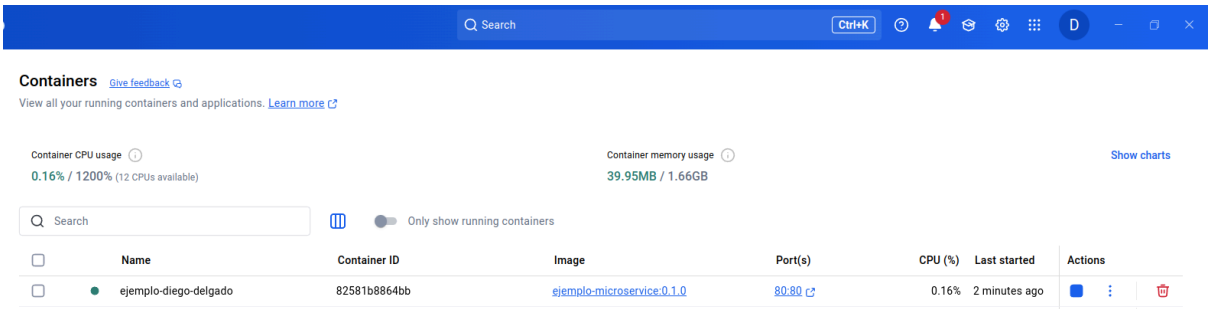
ACTIVIDAD 19

1. Construir la imagen

```
diegodev@HPavilion:~/Desktop/Computer-Science/ds-practice/Laboratorio10$ docker build -t ejemplo-microservice:0.1.0 .
[+] Building 27.1s (15/15) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 1.16kB
 => [internal] load metadata for docker.io/library/python:3.12-slim
 => [auth] library/python:pull token for registry-1.docker.io
 => [internal] load .dockerignore
 => => transferring context: 257B
 => [internal] load build context
 => => transferring context: 32.43kB
 => [builder 1/4] FROM docker.io/library/python:3.12-slim@sha256:b43ff04d5df04ad5cabb80890b7ef74e8410e3395b19af970dcd52d7a4bff921
 => => resolve docker.io/library/python:3.12-slim@sha256:b43ff04d5df04ad5cabb80890b7ef74e8410e3395b19af970dcd52d7a4bff921
 => => sha256:b7ba6d2a1fc72f9587288b3b60221d1b07dae4f61a41360d2cc281fe7c007b3a 250B / 250B
 => => sha256:0674d14a155c94f13e648265aa3ba62410e1fe0387fef64cc608388e54232880 12.11MB / 12.11MB
 => => sha256:490b9a1c25e472ab1cceab4ed6f3bd252006f7c6eea0ef7ba845c996adb6f302 1.29MB / 1.29MB
 => => sha256:0e4bc2bd6656e6e004e3c749af70e5650bac2258243eb0949dea51cb8b7863db 29.78MB / 29.78MB
 => => extracting sha256:0e4bc2bd6656e6e004e3c749af70e5650bac2258243eb0949dea51cb8b7863db
 => => extracting sha256:490b9a1c25e472ab1cceab4ed6f3bd252006f7c6eea0ef7ba845c996adb6f302
 => => extracting sha256:0674d14a155c94f13e648265aa3ba62410e1fe0387fef64cc608388e54232880
 => => extracting sha256:b7ba6d2a1fc72f9587288b3b60221d1b07dae4f61a41360d2cc281fe7c007b3a
 => [production 2/6] RUN groupadd -r appuser     && useradd -m -r -g appuser appuser
 => [builder 2/4] WORKDIR /build
 => [builder 3/4] COPY requirements.txt .
 => [builder 4/4] RUN pip install --user --no-cache-dir -r requirements.txt
 => [production 3/6] WORKDIR /app
 => [production 4/6] COPY --from=builder /root/.local /home/appuser/.local
 => [production 5/6] COPY . /app
 => [production 6/6] RUN chown -R appuser:appuser /app
 => exporting to image
 => => exporting layers
 => => exporting manifest sha256:18b642d958e4f073fe01ad9c2340af49298706cc5fb62f9d1e10b3065c7612c5
 => => exporting config sha256:0dab83588e970709ba700fb2f293e5fc85a8f550983c96b1350afa076a12862e
 => => exporting attestation manifest sha256:0eafe2afc723e72fac0827d7a990b84acb70976d1666a18318dcf95404965156
 => => exporting manifest list sha256:7478f37c99df3e2c432f1418e2209d2de83383922483f423b01a0ed82b4c85aa
 => => naming to docker.io/library/ejemplo-microservice:0.1.0
 => => unpacking to docker.io/library/ejemplo-microservice:0.1.0

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/4vmz8gc06rheqqkro1m20umqa
```

2. Arrancar el contenedor

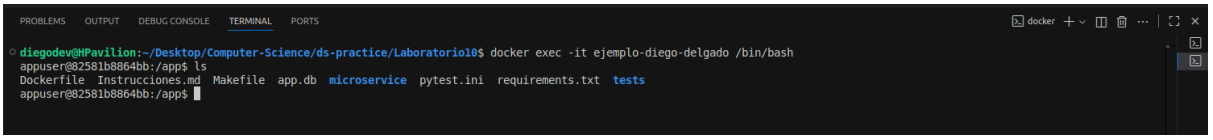
```
diegodev@HPavilion:~/Desktop/Computer-Science/ds-practice/Laboratorio10$ docker run -d \
  --name ejemplo-diego-delgado \
  -p 80:80 \
  ejemplo-microservice:0.1.0
82581b8864bb2968eadd4722224e9bdec7e8f4913940ef63b788e140b920b7c3
diegodev@HPavilion:~/Desktop/Computer-Science/ds-practice/Laboratorio10$
```



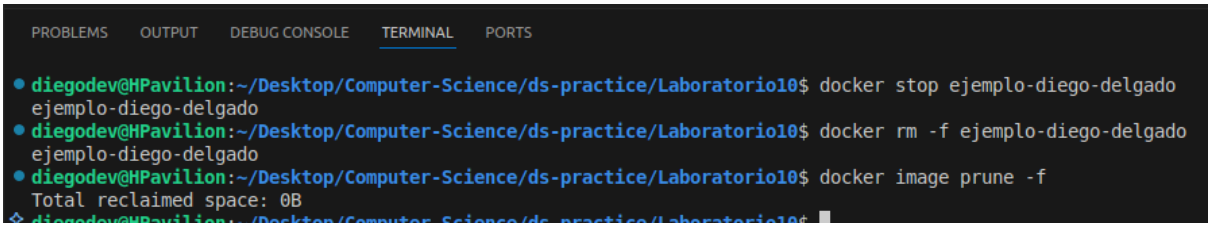
3. Verificar que responde

```
diegodev@HPavilion:~/Desktop/Computer-Science/ds-practice/Laboratorio10$ curl -i http://localhost/api/items/
HTTP/1.1 200 OK
date: Sat, 22 Nov 2025 13:19:47 GMT
server: uvicorn
content-length: 68
content-type: application/json

[{"name":"test-item","description":"Descripción de prueba","id":1}]diegodev@HPavilion:~/Desktop/Computer-Science/ds-practice/Laboratorio10$
```

4. Depurar



5. Detener y limpiar

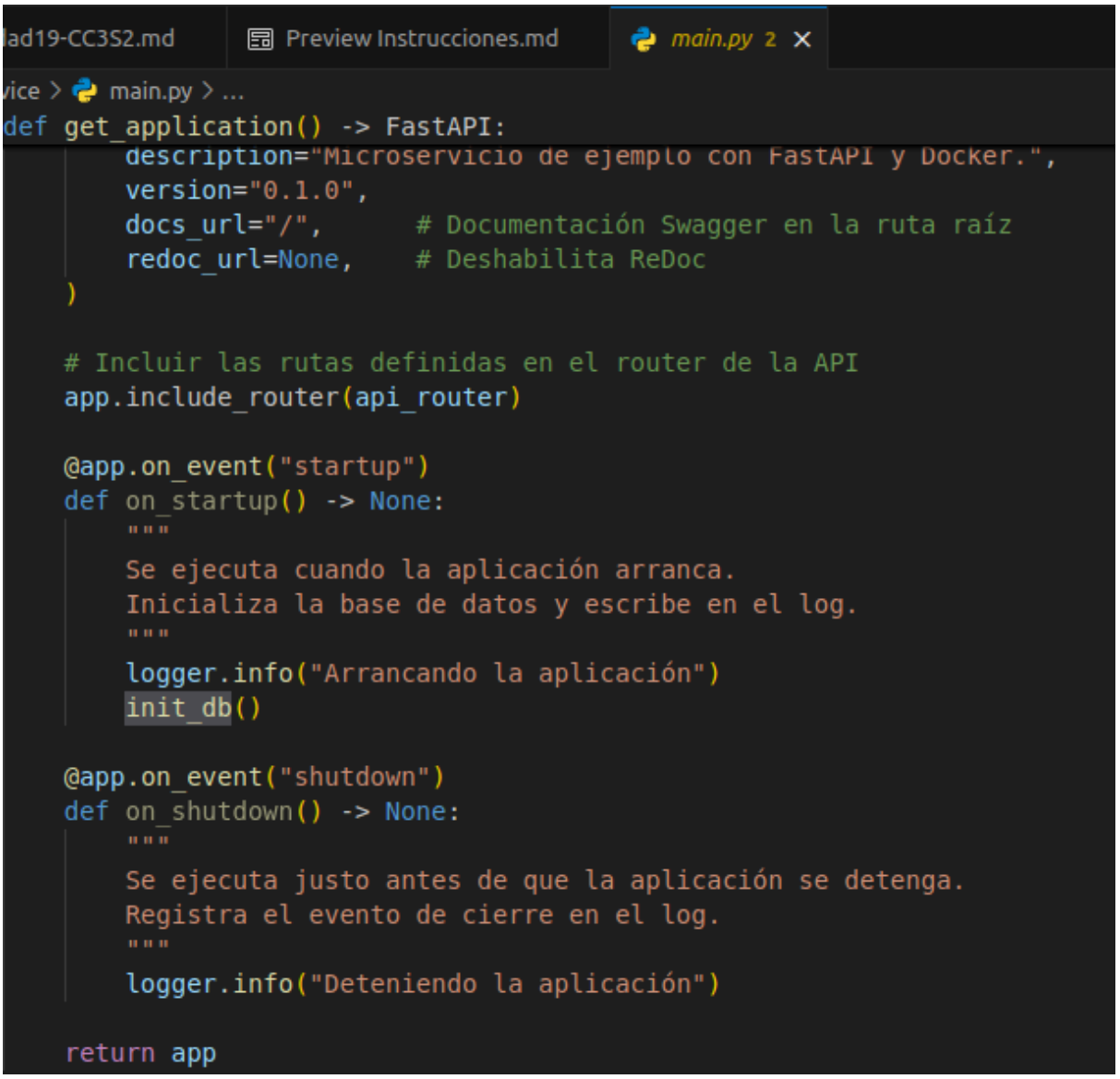


Teniendo en cuenta el laboratorio 10, podemos proseguir con la actividad 22

La base de datos SQLite ya se inicializa correctamente.

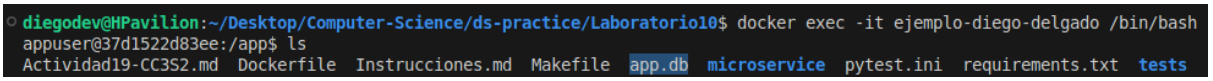
```python
def init_db() -> None:
    """
    Inicializa la base de datos SQLite creando la tabla `items` si no existe todavía.
    """
    logger.info("Inicializando base de datos en %s", DB_PATH)
    with sqlite3.connect(DB_PATH) as conn:
        conn.execute(
            """
            CREATE TABLE IF NOT EXISTS items (
                id INTEGER PRIMARY KEY AUTOINCREMENT,
                name TEXT NOT NULL UNIQUE,
                description TEXT,
                created_at DATETIME DEFAULT CURRENT_TIMESTAMP
            )
            """
        )
        conn.commit()
```

Notamos que FastAPI inicia, ejecuta init_db()

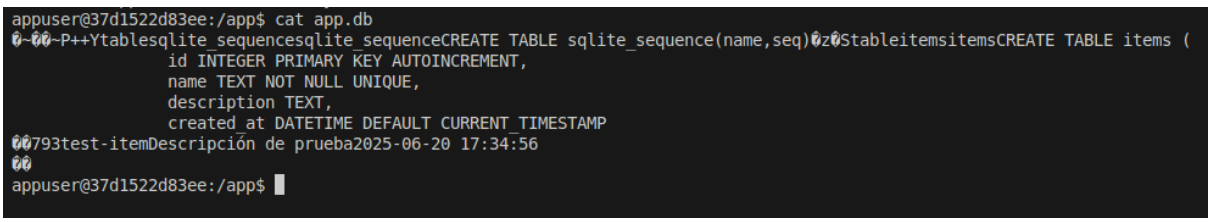


```
lad19-CC3S2.md    Preview Instrucciones.md    main.py 2
vice > main.py > ...
def get_application() -> FastAPI:
        description="Microservicio de ejemplo con FastAPI y Docker.",
        version="0.1.0",
        docs_url="/",       # Documentación Swagger en la ruta raíz
        redoc_url=None,     # Deshabilita ReDoc
    )

    # Incluir las rutas definidas en el router de la API
    app.include_router(api_router)

    @app.on_event("startup")
    def on_startup() -> None:
        """
        Se ejecuta cuando la aplicación arranca.
        Inicializa la base de datos y escribe en el log.
        """
        logger.info("Arrancando la aplicación")
        init_db()

    @app.on_event("shutdown")
    def on_shutdown() -> None:
        """
        Se ejecuta justo antes de que la aplicación se detenga.
        Registra el evento de cierre en el log.
        """
        logger.info("Deteniendo la aplicación")

    return app
```

Ingresamos al contenedor para visualizar la persistencia de la base de datos app.db



```
diegodev@HPavilion:~/Desktop/Computer-Science/ds-practice/Laboratorio10$ docker exec -it ejemplo-diego-delgado /bin/bash
appuser@37d1522d83ee:/app$ ls
Actividad19-CC3S2.md  Dockerfile  Instrucciones.md  Makefile  app.db  microservice  pytest.ini  requirements.txt  tests
```



```
appuser@37d1522d83ee:/app$ cat app.db
SQLite format 3@ -P++Ytablesqlite_sequencesqlite_sequenceCREATE TABLE sqlite_sequence(name,seq)tableitemsitemsCREATE TABLE items (
                id INTEGER PRIMARY KEY AUTOINCREMENT,
                name TEXT NOT NULL UNIQUE,
                description TEXT,
                created_at DATETIME DEFAULT CURRENT_TIMESTAMP
)793test-itemDescripción de prueba2025-06-20 17:34:56
appuser@37d1522d83ee:/app$
```

```
diegodev@HPavilion:~/Desktop/Computer-Science/ds-practice/Laboratorio10$ docker exec -it ejemplo-diego-delgado /bin/bash
>>> import sqlite3
>>> conn = sqlite3.connect('app.db')
>>> cursor = conn.execute("SELECT name FROM sqlite_master WHERE type='table'")
>>> print("Tablas:", cursor.fetchall())
Tablas: [('items',), ('sqlite_sequence',)]
>>> cursor = conn.execute("PRAGMA table_info(items)")
>>> print("\nColumnas de 'items':")

Columnas de 'items':
>>> cursor = conn.execute("SELECT * FROM items")
>>> print("\nDatos:")

Datos:
>>> for row in cursor.fetchall():
...         print(row)
...
(1, 'test-item', 'Descripción de prueba', '2025-06-20 17:34:56')
>>>
```

Finalmente se ejecuta las pruebas con pytest -q

```
appuser@37d1522d83ee:/app$ pytest -q
..                                                                                                                [100%]
============================================================================ warnings summary ============================================================================
../home/appuser/.local/lib/python3.12/site-packages/_pytest/config/__init__.py:1448
  /home/appuser/.local/lib/python3.12/site-packages/_pytest/config/__init__.py:1448: PytestConfigWarning: Unknown config option: python_paths

    self._warn_or_fail_if_strict(f"Unknown config option: {key}\n")

../home/appuser/.local/lib/python3.12/site-packages/pydantic/fields.py:814
../home/appuser/.local/lib/python3.12/site-packages/pydantic/fields.py:814
  /home/appuser/.local/lib/python3.12/site-packages/pydantic/fields.py:814: PydanticDeprecatedSince20: Using extra keyword arguments on `Field` is deprecated and will be removed
. Use `json_schema_extra` instead. (Extra keys: 'example'). Deprecated in Pydantic V2.0 to be removed in V3.0. See Pydantic V2 Migration Guide at https://errors.pydantic.dev/2.8
/migration/
    warn(

microservice/main.py:24
  /app/microservice/main.py:24: DeprecationWarning:
          on_event is deprecated, use lifespan event handlers instead.

          Read more about it in the
          [FastAPI docs for Lifespan Events](https://fastapi.tiangolo.com/advanced/events/).

    @app.on_event("startup")

../home/appuser/.local/lib/python3.12/site-packages/fastapi/applications.py:4495
../home/appuser/.local/lib/python3.12/site-packages/fastapi/applications.py:4495
  /home/appuser/.local/lib/python3.12/site-packages/fastapi/applications.py:4495: DeprecationWarning:
          on_event is deprecated, use lifespan event handlers instead.

          Read more about it in the
          [FastAPI docs for Lifespan Events](https://fastapi.tiangolo.com/advanced/events/).

    return self.router.on_event(event_type)

microservice/main.py:33
  /app/microservice/main.py:33: DeprecationWarning:
          on_event is deprecated, use lifespan event handlers instead.

          Read more about it in the
```

```
diegodev@HPavilion:~/Desktop/Computer-Science/ds-practice/Laboratorio10$ docker exec -it ejemplo-diego-delgado /bin/bash
../home/appuser/.local/lib/python3.12/site-packages/pydantic/fields.py:814
  /home/appuser/.local/lib/python3.12/site-packages/pydantic/fields.py:814: PydanticDeprecatedSince20: Using extra keyword arguments on `Field` is deprecated and will be removed
. Use `json_schema_extra` instead. (Extra keys: 'example'). Deprecated in Pydantic V2.0 to be removed in V3.0. See Pydantic V2 Migration Guide at https://errors.pydantic.dev/2.8
/migration/
    warn(

microservice/main.py:24
  /app/microservice/main.py:24: DeprecationWarning:
          on_event is deprecated, use lifespan event handlers instead.

          Read more about it in the
          [FastAPI docs for Lifespan Events](https://fastapi.tiangolo.com/advanced/events/).

    @app.on_event("startup")

../home/appuser/.local/lib/python3.12/site-packages/fastapi/applications.py:4495
../home/appuser/.local/lib/python3.12/site-packages/fastapi/applications.py:4495
  /home/appuser/.local/lib/python3.12/site-packages/fastapi/applications.py:4495: DeprecationWarning:
          on_event is deprecated, use lifespan event handlers instead.

          Read more about it in the
          [FastAPI docs for Lifespan Events](https://fastapi.tiangolo.com/advanced/events/).

    return self.router.on_event(event_type)

microservice/main.py:33
  /app/microservice/main.py:33: DeprecationWarning:
          on_event is deprecated, use lifespan event handlers instead.

          Read more about it in the
          [FastAPI docs for Lifespan Events](https://fastapi.tiangolo.com/advanced/events/).

    @app.on_event("shutdown")

tests/test_api.py::test_healthcheck_items_endpoint
  /home/appuser/.local/lib/python3.12/site-packages/httpx/_client.py:690: DeprecationWarning: The 'app' shortcut is now deprecated. Use the explicit style 'transport=WSGITranspo
rt(app=...)' instead.
    warnings.warn(message, DeprecationWarning)

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
2 passed, 8 warnings in 0.76s
appuser@37d1522d83ee:/app$
```

Ahora se procede a realizar el despliegue en Kubernetes local

```
diegodev@HPavilion:~/Desktop/Computer-Science/ds-practice/Laboratorio10$ minikube start
😄  minikube v1.36.0 on Ubuntu 22.04
✨  Automatically selected the docker driver. Other choices: qemu2, ssh
💨  For improved Docker performance, enable the overlay Linux kernel module using 'modprobe overlay'
📌  Using Docker driver with root privileges
❗  For an improved experience it's recommended to use Docker Engine instead of Docker Desktop.
Docker Engine installation instructions: https://docs.docker.com/engine/install/#server
👍  Starting "minikube" primary control-plane node in "minikube" cluster
🚜  Pulling base image v0.0.47 ...
💾  Downloading Kubernetes v1.33.1 preload ...
    > preloaded-images-k8s-v18-v1...:  347.04 MiB / 347.04 MiB  100.00% 23.81 M
    > gcr.io/k8s-minikube/kicbase...:  502.26 MiB / 502.26 MiB  100.00% 15.31 M
🔥  Creating docker container (CPUs=2, Memory=1963MB) ...
🐳  Preparing Kubernetes v1.33.1 on Docker 28.1.1 ...
    ▪ Generating certificates and keys ...
    ▪ Booting up control plane ...
    ▪ Configuring RBAC rules ...
🔗  Configuring bridge CNI (Container Networking Interface) ...
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
```

Configuramos el entorno de Docker para Minikube y construimos la imagen, pero esta vez, dentro del entorno de Minikub.

```
diegodev@HPavilion:~/Desktop/Computer-Science/ds-practice/Laboratorio10$ eval $(minikube docker-env)
diegodev@HPavilion:~/Desktop/Computer-Science/ds-practice/Laboratorio10$ docker build -t ejemplo-microservice:0.1.0 .
[+] Building 27.9s (15/15) FINISHED                                                                              do
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 1.16kB
 => [internal] load metadata for docker.io/library/python:3.12-slim
 => [auth] library/python:pull token for registry-1.docker.io
 => [internal] load .dockerignore
 => => transferring context: 257B
 => [internal] load build context
 => => transferring context: 51.78kB
 => [builder 1/4] FROM docker.io/library/python:3.12-slim@sha256:b43ff04d5df04ad5cabb80890b7ef74e8410e3395b19af970dcd52d7a4bff921
 => => resolve docker.io/library/python:3.12-slim@sha256:b43ff04d5df04ad5cabb80890b7ef74e8410e3395b19af970dcd52d7a4bff921
 => => sha256:b43ff04d5df04ad5cabb80890b7ef74e8410e3395b19af970dcd52d7a4bff921 10.37kB / 10.37kB
 => => sha256:971f04b358cf483ec445a8d388fb55267451f080d90fb136c8e69684a02a9604 1.75kB / 1.75kB
 => => sha256:445121148b187db67e48799f002500623fa22d9f635e522f4e0f345414bd9107 5.68kB / 5.68kB
 => => sha256:0e4bc2bd6656e6e004e3c749af70e5650bac2258243eb0949dea51cb8b7863db 29.78MB / 29.78MB
 => => sha256:490b9a1c25e472ab1cceab4ed6f3bd252006f7c6eea0ef7ba845c996adb6f302 1.29MB / 1.29MB
 => => sha256:0674d14a155c94f13e648265aa3ba62410e1fe0387fef64cc608388e54232880 12.11MB / 12.11MB
 => => sha256:b7ba6d2a1fc72f9587288b3b60221d1b07dae4f61a41360d2cc281fe7c007b3a 250B / 250B
 => => extracting sha256:0e4bc2bd6656e6e004e3c749af70e5650bac2258243eb0949dea51cb8b7863db
 => => extracting sha256:490b9a1c25e472ab1cceab4ed6f3bd252006f7c6eea0ef7ba845c996adb6f302
 => => extracting sha256:0674d14a155c94f13e648265aa3ba62410e1fe0387fef64cc608388e54232880
 => => extracting sha256:b7ba6d2a1fc72f9587288b3b60221d1b07dae4f61a41360d2cc281fe7c007b3a
```

Aplicamos los manifiestos de kubernetes

```
diegodev@HPavilion:~/Desktop/Computer-Science/ds-practice/Laboratorio10$ kubectl apply -f k8s/
^[[3~deployment.apps/ejemplo-microservice-deployment created
service/ejemplo-microservice-service created
```

Verificamos el despliegue

```
diegodev@HPavilion:~/Desktop/Computer-Science/ds-practice/Laboratorio10$ kubectl get pods,svc -o wide
NAME                                                  READY   STATUS    RESTARTS   AGE   IP           NODE       NOMINATED NODE   READINESS GATES
pod/ejemplo-microservice-deployment-794685697b-7xch2  1/1     Running   0          87s   10.244.0.4   minikube   <none>           <none>
pod/ejemplo-microservice-deployment-794685697b-wcckl  1/1     Running   0          87s   10.244.0.3   minikube   <none>           <none>

NAME                                   TYPE        CLUSTER-IP     EXTERNAL-IP   PORT(S)        AGE     SELECTOR
service/ejemplo-microservice-service   NodePort    10.98.173.76   <none>        80:30080/TCP   87s     app=ejemplo-microservice
service/kubernetes                     ClusterIP   10.96.0.1      <none>        443/TCP        6m23s   <none>
```

Accede al microservicio localmente

```
diegodev@HPavilion:~/Desktop/Computer-Science/ds-practice/Laboratorio10$ kubectl port-forward service/ejemplo-microservice-service 8080:80
Forwarding from 127.0.0.1:8080 -> 80
Forwarding from [::1]:8080 -> 80
```

Verificamos el log del primer pod

```
diegodev@HPavilion:~/Desktop/Computer-Science/ds-practice/Laboratorio10$ kubectl logs ejemplo-microservice-deployment-794685697b-7xch2 -f
INFO:     Started server process [1]
INFO:     Waiting for application startup.
2025-11-30 02:21:34,077 - INFO - microservice - Arrancando la aplicación
2025-11-30 02:21:34,077 - INFO - microservice - Inicializando base de datos en app.db
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:80 (Press CTRL+C to quit)
INFO:     10.244.0.1:48614 - "GET /api/items/ HTTP/1.1" 200 OK
INFO:     10.244.0.1:48622 - "GET /api/items/ HTTP/1.1" 200 OK
INFO:     10.244.0.1:36520 - "GET /api/items/ HTTP/1.1" 200 OK
INFO:     10.244.0.1:36534 - "GET /api/items/ HTTP/1.1" 200 OK
```