

Proyecto Final

**DAVID ESTRELLA
YORDHAN GUERRÓN
HENRY LAZO**

A red handwritten signature, possibly reading 'u0/a', is centered on the page.

17/01/2023

—

ADM. BASES DE DATOS

—

ING. LUIS MORENO

Introducción:

En el presente proyecto vamos a realizar el proceso de simular una base de datos de una cuenta bancaria contenerizada, para esto debemos aplicar los conocimientos adquiridos durante el semestre y aplicarlos en un ejemplo práctico

Objetivo:

Diseñar e implementar una solución de base de datos contenerizada, así como el análisis de su verificación y efectividad basándonos en las mejores practicas y vistas aplicadas durante las clases. De igual manera se tiene como objetivo analizar los riesgos de seguridad que podría existir en el acceso de los datos asegurando y planteando una eficaz solución basada en un enfoque RDA.

Planteamiento del caso:

Se pretende crear una base de datos de una Banca en la que su creación sea mediante PostgreSQL y la administración sea mediante un proceso de contenerización de la base en un Docker desde una máquina virtual con SO Linux.

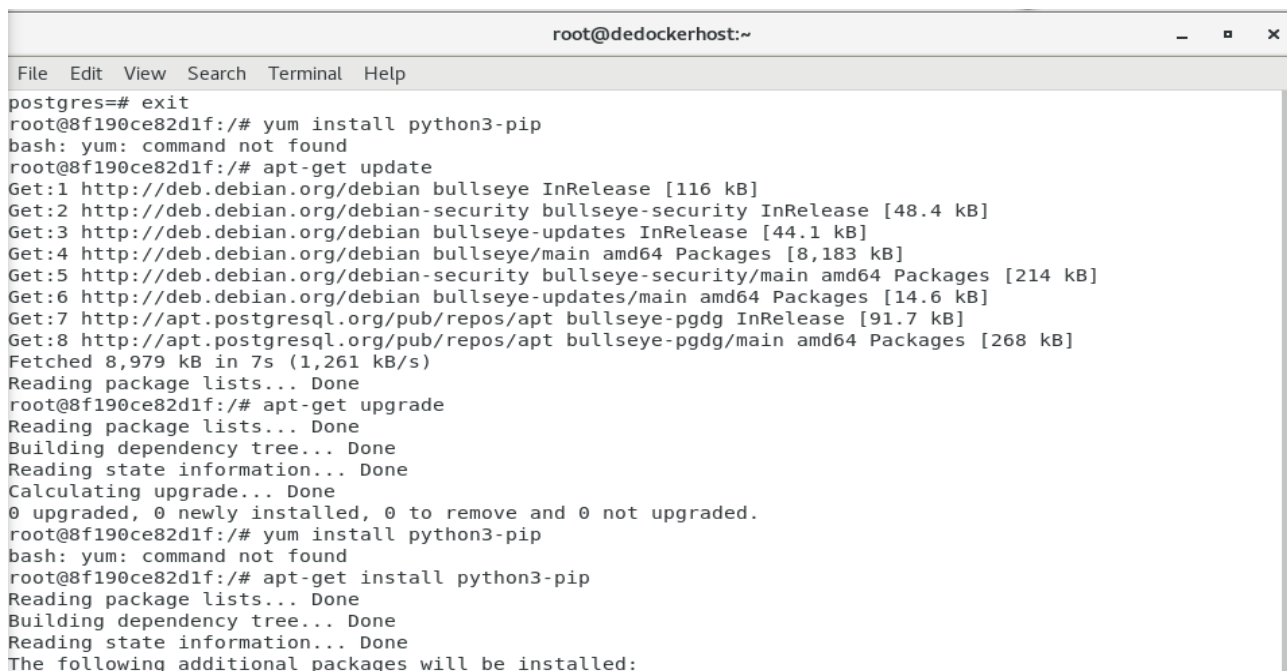
Recursos:

- 3 computadoras portátiles (de cada integrante del grupo).
- Descarga e instalación de PostgreSQL
- Máquina virtual - Virtual Box
- Descarga e instalación de SO Linux – Oracle

Proceso para el funcionamiento:

- Instalación y configuración de la máquina virtual
- Configuración del Docker desde la terminal en Linux.
- Instalación de PostgreSQL
- Creación de la base de datos en PostgreSQL
- Conexión ente el PostgreSQL y el Docker en la máquina virtual.
- Administración de la base de datos creada en PostgreSQL desde el Docker (máquina virtual).

Instalación de Python



```
root@dedockerhost:~  
File Edit View Search Terminal Help  
postgres=# exit  
root@8f190ce82d1f:/# yum install python3-pip  
bash: yum: command not found  
root@8f190ce82d1f:/# apt-get update  
Get:1 http://deb.debian.org/debian bullseye InRelease [116 kB]  
Get:2 http://deb.debian.org/debian-security bullseye-security InRelease [48.4 kB]  
Get:3 http://deb.debian.org/debian bullseye-updates InRelease [44.1 kB]  
Get:4 http://deb.debian.org/debian bullseye/main amd64 Packages [8,183 kB]  
Get:5 http://deb.debian.org/debian-security bullseye-security/main amd64 Packages [214 kB]  
Get:6 http://deb.debian.org/debian bullseye-updates/main amd64 Packages [14.6 kB]  
Get:7 http://apt.postgresql.org/pub/repos/apt bullseye-pgdg InRelease [91.7 kB]  
Get:8 http://apt.postgresql.org/pub/repos/apt bullseye-pgdg/main amd64 Packages [268 kB]  
Fetched 8,979 kB in 7s (1,261 kB/s)  
Reading package lists... Done  
root@8f190ce82d1f:/# apt-get upgrade  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
Calculating upgrade... Done  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
root@8f190ce82d1f:/# yum install python3-pip  
bash: yum: command not found  
root@8f190ce82d1f:/# apt-get install python3-pip  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:
```

Ilustración 1 - Instalación de Python

Instalación del módulo Faker en el Docker

```
root@8f190ce82d1f:/# pip3 install faker
Collecting faker
  Downloading Faker-16.4.0-py3-none-any.whl (1.7 MB)
    |████████████████████| 1.7 MB 1.4 MB/s
Collecting python-dateutil>=2.4
  Downloading python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
    |████████████████████| 247 kB 11.5 MB/s
Collecting six>=1.5
  Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: six, python-dateutil, faker
Successfully installed faker-16.4.0 python-dateutil-2.8.2 six-1.16.0
root@8f190ce82d1f:/#
```

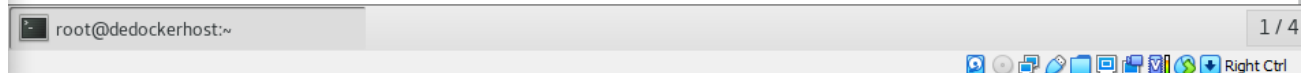


Ilustración 2 - Instalación del módulo Faker

Ingreso a la herramienta Postgres

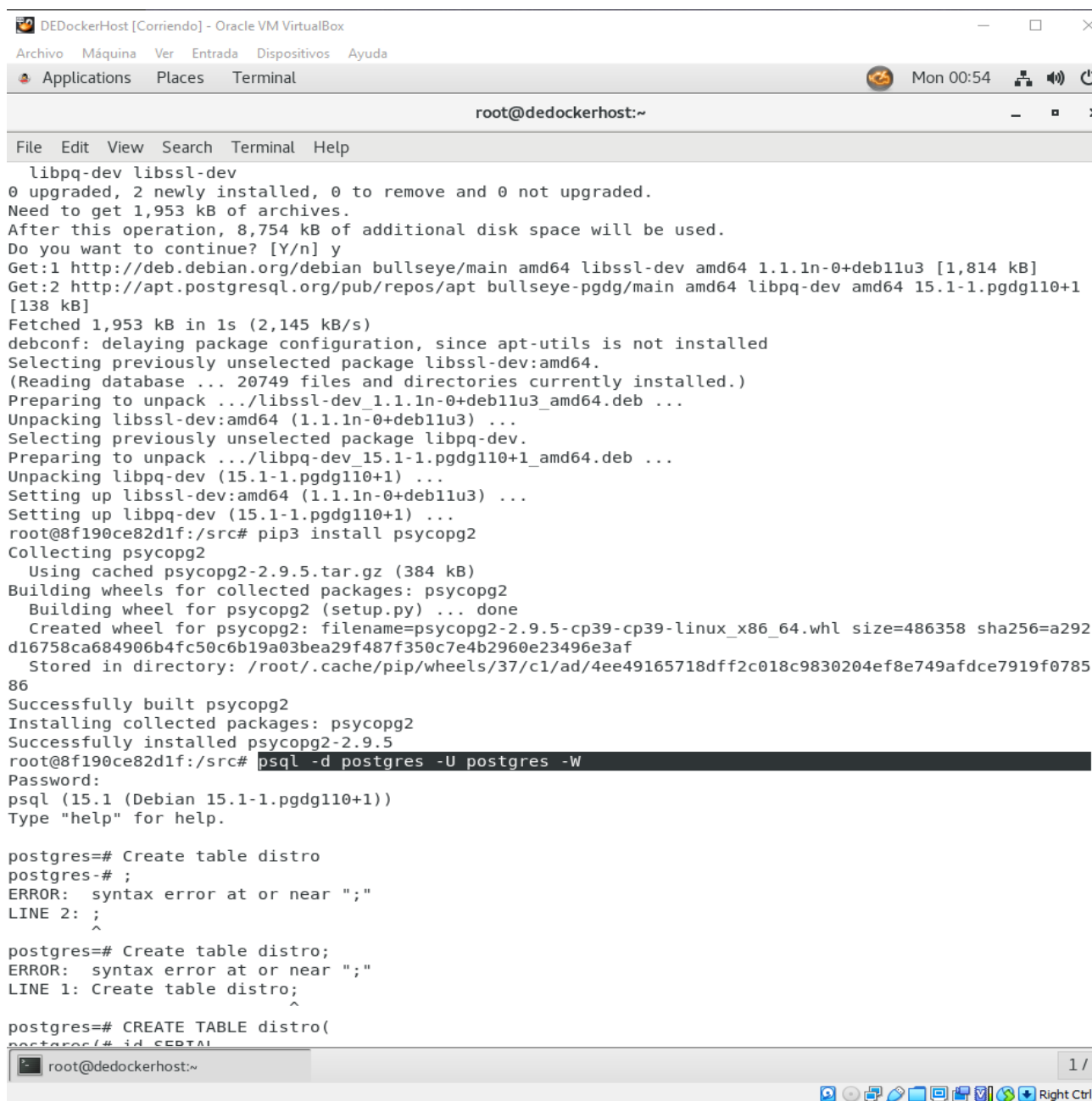


Ilustración 3 - Ejecución de la herramienta Postgres

Creación de Docker - Ingreso al docker

```
[root@dedockerhost ~]# docker run --name postgres -p 5432:5432 -v/docker-storage/postgres/conf.d:/etc/my  
sql -e POSTGRES_PASSWORD=udla -d postgres  
[root@dedockerhost ~]# docker exec -it postgres bash
```

Ilustración 4 - Creacion e Ingreso al Docker

Conexión de Python a Postgres

```
import psycopg2

# Global constant
PSQL_HOST = "10.10.10.2"
PSQL_PORT = "5432"
PSQL_USER = "postgres"
PSQL_PASS = "udla"
PSQL_DB = "Banco"

# Connection
connection_address = ""
"""" %(PSQL_HOST, PSQL_PORT, PSQL_USER, PSQL_PASS, PSQL_DB)
connection = psycopg2.connect(connection_address)

cursor= connection.cursor()

SQL= "SELECT * FROM distro;"
cursor.execute(SQL)

all_values = cursor.fetchall()

cursor.close()
connection.close()

print('Get values: ', all_values)
```

Ilustración 5 - Conexión de Python con Postgres

Creación de las tablas

Tabla cliente:

```
CREATE TABLE Cliente(  
ID_CLIENTE int NOT NULL primary key ,  
nombre_cliente nchar(50) NOT NULL,  
apellido_cliente nchar(50) NOT NULL,  
tlf_cliente int NOT NULL,  
edad_cliente int NOT NULL  
)
```

Tabla cuenta:

```
CREATE TABLE Cuenta(  
id_cuenta int NOT NULL PRIMARY KEY,  
saldo_cuenta float NULL  
)
```

Tabla movimiento:

```
CREATE TABLE Movimiento(  
id_movimiento int NOT NULL PRIMARY KEY ,  
tipo_movimiento nchar(50) NOT NULL,  
monto_movimiento float NOT NULL,  
cuentaID int NOT NULL,  
CONSTRAINT Fk_CuentaID FOREIGN KEY (cuentaID) REFERENCES Cuenta(id_cuenta)  
)
```

Tabla tiempo:

```
CREATE TABLE Tiempo(  
id_Tiempo int NOT NULL PRIMARY KEY,  
fecha_tiempo date NOT NULL,  
anio_tiempo int NOT NULL,  
trimestre_tiempo int NOT NULL,  
mes_tiempo varchar(50) NOT NULL,  
dia_tiempo int NOT NULL  
)
```

Tabla sucursal:

```
CREATE TABLE Sucursal(  
id_sucursal int NOT NULL PRIMARY KEY ,  
nombre_sucursal varchar(50) NOT NULL,  
ciudad_sucursal varchar(50) NOT NULL,  
telf_sucursal int NOT NULL  
)
```

Tabla comprobante:

```
CREATE TABLE Comprobante(  
id_Comprobante int NOT NULL PRIMARY KEY ,  
clienteID int NOT NULL,  
fechaID int NOT NULL,  
sucursalID int NOT NULL,  
movimientoID int NOT NULL,  
CONSTRAINT Fk_ClienteID FOREIGN KEY (clienteID) REFERENCES cliente(id_cliente),  
CONSTRAINT Fk_FechaID FOREIGN KEY (fechaID) REFERENCES Tiempo(id_tiempo),  
CONSTRAINT Fk_SucursalID FOREIGN KEY (sucursalID) REFERENCES Sucursal(id_sucursal),  
CONSTRAINT Fk_MovimientoID FOREIGN KEY (movimientoID) REFERENCES  
Movimiento(id_movimiento)  
)
```

```
INSERT INTO Cliente (ID_CLIENTE, nombre_cliente, apellido_cliente, tlf_cliente, edad_cliente)  
VALUES (11111111,'David','Estrella',999999999,22)  
INSERT INTO Cliente (ID_CLIENTE, nombre_cliente, apellido_cliente, tlf_cliente, edad_cliente)  
VALUES (22222222,'Henry','Lazo',888888888,22)
```

```
INSERT INTO Cliente (ID_CLIENTE, nombre_cliente, apellido_cliente, tlf_cliente, edad_cliente)  
VALUES (33333333,'Yordhan','Guerron',777777777,29)  
INSERT INTO Cliente (ID_CLIENTE, nombre_cliente, apellido_cliente, tlf_cliente, edad_cliente)  
VALUES (44444444,'Abigail','Fernandez',666666666,18)  
INSERT INTO Cliente (ID_CLIENTE, nombre_cliente, apellido_cliente, tlf_cliente, edad_cliente)  
VALUES (55555555,'Alexander','Fierro',555555555,26)  
INSERT INTO Cliente (ID_CLIENTE, nombre_cliente, apellido_cliente, tlf_cliente, edad_cliente)  
VALUES (66666666,'Esteban','Martinez',444444444,27)
```



```

INSERT INTO Cliente (ID_CLIENTE, nombre_cliente, apellido_cliente, tlf_cliente, edad_cliente)
VALUES (777777777,'Lizbeth','Flores',333333333,20)
INSERT INTO Cliente (ID_CLIENTE, nombre_cliente, apellido_cliente, tlf_cliente, edad_cliente)
VALUES (888888888,'Mateo','Davila',222222222,37)
INSERT INTO Cliente (ID_CLIENTE, nombre_cliente, apellido_cliente, tlf_cliente, edad_cliente)
VALUES (999999999,'Omar','Valenzuela',111111111,47)
INSERT INTO Cliente (ID_CLIENTE, nombre_cliente, apellido_cliente, tlf_cliente, edad_cliente)
VALUES (101010110,'Karen','Viteri',101010110,28)

```

```

INSERT INTO Sucursal(ID_SUCURSAL, NOMBRE_SUCURSAL, CIUDAD_SUCURSAL,
TELF_SUCURSAL) VALUES (1,'Condado','Quito',99999999)
INSERT INTO Sucursal(ID_SUCURSAL, NOMBRE_SUCURSAL, CIUDAD_SUCURSAL,
TELF_SUCURSAL) VALUES (2,'Guasmo','Guayaquil',88888888)
INSERT INTO Sucursal(ID_SUCURSAL, NOMBRE_SUCURSAL, CIUDAD_SUCURSAL,
TELF_SUCURSAL) VALUES (3,'Obrero','Puyo',7777777)
INSERT INTO Sucursal(ID_SUCURSAL, NOMBRE_SUCURSAL, CIUDAD_SUCURSAL,
TELF_SUCURSAL) VALUES (4,'Pifo','Quito',66666666)
INSERT INTO Sucursal(ID_SUCURSAL, NOMBRE_SUCURSAL, CIUDAD_SUCURSAL,
TELF_SUCURSAL) VALUES (5,'Malecon','Guayaquil',5555555)
INSERT INTO Sucursal(ID_SUCURSAL, NOMBRE_SUCURSAL, CIUDAD_SUCURSAL,
TELF_SUCURSAL) VALUES (6,'Tumbaco','Quito',44444444)
INSERT INTO Sucursal(ID_SUCURSAL, NOMBRE_SUCURSAL, CIUDAD_SUCURSAL,
TELF_SUCURSAL) VALUES (7,'Pomasqui','Quito',33333333)
INSERT INTO Sucursal(ID_SUCURSAL, NOMBRE_SUCURSAL, CIUDAD_SUCURSAL,
TELF_SUCURSAL) VALUES (8,'Conocoto','Quito',22222222)

```

```

INSERT INTO Cuenta(id_cuenta, saldo_cuenta) VALUES (1,500)
INSERT INTO Cuenta(id_cuenta, saldo_cuenta) VALUES (2,2500)
INSERT INTO Cuenta(id_cuenta, saldo_cuenta) VALUES (3,4500)
INSERT INTO Cuenta(id_cuenta, saldo_cuenta) VALUES (4,310)
INSERT INTO Cuenta(id_cuenta, saldo_cuenta) VALUES (5,2500)
INSERT INTO Cuenta(id_cuenta, saldo_cuenta) VALUES (6,600)
INSERT INTO Cuenta(id_cuenta, saldo_cuenta) VALUES (7,6600)
INSERT INTO Cuenta(id_cuenta, saldo_cuenta) VALUES (8,70)
INSERT INTO Cuenta(id_cuenta, saldo_cuenta) VALUES (9,800)
INSERT INTO Cuenta(id_cuenta, saldo_cuenta) VALUES (10,950)

```

ROLES:

```

CREATE USER david With PASSWORD 'udla';
CREATE ROLE DB_developer;
grant CREATE, CONNECT, TEMPORARY on DATABASE postgres to DB_developer;
grant DB_developer to david;

```

Evaluación de Conceso

| Nombre del estudiante | Henry Lazo | | | | |
|-----------------------------------------------------------------------|------------|---|---|---|---|
| Criterios | 1 | 2 | 3 | 4 | 5 |
| Mostro responsabilidad con la parte del trabajo asignado. | | | | | ✓ |
| Participo en las discusiones del grupo aportando ideas, clarificando. | | | | | ✓ |
| Escucho activamente a los demás. | | | | | ✓ |
| Acepto las opiniones de los miembros del grupo | | | | | ✓ |
| Fue respetuoso y no obstaculizo el trabajo del grupo | | | | | ✓ |
| Ánimo y apoyo al resto de los compañeros | | | | | ✓ |

| Nombre del estudiante | David Estrella | | | | |
|-----------------------------------------------------------------------|----------------|---|---|---|---|
| Criterios | 1 | 2 | 3 | 4 | 5 |
| Mostro responsabilidad con la parte del trabajo asignado. | | | | | ✓ |
| Participo en las discusiones del grupo aportando ideas, clarificando. | | | | | ✓ |
| Escucho activamente a los demás. | | | | | ✓ |
| Acepto las opiniones de los miembros del grupo | | | | | ✓ |
| Fue respetuoso y no obstaculizo el trabajo del grupo | | | | | ✓ |
| Ánimo y apoyo al resto de los compañeros | | | | | ✓ |

| Nombre del estudiante | Yhordan Guerron | | | | |
|-----------------------------------------------------------------------|-----------------|---|---|---|---|
| Criterios | 1 | 2 | 3 | 4 | 5 |
| Mostro responsabilidad con la parte del trabajo asignado. | | | | | ✓ |
| Participo en las discusiones del grupo aportando ideas, clarificando. | | | | | ✓ |
| Escucho activamente a los demás. | | | | | ✓ |
| Acepto las opiniones de los miembros del grupo | | | | | ✓ |
| Fue respetuoso y no obstaculizo el trabajo del grupo | | | | | ✓ |
| Ánimo y apoyo al resto de los compañeros | | | | | ✓ |