

RESTAURANTE FAST FOOD

Santiago de Cali, noviembre del 2024

Luis Ángel García / 2230177

Antonio Cardenas / 2230433

Juliana Toro Serrano / 2225658



Fast food

FAST FOOD



Identificación del Problema

FASTFOOD es una empresa de comida rápida con problemas de gestión de órdenes y personal por falta de un sistema inteligente o página web, lo que causa pérdida de clientes y mala organización de tareas.

Como objetivo, vamos a implementar una página web con roles de cliente, gerente y trabajador, mejorando la gestión de personal y órdenes en FASTFOOD, aumentará las ventas y optimizará la asignación y reporte de tareas.

Modulo

Este módulo se enfoca en diseñar y gestionar infraestructuras escalables y distribuidas. Usando herramienta como Docker, los estudiantes aprenden a implementar aplicaciones de microservicios y procesamiento de datos a gran escala.



Objetivo general

Implementar una página web con roles de cliente, gerente y trabajador, mejorando la gestión de personal y órdenes en FASTFOOD, aumentará las ventas y optimizará la asignación y reporte de tareas.



Automatizar la Gestión de Órdenes

Implementar un sistema de pedidos en línea que permita a los clientes realizar órdenes de manera rápida y sencilla, optimizando así el proceso de atención y reduciendo los tiempos de espera.



Optimizar el Proceso de Gestión de Órdenes

Centralizar y automatizar la recepción, seguimiento y actualización de órdenes, permitiendo a los clientes visualizar el estado de sus pedidos en tiempo real y facilitando su gestión para el personal



Facilitar el Monitoreo en Tiempo Real

Implementar un sistema de visualización en tiempo real de las métricas operativas (como el estado de órdenes y desempeño del personal) para que los gerentes puedan tomar decisiones rápidas y mejorar la eficiencia diaria.

Seleccion de dataset



Balaji Fast Food Sales.csv (62.49 kB)

DetailCompactColumn

10 of 10 columns

About this file

This file does not have a description yet.

Add Suggestion

order_id	date	item_name	item_type	item_price	quantity	transaction_amount	transaction_type	received_by	time_of_sale
<div><div></div><div>11000</div></div>	<div><div></div><div>2022-03-312023-03-29</div></div>	<div><div></div><div>Cold coffee16%Fastfood69%</div><div>Sugarcane juice15%Beverages31%</div><div>Other (686)69%</div></div>		<div><div></div><div>2060</div></div>	<div><div></div><div>115</div></div>	<div><div></div><div>20900</div></div>	<div><div></div><div>Cash48%Mr.51%</div><div>Online42%Mrs.49%</div><div>Other (107)11%</div></div>		<div><div></div><div>Night21%</div><div>Afternoon21%</div><div>Other (590)59%</div></div>
1	07-03-2022	Aalopuri	Fastfood	20	13	260		Mr.	Night
2	8/23/2022	Vadapav	Fastfood	20	15	300	Cash	Mr.	Afternoon
3	11/20/2022	Vadapav	Fastfood	20	1	20	Cash	Mr.	Afternoon
4	02-03-2023	Sugarcane juice	Beverages	25	6	150	Online	Mr.	Night
5	10-02-2022	Sugarcane juice	Beverages	25	8	200	Online	Mr.	Evening
6	11/14/2022	Vadapav	Fastfood	20	10	200	Cash	Mr.	Evening
7	05-03-2022	Sugarcane juice	Beverages	25	9	225	Cash	Mr.	Evening
8	12/22/2022	Panipuri	Fastfood	20	14	280	Online	Mr.	Night
9	06-10-2022	Panipuri	Fastfood	20	1	20	Cash	Mrs.	Morning
10	9/16/2022	Panipuri	Fastfood	20	5	100	Online	Mr.	Afternoon
11	12-01-2022	Frankie	Fastfood	50	8	400	Online	Mrs.	Afternoon
12	07-12-2022	Vadapav	Fastfood	20	8	160	Online	Mrs.	Night
13	12/22/2022	Panipuri	Fastfood	20	9	180	Online	Mrs.	Afternoon
14	11/25/2022	Frankie	Fastfood	50	4	200	Online	Mr.	Morning
15	02-03-2023	Aalopuri	Fastfood	20	3	60	Cash	Mrs.	Evening

Elegimos el dataset de FASTFOOD porque la empresa enfrenta problemas en la gestión de órdenes y personal debido a la falta de un sistema inteligente o página web. Esto afecta la atracción de clientes y la eficiencia operativa, y buscamos soluciones tecnológicas para mejorar ambos aspectos.



Dado que el dataset original no contaba con suficientes datos, se creó un generador basado en su estructura, simplificando los elementos y obteniendo un conjunto de 20 datos relevantes.

```
import random
from datetime import datetime, timedelta

# Listado de usuarios y productos
usuarios = [
    ('Ana', 'anagarcia'), ('Andrea', 'andreamg'), ('Camilo', 'camip1234'),
    ('Camila', 'cami_sm23'), ('Carlos', 'carlito01'), ('Daniel', 'dannyg123'),
    ('Valeria', 'valeria_1')
]

productos = [
    ('Aalopuri', 20), ('Cold coffee', 40), ('Frankie', 50),
    ('Panipuri', 20), ('Sandwich', 60), ('Sugarcane juice', 25),
    ('Vadapav', 20)
]

# Generar 1000 órdenes con datos aleatorios
ordenes = []
fecha_base = datetime(2024, 10, 31, 9, 0, 0)

for i in range(1, 20001):
    usuario = random.choice(usuarios)
    producto = random.choice(productos)
    estado = 'creada'
    precio = producto[1]
    fecha_orden = fecha_base + timedelta(minutes=random.randint(1, 1440))

    orden = (i, usuario[0], usuario[1], estado, precio, fecha_orden.strftime('%Y-%m-%d %H:%M:%S'))
    ordenes.append(orden)

# Generar el script SQL
with open('insert_ordenes.sql', 'w') as file:
    file.write("INSERT INTO `orden` VALUES\n")
    for orden in ordenes[:-1]: # Agregar todos menos el último con coma
        file.write(f"({orden[0]}, '{orden[1]}', '{orden[2]}', '{orden[3]}', {orden[4]}, '{orden[5]}'),\n")
    # Agregar el último sin coma
    orden = ordenes[-1]
    file.write(f"({orden[0]}, '{orden[1]}', '{orden[2]}', '{orden[3]}', {orden[4]}, '{orden[5]}');\n")

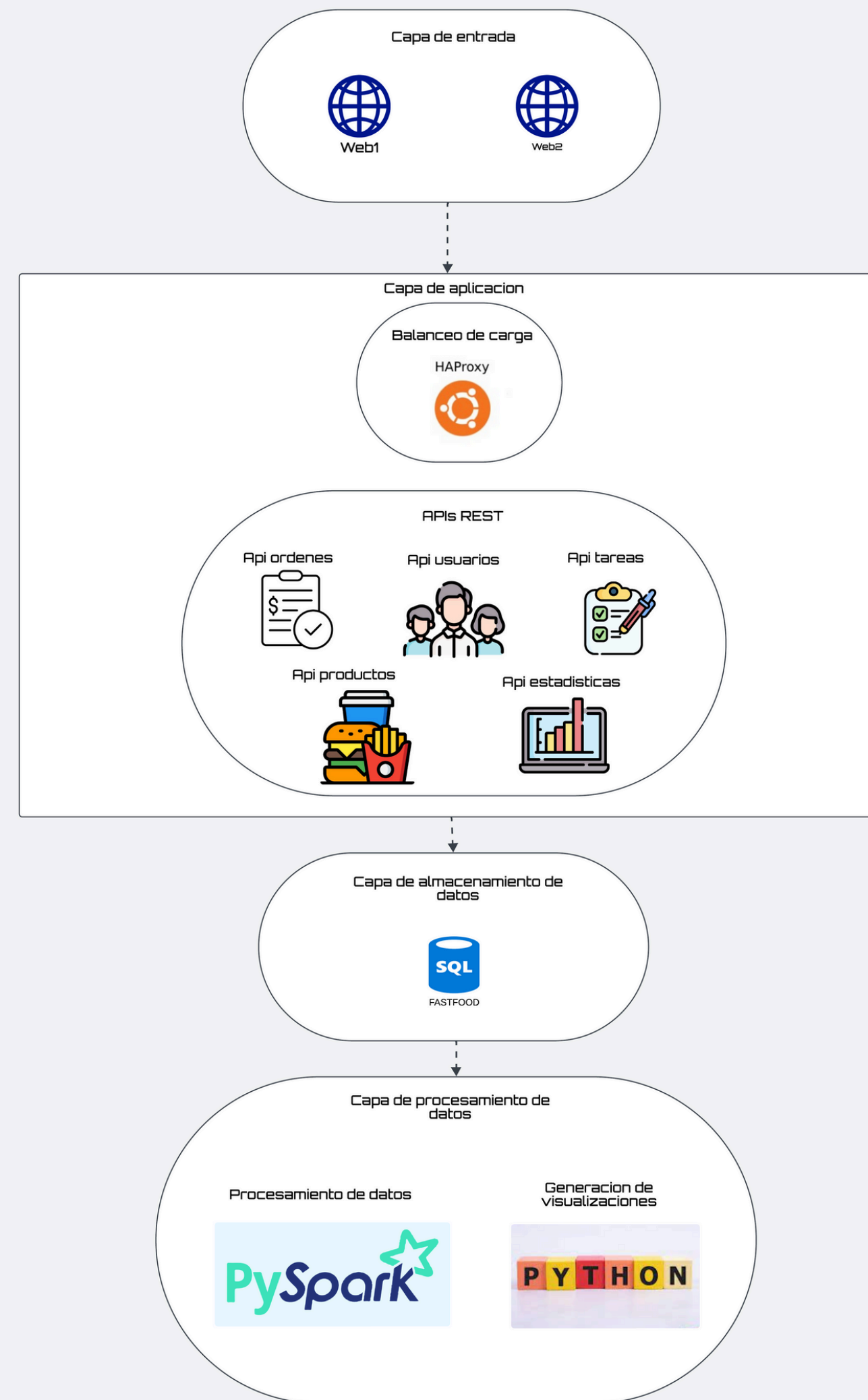
print("Archivo insert_ordenes.sql creado con éxito.")
```

Arquitectura de microservicios

- Capa de Entrada: Los servidores web reciben las solicitudes de los usuarios.
- Capa de Aplicación: HAProxy balancea las solicitudes entre las APIs (órdenes, usuarios, tareas, productos, estadísticas) para una respuesta equilibrada.
- Capa de Almacenamiento de Datos: La base de datos SQL almacena toda la información necesaria.
- Capa de Procesamiento de Datos: PySpark procesa los datos, y Python genera visualizaciones para facilitar el análisis.



Diagrama de arquitectura



Flujo de Trabajo y Descripción de Componentes

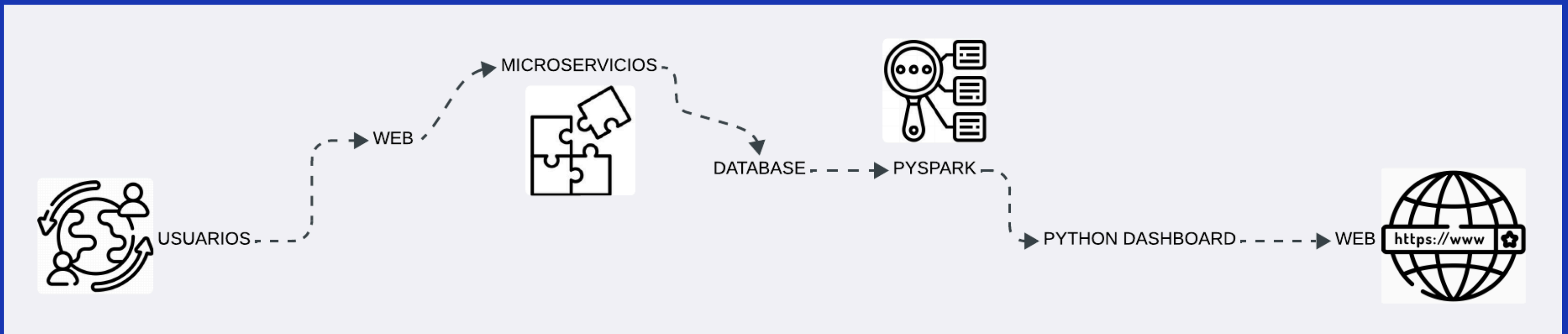
Flujo de Trabajo:

1. Cliente: Realiza pedidos y consulta el estado en tiempo real.
2. Gerente: Gestiona órdenes, asigna tareas y supervisa el personal.
3. Trabajador: Recibe tareas asignadas y actualiza el estado de los pedidos

1. Frontend: Páginas para clientes, gerentes y trabajadores.
2. Backend: Gestión de órdenes, asignación de tareas y notificaciones.
3. Base de Datos: Almacena pedidos, usuarios y estado de tareas.
4. Docker: Escalabilidad, contenedores y orquestación.



Diagrama pipeline



Utilizamos el pipeline ya que nos ayuda a visualizar cómo los datos fluyen desde los usuarios hasta el procesamiento y su presentación en la interfaz web

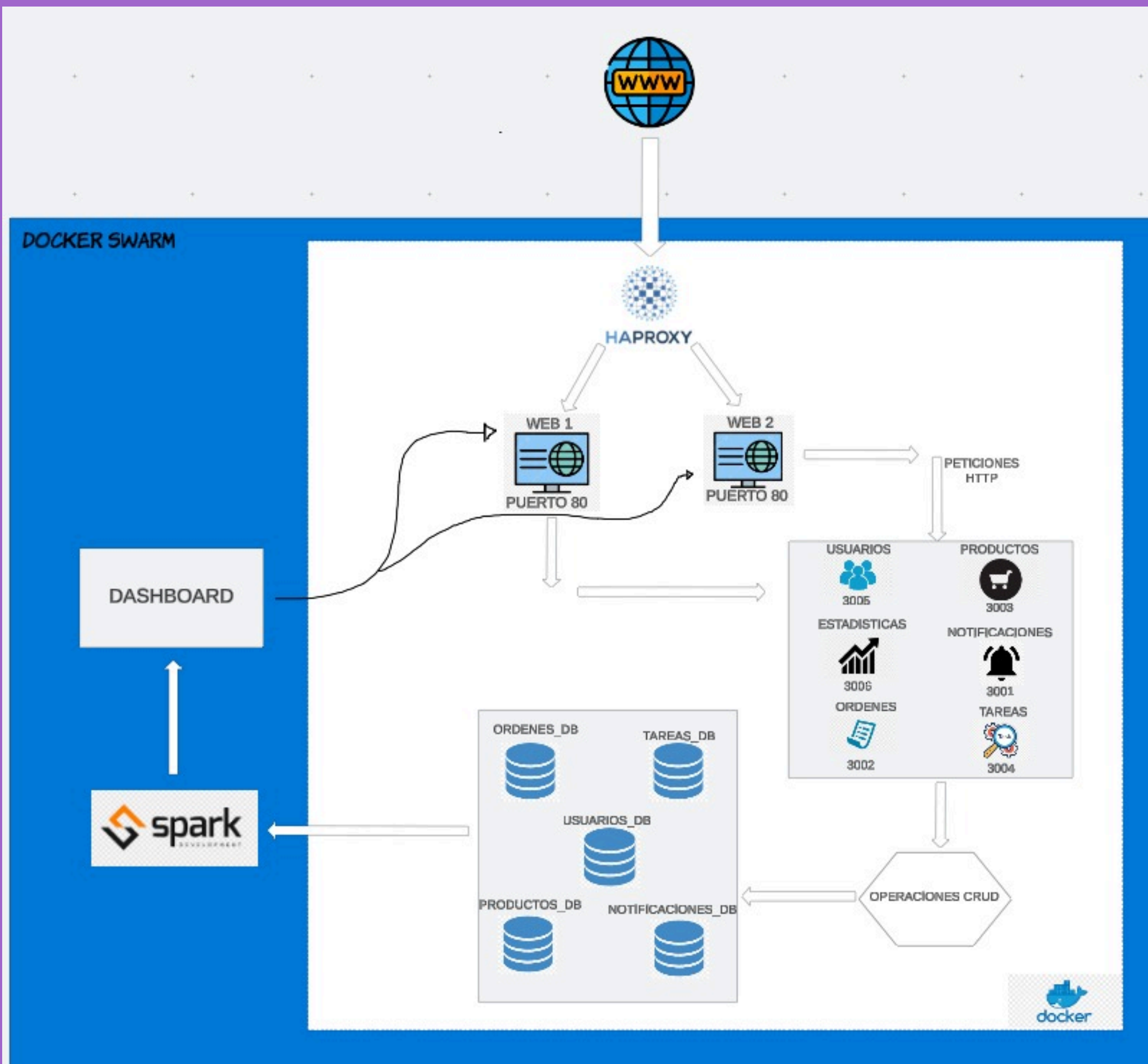
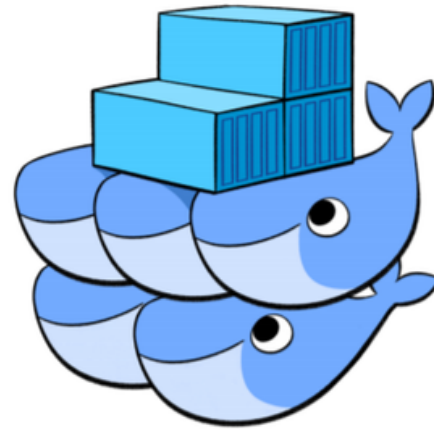


Diagrama de Despliegue

Implantación de herramientas



Usamos PySpark para gestionar pagos y transacciones en nuestra plataforma de datos distribuida.



Usamos Docker Swarm para orquestar y gestionar nuestros contenedores en un entorno distribuido.

HAProxy



Usamos HAProxy para balancear la carga entre servidores y mejorar la eficiencia de las solicitudes en tiempo real

Pruebas de Funcionamiento

```
root@servidorUbuntu:~/prueba/FASTFOOD# docker service ls
ID                NAME                MODE                REPLICAS    IMAGE                PORTS
kxczmnxnmp5b     fastfood_db         replicated          1/1         mysql:8.0           *:3307->3306/tcp
um4g9eib5hed     fastfood_estadisticas replicated          1/1         luisgar0821/estadisticas:latest *:3006->3006/tcp
vjyq85x2hsct     fastfood_haproxy    replicated          1/1         haproxy:latest       *:5080->80/tcp
k9cnz5td9jlx     fastfood_notificaciones replicated          1/1         luisgar0821/notificaciones:latest *:3001->3001/tcp
vax3k5kw2ulx     fastfood_ordenes    replicated          1/1         luisgar0821/ordenes:latest      *:3002->3002/tcp
rp7spqgfmbja     fastfood_productos  replicated          1/1         luisgar0821/productos:latest    *:3003->3003/tcp
ywyuf0hcwrli     fastfood_pyspark    replicated          1/1         luisgar0821/pyspark:latest      *:3004->3004/tcp
hhmpt60sq5p7     fastfood_tareas     replicated          1/1         luisgar0821/tareas:latest       *:3005->3005/tcp
mq77l65dz6vb     fastfood_usuarios   replicated          1/1         luisgar0821/usuarios:latest
z5g1e8j9h7ga     fastfood_web1       replicated          1/1         luisgar0821/webf:latest
ircbncnssw1r     fastfood_web2       replicated          1/1         luisgar0821/webf:latest
```

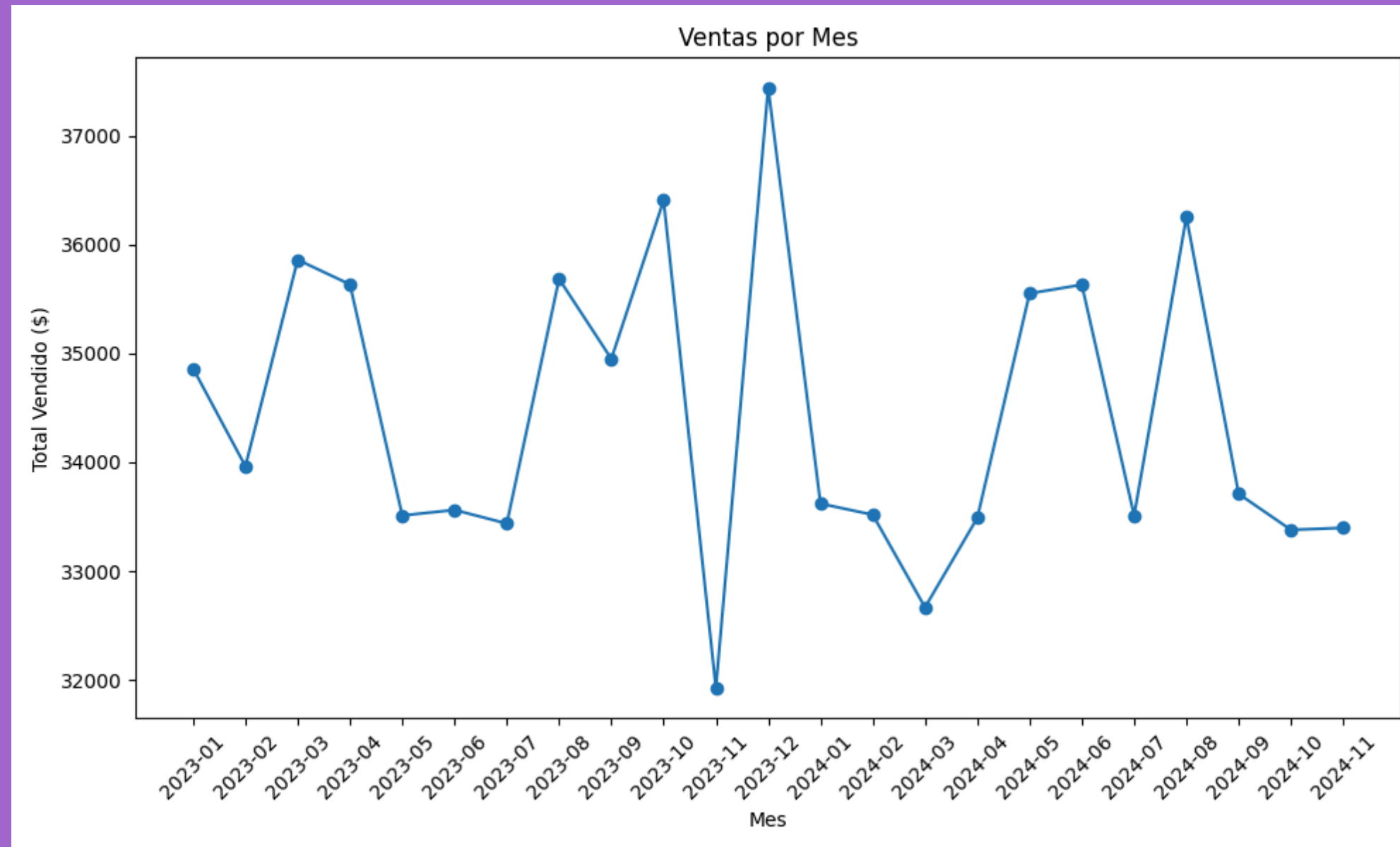
Despliegue

Escalabilidad

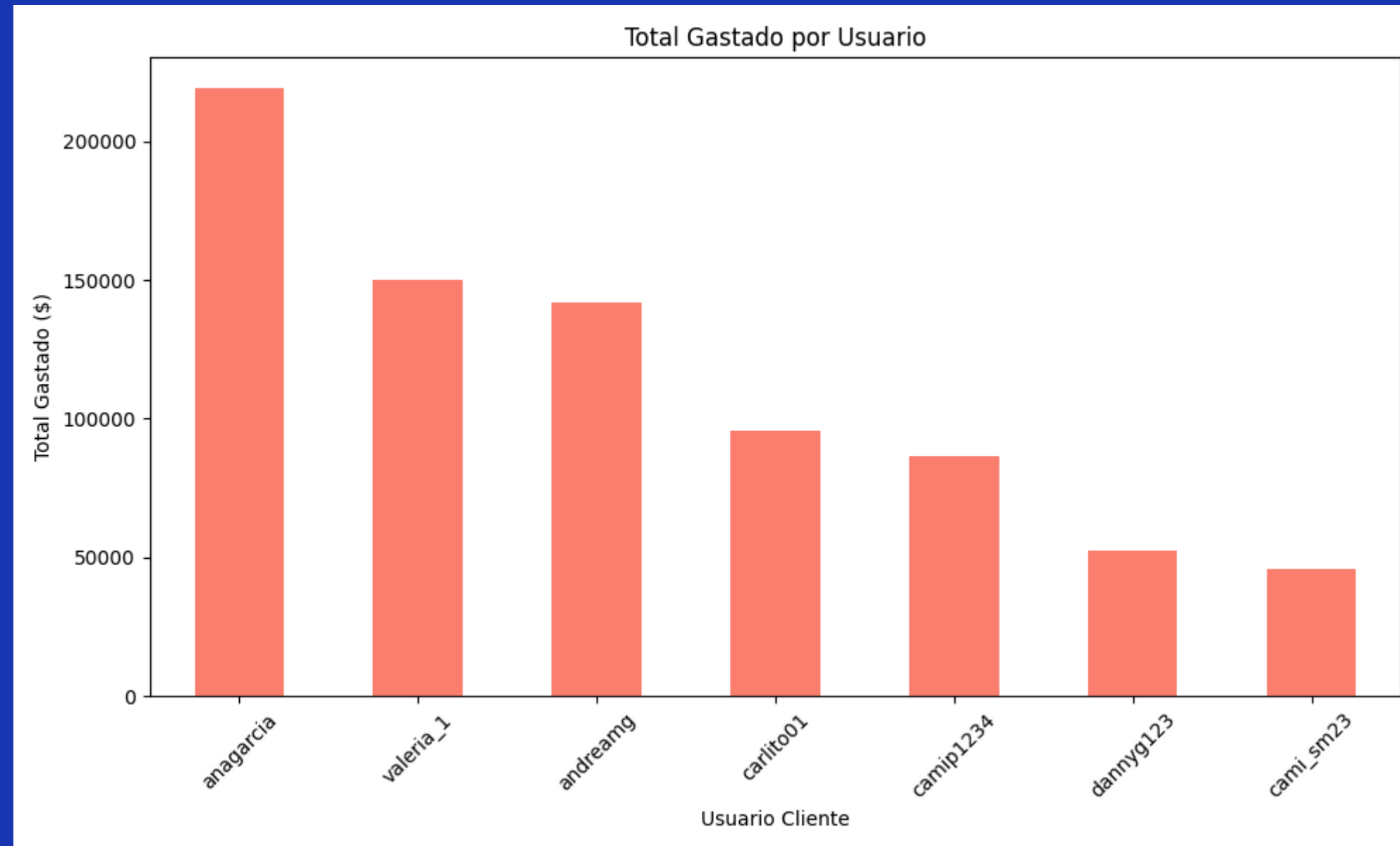
Estrés	Muestras	Replicas	% de error	rendimiento
Baja	300	1	0,00%	93,8/Sec
Media	3000	1	0,00%	315,7/Sec
Alta	10000	1	0,49%	269,0/Sec



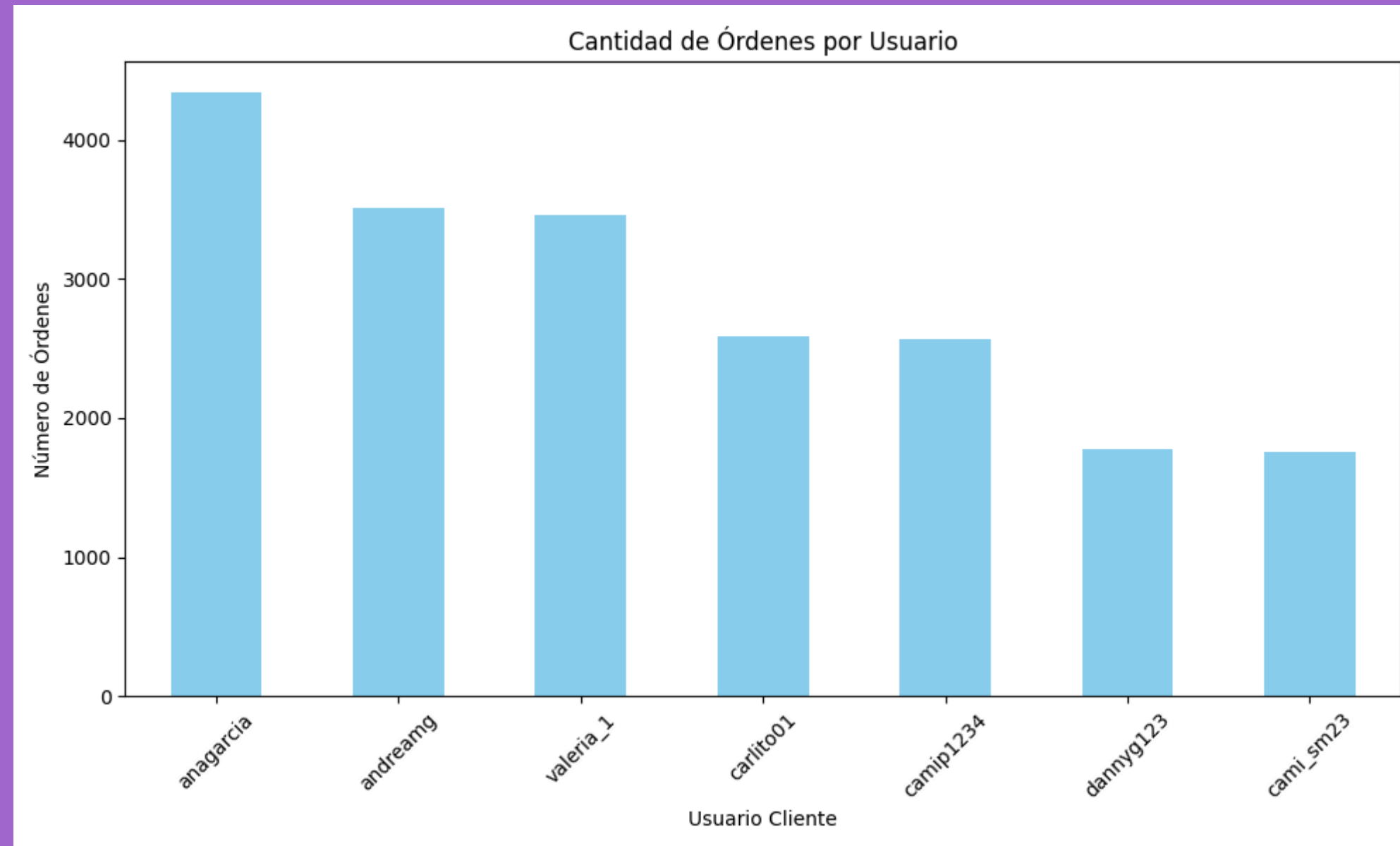
Resultados y Análisis



Resultados y Análisis



Resultados y Análisis





Conclusiones





Repositorio



Repositorio GitHub

<https://github.com/JulianaSerranoT/FASTFOOD>



Dataset venta fast food

<https://www.kaggle.com/datasets/julianaserrano/dataset-venta-fast-food>



Notion

<https://www.notion.so/FAST-FOOD-1372dc83e5ad80cc8822dbd4cd908d1b>



Documento tipo IEEE

<https://www.notion.so/FAST-FOOD-1372dc83e5ad80cc8822dbd4cd908d1b>