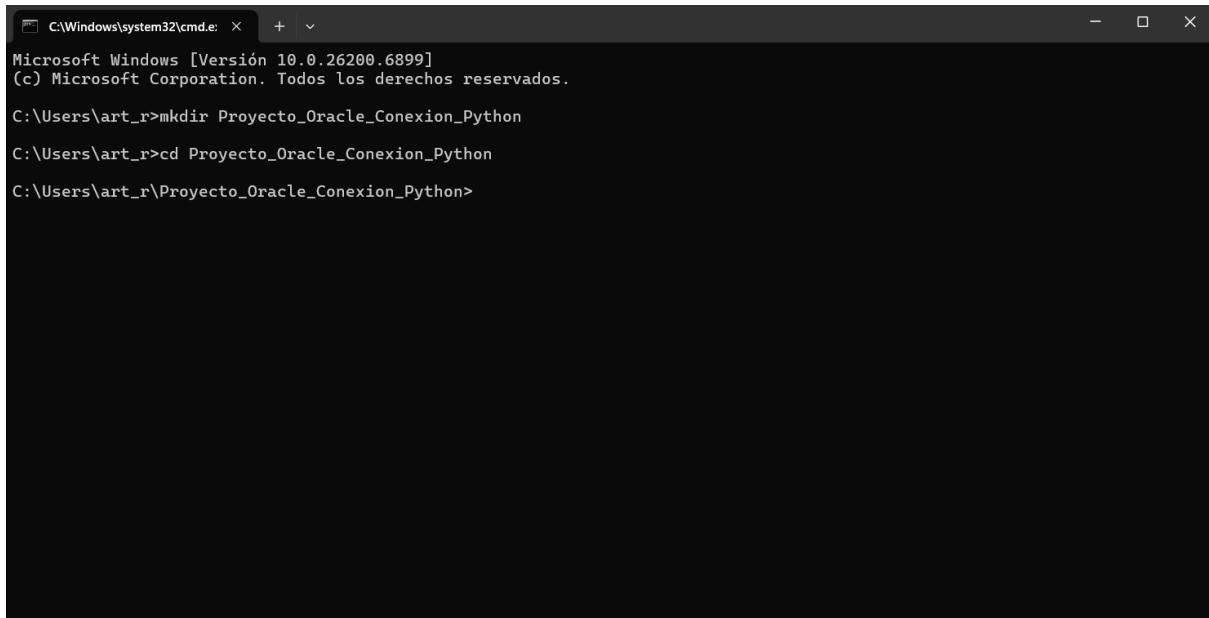


Documentación: Conexión a Oracle 11g desde Python. Elaboración de plantilla para Github.

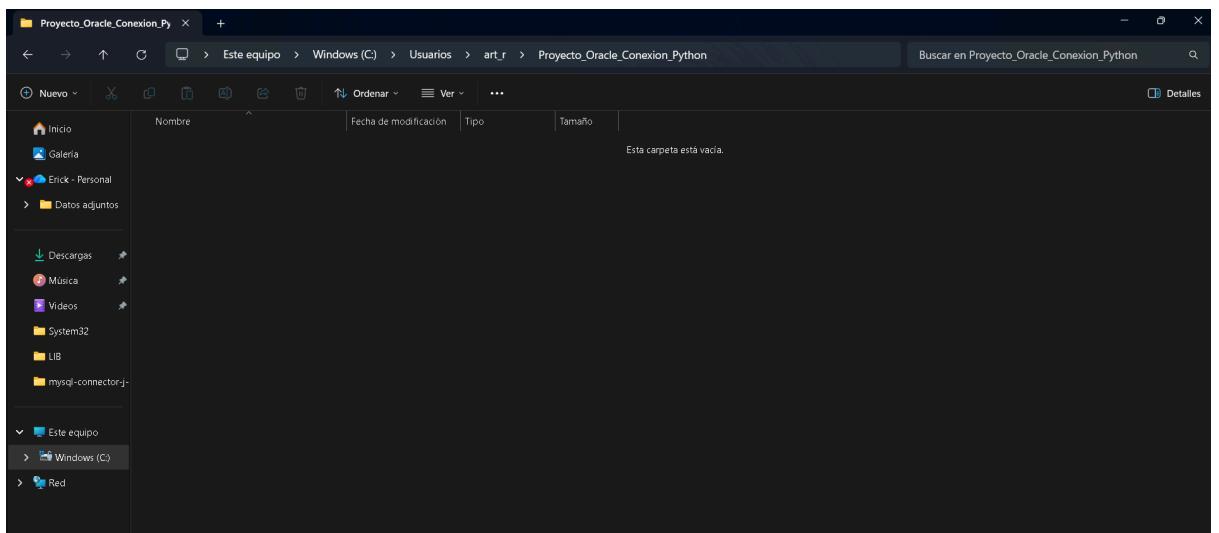
1. Crear una carpeta repositorio con toda la información que se subirá a Github. Esta carpeta del proyecto se encontrará estructurada para distribuir de forma ordenada los datos a cargar. En CMD creamos un directorio para poder asignar una carpeta fuente.



```
C:\Windows\system32\cmd.exe: × + ▾ Microsoft Windows [Versión 10.0.26200.6899]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\art_r>mkdir Proyecto_Oracle_Conexion_Python
C:\Users\art_r>cd Proyecto_Oracle_Conexion_Python
C:\Users\art_r\Proyecto_Oracle_Conexion_Python>
```

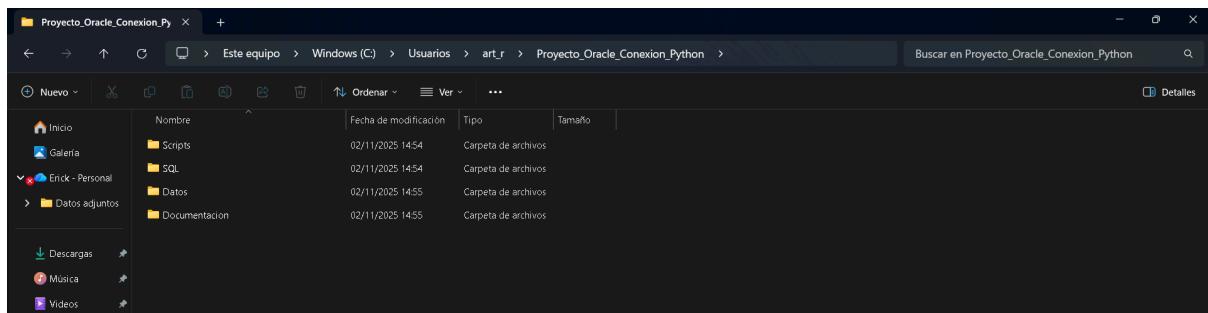
- Al escribir “`mkdir Proyecto_Oracle_Conexion_Python`” en el bash, creamos una carpeta base. Y al escribir después el comando “`cd Proyecto_Oracle_Conexion_Python`” asignamos una ubicación raíz.
- Seguido a esto, la ubicación sería la siguiente:
`C:\Users\art_r\Proyecto_Oracle_Conexion_Python>`



2. En estos momentos la ruta se encuentra vacía, por lo que crearemos una serie de carpetas donde estructuramos todos los archivos.

Dentro de la carpeta del proyecto, creamos las carpetas con nombre:

- Scripts: La carpeta donde irán nuestros scripts en Python.
- SQL: Donde se contendrán los archivos .sql del proyecto.
- Datos: En esta carpeta se almacenan todos los archivos necesarios.
- Documentación: Contiene notas y datos auxiliares, sirven de ayuda.



3. Empezando con los scripts de Python, es necesario tenerlo instalado antes. Para revisar si está instalado en el equipo ir a CMD y escribir el comando “`python --version`”.

```
Microsoft Windows [Versión 10.0.26200.6899]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\art_r>python --version
Python was not found; run without arguments to install from the Microsoft Store, or disable this shortcut from Settings
> Apps > Advanced app settings > App execution aliases.

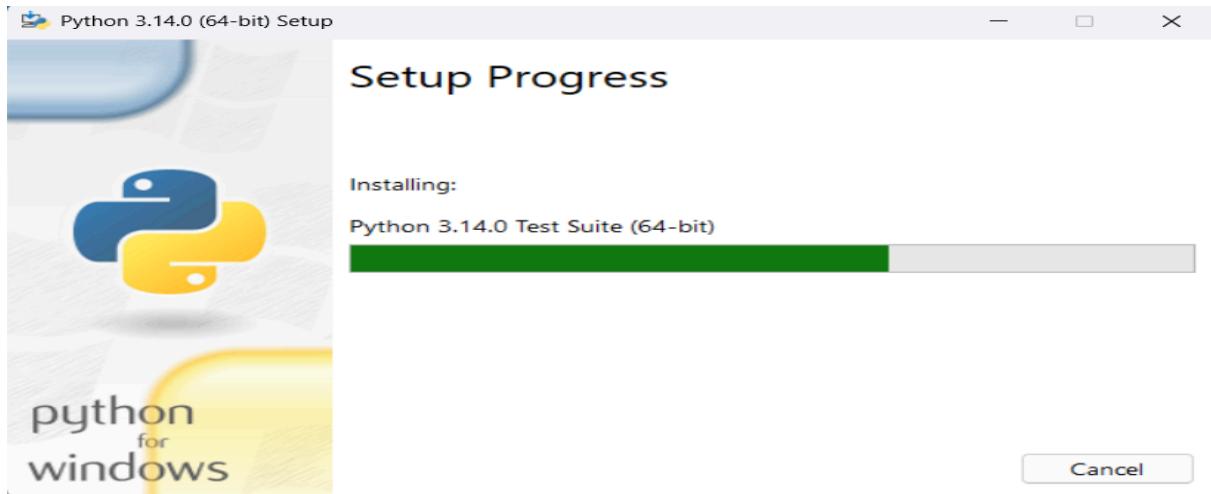
C:\Users\art_r>
```

- Si el mensaje aparece como “...not found..”, entonces instalarlo antes de continuar en la página oficial: [Python Release Python 3.14.0 | Python.org](https://www.python.org/downloads/windows/), y darle a la última versión de Python 3. Una vez listo, seguir todo el proceso de instalación eligiendo la versión de win64.exe si este es su OS predeterminado (si ya lo tenía instalado, omita esta parte).

The screenshot shows a web browser displaying the Python.org download page for Windows. The URL in the address bar is <https://www.python.org/downloads/windows/>. The page title is 'Python Releases for Windows'. Below the title, there are two bullet points:

- Latest Python install manager - [Python install manager 25.0](#)
- Latest Python 3 Release - [Python 3.14.0](#)

- Dentro del instalador, es necesario darle al instalador incluyendo Python al PATH.
- Dentro de la opción final, darle aceptar a la opción de quitar el limitador de cantidad de caracteres en PATH (necesario para proyectos amplios).



4. Una vez instalado Python, revisar en el CMD para asegurarnos de que esté activo.

```
C:\Windows\system32\cmd.e: + 
Microsoft Windows [Versión 10.0.26200.6899]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\art_r>python --version
Python 3.14.0

C:\Users\art_r>
```

5. Dentro del CMD, nos aseguramos que los gestores de paquetes ‘pip’ de Python estén instalados antes de proceder con las librerías con el comando: `python -m pip --version`

```
C:\Windows\system32\cmd.e: + 
Microsoft Windows [Versión 10.0.26200.6899]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\art_r>python -m pip --version
pip 25.2 from C:\Users\art_r\AppData\Local\Programs\Python\Python314\Lib\site-packages\pip (python 3.14)

C:\Users\art_r>
```

- Si lo que aparece es el siguiente mensaje, entonces confirmamos que Python está instalado correctamente y las librerías también están listas para usarse.
6. Una vez listo lo anterior, procedemos a instalar el módulo “oracledb”, el cual es el puente entre los scripts en Python para poder conectarse con las bases de datos de Oracle. En el CMD, usamos el comando: “`python -m pip install oracledb`”.

```
C:\Windows\system32\cmd.exe + - x
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\art_r>python -m pip --version
pip 25.2 from C:\Users\art_r\AppData\Local\Programs\Python\Python314\Lib\site-packages\pip (python 3.14)

C:\Users\art_r>python -m pip install oracledb
Collecting oracledb
  Downloading oracledb-3.4.0-cp314-cp314-win_amd64.whl.metadata (7.9 kB)
Collecting cryptography>=3.2.1 (from oracledb)
  Downloading cryptography-46.0.3-cp311-abi3-win_amd64.whl.metadata (5.7 kB)
Collecting typing_extensions>=4.14.0 (from oracledb)
  Downloading typing_extensions-4.15.0-py3-none-any.whl.metadata (3.3 kB)
Collecting cffi>=2.0.0 (from cryptography>=3.2.1->oracledb)
  Downloading cffi-2.0.0-cp314-cp314-win_amd64.whl.metadata (2.6 kB)
Collecting pycparser (from cffi>=2.0.0->cryptography>=3.2.1->oracledb)
  Downloading pycparser-2.23-py3-none-any.whl.metadata (993 bytes)
Collecting oracledb-3.4.0-cp314-cp314-win_amd64.whl (1.9 MB)
  1.9/1.9 MB 7.8 MB/s 0:00:00
  3.5/3.5 MB 7.7 MB/s 0:00:00
Downloading cryptography-46.0.3-cp311-abi3-win_amd64.whl (3.5 MB)
  3.5/3.5 MB 7.7 MB/s 0:00:00
Downloading cffi-2.0.0-cp314-cp314-win_amd64.whl (189 kB)
Downloading typing_extensions-4.15.0-py3-none-any.whl (44 kB)
Downloading pycparser-2.23-py3-none-any.whl (118 kB)
Installing collected packages: typing_extensions, pycparser, cffi, cryptography, oracledb
Successfully installed cffi-2.0.0 cryptography-46.0.3 oracledb-3.4.0 pycparser-2.23 typing_extensions-4.15.0

[notice] A new release of pip is available: 25.2 -> 25.3
[notice] To update, run: python.exe -m pip install --upgrade pip
C:\Users\art_r>
```

- Una vez listo, se confirma la actualización de dependencias necesarias para la lectura de los scripts, y también ahora podemos conectarnos directamente a la base de datos que tengamos lista.
- Para asegurarnos de que el módulo “oracledb” se encuentra listo para funcionar, probamos el siguiente comando en el CMD accediendo a Python:

I. > python

- Permite escribir comandos de Python dentro de la consola, esto gracias a instalar Python y su conjunto de librerías dentro del sistema.

```
>>>import oracledb
```

```
>>>print(oracledb.version)
```

- Se importa el módulo Oracle para verificar su estado. Si se encuentra instalado te dejará acceder a funciones de consulta desde Python a Oracle.
- El *print* lo que hace es mostrar la versión y el estatus en el que se encuentra.

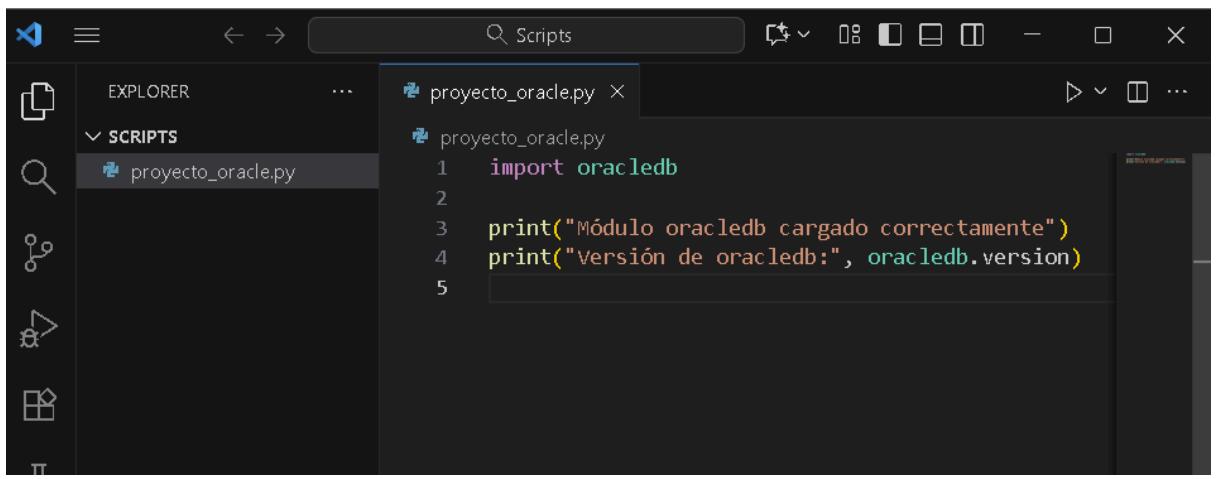
```
>>>exit()
```

- Sale de la operación.

```
C:\Users\art_r>python
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct  7 2025, 10:15:03) [MSC v.1944 64
  bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import oracledb
>>> print(oracledb.version)
3.4.0
>>> exit()

C:\Users\art_r>
```

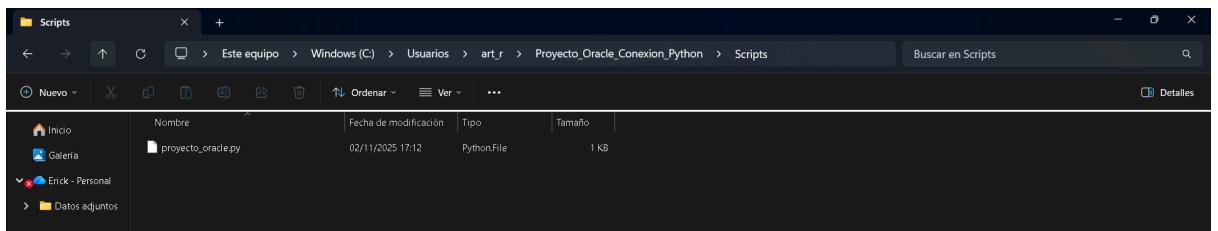
7. Procedemos a crear un archivo .py donde importamos el módulo “oracledb”, para asegurarnos que es posible ejecutarlo. Vamos al directorio de la carpeta Scripts, y ejecutamos en el CMD el .py con los mismos comandos utilizados antes.



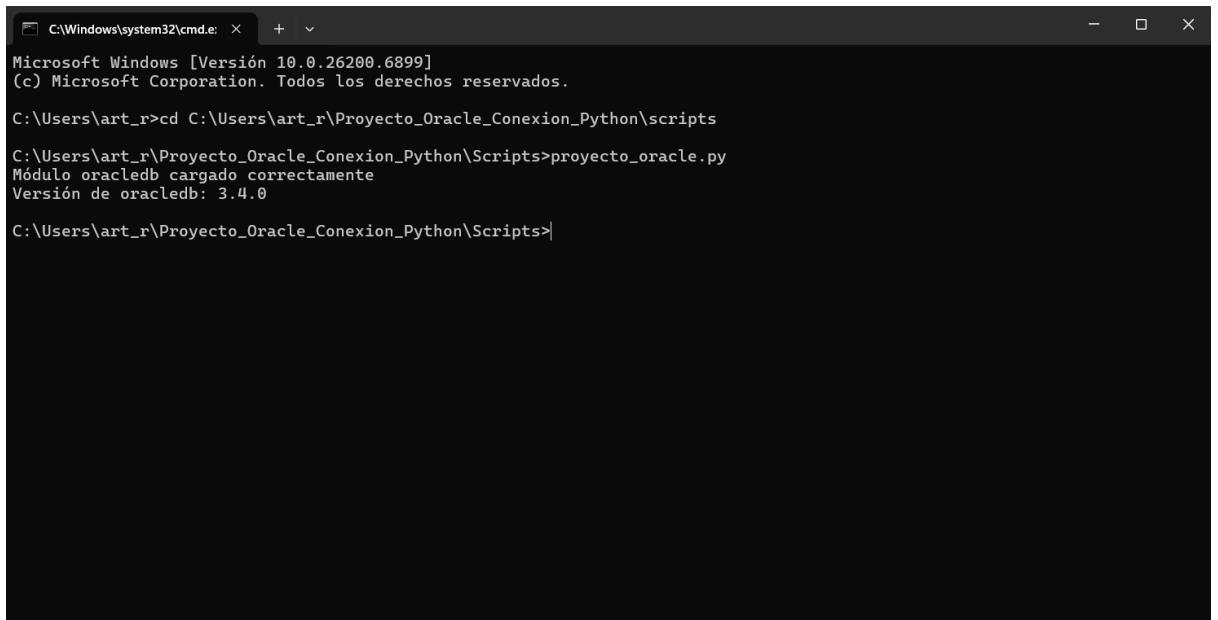
The screenshot shows a code editor window titled "Scripts". In the left sidebar, under "EXPLORER", there is a folder named "SCRIPTS" which contains a file named "proyecto_oracle.py". The main pane displays the contents of this file:

```
proyecto_oracle.py
1 import oracledb
2
3 print("Módulo oracledb cargado correctamente")
4 print("Versión de oracledb:", oracledb.version)
5
```

Según la ruta:



Desde el CMD:



The screenshot shows a Windows Command Prompt window with the following session:

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Versión 10.0.26200.6899]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\art_r>cd C:\Users\art_r\Proyecto_Oracle_Conexion_Python\scripts
C:\Users\art_r\Proyecto_Oracle_Conexion_Python\Scripts>proyecto_oracle.py
Módulo oracledb cargado correctamente
Versión de oracledb: 3.4.0

C:\Users\art_r\Proyecto_Oracle_Conexion_Python\Scripts>
```

- Como se puede observar, el servicio se encuentra activo y el módulo es accesible desde el CMD.

8. Antes de proceder, tener en cuenta que la versión del Oracle es el 11g XE, por lo tanto es una versión predecesora a las modernas. Para asegurar la compatibilidad entre Oracle y los scripts en Python, instalamos el Oracle Instant Client, ya que esta versión

de Oracle necesita de librerías nativas que no tienen versiones más modernas, además que nos abre funciones más extensas para manipular los datos de nuestras tablas sql.

Vamos a Link: [Oracle Instant Client Downloads](https://www.oracle.com/database/technologies/instant-client/downloads.html?utm_source=chatgpt.com) para descargar el OIC, y seleccionamos la versión de winx64 la cual es la versión que utilizaremos. Una vez escogido, nos redirigirá a una pestaña de versiones, y elegimos la versión Basic Package. Lo descargamos como .zip y procedemos a instalarlo.

- La página (Seleccionamos el que se encuentra en el cursor):

The screenshot shows the Oracle Instant Client Downloads page. It lists download links for various platforms:

- Instant Client for Microsoft Windows:**
 - [Instant Client for Microsoft Windows \(x64\)](#)
 - [Instant Client for Microsoft Windows \(32-bit\)](#)
- Instant Client for Linux:**
 - [Instant Client for Linux x86-64](#)
 - [Instant Client for Linux x86](#)
 - [Instant Client for Linux ARM \(arch64\)](#)
 - [Instant Client for z/Linux \(64-bit\)](#)
- Instant Client for HP:**
 - [Instant Client for HP-UX Itanium \(64-bit\)](#)
 - [Instant Client for HP-UX Itanium \(32-bit\)](#)
- Instant Client for macOS:**
 - [Instant Client for macOS \(ARM64\)](#)
 - [Instant Client for macOS \(Intel x86\)](#)
- Instant Client for Solaris:**
 - [Instant Client for Solaris Operating System \(SPARC\) \(64-bit\)](#)
 - [Instant Client for Solaris Operating System \(SPARC\) \(32-bit\)](#)
 - [Instant Client for Solaris x86-64](#)
 - [Instant Client for Solaris x86](#)
- Instant Client for AIX:**
 - [Instant Client for AIX \(PPC64\)](#)
 - [Instant Client for AIX \(PPC32\)](#)

Other Platforms

- Seleccionamos la versión Basic Package:

The screenshot shows the Oracle Instant Client Downloads for Microsoft Windows (x64) 64-bit page. It highlights the "Basic" package download link:

See the [Instant Client Home Page](#) for more information about instant Client.
The [Installation instructions](#) are at the foot of the page.
Client-server version interoperability is detailed in [Doc ID 2073031](#). For example, Oracle Call Interface 23 can connect to Oracle Database 19c or later, while Oracle Call Interface 19.3 can connect to Oracle Database 11.2 or later. Some tools may have other restrictions.
Permanent links to the latest packages are: [Basic](#), [Basic Light](#), [SQL*Plus](#), [Tools](#), [SDK](#), [JDBC Supplement](#), [ODBC](#)

Version 23.9.0.25.07

Base - one of these packages is required

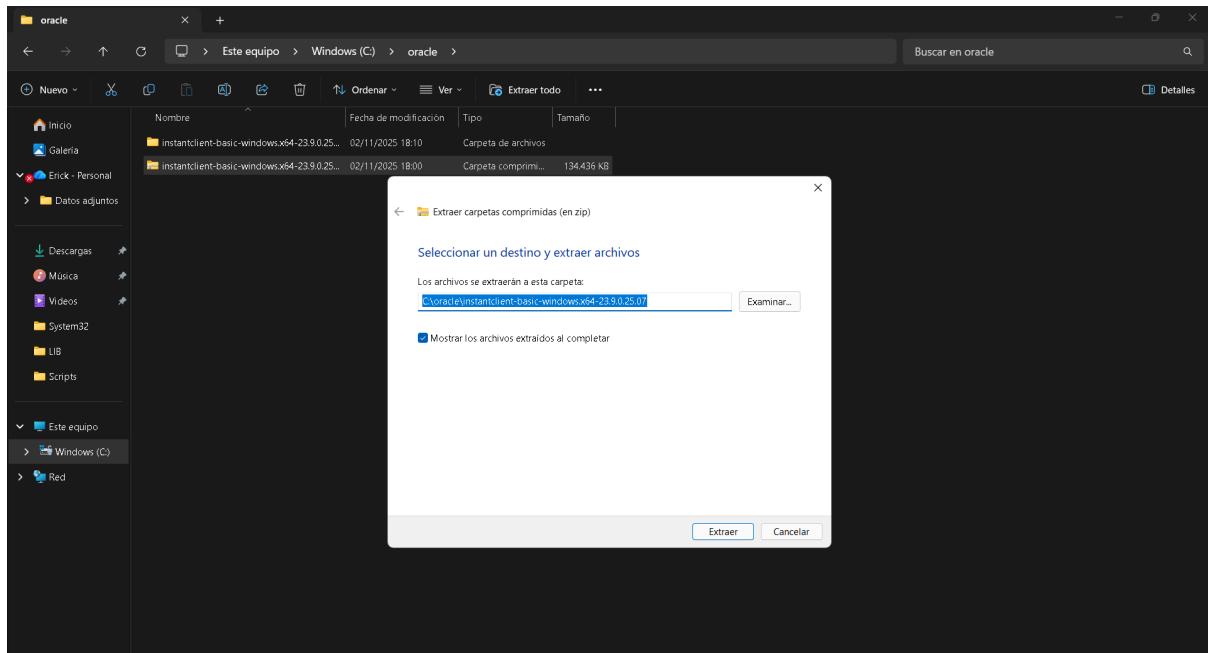
Name	Download	Description
Basic Package	instantclient-basic-windows.x64-23.9.0.25.07.zip	All files required to run OCI, OCCI, and JDBC:OCI applications (157,662,217 bytes) (cksum - 1523760) Review the Operating System Check Database Client Installation . Note Wi supported. The 23.9 Basic package requires the Visual C++ Redistributable package

Open all

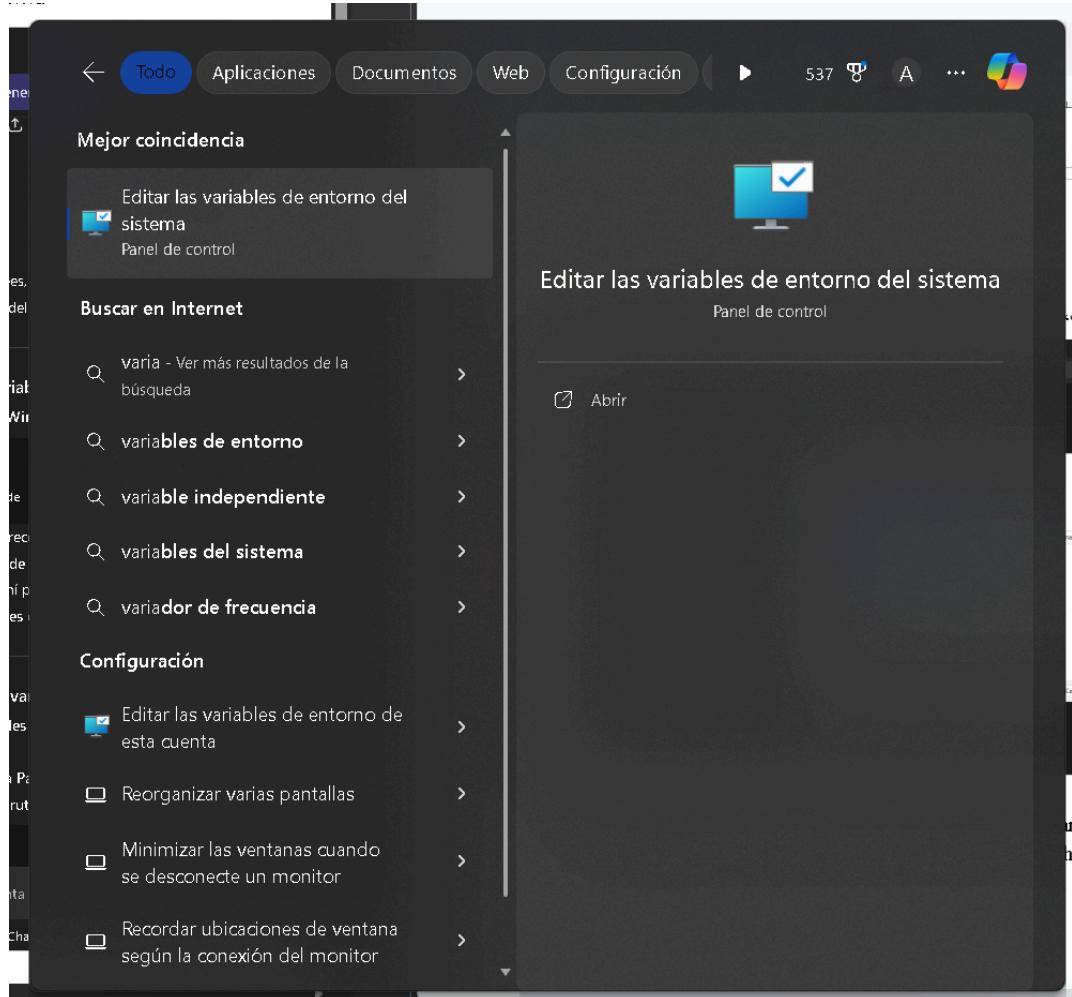
Chat now

Call US Sales +1.800.633.0738

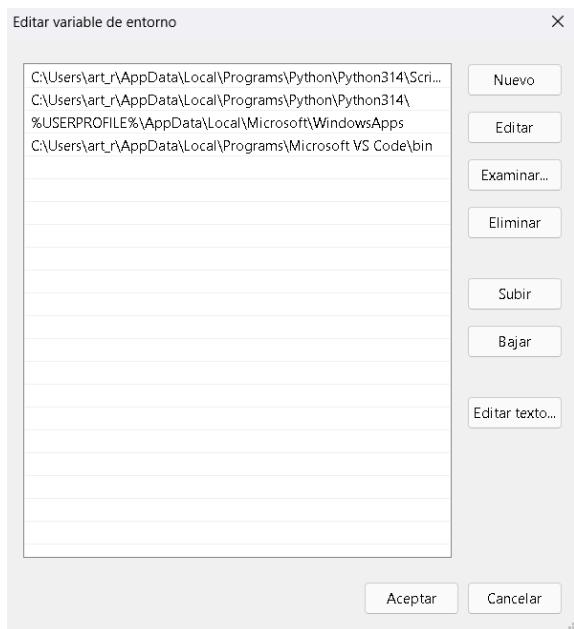
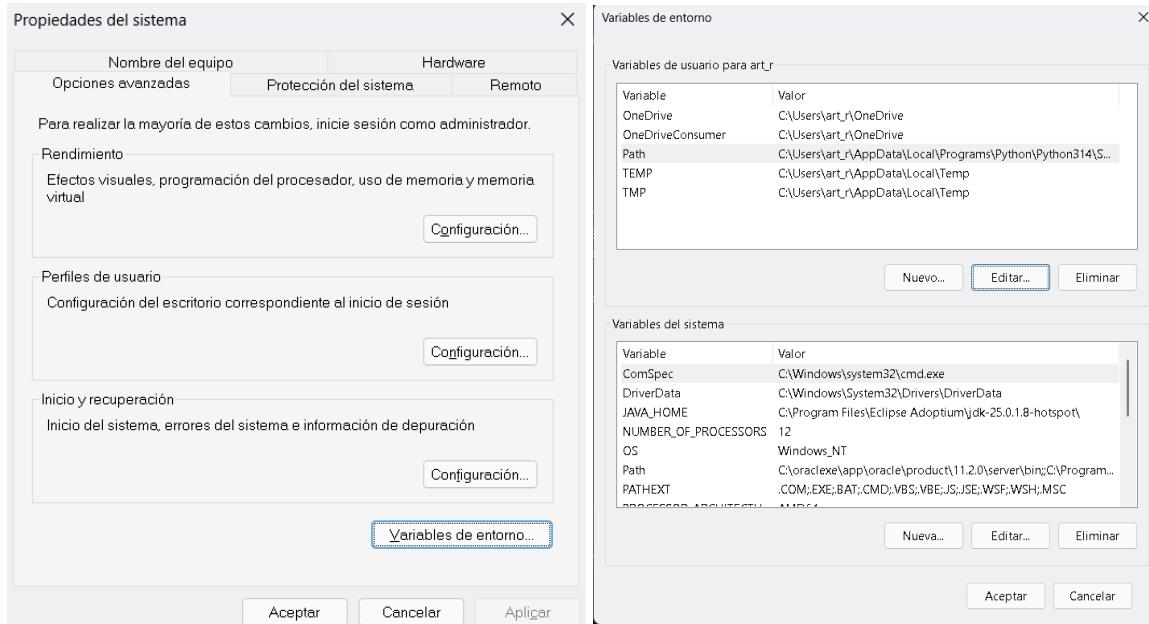
- Creamos un directorio en: “ C:\oracle\instantclient_11_2“, y descomprimimos el .zip allí.



9. Para configurar el PATH correctamente y asegurar el funcionamiento de Python, vamos al entorno del sistema de Windows para cambiar la ruta con el archivo Instant Client:



10. Dentro del recuadro de Propiedades del sistema, vamos a Opciones avanzadas > Variables del entorno > En variables del entorno del usuario, seleccionamos el Path y luego en Editar > Aparecerá una lista, le damos a Nuevo y añadimos la ruta del Instant Client, y le damos Aceptar en todas las ventanas. Una vez realizado todo esto revisamos el CMD y nos dará una confirmación de que todo funciona correctamente.



```

Microsoft Windows [Versión 10.0.26200.6899]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\art_r>echo %PATH%
C:\oraclexe\app\oracle\product\11.2.0\server\bin;;C:\Program Files\Eclipse Adoptium\jdk-25.0.1.8-hotspot\bin;C:\Windows\system32;C:\Windows;C:\Windows\System32\Wbem;C:\Windows\System32\WindowsPowerShell\v1.0\;C:\Windows\System32\OpenSSH\;C:\Program Files\dotnet\;C:\Users\art_r\AppData\Local\Programs\Python\Python314\Scripts\;C:\Users\art_r\AppData\Local\Programs\Python\Python314\;C:\Users\art_r\AppData\Local\Microsoft\WindowsApps;C:\Users\art_r\AppData\Local\Programs\Microsoft VS Code\bin;C:\oracle\instantclient-basic-windows.x64-23.9.0.25.07\instantclient_23_9;

C:\Users\art_r>python
Python 3.14.0 (tags/v3.14.0:ebf955d, Oct  7 2025, 10:15:03) [MSC v.1944 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import oracledb
>>> oracledb.init_oracle_client()
>>> print("Oracle Client cargado correctamente")
Oracle Client cargado correctamente
>>>

```

- Con el comando echo %PATH% vemos que aparece el Instant Client y su versión.

11. Una vez lista la conexión, creamos los scripts necesarios y los añadimos a la carpeta Scripts de nuestro proyecto, los cuales son:

- conexion_oracle.py >Define la conexión a Oracle
- crear_tablas.py >Crea la estructura de las tablas con relaciones y restricciones.
- insertar_datos.py >Inserta los registros y datos a cada tabla del SQL
- manipular_tablas.py >Permite hacer consultas para verificar los datos de las tablas.
- proyecto_oracle.py >Script principal que integra las funciones anteriores.

a. Script > conexion_oracle.py

```

# conexion_oracle.py
import oracledb

# Inicializa el cliente Oracle (modo thick)
oracledb.init_oracle_client(lib_dir=r"C:\oracle\instantclient-basic-windows.x64-23.9.0.25.07\instantclient_23_9")

def get_connection():
    """Retorna una conexión a la base de datos Oracle."""
    return oracledb.connect(
        user="SYSTEM",
        password="ORACLE",
        dsn="localhost/XE"

```

```

)
if __name__ == "__main__":
    try:
        conn = get_connection()
        print("Conexión exitosa")
        conn.close()
    except Exception as e:
        print("Error al conectar:", e)
print("Conexión exitosa")

```

b. Script >crear_tablas.py

```

# crear_tablas.py
import oracledb
from conexion_oracle import get_connection # Reutiliza la conexión

conn = get_connection()
cur = conn.cursor()

# Crear las tablas según la estructura proporcionada
cur.execute("""
CREATE TABLE CLIENTES (
    Cod_Client NUMBER(5) NOT NULL,
    Nombre VARCHAR2(30) NOT NULL,
    Apellido VARCHAR2(30) NOT NULL,
    DNI NUMBER(8) UNIQUE NOT NULL,
    Telefono NUMBER(9) NOT NULL,
    Correo VARCHAR2(30) UNIQUE NOT NULL,
    Direccion VARCHAR2(30) NOT NULL,
    CONSTRAINT PK_CLIENTES PRIMARY KEY (Cod_Client)
)
""")

cur.execute("""
CREATE TABLE PRODUCTOS (
    Cod_Prod NUMBER(8) NOT NULL,
    Nom_Prod VARCHAR2(100) NOT NULL,
    Precio NUMBER(6, 2) NOT NULL,
    Stock NUMBER NOT NULL,
    CONSTRAINT pk_productos PRIMARY KEY (Cod_Prod),
    CONSTRAINT chk_precio_stock CHECK (precio >= 0 AND stock >= 0)
)
""")

```

```

""")

cur.execute("""
CREATE TABLE PEDIDOS (
    Cod_Ped NUMBER(10) NOT NULL,
    Cod_Client NUMBER(5) NOT NULL,
    fecha_pedido DATE DEFAULT SYSDATE NOT NULL,
    estado VARCHAR2(50) DEFAULT 'Pendiente' NOT NULL,
    CONSTRAINT pk_pedidos PRIMARY KEY (Cod_Ped),
    CONSTRAINT fk_pedidos_cliente FOREIGN KEY (Cod_Client)
        REFERENCES CLIENTES(Cod_Client)
)
""")

cur.execute("""
CREATE TABLE DETALLE_PEDIDO (
    Cod_ped NUMBER NOT NULL,
    Cod_Prod NUMBER NOT NULL,
    Cantidad NUMBER NOT NULL,
    Precio_Total NUMBER(8, 2) NOT NULL,
    CONSTRAINT pk_detalle_pedido PRIMARY KEY (Cod_Ped, Cod_Prod),
    CONSTRAINT fk_detalle_pedido_ped FOREIGN KEY (Cod_Ped)
        REFERENCES PEDIDOS(Cod_Ped),
    CONSTRAINT fk_detalle_pedido_prod FOREIGN KEY (Cod_Prod)
        REFERENCES PRODUCTOS(Cod_Prod),
    CONSTRAINT chk_cantidad CHECK (cantidad > 0)
)
""")

conn.commit()
cur.close()
conn.close()
print("Tablas creadas correctamente")

```

c. Script > insertar_datos.py

```

# insertar_datos.py
from conexion_oracle import get_connection

# Datos a insertar
clientes = [

```

```

        (1, 'Sofía', 'Ramírez', 25896314, 945123789,
'sofia.r@mailcorp.com', 'Jr. Los Álamos 105'),
        (2, 'Javier', 'Méndez', 14785236, 933567123,
'javi.mendez@soluciones.net', 'Av. El Sol 402'),
        (3, 'Valeria', 'Castro', 36985214, 978456123,
'veleria.c@webmail.com', 'Calle Santa Elena 55'),
        (4, 'Carlos', 'Rojas', 95126347, 960987321, 'carlos.rojas@mail.pe',
'Jr. Los Sauces 88'),
        (5, 'Luisa', 'Herrera', 74125896, 921543876,
'luisa.herrera@techmail.com', 'Pasaje Las Flores 15'),
        (6, 'Andrés', 'Silva', 85214736, 998765432,
'andres.silva@inbox.com', 'Av. Primavera 777')
]

productos = [
    (101, 'Leche Fresca Entera 1L', 4.80, 150),
    (102, 'Gaseosa Cola 2.5L', 8.50, 90),
    (103, 'Bolsa de Papas Fritas 150g', 3.20, 240),
    (104, 'Pan de Molde Blanco', 6.90, 80),
    (105, 'Agua Mineral sin Gas 600ml', 1.50, 350),
    (106, 'Barra de Chocolate con Leche', 2.80, 180)
]

pedidos = [
    (10001, 1, '2025-11-01', 'Entregado'),
    (10002, 3, '2025-10-30', 'Pendiente'),
    (10003, 5, '2025-10-25', 'En Proceso'),
    (10004, 2, '2025-11-01', 'Entregado'),
    (10005, 4, '2025-10-28', 'Pendiente'),
    (10006, 6, '2025-10-20', 'Cancelado')
]

detalle_pedidos = [
    (10001, 101, 3, 14.40),
    (10002, 103, 2, 6.40),
    (10003, 102, 1, 8.50),
    (10004, 105, 4, 6.00),
    (10005, 106, 5, 14.00),
    (10006, 104, 1, 6.90)
]

# Conexión a la base
conn = get_connection()

```

```

cur = conn.cursor()

# Insertar datos
cur.executemany("""
    INSERT INTO CLIENTES (Cod_Client, Nombre, Apellido, DNI, Telefono,
Correo, Direccion)
    VALUES (:1, :2, :3, :4, :5, :6, :7)
""", clientes)

cur.executemany("""
    INSERT INTO PRODUCTOS (Cod_Prod, Nom_Prod, Precio, Stock)
    VALUES (:1, :2, :3, :4)
""", productos)

cur.executemany("""
    INSERT INTO PEDIDOS (Cod_Ped, Cod_Client, fecha_pedido, estado)
    VALUES (:1, :2, TO_DATE(:3, 'YYYY-MM-DD'), :4)
""", pedidos)

cur.executemany("""
    INSERT INTO DETALLE_PEDIDO (Cod_Ped, Cod_Prod, Cantidad,
Precio_Total)
    VALUES (:1, :2, :3, :4)
""", detalle_pedidos)

conn.commit()
cur.close()
conn.close()

print("Datos insertados correctamente.")

```

d. Script > manipular_datos.py

```

import oracledb

# Inicializa el cliente Oracle
oracledb.init_oracle_client(
    lib_dir=r"C:\oracle\instantclient-basic-windows.x64-23.9.0.25.07\instantclient_23_9"
)

```

```

def obtenerConexion():
    return oracledb.connect(
        user="SYSTEM",
        password="ORACLE",
        dsn="localhost/xe"
    )

def listarTablas():
    tablas = ['CLIENTES', 'PRODUCTOS', 'PEDIDOS', 'DETALLE_PEDIDO']
    conn = obtenerConexion()
    cur = conn.cursor()
    query = f"""
        SELECT table_name
        FROM user_tables
        WHERE table_name IN ({','.join("'" + t + "'" for t in tablas)})
        ORDER BY table_name
    """
    cur.execute(query)
    for row in cur:
        print(row[0])
    cur.close()
    conn.close()

if __name__ == "__main__":
    listarTablas()

```

e. Script > proyecto_oracle.py

```

import oracledb

print("Módulo oracledb cargado correctamente")
print("Versión de oracledb:", oracledb.version)

```

12. Ahora verificamos que los scripts y la información utilizada de nuestro SQL se encuentran funcionales. Vamos a CMD y usamos el comando **SQLPLUS**

SYSTEM/ORACLE@localhost/XE

Este comando nos permite ingresar a Oracle con los datos del sistema de tablas que hemos utilizado:

- **SQLPLUS**: El servicio que nos conecta a la base de datos de Oracle

- SYSTEM: El usuario (nosotros) que utilizamos
- ORACLE: La contraseña del usuario SYSTEM (establecida)
- localhost/XE: Indica la máquina donde se está ejecutando la base de datos y el XE es el nombre del servicio de la base de datos (Oracle Express Edition).

```
C:\Windows\system32\cmd.exe: × + ▾
Microsoft Windows [Versión 10.0.26200.6899]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\art_r>sqlplus system/ORACLE@localhost/XE

SQL*Plus: Release 11.2.0.2.0 Production on Dom Nov 2 19:55:34 2025

Copyright (c) 1982, 2014, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production

SQL> |
```

- Como se puede observar, la conexión con Oracle fue exitosa. Ahora probemos el comando SELECT para asegurarnos de que está conectada a la base de datos.

```
Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production

SQL> SELECT * FROM CLIENTES;

COD_CLIENT NOMBRE          APELLIDO
----- -----
DNI   TELEFONO CORREO
----- -----
DIRECCION
-----
1 Sofía                    Ramírez
25896314 945123789 sofia.r@mailcorp.com
Jr. Los Llamados 105

2 Javier                   Mendoza
14785236 935367123 javi.mendez@soluciones.net
Av. El Sol 402

COD_CLIENT NOMBRE          APELLIDO
----- -----
DNI   TELEFONO CORREO
----- -----
DIRECCION
-----
3 Valeria                 Castro
36985214 978456123 valeria.c@webmail.com
Calle Santa Elena 55

4 Carlos                   Rojas
95126347 960987321 carlos.rojas@mail.pe

COD_CLIENT NOMBRE          APELLIDO
----- -----
DNI   TELEFONO CORREO
----- -----
DIRECCION
-----
5 Luisa                   Herrera
74125896 921543876 luisa.herrera@techmail.com
Pasaje Las Flores 15

6 Andrés                  Silva
COD_CLIENT NOMBRE          APELLIDO
----- -----
DNI   TELEFONO CORREO
----- -----
DIRECCION
-----
85214736 998765432 andres.silva@inbox.com
Av. Primavera 777

6 rows selected.

SQL> |
```

- Como se puede apreciar, la conexión fue un éxito. La selección de datos de cada una de las tablas es realizable y funcional.
13. Una vez ordenados los scripts y teniendo listas las demás carpetas agregadas, nuestro entorno de carpetas quedaría así:

```

Listado de rutas de carpetas para el volumen Windows
El n mero de serie del volumen es 5E3B-C836
C:.

-- Datos
    OracleXE112_Win64.zip
    python-3.14.0-amd64.exe
    README.txt
    VSCodeUserSetup-x64-1.105.1.exe

    instantclient-basic-windows.x64-23.9.0.25.07
        instantclient_23_9
            adrci.exe
            adrci.sym
            BASIC_LICENSE
            BASIC_README
            extks.dll
            fips.dll
            genezi.exe
            genezi.sym
            legacy.dll
            oci.dll
            oci.sym
            ocijdbc23.dll
            ocijdbc23.sym
            ociw32.dll
            ociw32.sym
            ojdbc11.jar
            ojdbc17.jar
            ojdbc8.jar
            orannz.dll
            orannz.sym
            oraocci23.dll
            oraocci23.sym
            oraocci23d.dll
            oraocci23d.sym
            oraociei.dll
            oraociei.sym
            orasql.dll
            orasql.sym
            pkcs11.dll
            ucp.jar
            ucp11.jar
            ucp17.jar
            uidrvci.exe
            uidrvci.sym
            xstreams.jar

        network
            admin
                README

-- META-INF
    MANIFEST.MF
    ORACLE_C.RSA
    ORACLE_C.SF

-- Documentacion
    README.txt

-- Scripts
    conexion_oracle.py
    consultas.py
    crear_tablas.py
    insertar_datos.py
    manipular_tablas.py
    proyecto_oracle.py
    README.md

    __pycache__
        conexion_oracle.cpython-314.pyc

-- SQL
    ENTREGA 1.sql
    README.txt

PS C:\Users\art_r\Proyecto_Oracle_Conexion_Python> |

```

14. Una vez listo todo, vamos a Github e iniciamos sesi n. Si no tiene cuenta cree una para poder acceder.

The screenshot shows a Bing search interface with the query "github" entered into the search bar. Below the search bar, there are several navigation tabs: TODO, BÚSQUEDA (highlighted), VÍDEOS, IMÁGENES, MAPAS, COPILOT, and MÁS. The main content area displays a Copilot Search result for GitHub. The result includes the GitHub logo, the URL "https://github.com", and the headline "GitHub · Change is constant. GitHub keeps you ahead." A brief description follows: "Whether you're scaling your development process or just learning how to code, GitHub is where you belong. Join the world's most widely adopted developer platform to build the ...". Below this, there are four featured sections: "Login", "GitHub Copilot Write better co...", "GitHub Models New Manage a...", and "Actions Automate any workflow". Each section has a brief description and a "Read more" link indicated by a right-pointing arrow. At the bottom of the search results, there are three additional cards: "Top GitHub features you should know!", "Top GitHub features for developers", and "Exploring C".

Bing

github

TODO BÚSQUEDA VÍDEOS IMÁGENES MAPAS COPILOT MÁS

Copilot Search

Github
https://github.com

GitHub · Change is constant. GitHub keeps you ahead.

Whether you're scaling your development process or just learning how to code, GitHub is where you belong. Join the world's most widely adopted developer platform to build the ...

Login
GitHub is where people build software.
More than 150 million people use ...

GitHub Copilot Write better co...
GitHub Copilot works alongside you directly in your editor, suggesting ...

GitHub Models New Manage a...
GitHub Models brings top-tier AI models into your workflow—securely and at ...

GitHub Advanced Security Find...
GitHub Advanced Security (GHAS) encompasses GitHub's application ...

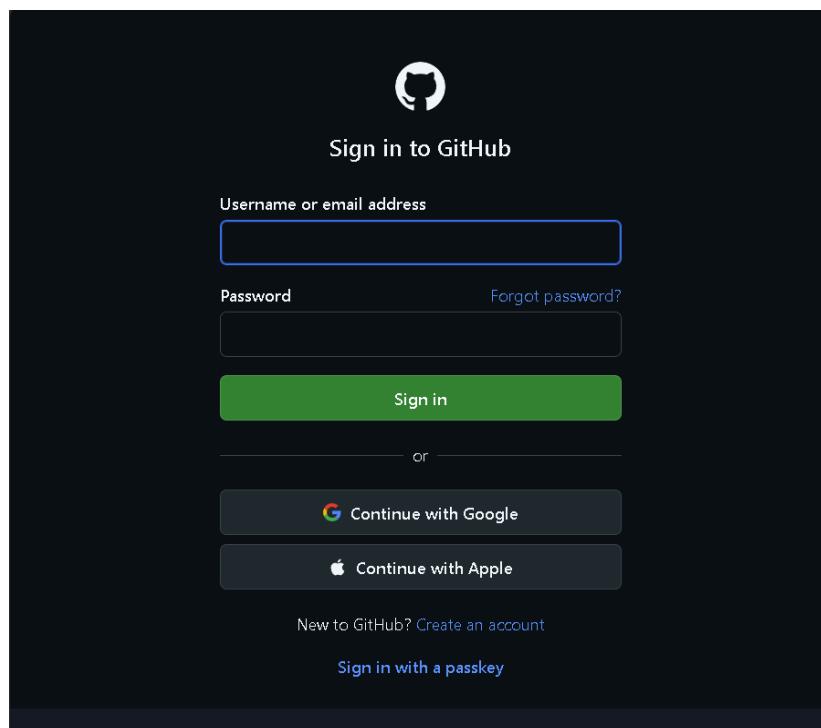
Actions Automate any workflow
GitHub Actions Automate your workflow from idea to production GitHub Actions ...

Codespaces Instant dev enviro...
GitHub Codespaces gets you up and coding faster with fully configured, ...

Top GitHub features you should know! >

Top GitHub features for developers >

Exploring C >



15. Una vez dentro, vamos a la opción “Create repository”

The image shows the GitHub home page. At the top left is the "Home" button. On the right, there is a "Try the new experience" link. Below the header is a search bar with the placeholder "Ask anything" and a "Task" button with a "New" badge. There is also a "Create issue" and a "Spark" button. Under the search bar, there is a section titled "Top Repositories". Below that is a "Create your first project" section with the text "Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it." A green "Create repository" button is prominently displayed, and it is circled with a red oval. Next to it is a "Import repository" link. At the bottom of the page, there is a "Getting started" section with a progress bar showing "0/3 complete" and three dots. Inside this section is a radio button next to the text "Create your first code project" and a descriptive paragraph about creating a project on GitHub. A green "Create project" button is at the bottom of this section.

16. Llenamos los campos y creamos el repositorio, donde subiremos el Proyecto.

Required fields are marked with an asterisk (*).

1 General

Owner * Repository name *

ErickGamarra / proyecto_bd
✓ proyecto_bd is available.

Great repository names are short and memorable. How about friendly-disco?

Description

Proyecto desarrollado con el fin de emplear tablas desarrolladas en Python con OracleSQL
88 / 350 characters

2 Configuration

Choose visibility * Public

Choose who can see and commit to this repository

Add README Off

READMEs can be used as longer descriptions. [About READMEs](#)

Add .gitignore

.gitignore tells git which files not to track. [About ignoring files](#)

No .gitignore

Add license

Licenses explain how others can use your code. [About licenses](#)

No license

Create repository

17. Una vez dentro del repositorio creado, le damos a la opción ‘..or upload an existing file..’, y subimos el archivo raíz ya creado. Luego le damos a Commit changes y se subirá.

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

projecto_bd (Public)

Set up GitHub Copilot Use GitHub's AI pair programmer to autocomplete suggestions as you code. Get started with GitHub Copilot

Add collaborators to this repository Search for people using their GitHub username or email address. Invite collaborators

Quick setup — if you've done this kind of thing before

Setup in Desktop or HTTPS SSH https://github.com/ErickGamarra/projecto_bd.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

git clone "https://github.com/ErickGamarra/projecto_bd.git"

ErickGamarra / proyecto_bd

Code Issues Pull requests Actions Projects Wiki ...

proyecto_bd /

Drag files here to add them to your repository
Or choose your files

Commit changes

Add files via upload

Add an optional extended description...

Commit changes Cancel

This screenshot shows a GitHub repository interface. At the top, there's a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Wiki, and more. Below the navigation, the repository name 'proyecto_bd' is shown. A large central area has a placeholder for dragging files or choosing them. Below this, a modal window titled 'Commit changes' is open, containing fields for adding files via upload and an optional extended description, along with 'Commit changes' and 'Cancel' buttons.

main ▾

Go to file + <> Code ▾

ErickGamarra Add files via upload ... 1483319 · now ⏱

Proyecto_Oracle_Conexion_Pyt... Add files via upload now

README

This screenshot shows the main page of the 'proyecto_bd' repository. It features a dropdown menu for the 'main' branch, search and filter icons, and a 'Go to file' button. Below these are two recent commits: one from 'ErickGamarra' adding files via upload, and another from 'Proyecto_Oracle_Conexion_Pyt...' also adding files via upload. At the bottom, there's a section for the 'README' file.

