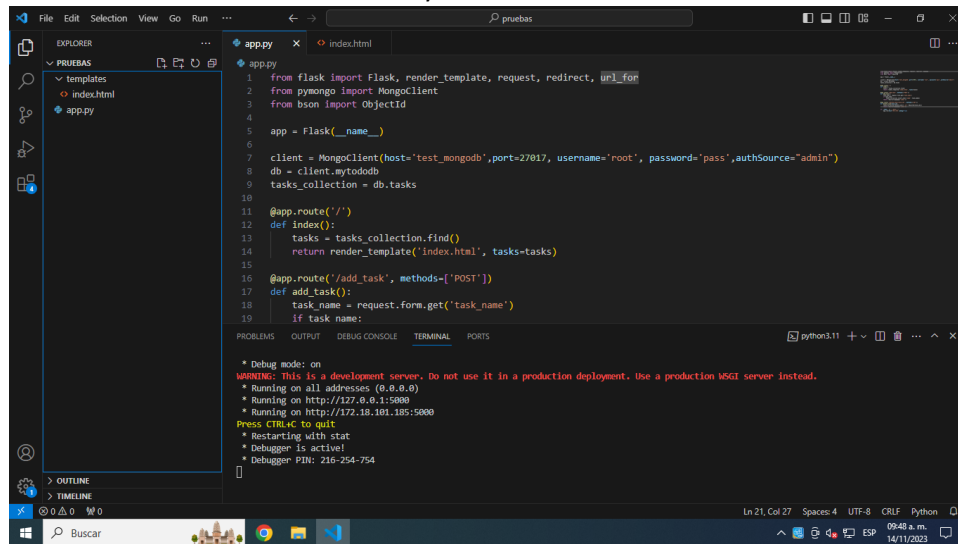


## **Distribución de aplicaciones: Contenedores**

Irvyn Xicale Cabrera

201963582

Creamos nuestra estructura inicial y verificamos nuestro servidor flask



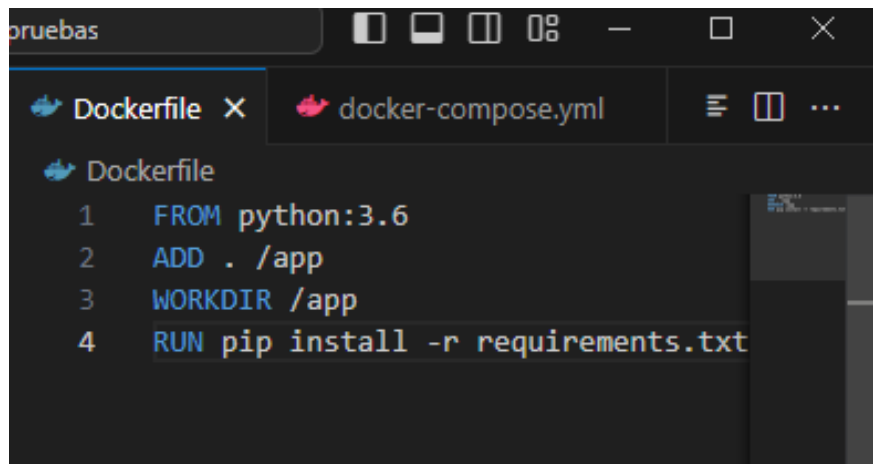
The screenshot shows a VS Code editor with a project named 'pruebas'. The Explorer sidebar on the left shows a file structure with 'templates', 'index.html', and 'app.py'. The main editor displays the content of 'app.py', which is a Flask application. The code includes imports for Flask, render\_template, request, redirect, url\_for, pymongo, MongoClient, and bson. It initializes a Flask app, connects to a MongoDB database, and defines two routes: a GET route for the index and a POST route for adding a task. The terminal at the bottom shows the output of running the application, indicating it is running on http://172.18.181.185:5000.

```
1 from flask import Flask, render_template, request, redirect, url_for
2 from pymongo import MongoClient
3 from bson import ObjectId
4
5 app = Flask(__name__)
6
7 client = MongoClient(host='test_mongodb', port=27017, username='root', password='pass', authSource='admin')
8 db = client.mytododb
9 tasks_collection = db.tasks
10
11 @app.route('/')
12 def index():
13     tasks = tasks_collection.find()
14     return render_template('index.html', tasks=tasks)
15
16 @app.route('/add_task', methods=['POST'])
17 def add_task():
18     task_name = request.form.get('task_name')
19     if task_name:
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.18.181.185:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active
* Debugger PIN: 216-254-754
```

Dockerfile se utiliza para especificar la imagen base, copiar el código de la aplicación al contenedor, configurar el entorno y las dependencias.



The screenshot shows a VS Code editor with a file named 'Dockerfile' open. The file contains four lines of Docker instructions: FROM python:3.6, ADD . /app, WORKDIR /app, and RUN pip install -r requirements.txt.

```
1 FROM python:3.6
2 ADD . /app
3 WORKDIR /app
4 RUN pip install -r requirements.txt
```

docker-compose.yml se utiliza para definir servicios, redes, puertos, conexión entre contenedores y volúmenes.

```

version: '3'

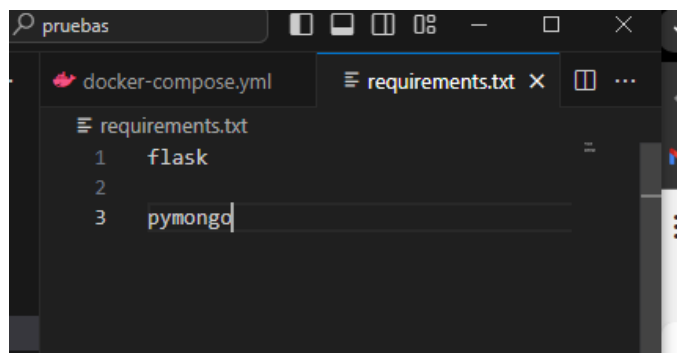
services:
  web:
    build: .
    command: ["python", "-u", "app.py"]
    ports:
      - "5000:5000"
    volumes:
      - ./app
    depends_on:
      - db
    networks:
      - my_network

  db:
    image: mongo:latest
    hostname: test_mongodb
    environment:
      - MONGO_INITDB_DATABASE=animal_db
      - MONGO_INITDB_ROOT_USERNAME=root
      - MONGO_INITDB_ROOT_PASSWORD=pass
    ports:
      - "27017:27017"
    networks:
      - my_network

networks:
  my_network:
    driver: bridge

```

Requisitos.txt se utiliza para instalar las dependencias de la aplicación



The screenshot shows a code editor window with the title 'pruebas'. It has two tabs: 'docker-compose.yml' and 'requirements.txt'. The 'requirements.txt' tab is active, showing the following content:

```

1 flask
2
3 pymongo

```

Debemos instalar docker en nuestro equipo y nos ayudaremos de la documentacion oficial.



