



# GUÍA DE POSTGRES CON DOCKER

Os dejo una guía paso a paso para que os funcione este contenedor de **Docker** en vuestros equipos sin necesidad de descargarnos postgres en nuestro equipo local.

En primer lugar descargamos/abrimos Docker en nuestro equipo local (el servicio también activo)

Ahora en el directorio “Docker” veremos un archivo docker-compose y Dockerfile.

 docker-compose	02/04/2022 18:16	Archivo de origen ...	1 KB
 Dockerfile	19/10/2022 19:41	Archivo	1 KB

Ahora, abrimos con VSC el archivo **docker-compose.yml** :

```
docker-compose.yml x
docker-compose.yml
1  version: "3.8"
2
3  services:
4    postgres:
5      image: postgres
6      container_name: postgres-pgadmin
7      restart: always
8      ports:
9        - "5432:5432"
10     environment:
11       - DATABASE_HOST=localhost
12       - POSTGRES_USER=admin
13       - POSTGRES_PASSWORD=admin
14     volumes:
15       - ./pgadmin-postgres:/var/lib/postgresql/data
16
17   pgadmin:
18     image: dpage/pgadmin4
19     environment:
20       PGADMIN_DEFAULT_EMAIL: "rafael.martinezl@campusfp.es"
21       PGADMIN_DEFAULT_PASSWORD: "admin"
22     ports:
23       - "80:80"
24     depends_on:
25       - postgres
```

Como vemos, hay unas flechas en rojo con los datos que podremos cambiar a nuestro gusto.

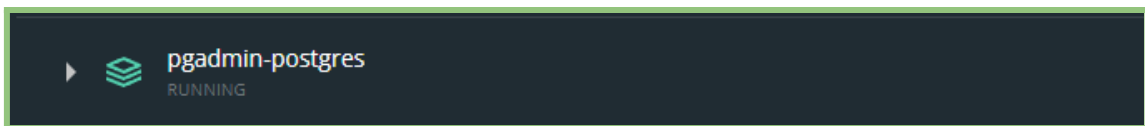
- **POSTGRES\_USER** = El nombre de usuario que nosotros queramos para nuestra conexión.
- **POSTGRES\_PASSWORD** = La contraseña que nosotros queramos para nuestra conexión.
- **PGADMIN\_DEFAULT\_EMAIL** = Este va a ser el email que pondremos más adelante para acceder a **PGADMIN**.
- **PGADMIN\_DEFAULT\_PASSWORD** = Esta va a ser la contraseña que pondremos más adelante para acceder a **PGADMIN**.

Una vez tengamos configurado este archivo a nuestro gusto, procederemos a ejecutarlo. Para ello, abriremos el terminal desde VSC con **CTRL+Ñ**.

Cuando lo hayamos abierto ejecutaremos el siguiente comando (docker-compose up)  
 Docker-compose hace referencia al archivo a ejecutar. No nos olvidemos que para ejecutar el comando debemos estar en la ruta del archivo que vamos a ejecutar.

```
docker compose up --build -d
```

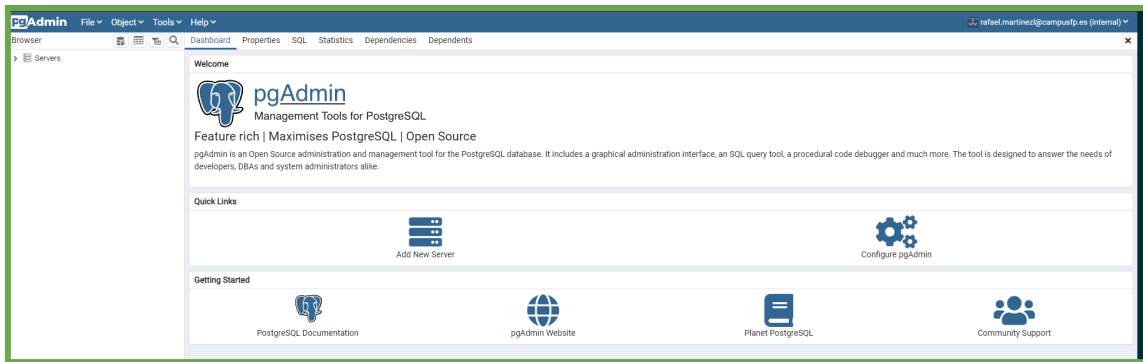
Lo ejecutamos y empezará a montarnos nuestros contenedores.



Cuando ya hayamos levantado los contenedores, vamos a ir a **localhost:8080** que es donde configuraremos nuestra base de datos.



Aquí es donde vamos a introducir el **PGADMIN\_DEFAULT\_EMAIL** y la **PGADMIN\_DEFAULT\_PASSWORD** que hayamos indicado en el docker-compose.yml.



Ahora vamos a configurar la conexión de **pgadmin**, para ello haremos un clic derecho sobre **“server”**, se nos abrirá un menú y clickeamos en **register > server..** y se nos abrirá una ventana como esta.

**Register - Server**

General Connection SSL SSH Tunnel Advanced

**Name** !

**Server group** Servers

**Background** ☐

**Foreground** ☐

**Connect now?** ☒

**Shared?** ☐

**Comments**

! 'Name' cannot be empty. ×

i ? Close Reset Save

En **amarillo** pondremos el nombre que queramos para nuestra conexión y en la **flecha roja** clickearemos para configurar esa conexión.

The screenshot shows the 'Register - Server' dialog box with the 'Connection' tab selected. The 'Host name/address' field is highlighted with a red arrow labeled '1'. The 'Maintenance database' field contains the text 'postgres'. The 'Username' field contains 'rafael.martinezl' and is highlighted with a red arrow labeled '2'. The 'Password' field is empty and highlighted with a red arrow labeled '3'. A red error message at the bottom states: 'Name' cannot be empty. The dialog also includes tabs for General, SSL, SSH Tunnel, and Advanced, and buttons for Close, Reset, and Save.




En la **flecha 1** pondremos el contenedor de postgres que hemos creado con la imagen (en este caso se llama postgres, así que colocaremos como nombre **“postgres”**, **NO PONER LOCALHOST**)

En la **flecha 2** colocaremos el usuario que hayamos editado antes para postgres, en mi caso es **“admin”**.

En la **flecha 3** colocaremos la contraseña que hayamos editado antes para postgres, en mi caso es **“admin”**.

Con esto ya tendríamos la conexión configurada. Ahora vamos a hacer la restauración de la base de datos del proyecto.

Primero, deberemos colocar el archivo **“recuperar.SQL”** en el siguiente directorio :

 docker-compose	27/11/2022 14:22	Archivo de origen ...	1 KB
 Dockerfile	27/11/2022 14:20	Archivo	1 KB
 recuperar	27/11/2022 14:18	SQL Text File	9 KB

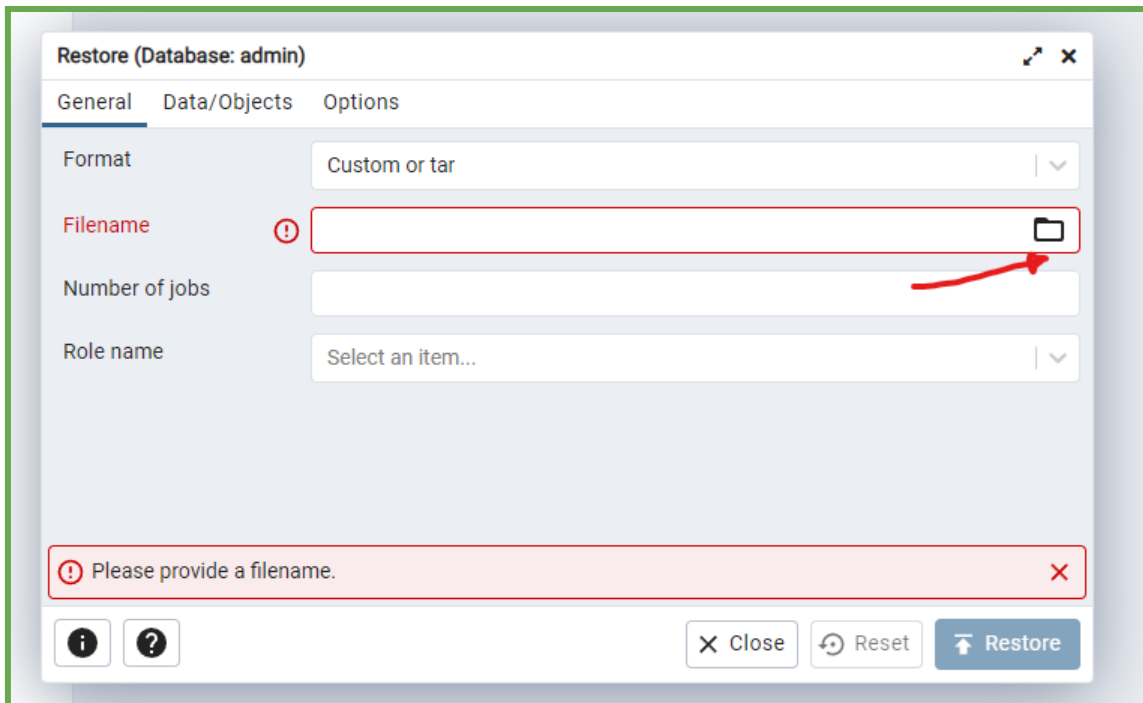


```
\pgadmin\storage\admin_gmail.com>
```

El último directorio hace alusión al usuario que hemos creado para acceder al Pgadmin. Es necesario hacer esto ya que la información de la base de datos se encuentra dentro del contenedor de docker, por lo que hemos tenido que persistir ciertos directorios para trabajar más cómodamente.

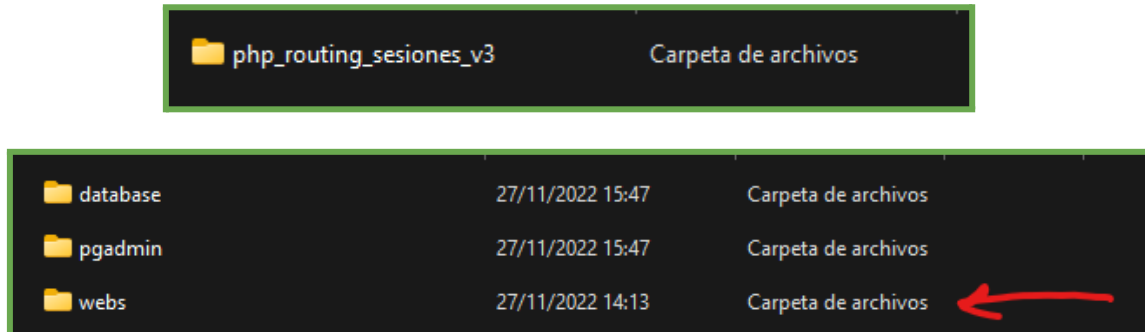
Ahora desplegamos la lista con las bases de datos que existen y veremos una con el nombre de **admin** (En caso de que no la veamos, la podemos crear nosotros con el mismo nombre, ya que es el nombre que está en la cadena de conexión del proyecto).

Hacemos clic derecho sobre la base de datos y seleccionaremos **"restore"**. Se nos abrirá lo siguiente :

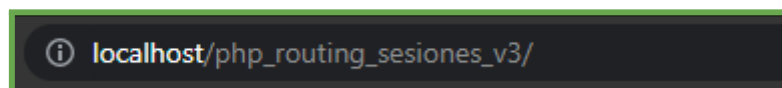


Clickeamos en el icono y seleccionamos el archivo **"recuperar.SQL"** que hemos ubicado anteriormente. Una vez hecho esto, ya tendríamos lista nuestra base de datos del proyecto.

Ahora solo nos queda descomprimir el **archivo.zip** del proyecto y colocar el directorio que contiene el zip en la siguiente ruta que se nos ha creado con el docker-compose que hicimos anteriormente :



Ahora en el navegador ponemos lo siguiente :



Y listo, ya tenemos nuestro proyecto configurado para usarlo. Espero que disfrutes del proyecto.

El usuario para el rol de administrador es :

- **user : admin**
- **password : admin**

El usuario para el rol de usuario es :

- **user : pruebas**
- **password : 1234**

