

Universidad Nacional Autónoma de México  
Facultad de Ciencias

## Práctica 01: Bitácora Escobar

Curso: Fundamentos de Bases de Datos  
Alumno: **Escobar Gonzalez Isaac Giovani**  
Profesor: Gerardo Avilés Rosas  
AyudanteLab: Ricardo Badillo Macías  
AyudanteTeorico: Luis Enrique García Gómez/ Jaime Octavio Delfín López

Ciclo Escolar: 2026-I

1 de septiembre de 2025

# Bitácora

## Sistema operativo y versión

- Memoria: 16 GB
- Procesador: 12th Gen Intel® Core™ i5-12400 (12) @ 5.60 GHz
- Tarjeta gráfica: NVIDIA GeForce RTX 3060
- Sistema Operativo: Fedora Linux 42 (Workstation Edition) x86\_64
- Entorno de escritorio: GNOME 48.4
- Kernel Version: 6.16.3-200.fc42.x86\_64
- Arquitectura: 64-bit

## Distribución de Linux

Distribución: Fedora Linux 42 (Workstation Edition).

## Versión de la instalación

Docker 28.3.3, build 980b856

SMBD PostgreSQL no se instalo solo se utilizara con contenedores

DBeaver 25.2.0.202508311659

## Tiempo requerido

Aproximadamente: **15** a **20** minutos

## Paso a paso de la instalación

### 1. Instalación de Docker en Linux (Fedora 42).

#### a) Actualizar el sistema y preparar herramientas para manejar repositorios DNF:

```
sudo dnf -y update my_user_password
sudo dnf -y install dnf-plugins-core
```

```

> sudo dnf -y update
[sudo] contraseña para isaac:
Actualizando y cargando repositorios:
RPM Fusion for Fedora 42 - Nonfree - NVIDIA Driver 100% | 2.1 KiB/s | 6.0 KiB | 00m03s
google-chrome 100% | 1.7 KiB/s | 1.3 KiB | 00m01s
Copr repo for PyCharm owned by phracek 100% | 2.7 KiB/s | 2.1 KiB | 00m01s
Visual Studio Code 100% | 2.0 KiB/s | 1.5 KiB | 00m01s
RPM Fusion for Fedora 42 - Nonfree - Steam 100% | 2.8 KiB/s | 5.7 KiB | 00m02s
Docker CE Stable - x86_64 100% | 6.8 KiB/s | 3.5 KiB | 00m01s
Fedora 42 - x86_64 - Updates 100% | 20.4 KiB/s | 30.4 KiB | 00m01s
Repositorios cargados.
Nada que hacer.

> sudo dnf -y install dnf-plugins-core
Actualizando y cargando repositorios:
Repositorios cargados.
El paquete "dnf-plugins-core-4.10.1-1.fc42.noarch" ya está instalado.
Nada que hacer.

```

#### b) Agregar el repositorio oficial de Docker:

```
sudo dnf-3 config-manager --add-repo https://download.docker.com/linux/fedora/docker-ce.repo
```

```

> sudo dnf-3 config-manager --add-repo https://download.docker.com/linux/fedora/docker-ce.repo

Agregando repositorio de: https://download.docker.com/linux/fedora/docker-ce.repo

```

#### c) Instalar Docker Engine y sus complementos (ultima versión):

```
sudo dnf install -y docker-ce docker-ce-cli containerd.io
docker-buildx-plugin docker-compose-plugin
```

```

> sudo dnf install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

Install Docker Engine
Actualizando y cargando repositorios:
  Docker CE Stable - x86_64                                100% | 70.5 KiB/s | 3.5 KiB | 00m00s
Repositorios cargados.
El paquete "containerd.io-1.7.27-3.1.fc42.x86_64" ya está instalado.
El paquete "docker-buildx-plugin-0.26.1-1.fc42.x86_64" ya está instalado.

Paquete                Arq.      Versión                Repositorio                Tamaño
Instalando:
docker-ce              x86_64    3:28.3.3-1.fc42        docker-ce-stable           86.0 MiB
docker-ce-cli          x86_64    1:28.3.3-1.fc42        docker-ce-stable           35.3 MiB
docker-compose-plugin  x86_64    2.39.1-1.fc42          docker-ce-stable           73.0 MiB
Instalando dependencias:
libcgroup              x86_64    3.0-8.fc42             fedora                     157.7 KiB
slirp4netns            x86_64    1.3.1-2.fc42           fedora                     89.3 KiB
Instalando dependencias débiles:
docker-ce-rootless-extras x86_64    28.3.3-1.fc42          docker-ce-stable           11.2 MiB

Resumen de la transacción:
Instalando:          6 paquetes

El tamaño total de paquetes entrantes es 47 MiB. Se necesita descargar 47 MiB.
Después de esta operación, 206 MiB extra serán utilizados (instalar 206 MiB, eliminar 0 B).
[1/6] docker-ce-cli-1:28.3.3-1.fc42.x86_64                100% | 4.5 MiB/s | 8.6 MiB | 00m02s
[2/6] docker-compose-plugin-0:2.39.1-1.fc42.x86_64        100% | 7.6 MiB/s | 15.0 MiB | 00m02s
[3/6] libcgroup-0:3.0-8.fc42.x86_64                       100% | 147.3 KiB/s | 73.8 KiB | 00m01s
[4/6] slirp4netns-0:1.3.1-2.fc42.x86_64                   100% | 305.1 KiB/s | 47.3 KiB | 00m00s
[5/6] docker-ce-rootless-extras-0:28.3.3-1.fc42.x86_64   100% | 3.4 MiB/s | 3.4 MiB | 00m01s
[6/6] docker-ce-3:28.3.3-1.fc42.x86_64                   100% | 6.6 MiB/s | 20.3 MiB | 00m03s
-----
[6/6] Total                                                100% | 13.7 MiB/s | 47.3 MiB | 00m03s
Ejecutando transacción
[1/8] Verificar archivos de paquete                        100% | 51.0 B/s | 6.0 B | 00m00s
[2/8] Preparar transacción                                100% | 16.0 B/s | 6.0 B | 00m00s
[3/8] Instalando slirp4netns-0:1.3.1-2.fc42.x86_64        100% | 5.5 MiB/s | 90.7 KiB | 00m00s
[4/8] Instalando libcgroup-0:3.0-8.fc42.x86_64            100% | 51.8 MiB/s | 159.1 KiB | 00m00s
[5/8] Instalando docker-ce-cli-1:28.3.3-1.fc42.x86_64    100% | 327.6 MiB/s | 35.4 MiB | 00m00s
[6/8] Instalando docker-ce-3:28.3.3-1.fc42.x86_64        100% | 452.6 MiB/s | 86.0 MiB | 00m00s
[7/8] Instalando docker-ce-rootless-extras-0:28.3.3-1.fc42.x86_64 100% | 259.8 MiB/s | 11.2 MiB | 00m00s
[8/8] Instalando docker-compose-plugin-0:2.39.1-1.fc42.x86_64 100% | 78.2 MiB/s | 73.0 MiB | 00m01s
¡Completado!

```

d) Habilitar y arrancar el servicio de Docker:

```
sudo systemctl enable --now docker
```

```

> sudo systemctl enable --now docker

Created symlink '/etc/systemd/system/multi-user.target.wants/docker.service' -> '/usr/lib/systemd/system/docker.service'.

```

e) Verificar que Docker funciona correctamente con un contenedor de prueba:

```
sudo docker run hello-world
```

```

$ > ~
$ sudo docker run hello-world

Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
17eec7bbc9d7: Pull complete
Digest: sha256:a0dfb02aac212703bfc339d77d47ec32c8706ff250850ecc0e19c8737b18567
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

```

## 2. Instalación de PostgreSQL con contenedores.

a) Descargar la imagen oficial de PostgreSQL desde Docker Hub:)

```
docker pull postgres
```

```

$ > ~
$ docker pull postgres

Using default tag: latest
latest: Pulling from library/postgres
396b1da7636e: Pull complete
f5465e2fc020: Pull complete
c166c949e1c3: Pull complete
7fa725c973af: Pull complete
1f6dfcaad4e9: Pull complete
b7a79609094c: Pull complete
901a9540064a: Pull complete
085f0a899c07: Pull complete
5d91a345d79a: Pull complete
f7f2afaa1b41: Pull complete
36b4e7f51364: Pull complete
85558a023eea: Pull complete
be9fdbdba096: Pull complete
ae28e2b99a62: Pull complete
Digest: sha256:29e0bb09c8e7e7fc265ea9f4367de9622e55bae6b0b97e7cce740c2d63c2ebc0
Status: Downloaded newer image for postgres:latest
docker.io/library/postgres:latest

```

- b) Crear y ejecutar un contenedor de PostgreSQL en segundo plano con contraseña y puerto mapeado:

```
docker run -d --name postgres -e POSTGRES_PASSWORD=
mysecretpassword -p 5432:5432 postgres
```

```

> docker run -d --name postgres -e POSTGRES_PASSWORD=mysecretpassword -p 5432:5432 postgres
90a5e953b6da8c3b67c6b1813ddb9631f23440eb058b85c029c389d30bee02a6

```

- c) Listar contenedores creados:

```
docker ps -a
```

```

> docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
86837b654db8   postgres      "docker-entrypoint.s..." 2 seconds ago  Up 1 second   0.0.0.0:5432->5432/tcp,
[::]:5432->5432/tcp
1a1821373e52   hello-world   "/hello"                 6 minutes ago  Exited (0) 5 minutes ago
sad_bhabha

```

- d) Iniciar un contenedor detenido:

```
docker start <CONTAINER_ID>
```

```

> docker start 86837b654db8
86837b654db8

```

- e) Detener un contenedor en ejecución:

```
docker stop <CONTAINER_ID>
```

```

> docker stop 86837b654db8
86837b654db8

```

- f) Conectarse a PostgreSQL usando psql:

```
docker run -it --link postgres:postgres \
postgres psql -h postgres -U postgres
```

```

> docker run -it --link postgres:postgres postgres psql -h postgres -U postgres
Password for user postgres:
psql (17.6 (Debian 17.6-1.pgdg13+1))
Type "help" for help.

postgres=# _

```

### 3. Instalación de herramientas gráficas (Database Tools).

En este caso instalare **DBeaver** por recomendación del ayudante desde su página oficial mediante su archivo **.rpm**

**DBeaver.** Cliente SQL y herramienta de administración de bases de datos.

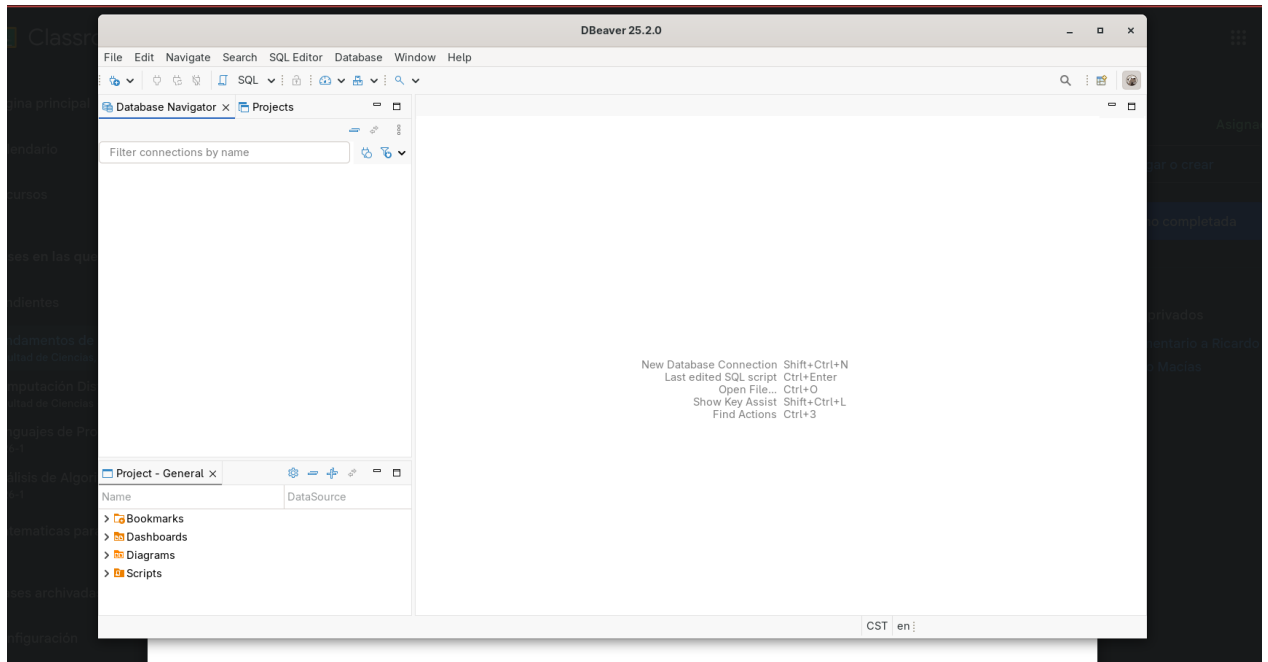
Enlace: <https://dbeaver.io/download/>

```

> sudo dnf install ./dbeaver-ce-25.2.0-stable.x86_64.rpm
[sudo] contraseña para isaac:
Actualizando y cargando repositorios:
Repositorios cargados.
Paquete      Arq.      Versión      Repositorio      Tamaño
Instalando:  dbeaver-ce  x86_64      25.2.0-stable  @commandline      166.2 MiB
Resumen de la transacción:
  Instalando: 1 paquete
El tamaño total de paquetes entrantes es 129 MiB. Se necesita descargar 0 B.
Después de esta operación, 166 MiB extra serán utilizados (instalar 166 MiB, eliminar 0 B).
Is this ok [y/N]: y
Ejecutando transacción
[1/3] Verificar archivos de paquete
[2/3] Preparar transacción
[3/3] Instalando dbeaver-ce-0:25.2.0-stable.x86_64
Aviso: comprobante OpenPGP omitido para 1 paquete desde repositorio: @commandline
¡Completado!

> dbeaver
DBeaver: Es un cliente SQL y una herramienta de administración de bases de datos, utiliza la API JDBC
para interactuar con ellas a través de un controlador JDBC. Para otras, como NoSQL, utiliza controladores
para Application Standalone [org.jkiss.dbeaver.ui.app.standalone 25.2.0.202508311659]
> Start Eclipse IDE UI Application
> Start Eclipse Jobs Mechanism [org.eclipse.jobs 3.10.0]
> Start DBeaver Command Line Mode
> Start Eclipse IDE UI [org.eclipse.ui 3.118.0]
> Start DBeaver Model Registry [org.jkiss.dbeaver.model 25.2.0.202508311659]
> Start jna [com.sun.jna 5.17.0]
> Start DBeaver UI Editors - Base [org.jkiss.dbeaver.ui.editors.base 25.2.0.202508311659]
> Start DBeaver UI [org.jkiss.dbeaver.ui 25.2.0.202508311659]
> Start Expression Language [org.jkiss.dbeaver.model.sql 25.2.0.202508311659]
> Start DBeaver Usage Statistics [org.jkiss.dbeaver.model.sql 25.2.0.202508311659]
> Start DBeaver UI Editors - Console [org.jkiss.dbeaver.ui.editors.console 25.2.0.202508311659]
> Start DBeaver Desktop Application Core [org.jkiss.dbeaver.core 25.2.0.202508311659]
2025-08-31 19:10:33.201 - DBeaver 25.2.0.202508311659 is starting
2025-08-31 19:10:33.201 - OS: Linux 6.16.3-200.fc42.x86_64 (amd64)
2025-08-31 19:10:33.201 - Java version: 21.0.5 by Eclipse Adoptium (64bit)
2025-08-31 19:10:33.201 - Install path: '/usr/share/dbeaver-ce'
2025-08-31 19:10:33.201 - Instance path: 'file:/home/isaac/.local/share/DBeaverData/workspace6/'
2025-08-31 19:10:33.202 - Memory available 64Mb/1024Mb
2025-08-31 19:10:33.207 - Create display
2025-08-31 19:10:33.267 - Initialize desktop platform...

```



## Comentarios y problemas encontrados

Al instalar Docker no tuve ningun inconveniente al momento de hacer la instalación, tuve facilidad de seguir los pasos e instrucciones de la página oficial de **Docker** para mi distribución de Linux **Fedora 42**.

Para **PostgreSQL** solamente tuve el detalle de que mi usuario carecía de los permisos para su contenedor de PostgreSQL, solo tuve que modificar los permisos de mi usuario para no tener que el 'sudo' cada vez que tenga que usar algun contenedor o descargar una imagen.

Por último, para **DBeaver** no tuve problemas pues su instalación me resulto relativamente fácil, solo descargar un archivo y ejecutarlo desde la terminal para su instalación.

El único problema que tuve en general de inicio, fue que tuve que hacer la desinstalación total de Docker puesto que ya lo tenía instalado previamente, pero casi no tuve ninguna otra complicación.