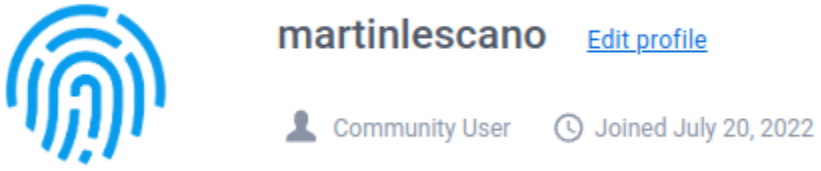
Trabajo Práctico 2 - Introducción a Docker

4- Desarrollo:

1- Instalar Docker Community Edition

```
tincho@IdeaPad-5-14ALC05:~/Downloads$ sudo dpkg -i docker-desktop-4.24.0-amd64.deb
(Reading database ... 201691 files and directories currently installed.)
Preparing to unpack docker-desktop-4.24.0-amd64.deb ...
Unpacking docker-desktop (4.24.0-122432) over (4.24.0-122432) ...
Setting up docker-desktop (4.24.0-122432) ...
Enabling use of privileged ports by Docker Desktop
kubernetes.docker.internal already in /etc/hosts
Reloading systemd daemon for logged in users
Done reloading systemd daemon for logged in users
Processing triggers for mailcap (3.70+nmu1ubuntu1) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu3) ...
Processing triggers for desktop-file-utils (0.26-1ubuntu3) ...
tincho@IdeaPad-5-14ALC05:~/Downloads$ docker version
Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?
Client: Docker Engine - Community
 Cloud integration: v1.0.35+desktop.5
 Version:           24.0.6
 API version:       1.43
 Go version:        go1.20.7
 Git commit:        ed223bc
 Built:             Mon Sep  4 12:31:44 2023
 OS/Arch:           linux/amd64
 Context:           default
```

2- Explorar DockerHub



3- Obtener la imagen BusyBox

```
tincho@IdeaPad-5-14ALC05:~$ docker pull busybox
Using default tag: latest
latest: Pulling from library/busybox
3f4d90098f5b: Pull complete
Digest: sha256:3fbc632167424a6d997e74f52b878d7cc478225cffac6bc977eedfe51c7f4e79
Status: Downloaded newer image for busybox:latest
docker.io/library/busybox:latest

What's Next?
  1. Sign in to your Docker account → docker login
  2. View a summary of image vulnerabilities and recommendations → docker scout quickview busybox
tincho@IdeaPad-5-14ALC05:~$ docker images
REPOSITORY   TAG       IMAGE ID       CREATED        SIZE
busybox      latest    a416a98b71e2   2 months ago   4.26MB
```

4- Ejecutando contenedores

```
tincho@IdeaPad-5-14ALC05:~$ docker run busybox
```

Como no especifique ningún comando adicional para ejecutar dentro del contenedor, el contenedor se inicia y luego finaliza inmediatamente. Esto significa que el contenedor se crea, realiza cualquier trabajo que tenga que hacer (que en este caso no es nada), y luego se detiene automáticamente. Como resultado, no hay salida en la terminal.

```
tincho@IdeaPad-5-14ALC05:~$ docker run busybox echo "Hola Mundo"
Hola Mundo
```

```
tincho@IdeaPad-5-14ALC05:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
tincho@IdeaPad-5-14ALC05:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND              CREATED          STATUS                      PORTS     NAMES
dc0923f066ac   busybox   "echo 'Hola Mundo'"  47 seconds ago   Exited (0) 46 seconds ago             galla
nt_bhaskara
8fa32db83645   busybox   "sh"                 2 minutes ago    Exited (0) 2 minutes ago              charm
ing_sutherland
```

Al ejecutar docker ps -a, obtengo una lista de todos los contenedores, tanto los que están en ejecución como los que han finalizado. El contenedor de BusyBox que ejecuté anteriormente aparece en la lista con un estado de "Exited". Además, puedo ver información sobre su ID, nombre y otros detalles relevantes.

5- Ejecutando en modo interactivo

```
tincho@IdeaPad-5-14ALC05:~$ docker run -it busybox sh
/ # ps
PID   USER     TIME  COMMAND
    1 root      0:00 sh
    7 root      0:00 ps
/ # uptime
 03:33:18 up 19 min,  0 users,  load average: 0.04, 0.01, 0.00
/ # free
              total        used        free      shared  buff/cache   available
Mem:        1789992      416468      118024      238560     1255500      984864
Swap:       1048572       23696     1024876
/ # ls -l /
total 40
drwxr-xr-x    2 root     root         12288 Jul 17 18:30 bin
drwxr-xr-x    5 root     root           360 Oct  4 03:33 dev
drwxr-xr-x    1 root     root          4096 Oct  4 03:33 etc
drwxr-xr-x    2 nobody   nobody        4096 Jul 17 18:30 home
drwxr-xr-x    2 root     root          4096 Jul 17 18:30 lib
lrwxrwxrwx    1 root     root             3 Jul 17 18:30 lib64 -> lib
dr-xr-xr-x  319 root     root             0 Oct  4 03:33 proc
drwx------    1 root     root          4096 Oct  4 03:33 root
dr-xr-xr-x   12 root     root             0 Oct  4 03:33 sys
drwxrwxrwt    2 root     root          4096 Jul 17 18:30 tmp
drwxr-xr-x    4 root     root          4096 Jul 17 18:30 usr
drwxr-xr-x    4 root     root          4096 Jul 17 18:30 var
/ # exit
```

6- Borrando contendores terminados

```
tincho@IdeaPad-5-14ALC05:~$ docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS                      PORTS     NAMES
47dfe2b3aa97   busybox   "sh"                     3 minutes ago    Exited (0) 2 minutes ago              cleve
r_hermann
dc0923f066ac   busybox   "echo 'Hola Mundo'"      12 minutes ago   Exited (0) 12 minutes ago             galla
nt_bhaskara
8fa32db83645   busybox   "sh"                     14 minutes ago   Exited (0) 14 minutes ago             charm
ing_sutherland
tincho@IdeaPad-5-14ALC05:~$ docker rm dc0923f066ac
dc0923f066ac
tincho@IdeaPad-5-14ALC05:~$ docker rm $(docker ps -a -q -f status=exited)
47dfe2b3aa97
8fa32db83645
tincho@IdeaPad-5-14ALC05:~$ docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Total reclaimed space: 0B
```

7- Construir una imagen

```
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP2$ git clone https://github.com/ingsoft3u
cc/SimpleWebAPI
Cloning into 'SimpleWebAPI'...
remote: Enumerating objects: 150, done.
remote: Counting objects: 100% (150/150), done.
remote: Compressing objects: 100% (115/115), done.
remote: Total 150 (delta 60), reused 33 (delta 9), pack-reused 0
Receiving objects: 100% (150/150), 28.41 KiB | 3.16 MiB/s, done.
Resolving deltas: 100% (60/60), done.
```


```
# Establecer la imagen base (en este caso, una imagen de ASP.NET Core)
FROM mcr.microsoft.com/dotnet/aspnet:5.0 AS base
```

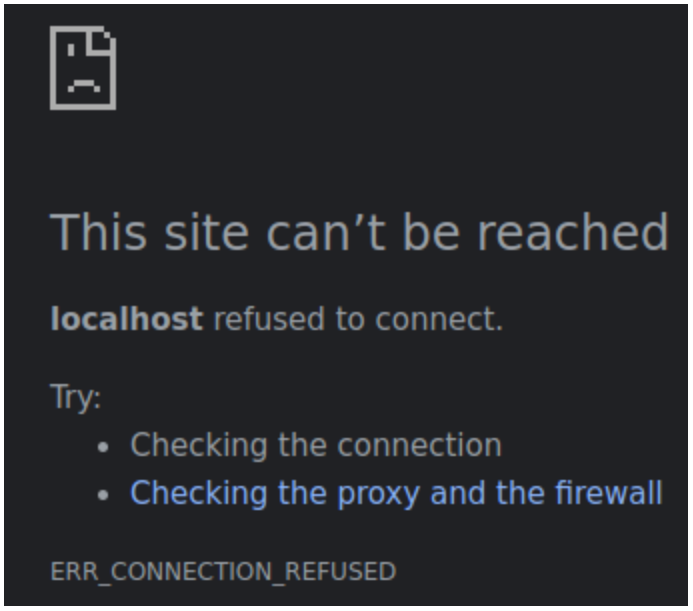
```
# Establecer el directorio de trabajo dentro del contenedor
WORKDIR /app
# Exponer el puerto 80 para que la aplicación pueda escuchar las solicitudes
EXPOSE 80
# Establecer la imagen base para compilar el código fuente
FROM mcr.microsoft.com/dotnet/sdk:5.0 AS build
WORKDIR /src
# Copiar el archivo csproj y restaurar las dependencias
COPY ["SimpleWebAPI/SimpleWebAPI.csproj", "SimpleWebAPI/"]
RUN dotnet restore "SimpleWebAPI/SimpleWebAPI.csproj"
# Copiar todo el código fuente al directorio de trabajo
COPY . .
WORKDIR "/src/SimpleWebAPI"
RUN dotnet build "SimpleWebAPI.csproj" -c Release -o /app/build
# Publicar la aplicación
FROM build AS publish
RUN dotnet publish "SimpleWebAPI.csproj" -c Release -o /app/publish
# Crear la imagen final utilizando la imagen base "base" y los archivos publicados
FROM base AS final
WORKDIR /app
COPY --from=publish /app/publish .
ENTRYPOINT ["dotnet", "SimpleWebAPI.dll"]
```

```
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP2/SimpleWebAPI$ docker images
REPOSITORY   TAG       IMAGE ID       CREATED         SIZE
mywebapi     latest    2c4c9ace0dbc   2 minutes ago   216MB
busybox      latest    a416a98b71e2   2 months ago    4.26MB
```

```
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP2/SimpleWebAPI$ docker run -d -p 8080:80 --name mywebapi-container mywe
bapi
23cf3b59a2ff6ee9eb71927059ff807e65ae52d624b857e1ef8ef607a13c3409
```

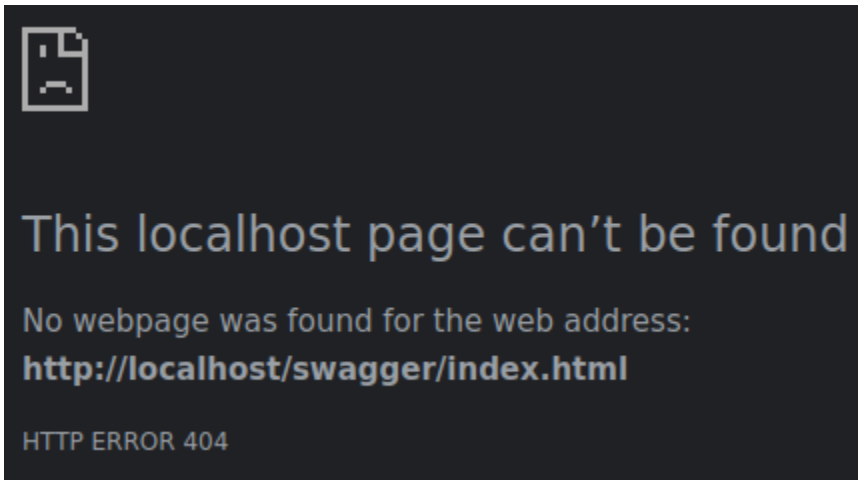
```
tincho@IdeaPad-5-14ALC05:~$ docker tag mywebapi martinlescano/mywebapi
tincho@IdeaPad-5-14ALC05:~$ docker push martinlescano/mywebapi
Using default tag: latest
The push refers to repository [docker.io/martinlescano/mywebapi]
a1881756479c: Pushed
5f70bf18a086: Pushed
1d05f981ce94: Pushed
621ac519e7ac: Pushed
13073704c26c: Pushed
2429ab113701: Pushed
7a17569cab26: Pushed
10764c37bcbc: Pushed
latest: digest: sha256:10aa467ae4c6dd30feb9501bfd7c7e736b7d6f114030ac60a01b4e1c3f665707 size: 1995
tincho@IdeaPad-5-14ALC05:~$
```

8- Publicando puertos



http://localhost/WeatherForecast

[{"date":"2023-10-05","temperatureC":-3,"temperatureF":27,"summary":"Fresquito"},{"date":"2023-10-06","temperatureC":54,"temperatureF":129,"summary":"Caloron"},{"date":"2023-10-07","temperatureC":40,"temperatureF":103,"summary":"Caloron"},{"date":"2023-10-08","temperatureC":12,"temperatureF":53,"summary":"Helado"},{"date":"2023-10-09","temperatureC":22,"temperatureF":71,"summary":"Sweltering"}]




```
tincho@IdeaPad-5-14ALC05:~$ docker run --name myapi -d mywebapi
cb165c661282c34a72cc918c042b04fea30c639c7c7cd63a3e9fc302da972e2d
tincho@IdeaPad-5-14ALC05:~$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED          STATUS          PORTS
       NAMES
cb165c661282   mywebapi   "dotnet SimpleWebAPI…"   13 seconds ago   Up 13 seconds   80/tcp, 443/tcp, 5254
/tcp   myapi
tincho@IdeaPad-5-14ALC05:~$ docker kill myapi
myapi
tincho@IdeaPad-5-14ALC05:~$ docker rm myapi
myapi
tincho@IdeaPad-5-14ALC05:~$ docker run --name myapi -d -p 80:80 -p 5254:5254 mywebapi
97ae7eed74ba112b3579fa8239dfe348e3518379bf752b27537ac79801efa9ec
```

Al acceder a http://localhost/WeatherForecast, pude ver la página de pronóstico del tiempo de la aplicación web. Esto funcionó correctamente debido al mapeo del puerto 80 en el contenedor al puerto 80 de mi sistema host.

Sin embargo, al intentar acceder a http://localhost/swagger/index.html, experimenté un problema y no pude ver la interfaz de Swagger.

9- Modificar Dockerfile para soportar bash

```
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP2/SimpleWebAPI$ docker build -t mywebap
i .
[+] Building 9.0s (18/18) FINISHED                                                docker:desktop-linux
 => [internal] load .dockerignore                                                                  0.0s
 => => transferring context: 2B                                                                    0.0s
 => [internal] load build definition from Dockerfile                                               0.0s
 => => transferring dockerfile: 812B                                                               0.0s
 => [internal] load metadata for mcr.microsoft.com/dotnet/sdk:7.0                                  0.8s
 => [internal] load metadata for mcr.microsoft.com/dotnet/aspnet:7.0                               1.1s
 => [base 1/2] FROM mcr.microsoft.com/dotnet/aspnet:7.0@sha256:933ae169296fe093776749cc47fd60e0a6e  0.0s
 => [build 1/7] FROM mcr.microsoft.com/dotnet/sdk:7.0@sha256:2dd6fa19392967b26d59228af0ec481c652b9  0.0s
 => [internal] load build context                                                                  0.0s
 => => transferring context: 3.54kB                                                                0.0s
 => CACHED [build 2/7] WORKDIR /src                                                                0.0s
 => CACHED [build 3/7] COPY [SimpleWebAPI/SimpleWebAPI.csproj, SimpleWebAPI/]                      0.0s
 => CACHED [build 4/7] RUN dotnet restore "SimpleWebAPI/SimpleWebAPI.csproj"                       0.0s
 => [build 5/7] COPY . .                                                                           0.1s
 => [build 6/7] WORKDIR /src/SimpleWebAPI                                                          0.0s
 => [build 7/7] RUN dotnet build "SimpleWebAPI.csproj" -c Release -o /app/build                    5.2s
 => [publish 1/1] RUN dotnet publish "SimpleWebAPI.csproj" -c Release -o /app/publish /p:UseAppHos  2.4s
 => CACHED [base 2/2] WORKDIR /app                                                                 0.0s
 => CACHED [final 1/2] WORKDIR /app                                                                0.0s
 => CACHED [final 2/2] COPY --from=publish /app/publish .                                          0.0s
 => exporting to image                                                                             0.0s
 => => exporting layers                                                                            0.0s
 => => writing image sha256:800cde7f1dbc7ef3b68cc644f4459f6001859d080940882b223f09e66371f4dc       0.0s
 => => naming to docker.io/library/mywebapi                                                        0.0s
```

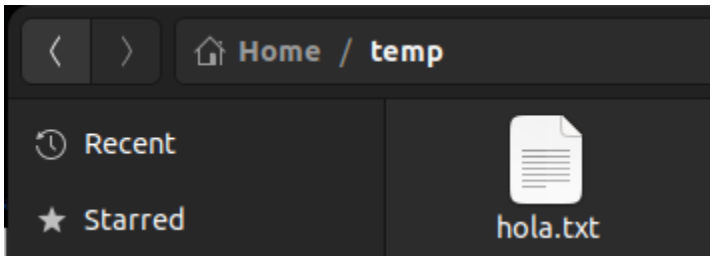
```
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP2/SimpleWebAPI$ docker run -it --rm -p
8080:80 mywebapi
root@e226981ad2d5:/app# dotnet SimpleWebAPI.dll
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://[::]:80
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Production
info: Microsoft.Hosting.Lifetime[0]
      Content root path: /app
```


```
[{"date":"2023-10-05","temperatureC":49,"temperatureF":120,"summary":"Calido"},{"date":"2023-10-
06","temperatureC":54,"temperatureF":129,"summary":"Mild"},{"date":"2023-10-
07","temperatureC":-10,"temperatureF":15,"summary":"Caloron"},{"date":"2023-10-
08","temperatureC":41,"temperatureF":105,"summary":"Sweltering"},{"date":"2023-10-
09","temperatureC":10,"temperatureF":49,"summary":"Caloron"}]
```

```
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP2/SimpleWebAPI$ docker run -it --rm -p
8080:80 mywebapi
root@e226981ad2d5:/app# dotnet SimpleWebAPI.dll
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://[::]:80
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Production
info: Microsoft.Hosting.Lifetime[0]
      Content root path: /app
warn: Microsoft.AspNetCore.HttpsPolicy.HttpsRedirectionMiddleware[3]
      Failed to determine the https port for redirect.

exit



^Cinfo: Microsoft.Hosting.Lifetime[0]
      Application is shutting down...
root@e226981ad2d5:/app# exit
exit
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP2/SimpleWebAPI$
```

10- Montando volúmenes

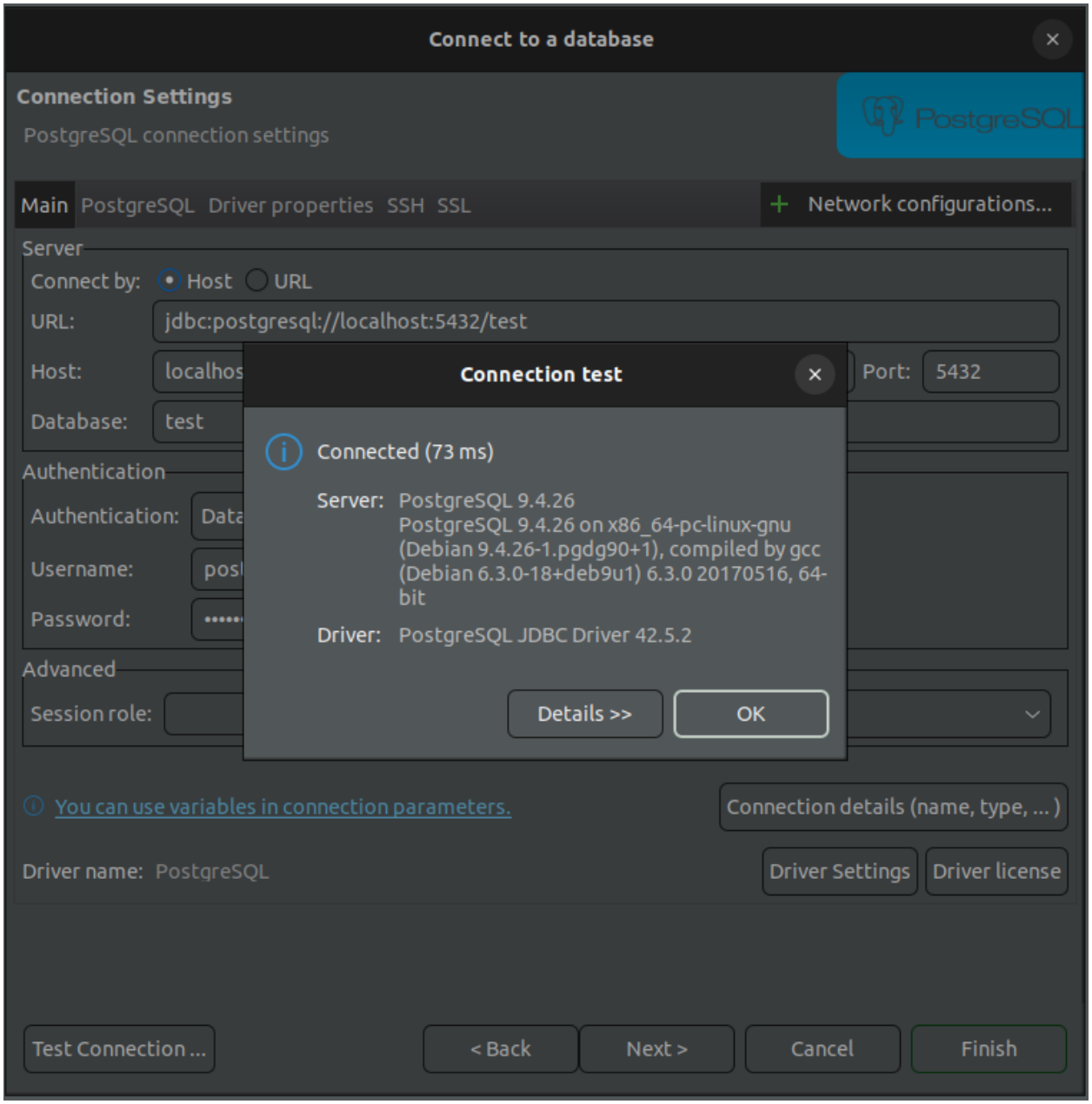
```
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP2/SimpleWebAPI$ docker run -it --rm -p
80:80 -v /home/tincho/temp:/var/temp mywebapi
root@67395fa5b422:/app# ls -l /var/temp
total 0
root@67395fa5b422:/app# touch /var/temp/hola.txt
root@67395fa5b422:/app#
```

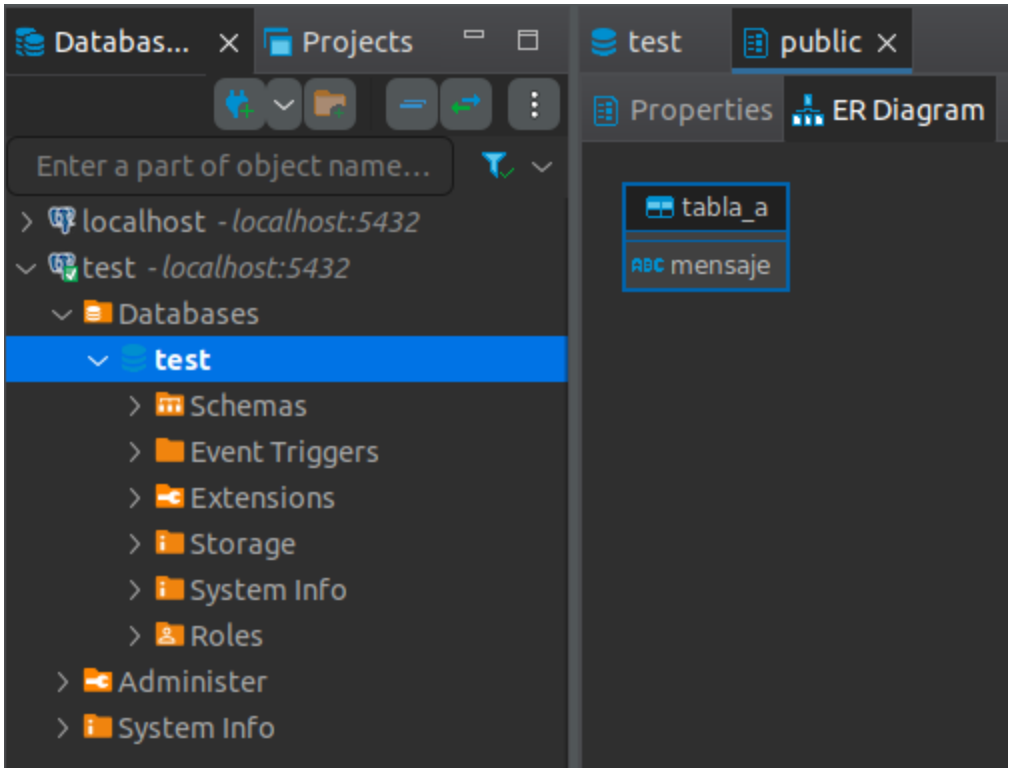


11- Utilizando una base de datos

```
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP2/SimpleWebAPI$ mkdir $HOME/.postgres
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP2/SimpleWebAPI$ docker run --name my-po
stgres -e POSTGRES_PASSWORD=mysecretpassword -v $HOME/.postgres:/var/lib/postgresql/data -p 5432:5432 -d
postgres:9.4
Unable to find image 'postgres:9.4' locally
9.4: Pulling from library/postgres
619014d83c02: Pull complete
7ec0fe6664f6: Pull complete
9ca7ba8f7764: Pull complete
9e1155d037e2: Pull complete
febcfb7f8870: Pull complete
8c78c79412b5: Pull complete
5a35744405c5: Pull complete
27717922e067: Pull complete
36f0c5255550: Pull complete
dbf0a396f422: Pull complete
ec4c06ea33e5: Pull complete
e8dd33eba6d1: Pull complete
51c81b3b2c20: Pull complete
2a03dd76f5d7: Pull complete
Digest: sha256:42a7a6a647a602efa9592edd1f56359800d079b93fa52c5d92244c58ac4a2ab9
Status: Downloaded newer image for postgres:9.4
86bf1e9bbfe7841b3b01c863de108321d1e2201fbe4dff8634e689447e425ce1
```

```
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP2/SimpleWebAPI$ docker exec -it my-post
gres /bin/bash
root@86bf1e9bbfe7:/# psql -h localhost -U postgres
psql (9.4.26)
Type "help" for help.

postgres=# \l
                                  List of databases
   Name    |  Owner   | Encoding |  Collate   |   Ctype    |   Access privileges
-----------+----------+----------+------------+------------+-----------------------
 postgres  | postgres | UTF8     | en_US.utf8 | en_US.utf8 |
 template0 | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres          +
           |          |          |            |            | postgres=CTc/postgres
 template1 | postgres | UTF8     | en_US.utf8 | en_US.utf8 | =c/postgres          +
           |          |          |            |            | postgres=CTc/postgres
(3 rows)

postgres=# create database test;
CREATE DATABASE
postgres=# \connect test
You are now connected to database "test" as user "postgres".
test=# create table tabla_a (mensaje varchar(50));
CREATE TABLE
test=# insert into tabla_a (mensaje) values('Hola mundo!');
INSERT 0 1
test=# select * from tabla_a;
   mensaje
-------------
 Hola mundo!
(1 row)

test=# \q
root@86bf1e9bbfe7:/# exit
exit
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP2/SimpleWebAPI$ cd
tincho@IdeaPad-5-14ALC05:~$ cd Downloads/
tincho@IdeaPad-5-14ALC05:~/Downloads$ sudo dpkg -i dbeaver-ce_23.2.1_amd64.deb
[sudo] password for tincho:
Selecting previously unselected package dbeaver-ce.
(Reading database ... 201691 files and directories currently installed.)
Preparing to unpack dbeaver-ce_23.2.1_amd64.deb ...
Unpacking dbeaver-ce (23.2.1) ...
Setting up dbeaver-ce (23.2.1) ...
Processing triggers for mailcap (3.70+nmu1ubuntu1) ...
Processing triggers for gnome-menus (3.36.0-1ubuntu3) ...
Processing triggers for desktop-file-utils (0.26-1ubuntu3) ...
```


Connect to a database
Connection Settings
PostgreSQL connection settings
PostgreSQL
Main PostgreSQL Driver properties SSH SSL
Network configurations...
Server
Connect by: Host URL
URL: jdbc:postgresql://localhost:5432/test
Host: localhos
Port: 5432
Database: test
Authentication
Authentication: Data
Username: posl
Password:
Advanced
Session role:
Connection test
Connected (73 ms)
Server: PostgreSQL 9.4.26
PostgreSQL 9.4.26 on x86_64-pc-linux-gnu (Debian 9.4.26-1.pgdg90+1), compiled by gcc (Debian 6.3.0-18+deb9u1) 6.3.0 20170516, 64-bit
Driver: PostgreSQL JDBC Driver 42.5.2
Details >>
OK
You can use variables in connection parameters.
Connection details (name, type, ...)
Driver name: PostgreSQL
Driver Settings
Driver license
Test Connection ...
< Back
Next >
Cancel
Finish



Con el comando docker run:

- Inicie un contenedor Docker de PostgreSQL.
- Establecí una contraseña para la base de datos PostgreSQL dentro del contenedor.
- Configure un volumen para que los datos de PostgreSQL se guarden en mi sistema host, lo que significa que los datos persisten incluso después de que el contenedor se detenga o elimine.
- Exponer el puerto 5432 del contenedor al puerto 5432 de mi sistema host, lo que me permitió acceder a la base de datos desde fuera del contenedor.

Con el comando docker exec:

- Ingrese al contenedor de PostgreSQL en modo interactivo.
- Utilicé la herramienta psql para conectarme a la base de datos PostgreSQL dentro del contenedor.
- Ejecute comandos SQL para crear una nueva base de datos, crear una tabla y realizar operaciones en la tabla.
- Luego, salí del contenedor y volví al entorno de mi sistema host.

12- Hacer el punto 11 con Microsoft SQL Server

```
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP2/SimpleWebAPI$ docker pull mcr.microso
ft.com/mssql/server
Using default tag: latest
latest: Pulling from mssql/server
e481c36a257f: Pull complete
7d8f784510c6: Pull complete
c6ac2a806f6c: Pull complete
1b1611617f54: Pull complete
Digest: sha256:d00c97dedf4f5c785bbbd5894f490d50124ff409d7751fbba2d53eb7de9965da
Status: Downloaded newer image for mcr.microsoft.com/mssql/server:latest
mcr.microsoft.com/mssql/server:latest

What's Next?
  View a summary of image vulnerabilities and recommendations → docker scout quickview mcr.microsoft.com/
mssql/server
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP2/SimpleWebAPI$ docker run -e "ACCEPT_E
ULA=Y" -e "SA_PASSWORD=Lescano2015" -p 1433:1433 --name mi-sql-server -d mcr.microsoft.com/mssql/server
9290aa50e79dff324c22ecc77106bdc4ea778310a6dc716d3b9a634d0f0fdb4f
tincho@IdeaPad-5-14ALC05:~/Documents/Ingenieria de Software III/TP2/SimpleWebAPI$ docker logs mi-sql-serv
er
SQL Server 2022 will run as non-root by default.
This container is running as user mssql.
To learn more visit https://go.microsoft.com/fwlink/?linkid=2099216.
sqlservr: This program requires a machine with at least 2000 megabytes of memory.
/opt/mssql/bin/sqlservr: This program requires a machine with at least 2000 megabytes of memory.
```