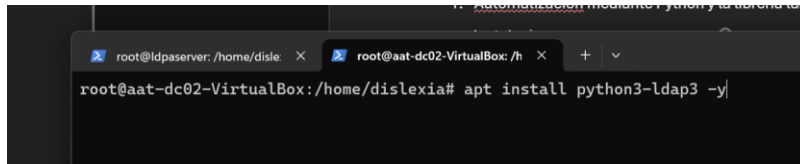


Alejandro Almagro Parte Opcional dentro de la practica ldap.

1. Automatizacion mediante Python y la librería ldap3:

Instalacion:

```
root@aat-dc02-VirtualBox:/home/dislexia# apt install python3-pip
```



```
root@aat-dc02-VirtualBox:/home/dislexia# apt install python3-ldap3 -y
```

Ahora crearemos nuestro script para el servidor en un nano create_users.py de la siguiente manera:

Conexion al servidor:

```
from ldap3 import Server, Connection, ALL, MODIFY_REPLACE
```

```
server = Server('ldap://dc02.aso.local:389', get_info=ALL)
```

```
conn = Connection(server, 'cn=admin,dc=aso,dc=local', '1234', auto_bind=True)
```

Lista de usuarios (ejemplos):

```
users = [
```

```
{'uid': 'user1', 'cn': 'User One', 'sn': 'One', 'mail': 'user1@example.com', 'password':  
'password1'},
```

```
{'uid': 'user2', 'cn': 'User Two', 'sn': 'Two', 'mail': 'user2@example.com', 'password':  
'password2'},
```

```
{'uid': 'user3', 'cn': 'User Three', 'sn': 'Three', 'mail': 'user3@example.com', 'password':  
'password3'},
```

```
]
```

Funcion para agregar usuarios:

```
def add_user(user):  
    dn = f"uid={user['uid']},ou=Users,dc=aso,dc=com"  
    attributes = {  
        'objectClass': ['inetOrgPerson'],  
        'uid': user['uid'], 'cn': user['cn'],  
        'sn': user['sn'], 'mail': user['mail'],  
        'userPassword': user['password']  
    }  
    conn.add(dn, attributes=attributes)  
    if conn.result['description'] == 'success':  
        print(f"Usuario {user['uid']} agregado.")  
    else:  
        print(f"Error al agregar {user['uid']}: {conn.result['description']}")
```

Crear los usuarios

```
for user in users:
```

```
    Add_user(user)
```

Cerramos la conexion:

```
Conn.unbind()
```

```
root@ldapserver: /home/disle x root@aat-dc02-VirtualBox: /h x + v
GNU nano 7.2 create_users.py *
from ldap3 import Server, Connection, ALL

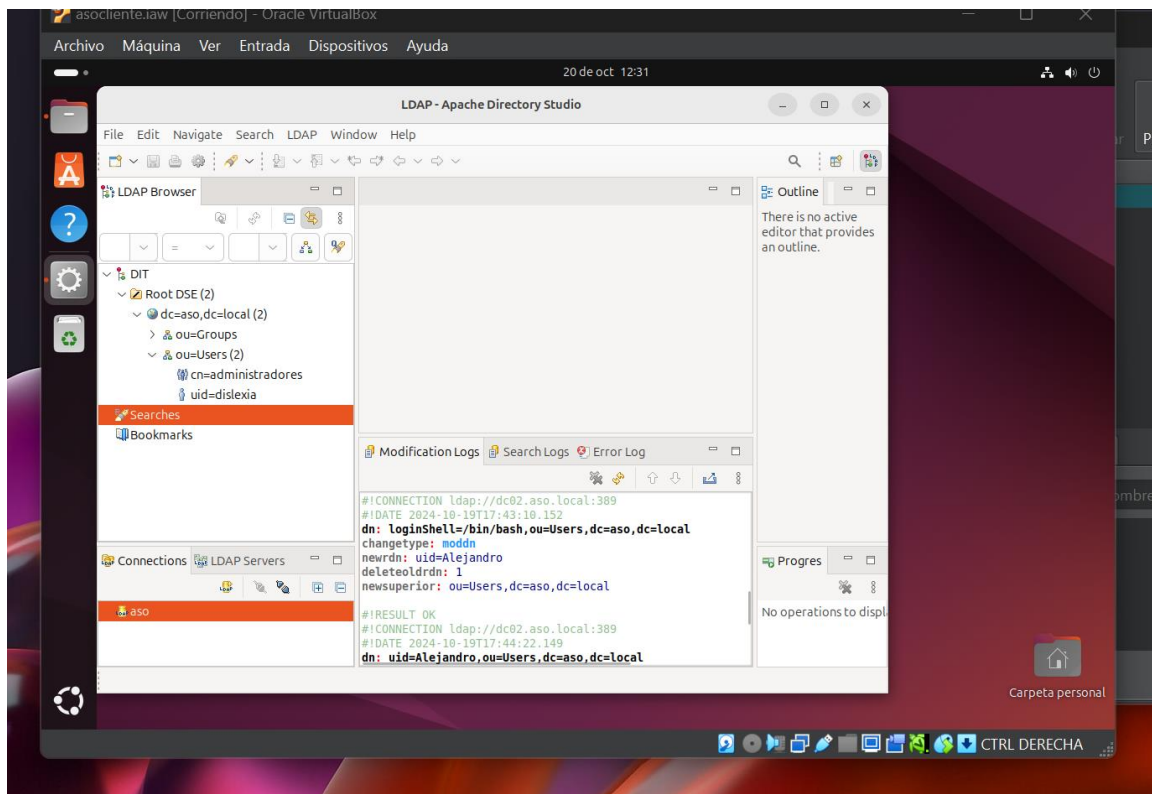
server = Server('ldap://dc02.aso.local:389', get_info=ALL)
conn = Connection(server, 'cn=admin,dc=aso,dc=local', '1234', auto_bind=True)

users = [
    {'uid': 'user1', 'cn': 'User One', 'sn': 'One', 'mail': 'user1@aso.local', 'password': 'password1'},
    {'uid': 'user2', 'cn': 'User Two', 'sn': 'Two', 'mail': 'user2@aso.local', 'password': 'password2'},
    {'uid': 'user3', 'cn': 'User Three', 'sn': 'Three', 'mail': 'user3@aso.local', 'password': 'password3'},
]

def add_user(user):
    dn = f"uid={user['uid']},ou=Users,dc=aso,dc=local"
    attributes = {
        'objectClass': ['inetOrgPerson'],
        'uid': user['uid'],
        'cn': user['cn'],
        'sn': user['sn'],
        'mail': user['mail'],
        'userPassword': user['password']
    }
    conn.add(dn, attributes=attributes)
    if conn.result['description'] == 'success':
        print(f"Usuario {user['uid']} agregado.")
    else:
```

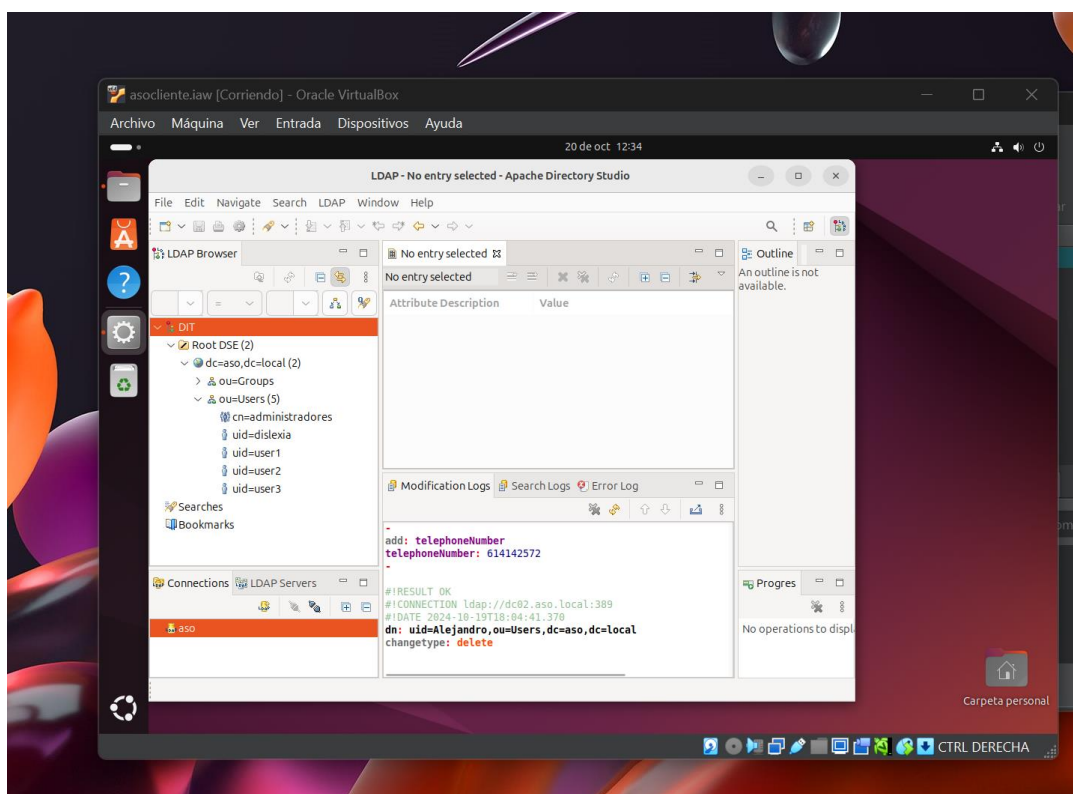
^G Ayuda ^O Guardar ^W Buscar ^K Cortar ^T Ejecutar ^C Ubicación M-U Deshacer M-A Poner marca
^X Salir ^R Leer fich. ^E Reemplazar ^U Pegar ^J Justificar ^_ Ir a línea M-E Rehacer M-G Copiar

Imagen de apache studio antes de ejecutar el script:



```
root@ldpserver: /home/disle: X root@aat-dc02-VirtualBox: /h X + v
root@aat-dc02-VirtualBox:/home/dislexia# python create_users.py
Usuario user1 agregado.
Usuario user2 agregado.
Usuario user3 agregado.
root@aat-dc02-VirtualBox:/home/dislexia# |
```

Cerramos conexion y volvemos a establecerla en apache studio:



Gestion de modificaciones con conn.modify:

Ejemplo para modificar el correo electronico masivos en usuarios:

Mediante otro script:

```
from ldap3 import Server, Connection, ALL, MODIFY_REPLACE
```

Igual que antes :

```
server = Server('ldap://dc02.aso.local:389', get_info=ALL)
```

```
conn = Connection(server, 'cn=admin,dc=aso,dc=local', '1234', auto_bind=True)
```

Usuarios seleccionados:

```
users = [  
    {'uid': 'user1', 'new_mail': 'alejandro3@aso.local'},  
    {'uid': 'user2', 'new_mail': 'naat2@aso.local'},  
    {'uid': 'user3', 'new_mail': 'manuel@aso.local'},  
]
```

Función de modificacion:

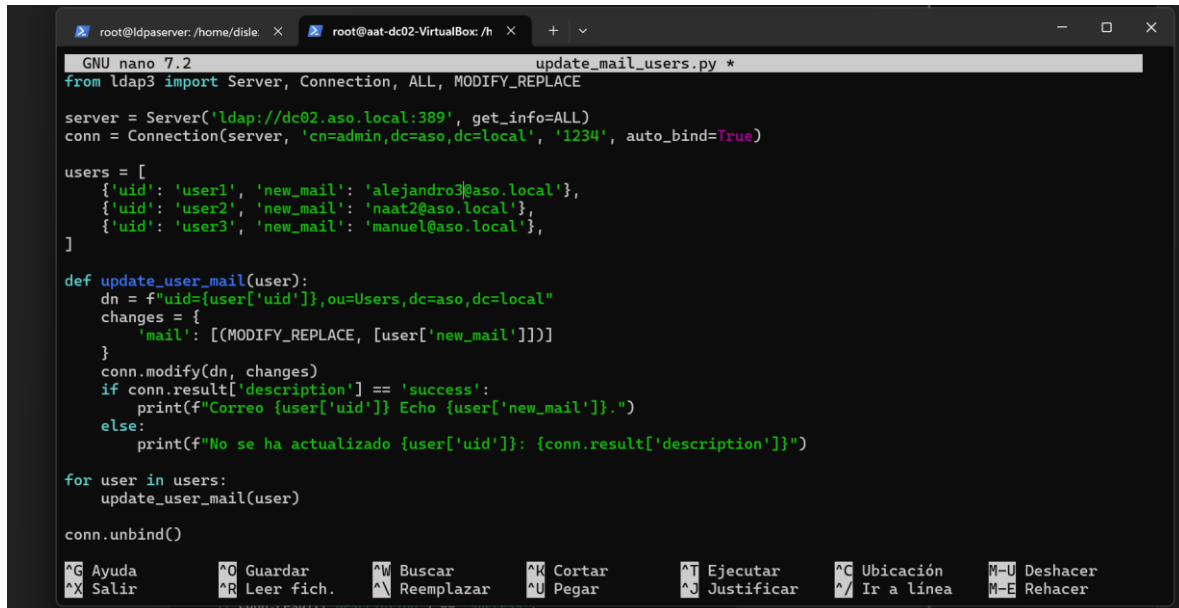
```
def update_user_mail(user):  
    dn = f"uid={user['uid']},ou=Users,dc=aso,dc=local"  
    changes = {  
        'mail': [(MODIFY_REPLACE, [user['new_mail']])]  
    }  
    conn.modify(dn, changes)  
    if conn.result['description'] == 'success':  
        print(f"Correo {user['uid']} Echo {user['new_mail']}.")  
    else:  
        print(f"No se ha actualizado {user['uid']}: {conn.result['description']}")
```

Modificación:

for user in users:

 update_user_mail(user)

conn.unbind()



```
GNU nano 7.2 update_mail_users.py *
from ldap3 import Server, Connection, ALL, MODIFY_REPLACE

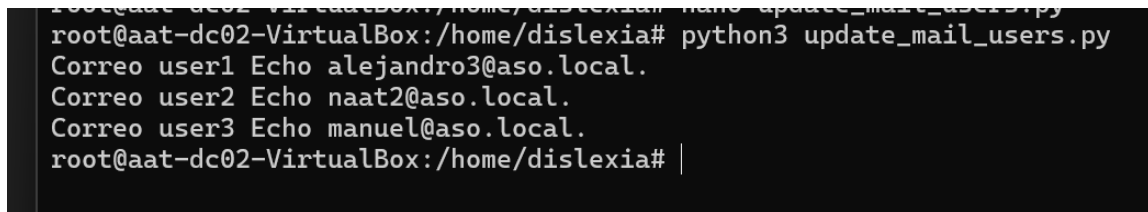
server = Server('ldap://dc02.aso.local:389', get_info=ALL)
conn = Connection(server, 'cn=admin,dc=aso,dc=local', '1234', auto_bind=True)

users = [
    {'uid': 'user1', 'new_mail': 'alejandro3@aso.local'},
    {'uid': 'user2', 'new_mail': 'naat2@aso.local'},
    {'uid': 'user3', 'new_mail': 'manuel@aso.local'},
]

def update_user_mail(user):
    dn = f"uid={user['uid']},ou=Users,dc=aso,dc=local"
    changes = {
        'mail': [(MODIFY_REPLACE, [user['new_mail']])]
    }
    conn.modify(dn, changes)
    if conn.result['description'] == 'success':
        print(f"Correo {user['uid']} Echo {user['new_mail']}")
    else:
        print(f"No se ha actualizado {user['uid']}: {conn.result['description']}")

for user in users:
    update_user_mail(user)

conn.unbind()
```



```
root@aat-dc02-VirtualBox:/home/dislexia# python3 update_mail_users.py
Correo user1 Echo alejandro3@aso.local.
Correo user2 Echo naat2@aso.local.
Correo user3 Echo manuel@aso.local.
root@aat-dc02-VirtualBox:/home/dislexia#
```

Refrescamos apache estudio:

u1	u1nre
mail	alejandro3@aso.local
uid	user1
userPassword	Plain text password

mail	naat2@aso.local
uid	user2
userPassword	Plain text password

u3	u3nre
mail	manuel@aso.local
uid	user3
userPassword	Plain text password

Eliminacion de usuarios creados:

```
root@ldpaserver: /home/disle: × root@aat-dc02-VirtualBox: /h × + v
GNU nano 7.2 delete_users.py *
from ldap3 import Server, Connection, ALL

server = Server('ldap://dc02.aso.local:389', get_info=ALL)
conn = Connection(server, 'cn=admin,dc=aso,dc=local', '1234', auto_bind=True)

users = ['user1', 'user2', 'user3']

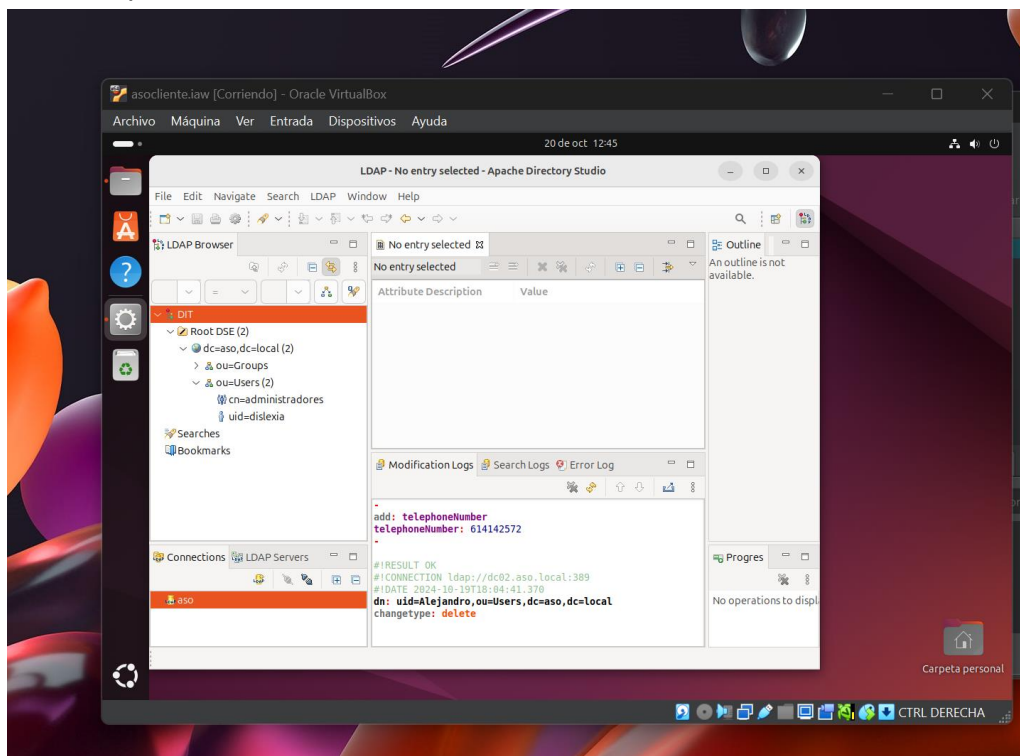
def delete_user(uid):
    dn = f"uid={uid},ou=Users,dc=aso,dc=local"
    conn.delete(dn)
    if conn.result['description'] == 'success':
        print(f"Usuario {uid} eliminado.")
    else:
        print(f"Error al eliminar {uid}: {conn.result['description']}")

for uid in users:
    delete_user(uid)

conn.unbind()
```

```
root@aat-dc02-VirtualBox:/home/dislexia# python3 delete_users.py
Usuario user1 eliminado.
Usuario user2 eliminado.
Usuario user3 eliminado.
root@aat-dc02-VirtualBox:/home/dislexia#
```

Vemos apache studio:



Mediante python podemos hacer un monton de cosas como scripts para estas tareas tediosas aunque tambien podriamos utilizarlo para tareas mas complicadas basandonos en automatizacion por ejemplo:

```
GNU nano 7.2                                     basededatos.py *
from ldap3 import Server, Connection, ALL, MODIFY_REPLACE

def connect_to_db():
    db = mysql.connector.connect(
        host="localhost",
        user="aat_user",
        password="aat_password",
        database="aat"
    )
    return db

def connect_to_ldap():
    server = Server('ldap://dc02.aso.local:389', get_info=ALL)
    conn = Connection(server, 'cn=admin,dc=aso,dc=local', '1234', auto_bind=True)
    return conn

def sync_users():
    db = connect_to_db()
    cursor = db.cursor(dictionary=True)

    cursor.execute("SELECT uid, cn, sn, mail FROM users")
    db_users = cursor.fetchall()

    conn = connect_to_ldap()

    for user in db_users:
```

```
GNU nano 7.2                                     basededatos.py *
    for user in db_users:
        dn = f"uid={user['uid']},ou=Users,dc=aso,dc=local"

        conn.search(dn, '(objectClass=*)')
        if len(conn.entries) == 0:
            attributes = {
                'objectClass': ['inetOrgPerson'],
                'uid': user['uid'],
                'cn': user['cn'],
                'sn': user['sn'],
                'mail': user['mail'],
                'userPassword': 'default_password'
            }
            conn.add(dn, attributes=attributes)
            print(f"Usuario {user['uid']} creado en LDAP.")
        else:
            changes = {
                'cn': [(MODIFY_REPLACE, [user['cn']])],
                'sn': [(MODIFY_REPLACE, [user['sn']])],
                'mail': [(MODIFY_REPLACE, [user['mail']])]
            }
            conn.modify(dn, changes)
            print(f"Usuario {user['uid']} actualizado en LDAP.")

    conn.unbind()
    cursor.close()
```

Es decir podemos hacer muchas tareas automatizadas gracias a python y si le sumamos el uso de IA o machine learning a nuestro dia a dia podriamos gestionar si sabemos lo que hacemos de manera muy simple y eficaz todo el bosque de nuestra empresa o poryecto personal.

