**ESCUELA POLITÉCNICA NACIONAL**

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

**APLICACIONES DISTRIBUIDAS**

|  |  |
| --- | --- |
|  | |
| ASIGNATURA: | Aplicaciones Distribuidas |
| PROFESOR: | Ing. Vanessa Guevara |
| PERÍODO ACADÉMICO: | 2024-B |
|  | |

**TAREA**

|  |  |
| --- | --- |
|  | |
| TÍTULO:  **Rendimiento y Tolerancia a fallos** | |
|  | |
|  | |
| Simplify Database Management with Docker Compose: A Comprehensive Guide |  by Akshat Gadodia | Medium |  |
|  | |

# OBJETIVOS

* Comprender y aplicar el concepto de balanceo de carga en aplicaciones distribuidas mediante la creación y configuración de un entorno con múltiples servicios utilizando Docker Compose.

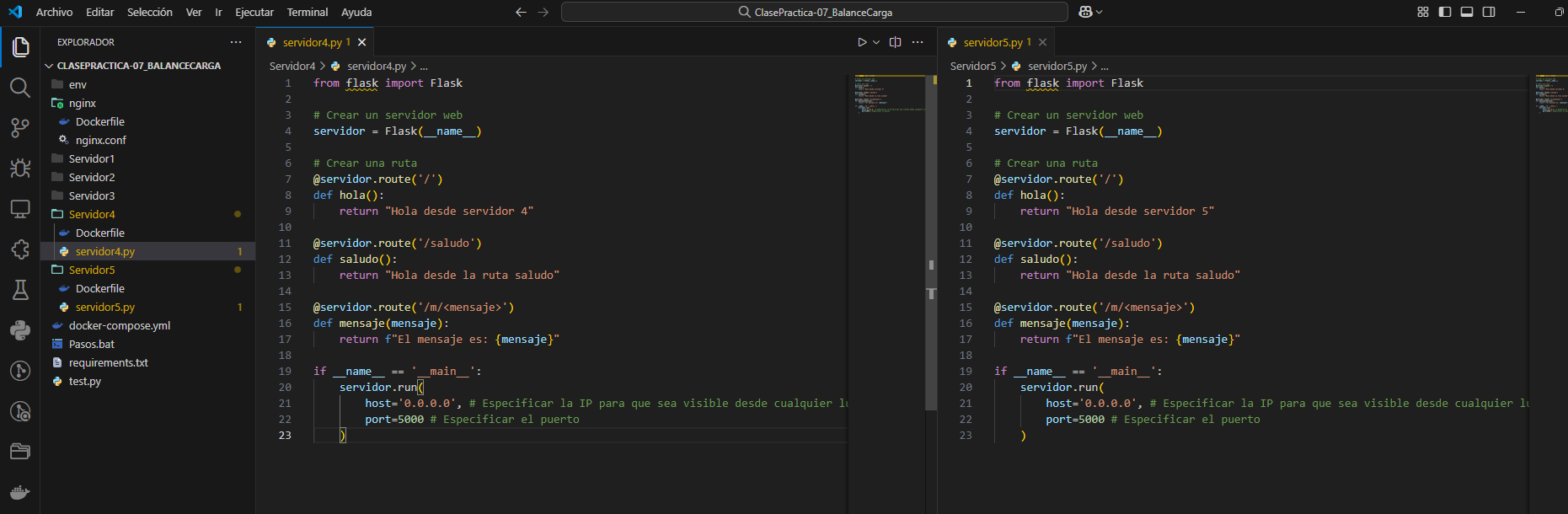
# INSTRUCCIONES

* **Balanceo de carga con Nginx**
* Configurar Nginx para realizar el balanceo de carga entre **5 servidores** (aplicación Python).

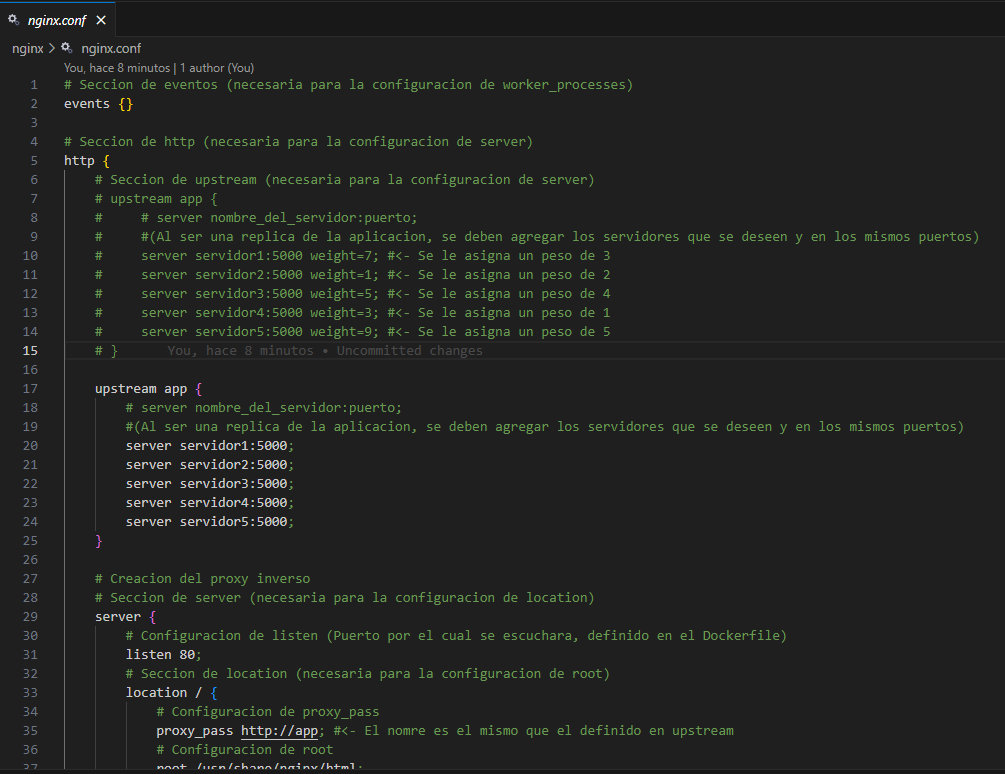
A screenshot of a computer program

Description automatically generated

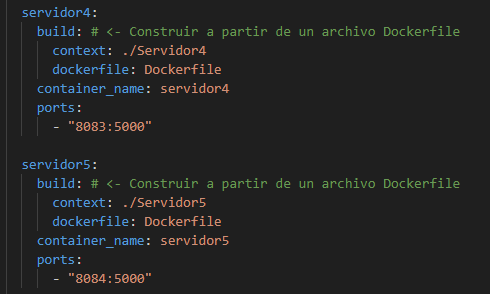
Creación de 2 servidores adicionales y su respectivo Dockerfile



Configuración de los servidores con *nginx* en el archivo ***nginx.conf***



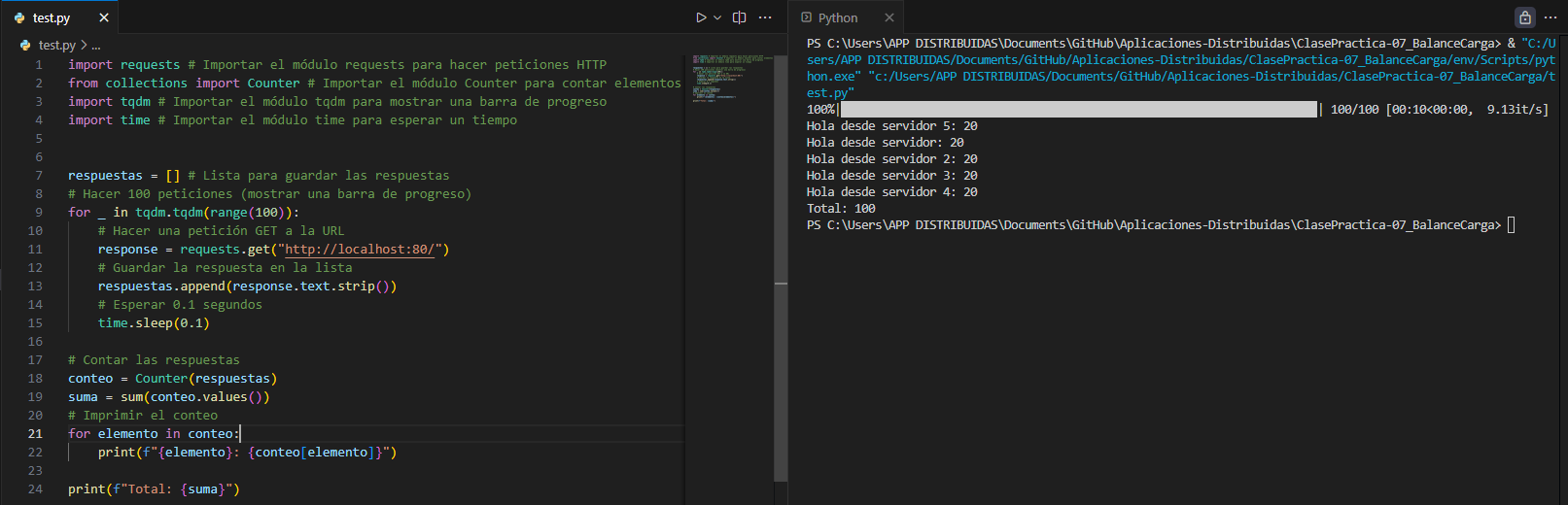
Y se crean los contenedores para estos servidores en el archivo *docker-compose.yml*



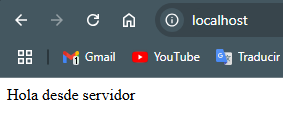
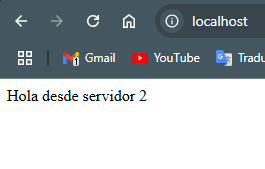
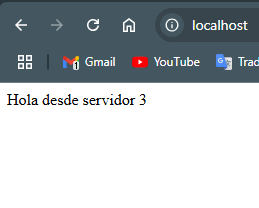
* **Consultar cómo NGINX distribuye las solicitudes entre las réplicas.**

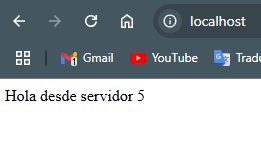
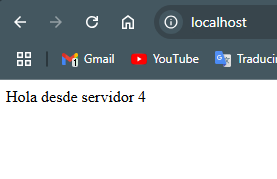
Nginx usa el algoritmo **Round Robin** por defecto, el algoritmo distribuye las solicitudes a los servidores de manera uniforme y en orden secuencial

Con Python se realiza las pruebas para verificar cuantas veces ha respondido un servidor



Respuestas de las solicitudes en el navegador

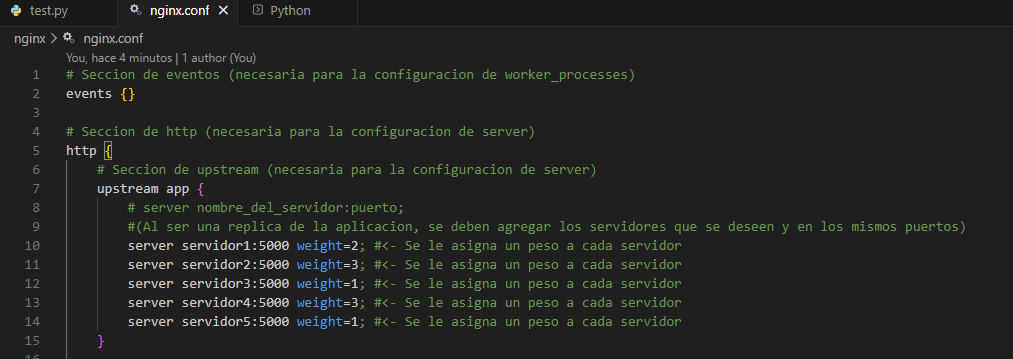


* **Balanceo de carga basado en peso** **con Nginx**
* Modificar el archivo **nginx.conf** para implementar balanceo de carga basado en peso, con los siguientes pesos:
  + Servidor 1: 2
  + Servidor 2: 3
  + Servidor 3: 1
  + Servidor 4: 3
  + Servidor 5: 1

**A screenshot of a computer program

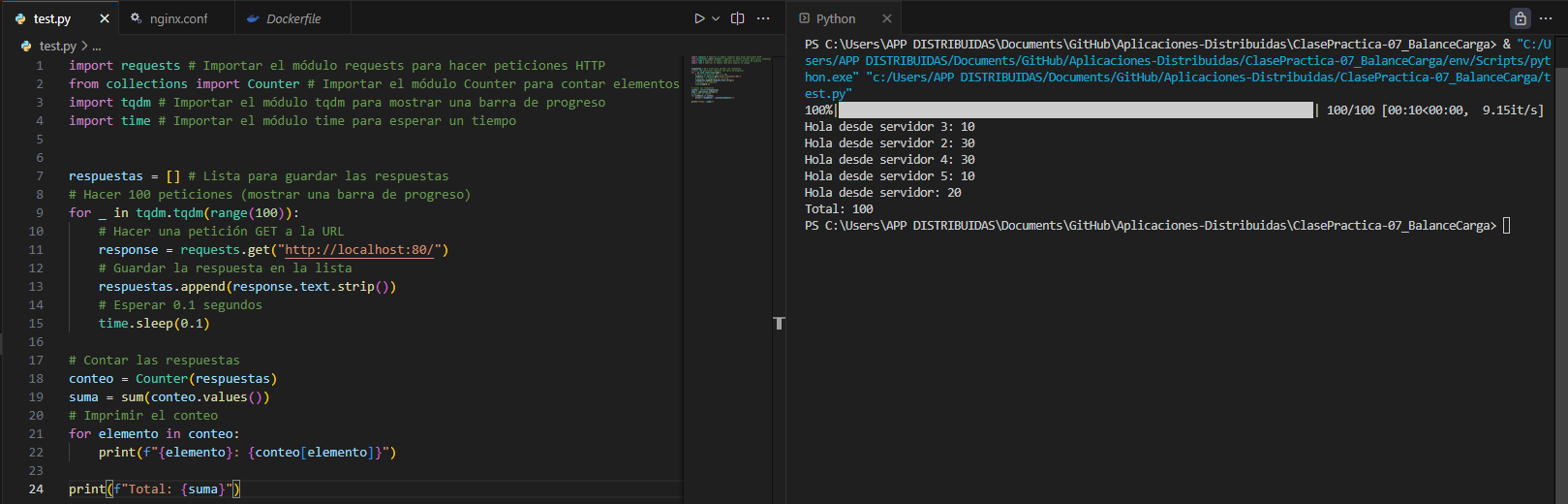
Description automatically generated**

Modificación del archivo **nginx.conf** para incluir los pesos de los servidores (esto permite la aleatoriedad en las respuestas recibidas al cliente)



* Utilizar un script de Python para enviar 100 solicitudes y contar las respuestas.

Usando un script de Python, se puede apreciar que ahora no hay uniformidad con el numero0 de respuestas recibidas del servidor al cliente, existen variaciones



* **Balanceo de Carga con Afinidad de Sesión**
* Modificar el archivo default.conf para implementar balanceo de carga donde las solicitudes de un mismo cliente siempre se envíen al mismo servidor backend.
* **Consultar como realizar pruebas con diferentes clientes o simulando distintas IPs.**

Colocar capturas con las respuestas de cada servidor.

A screen shot of a computer

Description automatically generated

# CONCLUSIONES

(3 CONCLUSIONES)

**PRESENTACIÓN**

Recuerda el nombre del archivo deberá ser: **TC\_APDIST\_2024B\_NApellido**.