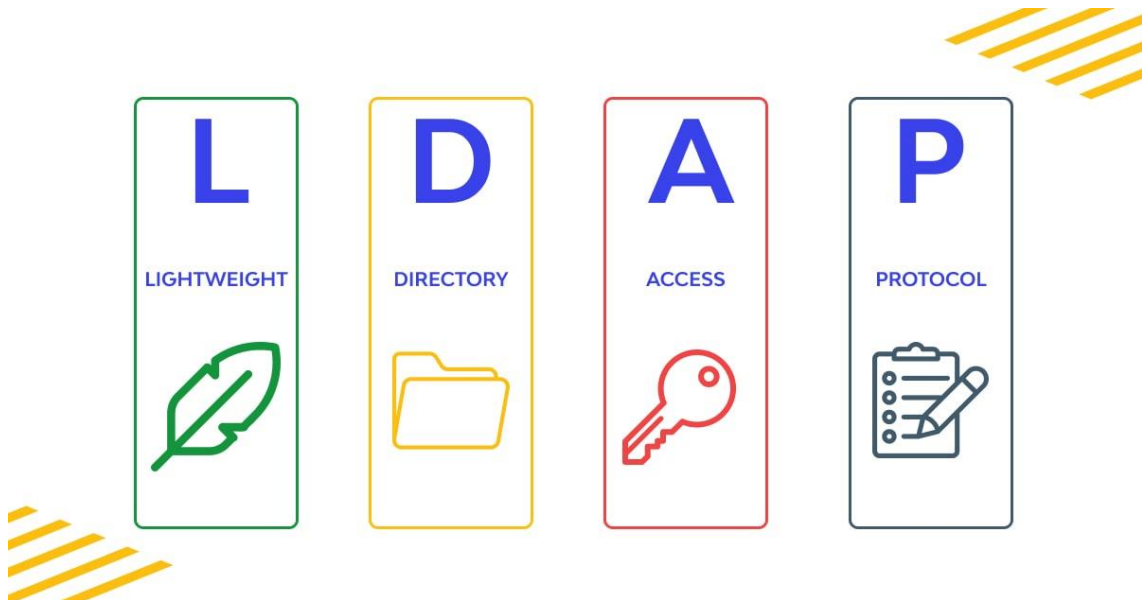


Configuración de Repositorio LDAP para Login con ExpressJ



Creo el fichero de configuración LDAP en el directorio raíz:

The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with a root directory containing files like .vscode, bin, public, routes, index.js, login.js, users.js, views, error.ejs, index.ejs, login.ejs, .gitignore, app.js, LICENSE, package-lock.json, package.json, README.md, and test-server.ldif. The code editor shows the content of test-server.ldif:

```
1 dn: dc=iesalixar,dc=org
2 objectclass: top
3 objectclass: domain
4 objectclass: extensibleObject
5 dc: iesalixar
6
7 dn: ou=alumnos,dc=iesalixar,dc=org
8 objectclass: top
9 objectclass: organizationalUnit
10 ou: alumnos
11
12
13
```

Añado los registros con el siguiente comando:

Primero el comando para copiarlo: `docker cp .\test-server.ldif ldap:/tmp/`

Segundo el comando para poder añadirlo:

```
docker exec ldap ldapadd -x -H ldap://localhost -D "cn=admin,dc=iesalixar,dc=org" -f /tmp/user.ldif -w passiesalixar
```

```
PS C:\Users\Manuel\Documents\2 DAW\Despliegue de Aplicaciones Web\UD5\T1-Configuración de Repositorio LDAP para Login con Express JS> docker exec ldap ldapadd -x -H ldap://localhost -D "cn=admin,dc=iesalixar,dc=org" -f /tmp/test-server.ldif -w passiesalixar
ldap_add: Already exists (68)
adding new entry "dc=iesalixar,dc=org"
```



```
login.js test-server.ldif X
test-server.ldif
1 dn: ou=alumnos,dc=iesalixar,dc=org
2 objectclass: top
3 objectclass: organizationalUnit
4 ou: alumnos
5
6 dn: uid=alum.mjaer.2021,ou=alumnos,dc=iesalixar,dc=org
7 objectclass: top
8 objectclass: person
9 objectclass: organizationalPerson
10 objectclass: inetOrgPerson
11 cn: Manuel Jaén Rojas
12 sn: Jaén
13 uid: alum.mjaer.2021
14 userPassword: usermanujaen
15
```

Y para poder revisar que esta todo totalmente insertado correctamente, utilizamos el siguiente comando para que lo muestre:

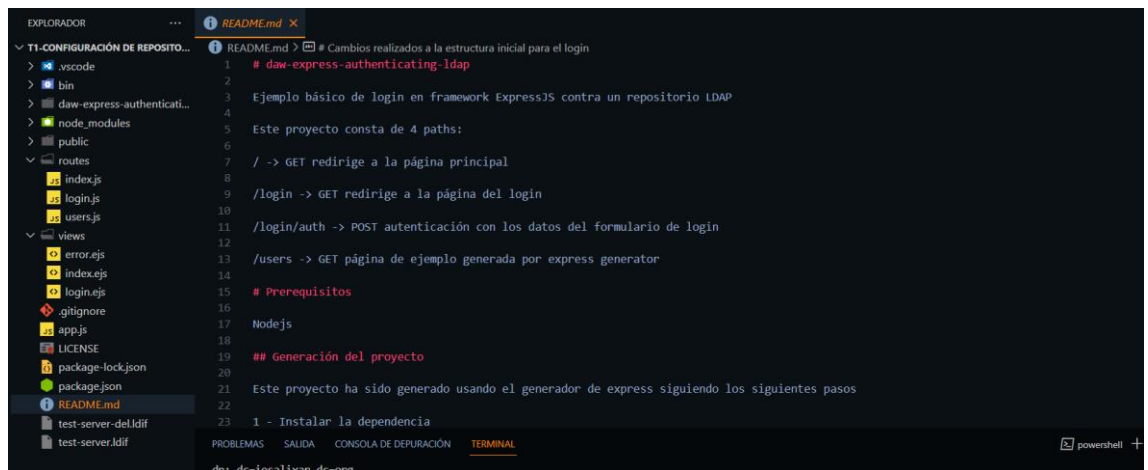
```
docker exec ldap ldapsearch -x -H ldap://localhost:389 -b dc=iesalixar,dc=org -D "cn=admin,dc=iesalixar,dc=org" -w passiesalixar
```

```
PS C:\Users\Manuel\Documents\2 DAW\Despliegue de Aplicaciones Web\UD5\T1-Configuración de Repositorio LDAP para Login con Express JS> docker exec ldap ldapsearch -x -H lda
p://localhost:389 -b dc=iesalixar,dc=org -D "cn=admin,dc=iesalixar,dc=org" -w passiesalixar
# extended LDIF
#
# LDAPv3
# base <dc=iesalixar,dc=org> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#
# iesalixar.org
dn: dc=iesalixar,dc=org
objectClass: top
objectClass: dcObject
# alumnos, iesalixar.org
dn: ou=alumnos,dc=iesalixar,dc=org
objectClass: top
objectClass: organizationalUnit
ou: alumnos
# alum.mjaer.2021, alumnos, iesalixar.org
dn: uid=alum.mjaer.2021,ou=alumnos,dc=iesalixar,dc=org
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
cn: TWFudmVsIEphw6luIFJvamFz
sn: SmFDbG94=
uid: alum.mjaer.2021
userPassword:: dXNlcm1hbnVqYXVw
# search result
search: 2
result: 0 Success
# numResponses: 4
# numEntries: 3
```

Implementar el login:

Primero descargamos [el repositorio que contiene ya el proyecto expressjs](#) preparado para implementar el login:

1. Sigue las instrucciones del README.MD del proyecto para instalar dependencias y arrancar el proyecto.



```
EXPLORADOR  ...  README.md X
T1-CONFIGURACIÓN DE REPOSITO...  README.md > # Cambios realizados a la estructura inicial para el login
1  # daw-express-authenticating-ldap
2
3  Ejemplo básico de login en framework ExpressJS contra un repositorio LDAP
4
5  Este proyecto consta de 4 paths:
6
7  / -> GET redirige a la página principal
8
9  /login -> GET redirige a la página del login
10
11 /login/auth -> POST autenticación con los datos del formulario de login
12
13 /users -> GET página de ejemplo generada por express generator
14
15 # Prerequisitos
16
17 Nodejs
18
19 ## Generación del proyecto
20
21 Este proyecto ha sido generado usando el generador de express siguiendo los siguientes pasos
22
23 1 - Instalar la dependencia
24
25 dn: dc=iesalixar,dc=org
```

2. Implementa el método post necesario para comprobar que el usuario existe en el LDAP. Si no existe debe redirigir a la página del login de nuevo. Si existe mostrar el uid del usuario y el dn. Tal y como se muestra en la imagen pero para tu usuario y la ou=alumnos.

Primero creamos la conexión con el post donde mostrara el usuario tras loguearse y que autentique el usuario y contraseña introducido, para que si diese un error vuelva al login:

```

router.post('/auth', (req, res) => {
  var ldap = new LdapAuth({
    url: 'ldap://localhost:389',
    bindDN: 'cn=admin, dc=iesalixar, dc=org',
    bindCredentials: 'passiesalixar',
    searchBase: 'ou=alumnos, dc=iesalixar, dc=org',
    searchFilter: '(uid={{username}})',
    reconnect: true,
  });

  ldap.authenticate(req.body.username, req.body.password, function (err, user) {
    if (err) {
      res.render('Login', {
        title: 'Express'
      });
    } else {
      res.send(`Username: ${user.uid}, DN: ${user.dn}`);
    }
  });
});
});

```

Compruebo si la contraseña es correcta con el siguiente comando:

```

docker exec -it ldap ldapwhoami -vvv -H ldap://localhost -D
"uid=alum.mjaenr.2021,ou=alumnos,dc=iesalixar,dc=org" -x -W

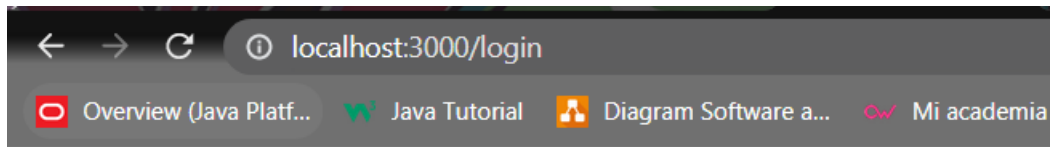
```

```

PS C:\Users\Manuel\Documents\2 DAW\Despliegue de Aplicaciones Web\UD5\T1-Configuración de Repositorio LDAP para Login con Express JS> docker exec -it ldap ldapwhoami -vvv
-H ldap://localhost -D "uid=alum.mjaenr.2021,ou=alumnos,dc=iesalixar,dc=org" -x -W
ldap_initialize( ldap://localhost:389/??base )
Enter LDAP Password:
dn:uid=alum.mjaenr.2021,ou=alumnos,dc=iesalixar,dc=org
Result: Success (0)

```

Resultado de la conexión Login:

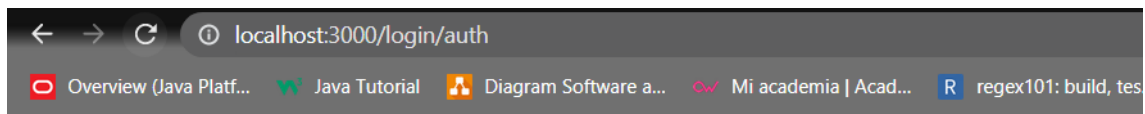


Express

Welcome to Express

Username:

Password:



Username: alum.mjaenr.2021,DN: uid=alum.mjaenr.2021,ou=alumnos,dc=iesalixar,dc=org