

## Actividad 19

Nombres y Apellidos: Sandro Alfredo Carrillo Jordán

Se comprueba de que se puede levantar el Docker con normalidad:

```
sandr@Sandro MINGW64 ~/OneDrive/Desktop/Universidad/7mo ciclo/DS/CC3S2/Actividad 19 (main)
$ docker build --no-cache -t ejemplo-microservice:0.1.0 .
-> [production 4/6] COPY --from=builder /root/.local /home/appuser/.local          0.4s
-> [production 5/6] COPY . /app                                                 0.1s
-> [production 6/6] RUN chown -R appuser:appuser /app                           0.6s
-> exporting to image                                                       4.1s
-> => exporting layers                                                       3.0s
=> => exporting manifest sha256:8ff95dcab52c6a42d9741125c800b1b01d4e13dc2b3b7e22f0e033da042305ae   0.0s
-> => exporting config sha256:699d731d0d15efa80ac1496257c796bd08769dbf654973e85656ad627422b605  0.0s
-> => exporting attestation manifest sha256:e1ecfa707b5137d607a14e7188b5fd03a89805de3503bb8f4f7906bfc17a5d  0.1s
-> => exporting manifest list sha256:b25b4b10897057478e491780f587a68cd7d8899e9fbfce9e7296126f40239b6  0.0s
-> => naming to docker.io/library/ejemplo-microservice:0.1.0                      0.0s
-> => unpacking to docker.io/library/ejemplo-microservice:0.1.0                     0.9s
```

```
sandr@Sandro MINGW64 ~/OneDrive/Desktop/Universidad/7mo ciclo/DS/CC3S2/Actividad 19 (main)
● $ docker run --rm -d --name ejemplo-ms -p 80:80 ejemplo-microservice:0.1.0
686916ec1520e36a919cb9af9c173010ecf41c8fe59f7a17d9691f3f8719fa44
```

Sin el app.db:

```
sandr@Sandro ~ MINGW64
● $ curl -i http://localhost/api/items/
HTTP/1.1 200 OK
date: Sat, 22 Nov 2025 14:36:19 GMT
server: uvicorn
content-length: 2
content-type: application/json

[]
```

Con el app.db:

```
● $ curl -i http://localhost/api/items/
HTTP/1.1 200 OK
date: Sat, 22 Nov 2025 14:38:57 GMT
server: uvicorn
content-length: 68
content-type: application/json

[{ "name": "test-item", "description": "Descripción de prueba", "id": 1}]
```

Además, las pruebas unitarias en el laboratorio ya cumplen por lo pedido. Solo habría que asegurar en una de las pruebas de que la respuesta .json es una lista.

```
def test_healthcheck_items_endpoint(client):
    """El listado debería responder 200 OK aun cuando no hay datos"""
    resp = client.get("/api/items")
    data = resp.json()
    assert resp.status_code == 200
    assert isinstance(data, list)
```

Para esto se guarda la respuesta obtenida en una variable data y mediante isinstance comprobamos de que devuelve una lista (de ítems en este caso).

Intenté correr los tests pero tuve un error debido a que el test para crear un usuario lo hace con una variable hardcodeada, o sea que siempre es la misma.

```
def test_create_item_and_verify_in_list(client):
    """
    1) Crea un ítem          -> 201 Created
    2) Devuelve los datos     -> nombre y descripción coinciden
    3) El ítem aparece luego en el listado general
    """

    payload = {"name": ITEM_NAME, "description": ITEM_DESCR}
    create_resp = client.post("/api/items", json=payload)
    assert create_resp.status_code == 201

    created = create_resp.json()
    assert created["name"] == ITEM_NAME
    assert created["description"] == ITEM_DESCRIPTION

    list_resp = client.get("/api/items")
    assert list_resp.status_code == 200
    items = list_resp.json()

    assert any(i["name"] == ITEM_NAME for i in items), (
        f"El ítem '{ITEM_NAME}' debería figurar en la lista"
    )
```

Debido a que la BD tiene de constraint UNIQUE en el name, sale este error:

```
$ pytest -v
File "C:\Users\sandr\OneDrive\Desktop\Universidad\7mo ciclo\DS\CC3S2\Actividad 19\microservice\services\database.py", line 53, in add_item
    cursor.execute(
~~~~~
        "INSERT INTO items (name, description) VALUES (?, ?)",
~~~~~
        (name, description)
~~~~~
    )
^
sqlite3.IntegrityError: UNIQUE constraint failed: items.name
```

Entonces borré el archivo app.db y ejecuté de nuevo, con esto pude comprobar que funciona correctamente todo.

```
sandro@sandro MINGW64 ~\OneDrive\Desktop\Universidad\7mo ciclo\DS\CC3S2\Actividad 19 (main)
$ pytest -v
tests/test_api.py::test_healthcheck_items_endpoint
  C:\Users\sandro\OneDrive\Desktop\Universidad\7mo ciclo\DS\CC3S2\Actividad 19\venv\Lib\site-packages\httpx\_client.py:690: DeprecationWarning:
The 'app' shortcut is now deprecated. Use the explicit style 'transport=WSGITransport(app=...)' instead.
    warnings.warn(message, DeprecationWarning)

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
===== 2 passed, 8 warnings in 1.00s =====
```

Más que nada hay warnings debido a que el decorador app está obsoleto