyecto y los usuarios deben estar de acuerdo. RUP se desarrolla mediante iteraciones, comenzando por los CU relevantes desde el punto de vista de la arquitectura.
- Iterativo e incremental: Propone que cada fase se desarrolle en iteraciones que involucran actividades de todos los flujos de trabajo. Es práctico dividir el trabajo en partes más pequeñas, cada una de ellas es una iteración que resulta un incremento. Las iteraciones hacen referencia a



Capítulo 1: Fundamentación Teórica

pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto.

En RUP se han definido 9 flujos de trabajo y 4 fases que se extienden a lo largo de todo el desarrollo. A continuación son especificados:

Flujos de trabajo:

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

Fases:

- Inicio: Obtener la visión general del proyecto.
- Elaboración: Determinar la arquitectura óptima.



- Construcción: Lograr un producto con capacidad operacional.
- Transición: Obtener el *release* del producto.

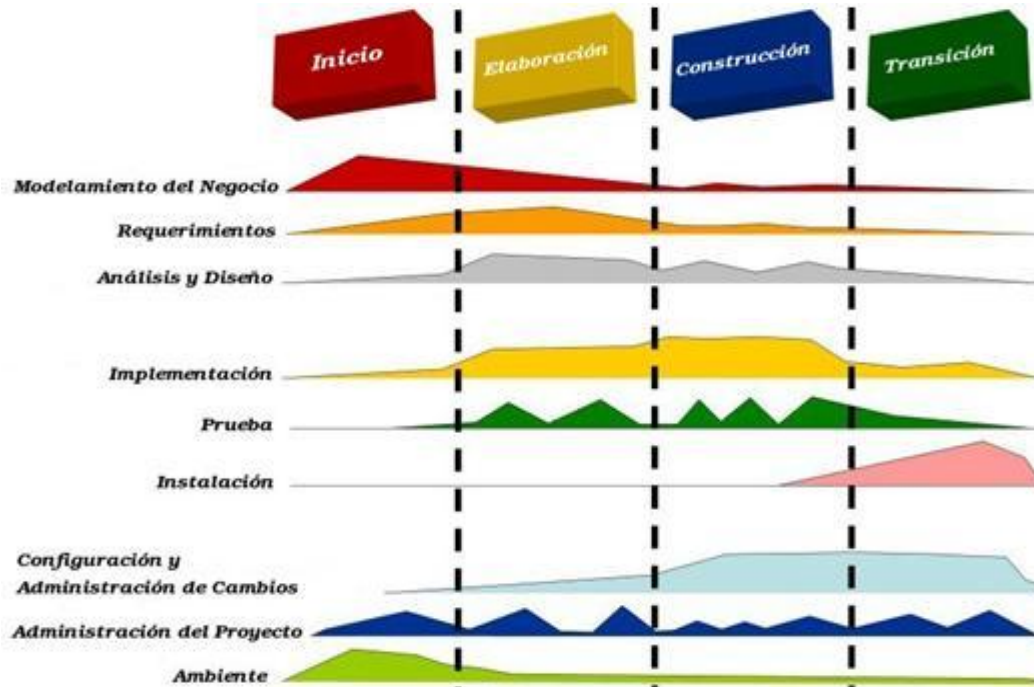


Figura 1: RUP en dos dimensiones

1.6 Modelación de funciones mediante IDEF0

Durante la década del 70 del siglo pasado la Fuerza Aérea de los EEUU desarrolló un programa para la fabricación integrada asistida por computador (Integrated Computer Aided Manufacturing, ICAM). El programa ICAM identificaba las necesidades de mejora en las técnicas y en el análisis de la comunicación para todo el personal que se involucraba en la producción. Como uno de los resultados más notables del proyecto, surgieron una serie de normas conocidas como IDEF (Integration Definition for Function Modeling) cuya traducción literal es definición de la integración para la modelización de funciones. Entre dichas normas se encontraba **IDEF0**, la cual muestra de manera estructurada y jerárquica las actividades que conforman un sistema o empresa.



Capítulo 1: Fundamentación Teórica

IDEFO está caracterizado por ser:

- Comprensivo y expresivo, capaz gráficamente, de representar una variedad amplia de procesos de negocio a cualquier nivel del detalle.
- Coherente y simple, garantizando una expresión rigurosa y exacta.
- De fácil generación por una gran variedad de herramientas gráficas.

Además, permite analizar, documentar y mejorar los procesos del negocio; empleando para ello las notaciones AS-IS y TO-BE. La forma AS-IS es la empleada por el negocio para representar los procesos de la manera actual, antes de la automatización.

La forma TO-BE es utilizada para transformar el proceso a la manera deseada para los objetivos de la organización, pudiendo identificarse mejoras en la disposición de los procesos.

Los elementos gráficos utilizados para la construcción de los diagramas IDEF0 son unos rectángulos o cajas y una serie de flechas.



Figura 2: Componentes básicos de IDEF0

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

Entrada: se representa con una flecha entrando por el lado izquierdo de la actividad, indica los materiales o informaciones que se transformarán en la actividad para obtener la salida.



Capítulo 1: Fundamentación Teórica

Salida: se representa con una flecha saliendo del lado derecho de la actividad, indica los objetos o informaciones producidos por la ocurrencia de la actividad.

Control: se representa con una flecha entrando por la parte superior, indica las regulaciones que determinan si una actividad se realiza o no; tales como normas, guías, reglas, políticas, etc.

Sujeto: se representa con una flecha entrando por la parte inferior, indica los recursos que ejecutan una actividad como personas, maquinarias, entre otros.

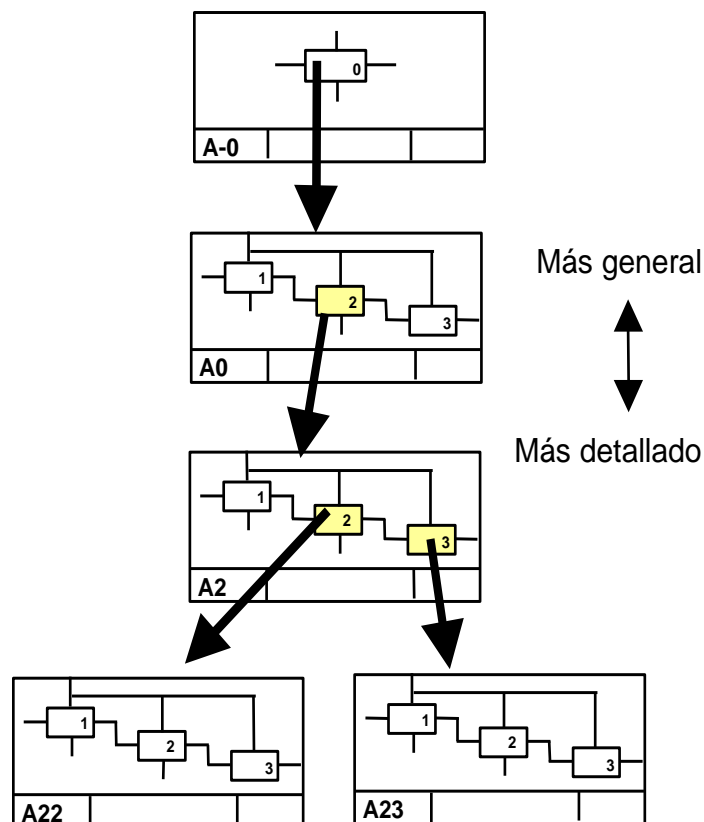


Figura 3: Jerarquía de procesos en IDEF0



1.8 Lenguaje de programación y herramientas

1.8.1 Elección de un lenguaje adecuado

Debatirse en la interrogante de elegir el lenguaje de programación para la construcción de aplicaciones hace muchas veces preguntarse: ¿Cuál es el mejor lenguaje que existe? Si se enfoca la búsqueda de esta manera seguramente se encontrarán criterios tan variados como la cantidad de lenguajes existentes; sería mejor preguntarse: ¿Realmente existe un “mejor lenguaje”? La experiencia en el mundo de la informática ha demostrado que existen lenguajes más o menos adecuados según los propósitos del software que se desea construir. Digamos que si quisiéramos construir un sistema de gestión de información para usar desde la Web; sin lugar a dudas que PHP sería un candidato importante, o si pretendiéramos construir algún tipo de driver en específico pudiera pensarse en lenguajes como C o C++, sin embargo si el objetivo constituye elaborar aplicaciones altamente portables, con facilidad para el manejo de ficheros, de código entendible incluso para quienes se inician en el mundo de la programación; sin lugar a dudas es necesario valorar cuantas ventajas aporta emplear a python como solución integral.

1.8.1.1 Python

Python es un potente lenguaje de programación creado por Guido van Rossum en 1991; representa la elección para sostener el desarrollo del sistema propuesto, entre otros factores por ser:

- Simple: Python constituye un lenguaje simple y minimalista. Poseedor de una estructura natural de pseudo-código que ilustra una de sus grandes fortalezas, permite concentrarse en la solución del problema en lugar de la sintaxis.
- Sencillo de aprender: Es extremadamente sencillo para iniciarse en la programación ya que ofrece una sintaxis extraordinariamente simple.
- Libre y Fuente Abierta: Representa un claro ejemplo de un FLOSS (Free/Libre and Open Source Software - Gratuito/Libre y Software de Fuente Abierta). Lo cual significa que es posible distribuir libremente copias de este software, leer su código fuente, hacerle cambios, usar partes del mismo en nuevos programas libres, entre otros.



- Lenguaje de Alto Nivel: Brinda la posibilidad de despreocuparse por detalles de bajo nivel, como manejar la memoria empleada por tu programa.
- Portable: Debido a su naturaleza Open Source, Python ha sido portado a diversas plataformas sin requerir cambio alguno. Sin embargo, se debe evitar las características dependientes del sistema como librerías o módulos que funcionan sólo en un sistema particular.
- Interpretado: En Python, solo ejecutas el programa desde el código fuente. Internamente, se convierte el código fuente en una forma intermedia llamada *bytecodes*, después los traduce en el lenguaje nativo de tu computadora y se ejecuta. Todo lo cual hace su uso mucho más sencillo. Solo es necesario ejecutar los programas realizados. Esto garantiza su portabilidad, ya que solo se debe copiar el código del programa Python en cualquier otro sistema y trabajará igualmente.
- Orientado a Objetos: Permite programación orientada 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, especialmente, cuando se compara con lenguajes como C++ o Java. (7)

1.8.2 Herramientas Case

1.8.2.1 Visual Paradigm

Visual Paradigm para UML es una herramienta multiplataforma con carácter profesional capaz de soportar el ciclo de vida completo del desarrollo de software. Diseñado para lograr el entendimiento entre un rango amplio de usuarios, que cubre desde los analistas de negocio hasta arquitectos del sistema, integrándose a la construcción de sistemas confiables mediante el uso de la Orientación a Objetos. Visual Paradigm gestiona una rápida construcción de aplicaciones con calidad a un menor costo. Además, soporta UML en su versión 2.1, posibilita el desarrollo de la ingeniería directa e inversa y se encuentra disponible en múltiples versiones y plataformas.

1.8.3 Entorno de Desarrollo Integrado

Un entorno de desarrollo integrado o Integrated Development Environment (IDE) es un programa que le brinda a los desarrolladores de software un conjunto de herramientas que facilitan su desempeño laboral y



proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación existentes, tales como Java, C++, C#, Python, etc. 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.

1.8.3.1 Alternativas aplicables

Realizar la elección de un lenguaje de programación para comenzar el desarrollo de un producto de software, es un elemento que acota futuras decisiones asociadas al entorno de desarrollo a emplear u otros aspectos técnicos. En este sentido, python es realmente flexible pues permite colocar su código desde sencillos editores de texto, incluso en terminales o consolas para usuarios Linux hasta modernos y complejos entornos de desarrollo que lideran el marco internacional de construcción de programas informáticos. Solamente es necesario tener instalado en el ordenador el intérprete de python para lograr entender el código. En los sistemas de la familia GNU/Linux dicho intérprete se encuentra instalado adjunto a la distribución; garantizando un primer e imprescindible paso. Es posible comenzar a escribir programas en las consolas del sistema tan solo colocando el comando *python*, lo cual incluye acceso incluso a la ayuda del lenguaje; dicho ambiente es posible mejorarlo con tan solo instalar *ipython* disponible en la mayoría de los repositorios de paquetes existentes; el cual logra una personalización aun mayor del entorno de trabajo. Los editores visuales de texto ofrecen la posibilidad de generar clases o scripts, teniendo generalmente como única condición el conocimiento de la sintaxis propia del lenguaje; entre ellos son notables las prestaciones que ofrece *Scite* para gestionar el trabajo con más de 30 lenguajes soportados, entre ellos python. Finalmente existen entornos realmente completos como *Easy Eclipse* con capacidad de completamiento de código, ambiente flexible, rápido, entre otras. Los sistemas operativos Windows o Mac OS poseen también una larga lista de opciones para garantizar el desarrollo de aplicaciones en python.

1.8.3.2 EasyEclipse

Easy Eclipse constituye un proyecto de software de código abierto el cual provee un IDE previamente compilado con pluggins de propósitos especiales para la programación en lenguajes como Java, PHP, Ruby, Python, C y C++. Precisamente, **PyDev** es un plugin que permite programar en Python usando



este IDE y es distribuido bajo la Eclipse Public License. Easy Eclipse integra editor de código, depurador e intérprete. Al estar escrito en Java es multiplataforma, por lo que no existen problemas con el sistema operativo en el que se instale. (8)

1.8.4 Interfaces gráficas de usuario

La elección de Python como lenguaje de programación brinda la posibilidad de elegir un estándar adecuado para la construcción de interfaces de usuario acorde a los objetivos. Entre los ejemplos más significativos se encuentran TkInter, wxPython, PyGTK y PyQt.

1.8.4.1 TkInter

Permite construir interfaces gráficas de usuario multiplataforma en Python utilizando el conocido toolkit Tk. Python incluye este módulo por defecto, lo que hace que sea muy popular. TkInter, además, es robusto, maduro y muy sencillo de aprender y de utilizar, contando con una amplia documentación.

TkInter se distribuye bajo la PSFL (Python Software Foundation License) una licencia compatible con la GPL creada para la distribución de software relacionado con el proyecto Python. La PSFL permite crear trabajos derivados sin que estos se conviertan necesariamente en Software Libre.

Sin embargo, presenta algunas desventajas en relación con las herramientas para gestionar su trabajo, problemas de integración con el sistema operativo y una significativa lentitud que afecta la calidad de los productos de software.

1.8.4.2 wxPython

wxPython es posiblemente el toolkit para desarrollo de interfaces gráficas en Python con mayor popularidad, superando incluso a Tkinter. El mismo se incluye por defecto con el intérprete de Python. Cuenta con más y mejores widgets que Tkinter, y ofrece un muy buen aspecto en todas las plataformas, utilizando MFC en Windows y GTK en Linux. Además existen herramientas muy interesantes como wxGlade para diseñar las interfaces gráficas de forma visual.



Sin embargo, carece de cierta falta de consistencia y un estilo muy alejado de Python, que se acerca más al estilo C++. Esto ha provocado que hayan aparecido distintos proyectos para abstraer al programador de los detalles del toolkit, como es el caso de Dabo.

wxPython se distribuyen bajo una licencia "wxWindows Licence", que consiste esencialmente en una LGPL con la excepción de que las obras derivadas en formato binario se pueden distribuir como el usuario crea conveniente.

1.8.4.3 PyGTK

PyGTK cuenta con una API muy clara, limpia y elegante y es, además, muy sencillo de aprender. Cuenta con grandes herramientas para construir la interfaz de forma gráfica, como Glade o Gazpacho. PyGTK se distribuye bajo licencia LGPL.

Aunque se recomienda para cualquier tipo de aplicación, especialmente interesante para *Gnome*; es ligeramente más complicado de instalar y distribuir en sistemas operativos como Mac OS.

1.8.4.4 PyQt

PyQt está poco a poco aumentando su popularidad entre los diseñadores de interfaces para Python. Aunque es un toolkit sencillo de utilizar y con muchas posibilidades. Es especialmente interesante para el desarrollo en KDE, dado que Qt es la librería utilizada para crear este entorno.

No obstante el interés en Qt no se limita a KDE, sino que es una biblioteca multiplataforma que, además, desde la versión 4, utiliza widgets nativos para las distintas plataformas; en contraste a lo que hacia anteriormente al emular el aspecto de la plataforma en la que corría.

Como aplicación de soporte para elaborar prototipos de alta calidad se puede utilizar Qt Designer. PyQt distingue por su sencillez, por poseer un número importantes de herramientas que gestionen su manipulación, por su posibilidad de adecuarse a las distintas plataformas de software. Está especialmente concebido para cualquier tipo de aplicación en especial para las que se ejecutan sobre el entorno KDE. Son solo estas algunas de las características que hacen de PyQt la elección para desarrollar el ambiente



gráfico del sistema de gestión de servicios telemáticos tratados. (9)

1.8.4.5 Qt Designer

En correspondencia con la elección anterior de PyQt como elemento que soportará el diseño de las interfaces se ha decidido emplear Qt Designer como herramienta necesaria para el desarrollo de formularios y presentaciones gráficas de nuestras aplicaciones. Qt Designer permite acelerar el desarrollo de interfaces de alto rendimiento de forma fácil, generando el código fuente para las mismas, lo que permite al desarrollador ajustarlo a sus necesidades. Inicialmente fue creado por la empresa TROLLTECH para trabajar en varias distribuciones Linux, sin embargo actualmente puede instalarse en otras plataformas como Windows y Mac OS. Esta herramienta provee características muy poderosas como la previa visualización de la interfaz, soporte para widgets y un editor de propiedades con gran variedad de opciones.

1.9 Conclusiones

En el capítulo fueron analizados a fondo temas que contribuyen al futuro desarrollo del sistema informático propuesto, centrándose fundamentalmente en la situación y tendencias actuales asociadas a los procesos de administración de servicios telemáticos en Cuba y el resto del mundo; puntualizando aspectos referentes a licencias de uso, garantías, soporte, tipo de solución y buenas prácticas implementadas; para con ello contribuir a la elaboración de un producto que se encuentre a la altura de las necesidades de una sociedad consciente de lo que consume. Finalmente se abordaron los términos generales que conforman las herramientas para automatizar el trabajo; herramientas que cubren desde el diseño de clases e interfaces hasta su posterior implementación.



CAPÍTULO 2 “Características del sistema”

2.1 Introducción

En el presente capítulo se describen de manera concreta los procesos del negocio asociados a la administración del servidor de “Directorio” en sistemas operativos GNU/Linux, con el objetivo primario de comprender su estructura básica; debido a lo cual resulta imprescindible conocer la situación problemática creada a partir del desempeño actual de la organización tratada, realizando un análisis crítico de la misma. Se expresa una propuesta del sistema en términos de calidad asociada a la usabilidad, el soporte, la portabilidad, entre otros aspectos; elementos estos que distinguen el producto de similares en el mercado del software. Además, son presentados las características o requisitos funcionales del sistema que logran suplir la necesidad de los clientes.

2.2 Problema y situación problemática

Cuba cuenta hoy con una institución dedicada a la seguridad en redes informáticas, se trata de la Oficina de Seguridad para las Redes Informáticas (OSRI), entidad nacional asociada al Ministerio de la Informática y las Comunicaciones, que tiene por objeto social llevar a cabo la prevención, evaluación, aviso, investigación y respuesta a las acciones, tanto internas como externas, que afecten el normal funcionamiento de las tecnologías de la información del país; así como trabajar por fortalecer la seguridad. Su propósito consiste en implementar un sistema de planificación que contribuya al ordenamiento de las actividades asociadas con las redes informáticas y de comunicaciones, mediante el establecimiento de un esquema que garantice niveles aceptables de seguridad. (10)

La OSRI como parte de su arduo trabajo realiza reiteradas inspecciones a muchos de los centros nacionales que cuentan con soporte tecnológico para la informática, pudiendo constatar el estado actual que presentan los servidores de red. Con un panorama realmente variado que se extiende desde grandes infraestructuras de redes con sistemas operativos como Windows Server o Red Hat, en alguna de sus versiones; hasta locales con redes muy pequeñas que carecen incluso de los más básicos servicios. Algunos administradores son capaces de gestionar sus servidores y garantizar sus servicios a través de la administración carente de interfaces en entornos GNU/Linux, mientras otros se debaten entre utilizar



Capítulo 2: Características del Sistema

soluciones independientes como Webmin o soluciones acopladas a sistemas operativos. Sin embargo, existe un problema aún mayor que sobrepasa meros conocimientos técnicos, un problema que está vinculado directamente a la consciencia sobre el uso de muchas de estas soluciones de mercado, que en gran cantidad de casos son soluciones de pago que no ofrecen su código fuente, con excesivas restricciones de licencias y a las cuales no tenemos derecho de asistencia y ayuda técnica en caso de necesitarla. La generación actual ha sufrido en gran medida con el flagelo de Windows Server 2003, como exponente claro de restricciones a gran parte de sus usuarios potenciales. Para Cuba ha resultado una tarea realmente difícil comenzar todo un proceso de migración de sus principales servidores de red a Software Libre, y varios son los organismos o entidades que han asumido este nuevo reto, entre ellos; el sistema de información médica (infomed.sld.cu), los Joven Club de Computación y Electrónica, la Aduana General de la República, varias universidades y empresas con directivas precisas, de basar su desarrollo en este paradigma, entre otros. Aunque realmente la primera migración que debe hacerse es en la concepción y las ideas de cada cual, en entender a profundidad la ventaja de ser libres, no solo en el ámbito político y económico, sino también en el tecnológico, en lograr que exista verdadera libertad del conocimiento, que nadie pueda arrebatarse algo que por derecho le pertenece a todos. (11)

A continuación se expresa una de las estructuras que siguen los procesos del negocio de gran parte de las organizaciones cubanas actuales, está asociada a la utilización de soluciones acopladas a sistemas operativos con restricciones de soporte, licencias y costo. La modelación de estos procesos de negocio está sustentada en la notación IDEF0 para garantizar una visión clara de los mismos en el marco de las organizaciones; en la cual participan los clientes y el equipo de desarrollo con los objetivos de identificar los procesos del negocio, definir sus objetivos y recursos implicados, entre otros.

Mediante esta técnica los desarrolladores explican a los directivos y trabajadores implicados los elementos gráficos que corresponden a IDEF0, elaborando de manera conjunta el modelo de procesos **as-is** de esta novedosa técnica. El **as-is** no es más que la modelación del cómo ocurren de manera global los procesos de la organización en su situación actual.



Capítulo 2: Características del Sistema

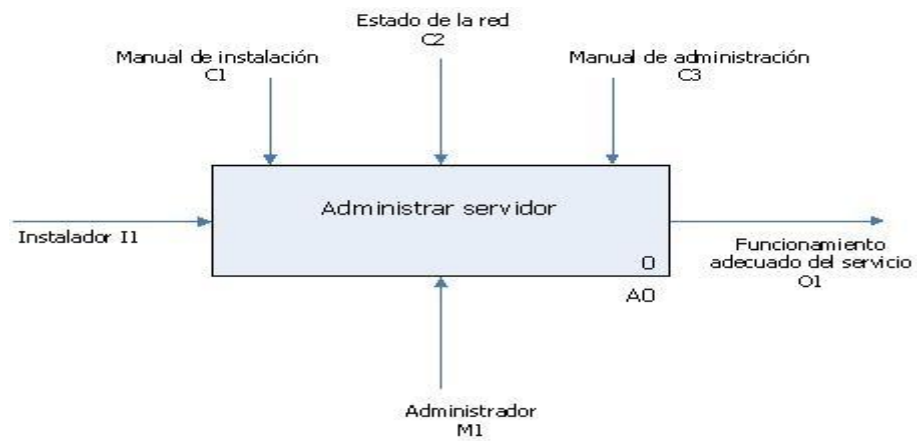


Figura 4: Representación del proceso base empleando as-is

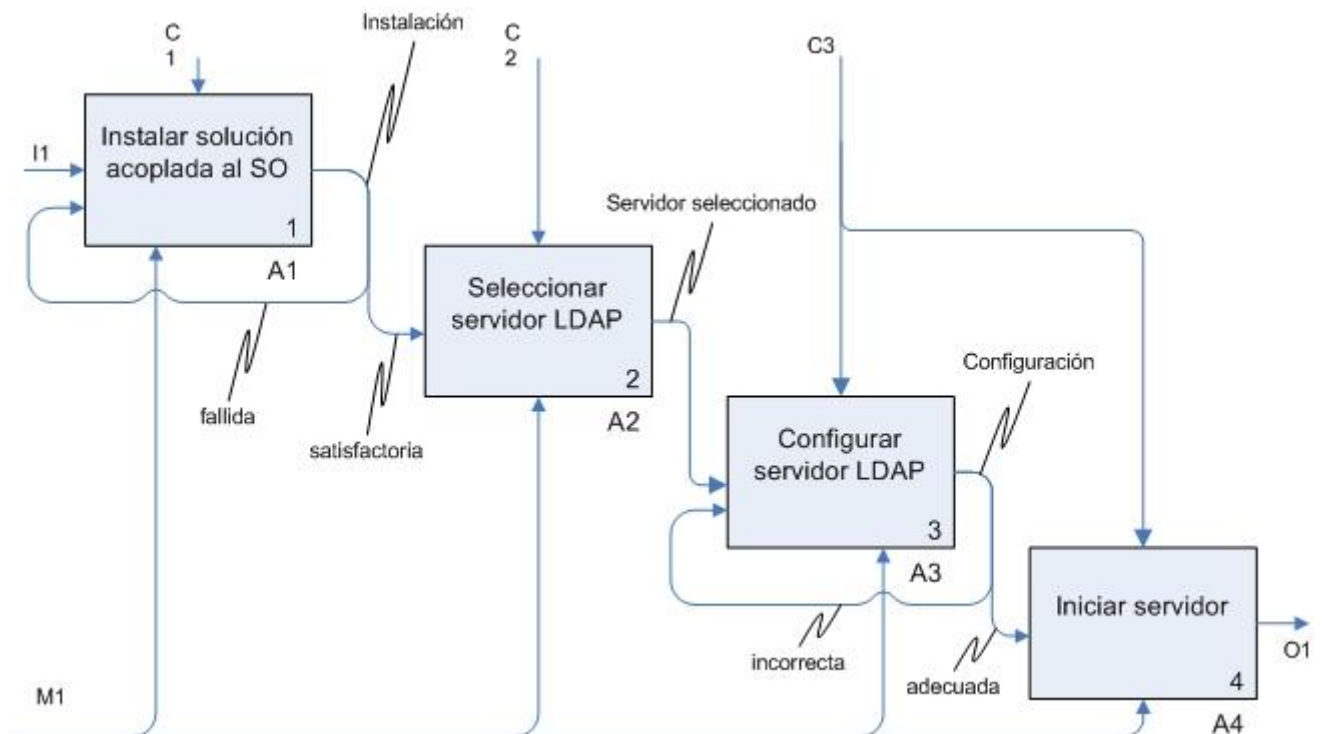


Figura 5: Detalles del proceso base empleando as-is



2.2.1 Descripción de los procesos de negocio

- Instalar solución acoplada al SO: El proceso se inicia cuando el administrador de red guiado por un manual de instalación sigue los pasos necesarios para la instalación de una solución de software asociada a un sistema operativo como Windows Server, Red Hat, o cualquier otro que permita gestionar el trabajo con los servidores de la red.
- Seleccionar servidor LDAP: El administrador una vez instalado el sistema operativo procede a la selección del servidor que brindará el servicio, en este caso el servidor LDAP, para lo cual debe poseer un conocimiento del estado actual de su red referente a aspectos tales como: servidores previamente instalados, hardware existente, número de usuarios potenciales, entre otros aspectos.
- Configurar servidor LDAP: Consiste en adecuar un grupo importante de parámetros del servidor a las necesidades propias de la organización en que se encuentre implantado.
- Iniciar servidor: Tras haber ejecutado los procesos anteriores de selección y configuración del servidor debemos iniciarlo. Iniciar un servidor significa tenerlo listo para comenzar a brindar su servicio en la red, basado en las configuraciones realizadas previamente.

2.3 Objeto de automatización

La construcción de este sistema pretende automatizar varios de los procesos tratados en la sección anterior debido a la importancia que ello representa. En próximas secciones serán especificados otros procesos de negocio que resultan significativos en el marco del trabajo, pero que hasta este momento no han sido tratados. Hasta este punto pueden definirse como procesos a automatizar los siguientes:

- Seleccionar servidor LDAP
- Configurar servidor LDAP
- Iniciar servidor

2.4 Información que se maneja

Resulta importante lograr comprender a plenitud como funcionan aspectos relacionados con el tratamiento de la información del sistema. Se hace imprescindible entonces, analizar los procesos existentes; entre ellos uno muy especial que permite entender lo anteriormente planteado: “Configurar servidor LDAP”,



proceso este que actualiza ficheros relativos a las configuraciones que por defecto acompañan a los servidores. La generalidad de los documentos tratados hace referencia a archivos de configuración asociados a los servidores que se instalan; particularmente el marco de este trabajo se encuentra limitado al servidor de directorio. Los archivos de configuración constituyen prácticamente el núcleo del sistema; una incorrecta estructura de su contenido pudiera ocasionar un inadecuado trabajo en su servidor asociado, incluso pudiera ser un motivo para deshabilitarlo y por consiguiente ocasionar la culminación del servicio brindado con afectación directa sobre los consumidores del mismo.

2.5 Solución propuesta

Se ha decidido contribuir a la eliminación de la problemática existente elaborando una aplicación de escritorio que gestione el trabajo con el servidor de directorio, la cual permita instalarlo desde su código fuente en los más variados sistemas operativos de la plataforma GNU/Linux; característica esta que lo hace superior a muchas de las soluciones de mercado que existen actualmente. La instalación desde los ficheros fuentes garantiza un mayor aprovechamiento de la arquitectura del ordenador.

No solamente será esta la única característica del sistema, sino que permitirá realizarle configuraciones a los servidores de manera local o remota. Este último aspecto alcanza su cima en entornos donde los administradores de redes se enfrentan a la difícil tarea de administrar un servidor ubicado en un ordenador carente de ambiente gráfico, debido a que por cuestiones principalmente de eficiencia los ordenadores dedicados a dicha tarea destinan sus recursos a obtener un funcionamiento adecuado y no un marco de trabajo agradable para cualquier tipo de usuario. Además, al estar construido esencialmente con python, permite ser una solución fácilmente portable a otros sistemas operativos.

Un producto con las distinciones anteriores, pero que además no presente limitantes por concepto de costos de licencias, soporte o mantenimiento para la comunidad cubana, con la posibilidad de ser portado a otros sistemas operativos de la familia Windows, con un ambiente de trabajo agradable y profesional para los usuarios, es sin lugar a dudas un recurso invaluable para los administradores de la red.

2.6 Modelo de negocio

En secciones anteriores se ha tratado el término IDEF0 como notación para el modelado de los procesos



de negocio, opción esta que facilita el entendimiento entre los desarrolladores y clientes. Existe en este ámbito una actividad denominada “modelado de procesos globales mejorados” que implica al equipo de desarrollo y pretende actualizar el modelo de procesos con las mejoras identificadas. En esta actividad se actualiza el modelo de procesos de la organización. Añadiendo al modelo una breve descripción de como se realiza cada actividad. En este punto se realizan en el modelo los cambios que impliquen una propuesta de mejora en los procesos. Estos cambios deberían basarse en la experiencia y el estudio del arte realizado previamente a la etapa de modelación del negocio, por parte del equipo de desarrollo sobre procesos de negocio similares a nivel nacional e internacional. Este nuevo modelo se corresponde con el modelo de procesos **to-be** de IDEF0.

Posteriormente se deben validar las mejoras propuestas con el cliente para establecer un acuerdo acerca de cómo deberían ser los procesos de la organización, antes de pasar a su adecuada informatización. En esta actividad el equipo de desarrollo presenta el *Modelo de Procesos Globales Mejorados* al cliente, para que este indique su conformidad con la propuesta o realice los señalamientos que considere pertinentes.

2.6.1 Representación de los procesos con IDEF0

Se ha definido un proceso general denominado “Administrar Servidor”, que a su vez está dividido en otros seis procesos que detallan su funcionamiento:

- Instalar solución independiente al sistema operativo (SO)
- Seleccionar servidor LDAP
- Compilar paquetes
- Instalar servidor compilado
- Configurar servidor LDAP
- Modificar estado



Capítulo 2: Características del Sistema

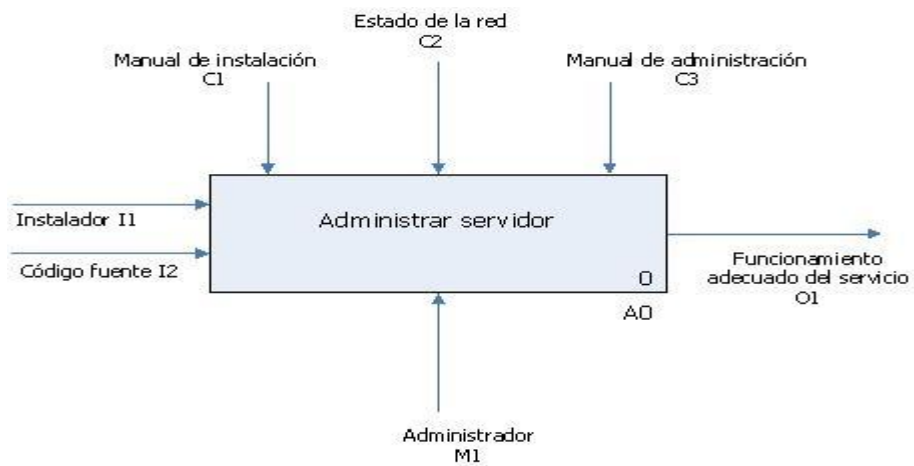


Figura 6: Representación del proceso base empleando to-be

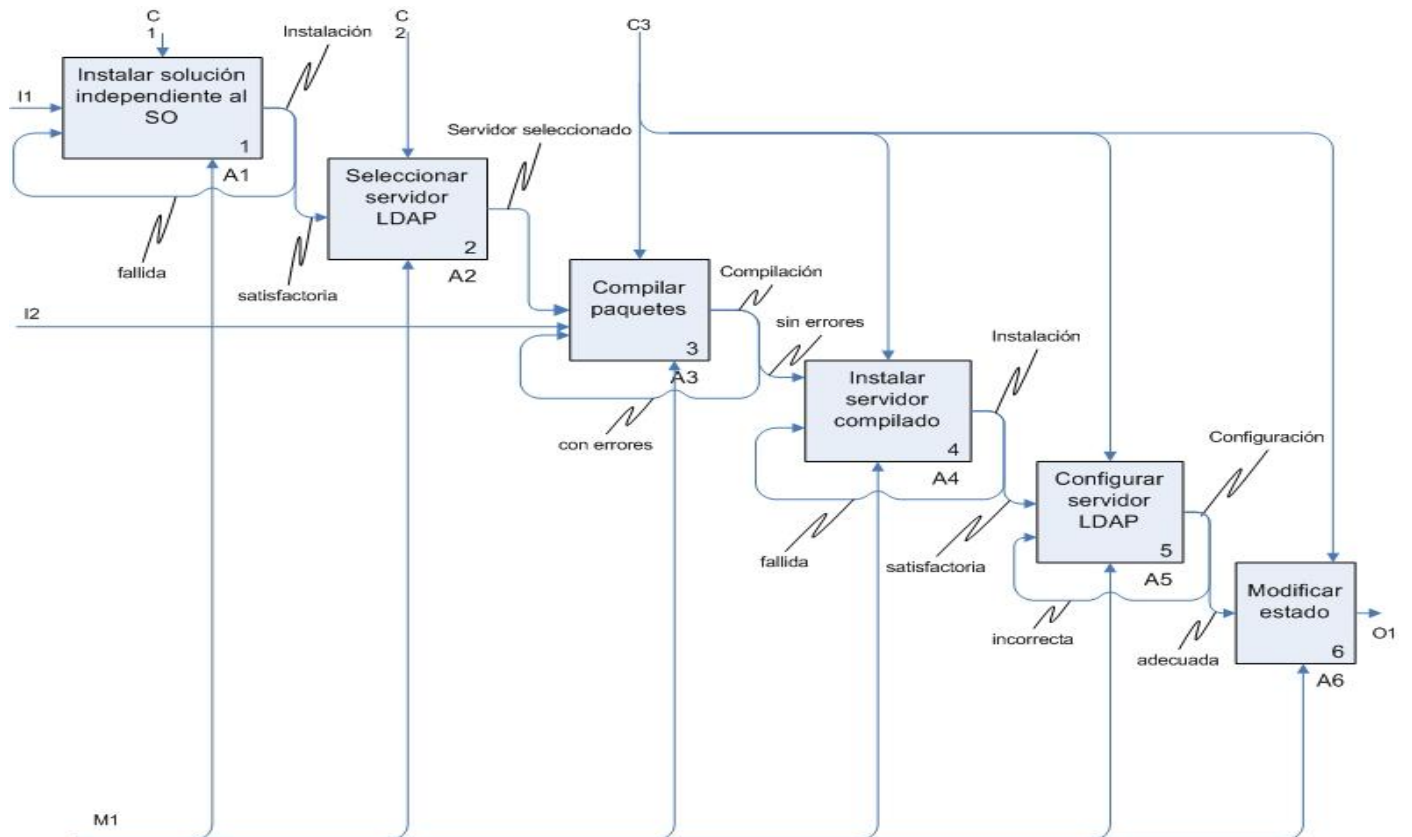


Figura 7: Detalles del proceso base empleando to-be



2.6.2 Descripción detallada de los procesos de negocio

- Instalar solución independiente al sistema operativo: El proceso consiste en instalar una solución para la gestión del servidor de directorio totalmente desacoplada del sistema operativo del ordenador.
- Seleccionar servidor LDAP: Tras haber instalado el sistema gestor debemos seleccionar el o los servidores que brindaran su servicio a la red tratada, en este caso particular está referido al servidor LDAP.
- Compilar paquetes: Se orienta a compilar los paquetes o servidores previamente seleccionados en el paso anterior. Una de las ventajas fundamentales del proceso de compilación de paquetes está ligada al aprovechamiento de la arquitectura del ordenador sobre el cual se ejecuta dicho proceso.
- Instalar servidor compilado: La instalación del servidor es un proceso totalmente imprescindible para lograr brindar el servicio, consiste en hacer residir el servidor en una estación de trabajo específica, para desde ahí, comenzar a brindar el servicio a los clientes.
- Configurar servidor LDAP: Instalar un servidor no es suficiente, es necesario configurarlo para ajustarlo a las características propias del entorno. La configuración hace referencia al servidor de directorio; donde pudieran tratarse aspectos como la gestión de puertos, certificados digitales, réplicas, entre otros aspectos relevantes. Aunque estos son solo algunos de los aspectos que pudieran modificarse, en cualquier momento puede procederse a una nueva configuración de la estructura del servidor, solo basta con modificar el estado del servidor de manera que se actualice los cambios realizados.
- Modificar estado: La configuración de cualquier parámetro del servidor no es aceptada hasta tanto el servidor no se reinicia y es capaz de usar la nueva información para brindar su servicio. Los servidores de red presentan por lo general un mínimo de tres operaciones permitidas sobre sus estados: detenido, iniciado o reiniciado.

En la sección 2.3 fueron detallados un grupo de procesos a automatizar por el sistema; como la selección, configuración e iniciación del servidor; sin embargo la modelación del negocio con IDEF0 empleando la técnica **to-be** expande un tanto la visión de negocio existente y propone algunas modificaciones a los procesos y su orden de aparición en el contexto de la organización. Debido a ello ha



sido necesario incluir como procesos a automatizar los tratados en el presente epígrafe. Los procesos expuestos en el epígrafe 2.3 quedan contenidos en esta nueva decisión basada en las ventajas de proponer mejoras a la estructura organizativa existente.

2.7 Especificación de los requisitos de software

La modelación del negocio brinda una visión general de los procesos que existen en un determinado entorno, permitiendo comprender a que se dedica el mismo, estableciendo una comunicación entre el cliente y el equipo de desarrollo. Una vez en este punto, se hace necesario comenzar a definir que es lo que debe hacer el sistema, siendo necesario ir a la captura de los requerimientos que este debe cumplir. Debido a ellos todas las ideas que los clientes, usuarios y miembros del equipo de proyecto tengan acerca de lo que debe hacer el sistema, deben ser analizadas como candidatas a requisitos.

Lograr una comunicación efectiva entre los usuarios y el equipo de proyecto con el objetivo de llegar a un entendimiento de lo que hay que hacer, es la clave del éxito en la producción de un software con calidad. Durante años muchas aplicaciones han fallado debido a incongruencias entre lo que el usuario deseaba, lo que realmente necesitaba, lo que interpretaba cada miembro del equipo de proyecto y lo que realmente se obtiene.

2.7.1 Requerimientos funcionales

Los requerimientos funcionales constituyen capacidades o condiciones que el sistema debe cumplir, los mismos se mantienen invariables sin importar con que propiedad o cualidad se relacionen. A continuación serán tratados los requerimientos funcionales asociados al módulo:

2.7.1.2 Requerimientos funcionales del módulo Directorio

R1. Autenticar usuario.

R2. Gestionar entrada.

R2.1 Agregar entrada al servidor de directorio.



Capítulo 2: Características del Sistema

R2.2 Modificar entrada existente en el servidor.

R2.3 Eliminar entrada existente en el servidor.

R2.4 Listar entradas existentes en el servidor.

R3. Buscar entradas

R4. Obtener puertos existentes.

R5. Gestionar puertos de acceso al servidor de directorio.

R5.1 Agregar puerto de acceso.

R5.2 Modificar puerto de acceso al servidor de directorio.

R5.3 Eliminar puerto de acceso al servidor de directorio.

R6. Gestionar estado del servicio de directorio.

R6.1 Detener servicio

R6.2 Iniciar servicio

R6.3 Reiniciar servicio

R7. Certificar servidor.

R8. Gestionar réplicas para garantizar la distribución de la información.

R8.1 Agregar réplica al servidor de directorio.

R8.2 Modificar réplica del servidor.



R8.3 Eliminar réplica del servidor.

R9. Gestionar configuraciones del servidor.

R9.1 Exportar parcial o totalmente la configuración del directorio.

R9.2 Importar configuraciones hacia el directorio.

R10. Mostrar estructura general del directorio.

2.7.1 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Representan características que hacen al producto atractivo, rápido, confiable, entre otras. En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requerimientos funcionales, es decir una vez se conozca lo que el sistema debe hacer podemos determinar cómo ha de comportarse y qué cualidades debe tener.

Requerimientos de software:

- Las estaciones de trabajo donde se instalará cualquiera de los servidores deben tener asociada algún sistema operativo 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:

Los requisitos de hardware están asociados fundamentalmente al tamaño de la red y a la cantidad de usuarios que recibirán el servicio. Es importante acotar que los servidores debido a su funcionalidad específica de atender las solicitudes de un grupo de clientes necesitan de una estructura de hardware



Capítulo 2: Características del Sistema

superior que les permita dar cumplimiento a su labor. Si se tratase de una pequeña red, con servicio para algunas decenas de usuarios, podría emplearse:

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

Incluso pudiera operar correctamente en condiciones inferiores; sin embargo, si el tamaño de la red aumentara y se tratase ya de algunos miles de usuarios, el cambio más significativo estaría en aumentar la capacidad de almacenamiento empleando discos duros en RAID como una de las mejores alternativas para las grandes empresas, además debe emplearse un procesador con capacidad de multiprocesamiento. Por otra parte se hace necesario aumentar la capacidad de memoria RAM hasta un rango de 2 a 8 GB.

Restricciones en el diseño y la implementación:

Este tipo de requerimiento especifica o restringe la codificación o construcción del sistema, constituyen restricciones que deben ser cumplidas estrictamente.

- 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 fundamentalmente en dos grupos: herramientas de diseño y herramientas de codificación; en el primer caso se ha seleccionado *Qt Designer* en su versión 4.2.1 para elaborar las interfaces de usuario, en el segundo caso aunque python permite extender su codificación a los más simples editores de texto, se ha decidido emplear *Eclipse con Pydev*. Pydev es un *pluggin* o complemento para el entorno de desarrollo Eclipse que permite la programación en python; estos complementos constituyen una forma de expandir programas de forma modular, de manera que se puedan añadir funcionalidades sin afectar a las ya existentes.
- Como arquitectura general del sistema se ha concluido la utilización de una arquitectura en capas debido a las ventajas que ello proporciona en términos de claridad, facilidad en la corrección de



Capítulo 2: Características del Sistema

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 para resolver problemas puntuales asociados a la gestión de servidores.

Requerimientos de apariencia o interfaz externa:

Particularmente este tipo de requerimiento describe la apariencia que tendrá el producto, sin llegar a tratar el diseño de la interfaz a un nivel alto de detalle.

- El sistema deberá contener una interfaz agradable, sencilla y sugerente; presentará por defecto el color azul en todas sus vistas, 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 iconos 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 asociados a la seguridad son uno de los cuales provocaría los mayores riesgos si no se manejan correctamente. El sistema se ha centrado en:

- Garantizar la autenticación al sistema empleando contraseñas cifradas.



Capítulo 2: Características del Sistema

- Ofrecer protección contra acciones no autorizadas o que puedan afectar la integridad de los datos mediante mecanismos de autenticación eficientes y políticas implementadas por la empresa.
- La gestión de la información del sistema se realizará de forma centralizada por el administrador.
- Garantizar la verificación de acciones irreversibles como la eliminación de los datos.
- La conexión al servidor para gestionar el trabajo con sus ficheros de configuración hará uso del protocolo SSH.

Requerimientos de usabilidad:

Este tipo de requerimientos describen los niveles apropiados de usabilidad, dados los usuarios finales del producto, para ello deben revisarse las especificaciones de los perfiles de usuarios y sus niveles de experiencia; en este caso los administradores de redes. Debe conocerse además que tipo de producto necesitan y que tipo de requisito haría el producto adecuado para ellos. Por tanto, se han definido de la siguiente manera:

- 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:

El soporte es un aspecto realmente amplio y abarca 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. Puede incluir: pruebas, adaptabilidad, mantenimiento, compatibilidad, configuración, servicios, instalación y portabilidad, por solo mencionar algunos.

- El sistema contará antes de su puesta en marcha con un período de pruebas que permitirá identificar y corregir errores cometidos durante el desarrollo, posteriormente se brindará el servicio de instalación del producto.
- Se generará la documentación apropiada para agilizar su mantenimiento y configuración.



- Se ofrecerán cursos de adiestramiento al personal implicado en el manejo del software.
- 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 general de gestión de servidores 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.
- Será un producto con la posibilidad de extenderlo a otros idiomas en futuras versiones del sistema.
- Se deben respetar los términos empleados normalmente por los especialistas en el tema de la esfera que se automatiza.

2.8 Modelo de Casos de Uso del Sistema

2.8.1 Actores del Sistema

En el módulo de directorio interviene un actor: el administrador del servidor, como agente principal del mismo.

Actores	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

2.8.2 Casos de Uso del Sistema (CUS)

Los casos de uso son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. Por lo tanto, establece un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema.

2.8.2.2 CUS del módulo “Directorio”

1. Autenticar usuario.
2. Gestionar entrada.
3. Buscar entradas
4. Obtener puertos.
5. Gestionar puertos de acceso.
6. Gestionar estado.
7. Certificar servidor.
8. Gestionar réplicas.
9. Gestionar configuraciones.
10. Mostrar estructura general.

CU - 1	Autenticar usuario
Actor	Administrador
Descripción	El CUS consiste en brindar la posibilidad de autenticar un usuario como administrador para garantizar la seguridad del sistema de manera general, con ello se asegura que solo accede y modifica la información quien realmente tenga



Capítulo 2: Características del Sistema

	permisos para ello. La autenticación necesita de un usuario, una contraseña y la dirección del servidor de directorio en la red.
Referencia	R1

Tabla 2: Descripción CU Autenticar usuario

CU - 2	Gestionar entrada
Actor	Administrador
Descripción	Consiste en ofrecerle al administrador la posibilidad de adicionar, modificar, eliminar y listar entradas en el servidor de directorios. Entiéndase por entrada todo aquel elemento que en el ámbito del servidor de directorio puede ser gestionado; entre ellos: cuentas de usuario, grupos, unidades organizativas, computadoras, entre otras.
Referencia	R2

Tabla 3: Descripción CU Gestionar entrada

CU - 3	Buscar entradas
Actor	
Descripción	Funcionalidad de apoyo al CU Gestionar entrada, siempre se ejecuta cuando se traten de hacer operaciones como las de adición y listado.
Referencia	R3

Tabla 4: Descripción CU Buscar entradas

CU - 4	Obtener puertos
---------------	------------------------



Capítulo 2: Características del Sistema

Actor	Administrador
Descripción	Mediante esta funcionalidad el administrador del sistema podrá visualizar los puertos que se encuentran activos en el servidor de directorios para dar servicio al mismo, generalmente el 389 y el 636.
Referencia	R4

Tabla 5: Descripción CU Obtener puertos

CU - 5	Gestionar puertos de acceso
Actor	
Descripción	La gestión de puertos, específicamente la agregación, modificación o eliminación de los mismos se realiza a partir de la funcionalidad de certificación del servidor. Cuando un servidor comienza a utilizar un servidor para garantizar que las conexiones que establezcan los clientes se produzcan de manera cifrada, entonces debe comenzar a utilizarse el puerto 636.
Referencia	R5

Tabla 6: Descripción CU Gestionar puertos de acceso

CU - 6	Gestionar estado
Actor	
Descripción	Representa un CU de apoyo o inclusión para los CU: <i>Certificar servidor</i> y <i>Gestionar réplicas</i> ; mediante el cual es posible cambiar el estado del servicio que brinda el servidor de directorio



Capítulo 2: Características del Sistema

	a: detenido, iniciado o reiniciado. Si un servicio en GNU/Linux se encuentra en ejecución y realizamos sobre alguno de sus ficheros de configuración una modificación, dicha modificación no será tomada en cuenta hasta tanto no sea recargado o reiniciado el servicio.
Referencia	R6

Tabla 7: Descripción CU Gestionar estado

CU - 7	Certificar servidor
Actor	Administrador
Descripción	Si queremos establecer una conexión segura entre los clientes y el servidor de directorio será necesario emplear un certificado digital del lado del servidor; a ello se dedica el presente CU.
Referencia	R7

Tabla 8: Descripción CU Certificar servidor

CU - 8	Gestionar réplicas
Actor	Administrador
Descripción	Es importante para garantizar la permanencia de la información que se almacena en el servidor de directorio, que la misma sea replicada en otros servidores similares para evitar afectaciones directas sobre los consumidores de los servicios de red en caso de alguna irregularidad sobre el servidor primario. Mediante la gestión de réplicas el



	administrador podrá agregar un nuevo servidor de replicación, modificarlo y eliminarlo en cualquier momento.
Referencia	R8

Tabla 9: Descripción CU Gestionar réplicas

CU - 9	Gestionar configuraciones
Actor	Administrador
Descripción	La gestión de configuraciones se enfoca en realizar operaciones de exportación e importación de configuraciones empleando ficheros LDIF.
Referencia	R9

Tabla 10: Descripción CU Gestionar configuraciones

CU - 10	Mostrar estructura general
Actor	Administrador
Descripción	El funcionamiento del CU se centra en mostrar la estructura general del servidor de directorio, incluyendo unidades organizativas, grupos, usuarios, recursos, entre otros.
Referencia	R10

Tabla 11: Descripción CU Mostrar estructura general

2.8.3 Diagrama de paquetes

Con el objetivo de hacer más claro el modelo de casos de uso éste se puede dividir en elementos más pequeños y manejables. Los paquetes, que es el nombre que reciben estos elementos, tienen el propósito de organizar en grupos funcionalidades semejantes. En el módulo de directorio quedaron agrupados los



casos de usos según su funcionalidad y los diagramas de paquetes se conformaron de la siguiente manera:

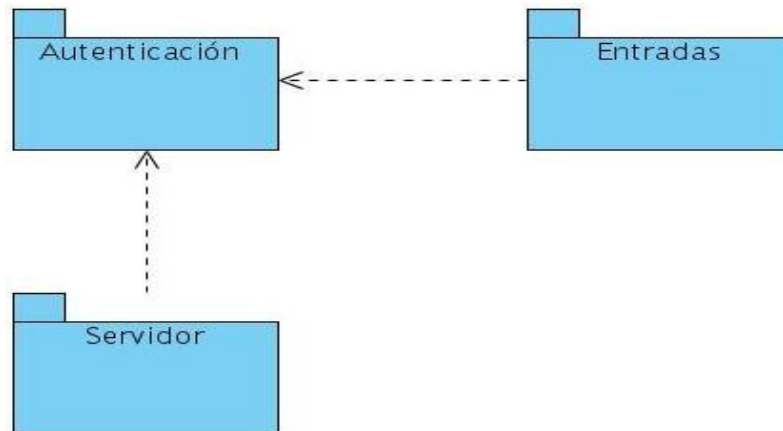


Figura 8: Diagrama de paquetes del módulo Directorio

2.8.4 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.

Cada caso de uso debe comunicarse con al menos un actor, si no aparece ningún actor que se comunique con un caso de uso esto indica error en el modelo de caso de uso o en los requerimientos planteados. Como excepciones a esta última regla podrían considerarse:

- Si el caso de uso es abstracto (no instanciable) no tiene porque incluir relación con actores (aunque puede tenerlas).
- Un caso de uso Padre en una relación de generalización-especialización no tiene porque tener relación con un actor si el caso de uso Hijo describe completamente toda la relación con el actor.
- Algunos casos de uso se inician acorde a un horario (ejemplo: una vez a la semana o al día) lo que significa que el reloj del sistema es su iniciador. El reloj es interno al sistema, de esta forma el caso de uso no tendría relación de iniciación con ningún actor, pero para esclarecer el modelo debe



colocarse un actor ficticio denominado "Reloj". (12)

2.8.4.1 DCUS del módulo: "Directorio"

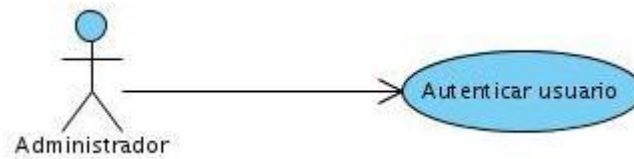


Figura 9: DCUS del paquete de Autenticación

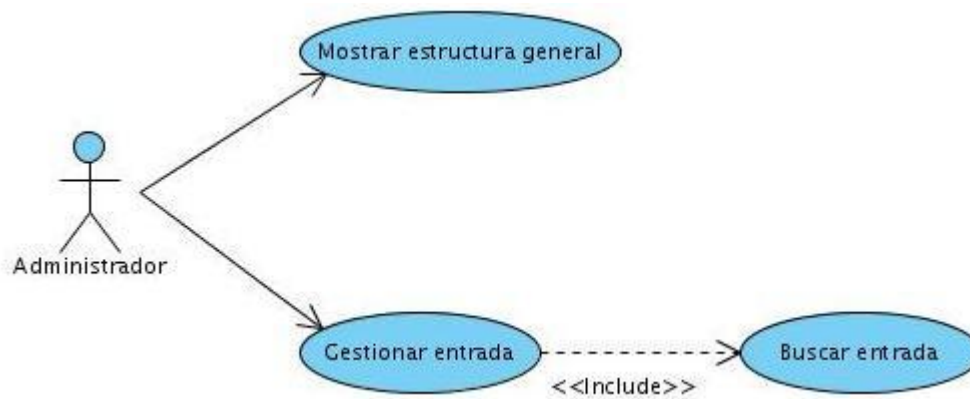


Figura 10: DCUS del paquete de Entradas

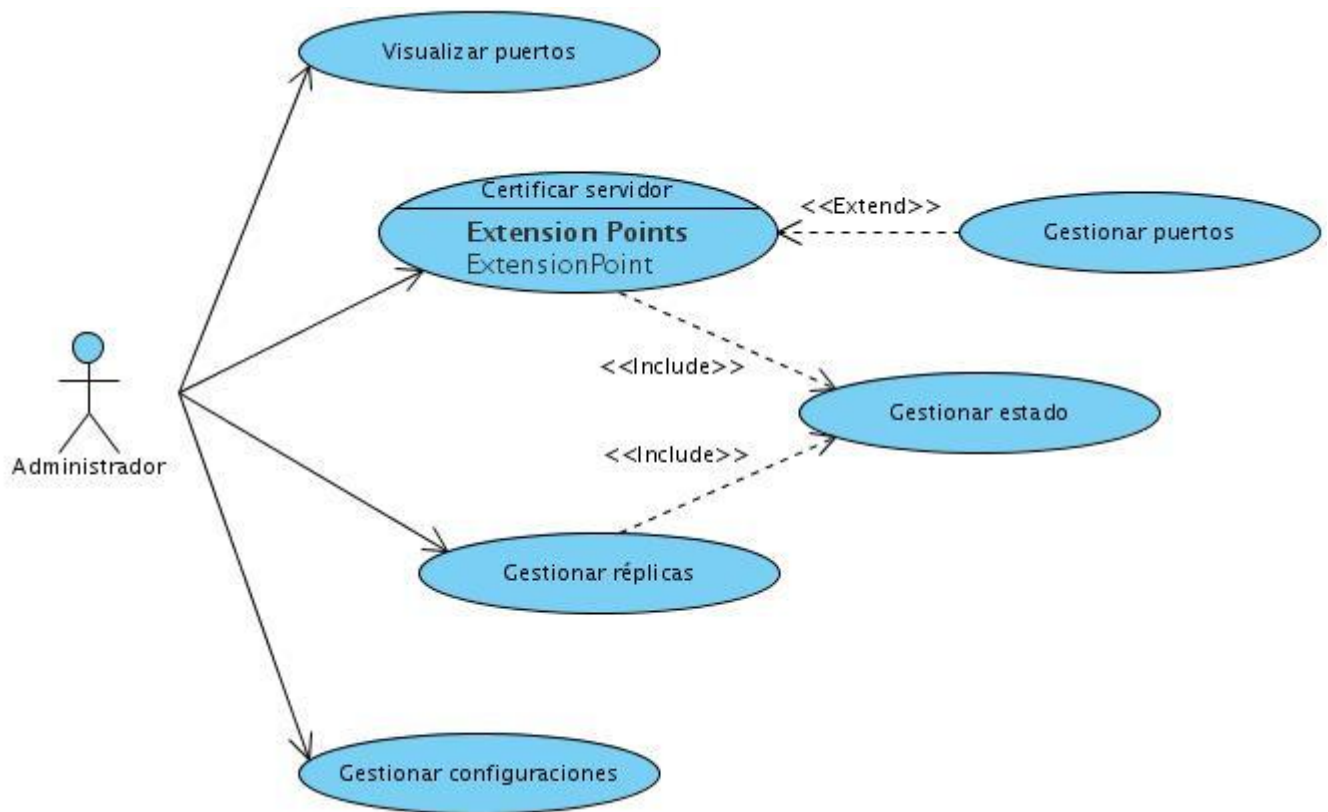


Figura 11: DCUS del paquete de Servidor

2.8.5 Descripción detallada de los CUS y prototipos de interfaces

La descripción detallada de los CUS describe como interactúa el sistema con sus actores, por lo cual constituye un elemento sumamente útil para lograr comprender en toda su amplitud el funcionamiento de el sistema. Todo lo anteriormente descrito será recogido íntegramente en el presente documento.

En esta sección se maneja frecuentemente el término entrada, referente sobre todo a elementos como cuentas de usuario, grupos, unidades organizativas, computadoras, entre otros. De manera genérica pudiera definirse a la entrada como un elemento que pertenece a una o varias clases establecidas por los estándares del servidor OpenLDAP. Cada una de estas clases presentan atributos necesarios y opcionales; debido a ello cuando realizamos operaciones como la adición de una nueva entrada; resulta imprescindible que se coloque toda la información asociada a los atributos obligatorios que están



Capítulo 2: Características del Sistema

presentes en las clases que forman a la entrada. El *Anexo 1* muestra las distintas entradas existentes con sus clases asociadas y el *Anexo 2* muestra el listado de clases con sus atributos necesarios y opcionales. Además, existen clases que definen la estructura de las entradas y cada entrada tiene asociada una única clase de este tipo; debido a que resultaría imposible que clases que definen la estructura de las unidades organizativas y otras que lo hacen de las cuentas de usuario estén asociadas a una misma entrada.

Caso de Uso:	Autenticar usuario	
Actores:	Administrador	
Resumen:	El caso de uso se inicia cuando al administrador del servidor de directorio desea ejecutar alguna de las tareas administrativas permitidas, para ello necesita colocar sus credenciales. Posteriormente el sistema las valida y asigna los permisos necesarios.	
Precondiciones:	El servidor de directorio debe haber sido instalado correctamente, asignando a su configuración un usuario con privilegios administrativos y la correspondiente contraseña.	
Referencias	R1	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Administrador ejecuta el software de gestión.	2. El sistema muestra una interfaz donde le solicita al Administrador su nombre distintivo y contraseña; así como el nombre o dirección IP donde reside el servidor de directorio.	
3. El Administrador introduce los datos correspondientes y presiona el botón aceptar.	4. El sistema valida los datos y brinda acceso al servidor de directorio.	
Flujos Alternos		
4.1 Si los datos no son válidos el sistema muestra el mensaje “Datos no válidos” y		



automáticamente pasa a la acción 2.

Poscondiciones	Se autenticó debidamente el Administrador del servidor de directorio.
-----------------------	---

Tabla 12: Descripción detallada CU Autenticar usuario



Figura 12: Interfaz del CU Autenticar Usuario

Caso de Uso:	Gestionar entrada
Actores:	Administrador
Resumen:	El caso de uso se inicia cuando el Administrador desea registrar, modificar, eliminar o visualizar una o varias entradas en el servidor de directorio. Las entradas representan objetos que contienen información, entre las más notables; aunque existen otras encontramos: cuentas de usuario, grupos, unidades organizativas y computadoras. Los detalles de las entradas serán tratados en el



Capítulo 2: Características del Sistema

	Anexo 1 del presente documento.
Precondiciones:	El Administrador debe estar autenticado en el sistema.
Referencias	R2.1, R2.2, R2.3, R2.4
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Administrador del servidor selecciona la opción: a) Registrar entrada en el servidor, perteneciente a la sección de registro. b) Modificar entrada del servidor, perteneciente a la sección de modificación. c) Eliminar entrada del servidor, perteneciente a la sección de eliminación. d) Listar entradas registradas hasta el momento en el servidor, perteneciente a la sección de visualización.	
Sección “Registrar entrada”	
Acción del Actor	Respuesta del Sistema
	2. El sistema muestra un formulario con los campos requeridos para registrar una nueva entrada. Ver <i>Anexos 1 y 2</i> .



Capítulo 2: Características del Sistema

3. El Administrador introduce los datos correspondientes y presiona el botón aceptar.	4. El sistema valida y registra los datos de la entrada.
Flujos Alternos	
4.1 Si los datos no son válidos el sistema muestra el mensaje “Datos no válidos” y automáticamente pasa a la acción 2.	
Sección “Modificar entrada”	
Acción del Actor	Respuesta del Sistema
	2. El sistema muestra un formulario con la estructura en forma de árbol del servidor de directorio, en cuya estructura se ubicará el listado de entradas registradas hasta el momento.
3. El Administrador selecciona la entrada que desea modificar.	4. El sistema muestra los datos de la entrada seleccionada.
5. El Administrador modifica los datos deseados y presiona el botón aceptar.	6. El sistema valida los datos modificados y actualiza la información asociada a la entrada.
Flujos Alternos	
5.1 Si los datos no son válidos el sistema muestra el mensaje “Datos no válidos” y automáticamente pasa a la acción 4 del flujo normal de eventos.	
Sección “Eliminar entrada”	
Acción del Actor	Respuesta del Sistema
	2. El sistema muestra un formulario con la estructura en forma de árbol del servidor de directorio, en cuya estructura se ubicará el listado de entradas registradas hasta el momento.
3. El Administrador selecciona la entrada que desea eliminar.	4. El sistema muestra en el panel derecho de la aplicación los datos referentes a la entrada y un botón que ofrece la posibilidad de eliminar la entrada.



5. El Administrador da clic en el botón nombrado “Eliminar este objeto”.	6. El sistema muestra un mensaje de alerta notificando que el usuario está a punto de ejecutar una operación irreversible.
7. El Administrador acepta ejecutar la operación.	8. El sistema elimina completamente la entrada seleccionada, incluso de manera recursiva en caso de que la misma posea otras entradas anidadas.
Sección “Visualizar entrada”	
Acción del Actor	Respuesta del Sistema
	2. El sistema muestra un formulario que expresa los datos requeridos para visualizar entradas de acuerdo con parámetros como el nombre y sus atributos.
3. El Administrador coloca la información requerida por el sistema para efectuar el proceso de visualización y presiona el botón aceptar.	4. El sistema apoyándose en la funcionalidad de búsqueda implementada muestra el listado de las entradas que se ajustan a los parámetros definidos anteriormente.
Flujos Alternos	
4.1 Si los datos para iniciar el proceso de visualización introducidos por el administrador no son correctos, se emitirá un mensaje con el texto “Datos no válidos” y automáticamente se pasará a la sección 2 del flujo normal de eventos.	
Poscondiciones	Se registró, modificó, eliminó o listó una o varias entradas.

Tabla 13: Descripción detallada CU Gestionar entrada

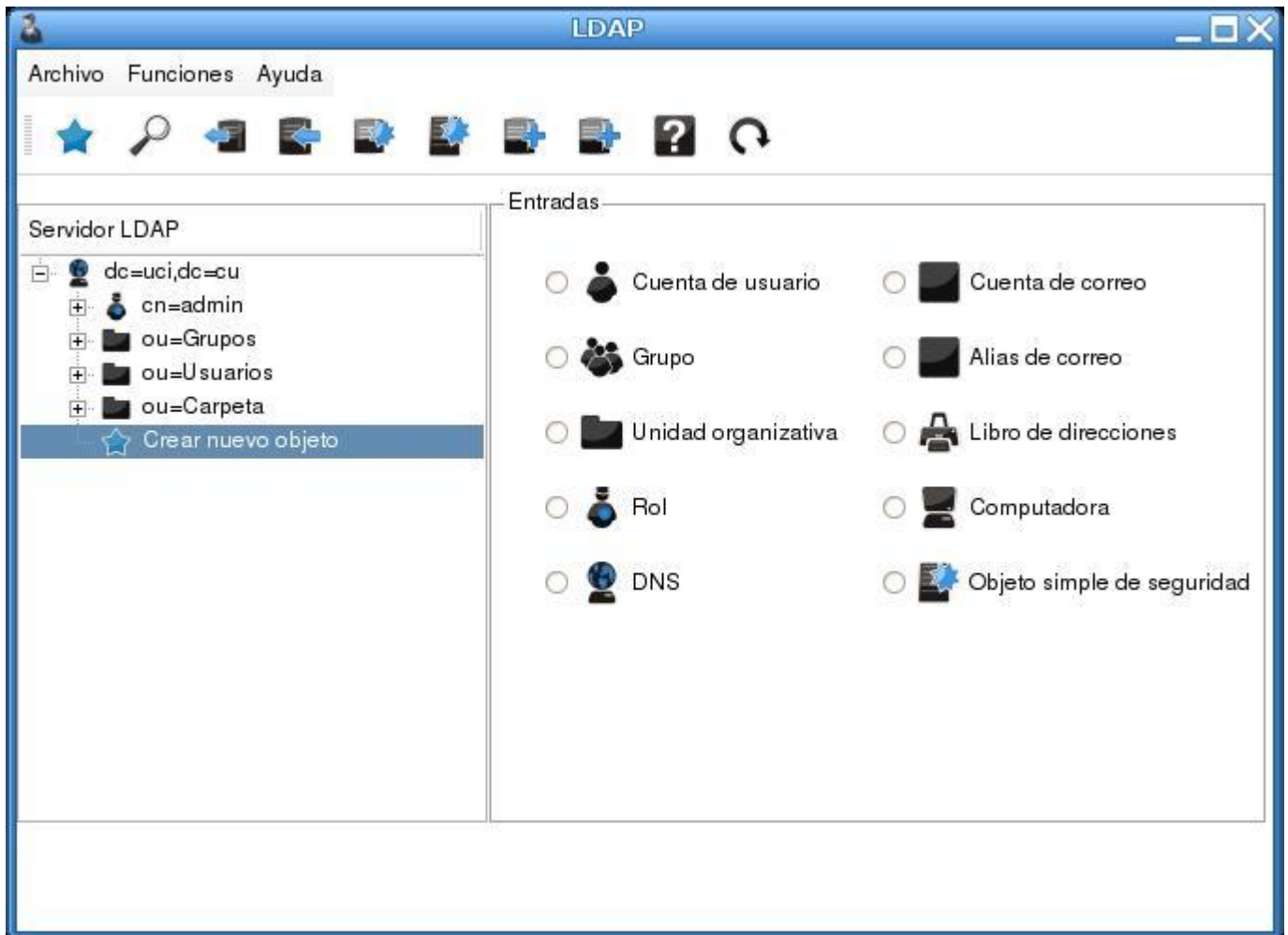


Figura 13: Interfaz del CU Gestionar entradas, sección de Adición. Parte I



LDAP

Archivo Funciones Ayuda

Servidor LDAP

- dc=uci,dc=cu
 - cn=admin
 - ou=Grupos
 - ou=Usuarios
 - ou=Carpeta
 - Crear nuevo objeto

Cuenta de usuario

Nombre:

Apellidos:

Nombre Común:

Usuario:

Contraseña:

Repetir contraseña:

Algoritmo de cifrado: MD5

Grupo: Grupo1 '3000'

Directorio:

Aceptar Cancelar

Figura 14: Interfaz del CU Gestionar entradas, sección de Adición. Parte II

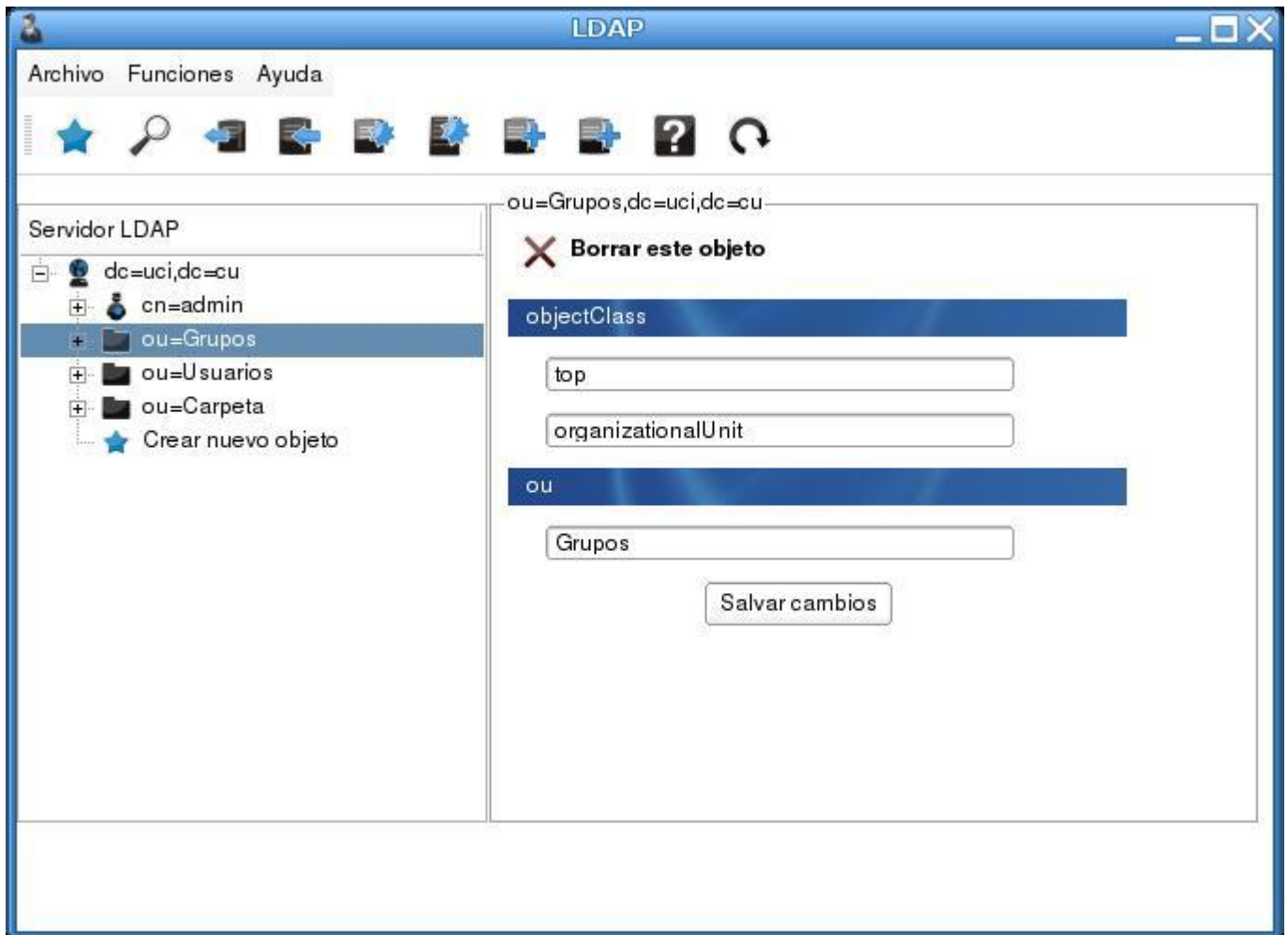


Figura 15: Interfaz del CU Gestionar entradas, sección de Eliminación



Capítulo 2: Características del Sistema

Caso de Uso:	Buscar entrada	
Actores:		
Resumen:	El caso de uso se inicia debido a un evento interno del sistema durante los procesos para registrar, modificar, eliminar o visualizar una o varias entradas en el servidor de directorio. Su función es realizar una búsqueda sobre las entradas existentes.	
Precondiciones:	El Administrador debe haber iniciado alguno de los casos de uso para registrar, modificar, eliminar o visualizar entradas del servidor.	
Referencias	R3	
Prioridad	Crítico	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
	1. El sistema se encarga de devolver una entrada con todas sus características para ser posteriormente modificada, visualizada, eliminada o solo para asegurar la existencia o no de la referida entrada.	
Poscondiciones	Se realizó la búsqueda de una o varias entradas.	

Tabla 14: Descripción detallada CU Buscar entrada

Caso de Uso:	Obtener puertos	
Actores:	Administrador	
Resumen:	El caso de uso se inicia cuando el Administrador desea visualizar los puertos disponibles para el servidor de directorio.	
Precondiciones:	El Administrador debe estar autenticado en el sistema.	
Referencias	R4	
Prioridad	Auxiliar	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Administrador del servidor		



Capítulo 2: Características del Sistema

selecciona la opción:	
a) Listar puertos de acceso registrados hasta el momento en el servidor, perteneciente a la sección de visualización.	
Sección “Visualizar puerto de acceso”	
Acción del Actor	Respuesta del Sistema
	2. El sistema muestra un listado con todos los puertos de acceso registrados hasta el momento.
Poscondiciones	Fueron listados los puertos de acceso al servidor de directorio.

Tabla 15: Descripción detallada CU Obtener puertos

Caso de Uso:	Gestionar puertos
Actores:	
Resumen:	El caso de uso se inicia opcionalmente mediante un evento interno del sistema cuando se trata de certificar el servidor, por tanto se requiere el funcionamiento del puerto 636. Otras modificaciones sobre los puertos también son efectuadas a partir de los procesos que genera el trabajo con certificados.
Precondiciones:	El Administrador debe haber realizado operaciones de certificación en el sistema.
Referencias	R5.1, R5.2, R5.3
Prioridad	Auxiliar
Flujo Normal de Eventos	
Sección “Registrar puerto de acceso”	
Acción del Actor	Respuesta del Sistema



Capítulo 2: Características del Sistema

	1. El sistema es certificado por primera vez. Se agrega el puerto 636 a las configuraciones del servidor de directorio para permitir el acceso seguro.
Sección “Modificar puerto de acceso”	
Acción del Actor	Respuesta del Sistema
	1. El sistema recibe operaciones de certificación que puede conllevar a cambios en los puertos de acceso al servidor de directorio.
Sección “Eliminar puerto de acceso”	
Acción del Actor	Respuesta del Sistema
	1. Se eliminan todos los certificados que hasta el momento hayan sido utilizados por el servidor, por tanto el sistema deja de brindar sus servicios por el puerto 636.
Poscondiciones	Se registró, modificó o eliminó los puertos de acceso al servidor de directorio.

Tabla 16: Descripción detallada CU Gestionar puertos

Caso de Uso:	Gestionar estado
Actores:	
Resumen:	El caso de uso se inicia debido a un evento interno del sistema durante los procesos para certificar el servidor o replicar su información. Consiste en modificar el estado del servidor a detenido, iniciado o reiniciado.
Precondiciones:	El Administrador debe haber iniciado alguno de los casos de uso para certificar o replicar el servidor.
Referencias	R6
Prioridad	Auxiliar
Flujo Normal de Eventos	



Capítulo 2: Características del Sistema

Acción del Actor	Respuesta del Sistema
	1. El sistema se encarga de reiniciar el servicio de directorio tras alguna operación de certificación o replicación.
Flujos Alternos	
1.1 Si el servicio muestra algún error tratando de reiniciarse, entonces se da la posibilidad al Administrador de editar directamente el archivo de configuración asociado y posteriormente iniciar o detener el servicio.	
Poscondiciones	Se restableció el servicio de directorio con las configuraciones impuestas.

Tabla 17: Descripción detallada CU Gestionar estado

Caso de Uso:	Certificar servidor
Actores:	Administrador
Resumen:	El caso de uso se inicia cuando el administrador del servidor desea establecer conexiones seguras entre el servidor y los clientes, por lo cual se asocia el certificado digital al servidor.
Precondiciones:	El Administrador debe estar autenticado en el sistema.
Referencias	R7
Prioridad	Auxiliar
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Administrador del servidor selecciona la opción para la construcción del certificado digital.	2. El sistema muestra una interfaz solicitando los datos requeridos del nuevo certificado.
3. El Administrador introduce los datos correspondientes y presiona el botón aceptar.	4. El sistema valida y registra el nuevo certificado.
Flujos Alternos	



4.1 Si los datos no son válidos el sistema muestra el mensaje “Datos no válidos” y automáticamente pasa a la acción 2.

Poscondiciones	Fue certificado el servidor de directorio.
-----------------------	--

Tabla 18: Descripción detallada CU Certificar servidor

Caso de Uso:	Gestionar réplicas
Actores:	Administrador
Resumen:	El caso de uso se inicia cuando el Administrador desea registrar, modificar o eliminar una o varias réplicas en el servidor de directorio.
Precondiciones:	El Administrador debe estar autenticado en el sistema.
Referencias	R8.1, R8.2, R8.3
Prioridad	Auxiliar
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
<p>1. El Administrador de la red selecciona la opción:</p> <p>a) Registrar réplicas en el servidor, perteneciente a la sección de registro.</p> <p>b) Modificar réplicas del servidor, perteneciente a la sección de modificación.</p> <p>c) Eliminar réplicas del servidor, perteneciente a la sección de eliminación.</p>	
Sección “Registrar réplicas”	
Acción del Actor	Respuesta del Sistema



Capítulo 2: Características del Sistema

	2. El sistema muestra un formulario con los campos requeridos para registrar una nueva réplica.
3. El Administrador introduce los datos correspondientes y presiona el botón aceptar.	4. El sistema valida y registra los datos de la réplica.
Flujos Alternos	
4.1 Si los datos no son válidos el sistema muestra el mensaje “Datos no válidos” y automáticamente pasa a la acción 2.	
Sección “Modificar réplicas”	
Acción del Actor	Respuesta del Sistema
	2. El sistema muestra un listado con las réplicas registrados hasta el momento.
3. El Administrador selecciona la réplica que desea modificar.	4. El sistema muestra los datos de la réplica seleccionada.
5. El Administrador modifica los datos deseados y presiona el botón guardar.	6. El sistema valida los datos modificados y actualiza la información asociada a la réplica.
Flujos Alternos	
5.1 Si los datos no son válidos el sistema muestra el mensaje “Datos no válidos” y automáticamente pasa a la acción 4 del flujo normal de eventos.	
Sección “Eliminar réplicas”	
Acción del Actor	Respuesta del Sistema
	2. El sistema muestra un listado con las réplicas registrados hasta el momento.
3. El Administrador selecciona la réplica que desea eliminar.	4. El sistema elimina la réplica seleccionada.
Flujos Alternos	



4.1 Si no hay una réplica seleccionada el sistema muestra el mensaje “Debe seleccionar una réplica”.	
Poscondiciones	Se registró, modificó o eliminó una o varias réplicas.

Tabla 19: Descripción detallada CU Gestionar réplicas

Caso de Uso:	Gestionar configuraciones	
Actores:	Administrador	
Resumen:	El caso de uso se inicia cuando el Administrador desea importar hacia el servidor o exportar desde el servidor la configuración en formato LDIF.	
Precondiciones:	El Administrador debe estar autenticado en el sistema.	
Referencias	R9.1, R9.2	
Prioridad	Auxiliar	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Sistema	
1. El Administrador de la red selecciona la opción: a) Exportar la configuración del servidor, perteneciente a la sección de exportación. b) Importar la configuración seleccionada, perteneciente a la sección de importación.		
Sección “Exportar configuración”		
Acción del Actor	Respuesta del Sistema	
	2. El sistema muestra un formulario con los campos requeridos para exportar la configuración del servidor	



Capítulo 2: Características del Sistema

	total o parcialmente.
3. El Administrador selecciona la sección del servidor de directorio que desea exportar y presiona el botón <i>Exportar</i> .	4. El sistema exporta los datos seleccionados por el administrador en formato LDIF hacia un fichero según lo haya previsto el usuario.
Flujos Alternos	
4.1 Si no se ha seleccionado ninguna sección para exportar y se presiona el botón aceptar, el sistema muestra el mensaje “Datos no válidos” y automáticamente pasa a la acción 2.	
Sección “Importar configuración”	
Acción del Actor	Respuesta del Sistema
	2. Para lograr importar configuraciones al servidor de directorio desde un fichero LDIF, el sistema mostrará un formulario que permita cargar el fichero desde alguna ubicación del ordenador.
3. El Administrador selecciona la ubicación del fichero y presiona el botón <i>Importar</i> .	4. El sistema valida los datos del nuevo fichero y actualiza la información contenida en el servidor de directorio.
Flujos Alternos	
4.1 Si la estructura del fichero LDIF no es válida el sistema muestra el mensaje “Datos no válidos” y automáticamente pasa a la acción 2 del flujo normal de eventos.	
Poscondiciones	Se logró importar o exportar la configuración del servidor de directorio.

Tabla 20: Descripción detallada CU Gestionar configuraciones

Caso de Uso:	Mostrar estructura general
Actores:	Administrador
Resumen:	El caso de uso se inicia cuando el Administrador accede al sistema,



	después de haber sido correctamente autenticado, para visualizar la estructura general del servidor de directorio.
Precondiciones:	El Administrador debe haber accedido al sistema.
Referencias	R10
Prioridad	Crítico
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El Administrador accede al servidor colocando su usuario y contraseña.	2. En caso de presentar las credenciales adecuadas, el sistema muestra la estructura en forma de árbol del servidor de directorio.
Flujos Alternos	
2.1 Si el servidor verifica que los datos de autenticación son incorrectos, envía un mensaje de “Datos no válidos” y remite al usuario a la sección de autenticación nuevamente.	
Poscondiciones	Se logró mostrar la estructura general del servidor de directorio.

Tabla 21: Descripción detallada CU Mostrar estructura general

Las operaciones descritas anteriormente asociadas a la importación o exportación de configuraciones del servidor, la gestión de puertos y réplicas; así como la certificación del servidor se realizarán a través del menú principal del sistema, en la opción *Funciones*.

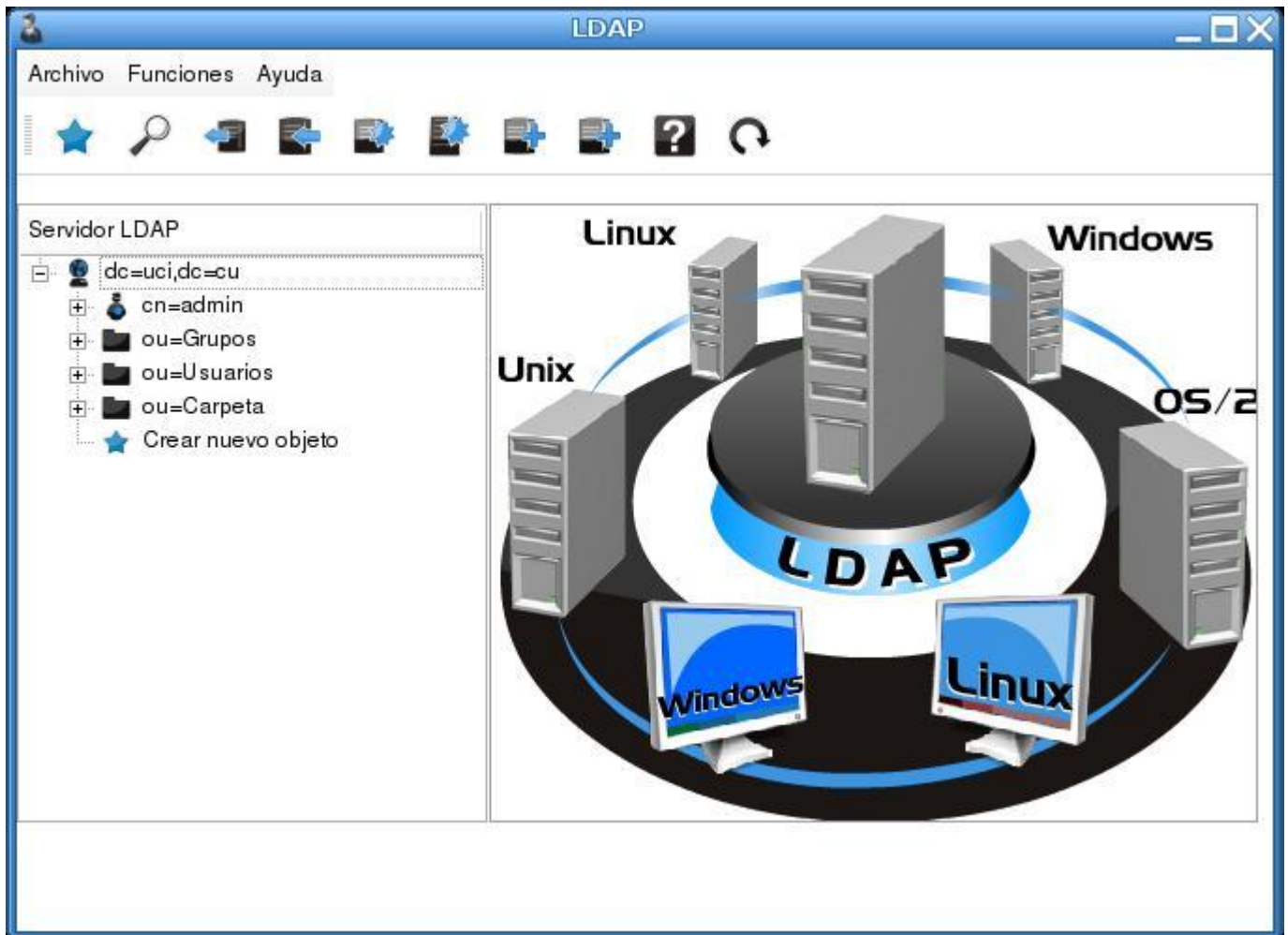


Figura 16: Interfaz general del sistema

2.9 Conclusiones

En el presente capítulo se realizó un estudio detallado de los procesos fundamentales del negocio que sirvieron de guía para la correcta identificación de los requisitos del sistema. Posteriormente fueron identificados los casos de uso del sistema, los cuales constituyen una entrada esencial para el flujo de trabajo de “Análisis y Diseño”. Además, el objeto de estudio; centrado básicamente en el análisis de la situación problemática, el objeto de automatización y la propuesta de sistema complementan una visión general del funcionamiento de la aplicación.



CAPÍTULO 3 “Análisis y Diseño del Sistema”

3.1 Introducción

Como uno de los resultados principales del flujo de trabajo de requerimientos se obtiene una vista externa del sistema, reflejando lo que se espera del mismo a través del Diagrama de Casos de Uso. A partir de este punto se hace imprescindible profundizar en los Casos de Uso, detallándolos; de manera que se logre obtener una vista de la estructura interna del sistema, descrita en el lenguaje de los desarrolladores de software. Dicha vista refleja las clases necesarias para dar cumplimiento a las funcionalidades descritas por los Casos de Uso. De acuerdo con la metodología RUP este proceso debe efectuarse fundamentalmente durante la fase de Elaboración asociada al ciclo de vida del producto correspondiente al flujo de Análisis y Diseño.

El presente capítulo tiene como objetivo la realización del Modelo de Análisis con las clases que lo componen, el Diagrama de Clases del Diseño de cada caso de uso base, así como los diagramas de secuencia del diseño.

3.2 Modelo de Análisis

3.2.1 Diagrama de Clases del Análisis

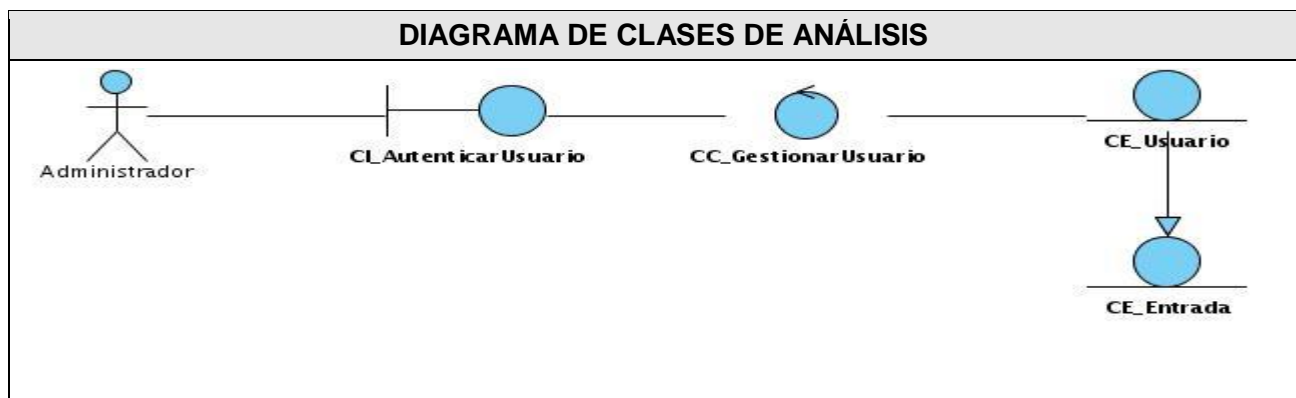


Figura 17: Diagrama de Clases del Análisis: CU Autenticar usuario

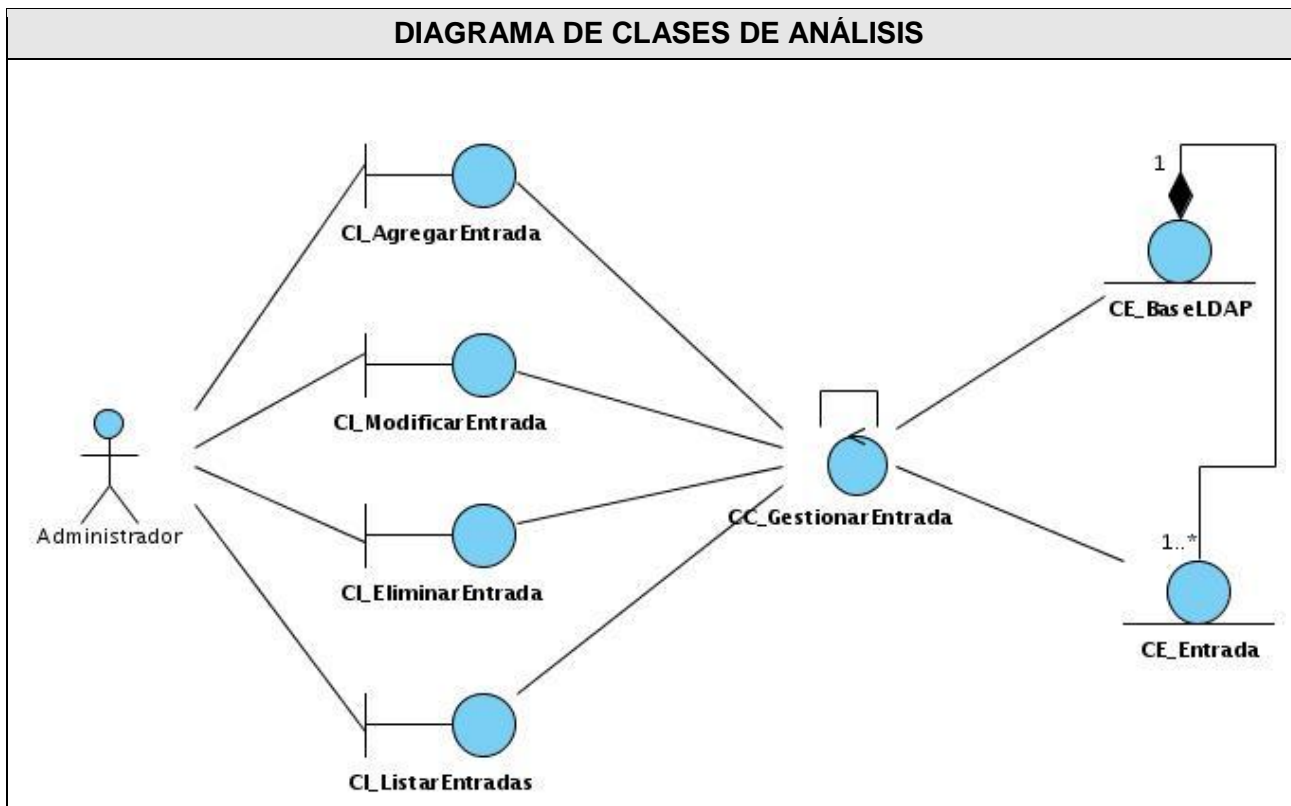


Figura 18: Diagrama de Clases del Análisis: CU Gestionar entrada

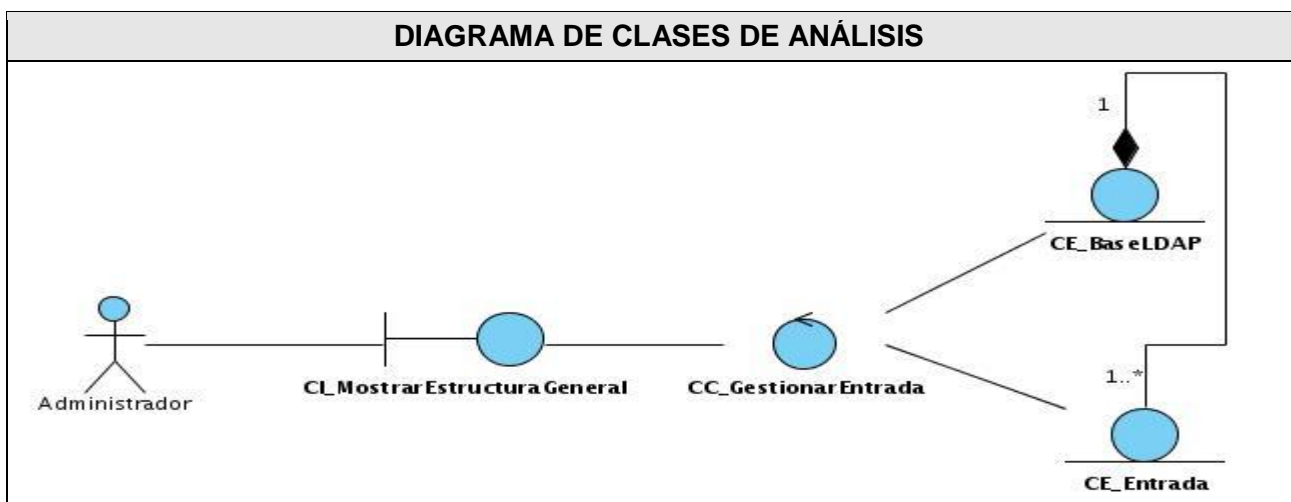


Figura 19: Diagrama de Clases del Análisis: CU Mostrar estructura general

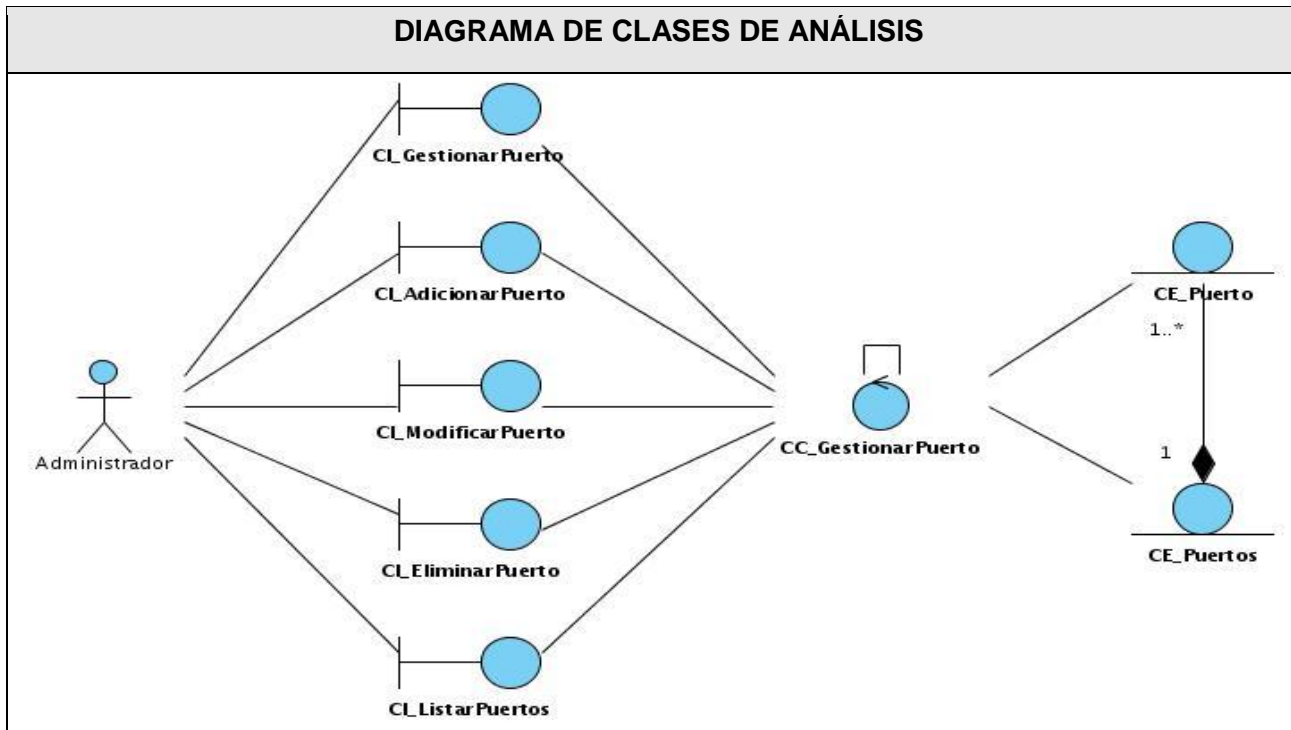


Figura 20: Diagrama de Clases del Análisis: CU Gestionar puerto

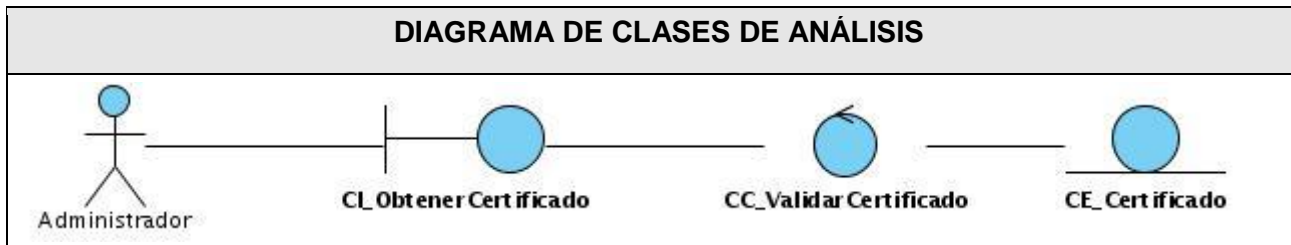


Figura 21: Diagrama de Clases del Análisis: CU Certificar servidor

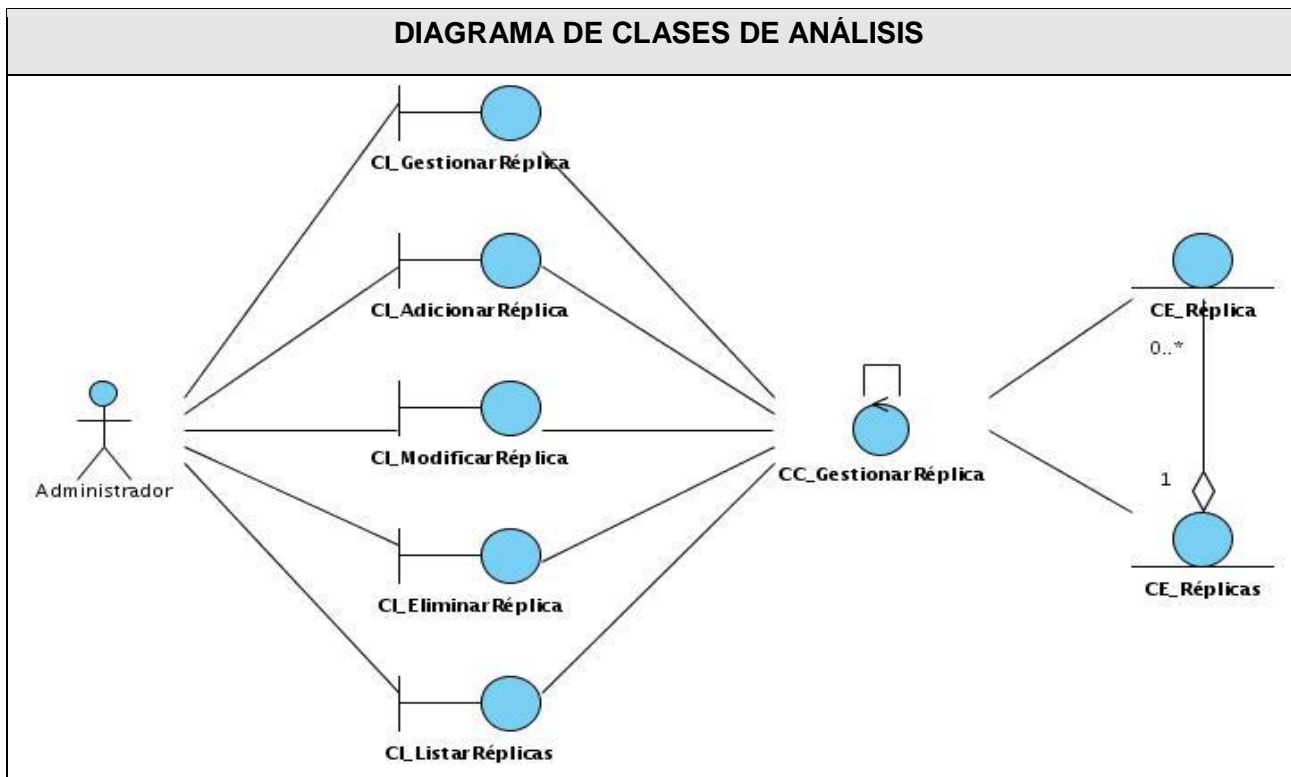


Figura 22: Diagrama de Clases del Análisis: CU Gestionar réplica

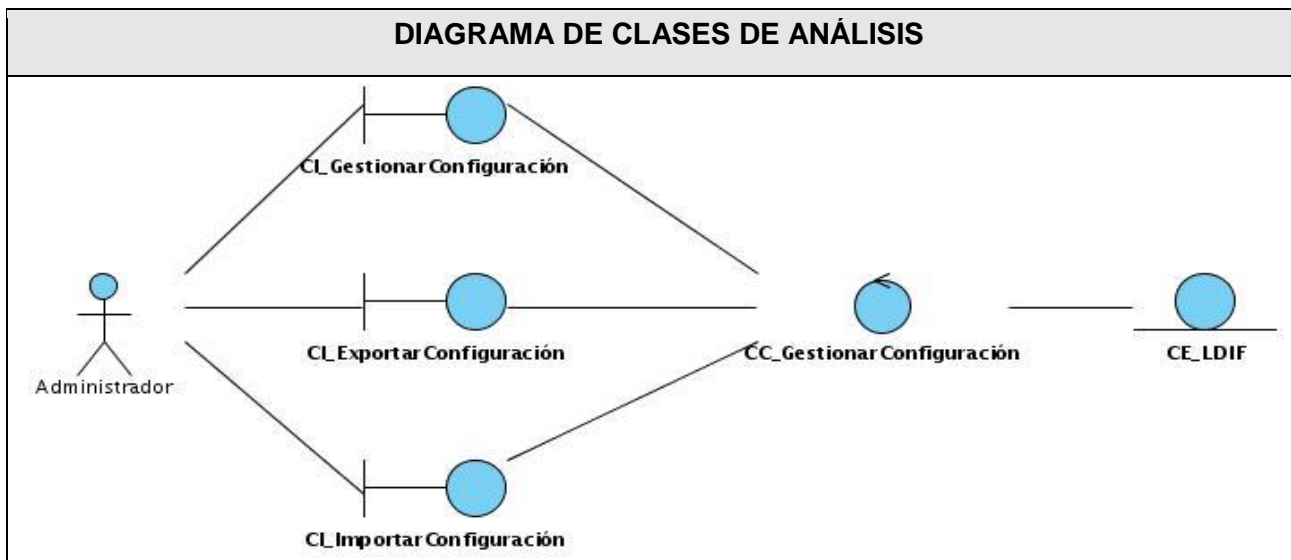


Figura 23: Diagrama de Clases del Análisis: CU Gestionar configuración



3.3 Modelo de Diseño

El diseño de un sistema está estrechamente ligado a la arquitectura escogida para su desarrollo. Una arquitectura con clara definición de capas que permitan la descentralización del trabajo, con componentes que cumplan funciones específicas asociados a cada una de ellas; es sin lugar a dudas una alternativa a valorar para elaborar un sistema informático. La arquitectura en capas permite ejercitar el trabajo en paralelo de los desarrolladores, mejora la detección y corrección de errores, además de ofrecer una mayor transparencia y entendimiento del sistema que se construye.

Los siguientes diagramas muestran la estructura general del sistema en términos de capas y componentes:

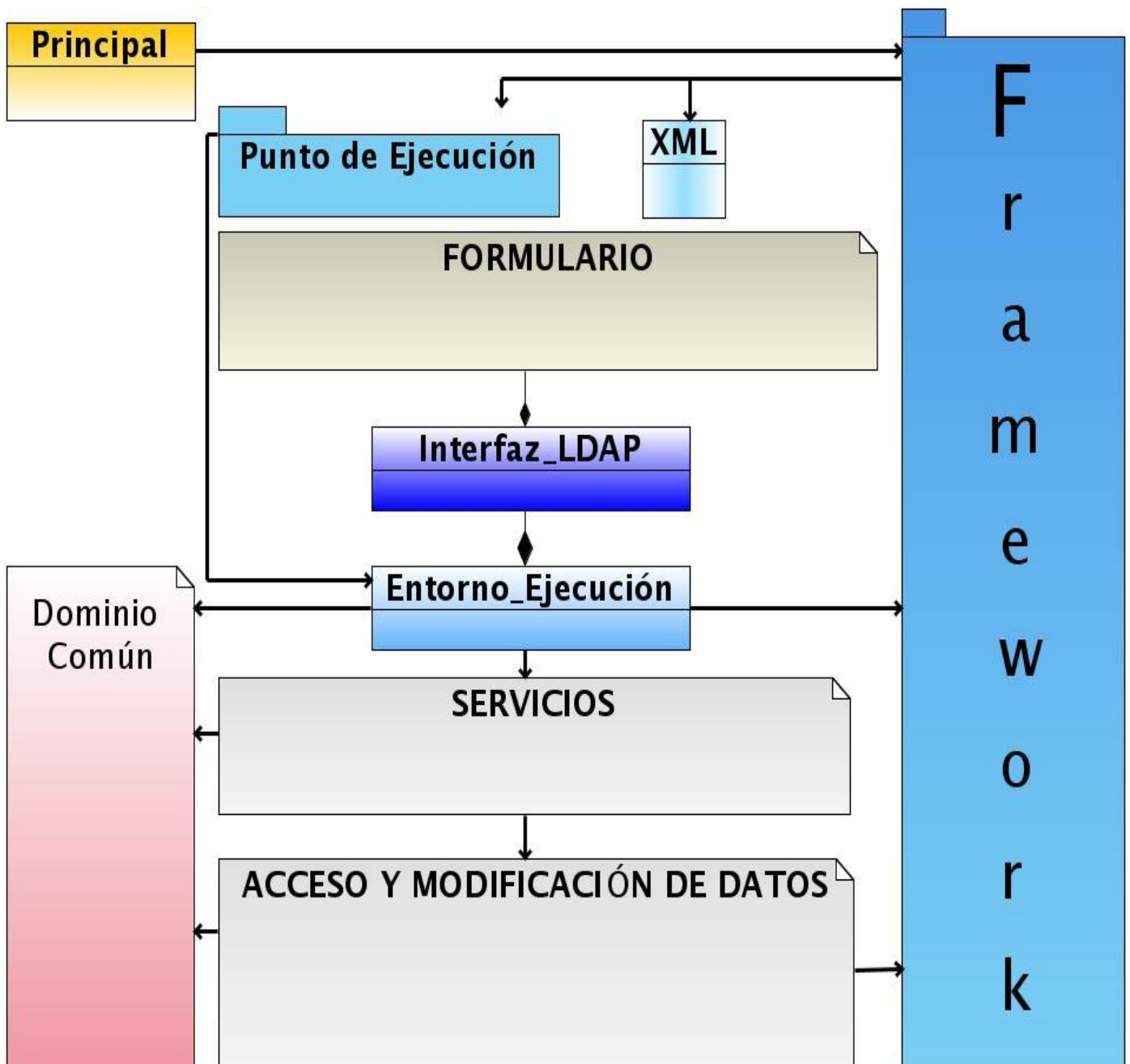


Figura 24: Estructura general del sistema



Capítulo 3: Análisis y Diseño del Sistema

Además de las capas de Interfaz, Servicios y Acceso y Modificación de Datos; existen algunos elementos importantes. En primer lugar se ha decidido construir un *framework* que se encargue de resolver problemas puntuales relacionados con la iniciación del sistema, la instalación desde fuentes, 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. Se ha generado una nueva estructura denominada *Entorno_Ejecución* cuya función principal es tratar los eventos generados a partir de la interfaz y conectarlos con la capa de servicios. La posibilidad que brinda python como lenguaje para modelar las interfaces de manera separada a su codificación, haciendo que solo la *Interfaz* contenga la definición de los componentes empleados ofrece aún mayor claridad a la solución de software. 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 empleadas y donde residirá su implementación particular. Otro aspecto importante es entender el funcionamiento del componente *Dominio Común*, el mismo contiene entidades o atributos que serán reutilizados por más de una clase en el sistema y es inicializado a través del *Entorno_Ejecución*.

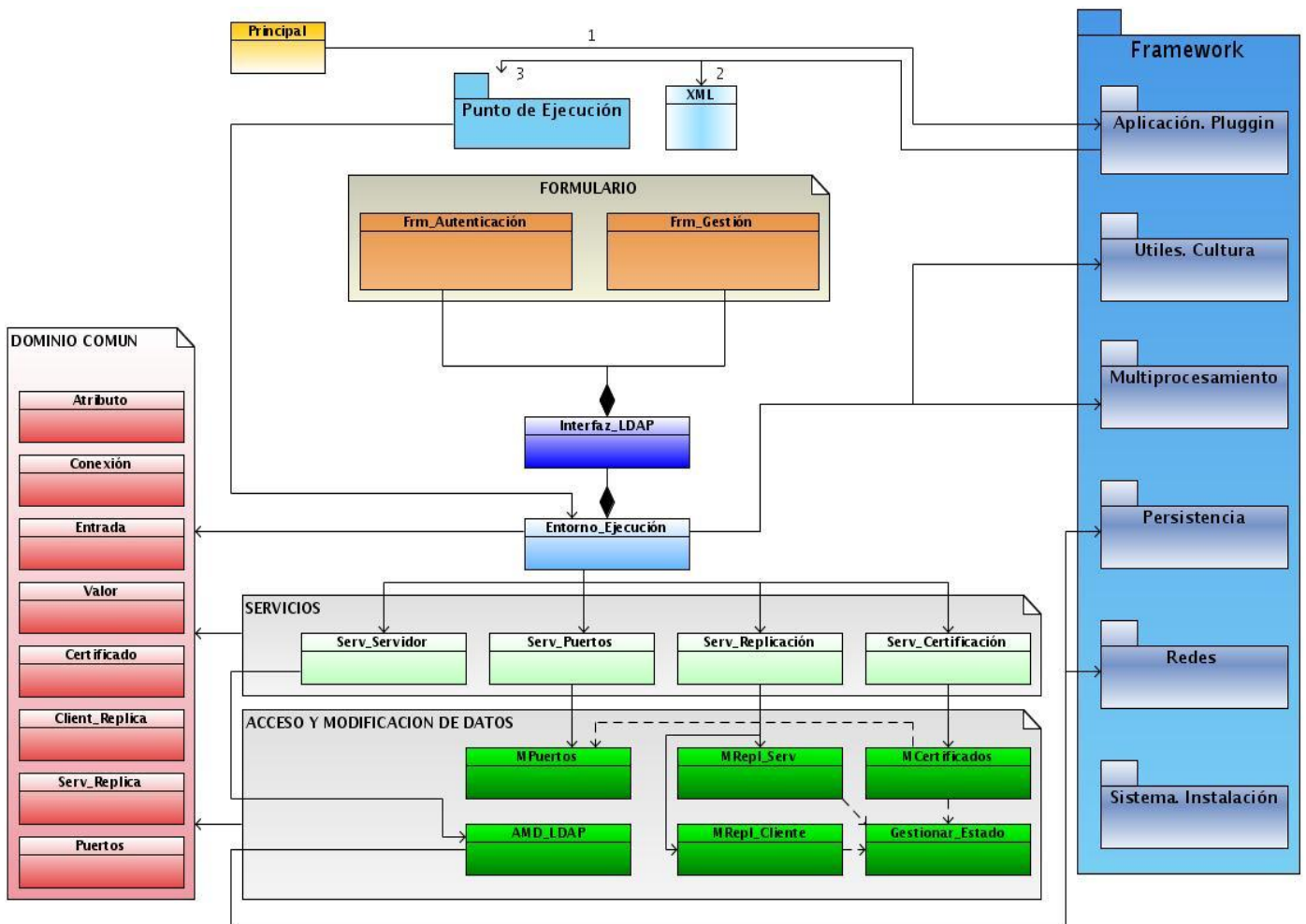


Figura 25: Estructura detallada del sistema

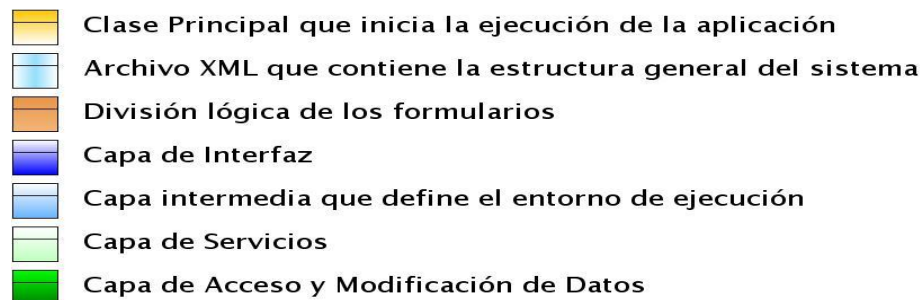


Figura 26: Definición de los términos fundamentales



3.3.1 Diagrama de Clases del Diseño

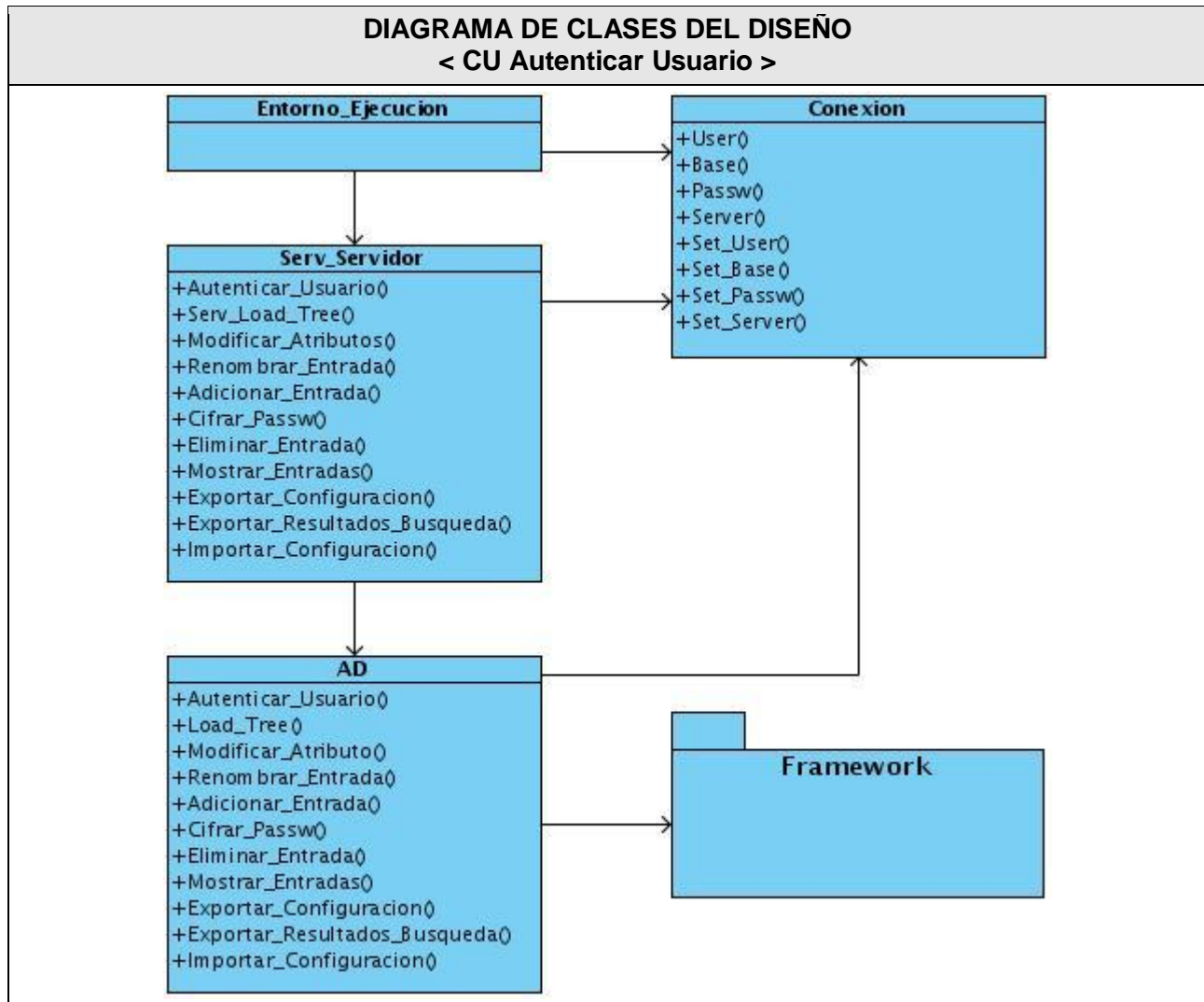


Figura 27: Diagrama de Clases del Diseño: CU Autenticar Usuario

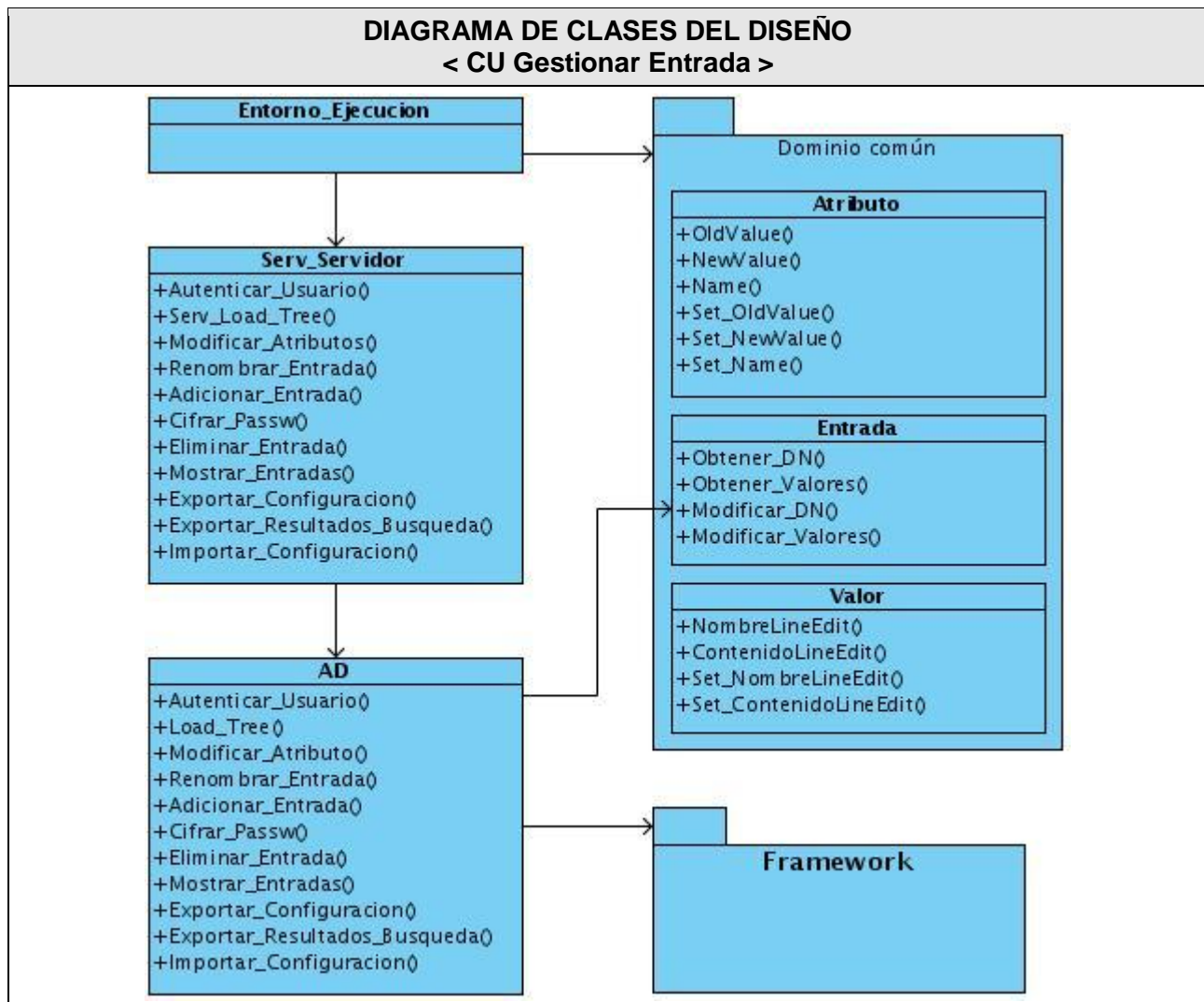


Figura 28: Diagrama de Clases del Diseño: CU Gestionar Entrada

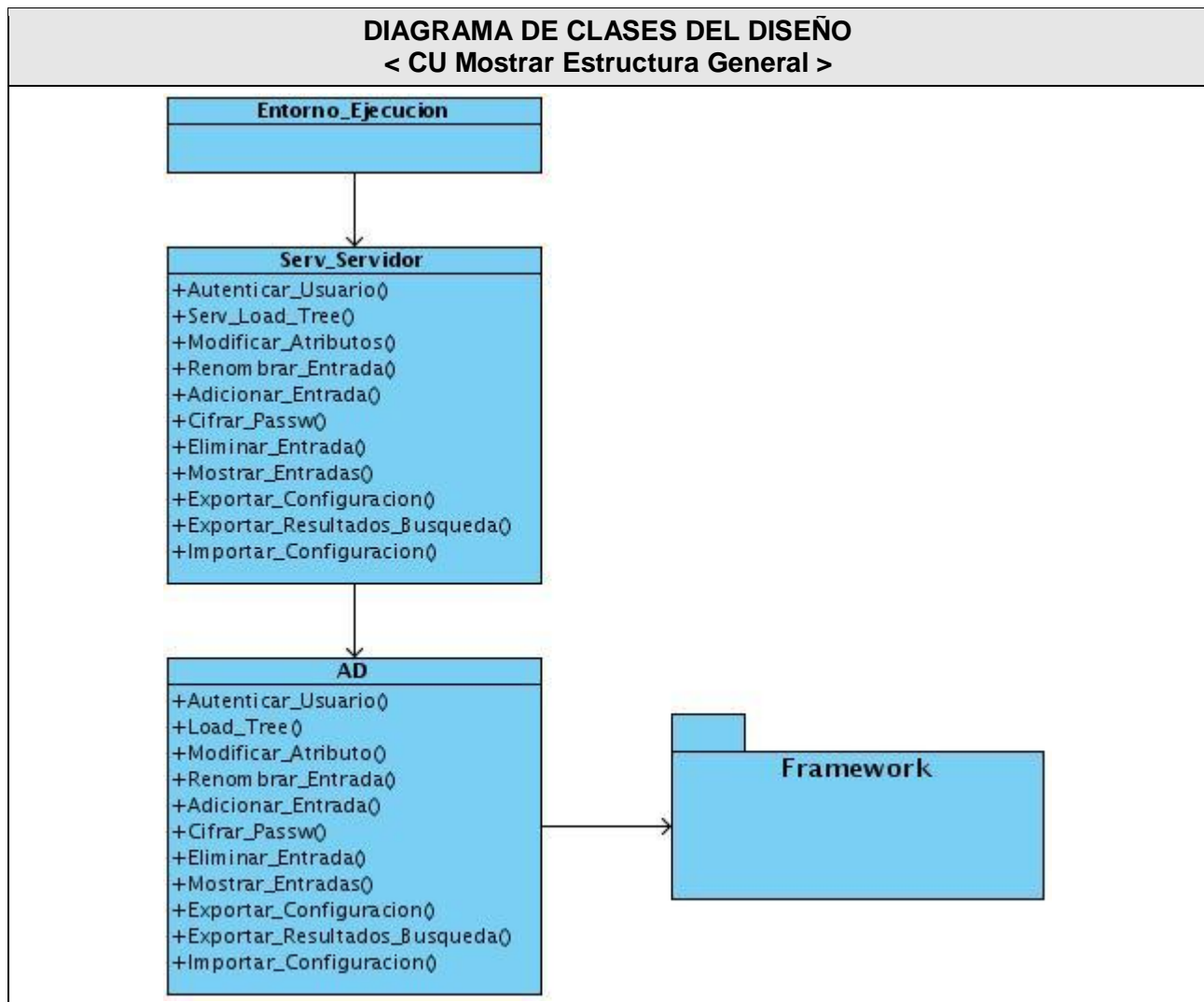


Figura 29: Diagrama de Clases del Diseño: CU Mostrar Estructura General

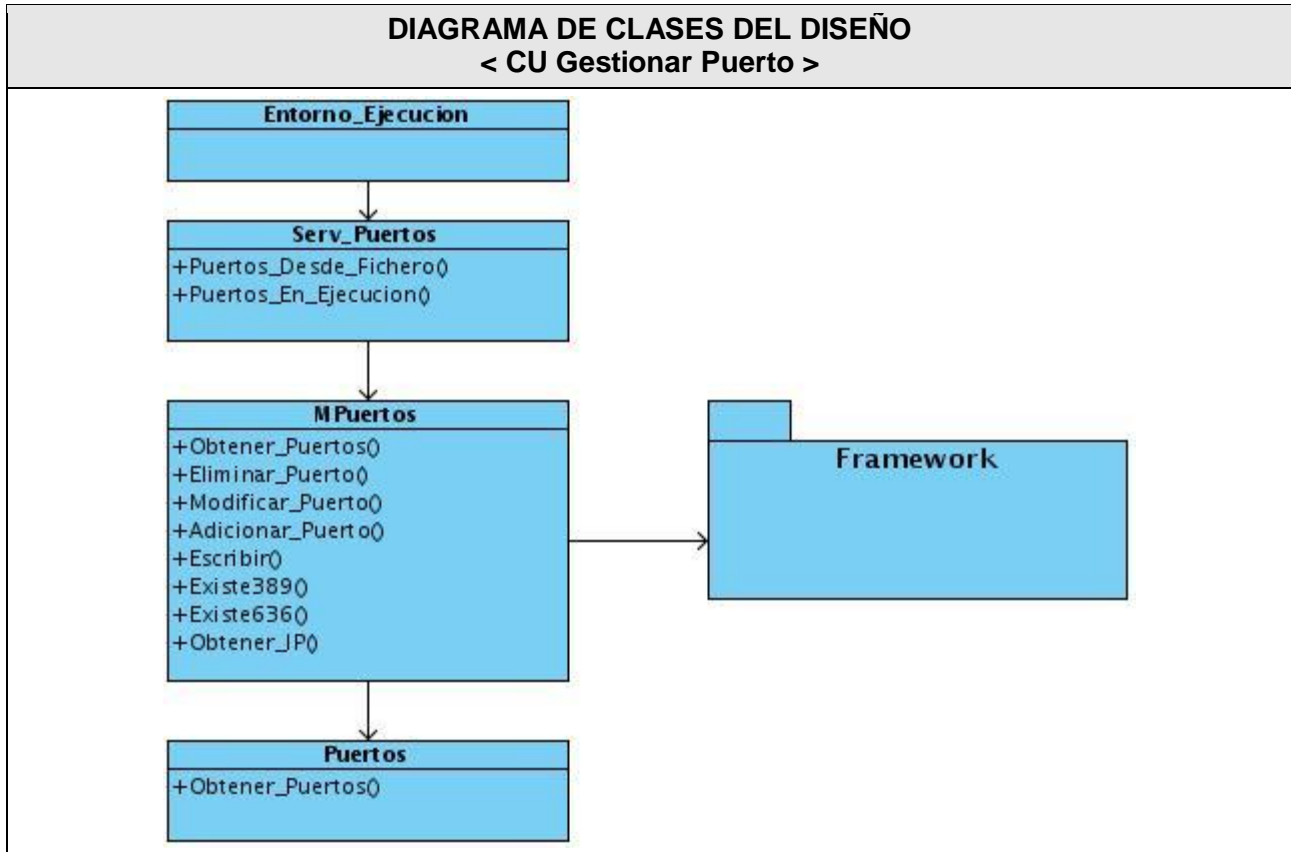


Figura 30: Diagrama de Clases del Diseño: CU Gestionar Puerto

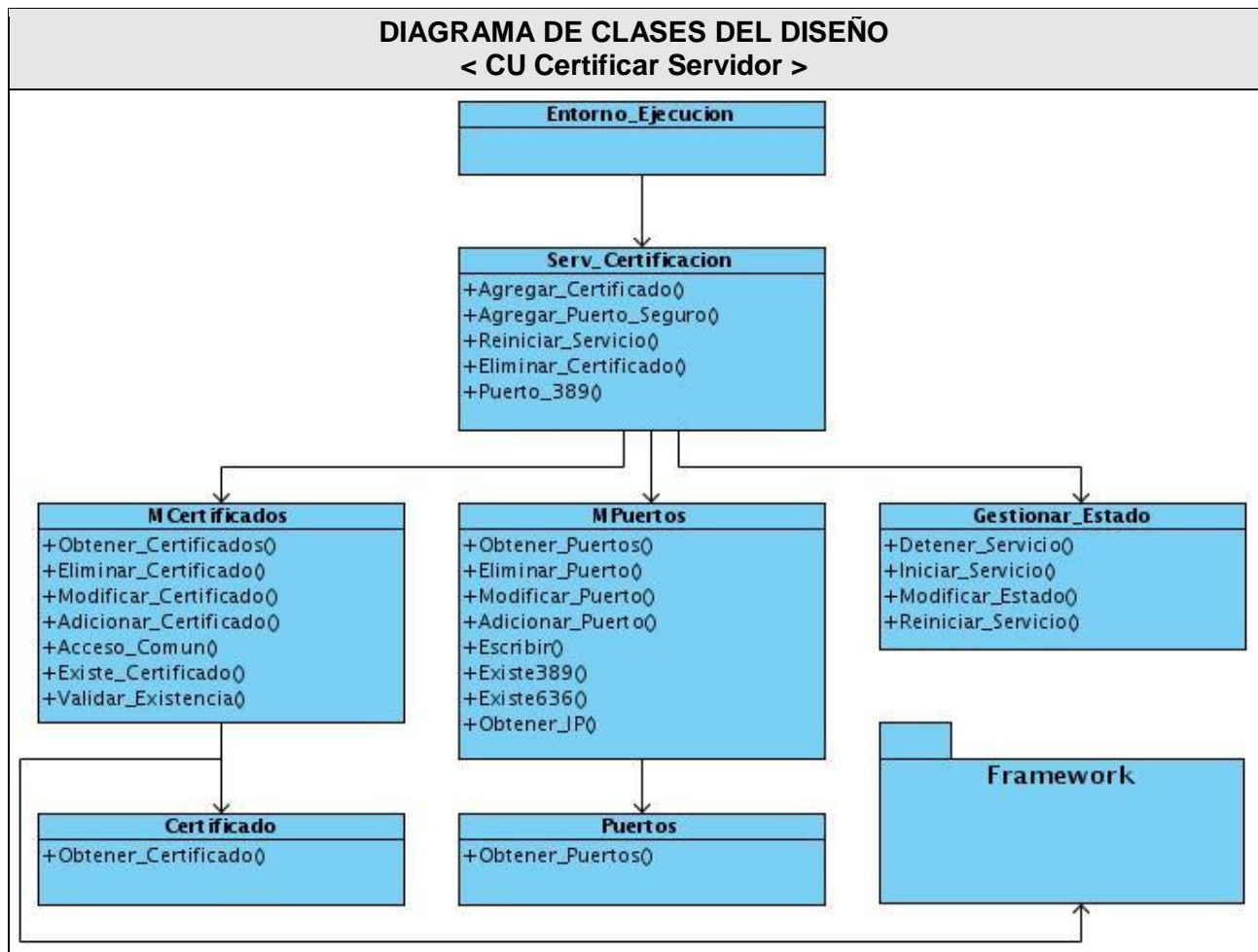


Figura 31: Diagrama de Clases del Diseño: CU Construir Certificado

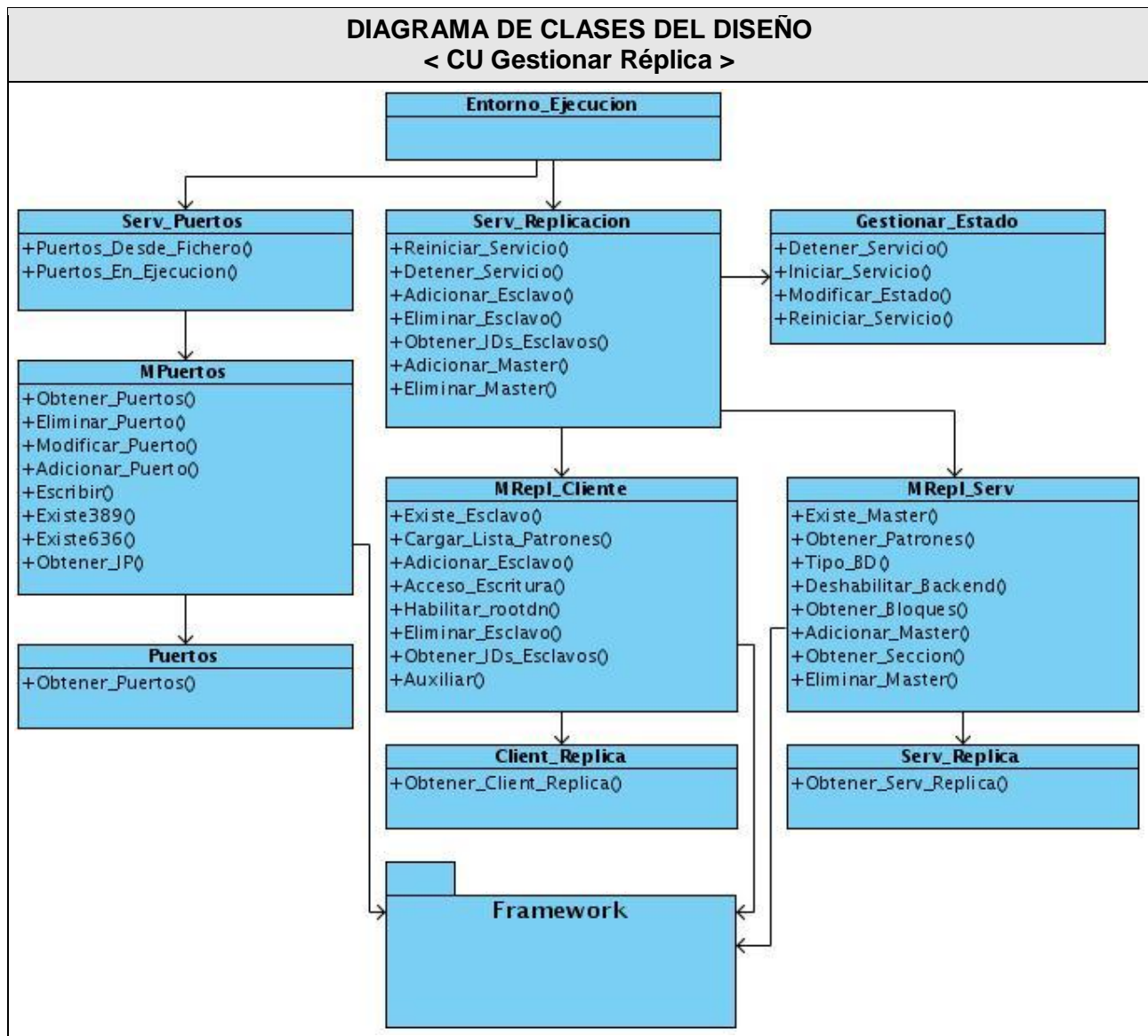


Figura 32: Diagrama de Clases del Diseño: CU Gestionar Réplica

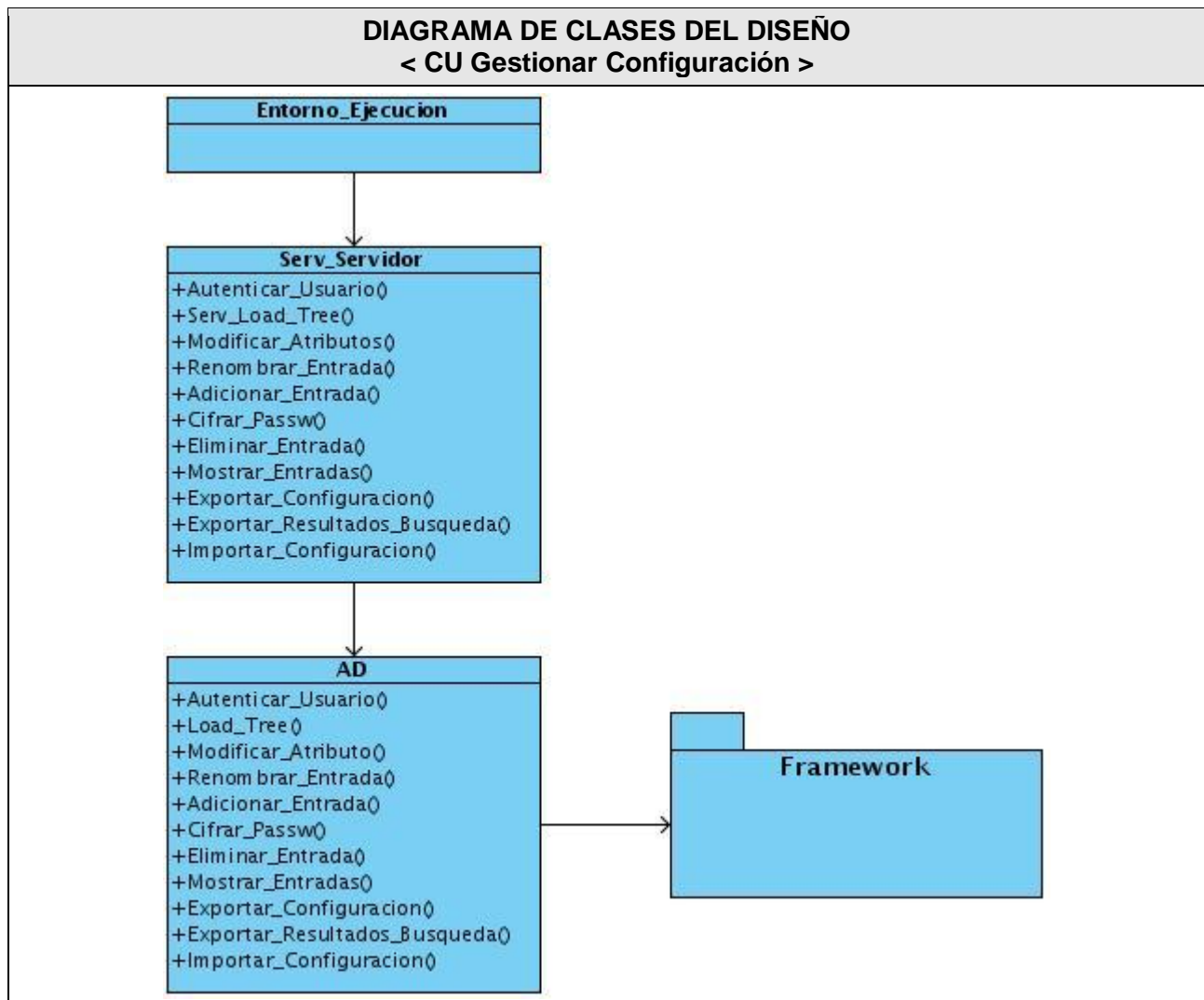


Figura 33: Diagrama de Clases del Diseño: CU Gestionar Configuración



3.4 Conclusiones

En el capítulo fueron presentados los artefactos fundamentales correspondientes al flujo de trabajo de *Análisis y Diseño*, entre ellos los modelos de análisis y diseño. La concepción del *Análisis* como parte fundamental del sistema ofrece una visión que sirve de base para la elaboración del resultado final de este flujo: el modelo de diseño, el cual constituye la entrada principal para la futura implementación del sistema.



CAPÍTULO 4 “Implementación”

4.1 Introducción

El presente capítulo tiene como objetivo desarrollar los artefactos correspondientes a la implementación del sistema, comenzando con el resultado más importante del capítulo anterior: el modelo de diseño. A partir de este artefacto se realizan los diagramas de componentes y despliegue que conforman lo que se conoce como: Modelo de Implementación; de esta forma se describen cómo los elementos del modelo del diseño se implementan en términos de componentes y se organizan de acuerdo a los nodos específicos en el modelo de despliegue.

4.2 Diagrama de despliegue

Los diagramas de despliegue describen la arquitectura física del sistema durante la ejecución, en términos de: procesadores, dispositivos y componentes de software. Básicamente este tipo de diagrama se utiliza para modelar el hardware utilizado en la implementación del sistema y las relaciones entre sus componentes.

En la Figura 34 se representan los aspectos más importantes:

- 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 34: Diagrama de Despliegue del sistema



4.3 Diagrama de componentes

Los diagramas de componentes muestran la separación de un sistema de software en componentes físicos y las dependencias entre ellos.

- El paquete “Interfaz” contiene un componente con el contenido gráfico de la aplicación y a su vez otro paquete con los módulos: *Principal.py*, *Iconos.py* e *Imágenes_rc.py*.
- El paquete “Entorno de Ejecución” contiene un único componente denominado *Entorno_Ejecucion.py* encargado de recibir los eventos externos al sistema y asociarle un tratamiento adecuado mediante otros componentes de software.
- El paquete “Servicios” contiene los componentes: *Serv_Servidor.py*, *Serv_Certificacion.py*, *Serv_Puertos.py* y *Serv_Replicacion.py*; los mismos funcionan como componentes del negocio de la aplicación.
- El paquete “Acceso y modificación de datos” contiene el componente *AD.py* encargado de acceder y modificar los datos del LDAP. Presenta además otros dos paquetes con componentes para gestionar el estado del servicio y manipular ficheros con la información de certificados, puertos y réplicas del servidor.
- El paquete “Dominio común” contiene un grupo de componente que son reutilizados por la mayoría de las capas del sistema.

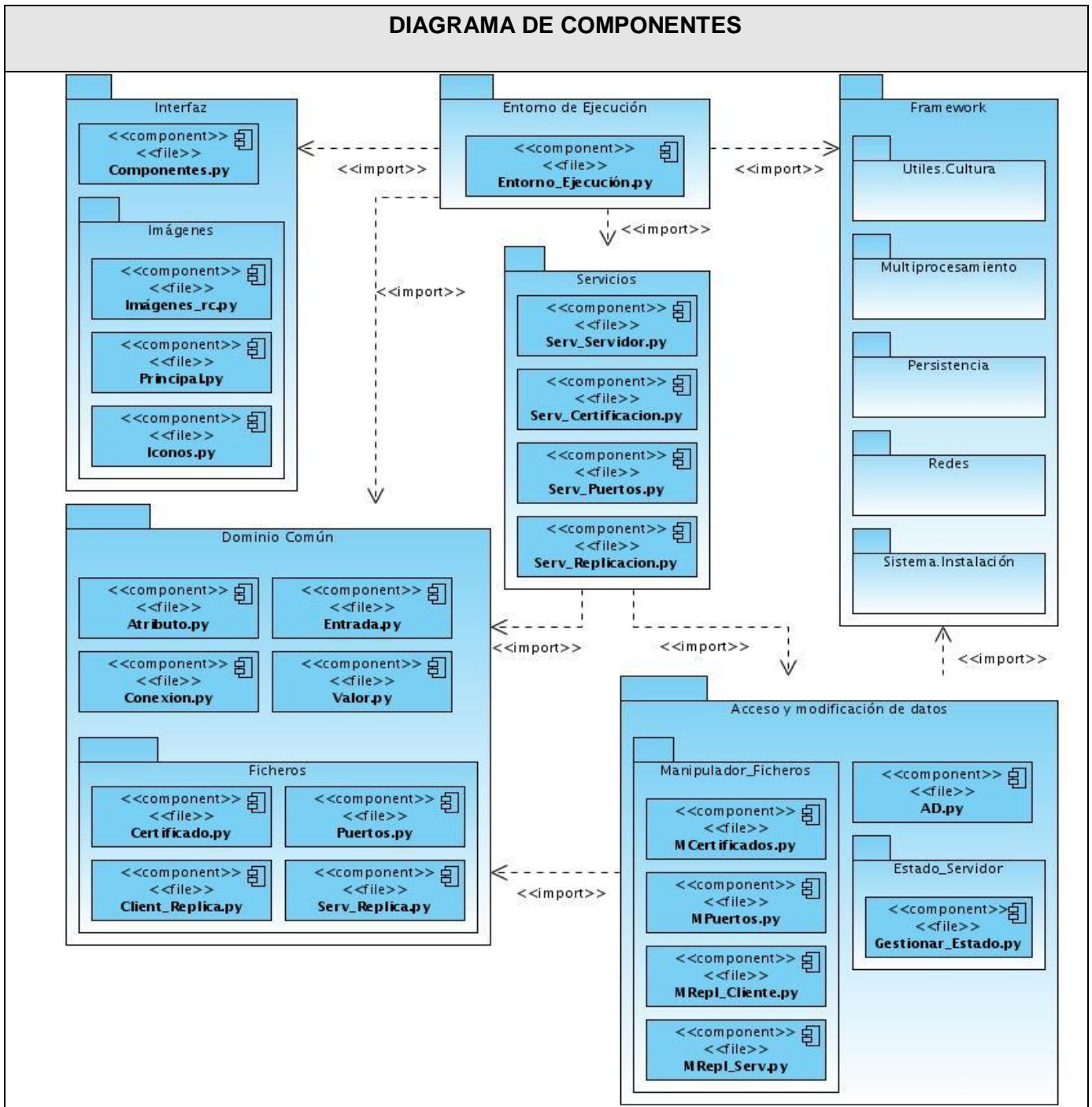


Figura 35: Diagrama General de Componentes



A continuación se muestran los diagramas de componentes correspondientes a cada uno de los casos de uso del Módulo de Directorio.

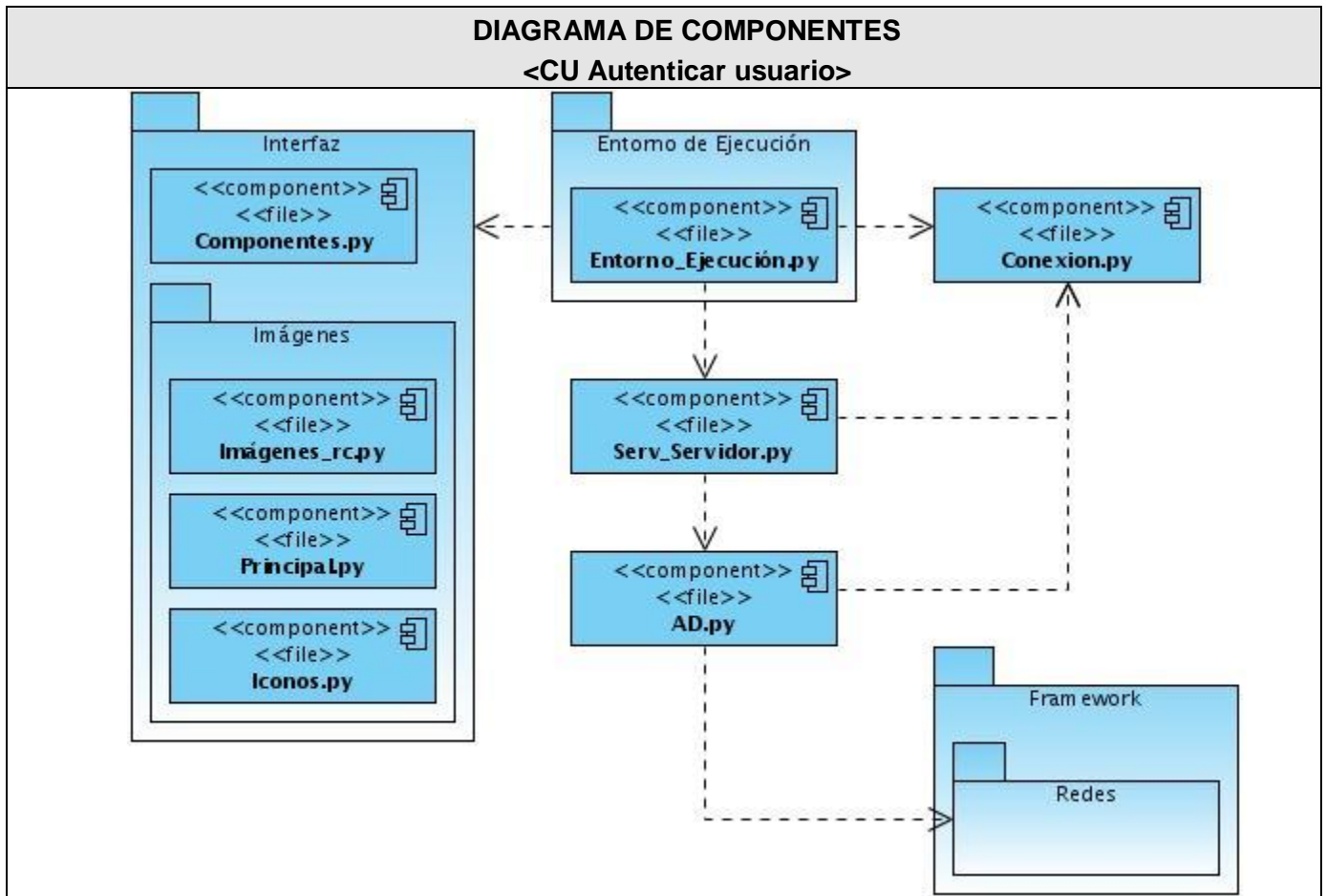


Figura 36: Diagrama de Componentes: CU Autenticar usuario

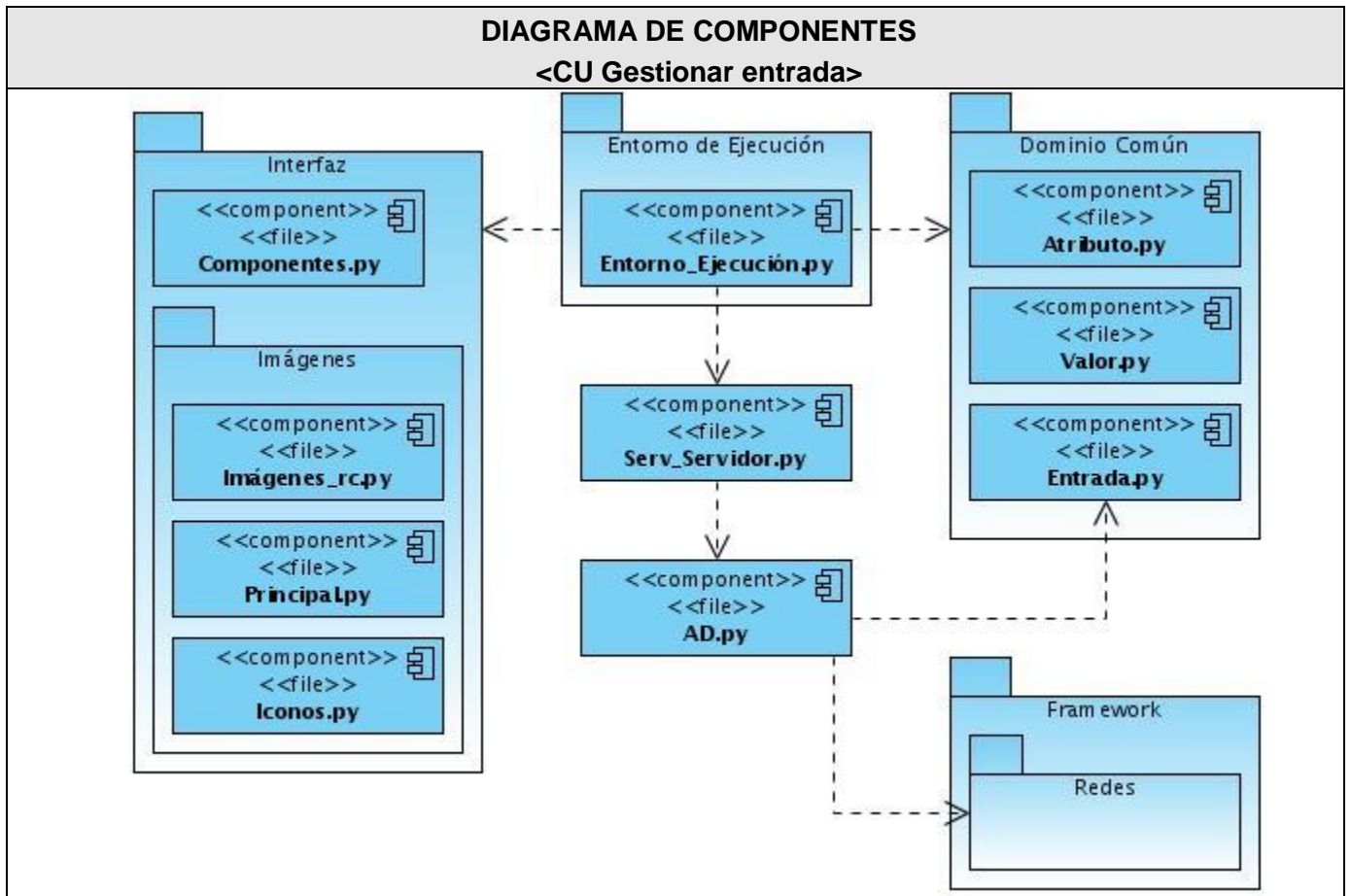


Figura 37: Diagrama de Componentes: CU Gestionar entrada

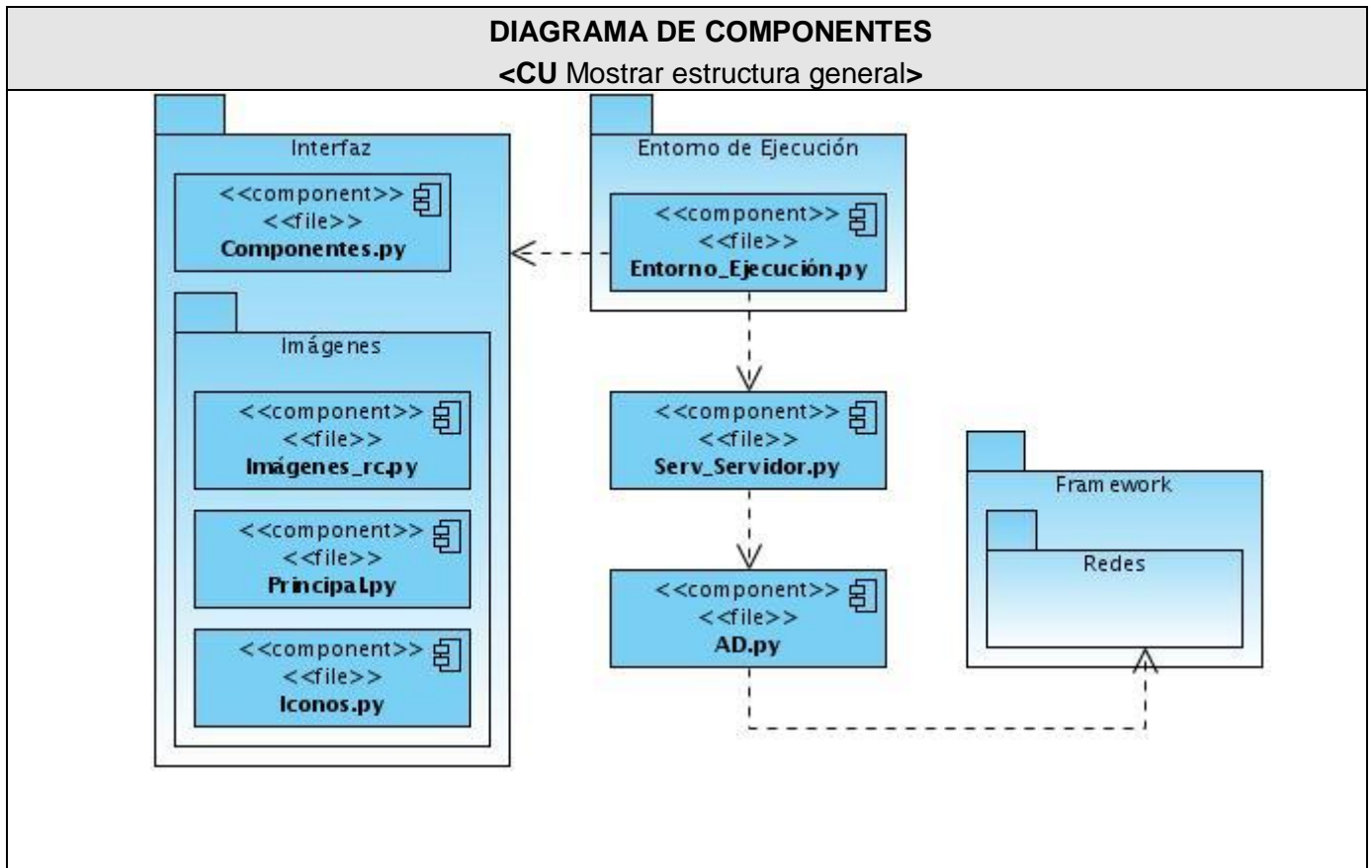


Figura 38: Diagrama de Componentes: CU Mostrar estructura general

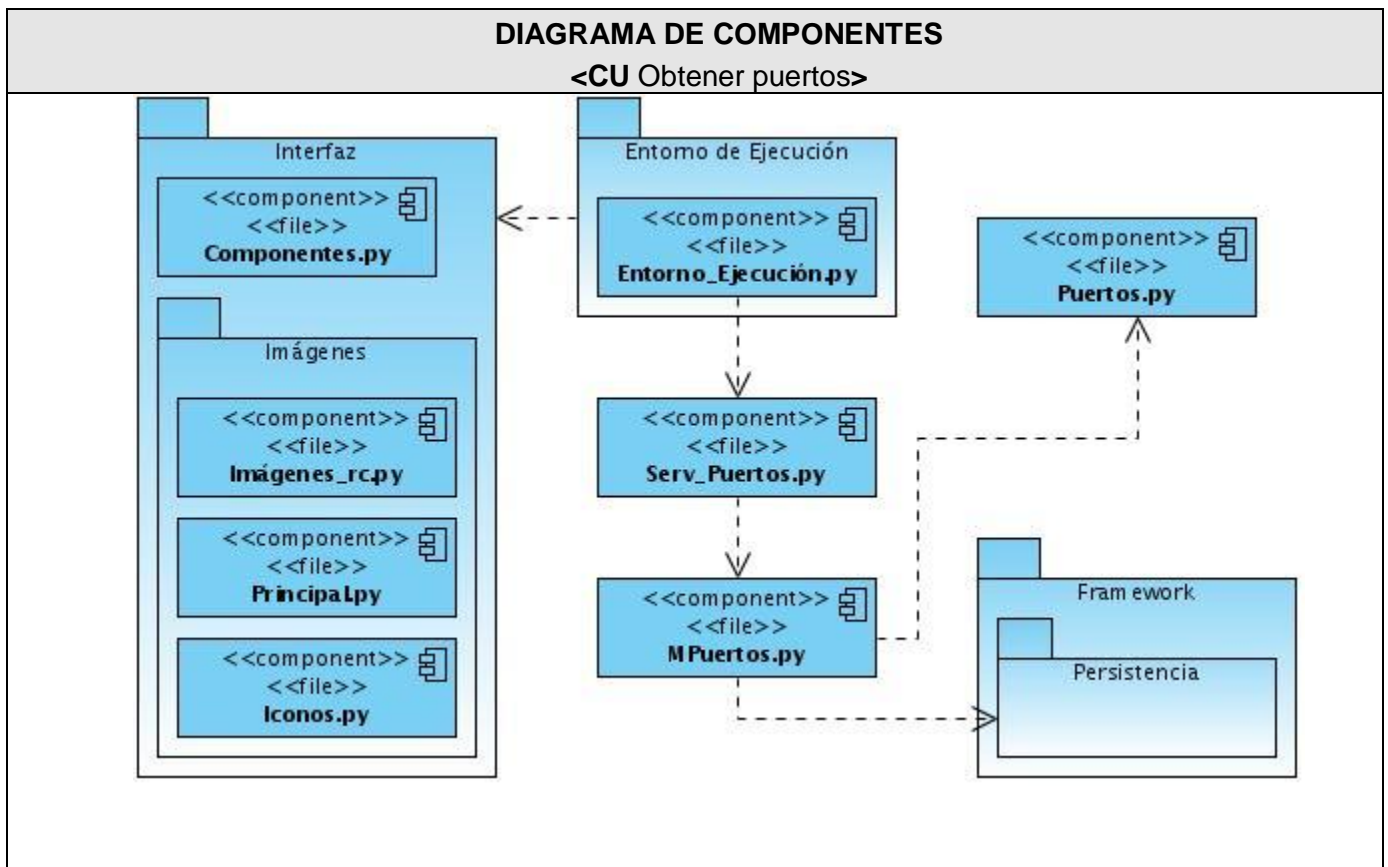


Figura 39: Diagrama de Componentes: CU Obtener puertos

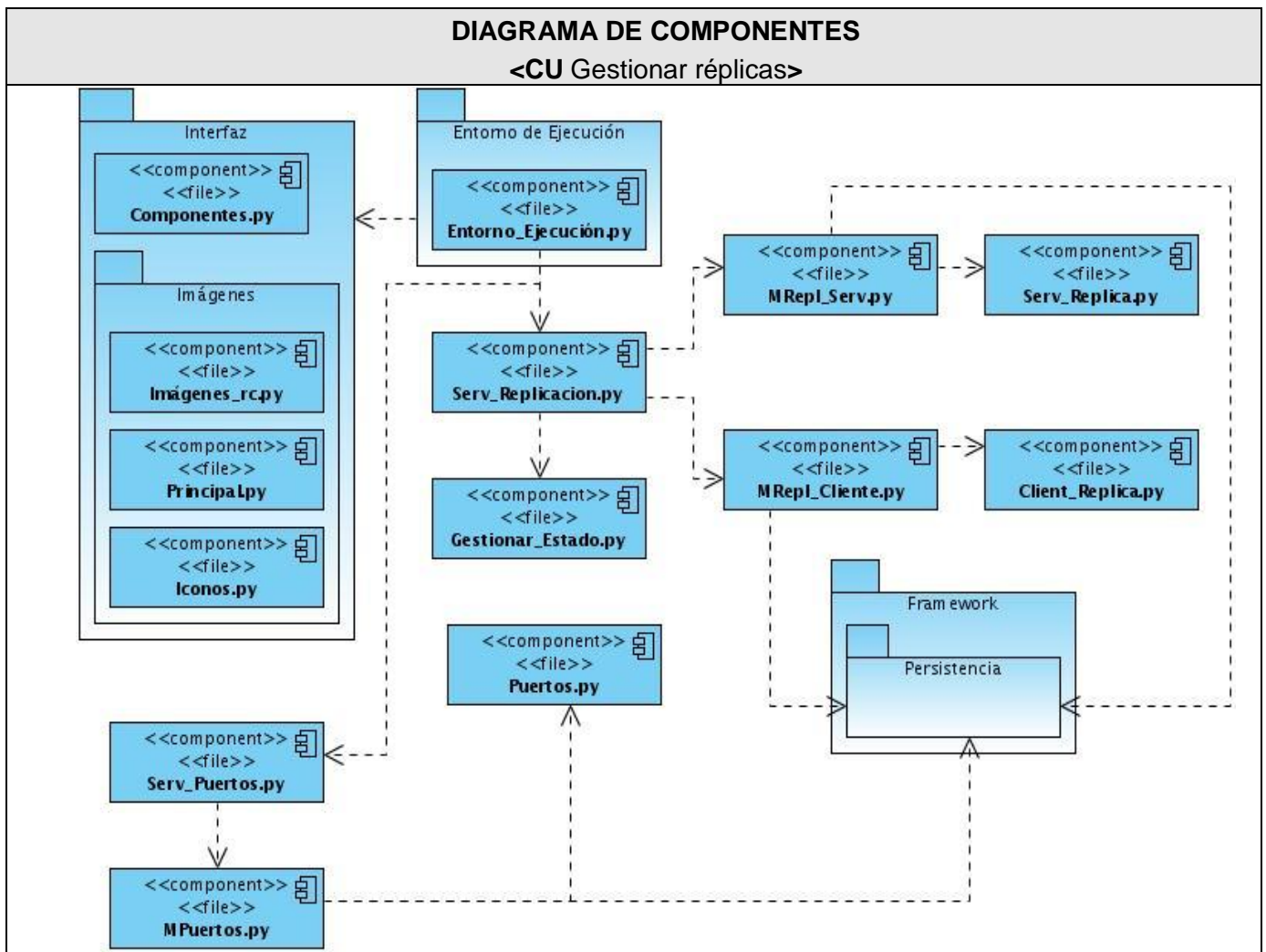


Figura 40: Diagrama de Componentes: CU Gestionar réplicas

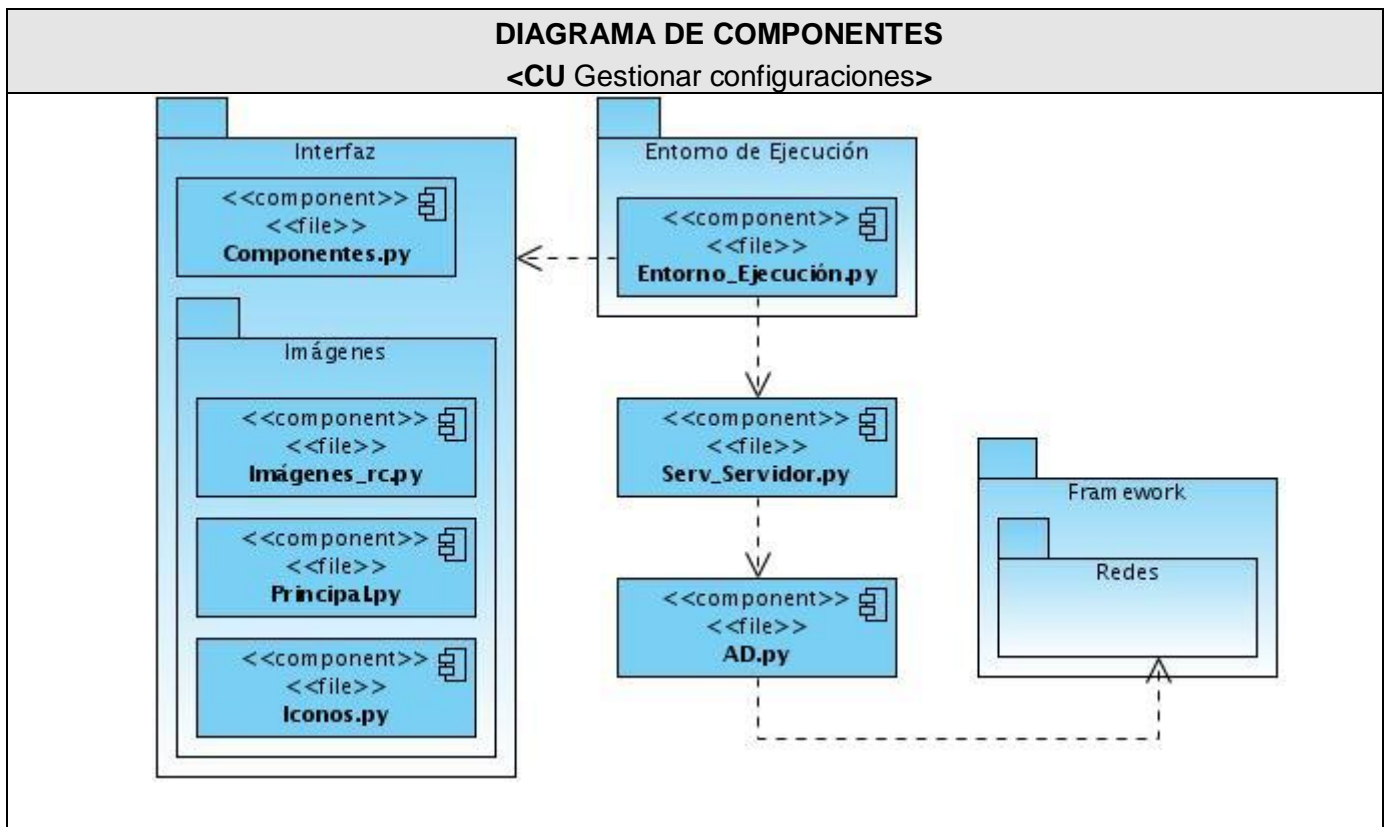


Figura 41: Diagrama de Componentes: CU Gestionar configuraciones

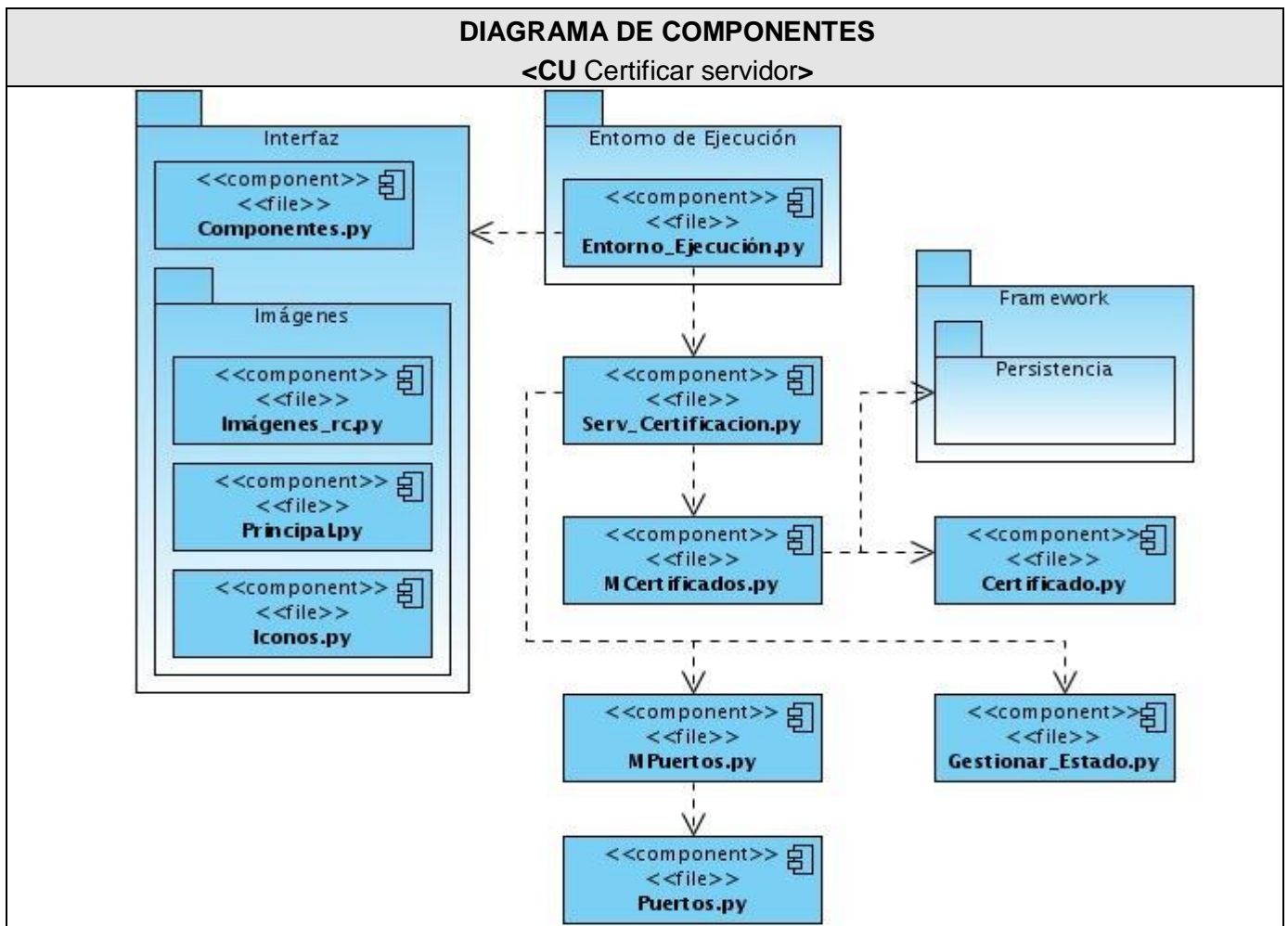


Figura 42: Diagrama de Componentes: CU Certificar servidor

4.4 Modelo de Prueba

La intención principal del Flujo de Trabajo de Pruebas es ejecutar un programa para descubrir sus errores. Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces; por lo cual una prueba tiene éxito si descubre un error no detectado.

A continuación son descritos los casos de prueba de integración asociados a cada caso de uso.



Caso de Uso: Autenticar usuario

Entrada	El caso de prueba se ejecuta dada la necesidad de validar el correcto funcionamiento de la aplicación de acuerdo a un conjunto de acciones posibles a realizar por el usuario final del sistema, en este caso: el administrador. La acción tratada está referida a la posibilidad de autenticarse en el sistema. El proceso requiere de cuatro parámetros esenciales: el usuario, la contraseña, el dominio y la dirección IP donde reside el servidor. Cada información solicitada presenta un formato específico, el cual es mostrado cuando el usuario se coloca encima de la región editable.
Resultados	Se muestra un mensaje de error con el texto: "Por favor verifique el formato de los datos" en caso de que: <ul style="list-style-type: none">• No se hayan colocado todos los datos solicitados.• No se hayan colocado los datos con el formato correcto.• El servidor LDAP descrito no esté disponible. Se muestra la interfaz que contiene la estructura jerárquica del árbol LDAP.
Condiciones	



Tabla 22: Descripción del caso de prueba Autenticar usuario

Caso de Uso: Gestionar entradas

Entrada	<p>El caso de prueba se ejecuta dada la necesidad de validar el correcto funcionamiento de la aplicación de acuerdo a un conjunto de acciones posibles a realizar por el usuario final del sistema, en este caso: el administrador. La acción tratada está referida a la posibilidad de gestionar las entradas del sistema. Las entradas del sistema están referidas a entidades lógicas que presentan información asociada, como cuentas de usuario, computadoras, grupos, unidades organizativas, entre otras. Cada entrada puede ser añadida, eliminada, modificada o leída por el administrador del sistema; aunque solo los procesos de adición y modificación requieren de una entrada de datos. Para adicionar una entrada primeramente se selecciona su tipo y posteriormente se muestran todos los datos requeridos para su construcción, los cuales varían de acuerdo al tipo. Para la construcción de una nueva entrada, generalmente son solicitados datos opcionales y obligatorios; la región editable donde se debe colocar la información imprescindible presenta un color de fondo más intenso. Para actualizar la información de alguna entrada, el usuario debe</p>
---------	--



	<p>dar clic sobre la misma y obtendrá en el panel derecho de la aplicación todos los datos actuales, muchos de los cuales podrán ser modificados y salvados mediante el botón <i>Salvar</i> al final del panel.</p>
Resultados	<p><u>Adicionar entrada:</u></p> <p>Se muestra un mensaje de error con el texto: “Datos requeridos” en caso de que:</p> <ul style="list-style-type: none">• Se trate de adicionar una entrada sin la totalidad de sus datos requeridos. <p>Se muestra un mensaje de error con el texto: “Identificador existente” en caso de que:</p> <ul style="list-style-type: none">• Se trate de adicionar una cuenta de usuario o computadora con un identificador existente. <p>Se muestra un mensaje de error con el texto: “El objeto que trata de adicionar ya existe” en caso de que:</p> <ul style="list-style-type: none">• Se trate de adicionar una entrada con el mismo identificador de otra en la misma ubicación del árbol jerárquico. <p><u>Modificar entrada:</u></p> <p>La entrada asociada es modificada</p>



	<p>parcialmente.</p> <p><u>Eliminar entrada:</u></p> <p>Se muestra un mensaje de alerta con el texto: “Realmente desea continuar” en caso de que se solicite eliminar la entrada correspondiente. En caso de aceptar, el proceso de eliminación se hará de manera recursiva con todas las entradas interiores en caso de existir.</p> <p><u>Leer entrada:</u></p> <p>Cuando el usuario da clic sobre una entrada, son mostrados sus datos en el panel derecho de la aplicación.</p>
Condiciones	El usuario debe estar autenticado en el sistema como administrador.

Tabla 23: Descripción del caso de prueba Gestionar entradas

Caso de Uso: Mostrar estructura general

Entrada	El caso de prueba se ejecuta dada la necesidad de validar el correcto funcionamiento de la aplicación de acuerdo a un conjunto de acciones posibles a realizar por el usuario final del sistema, en este caso: el administrador. La acción tratada está referida a la posibilidad de mostrar la estructura general del árbol de directorios. El proceso requiere como entrada y
---------	---



	requisito principal que el usuario que intenta gestionar el servidor se haya autenticado correctamente como administrador del mismo.
Resultados	Se muestran inicialmente dos paneles: izquierdo y derecho. En el panel de la izquierda se muestra la estructura en árbol del contenido perteneciente al servidor; en el panel derecho se muestra una imagen representativa del sistema de SW propuesto. Dicha imagen es sustituida por los datos de alguna entrada o las opciones del menú en caso de acceder a ellas.
Condiciones	El usuario debe estar autenticado en el sistema como administrador.

Tabla 24: Descripción del caso de prueba Mostrar estructura general

Caso de Uso: Gestionar configuración

Entrada	El caso de prueba se ejecuta dada la necesidad de validar el correcto funcionamiento de la aplicación de acuerdo a un conjunto de acciones posibles a realizar por el usuario final del sistema, en este caso: el administrador. La acción tratada está referida a la posibilidad de gestionar la configuración parcial o total del árbol de directorios. La configuración es manejada a través de archivos con formato LDIF (LDAP Data Interchange Format); dicho formato es el establecido para el intercambio de datos entre servidores LDAP. Las operaciones
---------	--



	<p>posibles son importar y exportar las configuraciones. Para importar las configuraciones se requiere de un archivo LDIF adecuado. El proceso para exportar puede ejecutarse a partir de dos escenarios distintos; primeramente como resultado de un proceso de búsqueda y en segundo lugar exportando directamente la sección del árbol deseada. Este último modo permite definir el ámbito de exportación como: subárbol completo, un nivel bajo la base o la base solamente.</p>
Resultados	<p><u>Exportar configuración</u></p> <p>La configuración es exportada con formato LDIF hacia un fichero en la PC.</p> <p><u>Importar configuración</u></p> <p>La configuración es importada desde un fichero de texto con formato LDIF, si el fichero presenta errores en su contenido, se muestra el mensaje “El fichero presenta un formato incorrecto”</p>
Condiciones	<p>El usuario debe estar autenticado en el sistema como administrador.</p>

Tabla 25: Descripción del caso de prueba Gestionar configuración

Caso de Uso: Certificar servidor

Entrada	El caso de prueba se ejecuta dada la necesidad
---------	--



	<p>de validar el correcto funcionamiento de la aplicación de acuerdo a un conjunto de acciones posibles a realizar por el usuario final del sistema, en este caso: el administrador. La acción tratada está referida a la posibilidad de certificar el servidor LDAP. Certificar el servidor consiste en asociarle un certificado que le permita a los clientes establecer conexiones cifradas por el puerto 636. Debido a que la certificación presupone una modificación a los ficheros de configuración del servidor se requiere de una nueva autenticación en la PC donde se ejecuta el servidor LDAP. Se requiere de un usuario con permisos de escritura en los archivos de configuración del sistema; generalmente <i>root</i>. Una vez autenticado se muestra una ventana donde debe seleccionarse la ubicación del certificado y su llave.</p>
Resultados	<p>Se muestra el mensaje de error “Información insuficiente” en caso de que:</p> <ul style="list-style-type: none">• Se trate de autenticar en la PC donde reside el servidor y no sean colocados todos los datos necesarios. <p>Se muestra el mensaje de error: “La autenticación ha fallado” en caso de que:</p> <ul style="list-style-type: none">• Se trate de autenticar en la PC donde



	<p>reside el servidor y los datos solicitados no sean correctos.</p> <p>Se muestra el mensaje de error: “Debe colocar el certificado y su llave” en caso de que:</p> <ul style="list-style-type: none">• No se hayan seleccionado la ruta del certificado y la llave. <p>Se muestra el mensaje de error: “Imposible adicionar el certificado en caso de que:</p> <ul style="list-style-type: none">• Se produzca un error de alguna naturaleza al tratar de adicionar el certificado. <p>Se muestra el mensaje: “El certificado ha sido añadido satisfactoriamente” en caso de que:</p> <ul style="list-style-type: none">• No se hayan presentado problemas en el proceso de adición.
Condiciones	El usuario debe estar autenticado en el sistema como administrador y en el ordenador donde reside el servidor como un usuario con permisos administrativos, generalmente <i>root</i> .

Tabla 26: Descripción del caso de prueba Certificar servidor

Caso de Uso: Obtener puertos

Entrada	El caso de prueba se ejecuta dada la necesidad
---------	--



	de validar el correcto funcionamiento de la aplicación de acuerdo a un conjunto de acciones posibles a realizar por el usuario final del sistema, en este caso: el administrador. La acción tratada está referida a la posibilidad de visualizar los puertos del servidor LDAP. Se requiere que el administrador coloque las credenciales válidas para acceder al ordenador donde reside el servidor.
Resultados	Tras la ejecución del caso de prueba son mostrados los puertos por los cuales se encuentra brindando su servicio el servidor.
Condiciones	El usuario debe estar autenticado en el sistema como administrador y en el ordenador donde reside el servidor como un usuario con permisos administrativos, generalmente <i>root</i> .

Tabla 27: Descripción del caso de prueba Gestionar puertos

Caso de Uso: Gestionar réplicas

Entrada	El caso de prueba se ejecuta dada la necesidad de validar el correcto funcionamiento de la aplicación de acuerdo a un conjunto de acciones posibles a realizar por el usuario final del sistema, en este caso: el administrador. La acción tratada está referida a la posibilidad de gestionar las réplicas del servidor LDAP. Las réplicas pueden ser adicionadas, eliminadas o modificadas.
---------	---



Resultados	Tras la ejecución del caso de prueba el servidor ha añadido, eliminado o actualizado una o varias réplicas.
Condiciones	El usuario debe estar autenticado en el sistema como administrador y en el ordenador donde reside el servidor como un usuario con permisos administrativos, generalmente <i>root</i> .

Tabla 28: Descripción del caso de prueba Gestionar réplicas

4.5 Conclusiones

En este capítulo se desarrollaron los elementos correspondientes al modelo de implementación: el diagrama de despliegue y los de componentes por cada caso de uso. Completando así los artefactos fundamentales que constituyen la base del módulo de Directorio, dando cumplimiento de esta forma, a los requerimientos planteados inicialmente.



CAPÍTULO 5 “Estudio de Factibilidad”

5.1 Introducción

Los estudios de factibilidad son investigaciones altamente enfocadas en un proyecto, diseñados para producir información crítica sobre la viabilidad de implementar el sistema de software, y con base en ello decidir si se va a llevar a cabo o no. En este capítulo se determina la estimación y cálculo de tamaño del software, así como el esfuerzo, los costos y beneficios que reportará el sistema realizado para gestionar el trabajo del servidor de directorio OpenLDAP.

5.2 Planificación

La planificación es una actividad de gran importancia en el desarrollo de software, al establecerse los objetivos y metas del sistema por desarrollar, a la vez que ayuda a valorar sus costos.

Para que la planificación se logre efectuar de una forma eficiente, resulta fundamental evaluar el sistema de software por desarrollar, con el fin de estimar su nivel de dificultad, buscando obtener un aproximado del tiempo que será requerido en el desarrollo del mismo.

Además, permite determinar el esfuerzo o bien la cantidad de personas necesarias, para lograr finalizar dicho proyecto en un tiempo establecido.

5.2.1 Puntos de Casos de Uso

Constituye un método de estimación del esfuerzo de un proyecto de desarrollo de software a partir de los casos de uso, introducido en 1993. Utiliza los actores y casos de uso identificados para calcular el esfuerzo que constará desarrollarlos. A los casos de uso se les asigna una complejidad basada en transacciones, que son pares de pasos: acción del usuario y respuesta del sistema por cada uno de los escenarios de los casos de uso. Por otro lado, a los actores se les asigna una complejidad basada en su tipo. También se utilizan factores de entorno y de complejidad técnica para refinar aún más el resultado.



Una vez realizado los procesos anteriores se calculan los puntos de caso de uso no ajustados o UUCP, el TCF (factor de complejidad técnica) y el EF (factor del entorno). Con ellos, se calculan los puntos de caso de uso o UCP, que finalmente se traducen a esfuerzo en horas-hombre mediante la aplicación de un sencillo cálculo.

5.2.1.1 Cálculo de Puntos de Casos de Uso sin ajustar

El cálculo de los Puntos de Casos de Uso sin ajustar constituye el primer paso para realizar la estimación del esfuerzo, calculándose a partir de la ecuación:

UUCP = UAW + UUCW	Donde, UUCP: Puntos de Casos de Uso sin ajustar. UAW: Factor de Peso de los Actores sin ajustar. UUCW: Factor de Peso de los Casos de Uso sin ajustar.
--------------------------	---

Tabla 29: Puntos de Casos de Uso sin ajustar

El Factor de Peso de los Actores sin ajustar se refiere a la evaluación de la complejidad de los actores con los que tendrá que interactuar el sistema. Determinando si cada actor es una persona u otro sistema, evaluando la forma en la que este interactúa con el caso de uso, así como la cantidad de actores que existen de cada tipo. Los criterios se muestran en la siguiente tabla:

Tipo de Actor	Descripción	Peso	Cantidad * Peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, Application Programming Interface).	1	0 * 1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2	0 * 2



Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3	1 * 3
Total			3

Tabla 30: Factor de Peso de los Actores sin ajustar

Para determinar el Factor de Peso de los Casos de Uso sin ajustar se propone emplear el método basado en transacciones. Una transacción es un conjunto de actividades atómicas, lo que quiere decir que se ejecutan todas o no se ejecuta ninguna. Este proceso toma en cuenta el número de transacciones que se pueden realizar en un caso de uso y lo evalúa según la siguiente tabla:

Tipo de Caso de Uso	Descripción	Peso	Cantidad * Peso
Simple	El Caso de Uso contiene de 1 a 3 transacciones.	5	10 * 5
Medio	El Caso de Uso contiene de 4 a 7 transacciones.	10	0 * 10
Complejo	El Caso de Uso contiene más de 8 transacciones.	15	0 * 15
Total			50

Tabla 31: Factor de peso de los Casos de Uso sin ajustar

Finalmente podemos determinar que los Puntos de Casos de Uso sin ajustar resultan:

$$UUCP = UAW + UUCW$$

$$UUCP = 3 + 50$$

$$UUCP = 53$$

5.2.1.1 Cálculo de Puntos de Casos de Uso ajustados

Los Puntos de Casos de Uso ajustados son representados mediante las siglas UCP y se obtienen al



multiplicar el UUCP el TCF y el EF quedando la operación de la siguiente forma:

UCP = UUCP * TCF * EF	Donde, UCP: Puntos de Casos de Uso ajustados. UUCP: Puntos de Casos de Uso sin ajustar. TCF: Factor de Complejidad Técnica. EF: Factor de Ambiente.
------------------------------	---

Tabla 32: Cálculo de Puntos de Casos de Uso ajustados

El *Factor de Complejidad Técnica* se compone de 13 puntos que evalúan la complejidad de los módulos del sistema que se desarrolla, cada uno de estos factores tiene un peso definido y es cuantificado con un valor desde 0 a 5; donde 0 significa un aporte irrelevante y 5 un aporte muy importante. La ecuación para efectuar su cálculo es:

$$\text{TCF} = 0.6 + 0.01 * \Sigma (\text{Peso } i * \text{Valor Asignado } i)$$

Factor	Descripción	Peso	Peso * Valor
T1	Sistema distribuido.	2	2 * 4
T2	Objetivos de performance o tiempo de respuesta.	1	1 * 4
T3	Eficiencia del usuario final.	1	1 * 5
T4	Procesamiento interno complejo.	1	1 * 1
T5	El código debe ser reutilizable.	1	1 * 5
T6	Facilidad de instalación.	0.5	0.5 * 5
T7	Facilidad de uso.	0.5	0.5 * 5
T8	Portabilidad.	2	2 * 4



T9	Facilidad de cambio.	1	1 * 5
T10	Concurrencia.	1	1 * 4
T11	Incluye objetivos especiales de seguridad.	1	1 * 3
T12	Provee acceso directo a terceras partes.	1	1 * 1
T13	Se requieren facilidades especiales de entrenamiento a los usuarios.	1	1 * 2
Total			51

Tabla 33: Valores del Factor de complejidad técnica

Luego, **TCF** = $0.6 + 0.01 * 51 = 1.11$

Por otro lado el *Factor de Ambiente* requiere la evaluación de 8 puntos, que están relacionados con las habilidades y experiencia del grupo de personas involucradas con el desarrollo del proyecto. Al igual que el factor de complejidad técnica se cuantifican con valores de 0 a 5. La ecuación para su cálculo es:

$$EF = 1.4 - 0.03 * \sum (\text{Peso } i * \text{Valor Asignado } i)$$

A continuación se muestra el significado, peso y valor asignado a cada uno de los factores:

Factor	Descripción	Peso	Peso * Valor
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	1.5 * 3
E2	Experiencia en la aplicación.	0.5	0.5 * 1
E3	Experiencia en orientación a objetos.	1	1 * 5
E4	Capacidad del analista líder.	0.5	0.5 * 4
E5	Motivación.	1	1 * 5
E6	Estabilidad de los requerimientos.	2	2 * 3
E7	Personal part - time.	-1	-1 * 1
E8	Dificultad del lenguaje de programación.	-1	-1 * 2
Total			20

Tabla 34: Valores del Factor de Ambiente



Luego, $EF = 1.4 - 0.03 * 20 = 0.8$

Finalmente, podemos determinar que los Puntos de Caso de Uso ajustados resultan:

$$UCP = UUCP * TCF * EF$$

$$UCP = 53 * 1.11 * 0.8$$

$$UCP = 47.064$$

5.2.2 Estimación del esfuerzo

El esfuerzo en horas - hombre está dado por:

$E = UCP * CF$	Donde, UCP: Puntos de Casos de Uso ajustados. CF: Factor de Conversión.
----------------	---

Tabla 35: Cálculo del esfuerzo

Para determinar el *Factor de Conversión* deben contabilizarse cuántos factores de los que afectan el factor ambiente están por debajo del valor medio, en el rango desde E1 a E6, así como los que están por encima en los factores E7 y E8.

Por lo cual se establece que si:

- Total $EF \leq 2$ entonces $CF = 20$ horas - hombre
- Total $EF = 3$ o Total $EF = 4$ entonces $CF = 28$ horas - hombre
- Total $EF \geq 5$ entonces es necesario hacer cambios en el proyecto ya que el riesgo de fracaso es alto.

Para el caso del sistema de software propuesto hay un total de 2 factores que cumplen con las



Capítulo 5: Estudio de la Factibilidad

condiciones planteadas por lo que el Factor de Conversión es 20 horas – hombre.

Finalmente podemos determinar el esfuerzo:

$$E = UCP * CF$$

$$E = 47.064 * 20$$

$$E = 941.28 \approx 941 \text{ horas – hombre.}$$

Sin embargo, para lograr establecer una estimación aún más completa de la duración total que conllevaría la construcción del sistema; se debe agregar el esfuerzo de las restantes actividades. Por lo cual se plantea establecer el esfuerzo total según la siguiente distribución.

Actividad	Porcentaje	Horas - Hombre
Análisis	10%	235.25
Diseño	20%	470.5
Implementación	40%	941
Pruebas	15%	352.875
Otras actividades	15%	352.875
Total	100%	2352.5

Tabla 36: Esfuerzo del proyecto

Conociendo además, que en el marco del proyecto se trabajan como promedio 25 días al mes con un ritmo de 8 horas aproximadamente por cada sesión diaria, entonces resulta fácil determinar que la cantidad de horas que trabaja una persona en 1 mes equivale a 200.

De acuerdo al Esfuerzo Total determinado anteriormente de 2352.5 horas – hombre, se estima a partir de los datos conocidos que una persona puede realizar el trabajo en aproximadamente 12 meses.

Por tanto se definen los siguientes términos:



- Cantidad de hombres (CH): 1
- Salario promedio mensual (SPM): \$100
- Costo hombre / mes (CHM): $CH * SPM = 1 * 100 = \$100.00 / \text{mes}$
- Costo: $CHM * ET (\text{meses}) / CH = 100 * 12/1 = \1200
- Tiempo = $ET (\text{meses}) / CH = 12 / 1 = 12 \text{ meses.}$

A partir de los resultados obtenidos se puede demostrar que con un hombre trabajando en el sistema para gestionar el trabajo del servidor OpenLDAP, se puede concluir su desarrollo en aproximadamente doce meses para un costo total asociado de \$1200.

5.3 Beneficios tangibles e intangibles

El módulo de directorio perteneciente a la plataforma general para la gestión de servidores en GNU/Linux está enfocado en lograr administrar de manera profesional y eficiente el servidor OpenLDAP. Su trabajo comprende desde una posible instalación desde las fuentes hasta las más variadas configuraciones a sus archivos, incluyendo la posibilidad de modificar el estado del servicio. Preguntarse y reflexionar acerca de los beneficios que se obtienen como fruto de un sistema de este tipo genera sin lugar a dudas una larga lista. Beneficios tangibles como la posibilidad de emplear herramientas para la construcción del software que no necesiten pagar el precio de elevadas licencias, la posibilidad de obtener el código fuente de variados paquetes de software para contribuir a su estudio, modificación y redistribución, así como la capacidad de brindarle al país una utilidad propia con soporte y capacitación internos serian de por si motivos suficientes para llevar a cabo la realización de un producto que resuelva la problemática que para los administradores de redes significa, gestionar las funciones de sus servidores. Sin embargo existen otros elementos que no pueden ser medidos en términos tan concretos y que están asociados al bienestar y la satisfacción de poder emplear un producto que:

- Asegure la gestión de la información centralizada en el servidor de directorio de manera fácil e intuitiva.
- Mejore los tiempos de respuesta al usuario basándose en la sencillez de las interfaces gráficas, evitando sobrecargas de imágenes u otros elementos innecesarios. Además de llevar a cabo una



codificación clara que impida repetir sentencias anteriormente expresadas.

- Garantice la portabilidad fundamentalmente a los sistemas operativos GNU/Linux y Windows.
- Ofrezca la posibilidad al usuario de escoger el lenguaje para interactuar con el sistema.

5.4 Análisis de costos y beneficios

Resulta imposible llevar a cabo la puesta en marcha de un proyecto de software sin antes valorar los costos asociados a su producción en contraposición a los resultados que del mismo se reciban. Fruto de un análisis profundo de los elementos anteriores podrá definirse cuan factible resulta materializar el proyecto. Debido a que las herramientas empleadas para la construcción del sistema son totalmente gratuitas y los beneficios reflejados en la sección anterior realmente sobresalen por su valor; se determinó la total factibilidad para elaborar el producto.

5.5 Conclusiones

En el presente capítulo fueron tratados temas como la estimación del tiempo y esfuerzo, los costos y los beneficios de la propuesta; todo lo cual refleja un análisis profundo de factibilidad para la puesta en marcha del proyecto de gestión del servidor OpenLDAP como parte imprescindible de la Plataforma para la Gestión de Servicios Telemáticos en GNU/Linux.



CONCLUSIONES

El desarrollo actual de las telecomunicaciones vinculadas a la informática ha permitido un mejoramiento sustancial en la calidad y cantidad de servicios brindados sobre las infraestructuras de redes. Generalmente no se concibe una red actual sin al menos alguno de estos servicios. El presente trabajo está dirigido esencialmente a la gestión del servicio de Directorio, basado en OpenLDAP.

OpenLDAP constituye una implementación libre del protocolo LDAP, el cual expresa un protocolo de acceso ligero a directorios. Los directorios almacenan información perteneciente a los recursos de red y sus usuarios.

La culminación de este trabajo ha permitido entender el estado actual de sistemas similares que operan el servicio de Directorio, basados en implementaciones tanto libres como privativas. Se ha podido constatar la calidad de los servicios brindados y llegar a conclusiones apropiadas.

Siendo consecuentes con la estrategia que se ha trazado el país para la migración de sus principales servicios de red a Software Libre fueron identificados los requisitos que garantizan la correcta gestión del servicio tratado, partiendo de lo cual, se diseñó e implementó un sistema capaz de permitir la instalación y/o administración del mismo, de forma local o remota. Además resultó factible el proceso de construcción del software basado en los resultados obtenidos mediante las técnicas de estimación de costos aplicadas.

Por lo cual se consideran cumplidos los objetivos trazados al inicio de la investigación. Se espera constituya un pilar importante de apoyo al proceso de migración.



RECOMENDACIONES

Con el desarrollo del presente trabajo se recomienda:

- Continuar trabajando en el perfeccionamiento e implementación de nuevas funcionalidades en la aplicación tales como: Autenticación de forma segura con certificado SSL, migración del servicio de Directorio de Windows a OpenLDAP.
- Extender la interfaz de administración a prototipos Web y de Consolas, reutilizando las funcionalidades previstas anteriormente en el framework, dándole al administrador más de una forma de realizar su labor de acuerdo a sus preferencias o condiciones específicas del entorno.
- Utilizar la herramienta en el proceso de migración de SWL que lleva a cabo la universidad y el país.
- Agregar la herramienta a la distribución Nova.



ANEXOS

Anexo 1: Entradas del servidor de directorio OpenLDAP

Nombre de la Entrada	Clases a las que pertenece
Cuenta de Usuario / User Account	<ul style="list-style-type: none">• inetOrgPerson• posixAccount• top
Grupo / Posix Group	<ul style="list-style-type: none">• posixGroup• top
Unidad Organizativa / Organisational Unit	<ul style="list-style-type: none">• organizationalUnit• top
Rol / Organisational Role	<ul style="list-style-type: none">• organizationalRole• top
DNS / DNS Entry	<ul style="list-style-type: none">• dNSDomain• domainRelatedObject• top
Alias de Correo / Courier Mail Alias	<ul style="list-style-type: none">• inetOrgPerson• top
Cuenta de Correo / Courier Mail Account	<ul style="list-style-type: none">• inetOrgPerson



	<ul style="list-style-type: none">• top
Libro de Direcciones / Address Book	<ul style="list-style-type: none">• inetOrgPerson• top
Cuenta de Samba / Samba3 Account	<ul style="list-style-type: none">• inetOrgPerson• posixAccount• top
Grupo de Samba / Samba3 Group Mapping	<ul style="list-style-type: none">• posixGroup• top
Maquina Samba / Samba3 Machine	<ul style="list-style-type: none">• posixAccount• account• top
Objeto Simple de Seguridad / Simple Security Object	<ul style="list-style-type: none">• account• simpleSecurityObject• top

Tabla 37: Entradas en el servidor de Directorio: OpenLDAP



Anexo 2: Tipos de clases y atributos en el servidor OpenLDAP

Nombre de Clase	Clase de Estructura	Atributos Necesarios	Atributos Opcionales
account	Si	<ul style="list-style-type: none">• userid	<ul style="list-style-type: none">• description• host• localityName• organizationName• organizationalUnitName• seeAlso
alias	Si	<ul style="list-style-type: none">• aliasedObjectName	
applicationEntity	Si	<ul style="list-style-type: none">• cn• presentationAddress	<ul style="list-style-type: none">• description• l• o• ou• seeAlso• supportedApplicationContext
applicationProcess	Si	<ul style="list-style-type: none">• cn	<ul style="list-style-type: none">• description• l• ou• seeAlso
bootableDevice	No		<ul style="list-style-type: none">• bootFile• bootParameter



certificationAuthority	No	<ul style="list-style-type: none">• authorityRevocationList• cACertificate• certificateRevocationList	<ul style="list-style-type: none">• crossCertificatePair
certificationAuthority-V2	No	<ul style="list-style-type: none">• authorityRevocationList• cACertificate• certificateRevocationList	<ul style="list-style-type: none">• crossCertificatePair• deltaRevocationList
country	Si	<ul style="list-style-type: none">• c	<ul style="list-style-type: none">• description• searchGuide
cRLDistributionPoint	Si	<ul style="list-style-type: none">• cn	<ul style="list-style-type: none">• authorityRevocationList• certificateRevocationList• deltaRevocationList
dcObject	No	<ul style="list-style-type: none">• dc	
deltaCRL	No		<ul style="list-style-type: none">• deltaRevocationList
device	Si	<ul style="list-style-type: none">• cn	<ul style="list-style-type: none">• description• l• o• ou• owner• seeAlso



			<ul style="list-style-type: none">serialNumber
dmd	Si	<ul style="list-style-type: none">dmdName	<ul style="list-style-type: none">businessCategorydescriptiondestinationIndicatorfacsimileTelephoneNumberinternationaliSDNNumberlphysicalDeliveryOfficeNamepostOfficeBoxpostalAddresspostalCodepreferredDeliveryMethodregisteredAddresssearchGuideseeAlsoststreettelephoneNumberteletexTerminalIdentifiertelexNumberuserPasswordx121Address
dNSDomain	Si	<ul style="list-style-type: none">domainComponent	<ul style="list-style-type: none">ARRecordCNAMERRecordMDRecordMXRecord



			<ul style="list-style-type: none">• NSRecord• SOARecord• associatedName• businessCategory• description• facsimileTelephoneNumber• internationalISDNNumber• destinationIndicator• localityName• organizationName• physicalDeliveryOfficeName• postOfficeBox• postalAddress• postalCode• preferredDeliveryMethod• registeredAddress• searchGuide• seeAlso• stateOrProvinceName• streetAddress• telephoneNumber• teletexTerminalIdentifier• telexNumber• userPassword• x121Address
document	Si	<ul style="list-style-type: none">• documentIdentifier	<ul style="list-style-type: none">• commonName• description



			<ul style="list-style-type: none">• documentAuthor• documentLocation• documentPublisher• documentTitle• documentVersion• localityName• organizationName• organizationalUnitName• seeAlso
documentSeries	Si	<ul style="list-style-type: none">• commonName	<ul style="list-style-type: none">• description• localityName• organizationName• organizationalUnitName• seeAlso• telephonenumber
domain	Si	<ul style="list-style-type: none">• domainComponent	<ul style="list-style-type: none">• associatedName• businessCategory• description• destinationIndicator• facsimileTelephoneNumber• internationalISDNNumber• localityName• organizationName• physicalDeliveryOfficeName• postOfficeBox• postalAddress



			<ul style="list-style-type: none">• postalCode• preferredDeliveryMethod• registeredAddress• searchGuide• seeAlso• stateOrProvinceName• streetAddress• telephoneNumber• teletexTerminalIdentifier• telexNumber• userPassword• x121Address
domainRelatedObject	No	<ul style="list-style-type: none">• associatedDomain	
dSA	Si	<ul style="list-style-type: none">• cn• presentationAddress	<ul style="list-style-type: none">• description• knowledgeInformation• l• o• ou• seeAlso• supportedApplicationContext
extensibleObject	No		
friendlyCountry	Si	<ul style="list-style-type: none">• c• friendlyCountryName	<ul style="list-style-type: none">• description• searchGuide
groupOfNames	Si	<ul style="list-style-type: none">• cn	<ul style="list-style-type: none">• businessCategory



		<ul style="list-style-type: none">• member	<ul style="list-style-type: none">• description• o• ou• owner• seeAlso
groupOfUniqueNames	Si	<ul style="list-style-type: none">• cn• uniqueMember	<ul style="list-style-type: none">• businessCategory• description• o• ou• owner• seeAlso
ieee802Device	No		<ul style="list-style-type: none">• macAddress
inetOrgPerson	Si	<ul style="list-style-type: none">• cn• sn	<ul style="list-style-type: none">• audio• businessCategory• carLicense• departmentNumber• description• destinationIndicator• displayName• employeeNumber• employeeType• facsimileTelephoneNumber• givenName• homePhone



- | | | | |
|--|--|--|---|
| | | | <ul style="list-style-type: none">• homePostalAddress• initials• internationaliSDNNumber• jpegPhoto• l• labeledURI• mail• manager• mobile• o• ou• pager• photo• physicalDeliveryOfficeName• postOfficeBox• postalAddress• postalCode• preferredDeliveryMethod• preferredLanguage• registeredAddress• roomNumber• secretary• seeAlso• st• street• telephoneNumber• teletexTerminalIdentifier• telexNumber |
|--|--|--|---|



			<ul style="list-style-type: none">• title• uid• userCertificate• userPKCS12• userPassword• userSMIMECertificate• x121Address• x500uniqueIdentifier
ipHost	No	<ul style="list-style-type: none">• cn• ipHostNumber	<ul style="list-style-type: none">• description• l• manager
ipNetwork	Si	<ul style="list-style-type: none">• cn• ipNetworkNumber	<ul style="list-style-type: none">• description• ipNetmaskNumber• l• manager
ipProtocol	Si	<ul style="list-style-type: none">• cn• description• ipProtocolNumber	<ul style="list-style-type: none">• description
ipService	Si	<ul style="list-style-type: none">• cn• ipServicePort• ipServiceProtocol	<ul style="list-style-type: none">• description
labeledURIObject	No		<ul style="list-style-type: none">• labeledURI



locality	Si		<ul style="list-style-type: none">• description• l• searchGuide• seeAlso• st• street
nisMap	Si	<ul style="list-style-type: none">• nisMapName	<ul style="list-style-type: none">• description
nisNetgroup	Si	<ul style="list-style-type: none">• cn	<ul style="list-style-type: none">• description• memberNisNetgroup• nisNetgroupTriple
nisObject	Si	<ul style="list-style-type: none">• cn• nisMapEntry• nisMapName	<ul style="list-style-type: none">• description
olcBackendConfig	Si	<ul style="list-style-type: none">• olcBackend	
olcBdbConfig	Si	<ul style="list-style-type: none">• olcDatabase• olcDbDirectory	<ul style="list-style-type: none">• olcAccess• olcDbCacheFree• olcDbCacheSize• olcDbCheckpoint• olcDbConfig• olcDbDirtyRead• olcDbIDLcacheSize• olcDbIndex• olcDbLinearIndex



			<ul style="list-style-type: none">• olcDbLockDetect• olcDbMode• olcDbNoSync• olcDbSearchStack• olcDbShmKey• olcLastMod• olcLimits• olcMaxDerefDepth• olcPlugin• olcReadOnly• olcReplica• olcReplLogFile• olcRequires• olcRestrict• olcRootDN• olcRootPW• olcSchemaDN• olcSecurity• olcSizeLimit• olcSubordinate• olcSuffix• olcSyncrepl• olcTimeLimit• olcUpdateDN• olcUpdateRef
olcConfig	No		



olcDatabaseConfig	Si	<ul style="list-style-type: none">• olcDatabase	<ul style="list-style-type: none">• olcAccess• olcLastMod• olcLimits• olcMaxDerefDepth• olcPlugin• olcReadOnly• olcReplica• olcRepllogFile• olcRequires• olcRestrict• olcRootDN• olcRootPW• olcSchemaDN• olcSecurity• olcSizeLimit• olcSubordinate• olcSuffix• olcSyncrepl• olcTimeLimit• olcUpdateDN• olcUpdateRef
olcFrontendConfig	No		<ul style="list-style-type: none">• olcDefaultSearchBase
olcGlobal	Si		<ul style="list-style-type: none">• cn• olcAllows• olcArgsFile• olcAttributeOptions



- | | | | |
|--|--|--|--|
| | | | <ul style="list-style-type: none">• olcAttributeTypes• olcAuthIDRewrite• olcAuthzPolicy• olcAuthzRegexp• olcConcurrency• olcConfigDir• olcConfigFile• olcConnMaxPending• olcConnMaxPendingAuth• olcDisallows• olcDitContentRules• olcGentleHUP• olcIdleTimeout• olcIndexSubstrAnyLen• olcIndexSubstrAnyStep• olcIndexSubstrIfMaxLen• olcIndexSubstrIfMinLen• olcLocalSSF• olcLogLevel• olcObjectClasses• olcObjectIdentifier• olcPasswordCryptSaltFormat• olcPasswordHash• olcPidFile• olcPluginLogFile• olcReadOnly• olcReferral• olcReplicaArgsFile |
|--|--|--|--|



- | | | | |
|--|--|--|---|
| | | | <ul style="list-style-type: none">• olcReplicaPidFile• olcReplicationInterval• olcRepllogFile• olcRequires• olcRestrict• olcReverseLookup• olcRootDSE• olcSaslHost• olcSaslRealm• olcSaslSecProps• olcSecurity• olcSizeLimit• olcSockbufMaxIncoming• olcSockbufMaxIncomingAuth• olcSrvtab• olcTLSCACertificateFile• olcTLSCACertificatePath• olcTLSCRLCheck• olcTLSCertificateFile• olcTLSCertificateKeyFile• olcTLSCipherSuite• olcTLSDHParamFile• olcTLSSRandFile• olcTLSVerifyClient• olcThreads• olcTimeLimit• olcToolThreads |
|--|--|--|---|



olcIncludeFile	Si	<ul style="list-style-type: none">• olcInclude	<ul style="list-style-type: none">• cn• olcRootDSE
olcLdifConfig	Si	<ul style="list-style-type: none">• olcDatabase• olcDbDirectory	<ul style="list-style-type: none">• olcAccess• olcLastMod• olcLimits• olcMaxDerefDepth• olcPlugin• olcReadOnly• olcReplica• olcRepllogFile• olcRequires• olcRestrict• olcRootDN• olcRootPW• olcSchemaDN• olcSecurity• olcSizeLimit• olcSubordinate• olcSuffix• olcSyncrepl• olcTimeLimit• olcUpdateDN• olcUpdateRef
olcModuleList	Si		<ul style="list-style-type: none">• cn• olcModuleLoad



			<ul style="list-style-type: none">• olcModulePath
olcOverlayConfig	Si	<ul style="list-style-type: none">• olcOverlay	<ul style="list-style-type: none">•
olcSchemaConfig	Si		<ul style="list-style-type: none">• cn• olcAttributeTypes• olcDitContentRules• olcObjectClasses• olcObjectIdentifier
oncRpc	Si	<ul style="list-style-type: none">• cn• description• oncRpcNumber	<ul style="list-style-type: none">• description
OpenLDAProotDSE	Si		<ul style="list-style-type: none">• cn
organization	Si	<ul style="list-style-type: none">• o	<ul style="list-style-type: none">• businessCategory• description• destinationIndicator• facsimileTelephoneNumber• internationaliSDNNumber• l• physicalDeliveryOfficeName• postOfficeBox• postalAddress• postalCode• preferredDeliveryMethod• registeredAddress



			<ul style="list-style-type: none">• searchGuide• seeAlso• st• street• telephoneNumber• teletexTerminalIdentifier• telexNumber• userPassword• x121Address
organizationalPerson	Si	<ul style="list-style-type: none">• cn• sn	<ul style="list-style-type: none">• description• destinationIndicator• facsimileTelephoneNumber• internationaliSDNNumber• l• ou• physicalDeliveryOfficeName• postOfficeBox• postalAddress• postalCode• preferredDeliveryMethod• registeredAddress• seeAlso• st• street• telephoneNumber• teletexTerminalIdentifier• telexNumber



			<ul style="list-style-type: none">• title• userPassword• x121Address
organizationalRole	Si	<ul style="list-style-type: none">• cn	<ul style="list-style-type: none">• description• destinationIndicator• facsimileTelephoneNumber• internationaliSDNNumber• l• ou• physicalDeliveryOfficeName• postOfficeBox• postalAddress• postalCode• preferredDeliveryMethod• registeredAddress• roleOccupant• seeAlso• st• street• telephoneNumber• teletexTerminalIdentifier• telexNumber• x121Address
organizationalUnit	Si	<ul style="list-style-type: none">• ou	<ul style="list-style-type: none">• businessCategory• description• destinationIndicator



			<ul style="list-style-type: none">• facsimileTelephoneNumber• internationaliSDNNumber• l• physicalDeliveryOfficeName• postOfficeBox• postalAddress• postalCode• preferredDeliveryMethod• registeredAddress• searchGuide• seeAlso• st• street• telephoneNumber• teletexTerminalIdentifier• telexNumber• userPassword• x121Address
person	Si	<ul style="list-style-type: none">• cn• sn	<ul style="list-style-type: none">• description• seeAlso• telephoneNumber• userPassword
pilotDSA	Si	<ul style="list-style-type: none">• cn• presentationAddress	<ul style="list-style-type: none">• dSAQuality• description• knowledgeInformation• l



			<ul style="list-style-type: none">• o• seeAlso• supportedApplicationContext
pilotOrganization	Si	<ul style="list-style-type: none">• o• ou	<ul style="list-style-type: none">• buildingName• businessCategory• description• destinationIndicator• facsimileTelephoneNumber• internationaliSDNNumber• l• physicalDeliveryOfficeName• postOfficeBox• postalAddress• postalCode• preferredDeliveryMethod• registeredAddress• searchGuide• seeAlso• st• street• telephoneNumber• teletexTerminalIdentifier• telexNumber• userPassword• x121Address
pilotPerson	Si	<ul style="list-style-type: none">• cn	<ul style="list-style-type: none">• businessCategory



		<ul style="list-style-type: none">• sn	<ul style="list-style-type: none">• description• favouriteDrink• homePostalAddress• homeTelephoneNumber• janetMailbox• mailPreferenceOption• mobileTelephoneNumber• organizationalStatus• otherMailbox• pagerTelephoneNumber• personalSignature• personalTitle• preferredDeliveryMethod• rfc822Mailbox• roomNumber• secretary• seeAlso• telephoneNumber• textEncodedORAddress• userClass• userPassword• userid
pkiCA	No		<ul style="list-style-type: none">• authorityRevocationList• cACertificate• certificateRevocationList• crossCertificatePair



pkiUser	No		<ul style="list-style-type: none"> • userCertificate
posixAccount	No	<ul style="list-style-type: none"> • cn • gidNumber • homeDirectory • uid • uidNumber 	<ul style="list-style-type: none"> • description • gecoc • loginShell • userPassword
posixGroup	Si	<ul style="list-style-type: none"> • cn • gidNumber 	<ul style="list-style-type: none"> • description • memberUid • userPassword
qualityLabelledData	No	<ul style="list-style-type: none"> • dsaQuality 	<ul style="list-style-type: none"> • subtreeMaximumQuality • subtreeMinimumQuality
referral	Si	<ul style="list-style-type: none"> • ref 	<ul style="list-style-type: none"> •
residentialPerson	Si	<ul style="list-style-type: none"> • cn • l • sn 	<ul style="list-style-type: none"> • businessCategory • description • destinationIndicator • facsimileTelephoneNumber • internationaliSDNNumber • l • physicalDeliveryOfficeName • postOfficeBox • postalAddress • postalCode • preferredDeliveryMethod



			<ul style="list-style-type: none">• registeredAddress• seeAlso• st• street• telephoneNumber• teletexTerminalIdentifier• telexNumber• userPassword• x121Address
RFC822localPart	Si	<ul style="list-style-type: none">• domainComponent	<ul style="list-style-type: none">• associatedName• businessCategory• commonName• description• destinationIndicator• facsimileTelephoneNumber• internationalISDNNumber• localityName• organizationName• physicalDeliveryOfficeName• postOfficeBox• postalAddress• postalCode• preferredDeliveryMethod• registeredAddress• searchGuide• seeAlso• stateOrProvinceName



			<ul style="list-style-type: none">• streetAddress• surname• telephoneNumber• teletexTerminalIdentifier• telexNumber• userPassword• x121Address
room	Si	<ul style="list-style-type: none">• commonName	<ul style="list-style-type: none">• description• roomNumber• seeAlso• telephoneNumber
shadowAccount	No	<ul style="list-style-type: none">• uid	<ul style="list-style-type: none">• description• shadowExpire• shadowFlag• shadowInactive• shadowLastChange• shadowMax• shadowMin• shadowWarning• userPassword
simpleSecurityObject	No	<ul style="list-style-type: none">• userPassword	
strongAuthenticationUser	No	<ul style="list-style-type: none">• userCertificate	
subentry	Si	<ul style="list-style-type: none">• cn	



		<ul style="list-style-type: none">• subtreeSpecification	
subschema	No		<ul style="list-style-type: none">• attributeTypes• dITStructureRules• ditContentRules• matchingRuleUse• matchingRules• nameForms• objectClasses
top	No	<ul style="list-style-type: none">• objectClass	
uidObject	No	<ul style="list-style-type: none">• uid	
userSecurityInformation	No		<ul style="list-style-type: none">• supportedAlgorithms

Tabla 38: Tipos de clases y atributos del servidor OpenLDAP



Bibliografía

1. novell.com. [En línea] [Citado el: 15 de octubre de 2008.] <http://www.novell.com/es-es/products/server/management.html>.
2. redhat.es. [En línea] [Citado el: 16 de octubre de 2008.] <http://www.redhat.es/rhel/server/>.
3. monografias.com. [En línea] 2007. [Citado el: 16 de octubre de 2008.] <http://www.monografias.com/trabajos14/microsoftwindows/microsoftwindows.shtml>.
4. microsoft.com. [En línea] 2008. [Citado el: 16 de octubre de 2008.] <http://www.microsoft.com/spain/windowsserver2008/default.msp>.
5. **James Rumbaugh, Ivar Jacobson, Grady Booch.** bibliodoc.uci.cu. [En línea] 1998. [Citado el: 16 de octubre de 2008.] <http://bibliodoc.uci.cu/pdf/reg03050.pdf>.
6. —. biblioteca.uci.cu. [En línea] 2000. [Citado el: 17 de octubre de 2008.] <http://biblioteca.uci.cu/titdigitales.htm#igs>.
7. **Guarda, Alfonso de la.** dev.laptop.org. [En línea] 2004. [Citado el: 21 de enero de 2009.] http://dev.laptop.org/~edsiper/byteofpython_spanish/index.html.
8. easyeclipse.org. [En línea] 2007. [Citado el: 22 de enero de 2009.] <http://www.easyeclipse.org/site/distributions/index.html>.
9. mundogeek.net. [En línea] [Citado el: 25 de enero de 2009.] <http://mundogeek.net/archivos/2008/11/24/interfaces-graficas-de-usuario-en-python/>.
10. mic.gov. [En línea] [Citado el: 26 de enero de 2009.] <http://www.mic.gov.cu/htcentity.aspx>.



11. **Rodríguez, Medardo.** mastermagazine.info. [En línea] 2008. [Citado el: 26 de enero de 2009.] <http://www.mastermagazine.info/articulo/12671.php>.
12. *Levantamiento de requisitos.* 2007-2008.



Glosario de términos

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

Dabo: Framework para el desarrollo de aplicaciones de escritorio escrito en python, con una estructura en capas para manejar las bases de dato, los objetos de negocio y las interfaces de usuario.

G

Gazpacho: Generador de interfaces gráficas de usuario para la librería de controles GTK construido totalmente con python e inspirado en el proyecto Glade.

Glade: Herramienta de desarrollo visual de interfaces gráficas mediante GTK/GNOME. Es independiente del lenguaje de programación y predeterminadamente no genera código fuente sino un archivo XML.

GNOME: Entorno de escritorio e infraestructura de desarrollo para sistemas operativos Unix/Linux, compuesto enteramente de Software Libre.

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.



I

ipython: Intérprete de comandos interactivo para el lenguaje de programación python que ofrece una mayor introspección, depósito adicional sintaxis y diferenciación de la sintaxis.

K

KDE: Entorno de Escritorio K, es un entorno de escritorio e infraestructura de desarrollo para sistemas Unix/Linux.

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.

LGPL: GNU Lesser General Public License, licencia de software creada por la Free Software Foundation. La diferencia fundamental con la licencia GPL radica en que puede enlazarse con un programa no GPL, que puede ser Software Libre o no.

M

MFC: Microsoft Foundation Class Library, es una biblioteca que contiene porciones de la API de Windows en clases C++, incluye funcionalidades que permiten utilizar por defecto un marco de trabajo.

O

OpenLDAP: Es una implementación libre y de código abierto del protocolo Lightweight Directory Access Protocol (LDAP) desarrollada por el proyecto OpenLDAP. Está liberada bajo su propia licencia OpenLDAP Public License. LDAP es un protocolo de comunicación independiente de la plataforma.

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

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

PSFL: Python Software Foundation License, licencia de Software Libre compatible con la GPL, su uso primario es para la distribución de proyectos de software con python.

Q

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

R

RAID: Redundant Array of Independent Disks, constituye un conjunto redundante de discos independientes. Hace referencia a un sistema de almacenamiento que usa múltiples discos duros entre los que distribuye o replica los datos.

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

S

Scite: Editor de textos multiplataforma, muy ligero, diseñado principalmente para edición de código fuente, y resaltado de sintaxis, tiene referencia de función en línea para muchos lenguajes de programación.

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.

W

widgets: Componentes visuales empleados para el diseño de interfaces, a partir de ellos pueden crearse



vistosos relojes en pantalla, notas, calculadoras, calendarios, agendas, juegos, ventanas con información, entre otros componentes.

wxGlade: Diseñador de interfaces escrito en python que emplea wxPython.