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. |

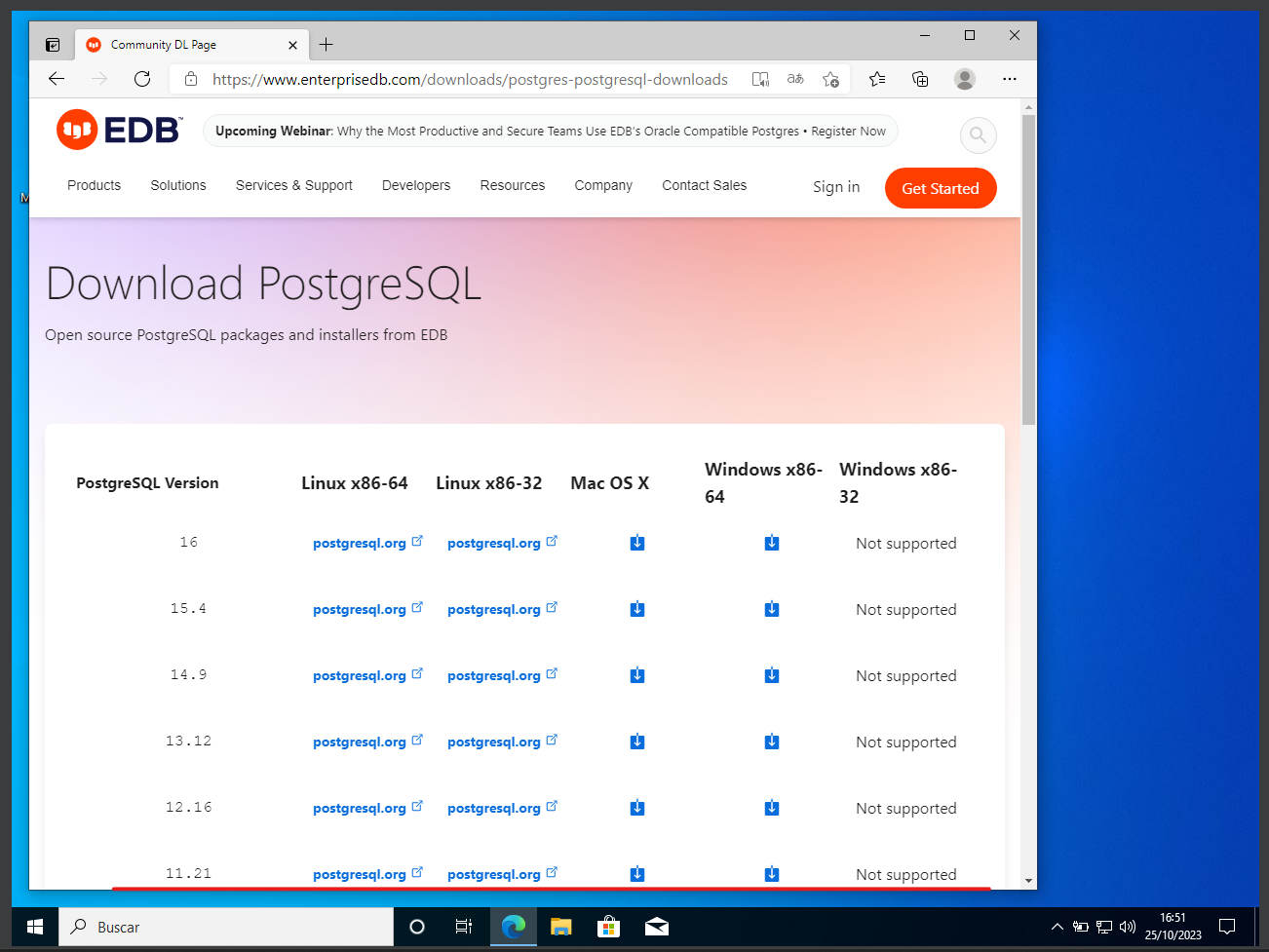
## Instalación y manipulación PostgreSQL y pgAdmin

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.



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

Interfaz de usuario gráfica, Aplicación

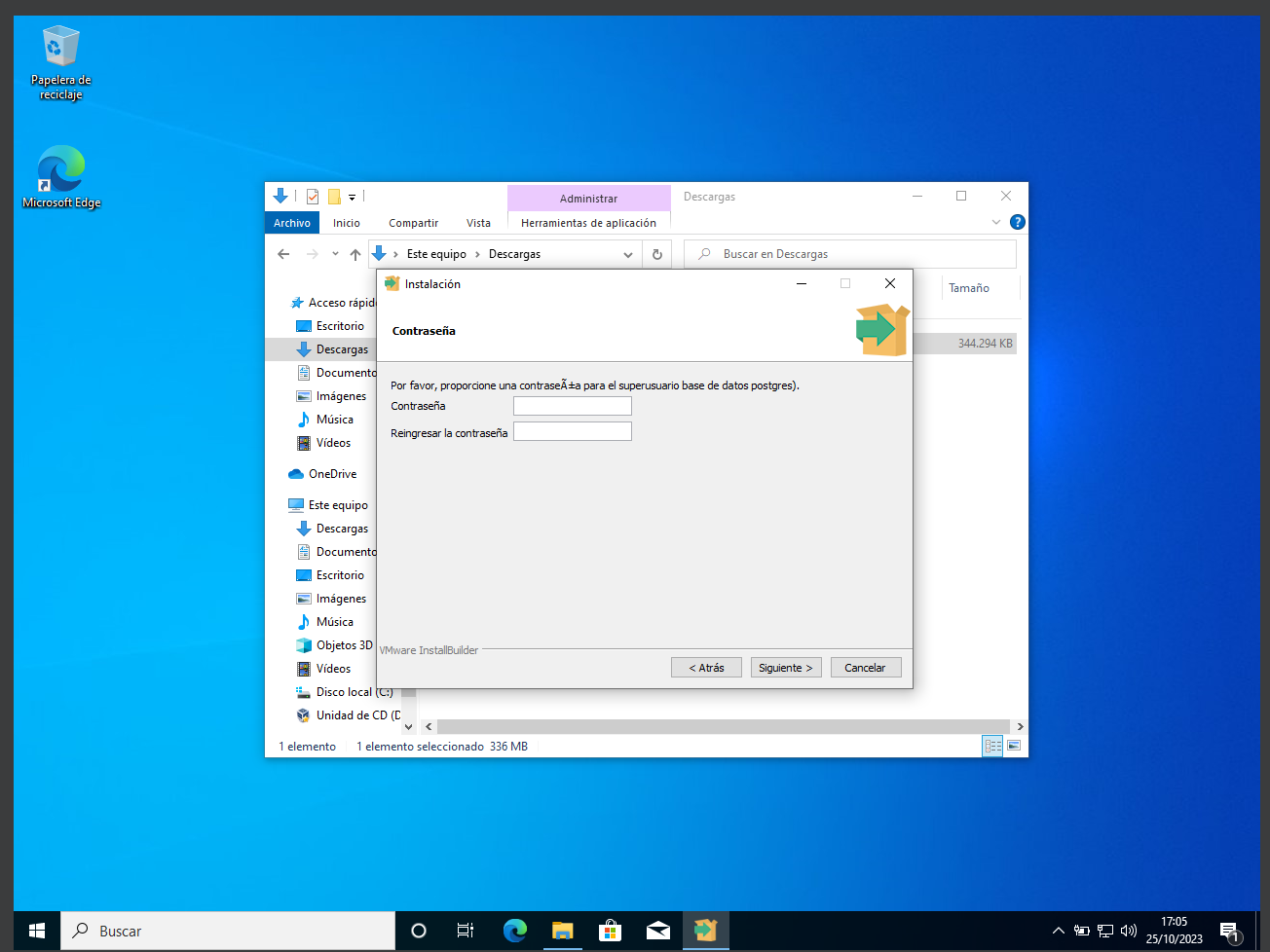
Descripción generada automáticamente

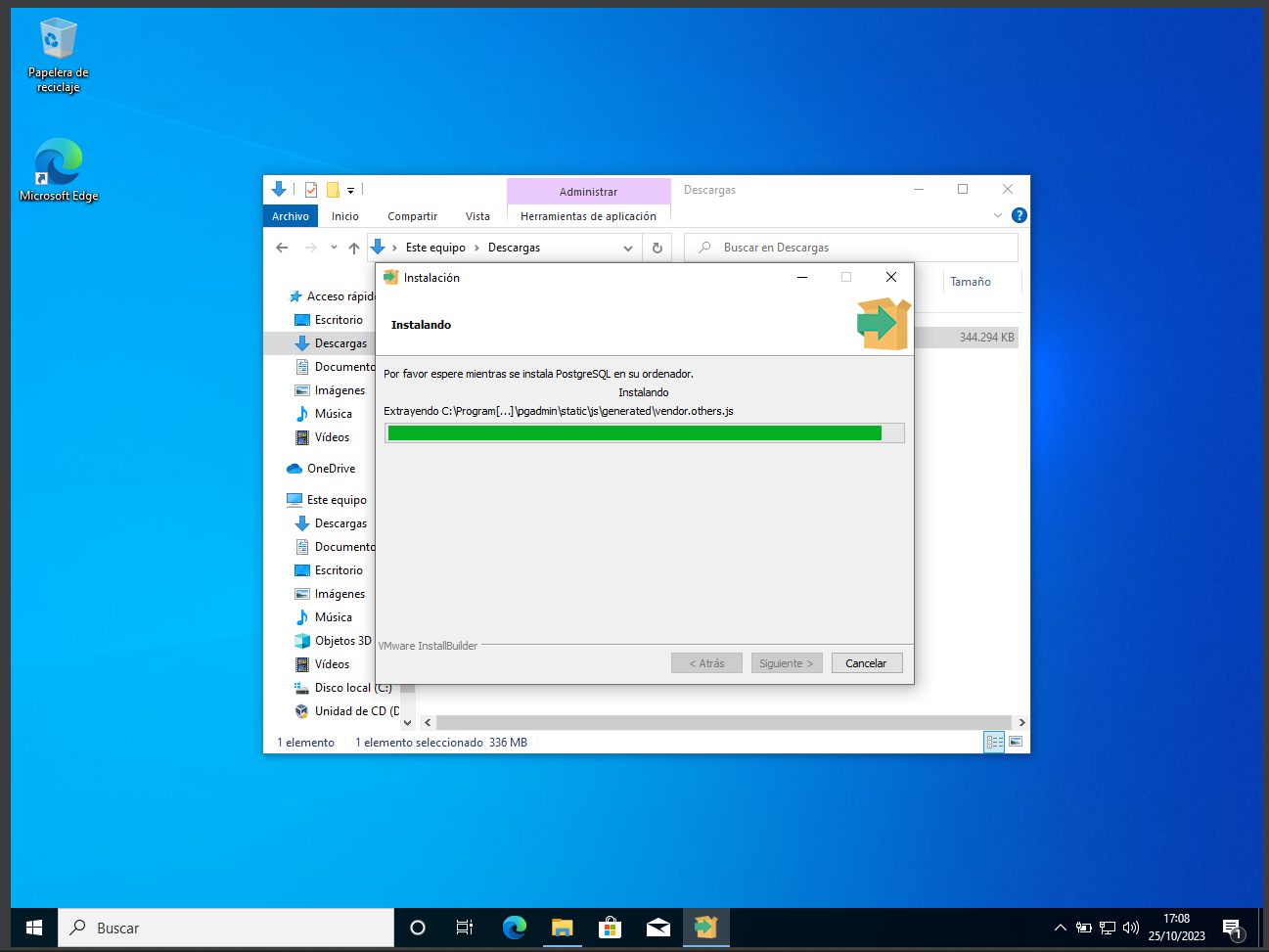
Una captura de pantalla de una computadora

Descripción generada automáticamente

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

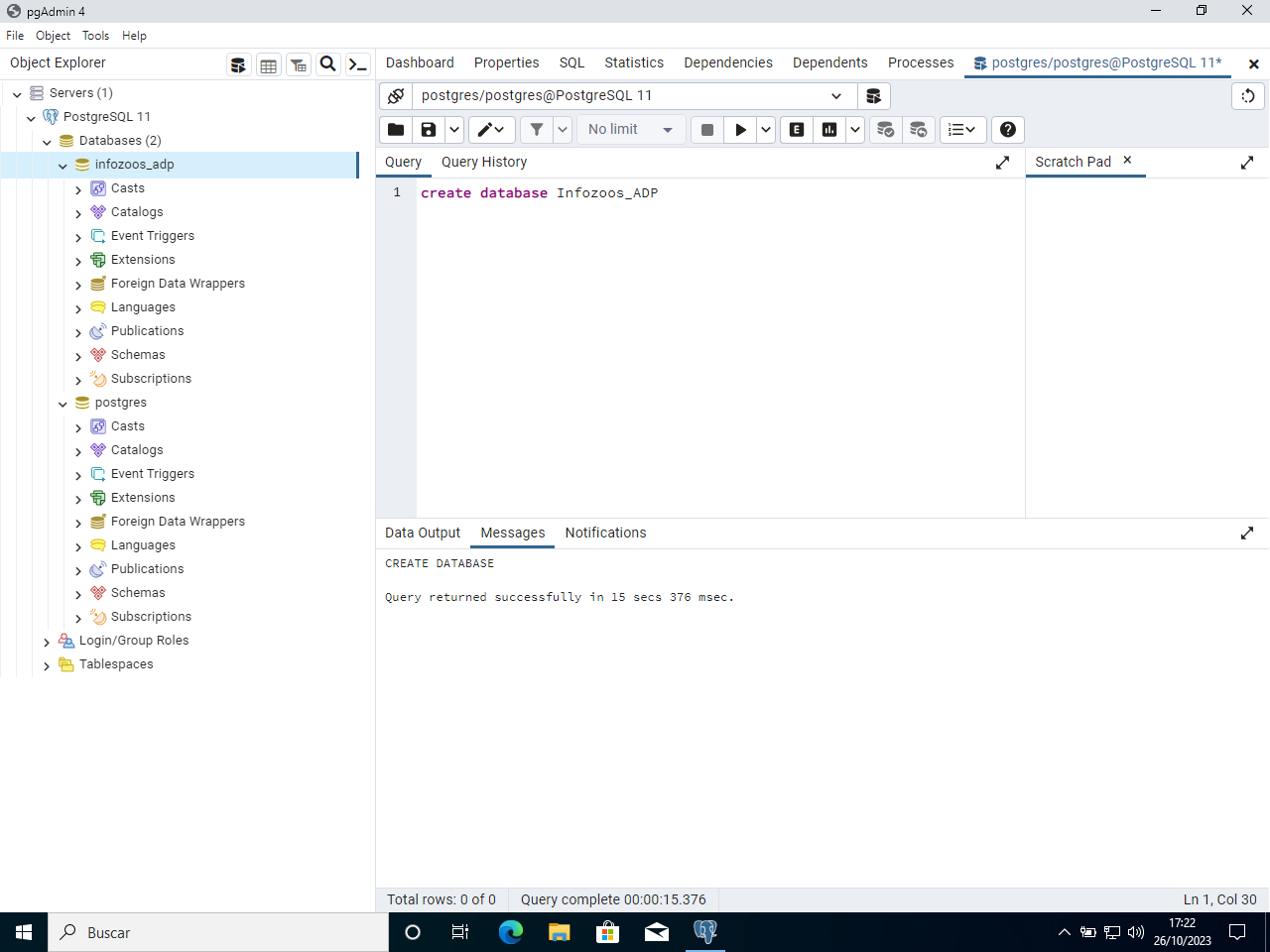
IMPORTANTE: configuración bien la contraseña del usuario root.





* **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)**



Sentencia SQL: Create database infozoos\_ADP;

* **Crear las siguientes tablas (2 puntos):**
  1. Interfaz de usuario gráfica, Texto, Aplicación

     Descripción generada automáticamente **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,

tamano double precision not null,

presupuestoAnual double precision not null

);

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

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

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

);

* 1. Interfaz de usuario gráfica, Texto, Aplicación

     Descripción generada automáticamente**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)

);

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

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteDatos en Zoo

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:

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

Datos Especie

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

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:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Sentencia SQL:

insert into animal(zooid,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):** 
  + 1. **Mostrar los animales que pertenecen a un determinado zoo.**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

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

* + 1. **Mostrar los zoos cuyo tamaño esté entre 15000 y 20000 m2.**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

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

* + 1. **Mostrar los animales que estén en peligro de extinción indicando, también a qué zoo pertenecen.**

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

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.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamenteComprobación:

* **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;

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteInterfaz de usuario gráfica

Descripción generada automáticamentePara restaurar se crea una base de datos nueva y restaurar con el archivo generado previamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* **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)**

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Sentencia SQL:

-- Crear Usuario

CREATE USER openpg\_AD with password '1234';

-- Dar Permisos Super Usuario

GRANT ALL PRIVILEGES ON DATABASE "Restauracion" TO openpg\_AD WITH GRANT OPTION