

## Documentacion carga de datos

Para cargar la base de datos se siguieron los siguientes pasos:

1. Primero se decidió el volumen de datos que se tiene por cada tabla. Se eligieron que las tablas con mayor porcentaje de registros fueran Usuarios, Reservas, Productos, Servicio\_producto, Reservas\_Servicios, Caract\_Habitaciones. Asimismo, los de menor volumen fueron los de hoteles, Servicios, CaracHabitaciones, Habitaciones y plans\_de\_consumo. Sumando todo, nos da un resultado de 637,500 datos para ser cargados a la base de datos.

	Datos
CaracHabitaciones	10,000
Caract_Habitacion	100,000
Habitaciones	2,000
Hoteles	500
Planes_de_consumo	20,000
Productos	100,000
Reservas	100,000
Reservas_Servicios	100,000
Servicios	5,000
Servicios_Productos	100,000
Usuarios	100,000
	637,500

2. Se creo un archivo en python y se proparo el ambiente para empezar a trabajar con la librería Random y faker.

```
import random
from faker import Faker

# Crear un objeto Faker
fake = Faker()
```

3. Se creo la función generate\_insert\_statement(): la cual crea sentencias sql y las agrega en un archivo sql. Las funciones con única primary key, tienen un contador que les permite evitar ser repetidas durante la ejecución.

```
# Generar datos para las tablas
def generate_insert_statements():
    i=0

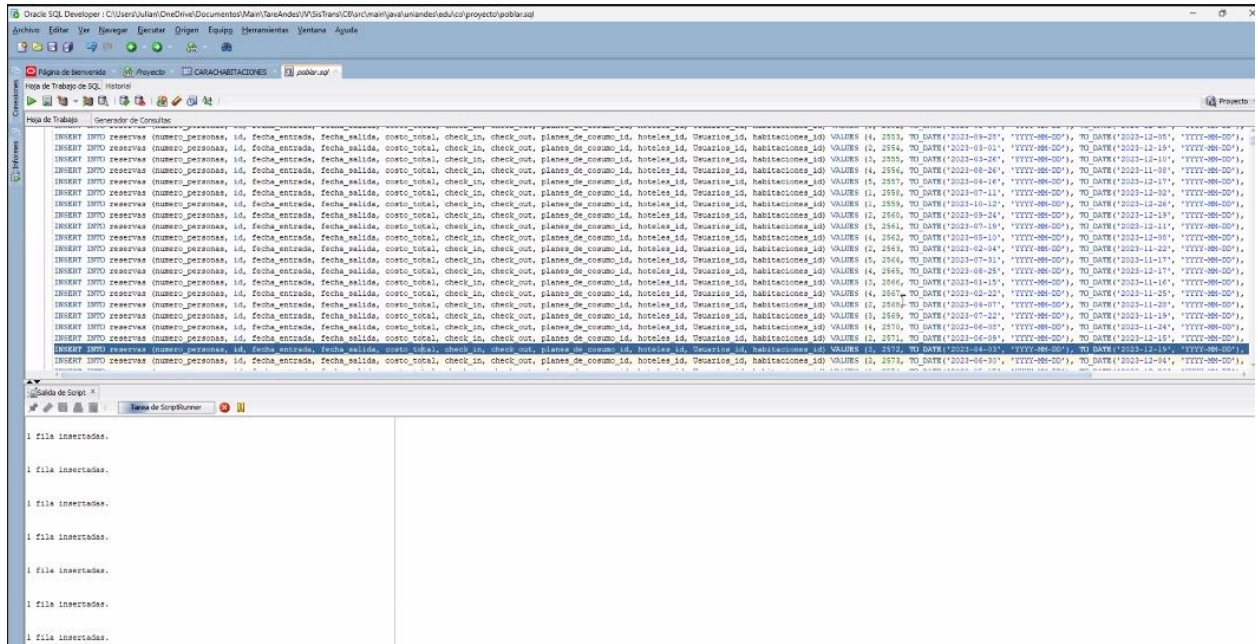
    for _ in range(500):
        # Generar sentencia SQL para insertar datos en la tabla 'hoteles'
        hoteles_insert = f"INSERT INTO hoteles (nombre, tipo, estrellas, id) VALUES ('{fake.company()}', '{fake.word()}', {random.randint(1, 5)}
        print(hoteles_insert)
        i+=1

    i=0
    for _ in range(100000):
        # Generar sentencia SQL para insertar datos en la tabla 'usuarios'
        usuarios_insert = f"INSERT INTO usuarios (id, login, contrasena, nombre, no_documento, tipo_usuario, hoteles_id) VALUES ({i}, '{fake.us"
        print(usuarios_insert)
        i+=1
```

```
i=0
for _ in range(100000):
    # Generar sentencia SQL para insertar datos en la tabla 'servicios_productos'
    servicios_productos_insert = f"INSERT INTO servicios_productos (servicios_id, productos_id) VALUES ({random.randint(1, 5000)}, {random.
    print(servicios_productos_insert)

pass
```

4. Ingresar el archivo SQL en sqldeveloper. Se tuvo que ingresar todos los datos de cada tabla por separado. Esto nos ayudo a que no tuviéramos errores con sqldeveloper evitando cancelaciones de la ejecución inesperados.



5. Finalizar el proceso y verificar que todos los registros estén cargados.