



ENIKIO

Integrantes:

Angélica Portocarrero

Camila Cardona

Mariana Mera

Santiago Rodriguez

Xilena Rojas

Vanessa Suarez

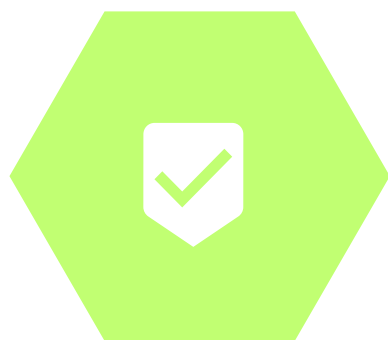


Problema

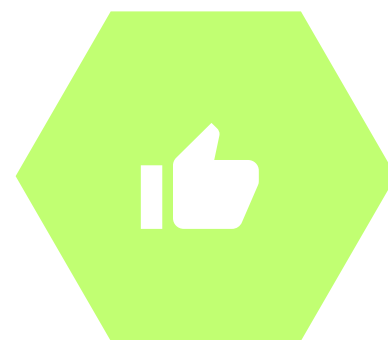
"Encuentra tu nuevo hogar cerca de la universidad y olvídate de los problemas de transporte y tiempo"



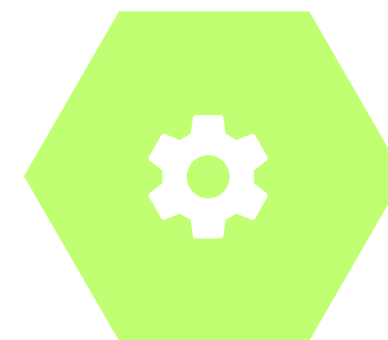
Usuarios del sistema



Postulantes



Arrendadores



Administrador



Base de datos 'enikio'

01

Tabla de usuarios

02

Tabla de Apartamentos

03

Tabla de postulaciones

04

Tabla de Universidades

05

Tabla de Ocupaciones

06

Tabla de Rangos de Precio



01

Tabla de Usuarios

Field	Type	Null	Key	Default	Extra
cc	varchar(10)	NO	PRI	NULL	
nombre	varchar(50)	YES		NULL	
email	varchar(50)	YES		NULL	
password	varchar(50)	YES		NULL	
celular	varchar(10)	YES		NULL	
rol	varchar(20)	YES		NULL	

02

Tabla de Apartamentos

Field	Type	Null	Key	Default	Extra
link	varchar(150)	YES		NULL	
id_apto	int(11)	NO	PRI	NULL	
precio	int(11)	YES		NULL	
cant_h	int(11)	YES		NULL	
coord	point	YES		NULL	
hab_disponibles	int(11)	YES		NULL	
id_arrendador	int(11)	YES		NULL	

03 Tabla de Postulaciones

Field	Type	Null	Key	Default	Extra
id_apto	int(11)	YES		NULL	
cc_postulado	varchar(10)	YES		NULL	
fecha	date	YES		NULL	
ocupacion	varchar(50)	YES		NULL	
interes	varchar(20)	YES		NULL	
estado	varchar(20)	YES		NULL	

04

Tabla de Universidades

Field	Type	Null	Key	Default	Extra
nombre	varchar(100)	YES		NULL	
coord	point	YES		NULL	
avg_precio_apto	float	YES		NULL	
num_aptos	int(11)	YES		NULL	
num_postulaciones_X_universidad	int(11)	YES		NULL	
promedio_habitacion_X_universidad	float	YES		NULL	

05 Tabla de Ocupaciones

Field	Type	Null	Key	Default	Extra
ocupacion	varchar(50)	YES		NULL	
count	int(11)	YES		NULL	

06 Tabla de Rangos de Precios

Field	Type	Null	Key	Default	Extra
rango_precio	varchar(50)	YES		NULL	
num_postulaciones_X_precio	int(11)	YES		NULL	
promedio_habitacion_X_rango_precio	float	YES		NULL	

Dataset Objetivo

ID_APTO	CC_POSTULADO	FECHA	OCUPACION	INTERES	ESTADO
---------	--------------	-------	-----------	---------	--------



Fincaraiz[®]



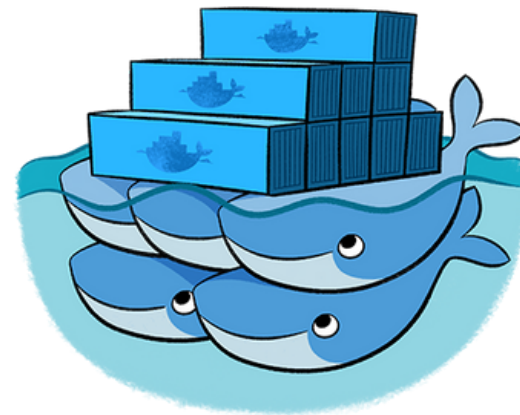
Alternativas de solución

01



Uso de Kubernetes para
implementar
contenedores

02



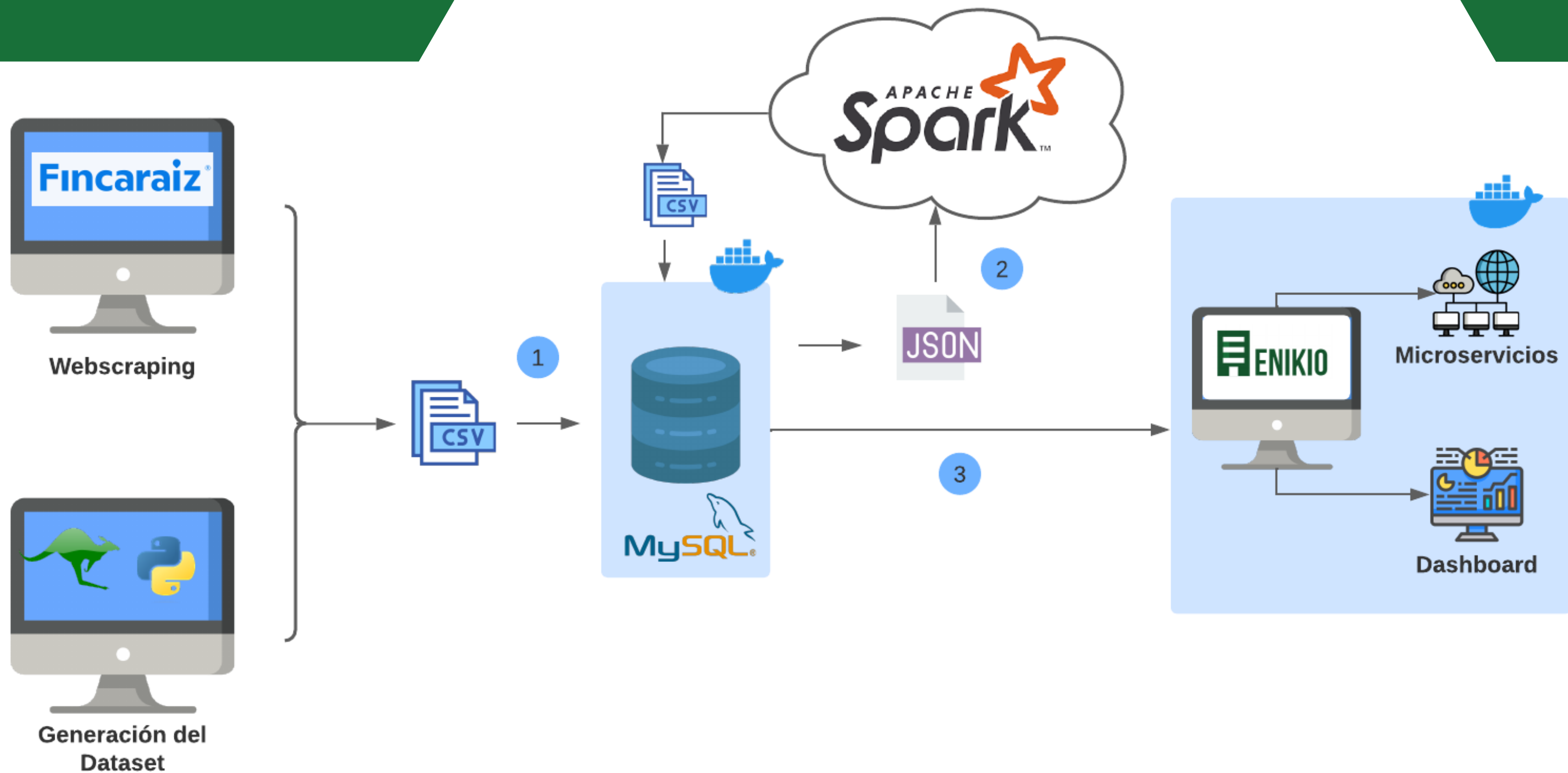
Docker Swarm para
orquestración de
contenedores

03



Spark Streaming para la
simulación de datos en
tiempo real

Pipeline



RECOLECCIÓN

PREPARACIÓN Y COMPUTACIÓN

PRESENTACIÓN

Arquitectura basada en microservicios

01

Usuarios- puerto: 3001

Se encarga de gestionar la información de los usuarios arrendadores, postulados y administradores, al igual que validar sus credenciales

02

Aptos- puerto: 3002

Gestionar la información de los apartamentos y universidades y encontrar los apartamentos más cercanos a cada universidad a través de sus coordenadas

03

Postulaciones- puerto: 3003

Gestionar la información de las postulaciones a cada apartamento para que sean visualizadas por el arrendador y añadir a la tabla de usuarios la información de un postulado la primera vez que hace una postulación

Otros servicios empaquetados

04

Microweb1

Se encarga de contener toda la parte del frontend de la aplicación, está relacionado al servicio de HAProxy para su balanceo.

05

Microweb2

Es una réplica de microweb1, tiene las mismas funciones que este y se involucra para tener un balanceo de las peticiones al servicio web.

06

Haproxy

Gestionar el balanceo de carga y redirecciona al usuario microweb1 o a microweb2. Funciona con planificación round-robin y tiene en cuenta las sesiones de cada usuario.

Pipeline

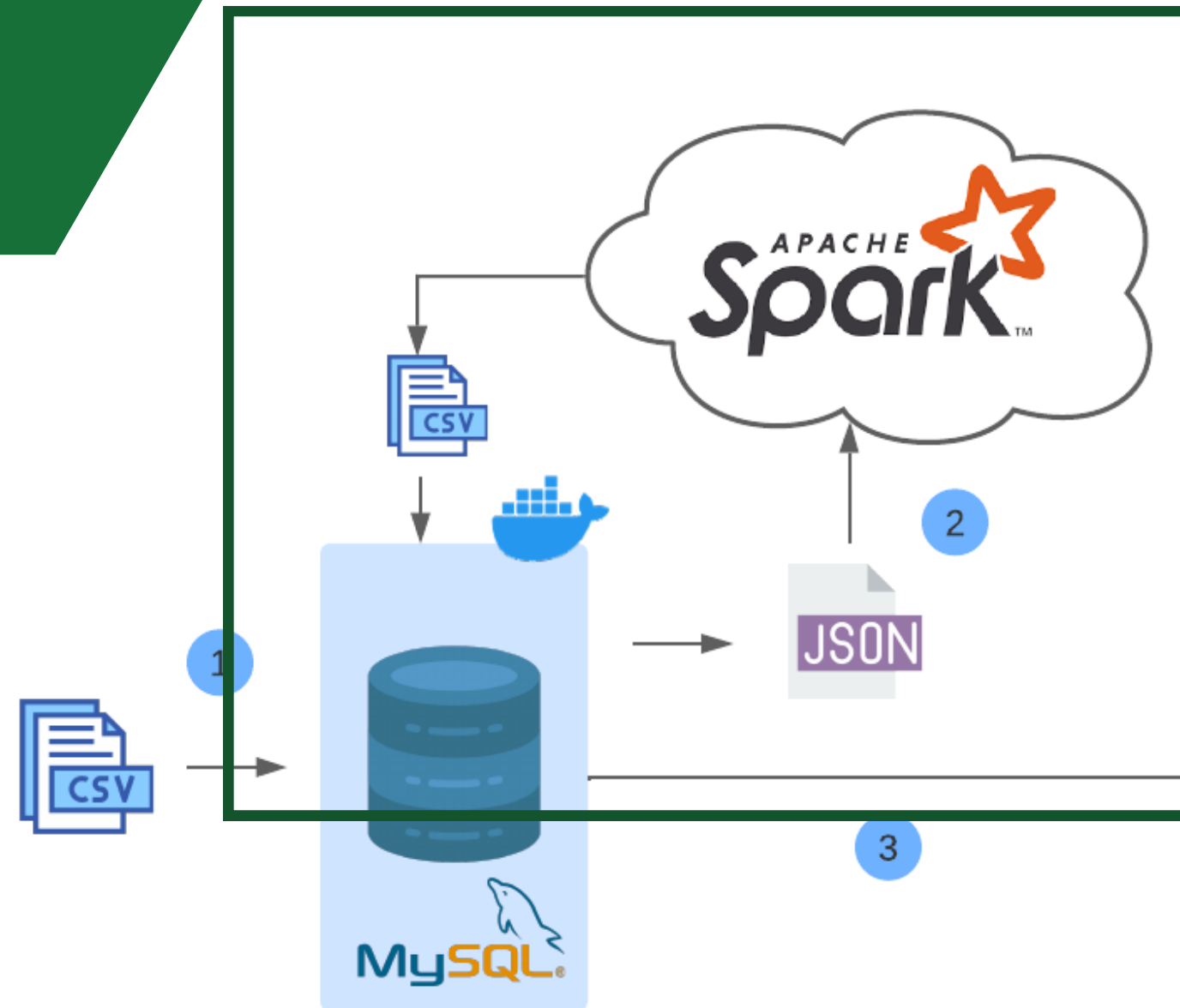


Webscrapping



Generación del
Dataset

RECOLECCIÓN



PREPARACIÓN Y COMPUTACIÓN



PRESENTACIÓN

Diagrama de Componentes

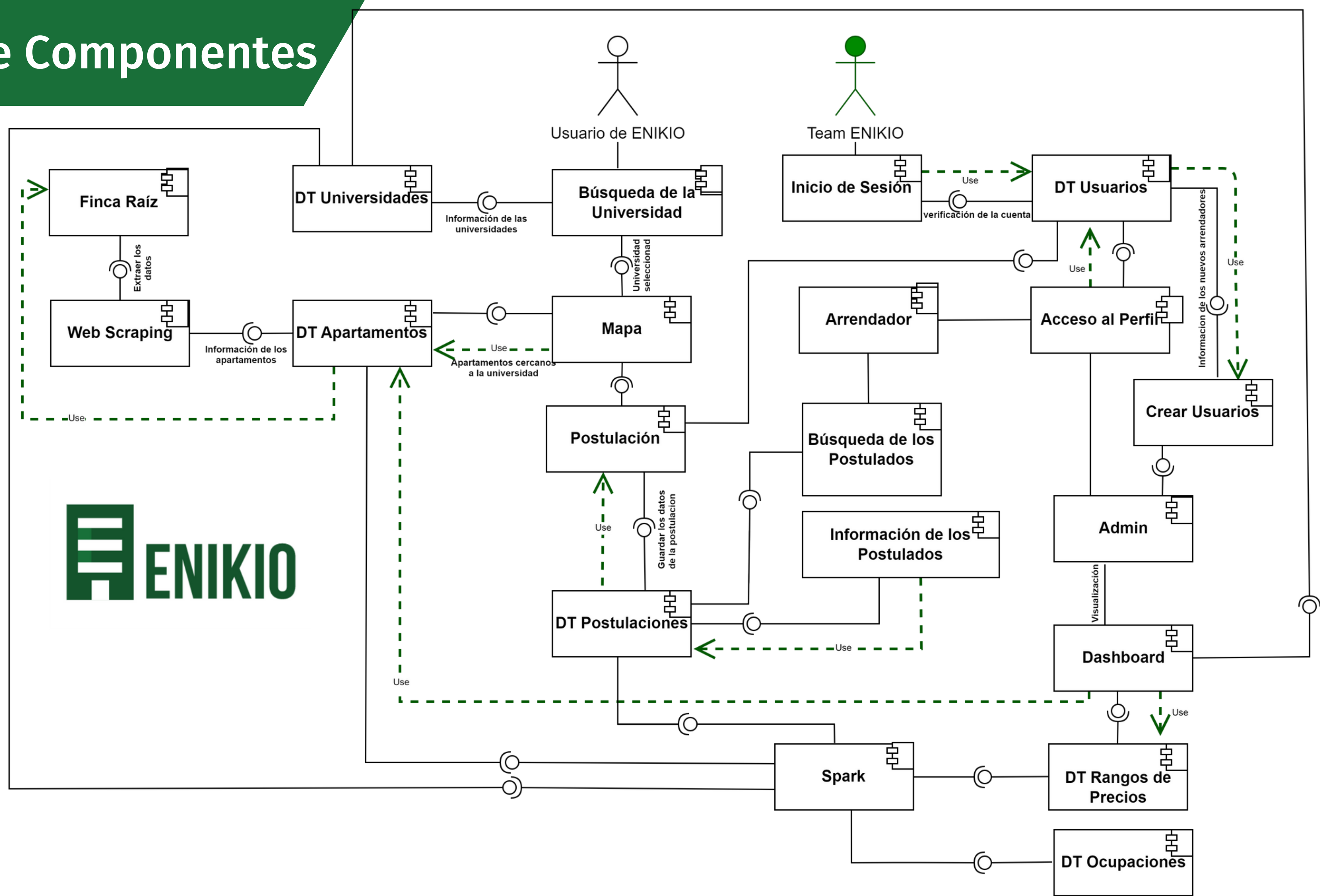
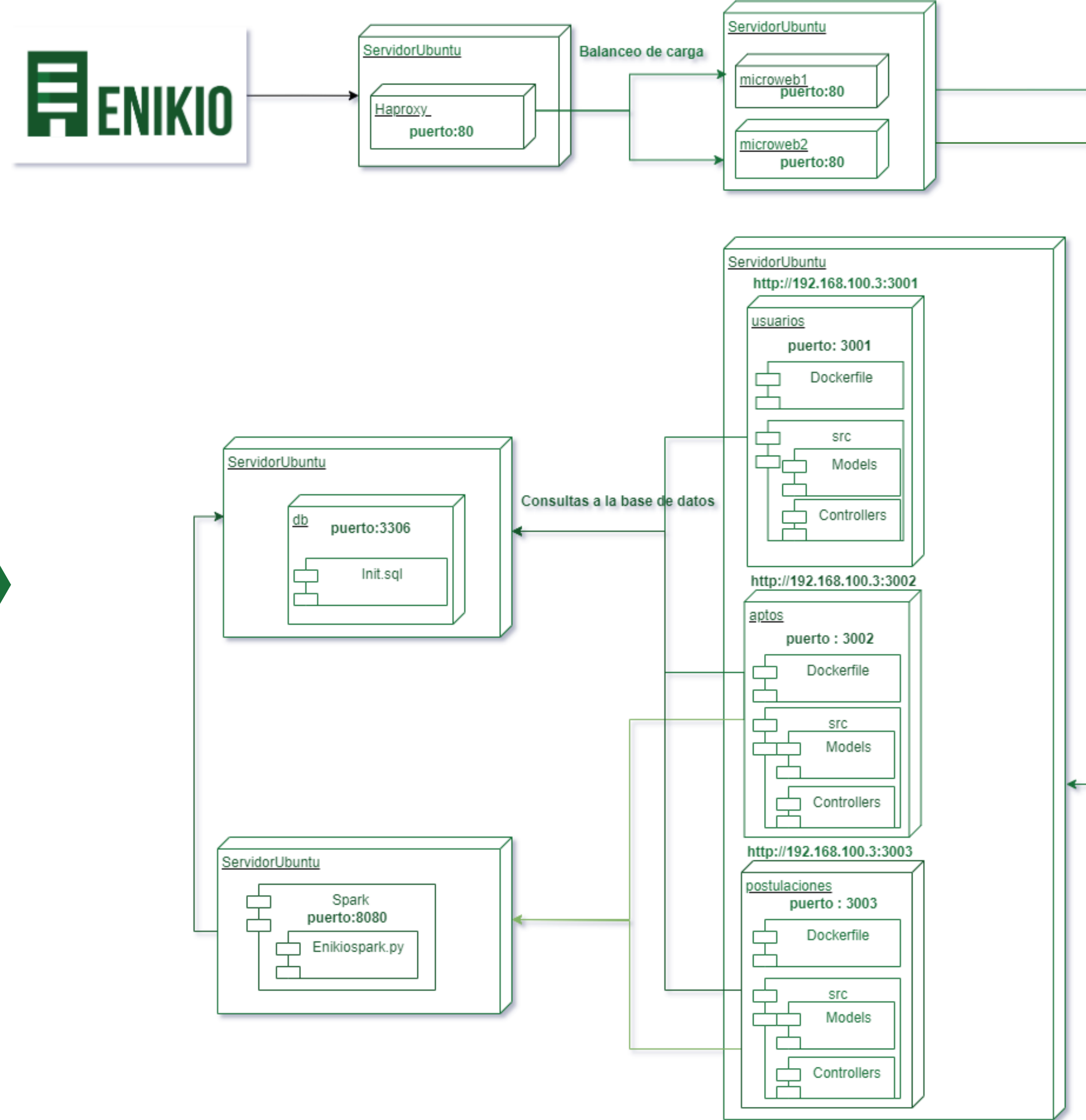
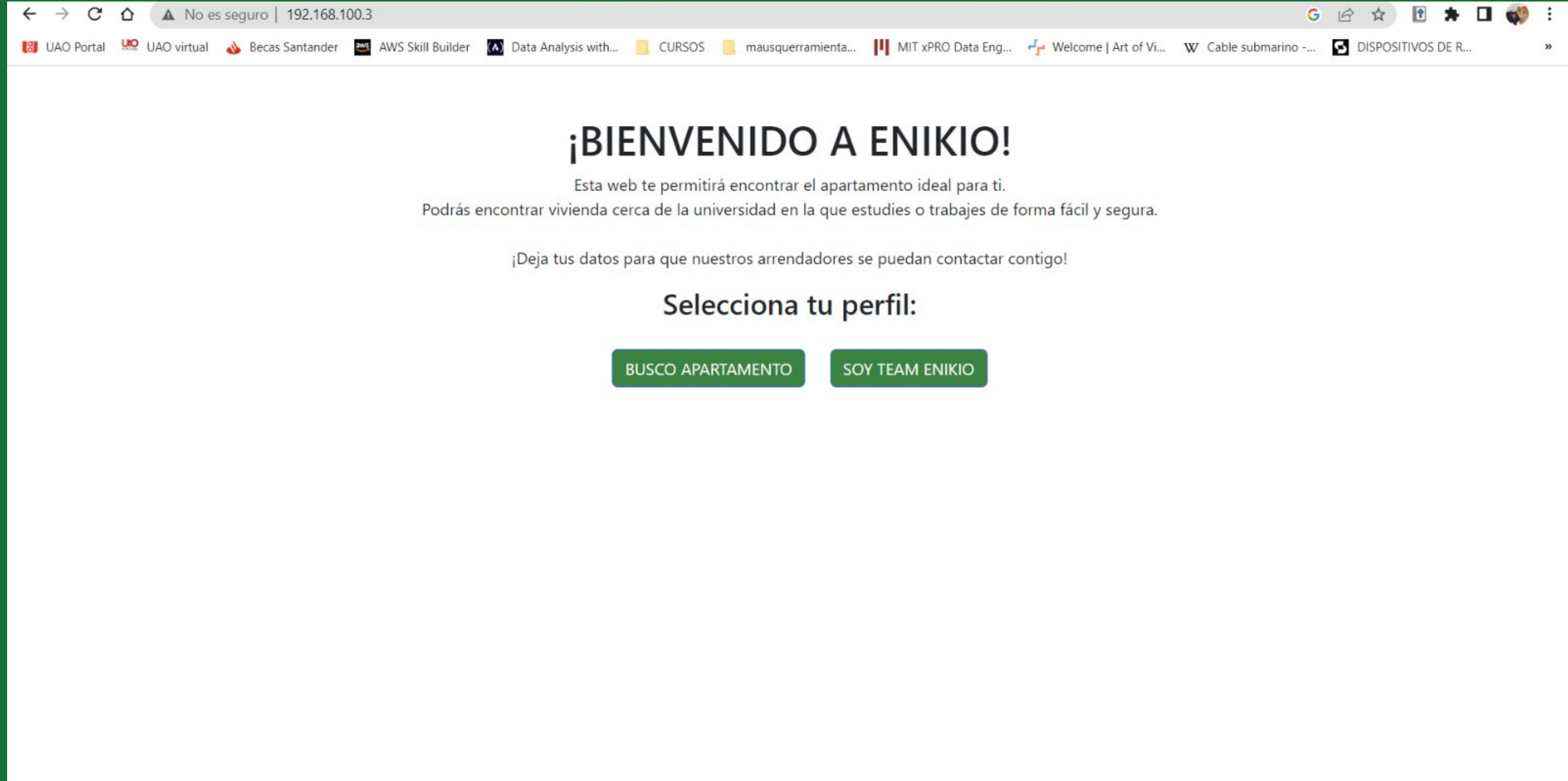


Diagrama de Despliegue



IMPLEMENTACIÓN Y PRUEBA





Conclusiones

01

El uso de una arquitectura de microservicios y API REST permite una mayor modularidad y flexibilidad en el desarrollo de la aplicación.

02

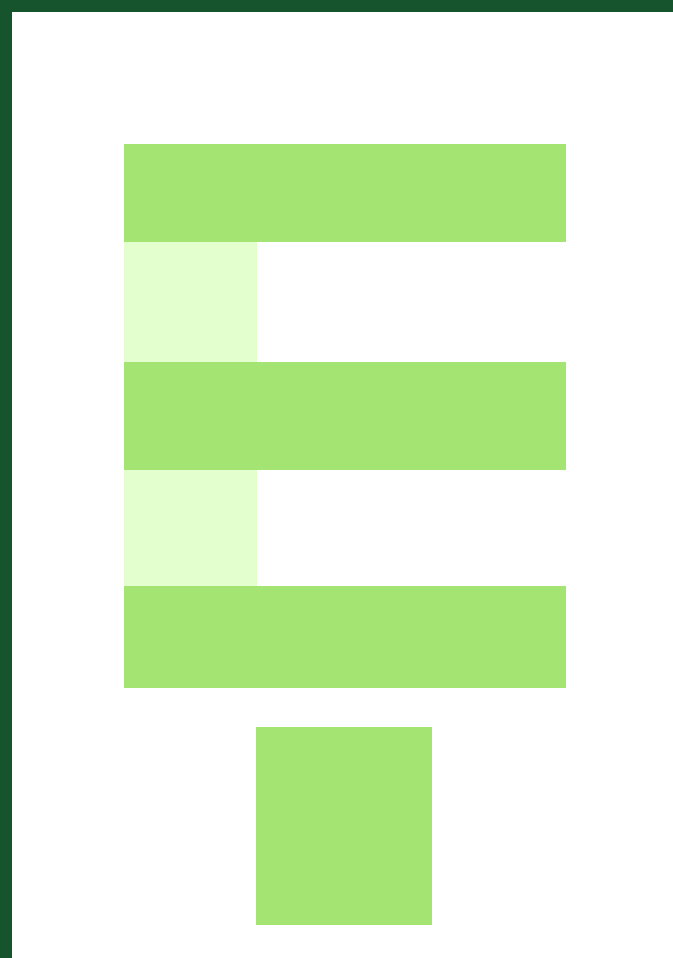
La inclusión del servicio Haproxy en el cluster permite distribuir la carga de trabajo del servicio web y garantizar la disponibilidad de la aplicación.

03

La integración de Apache Spark en el proyecto permite realizar un procesamiento eficiente de grandes volúmenes de datos, esto permite obtener insights y métricas relevantes.

04

El uso de contenedores, empaquetados en un cluster mediante Docker Compose facilita la portabilidad y la escalabilidad de los microservicios.



ENIKIO

"Tu apartamento soñado a tan solo un clic de distancia"