

Conectando Hilos Digitales: Implementación Maestra de OpenLDAP para Autenticación Centralizada en un Mundo Aplicativo Diverso

Benjamín J. Salguero, José Martín Miret, Lourdes R. Lezcano, Sergio Zapata¹

¹ Dpto. Informática, Facultad Cs. Exactas, Univeridad Nacional del Nordeste, Corrientes, República Argentina

benjaminsalg01@gmail.com, martin.miret@gmail.com
lezcanolourdes778@gmail.com,
sergio.zapata@comunidad.unne.edu.ar

Abstract. En la actualidad, en todo tipo de organización, la infraestructura de los servicios informáticos y su calidad son cruciales a la hora de satisfacer las crecientes demandas de rendimiento, alcance y eficiencia de los mismos. Además, con el continuo crecimiento de las organizaciones se ha vuelto esencial mejorar la gestión de identidades, la autorización y la seguridad de acceso.

En este trabajo de investigación, realizado en colaboración como parte del curso de Redes de Datos en el cuarto año de la carrera de Licenciatura en Sistemas de Información (LSI), se enfoca en la implementación OpenLDAP, una solución basada en código abierto para abordar los mencionados desafíos de autenticación y autorización. Este software brinda una solución versátil que se adapta a las necesidades de las organizaciones independientemente de su tamaño.

Keywords: Autenticación, Autorización, Software Libre, OpenLDAP.

1 Introducción

El protocolo LDAP (Lightweight Directory Access Protocol) es un protocolo de acceso a directorios que se utiliza para acceder y gestionar información almacenada en un servicio de directorio. Su propósito principal es proporcionar un mecanismo estándar para buscar, recuperar, modificar y eliminar información en un directorio de manera eficiente y segura.

LDAP se utiliza ampliamente en entornos de red para autenticación, autorización y búsqueda de información de usuarios y recursos. Permite a los clientes acceder a los servicios de directorio que se ejecutan en el protocolo de control de transmisión/protocolo de Internet (TCP/IP) y gestionarlos.

El protocolo LDAP se basa en el modelo cliente-servidor, donde un cliente realiza solicitudes al servidor LDAP para buscar, recuperar o modificar información en el

2 Benjamín J.Salguero, José Martín Miret, Lourdes R. Lezcano, Sergio Zapata

directorío. El servidor LDAP almacena la información en forma de entradas, que contienen atributos y valores asociados.

LDAP es utilizado por varios servicios de directorío, como Microsoft Active Directory, Apache, Red Hat Directory Service y OpenLDAP. También es compatible con diferentes sistemas operativos, como Windows, Linux, Unix y macOS.

En resumen, el protocolo LDAP es un estándar utilizado para acceder y gestionar información en un servicio de directorío, proporcionando un mecanismo eficiente y seguro para buscar y modificar datos en un entorno de red.[1][2]

2 Instrumentación

Para el despliegue del estudio realizado se requiere disponer de las siguientes herramientas:

- Oracle VM VirtualBox: Es una plataforma de virtualización que permite crear máquinas virtuales donde se pueden instalar sistemas operativos. En este caso, para alojar el sistema operativo Ubuntu Server.
- Ubuntu Server 23.10: Es una distribución de Linux diseñada para servidores. En este caso, se elige como sistema base para el servidor LDAP. Dentro de este, se instalan OpenLDAP y OpenSSH que son componentes clave para el funcionamiento del servidor LDAP y la conectividad segura a través de SSH.
- PuTTY: Es un cliente SSH que se utiliza para establecer una conexión segura con el servidor LDAP a crear. Permite la administración de manera remota y segura.
- Apache Directory Studio: Es una herramienta de código abierto que facilita la administración y navegación de servidores LDAP. Se utiliza para configurar y gestionar dicho servidor.
- IntelliJ IDEA: Es un entorno de desarrollo integrado (IDE) utilizado para el desarrollo de aplicaciones JAVA. En este caso para crear un proyecto Spring Boot que actúa como cliente LDAP.

3 Desarrollo

Para el desarrollo del trabajo podemos dividir las tareas en cinco partes principales, la configuración del servidor de directorío activo, la configuración de usuarios y la creación de tres clientes LDAP, una aplicación de Spring, un servidor Apache, y una terminal host con Ubuntu Desktop.

3.1 Configuración del Servidor LDAP

Desde un primer momento, se contempla que ya se tiene instalado el sistema operativo Ubuntu Server 22.04 ejecutándose desde VirtualBox, por lo que no se contemplan los pasos de instalación de este.

Una vez que el sistema operativo esté corriendo, procedemos a conectarnos mediante SSH utilizando PuTTY e iniciamos sesión.

Lo primero que debemos realizar una vez autenticados es instalar el paquete correspondiente a OpenLDAP con el siguiente comando en la consola

```
$sudo apt install slapd ldap-utils
```

El paquete se descargará y nos pedirá que ingresemos una contraseña para el administrador del directorio LDAP:



Figura 1. Interfaz de configuración de OpenLDAP.

Luego de esto la instalación continuará y, una vez terminada, podremos comprobar que se instaló correctamente ingresando el comando:

```
$sudo systemctl status slapd
```

Con esto se nos mostrará un resultado como el siguiente, el cual indica que el estado del servicio es activo y en funcionamiento:



Figura 2. Estado de proceso de OpenLDAP.

Ahora debemos establecer un host y dominio para el servidor, el cual puede ser uno verdadero o en este caso, uno ficticio para uso local.

```
$sudo hostnamectl set-hostname ldap.redes2023.com
```

y podemos comprobar el cambio con:

```
$hostname
```

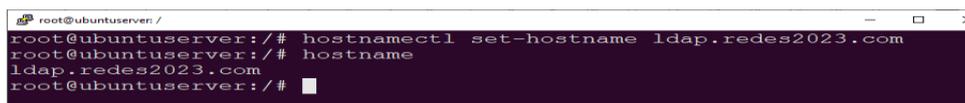
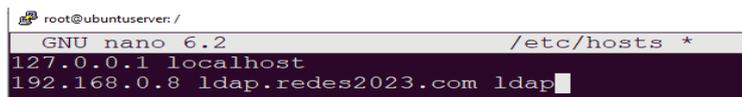


Figura 3. Configuración de hostname.

4 Benjamín J.Salguero, José Martín Miret, Lourdes R. Lezcano, Sergio Zapata

El paso siguiente es editar el archivo hosts y agregar nuestro servidor con el comando:

```
$sudo nano /etc/hosts
```



```
root@ubuntu: /
GNU nano 6.2 /etc/hosts *
127.0.0.1 localhost
192.168.0.8 ldap.redes2023.com ldap
```

Figura 4. Configuración del archivo hosts.

Ahora, debemos iniciar el asistente de configuración interactivo de OpenLDAP

```
$sudo dpkg-reconfigure slapd
```

Que nos abrirá una interfaz como la siguiente:

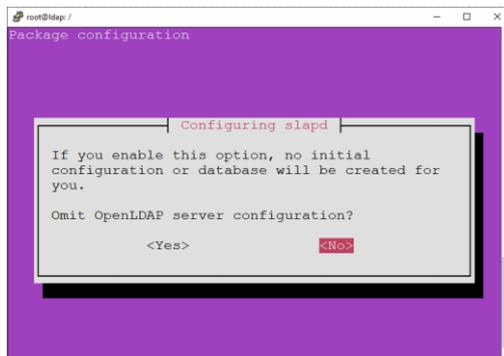


Figura 5. Interfaz de configuración de OpenLDAP.

Dentro de esta configuración, debemos responder a las siguientes preguntas con:

- Omitir la configuración del servidor OpenLDAP? -> NO
- DNS Domain name? -> redes2023.com
- Organization name? -> redes2023.com
- Administrator password? -> Elegimos una
- Eliminar la base de datos cuando slapd es purgado? -> NO
- Mover la base de datos anterior? -> SI

Una vez terminado, solo resta configurar el firewall del servidor con los siguientes comandos

```
$sudo ufw enable
$sudo ufw allow ldap
$sudo ufw allow ldaps
$sudo ufw allow ssh
$sudo ufw reload
$sudo ufw status
```

Con el último comando ingresado, podremos ver los puertos habilitados en el firewall.

```

root@ldap: /
root@ldap: /# ufw status
Status: active

To Action From
--
389 ALLOW Anywhere
636 ALLOW Anywhere
22/tcp ALLOW Anywhere
389 (v6) ALLOW Anywhere (v6)
636 (v6) ALLOW Anywhere (v6)
22/tcp (v6) ALLOW Anywhere (v6)

```

Figura 6. Puertos habilitados en el firewall de Ubuntu.

El puerto 389 se corresponde al utilizado por LDAP, mientras que el número 636 lo utiliza LDAPSecure. Por su parte, quien utiliza el puerto 22 es OpenSSH que lo utilizamos para conectarnos desde el PuTTY. Con esto, el servidor LDAP está completamente configurado.

Instalamos un editor de texto (paso opcional que puede ser reemplazado por otro editor a elección)

```
$sudo apt install nano
```

Y procedemos a editar los siguientes archivos de configuración:

```
$sudo nano /etc/ldap/ldap.conf
```

Descomentamos las siguientes líneas y las completamos con los datos correspondientes

```
BASE dc=redes2023,dc=com
URI ldap://192.168.0.8:389
```

```
$sudo nano /etc/nsswitch.conf
```

Y editamos también las siguientes líneas

```
passwd: files ldap
group: files ldap
shadow: files ldap
```

Procedemos a instalar un paquete de administración vía web (opcional)

```
$sudo apt install phpldapadmin
```

Editamos el archivo config.php buscando la línea correspondiente y reemplazando con la ip de nuestro servidor quedando de la siguiente manera:

6 Benjamín J.Salguero, José Martín Miret, Lourdes R. Lezcano, Sergio Zapata

```
$sudo nano /etc/phpldapadmin/config.php
```

```
$servers->setValue('server','host','192.168.0.175');  
$servers->setValue('server','base',array('dc=redes2023,dc=com'));
```

Instalamos los siguientes paquetes:

```
$sudo apt install libnss-ldap libpam-ldap
```

Dentro de esta configuración, debemos responder a las siguientes preguntas con:

```
-LDAP Server: ldap://192.168.0.8  
-Nombre del buscador base: dc=redes2023,dc=com  
-Version LDAP: 3  
-Hacer administrador root de la base de datos local?? -> SI  
-¿La base de datos LDAP requiere iniciar sesión? -> NO  
-Cuenta LDAP para root: cn=admin,dc=redes2023,dc=com  
-LDAP password: ...
```

A continuación, las siguientes configuraciones corresponden al uso de LDAP como servidor de autenticación para el host de ubuntu desktop.

```
$sudo nano /usr/share/pam-configs/mkhomedir
```

Escribimos lo siguiente:

Name: Create home directory on login

Default: yes

Priority: 900

Session-Type: Additional

Session: required pam_mkhomedir.so umask=0022 skel=/etc/skel

```
$sudo pam-auth-update
```

Seleccionamos la opción de “Crear directorio home al iniciar sesión”

```
$sudo nano /etc/pam.d/common-account
```

Agregamos lo siguiente al final del fichero:

```
account required pam_unix.so
```

```
$sudo nano /etc/pam.d/common-session
```

Agregamos lo siguiente al final del fichero:

```
session required pam_limits.so
```

3.2 Configuración de unidades organizacionales y usuarios

La creación de usuarios para ser administrados por el servidor LDAP puede realizarse de varias maneras, mediante la terminal o utilizando interfaces gráficas. En este trabajo se mostrará estos casos, creando un grupo en cada uno con sus usuarios.

El árbol de directorio final creado tendrá la siguiente forma:

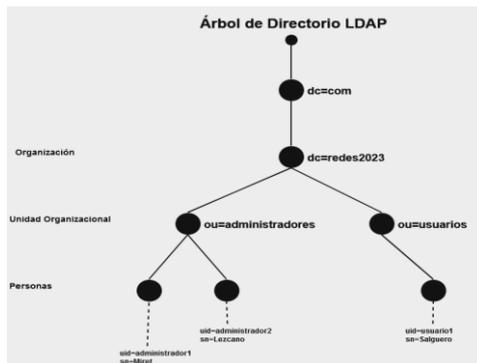


Figura 7. Estructura del árbol de directorios a ser creado.

Durante la configuración de credenciales, se utilizan siglas cuyas correspondencias son:
 dc = Domain Component. Representa partes de un nombre de dominio.
 ou = Organizational Unit.

Representa unidades organizativas, puede utilizarse también para hacer referencias a grupos.

uid = User ID. Es el identificador único de un usuario.

cn = Common Name. Es el nombre de una persona.

sn = Surname. Es el apellido de una persona.

objectClass = define las características de una entrada de directorio.

3.2.1 Configuración por terminal

Lo primero que debemos hacer es crear un archivo de configuración con el comando

```
$sudo nano /etc/ldap/ou.ldif
```

Y cargamos los datos para crear un grupo organizacional

```

root@ldap: /
GNU nano 6.2 /etc/ldap/ou.ldif
dn: ou=administradores,dc=redes2023,dc=com
objectClass: organizationalUnit
ou: administradores
  
```

Figura 8. Configuración de un grupo organizacional.

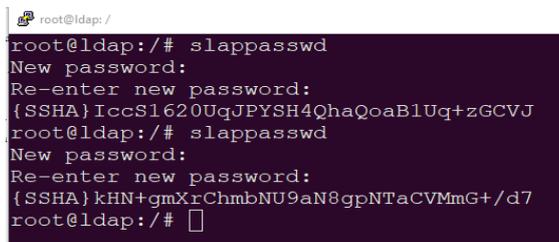
Con el siguiente comando cargamos la configuración LDAP contenida en el archivo

8 Benjamín J.Salguero, José Martín Miret, Lourdes R. Lezcano, Sergio Zapata

```
$sudo ldapadd -x -D cn=admin,dc=redes2023,dc=com -W -f /etc/ldap/ou.ldif
```

Con esto, el grupo de administradores estará creado y ahora solamente debemos añadir los usuarios. Para cada usuario, debemos generar una clave cifrada para luego ser utilizada al crear el mismo.

```
$sudo slappasswd
```

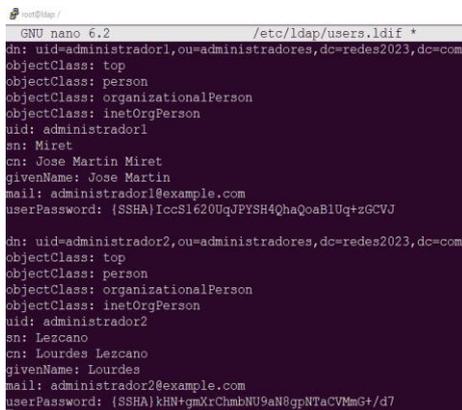


```
root@ldap: /
root@ldap: /# slappasswd
New password:
Re-enter new password:
{SSHA}IccS1620UqJPYSH4QhaQoaB1Uq+zGCVJ
root@ldap: /# slappasswd
New password:
Re-enter new password:
{SSHA}kHN+gmXrChmbNU9aN8gpNTaCVMmG+/d7
root@ldap: /#
```

Figura 9. Creación de contraseñas cifradas.

Similar a los pasos realizados con los grupos, creamos un archivo de configuración y colocamos los datos del usuario con las contraseñas creadas en el paso anterior.

```
$sudo nano /etc/ldap/users.ldif
```



```
GNU nano 6.2 /etc/ldap/users.ldif *
dn: uid=adminstrador1,ou=administradores,dc=redes2023,dc=com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
uid: administrador1
sn: Miret
cn: Jose Martin Miret
givenName: Jose Martin
mail: administrador1@example.com
userPassword: {SSHA}IccS1620UqJPYSH4QhaQoaB1Uq+zGCVJ

dn: uid=adminstrador2,ou=administradores,dc=redes2023,dc=com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
uid: administrador2
sn: Lezcano
cn: Lourdes Lezcano
givenName: Lourdes
mail: administrador2@example.com
userPassword: {SSHA}kHN+gmXrChmbNU9aN8gpNTaCVMmG+/d7
```

Figura 10. Creación de usuarios.

Y luego simplemente añadimos los usuario a la configuración

```
$sudo ldapadd -x -D cn=admin,dc=redes2023,dc=com -W -f /etc/ldap/users.ldif
```

Con los siguientes comandos podemos comprobar que los usuarios se agregaron correctamente

```
$sudo ldapwhoami -x -D "uid=adminstrador1,ou=administra-
dores,dc=redes2023,dc=com" -W
```

*\$sudo ldapwhoami -x -D "uid=administrador2,ou=administra-
dores,dc=redes2023,dc=com" -W*



```

root@ldap:~# sudo ldapwhoami -x -D "uid=administrador1,ou=administadores,dc=redes2023,dc=com" -W
Enter LDAP Password:
dn:uid=administrador1,ou=administadores,dc=redes2023,dc=com
root@ldap:~#
root@ldap:~# sudo ldapwhoami -x -D "uid=administrador2,ou=administadores,dc=redes2023,dc=com" -W
Enter LDAP Password:
ldap bind: Invalid credentials (49)
root@ldap:~#

```

Figura 11. Comprobación de usuarios creados.

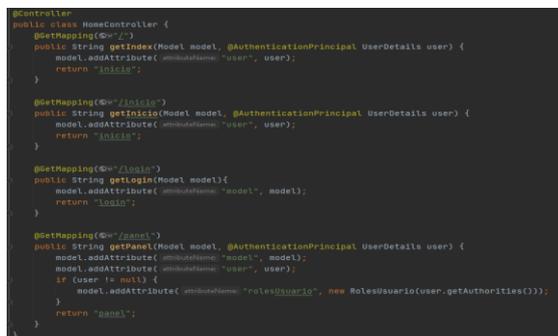
Para el caso del administrador1 se ingresó la contraseña correcta, y para el caso del administrador2 una incorrecta.

3.3 Codificación del cliente LDAP en Spring

Para crear el cliente, se debe generar un proyecto de Spring Boot 3 con las siguientes dependencias:

- Spring Boot Starter Web
- Spring Boot Starter Security
- Spring LDAP Core
- Spring Security LDAP
- Unboundid LDAP SDK
- Thymeleaf

Primero debemos crear el controlador y las vistas correspondientes.



```

@Controller
public class HomeController {
    @GetMapping("/")
    public String getIndex(Model model, @AuthenticationPrincipal UserDetails user) {
        model.addAttribute("username", user);
        return "inicio";
    }

    @GetMapping("/inicio")
    public String getInicio(Model model, @AuthenticationPrincipal UserDetails user) {
        model.addAttribute("username", user);
        return "inicio";
    }

    @GetMapping("/login")
    public String getLogin(Model model){
        model.addAttribute("username", model);
        return "login";
    }

    @GetMapping("/panel")
    public String getPanel(Model model, @AuthenticationPrincipal UserDetails user) {
        model.addAttribute("username", model);
        model.addAttribute("user", user);
        if (user != null) {
            model.addAttribute("roles", new RolesUserIo(user.getAuthorities()));
        }
        return "panel";
    }
}

```

Figura 12. Controlador del proyecto Spring Boot.

Implementamos la clase WebSecurityConfig, donde se realizará la configuración de acceso y autenticación.

10 Benjamín J.Salguero, José Martín Miret, Lourdes R. Lezcano, Sergio Zapata

```

@Configuration
public class WebSecurityConfig {
    @Bean
    SecurityFilterChain web(HttpSecurity http) throws Exception {
        http
            .csrf((csrf) -> csrf.disable()) //Desactiva el CSRF
            .authorizeHttpRequests(authorize -> authorize
                //Usuario no logueado
                .requestMatchers(anonymous).anonymous()
                //Cualquier usuario
                .requestMatchers(anonymous).requestMatchers("/css/**", "/js/**", "/inicio", "/error", "/logout").permitAll()
                //Solo administradores
                .requestMatchers(anonymous).hasAnyRole("ADMIN-VENTAS", "ADMIN-CATALOGO")
                //Cualquier otra dirección es prohibida automáticamente
                .anyRequest().denyAll()
            )
        //Config del login y logout
        .formLogin(form -> form
            .loginPage("/login")
            .defaultSuccessUrl("/inicio")
            .loginProcessingUrl("/login")
            .failureUrl("/login/error")
        )
        .logout()
        return http.build();
    }
}

```

Figura 13. Configuración de Spring Security.

En la configuración de acceso ingresada, se estableció que cualquier usuario existente en el servidor pueda iniciar sesión, mientras que a la sección “panel” solo puedan ingresar aquellos que sean administradores. Es decir, aquellos pertenecientes a las unidades organizacionales “admin-ventas” y “admin-catalogo”.

```

@Autowired
public void configure(AuthenticationManagerBuilder auth) throws Exception {
    auth
        .ldapAuthentication()
        //Cualquier usuario puede iniciar sesión
        .userSearchFilter("(uid={0})")
        .contextSource()
        //IP y dominio del servidor LDAP
        .url("ldap://192.168.0.22:389/dc=redes2023,dc=com");
}

```

Figura 14. Configuración de Spring Security.

Dentro de la misma clase `WebSecurityConfig`, se establece también la dirección del servidor LDAP donde se verificarán las credenciales ingresadas durante el inicio de sesión. Para eso, se colorea la IP y el nombre de dominio del Active Directory que se quiere utilizar.

El código fuente completo puede encontrarse en el repositorio `Spring-LDAP-Client` subido a Github[5].

3.4 Cliente LDAP Apache

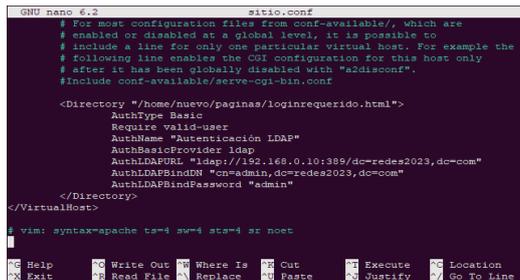
Si queremos modificar un servidor Apache en Ubuntu server existente para que este pueda utilizar credenciales provenientes de nuestro servidor LDAP, debemos instalar la librería correspondiente con el siguiente comando

```
$sudo apt install apache2 libapache2-mod-authnz-external libapache2-mod-authnz-ldap
```

Luego, debemos activar la librería instalada con:

```
$sudo a2enmod authnz_ldap
```

Una vez hecho esto, solamente resta modificar el archivo de configuración para que cuando el usuario quiera acceder a la página “loginrequerido.html”, este deba ingresar las credenciales de un usuario válido existente en el Active Directory configurado anteriormente.



```

GNU nano 6.2          sitio.conf
# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#include conf-available/serve-cgi-bin.conf

<Directory "/home/nuevo/paginas/loginrequerido.html">
    AuthType Basic
    Require valid-user
    AuthName "Autenticación LDAP"
    AuthBasicProvider ldap
    AuthLDAPURL "ldap://192.168.0.10:389/dc=redes2023,dc=com"
    AuthLDAPBindDN "cn=admin,dc=redes2023,dc=com"
    AuthLDAPBindPassword "admin"
</Directory>
</VirtualHost>
^ vim: syntax=apache ts=4 sw=4 sts=4 sr noet
  
```

Figura 15. Configuración del sitio web Apache.

Una vez hecha la configuración, debemos reiniciar el servicio de apache

```
$sudo systemctl restart apache2
```

3.5 Cliente Ubuntu Desktop

Iniciamos sesión en nuestro ubuntu desktop, abrimos la terminal como root y procedemos con la configuración:

```
$apt install libpam-ldap libnss-ldap nss-updatedb nscd ldap-utils -y
```

Dentro de esta configuración, debemos responder a las siguientes preguntas con:

- LDAP Server: ldap://192.168.0.8
- Nombre del buscador base: dc=redes2023,dc=com
- Version LDAP: 3
- Hacer administrador root de la base de datos local?? -> SI
- ¿La base de datos LDAP requiere iniciar sesión? -> NO
- Cuenta LDAP para root: cn=admin,dc=redes2023,dc=com
- LDAP password: ...

```
$apt install slapd
```

```
$dpkg-reconfigure slapd
```

Dentro de esta configuración, debemos responder a las siguientes preguntas con:

- Omitir la configuración del servidor OpenLDAP? -> NO
- DNS Domain name? -> redes2023.com
- Organization name? -> redes2023.com

12 Benjamín J.Salguero, José Martín Miret, Lourdes R. Lezcano, Sergio Zapata

- Administrator password? -> Elegimos una
- Eliminar la base de datos cuando slapd es purgado? -> NO
- Mover la base de datos anterior? -> SI

Procedemos a editar los siguientes archivos de configuración:

```
$sudo nano /etc/ldap/ldap.conf
```

Descomentamos las siguientes líneas y completamos con los datos correspondientes

```
BASE dc=redes2023,dc=com
```

```
URI ldap://192.168.0.8:389
```

```
$sudo nano /etc/nsswitch.conf
```

Cambiamos las siguientes líneas

```
passwd: files ldap
```

```
group: files ldap
```

```
shadow: files ldap
```

```
$nss-updatedb ldap
```

```
$sudo nano /usr/share/pam-configs/mkhomedir
```

Escribimos lo siguiente:

```
Name: Create home directory on login
```

```
Default: yes
```

```
Priority: 900
```

```
Session-Type: Additional
```

```
Session: required pam_mkhomedir.so umask=0022 skel=/etc/skel
```

```
$sudo nano /etc/pam.d/common-session
```

Agregamos lo siguiente al inicio del fichero en la primera línea:

```
session required pam_mkhomedir.so umask=0022 skel=/etc/skel
```

```
$sudo pam-auth-update
```

Seleccionamos la opción de “Crear directorio home al iniciar sesión”

```
$reboot
```

4 Resultados

4.1 Inicio de sesión en aplicación JAVA

Cuando ejecutamos el proyecto de Spring Boot, podemos ingresar desde el navegador al sitio web creado. En un primer momento cualquier usuario puede ingresar al inicio de manera normal.

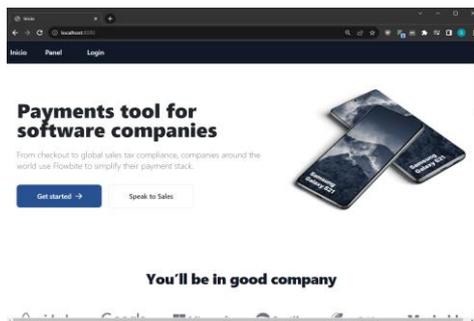


Figura 16. Acceso a la página de inicio.

Pero cuando se intenta ingresar a la página de panel aparece un formulario de inicio de sesión, donde debemos ingresar el usuario y contraseña de algún registro existente en el servidor OpenLDAP.

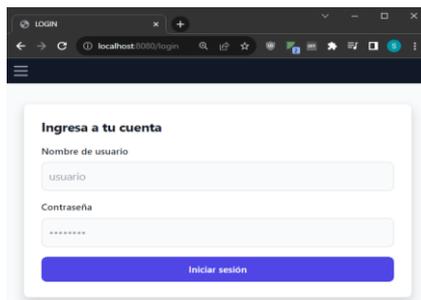


Figura 17. Formulario de inicio de sesión.

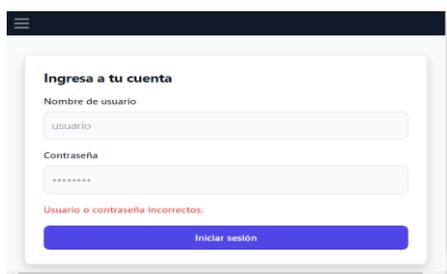


Figura 18. Intento de inicio de sesión no exitoso.

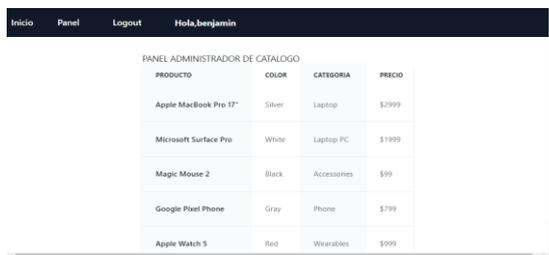
Cuando ingresamos las credenciales de un usuario administrador este puede acceder al panel de administradores, el cual cambiará el contenido según sea el rol del administrador que inició sesión.

14 Benjamín J.Salguero, José Martín Miret, Lourdes R. Lezcano, Sergio Zapata



NRO COMPONENTE	DNI	NOMBRE Y APELLIDO	MONTO
001	12345678	Juan Perez	\$500.00
002	87654321	Maria Gonzalez	\$750.00
003	55555555	Carlos Rodriguez	\$1200.00

Figura 19. Acceso al panel con usuario administrador de ventas.



PRODUCTO	COLOR	CATEGORIA	PRECIO
Apple MacBook Pro 17"	Silver	Laptop	\$2999
Microsoft Surface Pro	White	Laptop PC	\$1999
Magic Mouse 2	Black	Accessories	\$99
Google Pixel Phone	Gray	Phone	\$799
Apple Watch 5	Red	Wearables	\$599

Figura 20. Acceso al panel con usuario administrador de catálogo.

4.2 Inicio de sesión en servidor web Apache

Si ingresamos desde el navegador al servidor Apache e ingresamos a la página que configuramos para que pida las credenciales contenidas en el servidor LDAP, nos encontramos con el siguiente cuadro emergente.

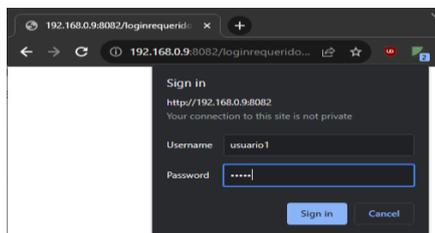


Figura 21. Intento de inicio de sesión con un usuario permitido.

Si ingresamos el usuario y contraseña de cualquier entrada de directorio, podemos iniciar sesión y ver la página protegida.



Redes de datos 2023

Login requerido para ver esta pagina

Figura 22. Inicio de sesión exitoso.

4.3 Inicio de sesión en ubuntu desktop

Al iniciar nuestro host con Ubuntu Desktop, no veremos ninguno de los usuarios creados en nuestro servidor con OpenLDAP por lo que debemos proceder con la opción de “No está en la lista?”, lo cual nos permite iniciar sesión escribiendo el usuario y el password, lo importante acá es iniciar sesión con una cuenta creada para tal efecto es decir una cuenta Samba. Una vez hecho esto veremos una leyenda abajo que nos avisará que se está creando el directorio home para dicho usuario, y al cabo de unos segundos iniciará sesión, con el usuario provisto, cosa que se puede comprobar abriendo la terminal.



```
usuarios-mmiret@redes-VirtualBox: ~  
usuarios-mmiret@redes-VirtualBox:~$ whoami  
usuarios-mmiret  
usuarios-mmiret@redes-VirtualBox:~$
```

Figura 23. Comprobación de usuario desde Ubuntu desktop.

5 Discusión

Luego de haber implementado tanto el servidor como el cliente LDAP y conectarlos, podemos hablar sobre las ventajas y desventajas de utilizar esta técnica para la autenticación y autorización de usuarios.

Ventajas

La principal ventaja es que este tipo de implementación permite centralizar la autenticación de usuarios en un solo lugar, lo que simplifica la administración de cuentas de usuario y contraseñas. Además, permite la posibilidad de utilizar opciones de seguridad como el cifrado SSL para garantizar que las contraseñas de los usuarios se transmitan de manera segura por la red. [3]

En cuanto a la escalabilidad, OpenLDAP permite manejar una gran cantidad de usuarios y grupos sin mayores problemas. De igual manera, la principal ventaja es que es compatible con una gran variedad de servicios y aplicaciones que admiten la autenticación mediante LDAP, lo que facilita la integración con otros sistemas.

Desventajas

Una de las desventajas puede ser que la configuración inicial de OpenLDAP puede ser compleja. Además, aunque muchas aplicaciones y servicios admitan la autenticación mediante directorios, es posible que algunas no sean compatibles y requieran configuraciones adicionales.

6 Conclusiones

Este trabajo de investigación, donde se configuró un Active Directory utilizando OpenLDAP y su integración con una aplicación de Spring para consumir las

16 Benjamín J.Salguero, José Martín Miret, Lourdes R. Lezcano, Sergio Zapata

credenciales creadas, resultó en una experiencia valiosa y la creación de un sistema sólido de autenticación.

Este proyecto demostró la versatilidad y potencia que posee OpenLDAP a la hora de crear un sistema de autenticación en un entorno basado en Linux junto a la capacidad que este brinda para configurar un Active directory con software de código abierto, presentando la ventaja en términos de costos.

Se pudo observar las diferencias que existen entre la configuración del directorio LDAP desde la terminal del servidor vía SSH y utilizando interfaces gráficas como Apache Directory Studio o PhpLdapAdmin, que presentan las mismas funcionalidades pero de manera mucho más intuitiva, especialmente el último caso. Además, la integración exitosa de múltiples clientes como son el proyecto Spring Boot, el servidor Apache y el Ubuntu Desktop demuestran la capacidad de interoperabilidad de sistemas donde se pueden crear varias aplicaciones diferentes que consuman el mismo directorio. Es decir, aunque todos los clientes sean totalmente diferentes estos pueden consumir el mismo servidor LDAP y utilizar las mismas credenciales para que los usuarios puedan autenticarse sin ningún problema.

References

1. Red Hat. (2023, 28 de septiembre). ¿Qué es la autenticación LDAP? Red Hat Customer Portal. Recuperado de <https://www.redhat.com/es/topics/security/what-is-ldap-authentication>
2. RedesZone.net. (2023, 28 de octubre). ¿Qué es LDAP? Funcionamiento y características. RedesZone.net. Recuperado de <https://www.redeszone.net/tutoriales/servidores/que-es-ldap-funcionamiento/>
3. OpenLDAP. OpenLDAP Faq-o-Matic <https://www.openldap.org/faq/data/cache/1.html>
4. Manuel Cabrera Caballero. (2022, 28 de marzo). Youtube. Instalación y configuración en Ubuntu Server. <https://www.youtube.com/watch?v=ZzApLI3UH68>
5. Spring. Authenticating a User with LDAP. <https://spring.io/guides/gs/authenticating-ldap/>
6. Repositorio código fuente Spring-LDAP-Client. <https://github.com/BenjaminSalg/Spring-LDAP-Client>
7. Ldap OpenLdap Active Directory <http://profesores.fi-b.unam.mx/yasmine/LDAP.pdf>
8. ProngerTV. (2023, 27 de febrero). Youtube. Cómo instalar y configurar LDAP Server y Cliente Ubuntu 20.04 - Tutorial 2023 <https://www.youtube.com/watch?v=oJBHbLUMSGY>