PROYECTO FINAL DE INGENIERÍA

AUTOMATIZACIÓN ACTIVE DIRECTORY/LDAP

López Baraldini, Luis Martín - LU: 1013229

Ingeniería en Informática

Casanova, German - LU: 1015967

Ingeniería en Informática

Tutor:

Mutti Andrés – UADE

Marzo 3, 2016

UNIVERSIDAD ARGENTINA DE LA EMPRESA

FACULTAD DE INGENIERÍA Y CIENCIAS EXACTAS

Agradecimiento

A nuestros padres, hermanos y a toda las personas que nos apoyaron y dieron fuerzas durante el desarrollo del trabajo, para aquellos que se alegran que se haya finalizado y a las comunidades de software libre, ya que sin ellas, este trabajo no se hubiera podido realizar.

Resumen

El presente documento tiene como objetivo: la descripción de un Sistema que realizará, la automatización y administración de permisos y usuarios de Active Directory o LDAP de una empresa/compañía, por medio de una interfaz Web; facilitando las tareas a los administradores de IT.

La herramienta desarrollada se encuentra enfocada en la estandarización y automatización de procedimientos de trabajo, independientemente de la plataforma que lo implemente. La aplicación permitirá reducir los tiempos de implementación, minimizar capacitaciones y reducir costos, con la idea de mantener al usuario final informado ante cualquier inconveniente o pedido de permisos que el mismo realice.

ABSTRACT

This document aims, automation and rights management and users from Active Directory or LDAP for a business/company, through a Web interface; facilitating the tasks of IT administrators.

With this tool we are focused on the standardization and automation of work processes regardless of the platform that implements it. This application will reduce the time of implementation, minimize training and reduce costs, with the idea of keeping the end user comfortable for any inconvenience or request permission to do the same.

Tabla de Contenidos

1. INTRODUCCIÓN

2. OBJETIVOS

2.1 General

Construcción de una herramienta de software para automatizar distintos procesos de Active Directory/LDAP (Lightweight Directory Access Protocol).

2.2 Específicos

Investigación, análisis y diseño de la aplicación.

Construcción del prototipo que soporte cada uno de los siguientes procesos:

- Unificar interfaz de ABM de usuario.
- Unificar interfaz de ABM de Grupos.
- Manejo de WORFLOW de aprobación.
- Envíos de notificaciones.
- Permita realizar la exportación de sus configuraciones.
- Automatización de las tareas de Active Directory.
- Generar transacciones de pedidos de cambios.

3. CONCEPTOS GENERALES

En esta sección se abarcaran los conceptos técnicos que fueron utilizados en el desarrollo analítico y que no son de uso cotidiano. Si bien no se ahondará demasiado en cada uno de ellos, es necesario que algunos de sus aspectos queden claro, ya que serán el soporte en varias las conclusiones.

Active Directory (AD)

Esta tecnología es utilizada por los servidores de Windows para centralizar la administración y almacenamiento de las políticas de permisos sobre una red. Esto implica que todos los usuarios que quieran utilizar esta red, tienen que registrarse en el servidor y a medida que obtenga los permisos necesarios podrán utilizar los recursos de la misma.

Para poder almacenar estas políticas de permisos, utiliza una base de datos jerarquizada de objetos, de tal manera que el acceso de esta información pueda ser eficiente. Para esto Microsoft decidió seguir el estándar .X500 que además le facilita la compatibilidad con protocolos como el LDAP.

Los objetos que almacena son:

- Servicios (Impresión, mail, internet, etc.)
- Recursos (Impresoras, escáner, etc.)
- Entidades (Usuarios, Grupos, OU, etc.)

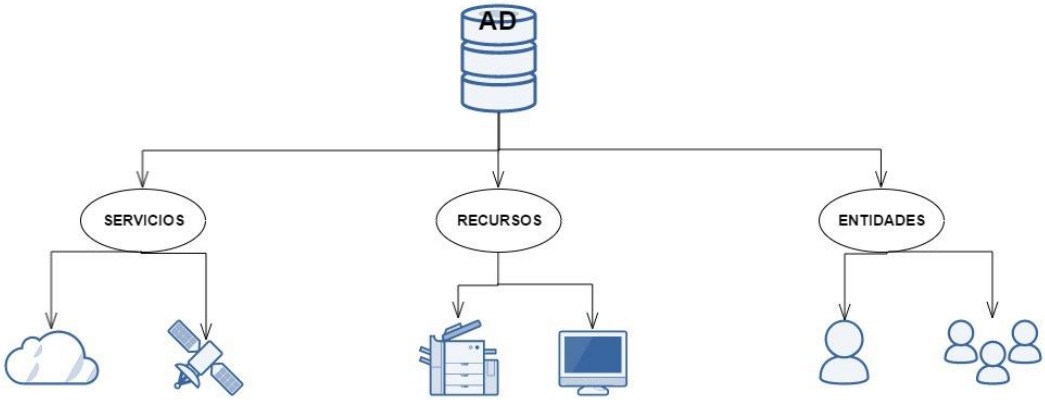


Figura 3.1: Objetos Estructura de Active Directory/LDAP

Posee un motor de búsqueda que permite realizar operaciones sobre los objetos almacenados de forma absoluta o de manera relativa.

Para obtener un objeto de manera absoluta, se requiere conocer el "nombre_distinguido" del mismo (DN). Que se relaciona unívocamente, debido a que la base tiene una restricción de

clave primaria. Este tipo de operación no tiene mayor impacto, ya que se puede determinar la ruta específica donde se encuentra la información.

El DN se compone de la siguiente forma:

- Esquema (indica en qué partición se encuentra el objeto)
- Objetos contenedores (Unidades Organizativas)
- Nombre absoluto del objeto (En casos de los usuarios se componen de: el nombre y el apellido)

```
DC=CN=Seguridad,OU=Sistemas,DC=empresa,DC=com
```

Code

Al realizar una búsqueda relativa se requiere conocer otro aspecto de los objetos, como por ejemplo los atributos. Aquí es donde subyace la información adicional del objeto y por lo tanto la que le proporciona sus características particulares, por lo que podemos inferir que la cantidad que posea una entidad no será igual a la de un recurso. La ventaja de utilizar el motor es que permite localizar los objetos a través de sus atributos y facilitar las búsquedas. Es importante remarcar que los escaneos de la base son más intensivos, por lo tanto es necesario tomarlos en consideración al momento de dimensionar el hardware del equipo.

Componentes de Active directory

Para poder llevar a cabo la administración de una red de gran escala se desarrollaron varios componentes que permitirán establecer una estructura lógica de todos sus recursos.

- Unidades organizativas
- Dominio
- Árboles
- Bosques

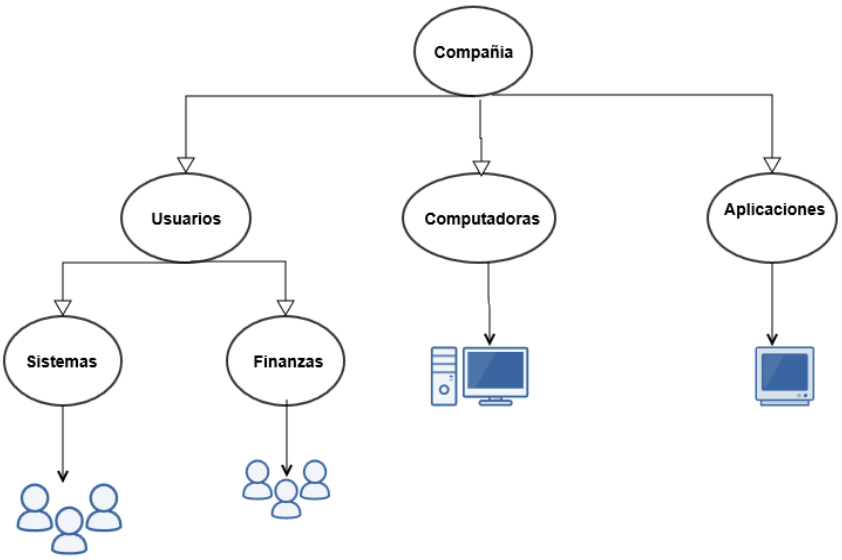


Figura 3.2: Componentes de Active Directory/LDAP

A su vez la estructura física de una organización está compuesta por:

- Red Lan (hubs, proxy, switch)
- Servidores (de dominio, de servicios).

Unidades organizativas (OU)

Las unidades organizativas, son objetos de la base como por ejemplo: los recursos, servicio y entidades. La característica particular que tienen, es que son objetos "contenedores" que pueden agrupar varios objetos. De esta manera cuando se realizan acciones que se aplican a las OU, se verán reflejados en todos los objetos que contenga.

Se puede realizar jerarquías de OU, lo que le permite generar permisos sectoriales como así también restringir accesos a ciertos niveles de la misma. Por otra parte se puede establecer OU utilizando nombres propios de los sectores de las empresas, lo que hará más fácil la comprensión de la estructura de la base.

La imagen N° 3.3 se podrá observar el recuadro las OU que tiene el dominio.

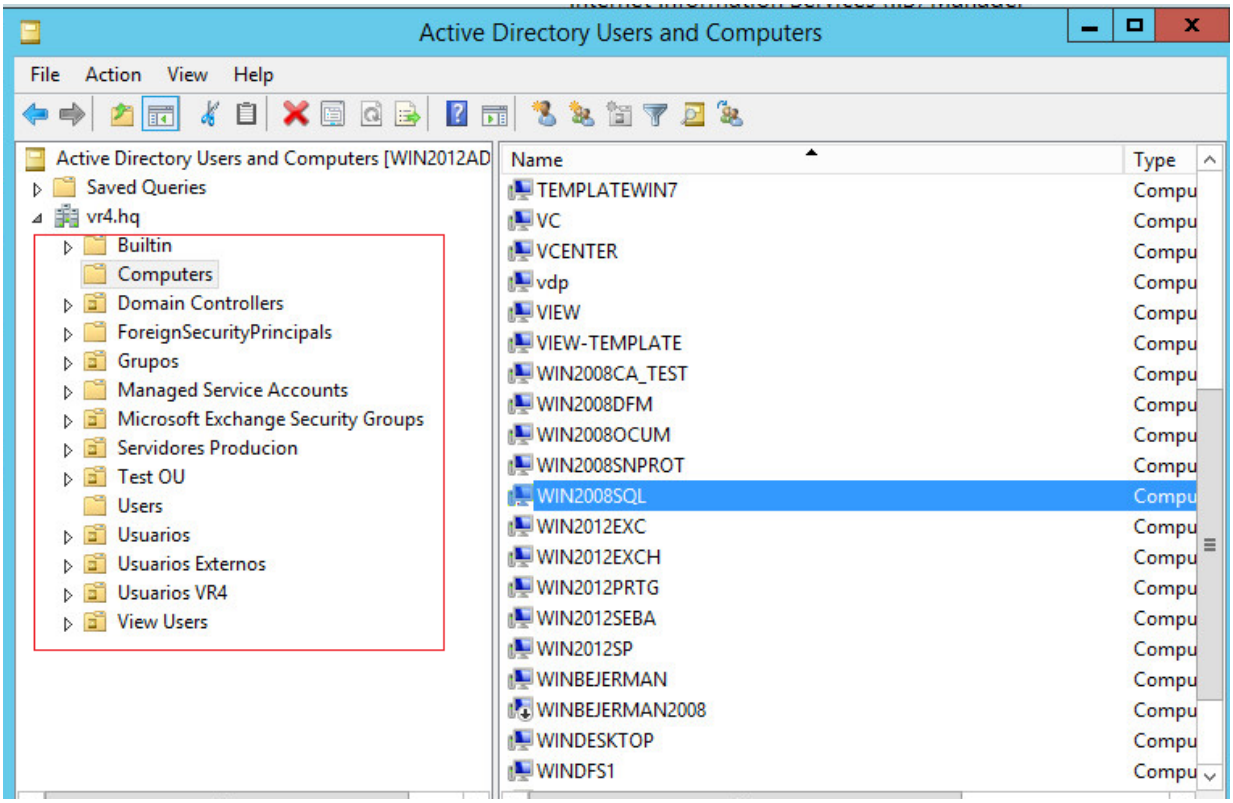


Figura 3.3: Unidad Organizativas en Active Directory. (Empresa VR4)

Dominios

Los dominios se pueden ver como separadores, es decir que son utilizados para contener y segmentar grupos de objetos que tienen un propósito en común dentro del mismo. Los AD pueden contener a su vez varios dominios que pueden ser creados a partir de diferentes criterios, los más comunes son:

- Por sector (ventas.empresa.com)
- Por producto (autos.empresa.com)

Cabe destacar que se puede tener inclusive una jerarquía de dominios como por ejemplo: ventas.empresa.com que es un subdominio de empresa.com. Esto permite compartir recursos, servicios entre diferentes subdominios o restringir el uso solo para un sector.

Todos los objetos se guardan en un mismo esquema dentro de la base de AD, pero los dominios establecen una frontera de seguridad, por la cual los objetos que se encuentran dentro se pueden relacionar entre sí. Según la recomendación de diseñadores de MICROSOFT, en los casos en que la empresa se encuentra distribuida por el mundo, las mismas deberán contener varios dominios, incluso en servidores de AD distintos, para evitar problemas de performance. Si fuese necesario relacionar objetos por fuera de esta barrera de seguridad se pueden utilizar elementos como Árboles.

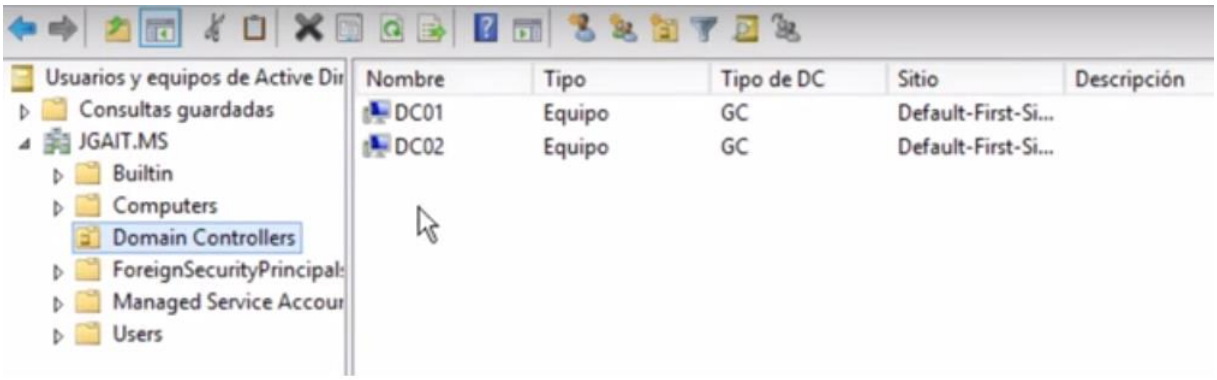


Figura 3.4: Domain Controllers registrados en Active Directory

Árboles

Cuando la configuración de la administración de la red posea más de un dominio, debido a su magnitud, existe la posibilidad de que aparezca la necesidad de que los objetos requieran interactuar inter dominios. Para esto AD tiene la capacidad de crear árboles que permite generar relaciones de confianza entre los dominios.

En este sentido aquellos dominios, que se encuentren dentro de un árbol, tienen la posibilidad de interactuar entre sí, dependiendo de cómo se establezca las relaciones de confianza, es decir, crea un nuevo límite de seguridad entre los objetos que componen todo conjunto.

En cuanto a las relaciones de confianza que se establecen entre los árboles, que pueden encontrarse dentro de un mismo AD o no, se puede realizar las autenticaciones y autorizaciones a través del protocolo de Kerberos.

Se pueden generar jerarquías entre los dominios relacionados, lo cual permite que se hereden los permisos que se otorgan en los dominios de niveles superiores.

Es necesario entender que al igual que en los dominios particulares, el motor realizar búsquedas absolutas y relativas. Esto permite buscar por más de un dominio.

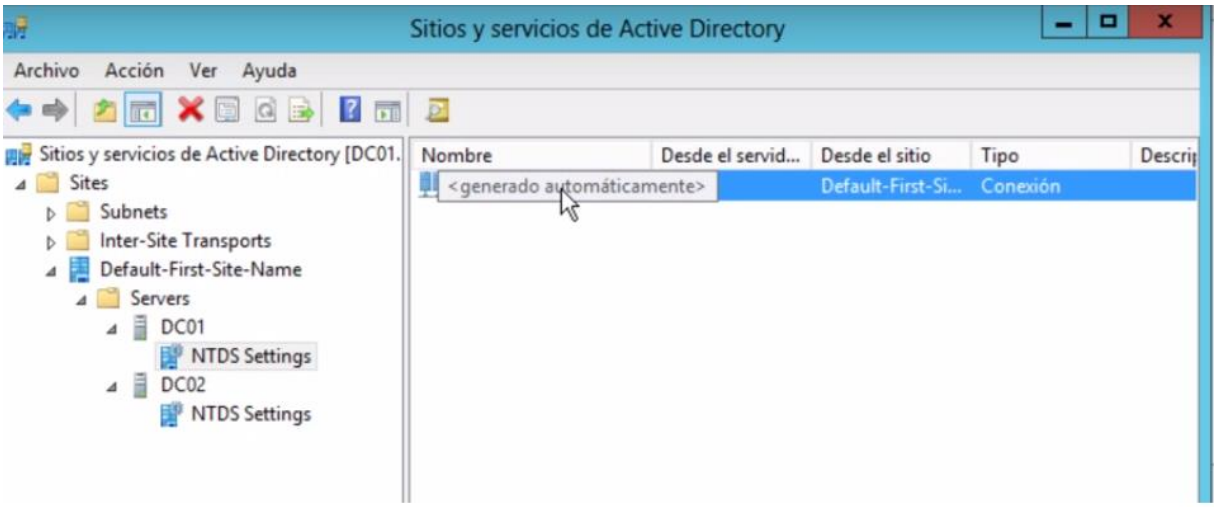


Figura 3.5: Sitios y Servicios de Active Directory. Configuración DNS

Bosques

Un bosque está formado por grupos jerárquicos de uno o más árboles de distintos dominios y completamente independientes entre sí. Lo que indica que, las estructuras de los árboles que lo componen pueden poseer estructuras jerárquicas y de nombres completamente distintas.

Cabe destacar que todo el bosque posee un esquema común, permitiendo que los integrantes puedan seguir realizando búsquedas absolutas o relativas de totalidad de los objetos.

Al igual que los dominios mencionados anteriormente, los árboles, generan relaciones de confianza a través del protocolo kerberos, esto permite mantener la seguridad de la red.

Cuentan con la característica particular de generar relaciones de confianza transitivas, esto quiere decir si el árbol A confía en el B y el B confía en el C, permitiendo que A confíe en C.

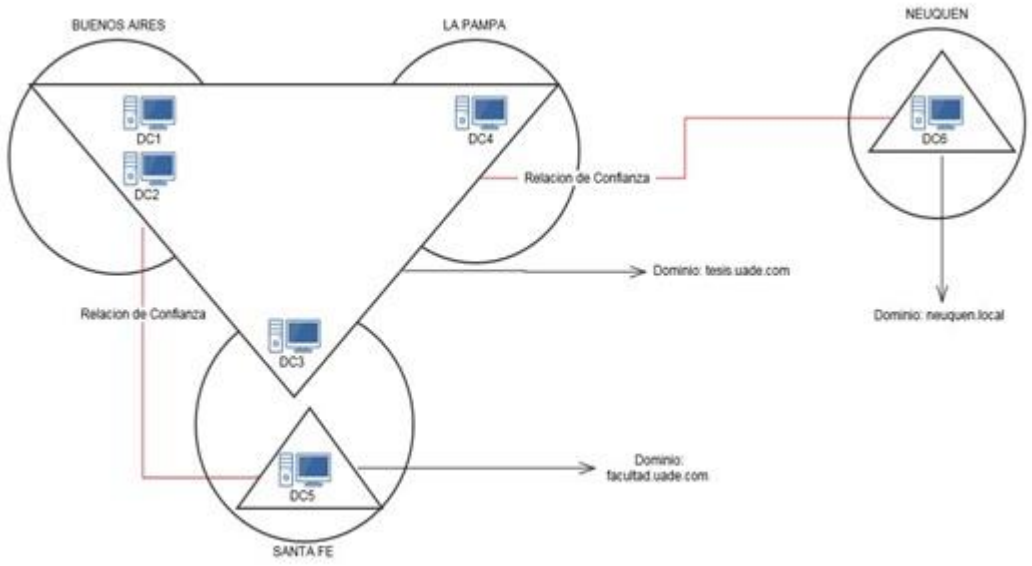


Figura 3.6: Modelo arquitectónico de bosque en Active Directory

LDAP v3

LDAP (Protocolo Ligero/Simplificado de Acceso a Directorios) sirve para almacenar y administrar directorios. Este protocolo está basado en .X500 que le permite compatibilizar con varios sistemas operativos. Se utiliza para poder registrar dominios en sistemas operativos unix y windows. Permite realizar búsquedas absolutas y relativas de sus directorios con una gran rapidez.

También interactúa a través del protocolo TCP/IP y en su v3 permite realizar estas acciones sobre conexiones seguras con TLS. Esto da el poder acceder fuera del servidor sin exponer nuestras credenciales en la red.

Una ventaja es que permite centralizar el punto de acceso a la información para mantener la integridad y mejorar la seguridad de la misma.

Los pedidos de cambio dentro del esquema de objetos de LDAP, son recibidos por el cliente Ldap y luego enviados al servidor central, este intercepta la petición y validar que los permisos del solicitante le permitan realizar la acción y posteriormente se conecta con el esquema para realizar la operación.

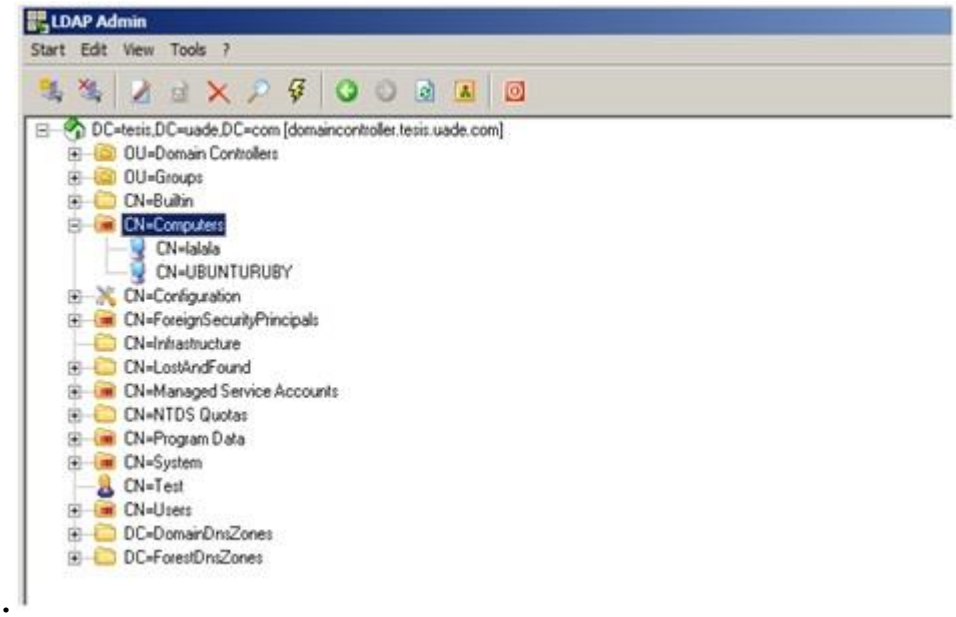


Figura 3.7: Estructura de LDAP

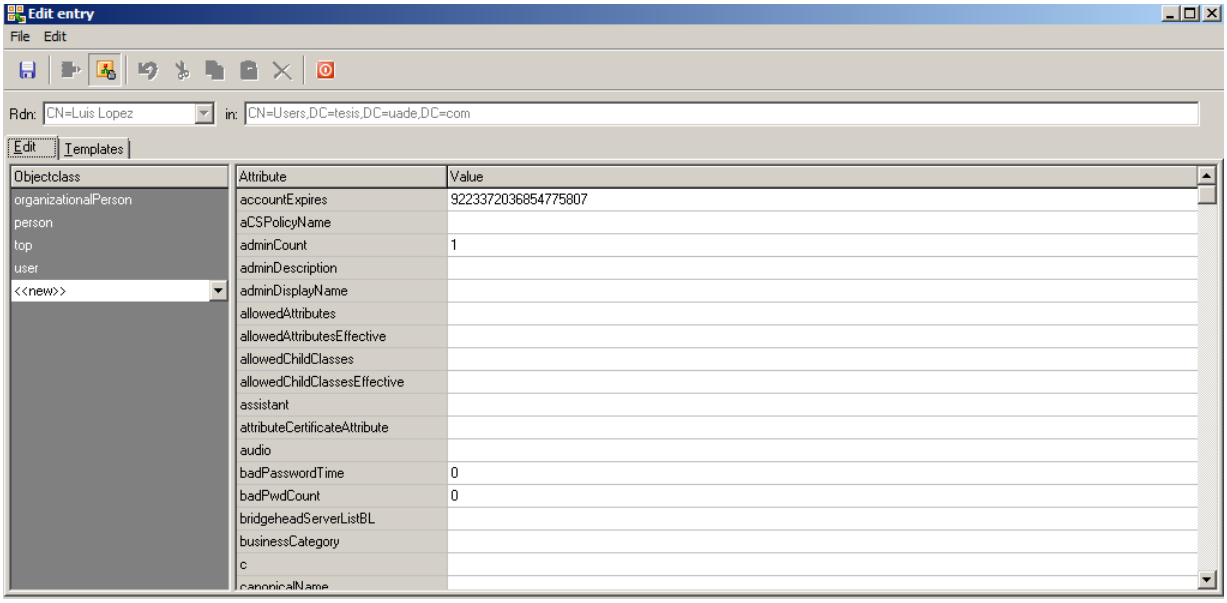


Figura 3.8: Atributos de un objeto usuario en LDAP

El componente principal de LDAP es una estructura de árbol llamada DIT (Árbol de información del directorio), se utiliza para establecer el orden de la información. Tiene el tipo de orden Jerárquico debido a que de esta forma puede establecer la ruta absoluta a todos los objetos (DN). El orden general no se puede varias y cumple con las siguientes características:

- Primero se encuentra el esquema
- Segundo se encuentra los niveles de Unidades organizativas (1 o más)
- Tercero se encuentra el nombre común de los objetos (CN)

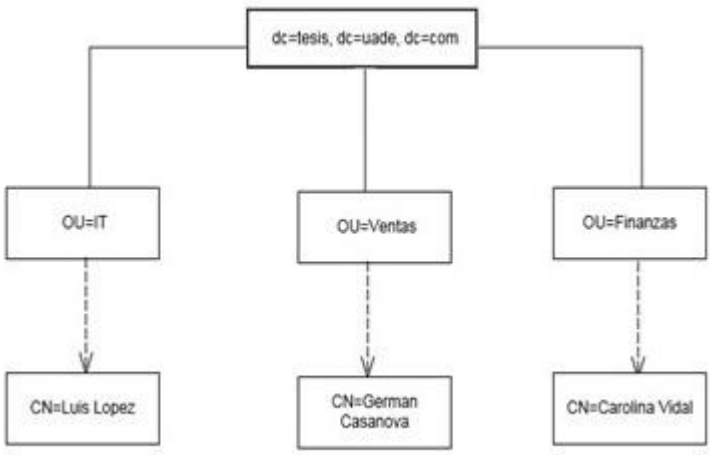


Figura 3.9: Estructura de Unidad Organizativas para AD/LDAP

Api Rest (Rest)

Roy Thomas Fielding en año 2000 en su tesis doctoral estableció un nuevo estilo de arquitectura para aplicaciones de servicios hipermedia, utiliza como protocolo de comunicación HTTP que permitirá integración con la gran mayoría de las aplicaciones creadas en la actualidad.

Para desarrollar este concepto esbozó una serie de principios que se tienen que cumplir para obtener un óptimo funcionamiento de las aplicaciones de servicios.

Utilizar explícitamente los métodos de HTTP

Las direcciones que tienen asociado los servicios, no poseen verbos sino sustantivos. Esto facilita el entendimiento de las estructuras de objetos y evita los usos ambiguos de los métodos.

- POST se utiliza para crear objetos
- GET se utiliza para obtener objetos
- PUT se utiliza para actualizar objetos
- DELETE se utiliza para eliminar objetos

Los cuatro métodos son utilizados para cualquier sistema que responda a los servicio CRUD, por lo tanto se deberá respetar el significado cuando se generan los servicios. A continuación se podrá observar un ejemplo correcto:

GET /updateuser?name=German&age=24 HTTP/1.1

Esta sentencia indica que el método es para obtener un recurso, en cambio esté, internamente realiza una actualización. Las prácticas previamente mencionadas son utilizadas para simplificar la implementación. Sin embargo son objeto de repudio cuando son analizadas, porque generan problemas de integración cuando nuevas personas desean utilizar los servicios.

No mantengas estados

Todas las invocaciones que se realicen, deben contar con la totalidad de la información necesaria para ejecutar dicha acción. En este punto es viable la flexibilidad de los lineamientos para la adaptación de la arquitectura y así verificar la factibilidad de la misma

El autor se basa en que no hay que complejizar la aplicación, con herramientas para persistir los estados de las comunicaciones, esto disminuye posibles fallas y problemas de sincronización. Se debe minimizar la transmisión de los datos sensibles, como así minimizar el uso de la red, A pesar de lo mencionados anteriormente, en su tesis doctoral estableció la plasticidad que deja en manifiesto las conveniencias para la arquitectura del sistema.

Expone tu estructura de conceptos a través de las URI

En una estructura compleja, se requiere una forma de navegarla. Este principio pretende establecer una ruta que al ser leída, se pueda interpretar que objeto es el que se quiere obtener y cómo se llegó al mismo. De esta manera se pueden establecer URI sencillas, intuitivas y fiables.

GET http://www.emperesa.org/user HTTP/1.1 (obtener todos los usuarios)

GET http://www.emperesa.org/group?name=QA HTTP/1.1 (obtener el grupo QA)

GET http://www.emperesa.org/user/group?name=QA HTTP/1.1 (obtener todos los usuarios del grupo QA)

Reglas básicas para evitar inconvenientes:

- Escribir en minúsculas
- Reemplazar espacios con guiones medios o bajos
- Evitar devolver 404 (error service not found, protocolo HTTP) cuando la ruta sea parcial, enviando una página de ayuda.

4. ESTADO DEL ARTE

Ldap command

Ldap permite conectarse a través del protocolo TCP/IP, el cual brinda la posibilidad de interactuar con su esquema. Esta api admite conectarse, agregar, modificar y borrar objetos, acciones que se realizan con Active Directory en caso de redes Windows o OpenLdap en redes Unix. Estos comandos se pueden clasificar en autenticación, integración, actualización y control.

Es importante destacar que el protocolo admite dos formas de conectarse: No segura a través del puerto 389 (default que puede ser modificado) o bien, en forma segura por medio de un túnel SSL del puerto 636 (default que puede ser modificado). Para este último tipo de conexión deberá generar certificados en el servidor e importarlos en el caso se interactúe desde un servidor externo.

- ldapbind
- ldapsearch
- ldapadd
- ldapdelete
- ldapmodify
- ldapmoddn

Ldap Bind

Es un comandos utilizado para autenticar una ruta específica o para chequear que el recurso se encuentre disponible.

```
ldapbind [options]
```

Ejemplo:

```
ldapbind -h myhost -p 389 -D "cn=german" -w 1234
```

Esta ejecución se va a autenticar con las contraseña 1234 y va a verificar que el common name german exista.

Ldap Search

Se utiliza para realizar búsquedas en el esquema del servidor, se puede especificar el formato de resultado.

```
ldapsearch  [options]  filter [attributes]
```

Ejemplo:

```
ldapsearch -h myhost -p 389 -s base -b "ou=ventas,dc=empresa,dc=com" \
"objectclass=*"
```

Se busca en el servicio sobre la unidad organizativa "ventas" dentro del dominio "empresa.com". El filtro ingresado indica que todas las clases van a ser devueltas. Este ejemplo no posee los datos de autenticación.

Ldap Add

El Ldap Add agrega información en los directorios. El modo de uso es proporcionar una, clave para obtener el registro al igual que el bind, también se agrega la dirección para el archivo con formato ldif, que indica cuales son los cambios. Este archivo se lee de forma secuencial y modifica los atributos que se encuentran especificados.

```
ldapadd [options] [-f LDIF-filename]
```

Ejemplo:

```
ldapadd -h myhost -p 389 -D "cn=german" -w 1234 -f adding.ldif
```

En el ejemplo se agregan los siguientes cambios: givenName, telephone, mail, description. Para realizar la siguiente acción, el formato del archivo debe ser de la siguiente manera:

adding.ldif

```
dn: cn=german, ou=desarrollo, dc=empresa,dc=com
objectclass: top
objectclass: person
cn: german
givenName: german
telephone: 5555-5555
mail: german@empresa.com
description: Software Developer
```

Ldap Delete

El comando se utiliza para borrar entradas del directorio, para ejecutar el comando el servidor autentica el usuario y si corresponde realiza la acción.

Code

```
ldapdelete [options] "entry DN"
```

Ejemplo:

Code

```
ldapdelete -h myhost -p 389 -D "cn=german" -w 1234 \
"uid=german,ou=desarrollo,dc=empresa,dc=com"
```

La búsqueda del registro utiliza el UID que es la manera de identificar unívocamente al mismo dentro del esquema. Solicita que tenga que autenticarse para realizar esta acción siendo fundamental ingresar las credenciales.

Ldap Modify

Cuando se requiere cambiar un registro existente, corresponde utilizar este comando. Para ello se debe especificar la base a la que se conecta y la credencial de seguridad, esto se debe a que es una acción de modificación. Hay que tener en cuenta que al igual que el comando ldapadd, se debe especificar la ruta del archivo que indica los cambios se van a realizar.

Code
```
ldapmodify [options] [-f LDIF-filename]
```

Ejemplo:

Code
```
ldapmodify -h myhost -p 389 -D "cn=german" -w 1234 -f mofify.ldif
```

En este ejemplo vamos a agregar el teléfono en el registro, y vamos a reemplazar el apellido del usuario.

modify.ldif
```
dn: cn=german,ou=desarrollo,dc=empresa,dc=com

changetype: modify

add: telephonenumber

telephonenumber: (011) 4555-5555

-

replace: sn

uid: casanova
```

Ldap Mod DN

Se utiliza para cambiar la ruta de acceso a un registro. Se ejecuta generalmente cuando se debe mover una entrada entre unidades organizativas. A diferencia de los comandos anteriores, este modifica la ruta de acceso.

```
ldapmoddn [options] -b "current DN" -R "new RDN" -N "new Parent"
```

Code

Ejemplo:

```
ldapmoddn -h myhost -p 389 -D "cn=german" -w 1234 \
-b "uid=german,ou=desarrollo,dc=empresa,dc=com" \
-N "ou=qa,dc=empresa,dc=com"
```

Code

En este ejemplo podemos ver que el usuario "german" que se encuentra en la unidad organizativa, se lo traslada de la OU "desarrollo" hacia "qa".

Ldap command + Script

Para empezar a automatizar ciertos comandos, la ejecución de los mismos eran realizados por medio del Scrip-bash. Dentro de los archivos es posible establecer las contraseñas de conexión encriptada en archivos de configuración. Otro aspecto positivo es que se pueden empezar a tener manejo de excepciones dentro de los archivos.

ldapadd.sh

```
#!/bin/sh

#Source properties
. ldap.properties

#Ejecucion del comando
ldapadd -h $host -p $port -D "cn=$user" -w $pass -f $1
```

ldap.properties

```
host=myshost
port=389
user=german
pass=235123ji29jf2cjdk!=
```

De esta manera se facilita la ejecución de un comando ldap. En este caso particular, es utilizado para agregar datos en el esquema de una entrada.

Al utilizar el script, indicando la ruta del archivo ldif que vimos anteriormente, se simplifica la ejecución del comando.

Code

```
./ldapadd.sh [ldif path]
```

Ejemplo:

Code

```
./ldapadd.sh adding.ldif
```

Para personal capacitado en el uso de las consolas de los sistemas operativos este cambio puede ser muy positivo, dado que puede ahorrar tiempo en la ejecución de los comandos y con el manejo de excepciones hacer más fácil la comprensión de los errores. También se puede realizar cron (ejecución periódica) para realizar controles de disponibilidad, realizando un *parseo* de las respuestas de los comandos.

Por otro lado se debe tener en cuenta que empleados de otras áreas, se encuentran sin la capacitación necesaria para la utilización del sistema operativo a través de la consola. Dando como resultado la falta de practicidad en el mismo lo que generará grandes rechazos a la hora de realizar una implementación de este estilo.

Apache directory Ldap API

Es un framework desarrollado en Java de la fundación "The apache software foundation", permite realizar una abstracción en la conexión hacia ldap; como así también ejecutar acciones atómicas de alta, baja y modificación de los registros. Al igual que en los comandos de ldap, es sencillo realizar una conexión y empezar a trabajar.

```
LdapConnection connection = new LdapNetworkConnection( "localhost", 389 );

connection.bind( "ou=empresa, dc=com", "{crypt}wSiewPyxdEC2c" );

EntryCursor cursor = connection.search( "ou=german", "(objectclass=*)", SearchSc
ope.ONELEVEL, "*" );

while ( cursor.next() )
{
    Entry entry = cursor.get();

    // Process the entry
    ...
}

connection.unbind();

connection.close();
```

En la actualidad se ha lanzado una versión estable 1.0.0-M028 (23 marzo 2015) que posee buenas prestaciones para realizar una integración con un sistema de gestión de usuario, implementado en Java. Pero a la hora revisar la documentación, no se encuentra definida en su totalidad, por ejemplo: La sección de conexiones segura y el manejo de excepciones.

Spring LDAP

Spring es un framework basado en java que permite al igual que J2EE resolver problemas cotidianos que pueden tener los desarrolladores que trabajan con el lenguaje. Se suelen basar en las estandarizaciones del lenguaje para poder aumentar la compatibilidad con otros frameworks. En esta oportunidad vamos a hablar sobre su implementación de LDAP.

Spring LDAP es un componente desarrollado con el objetivo de poder simplificar las conexiones hacia el servidor y las acciones que queremos realizar con los registros.

Posee una serie de características que facilitarán el uso como por ejemplo:

- Posee un pool de conexiones para paralelizar procesos
- Realiza el manejo de excepciones y recupero
- Se encuentra basado en JPA para la persistencia
- Establece control de las acciones a través de Transacciones que aseguran la atomicidad de la ejecución.
- Se crea *LDAPTemplate* que facilita las integraciones con las aplicaciones que desarrolla.

Para realizar la conexión hay que definir el archivo *xml* de configuraciones de *spring*.

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:ldap="http://www.springframework.org/schema/ldap"
       xsi:schemaLocation="http://www.springframework.org/schema/beans http://ww
w.springframework.org/schema/beans/spring-beans.xsd
       http://www.springframework.org/schema/ldap http://www.springframework.org
/schema/ldap/spring-ldap.xsd">

   <ldap:context-source
          url="ldap://localhost:389"
          base="dc=example,dc=com"
          username="cn=Manager"
          password="secret" />

   <ldap:ldap-template id="ldapTemplate" />

   <bean id="personRepo" class="com.example.repo.PersonRepoImpl">
      <property name="ldapTemplate" ref="ldapTemplate" />
   </bean>
</beans>
```

PersonRepoImpl utiliza *ldapTemplate* para realizar todas las acciones contra el servidor de *ldap*. Aquí se puede visualizar la implementación del *bean* que realiza las acciones.

Code

```
package com.example.repo;
import static org.springframework.ldap.query.LdapQueryBuilder.query;

public class PersonRepoImpl implements PersonRepo {
   private LdapTemplate ldapTemplate;

   public void setLdapTemplate(LdapTemplate ldapTemplate) {
      this.ldapTemplate = ldapTemplate;
   }

   public List<String> getAllPersonNames() {
      return ldapTemplate.search(
         query().where("objectclass").is("person"),
         new AttributesMapper<String>() {
            public String mapFromAttributes(Attributes attrs)
               throws NamingException {
               return (String) attrs.get("cn").get();
            }
         });
   }
}
```

LdapTemplate acepta como input para el desarrollo *LdapQueryBuilder* que facilita las construcciones de las consultas y se puede definir el formato de la respuesta. *AttributeMapper* se define como se va a realizar el mapeo de la respuesta y que tipo de objeto se va a devolver.

Este ejemplo en particular la consulta devuelve todos los objetos del tipo “person” el atributo *common name.*

5. CASOS DE ESTUDIO

En esta sección detallarán algunas empresas que poseen importantes departamentos de tecnología. No se profundizará en los objetivos de negocio, sino en poder entender las problemáticas que conlleva poseer dichos departamentos.

CARGILL

Es una empresa multinacional que se encuentra radicada hace unos 20 años en Argentina, el core business de la empresa es de agro industria realizando importante cantidades de exportación de materia agrícola. Se dedican a la producción, transporte y comercialización de los productos.

Este tipo de industria requiere que se esté permanentemente ofreciendo soluciones integrales a los productores regionales y soporte en la logística. Para ello la empresa cuenta con 50 sucursales a lo largo de país y 4000 empleados para realizar el servicio.

Para poder concretar sus acciones, posee un equipo de tecnología que les permite mantener en línea todos los sistemas para recuento de producto y las transacciones digitales. Particularmente el equipo de IT no solo da soporte a puntos de acopio de grano del país, sino que también provee servicios en la mayoría de las sucursales en todo el mundo.

El departamento de tecnología tiene la capacidad de resolver incidentes en la gestión de usuarios y permisos, en este caso en particular de resolver los de su banda horaria. Para llevar a cabo esta tarea posee un equipo de 20 analistas que tiene como principal objetivo la gestión de permisos de los usuarios de la empresa. Esto comprende desde el alta de usuario, hasta la baja por desvinculación del sistema.

Todos los tickets de solicitud cambio que posteriormente se derivan en una o más tareas para el equipo, son cargados a través de un sistema llamado Remedy. Este no solo se utiliza para la canalización del ticket y verificar el funcionamiento del área, sino que también permite cargar proyecto para la evolución del sistema que soporta las operaciones.

Hay que considerar que todos los tickets cargado tienen asociado un SLA (Acuerdo de nivel de servicio) el cual determina ciertas características del ticket, como por ejemplo.

- Tipo de validaciones que realizar
- Prioridad que posee
- Tiempo de resolución tiene
- Como debe ser comunicada la resolución.

Poseen un flujo de trabajo representado por un listado de tareas que tienen que realizar a medida que aparecen los requerimientos, pero cuando ocurre un evento de mayor prioridad, automáticamente se posiciona por encima de los requerimientos anteriores (léase una desvinculación). Estos requerimientos especiales provocan, que los objetivos del día se vean perjudicados generando demoras en los demás requerimientos.

Procedimientos

El equipo de trabajo se compone con un supervisor y un grupo de analistas que implementan las tareas. El líder del equipo tiene la responsabilidad de estar revisando permanentemente el listado de ticket para poder controlar el flujo de trabajo del equipo. Los analistas cuando el supervisor le asigna un ticket tiene que revisar las características del mismo, esto determina que tareas debe realizar. En general los tickets poseen tres tareas principales:

- Validación de autorizaciones y permisos.
- Ejecución del cambio o acción.
- Informe de resolución al supervisor e interesados.

Las validaciones comprenden: desde el pedido de autorización al personal indicando hasta verificar que la acción sea correcta.

Ejemplo:

Hay ciertos usuarios que no poseen la autorización de tener accesos a ciertos lugares. Estas autorizaciones se hacen vía mail y las acciones quedan en espera hasta la respuesta de los correos.

La ejecución de la acciones en los casos que hay cambios de permisos, se debe verificar que estos existan y si los usuarios disponen dentro de su dominio la autorización para el cambio a los permisos solicitados. Existen casos puntuales de copia de permisos de otros usuarios o quita de permisos específicos por cambio de rol dentro de la empresa.

El aviso de finalización de ticket puede ser por dos hitos principales: Realizado con éxito o la Denegación.

El rechazo se debe generalmente a:

- Petición realizada de manera indebida.
- Denegación del pedido de cambio.
- El cambio no se puede realizar por limitaciones de permisos. Este punto es esencial por la visibilidad del estado del pedido. Los usuarios pueden reclamar o hacer uso del nuevo cambio realizado.

El perfil de los analistas debe ser de personas prolijas y ordenadas, debido a que generalmente tienen más de una tarea en proceso.

Los trabajos generalmente se bloquean en las peticiones de validación, generando esperas innecesarias pero que pueden ser minimizadas.

Otro aspecto del trabajo de la administración de permisos y accesos, son los pedidos de cambios masivos o acciones (bulk). Se pueden pedir el mismo cambio de varias personas de un sector en simultáneo, debido a que se agregan nuevos servicios en la empresa y se requiere su uso. Al no poseer automatización de los procesos se deben pedir las validaciones por cada uno de los pedidos de manera individual y esto genera demoras excesivas.

MERCADO LIBRE

Es una empresa que opera en la región de Latinoamérica principalmente, Estados Unidos y Portugal. Posee 132 millones de usuario que se encuentran realizando transacciones con el sistema, siendo la principal compañía de comercio electrónico de Latinoamérica.

El personal de la empresa es de 2600 empleados entre los diferentes departamentos, teniendo equipos descentralizados para realizar las operaciones de mantenimientos en las diferentes locaciones que se encuentra radicada.

El equipo de IT está principalmente en Argentina teniendo a cargo las operaciones de modificación de permisos y accesos a los recursos de la empresa. Estas acciones están compuestas principalmente por Altas, Modificaciones y Baja de permisos siendo en algunas ocasiones estos accesos son temporales.

La empresa utiliza proveedores de servicio de contenido para el sitio web. Para el trabajo cotidiano se requieren acceso temporales para las personas externas a la compañía, hay que tener en cuenta que el alta a los servicios tiene que ser a través de AD. Por este motivo hay que solicitar accesos con anticipación, sino la persona no va a poder acceder a los sistemas. Ahora bien, cuando los trabajos no pueden ser anticipados por la dinámica del negocio, se requiere solicitar accesos con mayor prioridad y no siempre pueden ser otorgados a tiempo.

Actualmente (Enero de 2015) se está trabajando sobre la automatización de los pedidos. Este trabajo consta en la creación de script por tipo de acción que se realiza en el área, ejemplo de script por: Alta, Baja, Modificación de permisos. El lenguaje que se utilizó para la programación fue Ruby, por la afinidad de los integrantes del sector.

El componente desarrollado en Ruby

Se utiliza a través de la consola, esto quiere decir que no posee interfaz gráfica desktop ni web. Ofrece un menú con las acciones más comunes que realiza la empresa ya mencionadas. Para lograr este objetivo se desarrollaron varios métodos que se ocupan de cada una de estas acciones.

Alta de usuario se elige la opción indicada por el menú principal y el sistema paso a paso va solicitando los datos necesarios para realizar la acción. El componente realiza la carga de archivo "*add.ldif* "que es utilizado posteriormente interactuar con LDAP dentro del AD.

Este sistema resuelve los problemas de conexión contra el servidor y también facilita la carga de datos necesarios para completar las diferentes acciones. Permite generar *template* de usuario para agilizar la carga de datos por departamento y permisos necesarios.

Permite realizar acciones *bulk* a través de un archivo *.csv* respetando un formato indicado.

6. PROBLEMÁTICA PRINCIPAL

Las problemáticas son de las más diversas, debido a que se requiere automatizar procesos que pueden llegar a ser muy complejos sobre tecnologías que no están preparadas para este tipo de acciones.

Se pueden observar los siguientes puntos:

- El conocimiento que se requiere para operar los sistemas tiene que ser alto, debido que los sistemas de accesos pueden ser muy complejos y otorgar los permisos correctos puede ser una tarea difícil.
- La experiencia que de los analistas tiene que ser alta, esto se debe a que cuando uno trabaja con permisos puede presentarse problemas no estrictamente de permisos y solo el personal experimentado suele ser eficiente realizando estas operaciones.
- La relación entre usuarios del sistema analista suele ser de 100 a 1, esto quiere decir que a mayor volumen de trabajo indefectiblemente requiere más personal. El equipo tiene que crecer y decrecer con el volumen de peticiones. Si bien esta relación puede ser menor cuando el volumen es más alto, también es mayor cuando tenemos poco volumen.
- Poseer sucursales distribuidas con diferentes husos horarios, implica que haya un fraccionamiento del soporte de las acciones y poder tener una mayor coordinación entre equipos.
- Cuando los procesos requieren de varias aprobaciones esto queda atado al seguimiento del analista, esto puede generar demoras en las respuestas y/o que a las acciones no se realicen adecuadamente el seguimiento.
- Los tiempos de desarrollo no se pueden paralelizar por analista, esto quiere decir que el recurso tiene que resolver la acción en curso para tomar un siguiente ticket. La experiencia de realizar más de una acción al mismo tiempo, crea que se asignen mal los permisos
- Cuando los empleados de sistemas no pueden realizar una acción por falta de tiempo, esto genera pérdidas por costo de oportunidad. Puede verse claramente cuando se necesitan permisos temporales para realizar acciones contra los sistemas de las

empresas. También existen los riesgos en los casos de las bajas por desvinculación, esto implica que empleados que no pertenecen a la empresa pueden seguir interactuando con los sistemas y generar posibles pérdidas a la compañía.

- Las ejecuciones por comandos de consola pueden ser muy rápidos y efectivos, pero en la actualidad este tipo de interfaz genera rechazo en los empleados promedios que no son de área de sistemas.
- Completar formularios con estándar y forma de uso como por ejemplo los archivos **.ldif* son complejos de completar y tiene una alta posibilidad de ser cargados incorrectamente generando los siguientes problemas: error en la ejecución, carga con inconsistencias y pérdidas de datos importantes.
- Si se crea script y componente para que el usuario promedio ejecute las acciones contra los sistemas de AD, esto requiere que todos los equipos tengan accesos en los puertos del servidor. Este punto es peligroso debido a que el servidor que se encarga de establecer la seguridad esté expuesto a posibles ataques.
- Es muy costoso mantener estados de conversación y realizar manejo de excepciones cuando uno tiene que desarrollar scripts para realizar la comunicación contra AD.

.

7. SOLUCIÓN

Se realizará el desarrollo de un prototipo que permita resolver el alta, baja y modificación de permisos, grupos y usuarios de forma automática. Si bien las acciones que se realizarán sobre AD/LDAP no intervienen las personas, se debe desarrollar una interfaz que permita realizar las peticiones de cambio y sus respectivas autorizaciones.

El soporte web fue seleccionado para la construcción de la interfaz. Este permite poder usar frameworks que facilitan el uso de esta herramienta. Se puede obtener un sistema muy parecido a las redes sociales que son tan usadas y minimizar el tiempo de capacitación.

Los usuarios que interactúen con la herramienta no tienen que poseer ningún tipo de conocimiento previo de cómo se realizan las acciones dentro de los servidores. Esta capa de abstracción tiene como objetivo facilitar la interpretación del uso y acompañarlos para que puedan realizar peticiones con la menor ayuda posible.

Uno de los componentes del prototipo es el worflow de acciones, que permite mapear flujos de negocio y acciones para llevar a cabo una aprobación. Tiene la responsabilidad de mantener el estado de la petición y realizar el manejo de la comunicación, como el manejo de flujos de excepción. En conclusión cuando el flujo obtiene todas las aprobaciones positivas ejecuta la acción solicitada y comunica el resultado.

El diseño provee un aplicativo que tiene la capacidad de atender a cientos de peticiones en simultáneo las 24 horas del día, permitiendo que si se modifica el volumen de carga de trabajo no requiera de personal adicional para realizar las actividades.

Todos los componentes del desarrollo son de código libre evitando el cobro de licenciamiento por su uso.

REQUERIMIENTOS DEL SISTEMA

Visión

Dicha automatización de Active Directory/LDAP se utilizará para gestionar pedidos de altas, bajas, modificaciones, desbloqueo y reinicio de contraseña de todos los usuarios que sean necesarios a la vez. La herramienta mencionada, podrá ser usada por cualquier persona que tenga mínimos conocimientos de sistemas, ya que la interfaz es web e intuitiva.

Alcance y limitaciones

Dentro del alcance de la herramienta serán considerados los siguientes puntos:

- Altas, bajas y modificaciones de usuarios simple o en *bulk* (lote de acciones)
- Reinicio de contraseña y desbloqueo de cuentas de usuario simple o en *bulk*.
- Crear usuarios a través de *template* (formato base), esto implica que se pueden copiar los privilegios de otro usuario para facilitar la carga.
- Creación de grupos simple o en *bulk*.

Funcionamiento general del sistema

- Alta de usuario, se deberán llenar los campos referidos al usuario. En caso de tener un *template* para copiar los grupos a dicho usuario, se seleccionará el mismo y automáticamente se le copiaran todos los grupos.
- La baja de usuario se tendrá que escribir el nombre de cada usuario y automáticamente les sacara todos los grupos que tenga asignado. Esta acción implica que es un borrado lógico para mantener un histórico de los usuarios.
- El sistema tendrá la posibilidad de agregar/quitar permisos en los grupos de usuarios.
- El reinicio de contraseña se tendrá que ingresar cada usuario y automáticamente se le mandará un mail a cada uno con su nueva contraseña.
- El desbloqueo de contraseña se realizará de la misma forma que el reinicio. Se ingresa cada usuario y posteriormente se le enviará un mail por sistema indicando que su cuenta fue desbloqueada.

- Todas las acciones deben pasar por un flujo de aprobación en caso de ser requerido, esto implica que no todas las acciones son inmediatamente impactadas en el sistema de recursos

ARQUITECTURA

Para poder desarrollar la arquitectura de este prototipo es preciso entender los requerimientos de software para poder diseñar el esquema de componentes. Para ello hay que comprender que cada uno plantea problemáticas distintas.

Requerimientos de arquitectura:

- El sistema debe ser multiplataforma.
- El sistema debe ser flexible para soportar las diferentes tipos de integraciones posibles: Mobile App nativa, Web y debe permitir la integración de otros sistemas.
- El sistema debe ser seguro, la comunicación entre los componentes debe estar encriptada debido a que viajan datos sensibles.
- El sistema debe poseer unos componentes que centralice la comunicación con el servidor de recursos.
- El sistema debe minimizar las comunicaciones contra el servidor de recursos.
- El sistema debe ser escalable porque debe soportar una gran cantidad de operaciones debido a que todos los empleados de la empresa puede utilizarlo.
- El sistema debe ser usable por personas que no tengan conocimiento de tecnología.
- El sistema debe tener disponibilidad 7x24.
- El sistema debe soportar un sistema de aprobaciones para las acciones contra el servidor de recursos.

En relación con los requisitos de arquitectura ha sido detectada la necesidad de construir un componente que permita establecer comunicaciones de una manera estándar. En consecuencia la arquitectura que contiene una mayor adaptación es REST, que permite diseñar las interacción contra el sistema agnósticamente del lenguaje en que se desarrolle. La ventajas que provee este componentes son:

- Puede establecer una cache intermedia para minimizar las interacciones con el servidor de recursos y de esta manera minimizar el uso intensivo de las comunicaciones.
- Puede poseer persistencia para mantener el estado de los workflow de aprobaciones que soliciten los usuarios.
- Puede implementar servicios estándar para que cualquier otro sistema se pueda conectar
- Instalado en un servidor productivo puede soportar la disponibilidad requerida.

Para desarrollar la interfaz se puede utilizar un frameworks de trabajo livianos con ciclos de despliegue más simples. Este componente no está a cargo de mantener estados, sino que se comunica con la api rest por medio del protocolo http. Provee las siguientes ventajas:

- La interfaz puede ser construida con framework de visualizaciones utilizadas en las redes sociales que tienen una gran aceptación en general.
- La comunicación puede ser segura a través de certificados.

Sistemas operativos que soporta:

- Windows Server 2003 / 2008 / 2008 R2 / 2012 / 2012 R2.
- Windows XP/ 7 / 8.
- Ubuntu .
- Linux Red Hat.

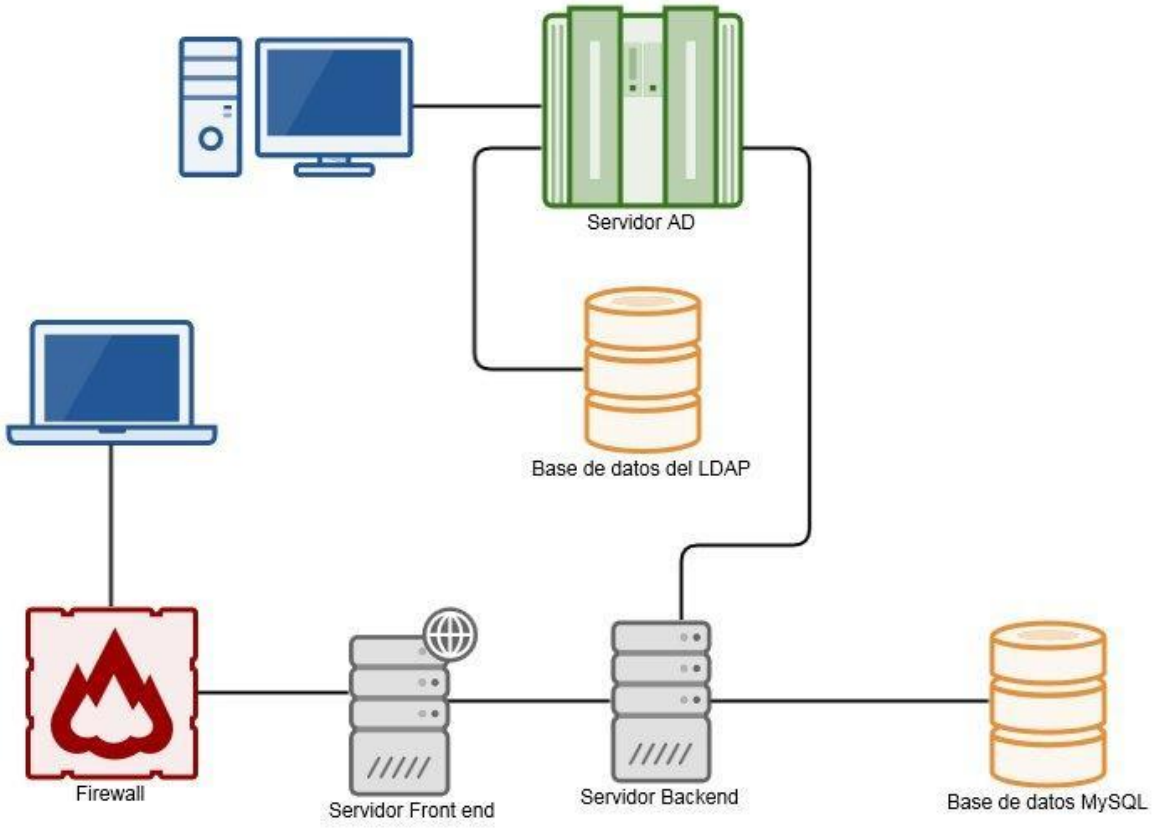


Figura 7.1: Modelo de Arquitectura del funcionamiento en Active Directory/LDAP

Selección de lenguajes

API REST

En la actualidad (2 de febrero 2014) existen muchos lenguajes que proporcionan estructuras de trabajo para realizar una api rest, como por ejemplo: Ruby, JavaScript, Java y .NET. Todos ellos de cuarta generación, poseen grandes comunidades que puede aportar soporte para la implementación de variadas problemáticas.

En primer lugar se expone el lenguaje .NET, Se tiene en consideración que para realizar desarrollos sobre esta tecnología se debe comprar las licencias de microsoft. En consecuencia

no respeta una de las motivaciones que indica que el sistema tiene que ser libre de licenciamientos.

En segundo lugar se realiza una comparación entre Ruby, JavaScript y Java. Ambos no poseen licenciamiento y son muy eficientes para realizar esta tarea. Para poder elegir entre ellos se consideran las prestaciones de cada uno estableciendo cual es el que mejor aplica.

Para comenzar analizaremos el marco de trabajo ruby-net-lap que se encuentra basado en Ruby. Con este se pueden construir servicios que facilitan la ejecución de los comando ldap y permite a su vez externalizar la configuraciones de los servidores, esto hace que los usuarios puedan ingresar unívocamente los datos pertinentes para realizar la acción deseada sin conocer cómo funcionan las conexiones y pueden recibir excepciones de ejecución parametrizadas para su mejor lectura.

Ahora bien, esta capa de abstracción de ejecución, al igual que los script previamente mencionados, realizan una acción por vez y no poseen un motor de transacciones que establezca una red de seguridad para ejecuciones concatenadas. Esto puede generar que el servidor AD/LDAP tenga incongruencias y se requiera un esfuerzo de desarrollo mayor para establecer políticasde recupero "rollback".

Ldapjs es un marco de trabajo realizado en JavaScri que al igual que Ruby-net-ldap posee las mismas ventajas de ejecución y externalización de configuraciones. Todos los navegadores de la actualidad tienen la capacidad de interpretar este lenguaje. Esta última observación es útil para distribuir el procesamiento, pero también implica que se deben realizar políticas de seguridad adicionales para evitar intrusos. Es necesario destacar que alberga la misma desventaja que mencionamos en el punto anterior, esta tecnología no poseen motor de transacciones que permita la ejecución concatenada de servicios.

Como ha sido indicado en el estado del Arte, el SpringL cuenta con las ventajas de los prototipos mencionados anteriormente. Adicionalmente tiene un motor de transacciones que determina el grabado en el servidor. De esta manera se asegura que la base de usuario no quede con inconsistencias y evita que el equipo de IT deba que realizar tareas adicionales.

En lo que respecta a la selección de lenguaje se puede concluir que, si alguna de la acciones no pudo ser realizada con éxito, Java es la única tecnología que tiene prestaciones que resuelve la problemática integralmente por lo que deberá ser utilizada para realizar este componente.

Base de datos

Una de las cuestiones que deben ser resueltas será la de motivos por los cuales se preside de ciertos datos. Aquí se encuentran los requerimientos adicionales que el servidor no contienen de manera nativa, por ejemplo la estandarización de campos necesarios como ser departamento, oficinas y posiciones de trabajo. Por otro lado, la práctica muestra que carece de sentido la utilización permanente del servidor para consultas de recursos preexistente, por lo cual ha cual ha sido primordial la realización de una cache de información que permita resolver estas peticiones sin tener que acceder al servidor AD/LDAP. Otra mejora adicional, es la centralización de la información de los usuarios para que otros departamentos puedan realizar acciones de baja por desvinculación de la empresa y poder mantener el estado del workflow de aprobaciones para realizar las acciones.

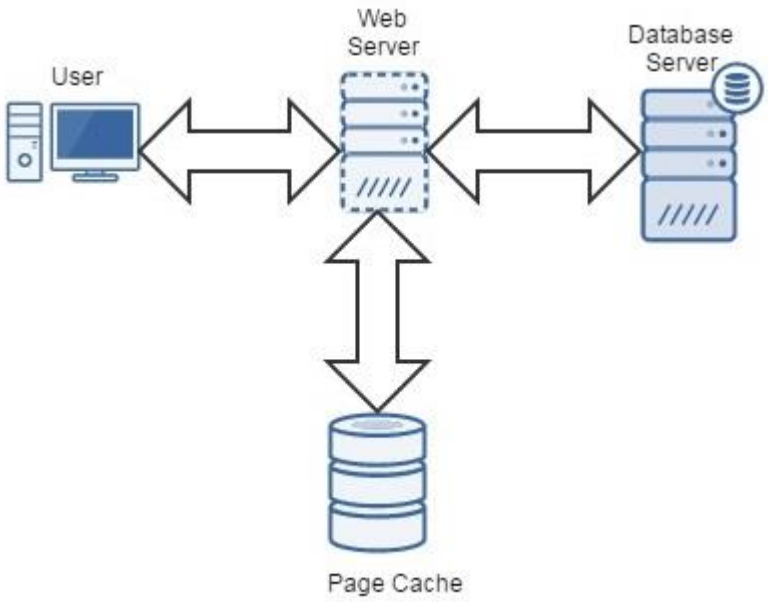


Figura 7.2: Modelo de cache del sistema implementado en base de datos.

Para comenzar el análisis se debe comprender que existen diferentes tipos de base de datos, como por ejemplo: relacionales y no relacionales.

Para las aplicaciones de alto rendimiento se utilizan generalmente bases no relacionadas con tecnologías como MongoDB y Redis, que permiten guardar documentos independientes. Pero el modelo que se utilizará en este prototipo se encuentra entre las entidades y esto requiere que las políticas de consistencia tengan que realizarse en la capa de aplicación haciendo que sea más lento.

Existen varios motores de persistencia relacionales como MySql, Oracle y SqlServer, que tienen una mayor eficiencia para realizar esta tarea. Para evitar los licenciamientos del prototipo se realizará la elección de Mysql.

Web - Interface del cliente

En esta parte se encuentra una problemática particular, los clientes en ocasiones tienen concepciones sobre la mejor usabilidad de sus sistemas, por tanto la interface tiene que poder proveer la flexibilidad de cambio para que cada usuario pueda adaptarlo a sus necesidades.

A continuación se analizará los siguientes lenguajes como ser PHP, Java y JavasCript para ver que tecnología se adapta mejor a esta necesidad.

JSP - Java Server Pages es un marco de trabajo creado por Sun Microsystems se utiliza en conjunto con HTML para crear páginas web. Es una tecnología que ejecuta *server side* y permite realizar la integración con servicios generados en java de manera nativa, esto implica que no se requiere establecer servicios públicos si la aplicación fuera realizada en java. La desventaja que posee, es que el procesamiento se encuentra centralizado, generando más carga en el servidor que provea la interfaz de usuario.

Symfony Es un framework de trabajo realizado en PHP que provee un esquema de trabajo para conectarse a través de los servicios de la api rest y simplifica la arquitectura del componente. Cabe destacar que este lenguaje es muy utilizado y posee una gran comunidad

para resolver diferentes necesidades de interfaces que puedan solicitar los clientes. Ahora bien al igual que JSP, es una tecnología *server side*, esto implica que la carga de trabajo para resolver la web estaría centralizada en un mismo servidor.

AngularJS - Construido en JavaScript es un framework que resuelve todas las necesidades de construcción de un sitio web, al igual que el punto anterior provee un marco de trabajo para construir páginas *single page*. Esta tecnología permite diferir el ciclo de vida de los objetos que son utilizados, facilitando que las páginas puedan refrescar sus datos sin la necesidad de tener que actualizarlas. Otra ventaja que posee, es que su ejecución es *client side*, esto significa que los navegadores de los usuarios son los que están encargados de resolver la construcción de las páginas minimizando la carga de los servidores.

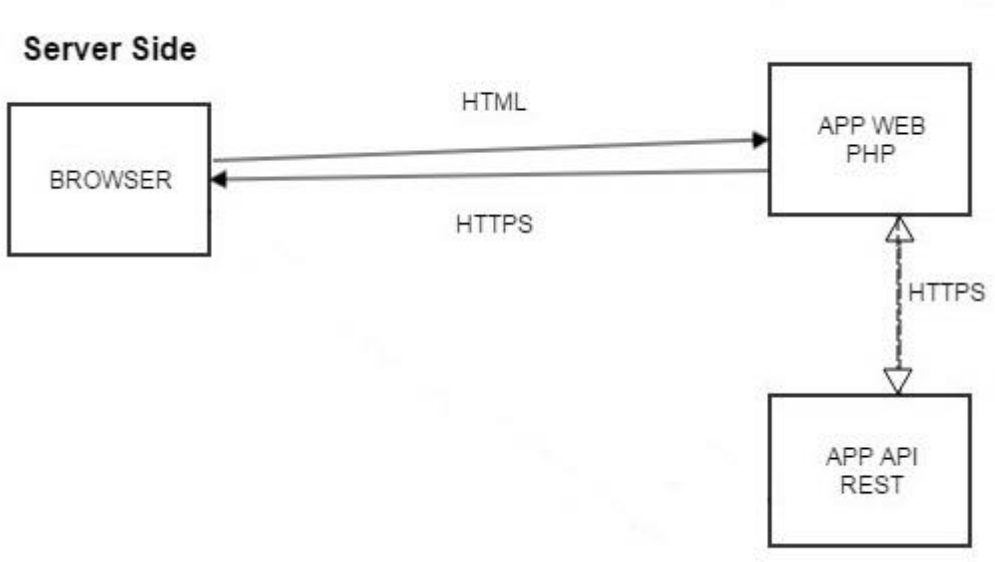


Figura 7.3: Modelo de de ejecucion de server side.

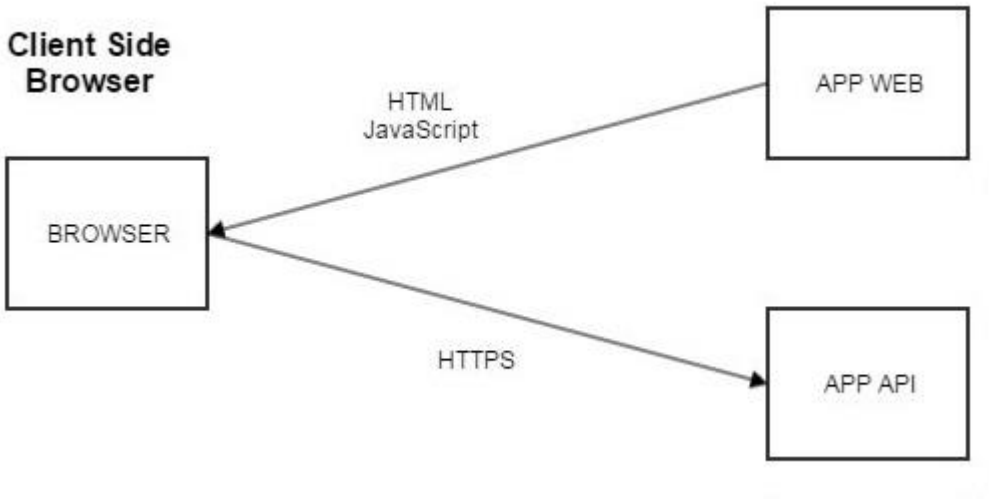


Figura 7.4: Modelo de de ejecucion de client side.

Para concluir con esta sección, queda en claro que todas las tecnologías propuestas proveen soluciones para poder resolver la problemática, pero una de ellas posee la ventaja adicional de no necesitar de un servidor para procesar el *render* de las mismas. Esta ventaja es fundamental a la hora de tener un volumen de tráfico alto, porque se requiere un equipamiento adicional con las tecnologías *server side*.

APP SERVER

Api Rest - Apache tomcat

Es una tecnología Open Source para trabajar como contenedor de aplicaciones como por ejemplo java. Hace años que está demostrando ser un aplicativo sólido y estable para usar sobre aplicaciones productivas de alto rendimiento.

Actualmente es el contenedor donde va a estar alojando la componente API, el cual estará ofreciendo servicios web para realizar posteriormente el *admin* web que consumirá estos servicios.

Ventajas características del Tomcat:

- Servidor soporta aplicaciones java.
- Es un light weight server (no EJB).
- Es de fácil integración con Apache HTTP Server y con IIS.
- Muy estable en sistemas UNIX.
- Buena documentación Online.
- Oracle *compliant*.
- No posee un uso intensivo de la memoria.
- Libre de licenciamientos para su uso.

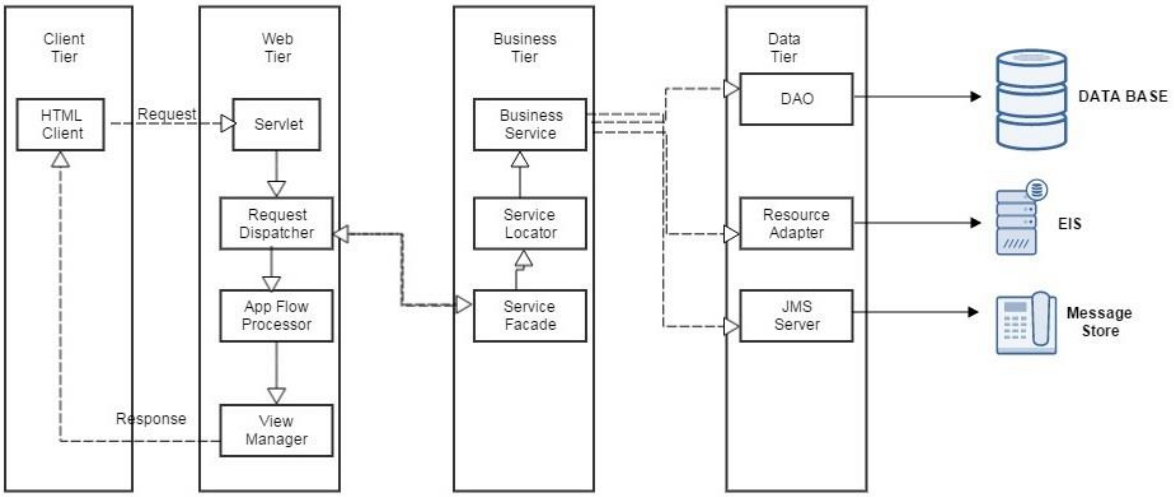


Figura 7.4: Arquitectura del application server tomcat.

Web - Apache 2

Apache 2, es de código abierto utilizado en plataformas UNIX/Linux, Microsoft Windows, Macintosh y otras que utilizan el protocolo HTTP. En la actualidad, tiene un aceptación del 70% en los sitios web de todo el mundo.

Este componente está siendo utilizado para alojar el *admin web;* lo llamamos *FrontEnd*; que se comunica con el api (componente de backend). Actualmente no tiene un backend propio el admin debido a que AngularJs se ejecuta íntegramente en el browser del cliente.

Alguna de las ventajas que posee esta tecnología:

- Es altamente configurable.
- Tiene amplia aceptación de la red.
- No posee licenciamiento
- Compatibilidad con los sistemas operativos.
- Posee código abierto y es fácil conseguir ayuda/soporte.

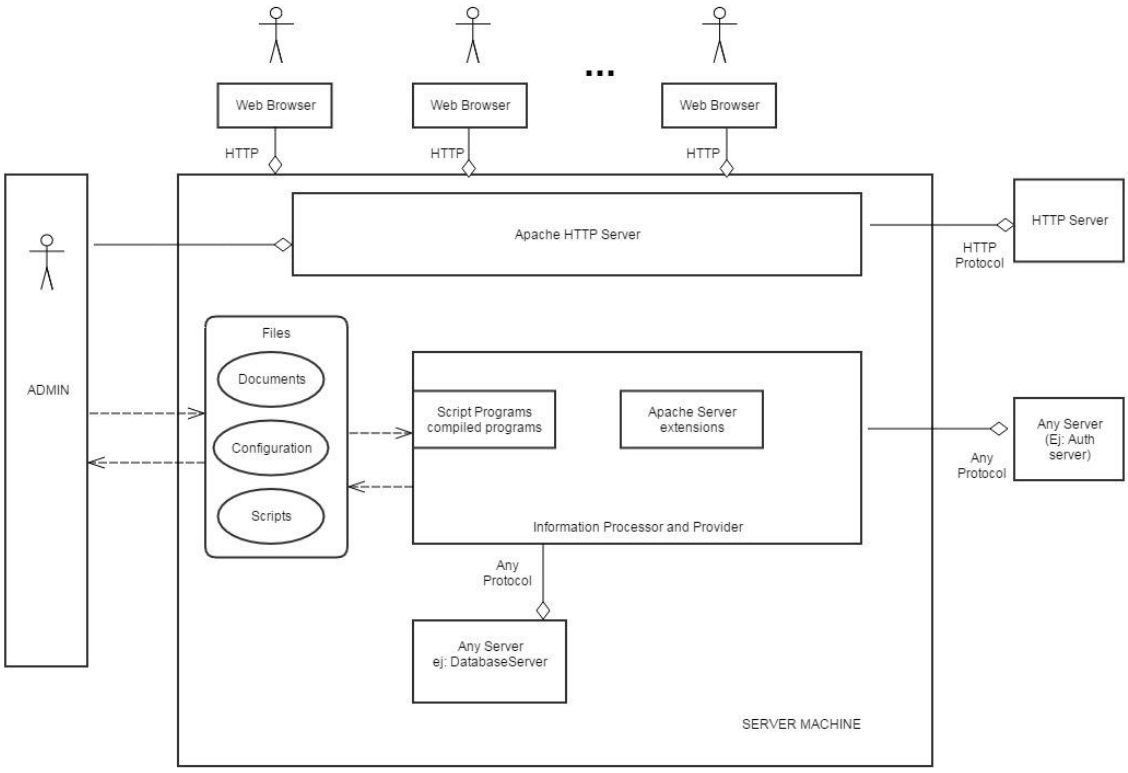


Figura 7.5: Arquitectura del application server apache 2.

SEGURIDAD INFORMÁTICA

En este punto se detallarán las consideraciones que se tener en cuenta al momento de establecer las comunicaciones entre los componentes, debido que se cuenta con información sensible de los usuarios como por ejemplo las credenciales del sistema de recursos.

- Las interfaces tienen que ser de acceso seguro, no pueden estar publicadas sin poseer un certificado de seguridad para establecer la comunicación.
- Todo acceso al sistema debe ser por medio de sus credenciales, para evitar acciones indebidas y ayudar al registro de acciones solicitadas.
- La base de persistencia solo puede ser accedida a través del componente de centralización de accesos. Esto se debe que es el único que puede realizar modificaciones para dejar siempre la base consistente.
- La comunicación con el servidor de recursos tiene que ser a través de certificados, debido a que hay acciones solo puede ser realizada en modo seguro.

Para resolver la comunicación entre la interfaz y el componente Rest Api se utilizara el protocolo Http que permite seguridad a través TLS 1.2. Este protocolo permite generar un certificado de clave privada para el servidor y una clave pública para los clientes, de esta manera podemos generar un túnel de conexión para encriptar los datos para evitar posibles capturas de paquetes.

Para evitar accesos ilegales a la persistencia del componente se puede establecer reglas de accesos con las credenciales del sistema y adicionalmente reglas de firewall para evitar conexiones indeseables.

El servidor de recursos AD/LDAP provee dos tipos de conexiones: ldap y ldaps. La segunda opción permite establecerse sobre TLS 1.2 pudiendo ser verificada al igual que la conexión de la interfaz web con la api rest.

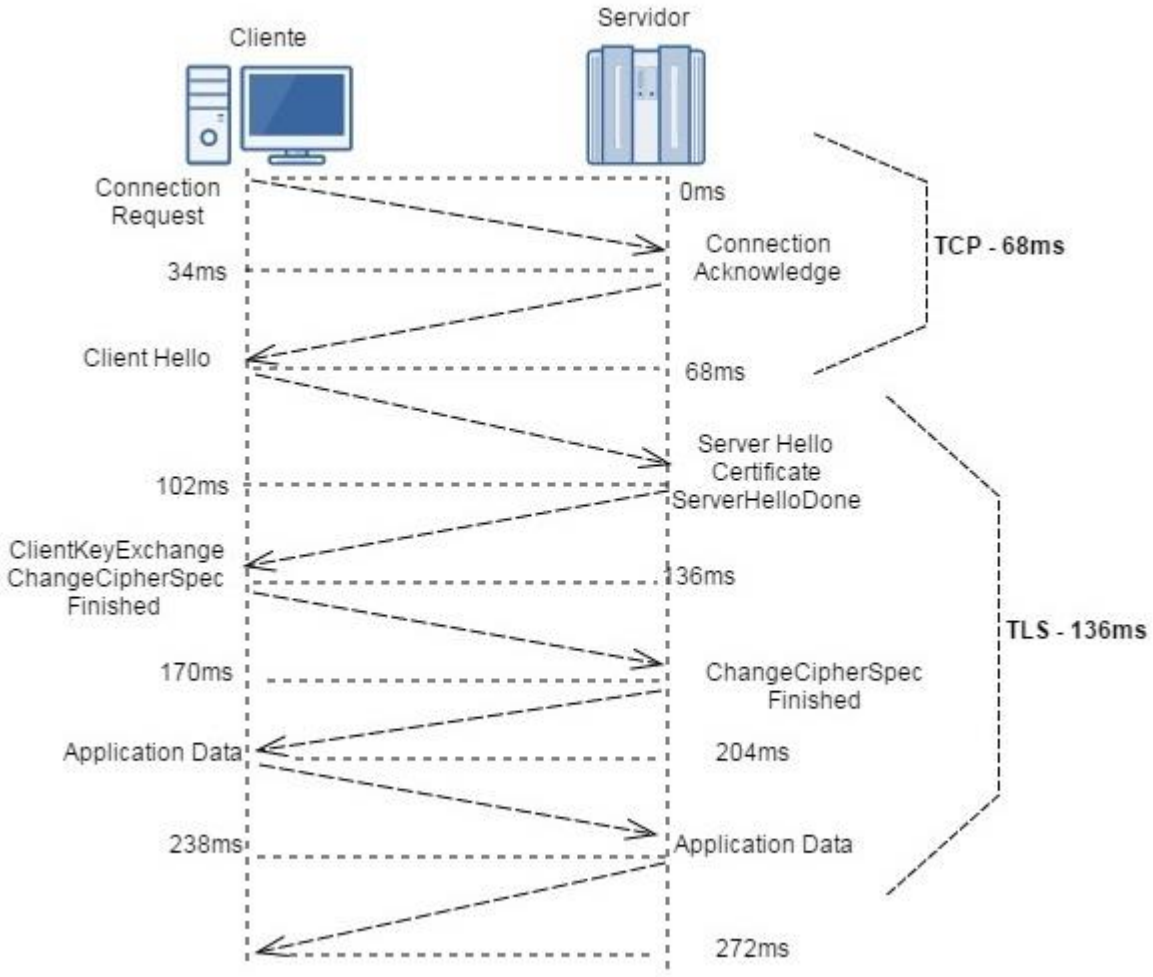


Figura 7.6: Comunicación con protocolo tls 1.2.

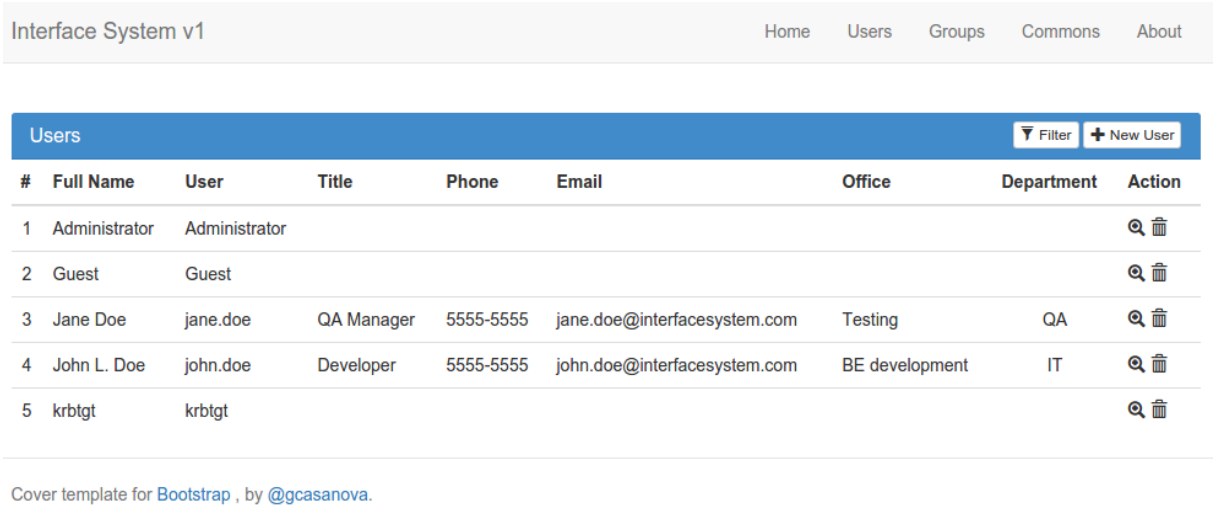
8. DOCUMENTACIÓN

Aquí se determina el manual de usuario para el uso del sistema. Se va a detallar el funcionamiento y las ventajas que provee. La interfaz es web, por tanto se indican los campos ingresados y qué sentido tienen dentro de los contextos de los flujos ya especificados en los requerimientos del sistema.

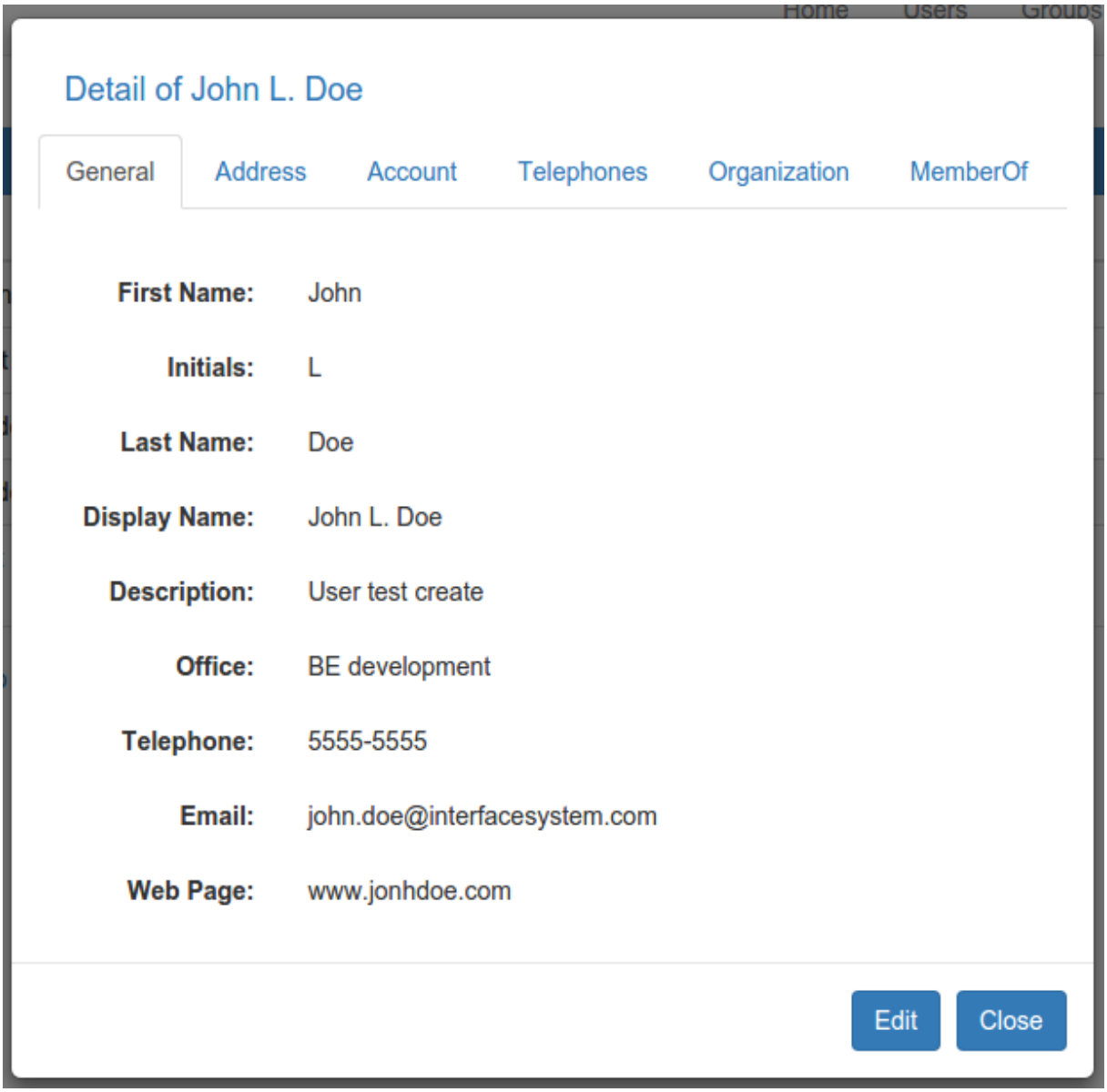
- *Home* o pantalla de ingreso al sistema, contiene la barra de notificaciones de la empresa y posee también el menú principal de la aplicación.

- En *About* vamos a especificar todas las tecnologías que usaron para diagramar el funcionamiento de nuestra aplicación.

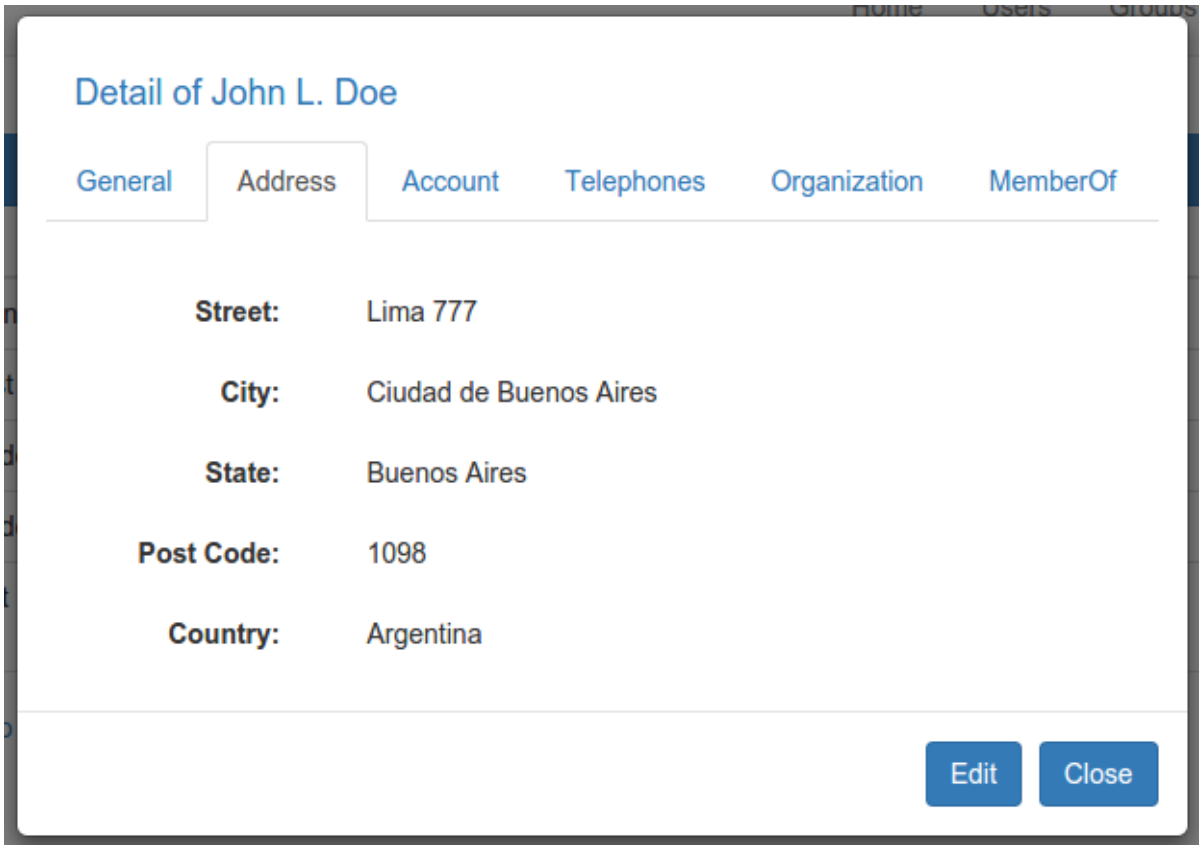
- "Users" esta pestaña se utiliza para realizar las todas las acciones referidas a los usuarios. La página principal posee una lista de usuarios activos con sus respectivos filtros para facilitar las búsquedas. Las acciones posibles son:
    - Filtros por todos los campos en la imagen.
    - Borrado lógico.
    - Ingresar nuevos usuarios.



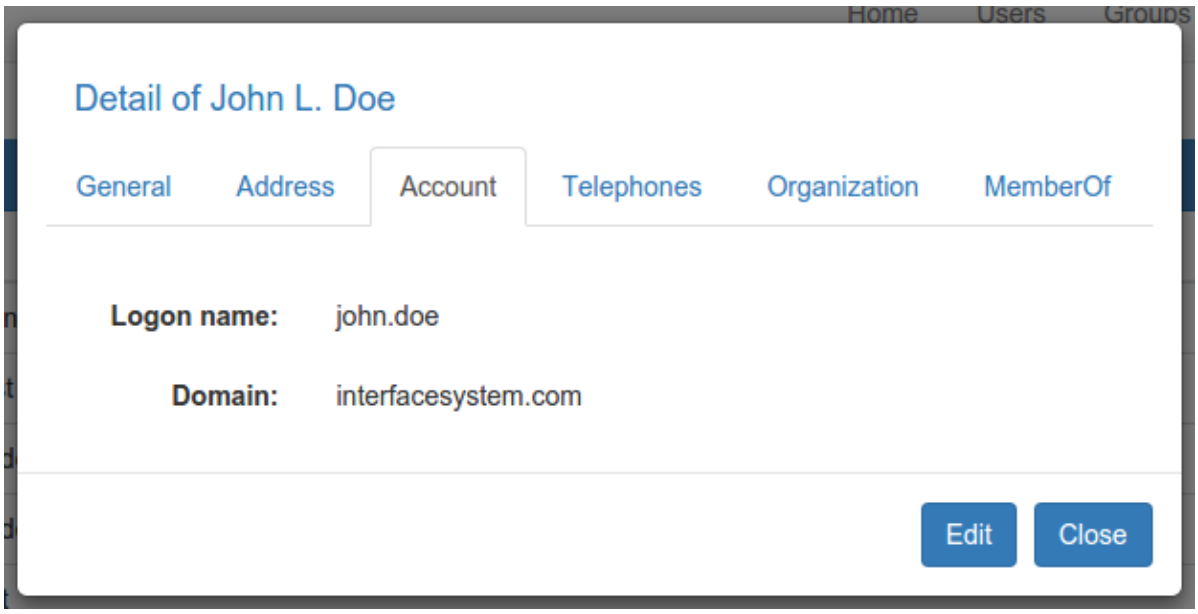
- “User Detail” permite conocer los datos registrados del usuario en el servidor de recursos. Hay que tener en cuenta que el componente realiza una búsqueda, la guarda en motor de persistencia para no tener que buscar los datos permanentemente. La página principal es "General" donde se muestra los datos de la persona.



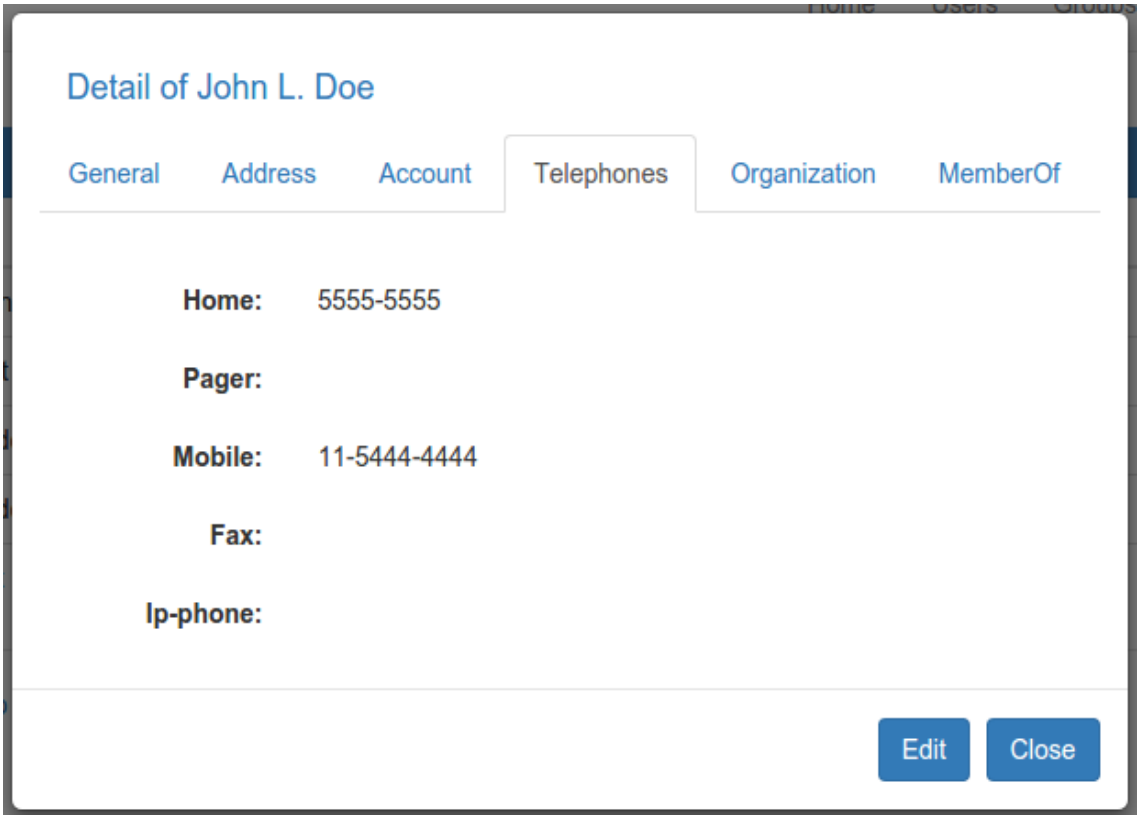
- "Address" indica la dirección donde trabaja el usuario.



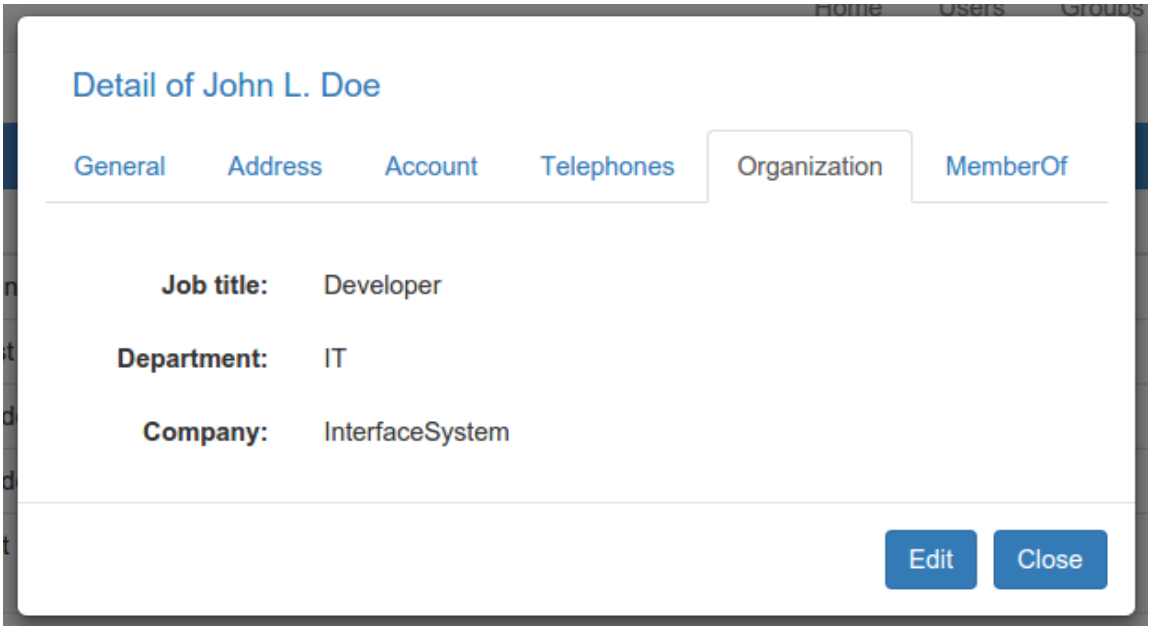
- "Account" nos indicará el nombre de usuario el cual utiliza para ingresar al dominio de su compañía.



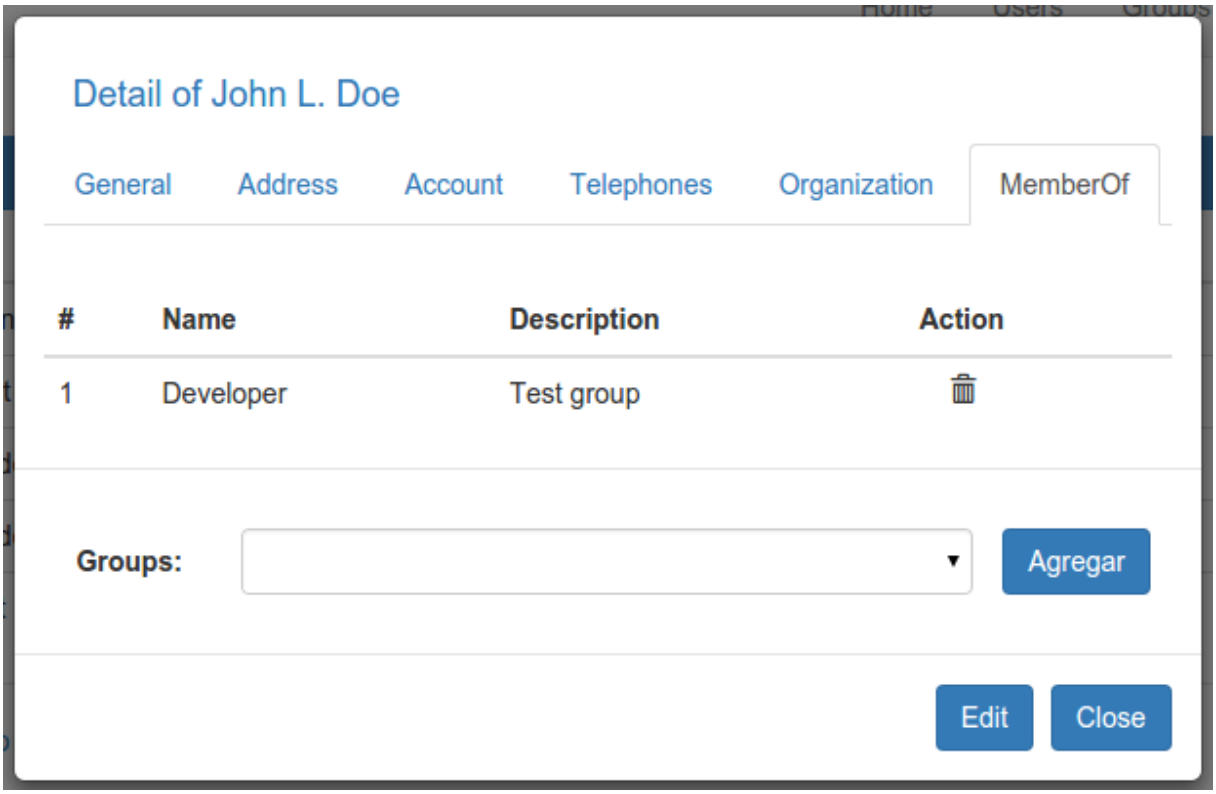
- "Telephones" indica los teléfonos de contacto del usuario.



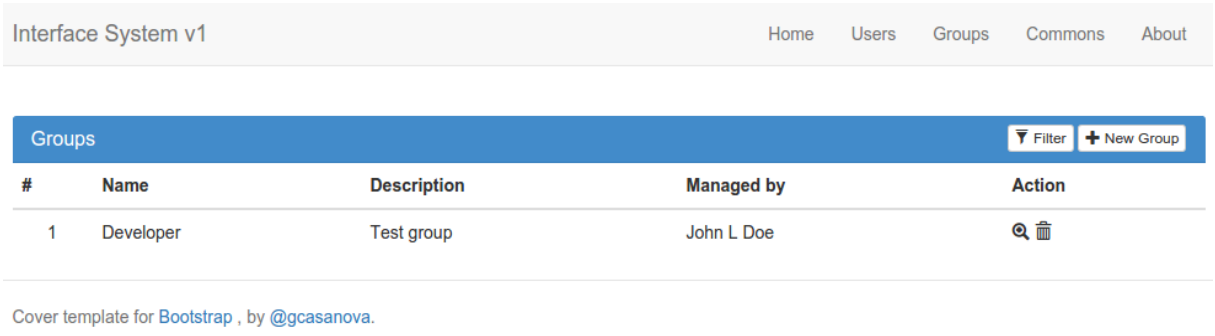
- "Organization" detalla los datos de la posición laboral del usuario:



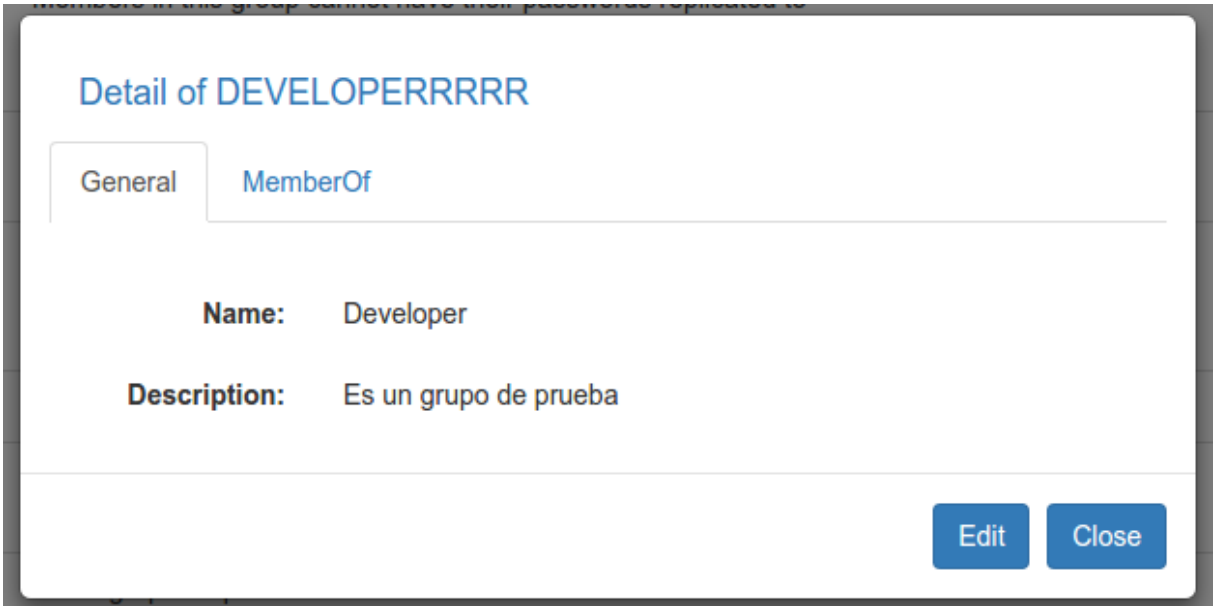
- "MemberOf" esta pestaña muestra los grupos asociados actualmente al usuario. Permite inclusive agrega nuevos grupos a los usuarios.



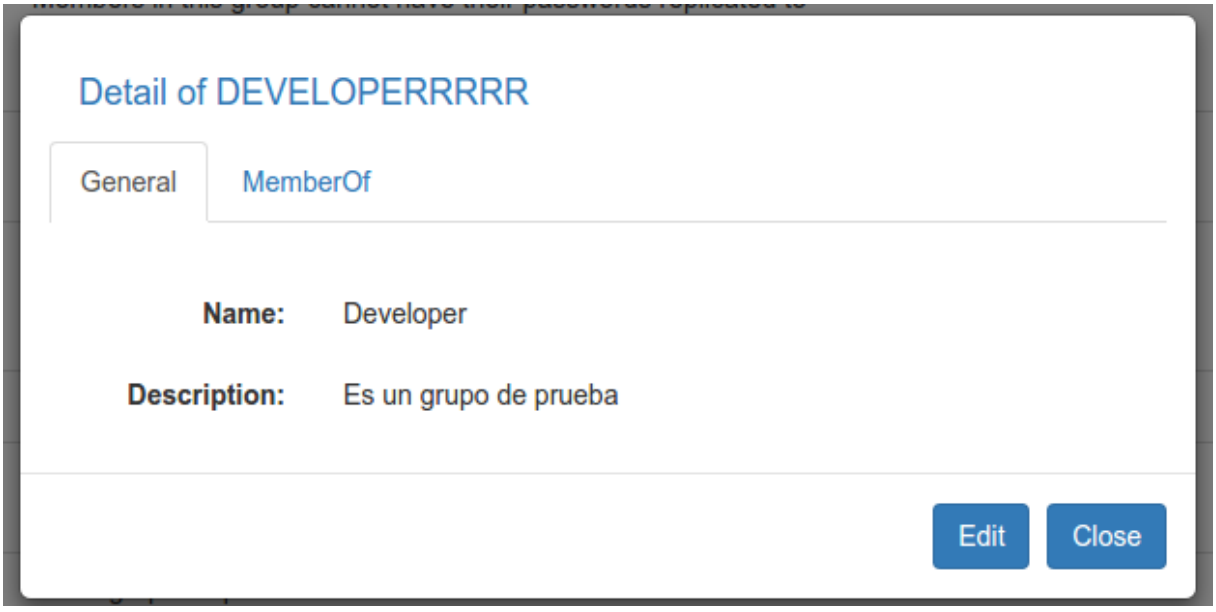
- "Grupos" solapa que se utiliza para la administración de los grupos del servidor de recursos. Las acciones aceptadas son:
  - Filtros para facilitar la búsqueda por parámetros
  - Nuevo grupo
  - Ver detalle de los grupos



- "Detalle de los grupos" La pagina principal se encuentran los datos generales del grupo. Los datos pueden ser editado presionando el botón "Edit"

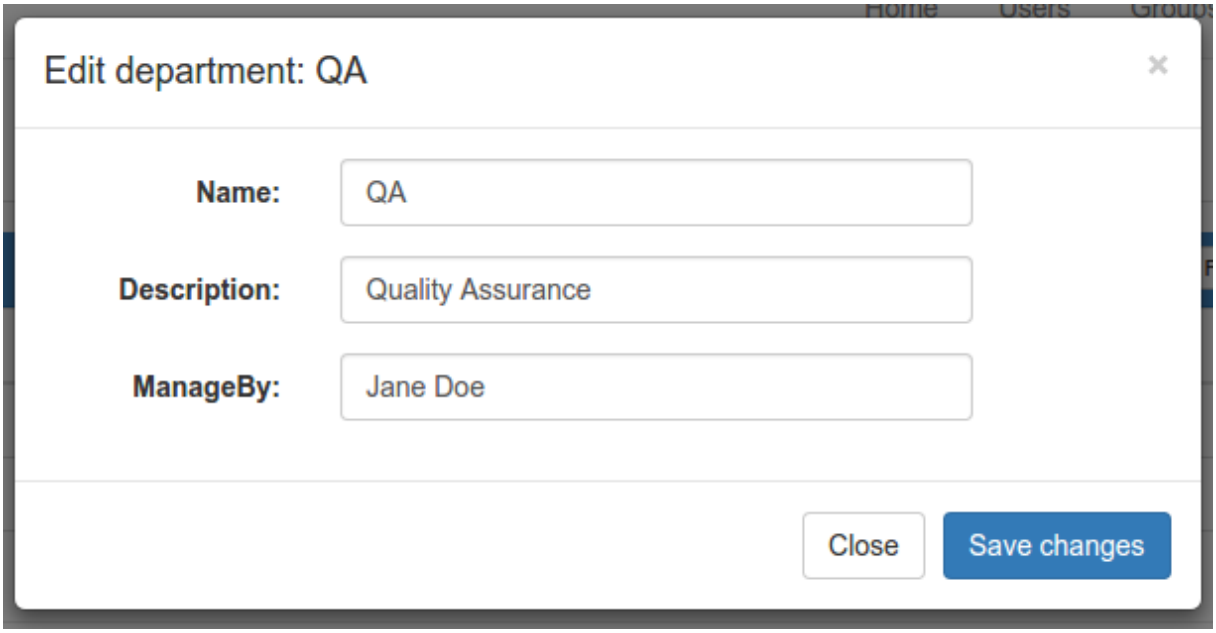


- “MemberOf” nos indicará todos los usuarios que se encuentra dentro de dicho grupo. Las acciones permitidas son:
    - Agregar usuarios
    - Eliminar usuarios del grupo

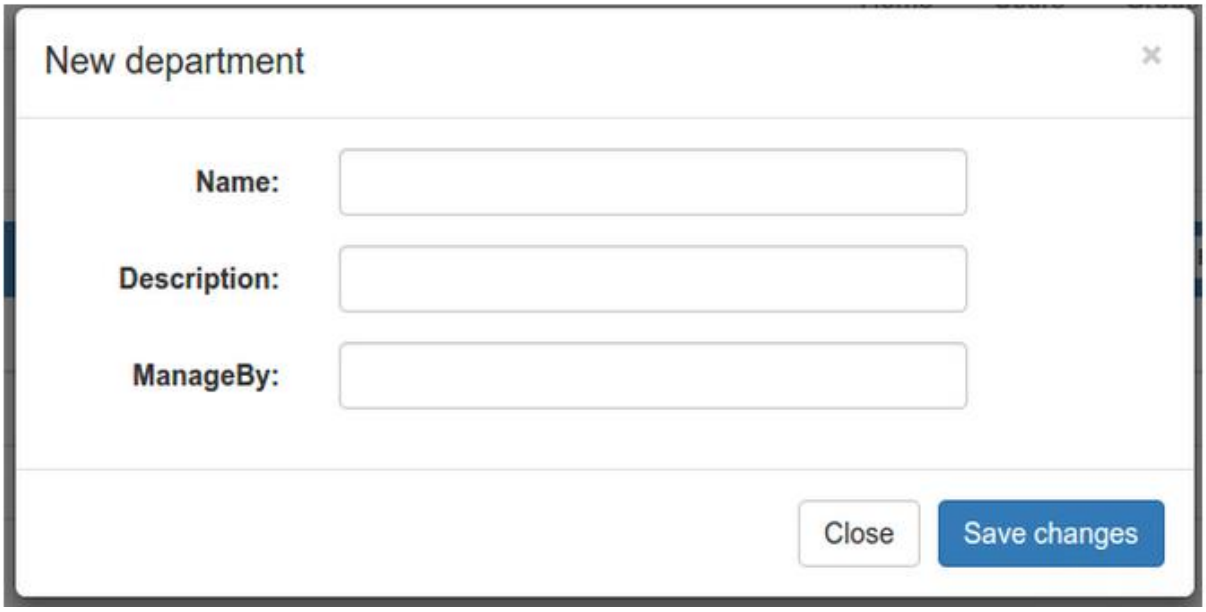


- “Commons” en esta solapa se encuentran todos los datos que se realizan la estandarización para la carga de nuevos registros, alguno de los ejemplo son: departamentos y oficinas. El servidor AD no tiene la capacidad de estandarizar estos campos debido a que son de texto libre dentro del LDAP. Las acciones que permite son:
    - Editar los registros
    - Borrar registros
    - Agregar registros

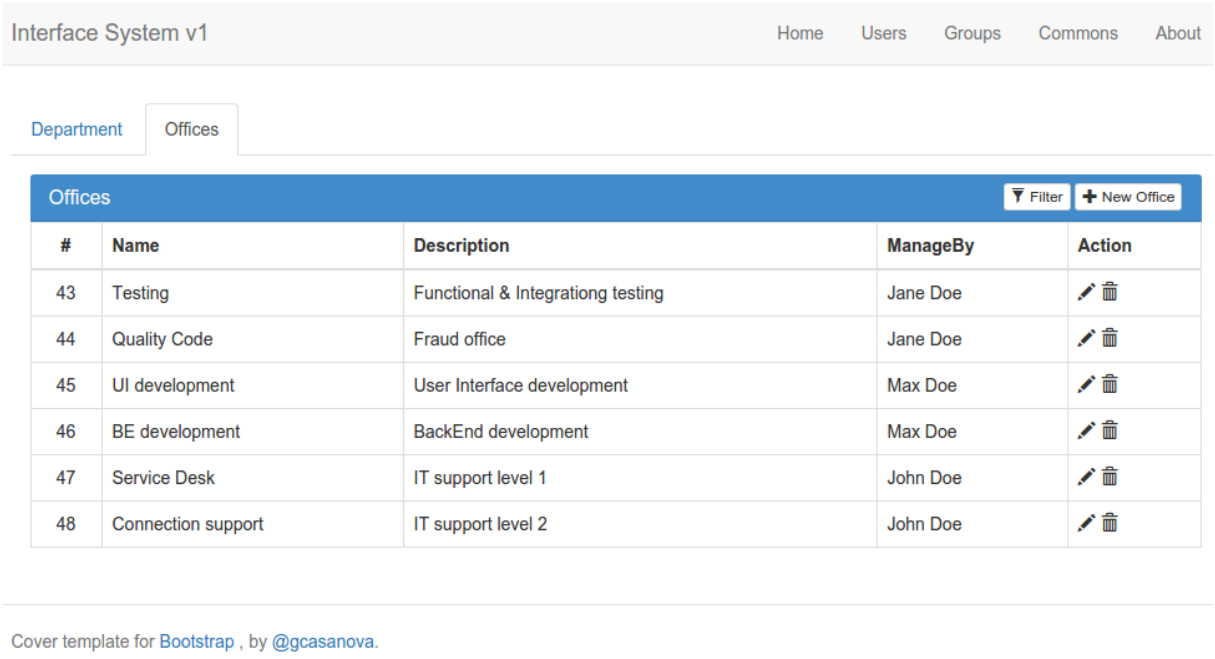
- Si se selecciona la solapa el lápiz que figura debajo de “Action” vamos a poder editar el departamento:



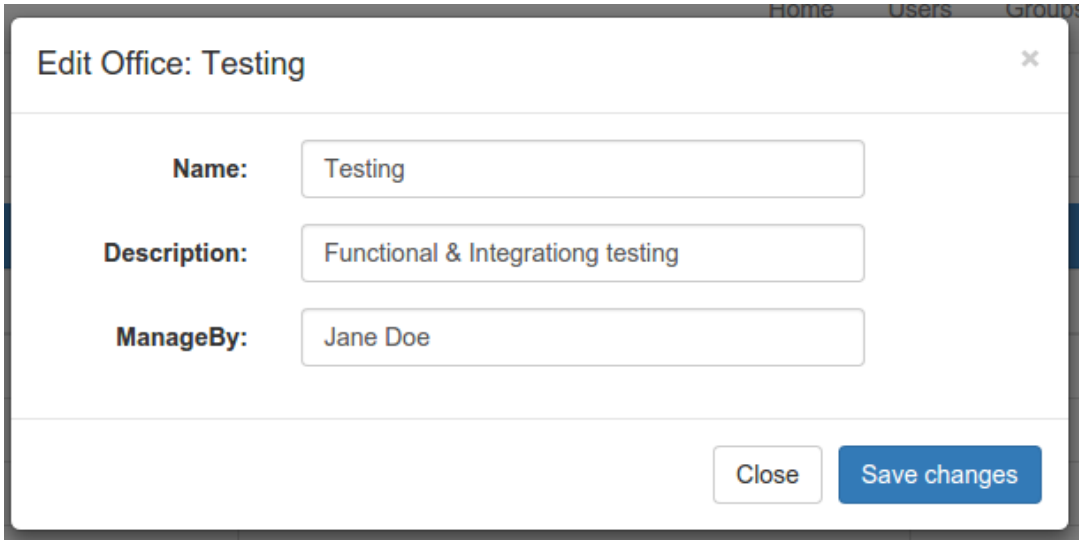
● También se puede crear nuevos departamentos seleccionando la el botón de “New Department”. El campo ManageBy indica quien es la persona que dirige el Departamento, esto es importante saber ya que la aprobación de unos nuevos usuarios debe ser aprobada por el Jefe del Departamento.



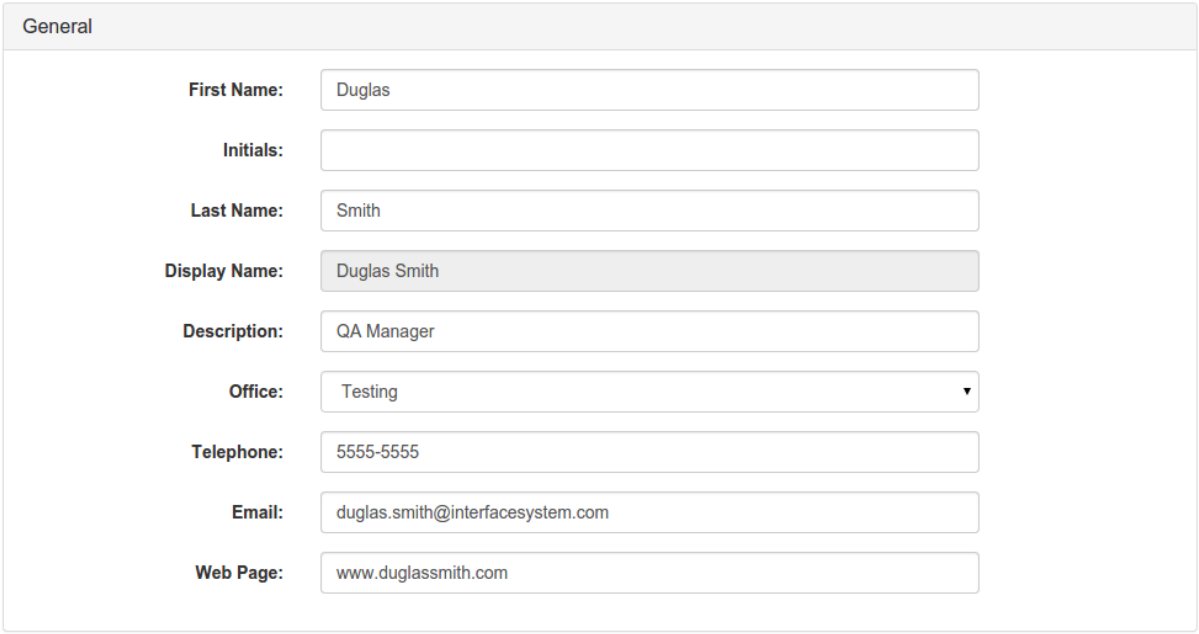
- “Offices” Están definidas las oficinas contenidas por los Departamentos, Se pueden realizar las mismas acciones que en Departamentos.

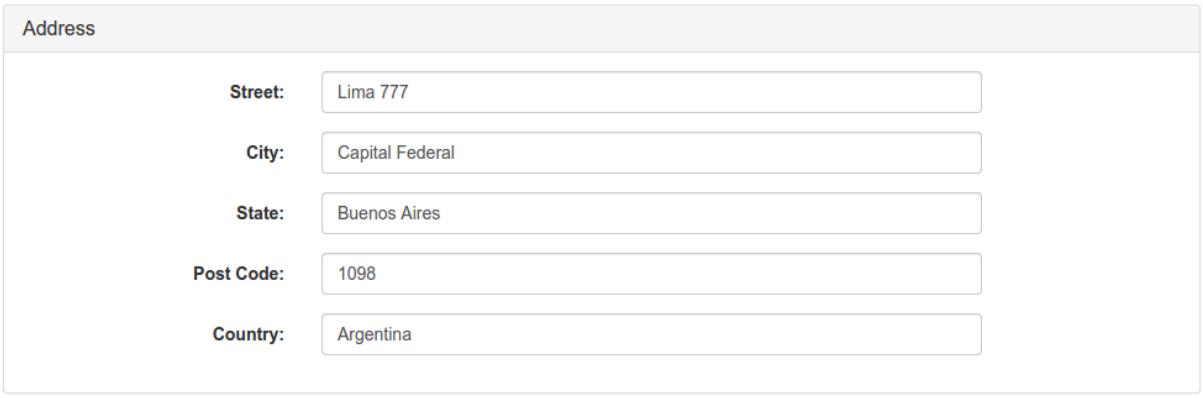


- Al igual que el Departamento se puede editar la oficina:

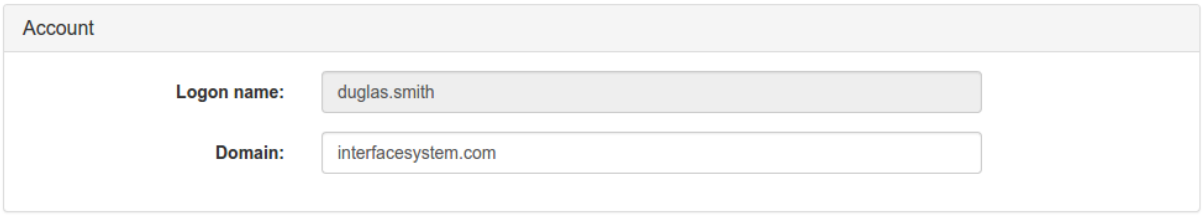


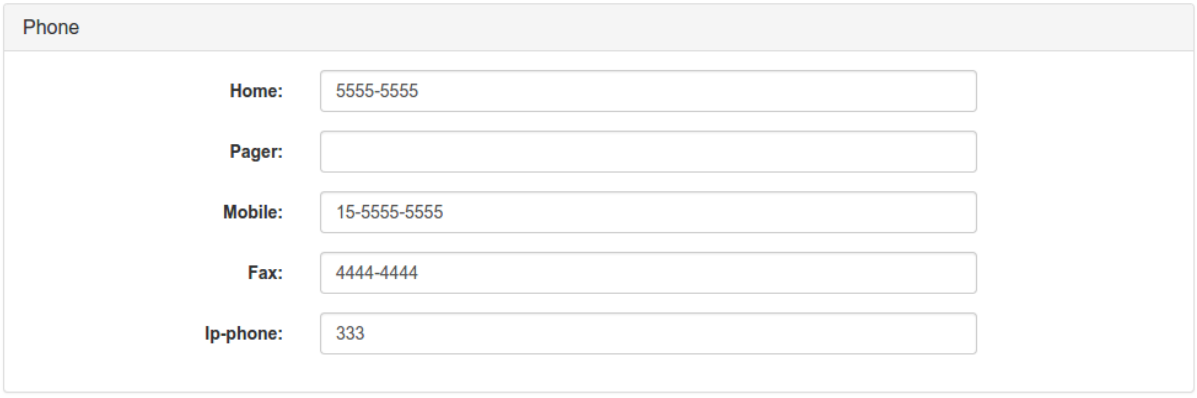
- "New User" permite la creación de los usuarios.
    - "Display Name" es el nombre que tiene para distinguir al usuario, se puede cambiar la manera de generarlo y el usuario que lo escribe no puede reescribir
    - "Office" se despliega una serie de opciones que se cargaron en la solapa de commons, de esta manera no tendremos una oficina cargadas por duplicado.



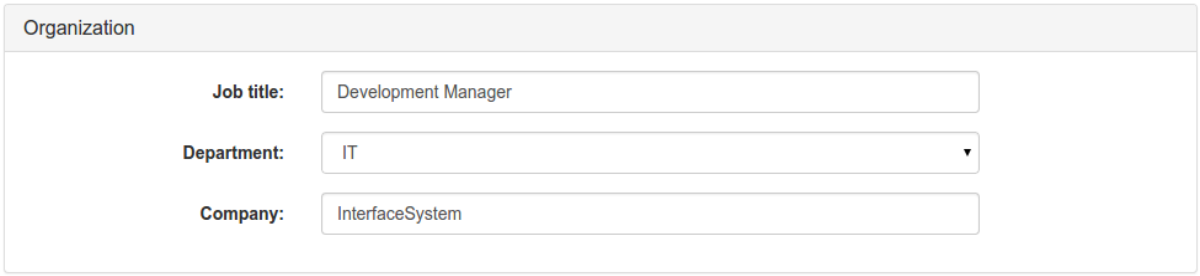


- El campo de "logon name" es un campo autogenerado que se registra a partir de los datos de usuario, la regla de generación puede ser modificada.

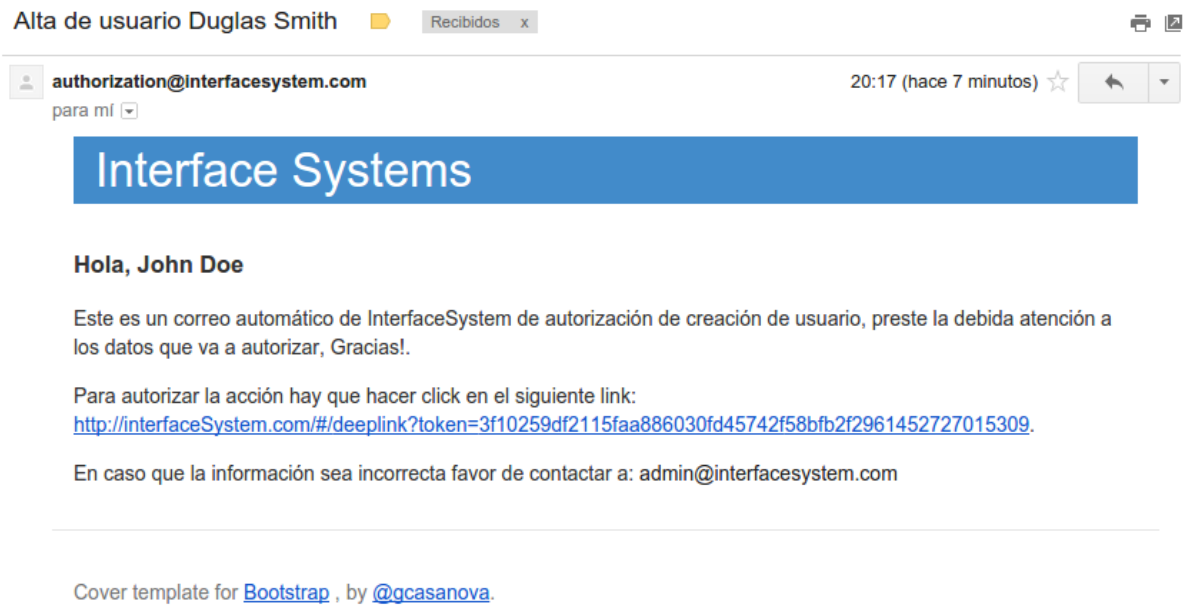




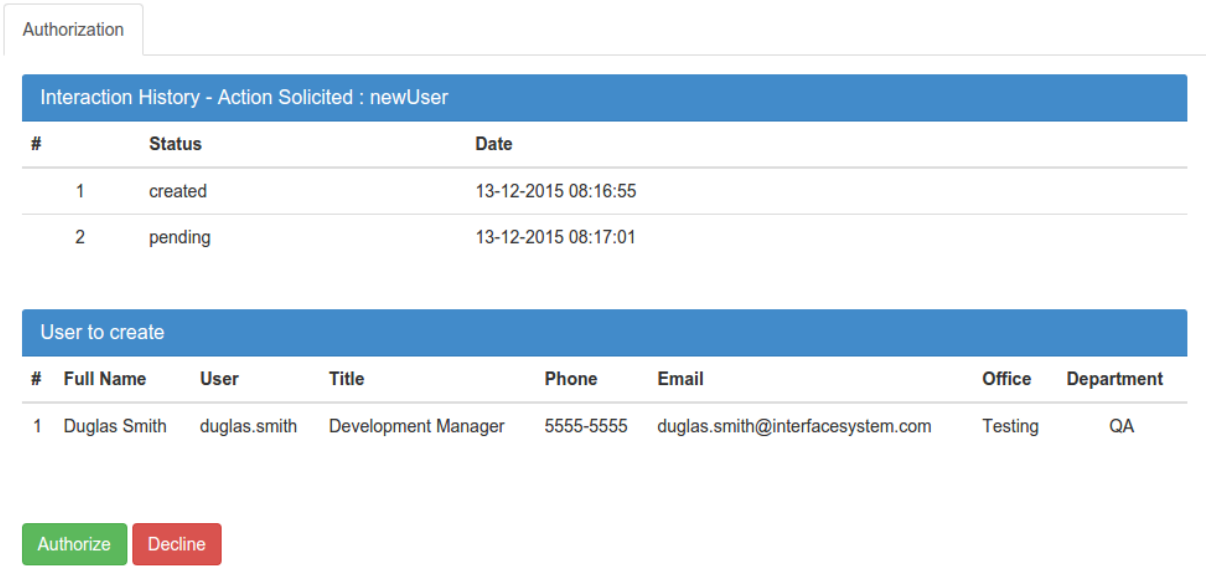
- Al igual que el campo "Office" el campo "Department" despliega una serie de posibles Departamentos que se puede seleccionar para evitar los duplicados.



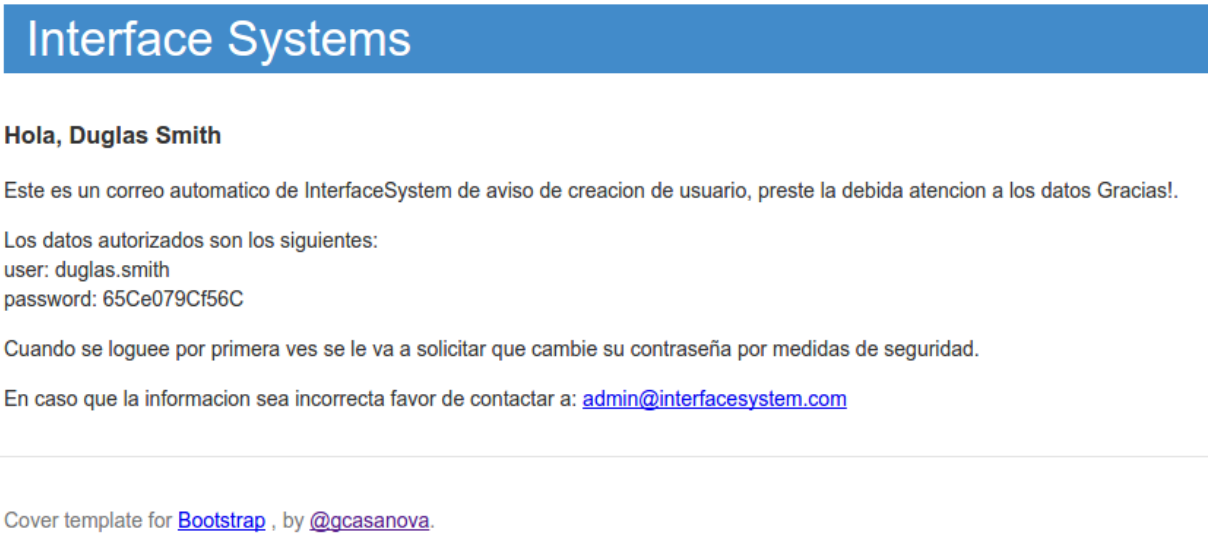
- Al dar de alta el usuario automáticamente el sistema de workflow guarda la solicitud pendiente y busca los aprobadores de la acción para enviarles un mail con el link de aprobación.



- El link de aprobación envía a la página correspondiente para realizar la autorización de la solicitud o declinarla. Permite visualizar el historial de la solicitud y los datos que se requieren aprobar.



- Cuando se autoriza la acción se envía automáticamente un mail al usuario indicando la password que va a utilizar en su primer logueo.



9. CONCLUSIONES

9.1 Objetivos Logrados

De acuerdo a lo definido en el alcance de este trabajo se pudo cumplir con los siguientes objetivos:

- Estudio de las necesidades, herramientas y procedimientos existentes que tienen las compañías para el manejo de su ABM de usuarios.
- Construcción de un prototipo que soporte todas las funcionalidades descritas en los Objetivos Específicos del documento.
- Aplicación del prototipo en distintas compañías.

En primer lugar, se pudo realizar un exhaustivo análisis de las necesidades de las medianas y grandes compañías al momento de realizar altas, bajas y modificaciones de usuarios, sus distintas herramientas ya existentes y el procedimiento de los analistas para llevarla a cabo. En base a este análisis, se determinó qué actividades deberían ser soportados por la herramienta y se pudo diseñar la solución a implementar.

Por último con base en el diseño realizado, se construyó un prototipo que permite automatizar cada una de las actividades que realizaban los analistas, con la finalidad de reducir los tiempos de ejecución y aumentar la disponibilidad del servicio sin tener costos adicionales.

9.2 Evaluación Crítica

La implementación de servicio requiere un exhaustivo análisis de la conformación de la empresa, esto implica que no todas las empresas pueden instalar automáticamente esta aplicación, requiriendo ajustes en las unidades organizativas y en la estandarización de las oficinas y departamentos. Aquí tenemos una barrera de entrada a la hora de adquirir este producto.

9.3 Trabajo Futuro

A continuación se resume los principales puntos sobre los que se desea continuar trabajando o que otras personas interesadas pueden afrontar. Nuevas funcionalidades prácticas al prototipo:

- Agregar equipamiento para aumentar la capacidad del producto.
- Mejorar el diseño del front-end para que soporte múltiples lenguaje.
- Mejorar la configuración del prototipo para soportar múltiples dominios.
- Crear la interfaz para poder realizar migraciones entre servidores que soporten el protocolo LDAP.
- Crear la interfaz para poder realizar imagen de las configuraciones actuales del servidor.

10. BIBLIOGRAFÍA

MICROSOFT Active Directory, [consultada 24 dic 2014]
https://support.microsoft.com/es-es/kb/196464

LDAP , [consultada 24 dic 2014]
http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-ldap.html

LDAP & AD , [consultada 24 dic 2014]
https://msdn.microsoft.com/en-us/library/aa367008(v=vs.85).aspx

LDAPS [consultada 24 dic 2014]
https://support.microsoft.com/en-us/kb/321051

Api Rest , [consultada 21 ene 2015]
https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
http://www.ibm.com/developerworks/library/ws-restful/

Ldap commands, [consultada 02 feb 2015]
https://docs.oracle.com/cd/B10501_01/network.920/a96579/comtools.htm

Spring ldap [consultada 12 Oct 2015]
http://projects.spring.io/spring-ldap/

Ruby net ldap [consultada 12 Oct 2015]
http://www.rubydoc.info/gems/ruby-net-ldap/Net/LDAP

LdapJs [consultada 12 Oct 2015]
http://ldapjs.org/

JSP JavaServer Pages [consultada 15 Oct 2015]
http://www.oracle.com/technetwork/java/javaee/jsp/index.html

PHP - Symfony [consultada 15 Oct 2015]
https://symfony.com/

ANGULARJS, Framework utilizado para soporte de lógica web [consultada 05 Sep 2015]
https://angularjs.org

BOOTSTRAP, Framework utilizado para la estética de la web [consultada 09 Sep 2015]
https://getbootstrap.com

GIT, Repositorio de versionado de código [consultada 10 Sep 2015]
http://git-scm.com

NODEJS, Servidor que soporta la aplicación web [consultada 29 Jul 2015]

http://nodejs.org

BOWER, Gestor de dependencias del proyecto web [consultada 29 Jul 2015]
http://bower.io

GRUNT, Gestor de tareas para el ciclo de despliegue [consultada 28 Jul 2015]
http://gruntjs.com/

TSL 1.2 [consultada 20 Nov 2015]
https://msdn.microsoft.com/en-us/library/windows/desktop/aa380516(v=vs.85).aspx

11. ANEXOS

CASOS DE USO

Caso de Uso ID:	AU01	
Caso de Uso Nombre:	Alta de Usuario	
Creado por:	Germán Casanova	Última actualización por: Luis López
Fecha Creación:	15/01/2015	Fecha última actualización: 18/01/2015
Actor:	Solicitante, aprobador, usuario final, sistema	
Descripción:	En la gestión de alta de usuario se va a crear un nuevo usuario en el dominio con los grupos correspondientes siempre y cuando los aprobadores de dichos grupos aprueben la solicitud.	
Precondiciones:	Que los grupos existan. Que los departamentos existan. Que las oficinas existan	
Postcondiciones:	Se crea el usuario	
Prioridad:	Alta	
Frecuencia de uso:	Alta	

Flujo Normal:		
	Actor	Sistema
	1) El solicitante ingresa a la página del login del sistema	
	2) El solicitante ingresa sus credenciales en la página del login del sistema	3) El sistema verifica los datos y devuelve un token de sesión.
	4) El solicitante ingresa a la sección de usuarios, creación de usuarios.	5) El sistema obtiene los departamentos habilitados por el usuario y unidades de negocio habilitados por el usuario
	6) El solicitante carga los datos del nuevo usuario, siendo: mail, password, departamento, primer nombre, segundo nombre, numero de contacto, título de puesto, unidad de negocio.	7) El sistema solicita si hay alguna carga extra de datos.
	8) El solicitante indica que no.	9) El sistema valida los datos y le indica que las aprobaciones han sido enviadas con éxito.
Flujos Alternativos:	Actor	Sistema
El solicitante desea cargar permisos adicionales a los que ya tienen precargados	8) EL solicitante indica que quiere cargar permisos adicionales.	9) El sistema realiza la búsqueda de los permisos adicionales que tiene habilitados la unidad de

a la unidad de negocio		negocio.
	10) El solicitante agrega el/los permisos que requiere	11) El sistema valida los permisos adicionales e indica si quiere agregar algún otro permiso adicional.
	12) el solicitante le indica que no	13) El sistema valida los datos y le indica que las aprobaciones han sido enviada con éxito
Flujos Alternativos:	Actor	Sistema
El solicitante desea copiar los permisos de otro usuario	8) El solicitante indica que quiere copiar los permisos de otro usuario	9) El sistema busca los usuarios correspondiente a la unidad de negocio y muestra los grupos
	10) El solicitante selecciona el usuario a ser copiado	11) El sistema valida los grupos e indica que las aprobaciones fueron enviadas con éxito

Caso de Uso ID:	AU02	
Caso de Uso Nombre:	Aprobación de solicitudes	
Creado por:	Luis López	Última actualización por: Luis López
Fecha Creación:	15/01/2015	Fecha última actualización: 18/01/2015

Actor:	aprobadores, solicitante, sistema	
Descripción:	En la gestión de aprobación de usuario se aceptar o rechazará el agregado del usuario al grupo solicitado	
Precondiciones:	Que exista una solicitud pendiente. Que el aprobador exista y se encuentre habilitado.	
Postcondiciones:	Se asocian los permisos al usuario.	
Prioridad:	Media	
Frecuencia de uso:	Alta	
Flujo Normal:		
	Actor	Sistema
	1) El aprobador ingresa a la página de login del sistema	2) El sistema valida las credenciales y devuelve un token de sesión.
		3) El sistema verifica las aprobaciones pendientes del aprobador y realiza un listado.
	4) El aprobador recorre las aprobaciones pendientes y aprueba o rechaza las solicitudes según corresponde.	5) El sistema le indica que sus acciones fueron grabadas con éxito y asocia los permisos a los usuarios correspondientes.

Caso de Uso ID:	AU03	
Caso de Uso Nombre:	Baja de Usuario	
Creado por:	Luis López	Última actualización por: Luis López
Fecha Creación:	15/01/2015	Fecha última actualización: 18/01/2015
Actor:	Solicitante, sistema	
Descripción:	En la gestión de baja de usuario se va a deshabilitar el usuario y se lo va a sacar de todos los grupos a los que pertenece para impedir su acceso.	
Precondiciones:	Que el usuario exista y se encuentre habilitado	
Postcondiciones:	Se deshabilitará el usuario	
Prioridad:	Alta	
Frecuencia de uso:	Alta	
Flujo Normal:		
	Actor	Sistema

	1) El solicitante ingresa a la página del login del sistema	
	2) El solicitante ingresa sus credenciales en la página del login del sistema	3) El sistema verifica los datos y devuelve un token de sesión.
	4) El solicitante ingresa al a sección de usuarios, baja de usuarios.	
	5) El solicitante carga el usuario al cual quiere dar de baja	6) El sistema obtiene todos los grupos donde se encuentra habilitado el usuario y le pide confirmación al solicitante
	7) El solicitante confirma la baja	7) El sistema verifica el usuario ingresado y lo saca de todos los grupos a los cuales tiene acceso y deshabilita el usuario.
Flujos Alternativos:	Actor	Sistema
El solicitante ingresa un usuario erróneo o que se encuentra deshabilitado	5) El solicitante carga el usuario al cual quiere dar de baja	6) El sistema no puede obtener los grupos de dicho usuario debido a que es erróneo o porque el mismo se encuentra deshabilitado y le pide al solicitante que vuelva a ingresar el usuario.

Caso de Uso ID:	AU05	
Caso de Uso Nombre:	Desbloqueo de Usuario	
Creado por:	German Casanova	Última actualización por: Luis López
Fecha Creación:	15/01/2015	Fecha última actualización: 18/01/2015
Actor:	Solicitante, sistema	
Descripción:	En la gestión de desbloqueo de usuario se va a desbloquear el usuario pudiendo así volver a ingresar al sistema.	
Precondiciones:	Que el usuario exista y se encuentre bloqueado	
Postcondiciones:	Usuario desbloqueado	
Prioridad:	Alta	
Frecuencia de uso:	Alta	
Flujo Normal:		
	Actor	Sistema

	1) El solicitante ingresa a la página del login del sistema	
	2) El solicitante ingresa sus credenciales en la página del login del sistema	3) El sistema verifica los datos y devuelve un token de sesión.
	4) El solicitante ingresa a la sección de usuarios, desbloqueo de usuarios	
	5) El solicitante carga el usuario al cual quiere desbloquear	6) El sistema desbloquea al usuario
Flujos Alternativos:	Actor	Sistema
El solicitante ingresa un usuario erróneo o que se encuentra deshabilitado	5) El solicitante carga el usuario al cual quiere desbloquear.	6) El sistema no puede obtener a dicho usuario debido a que es erróneo o porque el mismo se encuentra deshabilitado y le pide al solicitante que vuelva a ingresar el usuario.

Caso de Uso ID:	AU06	
Caso de Uso Nombre:	Reinicio de clave de Usuario	
Creado por:	Luis López	Última actualización por: Luis López
Fecha Creación:	15/01/2015	Fecha última actualización: 18/01/2015
Actor:	Solicitante, sistema	
Descripción:	En la gestión de reinicio de clave de usuario se va a reiniciar la clave del usuario pudiendo así volver a ingresar al sistema.	
Precondiciones:	Que el usuario exista.	
Postcondiciones:	Usuario con clave blanqueada	
Prioridad:	Alta	
Frecuencia de uso:	Alta	
Flujo Normal:		
	Actor	Sistema
	1) El solicitante ingresa a la página del login del sistema	

	2) El solicitante ingresa sus credenciales en la página del login del sistema	3) El sistema verifica los datos y devuelve un token de sesión.
	4) El solicitante ingresa a la sección de usuarios, reinicio de clave de usuarios	
	5) El solicitante carga el usuario al cual quiere reiniciar la clave	6) El sistema devuelve una clave randon al solicitante y setea para que el usuario la próxima vez que ingrese vuelva a ingresar una nueva contraseña
Flujos Alternativos:	Actor	Sistema
El solicitante ingresa un usuario erróneo o que se encuentra deshabilitado	5) El solicitante carga el usuario al cual quiere reiniciar la contraseña	6) El sistema no puede obtener a dicho usuario debido a que es erróneo o porque el mismo se encuentra deshabilitado y le pide al solicitante que vuelva a ingresar el usuario.

Caso de Uso ID:	AU07	

Caso de Uso Nombre:	Modificación Usuario	
Creado por:	Luis López	Última actualización por: Luis López
Fecha Creación:	15/01/2015	Fecha última actualización: 18/01/2015
Actor:	Solicitante, sistema	
Descripción:	En la gestión de modificación de usuario se realizan actualizaciones ya sea de su teléfono, descripción de puesto, etc.	
Precondiciones:	Que el usuario exista.	
Postcondiciones:	Usuario con perfil modificado	
Prioridad:	Baja	
Frecuencia de uso:	Media	
Flujo Normal:		
	Actor	Sistema
	1) El solicitante ingresa a la página del login del sistema	
	2) El solicitante ingresa sus credenciales en la página del	3) El sistema verifica los datos y devuelve un token de sesión.

	login del sistema	
	4) El solicitante ingresa a la sección de usuarios, modificación de usuarios.	
	5) El solicitante carga el usuario al cual quiere modificar su perfil	6) El sistema trae toda la información del usuario y le muestra los campos que pueden ser modificados.
	7) El solicitante realiza las modificaciones necesarias e ingresa guardar.	7) El sistema guarda la información.
Flujos Alternativos:	Actor	Sistema
El solicitante ingresa un usuario erróneo o que se encuentra deshabilitado	5) El solicitante carga el usuario al cual quiere actualizar	6) El sistema no puede obtener el usuario debido a que es erróneo o porque el mismo se encuentra deshabilitado y le pide al solicitante que vuelva a ingresar el usuario.

Caso de Uso ID:	AU09	
Caso de Uso Nombre:	Blanqueo Clave Bulk	
Creado por:	Luis López	Última actualización por: Luis López
Fecha Creación:	15/01/2015	Fecha última actualización: 18/01/2015
Actor:	Solicitante, sistema	
Descripción:	En la gestión de reinicio de clave en Bulk el solicitante va a ingresar X cantidad de usuario separados por coma o un departamento completo para reiniciar su clave y que el sistema automáticamente le envié al mail del solicitante/supervisor y al usuario su nueva password.	
Precondiciones:	Que los usuarios existan.	
Postcondiciones:	Usuarios con clave blanqueada	
Prioridad:	Alta	

Frecuencia de uso:	Alta	
Flujo Normal:		
	Actor	Sistema
	1) El solicitante ingresa a la página del login del sistema	
	2) El solicitante ingresa sus credenciales en la página del login del sistema	3) El sistema verifica los datos y devuelve un token de sesión.
	4) El solicitante ingresa a la sección de usuarios, reinicio de clave de usuarios en Bulk	
	5) El solicitante carga todos los usuarios separados por coma o también puede cargar un departamento completo para reiniciar la clave	6) El sistema devuelve una clave randon al solicitante y setea para que los usuarios la próxima vez que ingresen vuelvan a ingresar una nueva contraseña. También manda un mail al solicitante/supervisor y a los usuarios con la nueva clave para ingresar.
Flujos Alternativos:	Actor	Sistema

El solicitante ingresa un usuario erróneo o que se encuentra deshabilitado	5) El solicitante carga todos los usuarios separados por coma o también puede cargar un departamento completo para reiniciar la clave	6) El sistema no puede obtener algún/os usuario/s debido a que es erróneo o porque el mismo se encuentra deshabilitado y le pide al solicitante que vuelva a ingresar el/los usuario/s.

Caso de Uso ID:	AU10	
Caso de Uso Nombre:	Desbloqueo cuenta Bulk	
Creado por:	German Casanova	Última actualización por: Luis López
Fecha Creación:	15/01/2015	Fecha última actualización: 18/01/2015
Actor:	Solicitante, sistema	
Descripción:	En la gestión de desbloqueo de cuentas en Bulk el solicitante va a ingresar X cantidad de usuario separados por coma o un departamento completo para desbloquear su cuenta y que el sistema automáticamente le envié al mail del solicitante/supervisor y al usuario indicando que su cuenta fue desbloqueada.	

Precondiciones:	Que los usuarios existan.	
Postcondiciones:	Usuarios con cuenta desbloqueada.	
Prioridad:	Alta	
Frecuencia de uso:	Alta	
Flujo Normal:		
	Actor	Sistema
	1) El solicitante ingresa a la página del login del sistema	
	2) El solicitante ingresa sus credenciales en la página del login del sistema	3) El sistema verifica los datos y devuelve un token de sesión.
	4) El solicitante ingresa a la sección de usuarios, desbloqueo de clave de usuarios en Bulk	
	5) El solicitante carga todos los usuarios separados por coma o también puede cargar un departamento completo para desbloquear la cuenta.	6) El sistema indica que las cuentas fueron desbloqueada y también envía un mail al solicitante/supervisor y a los usuarios indicando que su cuenta fue desbloqueada.

Flujos Alternativos:	Actor	Sistema
El solicitante ingresa un usuario erróneo o que se encuentra deshabilitado	5) El solicitante carga todos los usuarios separados por coma o también puede cargar un departamento completo para desbloquear la cuenta.	6) El sistema no puede obtener algún/os usuario/s debido a que es erróneo o porque el mismo se encuentra deshabilitado y le pide al solicitante que vuelva a ingresar el/los usuario/s.

Caso de Uso ID:	AU11	
Caso de Uso Nombre:	Baja de usuarios en Bulk	
Creado por:	Luis López	Última actualización por: Luis López
Fecha Creación:	15/01/2015	Fecha última actualización: 18/01/2015
Actor:	Solicitante, sistema	

Descripción:	En la gestión de baja de usuarios de cuentas en Bulk el solicitante va a ingresar X cantidad de usuario separados por coma o un departamento completo para deshabilitar su cuenta.	
Precondiciones:	Que los usuarios existan y se encuentren habilitados	
Postcondiciones:	Usuarios con cuenta deshabilitada.	
Prioridad:	Alta	
Frecuencia de uso:	Alta	
Flujo Normal:		
	Actor	Sistema
	1) El solicitante ingresa a la página del login del sistema	
	2) El solicitante ingresa sus credenciales en la página del login del sistema	3) El sistema verifica los datos y devuelve un token de sesión.
	4) El solicitante ingresa a la sección de usuarios, baja de usuarios en bulk	

	5) El solicitante carga todos los usuarios separados por coma o también puede cargar un departamento completo para deshabilitar su cuenta.	6) El sistema indica que las cuentas fueron deshabilitadas.
Flujos Alternativos:	Actor	Sistema
El solicitante ingresa un usuario erróneo o que se encuentra deshabilitado	5) El solicitante carga todos los usuarios separados por coma o también puede cargar un departamento completo para deshabilitar su cuenta.	6) El sistema no puede obtener algún/os usuario/s debido a que es erróneo y le pide al solicitante que vuelva a ingresar el/los usuario/s.

Caso de Uso ID:	AU12	
Caso de Uso Nombre:	Agregar Grupos a usuarios	
Creado por:	Luis López	Última actualización por: Luis López
Fecha Creación:	15/01/2015	Fecha última actualización: 18/01/2015
Actor:	Solicitante, sistema	

Descripción:	En la gestión de agregar grupos a usuario/s se va a dar permisos para que el/los usuario/s puedan acceder a un determinado directorio/aplicación/servidor/etc.	
Precondiciones:	Que los usuarios existan y se encuentren habilitados Que el grupo exista	
Postcondiciones:	1) Usuarios con grupo/s agregados	
Prioridad:	Alta	
Frecuencia de uso:	Alta	
Flujo Normal:		
	Actor	Sistema
	1) El solicitante ingresa a la página del login del sistema	
	2) El solicitante ingresa sus credenciales en la página del login del sistema	3) El sistema verifica los datos y devuelve un token de sesión.
	4) El solicitante ingresa a la sección de usuarios, agregar grupo/s a usuario/s	

	5) El solicitante carga todos los usuarios separados por coma o también puede cargar un departamento completo para agregar posteriormente el grupo/s.	6) El sistema verifica la cuenta de los usuario y le pide que ingrese el/los grupo/s
	7) El solicitante carga todos los grupos separados por coma.	8) El sistema verifica los grupos.
		9) EL sistema genera la solicitud de aprobación a los distintos aprobadores de cada grupo y manda un mail al solicitante y usuario/s indicando que se gestionó el alta.
Flujos Alternativos:	Actor	Sistema
El solicitante ingresa un usuario erróneo o que se encuentra deshabilitado	5) El solicitante carga todos los usuarios separados por coma o también puede cargar un departamento completo para agregar posteriormente el grupo/s.	6) El sistema no puede obtener algún/os usuario/s debido a que es erróneo y le pide al solicitante que vuelva a ingresar el/los usuario/s.
Flujos Alternativos:	Actor	Sistema
El solicitante ingresa el/los grupo/s erróneos	7) El solicitante carga todos los grupos separados por coma.	6) El sistema no puede obtener algún/os grupos y le pide que los vuelva a cargar al solicitante.

Caso de Uso ID:	AU13	
Caso de Uso Nombre:	Quitar grupos a usuarios	
Creado por:	Luis López	Última actualización por: Luis López
Fecha Creación:	15/01/2015	Fecha última actualización: 18/01/2015
Actor:	Solicitante, sistema	
Descripción:	En la gestión de quitar grupos a usuario/s se va a sacar permisos para que el/los usuario/s no puedan acceder a un determinado directorio/aplicación/servidor/etc.	
Precondiciones:	Que los usuarios existan y se encuentren habilitados Que el grupo exista	
Postcondiciones:	1) Usuarios con grupo/s sacados	
Prioridad:	Alta	
Frecuencia de uso:	Alta	
Flujo Normal:		
	Actor	Sistema

	1) El solicitante ingresa a la página del login del sistema	
	2) El solicitante ingresa sus credenciales en la página del login del sistema	3) El sistema verifica los datos y devuelve un token de sesión.
	4) El solicitante ingresa a la sección de usuarios, sacar grupo/s a usuario/s	
	5) El solicitante carga todos los usuarios separados por coma o también puede cargar un departamento completo para sacar posteriormente el grupo/s.	6) El sistema verifica la cuenta de los usuario y le pide que ingrese el/los grupo/s
	7) El solicitante carga el grupo.	8) El sistema verifica los grupos.
		9) EL sistema genera la solicitud de aprobación a los distintos aprobadores de cada grupo y manda un mail al solicitante y usuario/s indicando que se gestionó la baja.
Flujos Alternativos:	Actor	Sistema

El solicitante ingresa un usuario erróneo o que se encuentra deshabilitado	5) El solicitante carga todos los usuarios separados por coma o también puede cargar un departamento completo para agregar posteriormente el grupo/s.	6) El sistema no puede obtener algún/os usuario/s debido a que es erróneo y le pide al solicitante que vuelva a ingresar el/los usuario/s.
Flujos Alternativos:	Actor	Sistema
El solicitante ingresa el/los grupo/s erróneos	7) El solicitante carga todos los grupos separados por coma.	6) El sistema no puede obtener algún/os grupos y le pide que los vuelva a cargar al solicitante.

Caso de Uso ID:	AU14	
Caso de Uso Nombre:	Crear Grupos	
Creado por:	German Casanova	Última actualización por: Luis López
Fecha Creación:	15/01/2015	Fecha última actualización: 18/01/2015
Actor:	Solicitante, sistema	

Descripción:	En la gestión de crear grupos se va agregar un nuevo grupo al dominio que va a cumplir con alguna funcionalidad específica.	
Precondiciones:	Que el nombre de grupo no se repita	
Postcondiciones:	Grupo creado	
Prioridad:	Media	
Frecuencia de uso:	Media	
Flujo Normal:		
	Actor	Sistema
	1) El solicitante ingresa a la página del login del sistema	
	2) El solicitante ingresa sus credenciales en la página del login del sistema	3) El sistema verifica los datos y devuelve un token de sesión.
	4) El solicitante ingresa a la sección de usuarios, crear grupos	
	5) El solicitante carga el nuevo grupo a crear con su respectivo	6) El sistema verifica que dicho grupo no se repita

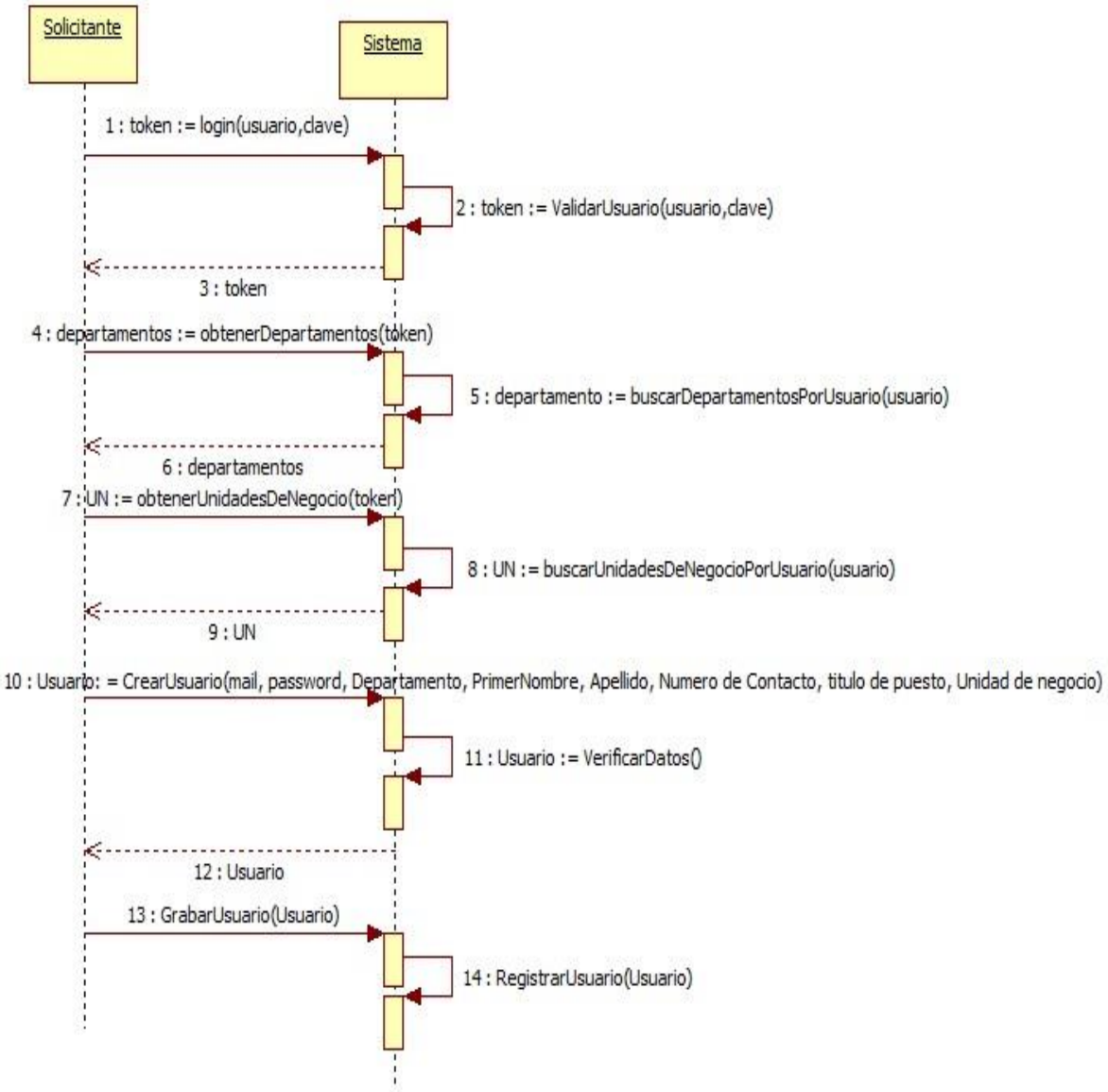
	aprobador	
		7) EL sistema crea el nuevo grupo con su respectivo aprobador.
Flujos Alternativos:	Actor	Sistema
El solicitante ingresa un grupo que ya existe.	5) El solicitante carga el nuevo grupo a crear con su respectivo aprobador	6) El sistema le indica al solicitante que el grupo ya existe.

Caso de Uso ID:	AU15	
Caso de Uso Nombre:	Borrar Grupos	
Creado por:	German Casanova	Última actualización por: Luis López
Fecha Creación:	15/01/2015	Fecha última actualización: 18/01/2015
Actor:	Solicitante, sistema	
Descripción:	En la gestión de remover grupos se va remover un grupo del LDAP/Active Directory	

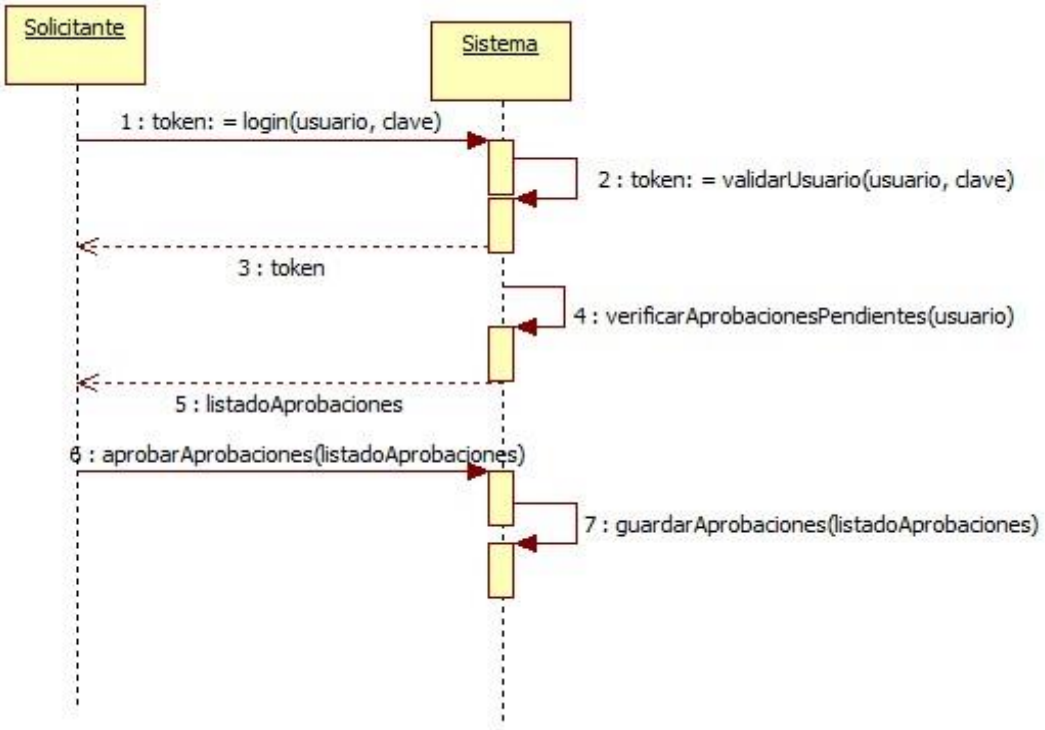
Precondiciones:	Que el nombre de grupo exista	
Postcondiciones:	Grupo removido.	
Prioridad:	Media	
Frecuencia de uso:	Media	
Flujo Normal:		
	Actor	Sistema
	1) El solicitante ingresa a la página del login del sistema	
	2) El solicitante ingresa sus credenciales en la página del login del sistema	3) El sistema verifica los datos y devuelve un token de sesión.
	4) El solicitante ingresa a la sección de usuarios, remover grupos	
	5) El solicitante carga el grupo a remover	6) El sistema verifica la existencia de dicho grupo
		7) EL sistema remueve el grupo
Flujos Alternativos:	Actor	Sistema
El solicitante ingresa un grupo que ya existe.	5) El solicitante carga el grupo a remover	6) El sistema le indica que el grupo no existe

Diagramas de Interacción Actores/Sistemas

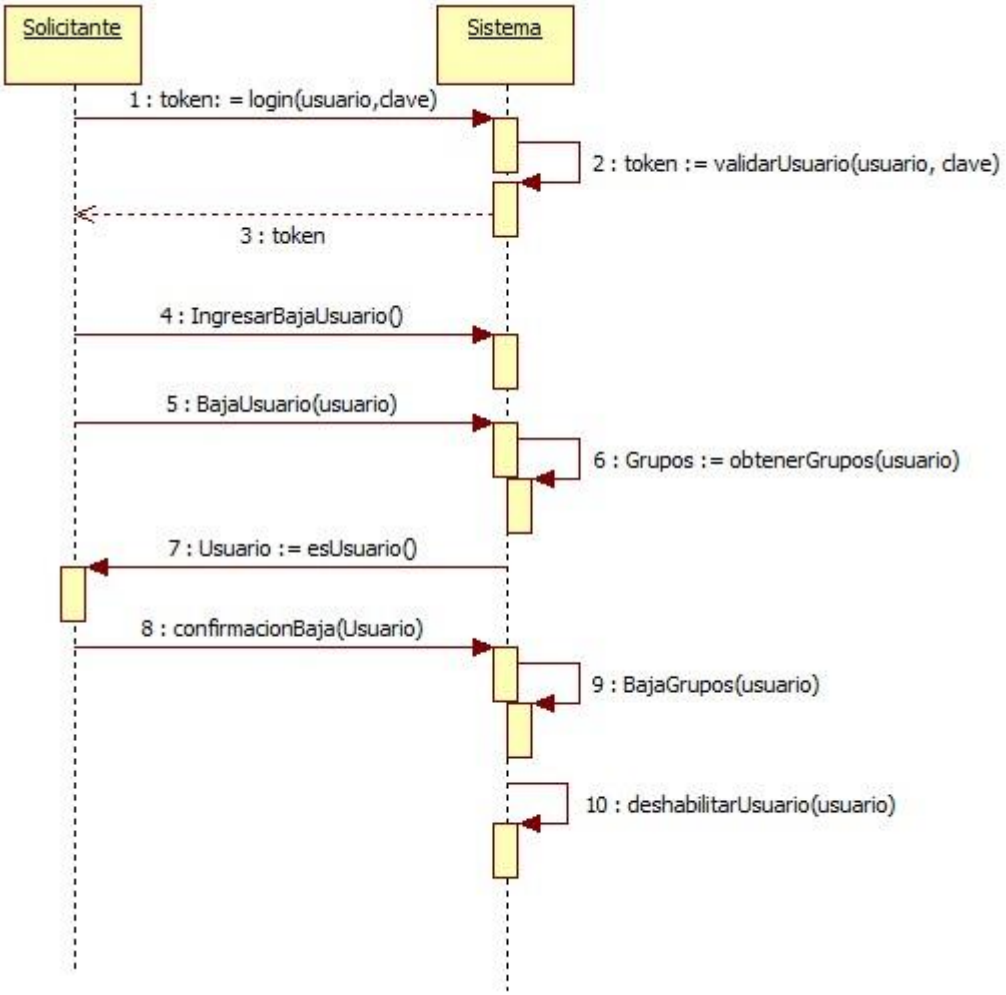
Alta Usuario



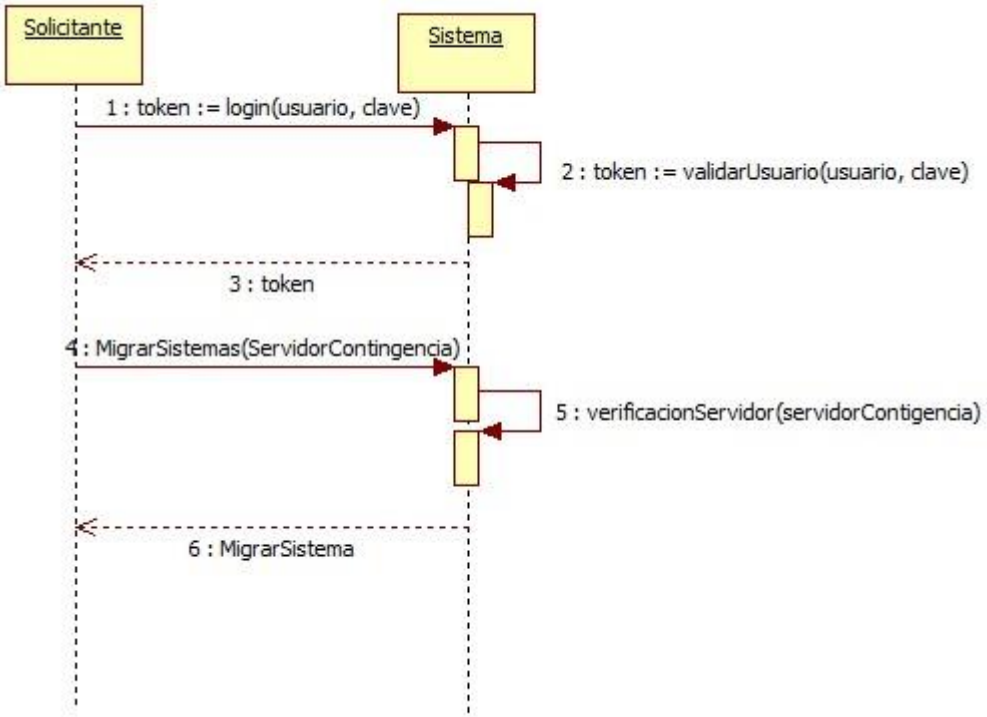
Aprobaciones Solicitudes



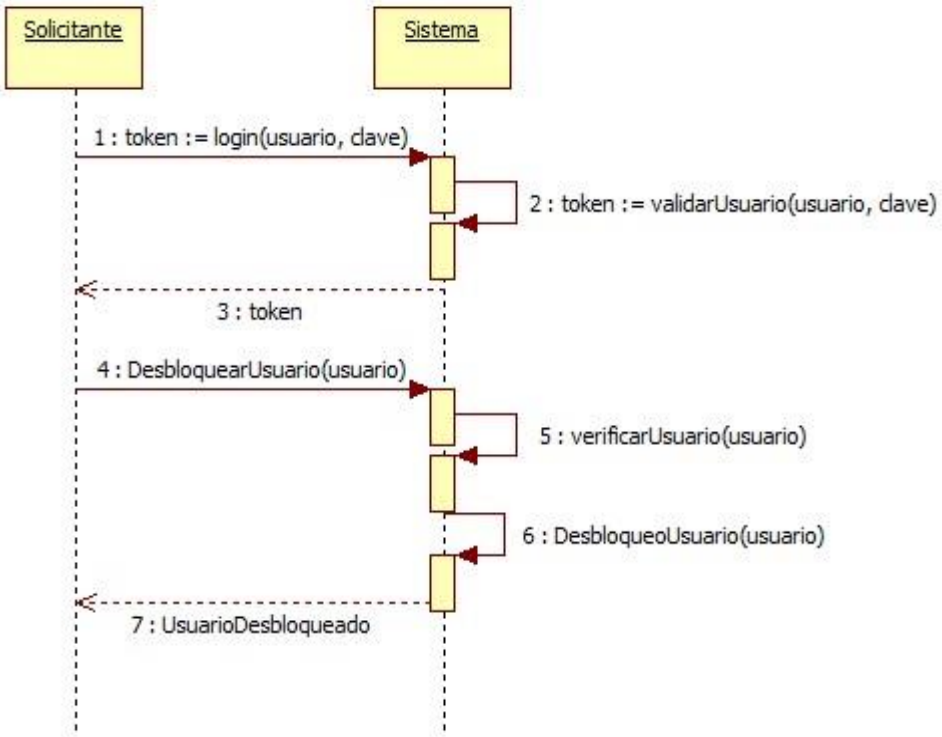
Baja Usuario



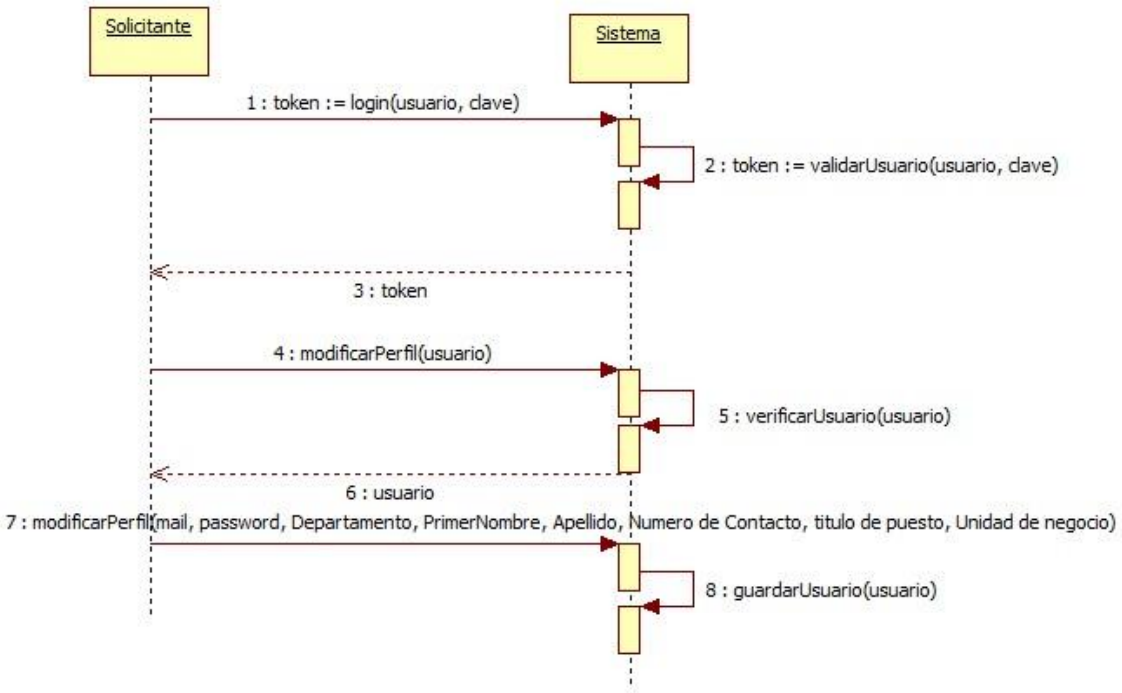
Migración sistema



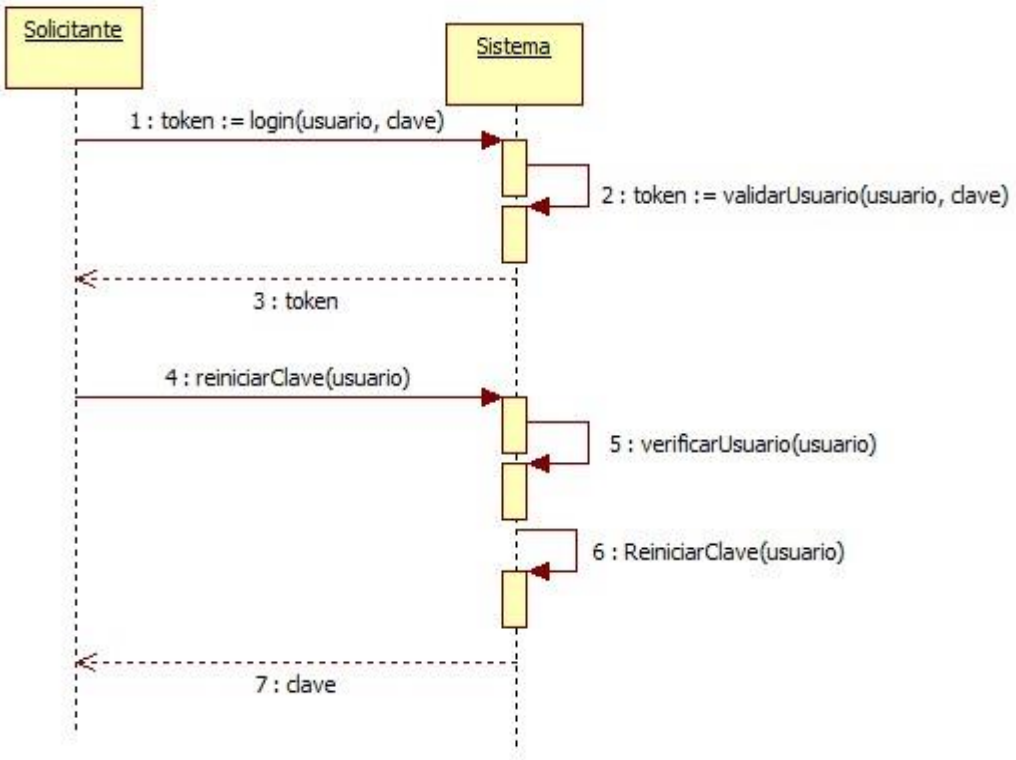
Desbloqueo Usuario



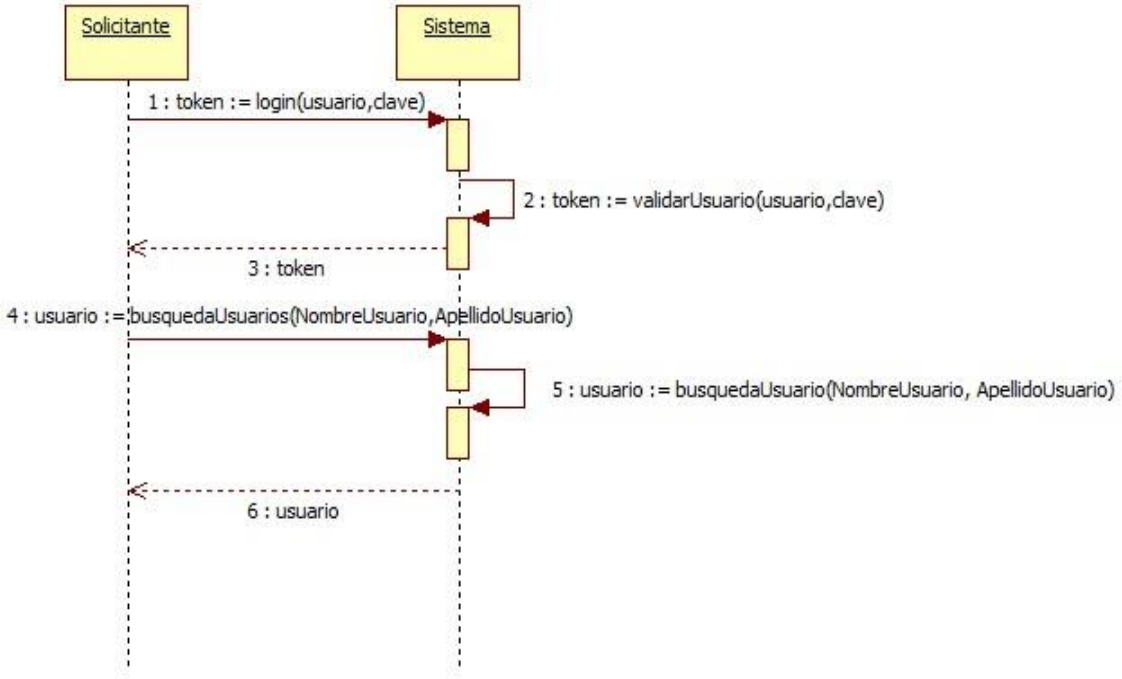
Modificación Perfil Usuario



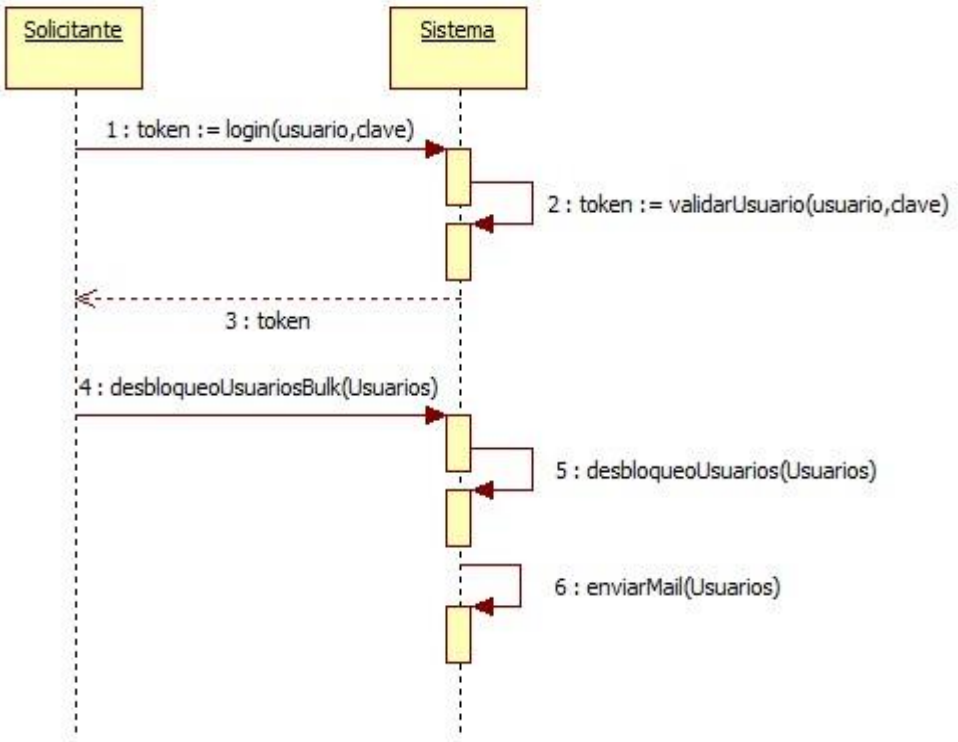
Reinicio Clave de Usuario



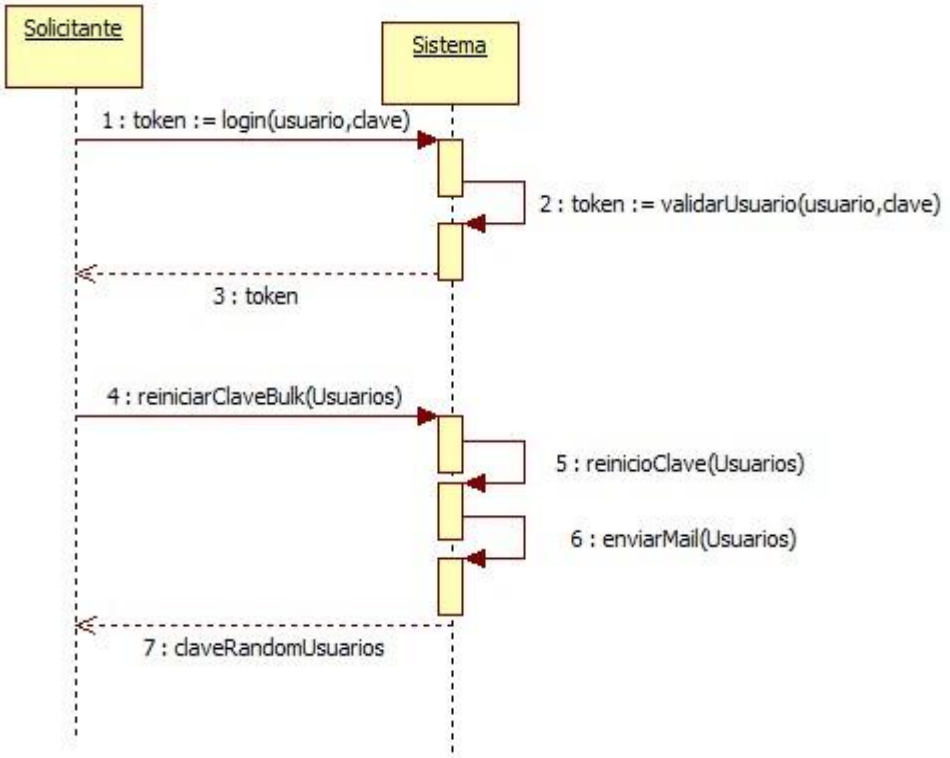
Búsqueda Usuario Cache



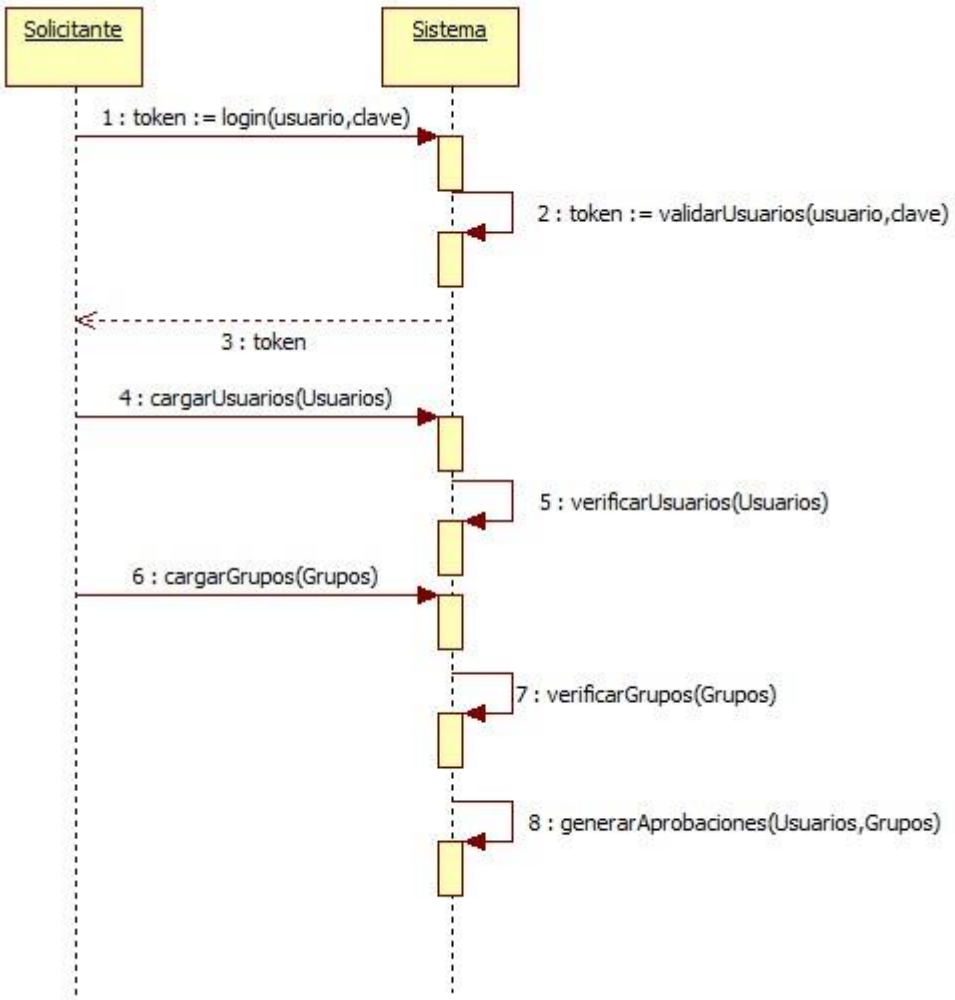
Desbloqueo de Usuarios en BULK



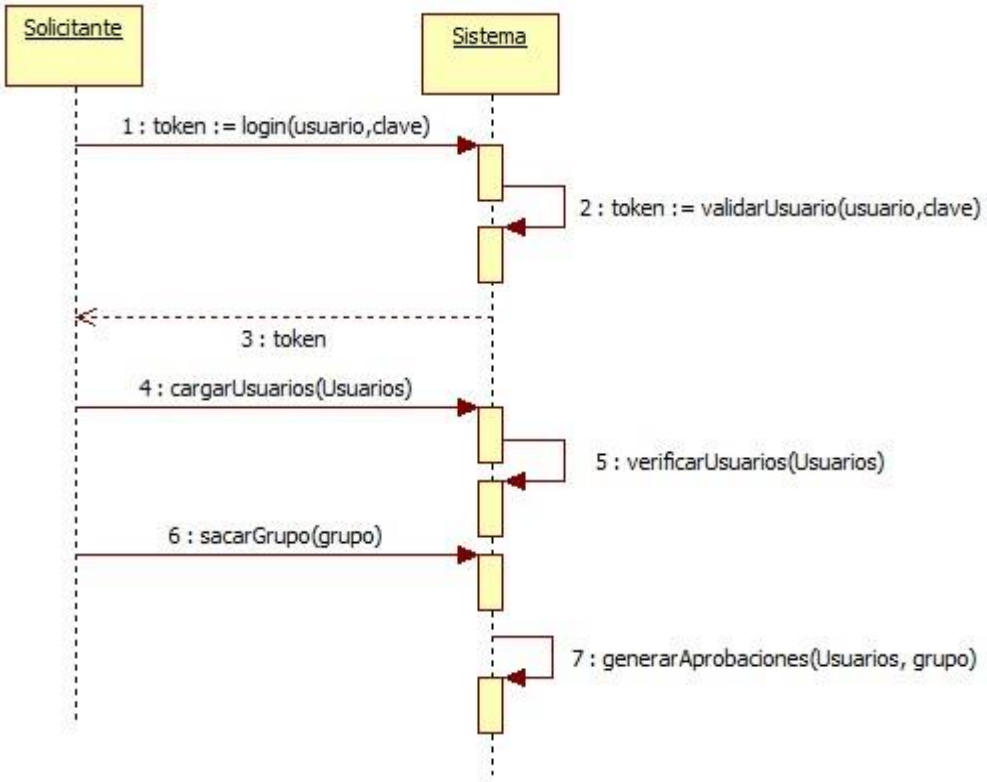
Reinicio de clave en BULK



Agregar Usuario/s a grupo/s



Sacar Usuarios de un grupo



Entrevista personal en CARGILL

Gerente de sistemas: Juan Pablo Barrirero (20 Ene 2014)

1. ¿Cuántas personas tiene a cargo para la gestión de altas/bajas, reset/desbloqueos de claves y modificaciones de perfil de Active Directory/LDAP?

Respuesta.- Hoy en día tenemos 40 personas que se dedican exclusivamente a realizar dichas tareas en las sucursales que tenemos asignadas. Dicho personal se encuentra distribuido parte en Argentina y otra parte en la India.

2. ¿Qué factores se basan para establecer la dimensión del equipo de soporte?

Respuesta.- Nos basamos en el tiempo promedio para realizar esas actividades y con la cantidad de tickets promedio de cada mes. En los reset/desbloqueos de contraseñas los usuarios muchas veces generan conflictos por los tiempos respuesta en dichas solicitudes brindando un servicio ineficiente a los empleados de la compañía.

3. ¿Dichos analistas están abocados exclusivamente a realizar tareas de Active Directory/LDAP?

Respuesta.- Sí, en los casos que tienen tiempo disponible, se ocupan de tareas de mantenimiento.

4. ¿Hoy en día cuánto costó les representa dicha área?

Respuesta.- Hoy en día nos representa un costo alto ya que se necesitan 40 analistas de Active Directory, 20 en Buenos Aires y 20 en la India. También contamos con 2 supervisores en su respectivo país. El salario bruto aproximado para los analistas en Argentina ronda por los 4000$ dólares mensuales y de 6000$ para los supervisores.

En la India el salario bruto de los analistas es de 2000$ dólares mensuales y el del supervisor de 3500$ mensuales.

5. ¿Tienen pensado algún proyecto para reducir ese costo del área?

Respuesta.- Actualmente hay un proyecto para automatizar dichos procesos con alguna herramienta/aplicación que nos sea útil. Todavía no hemos encontramos ninguna que podamos customizar al grado que nos sea útil para el área.

Entrevista personal de Mercado Libre

Supervisor seguridad informatica: Juan Berner (02 Mar 2014)

1. ¿Cuántas personas tiene a cargo para la gestión de altas/bajas, reset/desbloqueos de claves y modificaciones de perfil de Active Directory/LDAP?

   Respuesta.- Hoy en día el área de seguridad informática realiza dichas tareas y donde son 15 empleados que realizan dichas tareas, pero también están involucrados en muchos proyectos del área, y chequeo de vulnerabilidades sobre servidores y ciertas aplicaciones.

2. ¿Cuánto tiempo les demanda realizar tareas de Active Directory/LDAP?

Respuesta.- Actualmente la demanda de analistas es aproximadamente 5 analistas por día exclusivamente dedicadas a esas tareas.

3. ¿Hoy en día cuánto costo les representa dicha área?

Respuesta.- Hoy en día representa un costo alto ya que los 15 analistas del área de seguridad informática tienen una experiencia alta en el sector. Aproximadamente están en un salario bruto mensual de 25.000$ a 30.000$ pesos. Mensualmente representa un costo de 150.000$ pesos, a pesar de la cantidad de analistas el tiempo de respuesta es ineficaz.

4. ¿Tienen pensado algún proyecto para reducir ese costo del área?

Respuesta.- Actualmente tenemos un proyecto para automatizar dichos procesos. Los analistas están desarrollando un prototipo que les permita realizar acciones en simultáneo facilitando la carga de los datos.