**Desplegament d’una aplicació web client-servidor amb Docker Compose**

**PAS 0: Preparació de l’entorn**

0.1 Crear una carpeta de treball:

mkdir multi-app

cd multi-app

0.2 Crear tres carpetes per als serveis:

mkdir frontend backend db

**PAS 1: Crear el fitxer docker-compose.yml**

Crear el fitxer principal de la pràctica:

nano docker-compose.yml

Contingut del fitxer:

version: '3.8'

services:

 frontend:

 image: nginx:alpine

 container\_name: web-frontend

 ports:

 - "8080:80"

 volumes:

 - ./frontend:/usr/share/nginx/html:ro

 depends\_on:

 - backend

 backend:

 build: ./backend

 container\_name: api-backend

 ports:

 - "5000:5000"

 environment:

 - DB\_HOST=db

 - DB\_USER=demo

 - DB\_PASSWORD=demopass

 - DB\_NAME=demoapp

 depends\_on:

 - db

 db:

 image: mysql:5.7

 container\_name: mysql-db

 environment:

 - MYSQL\_ROOT\_PASSWORD=root

 - MYSQL\_DATABASE=demoapp

 - MYSQL\_USER=demo

 - MYSQL\_PASSWORD=demopass

 volumes:

 - db\_data:/var/lib/mysql

volumes:

 db\_data:

**PAS 2: Crear el frontend (client web)**

2.1 Crear el fitxer HTML:

nano frontend/index.html

Contingut:

<!DOCTYPE html>

<html>

 <head>

 <meta charset="UTF-8">

 <title>Aplicació demo</title>

 </head>

 <body>

 <h1>Benvingut!</h1>

 <div id="resultat"></div>

 <script>

 fetch('http://localhost:5000/dades')

 .then(res => res.json())

 .then(data => {

 document.getElementById('resultat').innerHTML =

 JSON.stringify(data);

 })

 .catch(err => {

 document.getElementById('resultat').innerHTML =

 "Error en la connexió amb l’API";

 });

 </script>

 </body>

</html>

**PAS 3: Crear el backend (API en Flask amb suport CORS)**

*Flask és un micro-framework per a Python que permet crear aplicacions web i APIs REST de manera ràpida i senzilla.*

3.1 Crear app.py:

nano backend/app.py

Contingut amb suport CORS:

from flask import Flask, jsonify

from flask\_cors import CORS

import mysql.connector

import os

app = Flask(\_\_name\_\_)

CORS(app)

@app.route('/dades')

def dades():

 db = mysql.connector.connect(

 host=os.environ['DB\_HOST'],

 user=os.environ['DB\_USER'],

 password=os.environ['DB\_PASSWORD'],

 database=os.environ['DB\_NAME']

 )

 cursor = db.cursor()

 cursor.execute("SELECT 'Hola món!'")

 result = cursor.fetchone()

 return jsonify({"missatge": result[0]})

if \_\_name\_\_ == '\_\_main\_\_':

 app.run(host='0.0.0.0')

3.2 Crear requirements.txt:

nano backend/requirements.txt

Contingut:

flask

flask-cors

mysql-connector-python

3.3 Crear el Dockerfile:

nano backend/Dockerfile

Contingut:

FROM python:3.10-slim

WORKDIR /app

COPY requirements.txt .

RUN pip install -r requirements.txt

COPY . .

CMD ["python", "app.py"]

**PAS 4: Posar en marxa l’aplicació**

4.1 Construir el backend amb els nous requisits:

docker compose build backend

4.2 Iniciar tots els serveis:

docker compose up -d

4.3 Comprovar l’estat:

docker compose ps

**PAS 5: Provar l’aplicació**

1. Obre el navegador i visita:

http://localhost:8080

1. Hauries de veure:

Benvingut!

{"missatge":"Hola món!"}

Si no apareix el missatge, obre la consola del navegador (F12) i comprova si hi ha errors de connexió o CORS.