

eman ta zabal zazu



universidad  
del país vasco

euskal herriko  
unibertsitatea

**Facultad de  
Informática**

**Informatika Fakultatea**

**TITULACIÓN: Ingeniería Informática**

**Desarrollo de la herramienta ADMINTOOL usando tecnología .NET**

**Alumno/a: D./Dña. Xabier Guereño Beares**

**Director/a: D./Dña. German Rigau Claramunt**

**Proyecto Fin de Carrera, julio de 2012**

© 2012 Xabier Guereño



## **AGRADECIMIENTOS**

Primero quiero dar las gracias a la Universidad del país Vasco (UPV/EHU) y a la empresa Ibermática Ciencia y Conocimiento que me han dado la oportunidad de realizar mi proyecto de fin de carrera.

En el plano personal, quiero agradecer la orientación y el apoyo de mi director del proyecto Germán Rigau, a mi tutor de la empresa César Otaño; porque sin su ayuda no hubiese sido posible llevar a buen término este proyecto y a los trabajadores de la empresa que en algún momento del desarrollo me han ayudado, Igor Ibarbia, Iñaki Colina, Julen Lizaso, David Montero, Iñaki Gorriti, Íñigo Lizaso y Guillermo Feijoo.

## **RESUMEN**

Este documento corresponde a la memoria del Proyecto Fin de Carrera para la obtención de la titulación superior de Ingeniería en Informática. Ha sido realizado por Xabier Guereño Beares en Ibermática Ciencia y Conocimiento, bajo la dirección de César Otaño (tutor por parte de la empresa) y Germán Rigau (Director de proyecto).

Título del Proyecto: **Desarrollo de la herramienta ADMINTOOL usando tecnología .NET**

El proyecto consiste en una aplicación para trabajadores de soporte del primer nivel de cualquier empresa en la que se utilice una red distribuida cuyo sistema de directorio sea el Directorio Activo de Microsoft. Se necesita que dicha aplicación permita llevar a cabo las tareas necesarias para la gestión y mantenimiento del citado Directorio Activo, la gestión de carpetas compartidas de los servidores de la red y el control de la configuración de los navegadores Internet Explorer en remoto.

# ÍNDICE

<b>1</b>	<b>Introducción.....</b>	<b>1</b>
<b>2</b>	<b>DOP.....</b>	<b>3</b>
2.1	Descripción:.....	3
2.2	Objetivos del proyecto:.....	3
2.3	Método del trabajo:.....	3
2.3.1	Estructura de las iteraciones:.....	4
2.4	Alcance:.....	5
2.4.1	Entregas:.....	5
2.4.2	Diagrama de dependencias / EDT:.....	6
2.4.3	Asignación de recursos:.....	6
2.4.4	Equipo de trabajo:.....	6
2.4.5	Estimación del esfuerzo:.....	6
2.5	Planificación temporal:.....	9
2.6	Plan de contingencia.....	10
2.7	Factibilidad.....	11
<b>3</b>	<b>Explicación del Entorno de trabajo.....</b>	<b>13</b>
3.1	Directorio Activo.....	13
3.1.1	Definición.....	13
3.1.2	Objetos.....	13
3.1.3	Unidades Organizativas .....	14
3.1.4	Dominios.....	15
3.1.5	Catálogo Global .....	16
3.1.6	Directorio Activo como Espacio de Nombres .....	17
3.1.7	Estándares más importantes que componen un Directorio Activo.....	18
3.1.7.1	DNS (Domain Name System).....	18
3.1.7.2	LDAP (Lightweight Directory Access Protocol): .....	18
3.1.7.3	Kerberos: .....	19
3.1.8	Estructura Física .....	19
3.1.9	Directo Activo a administrar por Admintool .....	20
3.2	Exchange.....	20
3.2.1	Definición, Exchange para usuarios y grupos.....	20
3.2.2	Grupos administrativos.....	21
3.2.3	Grupos de enrutamiento.....	21
3.2.4	Interoperabilidad con versiones anteriores de Exchange.....	21
3.3	Software utilizado.....	22
3.4	Arquitectura de la aplicación.....	23
3.4.1	Arquitectura por capas.....	23
<b>4</b>	<b>Captura de requisitos.....</b>	<b>25</b>
4.1	Diagrama de casos de uso.....	25
4.1.1	Diagrama completo:.....	26
4.1.2	Casos de uso que no gestionan las propiedades de objetos:.....	27
4.1.3	Propiedades de usuario:.....	27
4.1.4	Propiedades Grupo:.....	29
4.1.5	Propiedades Equipo:.....	29
4.2	Casos de Uso Reales.....	30

4.2.1 Casos de uso de la primera iteración.....	31
4.2.2 Casos de uso de la segunda iteración.....	36
4.2.2.1 objeto de tipo usuario.....	36
4.2.2.2 objeto de tipo grupo .....	56
4.2.2.3 objeto de tipo equipo.....	66
4.2.3 Casos de uso de la tercera iteración.....	72
4.2.3.1 Gestión directorio Activo.....	72
4.2.3.2 Buscar Objetos.....	74
4.2.3.3 Casos de uso relativos a las Tareas Exchange.....	76
4.2.3.4 Mover objeto.....	82
4.2.3.5 Eliminar objeto.....	83
4.2.3.6 Cambiar nombre al objeto.....	83
4.2.3.7 Habilitar/Deshabilitar objeto.....	84
4.2.3.8 Resetear contraseña de un usuario.....	84
4.2.3.9 Desbloquear usuario.....	85
4.2.4 Casos de uso de la cuarta iteración.....	85
4.2.4.1 Configuración IExplorer.....	85
4.2.4.2 Renombrar perfil.....	86
4.2.4.3 Gestionar carpetas compartidas.....	87
4.3 Modelo de dominio.....	89
4.3.1 Atributos principales de los objetos.....	89

## **5 Análisis.....93**

5.1 Casos de uso de la primera iteración.....	93
5.2 Caso de uso de la segunda iteración.....	96
5.2.1 Casos de uso de propiedades del usuario.....	96
5.2.2 Casos de uso relativos a las propiedades de grupo.....	113
5.2.3 Casos de uso relativos a las propiedades del equipo.....	120
5.3 Casos de uso de la tercera iteración.....	123
5.3.1 Caso de uso buscar objeto.....	123
5.3.2 Caso de uso Resetear contraseña usuario.....	124
5.3.3 Caso de uso Desbloquear usuario.....	125
5.3.4 Caso de uso Mover objeto.....	126
5.3.5 Caso de uso Eliminar objeto.....	127
5.3.6 Caso de uso Cambiar nombre del objeto.....	128
5.3.7 Caso de uso Cambiar nombre del objeto.....	128
5.3.8 Casos de uso relativos a las Tareas Exchange.....	129
5.4 Casos de uso de la cuarta iteración.....	133
5.4.1 Caso de uso cambiar configuración IExplorer.....	133
5.4.2 Caso de uso cambiar nombre perfil.....	134
5.4.3 Caso de uso gestionar carpetas.....	135

## **6 Diseño.....139**

6.1 Diseño de la capa de presentación:.....	139
6.2 Diseño de la capa de negocio:.....	139
6.2.1 Contratos de los casos de uso pertenecientes a la primera iteración.....	139
6.2.2 Contratos de los casos de uso pertenecientes a la segunda iteración.....	141
6.2.3 Contratos de los casos de uso pertenecientes a la tercera iteración.....	159
6.2.4 Contratos de los casos de uso pertenecientes a la cuarta iteración.....	164
6.3 Diseño de la capa de datos:.....	168
6.3.1 Gestión del Directorio Activo:.....	168

6.3.1.1 Usuarios, Grupos, Equipos:.....	168
6.3.1.2 Servidores y Servidores Exchange:.....	168
6.3.2 Configuración internet Explorer:.....	168
6.3.3 Renombrar Perfil y Gestión de carpetas compartidas:.....	169
<b>7 Implementación.....</b>	<b>171</b>
7.1 Clases utilizadas.....	171
7.2 Código.....	172
7.2.1 Gestión del Directorio Activo:.....	172
7.2.1.1 Entrada de Directorio Activo.....	172
7.2.1.2 Propiedades.....	173
7.2.1.3 Crear Objeto.....	174
7.2.1.4 Mover Objeto.....	174
7.2.1.5 Renombrar Objeto.....	175
7.2.1.6 Eliminar Objeto.....	175
7.2.1.7 Establecer Password.....	175
7.2.1.8 Establecer Reglas de Acceso.....	175
7.2.1.9 Buzón Exchange.....	177
7.2.1.10 Seguridad.....	178
7.2.2 Cambios en la configuración de Internet Explorer:.....	179
7.2.2.1 Habilitar/Deshabilitar el proxy actual.....	180
7.2.2.2 Obtener y modificar el Proxy actual.....	180
7.2.2.3 Obtener y modificar las excepciones del proxy.....	180
7.2.3 Acceso a carpetas en remoto para Gestión de Carpetas y Cambio de nombre de perfil.....	181
7.2.3.1 Crear carpeta.....	181
7.2.3.2 Eliminar carpeta.....	181
7.2.3.3 Renombrar carpeta, que en realidad la mueve a la dirección con su nuevo nombre.....	181
<b>8 Pruebas:.....</b>	<b>183</b>
8.1 Pruebas unitarias.....	183
8.1.1 caja negra.....	183
8.1.1.1 Pruebas en la primera iteración:.....	183
8.1.1.2 Pruebas para la segunda iteración.....	185
8.1.1.3 Pruebas para la tercera y cuarta iteración.....	188
8.1.2 Pruebas aceptación.....	190
8.1.2.1 Errores.....	190
8.1.2.2 Mejoras.....	190
<b>9 Gestión.....</b>	<b>193</b>
9.1 Incidencias.....	193
9.1.1 Seguimiento de la Fase 1:.....	193
9.1.2 Seguimiento de la Fase 2:.....	193
9.1.3 Incidencias en la Fase 3:.....	194
9.1.4 Incidencias en la fase 4:.....	195
9.1.5 Incidencias en la fase 5:.....	196
9.1.6 Incidencias en la Fase 6:.....	197
9.2 Horas totales.....	197
9.3 Gantt Planificado VS Gantt Real.....	199

<b>10 Conclusiones.....</b>	<b>201</b>
10.1 Plano personal.....	201
<b>11 BIBLIOGRAFÍA.....</b>	<b>203</b>
<b>12 Anexos.....</b>	<b>205</b>
12.1 Manual de Instalación:.....	205
12.1.1 Introducción.....	205
12.1.2 Requisitos previos.....	205
12.1.3 Instalación de AdminToolSetup.msi.....	206



# 1 INTRODUCCIÓN

*Desarrollo de la herramienta ADMINTOOL usando tecnología .NET* es un proyecto desarrollado por *Xabier Guereño* que se corresponde con el proyecto de fin de carrera de la titulación Ingeniería en Informática de la UPV/EHU en el curso académico 2011/2012.

El proyecto se ha llevado a cabo en la empresa Ibermática Ciencia y Conocimiento y ha consistido en realizar una aplicación que permita llevar a cabo las tareas necesarias para la gestión del directorio activo (DA), la gestión de carpetas compartidas de los servidores de la red y el control de la configuración de los navegadores Internet Explorer en remoto, por parte de un agente de soporte del cliente (primer nivel de soporte) de cualquier institución.

Como se ha podido apreciar, las tareas se pueden dividir en tres grupos distintos. Las de gestión del directorio activo son, por ejemplo, altas de usuario, agregar usuario a grupo, alta de grupo, mover usuario, etc... Para ello, la aplicación permitirá al agente poder realizar cambios en el directorio activo sin la necesidad de validarse como administrador, evitando así, comprometer la seguridad del dominio. Las de gestión de carpetas, le permitirá crear, eliminar carpetas compartidas de los servidores de la red. Para terminar, las de control de configuración del navegador Internet Explorer serán cambiar el proxy, habilitarlo o deshabilitarlo etc.

Al existir una versión previa de esta aplicación, al principio, se pensó en mejorarla para dotarla de más funcionalidades. Pero el diseño de la anterior versión no era apropiado para su ampliación. Además, como el título del proyecto indica, la nueva versión se ha desarrollado en Visual Basic .Net y la anterior lo estaba en Visual Basic 6.0, y se estimó que suponía un gran esfuerzo adaptar el código al nuevo lenguaje. Así que no se ha reutilizado código de la anterior versión de la aplicación para el desarrollo de la actual.

La aplicación viene a solventar la problemática que se genera cuando un agente de soporte debe de validarse como administrador del servidor del cliente, y para ello obtiene más permisos de los que necesita para llevar a cabo las tareas básicas de gestión. De esta manera, evitamos el riesgo que pueda existir de alguna acción que comprometa la seguridad del dominio que está gestionando.

Por último, aunque la aplicación se va a probar en el DA de la empresa Ibermática, la aplicación se ha realizado de manera que sea lo más genérica posible a fin de poder utilizarse en cualquier otra empresa.



## 2 DOP

### 2.1 Descripción:

GESTOR DEL DIRECTORIO ACTIVO ADMINTOOL es un proyecto a desarrollar por Xabier Guereño Beares, que se corresponde con el Proyecto Final de Carrera de la titulación Ingeniería Informática en el curso 2011/2012.

El proyecto consiste en realizar una aplicación que permita llevar a cabo las tareas necesarias para la gestión del directorio activo (DA<sup>1</sup>) por parte de un agente de soporte del cliente (primer nivel de soporte) de cualquier institución. Éstas tareas serán las propias de su rol dentro de la gestión de incidencias y peticiones según ITIL<sup>2</sup>. La aplicación permitirá al agente poder realizar cambios en el directorio activo sin la necesidad de validarse como administrador y así no comprometer la seguridad del dominio<sup>3</sup>.

La aplicación viene a solventar la problemática que se genera cuando un agente de soporte ha de validarse como administrador del servidor del cliente, y obtiene más permisos de los que necesita para llevar a cabo las tareas básicas de gestión. De esta manera, nos encontramos ante el riesgo de que pueda realizar alguna acción que comprometa la seguridad del dominio que está gestionando.

El proyecto se va a llevar a cabo de manera coordinada entre el grupo de sistemas de la empresa Ibermática y un tutor de la facultad de informática de San Sebastián (FISS) de la Universidad del País Vasco (UPV/EHU), el Sr. Germán Rigau. Y la fase de pruebas e implantación se llevarán a cabo en el centro de soporte de primer nivel de la empresa, el Centro de Atención del Cliente (CAU).

### 2.2 Objetivos del proyecto:

- Crear una aplicación con la que los trabajadores del CAU puedan realizar las tareas de gestión del DA que tienen encomendadas por la metodología ITIL. Éstas, al ser simples y rutinarias, están pensadas para que se puedan realizar en un primer nivel de soporte técnico. De esta manera, la aplicación que se obtendrá, ayudará a los agentes del CAU a realizar las tareas de manera más eficiente, rápida y segura.
- Que la aplicación sea genérica y fácilmente modificable, a fin de que se adapte a las necesidades de diferentes clientes.
- Que la interfaz de la aplicación sea lo más intuitiva posible. Consiguiendo que el tiempo de adaptación del usuario a la misma sea el menor posible.

### 2.3 Método del trabajo:

El proyecto se va a llevar a cabo por fases, de manera incremental, siguiendo el Proceso Unificado Racional (RUP).

---

1 **DA**, sigla de Directorio Activo, término utilizado por Microsoft para referirse a su implementación de servicio de directorio en una red distribuida de ordenadores.

2 **ITIL** (del inglés *Information Technology Infrastructure Library*), es un marco de trabajo de las mejores prácticas destinadas a facilitar la entrega de servicios de tecnologías de la información (TI).

3 **Dominio**, conjunto de ordenadores conectados en una red que confían a uno de los equipos de dicha red la administración de los usuarios y los privilegios que cada uno de los usuarios tiene en dicha red.

Se han definido cuatro iteraciones. Cada una tendrá asignadas algunas de las funcionalidades que queremos asignar a la aplicación. Al final, todas las funcionalidades estarán asignadas a una de las cuatro iteraciones.

- Las funcionalidades asignadas a la primera iteración son creación de usuarios, grupos y equipos.
- Las asignadas a la segunda son obtención y modificación de las propiedades de los usuarios, grupos y equipos, siendo ésta la fase en la que se estima que el esfuerzo será mayor.
- Para la tercera iteración, las funcionalidades que afectan a usuarios, grupos y equipos por igual (Eliminar, mover etc.).
- Por último, la cuarta iteración se ocupará de las funcionalidades relativas a la gestión de carpetas, configuración de Internet Explorer y perfiles de manera remota. Esta última iteración incluirá, además de las pruebas en local, las pruebas en un entorno real.

Una vez concluidas las iteraciones, la aplicación se trasladará a un entorno de producción.

Se acordarán reuniones puntuales con el director del proyecto, con el tutor del proyecto así como con el CAU para comentar los posibles problemas que puedan surgir y aclarar dudas que puedan surgir

### **2.3.1 Estructura de las iteraciones:**

Cada iteración constará de estas cinco partes:

Captura de requisitos

La captura de requisitos consiste en la identificación del modelo de casos de uso a desarrollar en función a las necesidades del usuario. Para ello, habría que encontrar los propios casos de uso y los actores que van a formar parte de ellos. Después se seleccionarán los más relevantes y se formará con ellos, una vez detallados, el modelo.

- Antecedentes
- Modelos de uso
- Diagramas de caso de uso
- Casos de uso de alto nivel
- Casos de uso expandidos
- Modelo dominio
- Diseño pantallas
- Reuniones

Análisis

Se trata de realizar los siguientes elementos para describir el comportamiento de la aplicación

- Diagramas de secuencia:

Se trata de un tipo de diagrama usado para modelar interacción entre objetos en un sistema. Para ello, muestra la interacción de los objetos en la aplicación a través del tiempo para cada caso de uso.

- Contratos:

Describen el comportamiento del sistema cuando se invoca una operación en términos de cuáles son los cambios de estado de los datos y cuáles son las salidas que el sistema proporciona. Establece un vínculo entre las operaciones y el esquema conceptual.

Diseño

Podemos dividir el diseño en interno y externo

- Interno:

En esta parte se definen las entradas y salidas de datos para cada caso de uso, el modelo de datos (en este caso, el DA), flujo de información (interdependencia entre módulos del sistema), funciones de sistema y transacciones entre las clases. En definitiva, se define la lógica de Negocio.

- Externo:

En esta parte se definen las distintas interfaces y el estilo visual que tendrá finalmente la aplicación.

Implementación:

Se trata de realizar la programación en un lenguaje de programación de los algoritmos obtenidos en el Diseño.

Pruebas:

Se realizarán sobre el programa obtenido en la implementación y servirán para detectar fallos en su funcionamiento.

## **2.4 Alcance:**

### **2.4.1 Entregas:**

- DOP
- Primera iteración
- Segunda iteración
- Tercera iteración
- Cuarta iteración
- Presentación pública

## 2.4.2 Diagrama de dependencias / EDT:

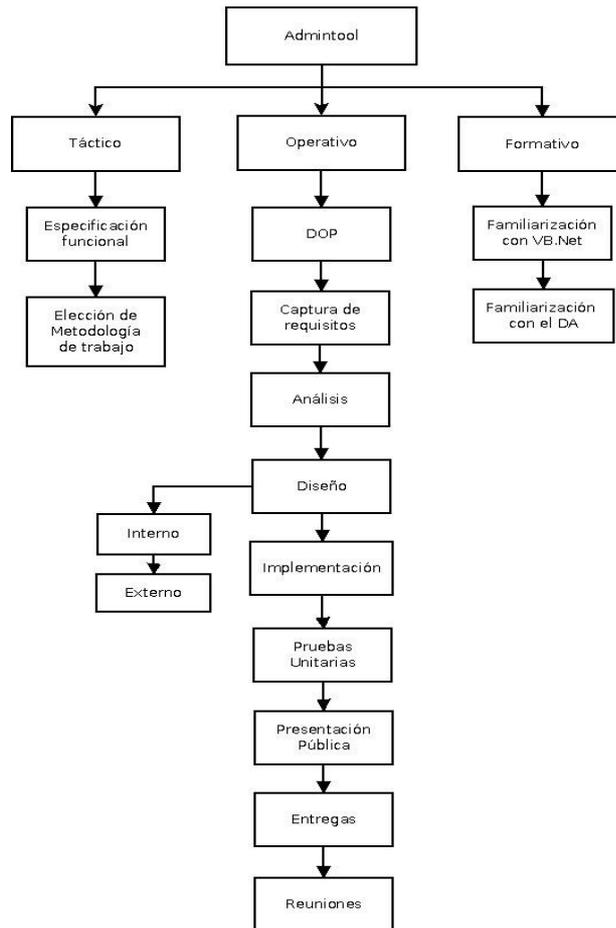


Figura 2.1 Estructura de descomposición del trabajo

## 2.4.3 Asignación de recursos:

- Fase1: DOP
- Fase2: Primera iteración
- Fase3: Segunda iteración
- Fase4: Tercera iteración
- Fase5: Cuarta iteración
- Fase6: Presentación pública

## 2.4.4 Equipo de trabajo:

Al tratarse de un proyecto personal, solo habrá una persona que se dedique a realizar todas las tareas.

## 2.4.5 Estimación del esfuerzo:

Se estima, por antecedentes y por la estimación de la carga de trabajo, que se necesitarán alrededor de 450 horas para llevar a cabo el proyecto. Suponiendo una carga lineal de trabajo, en la que se dedicarán 6 horas al desarrollo del proyecto por día, el total de días efectivos que se necesitarán para llevarlo a cabo serán 75.

450 horas \* 6h persona/día

Fase	Horas Dedicadas
fase1	2 días * 6h persona/día = 12 horas
fase2	15 días * 6h persona/día = 80 horas
fase3	25 días * 6h persona/día = 150 horas
fase4	14 días * 6h persona/día = 84 horas
fase5	18 días * 6h persona/día = 108 horas
fase6	1 días * 6h persona/día = 6 horas

#### Fase 1: DOP (12 h)

El Documento de Objetivos del Proyecto (DOP) se realiza al principio del mismo y marca las pautas a seguir durante todo el desarrollo del mismo. Quedando reflejados en él tanto en la parte de gestión del mismo como en el desarrollo técnico. Estará formado por:

- Descripción
- Objetivos
- Métodos de trabajo
- Alcance
- Planificación temporal
- Plan de contingencia

#### Fase2: Primera Iteración (80 h)

En esta fase se realiza lo siguiente:

- Se realiza la captura de requisitos con el fin de obtener los casos de uso relativos a la creación de usuarios, grupos y equipos
- Se realiza el análisis de esos casos de uso
- Se realiza el diseño de esos casos de uso
- Se implementan los casos de uso
- Se realizan las pruebas de lo implementado
- Se arreglan los problemas que se obtengan a la par que se mantienen reuniones con el tutor y el director de proyecto para llevar el seguimiento del proyecto.

#### Fase3: Segunda iteración (150h)

En esta fase se realiza lo siguiente:

- Se realiza la captura de requisitos con el fin de obtener los casos de uso relativos a la obtención y modificación de las propiedades de los usuarios, grupos y equipos
- Se realiza el análisis de esos casos de uso
- Se realiza el diseño de esos casos de uso

- Se implementan los casos de uso
- Se realizan las pruebas de lo implementado
- Se arreglan los problemas que se obtengan a la par que se mantienen reuniones con el tutor y el director de proyecto para llevar el seguimiento del proyecto.

#### Fase4: Tercera iteración (84h)

En esta fase se realiza lo siguiente:

- Se realiza la captura de requisitos con el fin de obtener los casos de uso generales que afectan a usuarios, grupos y equipos por igual
- Se realiza el análisis de esos casos de uso
- Se realiza el diseño de esos casos de uso
- Se implementan los casos de uso
- Se realizan las pruebas de lo implementado
- Se arreglan los problemas que se obtengan a la par que se mantienen reuniones con el tutor y el director de proyecto para llevar el seguimiento del proyecto.

#### Fase5: Cuarta iteración (108h)

En esta fase se realiza lo siguiente:

- Se realiza la captura de requisitos con el fin de obtener los casos de uso que afectan a la gestión de carpetas, la configuración del navegador Internet Explorer y el cambio de nombre de perfil.
- Se realiza el análisis de esos casos de uso
- Se realiza el diseño de esos casos de uso
- Se implementan los casos de uso
- Se realizan las pruebas de lo implementado en el entorno local
- Se arreglan los problemas que se obtengan a la par que se mantienen reuniones con el tutor y el director de proyecto para llevar el seguimiento del proyecto.
- Se realizan las pruebas, otra vez, de lo implementado en el entorno real.
- Se arreglan los problemas que se obtengan a la par que se mantienen reuniones con el tutor y el director de proyecto para llevar el seguimiento del proyecto. Y terminar la aplicación.

#### Fase6: Presentación Pública (6h)

Para terminar, se prepara la presentación del proyecto.

- Se mantiene una última reunión con el tutor para aclarar las dudas que puedan surgir en el punto anterior.
- Se ensaya la propia presentación

## 2.5 Planificación temporal:

Diagrama de Gantt:

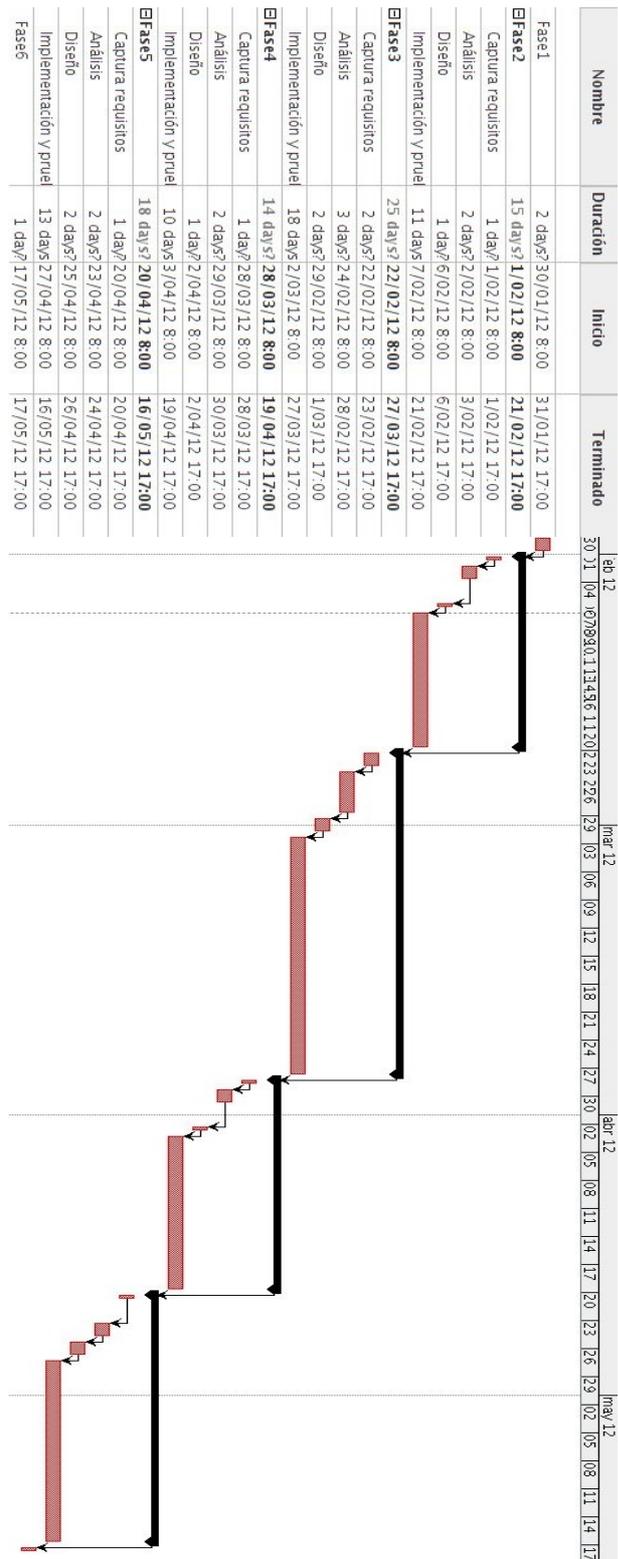


Figura 2.5.1 Diagrama de Gantt

## 2.6 Plan de contingencia

Se establecen ahora una serie de riesgos que pueden aparecer a lo largo de la realización del proyecto. Éstos, afectarían directamente al desarrollo del proyecto. Lo que puede ocasionar desde que se alarguen el tiempo necesario para llevarlo a cabo, hasta que directamente haya que empezarlo de cero. Seguido a los riesgos se ofrecen las medidas a tomar para minimizar en lo posible los daños que podrían causar.

**Riesgo:** Pérdidas de archivos o documentos

**Tipo:** Operativo

**Gravedad:** Baja

**Probabilidad:** Alta

**Prevención/Solución:** Se van a tener varias copias de respaldo (backups) de lo desarrollado, tanto en el equipo con el que se está llevando a cabo el trabajo como en otros equipos y en la nube gracias al servicio Dropbox. Se realizará una copia diaria del código y se subirá a la nube. Y dos veces por semana se realizará una copia de las máquinas virtuales en un disco duro externo.

**Riesgo:** Retraso en los plazos establecidos

**Tipo:** Operativo

**Gravedad:** Media

**Probabilidad:** Alta

**Prevención/Solución:** En caso de no llegar a cumplir alguno, se hará una replanificación del esfuerzo necesario para llevar a cabo el proyecto.

**Riesgo:** Problemas con el equipo que se está desarrollado

**Tipo:** Operativo

**Gravedad:** Media

**Probabilidad:** Baja

**Prevención/Solución:** Se tendrá un equipo de respaldo con la misma configuración para que pueda sustituir al que ha fallado.

**Riesgo:** Cambios inesperados a los requisitos obtenidos inicialmente

**Tipo:** Operativo

**Gravedad:** Baja

**Probabilidad:** Alta

**Prevención/Solución:** Se decidirá con el tutor y el director del proyecto si los cambios son posibles de realizar. Para, en caso contrario, no llevarlos a cabo.

**Riesgo:** Cambios inesperados en la interfaz de la aplicación

**Tipo:** Táctico

**Gravedad:** Alta

**Probabilidad:** Baja

**Prevención/Solución:** Se decidirá con el tutor y el director del proyecto si los cambios en la especificación funcional que afecten al Diseño Externo son posibles de realizar. Para, en caso contrario, no llevarlos a cabo.

**Riesgo:** Problemas de incompatibilidad entre las plataformas en las que se quiere implantar la aplicación y las librerías seleccionadas para la codificación de ésta.

**Tipo:** Formativo

**Gravedad:** Alta

**Probabilidad:** Baja

**Prevención/Solución:** Se llevará a cabo una búsqueda de alternativas a las librerías conflictivas consultando libros o páginas web especializadas, lo que obligará a una replanificar el esfuerzo necesario para llevar a cabo el proyecto

## **2.7 Factibilidad**

El tiempo en el que se ha de realizar el proyecto deber ser de aproximadamente 450 h/persona, teniendo en cuenta que la estimación no sobrepasa ese número de horas y que tengo un plan de contingencias que me permite abordar los problemas que puedan surgir durante la realización del mismo, considero factible la realización de este proyecto.



## 3 EXPLICACIÓN DEL ENTORNO DE TRABAJO



### 3.1 Directorio Activo

#### 3.1.1 Definición

El Directorio Activo (**Active Directory, AD**) es el término que usa Microsoft para referirse a su implementación de servicio de directorio en una red distribuida. Para la cual utiliza distintos protocolos (principalmente LDAP, DNS, Kerberos...).

Su estructura es jerárquica y permite mantener una serie de objetos relacionados de una red, como usuarios, grupos de usuarios, permisos, asignación de recursos y políticas de acceso. Además permite que los usuarios tengan un único inicio de sesión, a partir de lo cual pueden acceder a los recursos a los que está autorizado.

Active Directory permite a los administradores establecer políticas a nivel de empresa, desplegar programas en muchos ordenadores y aplicar actualizaciones críticas a una organización entera. Un Active Directory almacena información de una organización en una base de datos central, organizada y accesible, llamada Catálogo Global (GC). Pueden encontrarse desde directorios con cientos de objetos para una red pequeña hasta directorios con millones de objetos.

#### 3.1.2 Objetos

Como ya se ha dicho, Active Directory se basa en una estructura jerárquica de objetos. Los objetos se enmarcan en tres grandes categorías. — recursos (p.ej. impresoras), servicios (p.ej. correo electrónico), y usuarios (cuentas, o usuarios y grupos). El AD proporciona información sobre los objetos, los organiza, controla el acceso y establece la seguridad.

Cada objeto representa una entidad individual (ya sea un usuario, un grupo, un equipo, una impresora, una aplicación o una fuente compartida de datos) y sus atributos. Los objetos pueden contener otros objetos. Un objeto está unívocamente identificado por su nombre y tiene un conjunto de atributos (las características e información que el objeto puede contener) definidos por y dependientes del tipo. Los atributos, la estructura básica del objeto, se definen por un esquema, que también determina la clase de objetos que se pueden almacenar en el AD.

En la figura 3.1 podemos apreciar dos tipos de objeto, Equipo (Computers en la figura) y Usuario (Users) acompañados de varios de sus atributos, entre ellos el nombre y como éste les identifica luego en el Directorio Activo.

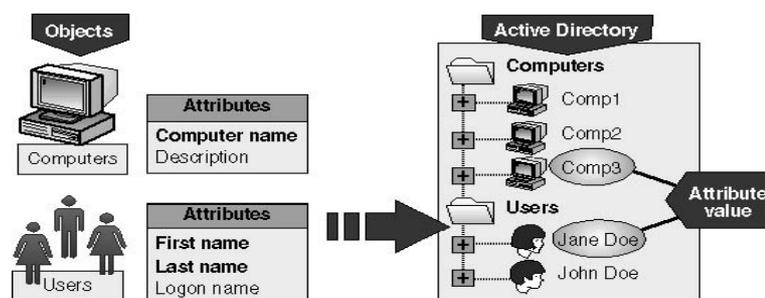


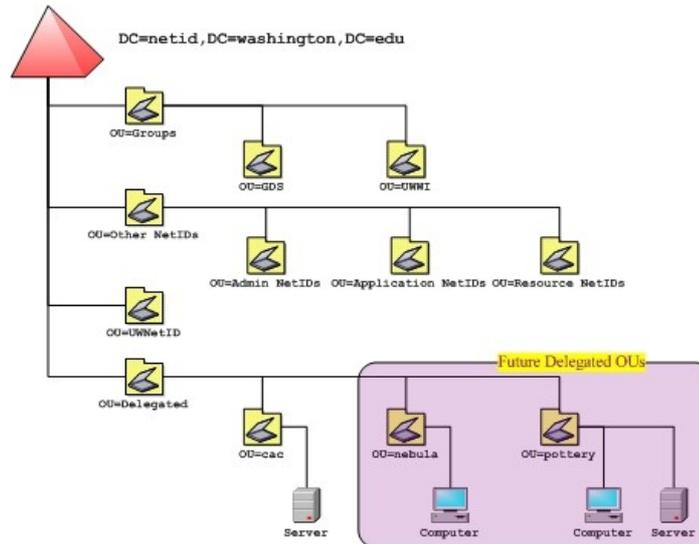
Figura 3.1 Ejemplo de objetos de Directorio Activo

### 3.1.3 Unidades Organizativas

Una Unidad Organizativa (Organizational Unit, OU) es un objeto del Directorio Activo que puede contener a otros objetos del directorio. Es decir, es un contenedor de otros objetos, de forma análoga a una carpeta o directorio en un sistema de archivos tradicional. En concreto, dentro de una unidad de este tipo pueden crearse cuentas de usuario, de grupo, de equipo, de recurso compartido, de impresora compartida, etc., además de otras unidades organizativas. Es decir, mediante unidades organizativas podemos crear una jerarquía de objetos en el directorio (lo cual se asemeja a un sistema de archivos típico de Windows). Los objetos ubicados dentro de una unidad organizativa pueden moverse más tarde a otra, si fuera necesario. Sin embargo, un objeto no puede copiarse: cada objeto es único en el directorio, y su existencia es independiente de la unidad organizativa a la que pertenece. Por tanto, el objetivo de las unidades organizativas es estructurar u organizar el conjunto de los objetos del directorio, agrupándolos de forma coherente. En el Directorio Activo, las unidades organizativas permiten:

- Conseguir una estructuración lógica de los objetos del directorio, de acuerdo con la organización de la empresa (por departamentos o secciones, sedes, delegaciones geográficas, etc.). Entre otras ventajas, esta organización le permite al administrador del dominio una gestión más lógica de usuarios, grupos, equipos, etc., pero también le permite a cualquier usuario una búsqueda de los objetos más sencilla cuando explora el directorio buscando recursos (por ejemplo, se podría localizar fácilmente las impresoras compartidas del edificio central de la delegación de Alicante).
- Delegar la administración. Cada unidad organizativa puede administrarse de forma independiente. En concreto, se puede otorgar la administración total o parcial de una unidad organizativa a un usuario o grupo de usuarios. Esto permite delegar la administración de subconjuntos estancos del dominio a ciertos usuarios que posean el nivel de responsabilidad adecuada.

- Establecer de forma centralizada comportamientos distintos a usuarios y equipos. A cada unidad organizativa pueden vincularse políticas de grupo, que aplican comportamientos (generalmente en forma de restricciones) a los usuarios y equipos cuyas cuentas se ubican en dicha unidad. De esta forma, podemos aplicar restricciones distintas a subconjuntos de usuarios y equipos del dominio, en función exclusivamente de la unidad organizativa donde se ubican. Por ejemplo, podemos limitar a los usuarios del departamento de contabilidad para que sólo puedan utilizar ciertas aplicaciones, pero que esto no se aplique a los



usuarios del departamento de informática.

Figura 3.2 Ejemplo de estructura de Unidades Organizativas

La figura 3.2 muestra un ejemplo de estructura formada por Unidades Organizativas. A modo de árbol, las OU hoja pueden contener objetos y son hijas de otras OU, algo que se repite con las OU de niveles superiores hasta las OU hijas de la raíz. Ésta puede ser otra OU o la raíz del dominio (Se explica en el siguiente apartado). Además, la zona sombreada, será administrada por un usuario o grupo de usuarios distinto al resto del árbol.

### 3.1.4 Dominios

La unidad central de la estructura lógica de Active Directory es el dominio. De hecho, es su menor unidad constructiva, es decir, Active Directory está compuesto por uno o más dominios. Se puede definir como un conjunto arbitrario de recursos de red, en el caso del AD objetos, bajo una administración común, tal que se cumpla:

- Límite de administración: Los administradores sólo tendrán poder de administración dentro de él. En el AD esta función la cumple el controlador de dominio.
- Límite de seguridad: Los administradores definirán los requerimientos de seguridad en su dominio. En el AD, serán las listas de control de acceso (ACL -Access Control List) las que controlen el acceso a los objetos del dominio. Las ACL contienen los permisos asociados con los objetos que controlan los usuarios que pueden acceder a un objeto, así como los tipos de acceso que pueden realizar.
- Límite de replicación: La información sensible de los recursos de red no se

replicará fuera de sus límites

Todos los objetos de la red existen en un dominio, y cada dominio almacena información exclusivamente sobre los objetos que contiene.

Al crear un dominio, el sistema crea una serie de contenedores predefinidos, y el administrador puede, y debería, crear una estructura jerárquica que facilite la administración. Esto es, debería crear una estructura de contenedores, Unidades Organizativas en el caso del Directorio Activo, que se adapte al entorno de trabajo y administración. En la figura 3.4 nos encontramos un ejemplo de dominio en el que el administrador es el controlador de dominio (Domain Controller) y los contenedores son las OU, que pueden a su vez contener objetos. Con el Schema se refiere al catálogo global que se explica más adelante.

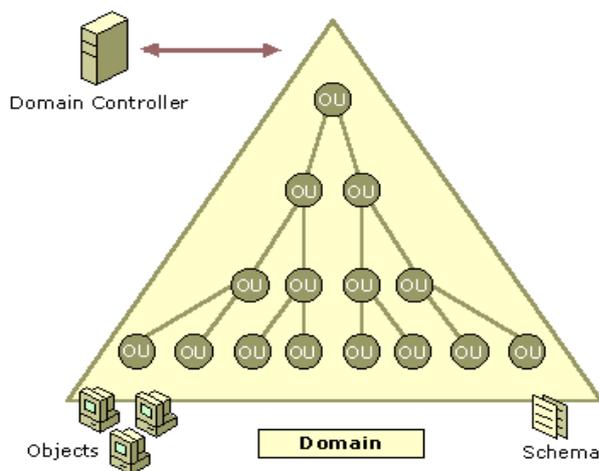


Figura 3.4 Ejemplo de dominio

Un dominio puede expandirse en más de una localización física. Para crear un dominio, los requisitos básicos son tener instalado un Windows, a partir de Windows 2000, en cualquiera de sus versiones de server, con protocolo de red TCP/IP instalado. Hay además un tercer requisito, que es la presencia de un servidor de nombres DNS instalado y configurado en la red.

### 3.1.5 Catálogo Global

El catálogo global es un depósito de información que contiene un subconjunto de atributos para todos los objetos del Active Directory. Los atributos que se almacenan en el catálogo global son los que se utilizan con más frecuencia en las consultas. Además, el catálogo global contiene la información necesaria para determinar la ubicación de cualquier objeto del directorio.

Un servidor de catálogo global es un controlador de dominio que almacena una copia del catálogo y procesa las consultas al mismo. El primer controlador de dominio que se crea en Active Directory es un servidor de catálogo global. Se pueden configurar controladores de dominio adicionales para que sean servidores de catálogo global con el fin de equilibrar el tráfico de autenticación de inicios de sesión y la transferencia de consultas. El catálogo global cumple dos funciones importantes en el directorio:

- Permite que un usuario inicie una sesión en la red mediante el suministro de la información de pertenencia a grupos universales a un controlador de dominio

- cuando inicia un proceso de sesión.
- Permite que un usuario busque información de directorio en todo el bosque, independiente de la ubicación de los datos.

### **3.1.6 Directorio Activo como Espacio de Nombres**

Un espacio de nombres es un área designada que tiene límites específicos donde se puede resolver un nombre lógico asignado a un objeto. El uso principal de un espacio de nombres es organizar las descripciones de los recursos para permitir a los usuarios localizarlos por sus características o propiedades. La base de datos del directorio para un espacio de nombres determinado se puede usar con el fin de localizar un objeto sin conocer su nombre. Si un usuario sabe el nombre de un recurso, puede consultar información útil acerca de ese objeto.

Una cuestión importante que hay que tener en cuenta es que el diseño del espacio de nombres determina, a la larga, el grado de utilidad que el Directorio Activo representará para los usuarios a medida que crezca. Los algoritmos de ordenación y búsqueda no pueden vencer los inconvenientes de un diseño lógico inadecuado.

En lo que se refiere a la lógica, el Directorio Activo es, simplemente, otro espacio de nombres. En él se almacenan dos tipos principales de información:

- La ubicación lógica del objeto.
- Una lista de atributos acerca del objeto.

Estos objetos al tener atributos asignados, como un número de teléfono, ubicación de oficina, etc., se pueden usar para localizar objetos en la base de datos del directorio. El uso de atributos para la búsqueda es incluso más importante cuando el esquema de el Directorio Activo se extiende, es decir, se modifica. Cuando se agregan objetos, clases de objetos o atributos de esos objetos a la base de datos del directorio, su estructura determina su utilidad para los usuarios del directorio.

Cada contenedor y objeto de un árbol tiene un nombre único. Un espacio de nombres es una colección de la ruta completa de todos los contenedores y objetos, o ramas y hojas, del árbol. La ubicación de un objeto en un árbol determina el nombre completo.

Un nombre completo se compone de la ruta de acceso completa desde el principio de un espacio de nombres específico a través de la jerarquía completa del árbol. Como los nombres diferenciados son útiles para organizar la base de datos de un directorio pero pueden no resultar de utilidad para recordar el objeto, en el Directorio Activo también se usan nombres en referencia relativa. Un nombre en referencia relativa es la parte del nombre de un objeto que es un atributo del propio objeto.

La base para el espacio de nombres usado en muchas redes se fundamenta en el Sistema de nombres de dominio (DNS, *Domain Name System*) usado en Internet. Esta conexión con DNS contribuye a determinar la forma del árbol de el Directorio Activo y la relación de los objetos entre sí. Los controladores de dominio son los dominios que se enumeran como nombres completos, mientras que los nombres comunes que son los formados por las rutas de acceso específicas de los objetos usuario del directorio.

### 3.1.7 Estándares más importantes que componen un Directorio Activo

#### 3.1.7.1 DNS (Domain Name System)

**Domain Name System** o **DNS (sistema de nombres de dominio)** es un sistema de nomenclatura jerárquica para equipos, servicios o cualquier recurso conectado a 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.

El servidor DNS utiliza una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet. Aunque como base de datos el DNS es capaz de asociar diferentes tipos de información a cada nombre, los usos más comunes son la asignación de nombres de dominio a direcciones IP y la localización de los servidores de correo electrónico de cada dominio.

Para la operación práctica del sistema DNS se utilizan tres componentes principales:

- Los **Cientes DNS** (objetos en AD): Un programa cliente DNS que se ejecuta en el equipo del usuario y que genera peticiones DNS de resolución de nombres a un servidor DNS (*Por ejemplo: ¿Qué dirección IP corresponde a nombre.dominio?*);
- Los **Servidores DNS** (Controladores de Dominio en AD): Que contestan las peticiones de los clientes. Los servidores recursivos tienen la capacidad de reenviar la petición a otro servidor si no disponen de la dirección solicitada.
- Y las **Zonas de autoridad** (Árboles en AD), porciones del espacio de nombres de dominio que almacenan los datos. Cada zona de autoridad abarca al menos un dominio y posiblemente sus subdominios, si estos últimos no son delegados a otras zonas de autoridad.

#### 3.1.7.2 LDAP (Lightweight Directory Access Protocol):

Protocolo ligero (o compacto) de acceso a directorio. Este es el protocolo mediante el que las aplicaciones acceden y modifican la información existente en el Directorio Activo, que a su vez está ordenado y distribuido para buscar diversa información en un entorno de red.

El formato de dirección LDAP es el que se muestra a continuación. Gracias a él se puede descubrir qué clientes soportan en variedad de grados, y qué servidores retornan como referentes y referencias de continuación.

La dirección comienza con los caracteres [ldap://](#) seguidos de diversos componentes, de los cuales los que poseen un interrogante delante de su nombre son opcionales, que se explican a continuación.

ldap://host:port/DN?attributes?scope?filter?extensions

- *host* es el nombre o la dirección IP del servidor LDAP donde se realiza la consulta.
- *port* es el puerto de red del servidor LDAP.
- *DN* es el nombre completo (o distinguido) a usar como base de búsqueda.
- *attributes* es una lista separada con comas de atributos a devolver.
- *scope* especifica el ámbito de búsqueda.
- *filter* es un filtro de búsqueda.
- *extensions* son extensiones al formato de la dirección LDAP.

Por ejemplo, "ldap://ldap.example.com/cn=John%20Doe,dc=example,dc=com" refiere a todos los usuarios en la entrada de John Doe en ldap.example.com, mientras "ldap:///dc=example,dc=com??sub?(givenName=John)" busca por la entrada en el servidor por defecto (notar el triple slash, omitiendo el host, y la marca de doble pregunta, omitiendo los atributos). Así como en otros URLs, los caracteres especiales deben ser codificados con signos de porcentaje.

### 3.1.7.3 Kerberos:

Protocolo utilizado para la autenticación de usuarios y máquinas. Permite a dos equipos demostrar su identidad mutuamente de manera segura. Sus diseñadores se concentraron primeramente en un modelo de cliente-servidor, y brinda autenticación mutua: tanto cliente como servidor verifican la identidad uno del otro.

Kerberos se basa en criptografía de clave simétrica y requiere un tercero de confianza. Además, existen extensiones del protocolo para poder utilizar criptografía de clave asimétrica.

### 3.1.8 Estructura Física

En Active Directory, la estructura lógica está separada de la estructura física. La estructura lógica se utiliza para organizar los recursos de red mientras que la estructura física se utiliza para configurar y administrar el tráfico de red. En concreto, la estructura física de Active Directory se compone de sitios y controladores de dominio. La estructura física de Active Directory define dónde y cuándo se producen el tráfico de replicación y de inicio de sesión. Una buena comprensión de los componentes físicos de Active Directory permite optimizar el tráfico de red y el proceso de inicio de sesión, así como solventar problemas de replicación.

**Sitio:** Una o varias subredes IP conectadas por vínculos de alta velocidad. Cada subred pertenece a un sitio. Un dominio puede estar formado por uno o mas sitios.

**Controlador de dominio (DC):** Equipo que permite administrar el directorio y contiene una réplica del mismo dividido en las siguientes "unidades de replicación" o "particiones del directorio":

- **Esquema:** contiene los tipo de objetos y atributos permitidos en el directorio. Se replica en todo el bosque.
- **Configuración:** contiene la estructura de los dominios y la topología de

- replicación. Se replica en todo el bosque.
- Dominio: contiene los objetos de un dominio. Se replica en un dominio.
- Aplicaciones: (opcional) contiene datos de aplicaciones. Se indica manualmente sus réplicas

### **3.1.9 Directo Activo a administrar por Admintool**

Si bien, en los apartados anteriores, se ha dado una idea general del Directorio Activo, la aplicación desarrollada en este proyecto de fin de carrera se va a centrar en la administración de un tipo concreto del mismo, el de dominio único, lo que provoca que el catálogo global tenga una importancia limitada.

## **3.2 Exchange**

Dentro de el directorio activo, una de las opciones para dotarlo de correo es el Microsoft Exchange Server.



### **3.2.1 Definición, Exchange para usuarios y grupos**

Exchange Server se basa en uno o varios servidores de correo, agrupados en uno o varios grupos administrativos, que formarán parte del dominio y una serie de buzones, agrupados en almacenes (MailboxStores), que se asociarán a los usuarios que necesiten una cuenta de correo. Además de los usuarios, los grupos podrán tener cuentas de correo (que no buzones), que se dedicarán a redireccionar los mensajes que se envíen a esa cuenta a todos sus miembros con buzón de correo, de un modo parecido a lo que haría una lista de distribución.

Cada usuario con buzón tiene la capacidad de realizar múltiples cambios en su configuración de correo, pudiendo incluso aceptar correos de servicios POP3 o IMAP4, aunque sobre estos no se tendrá ningún control (solo se recibirá). Los grupos también podrán realizar cambios sobre su cuenta de correo, como por ejemplo, controlar límites de remitentes o de tamaño de mensaje, pese a que los grupos no poseen buzón propio.

Además, Exchange obliga a los usuarios a utilizar Microsoft Outlook para gestionar su cuenta de correo.

De acuerdo con los requisitos administrativos, los requisitos de seguridad y la capacidad técnica, se puede optar por diseñar un modelo de administración centralizado, uno distribuido o la combinación de ambos. Con esta decisión, básicamente lo que se hace es determinar si crear uno o varios grupos administrativos.

### **3.2.2 Grupos administrativos**

Un grupo administrativo es un conjunto de objetos de Active Directory que se agrupan para simplificar la administración de permisos. Un grupo administrativo puede contener directivas, grupos de enrutamiento, jerarquías de carpetas públicas y redes de charla. El contenido de un grupo administrativo depende de las opciones seleccionadas durante el proceso de instalación. A menos que instale Exchange 2000, o una versión superior, en una organización de Exchange anterior, los grupos administrativos estarán deshabilitados en el Administrador del sistema. Puede habilitarlos si es necesario para dividir la administración de la organización de Exchange.

Una razón frecuente para crear un grupo administrativo es definir el ámbito de los permisos para los objetos del grupo. Por ejemplo, si la organización tiene dos conjuntos de administradores para dos conjuntos de servidores de Exchange, puede crear dos grupos administrativos que contengan estos dos conjuntos de servidores. Para establecer los permisos, puede agregar los usuarios y grupos de Windows 2000, o superior, que desee a la configuración de seguridad de los dos grupos administrativos. A continuación, Active Directory propaga esta configuración a todos los objetos de configuración del grupo administrativo.

Un grupo administrativo puede contener múltiples servidores y grupos de enrutamiento, pero no se puede mover servidores entre grupos administrativos. El grupo administrativo es similar a una carpeta en la que puede organizar las herramientas de administración del sistema Exchange. El uso de esta carpeta es similar a la forma de usar las carpetas para organizar archivos en el Explorador de Windows.

### **3.2.3 Grupos de enrutamiento**

Un grupo de enrutamiento es un conjunto de servidores de Exchange con conexiones permanentes y confiables. Los mensajes enviados entre dos servidores dentro de un grupo de enrutamiento van directamente del origen al destino. De forma análoga a los grupos administrativos, los grupos de enrutamiento son opcionales y no se ven en el Administrador del sistema a menos que los habilite.

La razón principal para crear grupos de enrutamiento es definir el enrutamiento de un único salto en la organización de Exchange; sin embargo, también puede crear grupos de enrutamiento entre servidores para administrar mensajes y controlar el flujo de mensajes entre límites administrativos.

### **3.2.4 Interoperabilidad con versiones anteriores de Exchange**

Cuando se instala Exchange a partir de la versión 2000 por primera vez, funciona en modo mixto. Este modo permite la coexistencia de servidores de Exchange de la versión 2000 en adelante y servidores que ejecutan versiones anteriores de Exchange en una misma organización. Al limitar la funcionalidad a las características que comparten ambos productos se posibilita la interoperabilidad entre versiones. Si sólo se ejecuta Exchange a partir de el 2000 Server en toda la organización y no tiene intención de incorporar versiones anteriores de Exchange, puede cambiar a modo nativo. El modo nativo significa que se usa una organización pura de Exchange 2000 Server en adelante, que permite aprovechar completamente la funcionalidad Exchange. El modo nativo

ofrece las siguientes ventajas:

- Aunque un servidor puede pertenecer a un grupo administrativo por razones administrativas, no tiene que pertenecer a uno de los grupos de enrutamiento del grupo administrativo.
- Los grupos administrativos no tienen que contener grupos de enrutamiento necesariamente.
- Un único grupo administrativo puede contener todos los grupos de enrutamiento de la organización.
- Puede mover buzones entre servidores de la organización.

Admintool gestionará el correo si se utiliza Exchange Server en el Directorio Activo para dotarlo de servicio de correo. Tanto en las pruebas en el entorno virtual como en las de entorno real se ha utilizado.

### 3.3 Software utilizado

A la hora de realizar este PFC, he utilizado las herramientas que me ha proporcionado la empresa.

La implementación se ha realizado en Visual Basic .Net y el entorno de desarrollo ha sido Visual Studio 2010. Esto se debe a que tienen otras aplicaciones en Visual Basic y querían mantener homogeneidad en toda la plataforma.



Figura 3.5 Caja del software Visual Studio 2010

Además, se ha interactuado con una base de datos centralizada que está implícita en los sistemas Windows Server a partir de 2003, el active directory. En mi caso, he utilizado un Windows Server 2003, ya que al ser las versiones posteriores retrocompatibles con la 2003, me aseguraba el buen funcionamiento de la aplicación en todas las demás.

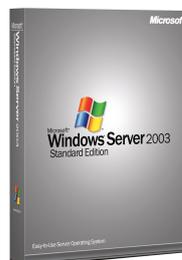


Figura 3.13 caja del SO Windows Server 2003

Para el desarrollo y pruebas de la aplicación he utilizado tres máquinas virtuales (dos Windows Server 2003 y una Cliente XP) para simular clientes y servidores. Para poder trabajar con ellas, se me proporcionó la versión 7.1 del software de virtualización

VMWare Workstation.

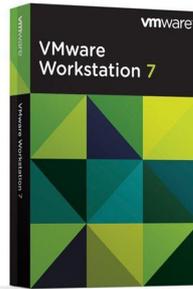


Figura 3.14 Caja del Software VMware Workstation 7

Además he instalado Exchange Server 2003 en las dos máquinas virtuales con Sistema Operativo Windows Server 2003, para que a su vez, sean servidores de correo. Su modo será el nativo. En cuanto a su ubicación lógica, cada servidor se encuentra en un grupo administrativo distinto.



Figura 3.15 Caja del Software Exchange Server 2003

### 3.4 Arquitectura de la aplicación

Para llevar a cabo la aplicación, me decanté por la arquitectura por capas. Ya que me parecía la mejor opción, puesto que, ésta, debía ser del tipo Cliente-Servidor.

#### 3.4.1 Arquitectura por capas

Separamos por capas el contenido de nuestra aplicación. Por un lado nos encontramos con la parte visual de la aplicación, otra parte será el código y todas aquellas funciones programadas, y por último la capa de datos.



Figura 7.1

- **Capa de Presentación**

Esta es la capa que ve el usuario. A través de esta capa el usuario puede comunicarse con la aplicación realizando las peticiones que estime oportunas. Mediante esta capa la aplicación también es capaz de comunicar una serie de avisos cuando se requiere que el usuario escoja entre una serie de opciones o cuando simplemente se le desea comunicar que ha sucedido cualquier acción. Para la aplicación Admintool, se tratará de la Interfaz de Usuario.

- Capa de la Lógica de Negocio

Es la capa donde se reciben peticiones del usuario y se envían las respuestas tras el proceso, es decir, es la capa que comunica la capa de presentación con la capa de datos. En ella se encontrarán todas aquellas funciones necesarias para el funcionamiento de la aplicación, como por ejemplo, crear un usuario, mover un grupo, renombrar un equipo, etc.

- Capa de Datos

Es la capa donde se encuentran almacenados los datos. A través de esta capa se podrá acceder a los mismos ,manipularlos mediante las funciones de la lógica de Negocio y volver a almacenarlos posteriormente. En el caso de Admintool, esta capa la formará el directorio Activo.

## **4 CAPTURA DE REQUISITOS**

### **4.1 Diagrama de casos de uso**

A continuación se mostrará el diagrama de casos de uso. Este diagrama posee setenta casos de uso, lo que provoca que sea demasiado grande como para poder apreciarse bien en una única figura. Por esa razón, se ha dividido en cuatro partes para poder tener una visión más detallada del mismo. No obstante, se ha incluido también el diagrama completo.

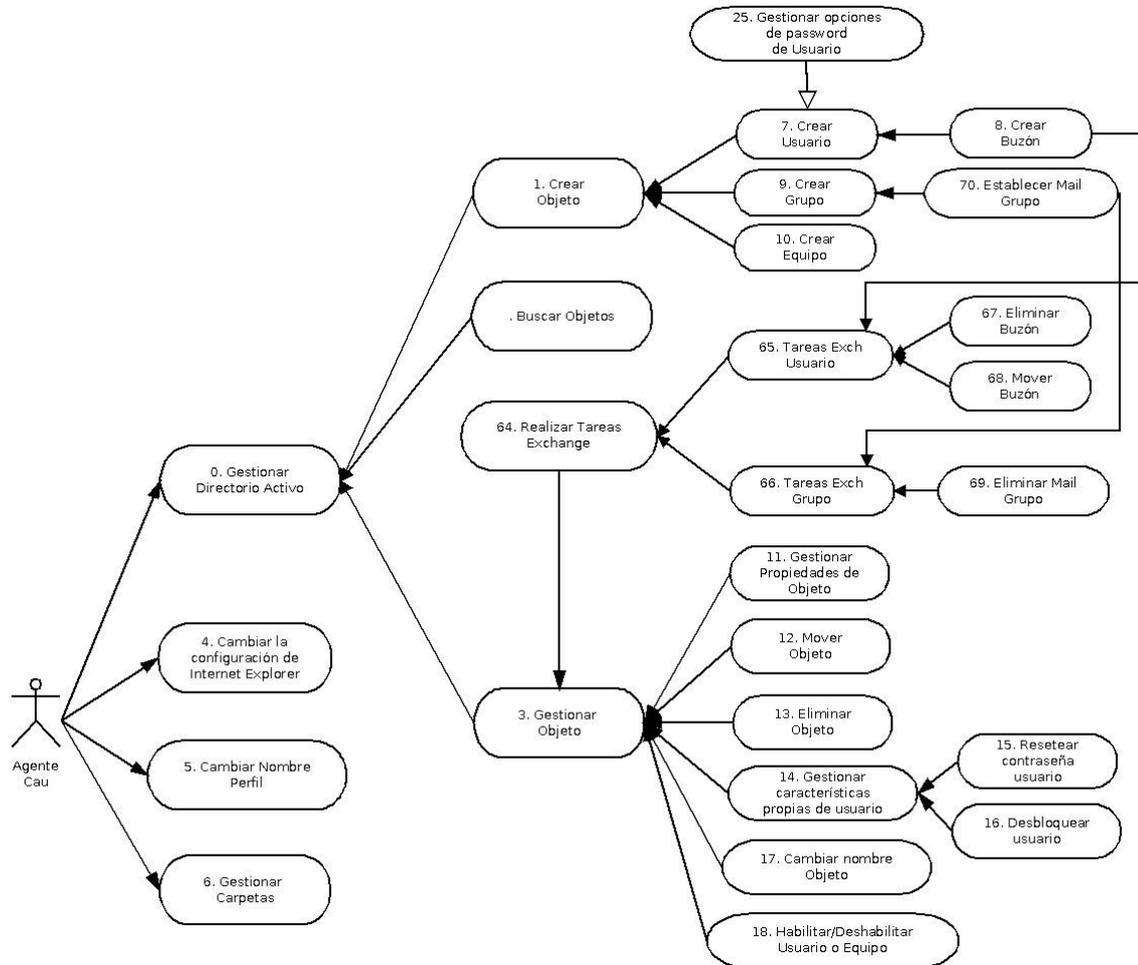
En cuanto a actores, solo habrá uno, el Agente del CAU. El Agente del CAU o Centro de Atención al Cliente, es el actor al que está destinada la aplicación. Se tratará de un agente de soporte del cliente de primer nivel de cualquier institución.

Las relaciones que aparecen en los diagramas se pueden dividir en dos tipos, extends y uses. Las flechas cuya punta aparece en negro serán extends y las que tienen la punta de color blanco son uses.



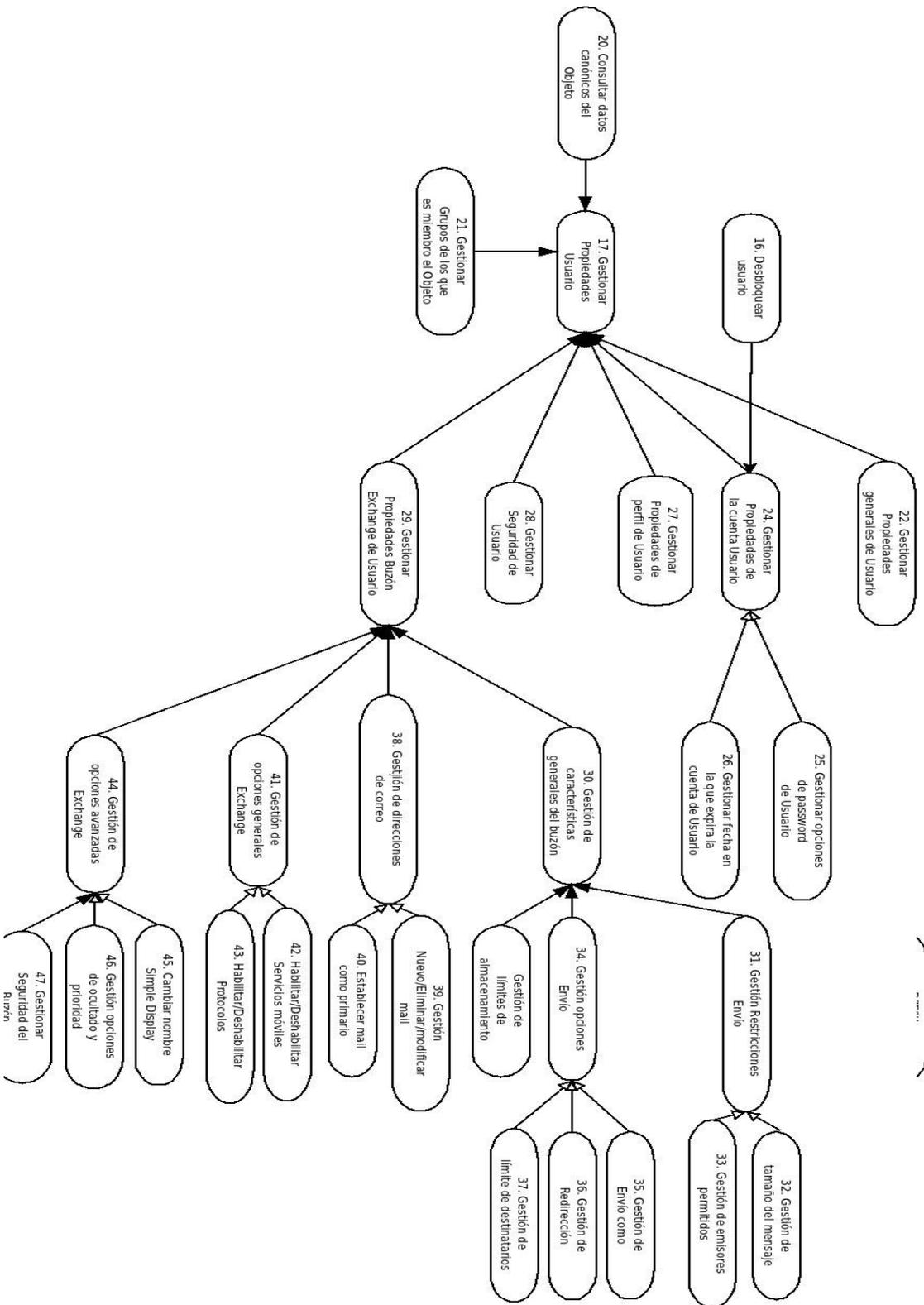
### 4.1.2 Casos de uso que no gestionan las propiedades de objetos:

En esta primera parte se muestran dos tipos de casos de uso. Por un lado, los casos de uso que se encargan de gestionar las partes de la aplicación que no afectan al directorio activo, es decir, los que afectan a la gestión de carpetas compartidas en remoto, el cambio de nombre de perfil y la gestión de la configuración del navegador web Internet Explorer en remoto. Y por el otro, los que afectan al directorio activo pero no se encargan de gestionar las propiedades de un objeto o extienden de los que sí lo hacen.



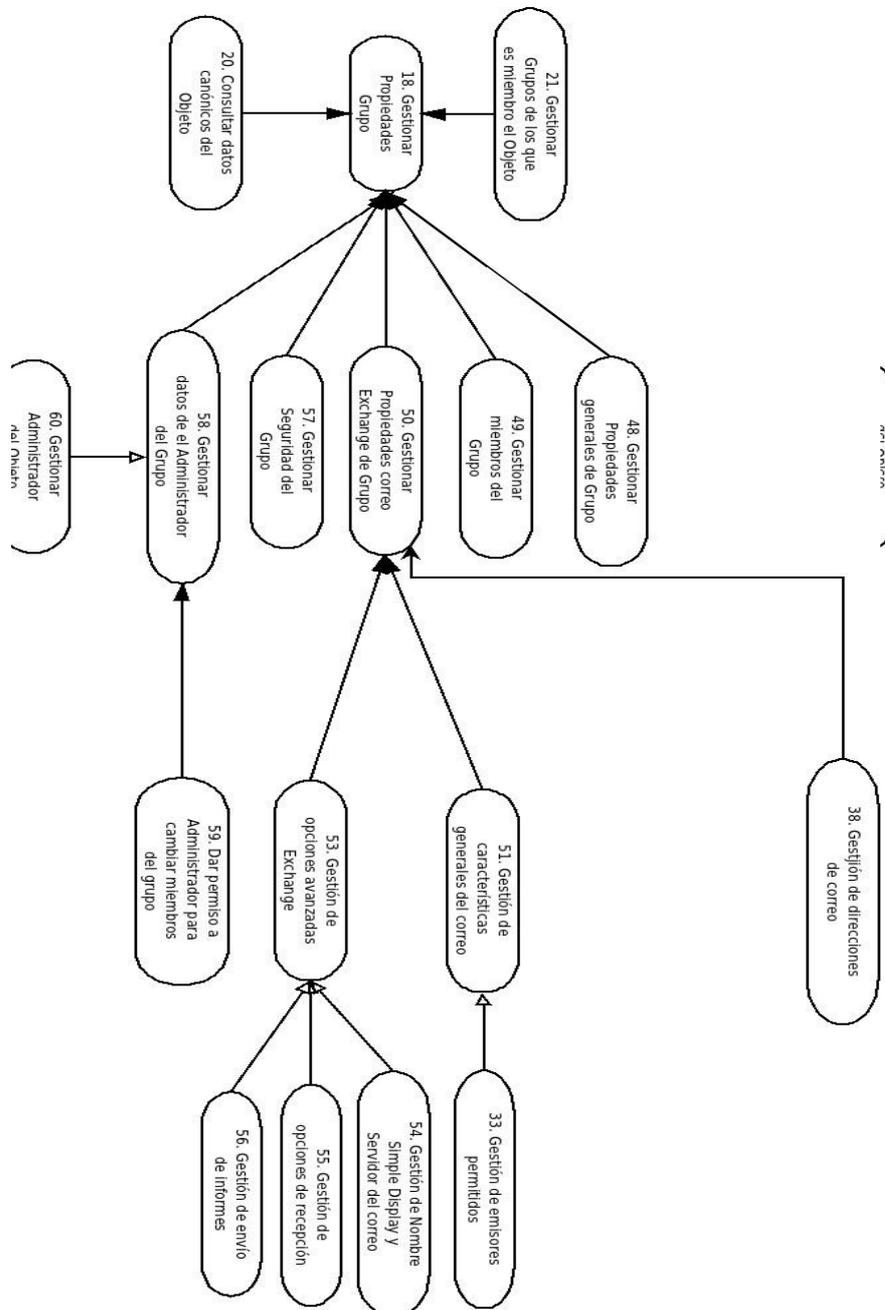
### 4.1.3 Propiedades de usuario:

En esta parte se muestran el caso de uso 17 (Gestión de propiedades de usuario) y todos los que extienden de él.



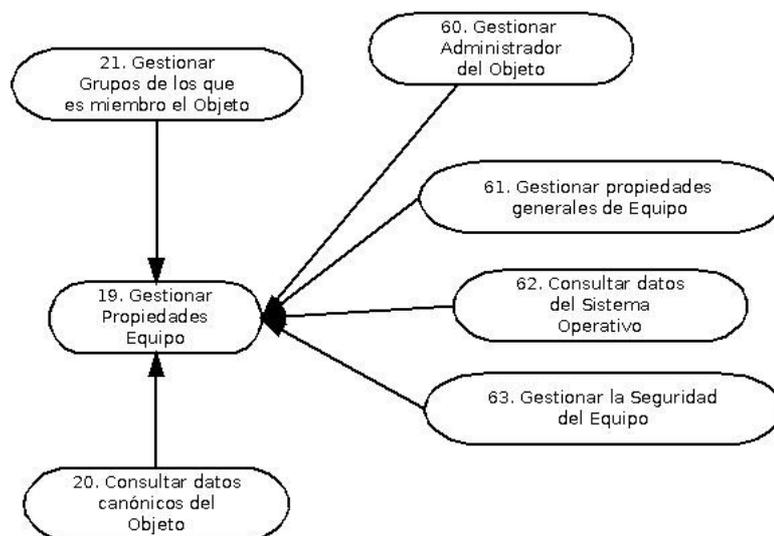
#### 4.1.4 Propiedades Grupo:

En esta parte se muestran el caso de uso 18 (Gestión de propiedades de grupo) y todos los que extienden de él.



#### 4.1.5 Propiedades Equipo:

Por último, se muestran el caso de uso 19 (Gestión de propiedades de equipo) y todos los que extienden de él.



## 4.2 Casos de Uso Reales

A continuación paso a hacer una breve descripción de los casos de uso más relevantes. Los no citados son meros contenedores de otros casos de uso.

Cada caso de uso citado a continuación se explicará junto con su interfaz correspondiente, ésta será la parte que verá el usuario de la aplicación y con la que interactuará.

Para extender los cuatro primeros casos de uso, la primera ventana que verá el usuario será la figura 4.0 Primera Admintool que se expone a continuación. La interfaz posee un desplegable, del cual al seleccionar la primera opción y hacer clic en aceptar el sistema extenderá el caso de uso Gestión del Directorio Activo (Número 0), al seleccionar la segunda extenderá Gestionar carpetas (Número 6), al hacer lo mismo con la tercera Cambiar configuración de Internet Explorer (Número 4) y al seleccionar la última Cambio Nombre perfil (Número 5). Todos estos casos de uso se explicarán a lo largo del tema 5, ya que las explicaciones detalladas de los casos de uso están ordenadas por la iteración a la que pertenezcan.



Figura 4.0 Primera Admintool

#### 4.2.1 Casos de uso de la primera iteración

Los casos de uso que se encargan de crear los objetos en el directorio activo

##### 7. Caso de Uso: Crear Usuario

Actores: Agente del CAU

Resumen: Este caso de uso se da cuando el agente del CAU se encuentra dentro del caso de uso “Crear Objeto”. Permite crear un usuario, para ello se deberán introducir varios datos, como el nombre del usuario, apellidos, etc. Posteriormente se deberá asignar un password al usuario y establecer las opciones de éste. Finalmente se podrá crear un buzón de correo Exchange que se asignará al usuario.

Precondición:

Postcondición: Se da de alta a un usuario dentro del directorio activo.

##### **Escenario principal (curso normal de eventos)**

1. Agente del CAU: El agente quiere crear un usuario
2. Sistema: Le muestra en la interfaz CrearUsuario1 (Figura 4.1) los campos que tiene que rellenar, de los cuales Nombre y Nombre Usuario Login son obligatorios.
3. Agente del CAU: Introduce los datos del usuario y hace clic en el botón Siguiente.
4. Sistema: Le muestra en la interfaz CrearUsuario2 (Figura 4.2) con los campos en los que debe introducir el password que quiere asignar al nuevo usuario. Usa el caso de uso de Gestionar Opciones de Password de Usuario (Número 25) para añadir las opciones de password a la interfaz.
5. Agente del CAU: Inserta el password en los dos campos y, en caso de que se quiera, marca una o varias de las opciones de password y hace clic en el botón Siguiente.
6. Sistema: Muestra la interfaz CrearUsuario3 (Figura 4.3) que posee la opción Crear Mailbox Exchange que el agente del CAU deberá marcar si quiere que el usuario tenga buzón de correo.
7. Agente del CAU: Establece si quiere crear un buzón de correo Exchange para el usuario, marcando la opción Crear Mailbox Exchange o no. En caso de no hacerlo, hace clic en el botón Siguiente.
8. Sistema: Si el agente quiere crear el buzón, extiende el caso de uso “crear buzón” (Número 8).
9. Agente del CAU: Si ha marcado que se quiere crear buzón, rellena el campo alias y selecciona una de las opciones del desplegable Server y otra del MailboxStore y hace clic en el botón Siguiente.
10. Sistema: Crea el usuario, crea el buzón si se han dado los pasos 8 y 9, y lo almacena en el Directorio activo

##### **Extensiones (cursos alternativos)**

Paso 3: El agente del CAU usuario introduce un nombre de usuario que ya existe en la red o deja en blanco los campos Nombre o Nombre Usuario Login.

- Sistema: Devuelve un mensaje de error y no cambia la siguiente interfaz.

Paso 5: El agente del CAU no ha introducido un password o el introducido no es igual al de confirmación.

- Sistema: Devuelve un mensaje de error y no cambia la siguiente interfaz.

Paso 5: El agente del CAU marca la opción “El usuario ha de cambiar el password en el siguiente login” y la opción “ El usuario no puede cambiar el password” a la vez.

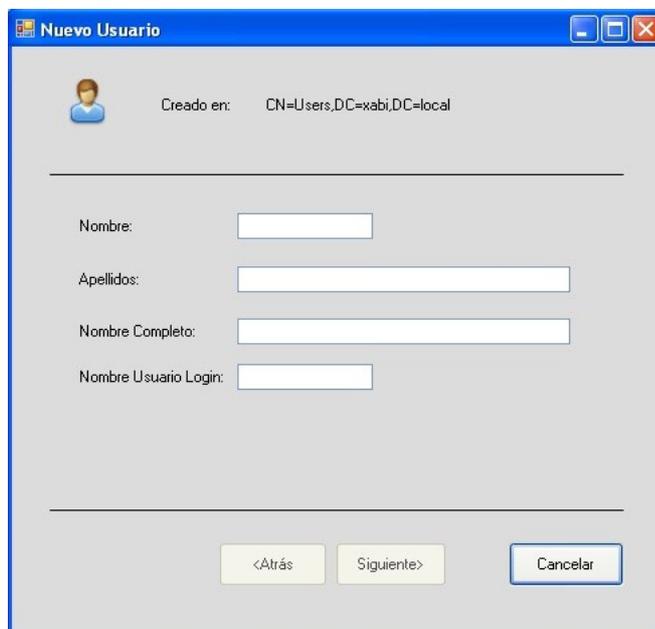
- Sistema: Devuelve un mensaje de error y no cambia la siguiente interfaz.

Paso 9: El agente del CAU deja vacío el campo alias o no selecciona ninguna de las opciones del desplegable Server o del Mailbox Store.

- Sistema: Devuelve un mensaje de error y no cambia la siguiente interfaz.

Paso 10: El agente del CAU ha insertado un password que no cumple con las medidas de seguridad de fortaleza de contraseña que requiere el directorio activo.

- Sistema: Devuelve un mensaje de error y vuelve al inicio del caso de uso.



Nuevo Usuario

Creado en: CN=Users,DC=xabi,DC=local

Nombre:

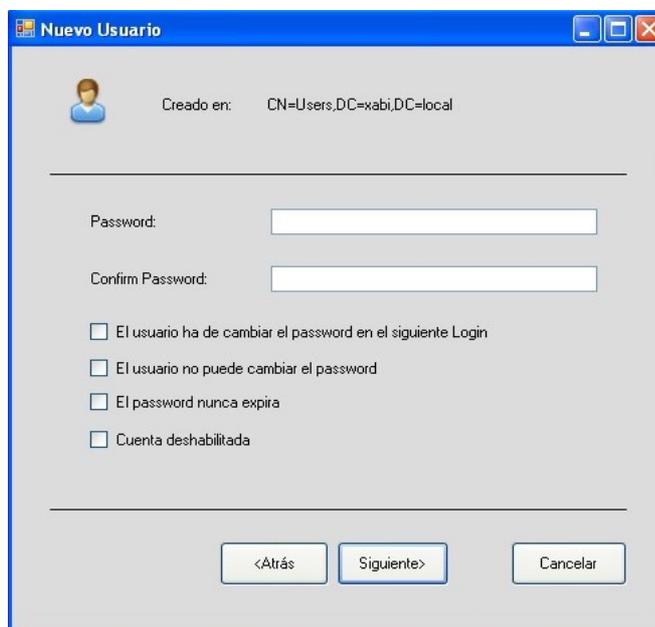
Apellidos:

Nombre Completo:

Nombre Usuario Login:

<Atrás   Siguiente>   Cancelar

Figura 4.1 CrearUsuario1



Nuevo Usuario

Creado en: CN=Users,DC=xabi,DC=local

Password:

Confirm Password:

El usuario ha de cambiar el password en el siguiente Login

El usuario no puede cambiar el password

El password nunca expira

Cuenta deshabilitada

<Atrás   Siguiente>   Cancelar

Figura 4.2 CrearUsuario2

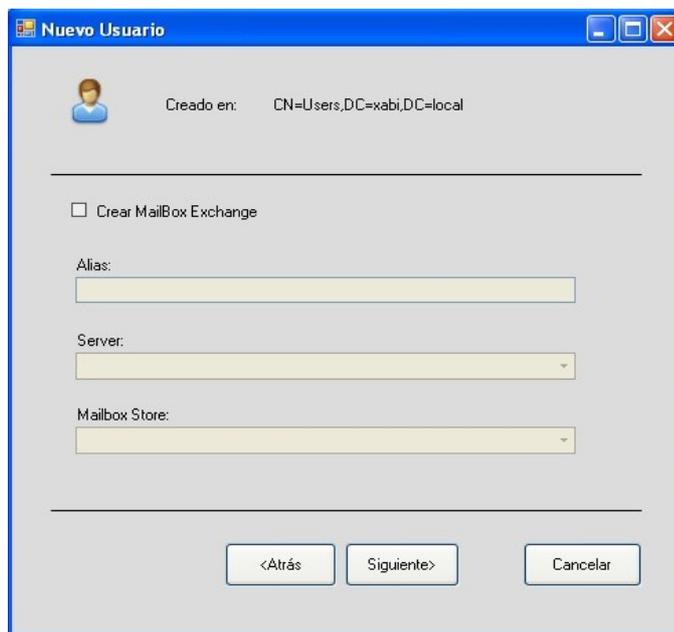


Figura 4.3 CrearUsuario3

## 9. Caso de Uso: Crear Grupo

**Actores:** Agente del CAU

**Resumen:** Este caso de uso se da cuando el agente del CAU se encuentra dentro del caso de uso “Crear Objeto”. Permite crear un grupo, para ello se deberán introducir varios datos, como el nombre del grupo y su tipo.

**Precondición:**

**Postcondición:** Se da de alta a un grupo dentro del directorio activo.

### Escenario principal (curso normal de eventos)

1. Agente del CAU: El agente quiere crear un grupo
2. Sistema: Le muestra en la interfaz CrearGrupo1 (Figura 4.4) en la que está el campo en el que se establece el nombre del grupo (Nombre del grupo) y los radio button en los que se selecciona el tipo y scope del grupo.
3. Agente del CAU: Introduce el nombre y selecciona un tipo y un scope para el grupo y hace clic en el botón Siguiete.
4. Sistema: Muestra la interfaz CrearGrupo2 (Figura 4.5) que posee la opción Crear una dirección email Exchange que el agente del CAU deberá marcar si quiere que el grupo posea una dirección de correo.
5. Agente del CAU: Establece si quiere crear una dirección de correo Exchange para el grupo, marcando la opción Crear una dirección email Exchange o no. En caso de no hacerlo, hace clic en el botón Siguiete.
6. Sistema: Si el agente quiere crear la dirección de correo, extiende el caso de uso “Establecer Mail Grupo” (Número 70).
7. Agente del CAU: Si ha marcado que se quiere crear la dirección de correo, rellena el campo alias y selecciona una de las opciones del desplegable Grupo Administrativo asociado y hace clic en el botón Siguiete.
8. Sistema: Crea el grupo y lo almacena en el Directorio activo

### Extensiones (cursos alternativos)

Paso 3: El agente del CAU ha introducido un nombre de grupo que ya existe en la red

- Sistema: Devuelve un mensaje de error y regresa al inicio del caso de uso.
- Paso 7: El agente del CAU deja vacío el campo alias o no selecciona ninguna de las opciones del desplegable Grupo Administrativo Asociado.
- Sistema: Devuelve un mensaje de error y no cambia la siguiente interfaz.

Figura 4.4 CrearGrupo1

Figura 4.5 CrearGrupo2

## 10. Caso de Uso: Crear Equipo

**Actores:** Agente del CAU

**Resumen:** Este caso de uso se da cuando el agente del CAU se encuentra dentro del caso de uso “Crear Objeto”. Permite crear un equipo, para ello se deberán introducir varios datos, como el nombre del grupo y de manera opcional su GUID y su administrador.

**Precondición:**

**Postcondición:** Se da de alta a un equipo dentro del directorio activo.

### Escenario principal (curso normal de eventos)

1. Agente del CAU: El agente quiere crear un equipo
2. Sistema: Le muestra en la interfaz CrearEquipo1 (Figura 4.6) en la que se establece el nombre que se le quiere asignar al equipo (Campo Nombre del Equipo) y se puede seleccionar un usuario o grupo que lo pueda insertar en otro dominio (Campo Usuario o Grupo).
3. Agente del CAU: Introduce el nombre del equipo. Si quiere cambiar el usuario o grupo que lo puede insertar en otro dominio, hace clic en el botón cambiar y el sistema extiende el caso de uso buscar objeto (número 2) que se explicará después de este caso de uso y hace clic en el botón Siguiente.
4. Sistema: Muestra la interfaz CrearEquipo2 (Figura 4.7) que posee la opción Este es un equipo administrado que el agente del CAU deberá marcar si la equipo que está creando es un equipo administrado.
5. Agente del CAU: Si ha marcado la opción Este es un equipo administrado, entonces, introduce el identificador unitario del equipo en el campo ID unitaria del ordenador (GUID/UUID). Marcada la opción o no hace clic en el botón Siguiente.
6. Sistema: Crea el equipo y lo almacena en el directorio activo.

### Extensiones (cursos alternativos)

Paso 3: El agente del CAU ha introducido un nombre de equipo que ya existe en la red

- Sistema: Devuelve un mensaje de error y no carga la siguiente interfaz.

Paso 5: El agente del CAU deja vacío el campo ID unitaria del ordenador

(GUID/UUID) pese a haber seleccionado la opción Este es un ordenador administrado.

- Sistema: Devuelve un mensaje de error y no cambia la siguiente interfaz.

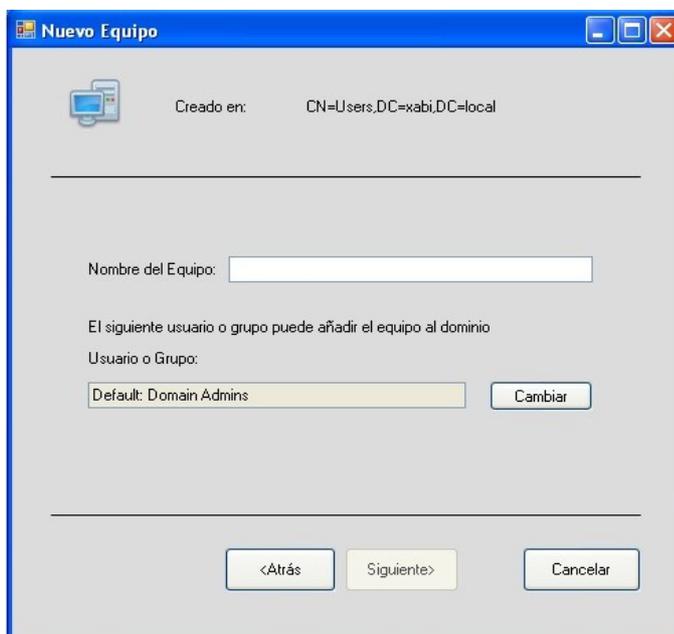


Figura 4.6 CrearEquipo1

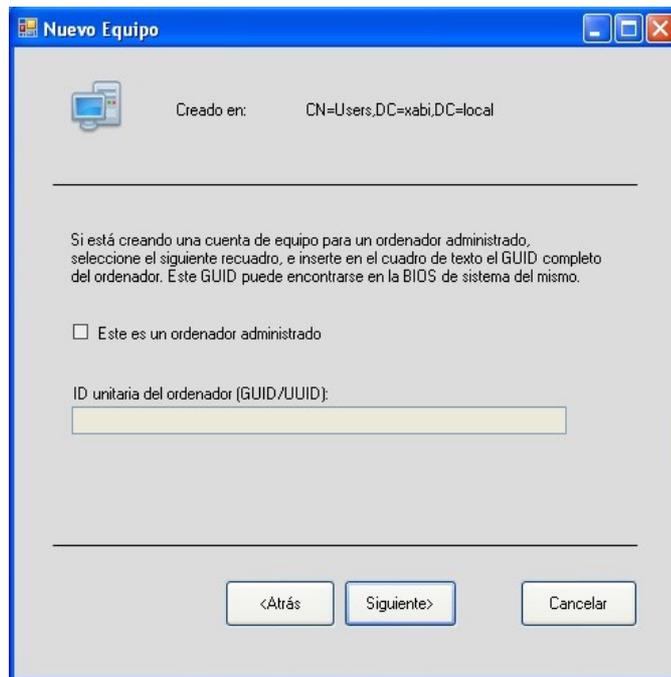


Figura 4.7 CrearEquipo2

## 4.2.2 Casos de uso de la segunda iteración

Los casos de uso que se encargan de gestionar las propiedades de los objetos en el directorio activo

### 4.2.2.1 objeto de tipo usuario

#### 17. Caso de Uso: Gestionar propiedades de un usuario

Actores: Agente del CAU

Resumen: Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso "Gestionar propiedades de objeto" y permite consultar y modificar los atributos de un objeto del tipo usuario.

Precondición:

Postcondición: Se le asignarán al usuario, las modificaciones realizadas sobre sus atributos.

#### **Escenario principal (curso normal de eventos)**

1. Agente del CAU: El agente quiere gestionar propiedades un usuario del directorio activo
2. Sistema: Le muestra la interfaz PropiedadesUsuario Inicial (Figura 4.22) y extiende el caso de uso Gestionar propiedades generales de Usuario (Número 22).
3. Agente del CAU:
  - Si quiere modificar las propiedades de la cuenta de usuario, hace clic en la pestaña Account de la barra de menú izquierda.
  - Si quiere modificar las propiedades de perfil de usuario, hace clic en la pestaña Profile de la barra de menú izquierda.
  - Si quiere modificar las propiedades del Buzón Exchange del usuario, hace clic en la pestaña Exchange de la barra de menú izquierda.
  - Si quiere modificar los grupos de los que es miembro el usuario, hace clic en la

pestaña Member of de la barra de menú izquierda.

- Si quiere consultar los datos canónicos del usuario, hace clic en la pestaña Object de la barra de menú izquierda.
- Si quiere gestionar la seguridad del objeto del usuario, hace clic en la pestaña Account de la barra de menú izquierda.

4. Sistema:

- Si el Agente del CAU ha hecho clic en la pestaña Account, extiende el caso de uso Gestionar propiedades de la cuenta de usuario (Número 24).
- Si el Agente del CAU ha hecho clic en la pestaña Profile, extiende el caso de uso Gestionar propiedades de perfil de usuario (Número 27).
- Si el Agente del CAU ha hecho clic en la pestaña Exchange, extiende el caso de uso Gestionar propiedades del Buzón Exchange de usuario (Número 29).
- Si el Agente del CAU ha hecho clic en la pestaña Member of, extiende el caso de uso Gestionar grupos de los que es miembro el objeto (Número 21).
- Si el Agente del CAU ha hecho clic en la pestaña Object extiende el caso de uso Consultar datos canónicos del Objeto (Número 20).
- Si el Agente del CAU ha hecho clic en la pestaña Security, extiende el caso de uso Gestionar Seguridad de usuario (Número 28).

5. Agente del CAU: Hace clic en el botón aceptar o aplicar.

6. Sistema: Realiza los cambios que ha indicado el Agente del CU sobre el usuario y, en el caso de que éste hubiese hecho clic en el botón aceptar, cierra la ventana.

### Extensiones (cursos alternativos)

Paso 5: Al modificar las propiedades generales, el Agente del CAU intenta cambiar el nombre del usuario por uno que ya existe en el dominio.

- Sistema: Devuelve un mensaje de error y no cambia el nombre del objeto.

Paso 5: Al modificar alguna propiedad, el Agente del CAU no le da al nuevo valor el formato adecuado.

- Sistema: Devuelve un mensaje de error y no realiza ese cambio.

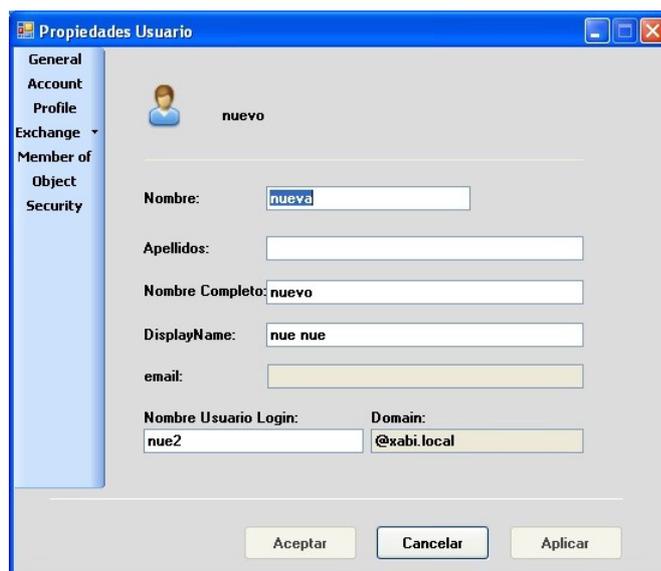


Figura 4.22 PropiedadesUsuario Inicial

## 22. Caso de Uso: Gestionar Propiedades generales de Usuario

Actores: Agente del CAU

Resumen: Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso “Gestionar propiedades usuario” y permite modificar los atributos que asignan el nombre, apellidos, nombre de login etc. del usuario.

Precondición:

Postcondición:

### Escenario principal (curso normal de eventos)

1. Agente del CAU: El agente quiere gestionar opciones generales del usuario.
2. Sistema: Le muestra la interfaz PropiedadesUsuario General (Figura 4.23) En el que están los campos Nombre, Apellidos, Nombre Completo, DisplayName y Nombre Usuario Login que podrán ser modificados.
3. Agente del CAU: Realiza los cambios que considere oportunos sobre los campos que se pueden modificar. Y si quiere que se hagan efectivos hace clic en aceptar o aplicar.
4. Sistema: Lleva a cabo los cambios sobre el usuario.

### Extensiones (cursos alternativos)

Paso 5: Al modificar alguno de los campos, el Agente del CAU intenta escribir mas caracteres de los permitidos.

- Sistema: No permite al Agente del CAU escribir más caracteres de los permitidos.

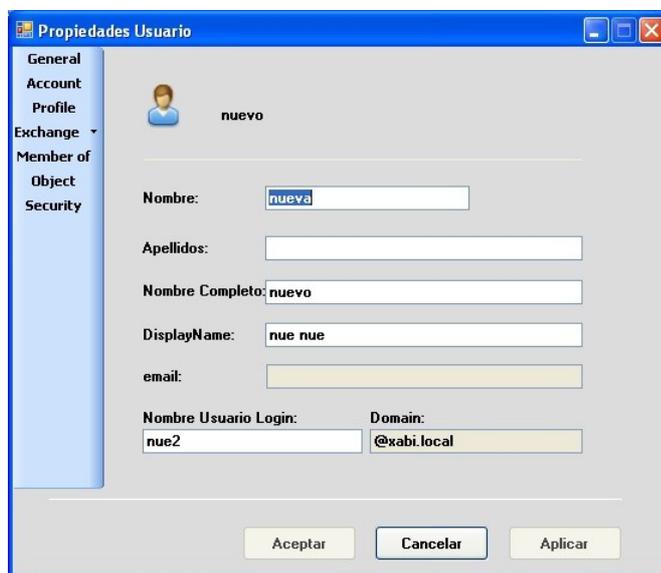
The image shows a Windows-style dialog box titled "Propiedades Usuario". On the left is a vertical navigation pane with tabs: "General", "Account", "Profile", "Exchange", "Member of", "Object", and "Security". The "General" tab is selected. The main area contains a user icon and the name "nuevo". Below are several text input fields: "Nombre:" with "nueva" entered, "Apellidos:" (empty), "Nombre Completo:" with "nuevo" entered, "DisplayName:" with "nue nue" entered, "email:" (empty), "Nombre Usuario Login:" with "nue2" entered, and "Domain:" with "@xabi.local" entered. At the bottom are three buttons: "Aceptar", "Cancelar", and "Aplicar".

Figura 4.23 PropiedadesUsuario General

## 24. Caso de Uso: Gestionar propiedades de la cuenta usuario

Actores: Agente del CAU

Resumen: Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso “Gestionar propiedades de la cuenta de un usuario” y usa otro dos, Gestionar opciones de password usuario (Número 25) que permite establecer las opciones de su

password y Gestionar fecha en la que expira la cuenta de usuario (Número 26) que permite asignar cuando expirará su cuenta.

Precondición:

Postcondición:

### Escenario principal (curso normal de eventos)

1. Agente del CAU: El agente quiere gestionar opciones de password de un usuario del directorio activo
2. Sistema: Carga la interfaz Propiedades usuario Account (Figura 4.24) y sus datos correspondientes. La interfaz se puede dividir en dos partes.
  - Una que posee unas opciones agrupadas en “Opciones de la cuenta” y que se carga su estado mediante el caso de uso Gestionar opciones de password usuario (Número 25).
  - Y otra parte que es la de los radio button y el desplegable del grupo “La cuenta expira” que se carga su estado mediante el caso de uso Gestionar fecha en la que expira la cuenta de usuario (Número 26).
3. Agente del CAU: Realiza las modificaciones que considere oportunas en los datos de la interfaz. Y si quiere que se hagan efectivos hace clic en aceptar o aplicar.
4. Sistema: Lleva a cabo los cambios sobre el usuario.

### Extensiones (cursos alternativos)

Paso 3: El agente del CAU marca opciones incompatibles entre sí, por ejemplo, que el usuario tenga que cambiar el password en el siguiente login y que el usuario no pueda cambiar el password

- Sistema: Devuelve un mensaje de error, y deselecciona una de las opciones.

Paso 3: El agente del CAU intenta seleccionar una fecha, para que expire la cuenta, en el desplegable anterior al día de hoy.

- Sistema: Devuelve un mensaje de error y no permite la selección.

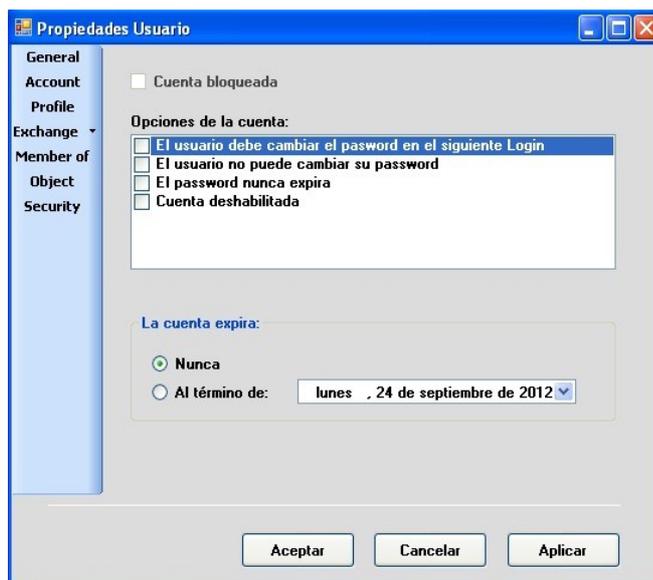


Figura 4.24 PropiedadesUsuario Account

## 27. Caso de Uso: Gestionar Propiedades de perfil de Usuario

Actores: Agente del CAU

Resumen: Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso “Gestionar propiedades usuario” y permite modificar las direcciones y el equipo en las que se encontrarán el perfil de usuario, su script de login y el poseedor de la Home.

Precondición:

Postcondición:

### Escenario principal (curso normal de eventos)

1. Agente del CAU: El agente quiere gestionar las propiedades de perfil del usuario.
2. Sistema: Le muestra la interfaz PropiedadesUsuario Profile (Figura 4.25) En la que hay dos grupos llamados Perfil usuario y poseedor de la home. En el primero se incluyen los campos Dirección del perfil y Logon script y en el segundo Dirección local y conectar.
3. Agente del CAU: Realiza los cambios que considere oportunos escribiendo las direcciones que considere oportunas sobre los campos del primer grupo y seleccionando uno de los radio buttons y rellenando su campo del segundo. Y si quiere que se hagan efectivos hace clic en aceptar o aplicar.
4. Sistema: Lleva a cabo los cambios sobre el usuario.

### Extensiones (cursos alternativos)

Paso 5: Al modificar alguno de los campos, el Agente del CAU no escribe las direcciones en el formato adecuado.

- Sistema: Devuelve un mensaje de error.

Paso 5: El Agente del CAU selecciona el radio button Conectar y no selecciona ninguna opción de las del desplegable.

- Sistema: Devuelve un mensaje de error.

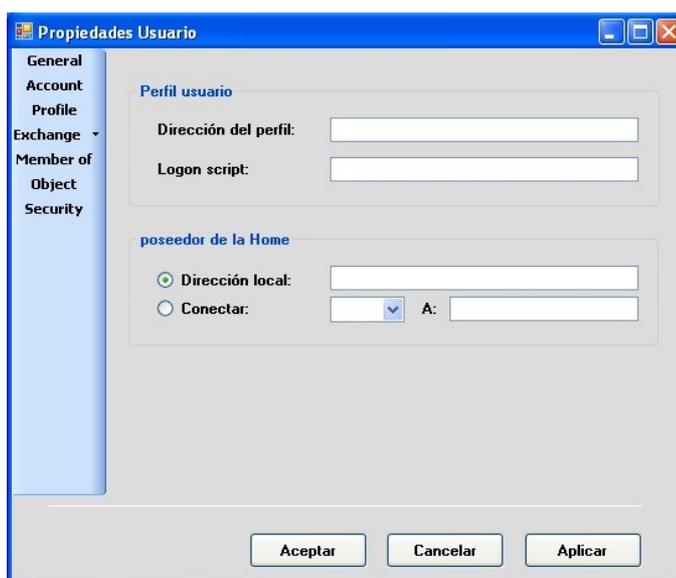


Figura 4.25 PropiedadesUsuario Profile

## 27. Caso de Uso: Gestionar Propiedades Buzón Exchange de Usuario

Actores: Agente del CAU

Resumen: Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso “Gestionar propiedades usuario” y permite modificar los atributos que se encargan de gestionar el buzón Exchange de un usuario.

Precondición: Que el usuario posea un buzón Exchange.

Postcondición:

### Escenario principal (curso normal de eventos)

1. Agente del CAU: El agente quiere gestionar las propiedades de perfil del usuario.
2. Sistema: Le muestra un menú desplegable con cuatro opciones, como se aprecia en la figura 4.26, Opciones generales de Exchange, Direcciones de correo, Características de Exchange y Opciones avanzadas de Exchange.
3. Agente del CAU: Selecciona una de las opciones del menú desplegable.
4. Sistema:
  - Si el usuario ha seleccionado Opciones generales de Exchange, extiende el caso de uso Gestión de características generales del buzón (Número 30).
  - Si el usuario ha seleccionado Direcciones de correo, extiende el caso de uso Gestión de direcciones de correo (Número 38).
  - Si el usuario ha seleccionado Características de Exchange, extiende el caso de uso Gestión de opciones generales de Exchange (Número 41).
  - Si el usuario ha seleccionado Opciones avanzadas de Exchange, extiende el caso de uso Gestión de opciones avanzadas del buzón (Número 44).

### Extensiones (cursos alternativos)



Figura 4.26 PropiedadesUsuario Exchange

### 30. Caso de Uso: Gestionar características generales del Buzón

Actores: Agente del CAU

Resumen: Este caso de uso se da cuando el agente del CAU se encuentra en el caso de

uso “Gestionar propiedades Buzón Exchange usuario” y permite modificar los atributos que se encargan de gestionar las características generales del buzón Exchange de un usuario.

Precondición: Que el usuario posea un buzón Exchange.

Postcondición:

#### **Escenario principal (curso normal de eventos)**

1. Agente del CAU: El agente quiere gestionar las propiedades de perfil del usuario.

2. Sistema: Carga la interfaz Le muestra la interfaz PropiedadesUsuario Exchange Generales (Figura 4.27) en la que habrá dos campos, uno modificable llamado Alias y otro no modificable llamado Mailbox Store en el que aparecerá la dirección en la que se encuentra el buzón del usuario y tres botones, que llamarán a una ventana cada uno para realizar otras gestiones que se explicarán en los casos de uso que se explicarán a continuación de este.

3. Agente del CAU: Puede modificar el campo alias y puede hacer clic en uno de los tres botones.

4. Sistema:

- Si el usuario ha hecho clic sobre el botón Restricciones de Envío, extiende el caso de uso Gestión Restricciones Envío (Número 31).
- Si el usuario ha hecho clic sobre el botón Opciones de Envío, extiende el caso de uso Gestión opciones Envío (Número 34).
- Si el usuario ha hecho clic sobre el botón Límites de Almacenamiento, extiende el caso de uso Gestión Límites Almacenamiento (Número 35).

5. Agente del CAU: Si quiere cambiar el alias del buzón de usuario hace clic en aceptar o aplicar

6. Sistema: Cambia el alias del buzón.

#### **Extensiones (cursos alternativos)**

Paso 3: El agente del CAU intenta establecer como alias del grupo una cadena de caracteres demasiado larga.

- Sistema: No le permite escribir más del límite de caracteres establecido.

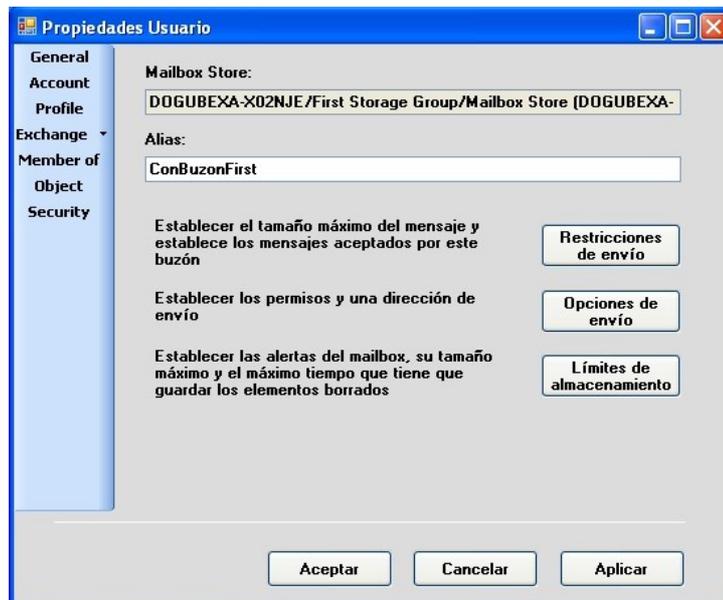


Figura 4.27 PropiedadesUsuario Exchange Generales

### 31. Caso de Uso: Gestionar Restricciones Envío

**Actores:** Agente del CAU

**Resumen:** Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso “Gestionar características generales de Buzón” y usa otro dos, Gestión del tamaño de mensaje (Número 32) que permite establecer los tamaños máximos de mensajes tanto de salida como de entrada del buzón y Gestión de emisores permitidos (Número 33) que permite asignar de qué usuarios se pueden recibir mensajes.

**Precondición:** Que el usuario posea un buzón Exchange.

**Postcondición:**

#### **Escenario principal (curso normal de eventos)**

1. Agente del CAU: El agente quiere gestionar las opciones de restricciones de envío del buzón del usuario.

2. Sistema: Carga la interfaz Restricciones de Envío (Figura 4.28) y sus datos correspondientes.

La interfaz se puede dividir en dos partes, una que posee dos grupos (Tamaño del mensaje de salida y tamaño del mensaje de entrada) con la misma estructura cada uno, dos radio buttons y un campo y que se carga su estado mediante el caso de uso Gestión del tamaño de mensaje (Número 32) y otra parte es la formada por los elementos incluidos en el grupo “Restricciones del mensaje” que se carga su estado mediante el caso de uso Gestión de emisores permitidos (Número 33).

3. Agente del CAU: Para cambiar los tamaños de los mensajes de salida y entrada, selecciona el radio button Máximo en KB y escribe ese máximo en el campo o selecciona el radio button Límite por defecto.

Selecciona la opción “Solo de usuarios autenticados” si es lo que considera oportuno. Para cambiar los emisores permitidos, selecciona uno de los radio button y, si el seleccionado no es “De cualquiera”, hace clic en añadir o eliminar

4. Sistema: Extiende el caso de uso buscar objeto.

5. Agente del CAU: selecciona el usuario a añadir y puede volver a darle a uno de los botones.

Una vez seleccionados todos los usuarios que se quiera y realizados los cambios

oportunos. Puede hacer clic en el botón aceptar.  
6. Sistema: Asigna los cambios al buzón de usuario.

### Extensiones (cursos alternativos)

Paso 3: El agente del CAU intenta escribir un carácter no numérico en uno de los campos.

- Sistema: No escribe nada en el campo.

Paso 3: El agente del CAU no selecciona ningún usuario y hace clic en el botón Eliminar.

- Sistema: Devuelve un mensaje de error.

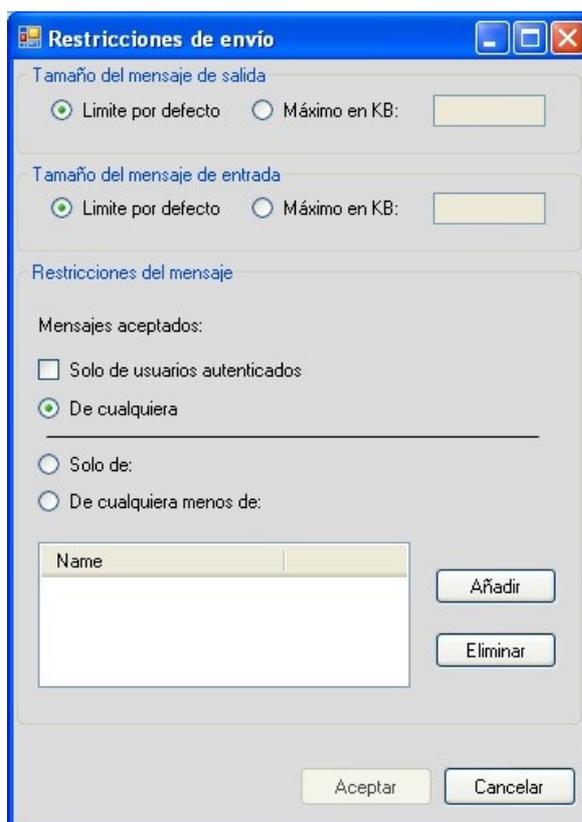


Figura 4.28 Restricciones de Envío

## 34. Caso de Uso: Gestionar Opciones Envío

Actores: Agente del CAU

Resumen: Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso “Gestionar características generales de Buzón” y usa otro tres, Gestión de Envío como (Número 35) que permite establecer a qué usuarios se les permitirá el envío como el usuario dueño del buzón, Gestión de redirección (Número 36) que permite redireccionar a otra dirección de correo los mensajes que lleguen al buzón y Gestión de límite de destinatarios (Número 37) que permite asignar un límite de destinatarios a los que se puede enviar un mensaje.

Precondición: Que el usuario posea un buzón Exchange.

Postcondición:

### **Escenario principal (curso normal de eventos)**

1. Agente del CAU: El agente quiere gestionar las opciones de envío del buzón del usuario.

2. Sistema: Carga la interfaz Opciones de Envío (Figura 4.29) y sus datos correspondientes.

La interfaz se puede dividir en tres partes.

- Una que posee una tabla, en la que se incluirán o no nombres de usuario, junto a dos botones y que carga su estado mediante el caso de uso Gestión de Envío como (Número 35).
- Otra parte es la formada por los elementos incluidos en el grupo “Dirección a la que redireccionar el correo” que carga su estado mediante el caso de uso Gestión de redirección (Número 36).
- Y una última que será la formada por los elementos incluidos en el grupo “Límite de destinatarios” que carga su estado mediante el caso de uso Gestión de límite de destinatarios (Número 37).

3. Agente del CAU: Para añadir un usuario al que darle permisos de envío como el dueño del buzón, hace clic en el botón añadir.

4. Sistema: Extiende el caso de uso buscar objeto.

5. Agente del CAU: selecciona el usuario a añadir y puede volver a realizar la misma operación si quiere.

Para eliminar un usuario que está en la tabla, debe seleccionarlo y hacer clic en eliminar.

Para añadir una dirección de correo a la que redireccionar los mensajes recibidos o modificar la actual en el buzón, selecciona el radio button Redireccionar a: y hacer clic en el botón Modificar.

6. Sistema: Extiende el caso de uso buscar objeto.

7. Agente del CAU: Para eliminar la dirección de correo a la que se redireccionan los mensajes que llegan al buzón, hace clic en el radio button Ninguna.

Para modificar el límite de destinatarios hace clic en el radio button n° Máximo de destinatarios y escribe el número deseado en la casilla contigua o hace clic en el otro radio button.

Si quiere guardar los cambios, hace clic en el botón aceptar.

6. Sistema: Asigna los cambios al buzón de usuario.

### **Extensiones (cursos alternativos)**

Paso 7: El agente del CAU intenta escribir un carácter no numérico en uno de los

campos.

- Sistema: No escribe nada en el campo.

Paso 5: El agente del CAU no selecciona ningún usuario y hace clic en el botón Eliminar.

- Sistema: Devuelve un mensaje de error.

Paso 5: El agente del CAU no selecciona ningún usuario al que redireccionar el correo y selecciona el radio button Redireccionar a.

- Sistema: Devuelve un mensaje de error y no redirecciona el correo a nadie.

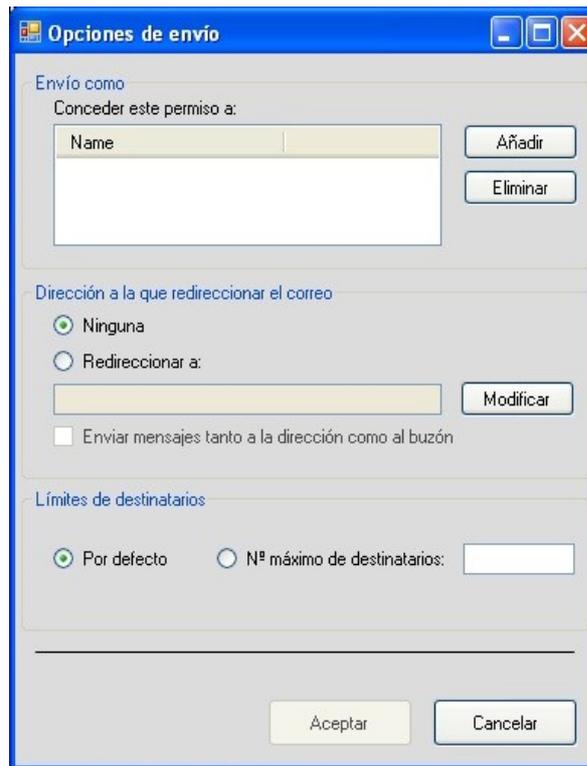


Figura 4.29 Opciones de Envío

?. Caso de Uso: Gestionar de límites de almacenamiento

Actores: Agente del CAU

Resumen: Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso “Gestionar características generales de Buzón” y permite modificar los límites de almacenamiento en KB del buzón del usuario.

Precondición: Que el usuario posea un buzón Exchange.

Postcondición:

#### **Escenario principal (curso normal de eventos)**

1. Agente del CAU: El agente quiere gestionar los límites de almacenamiento del buzón del usuario.
2. Sistema: Le muestra la interfaz Límites Almacenamiento (Figura 4.30) En la que se puede apreciar una opción en la parte superior que se puede marcar y que de no estarlo habilitará el resto de elementos de la interfaz.

3. Agente del CAU: Si quiere establecer algún límite, desmarcará la opción Usar límites por defecto, marcará la opción que quiera modificar y escribirá el número de KB que quiera asignar a ese límite en el campo contiguo.  
Si quiere usar los límites por defecto, marcará la opción de la parte superior.  
Si quiere llevar los cambios a cabo sobre el buzón del usuario, hará clic en el botón aceptar.
4. Sistema: Lleva a cabo los cambios sobre el buzón de usuario.

### Extensiones (cursos alternativos)

Paso 3: El agente del CAU intenta escribir un carácter no numérico en uno de los campos.

- Sistema: No escribe nada en el campo.

Paso 3: El agente del CAU hace clic en aceptar después de marcar una de las opciones de los límites y dejar el campo en blanco.

- Sistema: Devuelve un mensaje de error y no realiza el cambio.

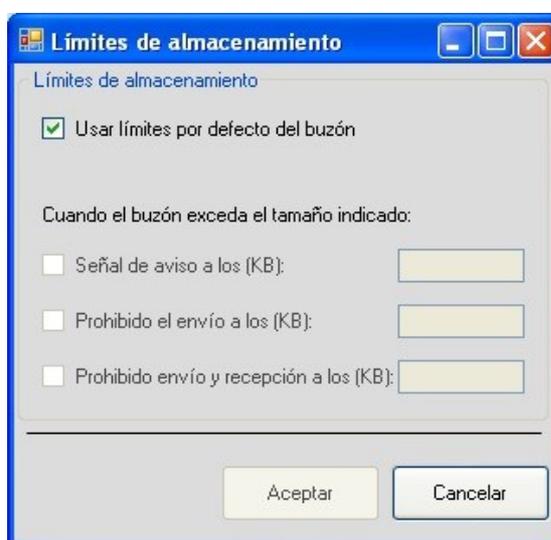


Figura 4.30 Límites Almacenamiento

### 38. Caso de Uso: Gestión de direcciones de correo.

Actores: Agente del CAU

Resumen: Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso “Gestionar propiedades Buzón Exchange usuario” y usa otro dos.

- Gestión Nuevo/Eliminar/Modificar mail (Número 39) que se encarga de añadir, modificar o eliminar una dirección de correo asociada al buzón del usuario.
- Y Establecer mail como primario (Número 40) para establecer una de las direcciones de correo asociadas al buzón como primaria.

Precondición: Que el usuario posea un buzón Exchange.

Postcondición:

### Escenario principal (curso normal de eventos)

1. Agente del CAU: El agente quiere gestionar las direcciones de correo del usuario.

2. Sistema: Carga la interfaz Le muestra la interfaz PropiedadesUsuario Exchange Direcciones de correo (Figura 4.31) en la que habrá una tabla con una o más direcciones de correo y cuatro botones en su parte inferior.
3. Agente del CAU: Puede hacer clic en el botón Nuevo.
4. Sistema: cargará una ventana con un campo de texto.
5. Agente del CAU: Escribe la nueva dirección de correo.
6. Sistema: La añade a la tabla.
7. Agente del CAU: Puede seleccionar una de las direcciones de correo y hacer clic en el botón editar.
8. Sistema: cargará una ventana con un campo de texto.
9. Agente del CAU: Escribe la nueva dirección de correo.
10. Sistema: modifica la dirección seleccionada por la que ha escrito el Agente del CAU.
11. Agente del CAU: Puede seleccionar una de las direcciones y hacer clic en el botón eliminar.
12. Sistema: Eliminará la dirección de correo de entre las de la tabla.
13. Agente del CAU: Selecciona una de las direcciones y hace clic en Establecer como primaria.
14. Sistema: Cambia el color de las direcciones a negro dejando en azul la dirección seleccionada.

### Extensiones (cursos alternativos)

Pasos 5 y 9: El agente del CAU no escribe una dirección de correo con el formato adecuado.

- Sistema: Devuelve un mensaje de error y no realiza ningún cambio en la tabla.

Pasos 7 y 13: El agente del CAU hace clic en editar o eliminar sin haber seleccionado ninguna dirección.

- Sistema: Devuelve un mensaje de error y no realiza ningún cambio en la tabla.

Paso 13: El agente del CAU hace clic en eliminar teniendo seleccionada una dirección primaria (en azul).

- Sistema: Devuelve un mensaje de error y no realiza ningún cambio en la tabla.



Figura 4.31 PropiedadesUsuario Exchange Direcciones de correo

#### 41. Caso de Uso: Gestión de opciones generales de Exchange.

Actores: Agente del CAU

Resumen: Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso “Gestionar propiedades Buzón Exchange usuario” y usa otro dos, Habilitar/Desabilitar Servicios móviles (Número 42) que habilitará o deshabilitará servicios móviles asignados al buzón y Habilitar/Desabilitar Protocolos (Número 43) que habilitará o deshabilitará protocolos asignados al buzón.

Precondición: Que el usuario posea un buzón Exchange.

Postcondición:

#### **Escenario principal (curso normal de eventos)**

1. Agente del CAU: El agente quiere gestionar las opciones generales de Exchange del usuario.

2. Sistema: Carga la interfaz Le muestra la interfaz PropiedadesUsuario Exchange Opciones Generales (Figura 4.32) en la que habrá una tabla con dos grupos de tres elementos cada uno, y dos etiquetas sobre las que se puede hacer clic justo debajo de ella.

El estado de los elementos de cada grupo se obtiene mediante los casos de uso:

- Habilitar/Desabilitar Servicios móviles (Número 42) para el primero.
- Habilitar/Desabilitar Protocolos (Número 43) para el segundo.

3. Agente del CAU: Selecciona el elemento que desee habilitar o deshabilitar y hace clic sobre una de las etiquetas de la parte inferior en función de lo que quiera hacer.

4. Sistema: Cambia el status del elemento

5. Agente del CAU: Repite el proceso hasta realizar todos los cambios de status que considere oportunos. Y si quiere asignar esos status a los servicios y los protocolos asociados al buzón del usuario, hace clic en el botón aceptar.

6. Sistema: Asigna los status a los servicios y protocolos asociados al buzón del usuario.

#### **Extensiones (cursos alternativos)**

Paso 3 y 5: El agente del CAU habilita el servicio Up-To-Date Notifications estando deshabilitado el servicio User Initiated Synchronization del que es dependiente el primero.

- Sistema: Habilita los dos.

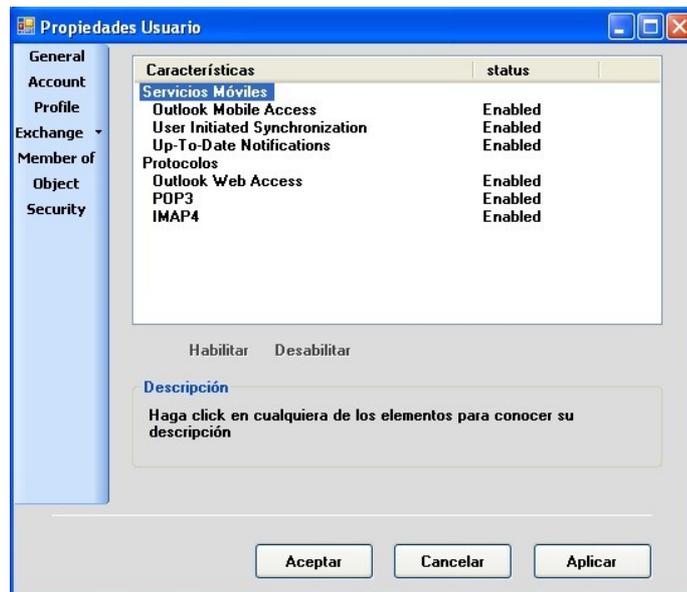


Figura 4.32 PropiedadesUsuario Exchange Opciones Generales

#### 41. Caso de Uso: Gestión de opciones avanzadas de Exchange.

**Actores:** Agente del CAU

**Resumen:** Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso “Gestionar propiedades Buzón Exchange usuario” y usa otro dos, Cambiar nombre Simple Display (Número 45) que permite establecer el Simple Display del buzón y Gestión opciones ocultado y prioridad (Número 46) que gestiona opciones de prioridad de correos tipo x400 y ocultado de entre las direcciones Exchange. Además, extiende el caso de uso Gestionar Seguridad del buzón (Número 47) que se encargará de gestionar los permisos del buzón.

**Precondición:** Que el usuario posea un buzón Exchange.

**Postcondición:**

#### **Escenario principal (curso normal de eventos)**

1. Agente del CAU: El agente quiere gestionar las opciones avanzadas de Exchange del usuario.
2. Sistema: Carga la interfaz PropiedadesUsuario Exchange Opciones Avanzadas (Figura 4.33) en la que habrá un campo con el Simple Display del buzón, dos opciones; una para establecer si tienen prioridad de recepción los mensajes de tipo x400 y si las direcciones de correo asignadas al buzón se ocultarán de la lista de direcciones Exchange.
  - El Simple Display se obtiene mediante el caso de uso Cambiar nombre Simple Display (Número 45).
  - Y se marcarán las opciones según lo indique el caso de uso Gestión opciones ocultado y prioridad (Número 46).
3. Agente del CAU: Si quiere modifica el campo y las opciones según le parezca. En caso de querer cambiar los permisos concedidos sobre el buzón hace clic en el botón Permisos del buzón
4. Sistema: Extiende el caso de uso Gestionar Seguridad del buzón (Número 47).
5. Agente del CAU: Si quiere guardar los cambios efectuados sobre el campo y las opciones, hace clic en el botón aceptar.

6. Sistema: Realiza los cambios en los atributos del usuario que se encargan de la prioridad y el ocultado para adecuarlos a los cambios marcados por el agente del CAU.

### Extensiones (cursos alternativos)

Paso 3: El agente del CAU escribe un nombre en el campo Simple Display que no tiene el formato adecuado.

- Sistema: devuelve error y no lo modifica.

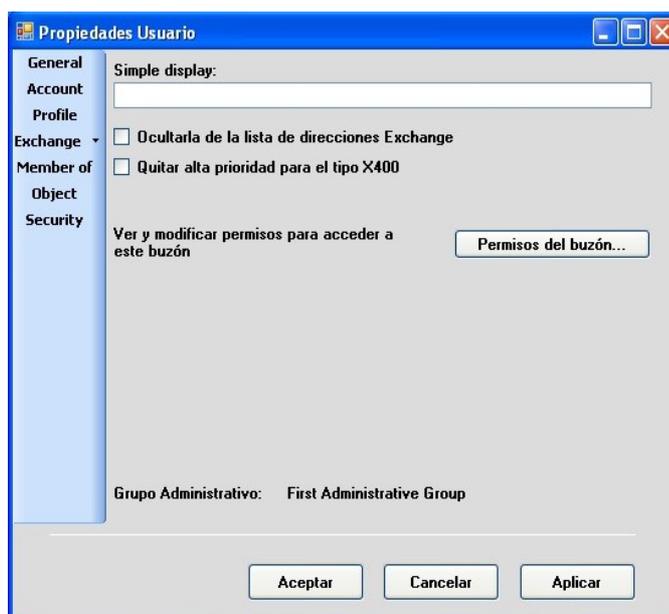


Figura 4.33 PropiedadesUsuario Exchange Opciones Avanzadas

## 47. Caso de Uso: Gestionar Seguridad del buzón

Actores: Agente del CAU

Resumen: Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso "Gestión de opciones avanzadas Exchange" y permite gestionar los permisos que se conceden sobre el buzón, tales como borrado de memoria del buzón, lectura etc.

Precondición: Que el usuario tenga buzón de correo Exchange

Postcondición: Se asignan eliminan o modifican los permisos sobre el buzón

### Escenario principal (curso normal de eventos)

1. Agente del CAU: El agente quiere establecer los permisos sobre el buzón
2. Sistema: Carga la interfaz PropiedadesUsuario Exchange Seguridad Buzón (Figura 4.34) en el que hay dos tablas, en la superior nos encontramos con los objetos que tienen permisos sobre el buzón y en la inferior, los permisos que posee el objeto seleccionado de la primera tabla sobre el buzón. Además hay dos botones para eliminar y añadir objetos con permisos.
3. Agente del CAU: Si quiere añadir permisos a un objeto hace clic en el botón añadir.
4. Sistema: Extiende el caso de uso Buscar de objetos (Número 2).
5. Agente del CAU: Selecciona el objeto a añadir.
6. Sistema: añade el objeto seleccionado a la tabla superior, lo selecciona y actualiza la

tabla inferior desmarcando todos los permisos.

7. Agente del CAU: Si quiere eliminar un objeto con permisos, selecciona el que quiere eliminar y hace clic en el botón eliminar.

8. Sistema: elimina el objeto seleccionado de la tabla superior, actualiza la tabla inferior con los permisos que posea el primero de la tabla superior que a su vez selecciona.

9. Agente del CAU: Si quiere modificar permisos concedidos a un objeto, lo selecciona y los modifica en la tabla inferior.

10. Sistema: Actualiza la tabla inferior.

11. Agente del CAU: Si quiere asignar al buzón los permisos tal y como están en la ventana, hace clic en el botón aceptar.

12. Sistema: Asigna los permisos que se marcan en la ventana como los permisos concedidos sobre el buzón.

### Extensiones (cursos alternativos)

Paso 7 y 9: El agente intenta asignar o eliminar permisos sin seleccionar ningún objeto

- Sistema: Devuelve un mensaje de error.

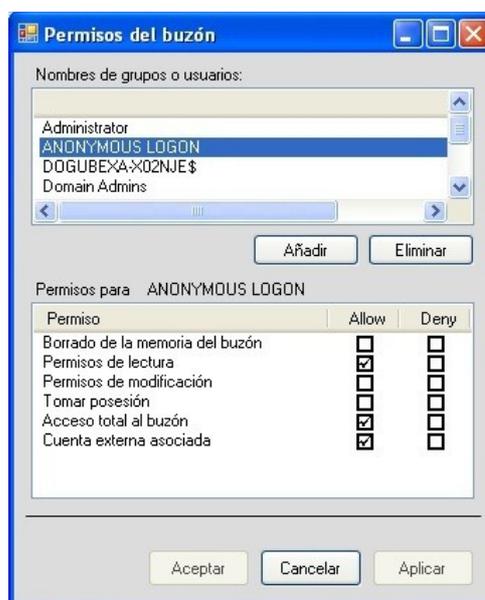


Figura 4.34 PropiedadesUsuario Exchange Seguridad Buzón

### 20. Caso de Uso: Consultar datos canónicos del objeto

Actores: Agente del CAU

Resumen: Este caso de uso se da cuando el agente del CAU se encuentra en los casos de uso “Gestionar propiedades de usuario, grupo y equipo” y permite consultar datos como: nombre canónico, clase de objeto, creado, modificado, USNs ...

Precondición:

Postcondición:

### Escenario principal (curso normal de eventos)

1. Agente del CAU: El agente quiere consultar datos canónicos del objeto
2. Sistema: Devuelve una interfaz PropiedadesUsuario Object (Figura 4.35) con los datos

### Extensiones (cursos alternativos)

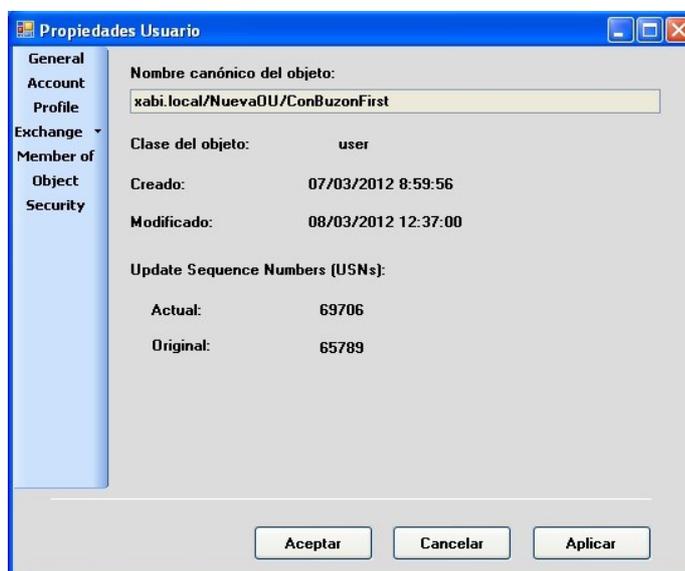


Figura 4.35 PropiedadesUsuario Object

### 21. Caso de Uso: Gestionar grupos de los que es miembro el objeto

Actores: Agente del CAU

Resumen: Este caso de uso se da cuando el agente del CAU se encuentra en los casos de uso “Gestionar propiedades de usuario, grupo y equipo” y permite establecer los grupos de los que es miembro el usuario, grupo o equipo

Precondición:

Postcondición: Se asignará el usuario, equipo o grupo como miembro de los grupos que se establecen en este caso de uso

#### Escenario principal (curso normal de eventos)

1. Agente del CAU: El agente quiere establecer los grupos de los que es miembro el objeto
2. Sistema: Devuelve una interfaz PropiedadesUsuario Miembro de (Figura 4.36) con los grupos de los que es miembro en una tabla y dos botones debajo de ella que permiten añadir o eliminar grupos.
3. Agente del CAU: Si quiere añadir un grupo hace clic en el botón añadir.
4. Sistema: Extiende el caso de uso Buscar Objeto (Número 2).
5. Agente del CAU: Selecciona uno de los grupos.
6. Sistema: Inserta el grupo elegido por el Agente del CAU en la tabla.
7. Agente del CAU: Si quiere eliminar uno de los grupos, lo selecciona y hace clic en el botón eliminar.
8. Sistema: Elimina el grupo de la tabla.
9. Agente del CAU: Si quiere asignar los grupos de la tabla como los grupos de los que

será miembro el objeto, hace clic en aceptar.

10. Sistema: Establece los grupos de la tabla como los grupos de los que es miembro el objeto.

### Extensiones (cursos alternativos)

Paso 3 y 7: Intenta asignar o quitar el objeto como miembro de un grupo sin seleccionar ninguno

- Sistema: Devuelve un mensaje de error y no hace nada.

Paso 3: Intenta asignar un grupo como miembro de otro, pero, por el tipo de los mismos no puede serlo.

- Sistema: Devuelve un mensaje de error y no hace nada.

Paso 7: Intenta quitar el objeto como miembro de su grupo primario. El que está en azul en vez de en negro.

- Sistema: Devuelve un mensaje de error y no hace nada.

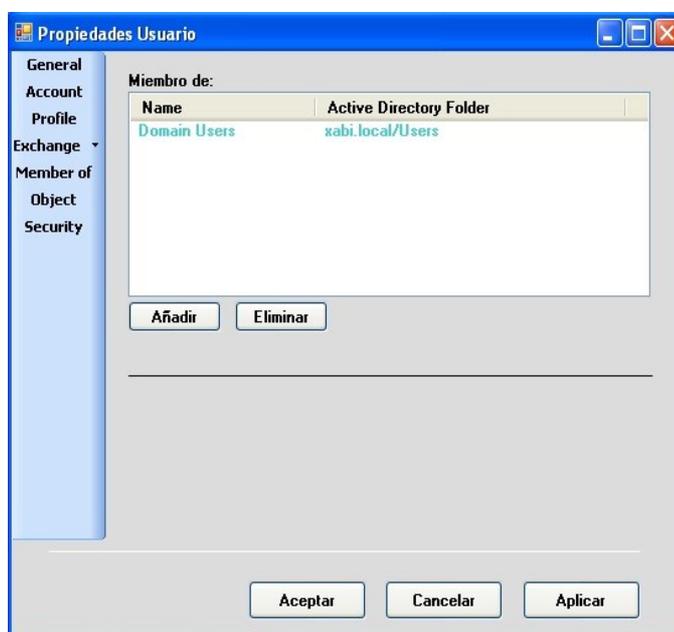


Figura 4.36 PropiedadesUsuario Miembro de

## 28. Caso de Uso: Gestionar Seguridad del usuario

**Actores:** Agente del CAU

**Resumen:** Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso “gestionar propiedades usuario” y permite gestionar veintiocho tipos de permiso que se conceden sobre el usuario, tales como control total, lectura, escritura etc.

**Precondición:**

**Postcondición:** Se asignan eliminan o modifican los permisos sobre el usuario

### Escenario principal (curso normal de eventos)

1. Agente del CAU: El agente quiere establecer los permisos sobre el usuario
2. Sistema: Carga la interfaz PropiedadesUsuario Seguridad (Figura 4.37) en el que hay dos tablas, en la superior nos encontramos con los objetos que tienen permisos

sobre el usuario y en la inferior, los permisos que posee el objeto seleccionado de la primera tabla sobre el usuario. Además hay dos botones para eliminar y añadir objetos con permisos.

3. Agente del CAU: Si quiere añadir permisos a un objeto hace clic en el botón añadir.
4. Sistema: Extiende el caso de uso Buscar de objetos (Número 2).
5. Agente del CAU: Selecciona el objeto a añadir.
6. Sistema: añade el objeto seleccionado a la tabla superior, lo selecciona y actualiza la tabla inferior desmarcando todos los permisos.
7. Agente del CAU: Si quiere eliminar un objeto con permisos, selecciona el que quiere eliminar y hace clic en el botón eliminar.
8. Sistema: elimina el objeto seleccionado de la tabla superior, actualiza la tabla inferior con los permisos que posea el primero de la tabla superior que a su vez selecciona.
9. Agente del CAU: Si quiere modificar permisos concedidos a un objeto, lo seleccionan y los modifica en la tabla inferior.
10. Actualiza la tabla inferior.
11. Agente del CAU: Si quiere asignar al usuario los permisos tal y como están en la ventana, hace clic en el botón aceptar.
12. Sistema: Asigna los permisos que se marcan en la ventana como los permisos concedidos sobre el usuario.

### Extensiones (cursos alternativos)

Paso 7 y 9: El agente intenta asignar o eliminar permisos sin seleccionar ningún objeto

- Sistema: Devuelve un mensaje de error y no asigna o elimina nada.

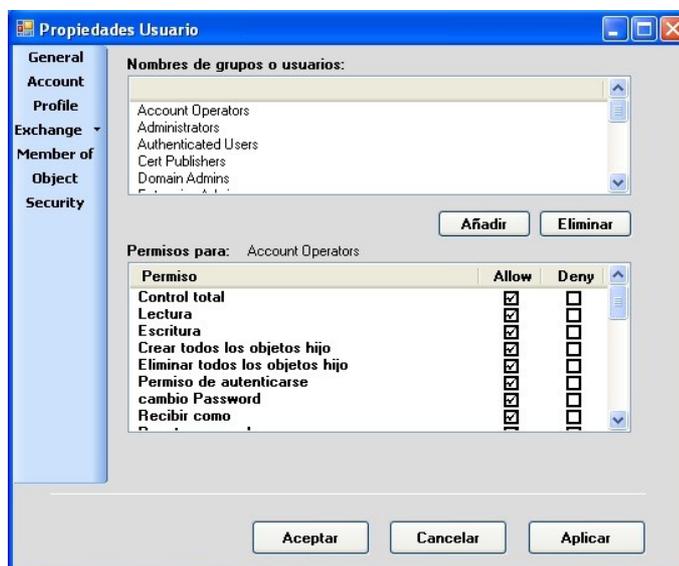


Figura 4.37 PropiedadesUsuario Seguridad

#### 4.2.2.2 objeto de tipo grupo

##### 18. Caso de Uso: Gestionar propiedades de un Grupo

Actores: Agente del CAU

Resumen: Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso “Gestionar propiedades de objeto” y permite consultar y modificar los atributos de un objeto del tipo grupo.

Precondición:

Postcondición: Se le asignarán al grupo, las modificaciones realizadas sobre sus atributos.

##### **Escenario principal (curso normal de eventos)**

1. Agente del CAU: El agente quiere gestionar propiedades un grupo del directorio activo
2. Sistema: Le muestra la interfaz PropiedadesGrupo Inicial (Figura 4.37) y extiende el caso de uso Gestionar propiedades generales de Grupo (Número 48).
3. Agente del CAU:
  - Si quiere modificar los miembros del grupo, hace clic en la pestaña Members de la barra de menú izquierda.
  - Si quiere modificar los grupos de los que es miembro el grupo, hace clic en la pestaña Member of de la barra de menú izquierda.
  - Si quiere modificar las propiedades de su dirección de la correo Exchange del grupo, hace clic en la pestaña Exchange de la barra de menú izquierda.
  - Si quiere modificar el usuario administrador del grupo, hace clic en la pestaña Managed by de la barra de menú izquierda.
  - Si quiere consultar los datos canónicos del grupo, hace clic en la pestaña Object de la barra de menú izquierda.
  - Si quiere gestionar la seguridad del grupo, hace clic en la pestaña Security de la barra de menú izquierda.
4. Sistema:
  - Si el Agente del CAU ha hecho clic en la pestaña Members, extiende el caso de uso Gestionar miembros del grupo (Número 49).
  - Si el Agente del CAU ha hecho clic en la pestaña Exchange, extiende el caso de uso Gestionar propiedades Exchange del grupo (Número 50).
  - Si el Agente del CAU ha hecho clic en la pestaña Member of, extiende el caso de uso Gestionar grupos de los que es miembro el objeto (Número 21).
  - Si el Agente del CAU ha hecho clic en la pestaña Managed By, extiende el caso de uso Gestionar datos del administrador del grupo (Número 58).
  - Si el Agente del CAU ha hecho clic en la pestaña Object extiende el caso de uso Consultar datos canónicos del Objeto (Número 20).
  - Si el Agente del CAU ha hecho clic en la pestaña Security, extiende el caso de uso Gestionar Seguridad de grupo (Número 57).
5. Agente del CAU: Hace clic en el botón aceptar o aplicar.
6. Sistema: Realiza los cambios que ha indicado el Agente del CU sobre el grupo y, en el caso de que éste hubiese hecho clic en el botón aceptar, cierra la ventana.

##### **Extensiones (cursos alternativos)**

Paso 5: Al modificar las propiedades generales, el Agente del CAU intenta cambiar el

nombre del grupo por uno que ya existe en el dominio.

- Sistema: Devuelve un mensaje de error y no cambia el nombre del objeto.

Paso 5: Al modificar alguna propiedad, el Agente del CAU no le da al nuevo valor el formato adecuado.

- Sistema: Devuelve un mensaje de error y no realiza ese cambio.

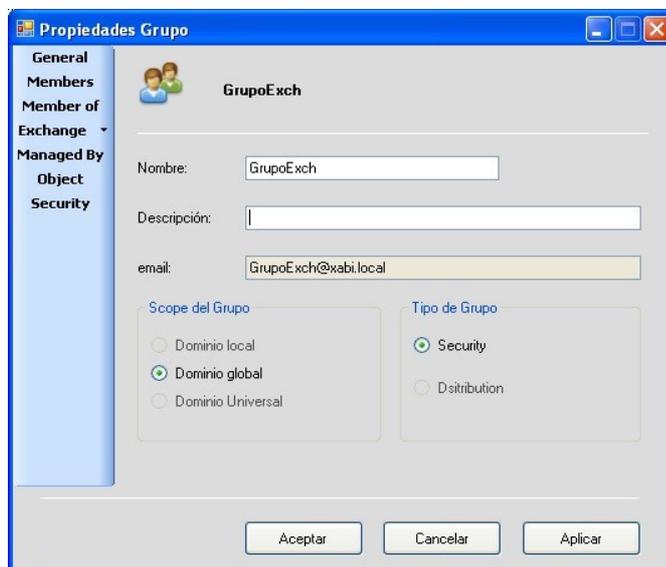


Figura 4.38 PropiedadesGrupo Inicial

Cuando se Extiende el caso de uso Consultar datos canónicos del Objeto (Número 20), la interfaz que se carga es la misma que cuando se extiende en el caso de las propiedades de usuario (Número 17), cambiando solo el menú de la izquierda. Como muestra la figura 4.38b.

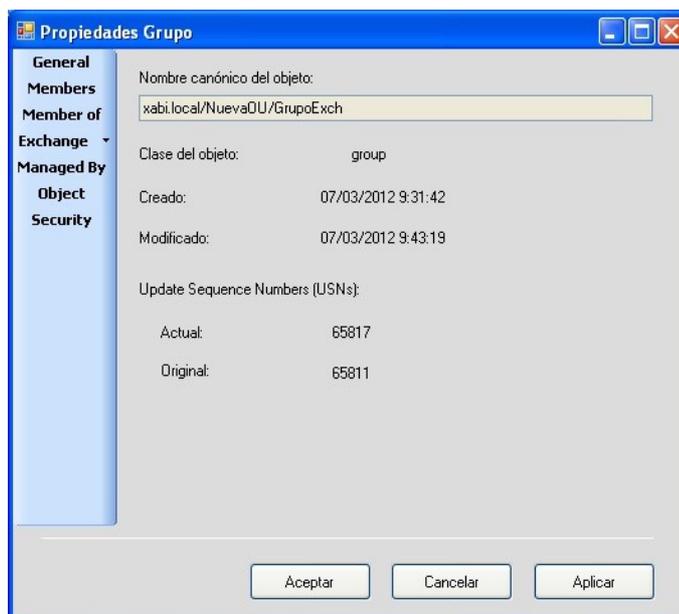


Figura 4.38b PropiedadesGrupo Object

#### 48. Caso de Uso: Gestionar propiedades generales del grupo

Actores: Agente del CAU

Resumen: Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso “Gestionar propiedades grupo” y permite cambiar la descripción del grupo y consultar su email primario y el tipo de grupo.

Precondición:

Postcondición:

#### Escenario principal (curso normal de eventos)

1. Agente del CAU: El agente quiere gestionar propiedades generales del grupo
2. Sistema: Devuelve una interfaz Propiedades Grupo General (Figura 4.39) con los datos Nombre, descripción, email, Scope y tipo. De estos datos el nombre y la descripción se pueden modificar
3. Agente del CAU: Escribe el nuevo nombre o la nueva descripción y hace clic en el botón aceptar.
4. Sistema: Asigna el nuevo nombre y la nueva descripción al grupo

#### Extensiones (cursos alternativos)

Paso 3: El campo donde el agente debe escribir la descripción está vacío

- Sistema: Elimina el valor actual de la descripción del grupo.

Paso 3: El nombre que le quiere asignar al grupo ya existe en el dominio.

- Sistema: Devuelve un mensaje de error y no lo cambia.

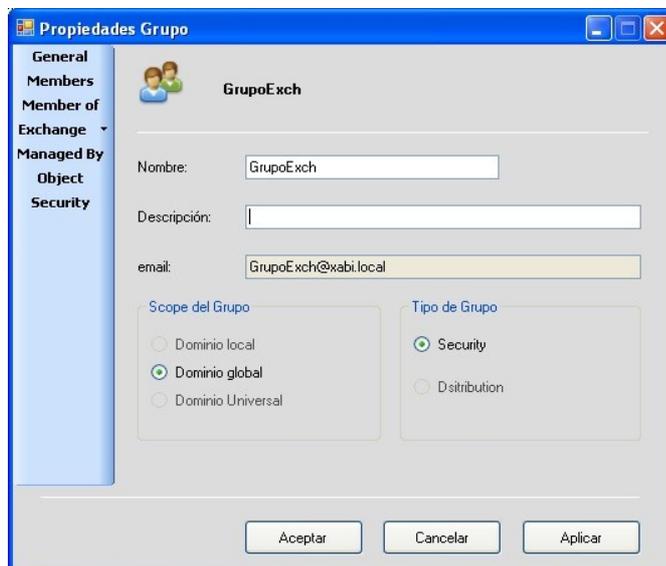


Figura 4.39 PropiedadesGrupo Generales

#### 49. Caso de Uso: Gestionar miembros del grupo

Actores: Agente del CAU

Resumen: Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso “Gestionar propiedades grupo” y permite establecer los miembros de el grupo.

Precondición:

Postcondición: Se asignarán los miembros del grupo

### Escenario principal (curso normal de eventos)

1. Agente del CAU: El agente quiere establecer los miembros de el grupo
2. Sistema: Devuelve la interfaz PropiedadesGrupo Miembros (Figura 4.40) con los miembros del grupo actuales en una tabla y dos botones debajo de ella que permiten añadir o eliminar grupos.
3. Agente del CAU: Si quiere añadir un miembro hace clic en el botón añadir.
4. Sistema: Extiende el caso de uso Buscar Objeto (Número 2).
5. Agente del CAU: Selecciona uno de los objetos de los resultados de la búsqueda.
6. Sistema: Inserta el objeto elegido por el Agente del CAU en la tabla.
7. Agente del CAU: Si quiere eliminar uno de los miembros, lo selecciona y hace clic en el botón eliminar.
8. Sistema: Elimina el grupo de la tabla.
9. Agente del CAU: Si quiere asignar los objetos de la tabla como los miembros del grupo, hace clic en aceptar.
10. Sistema: Establece los objetos de la tabla como los miembros actuales del grupo.

### Extensiones (cursos alternativos)

Paso 3 y 7: Intenta asignar o quitar un objeto como miembro del grupo sin seleccionar ninguno

- Sistema: Devuelve un mensaje de error y no hace nada.

Paso 3: Intenta asignar un grupo como miembro de otro, pero, por el tipo de los mismos no puede serlo.

- Sistema: Devuelve un mensaje de error y no hace nada.

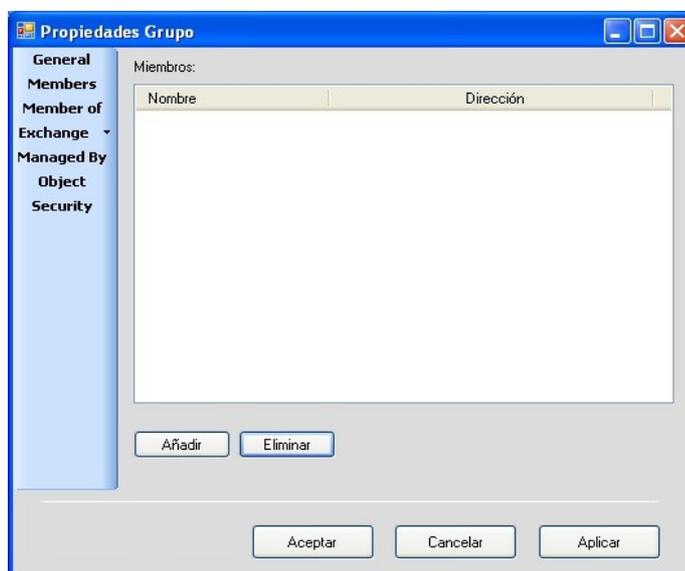


Figura 4.40 PropiedadesGrupo Miembros

50. Caso de Uso: Gestionar Propiedades Exchange del grupo

Actores: Agente del CAU

**Resumen:** Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso “Gestionar propiedades Grupo” y permite modificar los atributos que se encargan de gestionar el correo Exchange de un grupo.

**Precondición:** Que el grupo posea una dirección de correo Exchange.

**Postcondición:**

### Escenario principal (curso normal de eventos)

1. Agente del CAU: El agente quiere gestionar las propiedades Exchange del grupo.
2. Sistema: Le muestra un menú desplegable con tres opciones, como se aprecia en la figura 4.41, Direcciones de correo, Opciones generales y Opciones avanzadas.
3. Agente del CAU: Selecciona una de las opciones del menú desplegable.
4. Sistema:
  - Si el usuario ha seleccionado Direcciones de correo, extiende el caso de uso Gestión de direcciones de correo (Número 38).
  - Si el usuario ha seleccionado Opciones generales, extiende el caso de uso Gestión de características generales Exchange del grupo (Número 51).
  - Si el usuario ha seleccionado Opciones avanzadas de Exchange, extiende el caso de uso Gestión de opciones avanzadas Exchange del grupo (Número 53).

### Extensiones (cursos alternativos)

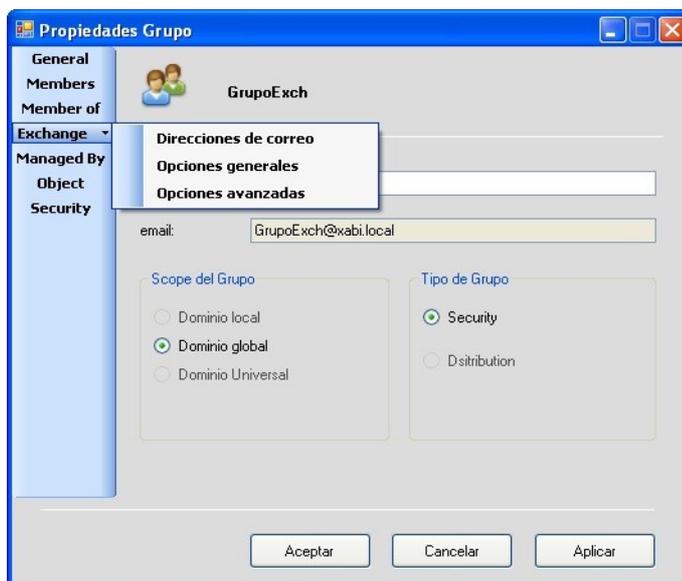


Figura 4.41 PropiedadesGrupo Exchange Desplegable

Cuando se Extiende el caso de uso Gestión de Direcciones de correo, la interfaz que se carga es la misma que cuando se extiende en el caso de las propiedades de usuario, cambiando solo el menú de la izquierda. Como muestra la figura 4.41b.

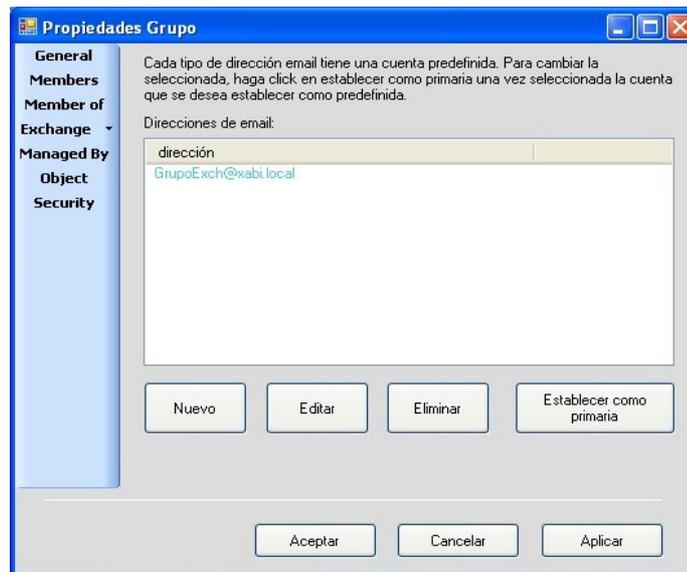


Figura 4.41b PropiedadesGrupo Exchange Direcciones de correo

## 51. Caso de Uso: Gestión de características generales Exchange del grupo

Actores: Agente del CAU

Resumen: Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso “Gestionar propiedades Exchange del Grupo” y usa otro, Gestión de emisores permitidos (Número 33) que permite asignar de qué usuarios se pueden recibir mensajes.

Además permite cambiar el Alias y el Nombre del Display del grupo y establecer el tamaño máximo de mensajes que podrá recibir el grupo.

Precondición: Que el grupo posea una dirección de correo Exchange.

Postcondición:

### Escenario principal (curso normal de eventos)

1. Agente del CAU: El agente quiere gestionar las características generales Exchange del grupo.

2. Sistema: Carga la interfaz Propiedades Grupo Exchange Opciones Generales (Figura 4.32) y sus datos correspondientes.

La interfaz se puede dividir en tres partes, una que posee dos campos (Alias y Nombre del Display). Un grupo formado por dos radio buttons y un campo (Tamaño del mensaje) con el que se puede cambiar el tamaño máximo de mensaje de entrada del grupo y, por último, la parte formada por los elementos incluidos en el grupo “Restricciones del mensaje” que se carga su estado mediante el caso de uso Gestión de emisores permitidos (Número 33).

3. Agente del CAU: Puede modificar los campos Alias y Nombre Display.

Para cambiar el tamaño de los mensajes de entrada, selecciona el radio button Máximo en KB y escribe ese máximo en el campo o selecciona el radio button Límite por defecto.

Selecciona la opción “Solo de usuarios autenticados” si es lo que considera oportuno.

Para cambiar los emisores permitidos, selecciona uno de los radio button y, si el seleccionado no es “De cualquiera”, hace clic en añadir o eliminar

4. Sistema: Extiende el caso de uso buscar objeto.

5. Agente del CAU: selecciona el usuario a añadir y puede volver a darle a uno de los botones.

Una vez seleccionados todos los usuarios que se quiera y realizados los cambios oportunos. Puede hacer clic en el botón aceptar.

6. Sistema: Asigna los cambios al grupo.

### Extensiones (cursos alternativos)

Paso 3: El agente del CAU intenta escribir un carácter no numérico en el campo de la parte de tamaño máximo de mensaje de entrada.

- Sistema: No escribe nada en el campo.

Paso 3: El agente del CAU no selecciona ningún usuario y hace clic en el botón Eliminar.

- Sistema: Devuelve un mensaje de error y no elimina ningún usuario.

Paso 3: El agente del CAU escribe un nombre de display o un alias en los campos de la parte superior que no tiene el formato adecuado.

- Sistema: Devuelve un mensaje de error y no actualiza el nombre o el Nombre de Display.

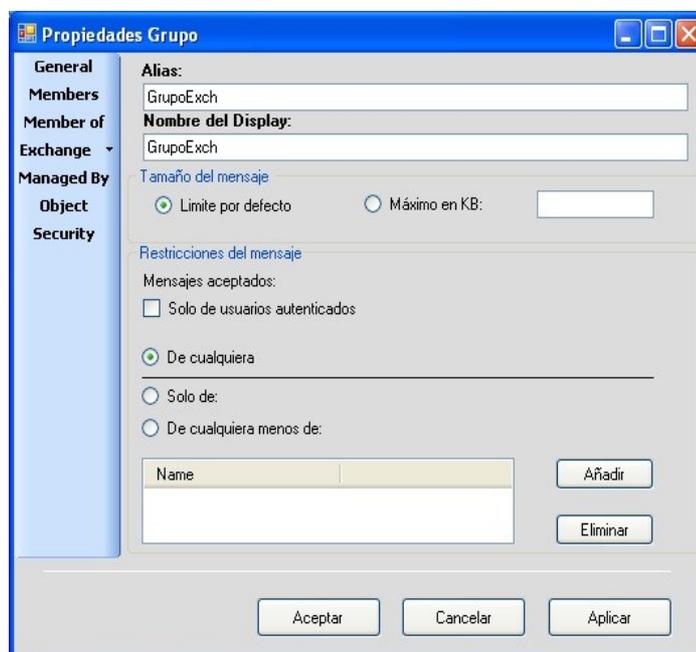


Figura 4.43 PropiedadesGrupo Exchange Opciones Generales

### 43. Caso de Uso: Gestión de opciones avanzadas de Exchange.

**Actores:** Agente del CAU

**Resumen:** Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso “Gestionar propiedades Buzón Exchange grupo” y usa otro tres, Gestión de nombre y servidor (Número 54) que permite establecer el Simple Display y el servidor de expansión para el correo del grupo, Gestión opciones de recepción (Número 55) que permite establecer el mensaje fuera-de-la-oficina y ocultar la dirección de correo para que no aparezca entre las direcciones Exchange y Gestión de envío de informes (Número 56) que se encargará de gestionar a quién se envía el informe de recepción de mensaje.

**Precondición:** Que el grupo posea una dirección de correo Exchange.

## Postcondición:

### **Escenario principal (curso normal de eventos)**

1. Agente del CAU: El agente quiere gestionar las opciones avanzadas de Exchange del grupo.

2. Sistema: Carga la interfaz PropiedadesGrupo Exchange Opciones Avanzadas (Figura 4.44).

- La parte gestionada por el caso de uso Gestión de Nombre y servidor (Número 54) está formada por un campo con el Simple Display (Simple Display) y un desplegable para seleccionar el servidor de expansión que se quiere utilizar (Servidor de Expansión).
- La parte gestionada por el caso de uso Gestión de opciones de recepción (Número 55) está formada por dos opciones, una para establecer si se activa el mensaje fuera-de-la-oficina y otra que para establecer si las direcciones de correo asignadas al grupo se ocultarán de la lista de direcciones Exchange.
- Por último, la parte gestionada por el caso de uso Gestión envío de informes (Número 56) está formada por tres radio buttons con las tres opciones para el envío de informe de recepción, envío al dueño del grupo, al emisor del mensaje o a nadie.

3. Agente del CAU: Si quiere modifica los campo, las opciones y los radio buttons según le parezca.

Si quiere guardar los cambios efectuados sobre el campo y las opciones, hace clic en el botón aceptar.

6. Sistema: Realiza los cambios en los atributos del grupo que correspondan.

### **Extensiones (cursos alternativos)**

Paso 3: El agente del CAU escribe un nombre en el campo Simple Display que no tiene el formato adecuado.

- Sistema: devuelve error y no lo modifica.

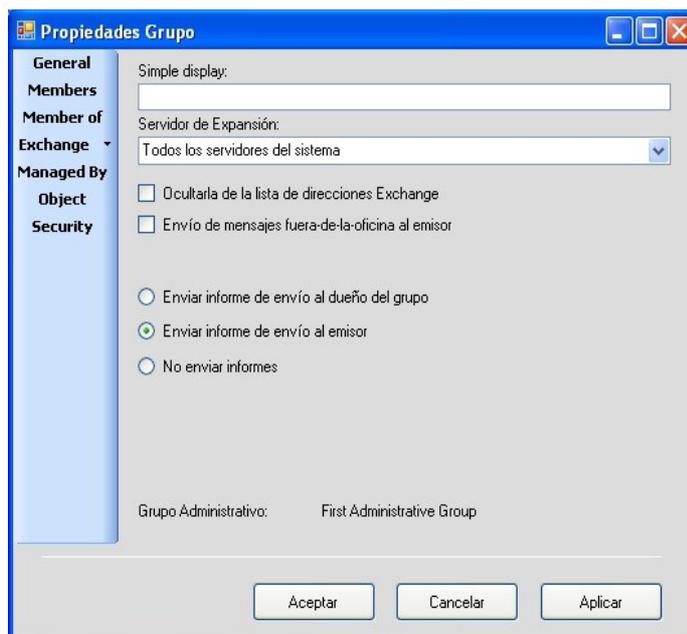


Figura 4.44 PropiedadesGrupo Exchange Opciones Avanzadas

## 58. Caso de Uso: Gestionar datos del Administrador del Grupo

Actores: Agente del CAU

Resumen: Este caso de uso se da cuando el agente del CAU se encuentra en los casos de uso “Gestionar propiedades equipo” y usa otro dos, Gestionar Administrador del Objeto (Número 60) que permite cambiar el administrador del objeto y Dar permiso al Administrador para cambiar miembros del grupo (Número 59) que permite darle permisos para modificar los miembros del grupo.

Precondición:

Postcondición: Se asignarán los cambios al grupo.

### **Escenario principal (curso normal de eventos)**

1. Agente del CAU: El agente quiere gestionar los datos del Administrador del grupo.
2. Sistema: Carga la interfaz PropiedadesGrupo Exchange Opciones Avanzadas (Figura 4.44).
  - La parte gestionada por el caso de uso Gestionar Administrador del Objeto (Número 60) está formada por un campo en el que no se puede escribir con el nombre del administrador actual del grupo (Nombre) y tres botones debajo de él, Cambio, Propiedades, Quitar.
  - La parte gestionada por el caso de uso Dar permiso al Administrador para cambiar miembros del grupo (Número 59) está formada por una opción, Puede cambiar la lista de miembros del grupo.
3. Agente del CAU: Si quiere modificar el administrador del grupo, hace clic en el botón Cambio.
4. Sistema: Extiende el caso de uso Buscar Objeto.
5. Agente del CAU: Selecciona un objeto de tipo usuario de entre los resultados.
6. Sistema: Inserta el nombre del usuario en el campo.
7. Agente del CAU: Si quiere ver las propiedades del usuario administrador del grupo, hace clic en el botón Propiedades.
8. Sistema: Extiende el caso de uso Gestionar propiedades Usuario.
9. Agente del CAU: Si quiere quitar el administrador del grupo, hace clic en el botón quitar.
10. Sistema: Elimina el nombre del usuario administrador del campo.
11. Agente del CAU: Si quiere darle permisos al administrador para poder modificar los miembros del grupo, marca la opción Puede cambiar la lista de miembros del grupo, si lo que quiere es quitárselos la desmarca.  
Si quiere asignar los cambios al grupo, hace clic en el botón aceptar.
12. Sistema: Asigna los cambios al grupo, cambiando los atributos oportunos.

### **Extensiones (cursos alternativos)**

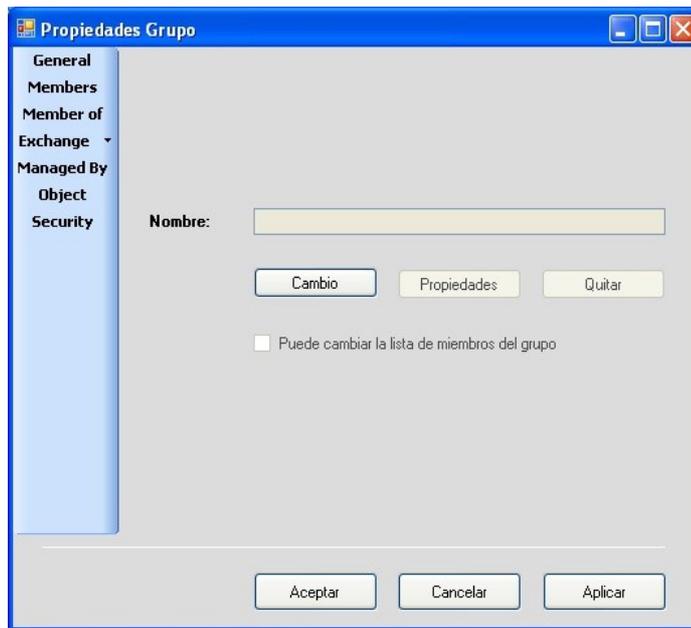


Figura 4.44 PropiedadesGrupo Exchange Opciones Avanzadas

### 57. Caso de Uso: Gestionar Seguridad del grupo

**Actores:** Agente del CAU

**Resumen:** Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso “gestionar propiedades grupo” y permite gestionar diez tipos de permiso que se conceden sobre el grupo, tales como control total, lectura, escritura etc.

**Precondición:**

**Postcondición:** Se asignan eliminan o modifican los permisos sobre el grupo

#### **Escenario principal (curso normal de eventos)**

1. Agente del CAU: El agente quiere establecer los permisos sobre el grupo.
2. Sistema: Carga la interfaz PropiedadesGrupo Seguridad (Figura 4.45) en el que hay dos tablas, en la superior nos encontramos con los objetos que tienen permisos sobre el grupo y en la inferior, los permisos que posee el objeto seleccionado de la primera tabla sobre el grupo. Además hay dos botones para eliminar y añadir objetos con permisos.
3. Agente del CAU: Si quiere añadir permisos a un objeto hace clic en el botón añadir.
4. Sistema: Extiende el caso de uso Buscar de objetos (Número 2).
5. Agente del CAU: Selecciona el objeto a añadir.
6. Sistema: añade el objeto seleccionado a la tabla superior, lo selecciona y actualiza la tabla inferior desmarcando todos los permisos.
7. Agente del CAU: Si quiere eliminar un objeto con permisos, selecciona el que quiere eliminar y hace clic en el botón eliminar.
8. Sistema: elimina el objeto seleccionado de la tabla superior, actualiza la tabla inferior con los permisos que posea el primero de la tabla superior que a su vez selecciona.
9. Agente del CAU: Si quiere modificar permisos concedidos a un objeto, lo seleccionan y los modifica en la tabla inferior.
10. Actualiza la tabla inferior.
11. Agente del CAU: Si quiere asignar al grupo los permisos tal y como están en la ventana, hace clic en el botón aceptar.

12. Sistema: Asigna los permisos que se marcan en la ventana como los permisos concedidos sobre el grupo.

### Extensiones (cursos alternativos)

Paso 7 y 9: El agente intenta asignar o eliminar permisos sin seleccionar ningún objeto

- Sistema: Devuelve un mensaje de error y no asigna o elimina nada.

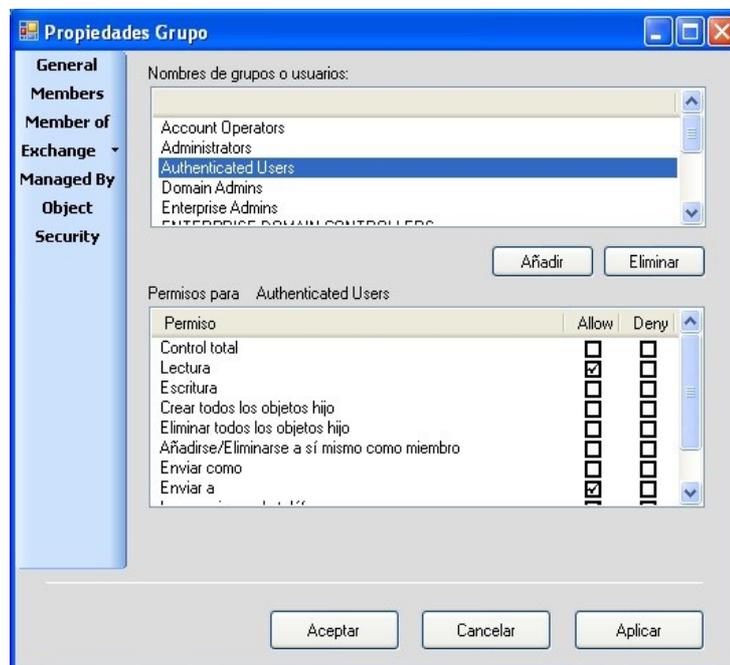


Figura 4.45 PropiedadesGrupo Seguridad

### 4.2.2.3 objeto de tipo equipo

19. Caso de Uso: Gestionar propiedades de un Equipo

Actores: Agente del CAU

Resumen: Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso "Gestionar propiedades de objeto" y permite consultar y modificar los atributos de un objeto del tipo equipo.

Precondición:

Postcondición: Se le asignarán al equipo, las modificaciones realizadas sobre sus atributos.

### Escenario principal (curso normal de eventos)

1. Agente del CAU: El agente quiere gestionar propiedades un equipo del directorio activo
2. Sistema: Le muestra la interfaz Propiedades Equipo Inicial (Figura 4.46) y extiende el caso de uso Gestionar propiedades generales de Equipo (Número 61).
3. Agente del CAU:
  - Si quiere consultar los datos del Sistema Operativo del equipo, hace clic en la pestaña Operating System de la barra de menú izquierda.

- Si quiere modificar los grupos de los que es miembro el grupo, hace clic en la pestaña Member of de la barra de menú izquierda.
- Si quiere modificar el usuario administrador del grupo, hace clic en la pestaña Managed by de la barra de menú izquierda.
- Si quiere consultar los datos canónicos del grupo, hace clic en la pestaña Object de la barra de menú izquierda.
- Si quiere gestionar la seguridad del grupo, hace clic en la pestaña Security de la barra de menú izquierda.

4. Sistema:

- Si el Agente del CAU ha hecho clic en la pestaña Operating System, extiende el caso de uso Consultar Datos del Sistema Operativo (Número 62).
- Si el Agente del CAU ha hecho clic en la pestaña Member of, extiende el caso de uso Gestionar grupos de los que es miembro el objeto (Número 21).
- Si el Agente del CAU ha hecho clic en la pestaña Managed By, extiende el caso de uso Gestionar administrador del objeto (Número 60).
- Si el Agente del CAU ha hecho clic en la pestaña Object extiende el caso de uso Consultar datos canónicos del Objeto (Número 20).
- Si el Agente del CAU ha hecho clic en la pestaña Security, extiende el caso de uso Gestionar Seguridad de equipo (Número 63).

5. Agente del CAU: Hace clic en el botón aceptar o aplicar.

6. Sistema: Realiza los cambios que ha indicado el Agente del CU sobre el equipo y, en el caso de que hubiese hecho clic en el botón aceptar, cierra la ventana.

**Extensiones (cursos alternativos)**

Paso 5: Al modificar las propiedades generales, el Agente del CAU intenta cambiar el nombre del equipo por uno que ya existe en el dominio.

- Sistema: Devuelve un mensaje de error y no cambia el nombre del objeto.

Paso 5: Al modificar alguna propiedad, el Agente del CAU no le da al nuevo valor el formato adecuado.

- Sistema: Devuelve un mensaje de error y no realiza ese cambio.

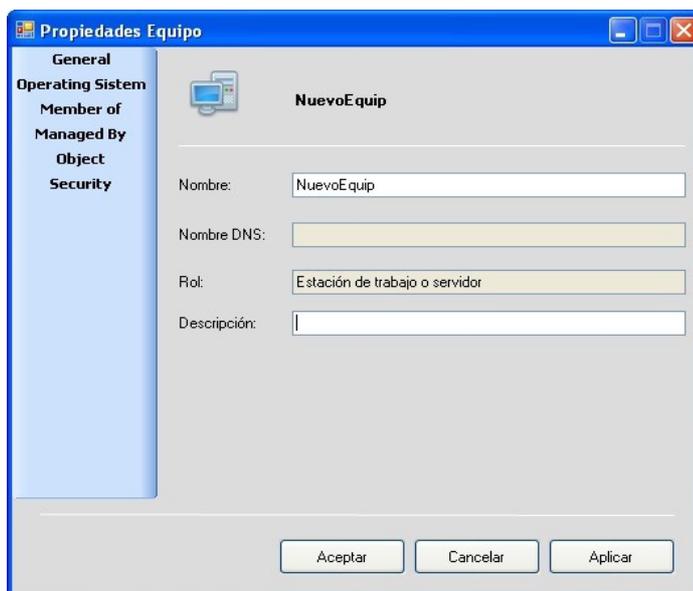


Figura 4.46 PropiedadesEquipo Inicial

Cuando se Extiende el caso de uso Gestionar administrador del objeto (Número 60), la interfaz que se carga es la misma que cuando se extiende en la parte que gestiona del caso de uso Gestionar datos del Administrador del grupo (Número 58) de las propiedades de grupo (Número 18), cambiando solo el menú de la izquierda. Como muestra la figura 4.46b.

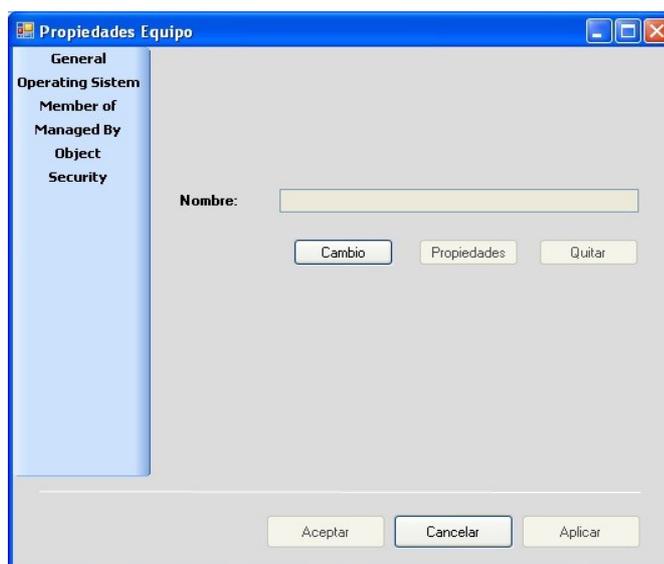


Figura 4.46b PropiedadesEquipo Managed By

Cuando se Extiende el caso de uso Consultar datos canónicos del Objeto (Número 20), la interfaz que se carga es la misma que cuando se extiende en el caso de las propiedades de usuario (Número 17) o en el de las propiedades de grupo (Número 18) cambiando solo el menú de la izquierda. Como muestra la figura 4.46c.

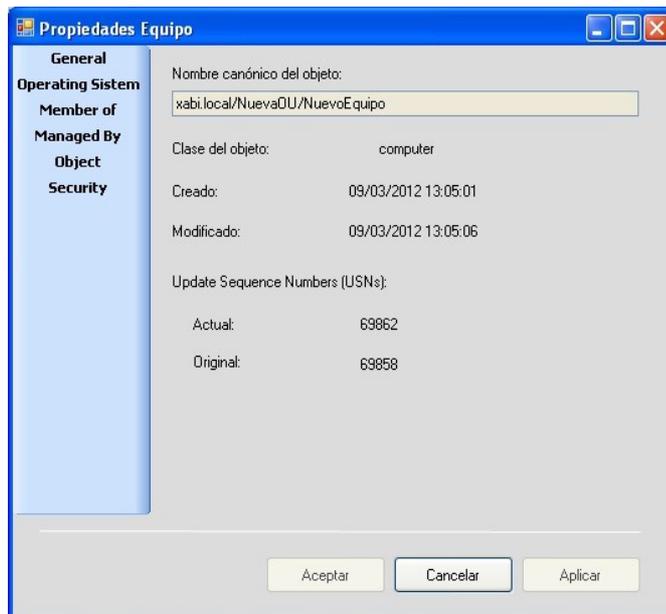


Figura 4.46c PropiedadesEquipo Object

## 61. Caso de Uso: Gestionar propiedades generales de Equipo

Actores: Agente del CAU

Resumen: Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso “gestionar propiedades equipo” y permite cambiar la descripción del grupo y consultar su rol y Nombre DNS

Precondición:

Postcondición:

### Escenario principal (curso normal de eventos)

1. Agente del CAU: El agente quiere gestionar propiedades generales del equipo
2. Sistema: Devuelve una interfaz Propiedades Grupo General (Figura 4.47) con los datos Nombre, NombreDNS, rol y descripción. De estos datos el nombre y la descripción se pueden modificar
3. Agente del CAU: Escribe el nuevo nombre o la nueva descripción y hace clic en el botón aceptar.
4. Sistema: Asigna el nuevo nombre y la nueva descripción al grupo

### Extensiones (cursos alternativos)

Paso 3: El campo donde el agente debe escribir la descripción está vacío

- Sistema: Elimina el valor actual de la descripción del grupo.

Paso 3: El nombre que le quiere asignar al grupo ya existe en el dominio.

- Sistema: Devuelve un mensaje de error y no lo cambia.

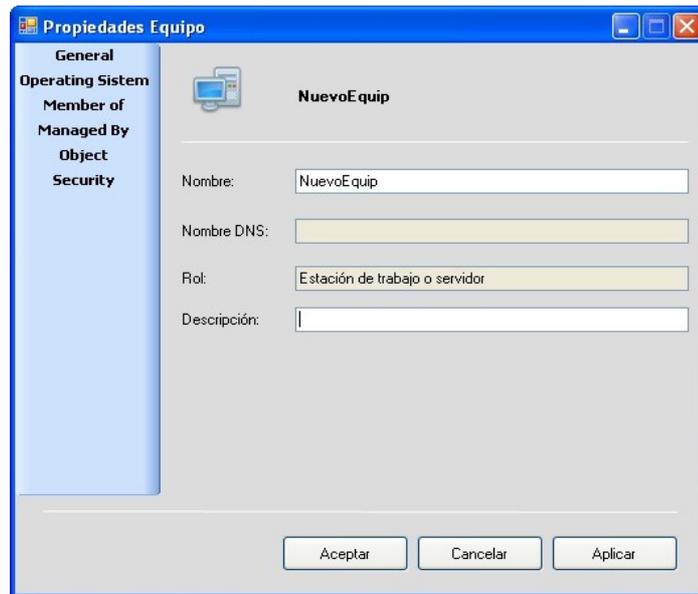


Figura 4.47 PropiedadesEquipo General

## 62. Caso de Uso: Consultar datos del sistema operativo

Actores: Agente del CAU

Resumen: Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso “gestionar propiedades equipo” y permite consultar datos como: nombre del SO, versión y service Pack

Precondición:

Postcondición:

### **Escenario principal (curso normal de eventos)**

1. Agente del CAU: El agente quiere consultar datos de SO del equipo
2. Sistema: Devuelve una interfaz PropiedadesEquipo SO (Figura 4.48) con tres campos con datos que no se pueden modificar.

### **Extensiones (cursos alternativos)**

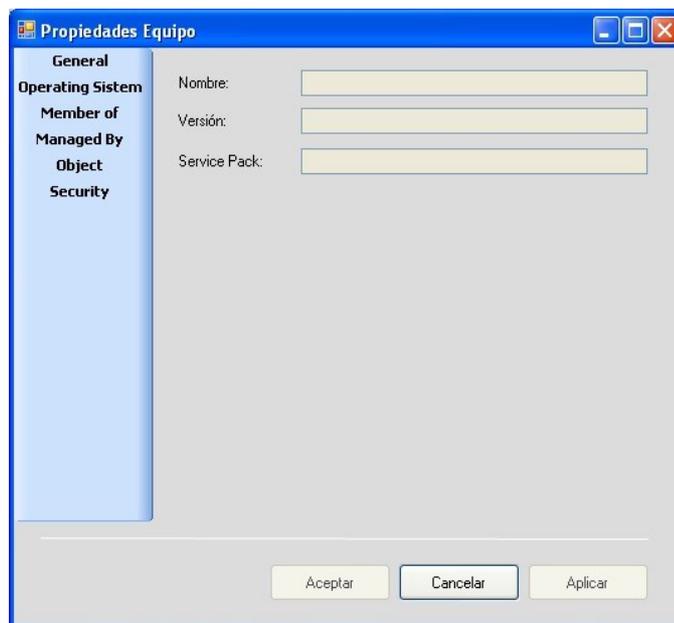


Figura 4.48 PropiedadesEquipo SO

### 63. Caso de Uso: Gestionar Seguridad del equipo

**Actores:** Agente del CAU

**Resumen:** Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso “gestionar propiedades equipo” y permite gestionar veinte tipos de permiso que se conceden sobre el equipo, tales como control total, lectura, escritura etc.

**Precondición:**

**Postcondición:** Se asignan eliminan o modifican los permisos sobre el equipo

#### **Escenario principal (curso normal de eventos)**

1. Agente del CAU: El agente quiere establecer los permisos sobre el equipo.
2. Sistema: Carga la interfaz PropiedadesEquipo Seguridad (Figura 4.49) en el que hay dos tablas, en la superior nos encontramos con los objetos que tienen permisos sobre el equipo y en la inferior, los permisos que posee el objeto seleccionado de la primera tabla sobre el equipo. Además hay dos botones para eliminar y añadir objetos con permisos.
3. Agente del CAU: Si quiere añadir permisos a un objeto hace clic en el botón añadir.
4. Sistema: Extiende el caso de uso Buscar de objetos (Número 2).
5. Agente del CAU: Selecciona el objeto a añadir.
6. Sistema: añade el objeto seleccionado a la tabla superior, lo selecciona y actualiza la tabla inferior desmarcando todos los permisos.
7. Agente del CAU: Si quiere eliminar un objeto con permisos, selecciona el que quiere eliminar y hace clic en el botón eliminar.
8. Sistema: elimina el objeto seleccionado de la tabla superior, actualiza la tabla inferior con los permisos que posea el primero de la tabla superior que a su vez selecciona.
9. Agente del CAU: Si quiere modificar permisos concedidos a un objeto, lo seleccionan y los modifica en la tabla inferior.
10. Actualiza la tabla inferior.
11. Agente del CAU: Si quiere asignar al equipo los permisos tal y como están en la ventana, hace clic en el botón aceptar.
12. Sistema: Asigna los permisos que se marcan en la ventana como los permisos

concedidos sobre el equipo.

### Extensiones (cursos alternativos)

Paso 7 y 9: El agente intenta asignar o eliminar permisos sin seleccionar ningún objeto

- Sistema: Devuelve un mensaje de error y no asigna o elimina nada.

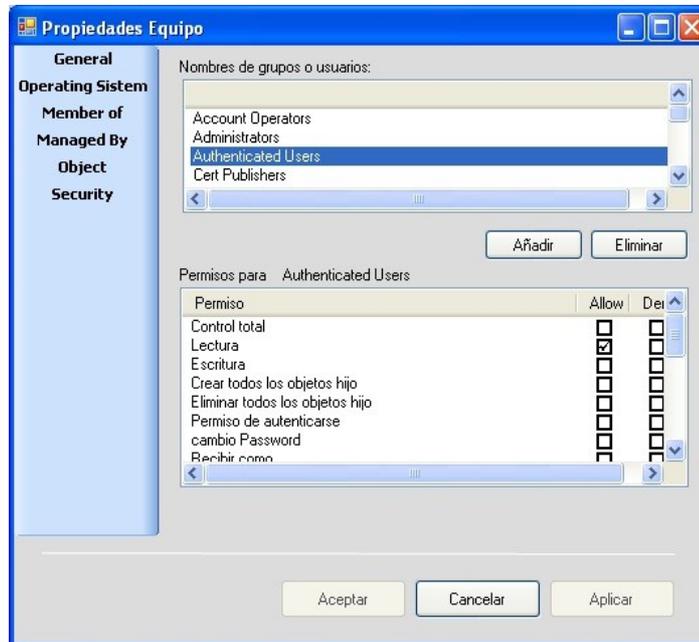


Figura 4.49 PropiedadesEquipo Seguridad

### 4.2.3 Casos de uso de la tercera iteración

Los casos de uso que afectan a los objetos del directorio activo pero que no se encargan de gestionar sus propiedades.

#### 4.2.3.1 Gestión directorio Activo

0. Caso de Uso: Gestionar Directorio Activo

Actores: Agente del CAU

Resumen: Se trata del caso de del que extenderán todos los que se encargan de gestionar el directorio activo.

Precondición:

Postcondición:

#### Escenario principal (curso normal de eventos)

1. Agente del CAU: El agente quiere gestionar el directorio activo.

2. Sistema: Carga la interfaz mostrada en las figuras 4.0a y 4.0b. Esta interfaz se puede dividir en tres partes.

- La barra superior que posee cuatro botones, marca a, en la que nos encontramos cuatro botones, los tres primeros para extender los casos de uso de creación de

objeto, el siguiente para extender el caso de uso búsqueda de objeto y el último que actualiza las otras partes de la interfaz.

- La marca b muestra una tabla con un grupo de carpetas estructuradas en forma de árbol, estas carpetas representan las Unidades Organizativas (OU's, contenedores) del directorio activo. Al hacer clic sobre una de ellas se modificará la tabla de la derecha, la de la marca c.
- La tabla con la marca c es en la que se representan los objetos contenidos en la unidad organizativa que esté seleccionada en la tabla de la marca b.

3. Agente del CAU: Si quiere crear un objeto, primero seleccionará la unidad organizativa en la que quiera crearlo y después hará clic en el botón situado más a la izquierda si quiere crear un usuario, en el siguiente si quiere crear un grupo y en el tercero si quiere crear un equipo

4. Sistema: Extenderá el caso de uso 7 (Crear Usuario), 8 (Crear grupo) o 9 (Crear Equipo) según el botón pulsado por el agente del CAU.

5. Agente del CAU: Si lo que quiere es buscar un objeto en el directorio activo, hará clic en el botón con la imagen de una lupa.

6. Sistema: Extenderá el caso de uso buscar objeto.

7. Agente del CAU: Si hace clic con el botón derecho sobre una de las carpetas de la tabla derecha, como muestra la figura 4.0a (Directorio Activo Contenedor), podrá seleccionar cualquiera de las mismas opciones que en la barra superior pero mediante el menú que se carga.

8. Sistema: Extenderá el caso de uso seleccionado.

9. Agente del CAU: Puede gestionar un objeto de los que forman parte de la Unidad Organizativa seleccionada en la tabla de la marca b, haciendo clic con el botón derecho sobre uno de sus objetos en la tabla con la marca c. En ese momento se carga un menú, como indica la figura 4.0b (Directorio Activo objeto), que incluye las opciones que se pueden llevar a cabo sobre un objeto. Que son:

- Propiedades: para el caso de uso gestionar propiedades del objeto (Número 11).
- Mover: para el caso de uso Mover objeto (Número 12).
- Eliminar: para Eliminar objeto (Número 13).
- Habilitar: para el caso de uso Habilitar/Deshabilitar usuario o equipo (Número 18). Y que solo aparecerá en el caso de hacer clic con el botón derecho sobre un usuario o un equipo. Solo se podrá hacer clic en ella si el usuario está Deshabilitado.
- Deshabilitar: para el caso de uso Habilitar/Deshabilitar usuario o equipo (Número 18). Y que solo aparecerá en el caso de hacer clic con el botón derecho sobre un usuario o un equipo. Solo se podrá hacer clic en ella si el usuario está Habilitado.
- Desbloquear: solo aparecerá si el objeto es de tipo usuario y es para el caso de uso Desbloquear usuario (Número 16). Solo se podrá hacer clic en ella si el usuario está bloqueado.
- Reseteo contraseña: solo aparecerá si el objeto es de tipo usuario y es para el caso de uso Resetear contraseña usuario (Número 15).
- Cambiar nombre: para el caso de uso Cambiar nombre objeto (Número 17).
- Tareas Exchange: para el caso de uso realizar Tareas Exchange (Número 64) y que solo aparecerá si el objeto es un usuario o grupo.

10. Sistema: Extenderá el caso de uso según la opción seleccionada por el Agente del CAU.

## Extensiones (cursos alternativos)

Paso 3: El agente del CAU intenta crear un objeto en la raíz del dominio.

- Sistema: Devuelve un mensaje de error y no crea nada.

Paso 5: El agente del CAU hace clic en el botón derecho sobre una unidad organizativa que aparece en la tabla de la derecha por estar incluida en otra.

- Sistema: No despliega ningún menú.

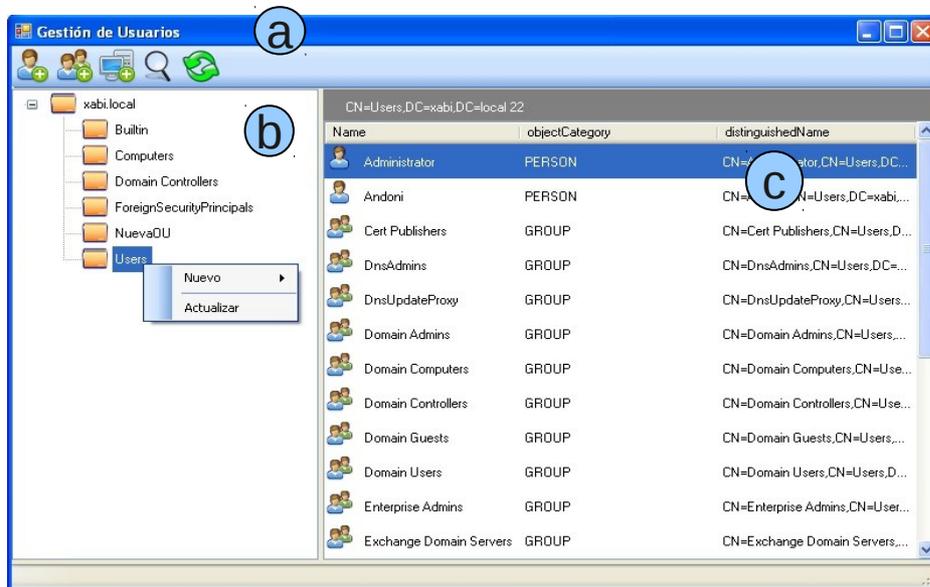


Figura 4.0a Directorio Activo contenedor

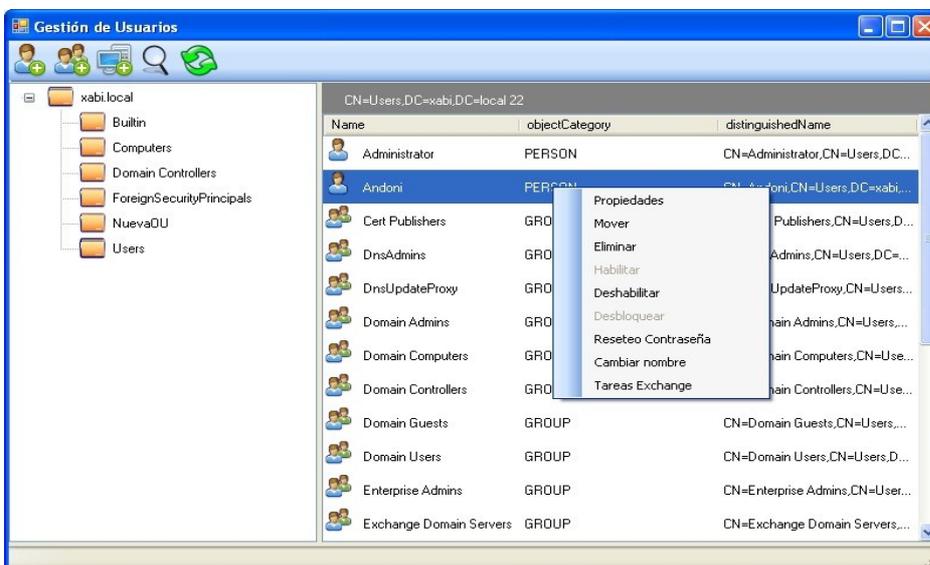


Figura 4.0a Directorio Activo objeto

### 4.2.3.2 Buscar Objetos

#### 2. Caso de Uso: Buscar Objetos

Actores: Agente del CAU

Resumen: Permite buscar usuarios, grupos y equipos por su nombre, nombre de login, etc

Precondición:

Postcondición: Se devuelven los resultados de la búsqueda.

**Escenario principal (curso normal de eventos)**

1. Agente del CAU: El agente quiere buscar un objeto
2. Sistema: Si el caso de uso no extiende de ningún otro, le muestra la interfaz Buscador General (Figura 4.8) sino le muestra Buscador Añadir (Figura 4.9). En ambas hay un campo en el que insertar el nombre o parte del nombre del objeto que se quiere buscar, Elemento a buscar y Elemento respectivamente.
3. Agente del CAU: Introduce la cadena a buscar y hace clic en el botón buscar.
4. Sistema: Devuelve, en la tabla de resultados de las interfaces, los atributos de los objetos encontrados, estos serán los que marcan los nombres de las columnas de la tabla.
5. Agente del CAU: En el caso de la interfaz Buscador Añadir (Figura 4.9) selecciona uno de los objetos de los resultados y hace clic en el botón aceptar.

**Extensiones (cursos alternativos)**

Paso 3: El agente del CAU no escribe nada para buscar

- Sistema: Devuelve un mensaje de error.

Paso 5: El agente del CAU hace clic en el botón aceptar sin haber seleccionado un objeto de los resultados.

- Sistema: Devuelve un mensaje de error.

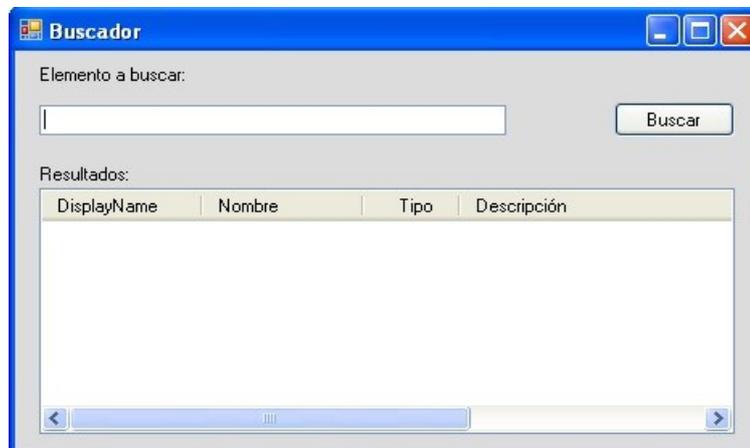


Figura 4.8 BuscadorGeneral



Figura 4.9 BuscadorAñadir

### 4.2.3.3 Casos de uso relativos a las Tareas Exchange

Cuando el Agente del CAU seleccione la opción Tareas Exchange de un usuario o grupo, el sistema llama al caso de uso Realizar Tareas Exchange, número 64, y de éste extenderán los casos de uso Tareas Exchange usuario (número 65) y Tareas Exchange grupo (número 66) dependiendo de si el objeto sobre el que se ha seleccionado la opción es un usuario o un grupo. Ambos comparten interfaz, que se muestra en la Figura 4.10.



Figura 4.10 TareasExchange

Como se puede apreciar en la interfaz, nos encontramos ante una tabla con diversos elementos, que cambiarán según cuales sean las circunstancias del objeto, es decir, en el caso de un usuario si posee un buzón de correo y en el caso del grupo si posee una dirección de correo. Al seleccionar una opción y al hacer clic en aceptar se extenderá su caso de uso correspondiente. Casos de uso que se explican a continuación.

#### 8. Caso de Uso: Crear Buzón

**Actores:** Agente del CAU

**Resumen:** El caso de uso comienza al estar el agente del CAU dentro del caso de uso “Crear Equipo” o en el “Realizar tareas Exchange de Usuario”. Permite crear un buzón de correo Exchange para el usuario que se está creando o para el que ya está creado pero no posee uno, para ello, necesitará el alias del buzón. Al estar ya explicado el caso cuando extiende de Crear Usuario, aquí se explicará cuando extiende de Realizar Tareas Exchange de Usuario.

**Precondición:** Que el agente del CAU haya decidido crear el buzón en el caso de uso “Crear Usuario”

**Postcondición:** Se crea y asigna un buzón al usuario.

#### **Escenario principal (curso normal de eventos)**

1. Agente del CAU: Selecciona la opción crear buzón y hace clic en el botón siguiente en la interfaz TareasExchangeUsuario crearBuzón1 (Figura 4.11).
2. Sistema: Le muestra en la interfaz TareasExchangeUsuario CrearBuzón2 (Figura 4.12) con el campo en el que escribir el alias del buzón y los desplegables de los que seleccionar una de sus opciones, Servidor y Almacén.
3. Agente del CAU: Rellena el campo alias y selecciona una de las opciones del desplegable Servidor y otra del Almacén y hace clic en el botón Siguiete.
4. Sistema: Crea el buzón y lo almacena en el servidor Exchange asociado al directorio activo.

#### **Extensiones (cursos alternativos)**

Paso 3: El agente del CAU no escribe ningún alias o no selecciona ninguna opción de

los desplegables.

- Sistema: Devuelve un mensaje de error.



Figura 4.11 TareasExchangeUsuario CrearBuzón1

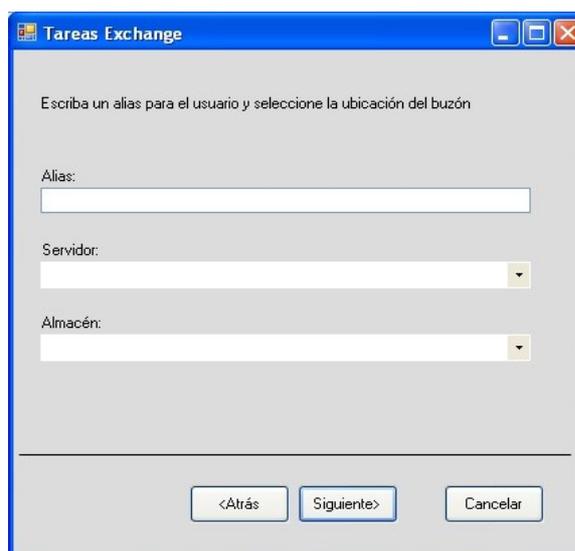


Figura 4.12 TareasExchangeUsuario CrearBuzón2

## 67. Caso de Uso: Mover Buzón

**Actores:** Agente del CAU

**Resumen:** El caso de uso comienza al estar el agente del CAU dentro del caso de uso “Realizar tareas Exchange de Usuario”. Permite mover un buzón de correo Exchange de un almacén de buzones a otro dentro de su mismo servidor Exchange o dentro de otro.

**Precondición:** Que el usuario tenga un buzón asignado

**Postcondición:** Se mueve el buzón del usuario.

### Escenario principal (curso normal de eventos)

1. Agente del CAU: Selecciona la opción mover buzón y hace clic en el botón

- siguiente en la interfaz TareasExchangeUsuario moverBuzón1 (Figura 4.13).
2. Sistema: Le muestra en la interfaz TareasExchangeUsuario moverBuzón2 (Figura 4.14) en el que aparece un campo que no se puede modificar con la dirección del almacén del buzón, Almacén actual del buzón, y con los desplegables Servidor y Almacén.
  3. Agente del CAU: Selecciona una de las opciones del desplegable Servidor y otra del Almacén y hace clic en el botón Siguiente.
  4. Sistema: Mueve el buzón del almacén de origen del buzón al almacén seleccionado.

### Extensiones (cursos alternativos)

Paso 3: El agente del CAU no elige ningún almacén.

- Sistema: Devuelve un mensaje de error.



Figura 4.13 TareasExchangeUsuario MoverBuzón1



Figura 4.14 TareasExchangeUsuario MoverBuzón2

## 68. Caso de Uso: Eliminar Buzón

**Actores:** Agente del CAU

**Resumen:** El caso de uso comienza al estar el agente del CAU dentro del caso de uso “Realizar tareas Exchange de Usuario”. Permite eliminar un buzón de correo Exchange para un usuario que posea uno.

**Precondición:** Que el usuario posea un buzón de correo Exchange.

**Postcondición:** Se elimina el buzón asignado al usuario.

### **Escenario principal (curso normal de eventos)**

1. Agente del CAU: Selecciona la opción eliminar buzón y hace clic en el botón siguiente en la interfaz TareasExchangeUsuario EliminarBuzón (Figura 4.14).
2. Sistema: Elimina el buzón.

### **Extensiones (cursos alternativos)**



Figura 4.14 TareasExchangeUsuario EliminarBuzón

## 69. Caso de Uso: Eliminar mail Grupo

**Actores:** Agente del CAU

**Resumen:** El caso de uso comienza al estar el agente del CAU dentro del caso de uso “Realizar tareas Exchange de Grupo”. Permite eliminar una dirección de correo Exchange para el grupo que posee una.

**Precondición:** Que el grupo tenga una dirección de correo Exchange asignada.

**Postcondición:** Se elimina la dirección.

### Escenario principal (curso normal de eventos)

1. Agente del CAU: Selecciona la opción eliminar correo grupo y hace clic en el botón siguiente en la interfaz TareasExchangeGrupo EliminarCorreoGrupo (Figura 4.15).
2. Sistema: Elimina la dirección de correo del grupo.

### Extensiones (cursos alternativos)



Figura 4.15 TareasExchangeGrupo EliminarCorreoGrupo

## 70. Caso de Uso: Establecer mail Grupo

**Actores:** Agente del CAU

**Resumen:** El caso de uso comienza al estar el agente del CAU dentro del caso de uso “Realizar tareas Exchange de Grupo” o dentro de “Crear Grupo”. Permite crear una dirección de correo Exchange, eligiendo el grupo administrativo en el que crearla, para el grupo que se está creando o para el que ya está creado pero no posee una.

**Precondición:** Que el agente del CAU haya decidido crear el mail en el caso de uso “Crear Grupo” o que estando ya creado, no posea una dirección de correo Exchange

**Postcondición:** Se crea y asigna un buzón al usuario.

### Escenario principal (curso normal de eventos)

1. Agente del CAU: Selecciona la opción establecer correo grupo y hace clic en el botón siguiente en la interfaz TareasExchangeGrupo EstablecerCorreoGrupo1 (Figura 4.16).
2. Sistema: Le muestra en la interfaz TareasExchangeGrupo EstablecerCorreoGrupo2 (Figura 4.17) en el que está el campo Alias y un desplegable, Grupo Administrativo asociado.
3. Agente del CAU: Selecciona el grupo administrativo de entre las opciones del desplegable Grupo Administrativo Asociado y rellena el campo Alias y hace clic en el botón Siguiente.
4. Sistema: Crea la dirección de correo para el grupo.

### Extensiones (cursos alternativos)

- Paso 3: El agente del CAU no elige ningún grupo administrativo o no escribe el Alias.
- Sistema: Devuelve un mensaje de error.



Figura 4.16 TareasExchangeGrupo ElstablecerCorreoGrupo1



Figura 4.17 TareasExchangeGrupo ElEstablecerCorreoGrupo2

#### 4.2.3.4 Mover objeto

<p>12. Caso de Uso: Mover Objeto</p> <p><b>Actores:</b> Agente del CAU</p> <p><b>Resumen:</b> Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso “Gestionar objeto”, permitiéndole mover un objeto de una ubicación del directorio activo a otra.</p> <p><b>Precondición:</b> Que el objeto exista en el directorio activo</p> <p><b>Postcondición:</b> Mueve el objeto a una nueva ubicación lógica dentro del directorio activo</p>
<p><b>Escenario principal (curso normal de eventos)</b></p> <ol style="list-style-type: none"> <li>1. Agente del CAU: El agente quiere mover un objeto dentro del directorio activo</li> <li>2. Sistema: Carga la interfaz Mover Objeto (Figura 4.21) en la que aparece una tabla con los Contenedores (OU's) del directorio activo formando un árbol.</li> <li>3. Agente del CAU: Selecciona el contenedor de destino del objeto.</li> <li>4. Sistema: Mueve el objeto a la ubicación seleccionada</li> </ol>
<p><b>Extensiones (cursos alternativos)</b></p> <p>Paso 3: El agente del CAU no establece la nueva ubicación para el objeto.</p> <ol style="list-style-type: none"> <li>1. Sistema: Devuelve un mensaje de error y no mueve el objeto.</li> </ol>



Figura 4.21 Mover Objeto

#### 4.2.3.5 Eliminar objeto

##### 13. Caso de Uso: Eliminar Objeto

Actores: Agente del CAU

Resumen: Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso "Gestionar objeto", permitiéndole eliminar un objeto del directorio activo.

Precondición: Que el objeto exista en el directorio activo

Postcondición: Elimina el objeto del directorio activo

##### **Escenario principal (curso normal de eventos)**

1. Agente del CAU: El agente quiere eliminar un objeto del directorio activo
2. Sistema: Elimina el objeto

##### **Extensiones (cursos alternativos)**

#### 4.2.3.6 Cambiar nombre al objeto

##### 17. Caso de Uso: Cambiar el nombre al Objeto

Actores: Agente del CAU

Resumen: Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso "Gestionar objeto", permitiéndole cambiar el nombre a un objeto del directorio activo.

Precondición: Que el objeto exista en el directorio activo

Postcondición: Renombra el objeto del directorio activo

##### **Escenario principal (curso normal de eventos)**

1. Agente del CAU: El agente quiere renombrar un objeto del directorio activo
2. Sistema: Carga la interfaz Mover Objeto (Figura 4.21) Que le pide el nuevo nombre a asignar al objeto mediante un campo de texto.

3. Agente del CAU: Devuelve el nuevo nombre
4. Sistema: Asigna el nuevo nombre al objeto

#### **Extensiones (cursos alternativos)**

Paso 3: El nombre devuelto por el agente del CAU ya existe en el directorio activo

1. El sistema devuelve un mensaje de error y “obliga” al agente a establecer otro nombre para el objeto

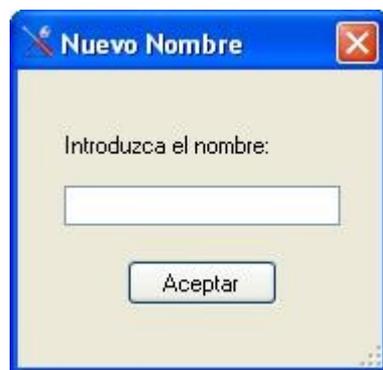


Figura 4.21 Cambiar nombre Objeto

#### **4.2.3.7 Habilitar/Deshabilitar objeto**

18. Caso de Uso: Habilitar/Deshabilitar usuario o equipo

Actores: Agente del CAU

Resumen: Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso “Gestionar objeto”, permitiéndole habilitar o deshabilitar un objeto del tipo usuario o equipo del directorio activo.

Precondición: Que el objeto exista en el directorio activo

Postcondición: Habilita o deshabilita el objeto del directorio activo

#### **Escenario principal (curso normal de eventos)**

1. Agente del CAU: El agente quiere habilitar o deshabilitar un objeto del directorio activo
2. Sistema: Habilita o deshabilita el objeto

#### **Extensiones (cursos alternativos)**

#### **4.2.3.8 Resetear contraseña de un usuario**

15. Caso de Uso: Resetear contraseña de un usuario

Actores: Agente del CAU

Resumen: Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso “Gestionar características propias de un usuario”, permitiéndole resetear la contraseña de un objeto del tipo usuario del directorio activo.

Precondición: Que el objeto exista en el directorio activo  
Postcondición: Se establece la nueva contraseña como la actual del usuario

**Escenario principal (curso normal de eventos)**

1. Agente del CAU: El agente quiere resetear la contraseña de un usuario del directorio activo
2. Sistema: Pide la nueva contraseña a asignarle al usuario
3. Agente del CAU: El agente envía la nueva contraseña
4. Sistema: Asigna la nueva contraseña al usuario

**Extensiones (cursos alternativos)**

- Paso 3: El agente del CAU envía un string vacío como contraseña
1. Sistema: Devuelve un mensaje de error y “obliga” al agente a escribir otra contraseña nueva.

#### 4.2.3.9 Desbloquear usuario

16. Caso de Uso: Desbloquear usuario

Actores: Agente del CAU

Resumen: Este caso de uso se da cuando el agente del CAU se encuentra en el caso de uso “Gestionar características propias de un usuario”, desbloquear un objeto del tipo usuario del directorio activo.

Precondición: Que el objeto exista en el directorio activo

Postcondición: Se desbloquea el usuario

**Escenario principal (curso normal de eventos)**

1. Agente del CAU: El agente quiere desbloquear un usuario del directorio activo
2. Sistema: Desbloquea el usuario

**Extensiones (cursos alternativos)**

- Paso 2: El agente del CAU intenta desbloquear un usuario no bloqueado
1. Sistema: Devuelve un mensaje de error

#### 4.2.4 Casos de uso de la cuarta iteración

Los casos de uso que no afectan a los objetos del directorio activo.

##### 4.2.4.1 Configuración IExplorer

4. Caso de Uso: Cambiar configuración de Internet Explorer

Actores: Agente del CAU

Resumen: Este caso de uso permite al agente del CAU establecer la configuración de conexión (configuración de proxy) de equipos de la red de manera remota.

Precondición:

**Postcondición:** Se cambia la configuración de conexión del Internet Explorer del equipo seleccionado.

**Escenario principal (curso normal de eventos)**

1. Agente del CAU: El agente quiere cambiar el nombre del perfil de usuario en un equipo de la red.
2. Sistema: Carga la interfaz Configurar Iexplorer (Figura 4.18) en la que solo está habilitado para la escritura el campo Equipo.
3. Agente del CAU: Introduce el nombre del equipo y hace clic en el botón Cargar
4. Sistema: Carga en la interfaz los datos de conexión de su Internet Explorer, si está habilitado el proxy o no, el nombre de éste y sus excepciones.
5. Agente del CAU: Modifica los campos con los datos de conexión y hace clic en el botón aceptar.
6. Sistema: Asigna los datos modificados al Internet Explorer del equipo

**Extensiones (cursos alternativos)**

Paso 3: El agente del CAU no escribe un nombre válido o deja el campo vacío.

- Sistema: Devuelve un mensaje de error y no carga ningún dato.

Paso 5: El agente del CAU deja el campo proxy vacío.

- Sistema: Devuelve un mensaje de error y no asigna ningún dato.

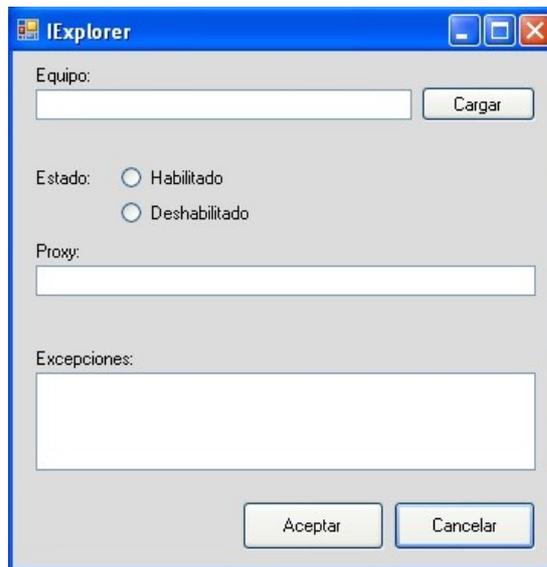


Figura 4.18 Configurar IE Explorer

#### 4.2.4.2 Renombrar perfil

##### 5. Caso de Uso: Cambiar nombre de Perfil

Actores: Agente del CAU

Resumen: Este caso de uso permite al agente del CAU renombrar un perfil de un usuario de Windows en un equipo de la red, de manera remota.

Precondición:

Postcondición: Se cambia el nombre del perfil de usuario seleccionado.

##### **Escenario principal (curso normal de eventos)**

1. Agente del CAU: El agente quiere cambiar el nombre de perfil de un usuario que se encuentra en una máquina de la red.
2. Sistema: Carga la interfaz Renombrar Perfil (Figura 4.19) que posee tres campos, Equipo para el nombre de equipo en el que se encuentra el perfil, Perfil que se quiere renombrar para el perfil al que se quiere cambiar el nombre y Nuevo nombre perfil para el nombre que se quiere asignar al perfil.
3. Agente del CAU: Introduce los datos y hace clic en el botón Renombrar
4. Sistema: Asigna el nuevo nombre al perfil

##### **Extensiones (cursos alternativos)**

Paso 3: El agente del CAU no escribe alguno de los campos.

- Sistema: Devuelve un mensaje de error y no renombra el perfil.

Paso 3: El nombre de la máquina en la que se encuentra el perfil a renombrar no es válido.

- Sistema: Devuelve un mensaje de error y “obliga” a reescribir el nombre de la máquina.

Paso 3: El nombre del perfil a renombrar no es válido.

- Sistema: Devuelve un mensaje de error y no renombra el perfil.



Figura 4.19 Renombrar Perfil

#### 4.2.4.3 Gestionar carpetas compartidas

##### 6. Caso de Uso: Gestionar Carpetas

Actores: Agente del CAU

Resumen: Este caso de uso permite al agente del CAU crear, modificar y eliminar carpetas dentro de los directorios compartidos de los equipos de la red.

Precondición: Que el equipo sobre el que se van a realizar los cambios conceda permisos de modificación al Administrador del directorio Activo

Postcondición: Se modifica el sistema de carpetas de los directorios compartidos por la máquina seleccionada.

##### **Escenario principal (curso normal de eventos)**

1. Agente del CAU: El agente quiere crear, eliminar o modificar carpetas compartidas de una máquina de la red
2. Sistema: Carga la interfaz Gestionar carpetas (Figura 4.20) en la que hay una tabla vacía y un campo en el que se debe insertar el nombre del servidor del cual se quieren gestionar sus carpetas compartidas.
3. Agente del CAU: Introduce el nombre del servidor.
4. Sistema: Devuelve las carpetas compartidas del equipo y las subcarpetas de éstas, formando un árbol que aparecerá en la tabla, como se ve en la Figura 4.20.
5. Agente del CAU: Selecciona una de las carpetas y establece que quiere crear una subcarpeta (haciendo clic en el botón Añadir carpeta), eliminar la carpeta (haciendo clic en el botón Eliminar carpeta) seleccionada o modificar su nombre (haciendo doble clic sobre la carpeta seleccionada ).
6. Sistema: Modifica o elimina la carpeta o crea una nueva subcarpeta en ésta

##### **Extensiones (cursos alternativos)**

Paso 3: El agente del CAU no establece la máquina sobre la que gestionar sus carpetas compartidas.

2. Sistema: Devuelve un mensaje de error.

Paso 5: El agente del CAU intenta gestionar una carpeta del sistema.

- Sistema: Devuelve un mensaje de error y no permite realizar acciones sobre ella.

Paso 5: El agente del CAU no selecciona ninguna carpeta que gestionar.

- Sistema: Devuelve un mensaje de error.

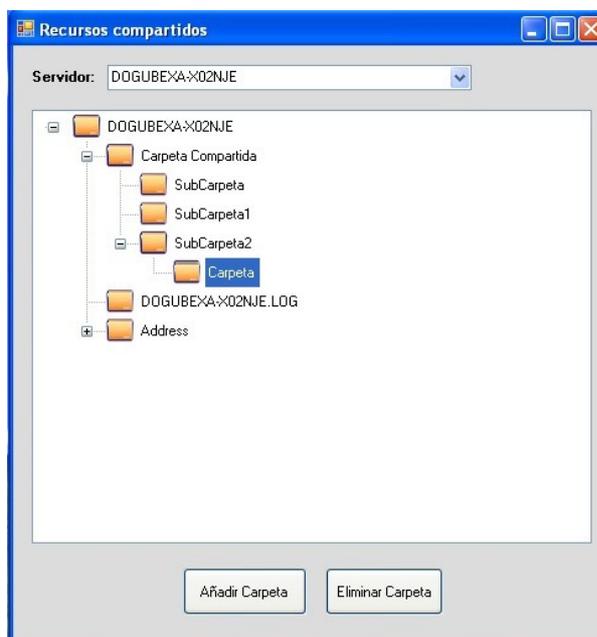


Figura 4.20 Gestionar carpetas

### 4.3 Modelo de dominio

Este es el modelo de dominio para la aplicación que se lleva a cabo en este proyecto de fin de carrera. El modelo de dominio de el directorio activo es bastante más grande, pero no merece la pena ir más allá de lo que la aplicación va a necesitar.

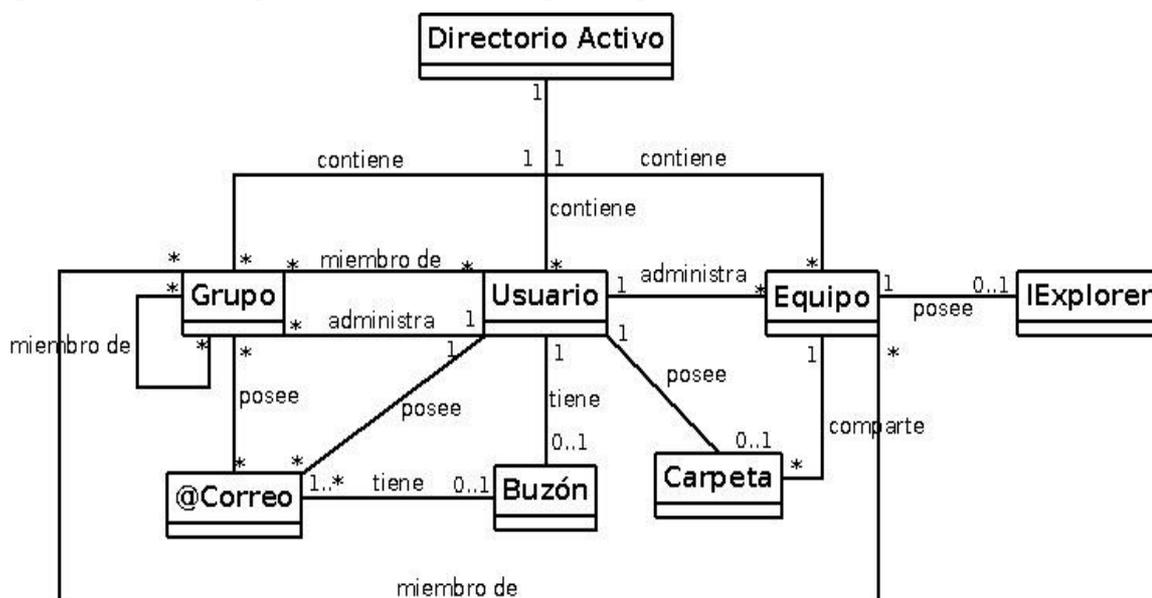


Figura 4.3.1 Modelo de dominio

#### 4.3.1 Atributos principales de los objetos

Para mayor comodidad a la hora de analizar el modelo de dominio, se ha optado por enumerar los atributos de cada objeto aparte. Al tener un volumen de atributos muy grande, se han agrupado los atributos más específicos y de menor relevancia.

#### Usuario:

- Dirección: Dirección del objeto en el directorio activo
- Nombre: Nombre del usuario
- Apellidos: Apellidos del usuario
- Contraseña: Contraseña de la cuenta de usuario
- NombreLogin: id del usuario
- OpcionesPassword: Representa un grupo de atributos, los que establecen si el usuario tiene que cambiar password en el siguiente login, el password nunca expira etc.
- FechaExpira: Fecha en la que expira la cuenta de usuario
- DireccionesPerfil: Representa un grupo de atributos, los que establecen las direcciones del perfil, script de Logon y poseedor de la Home
- MiembroDe: Lista con los nombres de los grupos de los que es miembro
- DatosCanónicos: Representa un grupo de atributos, los que establecen el nombre canonico, Usn's y fecha de creación y modificación del usuario
- DatosSeguridad: Representa un grupo de datos, los que establecen quienes tienen permisos sobre el usuario y cuales son
- Bloqueado: Indica si el usuario está bloqueado
- Habilitado/Deshabilitado: Indica si el usuario está habilitado o deshabilitado
- GruposAdministrados: Lista de grupos administrados por el usuario
- EquiposAdministrados: Lista de equipos administrados por el usuario

#### Buzón:

- UsuarioDueño: Usuario al que está asignado el buzón
- DatosTamañoMensj: Representa un grupo de datos, los que establecen el tamaño máximo de los mensajes de entrada y salida del buzón
- EmisoresPermitidos/Denegados: Lista de Nombres de los usuario de los que solo se puede recibir mensajes o de los que no se puede recibir mensajes.
- PermitidosEnvíoComo: Lista de usuarios que pueden tomar el rol del dueño del buzón para enviar mensajes
- Redirección: @correo a la que se redireccionan los mensajes, sin está vacío no se reenvían a nadie
- LímiteDestinatarios: Máximo número de destinatarios para un mensaje que envíe el buzón
- LímitesAlmacenaje: Representa un grupo de datos, los que establecen el máximo tamaño en KB de un mensaje que se puede recibir o enviar
- @correo: Direcciones de correo asignadas al buzón
- @correoPrimaria: Dirección de correo primaria
- ServiciosMóviles: Representa un grupo de datos, los que establecen que servicios móviles (Outlook mobile access, up-to-date notifications...) están habilitados y cuales deshabilitados
- Protocolos: Representa un grupo de datos, los que establecen que protocolos(POP3, IMAP4...) están habilitados y cuales deshabilitados
- SimpleDisplay: El nombre Simple Display del buzón

- DatosOcultaciónPrioridad: Representa un grupo de datos, los que establecen si las direcciones de correo gestionadas por el buzón no aparecerán entre los destinatarios al enviarles un mensaje o si tienen prioridad las de tipo x400
- DatosSeguridad: Representa un grupo de datos, los que establecen quienes tienen permisos sobre el buzón y cuales son
- Almacén: Almacén de buzones en el que se encuentra.

#### Grupo:

- Nombre: Nombre del grupo
- Tipo: Tipo del grupo (Security o distribución, dominio local, global o universal)
- @correo: Direcciones de correo asignadas al grupo
- @correoPrimaria: Dirección de correo primaria del grupo
- MiembroDe: Lista de nombres de los grupos de los que es miembro
- Miembros: Lista de nombres de los objetos miembros del grupo
- Administrador: Nombre del usuario administrador del grupo, si no hay nada en este atributo, es que no hay administrador
- Dirección: Dirección del objeto en el directorio activo
- DatosSeguridad: Representa un grupo de datos, los que establecen quienes tienen permisos sobre el grupo y cuales son
- DatosTamañoMensj: Representa un grupo de datos, los que establecen el tamaño máximo de los mensajes de entrada y salida del grupo (Este atributo solo tendrá un valor si el grupo tiene una dirección de correo).
- EmisoresPermitidos/Denegados: Lista de Nombres de los usuario de los que solo se puede recibir mensajes o de los que no se puede recibir mensajes (Este atributo solo tendrá un valor si el grupo tiene una dirección de correo).
- ServidorExch: Nombre del servidor exchange que da el servicio de correo (Este atributo solo tendrá un valor si el grupo tiene una dirección de correo).
- DatosOpcionesRecepción: Representa un grupo de datos, los que establecen si las direcciones de correo gestionadas por el buzón no aparecerán entre los destinatarios al enviarles un mensaje o si está activado el mensaje fuera-de-la-oficina (Este atributo solo tendrá un valor si el grupo tiene una dirección de correo).
- InformesDeRecepciónA: Dato que dice si se envía un informe de recepción al emisor, el dueño del buzón o a nadie (Este atributo solo tendrá un valor si el grupo tiene una dirección de correo).
- GrupoAdministrativo: Grupo administrativo en el que se encuentra su dirección de correo (Este atributo solo tendrá un valor si el grupo tiene una dirección de correo).

#### Equipo:

- Nombre: Nombre del equipo
- DatosSO: Representa un grupo de datos, los que establecen el sistema operativo de la máquina, su versión y su service pack
- Administrador: Nombre del usuario administrador del equipo, si es que lo tiene
- MiembroDe: Lista de nombres de los grupos de los que es miembro
- DatosSegurtidad: Representa un grupo de datos, los que establecen quienes tienen permisos sobre el equipo y cuales son

- Dirección: Dirección del objeto en el directorio activo
- CarpetasCompartidas: Lista de los nombres de las carpetas compartidas del equipo
- Habilitado/Deshabilitado: Indica si el equipo está habilitado o deshabilitado

#### Carpeta:

- Nombre: Nombre de la carpeta
- Dirección: Dirección del objeto en el directorio activo
- EquipoPertenencia: Nombre del equipo al que pertenece
- PerfilUsuario: Si es una carpeta de perfil de usuario, este atributo tiene su nombre
- Compartida?: Dato que establece si es una carpeta compartida

#### Iexplorer:

- EquipoPertenencia: Equipo en el que está instalado el Internet Explorer
- Proxy: Proxy por el que se conecta
- Habilitado/Deshabilitado: Indica si está habilitado el que Internet Explorer se conecte vía el proxy del anterior atributo
- Excepciones: Excepciones por las que IE no pasa por el proxy

#### @Correo:

- Dirección: La propia dirección de correo
- NombrePropietario: Nombre del usuario o grupo propietario de la dirección de correo
- NombreBuzón: Nombre del buzón al que pertenece, siempre y cuando pertenezca a alguno
- Primaria?: Dato que establece si es la dirección primaria del propietario

## 5 ANÁLISIS

### 5.1 Casos de uso de la primera iteración

#### Caso de uso 7: Crear Usuario

Diagrama de secuencia:

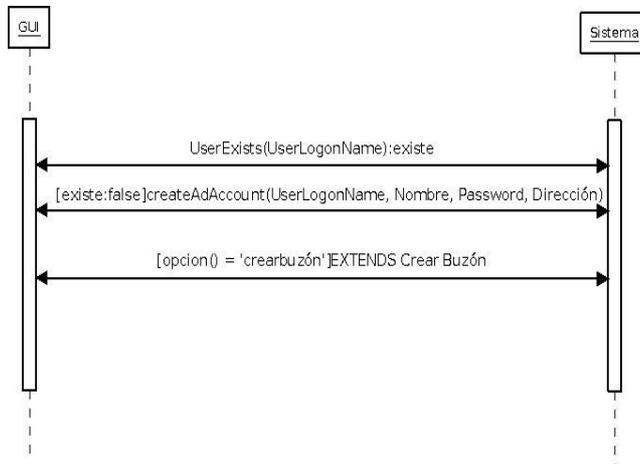


Figura 5.1.1 Crear Usuario

Contratos:

**Nombre:** UserExists(UserLogonName: string)

**Responsabilidades:** Comprueba si el nombre de login del usuario que se le pasa por parámetros pertenece a un usuario de el directorio activo.

**Referencias Cruzadas:** Casos de uso: Crear Usuario

**Precondiciones:** Que el parámetro de entrada de la función tenga algún valor

**Postcondiciones:**

**Salida:** Un booleano (existe) que devuelve si el usuario existe en el directorio activo o no.

**Nombre:** createAdAccount(UserLogonName: string, Nombre: String, Password: string, Direccion: string)

**Responsabilidades:** Crea un nuevo usuario con los datos que se le pasan por parámetros en la dirección que se establece en el parámetro dirección

**Referencias Cruzadas:** Casos de uso: Crear Usuario

**Precondiciones:** Que el nombre de login que se le quiera asignar al usuario no esté siendo utilizado en el directorio activo. Es decir, que el valor de salida de la función UserExists sea false.

**Postcondiciones:** Se creo una nueva cuenta de usuario en el directorio activo. Se le asignaron los datos que se obtienen por parámetros. Y la dirección en la que se crea es la que se obtiene como el último parámetro también.

**Salida:**

## Caso de uso 9: Crear Grupo

### Diagrama de secuencia:

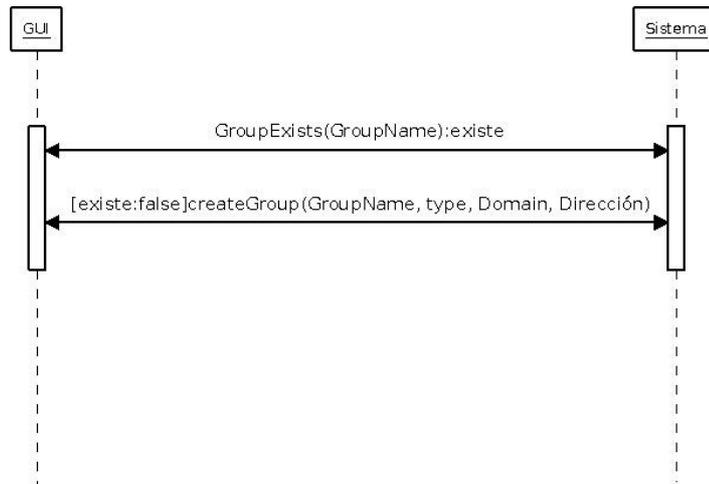


Figura 5.1.2 Crear Grupo

### Contratos:

**Nombre:** GroupExists(GroupName: string)

**Responsabilidades:** Comprueba si el nombre del grupo que se le pasa por parámetros pertenece a un grupo de el directorio activo.

**Referencias Cruzadas:** Casos de uso: Crear Grupo

**Precondiciones:** Que el parámetro de entrada de la función tenga algún valor

**Postcondiciones:**

**Salida:** Un booleano (existe) que devuelve si el grupo existe en el directorio activo o no.

**Nombre:** createGroup(GroupName: string, type: String, Domain: string, Direccion: string)

**Responsabilidades:** Crea un nuevo grupo con los datos que se le pasan por parámetros en la dirección que se establece en el parámetro dirección

**Referencias Cruzadas:** Casos de uso: Crear Grupo

**Precondiciones:** Que el nombre de login que se le quiera asignar al grupo no esté siendo utilizado en el directorio activo. Es decir, que el valor de salida de la función GroupExists sea false.

**Postcondiciones:** Se creo una nuevo grupo en el directorio activo. Se le asignaron los datos que se obtienen por parámetros. Y la dirección en la que se crea es la que se obtiene como el último parámetro también.

**Salida:**

## Caso de uso 10: Crear Equipo

Diagrama de secuencia:



Figura 5.1.3 Crear Equipo

Contratos:

**Nombre:** ComputerExists(ComputerName: string)

**Responsabilidades:** Comprueba si el nombre de login del equipo que se le pasa por parámetros pertenece a un equipo de el directorio activo.

**Referencias Cruzadas:** Casos de uso: Crear Equipo

**Precondiciones:** Que el parámetro de entrada de la función tenga algún valor

**Postcondiciones:**

**Salida:** Un booleano (existe) que devuelve si el equipo existe en el directorio activo o no.

**Nombre:** createComputer(ComputerName: string, Direccion: string)

**Responsabilidades:** Crea un nuevo equipo con los datos que se le pasan por parámetros en la dirección que se establece en el parámetro dirección

**Referencias Cruzadas:** Casos de uso: Crear Equipo

**Precondiciones:** Que el nombre de login que se le quiera asignar al equipo no esté siendo utilizado en el directorio activo. Es decir, que el valor de salida de la función ComputerExists sea false.

**Postcondiciones:** Se creo un nuevo equipo en el directorio activo. Se le asignaron los datos que se obtienen por parámetros. Y la dirección en la que se crea es la que se obtiene como el último parámetro también.

**Salida:**

## 5.2 Caso de uso de la segunda iteración

### 5.2.1 Casos de uso de propiedades del usuario

#### Caso de uso 22: Gestionar propiedades generales usuario

Diagrama de secuencia:



Figura 5.8.3 Cambiar nombre y apellidos usuario

#### Contratos:

**Nombre:** GetUserProperty(UserLogName: String, Atributo: String)

**Responsabilidades:** Obtiene el atributo del usuario cuyo nombre es el que se pasa por parámetros “Atributo”

**Referencias Cruzadas:** Casos de uso: Cambiar nombre y apellidos usuario, Cambiar Simple Display.

**Precondiciones:** El usuario de nombre de login UserLogName existe en el directorio activo. El atributo guarda un valor de tipo string.

**Postcondiciones:**

**Salida:** Un string con el valor del atributo

**Nombre:** SetUserProperty(UserLogName: String, Atributo: String, datos: String)

**Responsabilidades:** Modifica el atributo del usuario cuyo nombre es el que se pasa por parámetros “Atributo”

**Referencias Cruzadas:** Casos de uso: Cambiar nombre y apellidos usuario, Cambiar Simple Display.

**Precondiciones:** El usuario de nombre de login UserLogName existe en el directorio activo. El atributo guarda un valor de tipo string.

**Postcondiciones:** Se cambia el valor que guarda el atributo con el nombre que se pasa por parámetros “Atributo” con los datos que se pasan por parámetros también “datos”.

**Salida:**

## Caso de uso 25: Gestionar opciones de password

### Diagrama de secuencia:

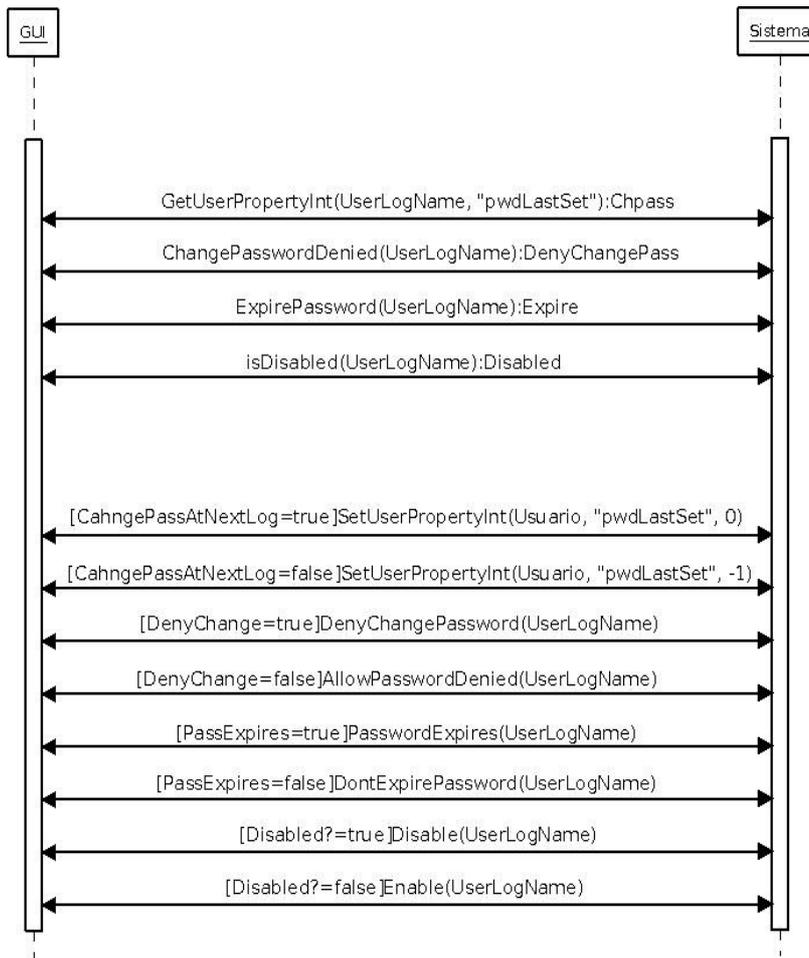


Figura 5.8.4 Gestionar Opciones de password

### Contratos:

**Nombre:** GetUserPropertyInt(UserLogName: String, Atributo: String)

**Responsabilidades:** Obtiene el atributo del usuario cuyo nombre es el que se pasa por parámetros “Atributo”

**Referencias Cruzadas:** Casos de uso: Gestionar opciones de password

**Precondiciones:** El usuario de nombre de login UserLogName existe en el directorio activo. El atributo guarda un valor de tipo integer.

**Postcondiciones:**

**Salida:** Un integer con el valor del atributo

**Nombre:** PasswordExpires(UserLogName: String)

**Responsabilidades:** Establecer que el password del usuario expire

**Referencias Cruzadas:** Casos de uso: Gestionar opciones password

**Precondiciones:** El usuario de nombre de login UserLogName existe en el directorio activo.

**Postcondiciones:** Se cambia el valor de el atributo que establece que expire el password del usuario.

**Salida:**

**Nombre:** DontExpirePassword(UserLogName: String)

**Responsabilidades:** Establecer que el password del usuario no expire

**Referencias Cruzadas:** Casos de uso: Gestionar opciones password

**Precondiciones:** El usuario de nombre de login UserLogName existe en el directorio activo.

**Postcondiciones:** Se cambia el valor de el atributo que establece que no expire el password del usuario.

**Salida:**

## Caso de uso 26: Gestionar fecha en la que expira la cuenta de usuario

Diagrama de secuencia:

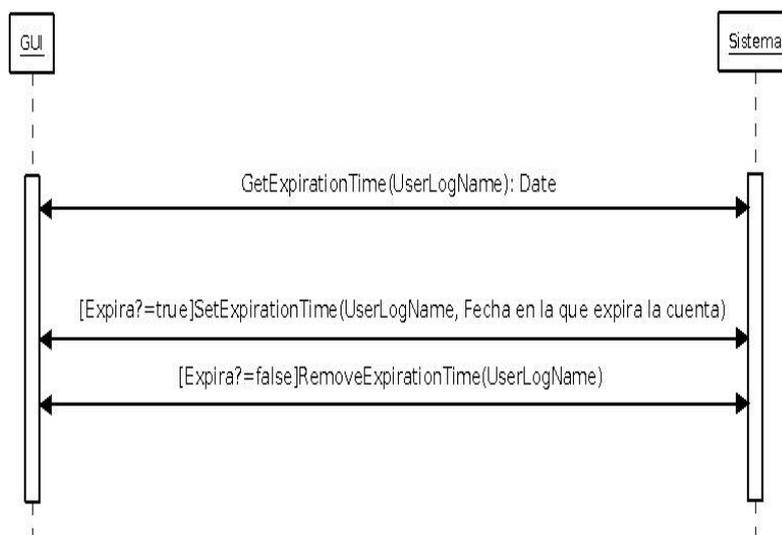


Figura 5.8.5 Gestionar fecha en la que expira la cuenta de usuario

Contratos:

**Nombre:** GetExpirationTime(UserLogName: String)

**Responsabilidades:** Obtiene la fecha en la que expira la cuenta de usuario.

**Referencias Cruzadas:** Casos de uso: Gestionar fecha en la que expira la cuenta de usuario.

**Precondiciones:** El usuario de nombre de login UserLogName existe en el directorio activo.

**Postcondiciones:** Si el dato devuelto es una fecha menor que hoy, eso quiere decir, que la cuenta nunca expira

**Salida:** Un dato de tipo fecha (date)

**Nombre:** SetExpirationTime(UserLogName: String, Fecha: Date)

**Responsabilidades:** Establece la fecha en la que expira la cuenta de usuario.

**Referencias Cruzadas:** Casos de uso: Gestionar fecha en la que expira la cuenta de usuario.

**Precondiciones:** El usuario de nombre de login UserLogName existe en el directorio activo.

**Postcondiciones:** La fecha en la que expira la cuenta de usuario será la que se le ha pasado por parámetros.

**Salida:**

**Nombre:** RemoveExpirationTime(UserLogName: String)

**Responsabilidades:** Establece que la cuenta de usuario no expira nunca.

**Referencias Cruzadas:** Casos de uso: Gestionar fecha en la que expira la cuenta de usuario.

**Precondiciones:** El usuario de nombre de login UserLogName existe en el directorio activo.

**Postcondiciones:** Modifica el atributo del objeto usuario que controla que su cuenta de usuario no expire.

**Salida:**

## Caso de uso 28: Gestionar seguridad usuario

### Diagrama de secuencia:

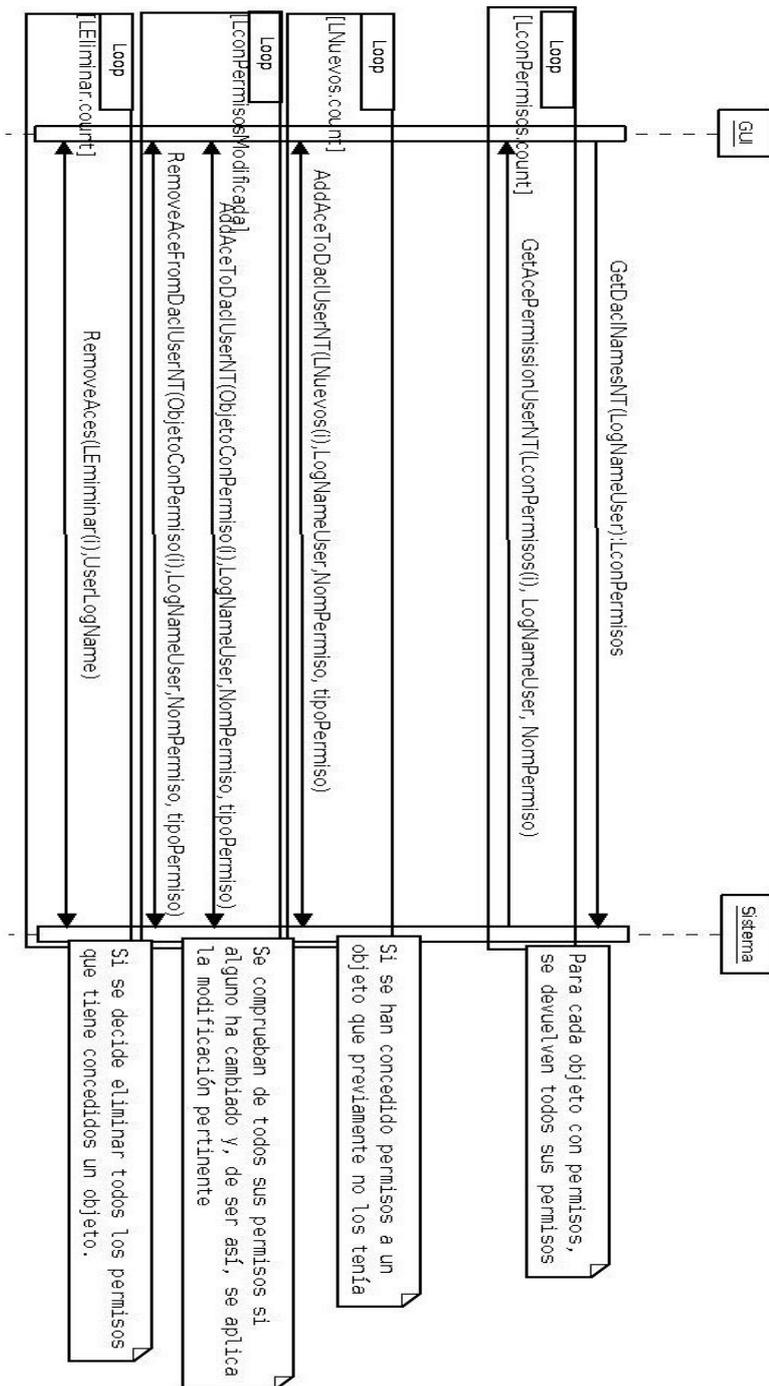


Figura 5.8.6 Gestionar seguridad usuario

Contratos:

**Nombre:** GetDaclNamesNT(UserLogName: String)

**Responsabilidades:** Obtiene la lista de los objetos con permisos sobre el objeto.

**Referencias Cruzadas:** Casos de uso: Gestionar seguridad usuario, Gestionar Seguridad grupo, Gestionar Seguridad equipo.

**Precondiciones:** El objeto de nombre de login UserLogName existe en el directorio activo.

**Postcondiciones:**

**Salida:** Una lista de nombres de objetos(string)

**Nombre:** GetAcePermissionUserNT(ObjConPermiso: String, UserLogName: String, NomPermiso: String)

**Responsabilidades:** Para el objeto de nombre "ObjConPermiso", devuelve una estructura que indica si tiene el permiso "NomPermiso" permitido o denegado sobre el usuario "UserLogName"

**Referencias Cruzadas:** Casos de uso: Gestionar Seguridad Usuario

**Precondiciones:** El usuario de nombre de login UserLogName existe en el directorio activo. Y ObjConPermiso es un objeto de la lista obtenida con GetDaclNames.

**Postcondiciones:**

**Salida:** Un array de dos booleanos, el primero indica si el permiso está permitido y el segundo si está denegado

**Nombre:** AddAceToDaclUserNT(ObjConPermiso: String, UserLogName: String, NomPermiso: String, Tipo: string)

**Responsabilidades:** Añade el permiso "NomPermiso" al objeto "ObjConPermiso" sobre el usuario "UserLogName" del tipo permitido (tipo = "allow") o denegado (tipo = "deny")

**Referencias Cruzadas:** Casos de uso: Gestionar Seguridad Usuario.

**Precondiciones:** El usuario de nombre de login UserLogName existe en el directorio activo. El objeto ObjConPermiso existe en el directorio activo. NomPermiso es un permiso del usuario.

**Postcondiciones:** Añade el permiso al atributo de seguridad del usuario

**Salida:**

**Nombre:** RemoveAceFromDaclUserNT(ObjConPermiso: String, UserLogName: String, NomPermiso: String, Tipo: string)

**Responsabilidades:** Elimina el permiso “NomPermiso” al objeto “ObjConPermiso” sobre el usuario “UserLogName” del tipo permitido (tipo = “allow”) o denegado (tipo = “deny”)

**Referencias Cruzadas:** Casos de uso: Gestionar Seguridad Usuario.

**Precondiciones:** El usuario de nombre de login UserLogName existe en el directorio activo. El objeto ObjConPermiso existe en el directorio activo. NomPermiso es un permiso del usuario.

**Postcondiciones:** Elimina el permiso al atributo de seguridad del usuario

**Salida:**

**Nombre:** RemoveAces(ObjConPermiso: String, UserLogName: String)

**Responsabilidades:** Elimina todos los permisos que posee el objeto “ObjetoConPermiso” sobre el usuario “UserLogName”.

**Referencias Cruzadas:** Casos de uso: Gestionar Seguridad Usuario, Gestionar Seguridad Buzón.

**Precondiciones:** El usuario de nombre de login UserLogName existe en el directorio activo. El objeto ObjConPermiso existe en el directorio activo.

**Postcondiciones:** Modifica el atributo de seguridad del usuario

**Salida:**

### Caso de uso 32: Gestionar tamaño mensaje

Diagrama de secuencia:

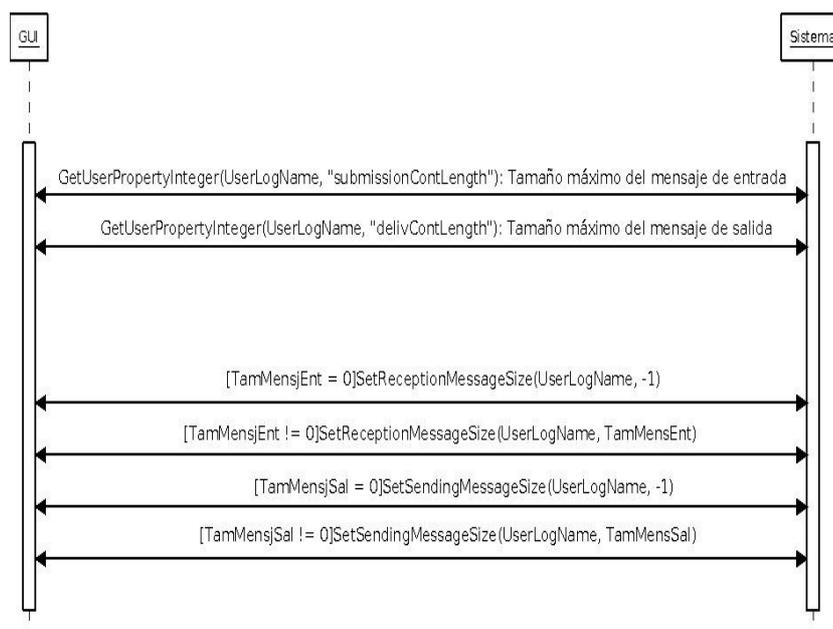


Figura 5.8.6 Gestionar tamaño mensaje

Contratos:

**Nombre:** GetUserPropertyInteger(UserLogName: String, Atributo: String)

**Responsabilidades:** Obtiene el valor de un atributo de tipo integer del objeto de tipo usuario.

**Referencias Cruzadas:** Casos de uso: Gestionar tamaño mensaje, Habilitar Servicios Móviles.

**Precondiciones:** El usuario de nombre de login UserLogName existe en el directorio activo.

**Postcondiciones:**

**Salida:** Un dato de tipo integer

**Nombre:** SetReceptionMessageSize(UserLogName: String, TamMensajeEnt: Integer)

**Responsabilidades:** Establece el tamaño máximo del mensaje de entrada del buzón del usuario.

**Referencias Cruzadas:** Casos de uso: Gestionar tamaño del mensaje.

**Precondiciones:** El usuario de nombre de login UserLogName existe en el directorio activo y debe tener un buzón Exchange asignado.

**Postcondiciones:** Asigna el valor que se obtiene por parámetros (TamMensajeEnt) al atributo que controla el tamaño máximo del mensaje de entrada.

**Salida:**

**Nombre:** SetSendingMessageSize(UserLogName: String, TamMensajeSal: Integer)

**Responsabilidades:** Establece el tamaño máximo del mensaje de salida del buzón del usuario.

**Referencias Cruzadas:** Casos de uso: Gestionar tamaño del mensaje.

**Precondiciones:** El usuario de nombre de login UserLogName existe en el directorio activo y debe tener un buzón Exchange asignado.

**Postcondiciones:** Asigna el valor que se obtiene por parámetros(TamMensajeSal) al atributo que controla el tamaño máximo del mensaje de salida.

**Salida:**

## Caso de uso 33: Gestionar emisores permitidos

Diagrama de secuencia:

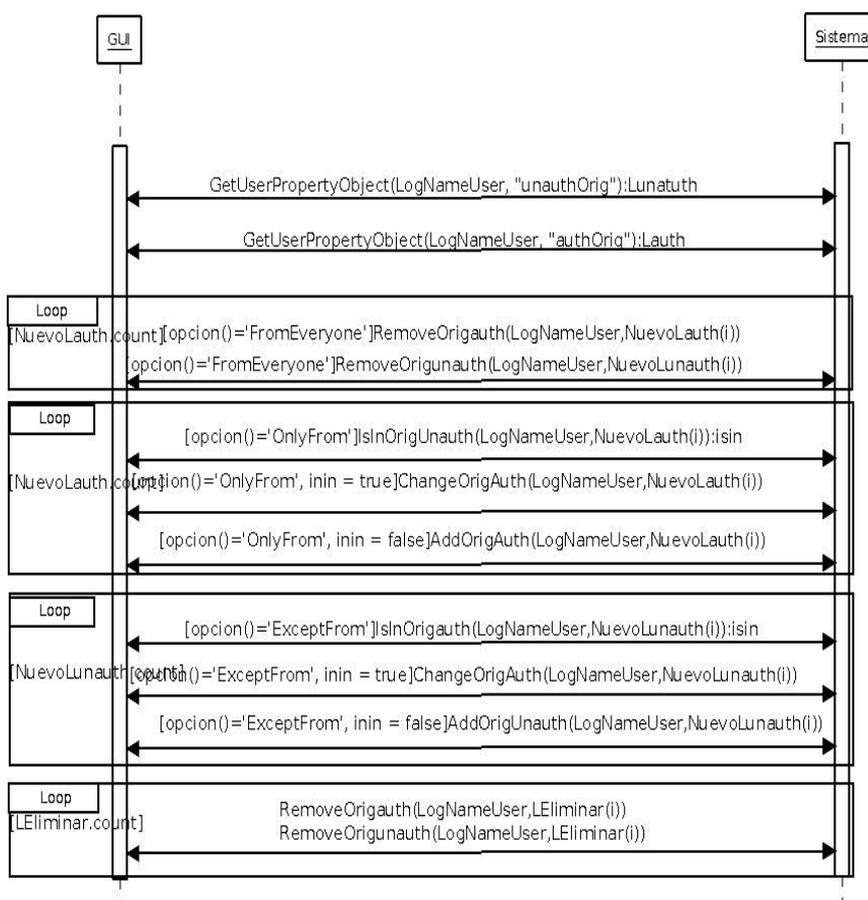


Figura 5.8.7 Gestionar emisores permitidos

### Contratos:

**Nombre:** GetUserPropertyObject(UserLogName: String, Atributo: String)

**Responsabilidades:** Obtiene el valor de un atributo, sin especificar su tipo, del objeto de tipo usuario.

**Referencias Cruzadas:** Casos de uso: Gestionar emisores permitidos, Gestionar envío como, Gestión redirección, Gestionar límite de destinatarios, Gestionar nuevo/eliminar/modificar mail, Establecer el mail como primario, Habilitar/Deshabilitar protocolos.

**Precondiciones:** El usuario de nombre de login UserLogName existe en el directorio activo.

**Postcondiciones:**

**Salida:** Un dato del que no se especifica su tipo.

## Caso de uso 35: Gestionar envío como

### Diagrama de secuencia del sistema:

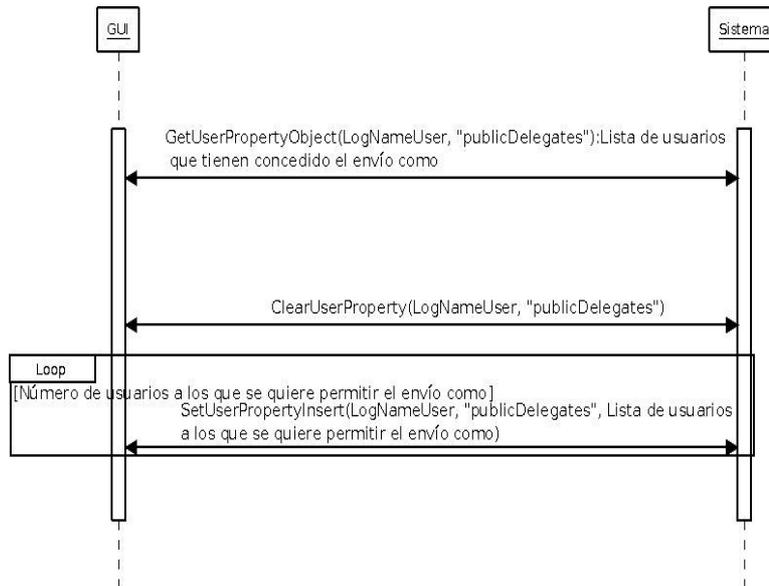


Figura 5.8.8 Gestionar envío como

### Contratos:

**Nombre:** ClearUserProperty(UserLogName: String, Atributo: String)

**Responsabilidades:** Elimina el valor almacenado en un atributo

**Referencias Cruzadas:** Casos de uso: Gestionar envío como, Gestionar nuevo/eliminar/modificar mail, Establecer mail como primario, Cambiar Simple Display.

**Precondiciones:** El usuario de nombre de login UserLogName existe y el nombre de atributo “Atributo” debe ser el nombre de un atributo del objeto.

**Postcondiciones:** Asigna el valor nulo al atributo del objeto de tipo usuario.

### Salida:

**Nombre:** SetUserPropertyInsert(UserLogName: String, Atributo: String, dato: String)

**Responsabilidades:** Asigna un nuevo dato de tipo string a un atributo del tipo lista de strings.

**Referencias Cruzadas:** Casos de uso: Gestionar emisores permitidos, Gestionar nuevo/eliminar/modificar mail, Establecer mail como primario, Habilitar/Deshabilitar protocolos.

**Precondiciones:** El usuario de nombre de login UserLogName existe y el nombre de atributo “Atributo” debe ser el nombre de un atributo del objeto. El dato tampoco puede ser un string vacío

**Postcondiciones:** Asigna el dato como último valor de la lista de strings que forman el atributo cuyo nombre se obtiene por los parámetros.

### Salida:

## Caso de uso 36: Gestión de redirección

### Diagrama de secuencia:

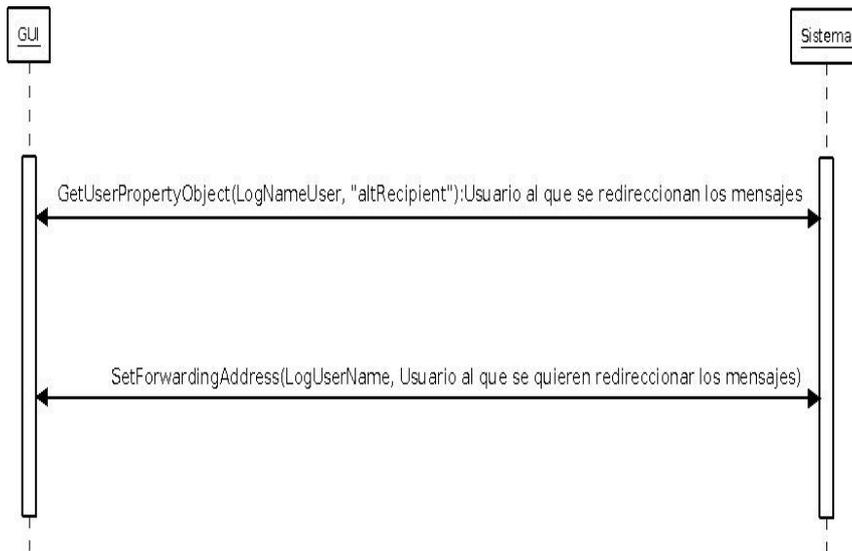


Figura 5.8.9 Gestionar de redirección

### Contratos:

**Nombre:** `SetForwardingAddress(UserLogName: String, NuevoRedirectTo: String)`

**Responsabilidades:** Establece al usuario cuyo nombre está en la variable “NuevoRedirectTo” como receptor de la redirección de los mensajes que recibe el usuario con nombre en la variable “UserLogName”

**Referencias Cruzadas:** Casos de uso: Gestión de redirección.

**Precondiciones:** Los usuarios existen en el directorio activo y tienen un buzón Exchange asignado cada uno.

**Postcondiciones:** Asigna el nombre del usuario a redireccionar al atributo que controla la redirección de los mensajes del usuario dueño del buzón.

**Salida:**

## Caso de uso 40: Establecer mail como primario

### Diagrama de secuencia:



Figura 5.8.12 Establecer mail como primario

### Contratos:

Las funciones utilizadas en este caso de uso ya se utilizan en los anteriores por lo que ya están explicadas

## Caso de uso 42: Habilitar/Deshabilitar Servicios Móviles

### Diagrama de secuencia:



Figura 5.8.13 Habilitar/Deshabilitar Servicios Móviles

### Contratos:

Las funciones utilizadas en este caso de uso ya se utilizan en los anteriores por lo que ya están explicadas

## Caso de uso 47: Gestionar seguridad buzón

### Diagrama de secuencia:

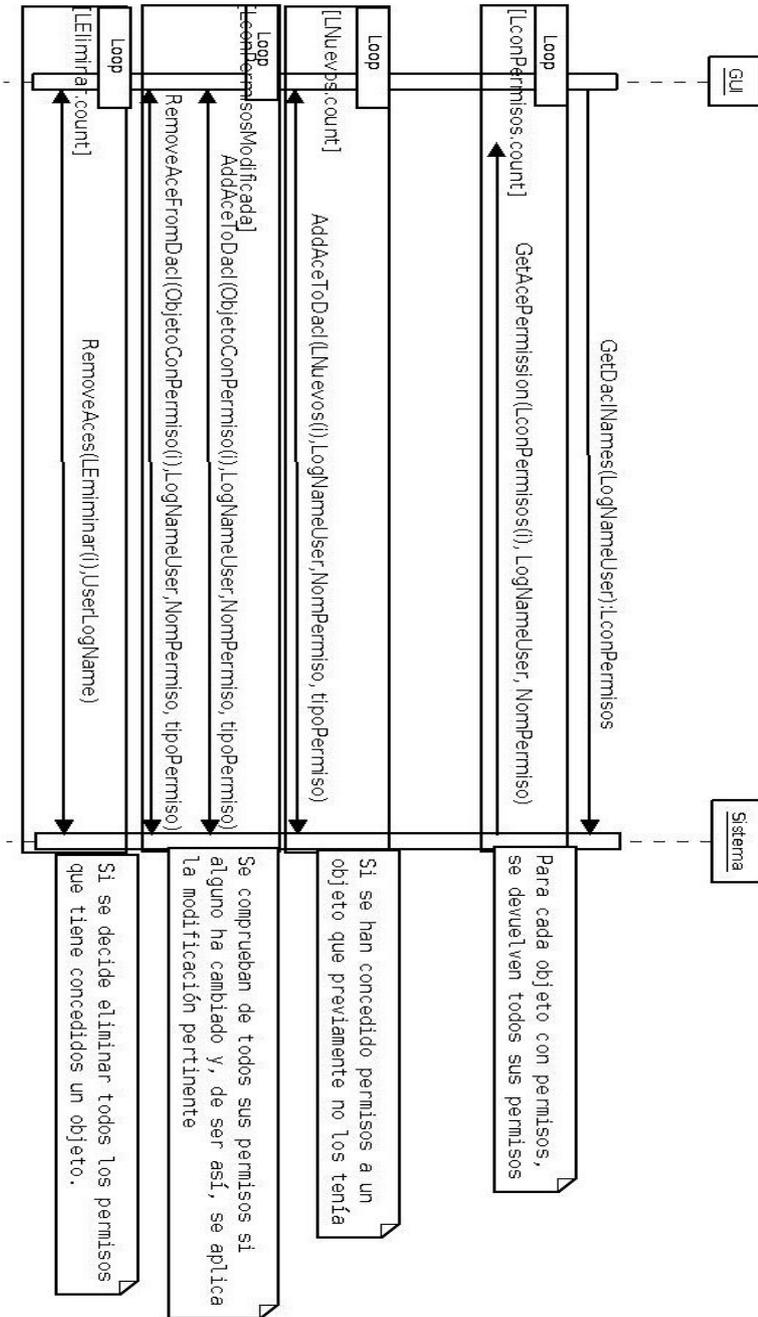


Figura 5.8.16 Gestionar Seguridad Buzón

Contratos:

**Nombre:** GetDaclNames(UserLogName: String)

**Responsabilidades:** Obtiene la lista de los objetos con permisos sobre el buzón asignado al usuario "UserLogName".

**Referencias Cruzadas:** Casos de uso: Gestionar seguridad buzón.

**Precondiciones:** El usuario de nombre de login UserLogName existe en el directorio activo y tiene buzón asignado.

**Postcondiciones:**

**Salida:** Una lista de nombres de objetos(string)

**Nombre:** GetAcePermission(ObjConPermiso: String, UserLogName: String, NomPermiso: String)

**Responsabilidades:** Para el objeto de nombre "ObjConPermiso", devuelve una estructura que indica si tiene el permiso "NomPermiso" permitido o denegado sobre el buzón del usuario "UserLogName"

**Referencias Cruzadas:** Casos de uso: Gestionar Seguridad Buzón

**Precondiciones:** El usuario de nombre de login UserLogName existe en el directorio activo y tiene buzón. Y ObjConPermiso es un objeto de la lista obtenida con GetDaclNames.

**Postcondiciones:**

**Salida:** Un array de dos booleanos, el primero indica si el permiso está permitido y el segundo si está denegado

**Nombre:** AddAceToDacl(ObjConPermiso: String, UserLogName: String, NomPermiso: String, Tipo: string)

**Responsabilidades:** Añade el permiso "NomPermiso" al objeto "ObjConPermiso" sobre el buzón del usuario "UserLogName" del tipo permitido (tipo = "allow") o denegado (tipo = "deny")

**Referencias Cruzadas:** Casos de uso: Gestionar Seguridad Buzón.

**Precondiciones:** El usuario de nombre de login UserLogName existe en el directorio activo y tiene buzón. El objeto ObjConPermiso existe en el directorio activo. NomPermiso es un permiso del buzón.

**Postcondiciones:** Añade el permiso al atributo de seguridad del buzón del usuario

**Salida:**

## Caso de uso 21: Gestionar grupos de los que el objeto es miembro

Diagrama de secuencia:



Figura 5.8.17 Gestionar grupos de los que el objeto es miembro

Contratos:

**Nombre:** GetUserGroups(UserLogName: String)

**Responsabilidades:** Obtiene la lista de los grupos de los que es miembro un objeto de tipo usuario.

**Referencias Cruzadas:** Casos de uso: Gestionar grupos de los que el objeto es miembro.

**Precondiciones:** El usuario de nombre de login UserLogName existe en el directorio activo.

**Postcondiciones:**

**Salida:** Lista de strings.

**Nombre:** GetGroupGroups(GroupLogName: String)

**Responsabilidades:** Obtiene la lista de los grupos de los que es miembro un objeto de tipo grupo.

**Referencias Cruzadas:** Casos de uso: Gestionar grupos de los que el objeto es miembro.

**Precondiciones:** El grupo de nombre de login GroupLogName existe en el directorio activo.

**Postcondiciones:**

**Salida:** Lista de strings.

**Nombre:** GetComputerGroups(ComputerLogName: String)

**Responsabilidades:** Obtiene la lista de los grupos de los que es miembro un objeto de tipo equipo.

**Referencias Cruzadas:** Casos de uso: Gestionar grupos de los que el objeto es miembro.

**Precondiciones:** El equipo de nombre de login ComputerLogName existe en el directorio activo.

**Postcondiciones:**

**Salida:** Lista de strings.

**Nombre:** ClearProperty(ObjectLogName: String, Atributo: string)

**Responsabilidades:** Elimina el valor almacenado en un atributo

**Referencias Cruzadas:** Casos de uso: Gestionar grupos de los que el objeto es miembro, Gestionar opciones generales grupo, Gestionar opciones generales equipo.

**Precondiciones:** El objeto de nombre de login ObjectLogName existe en el directorio activo y el .nombre de atributo que se pasa por parámetros “Atributo” debe ser el nombre de un atributo del objeto.

**Postcondiciones:** Asigna el valor nulo al atributo del objeto.

**Salida:**

**Nombre:** SetGroupPropertyInsert(GroupLogName: String, Atributo: String, dato: String)

**Responsabilidades:** Asigna un nuevo dato de tipo string a un atributo del tipo lista de strings.

**Referencias Cruzadas:** Casos de uso: Gestionar grupos de los que el objeto es miembro, Gestionar miembros grupo.

**Precondiciones:** El grupo de nombre de login GroupLogName existe en el directorio activo y el .nombre de atributo que se pasa por parámetros “Atributo” debe ser el nombre de un atributo del objeto de tipo grupo. El dato tampoco puede ser un string vacío

**Postcondiciones:** Asigna el dato como último valor de la lista de strings que forman el atributo.

**Salida:**



## Caso de uso 20: Consultar datos canónicos del objeto

### Diagrama de secuencia:



Figura 5.8.18 Consultar datos canónicos del objeto

### Contratos:

**Nombre:** GetPropertyRefresh(ObjectLogName: String, Atributo: String)

**Responsabilidades:** Obtiene el valor de un atributo de tipo string, previo refresco de la memoria.

**Referencias Cruzadas:** Casos de uso: Consultar datos canónicos del objeto

**Precondiciones:** El objeto existe en el directorio activo.

**Postcondiciones:**

**Salida:** Un dato de tipo string

## 5.2.2 Casos de uso relativos a las propiedades de grupo

**Nombre:** SetGroupProperty(GroupLogName: String, Atributo: string, dato: String)

**Responsabilidades:** Asigna un valor de tipo string a un atributo del objeto del tipo grupo.

**Referencias Cruzadas:** Casos de uso: Gestionar opciones generales de grupo, Gestionar nombres buzón, Gestionar nombre servidor buzón.

**Precondiciones:** El objeto existe en el directorio activo y el atributo pertenece a un objeto de tipo grupo.

**Postcondiciones:** El dato "dato" se asigna al atributo "Atributo" del objeto grupo "GroupLogName".

**Salida:**

### Caso de uso 49: Gestionar miembros del grupo

Diagrama de secuencia:



Figura 5.9.2 Gestionar miembros del grupo

Contratos:

**Nombre:** GetGroupPropertyObject(GroupLogName: String, Atributo: String)

**Responsabilidades:** Obtiene el valor de un atributo sin especificar su tipo.

**Referencias Cruzadas:** Casos de uso: Gestionar miembros del grupo.

**Precondiciones:** El grupo existe en el directorio activo y el atributo es un atributo de un objeto del tipo grupo.

**Postcondiciones:**

**Salida:** Un dato sin especificar su tipo

**Nombre:** SetGroupPropertyRemove(GroupLogName: String, Atributo: String)

**Responsabilidades:** Elimina el valor de un atributo.

**Referencias Cruzadas:** Casos de uso: Gestionar miembros del grupo.

**Precondiciones:** El objeto existe en el directorio activo y el atributo es un atributo de un objeto del tipo grupo.

**Postcondiciones:** Se establece como nulo el valor del atributo "Atributo" del grupo "GroupLogName".

Salida:

### Caso de uso 55: Gestionar opciones recepción

Diagrama de secuencia:

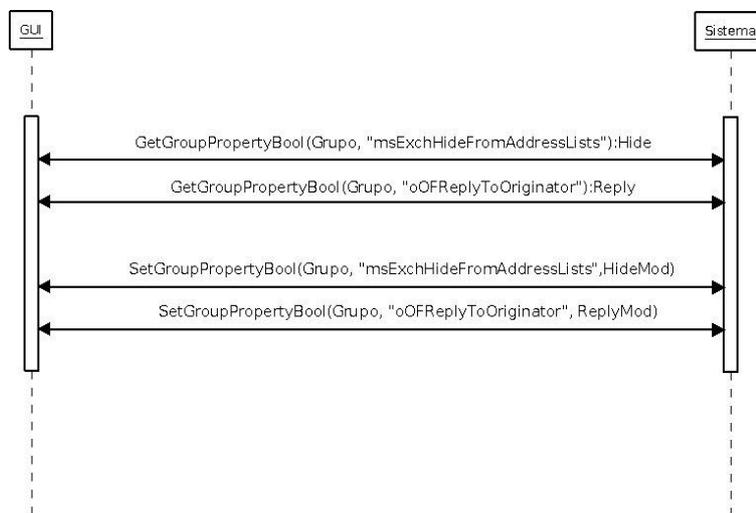


Figura 5.9.4 Gestionar Opciones recepción

Contratos:

**Nombre:** GetGroupPropertyBool(GroupLogName: String, Atributo: String)

**Responsabilidades:** Obtiene el valor de un atributo de tipo boolean.

**Referencias Cruzadas:** Casos de uso: Gestionar opciones recepción, Gestión Envío informes.

**Precondiciones:** El grupo existe en el directorio activo y el atributo es un atributo de un objeto del tipo grupo.

**Postcondiciones:**

**Salida:** Un dato de tipo boolean

**Nombre:** SetGroupPropertyBool(GroupLogName: String, Atributo: String, dato: boolean)

**Responsabilidades:** Asigna un valor de tipo boolean a un atributo del objeto del tipo grupo.

**Referencias Cruzadas: Casos de uso:** Gestionar opciones recepción, Gestión Envío Informes.

**Precondiciones:** El objeto existe en el directorio activo y el atributo pertenece a un objeto de tipo grupo.

**Postcondiciones:** El dato “dato” se asigna al atributo “Atributo” del objeto grupo “GroupLogName”.

**Salida:**

### Caso de uso 56: Gestionar envío informes

Diagrama de secuencia:



Figura 5.9.5 Gestionar envío informes

Contratos:

Las funciones utilizadas en este caso de uso ya se utilizan en los anteriores por lo que ya están explicadas

## Caso de uso 60: Gestionar Administrador Objeto

Diagrama de secuencia:



Figura 5.9.6 Gestionar Administrador Objeto

### Contratos:

**Nombre:** GetComputerProperty(ComputerLogName: String, Atributo: String)

**Responsabilidades:** Obtiene el valor de un atributo de tipo string.

**Referencias Cruzadas:** Casos de uso: Gestionar administrador objeto, Gestionar opciones generales equipo, Consultar datos del Sistema Operativo.

**Precondiciones:** El equipo existe en el directorio activo y el atributo es un atributo de un objeto del tipo equipo.

**Postcondiciones:**

**Salida:** Un dato de tipo string

**Nombre:** SetComputerProperty(ComputerLogName: String, Atributo: String, dato: boolean)

**Responsabilidades:** Asigna un valor de tipo string a un atributo del objeto del tipo equipo.

**Referencias Cruzadas:** Casos de uso: Gestionar administrador objeto, Gestionar opciones generales equipo.

**Precondiciones:** El equipo existe en el directorio activo y el atributo pertenece a un objeto de tipo equipo.

**Postcondiciones:** El dato "dato" se asigna al atributo "Atributo" del objeto grupo "ComputerLogName".

**Salida:**

## Caso de uso 59: Dar Permisos al administrador para gestionar miembros del grupo

Diagrama de secuencia:

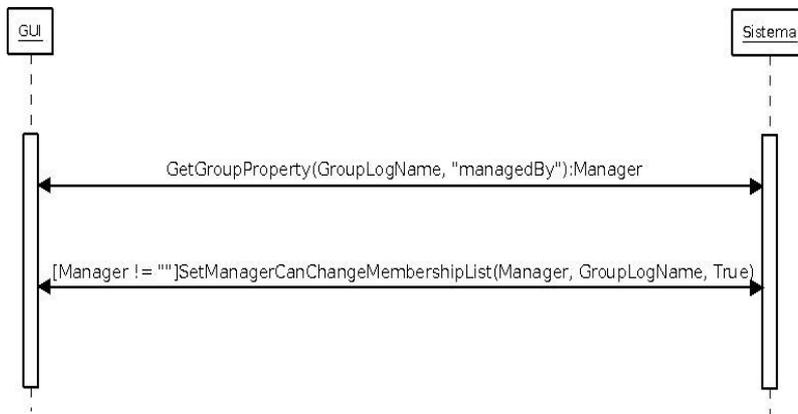


Figura 5.9.7 Dar permisos al administrador para gestionar miembros del grupo

Contratos:

**Nombre:** SetManagerCanChangeMembershipList(Manager: String, GroupLogName: String, Permiso: boolean)

**Responsabilidades:** Asigna o deniega permisos (según indique el booleano Permiso) para gestionar los miembros del grupo al administrador del mismo.

**Referencias Cruzadas:** Casos de uso: Dar Permisos al administrador para gestionar miembros del grupo.

**Precondiciones:** El objeto existe en el directorio activo y el usuario “manager” es su administrador.

**Postcondiciones:** Modifica el atributo de seguridad del grupo para conceder dicho permiso al usuario “Manager”.

**Salida:**









### Contratos:

**Nombre:** GetAcePermissionNT(ObjConPermiso: String, GroupLogName: String, NomPermiso: String)

**Responsabilidades:** Para el objeto de nombre “ObjConPermiso”, devuelve una estructura que indica si tiene el permiso “NomPermiso” permitido o denegado sobre el grupo “GroupLogName”

**Referencias Cruzadas:** Casos de uso: Gestionar Seguridad Grupo

**Precondiciones:** El grupo de nombre de login GroupLogName existe en el directorio activo. Y ObjConPermiso es un objeto de la lista obtenida con GetDaclNames.

**Postcondiciones:**

**Salida:** Un array de dos booleanos, el primero indica si el permiso está permitido y el segundo si está denegado

**Nombre:** AddAceToDaclNT(ObjConPermiso: String, GroupLogName: String, NomPermiso: String, Tipo: string)

**Responsabilidades:** Añade el permiso “NomPermiso” al objeto “ObjConPermiso” sobre el grupo “GroupLogName” del tipo permitido (tipo = “allow”) o denegado (tipo = “deny”)

**Referencias Cruzadas:** Casos de uso: Gestionar Seguridad Grupo.

**Precondiciones:** El grupo de nombre de login GroupLogName existe en el directorio activo. El objeto ObjConPermiso existe en el directorio activo. NomPermiso es un permiso del grupo.

**Postcondiciones:** Añade el permiso al atributo de seguridad del grupo

**Salida:**

**Nombre:** RemoveAceFromDaclNT(ObjConPermiso: String, GroupLogName: String, NomPermiso: String, Tipo: string)

**Responsabilidades:** Elimina el permiso “NomPermiso” al objeto “ObjConPermiso” sobre el grupo “GroupLogName” del tipo permitido (tipo = “allow”) o denegado (tipo = “deny”)

**Referencias Cruzadas:** Casos de uso: Gestionar Seguridad Grupo.

**Precondiciones:** El grupo de nombre de login GroupLogName existe en el directorio activo. El objeto ObjConPermiso existe en el directorio activo. NomPermiso es un permiso grupo.

**Postcondiciones:** Elimina el permiso al atributo de seguridad del grupo

**Salida:**

### 5.2.3 Casos de uso relativos a las propiedades del equipo

#### Caso de uso 61: Gestionar Opciones generales equipo

Diagrama de secuencia:



Figura 5.10.1 Gestionar Opciones generales Equipo

#### Contratos:

Las funciones utilizadas en este caso de uso ya se utilizan en los anteriores por lo que ya están explicadas

#### Caso de uso 62: Consultar datos del Sistema operativo

Diagrama de secuencia:



Figura 5.10.2 Consultar datos del Sistema operativo

#### Contratos:

Las funciones utilizadas en este caso de uso ya se utilizan en los anteriores por lo que ya están explicadas

## Caso de uso 63: Gestionar seguridad equipo

### Diagrama de secuencia del sistema:

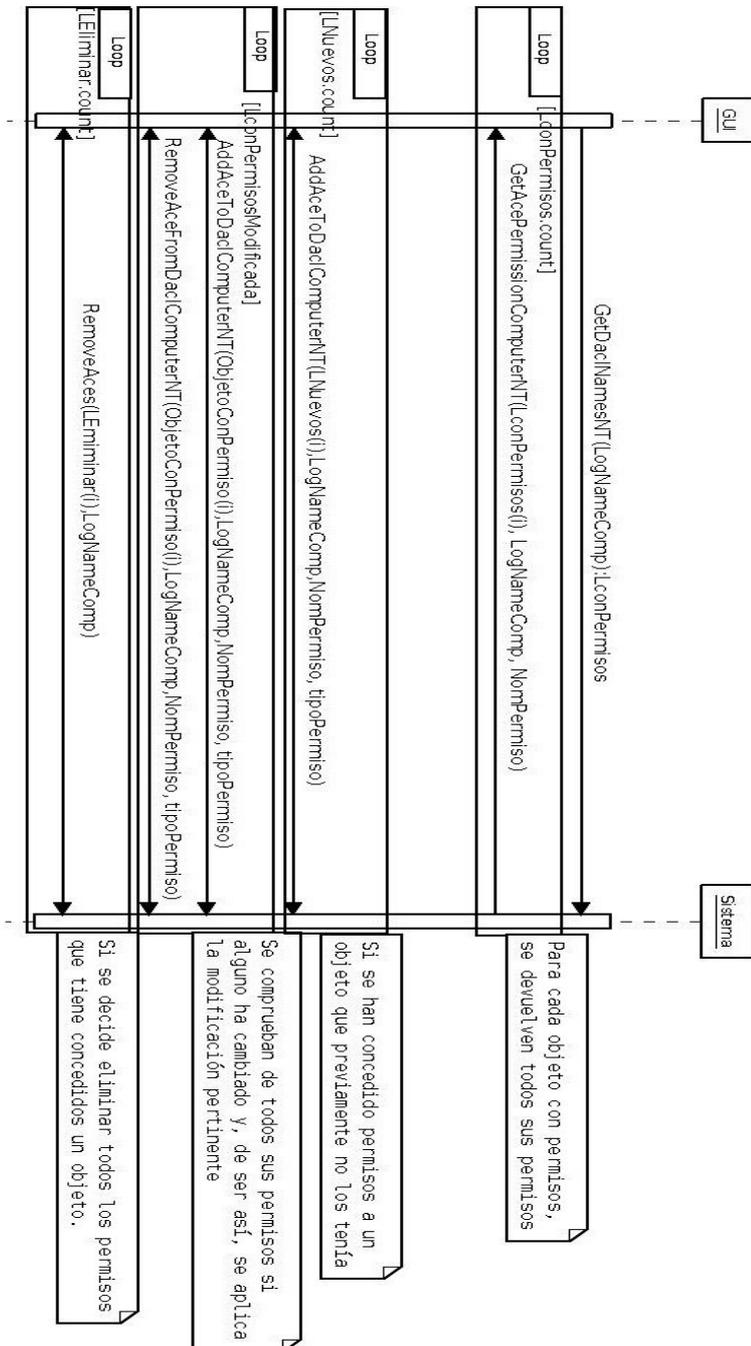


Figura 5.10.3 Gestionar Seguridad Equipo

### Contratos:

**Nombre:** GetAcePermissionComputerNT(ObjConPermiso: String, ComputerLogName: String, NomPermiso: String)

**Responsabilidades:** Para el objeto de nombre “ObjConPermiso”, devuelve una estructura que indica si tiene el permiso “NomPermiso” permitido o denegado sobre el equipo “ComputerLogName”

**Referencias Cruzadas:** Casos de uso: Gestionar Seguridad Equipo

**Precondiciones:** El equipo de nombre de login ComputerLogName existe en el directorio activo. Y ObjConPermiso es un objeto de la lista obtenida con GetDaclNames.

**Postcondiciones:**

**Salida:** Un array de dos booleanos, el primero indica si el permiso está permitido y el segundo si está denegado

**Nombre:** AddAceToDaclComputerNT(ObjConPermiso: String, ComputerLogName: String, NomPermiso: String, Tipo: string)

**Responsabilidades:** Añade el permiso “NomPermiso” al objeto “ObjConPermiso” sobre el equipo “ComputerLogName” del tipo permitido (tipo = “allow”) o denegado (tipo = “deny”)

**Referencias Cruzadas:** Casos de uso: Gestionar Seguridad Equipo.

**Precondiciones:** El equipo de nombre de login ComputerLogName existe en el directorio activo. El objeto ObjConPermiso existe en el directorio activo. NomPermiso es un permiso del equipo.

**Postcondiciones:** Añade el permiso al atributo de seguridad del equipo

**Salida:**

**Nombre:** RemoveAceFromDaclComputerNT(ObjConPermiso: String, ComputerLogName: String, NomPermiso: String, Tipo: string)

**Responsabilidades:** Elimina el permiso “NomPermiso” al objeto “ObjConPermiso” sobre el equipo “ComputerLogName” del tipo permitido (tipo = “allow”) o denegado (tipo = “deny”)

**Referencias Cruzadas:** Casos de uso: Gestionar Seguridad Equipo.

**Precondiciones:** El equipo de nombre de login ComputerLogName existe en el directorio activo. El objeto ObjConPermiso existe en el directorio activo. NomPermiso es un permiso de el equipo.

**Postcondiciones:** Elimina el permiso al atributo de seguridad del equipo

**Salida:**

## 5.3 Casos de uso de la tercera iteración

### 5.3.1 Caso de uso buscar objeto

#### Caso de uso 2: Buscar Objetos

Diagrama de secuencia:

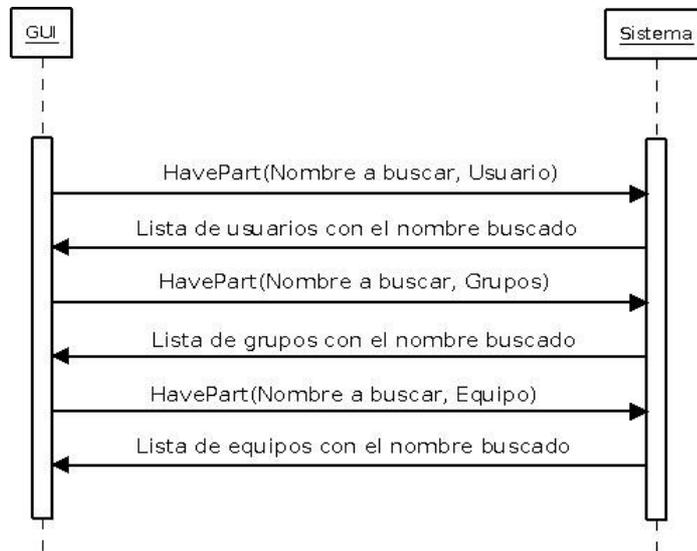


Figura 5.2.1 Buscar Objetos

Contratos:

**Nombre:** HavePart(ToSearch:String, tipo:String)

**Responsabilidades:** Devuelve una lista con los nombres de los objetos del tipo que el atributo tipo marque y que su nombre, nombre de login, o nombre de display empiecen por la cadena que contiene el atributo Tosearch.

**Referencias Cruzadas:** Casos de uso: Buscar Objetos

**Precondiciones:**

**Postcondiciones:** Los nombres de la lista, contienen la cadena de caracteres y por medio de la interfaz se devolverán al usuario de la aplicación.

**Salida:** Una lista con nombres de objetos de un determinado tipo del directorio activo.

### 5.3.2 Caso de uso Resetear contraseña usuario

#### Caso de uso 15: Resetear contraseña usuario

Diagrama de secuencia:



Figura 5.8.1 Resetear contraseña usuario

Contratos:

**Nombre:** ChangePassword(UserLogName: String, NuevoPassword: String)

**Responsabilidades:** Cambia la contraseña del usuario

**Referencias Cruzadas:** Casos de uso: Resetear contraseña usuario

**Precondiciones:** Vía interfaz se limita al agente a seleccionar objetos de tipo usuario que existen en el directorio activo.

**Postcondiciones:** Cambia el atributo que establece la contraseña

**Salida:**







### 5.3.3 Caso de uso Desbloquear usuario

#### **Caso de uso 16: Desbloquear usuario**

Diagrama de secuencia:



Figura 5.8.2 Desbloquear usuario

#### Contratos:

**Nombre:** UnlockUser(UserLogName: String)

**Responsabilidades:** Desbloquea el usuario

**Referencias Cruzadas:** Casos de uso: Desbloquear usuario

**Precondiciones:** Vía interfaz se limita al agente a seleccionar objetos de tipo usuario que existen en el directorio activo y están bloqueados.

**Postcondiciones:** Cambia el atributo que establece si el usuario está bloqueado o no.

**Salida:**

### 5.3.4 Caso de uso Mover objeto

#### Caso de uso 12: Mover Objeto

Diagrama de secuencia:



Figura 5.7.1 Mover Objeto

Contratos:

**Nombre:** CrearArbol(RaizDA: String, NodoRaiz: TreeNode)

**Responsabilidades:** Crea un árbol con la estructura de el directorio activo, esto es, la raíz y sus OU's.

**Referencias Cruzadas:** Casos de uso: Mover Objeto

**Precondiciones:** La RaizDA se tratará de la raíz del dominio, el nodo se forma con su dirección.

**Postcondiciones:** Crea el árbol en la interfaz

**Salida:**

**Nombre:** MoveTo(NombreObjeto: String, OUSeleccionada: String)

**Responsabilidades:** Mueve el objeto de su dirección actual a la OU seleccionada del árbol

**Referencias Cruzadas:** Casos de uso: Mover Objeto

**Precondiciones:** La OU seleccionada en el árbol de la interfaz no puede ser la raíz

**Postcondiciones:** Mueve el objeto, que pasará a formar parte de los elementos de la OU seleccionada.

**Salida:**

### 5.3.5 Caso de uso Eliminar objeto

#### Caso de uso 13: Eliminar Objeto

Diagrama de secuencia:

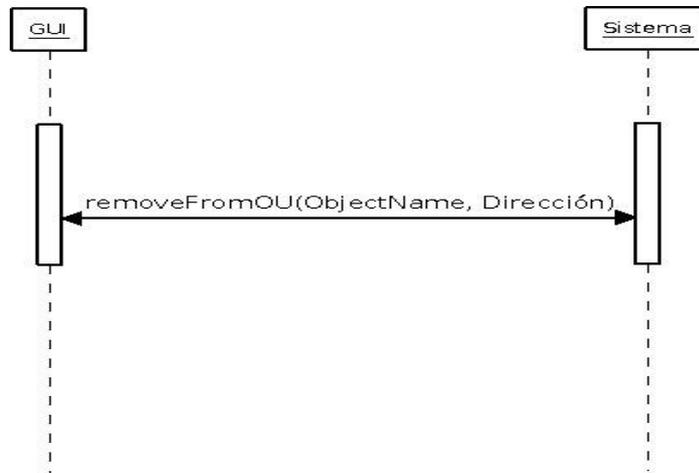


Figura 5.7.3 Eliminar Objeto

Contratos:

**Nombre:** removeFromOU(ObjectName: String, Dirección: String)

**Responsabilidades:** Elimina un objeto del directorio activo.

**Referencias Cruzadas:** Casos de uso: Eliminar Objeto

**Precondiciones:**

**Postcondiciones:** Elimina la cuenta del objeto del directorio activo.

**Salida:**

Contratos:

**Nombre:** CargarServidoresGeneral()

**Responsabilidades:** Devuelve la lista de los servidores que pertenecen al directorio activo.

**Referencias Cruzadas:** Casos de uso: Gestionar carpetas, Gestionar nombre servidor buzón.

**Precondiciones:**

**Postcondiciones:**

**Salida:** Una lista de nombres (strings) de los servidores pertenecientes al directorio activo.

**Nombre:** IsDirectory(DireccionCarpeta: String)

**Responsabilidades:** Establece si la carpeta cuya dirección se pasa por parámetros en un directorio o no.

**Referencias Cruzadas:** Casos de uso: Gestionar carpetas

**Precondiciones:**

**Postcondiciones:**

**Salida:** Un booleano “esDirectorio” que tendrá el valor true si la carpeta es un directorio y false en caso contrario.

**Nombre:** createFolderArbol(DireccionCarpeta: String, NuevaCarNombre: String)

**Responsabilidades:** Crea una nueva carpeta dentro de la carpeta seleccionada

**Referencias Cruzadas:** Casos de uso: Gestionar Carpetas

**Precondiciones:** La carpeta padre cuya dirección se pasa por parámetros “DireccionCarpeta” ha de ser un directorio. El nombre de la nueva carpeta NuevaCarNombre, no puede ser un string con valor vacío.

**Postcondiciones:** Se crea la subcarpeta con el nuevo nombre dentro de la seleccionada en el árbol de la interfaz.

**Salida:**

**Nombre:** renameFolder(DirecciónCarpeta: String, NuevoNombre: String)

**Responsabilidades:** Renombra la carpeta cuya dirección se obtiene por parámetros “DireccionCarpeta” con el nombre “NuevoNombre”

**Referencias Cruzadas:** Casos de uso: Gestionar Carpetas

**Precondiciones:** La carpeta que se quiere renombrar ha de ser un directorio. El nuevo nombre de la carpeta no puede ser un string vacío.

**Postcondiciones:** Se renombra la carpeta seleccionada en el árbol de la interfaz.

**Salida:**

### 5.3.6 Caso de uso Cambiar nombre del objeto

#### **Caso de uso 17: Cambiar el nombre al Objeto**

Diagrama de secuencia:

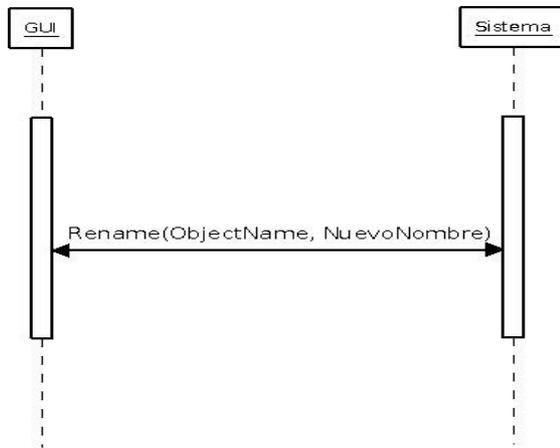


Figura 5.7.4 Cambiar Nombre al Objeto

Contratos:

**Nombre:** Rename(ObjectName: String, NuevoNombre: String)

**Responsabilidades:** Cambia el nombre del objeto.

**Referencias Cruzadas:** Casos de uso: Cambiar el nombre al objeto

**Precondiciones:** Que el nuevo nombre no exista ya en el directorio activo.

**Postcondiciones:** Cambia el atributo de dirección y nombre del objeto.

**Salida:**

### 5.3.7 Caso de uso Cambiar nombre del objeto

#### **Caso de uso 18: Habilitar/Deshabilitar usuario o equipo**

Diagrama de secuencia:

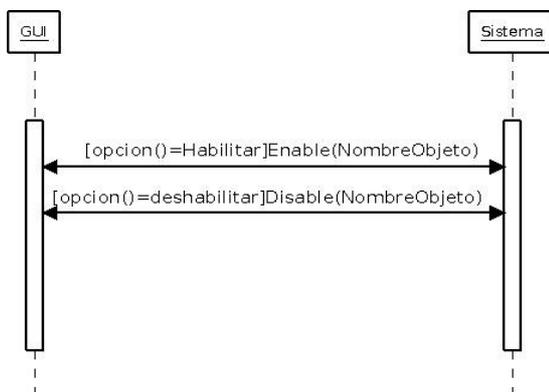


Figura 5.75 Habilitar/Deshabilitar usuario o equipo

### 5.3.8 Casos de uso relativos a las Tareas Exchange

#### Caso de uso 68: Mover buzón

Diagrama de secuencia:

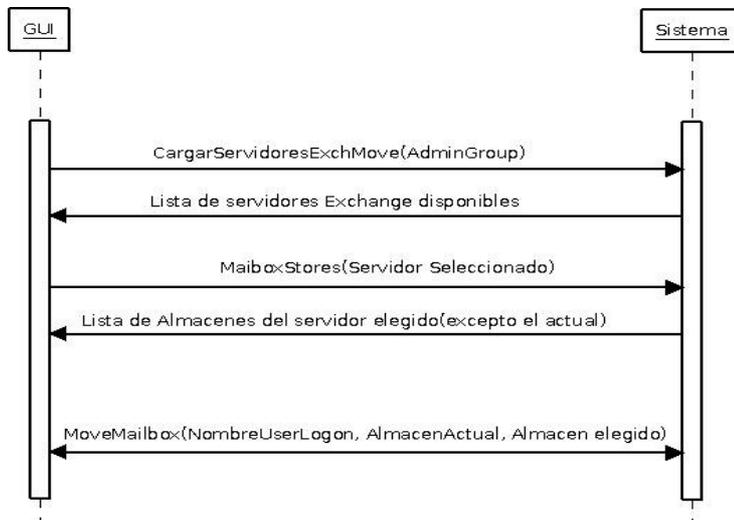


Figura 5.11.1 Mover Buzón

Contratos:

**Nombre:** CargarServidores(AdminGroup: String)

**Responsabilidades:** Obtiene la lista de los servidores a los que se puede mover el buzón. Es decir, el servidor al que pertenece el almacén actual, si es que tiene más almacenes, y el resto de servidores del grupo administrativo, si el correo Exchange está configurado en modo mixto, y el resto de servidores Exchange del directorio activo, estén donde estén; si el correo Exchange está configurado en modo nativo.

**Referencias Cruzadas:** Casos de uso: Realizar tareas Exchange usuario

**Precondiciones:** El usuario existe en el directorio activo y tiene un buzón asignado.

**Postcondiciones:** Se realiza una búsqueda de los servidores, decidiendo en ella si pueden o no pueden ser candidatos a enviar el buzón a uno de sus almacenes.

**Salida:** Una lista de objetos de tipo string.

**Nombre:** MoveMailbox(NombreUserLogon: String, AlmacenActual: String, AlmacenElegido: String)

**Responsabilidades:** Mueve el buzón del usuario, del almacén de buzones actual al almacén de buzones elegido.

**Referencias Cruzadas:** Casos de uso: Realizar tareas Exchange usuario.

**Precondiciones:** El usuario existe en el directorio activo y tiene un buzón asignado.

**Postcondiciones:**

**Salida:**

## Caso de uso 67: Eliminar buzón

Diagrama de secuencia:



Figura 5.11.2 Mover Buzón

Contratos:

**Nombre:** DeleteMailbox(NombreUserLogon: String)

**Responsabilidades:** Elimina el buzón del usuario.

**Referencias Cruzadas:** Casos de uso: Realizar tareas Exchange usuario.

**Precondiciones:** El usuario existe en el directorio activo y tiene un buzón asignado.

**Postcondiciones:** Se elimina el buzón de correo del directorio activo.

**Salida:**

## Caso de uso 69: Eliminar Mail Grupo

Diagrama de secuencia:



Figura 5.113 Eliminar Mail Grupo

Contratos:

**Nombre:** Group\_MailDisable(NombreGrupo: String)

**Responsabilidades:** Elimina las direcciones de correo del grupo.

**Referencias Cruzadas:** Casos de uso: Realizar tareas Exchange grupo.

**Precondiciones:** El grupo existe en el directorio activo y tiene por lo menos una dirección de correo asignada.

**Postcondiciones:** Se elimina el correo del grupo modificando sus atributos.

**Salida:**

## Caso de uso 70: Establecer Mail Grupo

Diagrama de secuencia del sistema:

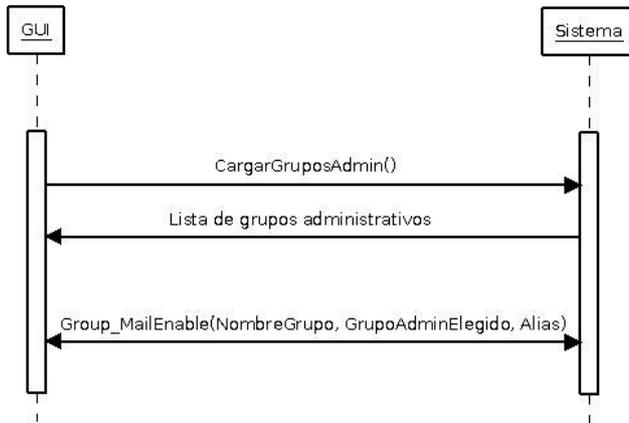


Figura 5.11.4 Establecer Mail Grupo

### Contratos:

**Nombre:** CargarGruposAdmin()

**Responsabilidades:** Obtiene los grupos administrativos del Servicio de correo Exchange del Directorio Activo. Y devuelve en una lista con sus nombres.

**Referencias Cruzadas:** Casos de uso: Realizar tareas Exchange grupo

**Precondiciones:** .

**Postcondiciones:** Se realiza una búsqueda de los objetos de tipo Grupo Administrativo en la configuración Exchange.

**Salida:** Una lista de objetos de tipo string.

**Nombre:** Group\_MailEnable(NombreUserLogon: String, GrupoAdminElegido: String, Alias: String)

**Responsabilidades:** Mueve el buzón del usuario, del almacén de buzones actual al almacén de buzones elegido.

**Referencias Cruzadas:** Casos de uso: Realizar tareas Exchange grupo.

**Precondiciones:** El grupo existe en el directorio activo y no tiene una dirección de correo asignada.

**Postcondiciones:** Modifica ciertos atributos del objeto grupo para que, éste, pueda utilizar una cuenta de correo cuya estructura será el alias seguido de arroba y su dominio.

**Salida:**

## Caso de uso: Crear Buzón

### Diagrama de secuencia:

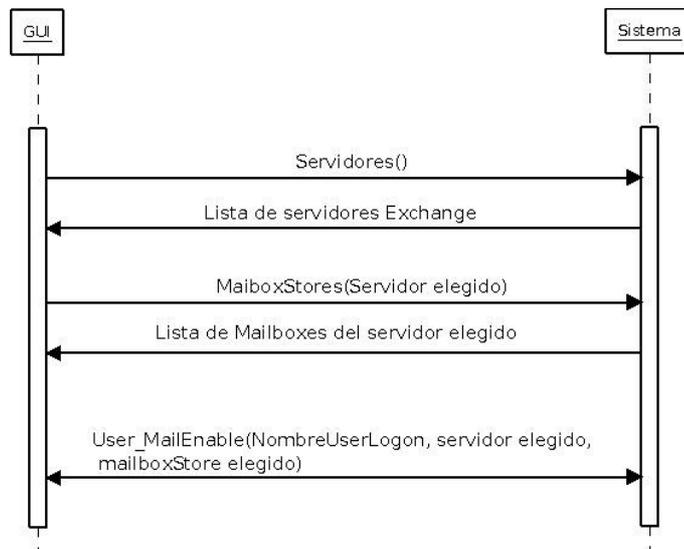


Figura 5.11.5 Crear Buzón

### Contratos:

**Nombre:** Servidores()

**Responsabilidades:** Devuelve la lista con los nombres de los servidores Exchange del directorio activo.

**Referencias Cruzadas:** Casos de uso: Crear Buzón

**Precondiciones:**

**Postcondiciones:**

**Salida:** Una lista con los nombres de los servidores Exchange del directorio activo.

**Nombre:** MailboxStores(Servidor: String)

**Responsabilidades:** Devuelve la lista de los nombres de los MailboxStores del servidor, cuyo nombre que se obtiene por parámetros.

**Referencias Cruzadas:** Casos de uso: Crear Buzón, Mover Buzón

**Precondiciones:** El parámetro Servidor no puede ser un string vacío y debe ser un servidor Exchange del directorio activo.

**Postcondiciones:**

**Salida:** Una lista con los nombres de los Mailboxes del servidor.

**Nombre:** User\_MailEnable(NombreUserLogon, servidor, mailboxStore)

**Responsabilidades:** Devuelve la lista de los nombres de los equipos del directorio activo.

**Referencias Cruzadas:** Casos de uso: Buscar Objetos

**Precondiciones:** El nombre de login del usuario se posee, ya que este caso de uso extiende del de crear usuario.

**Postcondiciones:** Se crea el buzón en el mailboxStore elegido del servidor elegido y se le asigna al usuario.

**Salida:** Una lista con los nombres de los equipos del directorio activo.

## 5.4 Casos de uso de la cuarta iteración

### 5.4.1 Caso de uso cambiar configuración IExplorer

**Caso de uso 4: Cambiar configuración Internet Explorer**

Diagrama de secuencia:

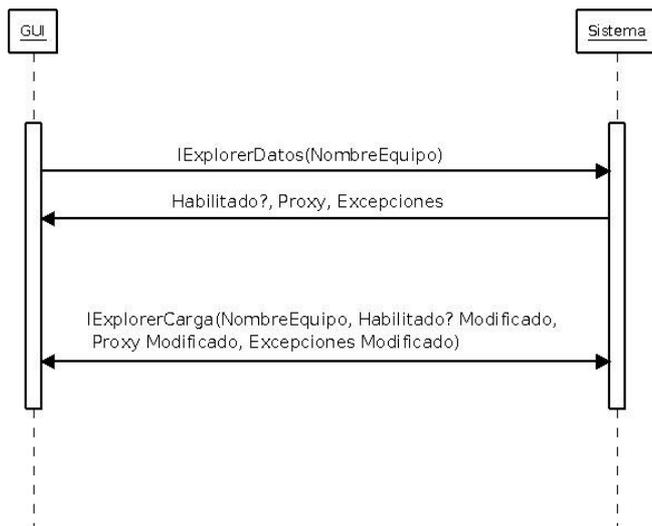


Figura 5.4.1 Cambiar configuración Internet Explorer

Contratos:

**Nombre:** IexplorerDatos(NombreEquipo: String)

**Responsabilidades:** Devuelve los datos proxy y excepciones de proxy del Internet Explorer, además de si está habilitada la opción de utilizar ese proxy con esas excepciones a la hora de conectarse.

**Referencias Cruzadas:** Casos de uso: Cambiar configuración Internet Explorer

**Precondiciones:** El nombre del equipo debe ser el de un equipo que exista en la red y que posea la aplicación InternetExplorer instalada.

**Postcondiciones:** Los datos, posteriormente, se mostrarán por la interfaz.

**Salida:** Devuelve los datos proxy y excepciones que serán dos string y el dato habilitado que será un booleano

**Nombre:** IExplorerCarga(NombreEquipo: String, Habilitado?: boolean, Proxy: String. Excepciones: String)

**Responsabilidades:** Asigna los datos que se pasan por parámetros al internet explorer instalado en el equipo “NombreEquipo”.

**Referencias Cruzadas:** Casos de uso: Cambiar configuración Internet Explorer

**Precondiciones:** El parámetro Proxy no puede ser un string vacío

**Postcondiciones:**

**Salida:**

#### 5.4.2 Caso de uso cambiar nombre perfil

##### Caso de uso 5: Cambiar nombre perfil

Diagrama de secuencia:



Figura 5.5.1 Cambiar nombre perfil

Contratos:

**Nombre:** IsProfile(NombreEquipo: String, Nomperfil: String)

**Responsabilidades:** Dice si existe un perfil en el equipo “NombreEquipo” con el nombre “Nomperfil”

**Referencias Cruzadas:** Casos de uso: Cambiar nombre perfil

**Precondiciones:**

**Postcondiciones:**

**Salida:** Devuelve el booleano perfil? A true si existe el perfil en el equipo.

**Nombre:** renameProfile(NombreEquipo: String, Nomperfil: String, NuevoNomPerfil: string)

**Responsabilidades:** Renombra al perfil “Nomperfil” del equipo “NombreEquipo” con un nuevo nombre “NuevoNomPerfil”

**Referencias Cruzadas:** Casos de uso: Cambiar nombre perfil

**Precondiciones:** El parámetro NombreEquipo debe pertenecer a un equipo de la red y el nombre Nomperfil debe pertenecer a uno de sus perfiles.

**Postcondiciones:** Dentro de las carpetas de perfiles del equipo, renombra la carpeta Nomperfil con el nombre, NuevoNomPerfil.

**Salida:**

### 5.4.3 Caso de uso gestionar carpetas

#### Caso de uso 6: Gestionar Carpetas

Diagrama de secuencia:

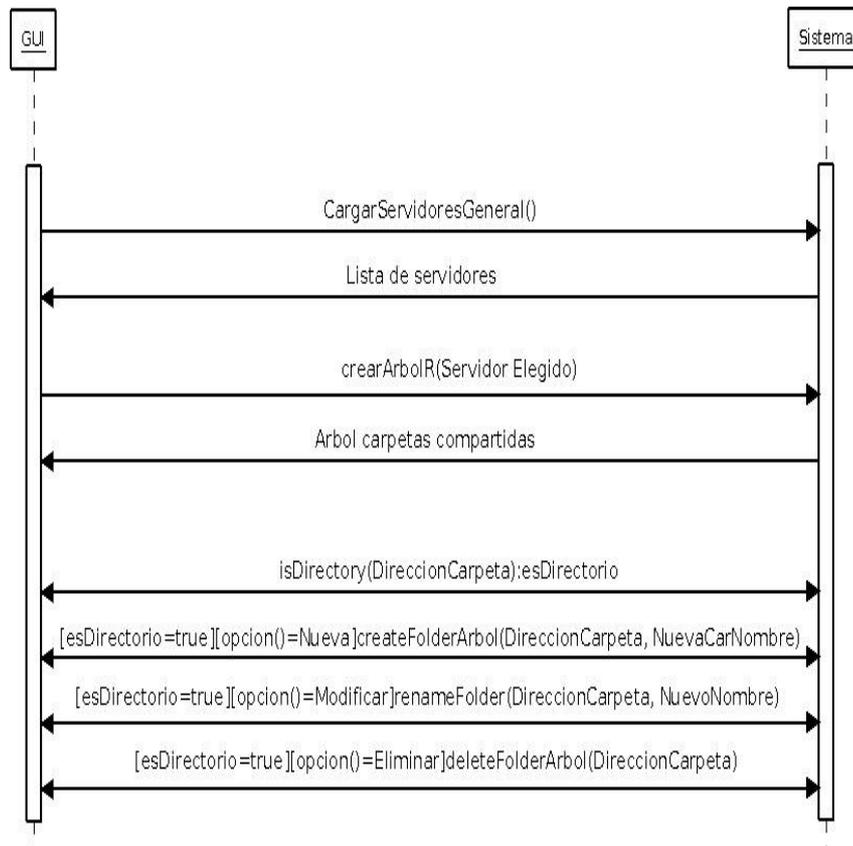


Figura 5.6.1 Gestionar Carpetas

**Nombre:** deleteFolder(DireccionCarpeta: String)

**Responsabilidades:** Elimina la carpeta cuya dirección se para por parámetros.

**Referencias Cruzadas:** Casos de uso: Gestionar Carpetas

**Precondiciones:** La carpeta que se quiere eliminar ha de ser un directorio

**Postcondiciones:** Elimina la carpeta seleccionada en el árbol de la interfaz

**Salida:**

**Nombre:** crearArbolR(Servidor: String)

**Responsabilidades:** Devolver vía interfaz el árbol que forman los recursos compartidos y las subcarpetas de éstos para el servidor seleccionado.

**Referencias Cruzadas:** Casos de uso: Gestionar carpetas

**Precondiciones:**

**Postcondiciones:**

**Salida:** Árbol con los nombres de las carpetas en la interfaz



## 6 DISEÑO

### 6.1 Diseño de la capa de presentación:

Para llevar a cabo el diseño externo de la aplicación he optado por realizar una versión reducida, pero con un estilo más amigable, del gestor del Directorio Activo (Active Directory Users and Computers) que se instala por defecto al instalar el directorio activo.

No obstante, como la gestión de carpetas, perfiles y la configuración de internet Explorer en remoto no entra dentro de las tareas que se pueden llevar a cabo con el gestor del Directorio activo. Las interfaces correspondientes a estas partes son totalmente originales.

A continuación, paso a mostrar las interfaces más importantes y los casos de uso para los que están concebidas. Se muestran las interfaces por ventanas de la aplicación. Estas, a su vez, se dividirán en diversos paneles. Por lo tanto, para que sea lo más fácil posible conocer qué acciones se realizan en cada interfaz y a qué caso de uso pertenecen, se pasará a explicarlas panel a panel.

### 6.2 Diseño de la capa de negocio:

A continuación se muestran los contratos obtenidos en el análisis utilizando pseudocódigo.

#### 6.2.1 Contratos de los casos de uso pertenecientes a la primera iteración

Contratos de los casos de uso Crear Usuario (Número 7), Crear Grupo (Número 9) y Crear Equipo (Número 10).

##### **UserExists(UserLogonName: string)**

De = Nueva Entrada al directorio activo

Results = Buscar objetos usuario en DE con el nombre que se pasa por parámetros (UserLogonName) como nombre de login de usuario.

**Si** Results > 0 **entonces**

    Devuelve true

**sino**

    Devuelve false

**fSi**

### **CreateAdAccount(ByVal sUserName As String, ByVal sPassword As String, ByVal sFirstName As String, ByVal OU\_AC As String)**

StroOU = Nombre de el contenedor actual de la dirección en la que se quiere crear la cuenta de usuario(OU\_AC).

DirE = Nueva entrada al directorio activo

DirEntry = Entrada al directorio activo que apunta al contenedor(StroOU). Buscada a partir de DirE

NewUser = Entrada al directorio activo que apunta al nuevo usuario de nombre sUserName creado mediante la función children de DirEntry.

//Asigna al nuevo usuario su nombre común y nombre de login

Se asigna sFristName al atributo "givenName" de NewUser

sUserName Se asigna sFristName al atributo "sAMAccountName" de NewUser

//Se le asigna el contraseña

NewUser invoca a la función "SetPassword" para asignarle sPassword como contraseña al usuario.

### **GroupExists(GroupName: string)**

De = Nueva Entrada al directorio activo

Results = Buscar objetos grupo en DE con el nombre que se pasa por parámetros (GroupName) como nombre de login de grupo.

**Si** Results > 0 **entonces**

Devuleve true

**sino**

Devuelve false

**fSi**

### **ComputerExists(ComputerName: string)**

De = Nueva Entrada al directorio activo

Results = Buscar objetos equipo en DE con el nombre que se pasa por parámetros (ComputerName) como nombre de equipo.

**Si** Results > 0 **entonces**

Devuleve true

**sino**

Devuelve false

**fSi**

## 6.2.2 Contratos de los casos de uso pertenecientes a la segunda iteración

Contratos de los casos de uso que extienden de Gestionar Propiedades Usuario (Número 17), Gestionar Propiedades Grupo (Número 18) y Gestionar Propiedades Equipo (Número 19).

### **GetUserProperty(UserLogName: String, Atributo: String)**

De = Nueva Entrada al directorio activo

Results = Buscar objetos de tipo usuario en DE con el nombre que se pasa por parámetros (ObjectName) como nombre de login.

#### **Si Results > 0 entonces**

```
//obtiene la entrada al directorio activo correspondiente al objeto
DEuser = GetDirectoryEntry(Results.direccion)
//Devuelve el valor de tipo string de la propiedad
Devolver Deuser.Propeties(Atributo)
```

#### **FSi**

Estas funciones sirven para objetos de tipo usuario y atributos con valores de tipo string. Pero habrá, otras funciones con el mismo pseudocódigo para los tipos:

Integer: GetUserPropertyInt, SetUserPropertyInt

Boolean: GetUserPropertyBool, SetUserPropertyBool

Objeto(para no especificar tipo) : GetUserPropertyObject,

SetUserPropertyObject

Dos tipos creados, que serán explicados en la parte de implementación

LargeInterval : GetUserPropertyLargeInterval, SetUserPropertyLargeInterval

Oidentifier: GetUserPropertyOidentifier, SetUserPropertyOidentifier

Además, de por tipo de dato, habrá otras funciones con exactamente el mismo pseudocódigo pero cambiando el tipo de objeto que se busca.

Grupo: GetgroupProperty, SetGroupProperty

Equipo: GetComputerProperty, SetComputerProperty

También los cambios por tipo de dato se pueden aplicar a éstas.

Por último habrá dos funciones por tipo Get(User,Group o Computer)PropertyRefresh y Set(User,Group o Computer)PropertyRefresh que realizarán lo mismo que GetUserproperty y SetUserProperty pero refrescando la memoria antes de acceder a la propiedad del objeto.

A continuación se muestran los cambios, en el pseudocódigo, de estas respecto de las anteriores.

### **ChangePasswordDenied(UserLogName: String)**

De = Nueva Entrada al directorio activo

DEuser = Buscar el objeto de tipo usuario en DE con el nombre que se pasa por parámetros (UserLogNameName) como nombre de login. Asignarle su Entrada del Directorio activo.

//Obtener la seguridad del usuario

userSecurity = Deuser.ObjectSecurity

//Recorrer sus reglas de acceso para saber si hay alguna que deniegue al usuario el cambio de password

**Para cada** regla de userSecurity.ReglasdeAcceso

**Si** regla.tipo = denegar y regla.tipoObjeto=sí mismo(self) **entonces**

        Devuelve true

**Fsi**

**Siguiente**

Devuelve false

### **AllowPasswordDenied(UserLogName: String)**

De = Nueva Entrada al directorio activo

DEuser = Buscar el objeto de tipo usuario en DE con el nombre que se pasa por parámetros (UserLogNameName) como nombre de login. Asignarle su Entrada del Directorio activo.

//Obtener la seguridad del usuario

userSecurity = Deuser.ObjectSecurity

//crear las reglas permitir y denegar cambio de contraseña para todo el mundo y para sí mismo sin asignarlos. Esta parte se explicará en la implementación

userSecurity.AñadirRegla(permitir cambio contraseña a todo el mundo)

userSecurity.EliminarRegla(denegar cambio contraseña a todo el mundo)

userSecurity.EliminarRegla(denegar cambio contraseña a sí mismo)

userSecurity.AñadirRegla(permitir cambio contraseña a sí mismo)

### **DenyChangePassword(UserLogName: String)**

De = Nueva Entrada al directorio activo

DEuser = Buscar el objeto de tipo usuario en DE con el nombre que se pasa por parámetros (UserLogNameName) como nombre de login. Asignarle su Entrada del Directorio activo.

//Obtener la seguridad del usuario

userSecurity = Deuser.ObjectSecurity

//crear las reglas permitir y denegar cambio de contraseña para todo el mundo y para sí mismo sin asignarlos. Esta parte se explicará en la implementación

userSecurity.AñadirRegla(denegar cambio contraseña a todo el mundo)

userSecurity.EliminarRegla(permitir cambio contraseña a todo el mundo)

userSecurity.EliminarRegla(permitir cambio contraseña a sí mismo)

userSecurity.AñadirRegla(denegar cambio contraseña a sí mismo)

### **PasswordExpires(UserLogName: String)**

De = Nueva Entrada al directorio activo

DEuser = Buscar el objeto de tipo usuario en DE con el nombre que se pasa por parámetros (UserLogNameName) como nombre de login. Asignarle su Entrada del Directorio activo.

//Cambiar el atributo del objeto de tipo usuario que establece si el password expira

val = Deuser.Properties("userAccountControl")

Deuser.Properties("userAccountControl") = val y **no** &H1

### **DontExpirePassword(UserLogName: String)**

De = Nueva Entrada al directorio activo

DEuser = Buscar el objeto de tipo usuario en DE con el nombre que se pasa por parámetros (UserLogNameName) como nombre de login. Asignarle su Entrada del Directorio activo.

//Cambiar el atributo del objeto de tipo usuario que establece si el password expira

val = Deuser.Properties("userAccountControl")

Deuser.Properties("userAccountControl") = val **o** & H1

### **GetExpirationTime(UserLogName: String)**

De = Nueva Entrada al directorio activo

DEuser = Buscar el objeto de tipo usuario en DE con el nombre que se pasa por parámetros (UserLogNameName) como nombre de login. Asignarle su Entrada del Directorio activo.

//Obtener el atributo del objeto de tipo usuario que establece la fecha en la que expira la contraseña

val = DEuser.ObjetoNativo.FechaExpiraContraseña()

Devolver val

### **RemoveExpirationTime(UserLogName: String)**

De = Nueva Entrada al directorio activo

DEuser = Buscar el objeto de tipo usuario en DE con el nombre que se pasa por parámetros (UserLogNameName) como nombre de login. Asignarle su Entrada del Directorio activo.

//Obtener el atributo del objeto de tipo usuario que establece si expira la cuenta de usuario

DEuser.Properties("accountExpires") = 0

### **GetDaclNamesNT(UserLogName: String)**

De = Nueva Entrada al directorio activo

DEuser = Buscar el objeto en DA con el nombre que se pasa por parámetros (UserLogNameName) como nombre de login. Asignarle su Entrada del Directorio activo.

//Obtener el atributo descriptor de seguridad del objeto

userSecurity = DEuser.Properties("ntSecurityDescriptor")

//obtener la lista de permisos que concede el objeto

dacl = userSecurity.Discretionaryacl

salida = nueva lista de strings

//Recorrer permisos uno a uno para añadir los nombres de los objetos a los que se les ha concedido.

**Para cada** permiso en lista

**Para cada** miembro de salida

**Si** miembro = permiso.nombreObjeto **entonces**

            puede = false

**Fsi**

**Siguiente**

**Si** puede = true **entonces**

        salida.añadir(permiso.nombreObjeto)

**Fsi**

**Siguiente**

**Devolver** salida

### **GetAceFromDaclNT(ObjConPermiso: String, UserLogName: String, NomPermiso: String)**

De = Nueva Entrada al directorio activo

DEuser = Buscar el objeto de tipo usuario, grupo o equipo en DA con el nombre que se pasa por parámetros (UserLogNameName) como nombre de login. Asignarle su Entrada del Directorio activo.

```
//Obtener el atributo descriptor de seguridad del objeto  
userSecurity = Deuser.Properties("ntSecurityDescriptor")
```

```
//obtener la lista de permisos que concede el objeto  
dacl = userSecurity.Discretionaryacl
```

```
//Obtener los atributos objectType y la máscara que corresponden al permiso que buscamos
```

```
ObjectType
```

```
mascara
```

```
//Recorrer permisos y si alguno coincide con el que buscamos devolvemos un array de 4 booleanos (permitido, denegado, permitidoHeredado, denegadoHeredado).
```

```
salida(3) = nuevo array de booleanos
```

**Para cada** permiso en lista

```
    Si permiso.nombreObjeto = ObjetoconPermiso y ObjectType =  
        permiso.ObjectType y mascara = permiso.mascara entonces  
        salida(0)=permiso.permitido  
        salida(1)=permiso.denegado  
        salida(2)=permiso.permitidoHeredado  
        salida(3)=permiso.denegadoHeredado
```

**Fsi**

**Siguiente**

Devolver salida

Este pseudocódigo es también válido para GetDaclNames(cambiando Deuser.Properties("ntSecurityDescriptor") por Deuser.Mailboxrights), que obtendrá los nombres de los objetos que poseen permisos sobre un buzón Exchange.

**AddAceToDaclNT(ObjConPermiso: String, UserLogName: String,  
NomPermiso: String, tipo = string)**

De = Nueva Entrada al directorio activo

DEuser = Buscar el objeto de tipo grupo en DA con el nombre que se pasa por parámetros (UserLogNameName) como nombre de login. Asignarle su Entrada del Directorio activo.

```
//Obtener el atributo descriptor de seguridad del objeto  
userSecurity = Deuser.Properties("ntSecurityDescriptor")
```

```
//obtener la lista de permisos que concede el objeto  
dacl = userSecurity.Discretionaryacl
```

```
//Obtener los atributos objectType y la máscara que corresponden al permiso que  
buscamos
```

```
ObjectType
```

```
mascara
```

```
//Crear el nuevo permiso
```

```
Nuevopermiso.nombreObjeto = ObjConPermiso
```

```
Nuevopermiso.objectType = ObjectType
```

```
NuevoPermiso.mascara = mascara
```

```
Si tipo = "Allow" entonces
```

```
    Nuevopermiso.permitido
```

```
Sino
```

```
    NuevoPermiso.denegado
```

```
FSi
```

```
//Añadir a la lista de permisos
```

```
dacl.añadir(NuevoPermiso)
```

```
Deuser.Properties("ntSecurityDescriptor").Discretionaryacl = dacl
```

Este pseudocódigo es también válido para:

AddAceToDacl(cambiando Deuser.Properties("ntSecurityDescriptor") por Deuser.Mailboxrights), que obtendrá los nombres de los objetos que poseen permisos sobre un buzón Exchange.

AddAceToDaclUserNT, que obtendrá los nombres de los objetos que poseen permisos sobre un objeto de tipo usuario.

AddAceToDaclUserNT, que obtendrá los nombres de los objetos que poseen permisos sobre un objeto de tipo equipo.

**RemoveAceFromDaclNT(ObjConPermiso: String, UserLogName: String, NomPermiso: String)**

De = Nueva Entrada al directorio activo

DEuser = Buscar el objeto de tipo grupo en DA con el nombre que se pasa por parámetros (UserLogNameName) como nombre de login. Asignarle su Entrada del Directorio activo.

//Obtener el atributo descriptor de seguridad del objeto

userSecurity = Deuser.Properties("ntSecurityDescriptor")

//obtener la lista de permisos que concede el objeto

dacl = userSecurity.Discretionaryacl

//Obtener los atributos objectType y la máscara que corresponden al permiso que buscamos

ObjectType

maskara

//Recorrer permisos y si alguno coincide con el que buscamos se elimina.

**Para cada** permiso en dacl

**Si** permiso.nombreObjeto = ObjetoconPermiso y ObjectType = permiso.ObjectType y maskara = permiso.maskara **entonces**

**Si** tipo = "Allow" y permiso.permittedo **entonces**

dacl.eliminarPermiso(permiso)

**SinoSi** tipo="Deny" y permiso.denegado **entonces**

dacl.eliminarPermiso(permiso)

**FSi**

**Fsi**

**Siguiente**

//Actualizar lista de permisos

Deuser.Properties("ntSecurityDescriptor").Discretionaryacl = dacl

Este pseudocódigo es también válido para:

RemoveAceFromDacl(cambiando Deuser.Properties("ntSecurityDescriptor") por

Deuser.Mailboxrights), que obtendrá los nombres de los objetos que poseen permisos sobre un buzón Exchange.

RemoveAceFromDaclUserNT, que obtendrá los nombres de los objetos que poseen permisos sobre un objeto de tipo usuario.

RemoveAceFromDaclUserNT, que obtendrá los nombres de los objetos que poseen permisos sobre un objeto de tipo equipo.

### **RemoveAces(ObjConPermiso: String, UserLogName: String)**

De = Nueva Entrada al directorio activo

DEuser = Buscar el objeto en DA con el nombre que se pasa por parámetros (UserLogNameName) como nombre de login. Asignarle su Entrada del Directorio activo.

//Obtener el atributo descriptor de seguridad del objeto

userSecurity = Deuser.Properties("ntSecurityDescriptor")

//obtener la lista de permisos que concede el objeto

dacl = userSecurity.Discretionaryacl

//Recorrer permisos y si alguno coincide con el que buscamos se elimina.

**Para cada** permiso en dacl

**Si** permiso.nombreObjeto = ObjetoconPermiso **entonces**

dacl.eliminarpermiso(permiso)

**Fsi**

### **Siguiente**

//Actualizar lista de permisos

Deuser.Properties("ntSecurityDescriptor").Discretionaryacl = dacl

### **SetReceptioMessageSize(UserLogName: String, TamMensjEnt: Integer)**

De = Nueva Entrada al directorio activo

DEuser = Buscar el objeto de tipo usuario en DE con el nombre que se pasa por parámetros (UserLogNameName) como nombre de login. Asignarle su Entrada del Directorio activo.

//Modificar el atributo del objeto de tipo usuario que establece el tamaño máximo del mensaje de entrada

Deuser.Properties("delivContLength") = TamMensjEnt

### **SetSendingMessageSize(UserLogName: String, TamMensjEnt: Integer)**

De = Nueva Entrada al directorio activo

DEuser = Buscar el objeto de tipo usuario en DE con el nombre que se pasa por parámetros (UserLogNameName) como nombre de login. Asignarle su Entrada del Directorio activo.

//Modificar el atributo del objeto de tipo usuario que establece el tamaño máximo del mensaje de entrada

Deuser.Properties("submissionContLength") = TamMensjEnt

### **RemoveOrigAuth(UserToRemove: String, UserLogName: String)**

De = Nueva Entrada al directorio activo

DEuser = Buscar el objeto en DA con el nombre que se pasa por parámetros (UserLogNameName) como nombre de login. Asignarle su Entrada del Directorio activo.

//Obtener el atributo descriptor que posee la lista de usuarios de los que se pueden recibir correos.

lista = Deuser.Properties("authOrig")

//Recorrer la lista y generar una auxiliar con todos los usuarios menos el que queremos eliminar

**Para cada** miembro en lista

**Si no** miembro = UserToRemove **entonces**

listaAux.añadir(miembro)

**Fsi**

**Siguiente**

//Actualizar lista de permisos

Deuser.Properties("authOrig") = listaAux

### **RemoveOrigUnauth(UserToRemove: String, UserLogName: String)**

De = Nueva Entrada al directorio activo

DEuser = Buscar el objeto en DA con el nombre que se pasa por parámetros (UserLogNameName) como nombre de login. Asignarle su Entrada del Directorio activo.

//Obtener el atributo descriptor que posee la lista de usuarios de los que no se pueden recibir correos.

lista = Deuser.Properties("unauthOrig")

//Recorrer la lista y generar una auxiliar con todos los usuarios menos el que queremos eliminar

**Para cada** miembro en lista

**Si no** miembro = UserToRemove **entonces**

listaAux.añadir(miembro)

**Fsi**

**Siguiente**

//Actualizar lista de permisos

Deuser.Properties("unauthOrig") = listaAux

### **AddOrigAuth(UserToAdd: String, UserLogName: String)**

De = Nueva Entrada al directorio activo

DEuser = Buscar el objeto en DA con el nombre que se pasa por parámetros (UserLogNameName) como nombre de login. Asignarle su Entrada del Directorio activo.

//Obtener el atributo descriptor que posee la lista de usuarios de los que se pueden recibir correos.

lista = Deuser.Properties("authOrig")

//Añadir a la lista el usuario que se quiere añadir

lista.añadir(UserToAdd)

//Actualizar lista de permisos

Deuser.Properties("authOrig") = lista

### **IsInOrigAuth(UserToCheck: String, UserLogName: String)**

De = Nueva Entrada al directorio activo

DEuser = Buscar el objeto en DA con el nombre que se pasa por parámetros (UserLogNameName) como nombre de login. Asignarle su Entrada del Directorio activo.

//Obtener el atributo descriptor que posee la lista de usuarios de los que se pueden recibir correos.

lista = Deuser.Properties("authOrig")

//Recorrer la lista y si encuentra el usuario que se busca devuelve true

**Para cada** miembro en lista

**Si** miembro = UserToRemove **entonces**

Devuelve true

**Fsi**

**Siguiente**

Devuelve false

### **IsInOrigUnauth(UserToCheck: String, UserLogName: String)**

De = Nueva Entrada al directorio activo

DEuser = Buscar el objeto en DA con el nombre que se pasa por parámetros (UserLogNameName) como nombre de login. Asignarle su Entrada del Directorio activo.

//Obtener el atributo descriptor que posee la lista de usuarios de los que no se pueden recibir correos.

lista = Deuser.Properties("unauthOrig")

//Recorrer la lista y si encuentra el usuario que se busca devuelve true

**Para cada** miembro en lista

**Si** miembro = UserToRemove **entonces**

Devuelve true

**Fsi**

**Siguiente**

Devuelve false

### **AddOrigUnauth(UserToAdd: String, UserLogName: String)**

De = Nueva Entrada al directorio activo

DEuser = Buscar el objeto en DA con el nombre que se pasa por parámetros (UserLogNameName) como nombre de login. Asignarle su Entrada del Directorio activo.

//Obtener el atributo que posee la lista de usuarios de los que no se pueden recibir correos.

lista = Deuser.Properties("unauthOrig")

//Añadir a la lista el usuario que se quiere añadir

lista.añadir(UserToAdd)

//Actualizar lista de permisos

Deuser.Properties("unauthOrig") = lista

### **ChangeOrigAuth(UserLogName: String)**

De = Nueva Entrada al directorio activo

DEuser = Buscar el objeto en DA con el nombre que se pasa por parámetros (UserLogNameName) como nombre de login. Asignarle su Entrada del Directorio activo.

//Obtener el atributo descriptor que posee la lista de usuarios de los no que se pueden recibir correos y de los que se pueden. Se guardan en listas auxiliares

listaAuth = Deuser.Properties("authOrig")

listaUnauth = Deuser.Properties("unauthOrig")

//Asignar los auxiliares cada uno al atributo contrario del que se ha obtenido

Deuser.Properties("authOrig") = listaUnauth

Deuser.Properties("unauthOrig") = listaAuth

### **ClearProperty(UserLogName: String, Atributo: String)**

De = Nueva Entrada al directorio activo

Results = Buscar objetos de tipo usuario en DE con el nombre que se pasa por parámetros (ObjectName) como nombre de login.

#### **Si Results > 0 entonces**

//obtiene la entrada al directorio activo correspondiente al objeto

DEuser = GetDirectoryEntry(Results.direccion)

//Elimina el valor de tipo string de la propiedad

DEuser.Properties(Atributo).clear

**FSi**

### **setPropertyInsert(UserLogName: String, Atributo: String, Dato: String)**

De = Nueva Entrada al directorio activo

Results = Buscar objetos de tipo usuario en DE con el nombre que se pasa por parámetros (ObjectName) como nombre de login.

#### **Si Results > 0 entonces**

//obtiene la entrada al directorio activo correspondiente al objeto

DEuser = GetDirectoryEntry(Results.direccion)

//Añade el dato a la propiedad

DEuser.Properties(Atributo).añadir(Dato)

**FSi**

### **SetForwardingAddress(UserLogName: String, NuevoRedirectTo: String)**

De = Nueva Entrada al directorio activo

DEuser = Buscar el objeto de tipo usuario en DE con el nombre que se pasa por parámetros (UserLogNameName) como nombre de login. Asignarle su Entrada del Directorio activo.

//Modificar el atributo que establece el usuario al que se redireccionarán los mensajes que se reciban

DEuser.Properties("altRecipient") = NuevoRedirectTo

### **SetRecipientLimits(UserLogName: String, NDestinatarios: Integer)**

De = Nueva Entrada al directorio activo

DEuser = Buscar el objeto de tipo usuario en DE con el nombre que se pasa por parámetros (UserLogNameName) como nombre de login. Asignarle su Entrada del Directorio activo.

//Modificar el atributo que establece el número máximo de destinatarios a los que se puede enviar un mensaje.

Deuser.Properties("msExchRecipientLimit") = NDestinatarios

### **x400Downgraded(UserLogName: String)**

De = Nueva Entrada al directorio activo

DEuser = Buscar el objeto de tipo usuario en DE con el nombre que se pasa por parámetros (UserLogNameName) como nombre de login. Asignarle su Entrada del Directorio activo.

//obtener el valor del atributo establece si los mensajes del tipo x400 están degradados en cuanto a su prioridad de llegada

val = Deuser.Properties("securityProtocol")

//El valor será un sistema de cuatro octetos. Para que se produzca la degradación, todos deben tener valor 0 menos el tercero que será 128.

**Si val(0)= y val(1)=0 y val(2)=128 y val(3)=0 entonces**

Devuelve true

**Sino**

Devuelve false

**FSi**

### **GetUserGroups(UserLogName: String)**

De = Nueva Entrada al directorio activo

DEuser = Buscar el objeto de tipo usuario en DE con el nombre que se pasa por parámetros (UserLogNameName) como nombre de login. Asignarle su Entrada del Directorio activo.

//Obtener su dirección

Dir = Deuser.Properties("distinguishedName")

Groups = Buscar todos los objetos de tipo grupo en DE y que posea Dir como miembro.

//Devolver los nombres de login de esos grupos. Que se introducirán en una lista de strings.

**Para cada** grupo en Groups

    salida.añadir(Grupo.Properties("sAMAccountNames"))

**Siguiente**

Devolver salida

Este pseudocódigo es también válido para:

GetGroupGroups, que hará lo mismo que para el usuario pero para un objeto de tipo grupo.

GetComputerGroups, que hará lo mismo que para el usuario pero para un objeto de tipo equipo.

**SetManagerCanChangeMembershipList(Manager: String, GroupLogName: String, Permiso: boolean)**

De = Nueva Entrada al directorio activo

DEuser = Buscar el objeto de tipo grupo en DA con el nombre que se pasa por parámetros (UserLogNameName) como nombre de login. Asignarle su Entrada del Directorio activo.

//Obtener el atributo descriptor de seguridad del objeto

userSecurity = Deuser.Properties("ntSecurityDescriptor")

//obtener la lista de permisos que concede el objeto

dacl = userSecurity.Discretionaryacl

//Obtener los atributos objectType y la máscara que corresponden al permiso que de cambio de miembros de grupo

ObjectType

mascara

//Crear el nuevo permiso

Nuevopermiso.nombreObjeto = Manager

Nuevopermiso.objectType = ObjectType

NuevoPermiso.mascara = mascara

Nuevopermiso.permitido

**Si** Permiso = boolean **entonces**

    dacl.añadir(NuevoPermiso)

**Sino**

**Para cada** permiso en dacl

**Si** permiso = NuevoPermiso **entonces**

            dacl.eliminar(permiso)

**Fsi**

**Siguiente**

**FSi**

//Actualizar la lista de permisos

Deuser.Properties("ntSecurityDescriptor").Discretionaryacl = dacl

## **CargarServidores(AdminGroup: String)**

RootDSE = Nueva Entrada a la raíz directorio activo

//Obtener la propiedad de la raíz del directorio activo que establece la dirección de la parte de configuración del directorio activo.

Rootpath = RootDSE.GetDirectoryEntry.Properties("configurationNamingContext")

Organization = Buscar la parte de organización de Exchange dentro de la parte de configuración del directorio activo(en la dirección Rootpath).

//Si Exchange está configurado en modo mixto, solo se buscan los servidores del grupo administrativo al que pertenece el buzón.

**Si** Organization.GetDirectoryEntry.Porpiedades("msExchMixedMode") **entonces**

AdministrationGroups = Buscar los grupos administrativos dentro de la parte de configuración del directorio activo(en la dirección Rootpath).

**Para cada** GrupoAdministrativo de AdministrationGroups

**Si** GrupoAdministrativo = AdminGroup **entonces**

BackEndServers = Buscar objetos del tipo Servidor Exchange en la entrada del directorio activo de dirección Rootpath.

**Para cada** Servidor en BackEndServers

//Añadir su nombre en una lista de strings

Lista.add(Servidor. GetDirectoryEntry("cn"))

**Siguiente**

**FSi**

**Siguiente**

**Sino**

Lista = Servidores()

**FSi**

Devolver Lista

### 6.2.3 Contratos de los casos de uso pertenecientes a la tercera iteración

Contratos de los casos de uso Realizar Tareas Exchange (Número 64), Buscar Objeto (Número 2), Mover Objeto (Número 12), Eliminar objeto (Número 13), Resetear Contraseña usuario (Número 15), Desbloquear usuario (Número 16), Cambiar nombre objeto (Número 17) y Habilitar/Deshabilitar usuario o Equipo (Número 18).

**MoveMailbox(UserLogonName: String, AlmacénElegido: String, ServidorAlmacen: String)**

RootDSE = Nueva Entrada a la raíz directorio activo

//Obtener la propiedad de la raíz del directorio activo que establece la dirección de la parte de configuración del directorio activo.

Rootpath =

RootDSE.GetDirectoryEntry.Properties("configurationNamingContext")

Server = Buscar el servidor Exchange de nombre ServidorAlmacen dentro de la parte de configuración del directorio activo(en la dirección Rootpath).

Direccion = Server.Properties("distinguishedName")

//Con la dirección del servidor obtener sus almacenes de buzones para obtener el Almacén elegido para mover el buzón allí.

AlmacenesDeBuzones = GetMailboxStores(Direccion)

Para cada Almacen en AlmacenesDeBuzones

    Si Almacen = AlmacenElegido entonces

        homeMDB =

        Almacen.GetDirectoryEntry.Properties("distinguishedName")

    Fsi

Siguiente

DEuser = Buscar al usuario dueño del buzón (UserLogon Name) en el DA. Y asignar a DEuser su entrada al Directorio activo.

//llamada a la función que mueve el buzón

DEuser.moveMailbox(homeMDB)

### **Group\_MailDisable(NombreGrupo: String)**

RootDSE = Nueva Entrada a la raíz directorio activo

DEgroup = Buscar al grupo (NombreGrupoe) en el DA. Y asignar a Deuser su entrada al Directorio activo.

//Eliminar los valores de los atributos que establecen el correo del grupo

Degroup.Properties("legacyExchangeDN").Eliminar

Degroup.Properties("MailNickName").Eliminar

### **DeleteMailbox(UserLogonName: String, AlmacénElegido: String, ServidorAlmacen: String)**

RootDSE = Nueva Entrada a la raíz directorio activo

DEuser = Buscar al usuario dueño del buzón (UserLogon Name) en el DA. Y asignar a Deuser su entrada al Directorio activo.

//llamada a la función que elimina su buzón

DEuser.DeleteMailbox()

### **Rename(ObjectName: String, NuevoNombre: String)**

De = Nueva Entrada al directorio activo

Results = Buscar objetos en DE con el nombre que se pasa por parámetros (ObjectName) como nombre de login.

//Se guarda en dos variables la dirección del objeto y la de su contenedor

dirObjeto = Results.GetDirectoryEntry.Properties("distinguishedName")

dirContainer = La parte hasta llegar al nombre del objeto de dirObjeto

//Se obtiene el objeto correspondiente al container y éste llama a la función que cambia de nombre a uno de los objetos que se encuentran dentro de él

oContainer = getObjeto(dirContainer)

oContainer.MoveHere(dirObjeto, NuevoNombre)

### **User\_MailEnable(NombreUserLogon: String, Servidor: String, MailboxStore: String)**

RootDSE = Nueva Entrada a la raíz directorio activo

//Obtener la propiedad de la raíz del directorio activo que establece la dirección de la parte de configuración del directorio activo.

```
Rootpath =  
RootDSE.GetDirectoryEntry.Properties("configurationNamingContext")  
BackEndServers = Buscar objetos del tipo Servidor Exchange en la entrada del directorio activo de dirección Rootpath.
```

**Para cada** Servidor en BackEndServers

**Si** Servidor.Properties("cn") = Servidor **entonces**

        MailboxStores = getMailboxStores(Servidor)

**Para cada** MailboxStoreR en oresults

**Si** MailboxStoreR.Properties("cn") = MailboxStore **entonces**  
                dirMailbox =

MailboxR.Properties("distinguishedName")

**Fsi**

**Siguiente**

**FSi**

**Siguiente**

//Busca el usuario

De = Nueva Entrada al directorio activo

Usuario = Buscar objetos usuario en DE con el nombre que se pasa por parámetros (NombreUserLogon) como nombre de login de usuario.

//Llamar a la función que crea el buzón

```
CreateMailbox(Usuario.Properties("distinguishedName"), dirMailbox)
```

### **RemoveFromOU(ObjectName: String)**

De = Nueva Entrada al directorio activo

Results = Buscar objetos en DE con el nombre que se pasa por parámetros (ObjectName) como nombre de login de usuario.

**Si** Results > 0 **entonces**

    //Llama a la función que elimina el objeto

```
    De.children.Remove(Results)
```

**fSi**

### **Disable(ObjectName: String)**

De = Nueva Entrada al directorio activo

Results = Buscar objetos en DE con el nombre que se pasa por parámetros (ObjectName) como nombre de login.

**Si Results > 0 entonces**

//obtiene la entrada al directorio activo correspondiente al objeto

DEobjeto = GetDirectoryEntry(Results.direccion)

//Obtiene la propiedad que establece el estado de la seguridad del objeto

val = Deobjeto.Properties("userAccountControl")

//Pasar esa propiedad a Deshabilitado

Deobjeto.Properties("userAccountControl") = val o &H2

**FSi**

### **Enable(ObjectName: String)**

De = Nueva Entrada al directorio activo

Results = Buscar objetos en DE con el nombre que se pasa por parámetros (ObjectName) como nombre de login.

**Si Results > 0 entonces**

//obtiene la entrada al directorio activo correspondiente al objeto

DEobjeto = GetDirectoryEntry(Results.direccion)

//Obtiene la propiedad que establece el estado de la seguridad del objeto

val = Deobjeto.Properties("userAccountControl")

//Habilitarlo para el caso de que la contraseña del objeto no se deshabilite nunca

Deobjeto.Properties("userAccountControl") = val o &H1

//Habilitarlo para el caso "normal"

Deobjeto.Properties("userAccountControl") = val y no &H2

**FSi**

### **ChangePassword(UserLogName: String, NuevoPassword: String)**

De = Nueva Entrada al directorio activo

Results = Buscar objetos de tipo usuario en DE con el nombre que se pasa por parámetros (ObjectName) como nombre de login.

**Si Results > 0 entonces**

*//obtiene la entrada al directorio activo correspondiente al objeto*

DEuser = GetDirectoryEntry(Results.direccion)

*//Invoca al método que cambia el password*

Deuser.invoke("SetPassword", NuevoPassword)

**FSi**

### **UnlockUser(UserLogName: String)**

De = Nueva Entrada al directorio activo

Results = Buscar objetos de tipo usuario en DE con el nombre que se pasa por parámetros (ObjectName) como nombre de login.

**Si Results > 0 entonces**

*//obtiene la entrada al directorio activo correspondiente al objeto*

DEuser = GetDirectoryEntry(Results.direccion)

*//Invoca al método que desbloquea el usuario*

Deuser.invoke("IsAccountLocked", False)

**FSi**

#### 6.2.4 Contratos de los casos de uso pertenecientes a la cuarta iteración

Contratos de los casos de uso Cambiar Configuración de Internet Explorer (Número 4), cambiar Nombre perfil (Número 5) y Gestionar carpetas (Número 6).

##### **IexplorerDatos(NombreEquipo: String)**

```
//Conectar al equipo remoto
oLocator = CreateObject("WbemScripting.SWbemLocator")
oService = oLocator.connectServer(NombreEquipo, Raíz, NombreAdministrador,
PasswordAdministrador)

//Se obtienen los datos de red de Internet Explorer
objRegistry= oService.Get("StdRegProv")
strKeyPath = "SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings"
//Estado del Proxy (Habilitado/Deshabilitado)
Habilitado = objRegistry.ObtenerValor(strKeyPath, "ProxyEnable")

//Proxy
Proxy = objRegistry.ObtenerValor(strKeyPath, "ProxyServer")

//Excepciones
Excep = objRegistry.ObtenerValor(strKeyPath, "ProxyOverride")
```

##### **IexplorerCarga(NombreEquipo: String, Habilitado: boolean, Proxy: String, Excepciones: String)**

```
//Conectar al equipo remoto
oLocator = CreateObject("WbemScripting.SWbemLocator")
oService = oLocator.connectServer(NombreEquipo, Raíz, NombreAdministrador,
PasswordAdministrador)

//Se asignan los datos de red de Internet Explorer
objRegistry= oService.Get("StdRegProv")
strKeyPath = "SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings"

//Estado del Proxy (Habilitado/Deshabilitado)
Habilitado = objRegistry.AsignarValor(strKeyPath, "ProxyEnable", Habilitado)

//Proxy
Proxy = objRegistry.AsignarValor (strKeyPath, "ProxyServer", Proxy)
//Excepciones
Excep = objRegistry.AsignarValor (strKeyPath, "ProxyOverride", Excepciones)
```

### **IsProfile(NombreEquipo: String, NombrePerfil: String)**

//Llamar a la función que produce la impersonación para conseguir que se obtengan permisos de administrador  
imp.EmpezarImpersonación(NombreAdministrador,  
ContraseñaAdministrador,NombreDominio)

//Crear la dirección al perfil  
direccion = "\\\" & cNombreEquipo & "\\\" & "C\$\Documents and Settings\" &  
NombrePerfil

//Llamada a la función de la clase Directory que dice si existe una carpeta en la dirección creada

**Si** Directory.EsCarpeta(direccion) **entonces**

    Devuelve true

**Sino**

    Devuelve false

**FSi**

### **RenameProfile(NombreEquipo: String, NombrePerfil: String, NuevoNombrePerfil: String)**

//Llamar a la función que produce la impersonación para conseguir que se obtengan permisos de administrador  
imp.EmpezarImpersonación(NombreAdministrador,  
ContraseñaAdministrador,NombreDominio)

//Crear la dirección al perfil con el nombre antiguo y otra con el nombre nuevo  
direccionAntigua = "\\\" & cNombreEquipo & "\\\" & "C\$\Documents and Settings\" &  
& NombrePerfil

direccionNueva = "\\\" & cNombreEquipo & "\\\" & "C\$\Documents and Settings\" &  
NuevoNombrePerfil

//Llamada a la función de la clase Directory que Mueve el contenido de la carpeta que se encuentra en la dirección antigua a la carpeta creada en la nueva, para posteriormente borrar la carpeta antigua que ha quedado vacía.

Directory.Mover(direccionAntigua, direccionNueva)

### **CargaServidoresGeneral()**

RootDSE = Nueva Entrada a la raíz directorio activo

//Obtener la propiedad de la raíz del directorio activo que establece la dirección de la parte de configuración del directorio activo.

Rootpath =

RootDSE.GetDirectoryEntry.Properties("configurationNamingContext")

BackEndServers = Buscar objetos del tipo Servidor Exchange y Servidor normal en la entrada del directorio activo de dirección Rootpath.

**Para cada** Servidor en BackEndServers

    //Añadir su nombre en una lista de strings

    Lista.add(Servidor. GetDirectoryEntry("cn"))

Siguiente

Devolver Lista

### **CrearArbolR(Servidor: String)**

//Se obtienen los recursos compartidos del equipo llamando a la función getSharedElements

elementos = getSharedElements(Servidor)

Crea el árbol

//Para cada elemento, si es una carpeta, se introduce en el árbol como nodo y se llama a la función recursiva que forma las ramas del árbol

**Para cada** elemento de la lista elementos

    dirección = "\\\" & Servidor & "\" & elemento

**Si** isDirectory(direccion) **entonces**

        NuevoNodo = elemento

        Arbol.AñadirNodo(NuevoNodo)

        CrearArbolF(direccion, NuevoNodo)

**fSi**

Siguiente

### **IsDirectory(Direccion: string)**

//Llama a la función de la clase directory que te dice si la dirección apunta a una carpeta existente o no

**Si** Directory.Existe(Direccion) **entonces**

    Devuelve true

**sino**

    Devuelve false

**fSi**

### **CrearArbolR(Direccion: String, NodoPadre: NodoArbol)**

//Se obtienen las subcarpetas del elemento

elementos = SFolderElements(Direccion)

//Para cada elemento, si es una carpeta, se introduce en el árbol como nodo y se produce la recursividad para seguir creando el árbol

**Para cada** elemento de la lista elementos

    Nuevadirección = "\\\" & Direccion & "\" & elemento

**Si** isDirectory(Nuevadireccion) **entonces**

        NuevoNodo = elemento

        NodoPadre.AñadirNodo(NuevoNodo)

        CrearArbolF(direccion, NuevoNodo)

**Fsi**

### **Siguiente**

### **CreateFolderArbol(Direccióncarpeta: String, NuevaCarpeta: String)**

//Llamar a la función que produce la impersonación para conseguir que se obtengan permisos de administrador

imp.EmpezarImpersonación(NombreAdministrador,  
ContraseñaAdministrador,NombreDominio)

//Llamada a la función de la clase Directory que dice si existe una carpeta en la dirección creada

**Si** Directory.EsCarpeta(Direccioncarpeta) **entonces**

    //Se crea la carpeta

    Directory.CrearCarpeta(Direccioncarpeta & "\" & NuevaCarpeta)

**FSi**

### **DeleteFolderArbol(Direccióncarpeta: String)**

//Llamar a la función que produce la impersonación para conseguir que se obtengan permisos de administrador

imp.EmpezarImpersonación(NombreAdministrador,  
ContraseñaAdministrador,NombreDominio)

//Llamada a la función de la clase Directory que dice si existe una carpeta en la dirección creada

**Si** Directory.EsCarpeta(Direccioncarpeta) **entonces**

    //Eliminar la carpeta

    Directory.EliminarCarpeta(Direccioncarpeta)

**FSi**

## **6.3 Diseño de la capa de datos:**

Se explicarán, a continuación, los datos que manipula la aplicación. Datos que, en su mayoría, pertenecerán al Directorio Activo.

### **6.3.1 Gestión del Directorio Activo:**

Como se ha explicado en el primer capítulo, el directorio activo está formado por un conjunto de objetos. Éstos, a su vez, poseen un determinado número de atributos mediante los cuales se podrá manipularlos.

#### **6.3.1.1 Usuarios, Grupos, Equipos:**

Al observar las operaciones podemos comprobar como para los objetos de tipo usuario, grupo y equipo los atributos pueden ser de tipo integer(GetUserPropertyInt, GetGroupPropertyInt, GetComputerÇPropertyInt), string(GetUserProperty, GetGroupProperty, GetComputerÇProperty), boolean (GetUserPropertyBool, GetGroupPropertyBool, GetComputerÇPropertyBool), int64 (integer de 64 bits, GEtUserPropertyLargeInterval) e incluso el usuario posee un atributo del tipo octet String (un array de cuatro hexadecimales, X400downgrad). Además, se pueden obtener atributos sin especificar su tipo(GetUserPropertyObject, GetGroupPropertyObject, GetComputerPropertyObject) que generalmente se usará para cuando el atributo está formado por un array de strings.

Por último, hay un atributo un poco más complejo, el que se refiere a la seguridad del objeto.

Éste poseerá una lista de Permisos que se conceden sobre el objeto llamada DiscretionaryAcl. Cada permiso de esta lista será un ACE (Access control Entry) cada ACE tendrá sus propios atributos, pero los que se utilizan en la aplicación son:

- trustee, que establecerá el objeto al que se concede el permiso.
- accessMask, máscara que establece qué permiso se concede.
- ObjectType, como la máscara puede ser compartida por varios permisos, mediante este string se distinguirán unos de otros.
- AceType, establece si el permiso está denegado o concedido al objeto
- Flags y AceFlags, establecen si el permiso se heredará

Por último. Con el fin de controlar que un usuario pueda o no pueda cambiar su password, utilizaremos reglas de acceso. Éstas se formarán con el identificador que indica si la regla de acceso se aplica al propio usuario (selfsid) o a todo el mundo (worldsid), si es de tipo permitida o denegada y, por último, el tipo de regla (en este caso el cambio de password) mediante un GUID (un identificador exclusivo global).

#### **6.3.1.2 Servidores y Servidores Exchange:**

Solo se accede al atributo que contiene su dirección, de tipo string.

### **6.3.2 Configuración internet Explorer:**

En esta parte, se acceden a tres datos distintos de tipo string. El primero indica si el Navegador está habilitado o no, para ello, devuelve un 1 en caso de estarlo y un 0 en caso contrario. El segundo devuelve el nombre del proxy asignado a internet Explorer. Y el tercero y último las excepciones por las que no pasará por el proxy.

### **6.3.3 Renombrar Perfil y Gestión de carpetas compartidas:**

En ambos casos, el tipo de dato que se maneja son carpetas. En el primero se renombrará y en el segundo, además, se crearán y eliminarán.



## 7 IMPLEMENTACIÓN

### 7.1 Clases utilizadas

Para la parte de la capa de presentación, se han implementado las interfaces mostradas anteriormente, que en vb.net toman el nombre de formularios.

Para la capa de lógica de negocio, se han implementado las funciones (operaciones en la parte de diseño) que llevarán a cabo los cambios sobre el directorio activo y que se han implementado en clases vb.net.

En la figura 7.2 se pueden apreciar tanto los formularios como las clases del proyecto:

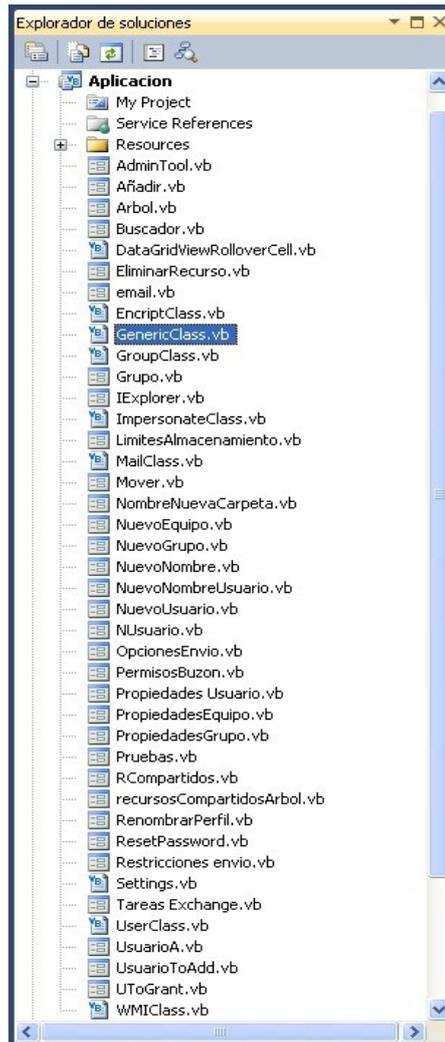


Figura 7.2

Dentro de las clases, UserClass y GroupClass incluirán las funciones que manipulen objetos de tipo usuario y grupo respectivamente, GenericClass incluirá funciones que manipulan objetos del tipo Equipo y otras que se encargan de operaciones generales sobre el directorio activo, como por ejemplo, cargar los servidores incluidos en el mismo. Las anteriores son las clases con mayor número de funciones. MailClass es la clase formada por las funciones que afectan al correo Exchange. WMIClass es la clase que incluye las funciones con las que se manipulan las carpetas. ImpersonateClass incluirá las funciones que permiten la impersonación. EncriptClass posee las funciones para poder acceder a los datos.

## 7.2 Código

En esta parte se mostrará el código utilizado para implementar las partes más significativas de la aplicación. Partes que o bien son claves, o bien no han quedado suficientemente claras en el análisis y el diseño.

### 7.2.1 Gestión del Directorio Activo:

#### 7.2.1.1 Entrada de Directorio Activo

La parte que se repite en todas las funciones de las clases que se utilizan para la gestión del directorio activo (userClass, groupClass y GenericClass) es en la que, previa búsqueda, se obtiene la entrada al directorio activo de un objeto. A continuación se explica su código paso a paso.

Primero se obtiene una entrada de directorio.

```
Dim de As DirectoryEntry = New DirectoryEntry(aquí se puede poner la
dirección del contenedor en el que está el objeto o se puede dejar vacío
para que obtenga la entrada de la raíz del directorio activo)
```

Se le ceden permisos de administrador del dominio. El nombre y password del administrador del dominio se han guardado previamente en la clase settings

```
de.Username = Settings.adminLogon
de.Password = Settings.adminPassword
de.AuthenticationType = AuthenticationTypes.Delegation
```

Se establece la consulta LDAP, en este caso objetos de tipo user que tengan el nombre que buscamos.

```
Dim deSearch As DirectorySearcher = New DirectorySearcher(de)
deSearch.Filter = "(&(objectClass=user) (sAMAccountName=" & Aquí
iría la variable con el nombre que buscamos & "))"
deSearch.SearchScope = SearchScope.Subtree
```

En result se carga el resultado de la consulta.

```
Dim result As SearchResult = deSearch.FindOne()
```

Y se puede crear la entrada al directorio activo del objeto puesto que uno de los atributos del resultado es su dirección LDAP.

```
Dim de As DirectoryEntry = New DirectoryEntry(results.path)
```

Para el caso de la búsqueda de objetos en el Directorio Activo, la consulta LDAP es un poco más compleja, ya que ha de buscar objetos cuyo nombre de usuario, nombre de login o de display comiencen por una cadena de caracteres determinada.

La consulta queda:

```
deSearch.Filter = "(&(objectClass=" & tipo de objeto & ")(|
(sAMAccountName=" & cadena a buscar & "*)|(cn=" & cadena a buscar & "*)
(displayName=" & cadena a buscar & "*))"
```

Y se obtendrán uno o mas resultados, así que el resultado de la consulta se cargará en

una lista de resultados.

```
Dim result As SearchResultCollection = deSearch.FindAll()
```

### 7.2.1.2 Propiedades

Todas las funciones que afectan a los objetos del directorio activo realizan cambios en los atributos de los mismos. Estos cambios se realizan de la manera que se muestra a continuación.

Una vez se han obtenido los resultados de una búsqueda de objetos, se puede obtener alguno de sus atributos

```
Dim oresult As SearchResult
```

Si se ha obtenido algún resultado en la consulta

```
If results.Count > 0 Then
```

Asigna a oresult el primer resultado y devuelve el valor de uno de sus atributos

```
    oresult = results.Item(0)
    Return
oreult.GetDirectoryEntry.Properties("NombreAtributo").Value
End If
```

Si se quiere establecer el valor de uno de sus atributos, se siguen los mismos pasos

```
Dim oresult As SearchResult
```

```
If results.Count > 0 Then
```

Asignarle los datos al atributo

```
    oresult = results.Item(0)
    oresult.GetDirectoryEntry.Properties("NombreAtributo").Value="
datos"
```

Se guardan los cambios realizados en el objeto

```
    oresult.GetDirectoryEntry.CommitChanges()
End If
```

Eliminar uno de sus atributos, lo que hace es asignarle el valor nulo.

```
Dim oresult As SearchResult
If results.Count > 0 Then
```

Se elimina el valor del atributo

```
    oresult = results.Item(0)
    Return
oreult.GetDirectoryEntry.Properties("NombreAtributo").Clear
    oresult.GetDirectoryEntry.CommitChanges()
End If
```

Si el atributo es una lista de elementos, se puede añadir otro

```
Dim oresult As SearchResult
If results.Count > 0 Then
```

Se añade el dato

```
    oresult = results.Item(0)
    oresult.GetDirectoryEntry.Properties("NombreAtributo").Add("dato")
    oresult.GetDirectoryEntry.CommitChanges()
End If
```

Aunque para este ejemplo se ha utilizado un objeto de tipo usuario, cambiando el valor que se asigna a objectclass, se pueden utilizar para cualquiera de los demás tipos de objeto.

### 7.2.1.3 Crear Objeto

Este código se usa en CreateAdAccount, CreateGroup y CreateComputer de las clases UserClass, GroupClass y GenericClass.

Para crear un objeto se obtiene la entrada del directorio activo que apunta al Contenedor, también llamado Unidad Organizativa (OU), en la que se va a crear el objeto.

```
Dim dirE As New DirectoryEntry()
Dim dirEntry As DirectoryEntry = dirE.Children.Find("OU en la que se va a crear el objeto")
dirEntry.Username = Settings.adminLogon
dirEntry.Password = Settings.adminPassword
dirEntry.AuthenticationType = AuthenticationTypes.Delegation
```

Se crea en esa OU un nuevo objeto con el tipo y el nombre deseado

```
Dim newUser As DirectoryEntry
newUser = dirEntry.Children.Add("CN=" & "Nombre de el objeto", "User")
```

Posteriormente, se debe asignarle más atributos como, por ejemplo, su nombre de login como hemos visto en el apartado anterior.

Este será el ejemplo para objetos de tipo usuario, pero para grupos y equipos solo habría que cambiar la palabra "User" por "Group" o "Computer".

### 7.2.1.4 Mover Objeto

Código de la función MoveTo de la clase GenericClass

Para eliminar un objeto se obtiene la entrada al directorio activo (oUser) que apunta al objeto y éste llama a la función MoveTo en cuyo parámetro introduce la nueva OU a la que se quiere enviar el objeto.

```
oUser.MoveTo(New DirectoryEntry("nueva dirección a la que se moverá el objeto"))
```

### 7.2.1.5 Renombrar Objeto

La función Rename de la clase GenericClass utiliza este código

De la misma manera que se utilizaba la función MoveTo para mover el objeto se utiliza la función MoveHere para renombrarla, salvo que en vez de un parámetro tendrá dos, su dirección Ldap y el nuevo nombre que se le quiere asignar.

```
oUser.MoveHere("dirección LDAP al objeto", "CN=" & nuevo nombre)
```

### 7.2.1.6 Eliminar Objeto

El código que se explica a continuación será el que utilice la función RemoveFromOU de la clase GenericClass

De una manera muy parecida a como se crean los objetos, para eliminarlos se necesita la entrada del directorio activo que apunta a la OU (de) donde se encuentra el objeto a eliminar y otra que apunte directamente al objeto(oUser). La diferencia será que en vez de llamar a la función add, se llama a la función Remove

```
de.Children.Remove(oUser)
```

### 7.2.1.7 Establecer Password

El código descrito a continuación se usa en createAdAccount y changePassword de la clase UserClass.

Para asignarle un password nuevo (tanto la primera vez como al resetearlo) al usuario, necesitaremos invocar a la función SetPassword  
Se obtiene el objeto de tipo usuario

```
Dim deuser As New DirectoryEntry("dirección del objeto usuario")
```

Se crea el objeto password

```
Dim oPassword As Object() = New Object() {"string con el nuevo  
password"}
```

Se asigna el nuevo password al usuario invocando la función Setpassword

```
deuser.Invoke("SetPassword", oPassword)
```

Se guardan los cambios realizados en el objeto

```
deuser.commitchanges()
```

Esta asignación se debe hacer obligatoriamente al crear un nuevo usuario.

### 7.2.1.8 Establecer Reglas de Acceso

El siguiente código se usa en las funciones DenyChangePassword, AllowPasswordDenied, ChangePaswordDenied, de la clase UserClass.

Para generar y asignar una regla de acceso que autorice o deniegue el cambio de password a un usuario se deben dar los siguientes pasos:

Crear el GUID (identificador único) de cambio de password

```
Dim changePasswordGuid As New Guid("{AB721A53-1E2F-11D0-9819-00AA0040529B}")
```

Obtener la seguridad del usuario, user es la entrada al directorio activo de un objeto del tipo usuario

```
Dim userSecurity As ActiveDirectorySecurity = user.ObjectSecurity
```

Obtener los identificadores(Sid) de seguridad para el usuario mismo y cualquiera

```
Dim everyoneSid As New SecurityIdentifier(WellKnownSidType.WorldSid, Nothing)
Dim selfSid As New SecurityIdentifier(WellKnownSidType.SelfSid, Nothing)
```

Crear las reglas de acceso denegar a sí mismo y a todo el mundo, así como permitir a sí mismo y a todo el mundo

```
Dim allowEveryone As New ActiveDirectoryAccessRule(everyoneSid, _
ActiveDirectoryRights.ExtendedRight, _
AccessControlType.Allow, _
changePasswordGuid)

Dim denyEveryone As New ActiveDirectoryAccessRule(everyoneSid, _
ActiveDirectoryRights.ExtendedRight, _
AccessControlType.Deny, _
changePasswordGuid)

Dim allowSelf As New ActiveDirectoryAccessRule(selfSid, _
ActiveDirectoryRights.ExtendedRight, _
AccessControlType.Allow, _
changePasswordGuid)

Dim denySelf As New ActiveDirectoryAccessRule(selfSid, _
ActiveDirectoryRights.ExtendedRight, _
AccessControlType.Deny, _
changePasswordGuid)
```

Añadir o eliminar las reglas de acceso según lo que se quiera hacer. Por ejemplo, para denegar el cambio de password quedaría:

```
userSecurity.RemoveAccessRuleSpecific(allowEveryone)
userSecurity.AddAccessRule(denyEveryone)
userSecurity.RemoveAccessRuleSpecific(allowSelf)
userSecurity.AddAccessRule(denySelf)
```

Para terminar se guardan los cambios

```
user.CommitChanges()
```

### 7.2.1.9 Buzón Exchange

El código que se muestra a continuación se usa en CreateMailbox, DeleteMailbox y MoveMailbox de la clase MailClass.

Para crear un buzón para un usuario existente, utilizamos la interfaz de Almacen (IMailboxStore) de buzones que nos lo permitirá.

*Crear el Buzón:*

Obtener la entrada al directorio activo del usuario dueño del buzón

```
Dim deMail As New DirectoryEntry("dirección del usuario dueño del buzón a crear")
```

Se realiza un casting para asignarle al usuario el tipo de la interfaz ImailboxStore

```
Dim mailboxStore As IMailboxStore  
mailboxStore = CType(deMail.NativeObject, IMailboxStore)
```

Se crea el buzón

```
mailboxStore.CreateMailbox("dirección del almacén en el que se creará el buzón")
```

Se guardan los cambios

```
deMail.CommitChanges()
```

*Mover el Buzón:*

Para mover un buzón realizamos una operación muy parecida a la anterior

Obtener usuario

```
Dim deMail As New DirectoryEntry("dirección del usuario dueño del buzón a crear")
```

Asignarle el tipo de la interfaz ImailboxStore

```
Dim mailboxStore As IMailboxStore  
mailboxStore = CType(deMail.NativeObject, IMailboxStore)
```

Se llama a la función para mover el buzón de un almacén a otro

```
mailboxStore.MoveMailbox("dirección del almacén al que se moverá el buzón")
```

Se guardan los cambios

```
deMail.CommitChanges()
```

*Eliminar el buzón:*

Por último, eliminar el buzón será prácticamente igual que los anteriores

```
Dim deMail As New DirectoryEntry("dirección del usuario dueño del buzón a crear")  
Dim mailboxStore As IMailboxStore
```

```
mailboxStore = CType(deMail.NativeObject, ImailboxStore)
```

La diferencia es que aquí se llama a la función que elimina el buzón

```
mailboxStore.DeleteMailbox()
```

Se guardan los cambios

```
deMail.CommitChanges()
```

### 7.2.1.10 Seguridad

Este código será común a las funciones GetAcePermission, GetAcePermissionUserNT, GetAcePermissioNT, GetAcePermissioComputerNT, AddAceToDaclNT, AddAceToDacl, AddAceToDaclUserNT, AddAceToDaclComputerNT, RemoveAceFromDaclNT, RemoveAceFromDacl, RemoveAceFromDaclUserNT, RemoveAceFromDaclComputerNT, GetDaclNames, GetDaclNamesNT, RemoveAces, RemoveAcesMailbox, de la clase MailClass.

Los permisos concedidos sobre un objeto se guardan en una lista dentro del atributo ntSecurityDescriptor del mismo. Para añadir permisos o eliminarlos no habrá más que modificar esa lista.

Paso a paso sería:

Se obtiene la entrada de directorio activo del objeto y se dan permisos de controlador de dominio

```
Dim deMail As New DirectoryEntry(Aquí se introduce la dirección  
LDAP del objeto en el directorio activo)  
deMail.Username = Settings.adminLogon  
deMail.Password = Settings.adminPassword  
deMail.AuthenticationType = AuthenticationTypes.Delegation
```

Se obtiene el atributo SecurityDescriptor y se asigna a una variable

```
Dim oSecurityDescriptor As IADsSecurityDescriptor =  
CType(deMail.Properties("ntSecurityDescriptor").Value,  
IADsSecurityDescriptor)
```

Se guarda en una variable la lista de permisos concedidos sobre el objeto

```
dacl = oSecurityDescriptor.DiscretionaryAcl
```

Se establece ace como una variable del tipo AccessControlEntry, el tipo que utilizan los permisos

```
ace = CreateObject("AccessControlEntry")
```

El tipo ace tiene cinco campos clave:

- ace.trustee – en el que se escribe qué objeto posee el permiso
- ace.AccessMask – se trata de la máscara que establece el permiso que se concede, lectura, escritura etc.
- ace.objectType – si el permiso que se quiere asignar es más específico de lo

que se puede indicar con la máscara , por ejemplo lectura de información personal cuando en la máscara solo se puede indicar lectura, se le añade otro identificador en este campo.

ace.AceType – indica si el permiso se deniega o se concede

ace.AceFlags – establece si el permiso lo heredan los hijos del objeto

Ahora se puede crear un nuevo permiso (ACE) y añadirlo a la lista.

```
dacl.AddAce(ace)
```

O recorrerla y al encontrar el deseado, eliminarlo.

```
dacl.RemoveAce(ace)
```

Se actualiza la lista de permisos del objeto

```
oSecurityDescriptor.DiscretionaryAcl = dacl  
deMail.Properties("ntSecurityDescriptor").Value =  
oSecurityDescriptor
```

Se guardan los cambios realizados

```
deMail.CommitChanges()
```

Para el caso de los permisos sobre el buzón de correo, solo cambia que en vez e utilizar el atributo ntSecurityDescriptor para obtener la lista de permisos, se utiliza el MailboxRights.

Respecto del código anterior solo cambia la carga de la variable oSecurityDescriptor, que quedará:

```
oSecurityDescriptor = objUser.MailboxRights
```

### **7.2.2 Cambios en la configuración de Internet Explorer:**

El código que se muestra a continuación se usará en las funciones IexplorerCarga e IexplorerDatos de la clase WMIClass.

Para gestionar la configuración de Internet Explorer de manera remota, utilizamos la clase WMI. Esta clase nos proporciona acceso a la información de los objetos del entorno administrativo. Así, podremos acceder a los datos de proxy y excepciones asignadas al usuario que, en el momento en el que se accede a ellos, está logeado en la máquina.

En una constante se guarda el hexadecimal que se utiliza para marcar posteriormente que los cambios se realicen sobre el usuario logeado en la máquina

```
Const HKEY_CURRENT_USER = &H80000001
```

Se crea el locator

```
Dim oLocator As Object =  
CreateObject("wbemScripting.SWbemLocator")  
Dim strComputer As String = "Nombre de la máquina"
```

Se crea el Service y conecta con la máquina con permisos de administrador de dominio

```

Dim oService As Object = oLocator.ConnectServer(strComputer,
"Root\DEFAULT", Settings.adminLogon, Settings.adminPassword)

```

Se obtiene el tipo de service

```

Dim objRegistry As Object = oService.Get("StdRegProv")

```

Se guarda la dirección a los datos de configuración de la conexión a internet de la máquina

```

Dim strKeyPath As String =
"SOFTWARE\Microsoft\Windows\CurrentVersion\Internet Settings"

```

Ahora se puede obtener los datos de configuración de internet asignados al usuario logeado en la máquina para alterarlos como se quiera.

### 7.2.2.1 Habilitar/Deshabilitar el proxy actual

```

strValueName = "ProxyEnable"
Dim strV As String
objRegistry.GetStringValue(HKEY_CURRENT_USER, strKeyPath,
strValueName, strV)

```

strV será 0 si está deshabilitado y 1 si está habilitado.

Cambio del estado del proxy

```

objRegistry.SetDWORDValue(HKEY_CURRENT_USER, strKeyPath,
strValueName, "0 si queremos deshabilitarlo, 1 en caso contrario")

```

### 7.2.2.2 Obtener y modificar el Proxy actual

```

strValueName = "ProxyServer"
Dim strV As String
objRegistry.GetStringValue(HKEY_CURRENT_USER, strKeyPath,
strValueName, strV)

```

strV tendrá el nombre del proxy actual

Cambio de proxy

```

objRegistry.SetStringValue(HKEY_CURRENT_USER, strKeyPath,
strValueName, "nombre nuevo proxy")

```

### 7.2.2.3 Obtener y modificar las excepciones del proxy

```

strValueName = "ProxyOverride"
Dim strV As String
objRegistry.GetStringValue(HKEY_CURRENT_USER, strKeyPath,
strValueName, strV)

```

strV será un string con las excepciones del proxy actuales.

Cambio de las excepciones del proxy

```

objRegistry.SetStringValue(HKEY_CURRENT_USER, strKeyPath,
strValueName, "nuevas excepciones de proxy")

```

### 7.2.3 Acceso a carpetas en remoto para Gestión de Carpetas y Cambio de nombre de perfil

El código que se muestra a continuación se usará en las funciones createFolder, deleteFolder, renameFolder y renameProfile de la clase WMIClass.

Se puede dividir las operaciones que manipulan carpetas en tres partes diferenciadas.

Primero se realiza la impersonificación, para que el usuario adquiera permisos de administrador de dominio y pueda crear, eliminar o modificar carpetas:

```
Dim oImp1 As Impersonation
oImp1 = New Impersonation()
oImp1.StartImpersonation(Settings.adminLogon,
Settings.userDNSDomain, Settings.adminPassword)
```

Después se accede a la carpeta que se quiere modificar, eliminar o crear en ella otra carpeta, para saber si es una carpeta de sistema que no puede ser manipulada:

```
objFSO = CreateObject("Scripting.FileSystemObject")
objFolder = objFSO.GetFolder(dirección de la carpeta)
If (objFolder.Attributes > 21) Then
    MsgBox("No se pueden modificar las carpetas de sistema")
End if
```

objFolder.Attributes es una máscara que establece el tipo de carpeta.

Para terminar se realiza la acción que se quiera mediante las librerías Directory:

#### 7.2.3.1 Crear carpeta

```
Directory.CreateDirectory(dirección en la que se quiere crear la
carpeta & nombre)
```

#### 7.2.3.2 Eliminar carpeta

```
Directory.Delete(dirección de la carpeta a eliminar, True)
```

#### 7.2.3.3 Renombrar carpeta, que en realidad la mueve a la dirección con su nuevo nombre

```
Directory.Move(dirección de la carpeta con el nombre nuevo)
```

El cambio de nombre de perfil, lo único que hace es renombrar una carpeta en la dirección C\$\Documents and Settings.



## 8 PRUEBAS:

Una de las últimas fases del ciclo de vida antes de entregar un programa para su explotación, es la fase de pruebas.

Como parte que es de un proceso industrial, la fase de pruebas añade valor al producto que se maneja: todos los programas tienen errores y la fase de pruebas los descubre; ese es el valor que añade. El objetivo específico de la fase de pruebas es encontrar cuantos más errores, mejor.

La prueba ideal de un sistema sería exponerlo en todas las situaciones posibles, así encontraríamos hasta el último fallo. Indirectamente, garantizamos su respuesta ante cualquier caso que se le presente en la ejecución real. Pero al no ser posible, puesto que someter la aplicación a todas las pruebas posibles resultaría en llevar a cabo un número infinito de ellas, y puesto que ésta está claramente orientada al usuario, se decidió realizar básicamente dos tipos de pruebas, de caja negra y de aceptación. No obstante, durante la fase de implementación ya se han realizado pruebas de una manera informal, ejecutando el código para convencerse de que el programa "básicamente, funciona".

### 8.1 Pruebas unitarias

#### 8.1.1 caja negra

Las pruebas de caja negra se centran en lo que se espera de un módulo, es decir, intentan encontrar casos en que el módulo no se atiene a su especificación. Por ello se denominan pruebas funcionales, y el probador se limita a suministrarle datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo el módulo por dentro.

A continuación aparecerá una enumeración de las pruebas de caja negra realizadas en cada iteración, éstas son las que deben devolver un mensaje de aviso porque se está intentando llevar a cabo una acción no permitida. También se probó a cambiar cada uno de los campos de la interfaz para; después de cada cambio; hacer clic en el botón aceptar, con el fin de comprobar que los cambios se hubiesen producido.

Todas las pruebas de caja negra se realizaron sobre cuatro máquinas virtuales. Por un lado dos Windows Server 2003, tomando uno de ellos el rol de controlador de dominio, en los que se instalos Exchange 2003 pasando así a ser servidores de correo. Por otro lado se utilizaron otras dos máquinas a modo de cliente. Un Windows XP y un Windows 7.

##### 8.1.1.1 Pruebas en la primera iteración:

*Pruebas en la creación de un objeto del tipo usuario*

Panel en la que se establece el nombre dentro de la interfaz para Crear un Usuario.

- Dejar los campos claves, como nombre de usuario o nombre de login en blanco.
- Insertar un nombre de usuario o de login que ya existen en el dominio.
- Insertar un nombre de usuario o de login que no tienen el formato adecuado.

Panel en la que se establece el password dentro de la interfaz para Crear un Usuario.

- Dejar uno o los dos campos correspondientes a las contraseñas en blanco.
- Escribir una contraseña en el campo de confirmación diferente a la del primer campo de las contraseñas.
- Escribir una contraseña que no cumpla con las exigencias de fortaleza de la misma que pide el dominio.
- Marcar opciones incompatibles entre sí en las opciones para la contraseña. Por ejemplo, que la contraseña no se pueda cambiar nunca y que se deba cambiar en el siguiente login (las dos primeras opciones).

Panel en la que se establece si tiene buzón el usuario de la interfaz para Crear un Usuario.

- No escribir el alias para el buzón una vez marcada la opción de que se quiere crear un buzón para el usuario.
- No seleccionar el servidor y el almacén de buzones en el que se quiere crear el buzón del usuario cuando está marcada la opción de crearlo.

#### *Pruebas en la creación del objeto del tipo grupo*

Panel en la que se establece el nombre dentro de la interfaz para Crear un Grupo.

- Dejar el campo clave nombre del grupo en blanco.
- Insertar como nombre de grupo uno que ya exista en el dominio.
- Insertar como nombre de grupo uno que no tenga el formato adecuado.

Panel en la que se establece si posee correo el grupo dentro de la interfaz para Crear un Grupo.

- No escribir el alias para la dirección de correo asignada al grupo una vez marcada la opción de que se quiere crear una dirección de correo electrónico para el grupo.
- No seleccionar el grupo administrativo en el que se quiere crear la dirección de correo del grupo cuando está marcada la opción de crearla.

#### *Pruebas en la creación del objeto del tipo equipo*

Panel en la que se establece el nombre dentro de la interfaz para Crear un Equipo.

- Dejar el campo clave nombre del equipo en blanco.
- Insertar como nombre de equipo uno que ya exista en el dominio.
- Insertar como nombre de equipo uno que no tenga el formato adecuado.
- Al cambiar el usuario o grupo que puede añadir el equipo al dominio, no elegir ninguno.

Panel en la que se establece si el equipo es administrado dentro de la interfaz para Crear un Equipo.

- Dejar en blanco el campo ID unitaria del equipo una vez marcada la opción de que es un ordenador administrado.
- Insertar una identidad unitaria del equipo que no es del formato adecuado una vez marcada la opción de que es un ordenador administrado.

### **8.1.1.2 Pruebas para la segunda iteración**

#### *Pruebas para los cambios en las propiedades del objeto del tipo usuario*

##### Panel General

- Dejar en blanco los campos nombre completo o nombre de login
- Insertar un nombre completo para el usuario o un nombre de login que ya existan en el dominio.
- Insertar un nombre completo o un nombre de login que no tengan el formato adecuado.
- En todos los campos intentar sobrepasar el tamaño máximo de caracteres de los mismos.

##### Panel Account

- Marcar opciones incompatibles entre sí en las opciones para la contraseña. Por ejemplo, que la contraseña no se pueda cambiar nunca y que se deba cambiar en el siguiente login (las dos primeras opciones).
- Comprobar que si el usuario está bloqueado la opción de desbloquearlo está habilitada y comprobar si funciona.
- Asignar a la cuenta de usuario una fecha de expiro anterior a la fecha actual.

##### Panel Profile

- Insertar en los paneles direcciones con formatos que no sean los adecuados.

##### Panel Member of

- Insertar un grupo vacío en la lista de grupos de los que es miembro el usuario.
- Eliminar el grupo primario del usuario.

##### Panel Security

- Alterar un permiso heredado.
- Asignar permisos a un objeto vacío, es decir, al añadir un objeto al que asignarle permisos no seleccionar ninguno.

##### Panel opciones generales de Exchange

##### Ventana Restricciones de envío

- Asignar valores no numéricos a los tamaños de los mensajes de salida y entrada
- Asignar la capacidad de recibir mensajes o denegar la misma a un usuario vacío, es decir, al añadir el usuario sobre el que se quiere establecer si se podrán recibir mensajes o no, no seleccionar ninguno.

## Ventana Opciones envío

- Asignar la capacidad de envío como el usuario a un usuario vacío, es decir, al añadir el usuario al que se quiere conceder la capacidad de enviar como el usuario, no seleccionar ninguno.
- Marcar la opción de reenvío o otro usuario y que éste sea vacío, es decir, al añadir el usuario al que se quiere reenviar los correos, no seleccionar ninguno.
- Asignar un valor no numérico al límite de destinatarios máximo para los correos de salida.

## Ventana Límites de almacenamiento

- Asignar valores no numéricos a los límites que se asignan en esa ventana.

## Panel Direcciones de correo

- Añadir una nueva dirección de correo o modificar una existente y que la dirección nueva ya exista en el dominio.
- Añadir una nueva dirección de correo o modificar una existente y que la dirección nueva sea una dirección en blanco.
- Intentar eliminar la dirección primaria del usuario.
- Intentar establecer como primaria una dirección de correo cuando no se ha seleccionado ninguna de las de la tabla.

## Panel Características de Exchange

- Intentar habilitar up-to-date notifications sin que esté habilitada User initiated synchronization.
- Intentar deshabilitar User initiated synchronization estando habilitada up-to-date notifications.

## Panel opciones avanzadas de Exchange

### Ventana Permisos de buzón

- Alterar un permiso heredado.
- Asignar permisos a un objeto vacío, es decir, al añadir un objeto al que asignarle permisos no seleccionar ninguno.

### *Pruebas para los cambios en las propiedades del objeto del tipo grupo*

### Panel general

- Dejar en blanco el campo nombre.
- Insertar un nombre para el grupo que ya existan en el dominio.
- Insertar un nombre que no tenga el formato adecuado.
- Intentar sobrepasar el tamaño máximo de caracteres del campo descripción.

### Panel members

- Intentar añadir un miembro al grupo pero no seleccionar ninguno en la ventana de búsqueda que aparece al darle al botón añadir.

-Intentar eliminar un miembro sin haber seleccionado ninguno.

#### Panel member of

-Intentar añadir un al grupo como miembro de otro pero no seleccionar ninguno en la ventana de búsqueda que aparece al darle al botón añadir.

-Intentar eliminar un grupo de los que es miembro el grupo sin haber seleccionado ninguno.

-Intentar añadir el grupo como miembro en otro del que no pueda ser miembro.

#### Panel direcciones de correo

-Añadir una nueva dirección de correo o modificar una existente y que la dirección nueva ya exista en el dominio.

-Añadir una nueva dirección de correo o modificar una existente y que la dirección nueva sea una dirección en blanco.

#### Panel opciones generales

-Asignar valores no numéricos a los tamaños de los mensajes.

-Asignar la capacidad de recibir mensajes o denegar la misma a un usuario vacío, es decir, al añadir el usuario sobre el que se quiere establecer si se podrán recibir mensajes o no, no seleccionar ninguno.

#### Panel opciones avanzadas

-Seleccionar a la vez mas de uno de los tres radio buttons.

#### Panel Managed by

-Asignar un usuario vacío como administrador del grupo

#### Panel Security

-Alterar un permiso heredado.

-Asignar permisos a un objeto vacío, es decir, al añadir un objeto al que asignarle permisos no seleccionar ninguno.

#### *Pruebas en las propiedades del objeto del tipo equipo*

#### Panel General

-Dejar en blanco el campo nombre.

-Insertar un nombre para el equipo que ya existan en el dominio.

-Insertar un nombre que no tenga el formato adecuado.

-Intentar sobrepasar el tamaño máximo de caracteres del campo descripción.

#### Panel Member of

-Insertar un grupo vacío en la lista de grupos de los que es miembro el equipo.

-Eliminar el grupo primario del equipo.

Panel Managed by

-Asignar un usuario vacío como administrador del equipo

Panel Security

-Alterar un permiso heredado.

-Asignar permisos a un objeto vacío, es decir, al añadir un objeto al que asignarle permisos no seleccionar ninguno.

### **8.1.1.3 Pruebas para la tercera y cuarta iteración**

*Pruebas sobre casos de uso no específicos de las propiedades de los objetos pero que forman parte de los incluidos en la gestión del directorio activo*

Habilitar/Deshabilitar usuario o equipo

-Habilitar un usuario o un equipo ya habilitado

-deshabilitar un usuario o un equipo ya deshabilitado

Mover un usuario, grupo o equipo

- No seleccionar la OU a la que mover el grupo y aceptar.

Cambiar nombre a un usuario, grupo o equipo

-Insertar un nombre para el equipo que ya existan en el dominio.

-Insertar un nombre que no tenga el formato adecuado.

Para el caso del usuario al abrirse una ventana con cuatro campos, las pruebas se realizarán sobre nombre completo y nombre de login.

Copiar un usuario, grupo o equipo

-Copiar un objeto cuando ya existe otro en el dominio con su nombre precedido del string "copia de".

Eliminar un usuario, grupo o equipo

-Eliminar un usuario o un equipo no deshabilitado.

-Eliminar uno de los objetos imprescindibles del directorio activo, como por ejemplo el grupo Administradores.

Reseteo contraseña de un usuario

-Dejar uno o los dos campos correspondientes a las contraseñas en blanco.

-Escribir una contraseña en el campo de confirmación diferente a la del primer campo de las contraseñas.

Desbloquear un usuario

-Desbloquear un usuario que no está bloqueado

Tareas Exchange usuario o grupo

Tareas Exchange usuario

Crear Buzón

-No escribir el alias para el buzón.

-No seleccionar el servidor y el almacén de buzones en el que se quiere crear el buzón del usuario.

Mover Buzón

-No seleccionar el servidor y el almacén de buzones al que se quiere mover el buzón del usuario.

Tareas Exchange grupo

Establecer correo

-No escribir el alias para la dirección de correo.

-No seleccionar el grupo administrativo en el que se quiere crear la dirección de correo del grupo.

*Pruebas para los casos de uso correspondientes a la gestión de carpetas compartidas*

-Añadir o eliminar un recurso compartido del servidor

-Crear o renombrar una carpeta y que el nuevo nombre sea vacío.

-Crear o renombrar una carpeta y que el nuevo nombre no tenga el formato adecuado.

Ventana de Seguridad carpeta

-Alterar un permiso heredado.

-Asignar permisos a un objeto vacío, es decir, al añadir un objeto al que asignarle permisos no seleccionar ninguno.

Pruebas para los casos de uso correspondientes a la gestión de la configuración de Internet Explorer

-Cargar los datos de un equipo que no existe en el dominio.

-Cargar los datos de Internet Explorer cuando no se ha escrito ningún nombre de equipo sobre el que cambiarlos.

-Una vez cargados los datos de manera satisfactoria añadir un nombre de proxy, número de puerto de proxy o excepción que no tengan el formato adecuado.

Pruebas para los casos de uso correspondientes al cambio de nombre de un perfil

-Cambiar el nombre de perfil que se encuentra en un equipo que no existe en el dominio.

-Cambiar el nombre de perfil cuando no se ha escrito el nombre de equipo en el que se

encuentra.

-Cambiar el nombre de perfil cuando no se ha escrito un nombre de perfil que se encuentre en ese equipo.

-Cambiar el nombre de perfil cuando el nuevo nombre a asignar al perfil no tenga el formato adecuado.

### **8.1.2 Pruebas aceptación**

Estas pruebas las realiza el cliente. Son básicamente pruebas funcionales, sobre el sistema completo, y buscan una cobertura de la especificación de requisitos y del manual del usuario. Estas pruebas no se realizan durante el desarrollo, pues sería impresentable de cara al cliente; sino una vez pasadas todas las pruebas de integración por parte del desarrollador.

Para el caso que nos ocupa, se instaló la aplicación AdminTool en el equipo de uno de los trabajadores del CAU, que se dedicó a realizar la labor habitual de su puesto de trabajo utilizándola. Avisando de los errores encontrados una vez terminaba su jornada laboral.

Las pruebas se realizaron durante tres semanas y se incluyeron tanto errores como mejoras para la aplicación. A continuación se incluye una descripción de los mismos.

#### **8.1.2.1 Errores**

Problemas con los permisos sobre las carpetas de los recursos compartidos

Se detectó que para acceder a las carpetas compartidas no se impersonaba previamente, así que no se tenían permisos para acceder a las mismas. Se añadió y se solucionó el problema

Falta de nombres en las etiquetas de añadir equipo

Se insertaron los que faltaban.

Búsqueda sin escribir nada

Se añadió un aviso y se dejó de permitir la búsqueda.

Quitar frontends de la lista de los servidores de correo

Se añadió una condición en la búsqueda ldap de los mismos para obtener solo los backends.

Orden de apellidos y luego nombre

Se cambió el orden con el fin de que apareciesen los apellidos seguidos de una coma y luego el nombre.

Problemas por falta de permisos en algunos casos de uso

En el entorno de pruebas no se había tenido en cuenta que se podría dar el caso en el que el usuario no tuviese los permisos suficientes como para ver ciertos objetos. En estos casos se añadió en la implementación el que se certificase el usuario tomando el rol de administrador.

#### **8.1.2.2 Mejoras**

Añadir los campos oficina, número de teléfono y página web en las propiedades del

usuario.  
Se añadieron.

Añadir menú del botón derecho en la búsqueda  
Se añadió el mismo menú que aparece en la tabla que incluye los objetos al hacer clic con el botón derecho sobre cualquiera de ellos.

Copiar permisos de otro usuario  
Se añadió un botón en el panel Security del usuario, que permite seleccionar otro y asignarle al usuario los mismos permisos que posee el seleccionado.

Copiar miembros del grupo en el portapapeles  
Se copian los miembros de un grupo con el carácter “;” entre ellos para poder asignarlos en el campo de destinatarios de un correo en outlook.

Mensaje al eliminar  
Se añade un mensaje que obliga a confirmar que se quiere eliminar un objeto o una carpeta, con el fin de evitar que se eliminen de manera accidental.



## 9 GESTIÓN

Una vez realizado el proyecto, gracias al seguimiento del mismo, se puede concluir que la planificación inicial ha sufrido importantes cambios. Éstos se han producido por diversos factores, la inexperiencia en la realización de proyectos de este tipo (sobre todo en cuanto a la duración de las pruebas, que se han extendido más de lo esperado), a añadidos que se le han hecho a la aplicación en mitad de su desarrollo y a problemas para adecuar la aplicación a las diversas plataformas en las que se quería instalar.

### 9.1 Incidencias

#### 9.1.1 Seguimiento de la Fase 1:

Se tardó un día más de lo esperado, porque se tuvo que discutir los aspectos del mismo con el director del proyecto, sobre todo los objetivos del mismo y el alcance. Algo apreciable al observar la figura 9.1.1.

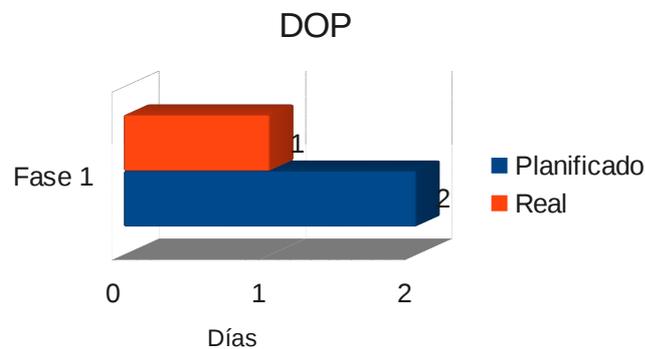


Figura 9.1.1 Seguimiento Fase 1

#### 9.1.2 Seguimiento de la Fase 2:

La primera iteración resultó más costosa de lo esperado. Este sobrecoste se debió a que la parte de análisis y la de diseño se alargaron por que no se conocía previamente el Directorio Activo. Como muestra la figura 9.1.2.1 queda.

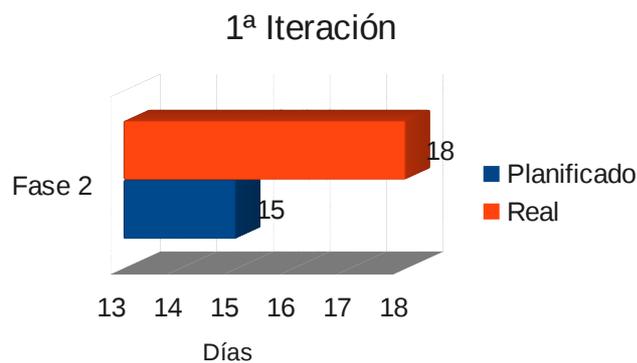


Figura 9.1.2.1 Seguimiento Fase 2

Como se muestra en la figura 9.1.2.2, los cambios en esta fase se produjeron en el análisis y en el diseño. El análisis es la parte que más aumentó, 12 horas, puesto que en ella se tuvo que buscar información sobre el Directorio activo y Visual Basic .NET para

decidir como afrontar la realización de los primeros casos de uso del proyecto.

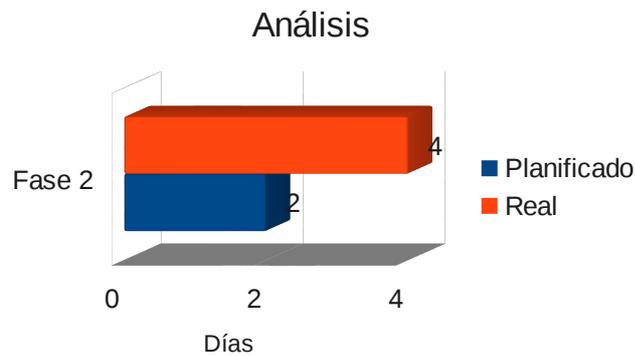


Figura 9.1.2.2 Análisis Fase 2

Por la misma razón que aumentó el análisis, por ser el comienzo de un proyecto basado en tecnologías que se desconocían, el diseño se alargó en 6 horas. Como se puede ver en la figura 9.1.2.3.

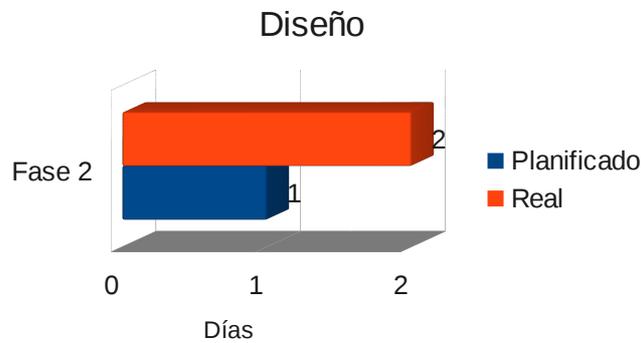
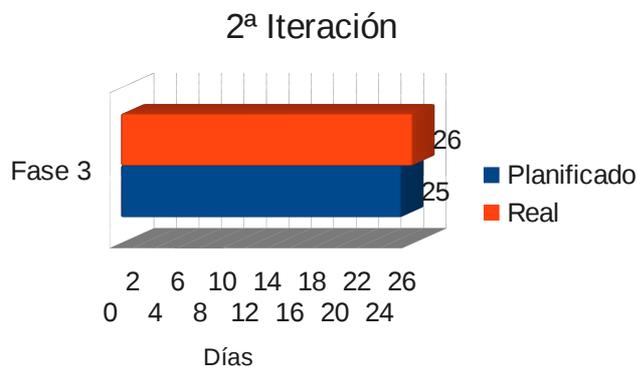


Figura 9.1.2.3 Diseño Fase 2

### 9.1.3 Incidencias en la Fase 3:

Una vez pasada la parte de adaptación a las tecnologías a utilizar. La segunda iteración resultó mas cercana a lo planificado, como se puede ver en la figura 9.1.3.1. No obstante, el número de casos de uso a implementar en esta fase resultó algo mayor de lo



esperado, lo que propició que se alargara el diseño.

Figura 9.1.3.1 Seguimiento Fase 3

El diseño se alargó puesto que al principio no se tenía muy claro como poder insertar todos los casos de uso que se encargan de la gestión de las propiedades de los objetos en

una sola interfaz por tipo del mismo. Lo que supuso un día mas de trabajo de lo esperado. Esto queda claro al ver la figura 9.1.3.2

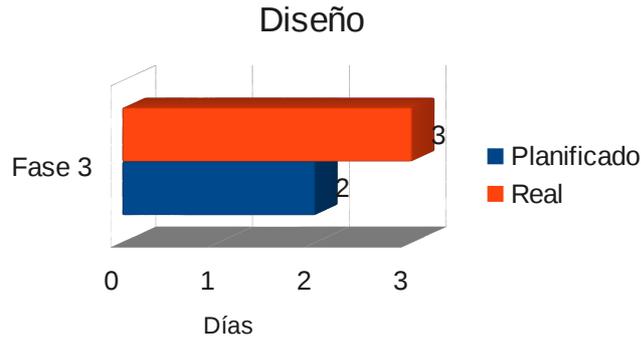


Figura 9.1.3.2 Diseño Fase 3

#### 9.1.4 Incidencias en la fase 4:

La tercera iteración resultó más sencilla de lo esperado puesto que a esas alturas del proyecto la soltura que se tenía tanto con el medio como con el lenguaje de programación era mayores. Al final, como se puede apreciar en la figura 9.1.4.1 se tardó un día menos de lo esperado en la implementación.

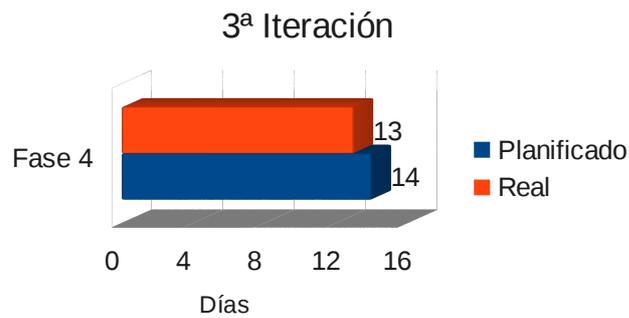


Figura 9.1.4.1 Seguimiento Fase 4

La implementación y pruebas se llevó a cabo en 54 horas en vez de las 60 planificadas, como se puede ver en la figura 9.1.4.2.

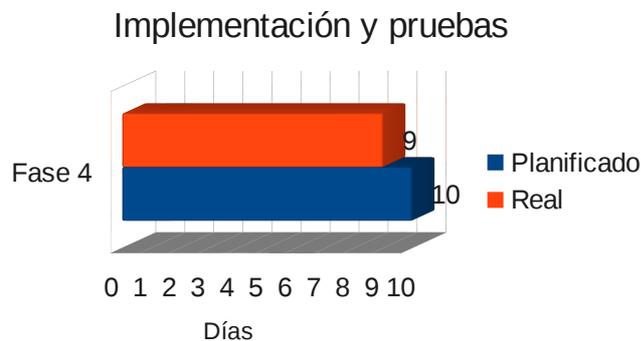


Figura 9.1.4.2 Implementación Fase 4

### 9.1.5 Incidencias en la fase 5:

La cuarta y última iteración, como es apreciable en la figura 9.1.5.1, aumentó de manera considerable. La razón es que al planificarla no se tuvo en cuenta que las tecnologías cambiaban, se pasó de implementar funciones que modificaban el directorio activo a otras que modificaban carpetas y la configuración del navegador internet Explorer en equipos remotos. Esto supuso que aumentarían todas las partes de la fase excepto la captura de requisitos.

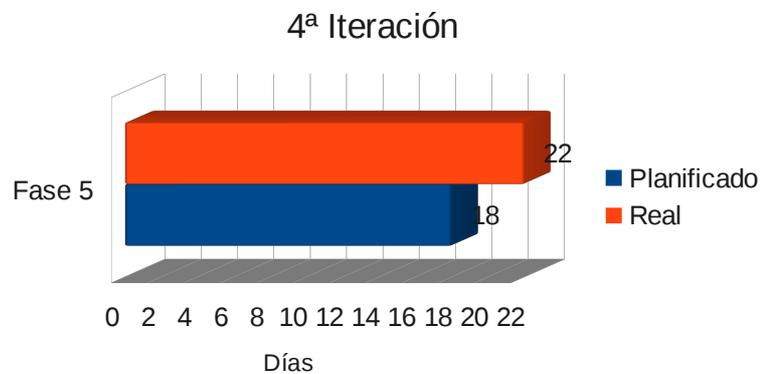


Figura 9.1.5.1 Seguimiento Fase 5

La parte de análisis y diseño, doblo el tiempo planificado debido a que se necesitó buscar una forma de llevar a cabo los casos de uso de esta fase. Al final se utilizaron unas nuevas librerías, las WMI, y el análisis y el diseño se tuvieron que llevar a cabo en base a ellas. Como se puede apreciar en las figuras 9.1.5.2 y 9.1.5.3.

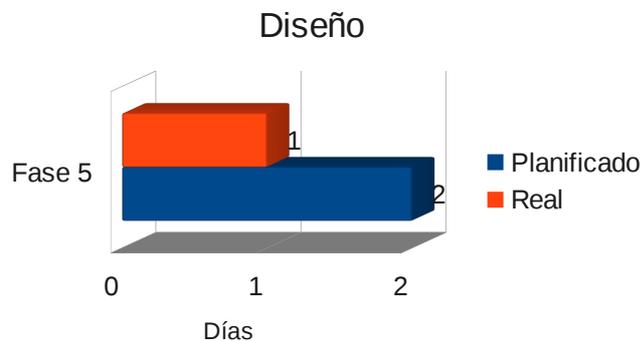


Figura 9.1.5.2 Diseño Fase 5

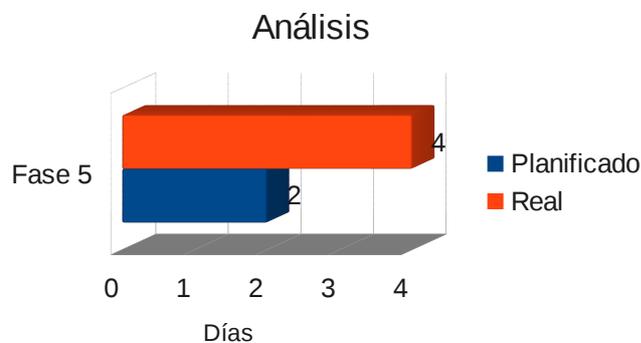


Figura 9.1.5.3 Análisis Fase 5

Por la misma razón que aumentó el esfuerzo dedicado a el análisis y el diseño lo hizo también para la implementación y pruebas. Aunque, como muestra la figura 9.1.5.4, estas últimas de una manera menos acusada.

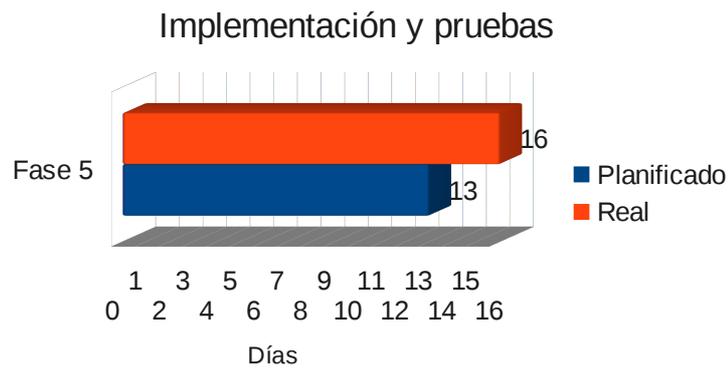


Figura 9.1.5.4 Implementación y pruebas Fase 5

### 9.1.6 Incidencias en la Fase 6:

Esta fase discurrió según lo estimado como muestra la figura 9.1.6.1.

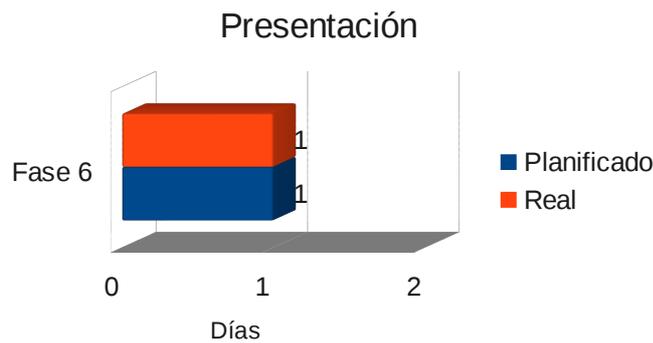


Figura 9.1.6.1 Seguimiento Fase 6

## 9.2 Horas totales

Una vez contabilizadas todas las horas que se han dedicado a desarrollar el proyecto, he llegado a la conclusión de que se han superado las horas planificadas en 48. Lo que supone un incremento del 11%. Esto es apreciable en la figura 9.2.1.

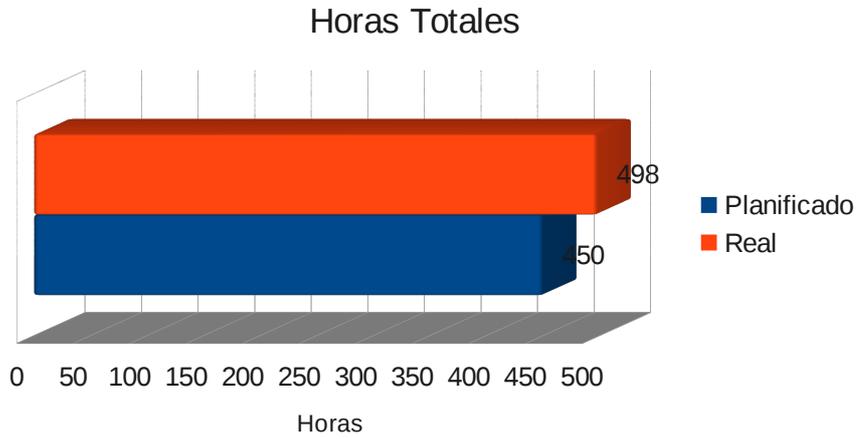


Figura 9.2.1 Horas totales

En cuanto a los esfuerzos por tipo de horas, como se puede apreciar en la figura 9.2.1, las Operativas representan un 64%, es decir más que la suma de las tácticas y las Formativas. Esto se debe principalmente a la implementación y las pruebas que, al final, han resultado la parte que ha requerido más esfuerzo. En cuanto a las Horas Tácticas y las formativas conllevaron un esfuerzo parecido, 16% y 10% de las horas respectivamente, puesto que las tecnologías en las que está basado el proyecto se desconocían antes de empezar el proyecto.

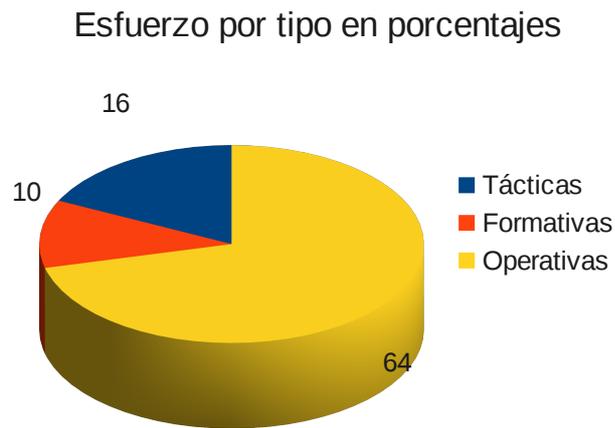


Figura 9.2.1 Horas totales

### 9.3 Gantt Planificado VS Gantt Real

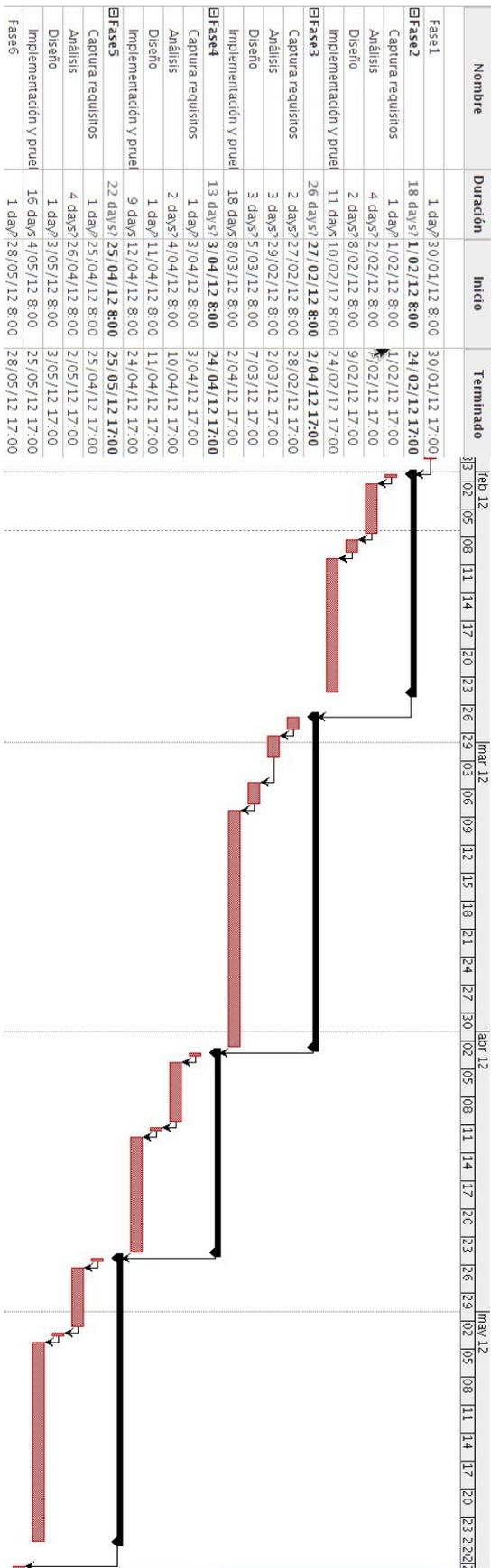


Figura 9.3.2 Diagrama de Gantt real

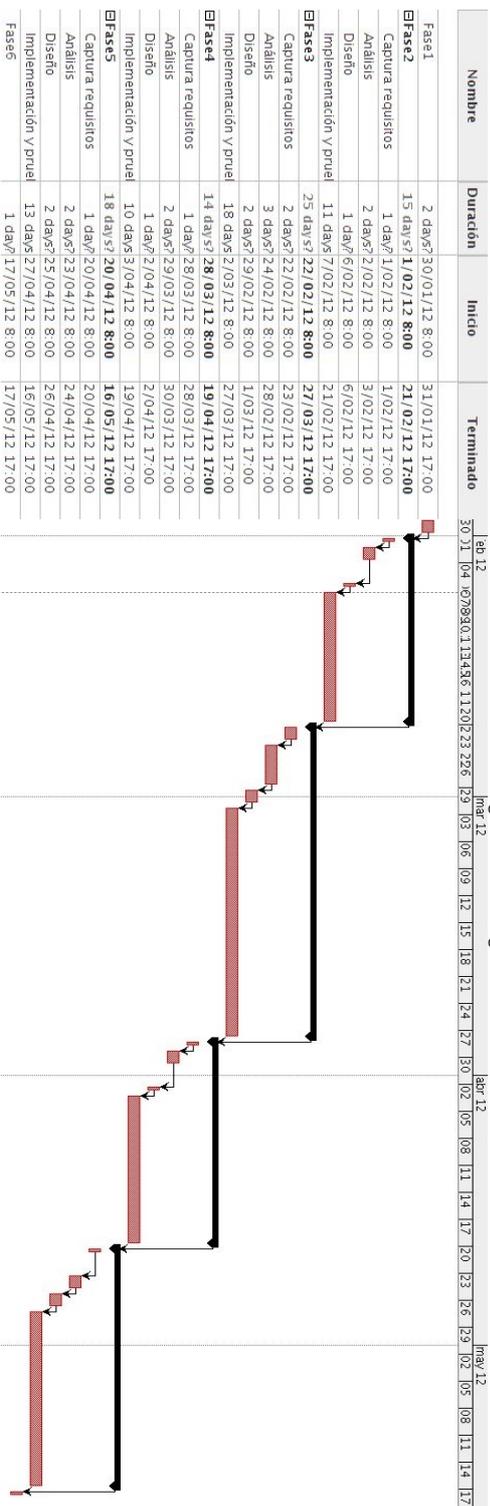


Figura 9.3.1 Diagrama de Gantt planificado



## 10 CONCLUSIONES

En este apartado se van a describir las conclusiones a las que he llegado una vez finalizado este proyecto.

Analizando los resultados, estimo que este proyecto se ha llevado a cabo de una forma razonable, consiguiendo crear una aplicación estable y que cumple con todos los objetivos establecidos al inicio. Incluso superándolos a medida que se le añadieron funcionalidades a la aplicación. Además, de cara a futuro, la aplicación es fácilmente modificable, ya sea para añadirle nuevos casos de uso como para modificar los que ya posee, algo que se tuvo en cuenta desde el inicio del proyecto.

En cuanto a la aplicación, de cara a futuras evoluciones hay dos aspectos a destacar. Por un lado, añadirle nuevas funcionalidades es muy sencillo, ya que desde el diseño se tuvo en cuenta el que fuese fácil de ampliar. Aunque, por otro, si se quiere adaptar para utilizar otras versiones de Windows Server y Exchange las cosas se complican. Toda la parte que no afecta al correo no tiene porque ser difícil de adaptar, ya que de unas versiones de Windows Server a otras los cambios afectan de manera limitada a las funciones de la aplicación, pero si se cambia la versión de Exchange o se pasa a una versión Server de 64 bits, las librerías con las que se gestionan los buzones de correo deberían cambiar, obligando a cambiar las funciones que crean, eliminan o mueven los buzones de correo.

Por otro lado, algo que hasta ahora no había vivido es que al llevar a producción la aplicación hay que tener mucho cuidado, puesto que se han de tener en cuenta las consecuencias que puede tener un fallo manipulando aspectos críticos de la red. De hecho los cambios, sobre todo los realizados sobre el Directorio Activo de la empresa, podían afectar negativamente a otros trabajadores, eliminando accidentalmente una cuenta de usuario por ejemplo.

Otra conclusión que he obtenido es el de la importancia de los planes de contingencia, ya que un par de veces tuve problemas con las máquinas virtuales y el poder echar mano de las copias de seguridad me evitó tener que realizar un trabajo extra.

### 10.1 Plano personal

Uno de las conclusiones más claras que he sacado, es la importancia de un buen análisis y diseño. Como he comentado en esta memoria, se realizaron más casos de uso de los inicialmente contemplados y de haber tenido un diseño rígido, estos cambios hubiesen supuesto un verdadero problema.

Por último, también se podría destacar que la realización del proyecto ha sido muy gratificante ya que es algo que se está utilizando, qué va a resultar útil para la empresa y ha logrado que adquiriera una perspectiva de lo que supone un trabajo de este tipo.



## 11 BIBLIOGRAFÍA

La bibliografía consultada ha sido toda a través de Internet:

Página oficial de la MSDN Library de Microsoft.

<http://msdn.microsoft.com/library/default.asp>

Página de soporte de productos Microsoft

<http://support.microsoft.com>

Página en la que consultar libros

<http://my.safaribooksonline.com>

En la que se consultó:

Joe Richards, Robbie Allen & Alistair G. Lowe-Norris. Design and Deployment of Microsoft Active Directory. O'Reilly, 2006

Enciclopedia on line

<http://es.wikipedia.org>

Páginas de consultas de código:

Generales:

<http://www.codeproject.com/>

<http://stackoverflow.com/>

Dedicadas a Exchange:

<http://www.msexchange.org>

<http://gsexdev.blogspot.com.es/>

Dedicadas a Servidores:

<http://serverfault.com/>

Páginas de consultas de errores:

<http://www.pcreview.co.uk>

<http://eventid.net/>



## 12 ANEXOS

### 12.1 Manual de Instalación:

#### 12.1.1 Introducción

A continuación se detallan los pasos que se deben dar para poder utilizar la aplicación Admintool en un equipo Windows XP o Windows 7 de una red en la que se utilice el Directorio Activo sobre servidores Windows Server 2003.

#### 12.1.2 Requisitos previos

Con el fin de que se puedan utilizar todas las funcionalidades de la aplicación, el equipo cliente deberá tener instalados e iniciados los servicios de IIS y WMI (De no estar instalados se deberá instalar en la opción de añadir componentes de Windows en la sección de desinstalación de programas del Panel de control).

Con el fin de que se pueda gestionar la parte Exchange de la aplicación hacen falta dos cosas. Uno, tener un cliente de correo MAPI instalado, por ejemplo Outlook. Y dos, tener instaladas las librerías CDO.dll y cdexm.dll en el equipo.

La instalación de las librerías necesarias para gestionar Exchange se realizarán de forma distinta si se quiere instalar Admintool en un equipo Windows XP o un equipo Windows 7.

*Windows XP:*

Para el Windows XP, se debe instalar el Windows Server Administrators tools pack, que instalará las librerías necesarias para poder realizar cambios en un servidor windows server 2003 en remoto. Y el Exchange System Manager que se puede encontrar en el CD de Exchange 2003, para que instale la librería cdoexm.dll que permitirá administrar los buzones en remoto.

*Windows 7:*

Para Windows 7 se debe instalar primero el Remote Server Administration Tools for Windows 7

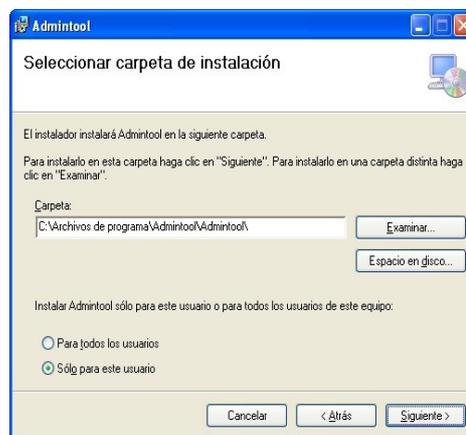
que permite realizar cambios en remoto al servidor y posteriormente se debe instalar el Exchange System Manager, pero la versión para Windows Vista, e instalarlo desde la consola, escribiendo ESMVISTA.MSI /q una vez situados en la carpeta donde esté esmvista.msi.

### 12.1.3 Instalación de AdminToolSetup.msi

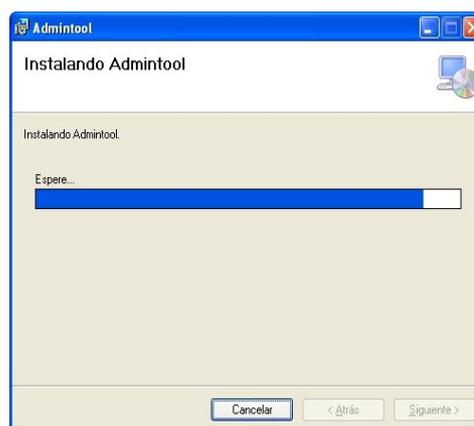
Se instala, como cualquier programa en el Sistema Operativo Windows, haciendo doble clic sobre el archivo AdmintoolSetup.



Ventana de inicio de la instalación:



Selección de la dirección de la carpeta de instalación:



Copia de archivos en la carpeta de instalación:

Una vez instalado, se debe copiar el archivo de configuración (en el que se incluye el nombre y el password del controlador de dominio) con el nombre Conf.txt en la carpeta Conf que se ha creado en la carpeta de instalación de Admintool, por defecto su dirección será C:\Archivos de programa\Admintool\Admintool\Conf. Por otro lado, en la carpeta Log de la carpeta de instalación se creará el log del uso de la

aplicación, cuyo nombre será log y entre paréntesis la fecha de creación, puesto que se creará uno por día.