



**Facultad de Ingeniería y Ciencias
Aplicadas**

Ingeniería en Ciberseguridad

Administración de Base de Datos

PROYECTO INTEGRADOR

Daniel José Galiano Varela

Daniela Azucena Martínez Bucheli

Matheo Julian Oviedo Rodas

02 de Julio del 2024

Quito, Ecuador

I. OBJETIVOS

El trabajo por realizar tiene como finalidad implementar una solución de base de datos contenerizada que permita cubrir los siguientes puntos:

- Plantear el despliegue de una solución de base de datos contenerizada.
- Efectuar el despliegue de una solución de base de datos contenerizada.
- Validar el despliegue de una solución de base de datos contenerizada.
- Analizar los riesgos de seguridad a nivel privilegios que puede poseer una base de datos contenerizada.
- Plantear una alternativa de solución a los riesgos de seguridad a nivel privilegios que puede poseer una base de datos contenerizada, basándose en un enfoque RBA.
- Implementar una solución de seguridad basada en RBA para la operación segura de una base de datos contenerizada

II. INDICACIONES

En grupos de 3 o 4 estudiantes se deberán construir, como habilitador para poder rendir las evaluaciones del tercer progreso y eventualmente del examen de recuperación, una solución de base de datos contenerizada.

Para empezar, se deberá implementar la mencionada solución de base de datos contenerizada, basándose en las buenas prácticas vistas y aplicadas en clase:

- Archivos de configuración deberán residir en un almacenamiento persistente.
- Archivos de la base de datos deberán residir en un almacenamiento persistente
- No se deberá utilizar el puerto de conexión por default.
- Se deberá mapear el puerto de conexión de la instancia de base de datos hacia el host donde se ejecuta el contenedor.
- Se deberá utilizar un cliente gráfico SQL para acceder y manejar la instancia de base de datos contenerizada.

Posteriormente, se deberá diseñar e implementar en la base de datos contenerizada, un modelo de datos que permita solventar una necesidad de almacenamiento de datos planteada por el docente. Una vez implementado el modelo de datos, se deberá analizar los riesgos de seguridad que podría existir en el acceso a dichos datos, con el fin de



plantear e implementar una solución basada en un enfoque RBA, la cual permitirá que la base de datos contenerizada opere de manera segura, una vez más, en base a las buenas prácticas vistas y aplicadas en clase:

- Deberá existir un rol de base de datos con privilegios de solo lectura.
- Deberá existir al menos un rol de base de datos con privilegios de manipular los datos.
- Deberá existir un primer usuario de base de datos a quien se le deberá conceder el primer rol.
- Deberá existir un segundo usuario de base de datos a quien se le deberá conceder el segundo rol.
- Deberán generarse evidencias del correcto funcionamiento de la solución de seguridad basada en RBA.

III. DESCRIPCIÓN DE LA SOLUCIÓN

Una agencia de viajes necesita modernizar su infraestructura de TI para mejorar la eficiencia operativa y la gestión de datos. La solución actual presenta problemas de escalabilidad, seguridad y mantenimiento. La agencia ha decidido implementar una base de datos contenerizada utilizando PostgreSQL, que proporcionará una mayor flexibilidad, seguridad y capacidad de recuperación.

La agencia solicita que se implemente una solución de base de datos contenerizada, utilizando PostgreSQL en un entorno Docker, asegurando la persistencia de los datos y mejorando la gestión de la infraestructura. También, requiere que se mejore la seguridad y se gestionen privilegios a través de un enfoque basado en roles (RBA), garantizando que el acceso a los datos sea seguro y controlado, por último, requiere que se asegure la escalabilidad y disponibilidad, permitiendo que la infraestructura crezca con la demanda y que los servicios estén siempre disponibles.

Para esta solicitud, se va a:

- Utilizar Docker para desplegar y gestionar contenedores de PostgreSQL, proporcionando aislamiento y facilidad de despliegue. Y utilizar pgAdmin para la visualización de la base de datos.
- Se configurará la base de datos con la última versión de PostgreSQL para aprovechar las mejoras y características de seguridad. Los archivos de configuración y datos se almacenarán en volúmenes persistentes de Docker para asegurar la persistencia a largo plazo.



- En cuanto a la seguridad y gestión de accesos, se va a implementar roles con diferentes niveles de acceso (lectura y manipulación de datos) para garantizar un acceso seguro. Y se cambiará el puerto por default a uno personalizado para la conexión de la base de datos para evitar accesos no autorizados.

IV. PLANEACIÓN

1. Equipo de trabajo.

Para realizar el proceso de planeación, implementación y documentación del proyecto se procedió a formar un grupo que juntos trabaja de manera eficaz, sabiendo dividir de manera equitativa las actividades por realizar, y apoyar de manera activa en el desarrollo del proyecto. Las personas que integran el equipo de trabajo son:

- Daniela Martínez: estudiante de 8vo semestre de Ingeniería en Ciberseguridad. Con habilidades en uso de máquinas virtuales y diversas distribuciones de Linux.
- Matheo Oviedo: estudiante de 8vo semestre de Ingeniería en Ciberseguridad. Con habilidades en identificación de datos, procesos relacionados a generar diagramas lógicos y diccionarios de datos.
- Daniel Galiano: estudiante de 8vo semestre de Ingeniería en Ciberseguridad. Con habilidades de codificación básica en Python, necesaria para entendimiento y generación de códigos.

2. Objetivos de trabajo.

- a. Realizar un análisis de la empresa identificado las tablas y columnas que se implementarán en el modelo de la base datos.
- b. Investigar el proceso de instalación y buenas prácticas para el uso de bases de datos PostgreSQL contenerizadas.
- c. Configurar de manera adecuada la base de datos y llenar las tablas, documentando el proceso.

3. Cronograma de actividades.

El siguiente cronograma de actividades se propone para tener una mejor organización durante el desarrollo del proyecto, este puede cambiar con el paso del tiempo.

CRONOGRAMA DE ACTIVIDADES			
ACTIVIDADES	INICIO	FINALIZACIÓN	TOTAL
1. Realizar un análisis de la empresa identificado las tablas y columnas que se implementarán en el modelo de la base datos.	7-may-24	17-may-24	7 días
1.1. Reunirse con la dueña de la agencia de viajes, para identificar las áreas de esta.	7-may-24	9-may-24	2 días
1.2. Realizar un diagrama lógico de datos para identificar las tablas y columnas que se utilizarán en la base de datos	10-may-24	14-may-24	2 días (se excluyen fin de semana 11 y 12 de mayo)
1.3. Realizar un diccionario de datos en donde se identifiquen: identificadores primarios y secundarios, claves únicas y validaciones (constraints)	14-may-24	17-may-24	3 días
2. Investigar el proceso de instalación y buenas prácticas para el uso de bases de datos PostgreSQL contenerizadas.	20-may-24	7-jun-24	12 días
2.1. Investigar manuales de creación y manejo de bases de datos contenerizadas.	20-may-24	27-may-24	5 días (se excluye fin de semana 25 y 26 de mayo).
2.2. Revisar manuales de creación de contenedores para bases de datos PostgreSQL e identificar archivo de configuración de puertos.	27-may-24	30-may-24	3 días
2.3. Realizar diversas pruebas de instalación en varios sistemas operativos para identificar el más adecuado	3-jun-24	7-jun-24	4 días
3. Configurar de manera adecuada la base de datos y llenar las tablas, documentando el proceso.	10-jun-24	28-jun-24	12 días
3.1. Instalar, configurar la base de datos para la agencia.	10-jun-24	13-jun-24	3 días
3.2. Llenar la base de datos utilizando un código de python encargado de llenar las tablas de manera eficiente.	17-jun-24	21-jun-24	4 días
3.3. Documentar el proceso y subir este a un repositorio en GitHub.	24-jun-24	28-jun-24	4 días

4. Definición de roles.

Definición de roles
Líder del Proyecto.
Administrador de base de Datos



Especialista en seguridad de bases de datos.
Administrador de sistemas.
Desarrollador de bases de datos.

5. Asignación de roles.

Definición de roles	
Líder del Proyecto.	Daniela Martínez
Administrador de base de Datos	Daniela Martínez
Especialista en seguridad de bases de datos.	Daniel Galiano
Administrado de sistemas.	Daniel Galiano
Desarrollador de bases de datos.	Matheo Oviedo
Documentador	Matheo Oviedo

6. Asignación de responsabilidades.

Definición de roles	Asignación de responsabilidades	Asignación de roles
Líder de Proyecto	- Coordinar y supervisar todas las actividades del proyecto.	Daniela Martínez
	- Establecer metas y plazos, y asegurar su cumplimiento.	
	- Facilitar la comunicación y colaboración dentro del equipo.	
	- Resolver conflictos y tomar decisiones estratégicas.	
Administrador de Base de Datos	- Supervisar la instalación y configuración de la base de datos.	Daniela Martínez
	- Gestionar el almacenamiento persistente de los archivos de configuración y datos.	
	- Asegurar el rendimiento, disponibilidad y seguridad de la base de datos.	

Especialista en Seguridad de Bases de Datos	- Crear y gestionar usuarios y roles de acceso para la base de datos.	Daniel Galiano
	- Implementar medidas de seguridad para proteger la integridad y confidencialidad de los datos.	
	- Analizar y mitigar riesgos de seguridad asociados con el acceso a la base de datos.	
Administrador de Sistemas	- Desarrollar scripts y programas para automatizar procesos de gestión de la base de datos.	Daniel Galiano
	- Configurar y gestionar la máquina virtual y los contenedores donde se ejecutará la base de datos.	
	- Asegurar la conectividad y mapeo de puertos entre la instancia de base de datos y el host.	
Desarrollador de Base de Datos	- Diseñar y desarrollar el modelo lógico y físico de la base de datos.	Matheo Oviedo
	- Crear el diccionario de datos y la documentación técnica del diseño de la base de datos.	
	- Desarrollar el código necesario para la creación y validación de las bases de datos.	
Documentador	- Documentar todo el proceso de desarrollo e implementación de la solución.	Matheo Oviedo
	- Generar y mantener un repositorio en GitHub con la estructura de carpetas y la documentación necesaria para la solución.	
	- Crear informes y presentaciones que evidencien el cumplimiento de los objetivos y la correcta implementación de la solución.	

7. Cronograma de reuniones de trabajo.

Cronograma de actividades				
Fecha	Hora	Duración	Temas para tratar	Responsables
10/5/2024	10:00 a. m.	2 horas	Presentación inicial del proyecto, asignación de roles, y revisión de objetivos.	Todos los miembros del equipo
14/5/2024	10:00 a. m.	2 horas	Revisión del diseño de la base de datos y avance en la instalación de la base de datos contenerizada.	Líder del proyecto, Desarrollador de BD, Administrador de BD
21/5/2024	10:00 a. m.	2 horas	Implementación del modelo de datos y generación de datos utilizando Python y Faker.	Desarrollador de BD, Administrador de Sistemas
28/5/2024	10:00 a. m.	2 horas	Análisis de riesgos de seguridad y diseño de la solución basada en RBA.	Especialista en seguridad de BD
4/6/2024	10:00 a. m.	2 horas	Implementación de la solución de seguridad y pruebas de funcionamiento.	Especialista en seguridad de BD, Administrador de BD
11/6/2024	10:00 a. m.	2 horas	Revisión y documentación del proyecto, preparación del repositorio en GitHub.	Documentador
28/6/2024	10:00 a. m.	2 horas	Revisión final del proyecto, recopilación de evidencias de funcionamiento y ensayo de la presentación y defensa.	Todos los miembros del equipo
2/7/2024	18:50 p. m.	3 horas	Presentación y defensa del proyecto.	Todos los miembros del equipo

V. IMPLEMENTACIÓN DE LA SOLUCIÓN

Modelo de Datos:

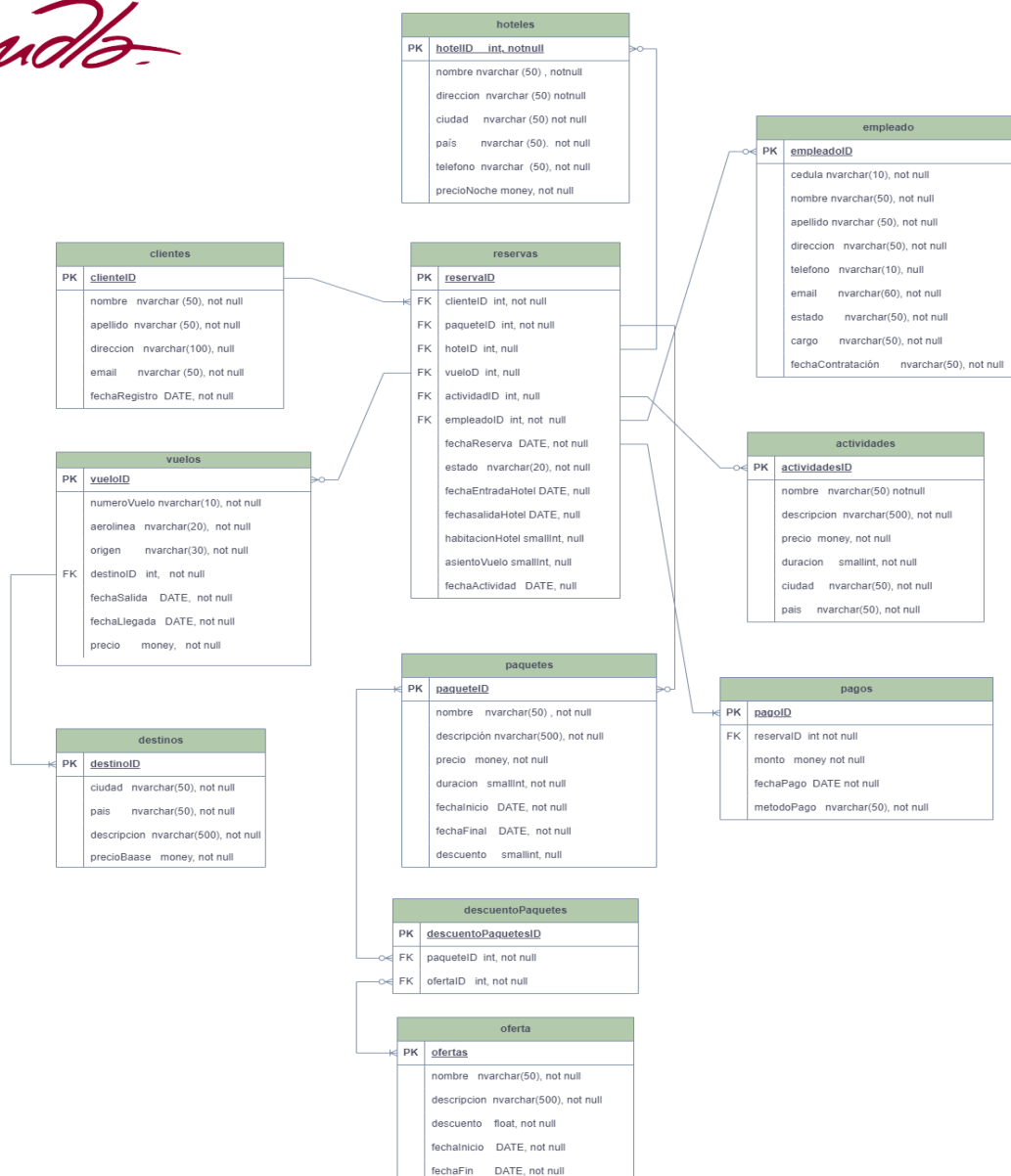
La base de datos incluirá las siguientes tablas:

- **hoteles:** Información de los hoteles asociados con la agencia.
- **destinos:** Datos de los destinos ofrecidos.



- **empleados:** Información del personal.
- **clientes:** Información de los clientes.
- **paquetes:** Paquetes de viajes ofrecidos.
- **ofertas:** Promociones y descuentos disponibles.
- **actividades:** Actividades turísticas en los destinos.
- **vuelos:** Detalles de los vuelos.
- **descuentosPaquetes:** Relación entre paquetes y ofertas.
- **pagos:** Información de los pagos realizados.
- **reservas:** Reservas hechas por los clientes, relacionando varias tablas

Diagrama lógico:





Diccionario de datos:

TABLAS		
NOMBRE	IDENTIFICADOR	DESCRIPCIÓN
clientes	clienteID	Tabla en donde se almacenan los datos personales de los clientes
hoteles	hotelID	Tabla en donde se almacena información sobre los hoteles que ofrece la agencia
empleados	empleadoID	Tabla en donde se almacena la información de todos los empleados de la agencia
actividades	actividadID	Tabla en donde se almacena la información de las actividades que los clientes pueden realizar.
Pagos	pagosID	Tabla en donde se almacena los pagos realizados
paquetes	paqueteID	Tabla en donde se almacena la información de los paquetes que ofrece la agencia.
Oferta	ofertaID	Tabla en donde se almacenan las ofertas temporales que ofrece la agencia.
descuentoPaquetes	descuentoPaquetesID	Tabla de conexión entre paquetes y ofertas en donde se almacena la o las ofertas que puede tener un paquete
destinos	destinoID	Tabla en donde se almacena los destinos turísticos que ofrece la agencia.





Vuelos	vueloID	Tabla en donde se almacena la información de los vuelos a los diversos destinos turisticos.
Reserva	reservaID	Tabla en donde se almacena la información de las reservas que el cliente realiza en la agencia, considerando vuelos, hoteles, actividades, paquetes, etc.

DETALLE TABLAS				
Tabla hoteles				
Nombre del Campo	Descripción	Tipo de Dato	Restricciones	NULL
hotelID	Identificador del hotel	nvarchar	UNIQUE	not null
nombre	Nombre del hotel	nvarchar	100 caracteres, UNIQUE	not null
direccion	Dirección del hotel	nvarchar	100 caracteres	not null
ciudad	Ciudad del hotel	nvarchar	50 caracteres	not null
pais	País del hotel	nvarchar	50 caracteres	not null
telefono	Teléfono del hotel	nvarchar	10 caracteres, UNIQUE	not null
precioNoche	Precio por noche	money		not null
Tabla empleados				
Nombre del Campo	Descripción	Tipo de Dato	Restricciones	NULL





empleadoID	Identificador del empleado	int	UNIQUE	not null
cedula	Cédula del empleado	int	10 caracteres, UNIQUE	not null
nombre	Nombre del empleado	nvarchar	50 caracteres	not null
apellido	Apellido del empleado	nvarchar	50 caracteres	not null
direccion	Dirección del empleado	nvarchar	100 caracteres	not null
telefono	Teléfono del empleado	nvarchar	10 caracteres, UNIQUE	not null
email	Email del empleado	nvarchar	100 caracteres, UNIQUE	not null
cargo	Cargo del empleado	nvarchar	50 caracteres	not null
fecha de contratación	Fecha de contratación	DATE		not null
Tabla pagos				
Nombre del Campo	Descripción	Tipo de Dato	Restricciones	NULL
pagosID	Identificador del pago	int	UNIQUE	not null
reservaID	Identificador de la reserva	int		not null
monto	Monto del pago	money		not null
fechaPago	Fecha del pago	nvarchar	50 caracteres	not null



metodoPago	Método de pago	nvarchar	50 caracteres	not null
Tabla clientes				
Nombre del Campo	Descripción	Tipo de Dato	Restricciones	NULL
clienteID	Identificador del cliente	int	Primary Key, UNIQUE	not null
cedula	Cédula del cliente	int	10 caracteres, UNIQUE	not null
nombre	Nombre del cliente	nvarchar	50 caracteres	not null
apellido	Apellido del cliente	nvarchar	50 caracteres	not null
direccion	Dirección del cliente	nvarchar	100 caracteres	not null
telefono	Teléfono del cliente	nvarchar	10 caracteres, UNIQUE	not null
email	Email del cliente	nvarchar	100 caracteres, UNIQUE	not null
fechaRegistro	Fecha de registro	DATE		not null
Tabla paquetes				
Nombre del Campo	Descripción	Tipo de Dato	Restricciones	NULL
paqueteID	Identificador del paquete	int	UNIQUE	not null
nombre	Nombre del paquete	nvarchar	100 caracteres, UNIQUE	not null



descripcion	Descripción del paquete	nvarchar	200 caracteres	not null
precio	Precio del paquete	money		not null
duracion	Duración del paquete	smallint		not null
fechaInicio	Fecha de inicio del paquete	DATE		not null
fechaFin	Fecha de fin del paquete	DATE		not null
Tabla descuentosPaquetes				
Nombre del Campo	Descripción	Tipo de Dato	Restricciones	NULL
descuentosPaquetesID	Identificador del descuento de paquetes	int	UNIQUE	not null
paqueteID	Identificador del paquete	int		not null
ofertaID	Identificador de la oferta	int		not null
Tabla ofertas				
Nombre del Campo	Descripción	Tipo de Dato	Restricciones	NULL
ofertaID	Identificador de la oferta	int	UNIQUE	not null
nombre	Nombre de la oferta	nvarchar	100 caracteres, UNIQUE	not null
descripcion	Descripción de la	nvarchar	200 caracteres	not null



	oferta			
descuento	Descuento ofrecido	decimal	(5,2)	not null
fechaInicio	Fecha de inicio de la oferta	DATE		not null
fechaFin	Fecha de fin de la oferta	DATE		not null
Tabla actividades				
Nombre del Campo	Descripción	Tipo de Dato	Restricciones	NULL
actividadID	Identificador de la actividad	int	UNIQUE	not null
nombre	Nombre de la actividad	nvarchar	100 caracteres, UNIQUE	not null
descripcion	Descripción de la actividad	nvarchar	200 caracteres	not null
precio	Precio de la actividad	money		not null
duracion	Duración de la actividad	tinyint		not null
ciudad	Ciudad de la actividad	nvarchar		not null
Tabla vuelos				
Nombre del Campo	Descripción	Tipo de Dato	Restricciones	NULL
vueloID	Identificador del vuelo	int	UNIQUE	not null



numeroVuelo	Número del vuelo	nvarchar	15 caracteres, UNIQUE	not null
aerolinea	Aerolínea	nvarchar	50 caracteres	not null
origen	Origen del vuelo	nvarchar	50 caracteres	not null
destinoID	Identificador del destino	int		not null
fechaSalida	Fecha de salida del vuelo	DATE		not null
fechaLlegada	Fecha de llegada del vuelo	DATE		not null
precio	Precio del vuelo	money		not null
Tabla destinos				
Nombre del Campo	Descripción	Tipo de Dato	Restricciones	NULL
destinoID	Identificador del destino	int	UNIQUE	not null
ciudad	Ciudad del destino	nvarchar	50 caracteres	not null
pais	País del destino	nvarchar	50 caracteres	not null
descripcion	Descripción del destino	nvarchar	200 caracteres	not null
precioBase	Precio base del destino	money		not null
Tabla reservas				
Nombre del Campo	Descripción	Tipo de Dato	Restricciones	NULL
reservaID	Identificador de la reserva	int	UNIQUE	not null



clienteID	Identificador del cliente	int		not null
paqueteID	Identificador del paquete	int		null
hotelID	Identificador del hotel	int		null
actividadID	Identificador de la actividad	int		null
empleadoID	Identificador del empleado	int		null
vueloID	Identificador del vuelo	int		null
fechaReserva	Fecha de la reserva	DATE		not null
estado	Estado de la reserva	nvarchar	50 caracteres	not null
fechaEntradaHotel	Fecha de entrada al hotel	DATE		NULL
fechaSalidaHotel	Fecha de salida del hotel	DATE		NULL
habitacionHotel	Habitación del hotel	smallint		NULL
asientoVuelo	Asiento del vuelo	smallint		NULL
fechaActividad	Fecha programada de la actividad	DATE		NULL





DETALLE CONSTRAINTS			
TABLA HOTELES			
NOMBRE	TIPO	APUNTA	DESCRIPCIÓN
pk_hotel	Primary Key	hotelID	Clave Primaria e identificador único de la tabla hoteles
uq_nombrehotel	Unique Key	nombre	Clave única de nombre del hotel.
uq_telefonoHotel	Unique Key	telefono	Clave unica del teléfono del hotel

DETALLE CONSTRAINTS			
TABLA Empleados			
NOMBRE	TIPO	APUNTA	DESCRIPCIÓN
pk_empleados	Primary Key	empleadID	Clave Primaria e identificador único de la tabla empleados
uq_cedulaempleado	Unique Key	cedula	Clave única de cédula de empelado
uq_telefonoempleado	Unique Key	telefono	Clave unica del teléfono del empelado
uq_emailempleado	Unique Key	email	Clave unica del emial del empelado

DETALLE CONSTRAINTS					
TABLA Pagos					
NOMBRE	TIPO	APUNTA	TABLA REFERENCIADA	COLUMNA REFERENCIADA	DESCRIPCIÓN



pk_Pagos	Primary Key	pagosID	N/A	N/A	Clave Primaria e identificador único de la tabla pagos
fk_Reservas	Foreign Key	reservaID	reservas	reservasID	Clave foránea que referencia a la tabla reservas a la columna reservasID

DETALLE CONSTRAINTS			
TABLA clientes			
NOMBRE	TIPO	APUNTA	DESCRIPCIÓN
pk_clientes	Primary Key	clienteID	Clave Primaria e identificador único de la tabla cliente
uq_cedulacliente	Unique Key	cedula	Clave única de cédula del cliente
uq_telefonocliente	Unique Key	telefono	Clave unica del teléfono del cliente
uq_emailcliente	Unique Key	email	Clave unica del emial del cliente

DETALLE CONSTRAINTS			
TABLA paquetes			
NOMBRE	TIPO	APUNTA	DESCRIPCIÓN
pk_paquete	Primary Key	paqueteID	Clave Primaria e identificador único de la tabla paquetes
uq_nombrePaquete	Unique Key	nombre	Clave única del nombre del paquete.



DETALLE CONSTRAINTS

TABLA DescuentosPaquetes

NOMBRE	TIPO	APUNTA	TABLA REFERENCIADA	COLUMNA REFERENCIADA	DESCRIPCIÓN
pk_descuentosPaquetes	Primary Key	descuentosPaquetesID	N/A	N/A	Clave Primaria e identificador único de la tabla descuentos paquetes
fk_paquete	Foreign Key	paqueteID	paquetes	paquetesID	Clave foránea que referencia a la tabla reservas a la columna reservasID
fk_oferta	Foreign Key	ofertaID	ofertas	ofertaID	Clave foránea que referencia a la tabla ofertas a la columna ofertaID



DETALLE CONSTRAINTS			
TABLA ofertas			
NOMBRE	TIPO	APUNTA	DESCRIPCIÓN
pk_oferta	Primary Key	ofertaID	Clave Primaria e identificador único de la tabla ofertas
uq_nombreOferta	Unique Key	nombre	Clave única del nombre de la oferta

DETALLE CONSTRAINTS			
TABLA actividades			
NOMBRE	TIPO	APUNTA	DESCRIPCIÓN
pk_actividad	Primary Key	actividadID	Clave Primaria e identificador único de la tabla actividad
uq_nombreOferta	Unique Key	nombre	Clave única del nombre de la oferta

DETALLE CONSTRAINTS			
TABLA vuelos			
NOMBRE	TIPO	APUNTA	DESCRIPCIÓN
pk_vuelo	Primary Key	vueloID	Clave Primaria e identificador único de la tabla vuelos
uq_numeroVuelo	Unique Key	numeroVuelo	Clave única del número del vuelo



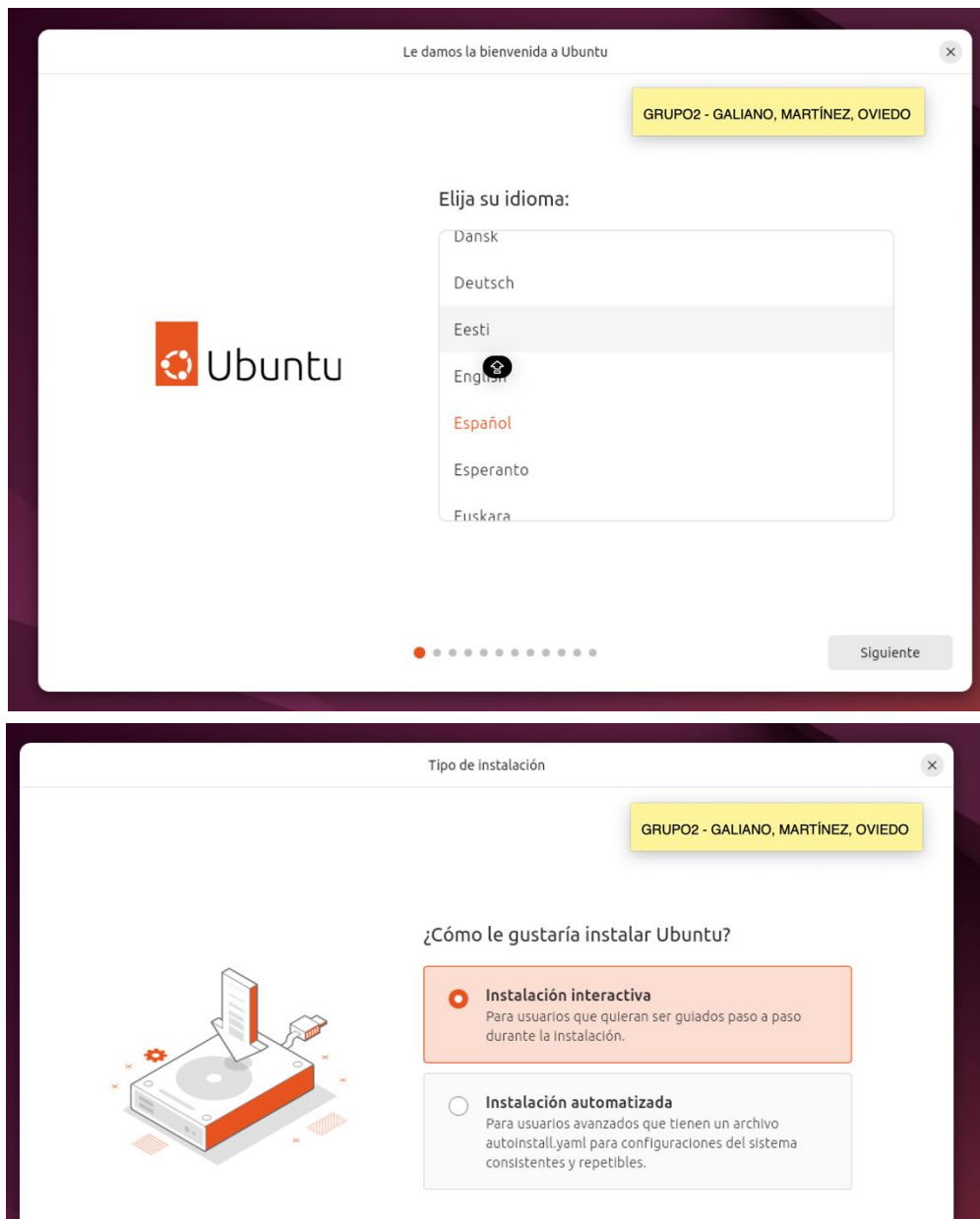
DETALLE CONSTRAINTS			
TABLA destinos			
NOMBRE	TIPO	APUNTA	DESCRIPCIÓN
pk_destino	Primary Key	destinoID	Clave Primaria e identificador único de la tabla destinos

DETALLE CONSTRAINTS					
TABLA reservas					
NOMBRE	TIPO	APUNTA	TABLA REFERENCIADA	COLUMNA REFERENCIADA	DESCRIPCIÓN
pk_reservas	Primary Key	reservaID	N/A	N/A	Clave Primaria e identificador único de la tabla reserva
fk_cliente	Foreign Key	paqueteID	paquetes	paquetesID	Clave foránea que referencia a la tabla reservas a la columna reservasID
fk_paquete	Foreign Key	ofertaID	ofertas	ofertaID	Clave foránea que referencia a la tabla ofertas a la columna ofertaID
fk_hotel	Foreign Key	ofertaID	ofertas	ofertaID	Clave foránea que referencia a la tabla ofertas a la columna ofertaID
fk_actividad	Foreign Key	actividadID	actividades	actividadID	Clave foránea que referencia a la tabla actividades a la columna actividadID
fk_empleado	Foreign Key	empleadoID	empleados	empleadoID	Clave foránea que referencia a la tabla empleados a la columna empleadoID
fk_vuelo	Foreign Key	vueloID	vuelos	vueloID	Clave foránea que referencia a la tabla vuelos a la columna vuelosID

Paso a paso:

1. Instalación de la máquina virtual:

Para este paso, vamos a instalar la máquina de Ubuntu, siguiendo el paso a paso de su instalador.



Aplicaciones

GRUPO2 - GALIANO, MARTÍNEZ, OVIEDO

¿Qué aplicaciones le gustaría instalar para comenzar?

☒ **Selección predeterminada**
Solo lo esencial, el navegador web y las utilidades básicas.

☐ **Selección ampliada**
Una selección de herramientas de oficina, utilidades y navegador web compatibles sin conexión.

Configuración del disco

GRUPO2 - GALIANO, MARTÍNEZ, OVIEDO

¿Cómo quieres instalar Ubuntu ?

☒ **Borrar disco e instalar Ubuntu**
Comience desde cero en el disco seleccionado.

Funcionalidades avanzadas...

Nada seleccionado

☐ **Instalación manual**
Para usuarios avanzados que buscan configuraciones de disco personalizadas.

Cree su cuenta

GRUPO2 - GALIANO, MARTÍNEZ, OVIEDO

Cree su cuenta

Su nombre

Grupo2

✓

El nombre del equipo

grupo2-VMware-Virtual-Platform

✓

Elija un nombre de usuario

grupo2

✓

Elija una contraseña

Mostrar

Contraseña aceptable

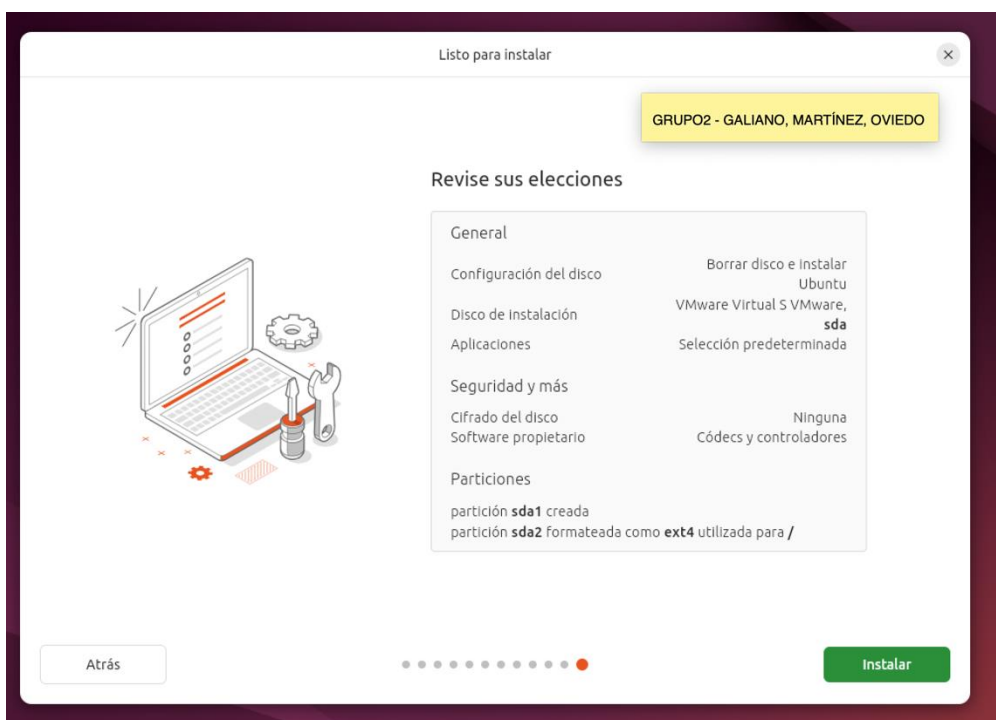
Confirme su contraseña

Mostrar

Contraseña aceptable

☒ Solicitar mi contraseña para acceder

☐ Utilizar Active Directory



2. Instalación de la base de datos contenerizada:

Una vez que se tenga instalada la máquina virtual, se procederá a configurar la base de datos contenerizada.

a. Primero actualizamos nuestro sistema.

```
grupo2@grupo2-VMware-Virtual-Platform:~/Escritorio$ sudo apt update
[sudo] contraseña para grupo2:
Warning: The unit file, source configuration file or drop-ins of apt-news.service
changed on disk. Run 'systemctl daemon-reload' to reload units.
Warning: The unit file, source configuration file or drop-ins of esm-cache.servi
ce changed on disk. Run 'systemctl daemon-reload' to reload units.
Obj:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Obj:2 http://ec.archive.ubuntu.com/ubuntu noble InRelease
Obj:3 http://ec.archive.ubuntu.com/ubuntu noble-updates InRelease
Obj:4 http://ec.archive.ubuntu.com/ubuntu noble-backports InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se pueden actualizar 62 paquetes. Ejecute «apt list --upgradable» para verlos.
```

b. Luego procedemos con la instalación de Docker.

```
grupo2@grupo2-VMware-Virtual-Platform:~/Escritorio$ sudo apt install apt-transpo
rt-https ca-certificates curl software-properties-common
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
ca-certificates ya está en su versión más reciente (20240203).
fijado ca-certificates como instalado manualmente.
software-properties-common ya está en su versión más reciente (0.99.48).
fijado software-properties-common como instalado manualmente.
Se instalarán los siguientes paquetes NUEVOS:
  apt-transport-https curl
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 62 no actualizados.
Se necesita descargar 230 kB de archivos.
Se utilizarán 568 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://ec.archive.ubuntu.com/ubuntu noble/universe amd64 apt-transport-htt
ps all 2.7.14build2 [3.974 B]
```

```
grupo2@grupo2-VMware-Virtual-Platform:~/Escritorio$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
grupo2@grupo2-VMware-Virtual-Platform:~/Escritorio$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"
Repositorio: «deb [arch=amd64] https://download.docker.com/linux/ubuntu noble stable»
Descripción:
Archive for codename: noble components: stable
Más información: https://download.docker.com/linux/ubuntu
Añadiendo repositorio.
Oprima [INTRO] para continuar o Ctrl+c para cancelar.
Añadiendo la entrada deb a /etc/apt/sources.list.d/archive_uri-https_download_docker_com_linux_ubuntu-noble.list

grupo2@grupo2-VMware-Virtual-Platform:~/Escritorio$ sudo apt update
Warning: The unit file, source configuration file or drop-ins of apt-news.service changed on disk. Run 'systemctl daemon-reload' to reload units.
Warning: The unit file, source configuration file or drop-ins of esm-cache.service changed on disk. Run 'systemctl daemon-reload' to reload units.
Obj:1 http://security.ubuntu.com/ubuntu noble-security InRelease
Obj:2 https://download.docker.com/linux/ubuntu noble InRelease
Obj:3 http://ec.archive.ubuntu.com/ubuntu noble InRelease
Obj:4 http://ec.archive.ubuntu.com/ubuntu noble-updates InRelease
Obj:5 http://ec.archive.ubuntu.com/ubuntu noble-backports InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se pueden actualizar 63 paquetes. Ejecute «apt list --upgradable» para verlos.
W: https://download.docker.com/linux/ubuntu/dists/noble/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
grupo2@grupo2-VMware-Virtual-Platform:~/Escritorio$ sudo apt install docker-ce
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras
```

- c. Se verifica que su instalación haya sido correcta.

```
grupo2@grupo2-VMware-Virtual-Platform:~/Escritorio$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: en>
   Active: active (running) since Tue 2024-07-02 10:11:23 -05; 19s ago
 TriggeredBy: ● docker.socket
   Docs: https://docs.docker.com
  Main PID: 6549 (dockerd)
    Tasks: 10
   Memory: 22.0M (peak: 22.4M)
      CPU: 863ms
   CGroup: /system.slice/docker.service
           └─6549 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/cont>

jul 02 10:11:22 grupo2-VMware-Virtual-Platform systemd[1]: Starting docker.serv>
jul 02 10:11:22 grupo2-VMware-Virtual-Platform dockerd[6549]: time="2024-07-02T>
jul 02 10:11:22 grupo2-VMware-Virtual-Platform dockerd[6549]: time="2024-07-02T>
jul 02 10:11:22 grupo2-VMware-Virtual-Platform dockerd[6549]: time="2024-07-02T>
jul 02 10:11:23 grupo2-VMware-Virtual-Platform dockerd[6549]: time="2024-07-02T>
jul 02 10:11:23 grupo2-VMware-Virtual-Platform dockerd[6549]: time="2024-07-02T>
jul 02 10:11:23 grupo2-VMware-Virtual-Platform dockerd[6549]: time="2024-07-02T>
jul 02 10:11:23 grupo2-VMware-Virtual-Platform dockerd[6549]: time="2024-07-02T>
jul 02 10:11:23 grupo2-VMware-Virtual-Platform systemd[1]: Started docker.servi>
lines 1-21/21 (END)
```



- d. Ahora vamos a implementar la última versión de PostgreSQL en un contenedor Docker y creamos un volumen para el almacenamiento persistente de Postgres en el escritorio.

```
grupo2@grupo2-VMware-Virtual-Platform:~/Escritorio$ sudo docker pull postgres:latest
latest: Pulling from library/postgres
f11c1adaa26e: Pull complete
76ce212b9153: Pull complete
919ca406a058: Pull complete
6b7a1245fe71: Pull complete
8064ffe06c65: Pull complete
4b5c59f2d82c: Pull complete
fe72764b9070: Pull complete
6ef8e2c0f4d9: Pull complete
e71fe9d7ff11: Pull complete
f3225d69190d: Pull complete
2bf90d17afc8: Pull complete
d3aee49eb079: Pull complete
e1e856658919: Pull complete
95c2c2ef9f02: Pull complete
Digest: sha256:6d2a810d24786a90bf4bf2f4d4ce7a1039696a9269f3293abf3d6c8aa0bab9da
Status: Downloaded newer image for postgres:latest
docker.io/library/postgres:latest
grupo2@grupo2-VMware-Virtual-Platform:~/Escritorio$ sudo docker volume create pgdata
pgdata
```

GRUPO2 - GALIANO, MARTÍNEZ, OVIEDO

- e. Ahora, iniciaremos un contenedor “ejemplo” con la imagen de postgres, de aquí copiaremos su configuración.

```
grupo2@grupo2-VMware-Virtual-Platform:~/Escritorio$ sudo docker run --name proyectorfinal -e POSTGRES_PASSWORD=proyectorfinal -d -p 5433:5432 -v pgdata:/var/lib/postgresql/data --restart always postgres
7811b4c47795c2fbad1694460fc0d5501ba01db20e16f2b879e239cf068e8398
grupo2@grupo2-VMware-Virtual-Platform:~/Escritorio$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS
PORTS         NAMES
7811b4c47795   postgres "docker-entrypoint.s..." 20 seconds ago Up 19 seconds
0.0.0.0:5433->5432/tcp, :::5433->5432/tcp proyectorfinal
grupo2@grupo2-VMware-Virtual-Platform:~/Escritorio$
```

GRUPO2 - GALIANO, MARTÍNEZ, OVIEDO

- f. Ahora, crearemos una carpeta para nuestros archivos llamada “proyectorfinal”, aquí copiaremos la configuración actual del contenedor de postgres ya creado.

```
grupo2@grupo2-VMware-Virtual-Platform:~/Escritorio$ sudo docker cp proyectorfinal:/var/lib/postgresql/data/postgresql.conf ./postgresql.conf
Successfully copied 31.7kB to /home/grupo2/Escritorio/postgresql.conf
grupo2@grupo2-VMware-Virtual-Platform:~/Escritorio$ mkdir proyectorfinal
grupo2@grupo2-VMware-Virtual-Platform:~/Escritorio$ cd proyectorfinal/
grupo2@grupo2-VMware-Virtual-Platform:~/Escritorio/proyectorfinal$ sudo docker cp proyectorfinal:/var/lib/postgresql/data/postgresql.conf ./postgresql.conf
Successfully copied 31.7kB to /home/grupo2/Escritorio/proyectorfinal/postgresql.conf
grupo2@grupo2-VMware-Virtual-Platform:~/Escritorio/proyectorfinal$ ls
postgresql.conf
grupo2@grupo2-VMware-Virtual-Platform:~/Escritorio/proyectorfinal$
```

GRUPO2 - GALIANO, MARTÍNEZ, OVIEDO

- g. Ahora, cambiaremos el puerto por default en la configuración que copiamos.

```

grupo2@grupo2-VMware-Virtual-Platform: ~/Escritorio/proyectofinal
GNU nano 7.2 postgresql.conf *
#external_pid_file = ''                                # write an extra PID file
                                                         # (change requires restart)

#-----
# CONNECTIONS AND AUTHENTICATION
#-----

# - Connection Settings -

listen_addresses = '*'                                # comma-separated list of addresses;
                                                         # defaults to 'localhost'; use '*' for
                                                         # (change requires restart)
port = 5555                                           # (change requires restart)
max_connections = 100                                # (change requires restart)
#reserved_connections = 0                            # (change requires restart)
#superuser_reserved_connections = 3                  # (change requires restart)
#unix_socket_directories = '/var/run/postgresql'     # comma-separated list of dire
                                                         # (change requires restart)

^G Ayuda      ^O Guardar    ^W Buscar     ^K Cortar     ^T Ejecutar   ^C Ubicación
^X Salir      ^R Leer fich. ^\ Reemplazar ^U Pegar      ^J Justificar ^_ Ir a línea

```

- h. Luego, vamos a crear un Dockerfile que use una configuración personalizada para PostgreSQL.

```

grupo2@grupo2-VMware-Virtual-Platform:~/Escritorio/proyectofinal$ cat <<EOL > Dockerfile
FROM postgres
COPY postgresql.conf /etc/postgresql/postgresql.conf
RUN chown postgres:postgres /etc/postgresql/postgresql.conf
CMD ["postgres", "-c", "config_file=/etc/postgresql/postgresql.conf"]
EOL
grupo2@grupo2-VMware-Virtual-Platform:~/Escritorio/proyectofinal$ ls
Dockerfile postgresql.conf
grupo2@grupo2-VMware-Virtual-Platform:~/Escritorio/proyectofinal$ sudo docker build -t mi_postgres .
[+] Building 0.7s (8/8) FINISHED
=> [internal] load build definition from Dockerfile                                docker:default 0.0s
=> => transferring dockerfile: 236B                                              0.0s
=> [internal] load metadata for docker.io/library/postgres:latest                0.0s
=> [internal] load .dockerignore                                                  0.0s
=> => transferring context: 2B                                                    0.0s
=> [internal] load build context                                                  0.0s
=> => transferring context: 29.81kB                                              0.0s
=> [1/3] FROM docker.io/library/postgres:latest                                0.1s
=> [2/3] COPY postgresql.conf /etc/postgresql/postgresql.conf                  0.0s
=> [3/3] RUN chown postgres:postgres /etc/postgresql/postgresql.conf            0.4s
=> exporting to image                                                            0.1s
=> => exporting layers                                                            0.0s
=> => writing image sha256:fd3ed839ae38d8698ec8371f4566b8821336722a1ddca      0.0s
=> => naming to docker.io/library/mi_postgres                                  0.0s
grupo2@grupo2-VMware-Virtual-Platform:~/Escritorio/proyectofinal$

```

- i. Ahora, paramos y borramos el contenedor de “ejemplo”.

```

grupo2@grupo2-VMware-Virtual-Platform:~/Escritorio/proyectofinal$ sudo docker stop proyecto-final
grupo2@grupo2-VMware-Virtual-Platform:~/Escritorio/proyectofinal$ sudo docker rm proyecto-final

```

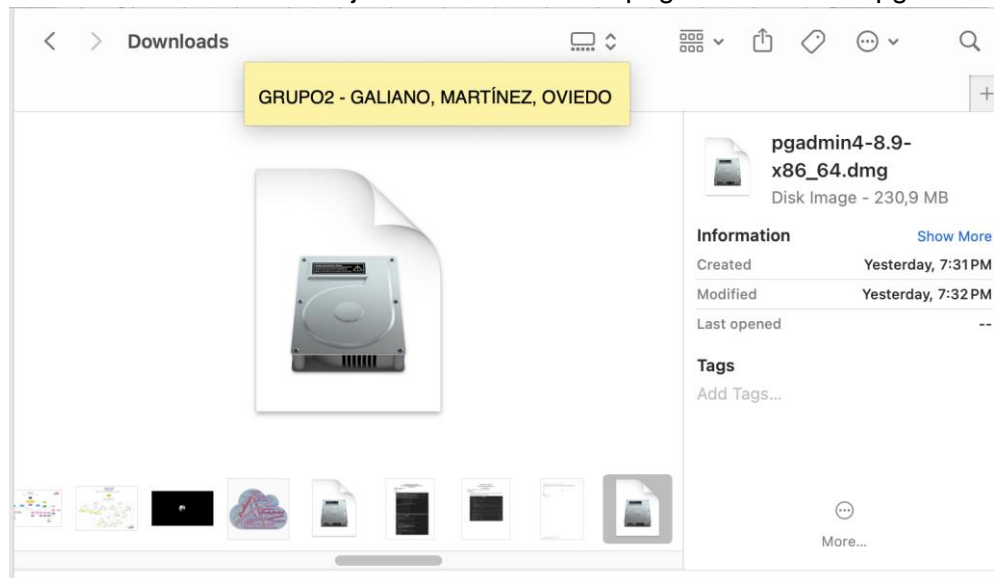
- j. Con esto listo, procedemos a crear el contenedor de la base de datos. Como se cambió anteriormente, postgres estará en el puerto 5555 y se mapeará al puerto 2222 para realizar la conexión en pgAdmin.


```
grupo2@grupo2-VMware-Virtual-Platform:~/Escritorio/proyectoFinal$ sudo docker run --name proyecto-final -e POSTGRES_PASSWORD=proyectoFinal -d -p 2222:5555 -v ~/Escritorio/pgdata:/var/lib/postgresql/data --restart always mi_postgres
grupo2@grupo2-VMware-Virtual-Platform:~/Escritorio/proyectoFinal$ sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
85969eb2d9cb	mi_postgres	"docker-entrypoint.s..."	4 seconds ago	Up 3 seconds	5432/tcp, 0.0.0.0:2222->5555/tcp, :2222->5555/tcp

3. Instalación del cliente gráfico SQL:

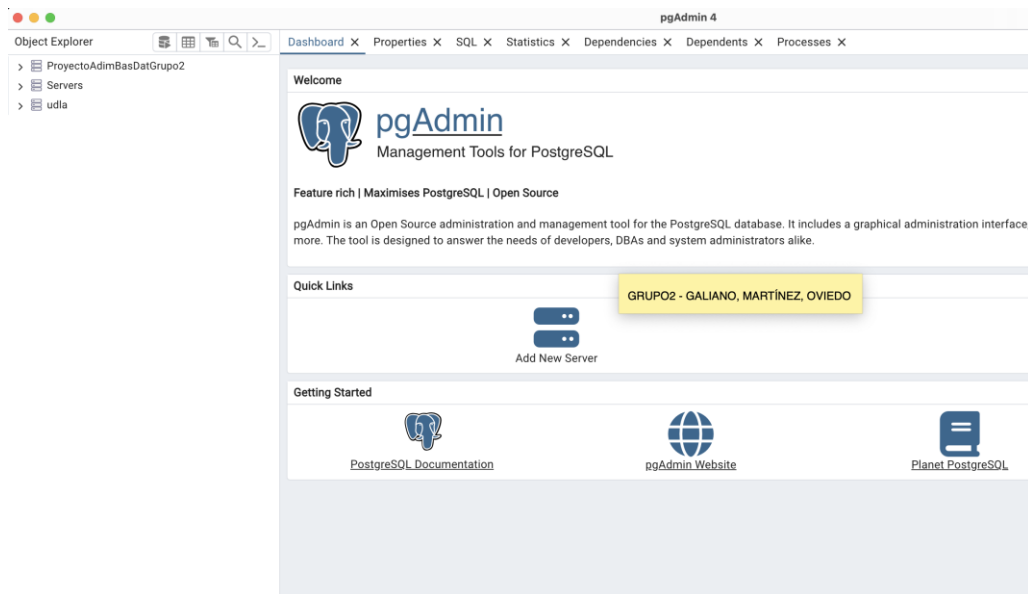
- a. Primero obtenemos el ejecutable desde la página oficial de pgAdmin.



- b. En el sistema operativo MacOs, solo basta con arrastrar la aplicación instalada hacia nuestra carpeta de aplicaciones.

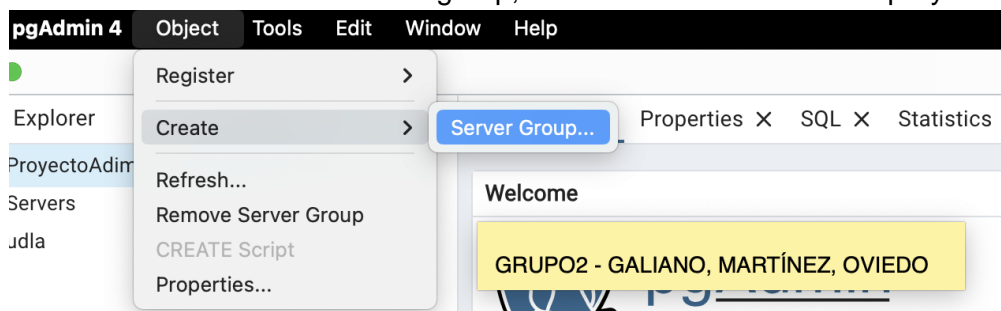


- c. Finalmente, tendremos pgAdmin.

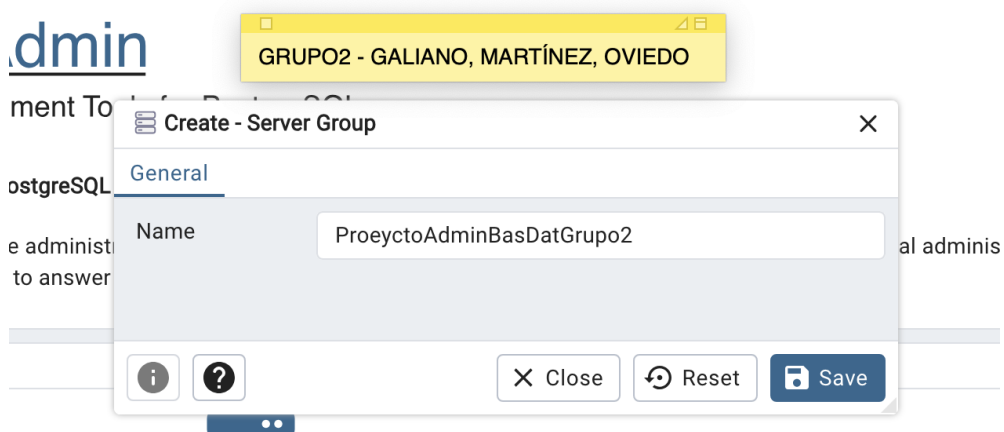


4. Conexión de la base de datos contenerizada con el cliente gráfico:

- a. Primero vamos a crear un servergroup, totalmente dedicado a este proyecto.

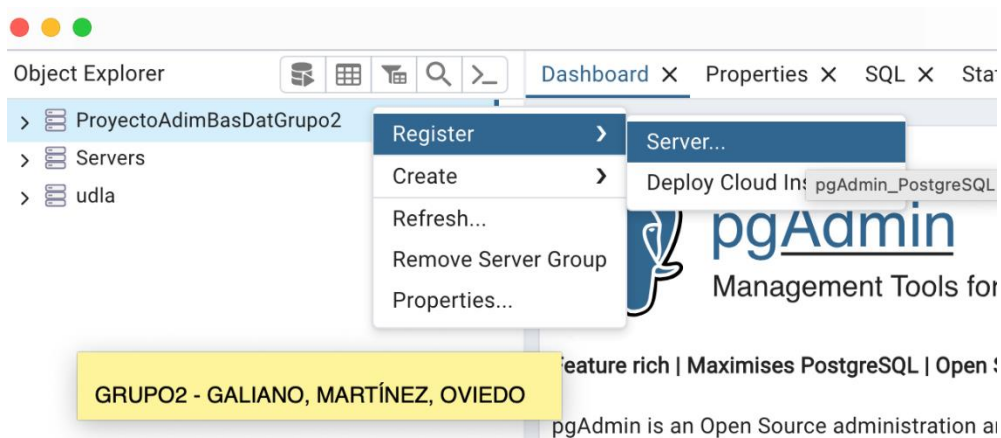


- b. Ahora, le colocaremos el nombre y crearemos.

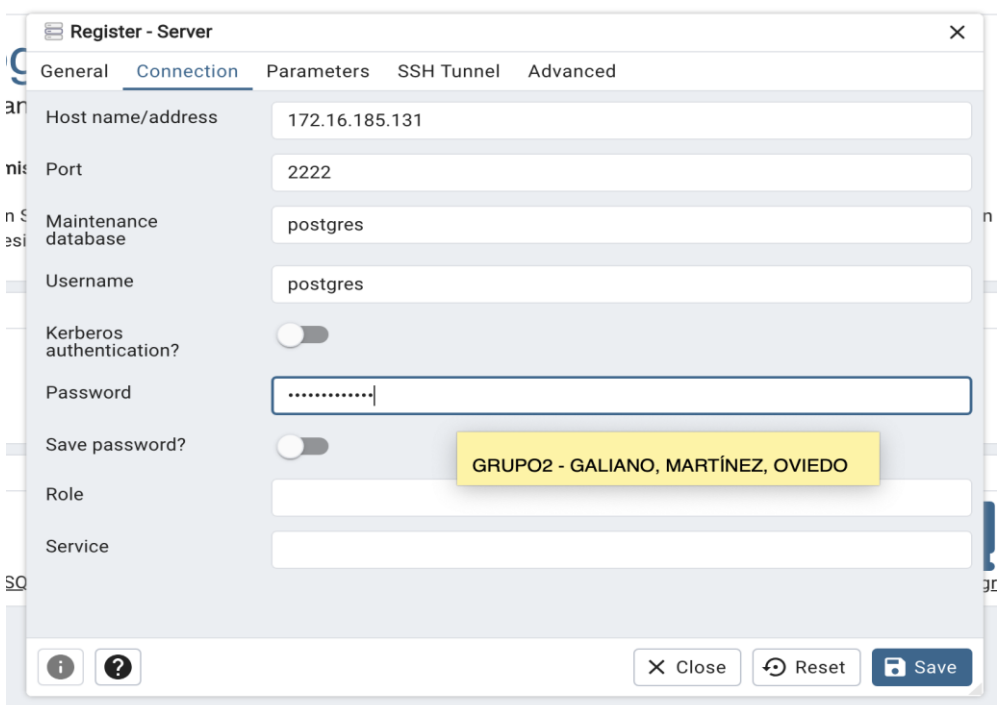
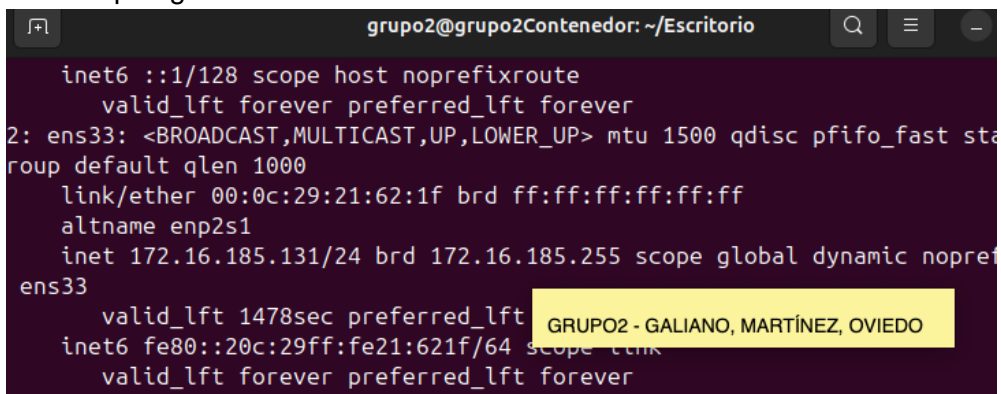




- c. Luego, vamos a registrar el servidor, aquí es cuando se realiza la conexión con nuestro contenedor.

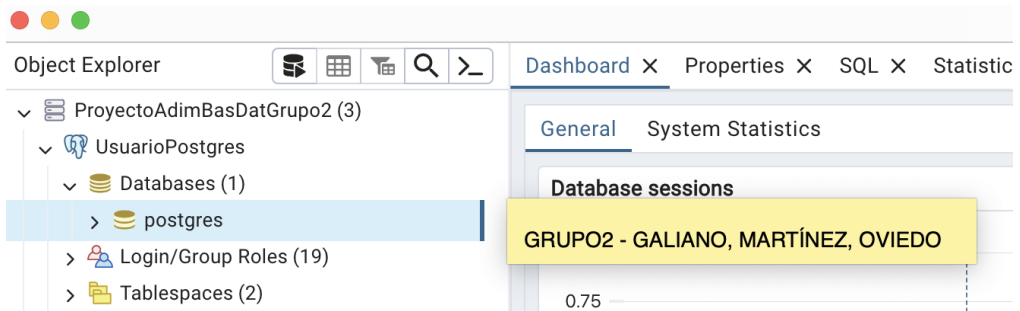


- d. Conociendo la ip de nuestra máquina virtual donde se encuentra alojado el contenedor, colocamos los datos que se nos solicita para crear la conexión. Recordando que mapeamos el puerto 5555 al 2222 y ese será por el cuál podamos ingresar. También debemos saber que, por defecto, al crear el contenedor con postgres, se va a crear una base de datos y un usuario de nombre postgres.



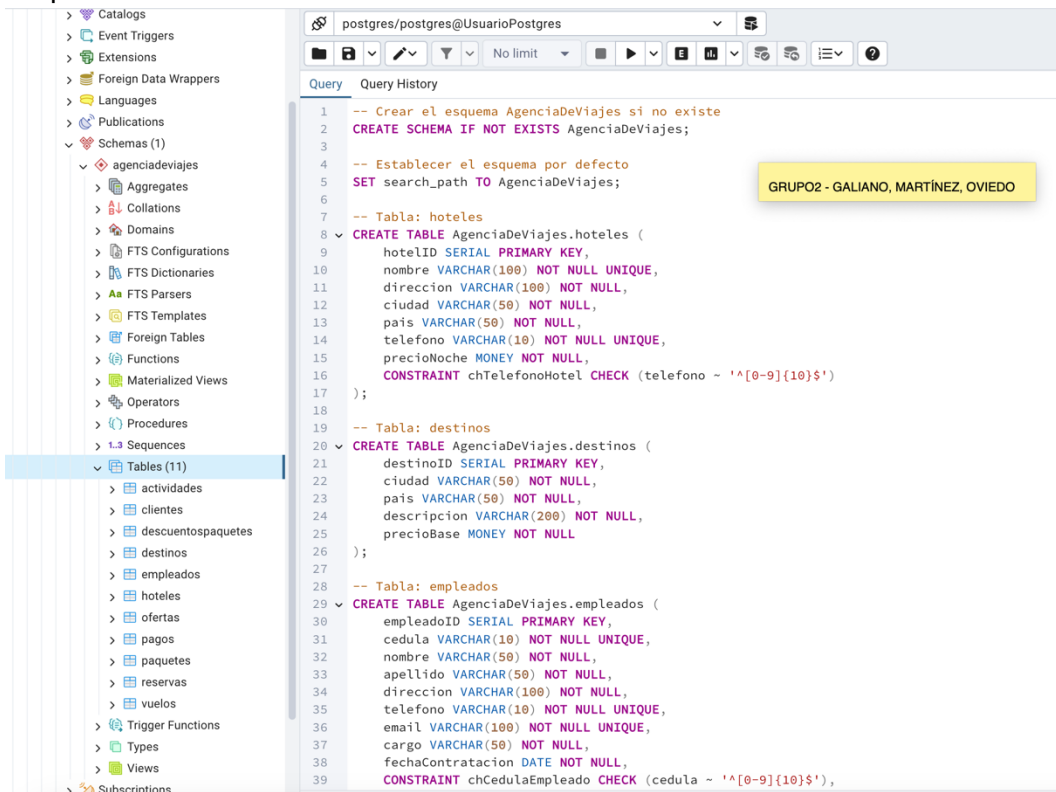


- e. Una vez dentro, podremos observar que tenemos la base de datos dentro de nuestra conexión.



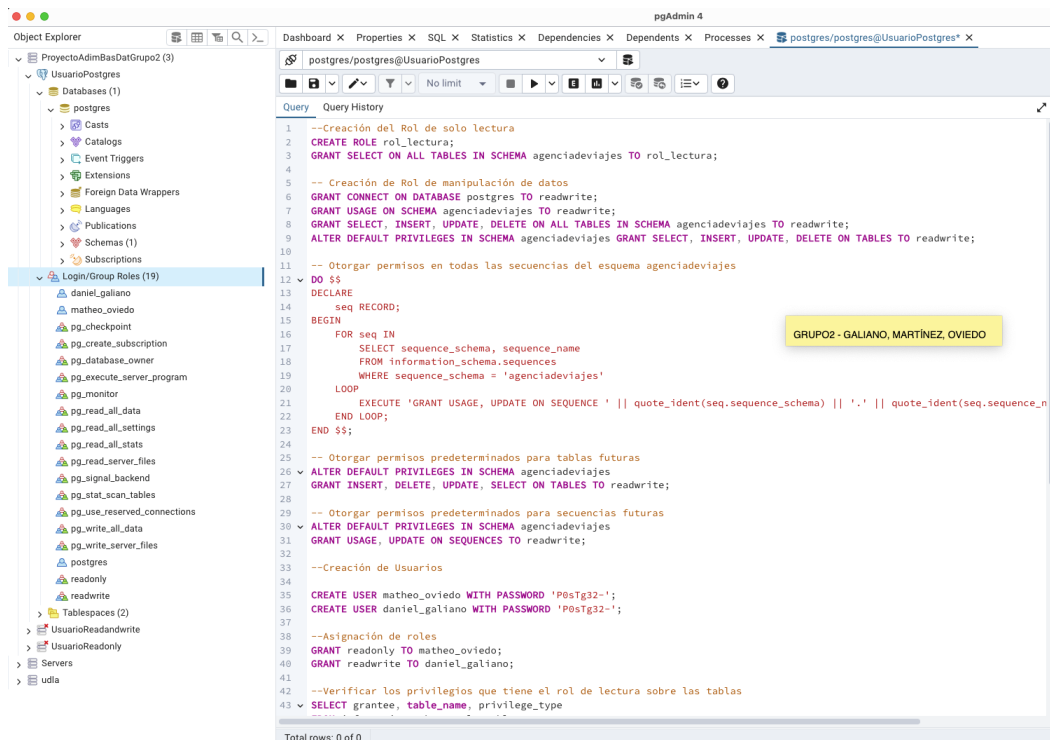
5. Creación de tablas:

Para crear las tablas, utilizaremos el lenguaje sql. A continuación se puede visualizar que el script fue ejecutado y que las tablas ahora se encuentran dentro de un esquema llamado “agenciadeviajes” que también fue creado mediante el script.



6. Creación de roles y usuarios:

Para crear los roles y los usuarios, también utilizaremos lenguaje sql que se ejecutará desde el query de pgAdmin.



```

--Creación del Rol de solo lectura
1 CREATE ROLE rol_lectura;
2 GRANT SELECT ON ALL TABLES IN SCHEMA agenciadeviajes TO rol_lectura;
3
4
5 -- Creación de Rol de manipulación de datos
6 GRANT CONNECT ON DATABASE postgres TO readwrite;
7 GRANT USAGE ON SCHEMA agenciadeviajes TO readwrite;
8 GRANT SELECT, INSERT, UPDATE, DELETE ON ALL TABLES IN SCHEMA agenciadeviajes TO readwrite;
9 ALTER DEFAULT PRIVILEGES IN SCHEMA agenciadeviajes GRANT SELECT, INSERT, UPDATE, DELETE ON TABLES TO readwrite;
10
11 -- Otorgar permisos en todas las secuencias del esquema agenciadeviajes
12 DO $$
13 DECLARE
14     seq RECORD;
15 BEGIN
16     FOR seq IN
17         SELECT sequence_schema, sequence_name
18         FROM information_schema.sequences
19         WHERE sequence_schema = 'agenciadeviajes'
20     LOOP
21         EXECUTE 'GRANT USAGE, UPDATE ON SEQUENCE ' || quote_ident(seq.sequence_schema) || '.' || quote_ident(seq.sequence_name) || ' TO readwrite;';
22     END LOOP;
23 END $$;
24
25 -- Otorgar permisos predeterminados para tablas futuras
26 ALTER DEFAULT PRIVILEGES IN SCHEMA agenciadeviajes
27 GRANT INSERT, DELETE, UPDATE, SELECT ON TABLES TO readwrite;
28
29 -- Otorgar permisos predeterminados para secuencias futuras
30 ALTER DEFAULT PRIVILEGES IN SCHEMA agenciadeviajes
31 GRANT USAGE, UPDATE ON SEQUENCES TO readwrite;
32
33 --Creación de Usuarios
34
35 CREATE USER matheo_oviedo WITH PASSWORD 'P0sTg32-';
36 CREATE USER daniel_galiano WITH PASSWORD 'P0sTg32-';
37
38 --Asignación de roles
39 GRANT readonly TO matheo_oviedo;
40 GRANT readwrite TO daniel_galiano;
41
42 --Verificar los privilegios que tiene el rol de lectura sobre las tablas
43 SELECT grantee, table_name, privilege_type

```

7. Llenado de datos:

- Lo primero que vamos a realizar es crear un entorno virtual dentro de Ubuntu, es para poder gestionar paquetes de Python para proyectos, debido a que el sistema está configurado para evitar cambios en la instalación de Python que puedan afectar los paquetes del sistema u otras aplicaciones de Python. Esto no afectará para nada el resultado de el llenado de datos.

```

grupo2@grupo2Contenedor:~/Escritorio/proyectofinal$ sudo apt install python3-venv
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
python3-venv ya está en su versión más reciente (3.12.3-0ubuntu1).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 71 no actualizados.
grupo2@grupo2Contenedor:~/Escritorio/proyectofinal$ python3 -m venv env
grupo2@grupo2Contenedor:~/Escritorio/proyectofinal$ source env/bin/activate
(env) grupo2@grupo2Contenedor:~/Escritorio/proyectofinal$

```

- Ahora instalaremos faker y algunas dependencias para poder ejecutar nuestro script de llenado.

```


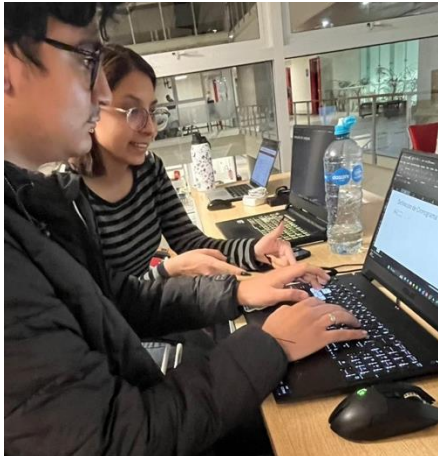
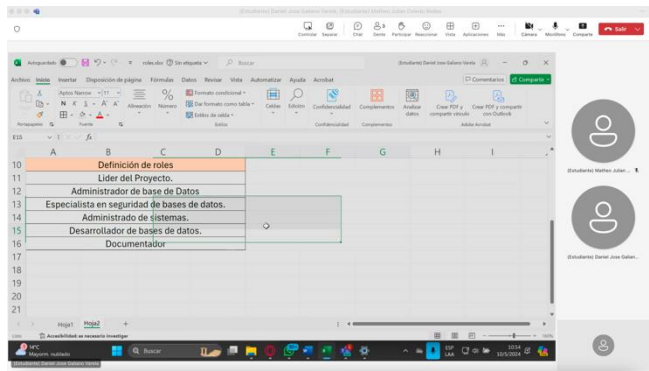
(env) grupo2@grupo2Contenedor:~/Escritorio/proyectofinal$ pip install faker
Requirement already satisfied: faker in ./env/lib/python3.12/site-packages (25.9.2)
Requirement already satisfied: python-dateutil>=2.4 in ./env/lib/python3.12/site-packages (from faker) (2.9.0.post0)
Requirement already satisfied: six>=1.5 in ./env/lib/python3.12/site-packages (from python-dateutil>=2.4->faker) (1.16.0)
(env) grupo2@grupo2Contenedor:~/Escritorio/proyectofinal$ sudo apt install libpq-dev python3-dev
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
libpq-dev ya está en su versión más reciente (16.3-0ubuntu0.24.04.1).
python3-dev ya está en su versión más reciente (3.12.3-0ubuntu1).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 71 no actualizados.
(env) grupo2@grupo2Contenedor:~/Escritorio/proyectofinal$ pip install psycopg2
Collecting psycopg2
  Using cached psycopg2-2.9.9-cp312-cp312-linux_x86_64.whl
Installing collected packages: psycopg2
Successfully installed psycopg2-2.9.9

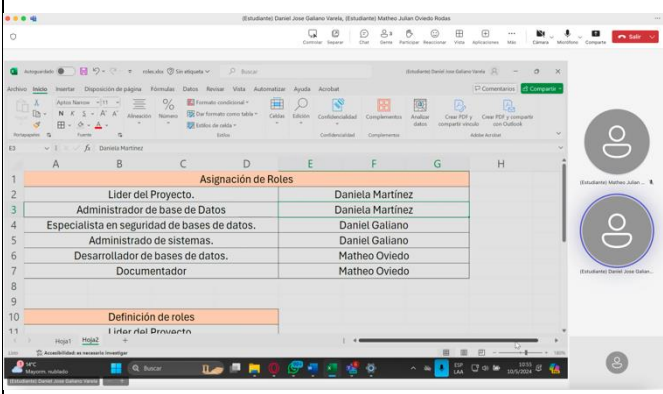
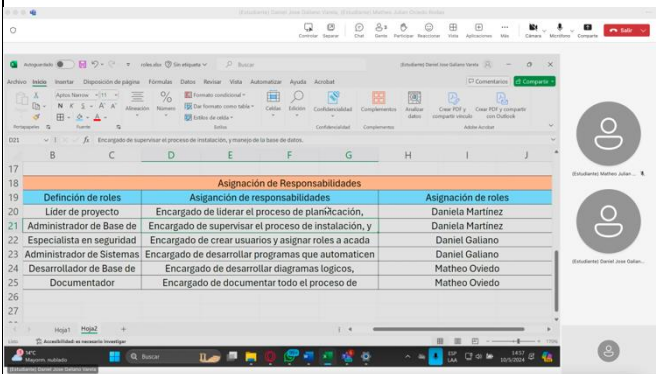
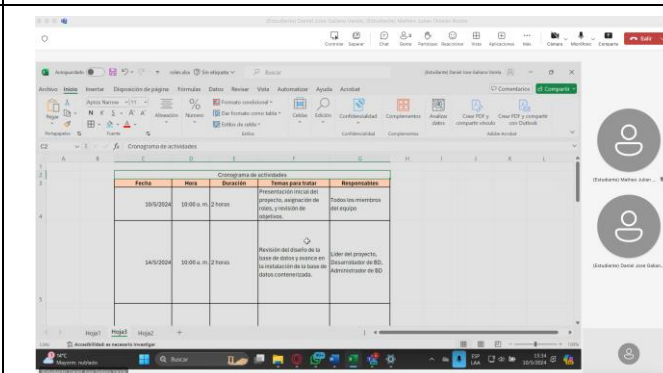
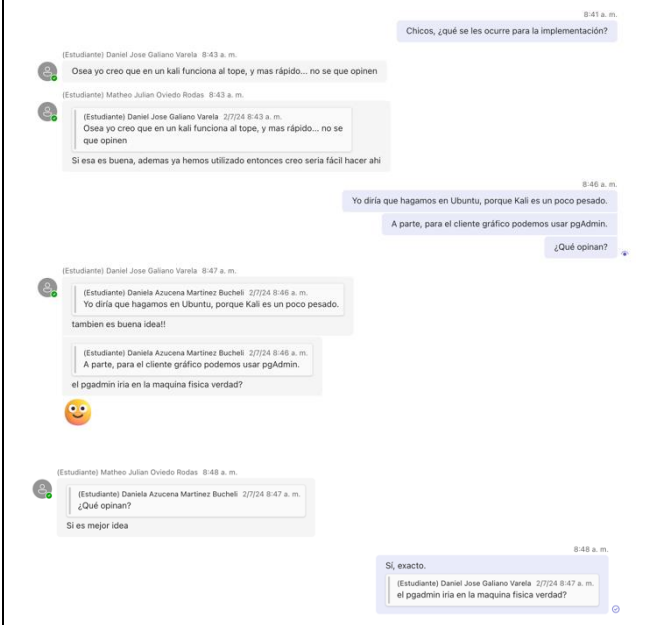
```


VI. INSTRUCCIONES PARA ACCEDER AL REPOSITORIO DE GITHUB

1. Abrir github (<https://github.com/>) e iniciar sesión.
2. Acceder al repositorio creado para la asignación de Administración de Base de Datos (<https://github.com/Matheo2702/Proyecto-Integrador-de-Administraci-n-de-Base-de-Datos>)
3. Leer el README para entender los documentos adjuntos.
4. En base a lo especificado comprender el uso de cada documento.

VII. CUMPLIMIENTO DE LOS CRITERIOS

Criterio	Porcentaje de Cumplimiento	Evidencia(s)
Definición de los Objetivos de Trabajo	100%	
Definición de Cronograma	100%	
Definición de Roles	100%	

Asignación de Roles	100%	 <table><tr><th colspan="2">Asignación de Roles</th></tr><tr><td>Lider del Proyecto.</td><td>Daniela Martinez</td></tr><tr><td>Administrador de base de Datos</td><td>Daniela Martinez</td></tr><tr><td>Especialista en seguridad de bases de datos.</td><td>Daniel Galiano</td></tr><tr><td>Administrado de sistemas.</td><td>Daniel Galiano</td></tr><tr><td>Desarrollador de bases de datos.</td><td>Matheo Oviedo</td></tr><tr><td>Documentador</td><td>Matheo Oviedo</td></tr></table>	Asignación de Roles		Lider del Proyecto.	Daniela Martinez	Administrador de base de Datos	Daniela Martinez	Especialista en seguridad de bases de datos.	Daniel Galiano	Administrado de sistemas.	Daniel Galiano	Desarrollador de bases de datos.	Matheo Oviedo	Documentador	Matheo Oviedo										
Asignación de Roles																										
Lider del Proyecto.	Daniela Martinez																									
Administrador de base de Datos	Daniela Martinez																									
Especialista en seguridad de bases de datos.	Daniel Galiano																									
Administrado de sistemas.	Daniel Galiano																									
Desarrollador de bases de datos.	Matheo Oviedo																									
Documentador	Matheo Oviedo																									
Asignación de Responsabilidades	100%	 <table><tr><th colspan="3">Asignación de Responsabilidades</th></tr><tr><td>Definición de roles</td><td>Asignación de responsabilidades</td><td>Asignación de roles</td></tr><tr><td>Lider de proyecto</td><td>Encargado de liderar el proceso de planificación,</td><td>Daniela Martinez</td></tr><tr><td>Administrador de Base de</td><td>Encargado de supervisar el proceso de instalación, y</td><td>Daniela Martinez</td></tr><tr><td>Especialista en seguridad</td><td>Encargado de crear usuarios y asignar roles a cada</td><td>Daniel Galiano</td></tr><tr><td>Administrador de Sistemas</td><td>Encargado de desarrollar programas que automaticen</td><td>Daniel Galiano</td></tr><tr><td>Desarrollador de Base de</td><td>Encargado de desarrollar diagramas logicos,</td><td>Matheo Oviedo</td></tr><tr><td>Documentador</td><td>Encargado de documentar todo el proceso de</td><td>Matheo Oviedo</td></tr></table>	Asignación de Responsabilidades			Definición de roles	Asignación de responsabilidades	Asignación de roles	Lider de proyecto	Encargado de liderar el proceso de planificación,	Daniela Martinez	Administrador de Base de	Encargado de supervisar el proceso de instalación, y	Daniela Martinez	Especialista en seguridad	Encargado de crear usuarios y asignar roles a cada	Daniel Galiano	Administrador de Sistemas	Encargado de desarrollar programas que automaticen	Daniel Galiano	Desarrollador de Base de	Encargado de desarrollar diagramas logicos,	Matheo Oviedo	Documentador	Encargado de documentar todo el proceso de	Matheo Oviedo
Asignación de Responsabilidades																										
Definición de roles	Asignación de responsabilidades	Asignación de roles																								
Lider de proyecto	Encargado de liderar el proceso de planificación,	Daniela Martinez																								
Administrador de Base de	Encargado de supervisar el proceso de instalación, y	Daniela Martinez																								
Especialista en seguridad	Encargado de crear usuarios y asignar roles a cada	Daniel Galiano																								
Administrador de Sistemas	Encargado de desarrollar programas que automaticen	Daniel Galiano																								
Desarrollador de Base de	Encargado de desarrollar diagramas logicos,	Matheo Oviedo																								
Documentador	Encargado de documentar todo el proceso de	Matheo Oviedo																								
Cronograma de reuniones de trabajo	100%	 <table><tr><th colspan="5">Cronograma de actividades</th></tr><tr><th>Fecha</th><th>Hora</th><th>Duración</th><th>Tareas para hacer</th><th>Responsables</th></tr><tr><td>18/03/2024</td><td>10:00 a.m.</td><td>2 horas</td><td>Presentación inicial del proyecto, asignación de roles, y revisión de objetivos.</td><td>Profesores asistentes del equipo</td></tr><tr><td>24/03/2024</td><td>10:00 a.m.</td><td>2 horas</td><td>Revisión del diseño de la base de datos y avance en la instalación de la base de datos contenida.</td><td>Lider del proyecto, Documentador de BD, Administrador de BD</td></tr></table>	Cronograma de actividades					Fecha	Hora	Duración	Tareas para hacer	Responsables	18/03/2024	10:00 a.m.	2 horas	Presentación inicial del proyecto, asignación de roles, y revisión de objetivos.	Profesores asistentes del equipo	24/03/2024	10:00 a.m.	2 horas	Revisión del diseño de la base de datos y avance en la instalación de la base de datos contenida.	Lider del proyecto, Documentador de BD, Administrador de BD				
Cronograma de actividades																										
Fecha	Hora	Duración	Tareas para hacer	Responsables																						
18/03/2024	10:00 a.m.	2 horas	Presentación inicial del proyecto, asignación de roles, y revisión de objetivos.	Profesores asistentes del equipo																						
24/03/2024	10:00 a.m.	2 horas	Revisión del diseño de la base de datos y avance en la instalación de la base de datos contenida.	Lider del proyecto, Documentador de BD, Administrador de BD																						
Ideas aportadas para la implementación de la solución por cada participante	100%	 <p>Chicos, ¿qué se les ocurre para la implementación?</p> <p>(Estudiante) Daniel Jose Galiano Varela 8:43 a. m. Osea yo creo que en un kali funciona al tope, y mas rápido... no se que opinen</p> <p>(Estudiante) Matheo Julian Oviedo Rodas 8:43 a. m. (Estudiante) Daniel Jose Galiano Varela 2/7/24 8:43 a. m. Osea yo creo que en un kali funciona al tope, y mas rápido... no se que opinen Si esa es buena, ademas ya hemos utilizado entonces creo sería fácil hacer ahi</p> <p>Yo diria que hagamos en Ubuntu, porque Kali es un poco pesado. A parte, para el cliente gráfico podemos usar pgAdmin. ¿Qué opinan?</p> <p>(Estudiante) Daniel Jose Galiano Varela 8:47 a. m. (Estudiante) Daniela Azucena Martinez Bucheli 2/7/24 8:46 a. m. Yo diria que hagamos en Ubuntu, porque Kali es un poco pesado. tambien es buena idea!</p> <p>(Estudiante) Daniela Azucena Martinez Bucheli 2/7/24 8:46 a. m. A parte, para el cliente gráfico podemos usar pgAdmin. el pgadmin iria en la maquina fisica verdad?</p> <p>(Estudiante) Matheo Julian Oviedo Rodas 8:48 a. m. (Estudiante) Daniela Azucena Martinez Bucheli 2/7/24 8:47 a. m. ¿Qué opinan? Si es mejor idea</p> <p>Si, exacto. (Estudiante) Daniel Jose Galiano Varela 2/7/24 8:47 a. m. el pgadmin iria en la maquina fisica verdad?</p>																								

Aporte individual para la consecución de los Objetivos de Trabajo	100%	
Aporte individual para la resolución de posibles conflictos	100%	

VII. CONCLUSIONES

La implementación del proyecto fue exitosa, pudiendo implementar PostgreSQL en Docker para tener una base de datos de una agencia de viajes bien definida y estructurada.

Adicionalmente se tuvo una mejora en la seguridad del proyecto aplicando lo aprendido en clase sobre control de accesos, teniendo roles definidos para que los usuarios tengan el menor privilegio y así se minimiza la exposición de información.

Por último, se tuvo la implementación gráfica para poder visualizar los datos de mejor manera con pgAdmin para facilitar el manejo de la base de datos.

El proyecto nos permitió consolidar y aplicar nuestros conocimientos técnicos, además nos enseñó la importancia de la planificación, la colaboración en equipo y la adopción de buenas prácticas en la implementación de soluciones tecnológicas.

IX. RECOMENDACIONES

En el proyecto propuesto en la materia de Administración de base de datos se pusieron en práctica varios conocimientos adquiridos a lo largo no solo de la materia sino de la carrera, como es el uso de sistemas operativos Linux, conectividad entre máquinas virtuales mediante puertos, configuración de bases de datos, entre otros. Por lo que se pudieron identificar varias recomendaciones para futuros trabajos.



1. Identificar de mejor manera los datos que se colocarán en la base de datos, permitiendo que esta sea optima al momento de realizar consultas.
2. Investigar los tipos de datos aceptados en cada DBMS en este caso PostgreSQL no acepta el tipo de datos TinyInt mientras que en SQL server es un tipo de dato muy común.
3. Realizar diversas pruebas de conectividad, esto se debe al cambio de puertos y al mapeo de este, para hacer más segura a la base de datos.
4. Realizar un código eficiente que además de que tenga las líneas comentadas realice inserciones de valores según las validaciones de cada tabla.
5. Identificar posibles problemas de permisos dentro de la máquina virtual.
6. Instalar las librerías necesarias para poder utilizar de manera el código desarrollado en Python.