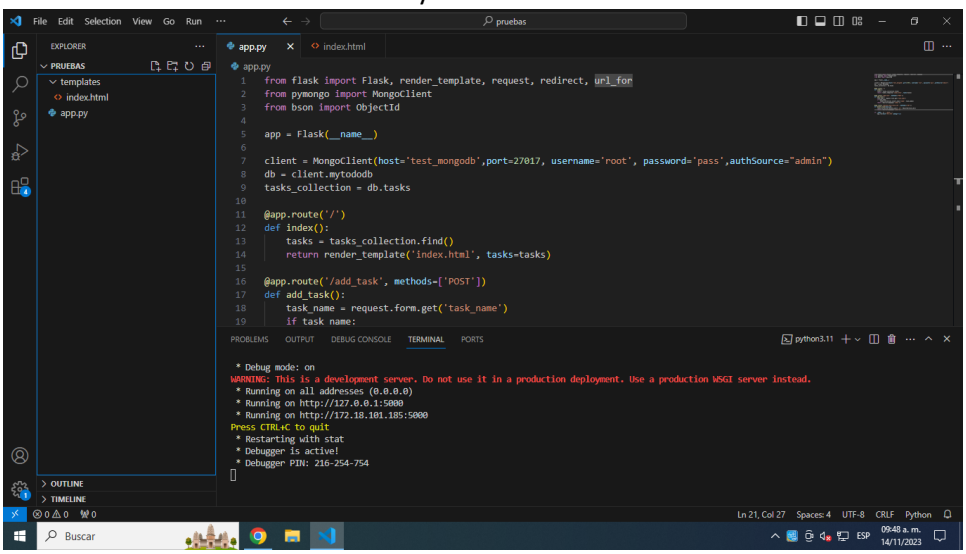
# Distribución de aplicaciones: Contenedores

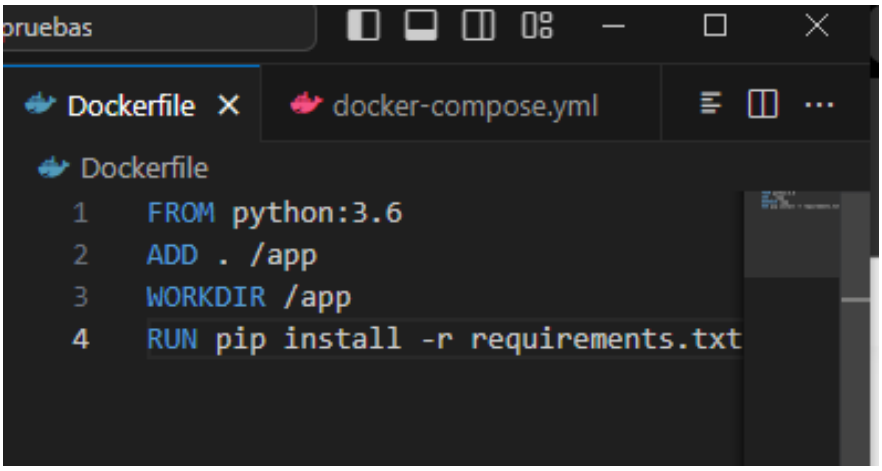
**Irvyn Xicale Cabrera**

**201963582**

Creamos nuestra estructura inicial y verificamos nuestro servidor flask




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




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

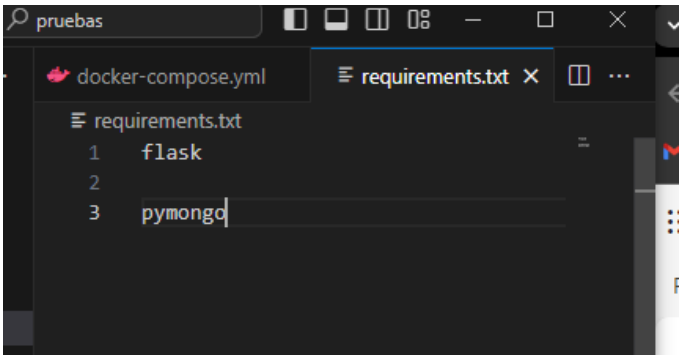
```yaml
version: '3'

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

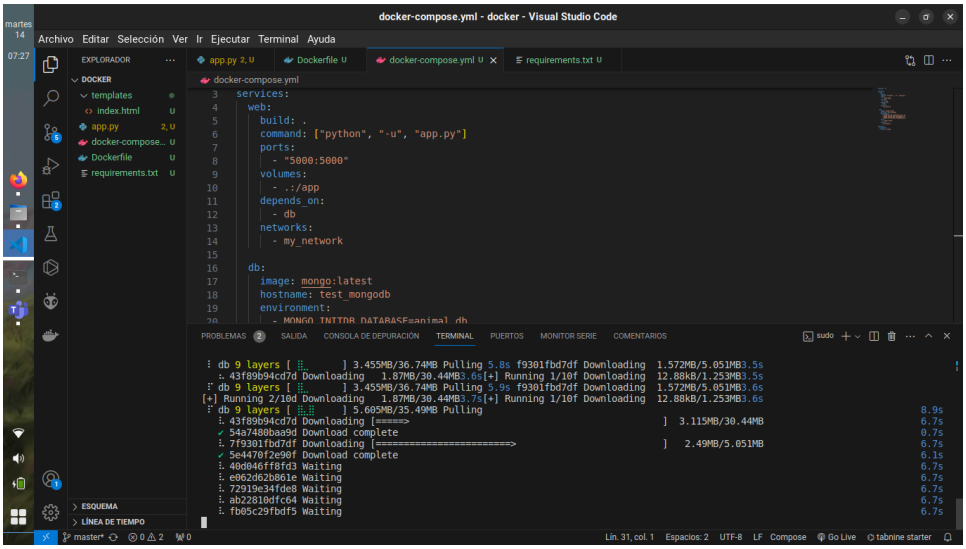



Debemos instalar docker en nuestro equipo y nos ayudaremos de la documentacion oficial.

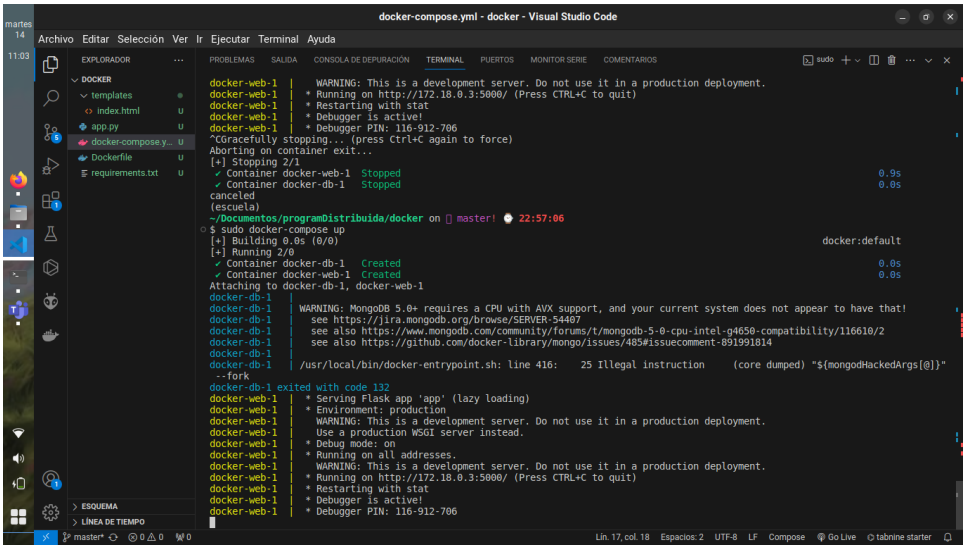
```
~/Documentos/programDistribuida/docker on  master! 22:45:37
$ docker-compose --version

Docker Compose version v2.23.0
(escuela)
~/Documentos/programDistribuida/docker on  master! 22:45:48
$ docker --version
Docker version 24.0.7, build afdd53b
(escuela)
~/Documentos/programDistribuida/docker on  master! 22:46:00
$
```

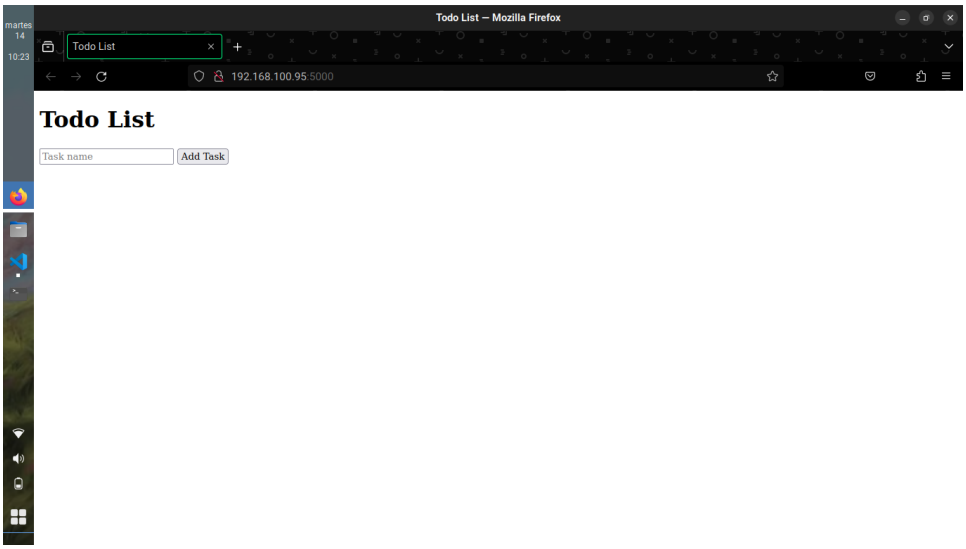
Ahora creamos la imagen docker con sudo docker-compose up




Finalmente ejecutamos el docker container con “sudo docker-compose up”




Y podemos observar nuestra pagina en la direccion [http://192.168.100.95:5000/](http://192.168.100.95:5000/)


martes
14
10:23
Todo List — Mozilla Firefox
Todo List
192.168.100.95:5000
Todo List
Task name
Add Task