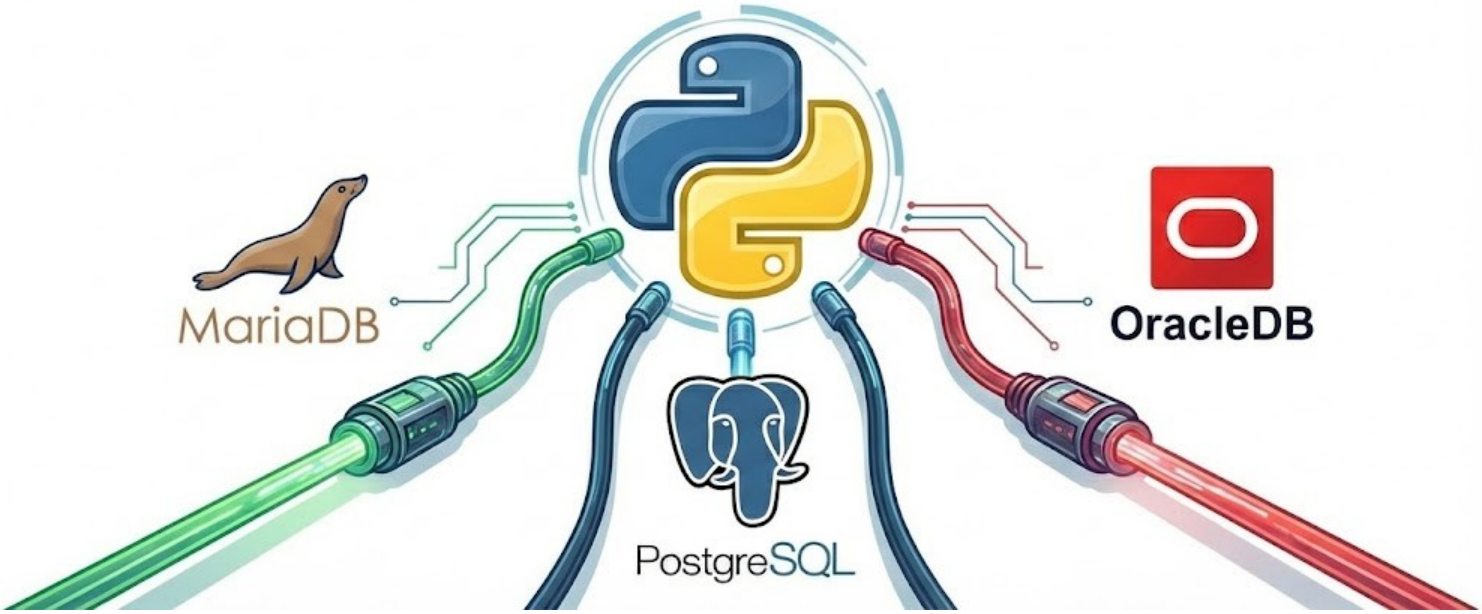
CONEXIÓN PYTHON

MariaDB | PostgreSQL | OracleDB



Gestión de bases de datos | I.E.S Gonzalo Nazreno

Concepción Guisado Jurado

1º A.S.I.R (Administración de sistemas informáticos en red)

2025-2026

Jesús Figueroa Roldán

Índice

Dependencias necesarias

Para que este programa en Python nos funcione correctamete primeramente tendremos que instalar unas dependencias necesarias.

```
jesusfigueroa ASIR …/SGBD-Python-Jesús-Figueroa-Roldán sudo apt install python3-mysqldb python3-psycopg2
 python3-dev default-libmysqlclient-dev build-essential libpq-dev build-essential pkg-config
[sudo] contraseña para jesusfigueroa:
python3-mysqldb ya está en su versión más reciente (1.4.6-2+b5).
python3-psycopg2 ya está en su versión más reciente (2.9.10-1+b1).
python3-dev ya está en su versión más reciente (3.13.5-1).
default-libmysqlclient-dev ya está en su versión más reciente (1.1.1).
build-essential ya está en su versión más reciente (12.12).
libpq-dev ya está en su versión más reciente (17.7-0+deb13u1).
pkg-config ya está en su versión más reciente (1.8.1-4).
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 0

jesusfigueroa ASIR …/SGBD-Python-Jesús-Figueroa-Roldán
```

Una vez que tengamos las dependencias necesarias instaladas vamos a crearnos un entorno virtual en python para trabajar desde ahí y tenerlo aislado del sistema. También esto es bueno para que no haya fallos entre las versiones de los repositorios y demás. Una vez que nos hagamos el entorno virtual lo vamos a activar y voy a instalar el conector de MariaDB y PostgreSQL aunque realmente con las dependencias que se han instalado antes con los repositorios se podría hacer pero por comodidad en mi caso lo voy a instalar todo en un entorno virtual.

```
jesusfigueroa ASIR …/python python3 -m venv proyecto

jesusfigueroa ASIR …/python source proyecto/bin/activate

jesusfigueroa ASIR …/python "proyecto" pip install mysqlclient psycopg2
Collecting mysqlclient
  Using cached mysqlclient-2.2.7-cp313-cp313-linux_x86_64.whl
Collecting psycopg2
  Using cached psycopg2-2.9.11-cp313-cp313-linux_x86_64.whl
Installing collected packages: psycopg2, mysqlclient
Successfully installed mysqlclient-2.2.7 psycopg2-2.9.11

jesusfigueroa ASIR …/python "proyecto"
```

Conector Python - MariaDB

Para que este programa en python nos funcione correctamente con MariaDB tendremos que tener instaladas las dependencias que he mencionado anteriormente, además de la base de datos creada con sus respectivas tablas. Una vez que tenemos la base de datos desplegada nos vamos a centrar en la función de conexión. Tenemos que asignar a una variable el modulo connect de MySQLdb y a esta función del módulo MySQLdb le vamos a pasar los parámetros de conexión, el host en mi caso es mi máquina pero si fuese un servidor remoto pues la dirección IP de ese servidor, el usuario de la base de datos con su respectiva contraseña y la base de datos, si todo esto lo tenemos bien cuando ejecutemos el programa nos vamos a poder conectar a MariaDB.

```python
def conectar_mysql():
    try:
        db = MySQLdb.connect(host="localhost", user="hotel", passwd="passwd", db="gestionhotel")
        print(">> Conectado a MySQL/MariaDB correctamente.")
        return db
    except MySQLdb.Error as error:
        print("Error al conectar a MySQL:", error)
```

Conector Python - PostgreSQL

Para que este programa en python nos funcione correctamente con el sistema gestor de base de datos PostgreSQL tendremos que tener las dependencias que he mencionado anteriormente, además de la base de datos creada con sus respectivas tablas. Para este sistema gestor de base de datos es similar a MariaDB. Para la conectivida vamos a usar el módulo psycopg2 y el método connect al cuál le vamos a pasar los parámetros de conexión, host en mi caso mi máquina pero si fuese un servidor la dirección IP, la base de datos, el usuario y la contraseña de dicho usuario.

Si la base de datos la hemos desplegado bien y hemos asignado bien los parámetros a la función de conexión nos podremos conectar a la base de datos.

```python
def conectar_postgresql():
    try:
        db = psycopg2.connect(host="localhost", database="gestionhotel", user="hotel", password="password")
        print(">> Conectado a PostgreSQL correctamente.")
        return db
    except psycopg2.Error as error:
        print("Error al conectar a PostgreSQL:", error)
```

Conector Python - Oracle

Una vez que tengamos las dependencias necesarias instaladas y el entorno virtual creado vamos a instalar el conector de Oracle. He probado varios conectores de Oracle pero el único que me ha funcionado correctamente es este que lo he conseguido de la documentación oficial.

IMPORTANTE: la ruta en la que creamos el entorno virtual no puede tener espacios ya que si tiene espacios nos va a dar error y no vamos a poder activarlo.

```
jesusfigueroa ASIR ~/Descargas/conector "proyecto" python3 -m pip install oracledb --upgrade
Collecting oracledb
  Downloading oracledb-3.4.2-cp313-cp313-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl.metadata (7.7 kB)
Collecting cryptography≥3.2.1 (from oracledb)
  Downloading cryptography-46.0.4-cp311-abi3-manylinux_2_34_x86_64.whl.metadata (5.7 kB)
Collecting typing_extensions≥4.14.0 (from oracledb)
  Using cached typing_extensions-4.15.0-py3-none-any.whl.metadata (3.3 kB)
Collecting cffi≥2.0.0 (from cryptography≥3.2.1→oracledb)
  Using cached cffi-2.0.0-cp313-cp313-manylinux2014_x86_64.manylinux_2_17_x86_64.whl.metadata (2.6 kB)
Collecting pycparser (from cffi≥2.0.0→cryptography≥3.2.1→oracledb)
  Downloading pycparser-3.0-py3-none-any.whl.metadata (8.2 kB)
Downloading oracledb-3.4.2-cp313-cp313-manylinux2014_x86_64.manylinux_2_17_x86_64.manylinux_2_28_x86_64.whl (2.4 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.4/2.4 MB 44.8 MB/s eta 0:00:00
Downloading cryptography-46.0.4-cp311-abi3-manylinux_2_34_x86_64.whl (4.5 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.5/4.5 MB 63.9 MB/s eta 0:00:00
Using cached cffi-2.0.0-cp313-cp313-manylinux2014_x86_64.manylinux_2_17_x86_64.whl (219 kB)
Using cached typing_extensions-4.15.0-py3-none-any.whl (44 kB)
Downloading pycparser-3.0-py3-none-any.whl (48 kB)
Installing collected packages: typing_extensions, pycparser, cffi, cryptography, oracledb
Successfully installed cffi-2.0.0 cryptography-46.0.4 oracledb-3.4.2 pycparser-3.0 typing_extensions-4.15.0

jesusfigueroa ASIR ~/Descargas/conector "proyecto"
```

Una vez que tenemos las dependencias necesarias y el entorno virtual creado, vamos a irnos a la máquina que tenemos instalado Oracle y vamos a poner el siguiente comando para ver si está en escucha y si lo está vamos a ver el puerto que está usando ya que esto va a ser importante para poner en el fichero de funciones para la conexión. Normalmente Oracle usa el puerto 1521 pero para asegurarnos ponemos este comando y lo miramos manualmente.

```
usuario@debian:~$ lsnrctl status

LSNRCTL for Linux: Version 21.0.0.0.0 - Production on 31-ENE-2026 12:13:38

Copyright (c) 1991, 2021, Oracle.  All rights reserved.

Conectándose a (ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521))
ESTADO del LISTENER
------------------------
Alias                     LISTENER
Versión                   TNSLSNR for Linux: Version 21.0.0.0.0 - Production
Fecha de Inicio       31-ENE-2026 11:21:01
Tiempo Actividad   0 días 0 hr. 52 min. 37 seg.
Nivel de Rastreo        off
Seguridad               ON: Local OS Authentication
SNMP                      OFF                                                  Servicio por Defecto          XE
Parámetros del Listener   /opt/oracle/homes/OraDBHome21cXE/network/admin/listener.ora
Log del Listener          /opt/oracle/diag/tnslsnr/debian/listener/alert/log.xml
Recibiendo Resumen de Puntos Finales...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=debian)(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST=127.0.0.1)(PORT=5500))(Security=(my_wallet_directory=/opt/oracle/homes/OraDBHome21cXE/admin/XE/xdb_wall
et))(Presentation=HTTP)(Session=RAW))
Resumen de Servicios...
El servicio "47311402cc3e129ee0630101007f72f4" tiene 1 instancia(s).
  La instancia "XE", con estado READY, tiene 1 manejador(es) para este servicio...
El servicio "XE" tiene 1 instancia(s).
  La instancia "XE", con estado READY, tiene 1 manejador(es) para este servicio...
El servicio "XEXDB" tiene 1 instancia(s).
  La instancia "XE", con estado READY, tiene 1 manejador(es) para este servicio...
El servicio "xepdb1" tiene 1 instancia(s).
  La instancia "XE", con estado READY, tiene 1 manejador(es) para este servicio...
El comando ha terminado correctamente
usuario@debian:~$
```

IMPORTANTE: una vez que hemos instalado las dependencias en el entorno virtual para que el editor de código te interpreta la librería hay que reiniciarlo normalmente. Dependiendo del editor de código que uses en mi caso he usado Thony y Visual Studio Code y en ambos tuve que reiniciarlo.

Aquí vemos la parte importante de la conexión con Oracle, en la variable dsn_str (Data source name) que es como la he sacado de la documentación oficial, tenemos que indicarle la dirección IP, el puerto y la versión de Oracle que está usando la máquina que lo tiene instalado. Si Oracle lo tienes en tu máquina física puedes poner localhost pero en mi caso lo tengo en una máquina virtual en Virt-Maganer por lo tanto indicamos la dirección IP y el puerto seguido de la versión. Una vez que tenemos eso pasamos a el usuario, que es el que tengamos creado con su contraseña y para la variable dsn (Data source name) indicamos la variable de arriba que la hemos definido previamente con los valores de conexión.

IMPORTANTE: Una vez que ejecutamos el programa para hacer operaciones debemos estar dentro del entorno virtual.

```python
def conectar_oracle():
    try:
        dsn_str = "192.168.122.167:1521/xe"

        db = oracledb.connect(
            user="hotel",
            password="passwd",
            dsn=dsn_str
        )
        print(">> Conectado a Oracle correctamente.")
```

Si todo esto funciona correctamente cuando elegimos la opción de Oracle ya estaremos conectado correctamente y podremos ejecutar las operaciones.

```
jesusfigueroa ASIR ~/Proyecto-SGBD/Python main ◔ "proyecto" python3 main.py

============================================================
MENÚ DE CONEXIÓN A SGBD
============================================================
    1. MySQL / MariaDB
    2. PostgreSQL
    3. Oracle
    4. Salir

Introduce una opción: 3
>> Conectado a Oracle correctamente.
```

Funcionamiento del programa

MariaDB

Opción 1

Con esta opción vamos a listar los tipos de habitación con una consulta sencilla (función de agregación).

```
Introduce una opción: 1
============================================================
RESUMEN DE HABITACIONES POR TIPO
============================================================
Tipo: Doble | Cantidad: 17
Tipo: Individual | Cantidad: 16
Tipo: Suite | Cantidad: 17
```

Opción 2

Esta opción del menú podemos filtrar por precio de habitaciones entre un precio mínimo y un máximo.

```
Introduce una opción: 2
Precio mínimo: 5
Precio máximo: 70
============================================================
HABITACIONES DISPONIBLES
============================================================
Habitación: 101 | Tipo: Individual | Precio: 66.00
Habitación: 104 | Tipo: Individual | Precio: 66.00
Habitación: 201 | Tipo: Individual | Precio: 55.00
Habitación: 204 | Tipo: Individual | Precio: 55.00
Habitación: 301 | Tipo: Individual | Precio: 60.00
Habitación: 304 | Tipo: Individual | Precio: 60.00
Habitación: 401 | Tipo: Individual | Precio: 65.00
Habitación: 404 | Tipo: Individual | Precio: 65.00
Habitación: 501 | Tipo: Individual | Precio: 70.00
Habitación: 504 | Tipo: Individual | Precio: 70.00
```

Opción 3

Con está opción vamos a buscar las reservas de un cliente por DNI.

```
Introduce una opción: 3
Introduce el DNI del cliente: 86878889Q
============================================================
RESERVAS DEL CLIENTE
============================================================
Nombre: Sonia Herrera | Fecha entrada: 2024-04-10 | Habitación: 705
```

Opción 4

Esta opción nos va a permitir dar de alta a clientes en nuestra base de datos, nos pedirá los campos de la tabla clientes y lo iremos rellenando.

```
Introduce una opción: 4
============================================================
REGISTRO DE NUEVO CLIENTE
============================================================
ID Cliente: 59
Nombre: Jesús Figueroa Roldán
Email: jesusfigueroa@iesgn
Teléfono: 690768192
DNI: 14592804P
>> Cliente registrado correctamente.
```

Una vez que lo añadimos si nos vamos al sistema gestor de base de datos que estabamos usando en el programa de python podemos observar que se ha añadido correctamente.

```
| 47 | Alicia Campos          | alicia.campos@mail.com    | 611111222 | 10111213W | 2024-02-16 |
| 48 | Antonio Moreno         | antonio.moreno@mail.com   | 611333444 | 14151617X | 2024-02-17 |
| 49 | Cristina Navarro       | cristina.navarro@mail.com | 611555666 | 18192022Y | 2024-02-18 |
| 50 | David Herrera          | david.herrera@mail.com    | 611777888 | 22232426Z | 2024-02-19 |
| 59 | Jesús Figueroa Roldán  | jesusfigueroa@iesgn       | 690768192 | 14592804P | 2026-02-05 |
```

Para la opción 5 vamos a borrar por id de reserva, antes de borrar nada vamos a mirar los datos de la tabla de nuestro SGBD correspondiente.

```
MariaDB [gestionhotel]> select * from servicios_reserva;
+-------------+------------+----------+--------+----------------+
| id_servicio | id_reserva | servicio | precio | fecha_servicio |
+-------------+------------+----------+--------+----------------+
|           3 |          3 | Cena     |  15.00 | 2024-03-04     |
|           4 |          4 | Spa      |  30.00 | 2024-03-05     |
|           5 |          5 | Desayuno |  10.00 | 2024-03-06     |
|           6 |          6 | Cena     |  15.00 | 2024-03-07     |
|           7 |          7 | Spa      |  30.00 | 2024-03-08     |
```

Opción 5

Nos vamos al programa en python y borramos por ejemplo el id de reserva 2 y ya estaría borrada de nuestra tabla del SGBD correspondiente.

```
Introduce una opción: 5
============================================================
ELIMINAR SERVICIOS DE RESERVA
============================================================
Introduce ID de reserva para borrar sus servicios: 3
>> Servicios eliminados. Filas afectadas: 1
```

Después de haberlo borrado si nos vamos a MariaDB manualmente y hacemos un select vemos que se ha borrado correctamente.

```
MariaDB [gestionhotel]> select * from servicios_reserva where id_reserva = 3;
Empty set (0,000 sec)

MariaDB [gestionhotel]>
```

Para la opción 6 del menú vamos a aumentar el precio de una noche en tanto por ciento para ello antes de actualizar nada vamos a mirar que precio tienen.

```
MariaDB [gestionhotel]> select * from habitaciones where piso = 3;
+---------------+--------+------------+------+-----------+--------------+------------+
| id_habitacion | numero | tipo       | piso | capacidad | precio_noche | disponible |
+---------------+--------+------------+------+-----------+--------------+------------+
|            13 |    301 | Individual |    3 |         1 |        60.00 | S          |
|            14 |    302 | Doble      |    3 |         2 |        80.00 | S          |
|            15 |    303 | Suite      |    3 |         3 |       140.00 | S          |
|            16 |    304 | Individual |    3 |         1 |        60.00 | S          |
|            17 |    305 | Doble      |    3 |         2 |        80.00 | S          |
|            18 |    306 | Suite      |    3 |         3 |       140.00 | S          |
+---------------+--------+------------+------+-----------+--------------+------------+
6 rows in set (0,000 sec)

MariaDB [gestionhotel]>
```

Opción 6

Nos vamos a nuestro programa en python y vamos a aumentar un 15% por ejemplo, nos dice que se han actualizado 6 filas.

```
Introduce una opción: 6
============================================================
ACTUALIZAR PRECIO POR PISO
============================================================
Número de piso a actualizar: 3
Porcentaje de incremento (ej: 10): 15
>> Precios actualizados. Filas actualizadas: 6
```

Volvemos a MariaDB y como vemos se ha actualizado correctamente.

```
MariaDB [gestionhotel]> select * from habitaciones where piso = 3;
+---------------+--------+------------+------+-----------+--------------+------------+
| id_habitacion | numero | tipo       | piso | capacidad | precio_noche | disponible |
+---------------+--------+------------+------+-----------+--------------+------------+
|            13 |    301 | Individual |    3 |         1 |        69.00 | S          |
|            14 |    302 | Doble      |    3 |         2 |        92.00 | S          |
|            15 |    303 | Suite      |    3 |         3 |       161.00 | S          |
|            16 |    304 | Individual |    3 |         1 |        69.00 | S          |
|            17 |    305 | Doble      |    3 |         2 |        92.00 | S          |
|            18 |    306 | Suite      |    3 |         3 |       161.00 | S          |
+---------------+--------+------------+------+-----------+--------------+------------+
6 rows in set (0,000 sec)

MariaDB [gestionhotel]>
```

Opción 7

Para esta opción vamos a poder listar los clientes con gasto superior a la media de los clientes.

```
Introduce una opción: 7
============================================================
CLIENTES CON GASTO EN SERVICIOS SUPERIOR A LA MEDIA
============================================================
Cliente: Lucía Herrera | Total gastado: 30.00
Cliente: Cristina Navarro | Total gastado: 30.00
Cliente: Antonio Molina | Total gastado: 30.00
Cliente: Raúl Pérez | Total gastado: 30.00
Cliente: Clara Vega | Total gastado: 30.00
Cliente: Carla Méndez | Total gastado: 30.00
Cliente: Raúl Castro | Total gastado: 30.00
Cliente: Rubén Molina | Total gastado: 30.00
Cliente: Sofía Torres | Total gastado: 30.00
Cliente: Daniela Ramírez | Total gastado: 30.00
Cliente: Luis Gómez | Total gastado: 30.00
Cliente: Paula Moreno | Total gastado: 30.00
Cliente: Óscar Navarro | Total gastado: 30.00
Cliente: Raquel Fuentes | Total gastado: 30.00
Cliente: Verónica Campos | Total gastado: 30.00
Cliente: Jorge Castillo | Total gastado: 30.00
```

Opción 8

Esta última opción muestra por cada cliente el informe completo por reserva, como vemos en la captura nos muestra el cliente, la habitación que ha reservado, el precio por noche de esa habitación, el número total de noches, el dinero total gastado en servicios (desayuno, parking …) y el total de reserva que es el total del número de noches y los servicios adicionales. Como he comentado muestra de todos los clientes pero solo he puesto una como muestra.

```
Introduce una opción: 8
============================================================
INFORME COMPLETO DE INGRESOS POR RESERVA
============================================================
Cliente: Cristina Navarro
Habitación: 901
Precio por noche: 200.00
Noches: 4
Total servicios: 30.00
Total reserva: 830.00
------------------------------------------------------------
```

Opción 9

Con esta última opción lo que hacemos es ir hacia atrás para elegir el sistema gestor de base de datos con el que queremos operar.

```
============================================================
OPERACIONES DML (mysql)
============================================================
    1. Listar tipos de habitaciones
    2. Buscar habitaciones por precio
    3. Ver reservas de un cliente
    4. Registrar nuevo cliente
    5. Eliminar servicios de reserva
    6. Subir precios por piso
    7. Clientes con gasto superior a la media
    8. Informe completo de ingresos por reserva
    9. Volver al menú de SGBD

Introduce una opción: 9
Volviendo al menú de selección de SGBD...
>> Base de datos desconectada.
```

PostgreSQL

Priramente tendremos que montar nuestra base de datos con el esquema correctamente y si todo funciona bien estaremos conectados. Las opciones son las mismas pero adaptadas a PostgreSQL.

```
jesusfigueroa ASIR ~/Proyecto-SGBD/Python main "proyecto" python3 main.py

============================================================
MENÚ DE CONEXIÓN A SGBD
============================================================
    1. MySQL / MariaDB
    2. PostgreSQL
    3. Oracle
    4. Salir

Introduce una opción: 2
>> Conectado a PostgreSQL correctamente.
```

Opción 1

Es lo mismo que en MariaDB lista por tipo de habitaciones y nos lo muestra ordenado ascendentemente..

```
Introduce una opción: 1
============================================================
RESUMEN DE HABITACIONES POR TIPO
============================================================
Tipo: Individual | Cantidad: 16
Tipo: Suite | Cantidad: 17
Tipo: Doble | Cantidad: 17
```

Opción 2

Buscamos pisos en un rango entre precio mínimo y precio máximo. Si no hay pisos nos muestra un mensaje que no se han encontrado registros.

```
Introduce una opción: 2
Precio mínimo: 10
Precio máximo: 50
============================================================
HABITACIONES DISPONIBLES
============================================================
No se encontraron registros.
```

Opción 3

Buscamos las reservas de un cliente por DNI y muestra el número de habitación con la fecha de entrada y el nombre del cliente.

```
Introduce una opción: 3
Introduce el DNI del cliente: 62636465K
============================================================
RESERVAS DEL CLIENTE
============================================================
Nombre: Iván Ruiz | Fecha entrada: 2024-04-04 | Habitación: 605
```

Opción 4

Podemos dar de alta a clientes con todos los campos de la tabla clientes.

```
Introduce una opción: 4
============================================================
REGISTRO DE NUEVO CLIENTE
============================================================
ID Cliente: 60
Nombre: Jesús Figueroa Roldán
Email: jesusfigueroaroldan@iesgm
Teléfono: 680821972
DNI: 48840401A
>> Cliente registrado correctamente.
```

Si nos vamos a la tabla clientes vemos que se ha añadido el cliente correctamente.

```
         49 | Cristina Navarro      | cristina.navarro@mail.com | 611555666 | 18192622Y | 2024-02-18
         50 | David Herrera         | david.herrera@mail.com    | 611777888 | 22232426Z | 2024-02-19
         57 | Jesús Figueroa        | jesusfigueroa@iesgn       | 671915815 | 22796049P | 2026-02-03
         60 | Jesús Figueroa Roldán | jesusfigueroaroldan@iesgm | 680821972 | 48840401A | 2026-02-05
(52 filas)

(END)
```

Para la opción 5 vamos a borrar servicios de reserva por id de reserva.

```
gestionhotel⇒ select * from servicios_reserva where id_reserva = 2;
 id_servicio | id_reserva | servicio | precio | fecha_servicio
-------------+------------+----------+--------+----------------
           2 |          2 | Desayuno |  10.00 | 2024-03-03
(1 fila)

gestionhotel⇒
```

Indicamos la opción 5 en el menú e indicamos el ID de reserva en este caso 2 por ejemplo y nos dice 1 filas afectadas.

```
Introduce una opción: 5
============================================================
ELIMINAR SERVICIOS DE RESERVA
============================================================
Introduce ID de reserva para borrar sus servicios: 2
>> Servicios eliminados. Filas afectadas: 1
```

Si nos vamos a nuestra base de datos vemos que nos lo ha borrado correctamente.

```
gestionhotel⇒ select * from servicios_reserva where id_reserva = 2;
 id_servicio | id_reserva | servicio | precio | fecha_servicio
-------------+------------+----------+--------+----------------
(0 filas)

gestionhotel⇒
```

Para la opción 6 vamos a actualizar el precio en porcentaje para ello vamos a ver como esta el piso 4 antes de ejecutar esta opción.

```
gestionhotel⇒ select * from habitaciones where piso = 4;
 id_habitacion | numero |    tipo    | piso | capacidad | precio_noche | disponible
---------------+--------+------------+------+-----------+--------------+------------
            19 |    401 | Individual |    4 |         1 |        65.00 | S
            20 |    402 | Doble      |    4 |         2 |        85.00 | S
            21 |    403 | Suite      |    4 |         3 |       150.00 | S
            22 |    404 | Individual |    4 |         1 |        65.00 | S
            23 |    405 | Doble      |    4 |         2 |        85.00 | S
            24 |    406 | Suite      |    4 |         3 |       150.00 | S
(6 filas)
```

Ponemos un 15% por ejemplo de incremento y nos lo ha actualizado correctamente.

```
Introduce una opción: 6
============================================================
ACTUALIZAR PRECIO POR PISO
============================================================
Número de piso a actualizar: 4
Porcentaje de incremento (ej: 10): 15
>> Precios actualizados. Filas actualizadas: 6
```

Si nos vamos a la base de datos vemos que nos lo ha subido en un 15% el piso 4.

```
gestionhotel⇒ select * from habitaciones where piso = 4;
 id_habitacion | numero |    tipo    | piso | capacidad | precio_noche | disponible
---------------+--------+------------+------+-----------+--------------+------------
            19 |    401 | Individual |    4 |         1 |        74.75 | S
            20 |    402 | Doble      |    4 |         2 |        97.75 | S
            21 |    403 | Suite      |    4 |         3 |       172.50 | S
            22 |    404 | Individual |    4 |         1 |        74.75 | S
            23 |    405 | Doble      |    4 |         2 |        97.75 | S
            24 |    406 | Suite      |    4 |         3 |       172.50 | S
(6 filas)

gestionhotel⇒
```

Opción 7

Para la opción 7 vamos a listar clientes con gastos en servicios superior a la media de los clientes.

```
Introduce una opción: 7
============================================================
CLIENTES CON GASTO EN SERVICIOS SUPERIOR A LA MEDIA
============================================================
Cliente: Paula Moreno | Total gastado: 30.00
Cliente: Antonio Molina | Total gastado: 30.00
Cliente: Luis Gómez | Total gastado: 30.00
Cliente: Carla Méndez | Total gastado: 30.00
Cliente: Clara Vega | Total gastado: 30.00
Cliente: Raúl Castro | Total gastado: 30.00
Cliente: Rubén Molina | Total gastado: 30.00
Cliente: Óscar Navarro | Total gastado: 30.00
Cliente: Lucía Herrera | Total gastado: 30.00
Cliente: Raúl Pérez | Total gastado: 30.00
Cliente: Jorge Castillo | Total gastado: 30.00
Cliente: Raquel Fuentes | Total gastado: 30.00
Cliente: Sofía Torres | Total gastado: 30.00
Cliente: Verónica Campos | Total gastado: 30.00
Cliente: Cristina Navarro | Total gastado: 30.00
Cliente: Daniela Ramírez | Total gastado: 30.00
```

Opción 8

Como he comentado anteriormente por cada cliente muestra la habitación que tiene reservada, el nombre del cliente, el precio por noche y el número de noches y el total gastado en servicios y reserva.

```
Introduce una opción: 8
============================================================
INFORME COMPLETO DE INGRESOS POR RESERVA
============================================================
Cliente: Cristina Navarro
Habitación: 901
Precio por noche: 200.00
Noches: 4
Total servicios: 30.00
Total reserva: 830.00
------------------------------------------------------------
```

Opción 9

Con esta opción salimos al menú de elección para elegir el sistema.

```
Introduce una opción: 9
Volviendo al menú de selección de SGBD...
>> Base de datos desconectada.

============================================================
MENÚ DE CONEXIÓN A SGBD
============================================================
    1. MySQL / MariaDB
    2. PostgreSQL
    3. Oracle
    4. Salir
```

Oracle

Para Oracle tendremos que tenerlo instalado ya sea en local, en una máquina virtual o en un servidor remoto como puede ser un VPS.

```
============================================================
MENÚ DE CONEXIÓN A SGBD
============================================================
    1. MySQL / MariaDB
    2. PostgreSQL
    3. Oracle
    4. Salir

Introduce una opción: 3
>> Conectado a Oracle correctamente.
```

Para la opción 1 tenemos lo mismo que es los 2 sistemas, nos filtra por tipo y nos muestra cuanto hay de cada tipo.

```
Introduce una opción: 1
============================================================
RESUMEN DE HABITACIONES POR TIPO
============================================================
Tipo: Individual | Cantidad: 16
Tipo: Doble | Cantidad: 17
Tipo: Suite | Cantidad: 17
```

Para la opción 2 buscamos habitaciones en un rango, si no hay ninguno no nos muestra ninguna.

```
Introduce una opción: 2
Precio mínimo: 10
Precio máximo: 40
============================================================
HABITACIONES DISPONIBLES
============================================================
No se encontraron registros.
```

Para la opción 3 vamos a poder buscar las reservas filtrando por el DNI del cliente.

```
Introduce una opción: 3
Introduce el DNI del cliente: 10111213W
============================================================
RESERVAS DEL CLIENTE
============================================================
Nombre: Alicia Campos | Fecha entrada: 2024-04-16 00:00:00 | Habitación: 805
```

Para la opción 4 vamos a dar de alta a un nuevo cliente, una vez que lo hayamos añadido vamos a ver que se haya añadido correctamente a nuestro sistema gestor de bases de datos.

```
Introduce una opción: 4
============================================================
REGISTRO DE NUEVO CLIENTE
============================================================
ID Cliente: 51
Nombre: Jesús Figueroa Roldán
Email: jesusfigueroa@iesgn.com
Teléfono: 626272173
DNI: 28727181A
>> Cliente registrado correctamente.
```

Y como vemos ahi tenemos la entrada que hemos añadido.

```
ID_CLIENTE NOMBRE                                        EMAIL                                                TELEFONO        DNI             FECHA_RE
---------- ------------------------------------------------ ---------------------------------------------------------- --------------- --------------- --------
        48 Antonio Moreno                                   antonio.moreno@mail.com                              611333444       14151617X       17/02/24
        49 Cristina Navarro                                 cristina.navarro@mail.com                            611555666       18192022Y       18/02/24
        50 David Herrera                                    david.herrera@mail.com                               611777888       22232426Z       19/02/24
        51 Jesús Figueroa Roldán                            jesusfigueroa@iesgn.com                                 626272173      28727181A       05/02/26

51 filas seleccionadas.

SQL>
```

Ahora para la opción 5 vamos a poder eliminar entradas de la tabla servicios_reserva por id reserva.

```
SQL> select * from servicios_reserva;

ID_SERVICIO ID_RESERVA SERVICIO                PRECIO FECHA_SE
----------- ---------- -------------------- ---------- --------
          2          2 Desayuno                     10 03/03/24
          3          3 Cena                         15 04/03/24
          4          4 Spa                          30 05/03/24
          5          5 Desayuno                     10 06/03/24
```

Como vemos nos pedirá un id de reserva y lo borramos de nuestra tabla.

```
Introduce una opción: 5
============================================================
ELIMINAR SERVICIOS DE RESERVA
============================================================
Introduce ID de reserva para borrar sus servicios: 2
>> Servicios eliminados. Filas afectadas: 1
```

Si nos vamos a Oracle vemos como lo ha borradod correctamente y que ya no tenemos esa entrada en la tabla servicios_reserva.

```
SQL> select * from servicios_reserva where id_reserva = 2;

ninguna fila seleccionada

SQL>
```

Para la opción 6 vamos a poder incrementar el precio por piso en un tanto por ciento que le indiquemos al programa. Esto sería una muestra antes de ejecutar esta opción.

```
SQL> select * from habitaciones where piso = 3;

ID_HABITACION     NUMERO TIPO                      PISO  CAPACIDAD PRECIO_NOCHE D
------------- ---------- -------------------- ---------- ---------- ------------ -
           13        301 Individual                    3          1           60 S
           14        302 Doble                         3          2           80 S
           15        303 Suite                         3          3          140 S
           16        304 Individual                    3          1           60 S
           17        305 Doble                         3          2           80 S
           18        306 Suite                         3          3          140 S

6 filas seleccionadas.
```

Nos vamos a nuestro programa vamos a elegir por ejemplo el piso 1 y vamos a incrementar un 25%.

```
Introduce una opción: 6
============================================================
ACTUALIZAR PRECIO POR PISO
============================================================
Número de piso a actualizar: 3
Porcentaje de incremento (ej: 10): 25
>> Precios actualizados. Filas actualizadas: 6
```

Cuando nos vamos a Oracle y vamos los precios vemos como ha incrementado un 25%.

```
SQL> select * from habitaciones where piso = 3;

ID_HABITACION     NUMERO TIPO                      PISO  CAPACIDAD PRECIO_NOCHE D
------------- ---------- -------------------- ---------- ---------- ------------ -
           13        301 Individual                    3          1           75 S
           14        302 Doble                         3          2          100 S
           15        303 Suite                         3          3          175 S
           16        304 Individual                    3          1           75 S
           17        305 Doble                         3          2          100 S
           18        306 Suite                         3          3          175 S

6 filas seleccionadas.
```

Bibliografía

https://python-oracledb.readthedocs.io/en/latest/user_guide/installation.html

https://www.josedomingo.org/pledin/2021/12/python3-mysql/

https://www.psycopg.org/docs/

https://mariadb.com/resources/blog/how-to-connect-python-programs-to-mariadb/