PROYECTO AGENCIA DE HOTELES

Redes e Infraestructura

Universidad Autónoma de Occidente

2025

Integrantes

- Sofia Gomez Rodriguez
- Victor Andres Martinez
- Juan Esteban Paredes
- Sofia Reyes Molina
- Salome Rivas Marulanda

Resumen del proyecto

La gestión de datos en sistemas de reservas hoteleras exige arquitecturas distribuidas escalables. Este trabajo presenta el diseño e implementación de una arquitectura basada en microservicios y procesamiento distribuido para analizar reservas internacionales.

Componentes claves

Aplicación de microservicios para gestión de reservas (usuarios, hoteles, habitaciones, reservas, reseñas) desplegada con Docker Swarm en un clúster de tres nodos.

Balanceo de carga

HAProxy para balanceo de carga, asegurando rendimiento óptimo y alta disponibilidad.

Procesamiento distribuido

Pipeline con Apache Spark para análisis del dataset "International Hotel Booking Analytics" de Kaggle.

La arquitectura implementa escalabilidad y balanceo de carga, logrando un despliegue exitoso de APIs REST con MySQL. Las pruebas demostraron la capacidad del sistema para procesar y visualizar análisis de datos masivos en tiempo real, generando reportes estadísticos en un dashboard.

Introducción y Análisis

Los sistemas tradicionales de agencias hoteleras, con arquitecturas monolíticas, presentan limitaciones en escalabilidad y procesamiento. La integración de microservicios con plataformas de procesamiento distribuido ofrece una solución eficiente para manejar cargas variables y grandes conjuntos de datos.

Objetivo principal

Diseñar e implementar una arquitectura distribuida basada en microservicios y Apache Spark para optimizar la gestión y análisis de información de reservas hoteleras internacionales.

Garantías del Sistema

Asegurar escalabilidad, disponibilidad y eficiencia mediante contenedores en Docker Swarm y balanceo de carga con HAProxy.

Análisis de Datos

Automatizar el procesamiento del dataset "International Hotel Booking Analytics" de Kaggle para análisis estadísticos, identificación de patrones y reportes visuales en tiempo real.

Dataset seleccionado

El dataset seleccionado de kaggle: **international-hotel-booking-analytics** contiene tres archivos principales:

- **Hoteles:** Información de hoteles globales, ubicación y algunas calificaciones.
- **Reseñas:** Detalles de reseñas de usuarios y calificaciones específicas.
- **Usuarios:** Datos demográficos de usuarios (género, edad) y fechas de entrada.

Se realizó una limpieza y unificación de estos tres archivos en uno solo, hotels_analytics.csv, para facilitar el procesamiento con Spark. Este archivo resultante contiene 29 columnas con información detallada sobre identificación, fechas, puntuaciones de aspectos del hotel (limpieza, confort), variables demográficas, datos de reseñas y usuarios, e información de referencia promedio por categoría de puntuación.

Alternativas de Despliegue y Orquestación

Para el despliegue de la aplicación y la gestión del clúster, se evaluaron diversas alternativas:

Kubernetes (K8s)	Automatiza despliegue, gestión y escalado de aplicaciones en contenedores.	Escalabilidad, administra múltiples servicios, redistribuye servicios si un nodo falla.	Necesita infraestructura robusta, sobrecoste para proyectos pequeños, alto consumo de recursos.	Múltiples microservicios que necesitan escalar automáticamente, entornos distribuidos o en la nube.
Docker Swarm	Orquesta contenedores, agrupa hosts Docker en un clúster, despliega servicios distribuidos.	Fácil de instalar y operar, se integra con CLI/API de Docker, balanceo de carga automático.	Funcionalidades limitadas, comunidad más pequeña, no ideal para infraestructuras a gran escala.	Trabajo principal con Docker, rapidez al montar entorno de orquestación sencillo, aplicaciones pequeñas/medianas.
Amazon ECS	Ejecuta, detiene y administra contenedores en un clúster sin gestión manual de servidores.	Integración con AWS, escalabilidad automática con Fargate, compatible con Docker.	Dependencia del ecosistema AWS, menos flexible que Kubernetes, puede ser costoso con Fargate.	Trabajo en AWS, despliegue rápido de contenedores sin gestión manual de nodos, microservicios de alta disponibilidad.

Alternativas de Procesamiento de Datos

Para el procesamiento de datos, se analizaron las siguientes herramientas:

Apache Spark	Motor de procesamiento distribuido para análisis de datos a gran escala.	Procesamiento en memoria, versatilidad (batch, streaming, SQL, MLlib).	Requiere mucha RAM/CPU, dependencia de sistemas de almacenamiento externos, problemas de compatibilidad JVM.	Análisis de datos masivos, procesamiento en tiempo real, aprendizaje automático, procesamiento de grafos.
Hadoop + HDFS + MapReduce	Almacena y procesa grandes volúmenes de datos (Big Data) de forma distribuida.	Escalabilidad masiva, tolerancia a fallos, procesa cualquier tipo de datos, bajo coste.	Complejidad, rendimiento lento para tiempo real, ineficiencia con archivos pequeños.	Análisis de clics, sistemas de recomendación, procesamiento de IoT, análisis de datos para sector público.
Dask (Python distribuido)	Procesa grandes volúmenes de datos en clústeres o máquinas, se integra con PyData.	API familiar para Python, escalable, flexible para procesamiento estructurado.	No es solución todo en uno, esfuerzo para Big Data empresarial, funcionalidades de orquestación limitadas.	Desarrollo en Python con dataframes/arreglos, escalabilidad necesaria, integración con ecosistema Python.

DIAGRAMA DE COMPONENTES

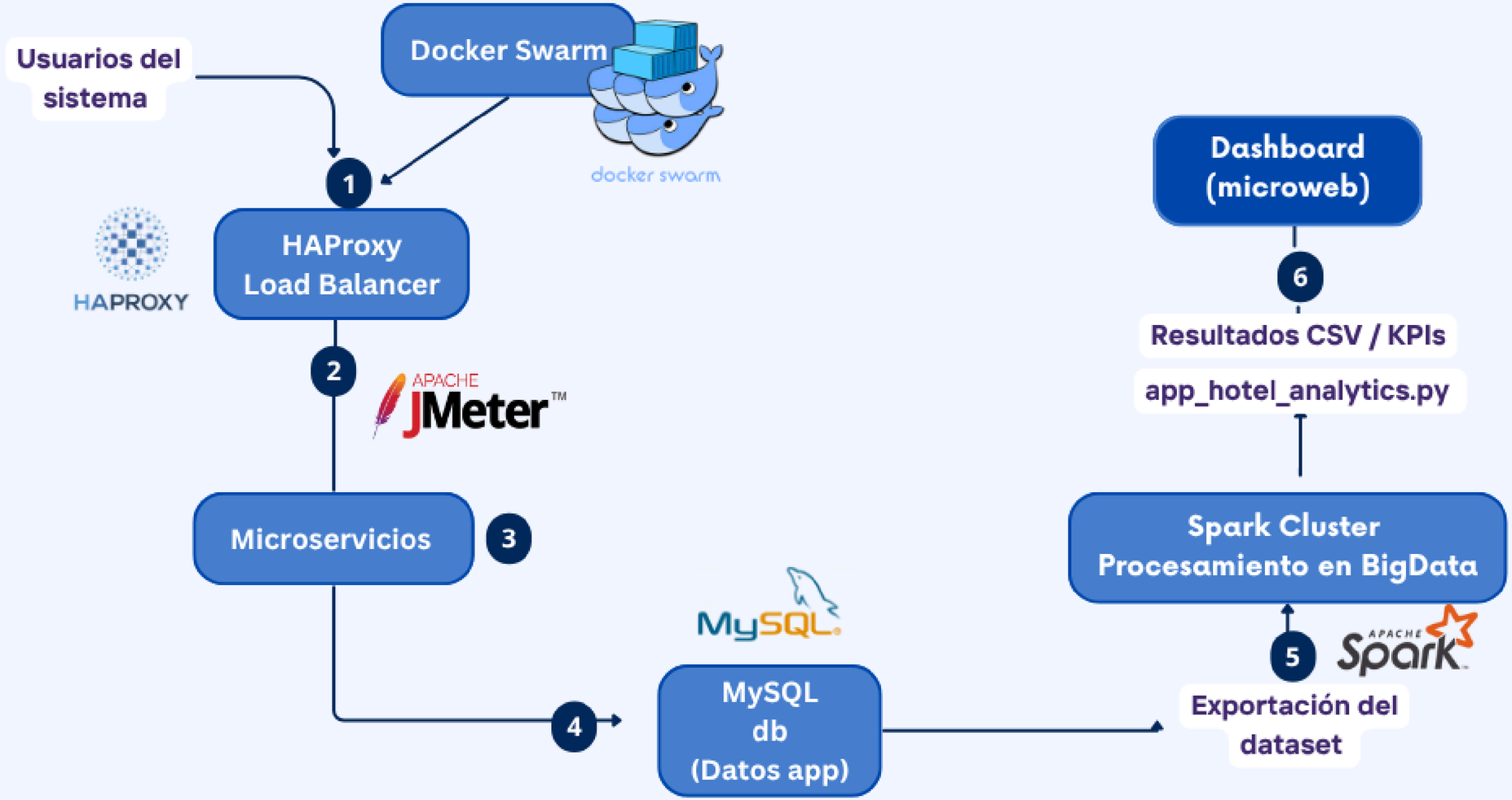
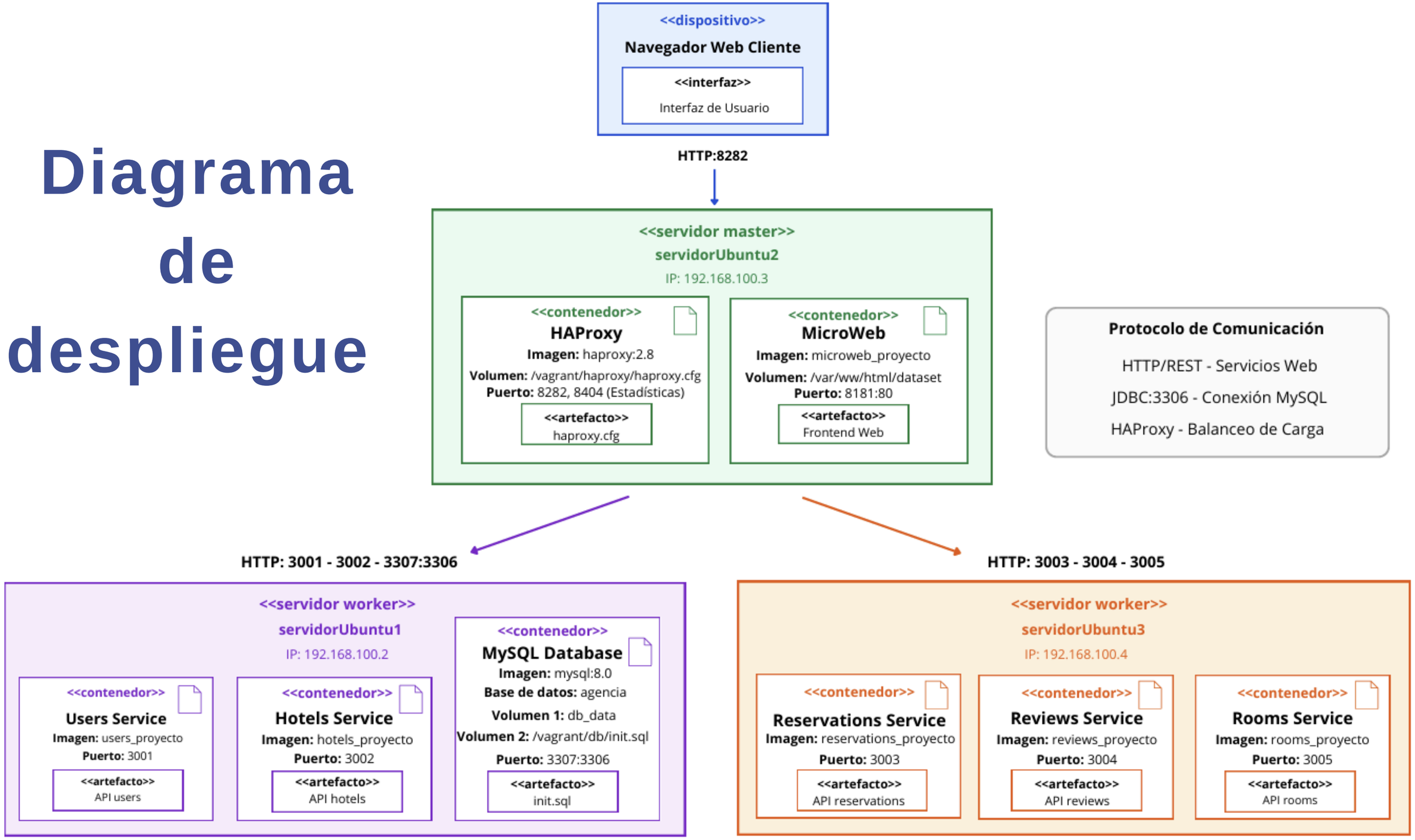
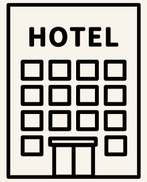


Diagrama de despliegue



Funciones de los microservicios

1 Users 

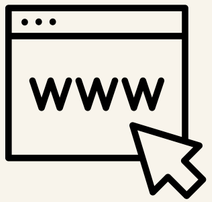


Hotels 2

3 Reservations 


Reviews 4

5 Rooms 



MICROWEB (Frontend) 6

CONCLUSIÓN

La arquitectura distribuida implementada permitió desplegar un sistema escalable, tolerante a fallos y capaz de manejar altas cargas gracias a Docker Swarm, HAProxy y MySQL. Además, Apache Spark automatizó el análisis de grandes volúmenes de datos, generando indicadores y visualizaciones útiles para la toma de decisiones.

GRACIAS