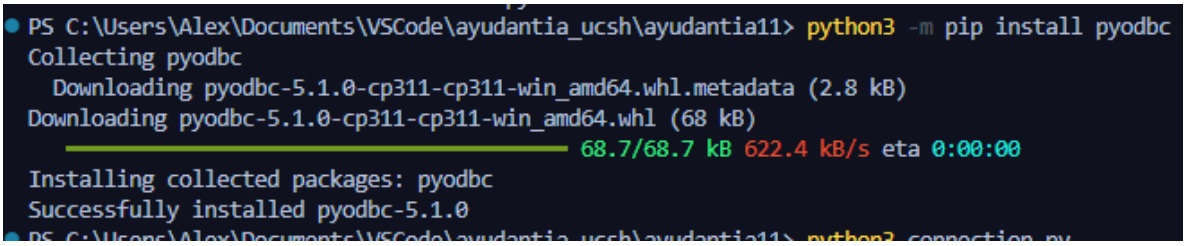
# Ayudantía 11

Alexander Inostroza – alexander.inostroza@usm.cl

**Conectarse a una DB**

Para conectarnos a una base de datos usando Python, primero debemos instalar un driver ODBC (Open DataBase Connectivity) en nuestro equipo. Algunos drivers ODBC se instalan al instalar una base de datos, como ocurre con SQL Server Express. El driver ODBC a instalar depende completamente de la base de datos que utilicen.

Para utilizar el driver ODBC en Python, utilizaremos pyodbc. Para instalar, en la cmd de Windows usar **python3 -m pip install pyodbc** (habiendo instalado previamente Python y pip).




También puede servir **python -m pip install pyodbc** .

Para conectarnos a nuestra base de datos, después de importar la librería pyodbc, debemos conectarnos usando una cadena de conexión:

```
import pyodbc

str_de_conexion = "DRIVER={SQL Server};\
    Server=localhost\SQLEXPRESS01;\
        Database=MultiUSM;\
            Trusted_Connection=True;"
```

Este es un ejemplo de una cadena en la que nos conectamos a una base de datos SQL Server instalada localmente. La cadena de conexión a utilizar depende de la base de datos a la que nos queremos conectar. Pueden encontrar más cadenas de conexión en connectionstrings.com .

Cuando utilizamos pyodbc, utilizamos un cursor para representar la conexión:

```
def main():
    connection = pyodbc.connect(str_de_conexion)
    cursor = connection.cursor()

    cursor.execute("CREATE TABLE Productos (prod_id bigint,\
                    prod_name VARCHAR(150), prod_description VARCHAR(150),\
                    prod_brand VARCHAR(150), category VARCHAR(150), \
                   prod_unit_price int)")

    connection.commit()
    cursor.close()
    connection.close()
```

Para obtener datos de la DB, podemos usar **fetchall** o **fetchone**.

```
nombre_producto = "algun producto"
cursor.execute("SELECT * FROM Productos WHERE prod_name='"+ nombre_producto +"'")
prod_encontrado = cursor.fetchall()
if len(prod_encontrado) > 0:
    # pueden haber varios productos con el mismo nombre
    for prod_id, prod_name, prod_description, prod_brand, category, prod_unit_price in
prod_encontrado:
        print("----------------------------------------------------------------")
        print("ID del producto: {}".format(prod_id))
        print("Nombre: {}".format(prod_name))
        print("Descripción: {}".format(prod_description))
        print("Marca: {}".format(prod_brand))
        print("Categoría: {}".format(category))
        print("Precio: {}".format(prod_unit_price))
        print("----------------------------------------------------------------")
else:
    print("----------------------------------------------------------------")
    print("PRODUCTO NO ENCONTRADO")
    print("----------------------------------------------------------------")
```

Toda la documentación de pyodbc está disponible en su wiki https://github.com/mkleehammer/pyodbc/wiki .

Pueden ocupar como referencia, si les es útil, una tarea que hice hace un tiempo. Ver archivo **main.py** .

Finalmente, si les interesa hacerlo con **Java**, la alternativa es **JDBC**.