

<b>Sistemas de Gestión Empresarial</b>	<b>Ejercicios</b>	1ª Evaluación
<b>P2.1 - PostgreSQL</b>	<b>Grupo</b>	<b>DAMT2</b>
<b>Nombre y apellidos: Díez de Paulino Albano</b>	<b>Fecha</b>	25/10/2023

En la presente práctica se evaluarán los siguientes criterios de evaluación:

RA1- CE e	Se han verificado las configuraciones del sistema operativo y del gestor de datos para garantizar la funcionalidad del ERP-CRM.
RA1- CE f	Se han documentado las operaciones realizadas.
RA1- CE g	Se han documentado las incidencias producidas durante el proceso.

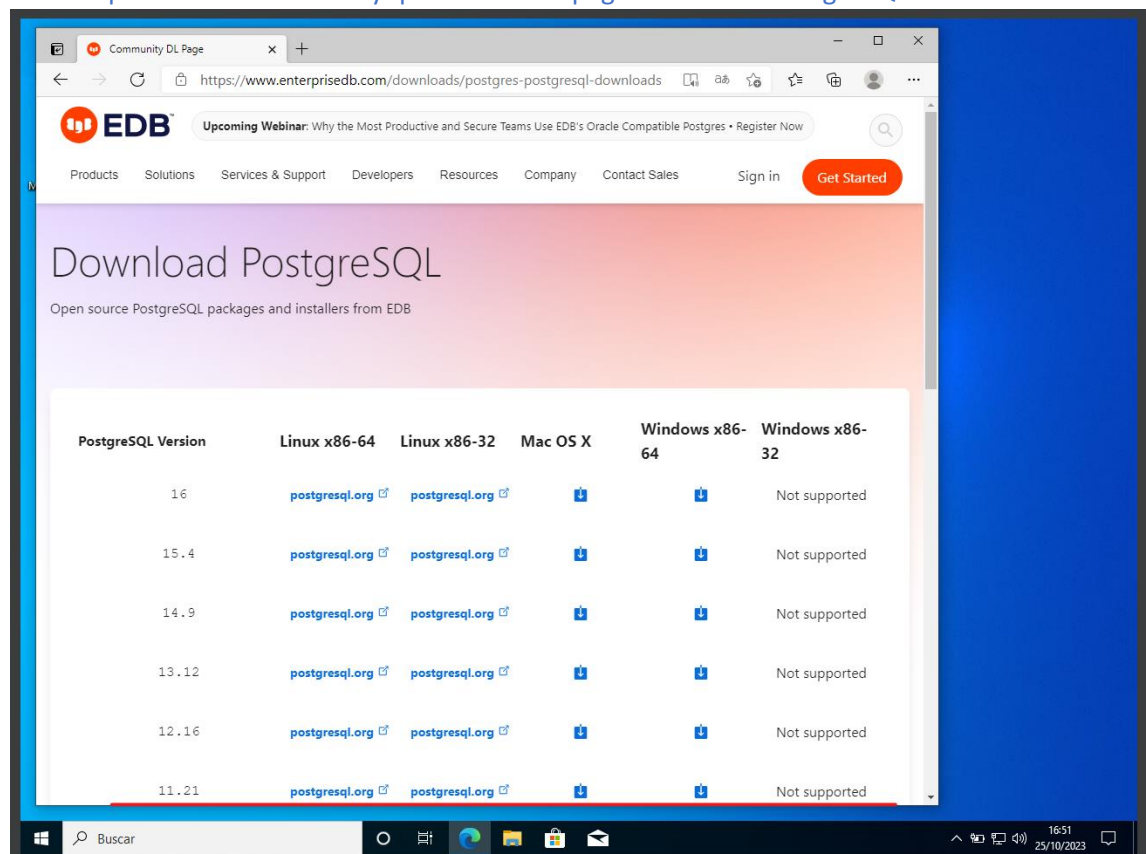
### Instalación y manipulación PostgreSQL y pgAdmin

El objetivo de esta práctica es instalar el gestor de bases de datos PostgreSQL y la herramienta gráfica pgAdmin y la creación y manipulación de BBDD usando dichas herramientas. La práctica se desarrollará en una **máquina virtual con Windows**.

Pasos:

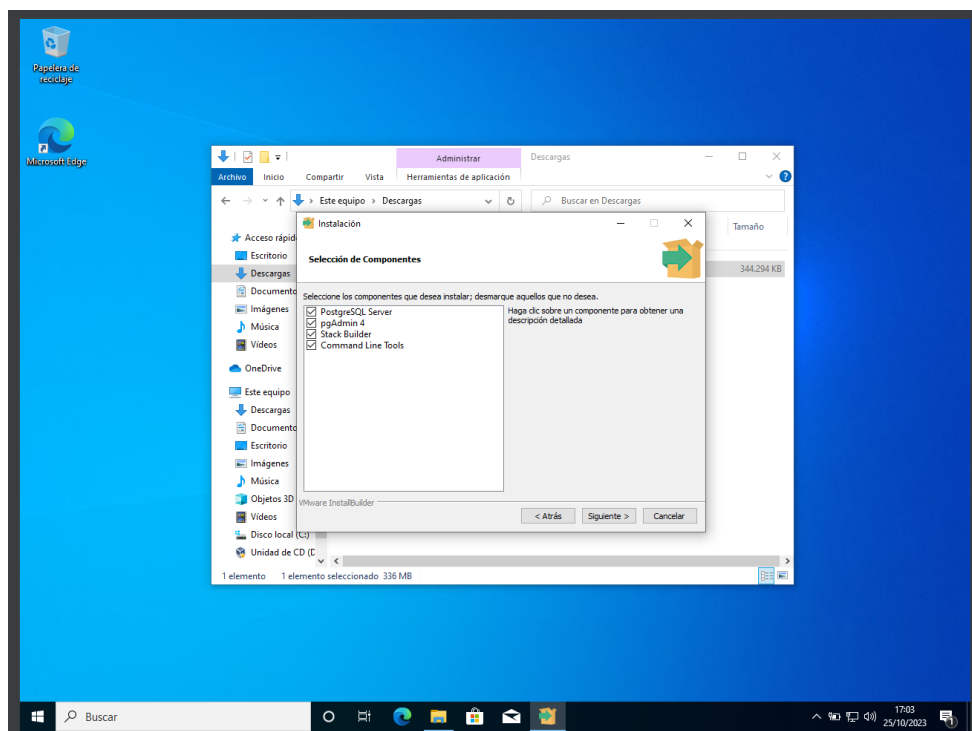
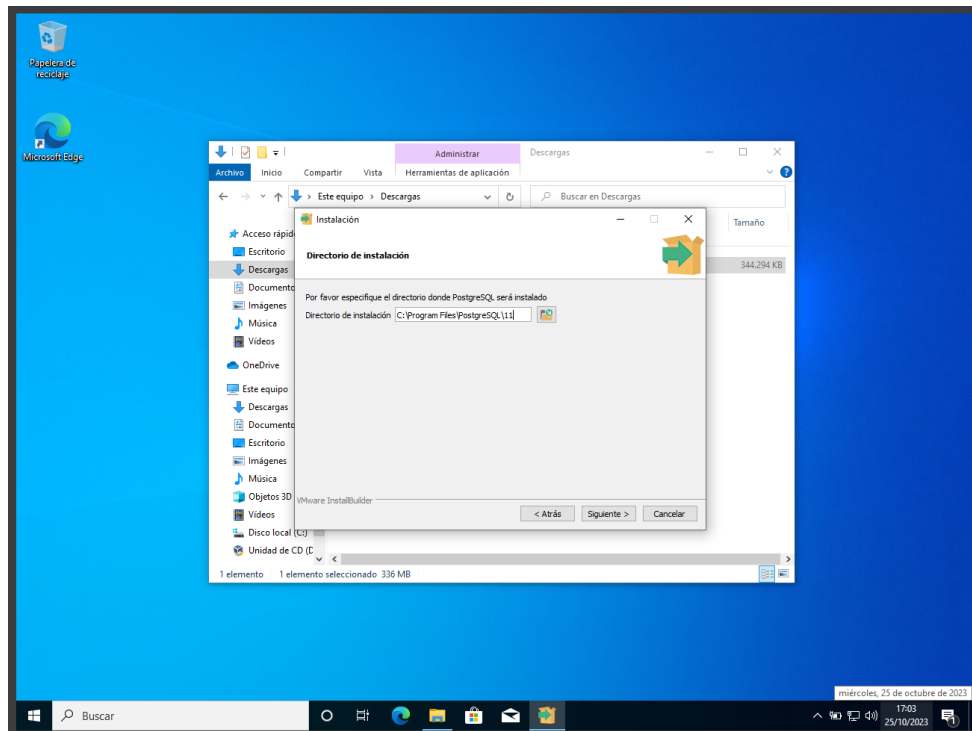
- **Instalar la versión 11 del SGBD PostgreSQL y la versión 4 de la herramienta gráfica pgAdmin (1 punto).**

Para empezar la instalación hay que acudir a la [página oficial de PostgreSQL](https://www.enterprisedb.com/downloads/postgres-postgresql-downloads).



Para Windows 10 se descarga el típico archivo instalable de Windows, que hay que darle siguiente hasta la configuración importante.

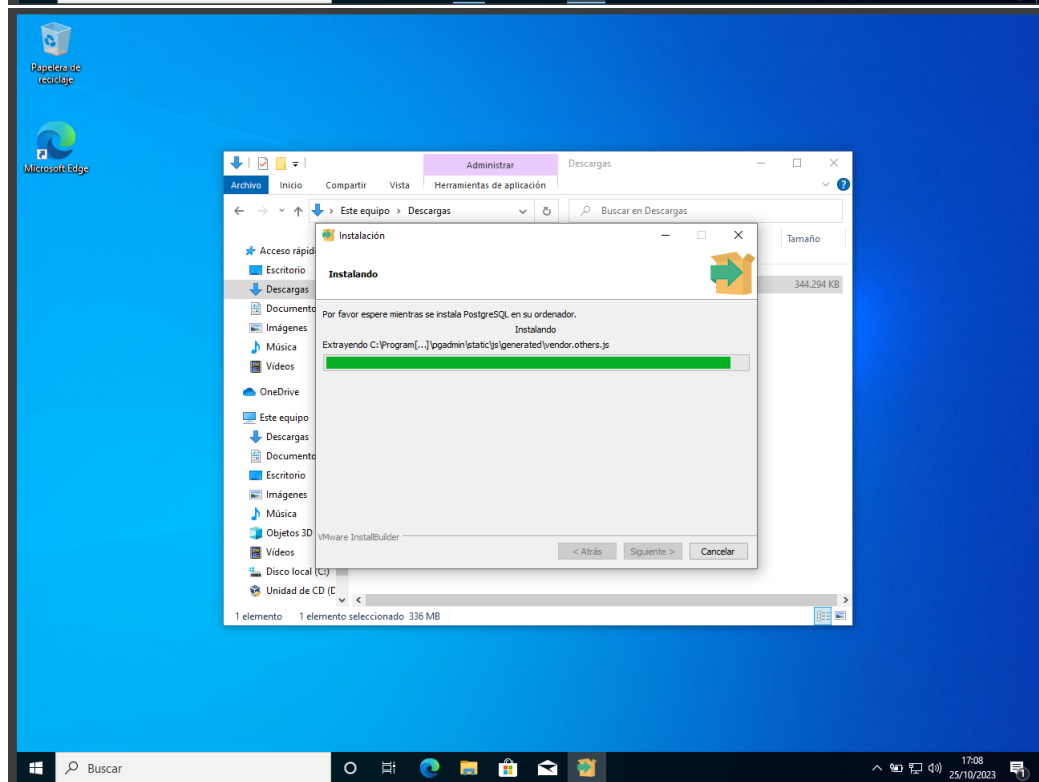
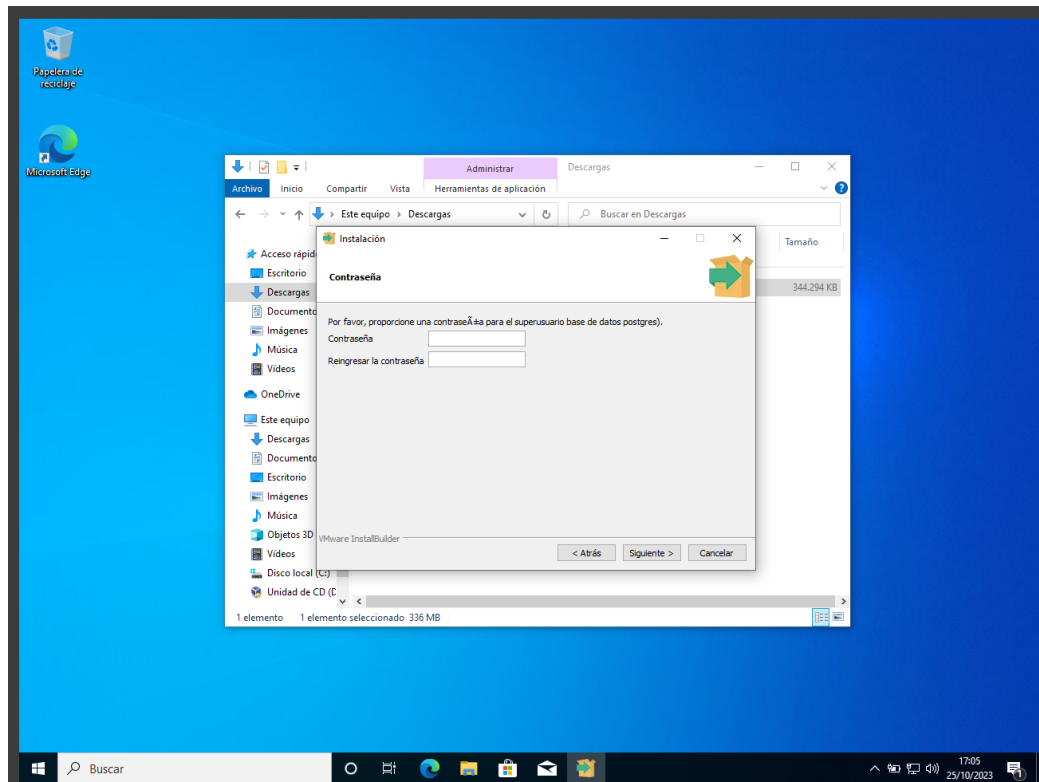
<b>Sistemas de Gestión Empresarial</b>	<b>Ejercicios</b>	<b>1ª Evaluación</b>
<b>P2.1 - PostgreSQL</b>	<b>Grupo</b>	<b>DAMT2</b>
<b>Nombre y apellidos: Díez de Paulino Albano</b>	<b>Fecha</b>	<b>25/10/2023</b>



El instalador automáticamente te descarga la consola de PostgreSQL y la interfaz grafica pgAdmin4.

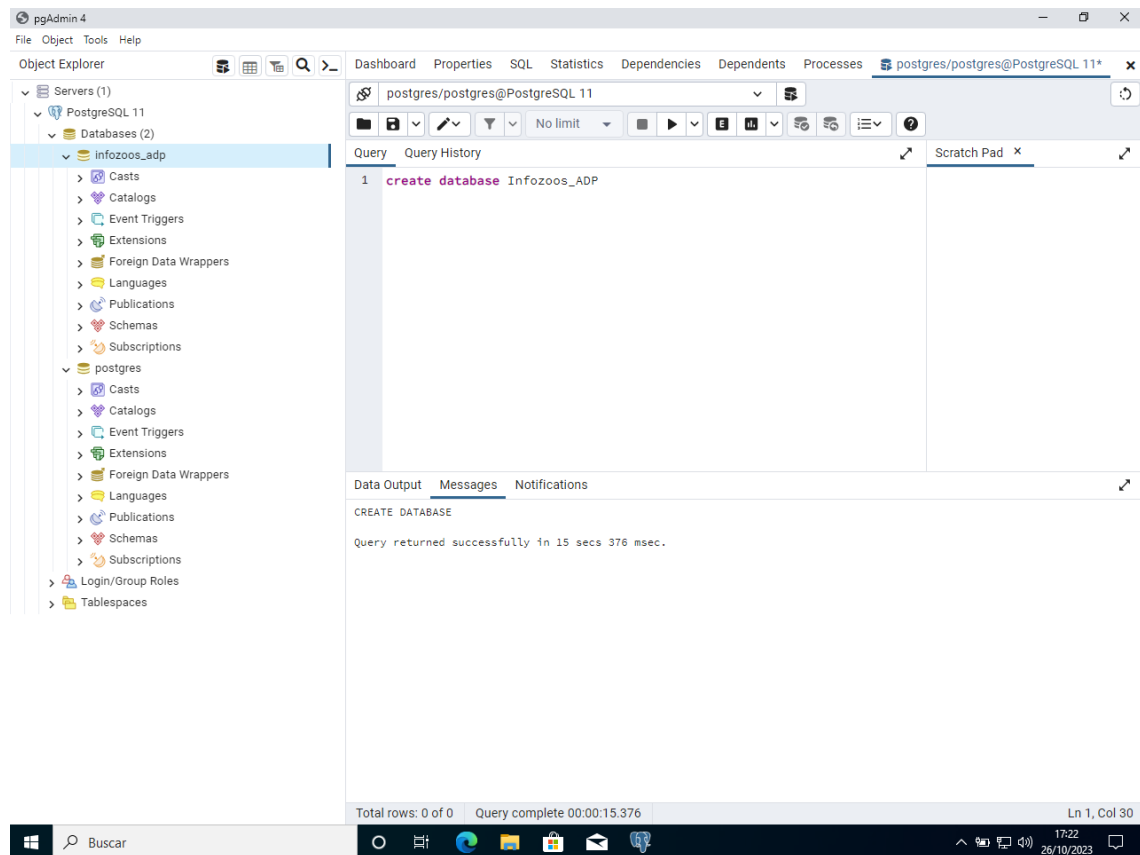
IMPORTANTE: configuración bien la contraseña del usuario root.

<b>Sistemas de Gestión Empresarial</b>	<b>Ejercicios</b>	<b>1ª Evaluación</b>
<b>P2.1 - PostgreSQL</b>	<b>Grupo</b>	<b>DAMT2</b>
<b>Nombre y apellidos: Díez de Paulino Albano</b>	<b>Fecha</b>	<b>25/10/2023</b>



- Crear una base de datos que guardará información relativa a zoos, así como las especies de animales que albergan. El nombre de la BBDD será: **Infozoos\_InicialNombreInicialApellido1InicialApellido2** (si te llamas Begoña Suárez Gómez, la BBDD se llamará **Infozoos\_BSG**). (0,5 puntos)

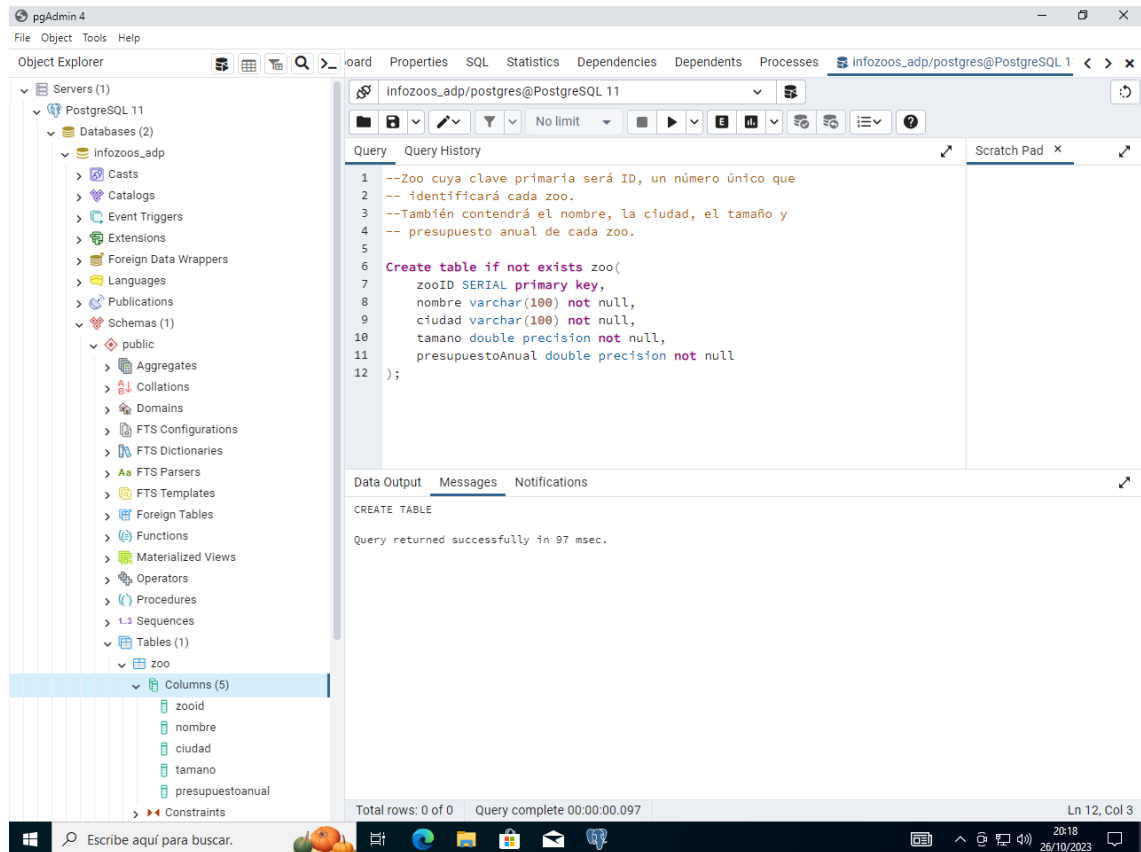
<b>Sistemas de Gestión Empresarial</b>	<b>Ejercicios</b>	1ª Evaluación
<b>P2.1 - PostgreSQL</b>	<b>Grupo</b>	<b>DAMT2</b>
<b>Nombre y apellidos: Díez de Paulino Albano</b>	<b>Fecha</b>	25/10/2023



Sentencia SQL: Create database infozoos\_ADP;

<b>Sistemas de Gestión Empresarial</b>	<b>Ejercicios</b>	<b>1ª Evaluación</b>
<b>P2.1 - PostgreSQL</b>	<b>Grupo</b>	<b>DAMT2</b>
<b>Nombre y apellidos: Díez de Paulino Albano</b>	<b>Fecha</b>	<b>25/10/2023</b>

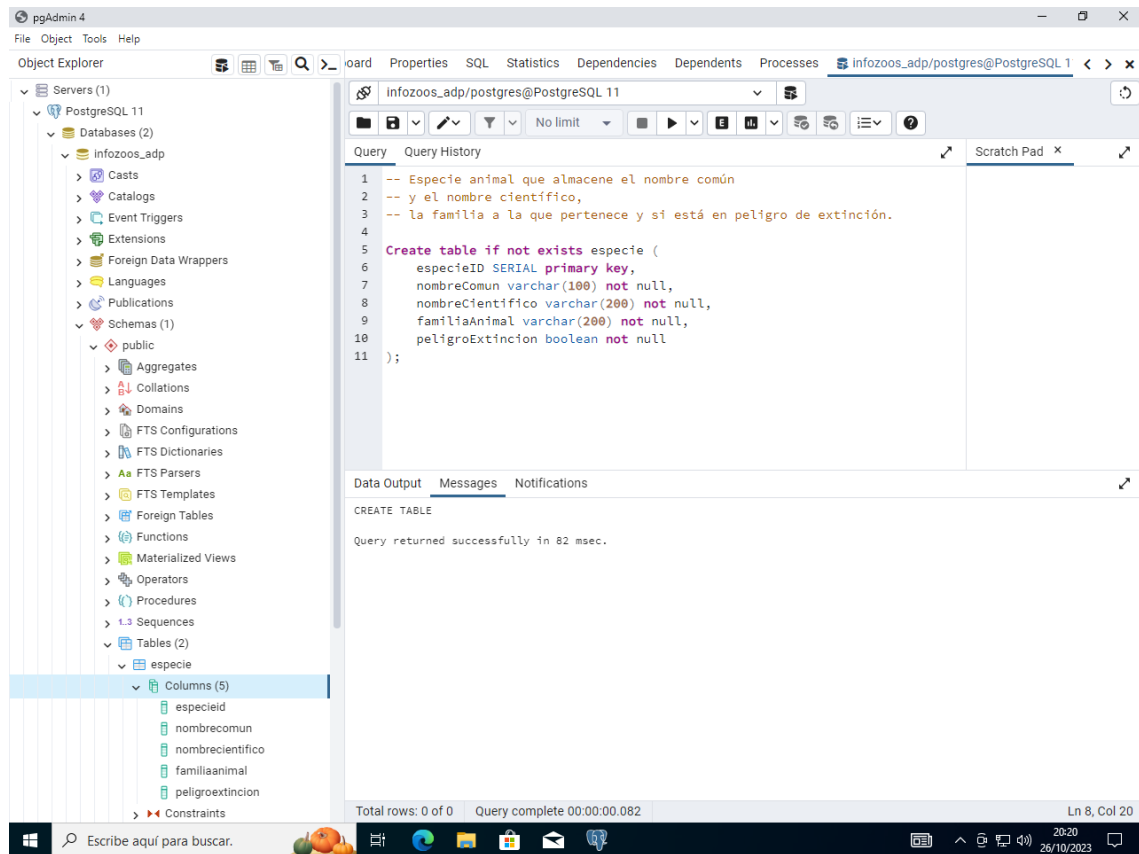
- Crear las siguientes tablas (2 puntos):
  - a. Zoo cuya clave primaria será ID, un número único que identificará cada zoo. También contendrá el nombre, la ciudad, el tamaño y presupuesto anual de cada zoo.



Sentencia SQL: Create table if not exists zoo(  
 zooID SERIAL primary key,  
 nombre varchar(100) not null,  
 ciudad varchar(100) not null,  
 tamaño double precision not null,  
 presupuestoAnual double precision not null  
 );

<b>Sistemas de Gestión Empresarial</b>	<b>Ejercicios</b>	<b>1ª Evaluación</b>
<b>P2.1 - PostgreSQL</b>	<b>Grupo</b>	<b>DAMT2</b>
<b>Nombre y apellidos: Díez de Paulino Albano</b>	<b>Fecha</b>	<b>25/10/2023</b>

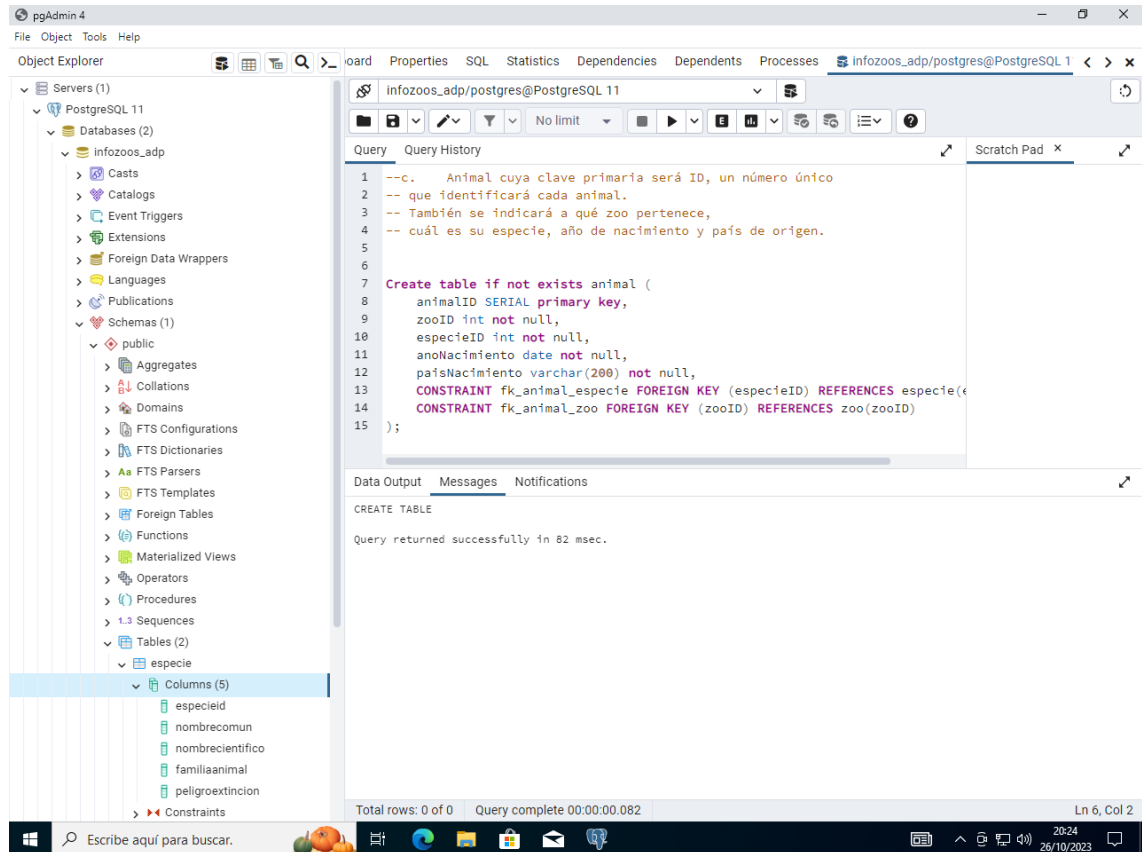
- b. Especie animal que almacene el nombre común y el nombre científico, la familia a la que pertenece y si está en peligro de extinción.



Sentencia SQL: Create table if not exists especie (  
 especieID SERIAL primary key,  
 nombreComun varchar(100) not null,  
 nombreCientifico varchar(200) not null,  
 familiaAnimal varchar(200) not null,  
 peligroExtincion boolean not null  
 );

<b>Sistemas de Gestión Empresarial</b>	<b>Ejercicios</b>	<b>1ª Evaluación</b>
<b>P2.1 - PostgreSQL</b>	<b>Grupo</b>	<b>DAMT2</b>
<b>Nombre y apellidos: Díez de Paulino Albano</b>	<b>Fecha</b>	<b>25/10/2023</b>

- c. Animal cuya clave primaria será ID, un número único que identificará cada animal. También se indicará a qué zoo pertenece, cuál es su especie, año de nacimiento y país de origen.

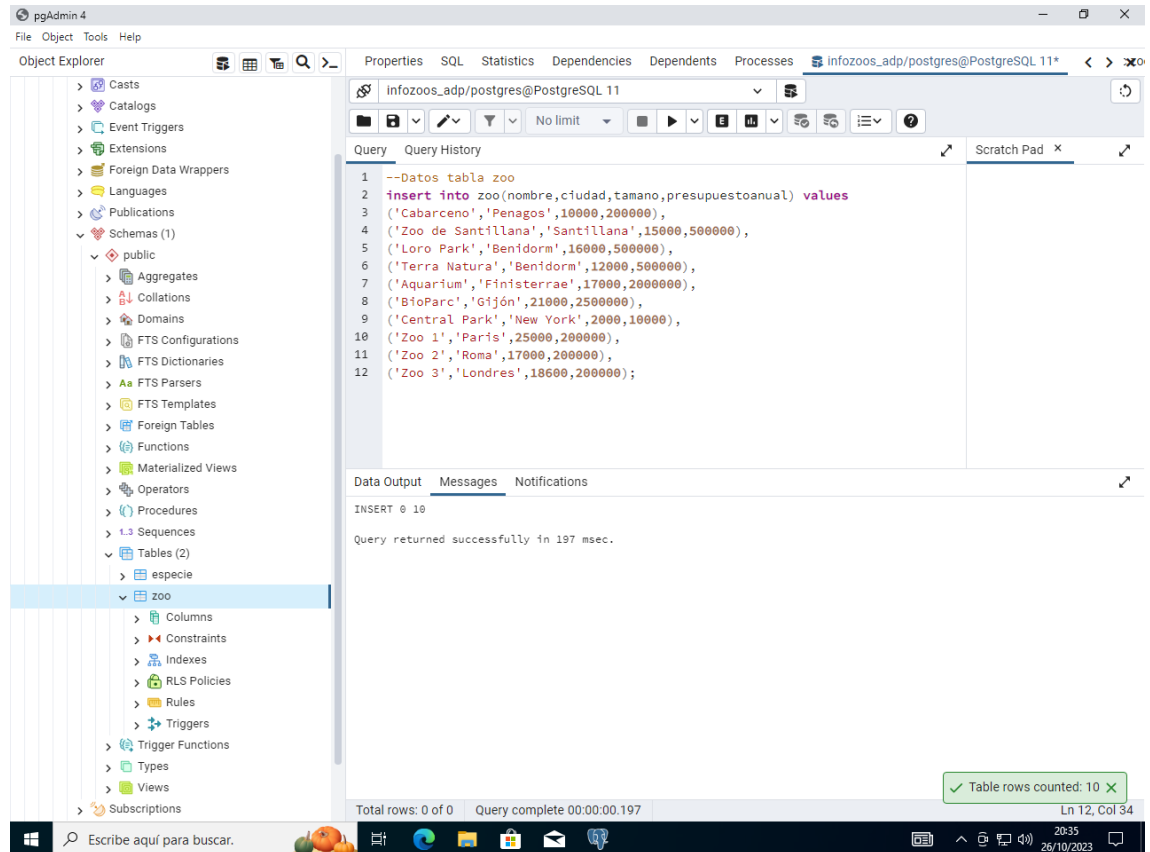


Sentencia SQL: Create table if not exists animal (  
 animalID SERIAL primary key,  
 zooID int not null,  
 especieID int not null,  
 anoNacimiento date not null,  
 paisNacimiento varchar(200) not null,  
 CONSTRAINT fk\_animal\_especie FOREIGN KEY (especieID) REFERENCES  
 especie(especieID),  
 CONSTRAINT fk\_animal\_zoo FOREIGN KEY (zooID) REFERENCES zoo(zooID)  
 );

<b>Sistemas de Gestión Empresarial</b>	<b>Ejercicios</b>	1ª Evaluación
<b>P2.1 - PostgreSQL</b>	<b>Grupo</b>	<b>DAMT2</b>
<b>Nombre y apellidos: Díez de Paulino Albano</b>	<b>Fecha</b>	25/10/2023

- **Añadir 10 registros a cada tabla (1 punto).**

#### Datos en Zoo



The screenshot shows the pgAdmin 4 interface. In the 'Query' tab, the following SQL query is entered:

```
--Datos tabla zoo
insert into zoo(nombre,ciudad,tamano,presupuestoanual) values
('Cabarceno','Penagos',10000,200000),
('Zoo de Santillana','Santillana',15000,500000),
('Loro Park','Benidorm',16000,500000),
('Terra Natura','Benidorm',12000,500000),
('Aquarium','Finisterrae',17000,2000000),
('BioParc','Gijón',21000,2500000),
('Central Park','New York',2000,10000),
('Zoo 1','Paris',25000,200000),
('Zoo 2','Roma',17000,200000),
('Zoo 3','Londres',18600,200000);
```

The 'Messages' tab shows the following output:

```
INSERT 0 10
Query returned successfully in 197 msec.
```

The 'Object Explorer' on the left shows the database structure, including the 'zoo' table under the 'public' schema.

Sentencia SQL: insert into zoo(nombre,ciudad,tamano,presupuestoanual) values  
('Cabarceno','Penagos',10000,200000),  
('Zoo de Santillana','Santillana',15000,500000),  
('Loro Park','Benidorm',16000,500000),  
('Terra Natura','Benidorm',12000,500000),  
('Aquarium','Finisterrae',17000,2000000),  
('BioParc','Gijón',21000,2500000),  
('Central Park','New York',2000,10000),  
('Zoo 1','Paris',25000,200000),  
('Zoo 2','Roma',17000,200000),  
('Zoo 3','Londres',18600,200000);

Comprobación:



<b>Sistemas de Gestión Empresarial</b>	<b>Ejercicios</b>	<b>1ª Evaluación</b>
<b>P2.1 - PostgreSQL</b>	<b>Grupo</b>	<b>DAMT2</b>
<b>Nombre y apellidos: Díez de Paulino Albano</b>	<b>Fecha</b>	<b>25/10/2023</b>

pgAdmin 4

File Object Tools Help

Object Explorer

- Castes
- Catalogs
- Event Triggers
- Extensions
- Foreign Data Wrappers
- Languages
- Publications
- Schemas (1)
  - public
    - Aggregates
    - Collations
    - Domains
    - FTS Configurations
    - FTS Dictionaries
    - FTS Parsers
    - FTS Templates
    - Foreign Tables
    - Functions
    - Materialized Views
    - Operators
    - Procedures
    - Sequences
    - Tables (2)
      - especie
      - zoo**
        - Columns
        - Constraints
        - Indexes
        - RLS Policies
        - Rules
        - Triggers
        - Trigger Functions
        - Types
        - Views
        - Subscriptions

Processes

infozoos\_adp/... public.zoo/infozoos\_adp/postgres@PostgreSQL 11

public.zoo/infozoos\_adp/postgres@PostgreSQL 11

Query Query History

```

1 SELECT * FROM public.zoo
2 ORDER BY zooid ASC

```

Data Output Messages Notifications

	zooid [PK] integer	nombre character varying (100)	ciudad character varying (100)	tamano double precision	presupuestoanual double precision
1	1	Cabarceno	Penagos	10000	200000
2	2	Zoo de Santillana	Santillana	15000	500000
3	3	Loro Park	Benidorm	16000	500000
4	4	Terra Natura	Benidorm	12000	500000
5	5	Aquarium	Finisterrae	17000	2000000
6	6	BioParc	Gijón	21000	2500000
7	7	Central Park	New York	2000	10000
8	8	Zoo 1	Paris	25000	200000
9	9	Zoo 2	Roma	17000	200000
10	10	Zoo 3	Londres	18600	200000

Total rows: 10 of 10 Query complete 00:00:00.731 Ln 1, Col 1

Table rows counted: 10 X

Escribe aquí para buscar.

## Datos Especie

pgAdmin 4

File Object Tools Help

Object Explorer

- PostgreSQL 11
  - Databases (2)
    - Infozoos\_adp
      - Castes
      - Catalogs
      - Event Triggers
      - Extensions
      - Foreign Data Wrappers
      - Languages
      - Publications
      - Schemas (1)
        - public
          - Aggregates
          - Collations
          - Domains
          - FTS Configurations
          - FTS Dictionaries
          - FTS Parsers
          - FTS Templates
          - Foreign Tables
          - Functions
          - Materialized Views
          - Operators
          - Procedures
          - Sequences
          - Tables (2)
            - especie
              - Columns (5)
              - Constraints
              - Indexes
              - RLS Policies
              - Rules
              - Triggers
              - Trigger Functions
            - zoo

Dependents Processes

infozoos\_adp/postgres@PostgreSQL 11\*

infozoos\_adp/postgres@PostgreSQL 11

Query Query History

```

1 --datos Especie
2 insert into especie(nombrecomun,nombrecientifico,familiaanimal,peligroextinc
3 ('Perro','Canis familiaris','Canidos',false),
4 ('Gato','Felis silvestris catus','Felidae',false),
5 ('Elefante africano','Loxodonta africana', 'Elephantidae',true),
6 ('León','Panthera leo', 'Felidae',true),
7 ('Cebra','Equus quagga','Equidae',true),
8 ('Girafa','Giraffa camelopardalis','Giraffidae',false),
9 ('Tigre de Bengala','Panthera tigris tigris','Felidae',false),
10 ('Oso polar','Ursus maritimus','Ursidae',false),
11 ('Rinoceronte de Java','Rhinoceros sondaicus','Rhinocerotidae',true),
12 ('Tigre de Sumatra','Panthera tigris sumatrae','Felidae',true),
13 ('Oso polar','Ursus maritimus','Ursidae',true);

```

Data Output Messages Notifications

INSERT 0 11

Query returned successfully in 234 msec.

Total rows: 0 of 0 Query complete 00:00:00.234 Ln 7, Col 10

Table rows counted: 10 X

Escribe aquí para buscar.

<b>Sistemas de Gestión Empresarial</b>	<b>Ejercicios</b>	1ª Evaluación
<b>P2.1 - PostgreSQL</b>	<b>Grupo</b>	<b>DAMT2</b>
<b>Nombre y apellidos: Díez de Paulino Albano</b>	<b>Fecha</b>	25/10/2023

Sentencia SQL: insert into  
 especie(nombrecomun,nombrecientifico,familiaanimal,peligroextincion) values  
 ('Perro','Canis familiaris','Canidos',false),  
 ('Gato','Felis silvestris catus','Felidae',false),  
 ('Elefante africano','Loxodonta africana','Elephantidae',true),  
 ('León','Panthera leo','Felidae',true),  
 ('Cebra','Equus quagga','Equidae',true),  
 ('Jirafa Giraffa','camelopardalis','Giraffidae',false),  
 ('Tigre de Bengala','Panthera tigris tigris','Felidae',false),  
 ('Oso polar','Ursus maritimus','Ursidae',false),  
 ('Rinoceronte de Java','Rhinoceros sondaicus','Rhinocerotidae',true),  
 ('Tigre de Sumatra','Panthera tigris sumatrae','Felidae',true),  
 ('Oso polar','Ursus maritimus','Ursidae',true);

Datos animal:

The screenshot displays the pgAdmin 4 interface. On the left, the 'Object Explorer' shows the database structure, including the 'public' schema and the 'animal' table. The central pane shows the SQL query editor with the following SQL statement:

```

1 insert into animal(zooId,especieId,anonacimiento,paisnacimiento) values
2 (1,2,'2018-01-01','Canada'),
3 (5,3,'2019-01-01','Japon'),
4 (5,3,'2020-01-01','Japon'),
5 (6,5,'2021-01-01','Japon'),
6 (6,5,'2022-01-01','Japon'),
7 (7,1,'2018-01-01','Venezuela'),
8 (7,1,'2019-01-01','Venezuela'),
9 (2,2,'2020-01-01','Portugal'),
10 (10,1,'2021-01-01','Nigeria'),
11 (9,4,'2022-01-01','Puerto Rico');
  
```

The 'Data Output' pane at the bottom shows the execution results:

```

INSERT 0 10
Query returned successfully in 447 msec.
  
```

A green notification box at the bottom right indicates: "Table rows counted: 10". The status bar at the bottom shows "Total rows: 0 of 0" and "Query complete 00:00:00.447".

<b>Sistemas de Gestión Empresarial</b>	<b>Ejercicios</b>	<b>1ª Evaluación</b>
<b>P2.1 - PostgreSQL</b>	<b>Grupo</b>	<b>DAMT2</b>
<b>Nombre y apellidos: Díez de Paulino Albano</b>	<b>Fecha</b>	<b>25/10/2023</b>

Sentencia SQL:

```
insert into animal(zoooid,especieid,anonacimiento,paisnacimiento) values
(1,2,'2018-01-01','Canada'),
(5,3,'2019-01-01','Japon'),
(5,3,'2020-01-01','Japon'),
(6,5,'2021-01-01','Japon'),
(6,5,'2022-01-01','Japon'),
(7,1,'2018-01-01','Venezuela'),
(7,1,'2019-01-01','Venezuela'),
(2,2,'2020-01-01','Portugal'),
(10,1,'2021-01-01','Nigeria'),
(9,4,'2022-01-01','Puerto Rico');
```

- Realizar las siguientes consultas (2 puntos):
  - Mostrar los animales que pertenecen a un determinado zoo.

The screenshot shows the pgAdmin 4 interface. On the left, the 'Object Explorer' pane displays the database structure, with the 'zoo' table selected under the 'public' schema. The main query editor shows the following SQL query:

```
1 select * from animal inner join zoo on animal.zoooid = zoo.zoooid where nombre
```

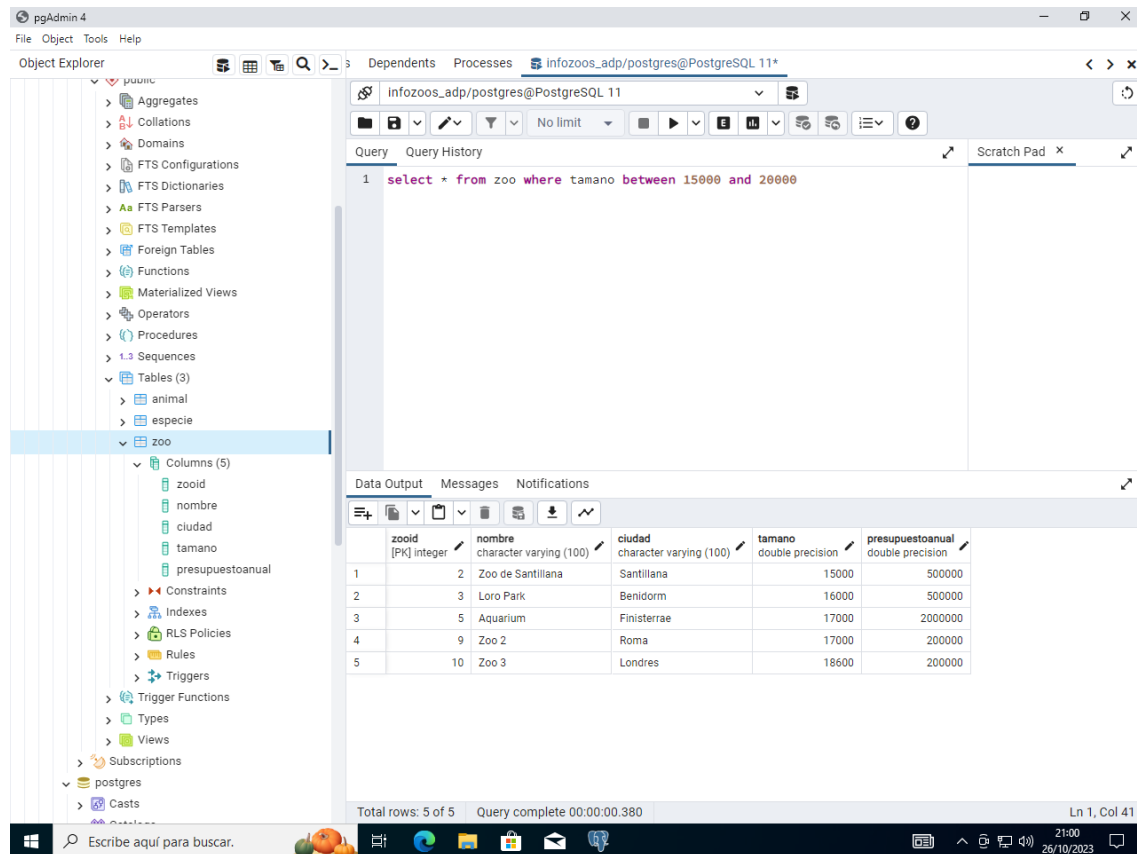
The 'Data Output' pane at the bottom displays the results of the query. The table has 9 columns: animalid, zoooid, especieid, anonacimiento, paisnacimiento, zoooid, nombre, and ciudad. The results show one row of data:

animalid	zoooid	especieid	anonacimiento	paisnacimiento	zoooid	nombre	ciudad
1	8	2	2020-01-01	Portugal	2	Zoo de Santillana	Santillana

Sentencia SQL: `select * from animal inner join zoo on animal.zoooid = zoo.zoooid where nombre = 'Zoo de Santillana'`

<b>Sistemas de Gestión Empresarial</b>	<b>Ejercicios</b>	<b>1ª Evaluación</b>
<b>P2.1 - PostgreSQL</b>	<b>Grupo</b>	<b>DAMT2</b>
<b>Nombre y apellidos: Díez de Paulino Albano</b>	<b>Fecha</b>	<b>25/10/2023</b>

b) **Mostrar los zoos cuyo tamaño esté entre 15000 y 20000 m2.**



The screenshot shows the pgAdmin 4 interface. On the left, the 'Object Explorer' shows the database structure, with the 'zoo' table selected under the 'public' schema. The 'Query' tab is active, displaying the following SQL query:

```
1 select * from zoo where tamano between 15000 and 20000
```

The 'Data Output' tab shows the results of the query in a table with 5 columns: **zoooid** (integer), **nombre** (character varying (100)), **ciudad** (character varying (100)), **tamano** (double precision), and **presupuestoanual** (double precision). The results are as follows:

zoooid	nombre	ciudad	tamano	presupuestoanual
1	Zoo de Santillana	Santillana	15000	500000
2	Loro Park	Benidorm	16000	500000
3	Aquarium	Finisterrae	17000	2000000
4	Zoo 2	Roma	17000	200000
5	Zoo 3	Londres	18600	200000

The status bar at the bottom indicates 'Total rows: 5 of 5' and 'Query complete 00:00:00.380'.

**Sentencia SQL: select \* from zoo where tamano between 15000 and 20000**

c) **Mostrar los animales que estén en peligro de extinción indicando, también a qué zoo pertenecen.**

<b>Sistemas de Gestión Empresarial</b>	<b>Ejercicios</b>	<b>1ª Evaluación</b>
<b>P2.1 - PostgreSQL</b>	<b>Grupo</b>	<b>DAMT2</b>
<b>Nombre y apellidos: Díez de Paulino Albano</b>	<b>Fecha</b>	<b>25/10/2023</b>

The screenshot shows the pgAdmin 4 interface. On the left, the 'Object Explorer' pane displays the database structure, with the 'zoo' table selected under the 'animal' schema. The main query editor shows the following SQL query:

```
1 select animalid,nombre,animal.especieid,anonacimiento,paisnacimiento,peligroextincion from animal inner join zoo on animal.zooid = zoo.zooid inner join especie on animal.especieid = especie.especieid where peligroextincion is true
```

The 'Data Output' pane displays the results of the query in a table format:

	animalid integer	nombre character varying (100)	especieid integer	anonacimiento date	paisnacimiento character varying (200)	peligroextincion boolean
1	3	Aquarium	3	2020-01-01	Japon	true
2	2	Aquarium	3	2019-01-01	Japon	true
3	5	BioParc	5	2022-01-01	Japon	true
4	4	BioParc	5	2021-01-01	Japon	true
5	10	Zoo 2	4	2022-01-01	Puerto Rico	true

The status bar at the bottom indicates 'Total rows: 5 of 5' and 'Query complete 00:00:00.662'.

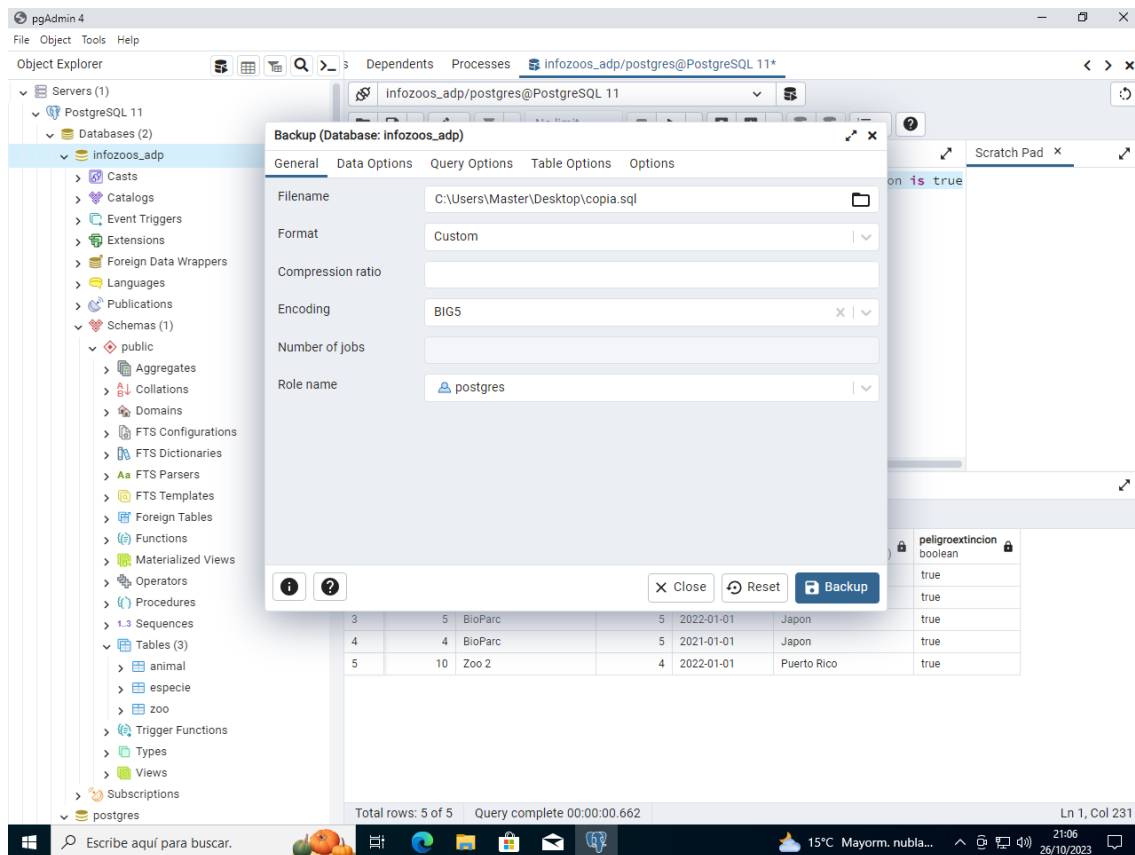
**Sentencia SQL: select**

**animalid,nombre,animal.especieid,anonacimiento,paisnacimiento,peligroextincion from animal inner join zoo on animal.zooid = zoo.zooid inner join especie on animal.especieid = especie.especieid where peligroextincion is true**

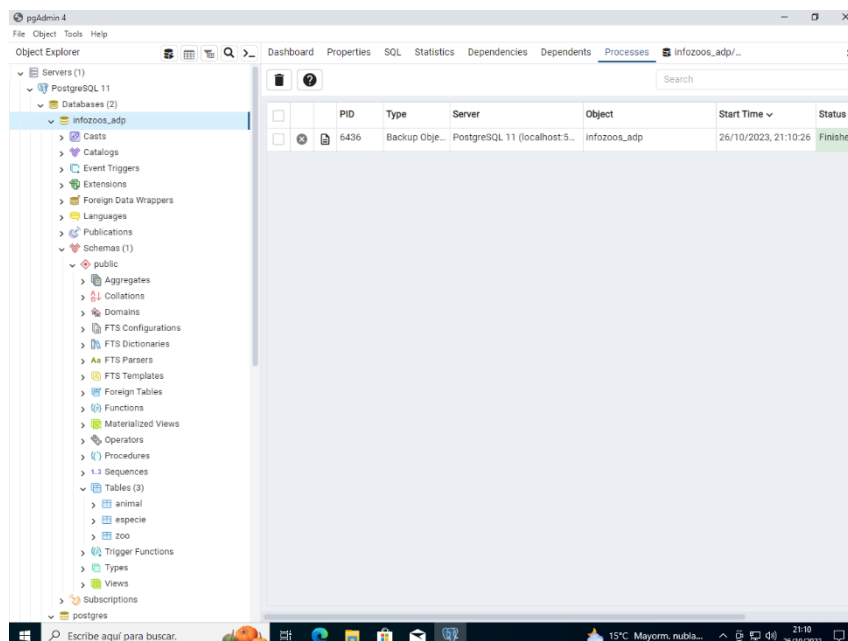
- **Crear una copia de seguridad de la base de datos (1 punto).**

Desplegando el menú contextual sobre la base de datos que quieres hacer la copia de seguridad, se selecciona la opción backup y se genera un archivo sql que se puede importar de vuelta.

<b>Sistemas de Gestión Empresarial</b>	<b>Ejercicios</b>	<b>1ª Evaluación</b>
<b>P2.1 - PostgreSQL</b>	<b>Grupo</b>	<b>DAMT2</b>
<b>Nombre y apellidos: Díez de Paulino Albano</b>	<b>Fecha</b>	<b>25/10/2023</b>



### Comprobación:

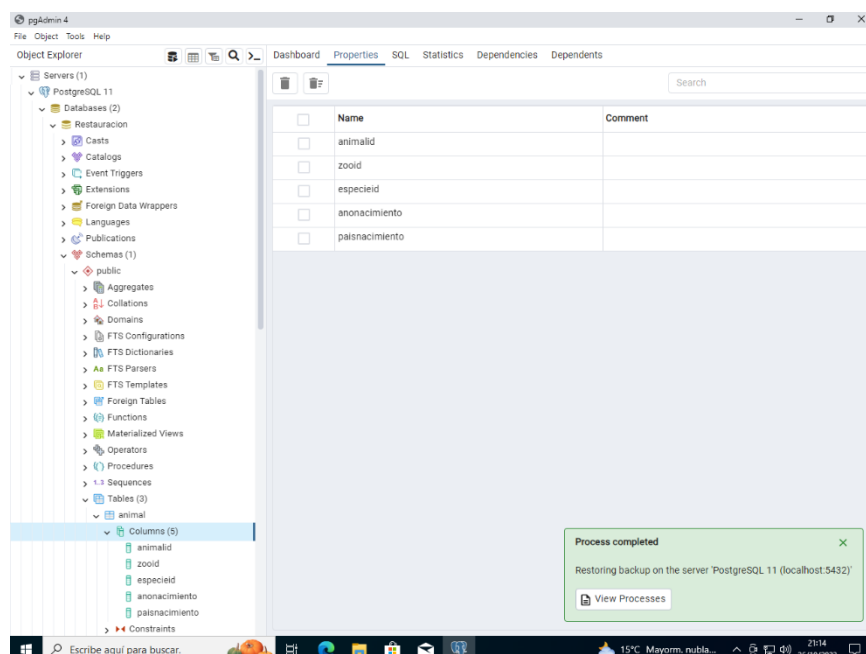
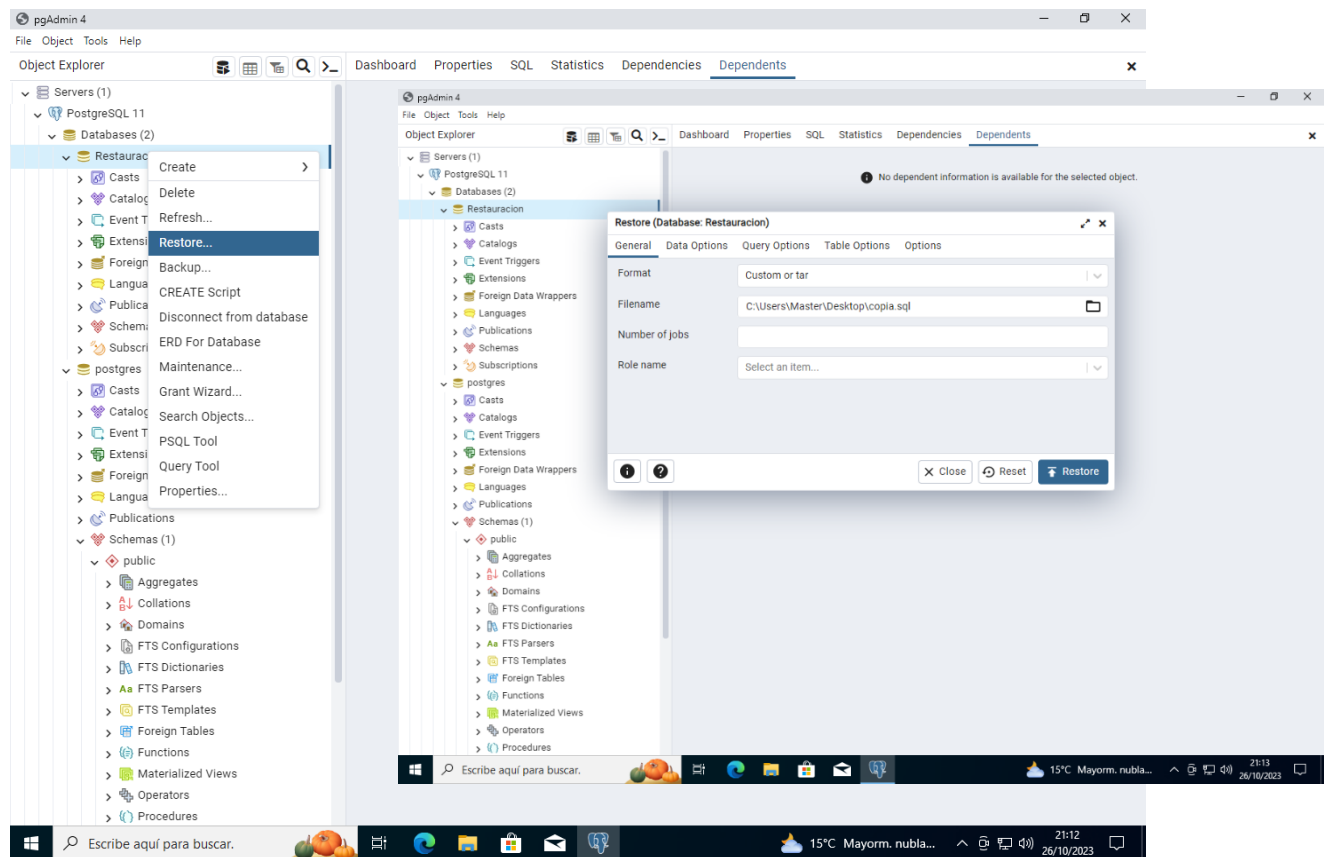


<b>Sistemas de Gestión Empresarial</b>	<b>Ejercicios</b>	<b>1ª Evaluación</b>
<b>P2.1 - PostgreSQL</b>	<b>Grupo</b>	<b>DAMT2</b>
<b>Nombre y apellidos: Díez de Paulino Albano</b>	<b>Fecha</b>	<b>25/10/2023</b>

- **Eliminar la base de datos y restaurarla (1 punto).**

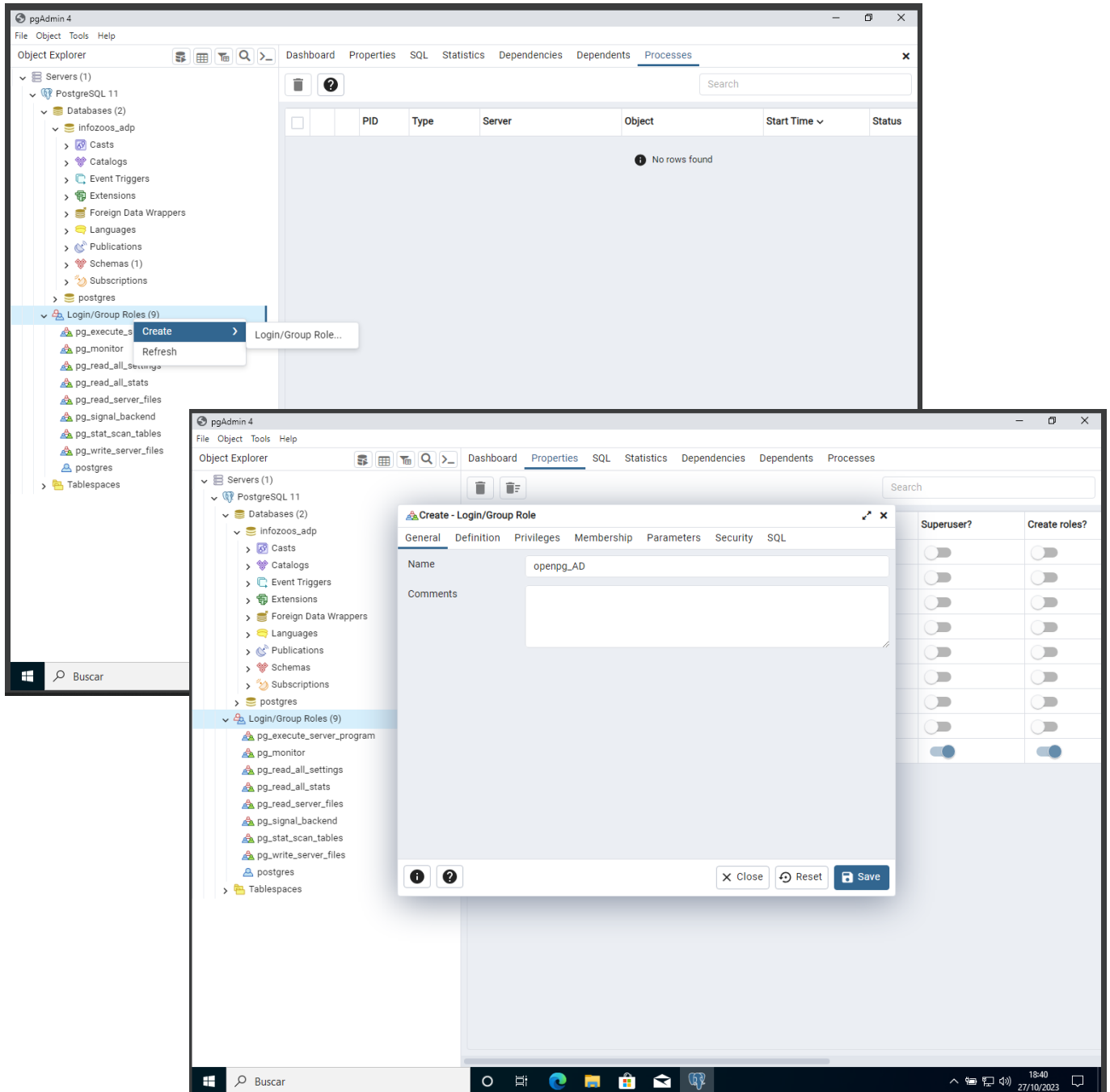
Para borrar la base de datos solo hay que usar esta sentencia SQL: `drop database infozoos_adp;`

Para restaurar se crea una base de datos nueva y restaurar con el archivo generado previamente



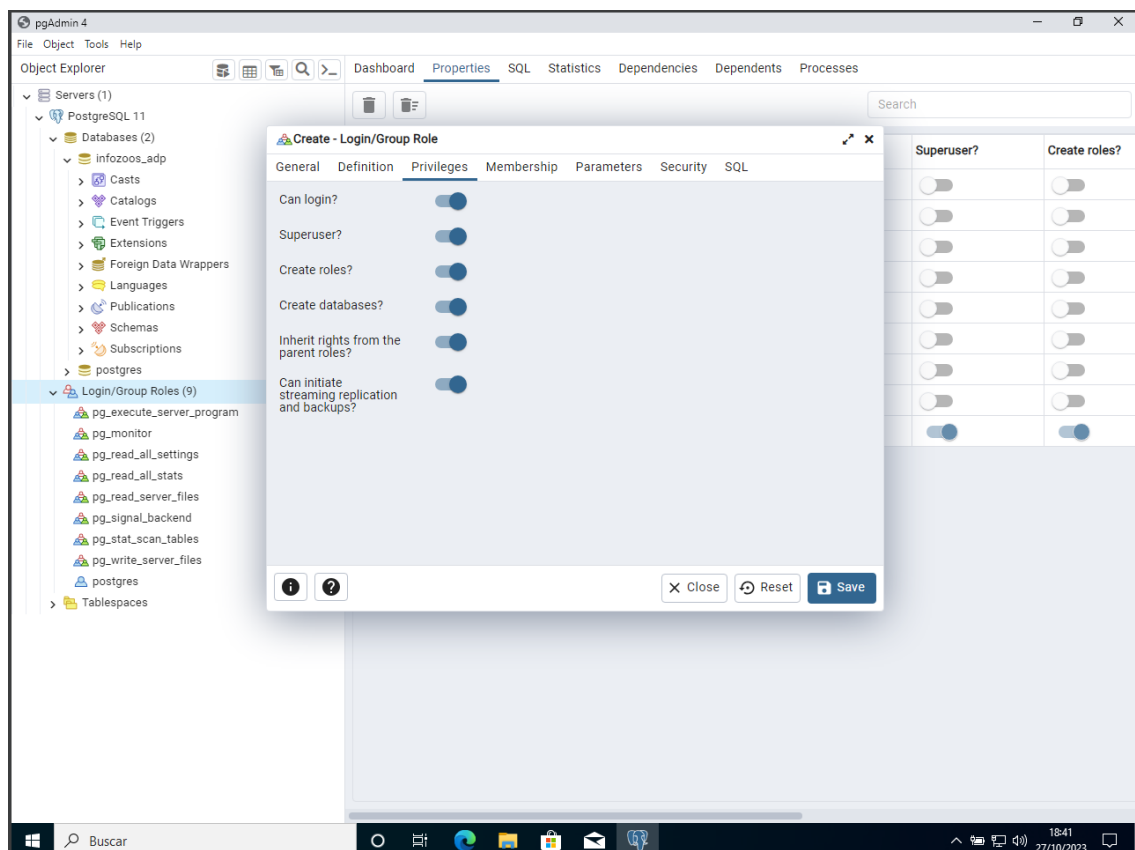
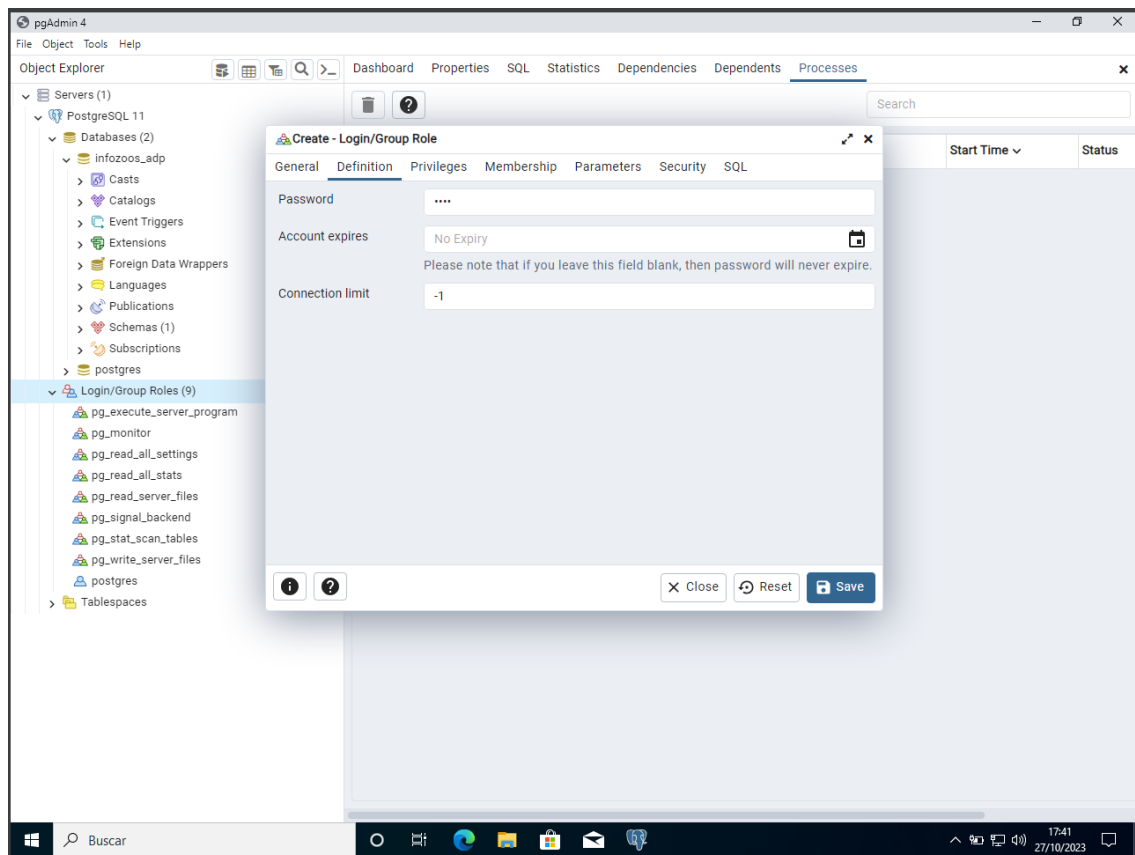
<b>Sistemas de Gestión Empresarial</b>	<b>Ejercicios</b>	1ª Evaluación
<b>P2.1 - PostgreSQL</b>	<b>Grupo</b>	<b>DAMT2</b>
<b>Nombre y apellidos: Díez de Paulino Albano</b>	<b>Fecha</b>	25/10/2023

- Crear el usuario **openpg\_XY** (donde X es la inicial de tu nombre e Y la inicial de tu apellido) con permisos de súper usuario. Este usuario es el usuario desde el que nos conectaremos a PostgreSQL cuando realicemos la instalación en un entorno Windows. (0,5 puntos)





<b>Sistemas de Gestión Empresarial</b>	<b>Ejercicios</b>	<b>1ª Evaluación</b>
<b>P2.1 - PostgreSQL</b>	<b>Grupo</b>	<b>DAMT2</b>
<b>Nombre y apellidos: Díez de Paulino Albano</b>	<b>Fecha</b>	<b>25/10/2023</b>



<b>Sistemas de Gestión Empresarial</b>	<b>Ejercicios</b>	1ª Evaluación
<b>P2.1 - PostgreSQL</b>	<b>Grupo</b>	<b>DAMT2</b>
<b>Nombre y apellidos: Díez de Paulino Albano</b>	<b>Fecha</b>	25/10/2023

### Comprobación de que se ha creado el usuario

