

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

Título: “Desarrollo de herramienta de integración con
Directorio Activo para Nova”

-Facultad 1-

Autor: Pedro Yaisel Arruebo Sosa

Tutor(es): Ing. Miguel Albalat Águila

Ing. Dairelys García Rivas

Consultantes: Msc. Dariem Pérez Herrera

Ing. Abel A. Fírvida Donéstevez

Ing. Amaury Viera Hernández

"La libertad no es poder elegir entre unas pocas opciones impuestas, sino tener el control de tu propia vida. La libertad no es elegir quien será tu amo, es no tener amo."

"Las obras de conocimiento deben ser libres, no hay excusas para que no sea así."

Richard Stallman

"El hombre más poderoso es aquel que es totalmente dueño de sí mismo."

Aristóteles

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter no exclusivo.

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

Pedro Yaisel Arruebo Sosa

Ing. Miguel Albalat Águila

Ing. Dairelys García Rivas

Dedicatoria:

Dedico mi trabajo de diploma a mi madre:

Eres la única persona del mundo que siempre estás, de forma incondicional. Si te rechazo, me perdonas. Si me equivoco, me acoges. Si los demás no pueden conmigo, me abres una puerta. Si estoy feliz, celebras conmigo. Si estoy triste, no sonríes hasta que me hagas reír. Eres mi amiga incondicional. Una madre es capaz de dar todo sin recibir nada. De querer con todo su corazón sin esperar nada a cambio. Mi madre encuentra la felicidad cuando yo la encuentro. Cuando yo vivo algo hermoso, lo vive a través de mi experiencia. Mi madre reza por mí, incluso cuando yo solo rezo por mí mismo. Mi madre me daría el mundo entero si fuese capaz. Mi madre es la mujer más bella que jamás conocí. Jamás en la vida encontraré ternura mejor, más profunda, más desinteresada ni verdadera que la de mi madre.

Agradecimientos:

A la Revolución:

Por crear esta Universidad de excelencia y darme la oportunidad de estudiar en ella.

A la FEU:

Porque ha hecho de mí el revolucionario que soy hoy y por los hermosos momentos que viví en la Universidad.

A mis Tutores:

Por el apoyo que me han dado, por sus consejos, muchas gracias. Este trabajo también es de ustedes.

A mis hermanos de causa y excelentes amigos:

Por la confianza depositada en mí, por los momentos que hemos pasado juntos, por lo que me enseñaron (Julio César, Joyce, Liván, Lester, José Joel, Víctor, Marcos, Gabriel, Luis Ángel, Eiler, Eliecer, Arai, Cire a todos gracias)

Al equipo de todos los días:

Carlos, Yoyo, Josué, Lorenzo, Jochy, Julio, Glendy, han sido muchas las victorias, gracias por su apoyo, son excelentes amigos.

A los buenos amigos:

Melissa, Tiquito, Iván, Heidy, Roexcy, José Daniel, gracias por soportarme.

A los de siempre:

Ustedes que siempre han estado presentes para mí, en las buenas y las malas: Yoel, Daily, Yissel, Diana, Ramón, Darien. Realmente gracias, les debo mucho.

A mi familia:

A mis tíos (Papi, Luisa, Anais, Neiralis, Cuqui), mi hermana, mis primos y mis sobrinos gracias por esperar lo mejor de mí, por los consejos, y por todo el apoyo que me han dado durante estos 25 años.

A la persona, más especial en mi vida:

Mami gracias y mil veces gracias, a ti debo que me encuentre hoy aquí, no hay palabras en el mundo, que me permitan agradecerte lo suficiente. Tú estás presente en cada párrafo, cada palabra de este trabajo.

Al niño robot:

Por tu esfuerzo, y empeño. Continúa siendo quien eres.

Resumen:

En Cuba, el Grupo Ejecutivo se hace cargo del proceso de migración a software libre y la OSRI dictamina las normas de seguridad para las redes informáticas. Nova, sistema operativo cubano, debe cumplir con las condiciones establecidas por los antes mencionados para poder ser desplegado en las empresas e instituciones cubanas. Sin embargo, no cuenta con una herramienta que realice el proceso de integración de los ordenadores con el sistema operativo al directorio activo. En la presente investigación se estudian las herramientas existentes en el mundo para ejecutar este tipo de tareas, en función de seleccionar una de ellas para incluirse en el sistema operativo. De esta forma, se ofrecerá la posibilidad a cada institución que utilice servicios telemáticos donde se gestionan usuarios, de unirse a un Directorio Activo (DA) para la gestión de los mismos.

Este programa permitirá integrar los ordenadores con bajas prestaciones de hardware existentes en las instituciones con los DA, solucionando la situación actual, donde este proceso se realiza manualmente o con la utilización de herramientas que entorpecen en gran medida el rendimiento del SO. Para la utilización del mismo el usuario no necesitará poseer amplios conocimientos sobre los SO GNU/Linux o los DA.

Palabras clave: Código Fuente, Directorio Activo, Integración, Nova.

Abstract:

In Cuba, the Executive Group it's the responsible of the free software migration process and the OSRI dictates the security rules for networks. Nova, Cuban operating system, must accomplish the conditions established by the previously mentioned to be deployed in Cuban institutions and enterprises. Nevertheless, it doesn't count with a tool to perform the integration process of the computers with the operating system to the active directory. In the present investigation the existing tools for executing this kind of tasks are investigated, in order to select one of them to be included in the operating system. This way, the possibility will be offered to each institution that uses telematics services where users are managed, to join the Active Directory to manage them.

This program will allow to integrate the low hardware computers existing on the institutions with the ADs, solving the current situation, where this process is carried manually or with tools that greatly hinders the performance of the OS. To use it the user won't need to have wide knowledge about GNU/Linux OS or ADs.

Keywords: Active Directory, Integration, Nova, Source Code.

Tabla de Contenido

| | |
|---|----|
| Introducción | 1 |
| Capítulo 1: Fundamentación Teórica..... | 5 |
| 1.1 Los Servicios de Directorio..... | 5 |
| 1.2 Descripción del protocolo LDAP | 5 |
| 1.2.1 Funcionamiento de LDAP | 6 |
| 1.2.2 Usos de LDAP | 7 |
| 1.3 Microsoft Active Directory | 7 |
| 1.3.1 Características generales | 7 |
| 1.3.2 Estructura | 8 |
| 1.4 Estudio del Estado del Arte | 9 |
| 1.4.1 Alternativas al Active Directory en la actualidad..... | 10 |
| A. 389 Directory Server | 10 |
| B. OpenLDAP..... | 11 |
| 1.4.2 Soluciones que permiten la integración con el AD desde el servidor..... | 12 |
| A. Centrify Express..... | 12 |
| B. Quest Authentication Services | 12 |
| 1.4.3 Soluciones que permiten la integración con el AD desde el Cliente | 13 |
| A. Likewise Open | 13 |
| B. SSSD..... | 14 |
| C. Forma Manual | 15 |
| 1.5 Comparación de las herramientas existentes..... | 15 |
| 1.6 Metodología y Herramientas | 17 |
| 1.6.1 Metodología de Desarrollo..... | 17 |
| 1.6.2 Herramienta CASE | 18 |
| 1.6.3 Lenguaje de Programación..... | 18 |
| 1.6.4 El entorno de desarrollo..... | 19 |
| 1.6.5 Distribución de la solución | 20 |
| 1.7 Conclusiones del Capítulo..... | 20 |
| Capítulo 2: Descripción de la solución propuesta | 21 |
| 2.1 Propuesta de Solución | 21 |
| 2.1.1 Especificación de los Requerimientos de Software | 22 |
| Requisitos Funcionales | 22 |
| Requisitos no Funcionales | 23 |
| 2.1.2 Modelo de Dominio | 24 |
| 2.2 Patrones utilizados en el desarrollo de la aplicación | 25 |
| 2.2.1 Patrones Generales de Software para la Asignación de Responsabilidades (GRASP)..... | 25 |
| 2.2.2 Patrones de creación de software (GoF) | 26 |
| 2.2.3 Estilo arquitectónico de CaliX | 27 |
| 2.2.4 CaliX | 27 |
| 2.3 Diseño | 27 |
| 2.3.1 Diagrama de Casos de Uso del Sistema | 28 |
| 2.3.2 Descripción de los actores del sistema..... | 28 |
| 2.3.3 Descripción de los Casos de Uso del Sistema | 29 |
| 2.3.4 Diagramas de clases | 34 |
| 2.4 Conclusiones del Capítulo..... | 35 |
| Capítulo 3: Implementación y Realización de Pruebas..... | 36 |
| 3.1 Implementación | 36 |

| | |
|---|----|
| 3.1.1 Vista de Despliegue | 36 |
| 3.1.2 Diagrama de Componentes | 37 |
| 3.2 Pruebas del Sistema | 38 |
| 3.2.1 Pruebas de Aceptación | 38 |
| 3.2.2 Pruebas de Caja Negra | 39 |
| 3.2.3 Casos de prueba..... | 39 |
| 3.3 Resultados obtenidos | 46 |
| 3.3.1 Sobre las funcionalidades obtenidas | 46 |
| 3.3.2 Resumen de las pruebas | 47 |
| 3.4 Conclusiones del capítulo..... | 48 |
| Conclusiones Generales | 49 |
| Recomendaciones | 50 |
| Referencias bibliográficas | 51 |
| Bibliografía | 54 |
| Anexos | 57 |
| Glosario de Términos..... | 62 |

Índice de Ilustraciones

| | |
|---|----|
| Figura 1 Funcionamiento de LDAP | 6 |
| Figura 2 Esquema de un DA..... | 9 |
| Figura 3 Metodología Open UP | 17 |
| Figura 4 Ciclo de vida Open UP..... | 18 |
| Figura 5 Modelo conceptual de la solución propuesta | 21 |
| Figura 6 Modelo de dominio..... | 25 |
| Figura 7 Implementación del patrón Singleton..... | 26 |
| Figura 8 Implementación del patrón Factory Method..... | 27 |
| Figura 9 Diagrama de casos de uso del sistema | 29 |
| Figura 10 Diagrama de clases del diseño | 35 |
| Figura 11 Diagrama de despliegue | 37 |
| Figura 12 Diagrama de componentes | 37 |
| Figura 13 Resultados de las pruebas de aceptación | 47 |

Índice de Tablas

| | |
|---|----|
| Tabla 1 Datos obtenidos en las encuestas realizadas a Empresas, Centros e Instituciones cubanas. | 10 |
| Tabla 2 Comparativa entre las herramientas de integración con los DA | 16 |
| Tabla 3 Descripción de los módulos que componen al módulo principal de la solución | 22 |
| Tabla 4 Descripción de los actores del sistema | 28 |
| Tabla 5 Descripción del CU Modificar dominio..... | 30 |
| Tabla 6 Descripción del CU Modificar nombre PC | 30 |
| Tabla 7 Descripción del CU Eliminar ordenador del dominio | 31 |
| Tabla 8 Descripción del CU Iniciar aplicación | 32 |
| Tabla 9 Descripción del CU Integrar al DA..... | 33 |
| Tabla 10 Descripción del CU Autenticar | 34 |
| Tabla 11 Sección Modificar dominio | 40 |
| Tabla 12 Sección Modificar nombre PC | 41 |
| Tabla 13 Sección Eliminar ordenador del dominio | 42 |
| Tabla 14 Sección Iniciar aplicación | 43 |
| Tabla 15 Sección Autenticar..... | 44 |
| Tabla 16 Sección Integrar al DA..... | 46 |

Introducción

En el mundo, el uso y la creación de distribuciones de sistemas basados en tecnologías de software libre y código abierto (FOSS), de tipo GNU/Linux, han ido evolucionando y ampliando sus horizontes. Esto se debe a que les permiten a los usuarios ciertas libertades: estudiar, ejecutar, copiar, distribuir, cambiar y mejorar el software, teniendo como elemento significativo el acceso al código fuente.

Cuba por razones de seguridad, política y soberanía tecnológica, es un ejemplo de los países latinoamericanos que han apostado por los sistemas FOSS. En el año 2004 con la aprobación del “Acuerdo 084/2004” por el Consejo de Ministros en el cual se determinó la necesidad de migrar hacia sistemas de código abierto, se creó el Grupo Ejecutivo Nacional encabezado por el Ministerio de la Informática y las Comunicaciones. Componen este grupo: el Grupo Legal, encargado de respaldar el proceso; el Grupo de Capacitación, cuya tarea reside en preparar a las personas para el cambio; y el Grupo Técnico, que ha elaborado una guía de migración, referencia para ordenar el proceso y señalar las aplicaciones sustitutas de aquellas en código propietario.

En el año 2005, con el objetivo de apoyar el proceso de migración un grupo de jóvenes de la Universidad de las Ciencias Informáticas (UCI), pone en marcha la creación de la Distribución Cubana de GNU/Linux Nova (Fundación Wikimedia, Inc., 2013). Este proceso se hizo extensivo a los Órganos de la Administración Central del Estado (OACE), por lo que Nova debió adaptarse a las condiciones imperantes en cada uno de los OACE.

La Oficina de Seguridad para las Redes Informáticas (OSRI) dictamina que para cada institución que utilice servicios telemáticos donde se gestionan usuarios se debe utilizar un Directorio Activo para la gestión de los mismos¹. Para esto es necesario que los ordenadores clientes se encuentren integrados con el Directorio Activo, requisito que en los sistemas GNU/Linux actuales, incluyendo a Nova, se logra de forma manual o con la utilización de herramientas que entorpecen en gran medida el rendimiento del Sistema Operativo(SO).

Nova debe adaptarse a esta política de forma segura y transparente. Además debe contemplar la situación del proceso de migración en la cual las instituciones por razones tecnológicas no puedan migrar su Directorio Activo de Windows hacia tecnologías libres como OpenLDAP.

Por lo que se hace necesario incluir en la distribución un método o herramienta que permita integrar los ordenadores con Nova a los Directorios Activos, de acuerdo con lo planteado anteriormente. Por ello se identificó como problema científico:

¹ **RESOLUCION No. 127 /2007** del Ministerio de la Informática y las Comunicaciones, Sección Quinta, Artículo 45 (Resolución 127 Sobre el Reglamento de Seguridad Informática, 2007).

¿Cómo hacer que Nova se adapte al entorno de las empresas cubanas que utilicen Directorios Activos (DA)?

Se define como objeto de estudio la integración de ordenadores con SO GNU/Linux Nova a DA, tomando como campo de acción la herramienta que permite la integración de ordenadores con SO GNU/LINUX Nova a DA.

El objetivo general de esta investigación es integrar Nova a los Directorios Activos mediante una herramienta de configuración de servicios, para ampliar la socio adaptabilidad² de la distribución cubana de GNU/Linux.

Como objetivos específicos se definieron:

- ✓ Caracterizar el entorno informático de las empresas cubanas.
- ✓ Estudiar acerca de la integración de SO GNU/Linux a DA.
- ✓ Estudiar las tecnologías utilizadas en la implementación de herramientas para la integración de SO a DA.
- ✓ Elaborar el diseño de una herramienta para la integración de SO a DA.
- ✓ Implementar una herramienta para la integración de SO a DA.
- ✓ Realizar pruebas de calidad al producto de esta investigación (Calix).

Teniendo definidos los objetivos específicos de la investigación, se determina conoidea a defender que: Sí se integra la Distribución Cubana de GNU/Linux Nova a los DA utilizando una herramienta de configuración de servicios, se logrará aumentar la socio adaptabilidad de la misma.

Con el propósito de dar cumplimiento a la idea a defender, se hace necesario realizar las siguientes tareas:

- ✓ Realización de una encuesta en empresas cubanas, para caracterizar como es su entorno informático, y así determinar cuáles son los DA más usados en ellas y el por ciento de ordenadores con bajas prestaciones de hardware.
- ✓ Realización de un estudio acerca de la integración de sistemas operativos GNU/Linux con DA.
- ✓ Realización de un estudio sobre las tecnologías utilizadas en la implementación de herramientas para la integración de los ordenadores con Nova a los DA.
- ✓ Realización del proceso de análisis y diseño a la herramienta para integrar los ordenadores con Nova a

² Socio-adaptabilidad: Las bases tecnológicas para la informatización de Cuba, deben ser hechas por cubanos y para los cubanos, logrando inigualable adaptabilidad a las condiciones de nuestro País.(Pierra Fuentes, 2011)

los DA.

- ✓ Implementación de la herramienta para integrar los ordenadores con Nova a los DA.
- ✓ Realización de un conjunto de pruebas de caja negra para determinar los posibles errores y así ir evaluando las no conformidades existentes.
- ✓ Realización de pruebas de aceptación para determinar los niveles de conformidad de los usuarios.

Esta investigación está sustentada sobre la base de la utilización de diferentes métodos científicos. Se decidió recurrir a los métodos tanto teóricos como empíricos. De los teóricos se utilizaron el histórico-lógico, analítico-sintético, causal y el de modelación. De los empíricos solo las encuestas y la observación.

- ✓ Histórico-Lógico: se aplicó con el objetivo de verificar teóricamente cómo ha evolucionado el tema tratado en esta investigación.
- ✓ Analítico-sintético: se utilizó en el análisis de la bibliografía relacionada con los sistemas que permiten realizar la integración de los sistemas operativos GNU/Linux con los Directorios Activos, posibilitando su comprensión y permitiendo la captura de aquellos elementos que se relacionan con el objeto de estudio.
- ✓ Modelación: se empleó para mostrar los diferentes artefactos que se construyen como resultado del proceso de Ingeniería de Software.
- ✓ Encuestas: se realizaron a los usuarios y a especialistas con experiencia en la administración de sistemas operativos GNU/Linux, y de redes privadas.
- ✓ Observación: se utilizó en la identificación del problema científico una vez analizada la situación problemática, además en la fase de pruebas con el fin de comprobar el cumplimiento de los requisitos.

El cumplimiento de todas las tareas propuestas permitirá obtener un producto que posea una documentación actualizada de los errores más críticos que puedan surgir durante el uso del mismo, que cuente además con una explicación detallada del proceso ingenieril llevado a cabo y la forma de implementar dicha solución. Contribuirá además a que las personas se muestren menos reacias al cambio que supone el proceso de migración.

El presente documento cuenta con 3 capítulos, además las secciones Glosario de términos, Referencias bibliográficas, Bibliografía y Anexos; los capítulos quedan organizados de la siguiente forma:

CAPÍTULO I: “Fundamentación Teórica” donde se explicarán los conceptos fundamentales sobre la tecnología con la que se trabajará y se realizará una investigación sobre las alternativas existentes actualmente a esta tecnología. Se hará una descripción de las herramientas y la metodología de las que se hará uso para el desarrollo de la solución propuesta.

CAPÍTULO II: “Descripción de la solución propuesta” donde se explicará el funcionamiento de la misma mediante un modelo conceptual, los requisitos funcionales y no funcionales de la misma. Se determinará la arquitectura, los casos de uso, los actores del sistema y otros artefactos que se generen en la Ingeniería de Software.

CAPÍTULO III: “Implementación de la Solución y Realización de Pruebas” donde se procederá a implementar la solución propuesta en el Capítulo II y se ejecutarán pruebas de caja negra al producto resultantes, además se realizaran pruebas de aceptación al mismo.

Capítulo 1: Fundamentación Teórica

Durante el presente capítulo se abordan los principales conceptos teóricos relacionados con los Servicios de Directorio. Se realizará el estudio del Estado del Arte actual sobre la tecnología de Servicios de Directorio y las herramientas que permiten la integración de los SO GNU/Linux a estos, se llevará a cabo una comparación de estas y se determinará qué solución se tomará para resolver la situación problemática expuesta en la introducción del trabajo. Además se describirán la metodología y las herramientas con que se llevará a cabo el proceso de construcción de la solución propuesta.

1.1 Los Servicios de Directorio

Los servicios de directorio son almacenes de información acerca de entidades de red, como aplicaciones, archivos, impresoras y usuarios. Estos proporcionan una manera consistente de nombrar, describir, localizar, acceder, administrar y asegurar información acerca de esos recursos. Muchas organizaciones crean almacenes especializados o servicios de directorio dentro de sus aplicaciones para permitir una funcionalidad específica que requieren sus clientes. Los servicios de directorio se manifiestan a través del protocolo LDAP³ acrónimo de (*Lightweight Directory Access Protocol*⁴), del cual a nivel mundial se han realizado diferentes implementaciones (Desmond, et al., 2008).

1.2 Descripción del protocolo LDAP

LDAP es un protocolo a nivel de aplicación que concede el acceso a un servicio de directorio ordenado y distribuido para buscar diversa información en un entorno de red. Normalmente se utiliza para acceder a la información almacenada de usuarios, grupos y equipos de una organización (Carter, 2003).

Conocido el significado de directorio, se hace necesario conocer en qué consiste LDAP. Existen diferentes estándares que especifican Servicios de Directorio, siendo el denominado X.500⁵ tal vez el más conocido. El estándar X.500 define de forma nativa un protocolo de acceso denominado DAP (Directory Access Protocol). Este protocolo de acceso resulta muy complejo (y computacionalmente pesado) porque está definido sobre la

³ La definición detallada de LDAP está disponible en el RFC2251 "The Lightweight Directory Access Protocol (v3)" (The Internet Society, 1997)

⁴ Llevado al español Lightweight Directory Access Protocol significa: Protocolo Ligero de Acceso a Directorios.

⁵ Conjunto de protocolos producido por la Unión Internacional de Telecomunicaciones (ITU por sus siglas en inglés) en la década de 1980.

pila completa de niveles del modelo OSI. Como alternativa a DAP para acceder a directorios de tipo X.500, LDAP ofrece un protocolo ligero casi equivalente, pero mucho más sencillo y eficiente, diseñado para operar directamente sobre TCP/IP (Carter, 2003).

1.2.1 Funcionamiento de LDAP

El protocolo LDAP está basado en un modelo cliente-servidor. Uno o más servidores incluyen los datos que conforman el árbol del directorio LDAP o base de datos troncal. El cliente se conecta con el servidor LDAP y le hace una consulta. El servidor contesta con la respuesta adecuada, o bien con una indicación de dónde puede el cliente hallar más información (normalmente otro servidor LDAP) (Donnelly, 2000).

No afecta con qué servidor se conecte el cliente: siempre se observará la misma vista del directorio; el nombre que se le presenta a un servidor LDAP hace referencia a la misma entrada a la que haría referencia en otro servidor LDAP. Esta es una particularidad importante de un servicio de directorios universal como LDAP (Donnelly, 2000). En la siguiente figura se visualiza el concepto explicado.



Figura 1 Funcionamiento de LDAP

1.2.2 Usos de LDAP

LDAP puede ser utilizado con varios propósitos, como por ejemplo la administración centralizada de usuarios, conservando todas las cuentas de usuario en una única ubicación, lo que no significa que quede albergada en un único servidor puesto que LDAP soporta alta disponibilidad y redundancia (Timothy A. Howes, et al., 2003). Sin embargo, puede destinarse igualmente para otros fines, como:

- ✓ Infraestructura de clave pública.
- ✓ Calendario compartido.
- ✓ Libreta de direcciones compartida.
- ✓ Almacenamiento para DHCP, DNS, etc.
- ✓ Directivas de configuración para las clases del sistema (guardando registros de las configuraciones de varios servidores).

La integración de sistemas es la pieza angular que debe afrontar un administrador de sistemas. Cuando en una organización deben convivir diferentes sistemas operativos, el administrador debe facilitar a los usuarios la forma de acceder a los recursos independientemente de la plataforma que éstos decidan utilizar. LDAP resuelve estas dificultades de manera eficiente, provee soporte para integración de cualquier sistema operativo, debido a su carácter estándar (Timothy A. Howes, et al., 2003).

1.3 Microsoft Active Directory

Active Directory (AD) permite la administración de Windows y otras aplicaciones de Microsoft desde un único punto. AD está diseñado específicamente para operar con otros directorios y administrarlos, independientemente de su ubicación o de los SO subyacentes. Para ello ofrece una amplia compatibilidad con las normas y los protocolos existentes y proporciona interfaces de programación de aplicaciones que facilitan la comunicación con otros directorios (Desmond, et al., 2013).

1.3.1 Características generales

Esta tecnología permite gestionar los recursos de una red, tratando estos como objetos, con sus respectivos atributos. Para esta gestión, la información de los objetos se guarda en una especie de base de datos de forma jerárquica. Los objetos son agrupados en forma de árboles, que interactúan entre ellos para su funciona-

miento, dependiendo en los dominios donde los recursos estén ubicados, y de ciertas relaciones de confianza; lo cual permite relaciones de objetos que estén en árboles distintos (Kouti, et al., 2004).

Un servicio de directorio, como AD, proporciona métodos para almacenar los datos del directorio y ponerlos a disposición de los administradores y los usuarios de la red. Por ejemplo, AD almacena información acerca de las cuentas de usuario (nombres, contraseñas, números de teléfono, etc.) y permite que otros usuarios autorizados de la misma red tengan acceso a esa información (Kouti, et al., 2004).

1.3.2 Estructura

El AD trabaja bajo una serie de protocolos para la comunicación entre los objetos que gestiona, tales como LDAP, DNS, DHCP, etc. (García Chico, 2011); a continuación una referencia a los estándares más destacados:

- ✓ DHCP (Dynamic Host Configuration Protocol). Protocolo de configuración dinámica de ordenadores, que permite la administración desatendida de direcciones de red.
- ✓ DNS (Domain Name System). Servicio de nombres de dominio que permite la administración de los nombres de ordenadores. Este servicio constituye el mecanismo de asignación y resolución de nombres (traducción de nombres simbólicos a direcciones IP) en Internet.
- ✓ SNTP (Simple Network Time Protocol). Protocolo simple de tiempo de red, que permite disponer de un servicio de tiempo distribuido.
- ✓ LDAP (Lightweight Directory Access Protocol). Protocolo ligero (o compacto) de acceso a directorio. Este es el protocolo mediante el cual las aplicaciones acceden y modifican la información existente en el directorio.
- ✓ KerberosV5. Versión 5 del protocolo utilizado para la autenticación de usuarios y máquinas.
- ✓ Certificados X.509. Estándar que permite distribuir información a través de la red de una forma segura.

Para un mejor entendimiento del concepto de servicio de directorio y su estructura vea la Figura 2 Esquema de un DA, la cual gráfica qué es la centralización de la información y la administración de una red de ordenadores, lo cual es el objetivo principal de un servicio de directorio.

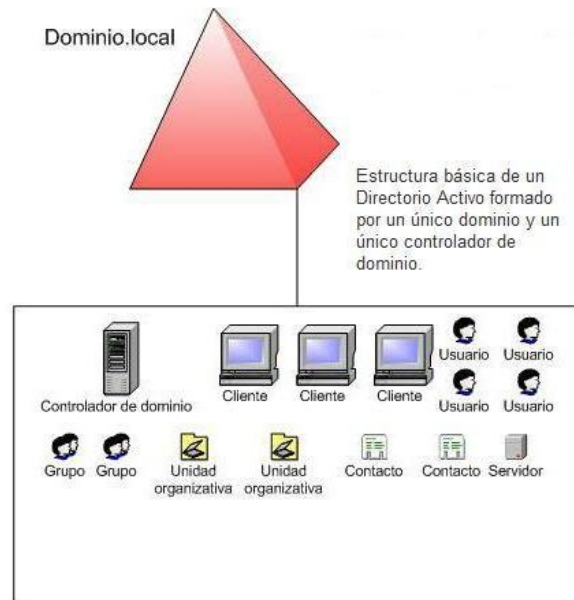


Figura 2 Esquema de un DA

1.4 Estudio del Estado del Arte

Una encuesta⁶ realizada a diez empresas, centros e instituciones cubanas arrojaron los datos expresados en la Tabla 1. De la misma se resume, que de las empresas, centros e instituciones encuestadas, el 89% usa Active Directory como tecnología para la administración centralizada de las entidad en su red privada y solo el 11% tecnologías libres (OpenLDAP). En todas ellas existen ordenadores con bajas prestaciones de hardware, el 56 % posee más de 100 ordenadores con estas características, otro 22% entre 50 y 100 ordenadores con estas características y el 22% restante menos de 50. Esto demuestra que sin duda alguna todas aquellas que posean una red privada de ordenadores, necesitan de las facilidades que brinda el uso de un DA, para la gestión y organización de los recursos existentes en la misma, pero es evidente que no existe una homogenización en el uso de la tecnología para resolver esta problemática así como una fuerte presencia de equipos de cómputos de bajas prestaciones. En nuestro país, debido a la necesidad real que existe de alcanzar una mayor independencia tecnológica, desde hace algunos años se puso en marcha un plan de migración a tecnologías libres. En el camino, uno de los mayores obstáculos que se encuentran las empresas, es integrar los sistemas GNU/Linux con el AD de Microsoft. Este es un producto comercial, y como tal está pensado para que sea compatible solo con los sistemas Windows, aunque existen actualmente otras alternativas libres al AD. Igualmente para lograr cierta integración de los sistemas UNIX con el AD existen un

⁶ Ver sección Anexos

conjunto de herramientas que funcionan desde el Controlador de Dominio, y otras desde el cliente, aunque de estas solo existen tres soluciones conocidas.

| Nombre Empresa, Centro o Institución. | Tecnología usada para la administración centralizada de las entidades. | ¿Existen en su empresa ordenadores con bajas prestaciones de hardware? | ¿Cuántas? |
|---------------------------------------|--|--|----------------|
| Banco Metropolitano | Active Directory | Si | entre 50 y 100 |
| Joven Club | Active Directory | Si | más de 100 |
| DATYS | Active Directory | Si | menos de 50 |
| Desoft | Active Directory | Si | más de 100 |
| AcinoX | Active Directory | Si | entre 50 y 100 |
| Aduana General de la República. | OpenLDAP | Si | menos de 50 |
| Infomed | Active Directory | Si | más de 100 |
| ABAPET | Active Directory | Si | más de 100 |
| UCI | Active Directory | Si | más de 100 |

Tabla 1 Datos obtenidos en las encuestas realizadas a Empresas, Centros e Instituciones cubanas.

1.4.1 Alternativas al Active Directory en la actualidad

A. 389 Directory Server

389 Directory Server o Fedora Directory Server, posteriormente bautizado como Red Hat Directory Server para su versión comercial, no es más que la simplificación gráfica del lenguaje LDAP para administrar directorios centralizados de usuarios en una red corporativa o personal (MediaWiki, 2013).

CARACTERÍSTICAS PRINCIPALES

- ✓ Replicación Multi-Master, para proporcionar tolerancia a fallos y alto rendimiento de escritura.
- ✓ Escalabilidad: miles de operaciones por segundo, decenas de miles de usuarios al mismo tiempo, decenas de millones de entradas, cientos de gigabytes de datos.
- ✓ El código base se ha desarrollado y desplegado continuamente por el mismo equipo de trabajo por más de una década.
- ✓ Extensa documentación, incluida la ayuda de instalación y guías de implementación.

- ✓ Sincronización con usuarios de AD y de grupos.
- ✓ Autenticación segura y transportable (SSLv3, TLSv1 y SASL).
- ✓ Apoyo a LDAPv3.
- ✓ El tiempo de inactividad es cero, basado en la actualización del esquema LDAP, configuración, administración y el árbol de acceso a la información de control (ACI).
- ✓ Consola gráfica para todas las facetas de usuario, grupo y gestión de servidores.

B. OpenLDAP

El proyecto OpenLDAP se inició en 1998 por Kurt Zeilenga. El mismo comenzó como un clon de la implementación LDAP de la Universidad de Michigan, entidad donde se desarrolló originalmente el protocolo LDAP y que también actualmente trabaja en la evolución del mismo. Esta implementación libre del protocolo LDAP desarrollada por el proyecto OpenLDAP, se encuentra bajo su propia licencia OpenLDAP Public License (Jackiewicz, 2004). La gran mayoría de las distribuciones de GNU/Linux incluyen este software para el soporte del protocolo LDAP, entre ellas Debian, CentOS, Ubuntu y sus derivados, destacando en este grupo a Nova.

CARACTERÍSTICAS PRINCIPALES

- ✓ Operaciones de lectura muy rápidas. Debido a la naturaleza de los datos almacenados en los directorios las lecturas son más comunes que las escrituras.
- ✓ Datos relativamente estáticos. Los datos almacenados en los directorios no suelen actualizarse con mucha frecuencia.
- ✓ Entorno distribuido, fácil replicación.
- ✓ Estructura jerárquica. Los directorios almacenan la información de forma jerárquica de manera nativa.
- ✓ Orientadas a objetos. El directorio representa a elementos y a objetos. Los objetos son creados como entradas, que representan a una colección de atributos.
- ✓ Esquema Standard. Los directorios utilizan un sistema estándar que pueden usar fácilmente diversas aplicaciones.
- ✓ Atributos Multi-valor. Los atributos pueden almacenar un valor único o varios.
- ✓ Replicación Multi-master. Muchos de los servidores LDAP permiten que se realicen escrituras o actualizaciones en múltiples servidores.(Jackiewicz, 2004)

1.4.2 Soluciones que permiten la integración con el AD desde el servidor

A. Centrify Express

Centrify Express es una completa suite de soluciones libres de la integración de AD basadas en: inicio de sesión único, acceso remoto, compartir archivos, monitorear la autenticación para la integración de Active Directory y la seguridad en la nube para los sistemas de plataforma cruzada. Centrify Express no es una solución puntual para un sistema operativo único, sino que abarca todas las versiones de los ampliamente desplegados en su mayoría de UNIX, Linux y Mac OS X (Centrify Corporation., 2013).

CARACTERÍSTICAS PRINCIPALES

- ✓ Single Sign-On (SSO) para estaciones de trabajo Mac y Linux. Centrify integra sistemas Mac y Linux en el dominio de AD con facilidad y no intrusiva, que permite la autenticación centralizada y control de acceso. (Centrify Corporation., 2013)
- ✓ Mejora de la seguridad y la racionalización de los procesos de TI mediante la sustitución de la infraestructura NIS legado con la autenticación centralizada y control de acceso.
- ✓ Descubre los sistemas no-Windows y se une a ellos en AD.(Centrify Corporation., 2013)

B. Quest Authentication Services

Quest Authentication Services (anteriormente Vintela Authentication Services) permite a las organizaciones establecer control de acceso, autenticación y autorización a lo largo de toda la compañía Quest Software, para sistemas Unix y Linux empleando la infraestructura existente de Gestión de Identidades para recursos Windows (AD). Con Quest Authentication Services, los recursos no basados en Windows se integran en el conjunto de recursos controlados por el AD, amplía su seguridad, sus funciones de autenticación y su capacidad para gestionar el cumplimiento de directivas a los sistemas Unix y Linux.(Dell, Inc., 2013)

CARACTERÍSTICAS PRINCIPALES

- ✓ Cumplimiento de normativas. Quest Authentication Services se basa en los mismos estándares del sector que Directorio Activo para organizar un sistema alternativo homologable capaz de integrar múltiples repositorios de identidad, múltiples puntos de autenticación y directorios no homologados como NIS.
- ✓ SSO. Al incorporar sistemas Unix y Linux (y muchas aplicaciones) dentro del conjunto de objetos gestionados desde el DA, ya puede disfrutar de las posibilidades de una validación de usuarios y

aplicaciones de SSO basada en DA aplicable a algunas de las parcelas más complejas de gestionar de su organización.

- ✓ Simplificación de la Gestión de Identidades. Quest Authentication Services simplifica la gestión de Identidades al ampliar las capacidades nativas del Directorio Activo a plataformas no basadas en Windows y añadiendo algunas funcionalidades avanzadas para la administración de identidades.
- ✓ Directorio Activo para Unix/Linux. Quest Authentication Services dispone de capacidad para ampliar de forma natural una infraestructura existente de AD al resto de la organización. Quest Authentication Services permite la integración nativa de sistemas Unix y Linux que van a poder operar como miembros normales del Directorio Activo, y beneficiándose de las ventajas que ofrece el AD en materia de seguridad y cumplimiento de normativas.
- ✓ SSO. Quest Authentication Services implementa Kerberos y LDAP de forma nativa sobre sistemas Unix y Linux de la misma forma que se emplean estos estándares en sistemas Windows. Además el producto ofrece capacidad de autenticación centralizada para muchas aplicaciones (p.ej. SAP). Una API potente brinda soporte de esta tecnología a aplicaciones desarrolladas internamente, guía para crear soluciones de SSO para muchas aplicaciones muy difundidas (p.ej. Oracle y DB2) y autenticación basada en smart-cards (tarjetas inteligentes) para Linux y Unix contra el AD (Dell Inc., 2013).
- ✓ Gestión de Identidades más sencilla. Quest Authentication Services aporta la tecnología necesaria para resolver la gestión de identidades empleando las infraestructuras existentes de DA. Con este producto, las soluciones de gestión de la identidad basadas en DA, como los de aprovisionamientos, gestión de contraseñas, auditoría y elaboración de informes, tanto de Quest como de otros fabricantes, pueden extenderse también a sistemas no basados en Windows.

1.4.3 Soluciones que permiten la integración con el AD desde el Cliente

A. Likewise Open

Likewise Open es un programa de código abierto que permite integrar plataformas distintas a Microsoft Windows con AD, de esta manera se podrá autenticar de forma segura y centralizada con las credenciales de este (Likewise Software., 2008).

Likewise Open es compatible con Linux, Unix y Mac OS X, por lo que puede centralizar la gestión de todos los equipos, autenticar usuarios y autorizar el acceso a los recursos. Likewise Open es libre de descargar y utilizar de acuerdo con los términos de la Licencia Pública General GNU. El Agente Likewise al estar instalado en Linux y UNIX, se integra con el SO e implementa la asignación para cualquier aplicación que utilice el

servicio de nombres (NSS) o módulo de autenticación conectable (PAM). Un ejemplo de una aplicación tipo PAM es el proceso de inicio de sesión (/bin/login) (Likewise Software., 2008).

Likewise actúa como un cliente de Kerberos 5 para la autenticación y como un cliente LDAP para la autorización (Likewise Software., 2008). El agente también incluye muchísimas bibliotecas o librerías que utiliza para configurar todos los servicios que necesita para su funcionamiento, lo cual lo hace muy pesado, y un alto consumidor de recursos. Por otra parte le facilita el trabajo al usuario al no tener que preocuparse por hacer todas esas configuraciones.

Likewise no resulta complicado de utilizar, el proceso de integrarse al DA con este es bastante intuitivo. La documentación disponible para Likewise aparece solo en inglés y se encuentra discontinuada.

B. SSSD

SSSD es un demonio⁷. Su función principal es proporcionar acceso a la identidad y la autenticación de recurso remoto a través de un marco común que puede proporcionar soporte de almacenamiento en caché y fuera de línea para el sistema. Se proporciona una interfaz de NSS y PAM para el sistema y también una mejor base de datos para almacenar los usuarios locales, así como de datos extendidos de usuario (Red Hat, Inc., 2011).

SSSD también es extensible, se puede configurar para utilizar nuevas fuentes y mecanismos de autenticación de identidad en caso de que surjan. Además, es totalmente compatible con IPv6 (Red Hat, Inc., 2011).

CARACTERÍSTICAS PRINCIPALES

- ✓ Autenticación fuera de línea: Uno de los principales beneficios de SSSD es la autenticación fuera de línea. Esto resuelve el caso de los usuarios que tienen una cuenta separada corporativa y una de equipo local debido a la exigencia común de implementar una red privada virtual (VPN). (Red Hat, Inc., 2011)
- ✓ Reducción de carga del servidor: El uso de SSSD también ayuda a reducir la carga de los servidores de identificación. Por ejemplo, al usar `nss_ldap`⁸, cada aplicación cliente que necesita solicitar información sobre el usuario abre su propia conexión con el servidor LDAP. La gestión de estas conexiones múltiples puede dar lugar a una pesada carga en el mismo. En un sistema de cliente

⁷ En sistemas UNIX se conoce como demonio o daemon (Disk And Execution Monitor) a un proceso que se ejecuta en segundo plano del sistema operativo, se ejecuta en todo momento y no posee interacción directa con el usuario, también se le conoce genéricamente como servicio o proceso, del cual no se percibe su ejecución. Un demonio realiza una operación específica en tiempos predefinidos o en respuesta a ciertos eventos del sistema.

⁸ Librería o biblioteca del protocolo LDAP, para hacer uso del servicio de nombres NSS.

SSSD, sólo el proceso SSSD proveedor de datos es quien se comunica con el servidor LDAP, disminuyendo la carga a una conexión por cada sistema cliente.(Red Hat, Inc., 2011)

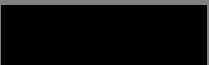






- ✓ Soporte para varios dominios: puede utilizar SSSD para especificar múltiples dominios del mismo tipo, lo cual, comparado con un archivo de configuración nsswitch.conf, con la que sólo se puede solicitar información de los usuarios desde un único servidor de cualquier tipo particular (LDAP, NIS, etc.), resulta superior. Con SSSD, se pueden crear múltiples dominios de la misma, o de diferentes tipos de identidad.(Red Hat, Inc., 2011)
- ✓ Diferenciación de los usuarios del mismo nombre: SSSD apoya la diferenciación de los usuarios del mismo nombre en diferentes dominios. Por ejemplo, se puede diferenciar al usuario kate en el dominio de ldap.example.com, del usuario kate en el dominio ldap.myhome.com.(Red Hat, Inc., 2011)
- ✓ La integración con IPA: va más allá de la autenticación fuera de línea, administración de dominio múltiple y otras características ya descritas, SSSD también está diseñado para integrarse y mejorar la funcionalidad de los clientes IPA. En un entorno con la última versión de la API instalada, SSSD proporciona funcionalidad adicional, incluyendo soporte para actualizaciones de DNS dinámico, basado en el host de control de acceso y contraseña, migración desde un entorno LDAP (sólo en el LDAP / Kerberos 5) ambiente empleado por el IPA.(Red Hat, Inc., 2011)

C. Forma Manual

Esta no es más que un script hecho en el lenguaje de programación BASH, el cual fue publicado por la Dirección de Redes y Seguridad Informática de la Universidad de las Ciencias Informáticas (UCI), el cual trae consigo algunas configuraciones básicas. Esta de todas las formas es la forma más compleja y menos factible para alcanzar el propósito de integrar un ordenador al AD. Puede encontrar el código fuente en la sección Anexos.

1.5 Comparación de las herramientas existentes

A continuación se muestra una tabla en la que se realiza una comparación de las herramientas ya señaladas en el presente capítulo, teniendo en cuenta aspectos como la facilidad de uso, si es compatible o no con GNU/Linux, el consumo de recursos, la cantidad de funcionalidades que brinda, la existencia de documentación sobre la misma y si es posible hacer uso de estas desde el cliente.

| <i>Características</i> | <i>Usabilidad (Alta, Media, Baja)</i> | <i>Compatibilidad con Plataformas Libres</i> | <i>Tipo de Licencia (Libre, Privativa)</i> | <i>Consumo de Recursos (Alto, Medio, Bajo)</i> | <i>Nivel de Funcionalidades (Avanzado, Medio, Básico)</i> | <i>Funciona desde el Cliente</i> | <i>Brinda documentación</i> |
|--|---------------------------------------|--|--|--|---|----------------------------------|-----------------------------|
|  | Alta | si | libre | alto | avanzado | no | si |
|  | Baja | si | libre | Medio | medio | si | si |
|  | Alta | si | libre | alto | avanzado | no | si |
|  | Alta | no | privativa | alto | avanzado | no | si |
|  | Alta | no | privativa | alto | avanzado | no | si |
|  | Baja | si | libre | bajo | básico | si | no |
|  | Alta | si | libre | medio | medio | si | si |

Simbología por Colores:

| | | | |
|------------------------|---|---------------------------|---|
| ✓ 389 Directory Server |  | ✓ Centrify Express |  |
| ✓ SSSD |  | ✓ Quest Software Services |  |
| ✓ OpenLDAP |  | ✓ Integración Manual |  |
| ✓ Likewise Open |  | | |

Tabla 2 Comparativa entre las herramientas de integración con los DA

Después de realizar el análisis de la tabla mostrada, se llega a la conclusión de que ninguna de estas herramientas por separado constituye una solución viable a la situación encontrada en las empresas cubanas luego de haberse sistematizado la situación de las mismas en este capítulo. Por lo que se decide implementar un producto informático que las resuelva.

1.6 Metodología y Herramientas

1.6.1 Metodología de Desarrollo

Como metodología de desarrollo se utilizará la metodología ágil Open Up⁹, la cual constituye una alternativa eficaz para proyectos pequeños y con pocos recursos, en los que se aprecia un desarrollo rápido, iterativo e incremental y cuenta con una personalización realizada para el trabajo en el proyecto productivo Nova (Fernández del Monte, y otros, 2011).

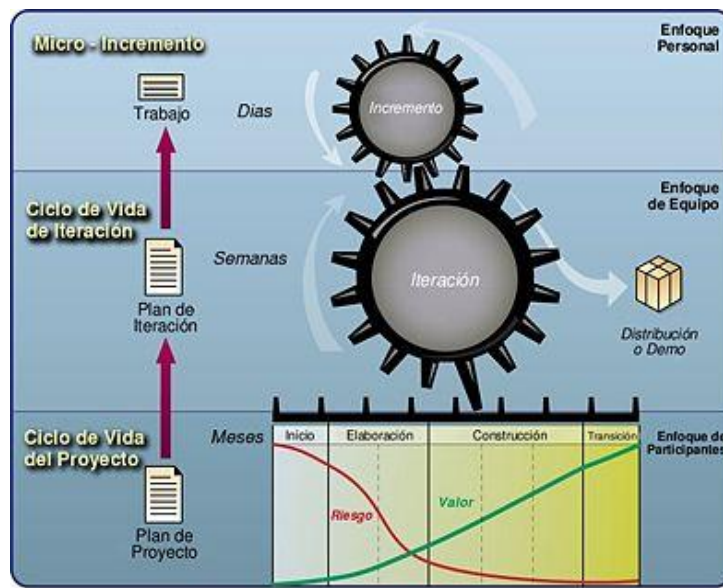


Figura 3 Metodología Open UP

Open UP define un ciclo de vida completo para el desarrollo de sistemas de software. Está diseñado para soportar y hacer seguimiento a las actividades diarias de pequeños equipos de trabajo (no superiores a 10 personas). Es un proceso iterativo cuyas diferentes iteraciones se distribuyen a través de cuatro fases: Inicio, Elaboración, Construcción y Transición.

⁹Acrónimo de "Open Unified Process", traducido al español como "Proceso unificado abierto". Es un método y proceso de desarrollo de software presentada por IBM (Acrónimo de "International Business Machines" en el año 2006 como producto de las mejores características de Rational Unified Process (RUP) (Kroll, y otros, 2006).

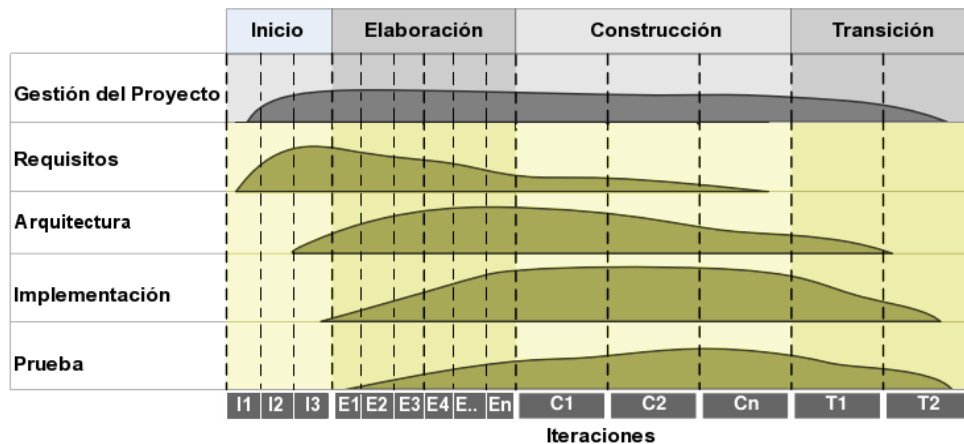


Figura 4 Ciclo de vida Open UP

1.6.2 Herramienta CASE

Como herramienta de modelado CASE¹⁰ se propone utilizar Visual Paradigm, la cual permite la representación gráfica en lenguaje UML¹¹ del ciclo de vida completo del desarrollo de software, características que podemos encontrar en herramientas similares a esta como: Rational Rose y ArgoUML ambas herramientas privativas; Visual Paradigm posee la característica de ser una aplicación multiplataforma de probada utilidad para el analista y diseñada para la construcción de sistemas de forma confiable a través de la utilización de un enfoque orientado a objetos.

1.6.3 Lenguaje de Programación

Se escoge como lenguaje de programación a Python, debido a:

- ✓ Posee una amplia biblioteca de módulos, los cuales tienen funcionalidades que pueden ser reutilizadas. Esto a su vez permite agilizar el proceso de desarrollo y pruebas.
- ✓ La curva de aprendizaje de este lenguaje es mayor que la de lenguajes como: C, C++, Java, Csharp, etc.
- ✓ El intérprete de Python se encuentra por defecto en el sistema base de la distribución y posee pocas dependencias de software.

¹⁰Acrónimo de “Computer Aided Software Engineering”, traducido al español como “Ingeniería de Software Asistida por Computadora”.

¹¹Acrónimo de “Unified Modeling Language”, traducido al español como “Lenguaje de Modelado Unificado”. Junto al proceso de desarrollo de software RUP, constituyen la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

- ✓ Al mismo tiempo, la gran mayoría de los desarrollos del proyecto Nova se realizan en este lenguaje, lo cual hace del mismo una opción fiable, pues facilitará la integración de la solución con el resto de las aplicaciones del SO.
- ✓ Python es interpretado, lo cual permite que las aplicaciones desarrolladas haciendo uso de él sean portables y se puedan ejecutar en cualquier SO.
- ✓ Como otro punto a tener en cuenta, está que el equipo de desarrollo posee un alto dominio de este lenguaje.

1.6.4 El entorno de desarrollo

Como herramienta de programación se utilizará el Geany, basado en el editor de textos Scintilla. Geany tiene soporte para varios lenguajes de programación, como C, C++, Java, JavaScript, PHP, HTML, CSS, Python, Perl, Ruby, Fortran, Pascal, Haskell entre tantos otros. Algunas de las características más destacadas de Geany son:

- ✓ Auto completado.
- ✓ Soporte multi-documento.
- ✓ Soporte de proyectos.
- ✓ Coloreado de sintaxis.
- ✓ Emulador de terminal incrustado.

Algunas de las utilidades más conocidas son:

- ✓ Compatible con la mayoría de los lenguajes.
- ✓ Varios paneles para acceder mejor a los datos.
- ✓ Herramientas para compilar.
- ✓ Buscador integrado.
- ✓ Posibilidad de compilar y ejecutar directamente desde el entorno (en todos los lenguajes orientados a esta labor).
- ✓ Descomposición y representación de las clases y estructuras del código.
- ✓ Posibilidad de ampliar funcionalidades mediante complementos.

1.6.5 Distribución de la solución

Para distribuir a los usuarios la solución obtenida, se incluirá en las versiones oficiales liberadas por el proyecto Nova y en el repositorio oficial del proyecto. Para esto será necesario realizar un proceso de empaquetamiento de software (H. Bahr, 2010) para distribuciones GNU/Linux. Dicho proceso consiste en proporcionar las aplicaciones en forma de paquetes instalables de software. Estos paquetes están formados por los programas ejecutables de la aplicación, así como por todas las bibliotecas de las que depende y otros tipos de ficheros (como imágenes, ficheros de audio, traducciones y localizaciones, etc.), de forma que se proporcionan como un conjunto. Las bibliotecas de las que depende el programa pueden haber sido enlazadas tanto de forma dinámica como también estática. Por tanto, el usuario percibe al paquete como un conjunto que representa al programa en sí, cuando en realidad incluye otros ficheros. En la distribución Debian, o las que están basadas en esta, el artefacto de salida de dicho proceso es un paquete de extensión .deb, listo para ser instalado (SPI et al., 2013).

1.7 Conclusiones del Capítulo

Luego de abordar los conceptos asociados a los Servicios de Directorio, se logra obtener un mejor dominio sobre el tema abordado en el desarrollo de la investigación. Describiendo algunos de los principales software existentes que permiten de una forma u otra, una integración parcial con el AD y haciendo una comparación entre los mismos, se concluye que las herramientas mencionadas no cumplen los objetivos trazados por la investigación, por lo cual no se hará uso de ellas. Se concluye además que no se conoce actualmente una herramienta que le permita a las empresas cubanas integrar la distribución cubana de GNU/Linux Nova a un AD, que sea compatible con la distribución, con un alto nivel de usabilidad, con un bajo consumo de recursos, e implemente las funcionalidades suficientes para una buena integración, además de poseer una buena documentación. Por lo tanto, se hace necesario el desarrollo de una herramienta en el proyecto Nova, que cumpla los objetivos del marco teórico de la investigación.

Capítulo 2: Descripción de la solución propuesta

En el presente capítulo se realiza una descripción detallada de la solución propuesta, exponiendo sus principales conceptos y características. Además se centrará en explicar el modo en que se hace uso de la metodología de desarrollo de software Open Up, que guiará los esfuerzos para la construcción exitosa de la solución propuesta, que ha sido nombrada CaliX.

Como parte de este proceso de desarrollo, está previsto generar sobre la marcha los artefactos propuestos por la metodología. Todos los artefactos elaborados se expondrán como anexos, y en caso de ser necesario se incluirán algunos de ellos en el cuerpo del capítulo.

2.1 Propuesta de Solución

Para lograr una integración de la Distribución Cubana de GNU/Linux Nova con Directorios Activos, se propone una herramienta que se encargue de automatizar el proceso desde el cliente. Se procura que sea fácil e intuitiva de usar para cualquier tipo de usuario, disminuyendo la cantidad de pasos en los que se complete el proceso de integración. A continuación, en la Figura 5 Modelo conceptual de la solución propuesta, se describe el funcionamiento de la solución de manera general.

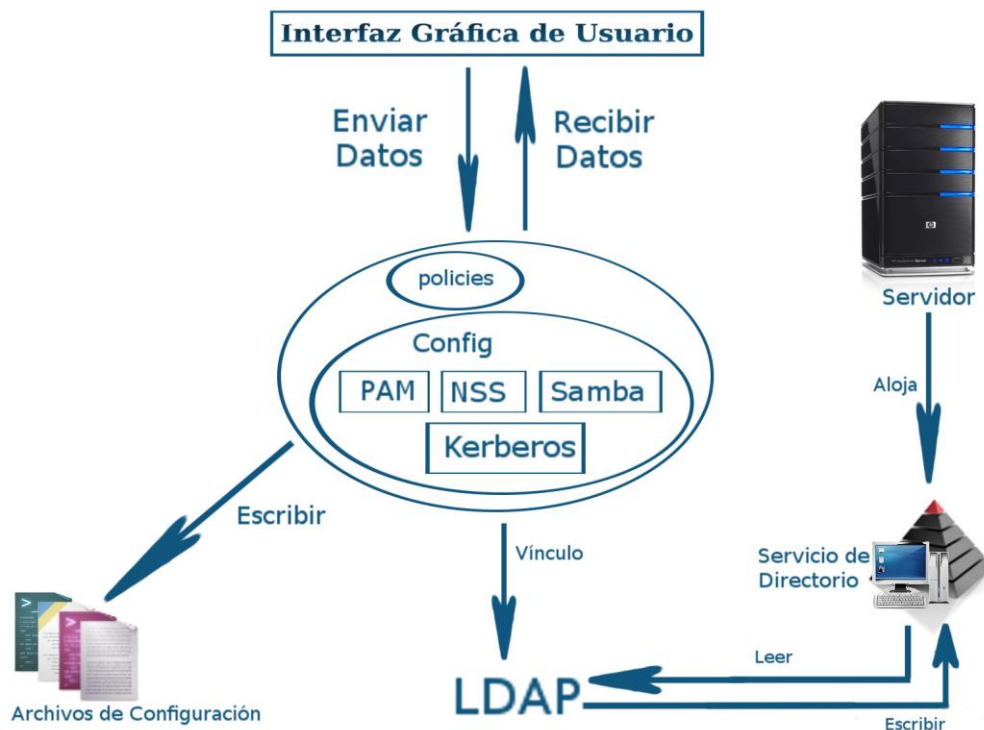


Figura 5 Modelo conceptual de la solución propuesta

Como se puede apreciar en la figura, la solución cuenta primeramente con una interfaz gráfica que se encarga de interactuar con el usuario del sistema, proporcionarle datos al módulo principal y mostrar la información que recibe de este al usuario. El módulo principal está compuesto por otros dos módulos, los cuales son descritos en la siguiente tabla.

| Nombre del Módulo | Descripción |
|-------------------|--|
| policias | El manejo de las políticas de seguridad establecidas en el DA según lo dictaminado por la OSRI, es uno de los RF propuestos para la solución dada, que permitirá una integración mucho más amplia con el mismo. Este módulo se encarga de realizar las configuraciones necesarias para establecer estas políticas en el ordenador cliente. |
| Config | Es el encargado de todas las configuraciones necesarias para una correcta integración con el DA. Hace uso de las funcionalidades que brindan módulos más pequeños para realizar los cambios en los archivos de configuración de las librerías del sistema (PAM, NSS, Samba y Kerberos). |

Tabla 3 Descripción de los módulos que componen al módulo principal de la solución

2.1.1 Especificación de los Requerimientos de Software

Los requisitos funcionales y no funcionales que se presentan a continuación brindarán más detalles de la solución propuesta.

Requisitos Funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir (E. Wiegers, 2009).

RF_1 Autenticar Usuario SO.

RF_2 Determinar estado de la Integración.

RF_3 Buscar Dominio.

RF_4 Buscar Nombre Ordenador.

RF_5 Modificar Dominio.

RF_6 Modificar Nombre Ordenador.

RF_7 Integrar al Directorio Activo.

RF_8 Configurar las políticas de seguridad.

RF_8.1 Configurar políticas de contraseña.

RF_8.2 Configurar políticas de actualización.

RF_8.3 Configurar políticas de permisos de usuario y grupo.

RF_9 Verificar Campos Vacíos.

RF_10 Validar Campos.

RF_11 Verificar Conexión de Red.

RF_12 Eliminar la Integración.

RF_13 Reiniciar el Sistema.

Requisitos no Funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable (Rumbaugh, 2000).

Eficiencia

- ✓ De acuerdo al volumen de información, las transacciones en las operaciones y la complejidad de las mismas en el servidor, el sistema debe consumir entre el 5% y el 10% del espacio en memoria, teniendo esta un espacio mínimo de 128MB.

Usabilidad

- ✓ La aplicación poseerá un ambiente sencillo y será fácil de manejar por los usuarios, incluso aquellos que no han tenido mucha experiencia en el trabajo con computadoras o con sistemas informáticos.
- ✓ El tiempo de respuesta de la aplicación debe ser entre 5 y 60 segundos, permitiendo al usuario hacer sus operaciones de manera satisfactoria.
- ✓ La aplicación deberá permitir que luego de concluido el proceso de integración, y una vez que el usuario haya iniciado por primera vez su sesión del dominio, la siguiente vez el sistema permita que el usuario se autentique e inicie su sesión aún sin conexión con el DA.

- ✓ Todo el proceso de integración deberá ser realizado por la aplicación, de forma tal que el usuario realice la menor cantidad de tareas para lograr su objetivo.

Restricciones de diseño

- ✓ La aplicación debe hacerse utilizando el lenguaje de programación Python.
- ✓ Para cumplir con los requisitos de diseño y creación de gráficos del módulo se debe utilizar la herramienta GLADE.
- ✓ Para la codificación de la herramienta se debe utilizar el editor de textos Geany.
- ✓ Para crear el módulo se debe utilizar la herramienta Visual Paradigm.

Apariencia o interfaz externa

- ✓ La aplicación debe poseer una apariencia acorde con los estándares establecidos por el proyecto Gnome. Los componentes visuales deben comportarse de manera consistente y predecible, o sea el comportamiento debe ser el mismo independientemente del entorno en que sea utilizado.

Software

- ✓ La aplicación debe funcionar sobre los sistemas operativos GNU/Linux.

Hardware

- ✓ Para un correcto funcionamiento, el ordenador donde se ejecute deberá poseer un Microprocesador 1,7 GHz o superior, 256 MB mínimo de memoria RAM y una tarjeta de red.

Seguridad

- ✓ Solo el usuario root¹² o un usuario con permisos de administración podrán hacer uso de la aplicación..

2.1.2 Modelo de Dominio

Debido a la relativa simplicidad del entorno donde está enmarcado el sistema y el conocimiento que se posee acerca de su funcionamiento, no es necesario realizar un Modelo de Negocio para comprender la problemática que ha de resolverse, siendo suficiente un Modelo de Dominio (Modelo Conceptual). Este último es una representación visual de los conceptos u objetos del mundo real significativos para un problema o área de interés. Representa clases conceptuales del dominio del problema, conceptos del mundo real, no de

¹² Nombre de usuario o la cuenta que por defecto tiene acceso a todos los comandos y archivos en un sistema operativo Linux o Unix. También se le conoce como la cuenta de root, usuario root y el superusuario.

componentes de software (Sommerville, 2005). La Figura 6 Modelo de dominio, muestra el Modelo de Dominio realizado, teniendo en cuenta lo planteado en la descripción del problema.

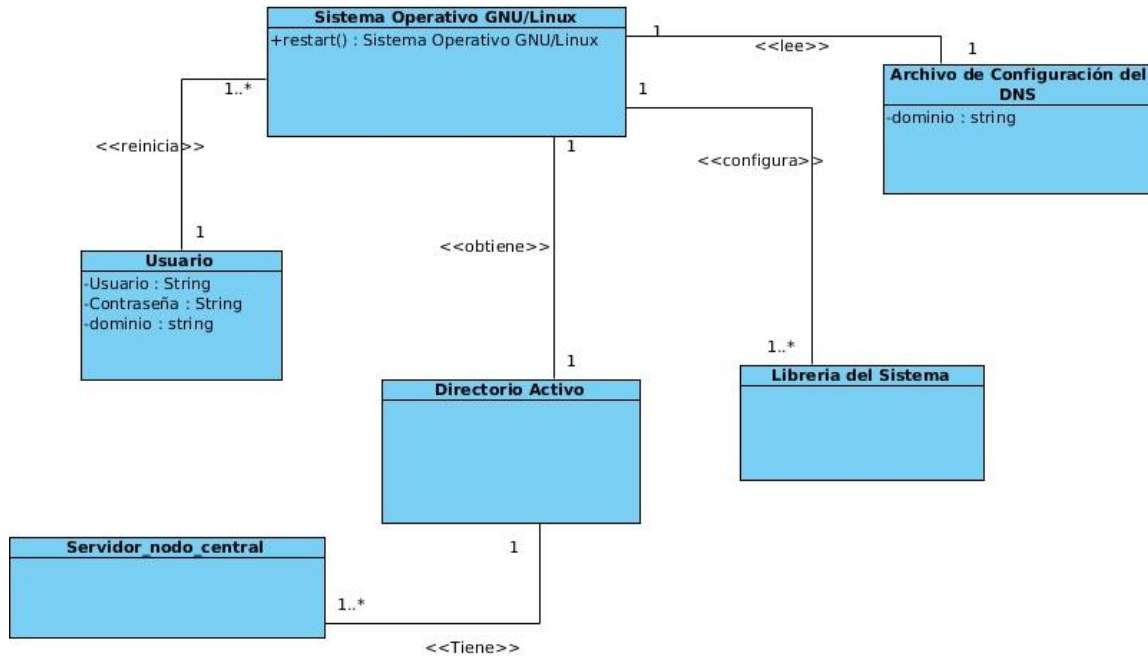


Figura 6 Modelo de dominio

2.2 Patrones utilizados en el desarrollo de la aplicación

Los patrones arquitectónicos, son patrones de diseño de software (con un mayor nivel de abstracción) que ofrecen soluciones a problemas de arquitectura de software en ingeniería de software. Brindan una descripción de los elementos y el tipo de relación que tienen junto con un conjunto de restricciones sobre cómo pueden ser usados. Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones. (Pressman, 2011)

2.2.1 Patrones Generales de Software para la Asignación de Responsabilidades (GRASP)

Experto: La responsabilidad de realizar una labor es de la clase que tiene o puede tener los datos involucrados. La aplicación se desarrolló de forma modular, donde cada módulo es independiente de los restantes, y a su vez el máximo responsable de la información que maneja.

Bajo acoplamiento: El bajo acoplamiento se muestra al existir la menor dependencia posible entre módulos, para que en caso de producirse una modificación en alguno de ellos, se tenga la mínima repercusión posible en el resto.

Alta cohesión: Los módulos poseen un número relativamente pequeño de responsabilidades, con una importante funcionalidad relacionada y poco trabajo por hacer, definiendo así que cada módulo realice sólo las funcionalidades para las cuales fue creado, generando un bajo acoplamiento y fomentando la reutilización.

Controlador: asigna la responsabilidad de controlar el flujo de eventos del sistema, a clases específicas. Esto facilita la centralización de actividades (validaciones, seguridad, etc.). El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Le ofrece mayor potencial de los componentes reutilizables, garantizando que la empresa o los procesos de dominio sean manejados por la capa de los objetos de dominio y no por la de la interfaz.

2.2.2 Patrones de creación de software (GoF)

Singleton: Para el desarrollo de la aplicación se decidió utilizar el patrón de creación Singleton, el cual garantiza que una clase tenga una sola instancia y proporciona un punto global a ella, lo cual es necesario para el sistema pues se hacen varias llamadas a una misma clase. Además, permite el acceso controlado a esta única instancia (Gamma, et al., 2005). A continuación se muestra un ejemplo de cómo se realiza la implementación del mismo:

```
class Aplicacion( object ):
    __instancia = None
    __componentes = None

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

Figura 7 Implementación del patrón Singleton

Factory method: Permite separar la clase que crea los objetos, de la jerarquía de objetos a instanciar. Además tiene como ventajas la centralización de la creación de objetos, el facilitar la escalabilidad del sistema y el hecho de que permite que el usuario se abstraiga de la instancia a crear por el programa (Gamma, et al., 2005). La implementación del patrón se realiza como se muestra a continuación:

```
def Programa(self, var):  
    """Implementación del patrón Factory Method"""  
    if var=="ok":  
        self.createA()  
    else:  
        self.createB()
```

Figura 8 Implementación del patrón Factory Method

2.2.3 Estilo arquitectónico de CaliX

La arquitectura definida para la implantación de la aplicación es una arquitectura por capas y se define como el estilo de programación a utilizar. Su objetivo principal es la separación de la capa de presentación, capa de negocios y la capa de datos. Este estilo en capa constituye una organización jerárquica. Las capas están compuestas por subsistemas y entidades complejas (Sommerville, 2005).

La capa de presentación es la que presenta el sistema al usuario, le comunica la información y la captura cuando este la introduce, en un mínimo de procesos. Esta capa se comunica únicamente con la capa de negocio. En la capa de negocio es donde se reciben las peticiones del usuario y se envían las respuestas tras el proceso, además se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar la creación de objetos y la manipulación de los mismos. En la capa de datos se conservarán archivos de configuración que constituirán la base del sistema.

2.2.4 CaliX

Se espera que **CaliX** como software sea capaz de automatizar el proceso de integración de los clientes GNU/Linux a los DA, permitiendo a los administradores de red conocer el estado de estos clientes. Además, debe brindar la posibilidad del manejo de las políticas establecidas en el DA, extendiendo las posibilidades de control de los administradores de red sobre estos ordenadores. La aplicación también contará con funciones avanzadas como la autenticación sin conexión. Brinda al mismo tiempo la posibilidad de utilizar un bajo consumo de recursos, lo que amplía su campo de acción a ordenadores con bajas prestaciones de hardware.

2.3 Diseño

Diseñar un sistema es crear la estructura interna y el comportamiento de tal forma que el mismo sea robusto, fácil de extender y de alta calidad. Un buen diseño mejora la calidad y hace que el sistema sea fácil de mantener y extender; por el contrario un diseño pobre puede aumentar de forma significativa los costos asociados con la producción y el mantenimiento del software (Pressman, 2011).

El diseño es una abstracción del código que presenta el sistema desde una perspectiva que hace más sencillo manejar la estructura y el comportamiento del software. Si bien es cierto que desde el código también se pueden definir los elementos estructurales y de comportamiento, es mucho más complicado tratar de afrontarlos a través de esta vía. Los diseños pueden ser modelos visuales, bosquejos simples, descripciones tipo texto, etc. En general un buen diseño describe claramente cómo los diferentes elementos del sistema interaccionan para cubrir los requerimientos.

2.3.1 Diagrama de Casos de Uso del Sistema

Un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. Los casos de uso pretenden ser herramientas simples para describir el comportamiento del software o de los sistemas. Los casos de uso no describen ninguna funcionalidad interna (oculta al exterior) del sistema, ni explican cómo se implementarán. Simplemente muestran los pasos que el actor sigue para realizar una tarea (Alarcón, 2000: Capítulo 5, pág. 56-58).

Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema. El diagrama de casos de uso documenta el comportamiento de un sistema desde el punto de vista del usuario. Representan las funciones que un sistema puede ejecutar (Alarcón, 2000: Capítulo 6, pág. 64-66).

2.3.2 Descripción de los actores del sistema

| Actores | Responsabilidad |
|----------------------------|---|
| Usuario del SO | Es el encargado de brindar los privilegios necesarios para iniciar la aplicación e interactuar con ella. |
| Usuario del Dominio | Es el encargado de suministrar los datos necesarios para realizar el proceso de integración al Directorio Activo. |

Tabla 4 Descripción de los actores del sistema

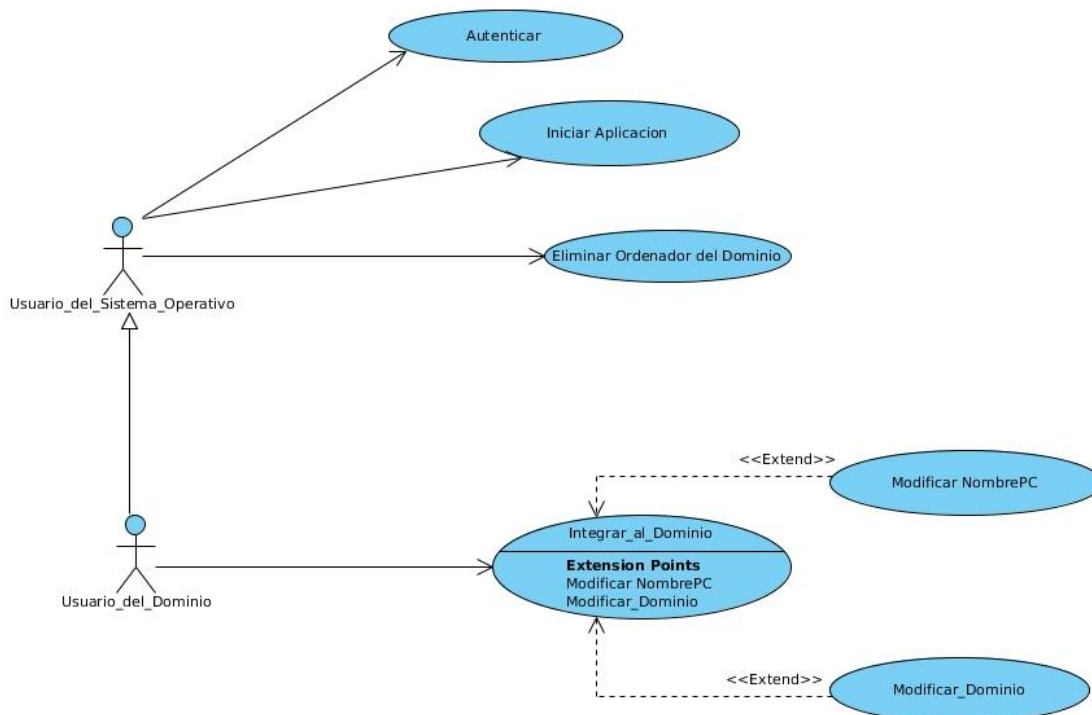


Figura 9 Diagrama de casos de uso del sistema

2.3.3 Descripción de los Casos de Uso del Sistema

| | | |
|--|--|--|
| Caso de Uso: | Modificar Dominio | |
| Actores: | Usuario_del_SO | |
| Resumen: | El caso de uso inicia cuando el Actor selecciona la opción “Modificar Dominio”, luego modifica el dominio y concluye el caso de uso. | |
| Precondiciones: | El usuario debe haber iniciado la aplicación | |
| Referencias | RF_5 | |
| Prioridad | Alto | |
| Flujo Normal de Eventos | | |
| Acción del Actor | Respuesta del Sistema | |
| 1. Selecciona la opción “Modificar Dominio”. | 2. Activa el campo de texto “dominio”. | |
| 3. Modifica el dominio. | 4. Verifica que el dato entrado se corresponda con un dominio. Termina el caso de uso. | |
| Prototipo de Interfaz | | |
| | | |
| Flujo Alternativo | | |
| | 4.1 Si el actor entra un valor que no se corresponda con el de un dominio, el sistema muestra | |

| | |
|------------------------|--|
| | un mensaje diciendo “Esto no es un dominio” y el campo de texto se pone en rojo. Ir a la acción 3 del actor. |
| Postcondiciones | Queda modificado el dominio. |

Tabla 5 Descripción del CU Modificar dominio

| | | |
|---|--|--|
| Caso de Uso: | Modificar Nombre PC | |
| Actores: | Usuario_del_SO | |
| Resumen: | El caso de uso inicia cuando el Actor selecciona la opción “Modificar Nombre Ordenador”. El actor modifica el nombre del ordenador y así concluye este caso de uso. | |
| Precondiciones: | El usuario de haber iniciado la aplicación | |
| Referencias | RF_6 | |
| Prioridad | Alto | |
| Flujo Normal de Eventos | | |
| Acción del Actor | Respuesta del Sistema | |
| 1. Selecciona la opción “Modificar Nombre Ordenador”. | 2. Cambia la propiedad del campo de texto a editable. | |
| 3. Modifica el nombre del ordenador. | 4. Verifica que el dato entrado no se corresponda con alguno que esté siendo usado por otro ordenador en la red. Termina el caso de uso. | |
| Prototipo de Interfaz | | |
| | | |
| Flujo Alternativo | | |
| | 4.1 Si el valor ya está siendo usado el sistema muestra un mensaje diciendo “Este nombre ya está siendo usado por otro ordenador” y pintará el campo de texto en rojo. Ir a la acción 3 del usuario. | |
| Postcondiciones | Queda modificado el nombre del ordenador. | |

Tabla 6 Descripción del CU Modificar nombre PC

| | | |
|---|---|--|
| Caso de Uso: | Eliminar Ordenador del Dominio. | |
| Actores: | Usuario del SO | |
| Resumen: | El caso de uso inicia cuando el Actor selecciona la opción “Eliminar Ordenador del Dominio”, el sistema realiza las configuraciones necesarias y de esta forma concluye el caso de uso. | |
| Precondiciones: | El usuario debe haber iniciado la aplicación. | |
| Referencias | RF_12, RF_13 | |
| Prioridad | Crítico | |
| Flujo Normal de Eventos | | |
| Acción del Actor | Respuesta del Sistema | |
| 1. Selecciona la opción “Eliminar Ordenador del Dominio”. | 2. Realiza las configuraciones necesarias para eliminar el ordenador del dominio. 3. Muestra una ventana diciendo al usuario “La operación concluyó con éxito, ahora debe Reiniciar el Sistema Operativo, para que surtan efecto los cambios realizados”, la cual posee la opción “Reiniciar”. Termina el caso de uso. | |
| Prototipo de Interfaz | | |
| | | |
| Flujo Alternativo | | |
| | | |
| Postcondiciones | Queda eliminado el ordenador del dominio. | |

Tabla 7 Descripción del CU Eliminar ordenador del dominio

| | |
|------------------------|--|
| Caso de Uso: | Iniciar Aplicación. |
| Actores: | Usuario del SO. |
| Resumen: | El caso de uso inicia cuando el Actor selecciona la aplicación en el menú de aplicaciones, esta pide la credencial de administrador del sistema operativo, busca información y muestra la interfaz con los datos, luego concluye el caso de uso. |
| Precondiciones: | El usuario debe haber iniciado sesión en el Sistema Operativo. |
| Referencias | RF_2, RF_3, RF_4. |

| | |
|---|---|
| Prioridad | Crítico |
| Flujo Normal de Eventos | |
| Acción del Actor | Respuesta del Sistema |
| 1. Selecciona la aplicación en el menú de aplicaciones. | 2. Verifica si existe conexión de red. 3. Determina si el ordenador está integrado al Directorio Activo o no. 4. Busca el nombre del dominio. 5. Busca el nombre del ordenador. Termina el caso de uso. |
| Prototipo de Interfaz | |
| Flujo Alternativo | |
| | 2.1 Si el ordenador no tiene conexión de red, se muestra el mensaje "Para iniciar la aplicación es necesario que su ordenador tenga conexión de red". |
| | 4.1 Si el archivo de configuración "resolv.conf" no se encuentra, se muestra una ventana con el mensaje: "El destino /etc/resolv.conf no se encontró". |
| | 5.1 Si el archivo de configuración "hostname" no se encuentra, se muestra una ventana con el mensaje: "El destino /etc/hosts no se encontró". |
| Postcondiciones | Se muestra la Interfaz de Usuario con los Datos. |

Tabla 8 Descripción del CU Iniciar aplicación

| | |
|------------------------|--|
| Caso de Uso: | Integrar al Directorio Activo |
| Actores: | Usuario del Dominio. |
| Resumen: | El caso de uso inicia cuando el Actor selecciona la opción "Integrar al Dominio". La aplicación realiza las configuraciones necesarias para integrar el ordenador al Directorio Activo, de esta forma concluye este caso de uso. |
| Precondiciones: | Debe haber iniciado la aplicación. |
| Referencias | RF_7, RF_8, RF_9, RF_10, RF_11, RF_13 |
| Prioridad | Crítico |

| Flujo Normal de Eventos | |
|--|---|
| Acción del Actor | Respuesta del Sistema |
| 1. Selecciona la opción "Integrar al Dominio". | 2. Verifica que los campos (usuario, contraseña y Directorio Activo) estén llenos. 3. Válida los datos entrados por el actor. 4. El sistema verifica que el ordenador tenga conexión de red. 5. El sistema configura las librerías del sistema que permiten integrar el ordenador al Directorio Activo. 6. El sistema realiza las configuraciones en el sistema operativo para establecer las políticas de seguridad establecidas por la OSRI. 7. El sistema sincroniza el ordenador con el Directorio Activo. 8. El sistema muestra una ventana de aviso diciendo "La Integración se realizó exitosamente, ahora debe Reiniciar para que surtan efecto los cambios realizados", que incluye la opción "Reiniciar". Termina el caso de uso. |
| Prototipo de Interfaz | |
| | |
| Flujo Alternativo | |
| | 4.1 Si el ordenador no tiene conexión, el sistema lanza un mensaje de alerta diciendo "La conexión de red esta inhabilitada". Ir a la acción 1 del usuario. |
| Postcondiciones | Queda integrado el ordenador al dominio. |

Tabla 9 Descripción del CU Integrar al DA

| | |
|---------------------|---|
| Caso de Uso: | Autenticar |
| Actores: | Usuario del SO. |
| Resumen: | El caso de uso inicia cuando el Actor selecciona la aplicación en el menú de aplicaciones, introduce la credencial de administrador del SO y de esta forma concluye este caso de uso. |

| | | |
|--|---|--|
| Precondiciones: | El usuario debe haber iniciado sesión en el SO. | |
| Referencias | RF_1 | |
| Prioridad | Crítico | |
| Flujo Normal de Eventos | | |
| Acción del Actor | Respuesta del Sistema | |
| 1. Selecciona la aplicación en el menú de aplicaciones. 3. Introduce la contraseña. | 2. Muestra una pantalla para introducir la contraseña. 4. Verifica que sea correcta la contraseña entrada. Termina el caso de uso. | |
| Prototipo de Interfaz | | |
| | | |
| Flujo Alternativo | | |
| | 4.1 Si la contraseña no es válida, muestra el mensaje "Contraseña Incorrecta". Ir a la acción 2 del sistema (la tercera vez que ocurra esto se cierra automáticamente la aplicación). | |
| Postcondiciones | Queda autenticado como administrador del SO. | |

Tabla 10 Descripción del CU Autenticar

2.3.4 Diagramas de clases

Un diagrama de clases sirve para visualizar las relaciones entre las clases que involucran el sistema, las cuales pueden ser asociativas, de herencia, de uso y de agregación, ya que una clase es una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, métodos, relaciones y semántica; mostrando un conjunto de elementos que son estáticos, como las clases y tipos junto con sus contenidos y relaciones. Un diagrama de clases está compuesto por los siguientes elementos: clases (atributos, métodos y visibilidad) y relaciones (herencia, composición, agregación, asociación y uso).

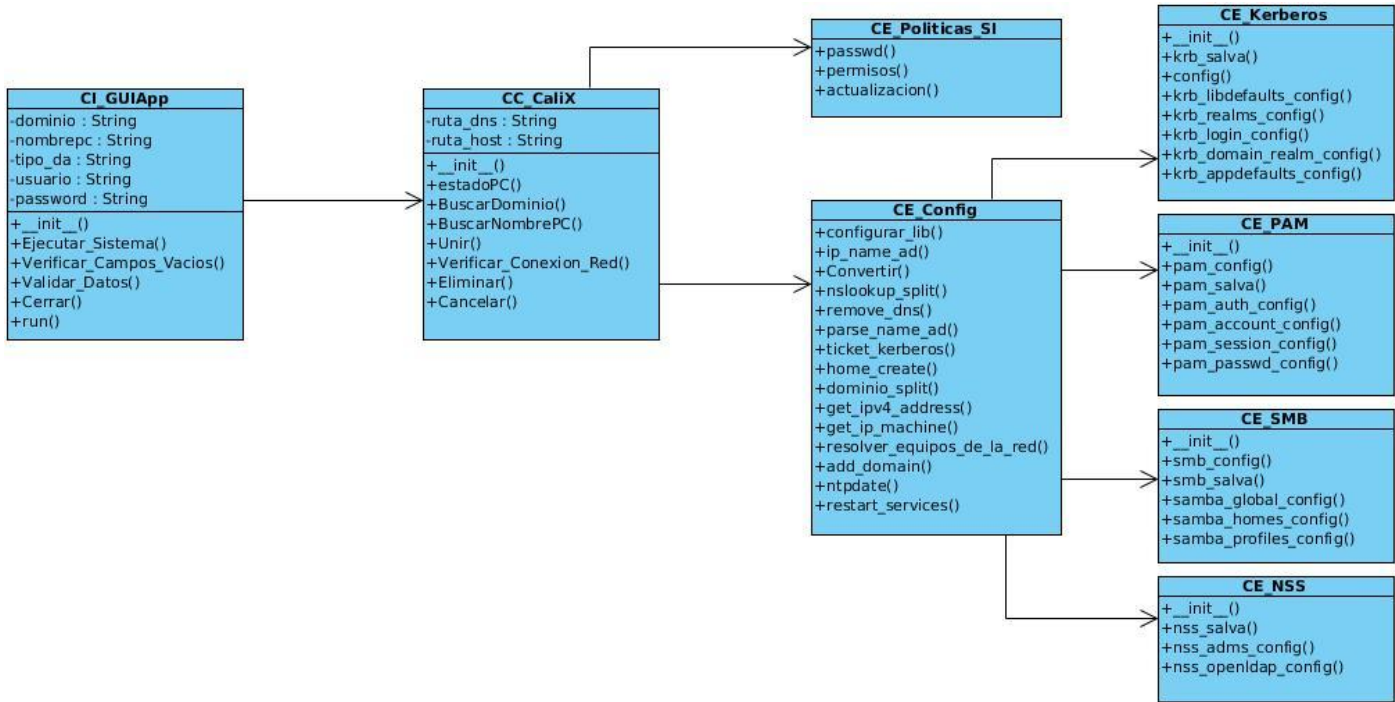


Figura 10 Diagrama de clases del diseño

2.4 Conclusiones del Capítulo

Haciendo uso de la metodología ágil Open Up, para lograr flexibilidad y rapidez, se llevó a cabo el proceso ingenieril de la herramienta CaliX, lo que permitió cumplir con el objetivo específico propuesto para el presente capítulo. Este proceso estuvo compuesto por la conceptualización del funcionamiento y la especificación de los requisitos de software de la solución que se propone, para permitir una mejor comprensión de lo que se desea lograr con el desarrollo de esta herramienta; se establecen los casos de uso con sus respectivos diagramas, para lograr un mayor entendimiento del sistema y facilitar la posterior implementación del código.

Capítulo 3: Implementación y Realización de Pruebas

En el capítulo anterior se realizó la propuesta del diseño ingenieril para la solución a implementar. En función del mismo, en el presente capítulo se lleva a cabo la implementación. Además, se definen las pruebas que se le van a realizar al sistema para verificar que las funcionalidades estén correctamente implementadas, con el fin de asegurar la calidad de la misma y poder entregar al cliente un mejor producto.

3.1 Implementación

El proceso de implementación consiste en desarrollar y organizar los componentes basándose en las especificaciones del diseño (Petitpierre, 2006). Se lleva a cabo mediante la realización de las especificaciones técnicas o algoritmos como un programa, componente de software, u otro sistema de cómputo (McConnell, 2004). Dentro de los principales objetivos que se desea alcanzar en una etapa de implementación se encuentran:

- ✓ Definir la organización del código, en términos de los subsistemas de implementación organizados en capas.
- ✓ Implementar los elementos de diseño en términos de los elementos de implementación (archivos de origen, binarios, programas ejecutables y otros).
- ✓ Probar y desarrollar componentes como unidades.
- ✓ Integrar los resultados producidos por los implementadores individuales, o equipos en un sistema ejecutable.

3.1.1 Vista de Despliegue

En un diagrama de despliegue se muestra la disposición física de los distintos nodos que componen un sistema y el reparto de los componentes sobre ellos. Los nodos no son más que elementos físicos que existen en tiempo de ejecución y representan un recurso computacional que generalmente poseen una asignación de memoria y capacidad de procesamiento. El diagrama de despliegue servirá para modelar la topología de hardware sobre la cual se ejecutará el sistema (Petitpierre, 2006).

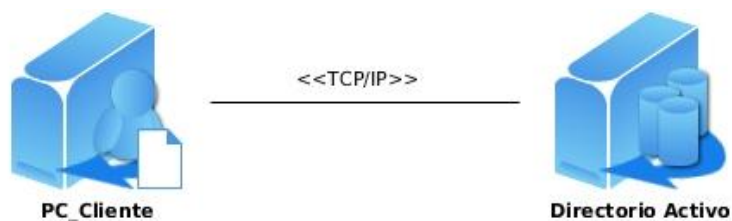


Figura 11 Diagrama de despliegue

3.1.2 Diagrama de Componentes

Un diagrama de componentes es un diagrama desarrollado en el lenguaje unificado de modelado (UML). Representa cómo un sistema de software es dividido en componentes y muestra las dependencias entre estos.

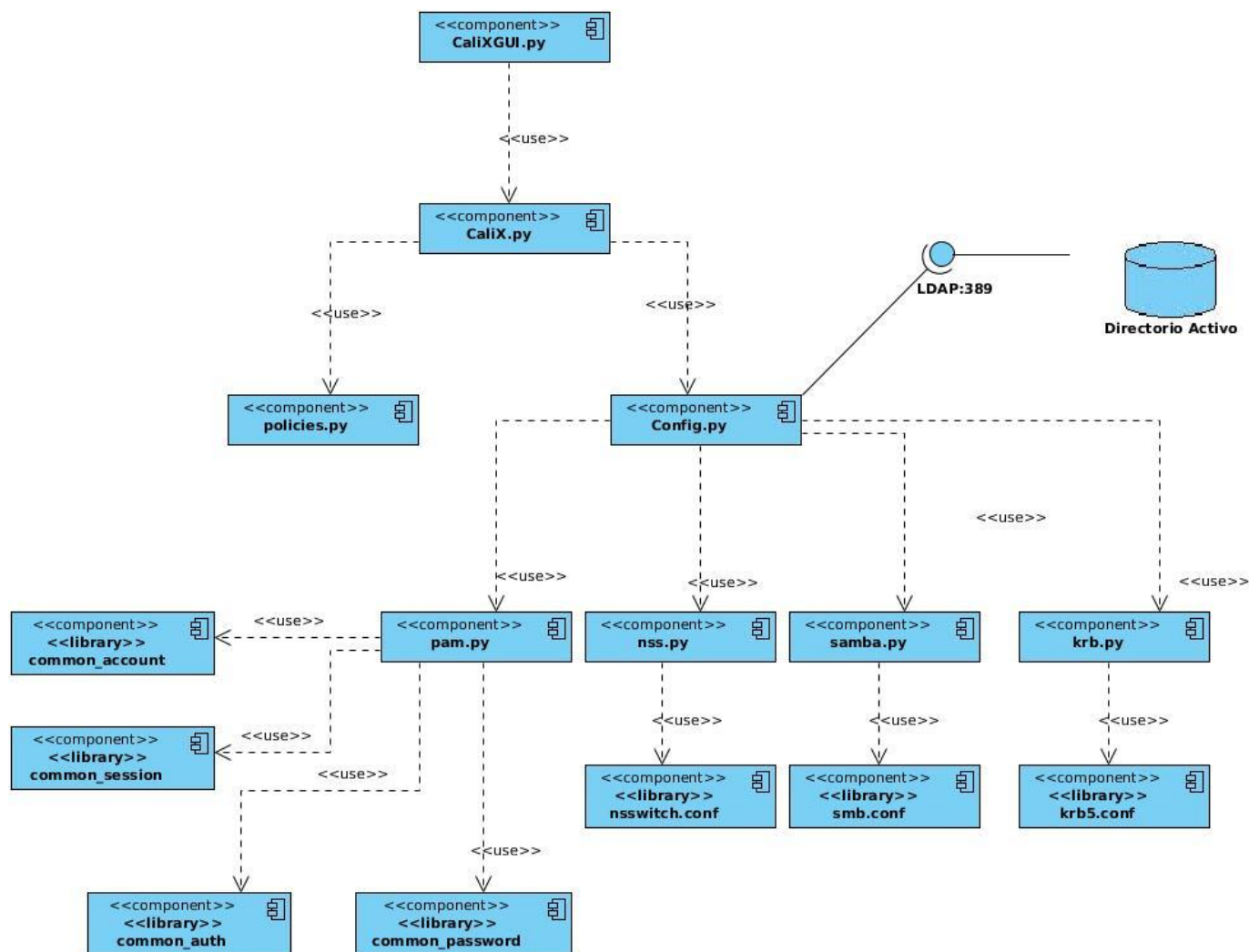


Figura 12 Diagrama de componentes

Normalmente los diagramas de componentes contienen:

- ✓ **Componentes:** son una parte física de un sistema (módulo, base de datos, programa ejecutable, etc.). Se puede decir que un componente es la materialización de una o más clases, debido a que una abstracción con atributos y métodos puede ser implementada en los componentes. Estos se pueden agrupar en paquetes, y pueden existir entre ellos relaciones de dependencia (Generalización, Asociación, Agregación, Realización). (McConnell, 2004)
- ✓ **Interfaces:** es el lazo de unión entre varios componentes.
- ✓ **Paquetes o subsistemas.**

3.2 Pruebas del Sistema

El interés por la calidad crece de forma continua, a medida que los clientes se vuelven más selectivos y comienzan a rechazar los productos poco fiables o que realmente no dan respuesta a sus necesidades. Es por esto que es necesario realizarle pruebas al sistema a medida que se implementan las funcionalidades.

Las pruebas de software son un proceso centrado en el objetivo de encontrar defectos a un software; puede ser por razones de depuración al software o de aceptación del mismo. Tratar de encontrar defectos es una parte esencial en todo proceso de pruebas, para ello, los probadores evalúan el producto desde un punto de vista crítico, sometiéndolo a una serie de acciones inquisitivas y el producto responde con su comportamiento como reacción (J. Myers, et al., 2011).

Objetivos de las pruebas de software:

- ✓ Encontrar y documentar los defectos que puedan afectar la calidad del software.
- ✓ Validar que el software trabaje como fue diseñado.
- ✓ Probar y validar los requisitos que debe cumplir el software.
- ✓ Validar que los requisitos fueron implementados correctamente.

3.2.1 Pruebas de Aceptación

Es el cliente quien lleva a cabo el testeo de la aplicación para determinar si el sistema cumple con los objetivos deseados. Se determina la conformidad del cliente antes de que el programa le sea entregado como una versión final. El usuario comprueba en su propio entorno de explotación si acepta el software como está o si precisa aplicar nuevas optimizaciones y soluciones de fallas. Las pruebas de aceptación se realizan en el entorno en que se va a explotar el sistema (incluido el personal que lo maneja).

Estas pruebas tienen vital importancia, pues los resultados proyectados deciden la aceptación o el rechazo del producto. Las pruebas de aceptación surgen por la necesidad de productos software con calidad, capaces de cumplir con las expectativas y los requisitos de los clientes finales. Para la generación de los casos de prueba de las pruebas de aceptación funcionales se utilizan las pruebas de caja negra. (J. Myers, y otros, 2011)

3.2.2 Pruebas de Caja Negra

Es denominada prueba de comportamiento, ya que se centra en los requisitos funcionales del software. Permite obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa y está referida a las pruebas que se llevan a cabo sobre la interfaz del software, examinando algunos aspectos del modelo fundamental del sistema sin tener mucho en cuenta la estructura lógica interna del software (código fuente de la aplicación). Para llevar a cabo las mismas se diseñan casos de prueba para estudiar las salidas del sistema y observar si concuerdan con la salida esperada (Pressman, 2011).

3.2.3 Casos de prueba

Un caso de prueba se diseña según las funcionalidades descritas en los casos de uso. El propósito que se persigue con este artefacto es lograr una comprensión común de las condiciones específicas que la solución debe cumplir. El punto de partida del proceso de pruebas lo constituye la descripción de los casos de usos del sistema, como apoyo para las revisiones. Cada planilla de caso de prueba recoge la especificación de un caso de uso, dividido en secciones y escenarios, detallando las funcionalidades descritas en él y describiendo cada una de las variables que recoge. Además quedan plasmadas las revisiones realizadas al caso de prueba; así como un registro de todo aquello que no corresponde a la calidad del software.

Partiendo de la descripción de los casos de usos del sistema, como apoyo para las revisiones, se diseñó un caso de prueba asociado a cada caso de uso. Se efectuaron los casos de prueba a 6 casos de uso, plasmándose en la documentación del proyecto.

Casos de Prueba del Sistema

| Nombre de la sección | Escenarios de la sección | Descripción de la funcionalidad | Flujo Central |
|------------------------------|--------------------------------------|--|--|
| SC1 Modificar Dominio | EC 1.1: Modificar Dominio con éxito. | Permitirá modificar el dominio mostrado por la aplicación. | <ul style="list-style-type: none"> ✓ Clic en la opción "Modificar Dominio" ✓ Se activa el campo de texto, el usuario modifi- |

| | | | |
|--|---|--|--|
| | | | ca el dominio y se muestra el campo en color verde. |
| | EC 1.2: Modificar el Dominio sin éxito. | Se intentará modificar el dominio mostrado por la aplicación con datos no válidos. | <ul style="list-style-type: none"> ✓ Clic en la opción "Modificar Dominio" ✓ Se activa el campo de texto, el usuario modifica el dominio y se muestra el campo en color rojo y un mensaje diciendo "Esto no es un Dominio" |

Tabla 11 Sección Modificar dominio

| Nombre de la sección | Escenarios de la sección | Descripción de la funcionalidad | Flujo Central |
|---------------------------------|--|---|---|
| SC2 Modificar Nombre PC. | EC 2.1: Modificar Nombre PC con éxito. | Permitirá modificar nombre del ordenador mostrado por la aplicación. | <ul style="list-style-type: none"> ✓ Clic en la opción "Modificar Nombre Ordenador". ✓ Se activa el campo de texto, el usuario modifica el nombre del ordenador y se muestra el campo en color verde. |
| | EC 2.2: Modificar Nombre PC sin éxito. | Se intentará modificar el nombre del ordenador mostrado por la aplicación con datos no válidos. | <ul style="list-style-type: none"> ✓ Clic en la opción "Modificar Nombre Ordenador". ✓ Se activa el campo de texto, el usuario modifica el nombre del orde- |

| | | | |
|--|--|--|--|
| | | | nador, se muestra el campo en color rojo y un mensaje diciendo “Ha introducido un nombre no válido”. |
| | EC 2.3: Modificar Nombre PC sin éxito. | Se intentará modificar el nombre del ordenador mostrado por la aplicación con un nombre válido, y este ya se encuentra en uso. | <ul style="list-style-type: none"> ✓ Clic en la opción “Modificar Nombre Ordenador” ✓ Se activa el campo de texto, el usuario modifica el nombre del ordenador, se muestra el campo en color rojo y un mensaje diciendo “El nombre introducido ya está siendo usado por otro ordenador en la red”. |

Tabla 12 Sección Modificar nombre PC

| Nombre de la sección | Escenarios de la sección | Descripción de la funcionalidad | Flujo Central |
|---|---|--|--|
| SC3 Eliminar Ordenador del Dominio | EC 3.1: Eliminar ordenador del dominio con éxito. | Permitirá eliminar el ordenador del dominio. | <ul style="list-style-type: none"> ✓ Clic en la opción “Eliminar Ordenador del Dominio”. ✓ Se muestra una ventana con una barra de progreso, cuando esta llega al 100% se muestra un mensaje diciendo: “Se ha eliminado el ordena- |

| | | | |
|--|---|--|--|
| | | | dor del Dominio, debe Reiniciar para que surtan efecto los cambios realizados”. Se activa el botón Reiniciar. |
| | EC 3.2: Eliminar ordenador del dominio sin éxito. | Se intentará eliminar el ordenador del dominio, pero las salvas han sido movidas o borradas. | <ul style="list-style-type: none"> ✓ Clic en la opción “Eliminar ordenador del dominio”. ✓ Se muestra una ventana con una barra de progreso, el cual se detiene y se muestra un mensaje diciendo: “El proceso ha sido interrumpido, porque no se encuentran las salvas”. |

Tabla 13 Sección Eliminar ordenador del dominio

| Nombre de la sección | Escenarios de la sección | Descripción de la funcionalidad | Flujo Central |
|--------------------------------|---------------------------------------|----------------------------------|---|
| SC4 Iniciar Aplicación. | EC 4.1: Iniciar Aplicación con éxito. | Permitirá iniciar la aplicación. | <ul style="list-style-type: none"> ✓ Se determina el estado de la integración, se localiza el nombre del ordenador y el nombre del dominio correctamente y se muestran en la interfaz. |

| | | | |
|--|---------------------------------------|---|---|
| | EC 4.2: Iniciar aplicación sin éxito. | Se intentará iniciar la aplicación pero no hay conexión de red. | ✓ Se muestra una ventana de alerta con el mensaje: "Para iniciar la aplicación es necesario que su ordenador tenga conexión de red". |
| | EC 4.3: Iniciar aplicación sin éxito. | Se intentará iniciar la aplicación pero no se encuentra disponible la ruta: "/etc/resolv.conf". | ✓ Se muestra una ventana de alerta con el mensaje: "No se pudo iniciar la aplicación porque la ruta "/etc/resolv.conf" no está disponible". |
| | EC 4.4: Iniciar aplicación sin éxito. | Se intentará iniciar la aplicación pero no se encuentra disponible la ruta: "/etc/hostname". | ✓ Se muestra una ventana de alerta con el mensaje: "No se pudo iniciar la aplicación porque la ruta "/etc/hostname" no está disponible". |

Tabla 14 Sección Iniciar aplicación

| Nombre de la sección | Escenarios de la sección | Descripción de la funcionalidad | Flujo Central |
|-----------------------|-------------------------------|---|--|
| SC5 Autenticar | EC 5.1: Autenticar con éxito. | Permitirá validar que el usuario del SO posea privilegios de administrador. | <ul style="list-style-type: none"> ✓ El usuario del SO selecciona la aplicación en el menú de aplicaciones. ✓ Se muestra una venta con un mensaje diciendo: "Se necesitan privilegios administrativos para ejecutar la aplicación" y un campo para |

| | | | |
|--|--------------------------------------|---|---|
| | | | <p>introducir la clave requerida.</p> <ul style="list-style-type: none"> ✓ El usuario introduce la clave de administrador del SO. |
| | <p>EC 5.2: Autenticar sin éxito.</p> | <p>Se intentará validar al usuario del SO como administrador, con una clave incorrecta.</p> | <ul style="list-style-type: none"> ✓ El usuario del SO selecciona la aplicación en el menú de aplicaciones. ✓ Se muestra una ventana con el mensaje: “Se necesitan privilegios administrativos para ejecutar la aplicación” y un campo para introducir la clave requerida. ✓ El usuario introduce una clave incorrecta, se muestra el mensaje: “La clave es incorrecta, inténtelo de nuevo”. |

Tabla 15 Sección Autenticar

| Nombre de la sección | Escenarios de la sección | Descripción de la funcionalidad | Flujo Central |
|---|---|---|--|
| <p>SC6 Integrar al Directorio Activo</p> | <p>EC 6.1: Validar nombre de usuario del dominio con éxito.</p> | <p>Permitirá validar que el dato entrado se corresponda con las características establecidas para un nombre de usuario.</p> | <ul style="list-style-type: none"> ✓ El usuario del dominio introduce su nombre de usuario. ✓ Se muestra el campo de texto en color verde. |

| | | | |
|--|---|--|--|
| | EC 6.2: Validar nombre de usuario del dominio sin éxito. | Se intentará validar el nombre de usuario del dominio, introduciendo un valor incorrecto. | <ul style="list-style-type: none"> ✓ El usuario del dominio introduce su nombre de usuario de forma incorrecta. ✓ Se muestra el campo de texto en rojo y el mensaje: “El valor introducido no cumple con las características de un usuario”. |
| | EC 6.3: Validar contraseña del usuario del dominio con éxito. | Permitirá validar que la contraseña del usuario del dominio cumpla las características requeridas. | <ul style="list-style-type: none"> ✓ El usuario del dominio introduce su contraseña. ✓ Se muestra el campo de texto en color verde. |
| | EC 6.4: Validar contraseña del usuario del dominio sin éxito. | Se intentará validar la contraseña del usuario del dominio, con un tamaño menor que 8 caracteres. | <ul style="list-style-type: none"> ✓ El usuario del dominio introduce la contraseña. ✓ Se muestra el campo de texto en rojo con el mensaje: “La contraseña introducida es demasiado corta”. |
| | EC 6.5: Validar campos vacíos con éxito. | Permitirá validar que los campos no estén vacíos. | <ul style="list-style-type: none"> ✓ Se activa el botón con la opción “Integrar al Directorio Activo”. |
| | EC 6.6: Validar campos vacíos sin éxito. | Se intentará dejar uno o varios de los campos vacíos. | <ul style="list-style-type: none"> ✓ Se muestra el mensaje en la ventana: “Todos los campos deben ser llenados”. |

| | | | |
|--|---|--|--|
| | <p>EC 6.7: Realizar proceso de Integración con éxito.</p> | <p>Permitirá realizar el proceso de integración al Directorio Activo.</p> | <p>✓ El usuario selecciona la opción: “Integrar al Directorio Activo”.</p> <p>✓ Se muestra una ventana con una barra de progreso, cuando esta llega al 100% se muestra el mensaje: “La Integración fue exitosa, debe Reiniciar para que surtan efecto los cambios realizados”. Se activa el botón Reiniciar.</p> |
| | <p>EC 6.8: Realizar proceso de Integración sin éxito.</p> | <p>Se intentará integrar al directorio activo pero no hay conexión de red.</p> | <p>✓ Se muestra una ventana de alerta con el mensaje: “Para integrar al Directorio Activo es necesario que su ordenador tenga conexión de red”.</p> |

Tabla 16 Sección Integrar al DA

3.3 Resultados obtenidos

En este Epígrafe se relacionan los resultados obtenidos hasta el momento por el equipo de desarrollo de **CaliX**. Es necesario resaltar que como resultado de este trabajo la herramienta **CaliX** se encuentra disponible en su versión 1.0, lo cual también significa que se pueden esperar mayores resultados en versiones superiores.

3.3.1 Sobre las funcionalidades obtenidas

Entre las principales funcionalidades que posee la herramienta **CaliX** se pueden mencionar:

- ✓ Configura correctamente las librerías que permiten la integración.

- ✓ Configura adecuadamente las políticas de seguridad en el ordenador cliente.
- ✓ Realiza eficazmente la integración del ordenador con los Directorios Activos.
- ✓ Permite la autenticación sin conexión.

3.3.2 Resumen de las pruebas

Para la realización de las pruebas se hizo uso de los casos de prueba y se tuvo en cuenta el criterio del cliente, logrando gran aceptación por parte del mismo, pues la aplicación cumple el objetivo general por el que fue creada: lograr la integración de los ordenadores con Nova al DA. En el proceso de las pruebas estuvieron involucrados dos estudiantes del centro, una secretaria de la facultad, un profesor del proyecto, arrojando como resultado en una escala de 1 a 5, un nivel 5 de aceptación, lo cual se demuestra gráficamente en la figura que se muestra a continuación.



Figura 13 Resultados de las pruebas de aceptación

Durante el proceso de depuración se obtuvieron 5 errores, entre los cuales cabe mencionar:

- ✓ Al validar los datos entrados por el usuario, siempre tomaba como valor del tipo de DA seleccionado el primero en la lista.
- ✓ La función que se implementó para devolver la dirección IP y el nombre del DA, no llenaba la lista con los datos, pues la expresión regular para determinarlos no era correcta.

Al finalizar el proceso de pruebas, se corrigieron todos los errores encontrados durante 4 iteraciones de software y se alcanzó la versión 1.0 del producto.

3.4 Conclusiones del capítulo

Luego de realizar el diseño de los casos de prueba basados en casos de uso y haberlos utilizado para verificar el funcionamiento de la aplicación, se puede llegar a la conclusión de que la misma se encuentra correctamente implementada, cumpliendo así con los objetivos planteados. Este proceso, además de validar que las funcionalidades se realizaron de forma correcta, permite asegurar la calidad del producto a la hora de entregarlo al cliente.

Conclusiones Generales

Una vez concluida la investigación, se arriban a las siguientes conclusiones:

- ✓ La caracterización del entorno informático de las empresas cubanas y el estudio del proceso y de las tecnologías utilizados en la implementación de herramientas para la integración de SO a DA, permitió obtener el conocimiento necesario para determinar qué solución tomar.
- ✓ Los estudios realizados contribuyeron además a realizar el diseño de la solución propuesta, eligiendo herramientas y tecnologías que respondieran a las necesidades del diseño e implementación del sistema, facilitando la elaboración de la documentación necesaria para extender el alcance del sistema en versiones posteriores.
- ✓ Luego de completar la implementación y de haber realizado un estudio del diseño de interfaces gráficas en la actualidad, la aplicación resultante ofrece una interfaz sencilla y agradable al usuario no experimentado, lo que también mejora el rendimiento de la misma.
- ✓ Realizar el proceso de pruebas a la solución, permitió demostrar que el sistema realiza una correcta integración del ordenador con los Directorios Activos y favorece una mayor agilidad en los resultados, reduciendo de esta forma el tiempo de respuesta del mismo.

Recomendaciones

A los Administradores de Red: en caso de utilizar Microsoft Active Directory como Servicio de Directorio, se recomienda migrar el mismo a tecnologías de código abierto, utilizando para ello alguno de los Servicios de Directorio expuestos en este trabajo como alternativa a Microsoft Active Directory, preferiblemente hacer uso del 389 Directory Server.

Al Equipo de desarrollo de la distribución cubana de GNU/Linux: se recomienda modificar el manejador de sesión (GDM, LightDM, otros), para que los usuarios puedan desde aquí cambiar su contraseña del dominio, haciendo uso de las librerías PAM, previamente configuradas por la solución obtenida en este trabajo.

Desarrollar en próximas versiones del producto, la siguiente funcionalidad:

- ✓ Módulo para la integración de las demás aplicaciones del SO mediante D-BUS.

Referencias bibliográficas

1. **Alarcón, Raúl. 2000.***Diseño orientado a objetos con UML*. s.l.: Grupo Eidos, 2000. Vol. 1. 84-88457-03-0.
2. **De la Rosa Pasteur, Jenny y Zulueta Saria, Edgardo. 2010.***Plataforma de Gestión de Servicios Telemáticos en GNU/Linux. Módulo de Directorio v2.0*. La Habana: Universidad de las Ciencias Informáticas, 2010.
3. **Dell, Inc. 2013.** IT Management Solutions. [En línea] Dell, Inc., 2013. <http://www.quest.com/>.
4. **Eclipse Foundation. 2013.**Open UP for Eclipse. [Online] Eclipse Foundation, 2013. <http://epf.eclipse.org/wikis/openup/index.htm>.
5. **Fedora. 2013.** Fedora Project. [Online] 2013. <http://www.fedoraproject.org>.
6. **freedesktop.org. 2013.** FreeDesktop.org. [En línea] 17 de Mayo de 2013. <http://www.freedesktop.org/wiki/>.
7. **García Chico, Joanna. 2011.***Estudio de viabilidad de Directorio Activo en Linux*. s.l.: Universidad Carlos III, 2011.
8. **H. Bahr, Daniel. 2010.***Sistema para la generación de paquetes binarios fragmentados a partir del código fuente*. Mayo: Universidad de las Ciencias Informáticas, 2010. págs. 13-23.
9. **Kroll, Per y Maclsaac, Bruce. 2006.***Agility and Discipline Made Easy: Practices from OpenUp and RUP*. Online. s.l. : ADDISON WESLEY Publishing Company Incorporated, 2006. pág. 417. 0-321-32130-8.
10. **OpenLDAP Foundation. 2013.** OpenLDAP. [Online] OpenLDAP Foundation, 2013. <http://www.openldap.org/>.
11. **Resolución 127 Sobre el Reglamento de Seguridad Informática. Ministro de la Informática y las Comunicaciones. 2007.** La Habana: Ministerio de la Informática y las Comunicaciones, 2007, Gaceta Oficial de la República de Cuba, Vol. 1, pág. 10.
12. **Pressman, Roger S. 2011.***Software Engineering, a practitioner's approach*. [ed.] McGraw-Hill. 7th. 2011. p. 895. 9780071267823.
13. **Rumbaugh, I.J.G.B.J. 2000.***El Proceso Unificado de Desarrollo de Software*. 3ra. s.l.: Addison Wesley, 2000. Págs. 125-202.
14. **SPI et al. 2013.** Debian Official Website. [Online] Software in the Public Interest, Inc., Mayo 5, 2013. <http://www.debian.org/>.
15. **The Internet Society. 1997.***RFC2251*. 1997.
16. **ROUYOY, Romain; CONAN, Denis; SEINTURIER, Lionel.** Software architecture patterns for a context-processing middleware framework. Distributed Systems Online, IEEE, 2008, vol. 9, no 6, p. 1-1.
17. **KIRCHER, Michael; JAIN, Prashant.** Pattern-oriented software architecture, Patterns for resource management. Wiley, 2005.

- 18. Petersen, Richard y Haddad, Ibrahim.** *Red Hat Linux Pocket Administrator*. [ed.] Brandon A. Nordin. 2004. pág. 289. 0-07-222974-8.
- 19. Timothy A. Howes, Ph.D., Smith, Mark C. and Good, Gordon S.** *Understanding and Deploying LDAP Directory Services*. 2nd. Michigan : Addison Wesley, 2003. p. 936. 9780672323164.
- 20. Seitz, Justin.** *Gray Hat Python*. 2009. 978-1-59327-224-1, 1-59327-224-3.
- 21. Petitpierre, Claude.** *Software Engineering: The Implementation Phase*. [ed.] EPFL Press. ilustrada. 2006. p. 349. 0849391725, 9780849391729.
- 22. O'Reilly, Mike.** developerWorks. [Online] 2013. <http://www.ibm.com/developerworks/library/l-openldap/>.
- 23. Lutz, Mark.** *Learning Python*. [ed.] O'Reilly Media Inc. 5th. 2013. pág. 1600. 978-1-4493-5568-5, 1-4493-5568-4.
- 24. Kouti, Sakari y Seitsonen, Mika .** *Inside Active Directory*. [ed.] Addison-Wesley. 2004. pág. 1248. 0321228480.
- 25. Jackiewicz, Tom.** *Deploying OpenLDAP*. [ed.] Apress. 2005. s.l. : O'Reilly, 2004. p. 344. 978-1590594131.
- 26. Donnelly, Michael.** ldapman.org. [Online] Abril 28, 2000. http://ldapman.org/articles/intro_to_ldap.html.
- 27. Desmond, Brian, y otros.** *Active Directory: Designing, Deploying, and Running Active Directory*. 4th. s.l. : O'Reilly Media Inc., 2008. pág. 866. 0596554281, 9780596554286.
- 28. Desmond, Brian, y otros.** *Active Directory*. 5th. s.l. : O'Reilly Media, 2013. p. 738. 978-1-4493-2000-3, 1-4493-2000-7.
- 29. Carter, Gerald.** *LDAP system administration*. [ed.] O'Reilly. ilustrada. 2003. p. 294. 1565924916, 9781565924918.
- 30. Beazley, David y K. Jones, Brian.** *Python Cookbook*. [ed.] O'Reilly Media Inc. 3rd. 2013. 978-1-4493-4036-0, 1-4493-4036-9.
- 31. Sommerville, Ian.** *Ingeniería del software 7*. [ed.] Pearson Educación. [trad.] María Isabel Alfonso Galipienso. 7ma, ilustrada. s.l. : Pearson Educación, 2005. pág. 687. 8478290745, 9788478290741.
- 32. J. Myers, Glenford, Sandler, Corey y Badgett, Tom .** *The Art of Software Testing*. [ed.] Wiley. 3rd. 2011. pág. 240. 1118031962, 978-1118031964.
- 33. Hunter, Laura E. and Allen, Robbie.** *Active Directory Cookbook*. [ed.] Inc. O'Reilly Media. 3. 2008. p. 1092. 0596554443, 9780596554446.
- 34. E. Wieggers, Karl.** *Software Requirements*. [ed.] Inc. O'Reilly Media. 2nd. 2009. pág. 544. 0735637083, 9780735637085.
- 35. MediaWiki.** Powerful Open Source LDAP. [En línea] 24 de Marzo de 2013. [Citado el: 16 de Abril de 2013.]
- 36. Centrify Corporation.** Centrify Express. [Online] Febrero 25, 2013. [Cited: Abril 15, 2013.] <http://www.centrify.com/express/free.asp>.

- 37.Dell Inc.**Authentication Services. [En línea] 16 de Abril de 2013. <http://www.quest.com/authentication-services/>.
- 38.Red Hat, Inc.** Deployment, configuration and administration of Fedora 14. [En línea] Red Hat, Inc. and others., 2011. http://docs.fedoraproject.org/en-US/Fedora/14/html/Deployment_Guide/index.html.
- 39.Likewise Software.***Likewise Open Installation And Administration Guide*. [ed.] Likewise Software. 2008. pág. 72.
- 40.Gamma, Erich, y otros.***Design Patterns: Elements of Reusable Object-Oriented Software*. 2005. pág. 432. 0-201-63361-2.
- 41.Pierra Fuentes, Allan .***Conceptualización y Reestructuración Estratégica de la Distribución Cubana de GNU/Linux “Nova”*. [ed.] UCI. La Habana : s.n., 2011. pág. 77.

Bibliografía

1. **Fernández del Monte, Yusleidy y Albo Castro, Mónica María. 2011.***Metodología Nova OpenUp para el desarrollo de distribuciones GNU/Linux.* La Habana: Universidad de las Ciencias Informáticas, 2011.
2. **De la Rosa Pasteur, Jenny y Zulueta Saria, Edgardo. 2010.***Plataforma de Gestión de Servicios Telemáticos en GNU/Linux. Módulo de Directorio v2.0.* La Habana: Universidad de las Ciencias Informáticas, 2010.
3. **García Chico, Joanna. 2011.***Estudio de viabilidad de Directorio Activo en Linux.* s.l.: Universidad Carlos III, 2011.
4. **Resolución 127 Sobre el Reglamento de Seguridad Informática. Ministro de la Informática y las Comunicaciones. 2007.** La Habana: Ministerio de la Informática y las Comunicaciones, 2007, Gaceta Oficial de la República de Cuba, Vol. 1, pág. 10.
5. **Lie Hetland, Magnus.***Beginning Python From Novice to Professional.* [ed.] Matt Moodie. s.l.: Springer-Verlag New York, Inc. 1-59059-519-X.
6. **Gift, Noab and M.Jones, Jeremy. 2008.***Python for Unix and Linux System.* [ed.] Julie Steele. Lorana Dimant. s.l.: nSight, Inc., 2008. 978-0-596-51582-9.
7. **Kroll, Per y Maclsaac, Bruce. 2006.***Agility and Discipline Made Easy: Practices from OpenUp and RUP.* Online. s.l.: ADDISON WESLEY Publishing Company Incorporated, 2006. pág. 417. 0-321-32130-8.
8. **Pressman, Roger S. 2011.***Software Engineering, a practitioner's approach.* [ed.] McGraw-Hill. 7th. 2011. p. 895. 9780071267823.
9. **Rossum, Guido van. 2005.***Guía de aprendizaje de Python.* [ed.] Jr. Fred L. Drake. 2005.
10. **The NetBSD Foundation. 2013.** The NetBSD Guide. [Online] The NetBSD Foundation, Inc., Enero 1, 2013. <http://www.netbsd.org/docs/guide/en/index.html>.
11. **Finlay, John, y otros. 2012.***Tutorial de PyGTK 2.0 versión 2.3.* 2012.
12. **Microsoft. 2013.**TechNET Magazine. [En línea] 2013. <http://technet.microsoft.com/es-es/magazine/2008.12.linux.aspx>.
13. **Python Software Foundation. 2013.** Python v2.7.5 documentation. [Online] 2013. <http://docs.python.org/2.7/>.
14. **Vernooi, Jelmer R., Terpstra, John H. and Carter, Gerald (Jerry). 2009.***The Official Samba 3.2.x HOWTO and Reference Guide.* EE.UU : s.n., 2009.
15. **Durán Toro, Amador y Bernárdez Jiménez, Beatriz. 2000.***Metodología para la Elicitación de Requisitos de Sistemas Software.* [ed.] Universidad de Sevilla. Sevilla : s.n., 2000. 97-0593-C05-01.

16. **ROUYOY, Romain; CONAN, Denis; SEINTURIER, Lionel.** Software architecture patterns for a context-processing middleware framework. Distributed Systems Online, IEEE, 2008, vol. 9, no 6, p. 1-1.
17. **KIRCHER, Michael; JAIN, Prashant.** Pattern-oriented software architecture, Patterns for resource management. Wiley, 2005.
18. **González Duque, Raúl. 2008.***Python para todos.* 2008.
19. **Jorba Esteve, Josep y Suppi Boldrito, Remo. 2004.***Administración Avanzada de GNU/Linux.* Barcelona : Eureka Media, SL, 2004. 84-9788-116-8.
20. **Petersen, Richard y Haddad, Ibrahim. 2004.***Red Hat Linux Pocket Administrator.* [ed.] Brandon A. Nordin. 2004. pág. 289. 0-07-222974-8.
21. **E. Wieggers, Karl. 2009.***Software Requirements.* [ed.] Inc. O'Reilly Media. 2nd. 2009. pág. 544. 0735637083, 9780735637085.
22. **J. Myers, Glenford, Sandler, Corey y Badgett, Tom . 2011.***The Art of Software Testing.* [ed.] Wiley. 3rd. 2011. pág. 240. 1118031962,978-1118031964.
23. **Lie Hetland, Magnus.***Beginning Python From Novice to Professional.* [ed.] Matt Moodie. s.l. : Springer-Verlag New York, Inc. 1-59059-519-X.
24. **Sommerville, Ian. 2005.***Ingeniería del software 7.* [ed.] Pearson Educación. [trad.] María Isabel Alfonso Galipienso. 7ma, ilustrada. s.l. : Pearson Educación, 2005. pág. 687. 8478290745, 9788478290741.
25. **Beazley, David y K. Jones, Brian. 2013.***Python Cookbook.* [ed.] O'Reilly Media Inc. 3rd. 2013. 978-1-4493-4036-0,1-4493-4036-9.
26. **Carter, Gerald. 2003.***LDAP system administration.* [ed.] O'Reilly. ilustrada. 2003. p. 294. 1565924916, 9781565924918.
27. **Donnelly, Michael. 2000.** ldapman.org. [Online] Abril 28, 2000. http://ldapman.org/articles/intro_to_ldap.html.
28. **Jackiewicz, Tom. 2004.***Deploying OpenLDAP.* [ed.] Apress. 2005. s.l. : O'Reilly, 2004. p. 344. 978-1590594131.
29. **Likewise Software. 2008.***Likewise Open Installation And Administration Guide.* [ed.] Likewise Software. 2008. pág. 72.
30. **Lutz, Mark. 2013.***Learning Python.* [ed.] O'Reilly Media Inc. 5th. 2013. pág. 1600. 978-1-4493-5568-5,1-4493-5568-4.
31. **MediaWiki. 2013.** Powerful Open Source LDAP. [En línea] 24 de Marzo de 2013. [Citado el: 16 de Abril de 2013.]
32. **Petersen, Richard y Haddad, Ibrahim. 2004.***Red Hat Linux Pocket Administrator.* [ed.] Brandon A. Nordin. 2004. pág. 289. 0-07-222974-8.

- 33. Petitpierre, Claude. 2006.** *Software Engineering: The Implementation Phase*. [ed.] EPFL Press. ilustrada. 2006. p. 349. 0849391725, 9780849391729.
- 34. Red Hat, Inc. 2011.** Deployment, configuration and administration of Fedora 14. [En línea] Red Hat, Inc. and others., 2011. http://docs.fedoraproject.org/en-US/Fedora/14/html/Deployment_Guide/index.html.
- 35. Seitz, Justin. 2009.** *Gray Hat Python*. 2009. 978-1-59327-224-1, 1-59327-224-3.
- 36. Timothy A. Howes, Ph.D., Smith, Mark C. and Good, Gordon S. 2003.** *Understanding and Deploying LDAP Directory Services*. 2nd. Michigan : Addison Wesley, 2003. p. 936. 9780672323164.

Anexos

Anexo 1. Código que permite integrar un ordenador con GNU/Linux a un AD.

```
#!/bin/sh -e
#   joinad.sh#
#   Copyright 2010
#   This program is free software; you can redistribute it and/or modify
#   it under the terms of the GNU General Public License as published by
#   the Free Software Foundation; either version 2 of the License, or
#   (at your option) any later version.
#   This program is distributed in the hope that it will be useful,
#   but WITHOUT ANY WARRANTY; without even the implied warranty of
#   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
#   GNU General Public License for more details.
#   You should have received a copy of the GNU General Public License
#   along with this program; if not, write to the Free Software
#   Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston,
#   MA 02110-1301, USA.
```

#Chequear dependencias

#Existencia de ficheros de configuración y backup

DOMAIN_C="ucidc1.uci.cu"

ID_USER=`id -u`

POOL="pool"

SAMBA_CONF="/etc/samba/smb.conf"

KRB_CONF="/etc/krb5.conf"

WINBIND_CONF="/etc/nsswitch.conf"

#DATE=`date +%N`

DATE="backup"

DATE_CONF="\$POOL/updatetime"

NTPDATE=`whereis -b ntpdate | awk '{print \$2}'`

HOSTNAME=`hostname`

DOMAIN="UCI.CU"

#¿Soy root?

if [\$ID_USER != "0"]

then

echo "El usuario no es root"

exit 0

fi

#Configuración de hosts y resolv.conf

cat << EOF

Verifique que las entradas en los ficheros /etc/hosts y /etc/resolv.conf sean como se muestra en el ejemplo:

```

#-----
#/etc/hosts
#-----
127.0.0.1    localhost.localdomain localhost
10.3.3.1    nombre_pc.uci.cu nombre_pc#-----
#/etc/resolv.conf
#-----
domain uci.cu
search uci.cu
nameserver 10.0.0.3
nameserver 10.0.0.4
EOF
echo "-----Contenido de /etc/hosts en $HOSTNAME-----"
cat /etc/hosts
echo "-----Contenido de /etc/resolv.conf en $HOSTNAME-----"
cat /etc/resolv.conf
echo "¿Es correcta la información mostrada? [s/n]"
read OPT
if [ $OPT = "n" ]
then
echo "Modifique los ficheros etc/resolv.conf y /etc/hosts de acuerdo al ejemplo mostrado."
exit 0
fi
#Ajustar Hora
if [ ! -z $NTPDATE ]
then
echo "copiando $DATE_CONF para /etc/cron.daily/"
cp $DATE_CONF /etc/cron.daily/
echo "Ejecutando $DATE_CONF"
sh $DATE_CONF
else
echo "No se puede encontrar ntpdate para sincronizar la hora del sistema con el controlador de dominio"
exit 128
fi
echo "Se añadirá la estación $HOSTNAME al dominio $DOMINIO"
#Backup

```

```

#Copiar ficheros
if [ -e $SAMBA_CONF ]
then
    echo "Se crea un fichero de backup para $SAMBA_CONF con el nombre: $SAMBA_CONF.${DATE}"
    mv $SAMBA_CONF $SAMBA_CONF.${DATE}
    cp "$POOL/smb.conf" /etc/samba/
else
    echo "No se encuentra el fichero $SAMBA_CONF. Pruebe: {Debian/ubuntu} aptitude install samba smbclient"
    exit 128fi

if [ -e $KRB_CONF ]
then
    echo "Se crea un fichero de backup para $KRB_CONF con el nombre: $KRB_CONF.${DATE}"
    mv $KRB_CONF $KRB_CONF.${DATE}
    cp $POOL/krb5.conf /etc/krb5.conf
else
    echo "No se encuentra el fichero $KRB_CONF. Pruebe: {Debian/ubuntu} aptitude install krb5-user krb5-config"
    exit 128

fi

if [ -e $WINBIND_CONF ]
then
    echo "Se crea un fichero de backup para $WINBIND_CONF con el nombre: $WINBIND_CONF.${DATE}"
    mv $WINBIND_CONF $WINBIND_CONF.${DATE}
    cp $POOL/nsswitch.conf /etc/nsswitch.conf
else
    echo "No se encuentra el fichero $WINBIND_CONF. Pruebe: {Debian/ubuntu} aptitude install winbind"
    exit 128

fi

#Modificar
sed -i -e 's/<NOMBRE_NETBIOS>/lola/' $SAMBA_CONF

#Reiniciar Servicios
/etc/init.d/samba restart
/etc/init.d/winbind restart

#Pedir ticket kerberos
echo "AutenticaciÃ³n Kerberos..."

echo "Usuario del dominio:"
read USERNAME

```

```
kinit $USERNAME@UCI.CU  
#Unir al dominio  
echo "Añadiendo PC al dominio..."  
net ads join -S $DOMAIN_C -U $USERNAME  
#Mostrar comprobaciones  
net ads info  
echo "Terminado"  
exit 0
```

Anexo 2. Encuesta realizada en empresas e instituciones cubanas

Encuesta a Especialista(s) que supervisan la Red.

Nombre de la Empresa:

Nombre del Especialista:

Seleccione que tecnología usa usted para realizar la administración centralizada de las entidades en la red de su empresa.

1. Microsoft Active Directory.
2. OpenLDAP.
3. Otra.

¿Existen en su empresa y las sucursales de la misma, ordenadores con bajas prestaciones de hardware?

1. Si.
¿Cuántas?
 - 1.1 menos de 50.
 - 1.2 entre 50 y 100.
 - 1.3 más de 100

2. No

¿ Considera usted necesario el desarrollo de una herramienta que permita integrar los ordenadores con GNU/Linux Nova al Directorio Activo, siendo posible que esta pueda ser ejecutada en ordenadores con bajas prestaciones, haciendo uso de la versión ligera de Nova?

1. Si. 2. No.

Glosario de Términos

API: (del inglés Application Programming Interface, lo que en español sería interfaz de programación de aplicaciones), es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas.

Código abierto: es el término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de poder acceder al código, que a las cuestiones éticas y morales las cuales se destacan en el software libre.

Código fuente: en informática se refiere al conjunto de instrucciones escritas en algún lenguaje de programación de computadoras, hechas para ser leídas y transformadas por alguna herramienta de software (compilador, intérprete, ensamblador) en lenguaje de máquina o instrucciones ejecutables en la máquina.

Código propietario: en informática un programa es de código propietario o cerrado cuando el código fuente no se encuentra disponible para cualquier usuario, es decir no se hace público. Se le llama así en contraposición al código abierto.

DB2: es una marca comercial, propiedad de IBM, bajo la cual se comercializa un sistema de gestión de base de datos del mismo nombre.

DHCP (sigla en inglés de Dynamic Host Configuration Protocol) es un protocolo de red que permite a los nodos de una red IP obtener sus parámetros de configuración automáticamente. Se trata de un protocolo de tipo cliente/servidor en el que generalmente un servidor posee una lista de direcciones IP dinámicas y las va asignando a los clientes conforme éstas van estando libres, sabiendo en todo momento quién ha estado en posesión de esa IP, cuánto tiempo la ha tenido y a quién se la ha asignado después.

DNS (Domain Name System, en español, sistema de nombre de dominio). Es un sistema de nomenclatura jerárquica para computadoras, servicios o cualquier recurso conectado al internet o a una red privada. Este sistema asocia información variada con nombres de dominios asignado a cada uno de los participantes. Su función más importante, es traducir (resolver) nombres inteligibles para los humanos en identificadores binarios asociados con los equipos conectados a la red, esto con el propósito de poder localizar y direccionar estos equipos mundialmente.

FOSS: Terminología utilizada para nombrar el software libre y de código abierto, por free/libre and open source software, en inglés.

Microsoft Corporation: Monopolio informático fundado en 1975 por Bill Gates y Paul Allen. Dedicado al sector de la informática, con sede en Redmond, Washington, Estados Unidos. Microsoft desarrolla, fabrica, licencia y produce software y equipos electrónicos. Sus productos más usados son el sistema operativo Microsoft Windows y la suite Microsoft Office, los cuales tienen una importante posición en el mundo de la informática.

Modelo OSI: Marco de referencia para la definición de arquitecturas de interconexión de sistemas de comunicaciones. Es un lineamiento funcional para tareas de comunicaciones y, por consiguiente, no especifica un estándar de comunicación para dichas tareas. Sin embargo, muchos estándares y protocolos cumplen con los lineamientos del Modelo OSI.

NIS: Siglas en inglés para Network Information Service (que en español significa *Sistema de Información de Red*), es el nombre de un protocolo de servicios de directorios cliente-servidor desarrollado por Sun Microsystems para el envío de datos de configuración en sistemas distribuidos tales como nombres de usuarios y hosts entre computadoras sobre una red.

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

Oracle: sistema de base de datos propiedad de Oracle Corporation, quien es una de las mayores compañías de software del mundo.

POO (Programación Orientada a Objetos), Es un paradigma de programación que usa objetos y sus interacciones, para diseñar aplicaciones y programas informáticos. Está basado en varias técnicas, incluyendo herencia, abstracción, polimorfismo y encapsulamiento. Su uso se popularizó a principios de la década de los años 1990. En la actualidad, existe variedad de lenguajes de programación que soportan la orientación a objetos.

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

SAP: software modular para la gestión empresarial creado por SAP AG.

SAP AG: compañía informática con sede en Walldorf, Alemania.

Single sign-on (SSO) es un procedimiento de autenticación que habilita al usuario para acceder a varios sistemas con una sola instancia de identificación. Su traducción literal sería algo como "sistema centralizado de autenticación y autorización".

TCP/IP: Conjunto de protocolos de red en los que se basa Internet y que permiten la transmisión de datos entre redes de computadoras. En ocasiones se le denomina conjunto de protocolos TCP/IP, en referencia a los dos protocolos más importantes que la componen: Protocolo de Control de Transmisión (TCP) y Protocolo de Internet (IP), que fueron los dos primeros en definirse, y que son los más utilizados de la familia.