

Práctica: Arquitectura web con Docker usando volumes e bind mounts

Obxectivos	1
Contido inicial.....	2
Parte 1: Git e GitHub	2
Parte 2: Estrutura do despregue.....	9
Frontend	9
Backend	9
Base de datos	11
Parte 3: docker-compose.yml	11
Parte 4: Execución e probas.....	12
Parte 5: Entregable	15
Recomendacións	16
Rúbrica de avaliación da práctica: Docker + Git + Compose	16

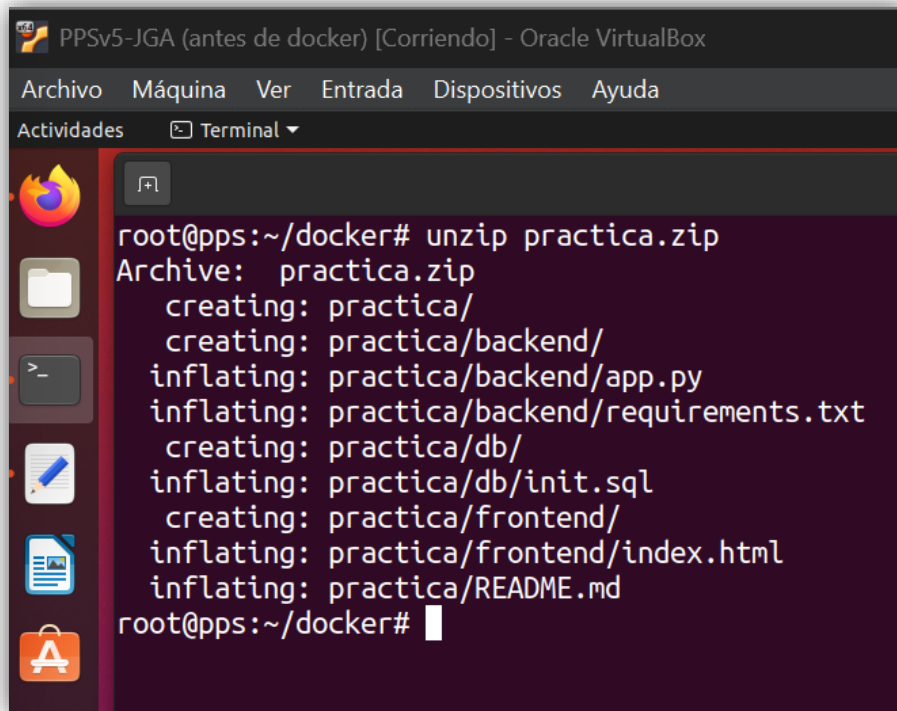
Obxectivos

- Construír e despregar unha arquitectura web básica (frontend + backend + base de datos) con Docker e Docker Compose.
 - Usar **bind mounts** para cargar código fonte directamente desde o host.
 - Empregar un **volume de Docker** para garantir a persistencia dos datos da base de datos.
 - Practicar o control de versións con **Git e GitHub**.
-

Contido inicial

Ao descomprimir o ZIP recibirás:

- Un directorio `frontend/` cunha páxina `index.html`.
- Un directorio `backend/` cun ficheiro Python (`app.py`), un `requirements.txt` e un `Dockerfile`.
- Un directorio `db/` cun script SQL de inicialización (`init.sql`).
- Un `README.md` de referencia.

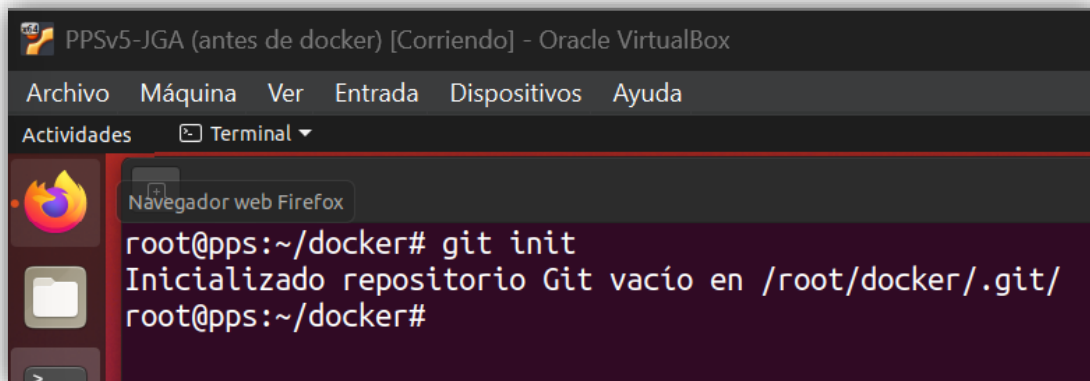


The screenshot shows a terminal window titled "PPSv5-JGA (antes de docker) [Corriendo] - Oracle VirtualBox". The terminal output shows the command `unzip practica.zip` being executed, which creates the directory structure and files for the project.

```
root@pps:~/docker# unzip practica.zip
Archive:  practica.zip
  creating: practica/
  creating: practica/backend/
  inflating: practica/backend/app.py
  inflating: practica/backend/requirements.txt
  creating: practica/db/
  inflating: practica/db/init.sql
  creating: practica/frontend/
  inflating: practica/frontend/index.html
  inflating: practica/README.md
root@pps:~/docker#
```

Parte 1: Git e GitHub

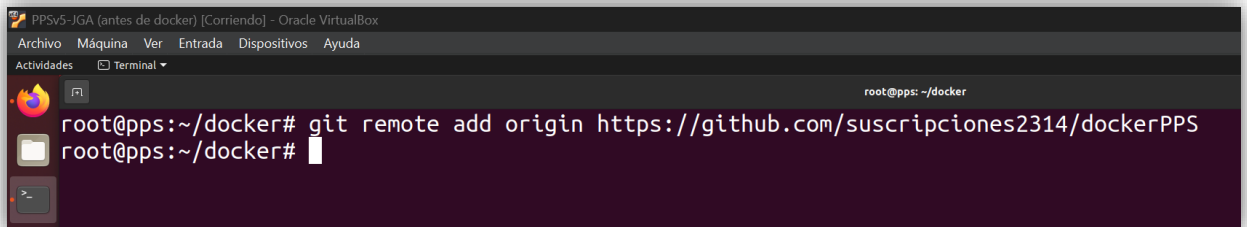
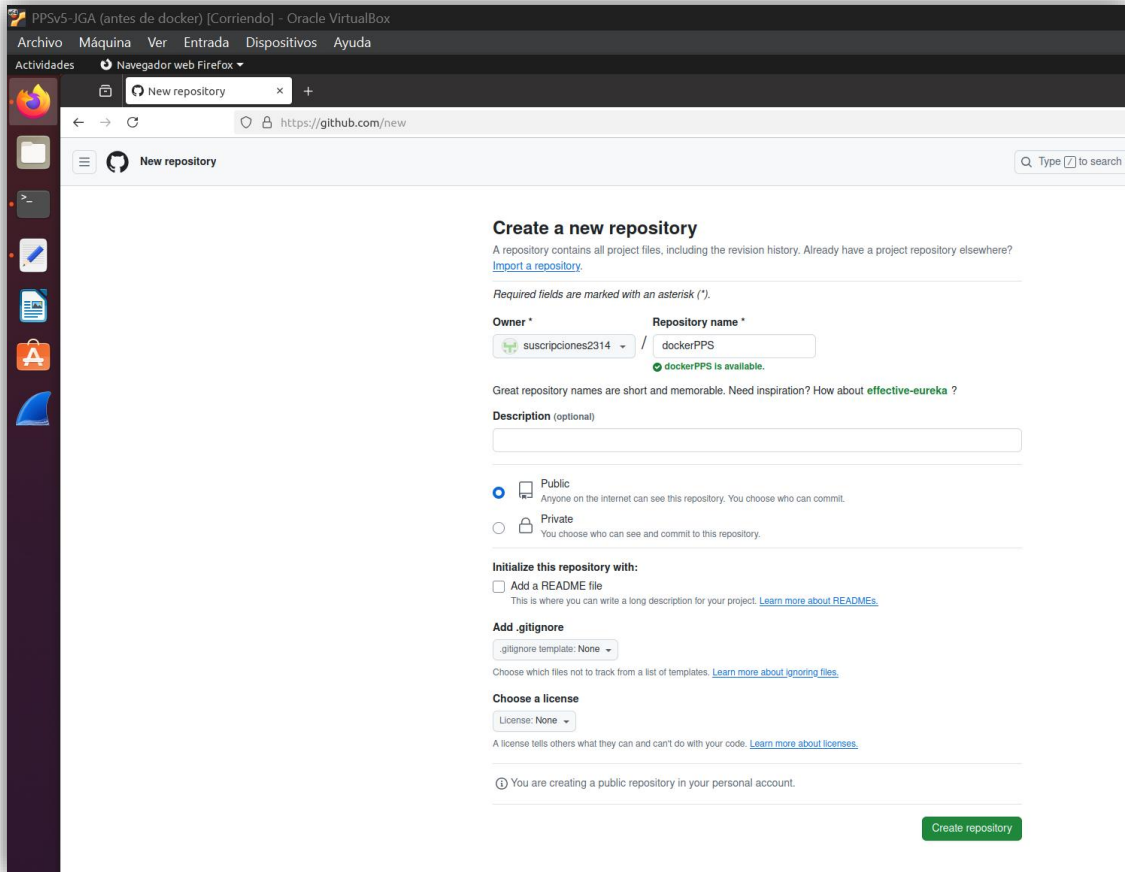
1. Inicializa un repositorio Git local (`git init`).



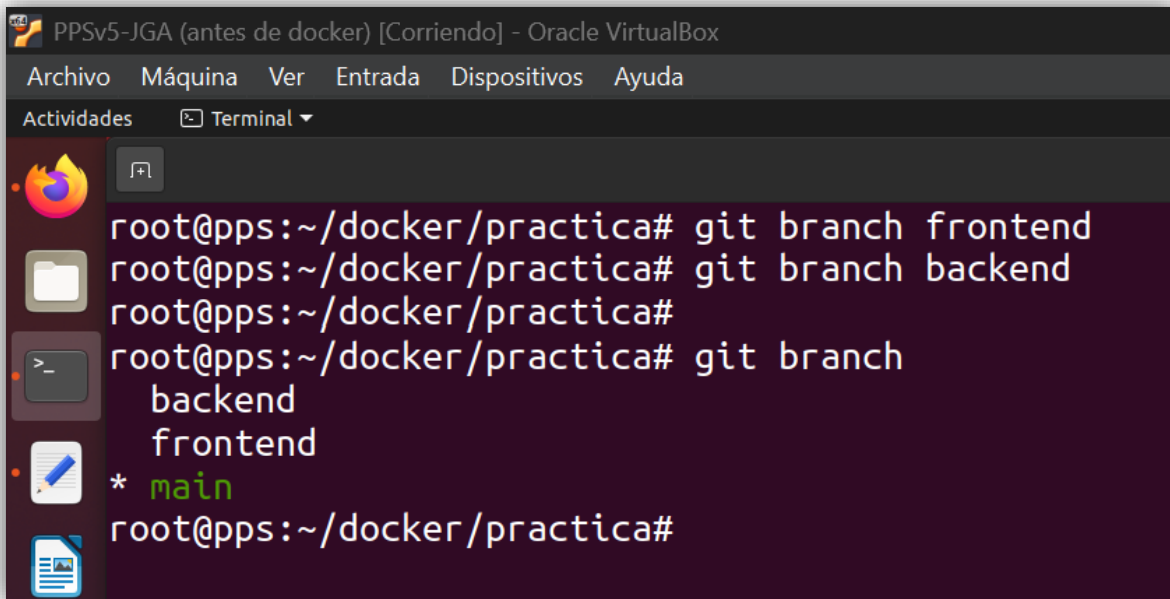
The screenshot shows a terminal window titled "PPSv5-JGA (antes de docker) [Corriendo] - Oracle VirtualBox". The terminal output shows the command `git init` being executed, which initializes a new Git repository in the current directory.

```
root@pps:~/docker# git init
Iniciado repositorio Git vacío en /root/docker/.git/
root@pps:~/docker#
```

2. Crea un repositorio remoto en GitHub e vincúlao.



3. Crea ramas separadas (**frontend**, **backend**) para trabajar de forma organizada.



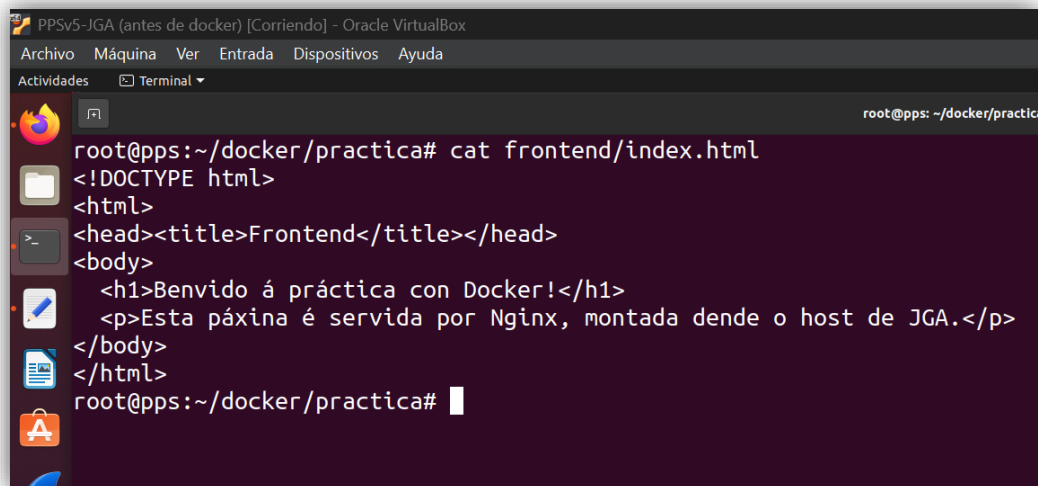
```
PPSv5-JGA (antes de docker) [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Actividades  Terminal
root@pps:~/docker/practica# git branch frontend
root@pps:~/docker/practica# git branch backend
root@pps:~/docker/practica#
root@pps:~/docker/practica# git branch
      backend
      frontend
* main
root@pps:~/docker/practica#
```

4. Fai **commit** dos cambios e realiza **merge** cara á rama principal.

Frontend

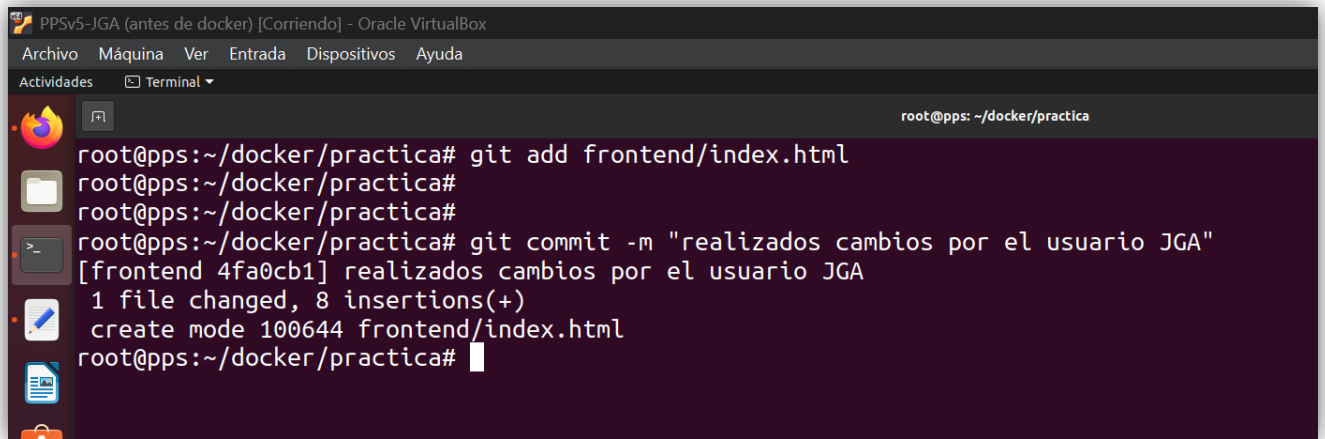
1º git checkout frontend (para cambiar a la rama frontend)

2º Edité el archivo index.html de frontend anteriormente



```
PPSv5-JGA (antes de docker) [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Actividades  Terminal
root@pps:~/docker/practica# cat frontend/index.html
<!DOCTYPE html>
<html>
<head><title>Frontend</title></head>
<body>
  <h1>Benvido á práctica con Docker!</h1>
  <p>Esta páxina é servida por Nginx, montada dende o host de JGA.</p>
</body>
</html>
root@pps:~/docker/practica#
```

3º Hacer el commit

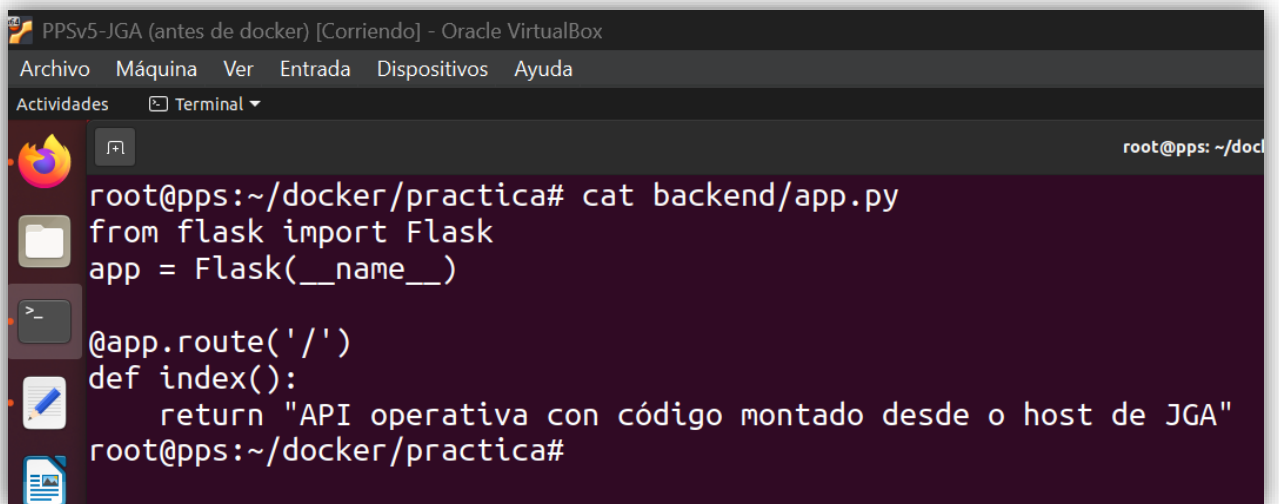


```
PPSV5-JGA (antes de docker) [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Actividades  Terminal
root@pps: ~/docker/practica
root@pps:~/docker/practica# git add frontend/index.html
root@pps:~/docker/practica#
root@pps:~/docker/practica# git commit -m "realizados cambios por el usuario JGA"
[frontend 4fa0cb1] realizados cambios por el usuario JGA
1 file changed, 8 insertions(+)
create mode 100644 frontend/index.html
root@pps:~/docker/practica#
```

Backend

1º git checkout backend

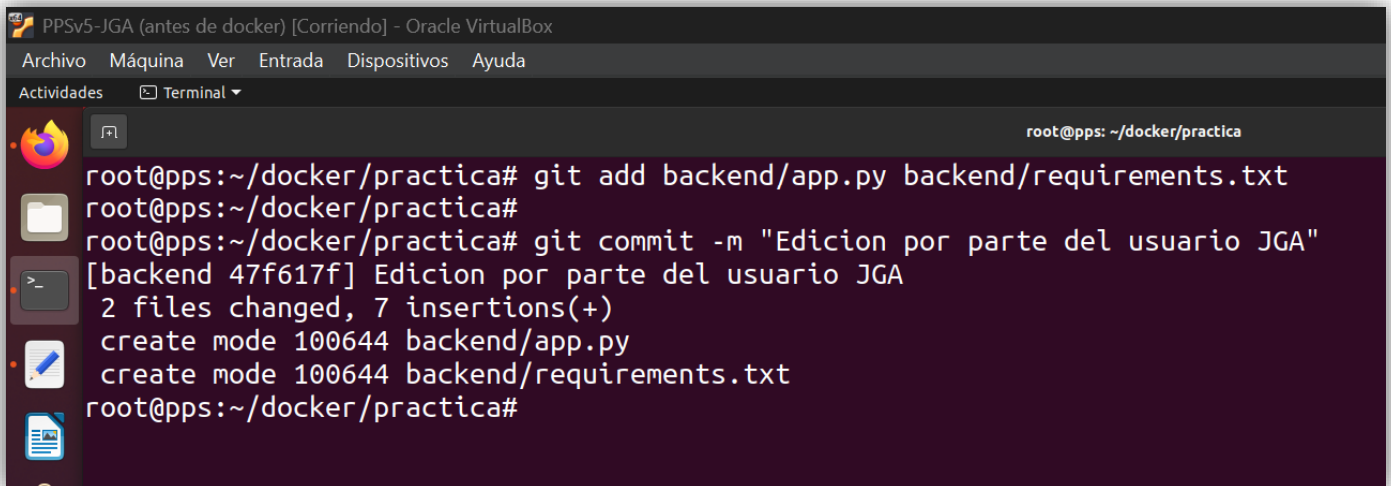
2º editamos el app.py



```
PPSV5-JGA (antes de docker) [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Actividades  Terminal
root@pps: ~/docker/practica
root@pps:~/docker/practica# cat backend/app.py
from flask import Flask
app = Flask(__name__)

@app.route('/')
def index():
    return "API operativa con código montado desde o host de JGA"
root@pps:~/docker/practica#
```

3º Hacer el commit



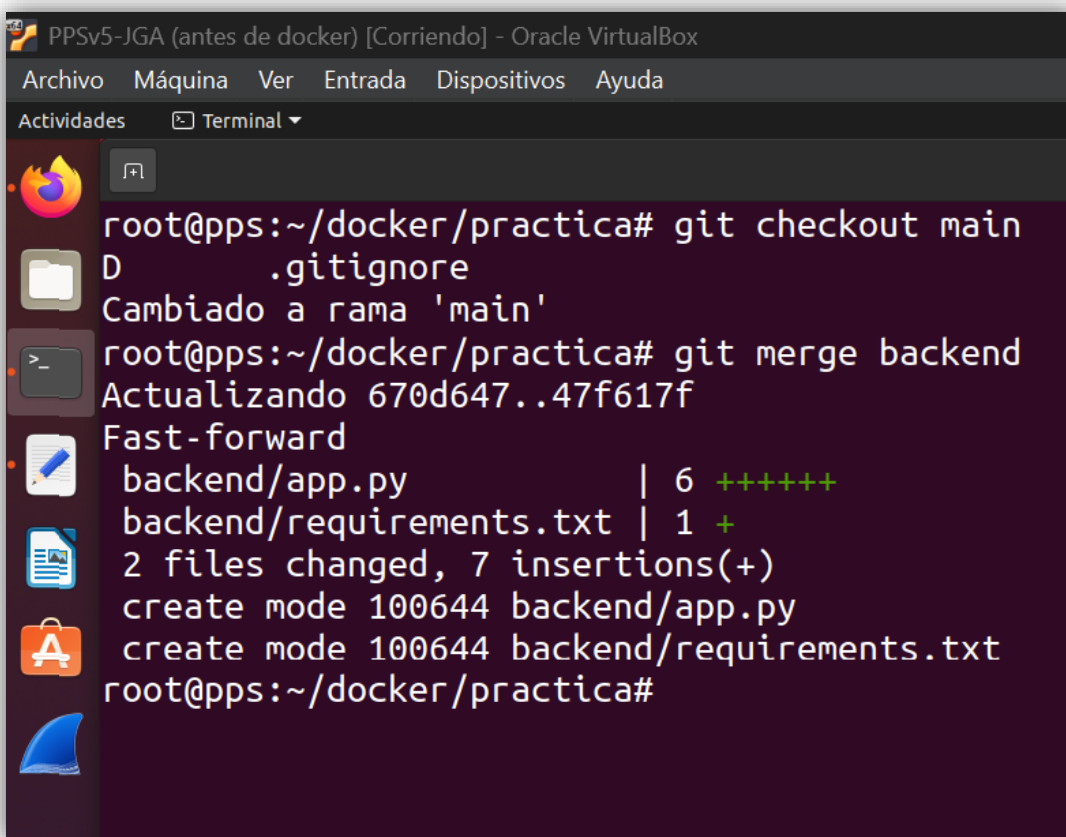
PPSV5-JGA (antes de docker) [Corriendo] - Oracle VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Actividades Terminal

```
root@pps: ~/docker/practica
root@pps:~/docker/practica# git add backend/app.py backend/requirements.txt
root@pps:~/docker/practica#
root@pps:~/docker/practica# git commit -m "Edicion por parte del usuario JGA"
[backend 47f617f] Edicion por parte del usuario JGA
2 files changed, 7 insertions(+)
create mode 100644 backend/app.py
create mode 100644 backend/requirements.txt
root@pps:~/docker/practica#
```

5. Sube o proxecto completo a GitHub.



PPSV5-JGA (antes de docker) [Corriendo] - Oracle VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Actividades Terminal

```
root@pps:~/docker/practica# git checkout main
D      .gitignore
Cambiado a rama 'main'
root@pps:~/docker/practica# git merge backend
Actualizando 670d647..47f617f
Fast-forward
 backend/app.py           | 6 ++++++
 backend/requirements.txt | 1 +
2 files changed, 7 insertions(+)
create mode 100644 backend/app.py
create mode 100644 backend/requirements.txt
root@pps:~/docker/practica#
```

Subiendo frontend

```
PPSV5-JGA (antes de docker) [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal
root@pps: ~/docker/practica
root@pps:~/docker/practica# git remote set-url origin git@github.com:suscripciones2314/dockerPPS.git
root@pps:~/docker/practica# git push origin frontend
Enumerando objetos: 10, listo.
Contando objetos: 100% (10/10), listo.
Compresión delta usando hasta 5 hilos
Comprimiendo objetos: 100% (7/7), listo.
Escribiendo objetos: 100% (10/10), 1.00 KiB | 1.00 MiB/s, listo.
Total 10 (delta 1), reusado 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To github.com:suscripciones2314/dockerPPS.git
 * [new branch]      frontend -> frontend
```

Subiendo backend

```
PPSV5-JGA (antes de docker) [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal
root@pps: ~/docker/practica
root@pps:~/docker/practica# git push origin backend
Warning: Permanently added the ECDSA host key for IP address '140.82.121.3' to the list of known hosts.
Enumerando objetos: 6, listo.
Contando objetos: 100% (6/6), listo.
Compresión delta usando hasta 5 hilos
Comprimiendo objetos: 100% (4/4), listo.
Escribiendo objetos: 100% (5/5), 579 bytes | 579.00 KiB/s, listo.
Total 5 (delta 0), reusado 0 (delta 0)
remote:
remote: Create a pull request for 'backend' on GitHub by visiting:
remote:   https://github.com/suscripciones2314/dockerPPS/pull/new/backend
remote:
To github.com:suscripciones2314/dockerPPS.git
 * [new branch]      backend -> backend
root@pps:~/docker/practica# git push -u origin main
fatal: 'origin' does not appear to be a git repository
fatal: No se pudo leer del repositorio remoto.

Por favor asegúrate de que tengas los permisos de acceso correctos
y que el repositorio exista.
root@pps:~/docker/practica#
root@pps:~/docker/practica# git push origin backend
Everything up-to-date
root@pps:~/docker/practica#
```

Proyecto frontend subido

suscripciones2314 / dockerPPS

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

dockerPPS

Public

Pin

Unwatch 1

backend had recent pushes 10 seconds ago

Compare & pull request

frontend

2 Branches

0 Tags

Go to file

Add file

<> Code

suscripciones2314

realizados cambios por el usuario JGA

4fa0cb1 · 44 minutes ago

2 Commits

db

Primer proyecto

1 hour ago

frontend

realizados cambios por el usuario JGA

44 minutes ago

.gitignore

Primer proyecto

1 hour ago

README.md

Primer proyecto

1 hour ago

README

Proyecto Docker con mounts e volume

Arquitectura web con frontend, backend e base de datos. Incluye persistencia e montaxes de código en tempo real.

Proyecto backend subido

suscripciones2314 / dockerPPS

<> Code

Issues

Pull requests

Actions

Projects

Wiki

Security

Insights

Settings

dockerPPS

Public

Pin

Unwatch 1

backend had recent pushes 2 minutes ago

Compare & pull request

backend

2 Branches

0 Tags

Go to file

Add file

<> Code

This branch is 1 commit ahead of, 1 commit behind frontend.

Contribute

suscripciones2314

Edicion por parte del usuario JGA

47f617f · 37 minutes ago

2 Commits

backend

Edicion por parte del usuario JGA

37 minutes ago

db

Primer proyecto

1 hour ago

.gitignore

Primer proyecto

1 hour ago

README.md

Primer proyecto

1 hour ago

README

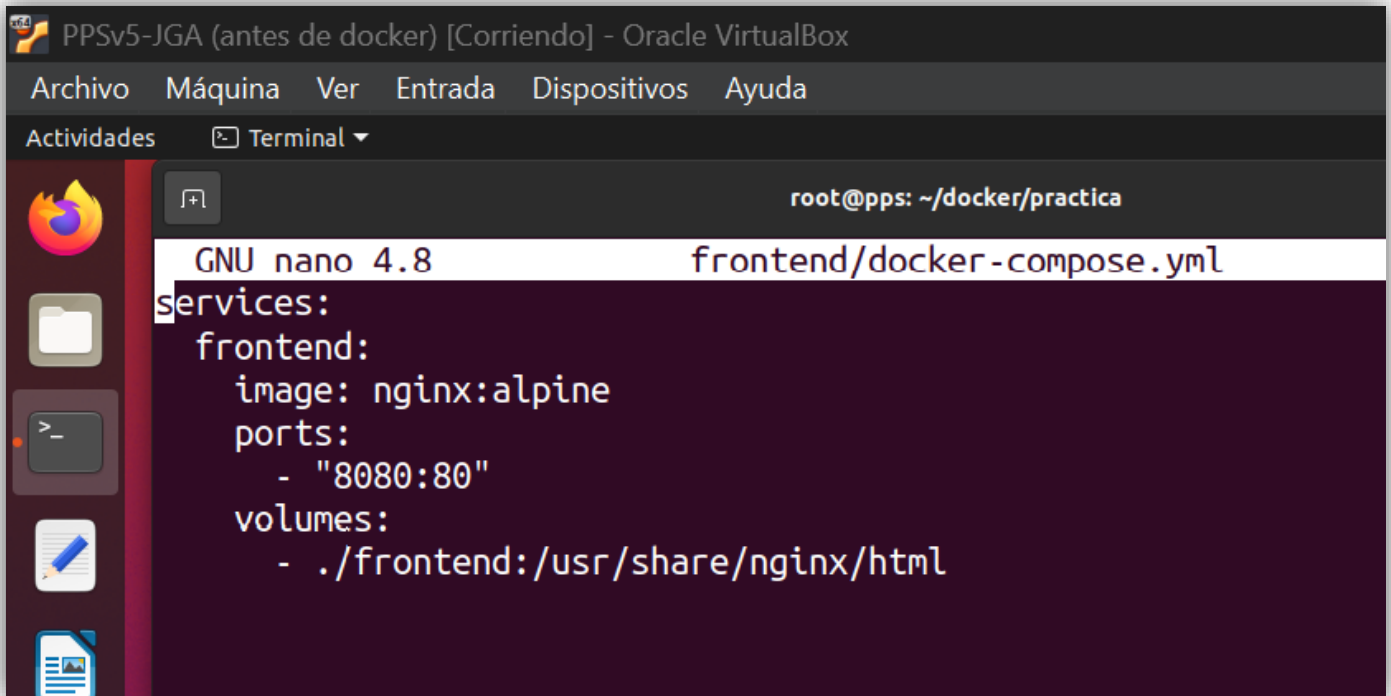
Proyecto Docker con mounts e volume

Arquitectura web con frontend, backend e base de datos. Incluye persistencia e montaxes de código en tempo real.

Parte 2: Estrutura do despregue

Frontend

- Non require un Dockerfile propio.
- Usarás directamente a imaxe `nginx:alpine`.
- Servirá a páxina `index.html` mediante un bind mount dende o host.



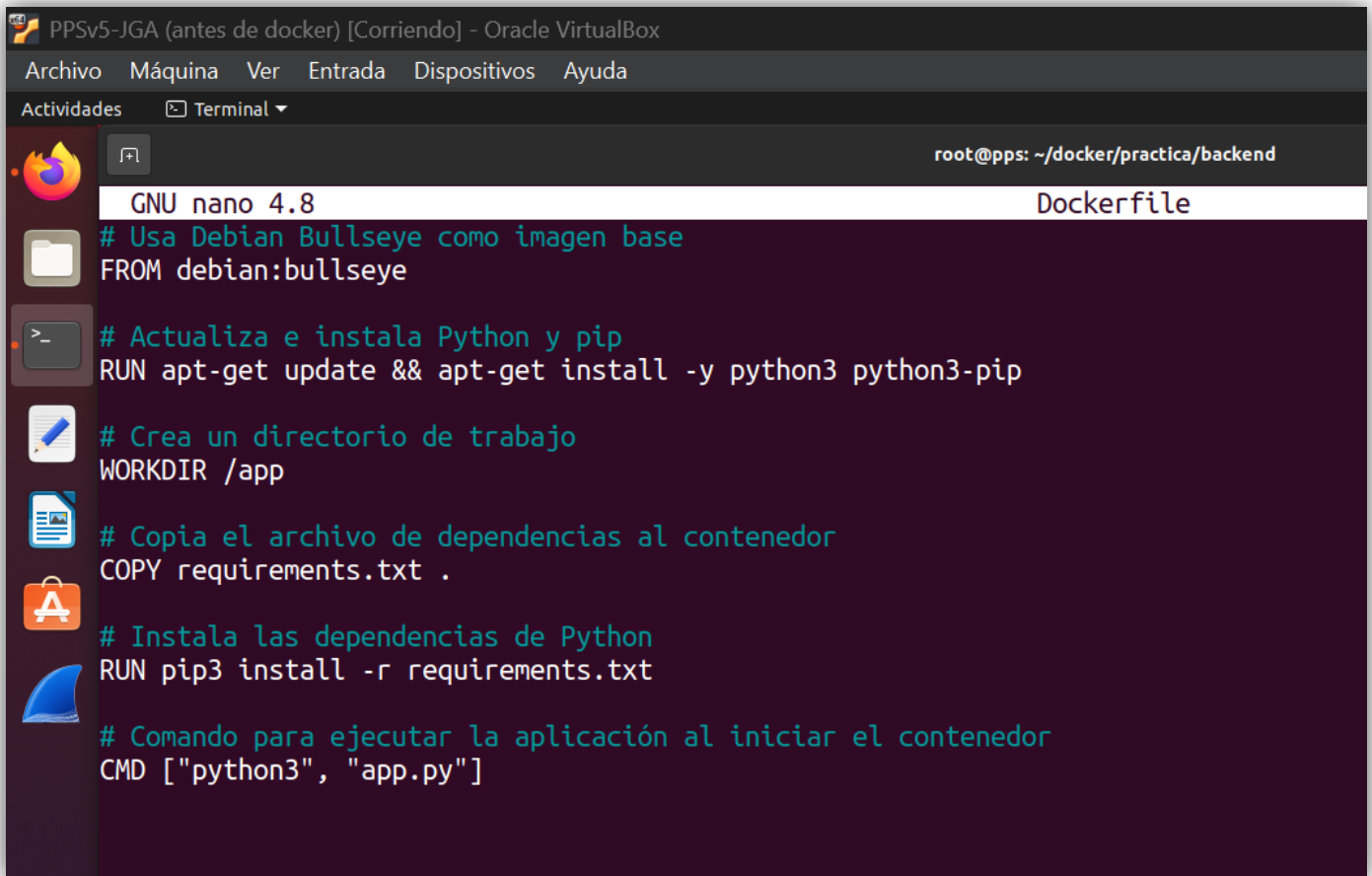
The screenshot shows a terminal window titled "PPSv5-JGA (antes de docker) [Corriendo] - Oracle VirtualBox". The terminal is running the nano text editor, editing the file `frontend/docker-compose.yml`. The content of the file is as follows:

```
services:
  frontend:
    image: nginx:alpine
    ports:
      - "8080:80"
    volumes:
      - ./frontend:/usr/share/nginx/html
```

Backend

- Usa un `Dockerfile` baseado en `debian:bullseye`.
- Instala Python 3 e pip.
- Instala as dependencias de `requirements.txt`.
- Lanza a aplicación `app.py`.
- O código da app será cargado dinamicamente cun bind mount, sen recompilar a imaxe tras cada cambio.

Creando el archivo Dockerfile



```
PPSV5-JGA (antes de docker) [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Actividades  Terminal
root@pps: ~/docker/practica/backend
GNU nano 4.8 Dockerfile
# Usa Debian Bullseye como imagen base
FROM debian:bullseye

# Actualiza e instala Python y pip
RUN apt-get update && apt-get install -y python3 python3-pip

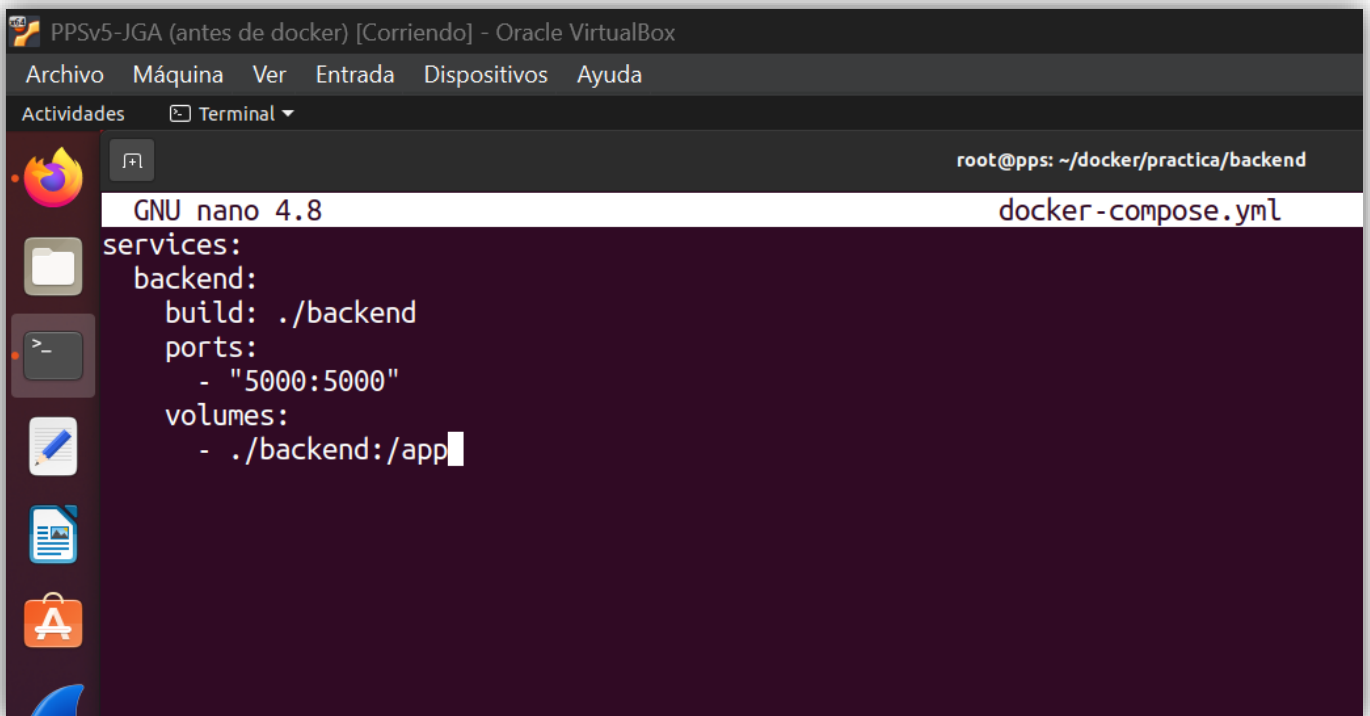
# Crea un directorio de trabajo
WORKDIR /app

# Copia el archivo de dependencias al contenedor
COPY requirements.txt .

# Instala las dependencias de Python
RUN pip3 install -r requirements.txt

# Comando para ejecutar la aplicación al iniciar el contenedor
CMD ["python3", "app.py"]
```

Archivo Docker compose

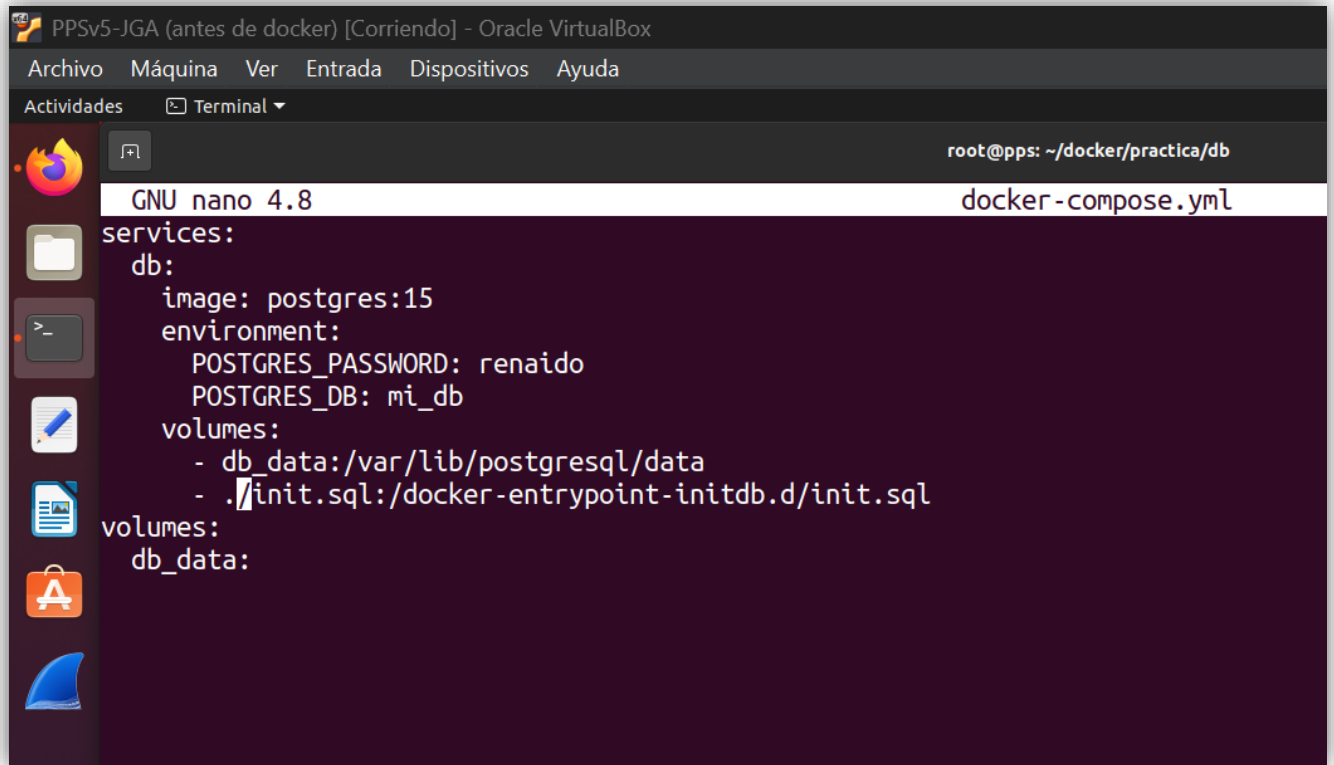


```
PPSV5-JGA (antes de docker) [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Actividades  Terminal
root@pps: ~/docker/practica/backend
GNU nano 4.8 docker-compose.yml
services:
  backend:
    build: ./backend
    ports:
      - "5000:5000"
    volumes:
      - ./backend:/app
```

Base de datos

- Usa a imaxe **postgres:15**.
- Executa un script SQL de inicialización.
- Usa un volume para conservar os datos entre sesións.

Archivo Docker compose



```
PPSV5-JGA (antes de docker) [Corriendo] - Oracle VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Actividades  Terminal
root@pps: ~/docker/practica/db
GNU nano 4.8  docker-compose.yml

services:
  db:
    image: postgres:15
    environment:
      POSTGRES_PASSWORD: renaido
      POSTGRES_DB: mi_db
    volumes:
      - db_data:/var/lib/postgresql/data
      - ./init.sql:/docker-entrypoint-initdb.d/init.sql
volumes:
  db_data:
```

Parte 3: docker-compose.yml

Crea un ficheiro **docker-compose.yml** que:

- Defina os tres servizos (**frontend**, **backend**, **db**).
- Use bind mounts para montar os directorios **./frontend** e **./backend** nos contedores.
- Use un volume chamado **db_data** para **/var/lib/postgresql/data**.

PPSv5-JGA (antes de docker) [Corriendo] - Oracle VirtualBox

Archivo Máquina Ver Entrada Dispositivos Ayuda

Actividades Terminal

root@pps: ~/docker/practica

```
GNU nano 4.8 docker-compose.yml
version: '3.8'

services:
  frontend:
    image: nginx:alpine
    ports:
      - "8080:80"
    volumes:
      - ./frontend:/usr/share/nginx/html:ro

  backend:
    build: ./backend
    ports:
      - "5000:5000"
    volumes:
      - ./backend:/app

  db:
    image: postgres:15
    environment:
      POSTGRES_PASSWORD: renaido
      POSTGRES_DB: mydb
    ports:
      - "5432:5432"
    volumes:
      - db_data:/var/lib/postgresql/data
      - ./db/init.sql:/docker-entrypoint-initdb.d/init.sql:ro

volumes:
  db_data:
```

Parte 4: Ejecución e pruebas

1. Lanza os servizos:

```
docker compose up
```

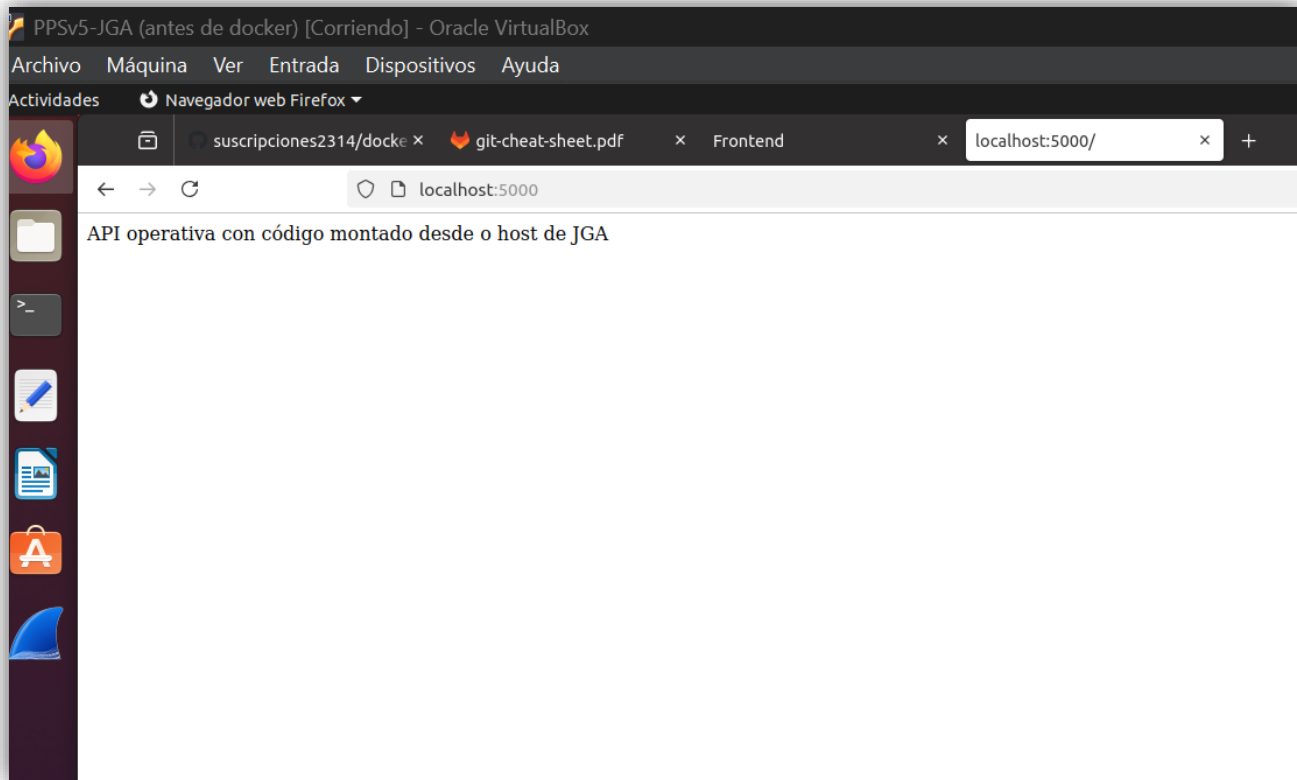
```
PPSV5-JGA (antes de docker) [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Actividades Terminal
root@pps: ~/docker/practica
root@pps:~/docker/practica# docker compose -f docker-compose.yml up -d
WARN[0000] /root/docker/practica/docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 9/9
  ✔ frontend Pulled 6.0s
    ✔ f18232174bc9 Pull complete 1.3s
    ✔ 61ca4f733c80 Pull complete 1.9s
    ✔ b464cfd72a63 Pull complete 2.0s
    ✔ d7e507024086 Pull complete 2.1s
    ✔ 81bd8ed7ec67 Pull complete 2.2s
    ✔ 197eb75867ef Pull complete 2.3s
    ✔ 34a64644b756 Pull complete 2.4s
    ✔ 39c2ddf6010 Pull complete 4.2s
[+] Running 0/0
[+] Running 0/1end Building 0.1s
[+] Building 28.4s (5/9)
=> [backend internal] load build definition from Dockerfile
=> => transferring dockerfile: 482B 0.0s
=> [backend internal] load metadata for docker.io/library/debian:bullseye 1.2s
=> [backend internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [backend 1/5] FROM docker.io/library/debian:bullseye@sha256:cf48c31af360e1c0a0aedd33aae4d928b68c2cdf093f1612650eb1ff434d1c34 13.2s
=> => resolve docker.io/library/debian:bullseye@sha256:cf48c31af360e1c0a0aedd33aae4d928b68c2cdf093f1612650eb1ff434d1c34 0.0s
=> => sha256:cf48c31af360e1c0a0aedd33aae4d928b68c2cdf093f1612650eb1ff434d1c34 4.52kB / 4.52kB 0.0s
=> => sha256:652b211ca872c3392daefabcc7d2f10ac50afdd9560554688334cf5903e0b868 1.02kB / 1.02kB 0.0s
=> => sha256:58db20b609564f247a241eb9fa3a83c55779e7c5d34800c126a1e2a2faf6efcc 453B / 453B 0.0s
=> => sha256:54107f2de180b7b6e9f909d2f1c6c18e10c700a6bd80a035d931768b06bb2905 53.75MB / 53.75MB 4.9s
=> => extracting sha256:54107f2de180b7b6e9f909d2f1c6c18e10c700a6bd80a035d931768b06bb2905 8.1s
=> [backend internal] load build context 0.0s
=> => transferring context: 49B 0.0s
=> [backend 2/5] RUN apt-get update && apt-get install -y python3 python3-pip 13.9s
=> => # Get:126 http://deb.debian.org/debian bullseye/main amd64 python3-distutils all 3.9.2-1 [143 kB]
=> => # Get:127 http://deb.debian.org/debian bullseye/main amd64 python3-dev amd64 3.9.2-3 [24.8 kB]
=> => # Get:128 http://deb.debian.org/debian-security bullseye-security/main amd64 python3-pkg-resources all 52.0.0-4+deb11u1 [190 kB]
=> => # Get:129 http://deb.debian.org/debian-security bullseye-security/main amd64 python3-setuptools all 52.0.0-4+deb11u1 [367 kB]
=> => # Get:130 http://deb.debian.org/debian bullseye/main amd64 python3-wheel all 0.34.2-1 [24.0 kB]
=> => # Get:131 http://deb.debian.org/debian bullseye/main amd64 python3-pip all 20.3.4-4+deb11u1 [337 kB]
```

2. Accede no navegador:

🔗 <http://localhost:8080> → parte frontend.



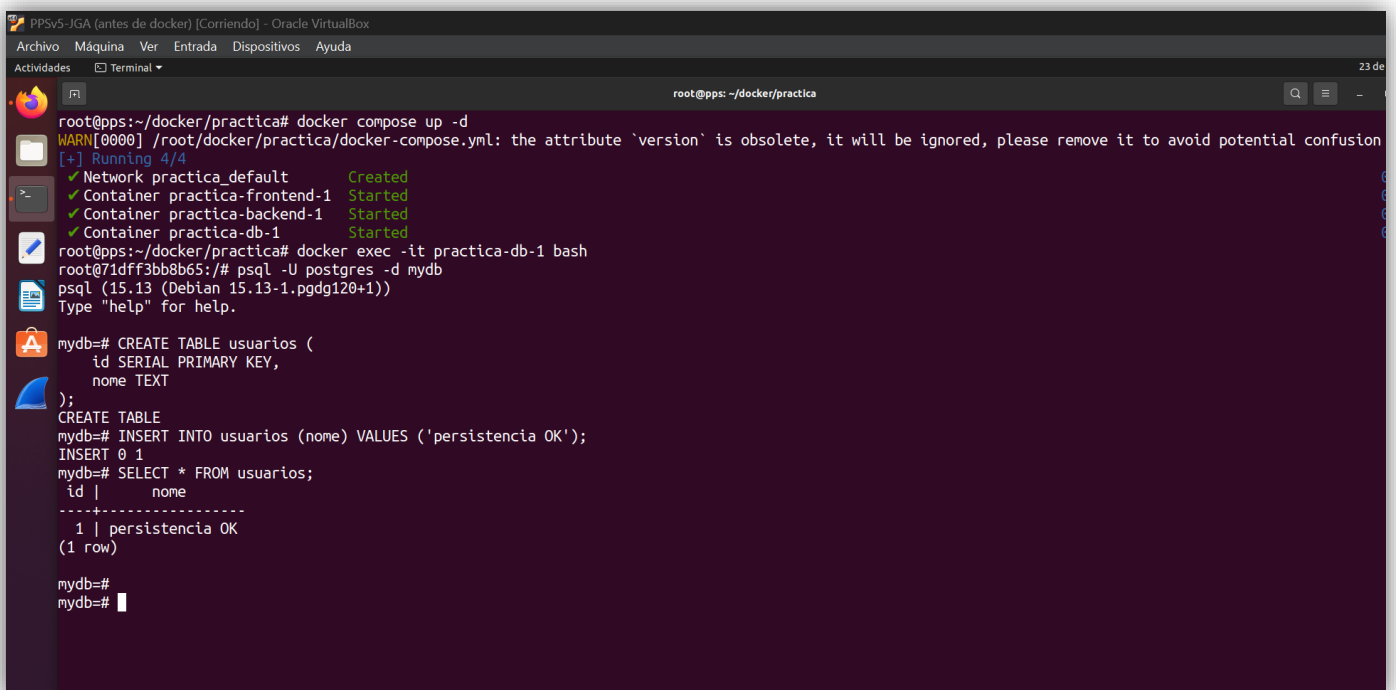
- <http://localhost:5000> → backend (respuesta tipo "API operativa...").



3. Apaga os contedores:

```
docker compose down
```

4. Verifica que os datos da base de datos persisten tras reiniciar os servicios.



```
root@pps:~/docker/practica# docker compose down
WARN[0000] /root/docker/practica/docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 4/4
✔ Container practica-db-1      Removed      0.4s
✔ Container practica-frontend-1 Removed      0.4s
✔ Container practica-backend-1 Removed     10.3s
✔ Network practica_default     Removed      0.1s
root@pps:~/docker/practica# docker compose up -d
WARN[0000] /root/docker/practica/docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 4/4
✔ Network practica_default     Created       0.1s
✔ Container practica-db-1      Started      2.1s
✔ Container practica-frontend-1 Started      2.1s
✔ Container practica-backend-1 Started      0.6s
root@pps:~/docker/practica# docker exec -it practica-db-1 bash
root@dad851679fa6:/# psql -U postgres -d mydb
psql (15.13 (Debian 15.13-1.pgdg120+1))
Type "help" for help.

mydb=# SELECT * FROM usuarios;
 id |      nome
-----+-----
  1 | persistencia OK
(1 row)

mydb=#
```

Parte 5: Entregable

1. Subir a GitHub:

- `docker-compose.yml`
- Dockerfile do backend
- Código HTML e Python
- Script SQL
- README con instrucciones

2. Incluir no README:

- Breve descripción do proxecto
- Composición dos servizos
- Comandos principais para executar o proxecto

3. Entregar un arquivo de texto coa URL do repositorio en *GitHub*

Recomendacións

- Fai **commit** frecuentes e con mensaxes claros.
- Usa ramas e merges de maneira organizada.
- Fai push de cada rama a GitHub

Rúbrica de avaliación da práctica: Docker + Git + Compose

Cada criterio puntúase sobre 2 puntos, podendo acadar un máximo de 6 puntos totais.

Criterio	Baixo (0,75 puntos)	Medio (1,5 puntos)	Alto (2 puntos)
Uso de Git e GitHub	O repositorio é incompleto ou con poucos commits . Sen ramas nin boas prácticas.	Inclúe commits e estrutura razoable. Uso básico de ramas ou mensaxes de commit claras.	Uso correcto de ramas, commits frecuentes e significativos, estrutura clara, repo publicado en GitHub.
Creación de imaxes Docker	A imaxe funciona con erros ou mal estruturada. Inclúe paquetes innecesarios ou malas prácticas básicas.	A imaxe funciona ben pero con algúns detalles mellorables: limpeza, separación de capas, permisos, etc.	Imaxe funcional, clara, segura e optimizada : imaxe base adecuada, non executa como root, instala só o necesario, sen secrets nin lixo.
Composición con Docker Compose	O ficheiro docker-compose.yml é incompleto ou contén erros aínda que desprega os servizos requiridos.	Composición funcional pero con limitacións (falta de volumes, dependencias mal definidas...).	Composición completa, funcional, con bind mounts , volumes e relacións ben establecidas entre servizos.