



Administración de bases de datos

D07

M-J 7-9 AM



## Práctica 14: IDE's para bases de datos y PostgreSQL

Centro Universitario de Ciencias Exactas e Ingenierías

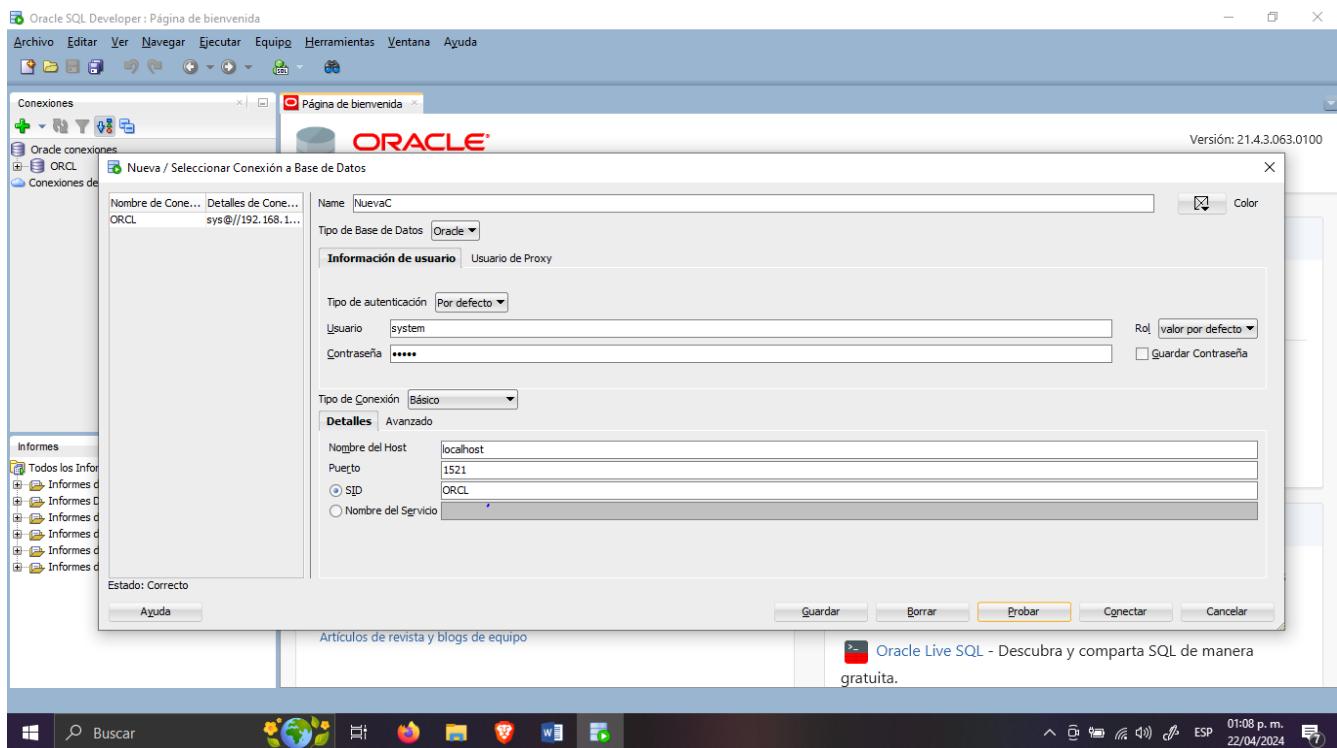
Universidad de Guadalajara

2024<sup>a</sup>

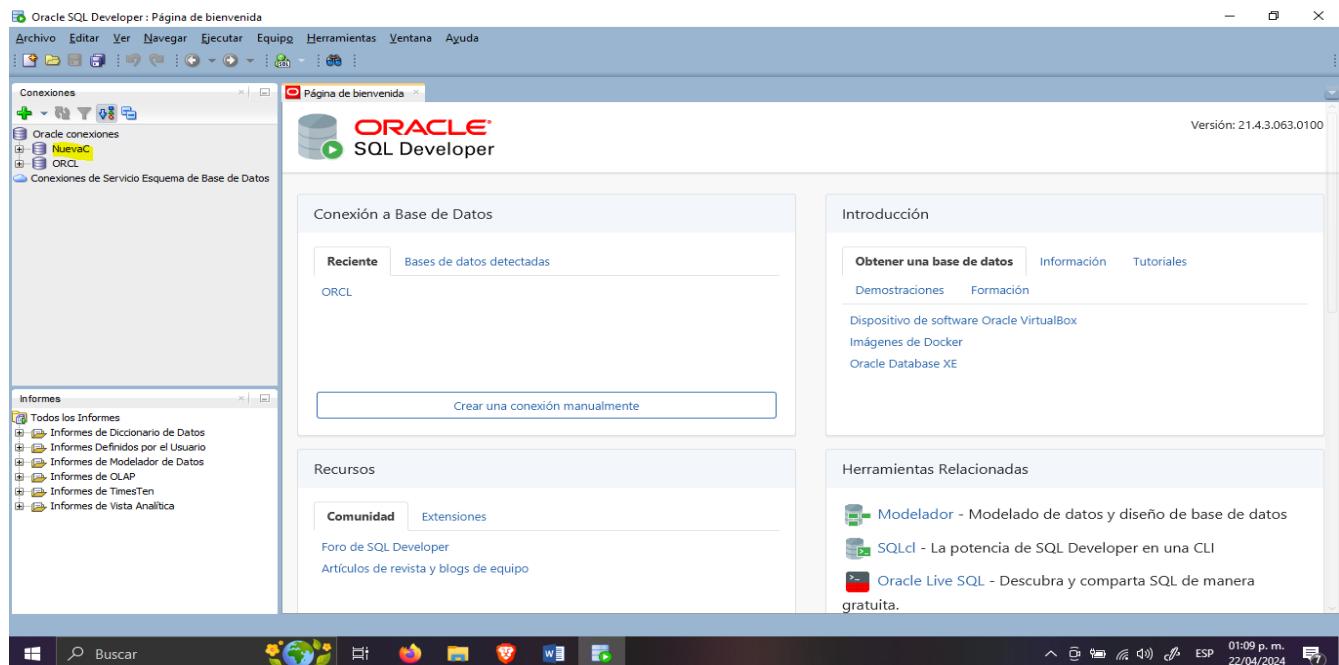
## Con SQL Developer

- Crear una conexión a la base de datos local
- Generar una ingeniería inversa del modelo de datos de la práctica 2 (creación de usuarios y tablas) si ya no tienes las tablas, vuélvelas a hacer con integridad referencial.

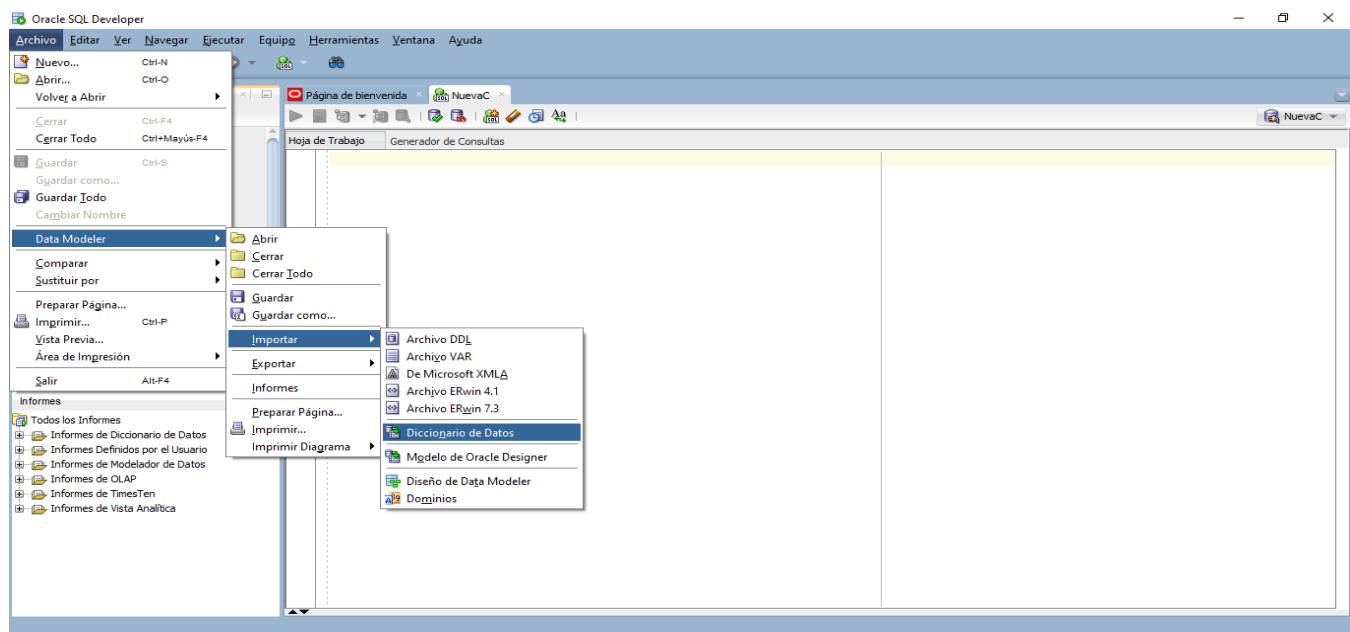
En SQL Developer creamos una nueva conexión con los siguientes parámetros para conectarnos a la base de datos local. Con el usuario SYSTEM y el SID de ORCL.



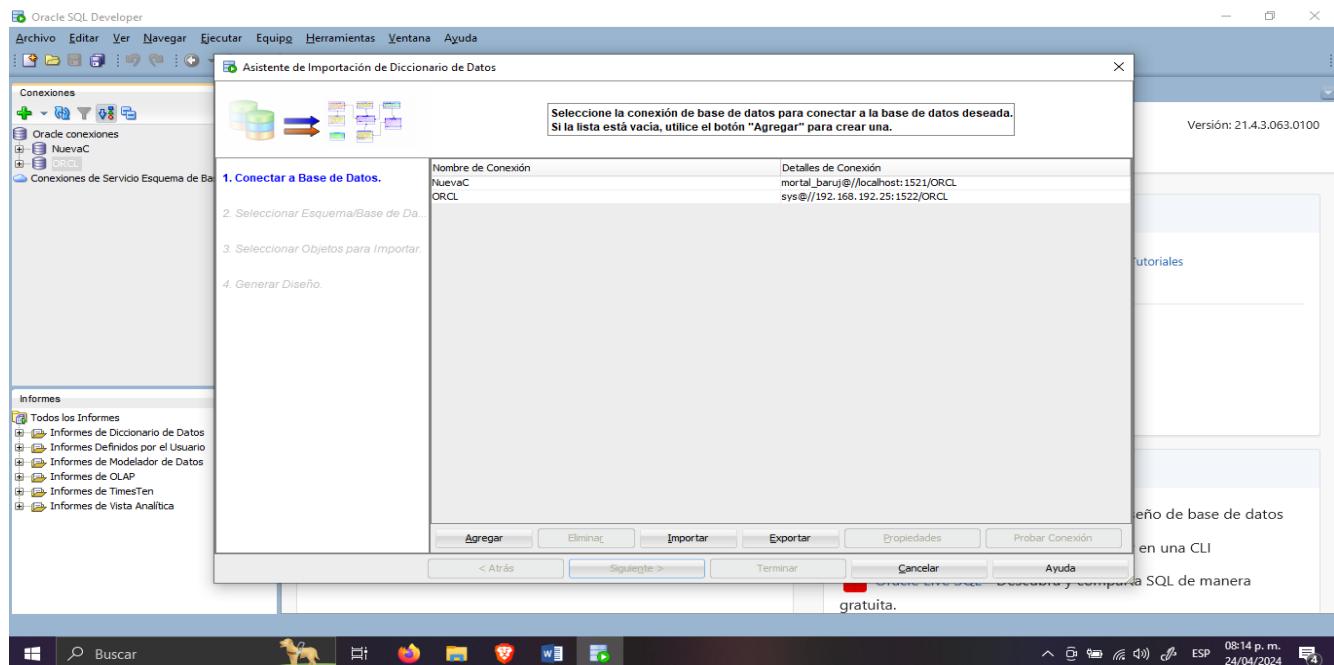
Aquí se ve la creación de la nueva conexión.



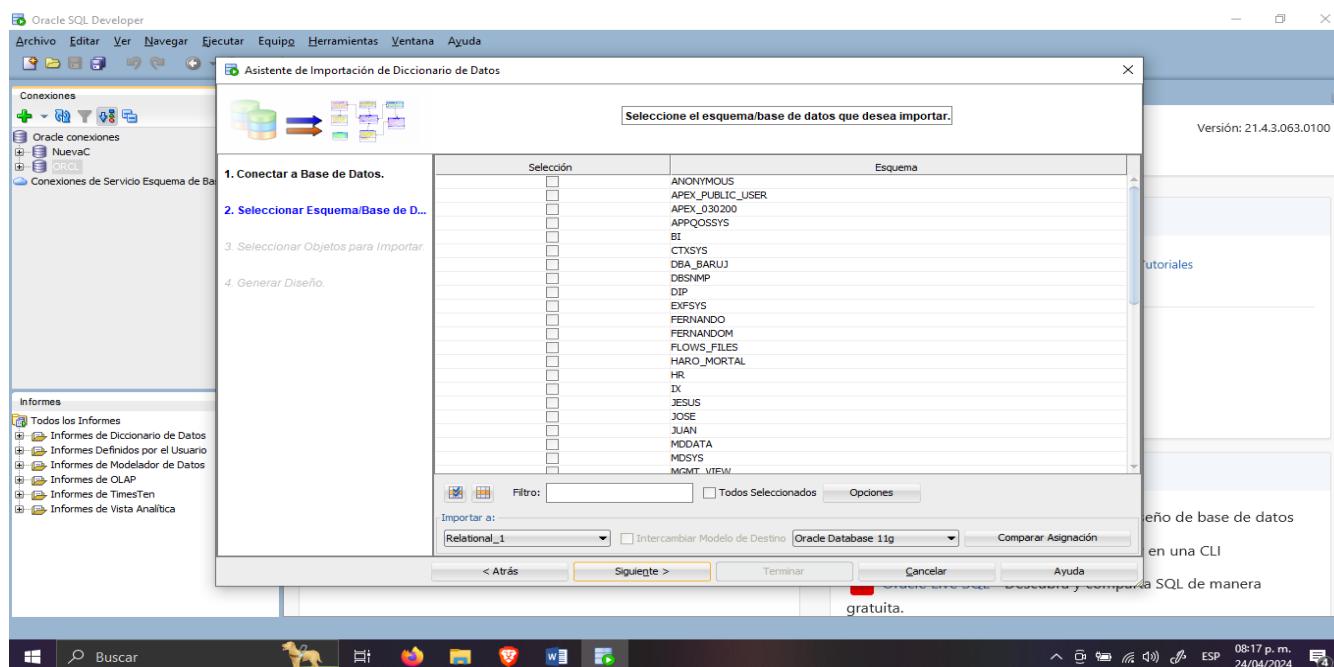
A continuación, procedemos a hacer ingeniería inversa del modelo de datos de la práctica anterior, para ello nos vamos Archivo->Data Modeler->Importar->Diccionario de Datos



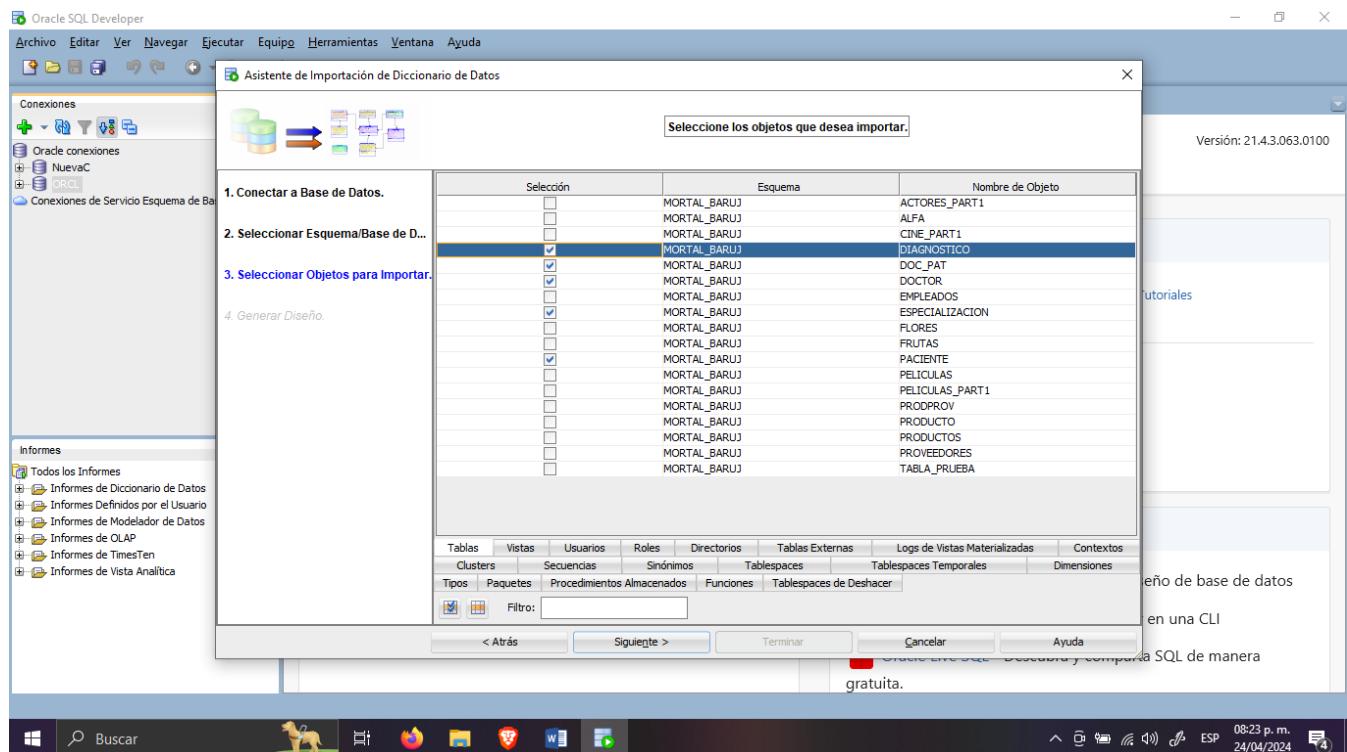
Eso nos abrirá esta pestaña para elegir entre las conexiones creadas.



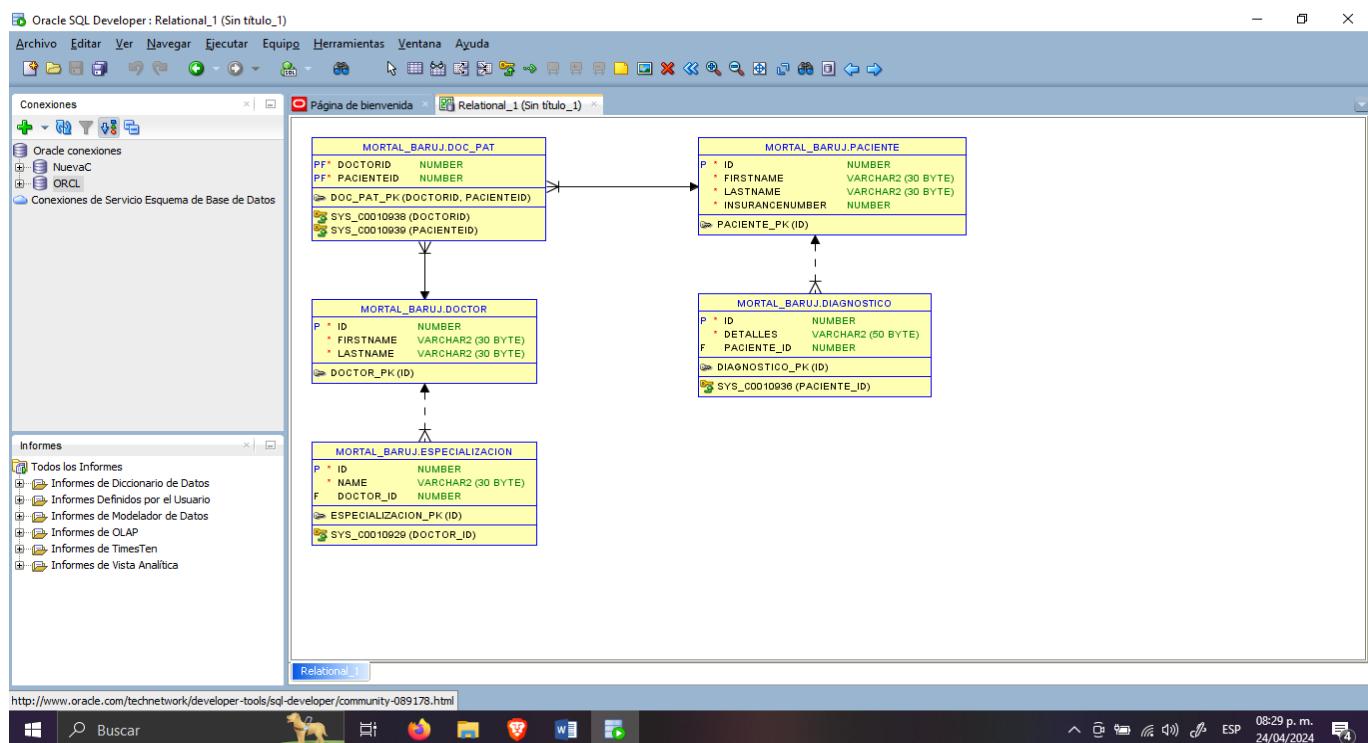
Se elige ORCL, que nos manda a esta pestaña para elegir los esquemas a importar.



Se elige el esquema del usuario MORTAL\_BARUJ, de ahí se nos manda a otra pestaña que muestra las tablas de ese esquema, las cuales vamos a elegir y picarle a siguiente.



Este fue el diagrama generado con las 5 tablas seleccionadas de la práctica 2.

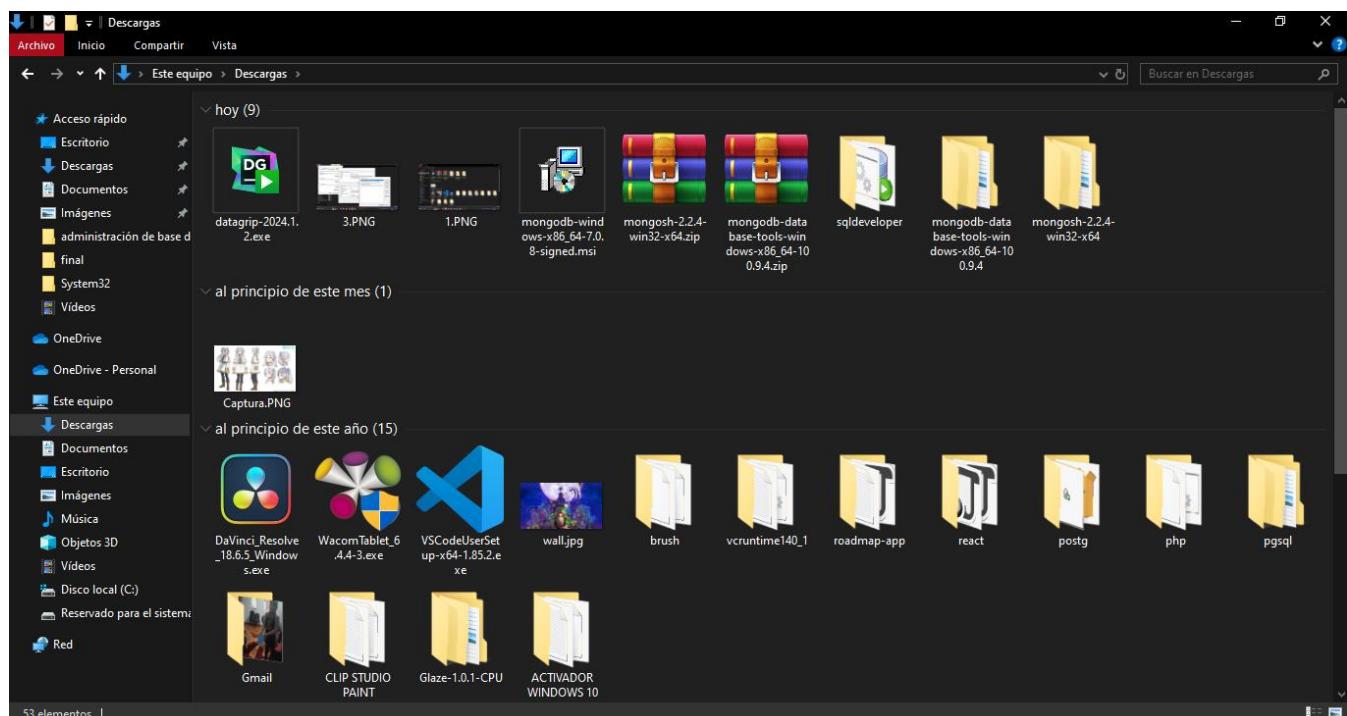


## Con DataGrip/DBeaver

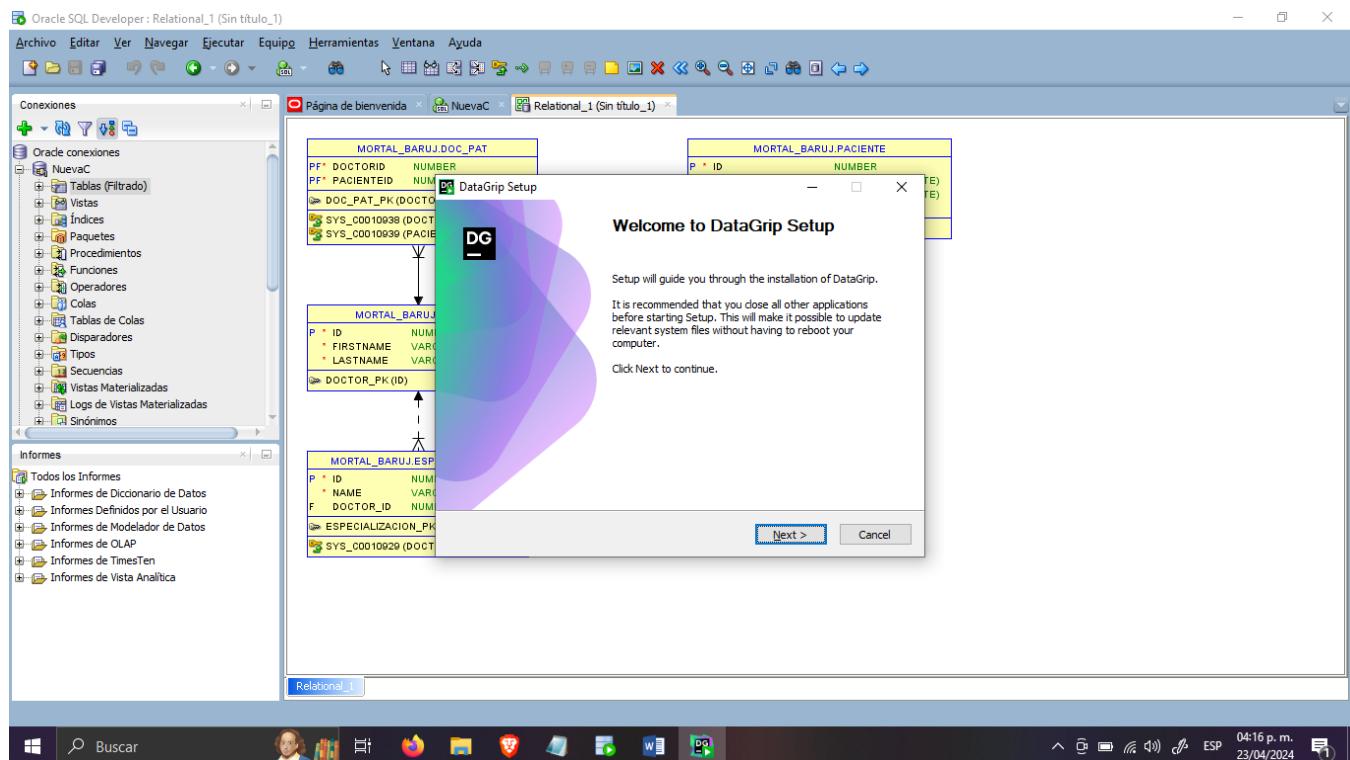
- Crear una conexión a la base de datos local
- Mediante la opción de "Copy tables to..." copiar las tablas de la práctica 2 (creación de usuarios y tablas) a un nuevo usuario
- Mediante la herramienta de "Compare structure" comparar ambos usuarios mortales para verificar el copiado de las tablas

Nota: cuidar la integridad referencial al momento de establecer el orden en el que se copian las tablas

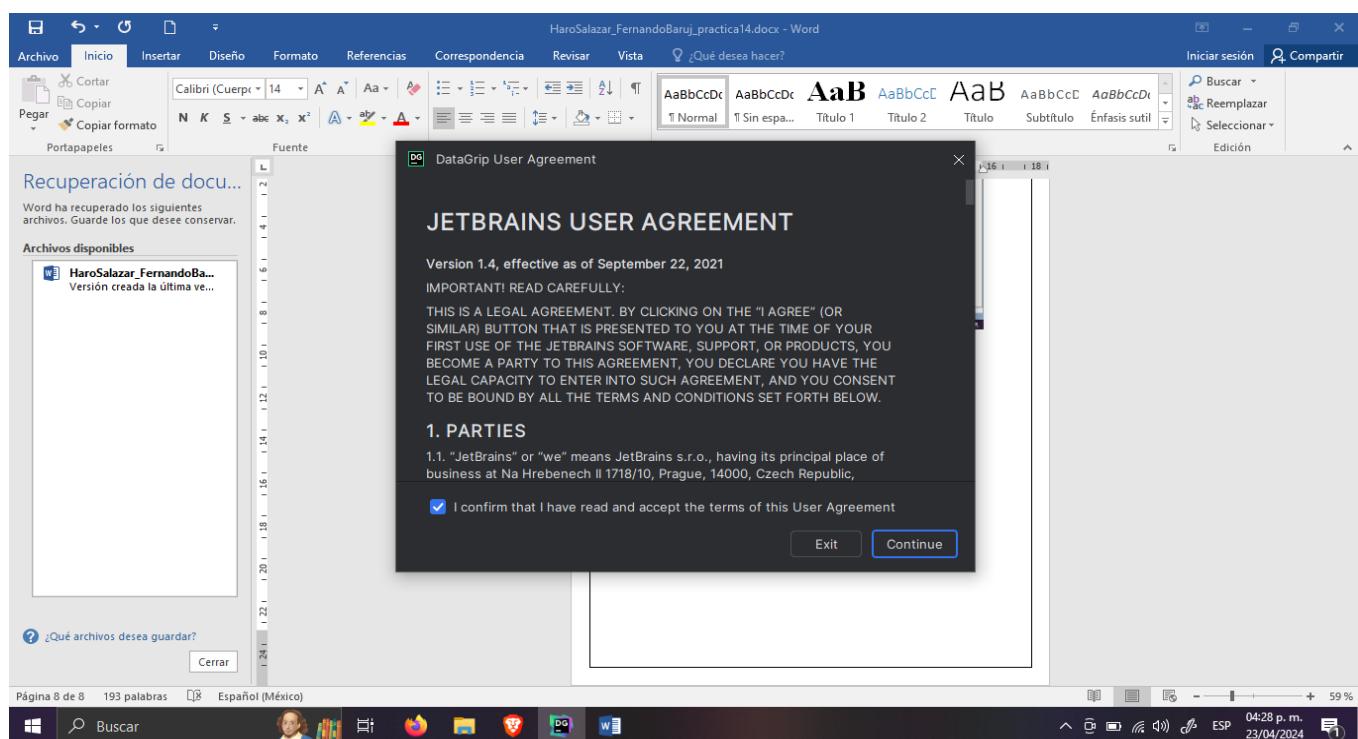
Descargamos y corremos Datagrip.



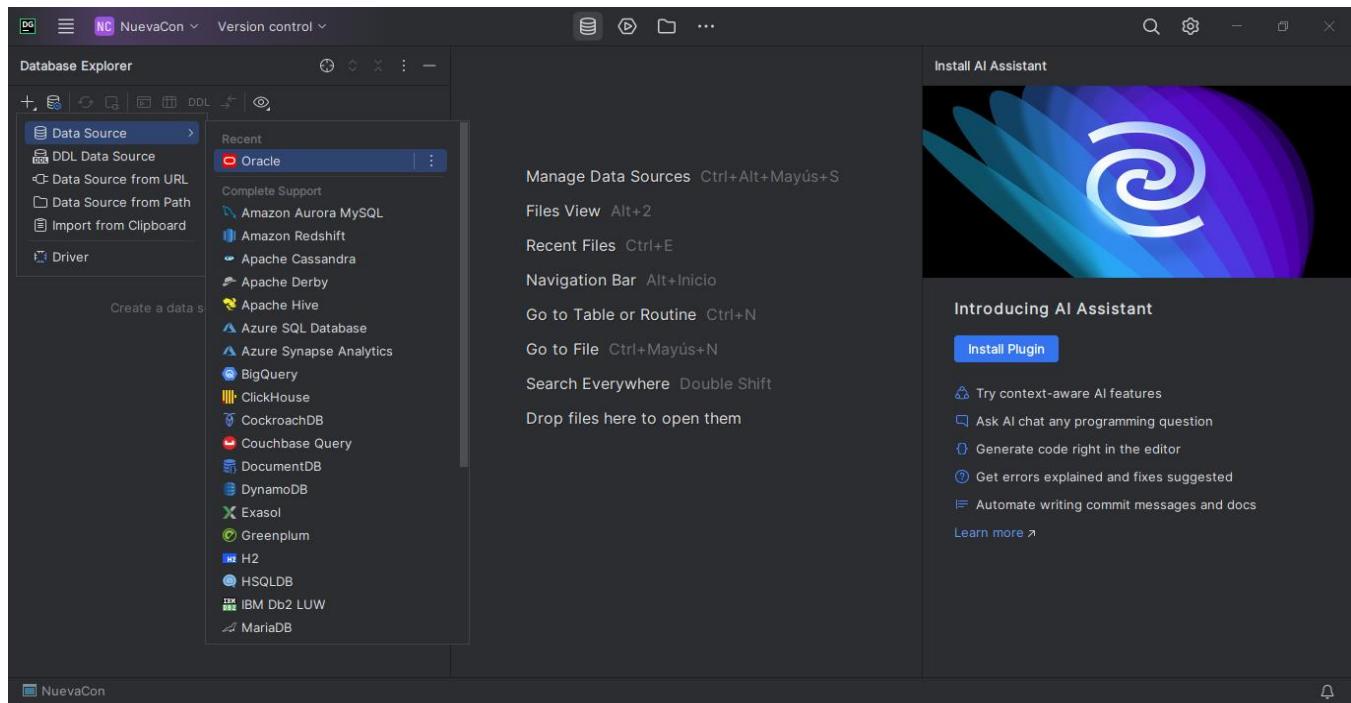
Se inicia la instalación de Datagrip.



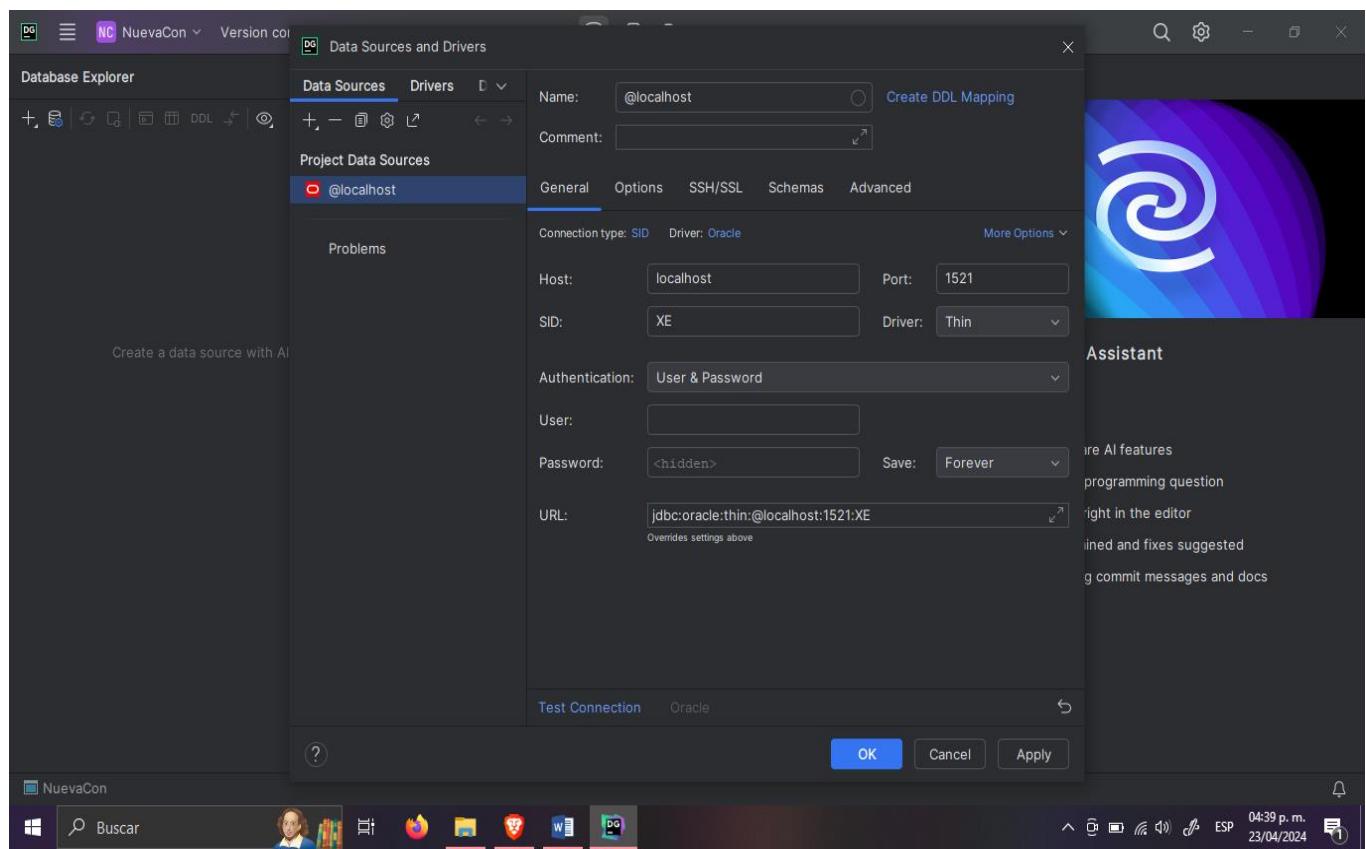
Aceptamos los términos y políticas.



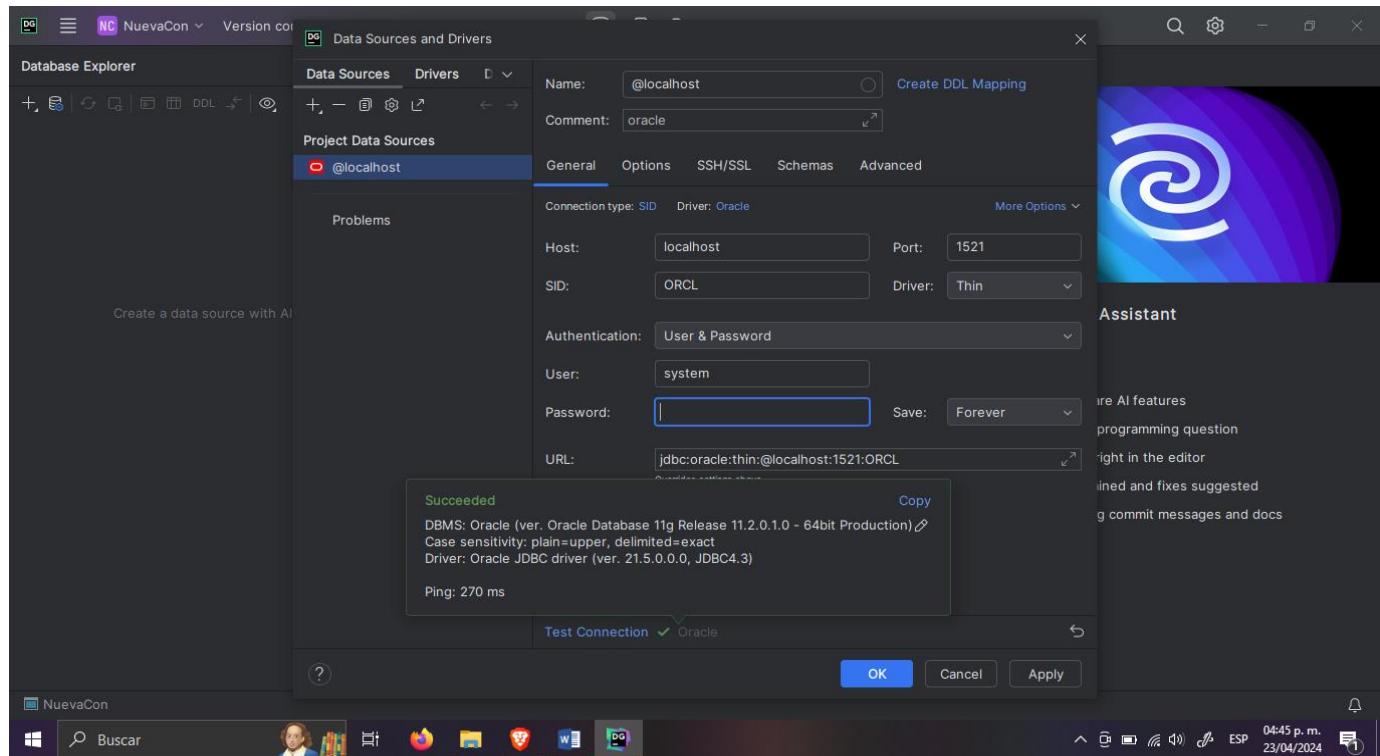
Abrimos Datagrip, creamos una nueva conexión en data source-> Oracle, para conectarnos a Oracle.



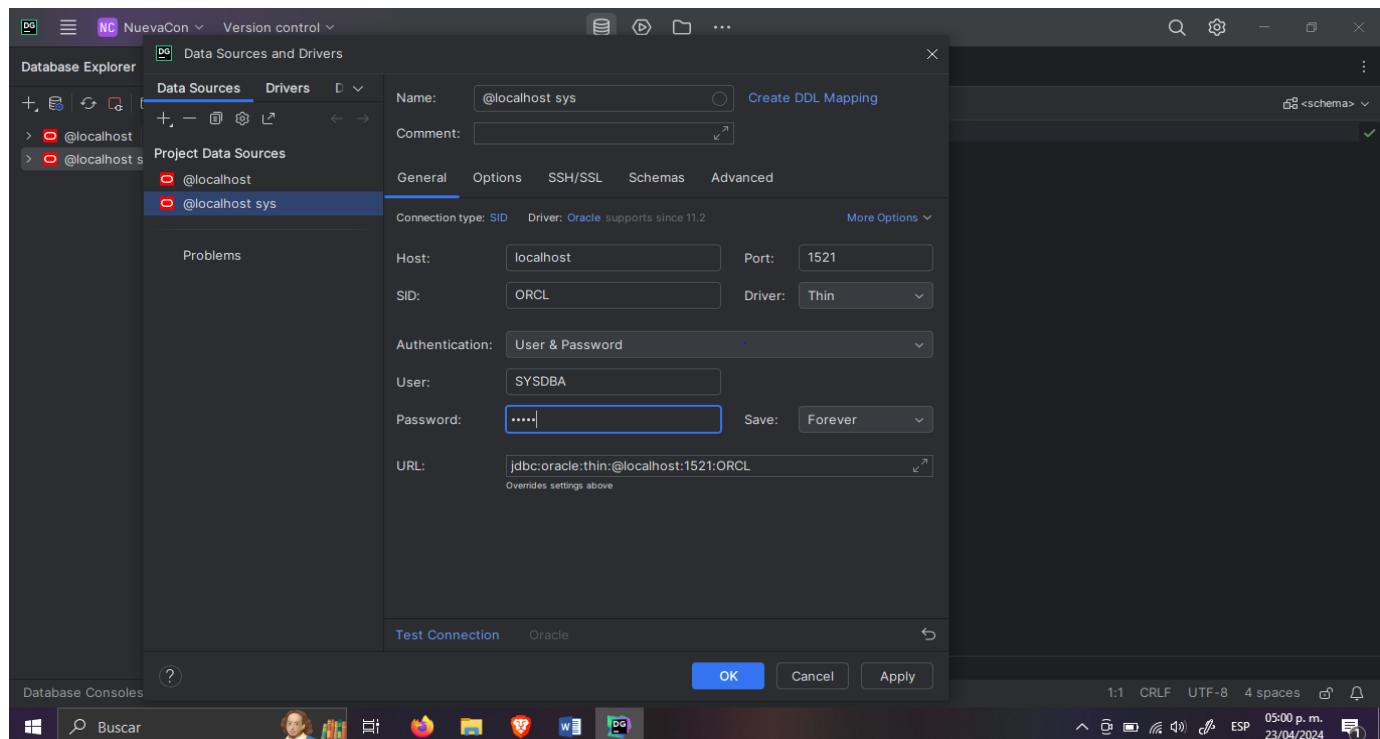
Esto dará una pestaña similar a sql developer, donde nos conectaremos al localhost.



Con el SID de ORCL y se modifica también el usuario.



Además de crear otra conexión para el usuario DBA.



Creamos otro usuario con contraseña y le damos el privilegio de crear sesión.

The screenshot shows the Oracle SQL Developer interface. In the Database Explorer, there are two connections: '@localhost' and '@localhost sys'. The main console window contains the following SQL code:

```

CREATE USER mor_bar2 IDENTIFIED BY "1234";
grant create session to mor_bar2;

```

A red error message at the bottom of the console window states: '[42000][1920] ORA-01920: conflicto entre el nombre de usuario 'MOR\_BAR2' y otro nombre de usuario o rol Position: 12'. Below the console, the Services pane shows a transaction history with several log entries related to the failed user creation.

Con el usuario Mortal que se creó en la practica 2 y el cual tiene las tablas, ese debe hacer la acción de copy tables to... con tabla->import->copytables to...

The screenshot shows the Oracle SQL Developer interface. In the Database Explorer, the 'PACIENTE' table is selected. A context menu is open over this table, with the 'Import/Export' option highlighted. The main console window shows the following command:

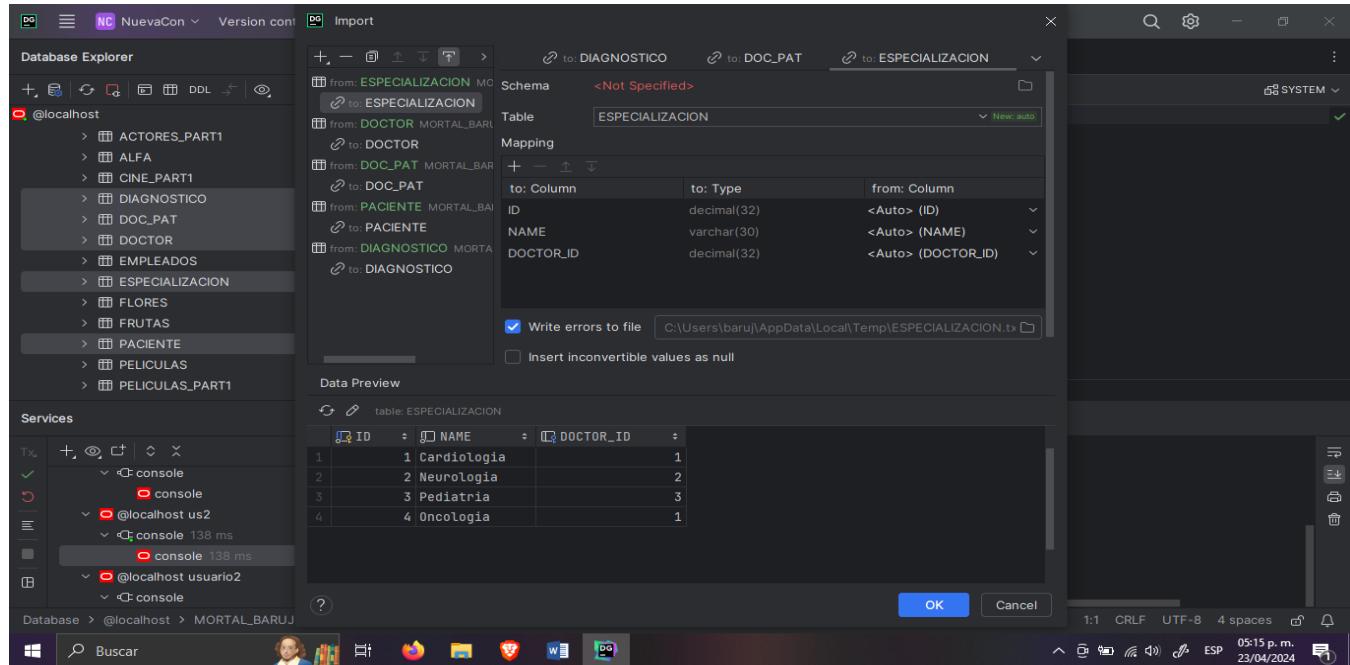
```

ALTER SESSION SET CURRENT_SCHEMA = SYSTEM

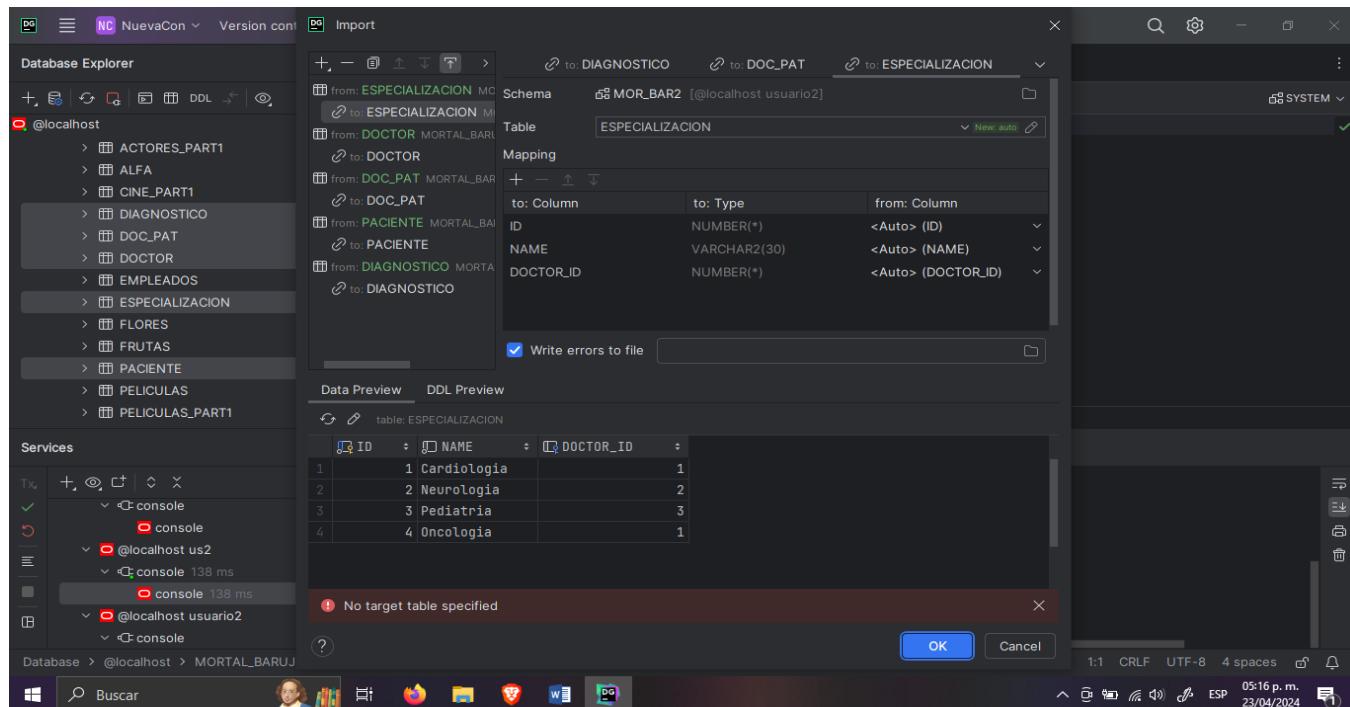
```

The Services pane shows a transaction history with log entries related to schema changes and grants.

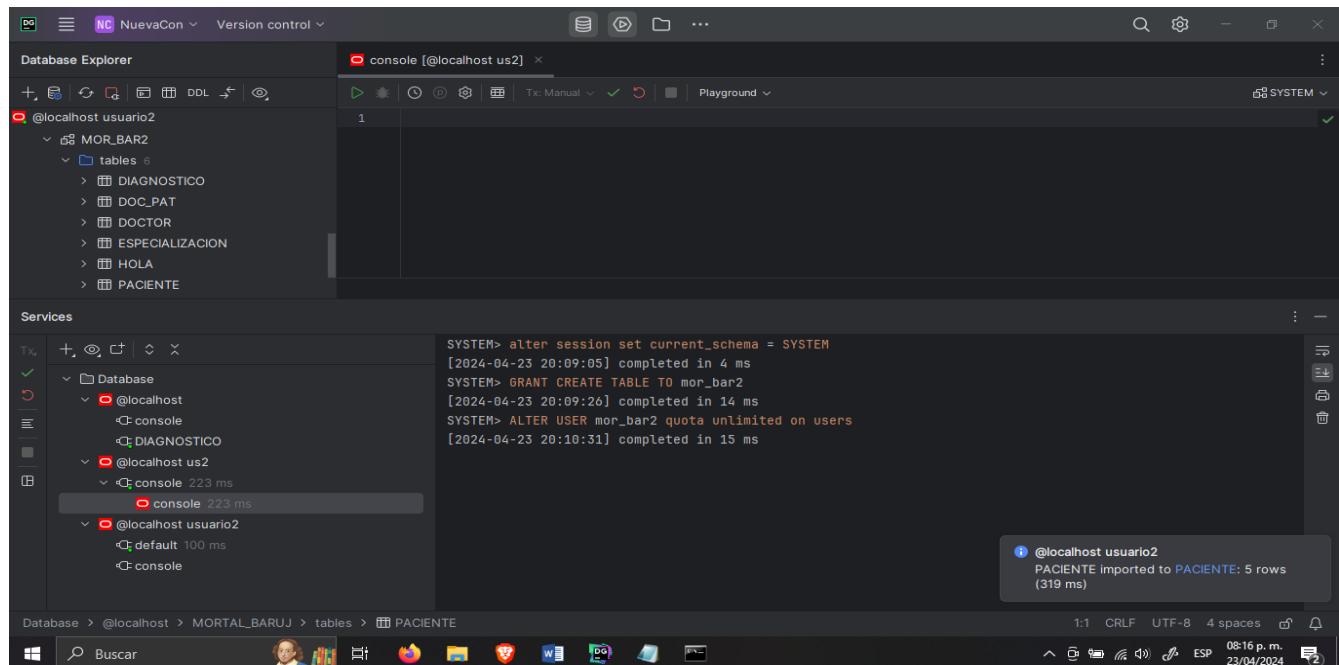
Esta pestaña se abrirá para poder elegir a que esquema se desean mandar las tablas.



Después se elige el usuario a donde se han de copiar las tablas.

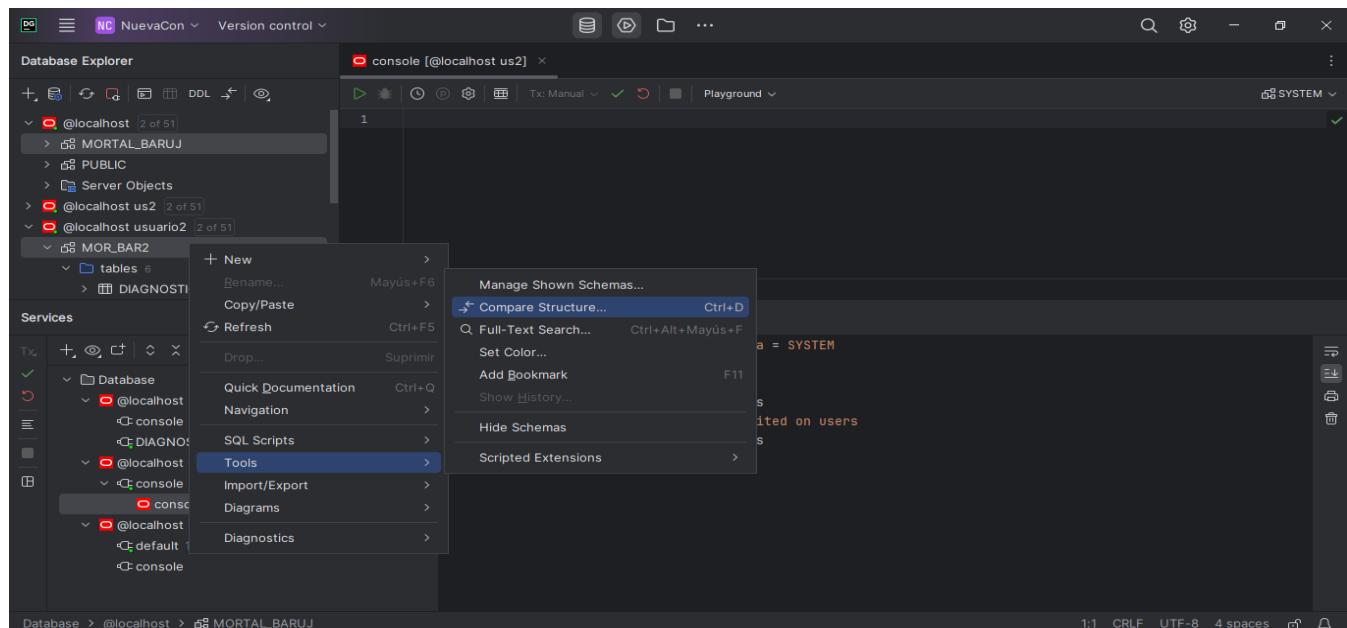


Podemos ver en la parte de conexiones, donde se encuentra el usuario mor\_bar2, este ya tiene las tablas que se copiaron.



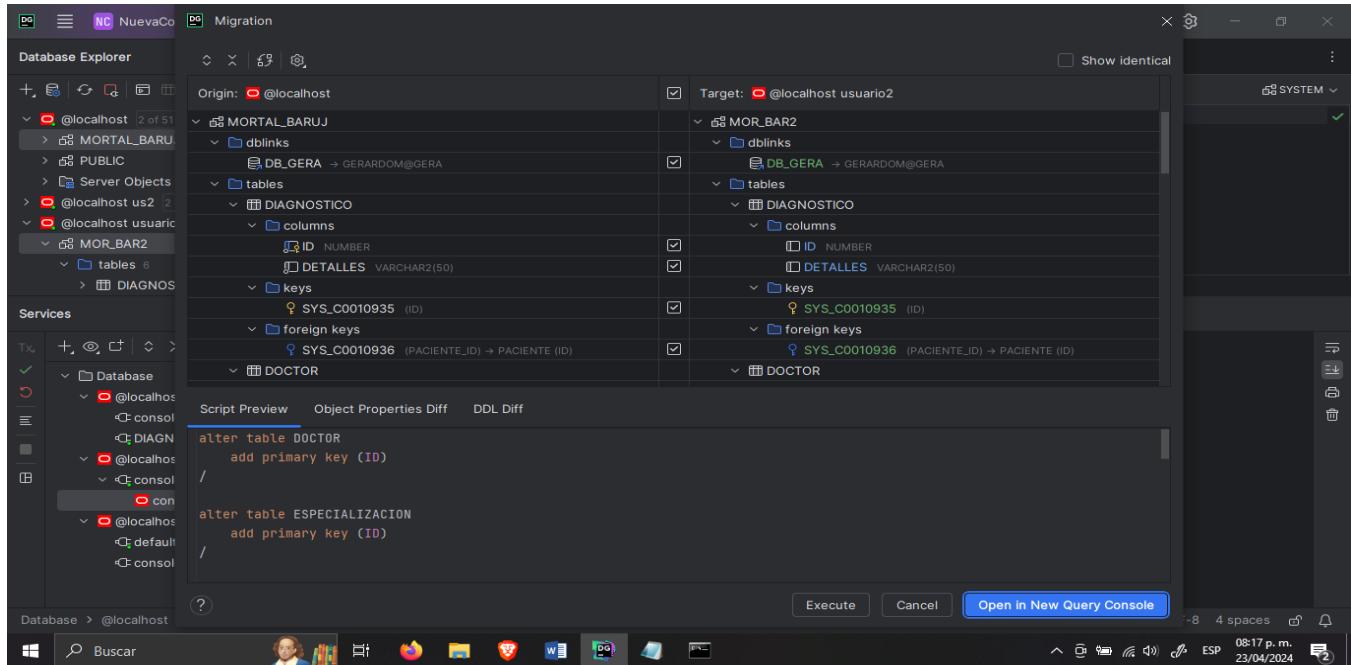
Se seleccionan los dos usuarios con click derecho->tools->compare structure...

Para hacer una comparación entre estos dos usuarios, sus tablas y referencias.

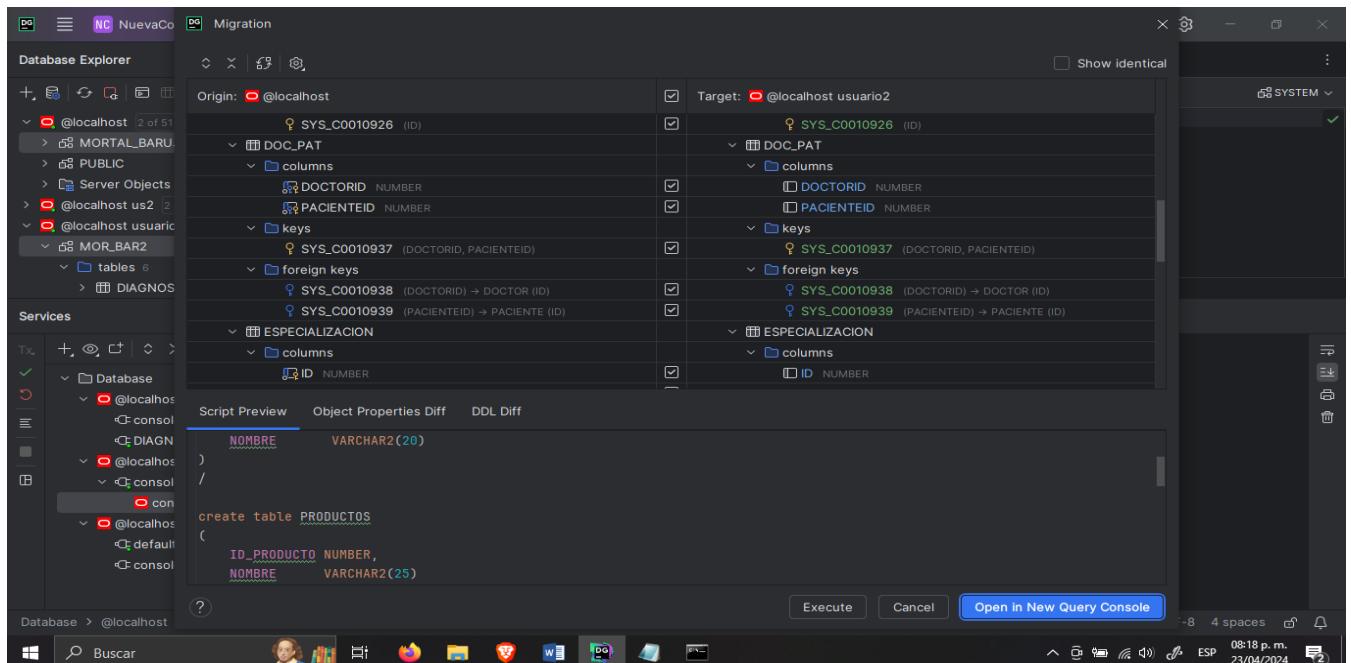


Como podemos ver, se abre esta pestaña que nos permite analizar las comparaciones, con similitudes y diferencias, en estructura y en sus tablas.

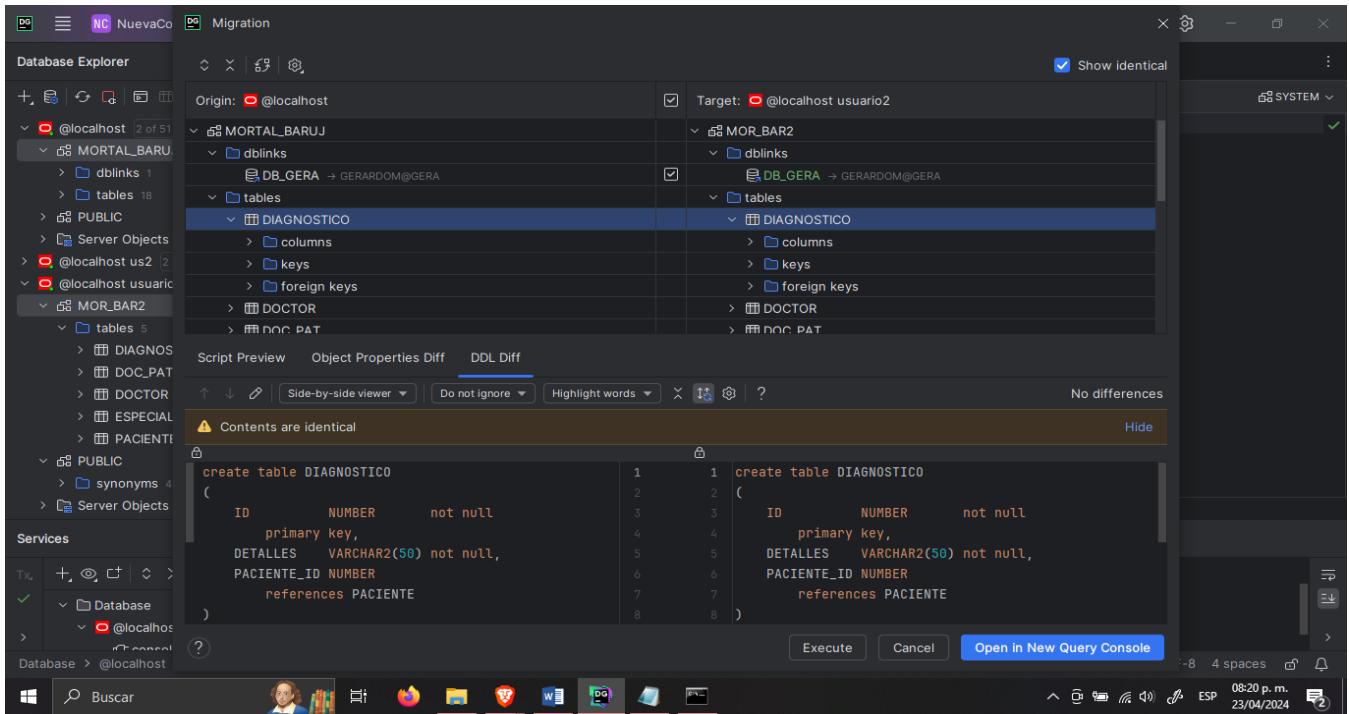
Como se puede observar los elementos de la derecha de color verde son aquellos que son diferentes.



Podemos ver en azul los elementos similares.



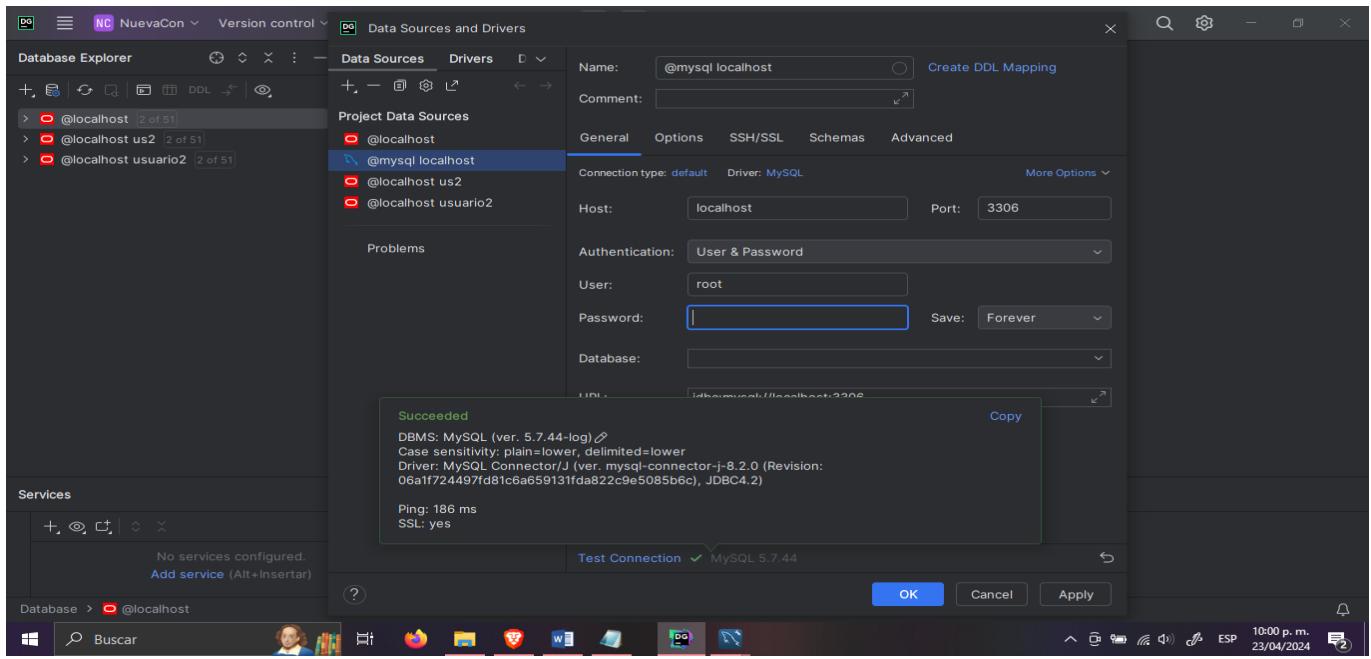
Si le picamos a "show identical" mostrará la comparación en los elementos que se elijan entre los usuarios.



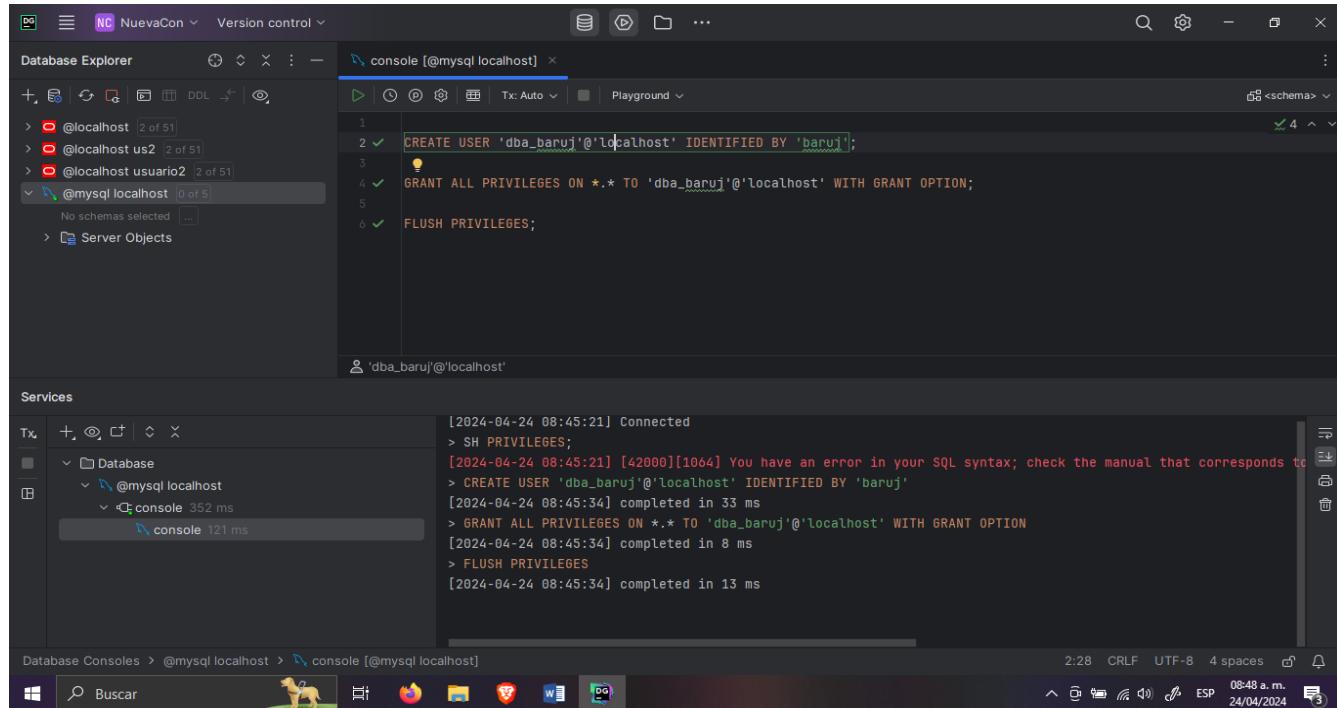
## Con PostgreSQL/MySQL

- Volver a generar la práctica 2 (creación de usuarios y tablas)
- Cuidar la integridad referencial que esté creada
- Queda a su consideración hacerlo mediante consola, PgAdmin o DataGrip

En este caso se utiliza MySQL, el cual ya está instalado, sin embargo se requiere la conexión de manera similar que con Oracle, cambiando el puerto a 3306 y el usuario de root.



Para generar la práctica 2, se creará el usuario dba con todos los privilegios.



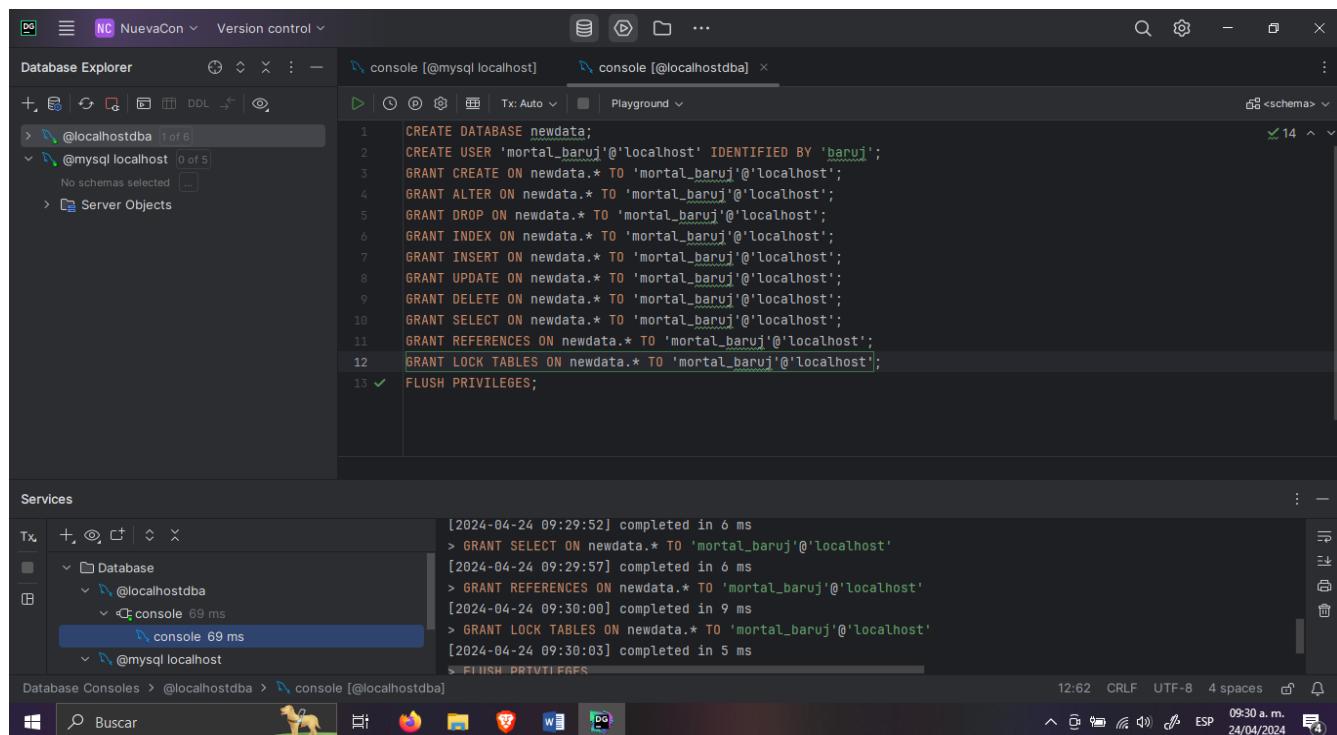
The screenshot shows the MySQL Workbench interface. In the Database Explorer, there are several connections listed under the 'localhost' node. In the main console window, the following SQL commands are run:

```
CREATE USER 'dba_baruj'@'localhost' IDENTIFIED BY 'baruj';
GRANT ALL PRIVILEGES ON *.* TO 'dba_baruj'@'localhost' WITH GRANT OPTION;
FLUSH PRIVILEGES;
```

The Services panel shows a transaction history with the following log entries:

```
[2024-04-24 08:45:21] Connected
> SH PRIVILEGES;
[2024-04-24 08:45:21] [42000][1064] You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'SH PRIVILEGES;' at line 1
> CREATE USER 'dba_baruj'@'localhost' IDENTIFIED BY 'baruj'
[2024-04-24 08:45:34] completed in 33 ms
> GRANT ALL PRIVILEGES ON *.* TO 'dba_baruj'@'localhost' WITH GRANT OPTION
[2024-04-24 08:45:34] completed in 8 ms
> FLUSH PRIVILEGES
[2024-04-24 08:45:34] completed in 13 ms
```

En la conexión del usuario dba, se crea el usuario mortal y la base de datos, además de darle permisos a este usuario mortal sobre esa base de datos.



The screenshot shows the MySQL Workbench interface. In the Database Explorer, there are connections listed under the 'localhost' node. In the main console window, the following SQL commands are run:

```
CREATE DATABASE newdata;
CREATE USER 'mortal_baruj'@'localhost' IDENTIFIED BY 'baruj';
GRANT CREATE ON newdata.* TO 'mortal_baruj'@'localhost';
GRANT ALTER ON newdata.* TO 'mortal_baruj'@'localhost';
GRANT DROP ON newdata.* TO 'mortal_baruj'@'localhost';
GRANT INDEX ON newdata.* TO 'mortal_baruj'@'localhost';
GRANT INSERT ON newdata.* TO 'mortal_baruj'@'localhost';
GRANT UPDATE ON newdata.* TO 'mortal_baruj'@'localhost';
GRANT DELETE ON newdata.* TO 'mortal_baruj'@'localhost';
GRANT SELECT ON newdata.* TO 'mortal_baruj'@'localhost';
GRANT REFERENCES ON newdata.* TO 'mortal_baruj'@'localhost';
GRANT LOCK TABLES ON newdata.* TO 'mortal_baruj'@'localhost';
FLUSH PRIVILEGES;
```

The Services panel shows a transaction history with the following log entries:

```
[2024-04-24 09:29:52] completed in 6 ms
> GRANT SELECT ON newdata.* TO 'mortal_baruj'@'localhost'
[2024-04-24 09:29:57] completed in 6 ms
> GRANT REFERENCES ON newdata.* TO 'mortal_baruj'@'localhost'
[2024-04-24 09:30:00] completed in 9 ms
> GRANT LOCK TABLES ON newdata.* TO 'mortal_baruj'@'localhost'
[2024-04-24 09:30:03] completed in 5 ms
> FINISH PRTVLFARS
```

Dentro de la conexión de usuario mortal. Se crea la tabla de Doctor.

The screenshot shows the MySQL Workbench interface. On the left, the Database Explorer pane displays databases like @localhostdba, @mortallocalhost, and @mysql localhost. The Services pane shows various MySQL connections. The main workspace contains a code editor with the following SQL script:

```
1 ✓ USE newdata;
2 ✓ CREATE TABLE Doctor (
3     ID INT PRIMARY KEY,
4     FirstName VARCHAR(30) NOT NULL,
5     LastName VARCHAR(30) NOT NULL
6 );
```

The status bar at the bottom indicates the command was completed in 35 ms.

Se crean las demás tablas de la práctica 2, adaptándolas a Mysql.

**CREATE TABLE Especializacion (**

**ID INT PRIMARY KEY,**

**Name VARCHAR(30) NOT NULL,**

**Doctor\_id INT REFERENCES Doctor(ID)**

**);**

**CREATE TABLE Paciente (**

**ID INT PRIMARY KEY,**

**FirstName VARCHAR(30) NOT NULL,**

**LastName VARCHAR(30) NOT NULL,**

**InsuranceNumber INT NOT NULL**

**);**

```
CREATE TABLE Diagnostico (
    ID INT PRIMARY KEY,
    Detalles VARCHAR(50) NOT NULL,
    Paciente_id INT REFERENCES Paciente(ID)
);


```

```
CREATE TABLE Doc_pat(
    DoctorID INT REFERENCES Doctor(ID),
    PacienteID INT REFERENCES Paciente(ID),
    PRIMARY KEY (DoctorID, PacienteID)
);


```

The screenshot shows the MySQL Workbench interface with the following details:

- Database Explorer:** Shows the database structure with two servers: @localhost and @mortallocalhost. Under @mortallocalhost, there is a schema named newdata which contains two tables: Especializacion and Paciente.
- Console:** Three consoles are open:
  - console [@mysql localhost]: Shows the creation of the Especializacion table.
  - console [@localhost]: Shows the creation of the Paciente table.
  - console [@mortallocalhost]: Shows the creation of the Doc\_pat table.
- Services:** A transaction history is displayed, showing the execution times for each query.
- Bottom Bar:** Includes icons for search, file operations, and system status.

```
CREATE TABLE Especializacion (
    ID INT PRIMARY KEY,
    Name VARCHAR(30) NOT NULL,
    Doctor_id INT REFERENCES Doctor(ID)
);

CREATE TABLE Paciente (
    ID INT PRIMARY KEY,
    FirstName VARCHAR(30) NOT NULL,
    LastName VARCHAR(30) NOT NULL,
    TelephoneNumber INT NOT NULL
);

CREATE TABLE Doc_pat (
    DoctorID INT REFERENCES Doctor(ID),
    PacienteID INT REFERENCES Paciente(ID),
    PRIMARY KEY (DoctorID, PacienteID)
);
```

Se insertan datos a la tabla de Doctor.

```
INSERT INTO Doctor (ID, FirstName, LastName) VALUES  
(1, 'John', 'Doe'),  
(2, 'Vane', 'Smith'),  
(3, 'Robert', 'Johnson');
```

The screenshot shows the MySQL Workbench application. In the Database Explorer, there are two databases: @localhostdba (1 of 6) and @mortallocalhost (1 of 2). Under @mortallocalhost, there is a newdata schema containing tables: diagnostico, doc\_pat, and doctor. The doctor table is selected. In the main console window, the following SQL code is shown:

```
INSERT INTO Doctor (ID, FirstName, LastName) VALUES  
(1, 'John', 'Doe'),  
(2, 'Vane', 'Smith'),  
(3, 'Robert', 'Johnson');
```

Below the code, the output of the query is displayed:

```
PRIMARY KEY (DoctorID, PacienteID)  
)  
[2024-04-24 09:43:15] completed in 28 ms  
newdata> INSERT INTO Doctor (ID, FirstName, LastName) VALUES  
(1, 'John', 'Doe'),  
(2, 'Vane', 'Smith'),  
(3, 'Robert', 'Johnson')  
[2024-04-24 09:52:31] 3 rows affected in 18 ms
```

The Services panel shows several active connections, including @localhost, @mortallocalhost, and @mysql localhost. The bottom status bar indicates the time as 09:52 a.m. on 24/04/2024.

Se insertan datos a la tabla Diagnostico.

```
INSERT INTO Diagnostico (ID, Detalles, Paciente_id) VALUES  
(1, 'Fiebre', '1'),  
(2, 'Presion alta', '2'),  
(3, 'Dolor de cabeza', '3'),  
(4, 'Demencia', '4'),  
(5, 'Hipertension', '5'),  
(6, 'Bronquitis', '4'),  
(7, 'Leucemia', '2');
```

The screenshot shows the MySQL Workbench application. In the Database Explorer, there are two databases: @localhostdb and @mortallocalhost. Under @mortallocalhost, there is a newdata schema which contains a table named 'diagnostico'. The table has three columns: ID, Detalles, and Paciente\_id. The console window displays the following SQL query and its execution results:

```
1 ✓ INSERT INTO Diagnostico (ID, Detalles, Paciente_id) VALUES
2   (1, 'Fiebre', '1'),
3   (2, 'Presion alta', '2'),
4   (3, 'Dolor de cabeza', '3'),
5   (4, 'Demencia', '4'),
6   (5, 'Hipertension', '5'),
7   (6, 'Bronquitis', '4'),
8   (7, 'Leucemia', '2');
```

The output window shows the results of the insertion:

```
newdata.dotor
(2, 'Presion alta', '2'),
(3, 'Dolor de cabeza', '3'),
(4, 'Demencia', '4'),
(5, 'Hipertension', '5'),
(6, 'Bronquitis', '4'),
(7, 'Leucemia', '2')
[2024-04-24 18:54:06] 7 rows affected in 20 ms
```

Se insertan datos a la tabla de especialización.

INSERT INTO Especializacion (ID, Name, Doctor\_id) VALUES

(1, 'Cardiologia', '1'),

(2, 'Neurologia', '2'),

(3, 'Pediatrica', '3'),

(4, 'Oncologia', '1');

The screenshot shows the MySQL Workbench application. In the Database Explorer, there are two databases: @localhostdb and @mortallocalhost. Under @mortallocalhost, there is a newdata schema which contains a table named 'especializacion'. The table has three columns: ID, Name, and Doctor\_id. The console window displays the following SQL query and its execution results:

```
1 ✓ INSERT INTO Especializacion (ID, Name, Doctor_id) VALUES
2   (1, 'Cardiologia', '1'),
3   (2, 'Neurologia', '2'),
4   (3, 'Pediatrica', '3'),
5   (4, 'Oncologia', '1');
```

The output window shows the results of the insertion:

```
newdata.dotor
[2024-04-24 18:54:06] 7 rows affected in 20 ms
newdata> INSERT INTO Especializacion (ID, Name, Doctor_id) VALUES
(1, 'Cardiologia', '1'),
(2, 'Neurologia', '2'),
(3, 'Pediatrica', '3'),
(4, 'Oncologia', '1')
[2024-04-24 18:57:44] 4 rows affected in 13 ms
```

Se insertan datos a la tabla de paciente.

```
INSERT INTO Paciente (ID, Firstname, Lastname, insurancenumber) VALUES
(1, 'Leon', 'White', '123'),
(2, 'Charlie', 'Williams', '234'),
(3, 'David', 'Jones', '456'),
(4, 'Eva', 'Black', '567'),
(5, 'Fernando', 'Haro', '678');
```

The screenshot shows the DataGrip IDE interface. On the left, the Database Explorer pane displays the schema of the database 'mortallocalhost'. It shows two tables: 'paciente' and 'doctor'. The 'paciente' table has columns 'ID' (int), 'FirstName' (varchar(30)), and 'LastName' (varchar(30)). The 'doctor' table has columns 'columns' (3), 'keys' (1), and 'indexes' (1). In the center, the 'console' tab is active, showing the execution of the provided SQL query. The output window below shows the command and the response: '5 rows affected in 25 ms'. At the bottom, the status bar indicates the date and time: '2024-04-24 19:07:56' and '07:08 p.m.'.

```
1 ✓ INSERT INTO Paciente (ID, Firstname, Lastname, insurancenumber) VALUES
2 (1, 'Leon', 'White', '123'),
3 (2, 'Charlie', 'Williams', '234'),
4 (3, 'David', 'Jones', '456'),
5 (4, 'Eva', 'Black', '567'),
6 (5, 'Fernando', 'Haro', '678');
```

```
newdata> INSERT INTO Paciente (ID, Firstname, Lastname, insurancenumber) VALUES
      (1, 'Leon', 'White', '123'),
      (2, 'Charlie', 'Williams', '234'),
      (3, 'David', 'Jones', '456'),
      (4, 'Eva', 'Black', '567'),
      (5, 'Fernando', 'Haro', '678')
[2024-04-24 19:07:56] 5 rows affected in 25 ms
```

Se insertan los datos a Doc\_pat.

```
INSERT INTO doc_pat (Doctorid, pacienteid) VALUES
(1, 1),
(1, 2),
(2, 3),
(2, 4),
(3, 2);
```

The screenshot shows the MySQL Workbench interface. In the Database Explorer, the schema @mortallocalhost is selected, displaying tables like doc\_pat, doctor, paciente, and especializacion. The Services panel shows a transaction tree with several connections. The main console window displays the following SQL query and its execution results:

```
1 ✓ INSERT INTO doc_pat (Doctorid, pacienteid) VALUES
2   (1, 1),
3   (1, 2),
4   (2, 3),
5   (2, 4),
6   (3, 2);
```

```
newdata> INSERT INTO doc_pat (Doctorid, pacienteid) VALUES
(1, 1),
(1, 2),
(2, 3),
(2, 4),
(3, 2)
[2024-04-24 19:13:48] 5 rows affected in 18 ms
```

The status bar at the bottom indicates the output is 1:1 (89 chars, 5 line breaks), CRLF, UTF-8, 4 spaces, and the current time is 07:13 p.m. on 24/04/2024.

Se hacen alter table para crear las llaves foráneas y enlazarlas con las tablas y que se cuide la integridad de las tablas.

ALTER TABLE Especializacion

ADD CONSTRAINT fk\_Doctor\_id

FOREIGN KEY (Doctor\_id) REFERENCES Doctor(ID);

ALTER TABLE Diagnostico

ADD CONSTRAINT fk\_Paciente\_id

FOREIGN KEY (Paciente\_id) REFERENCES Paciente(ID);

ALTER TABLE Doc\_pat

ADD CONSTRAINT fk\_DoctorID

FOREIGN KEY (DoctorID) REFERENCES Doctor(ID),

ADD CONSTRAINT fk\_PacienteID

FOREIGN KEY (PacienteID) REFERENCES Paciente(ID);

```

ALTER TABLE Especializacion
ADD CONSTRAINT fk_Doctor_id
FOREIGN KEY (Doctor_id) REFERENCES Doctor(ID);

ALTER TABLE Diagnostico
ADD CONSTRAINT fk_Paciente_id
FOREIGN KEY (Paciente_id) REFERENCES Paciente(ID);

ALTER TABLE Doc_pat
ADD CONSTRAINT fk_DocID
FOREIGN KEY (DoctorID) REFERENCES Doctor(ID),
ADD CONSTRAINT fk_PacientesID
FOREIGN KEY (PacienteID) REFERENCES Paciente(ID);

```

Ya con las tablas creadas podemos visualizar el esquema generado del mismo modo que con sql developer, asegurando el correcto funcionamiento y acomodo de los datos.

