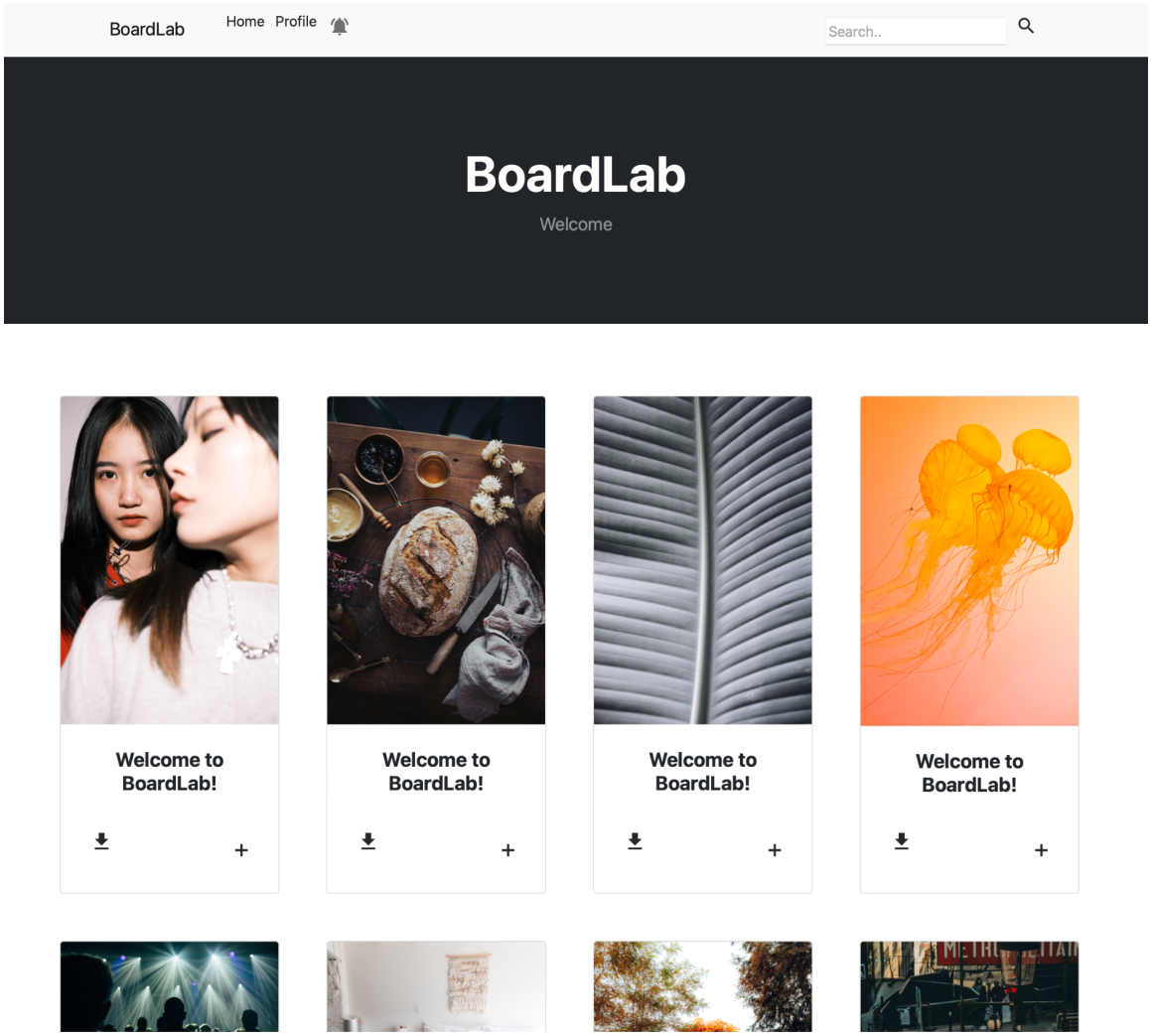
Proyecto Final Datos 2
Grupo: Piratas

- Fabricio Juarez 20190361
- Katherine García 20190418
- Andrea Reyes 20190265

Documentación BoardLab

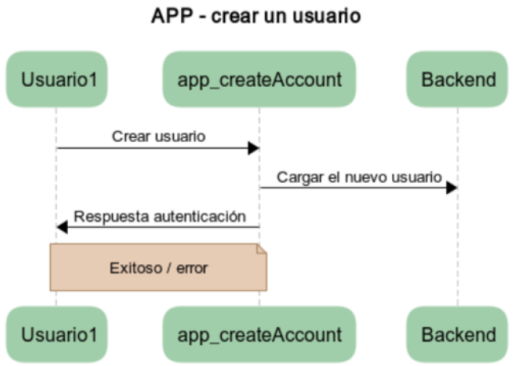
Descripción del proyecto

BoardLab es una plataforma desarrollada en Python basada en el concepto de Pinterest. El menú de la plataforma contiene tres opciones principales: home, profile y search. Los usuarios en la opción de home pueden visualizar imágenes random, las cuales pueden descargar para guardar localmente en sus computadoras o guardarlas en sus tableros personales dentro de la aplicación. En la barra de búsquedas del menú, los usuarios pueden buscar imágenes y visualizar una lista infinita de imágenes relacionadas con el tag que escribieron en el buscador. Cada usuario creado tiene acceso a su perfil que consta de un tablero con todas la imágenes que han guardado ya sea desde home o desde las búsquedas realizadas.

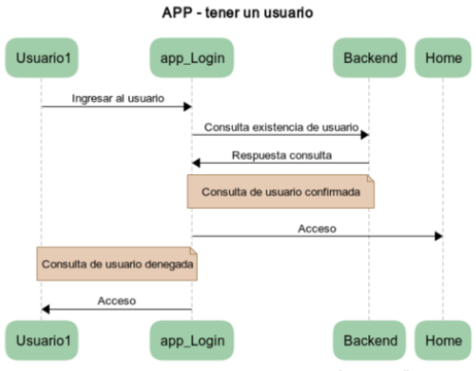


UML

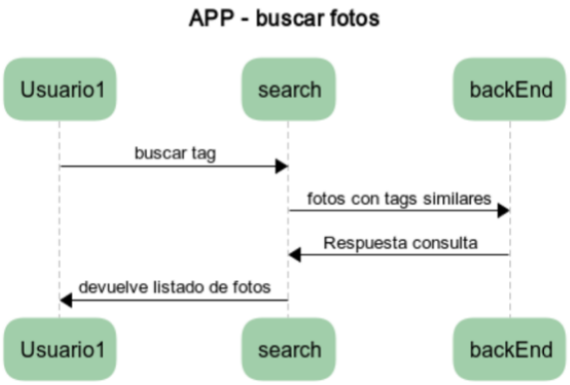
Crear usuario:



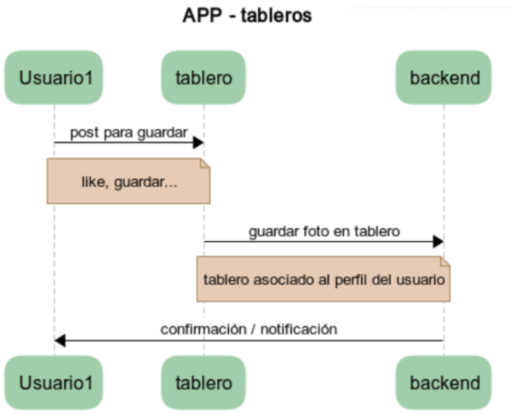
Login:



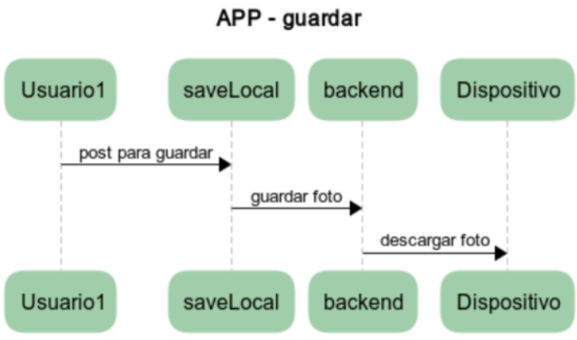
Buscar imagenes:



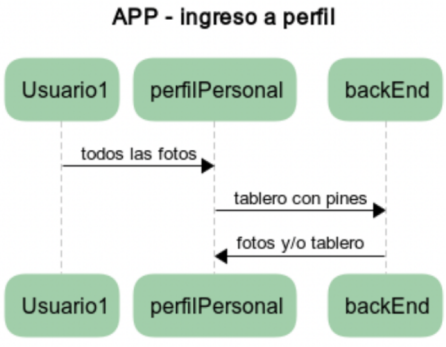
Tableros:



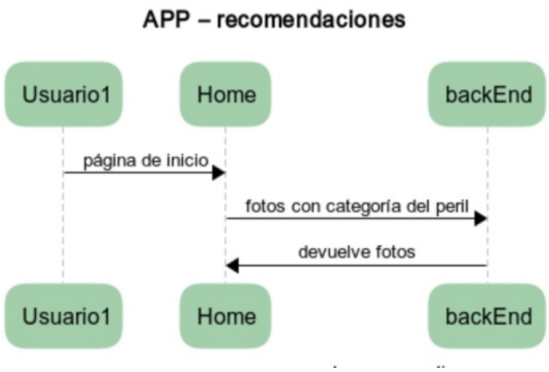
Guardar:



Perfil:



Home:



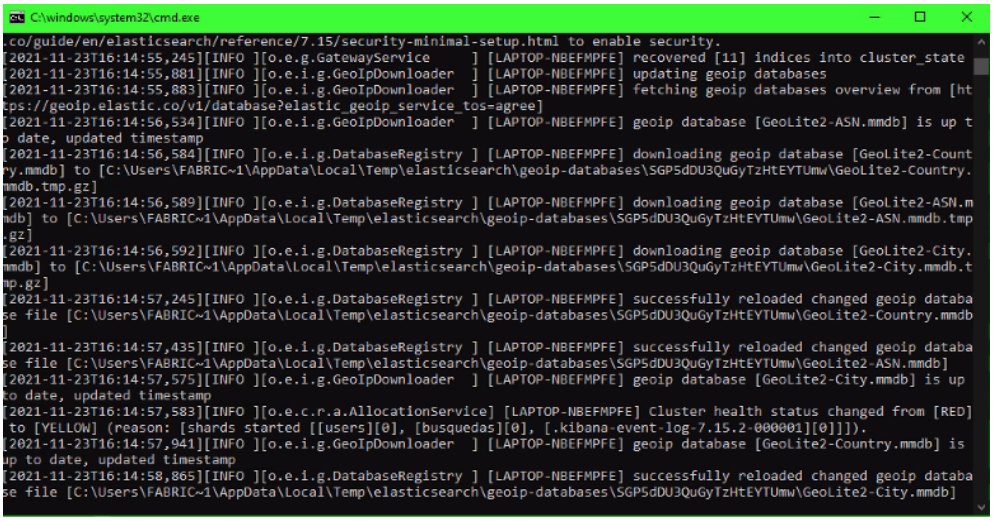
Ejecución

Para que este proyecto funcione a su máximo potencial es necesario levantar distintos servicios previos a correr la app principal:

- *ElasticSearch*:
  - .\elasticsearch-7.15.2\bin\elasticsearch.bat
- *Caché*:
  - memcached.exe -d start
- *Kafka*:
  - .\bin/zookeeper-server-start.sh config/zookeeper.properties
  - .\bin\windows\kafka-server-start.bat .\config\server.properties
- *Logstash*:
  - .\bin\logstash -f .\config\logstash.yml
- Python:
  - python3 app.py

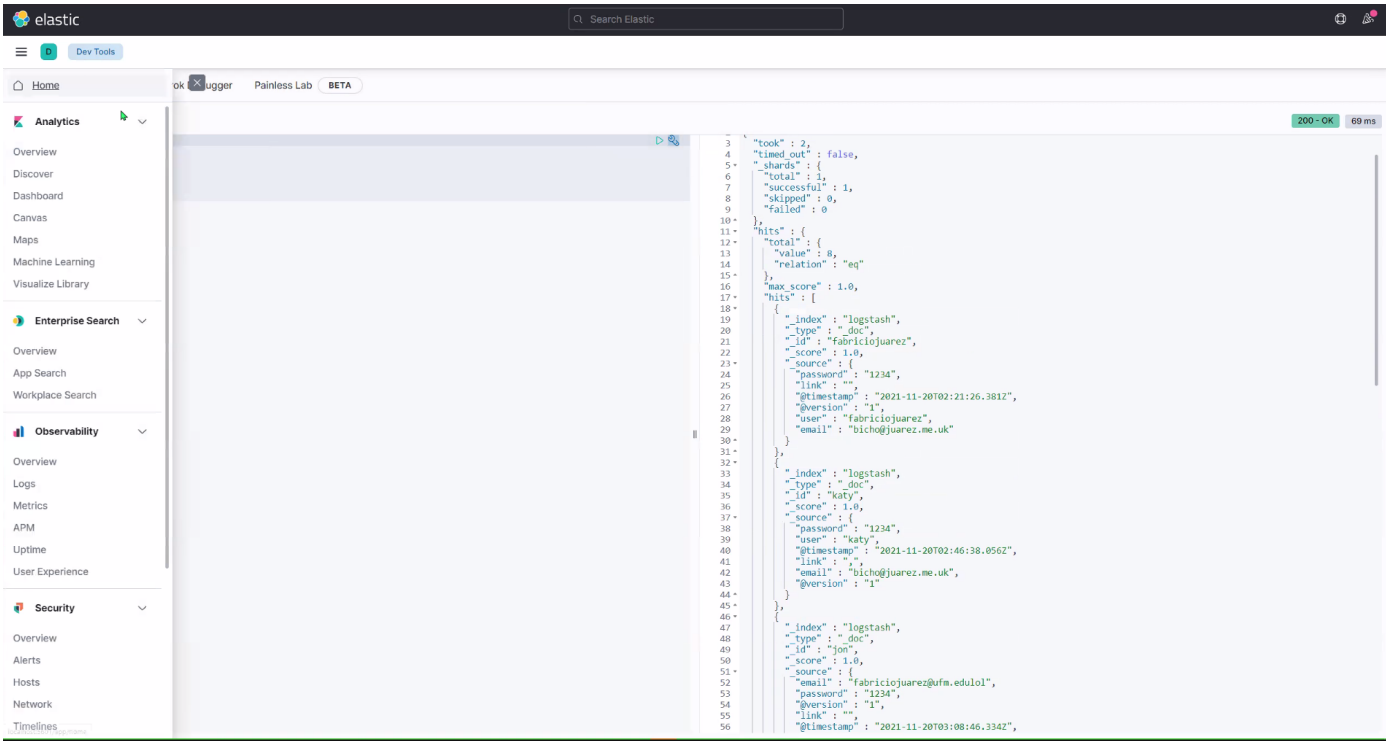
App
Se puede acceder a la plataforma con el siguiente url: http://localhost:5000/

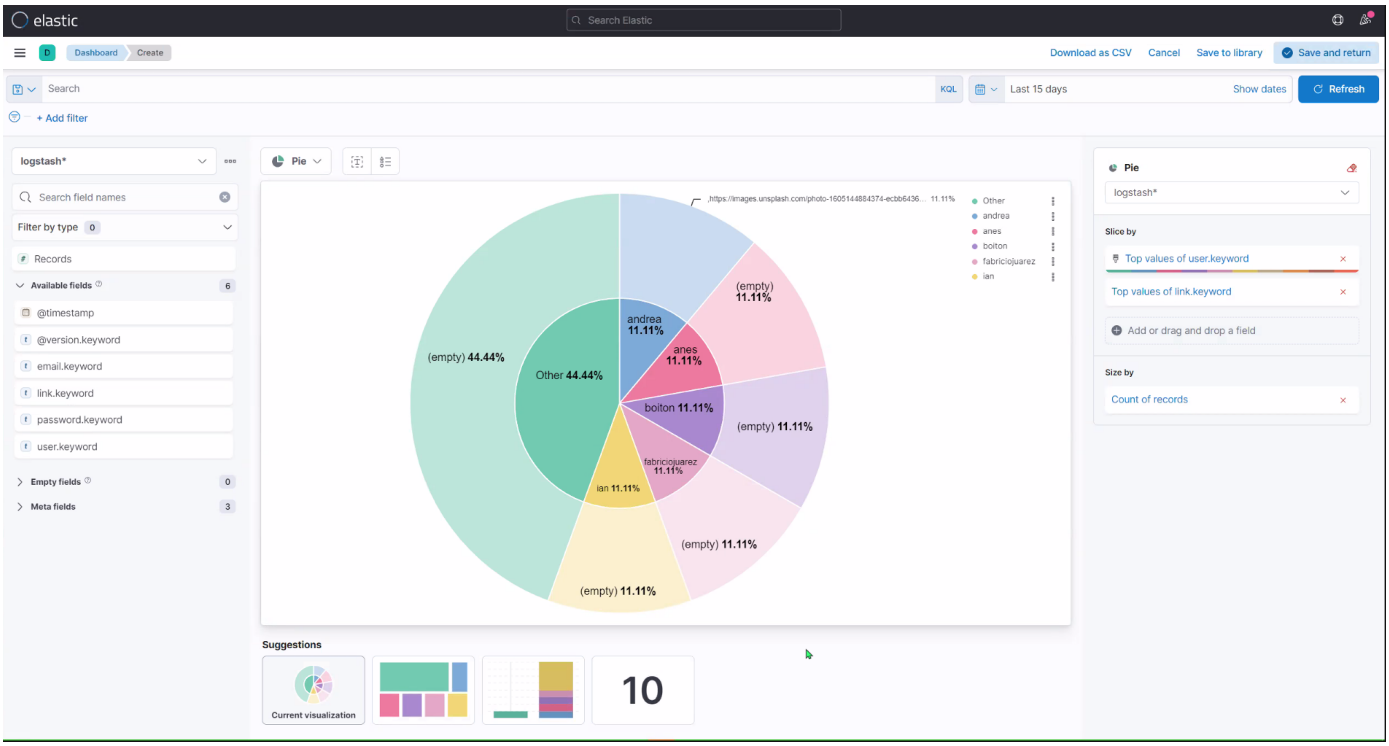
ElasticSearch
El proyecto cuenta con 2 nodos que se ejecutan de la siguiente manera:

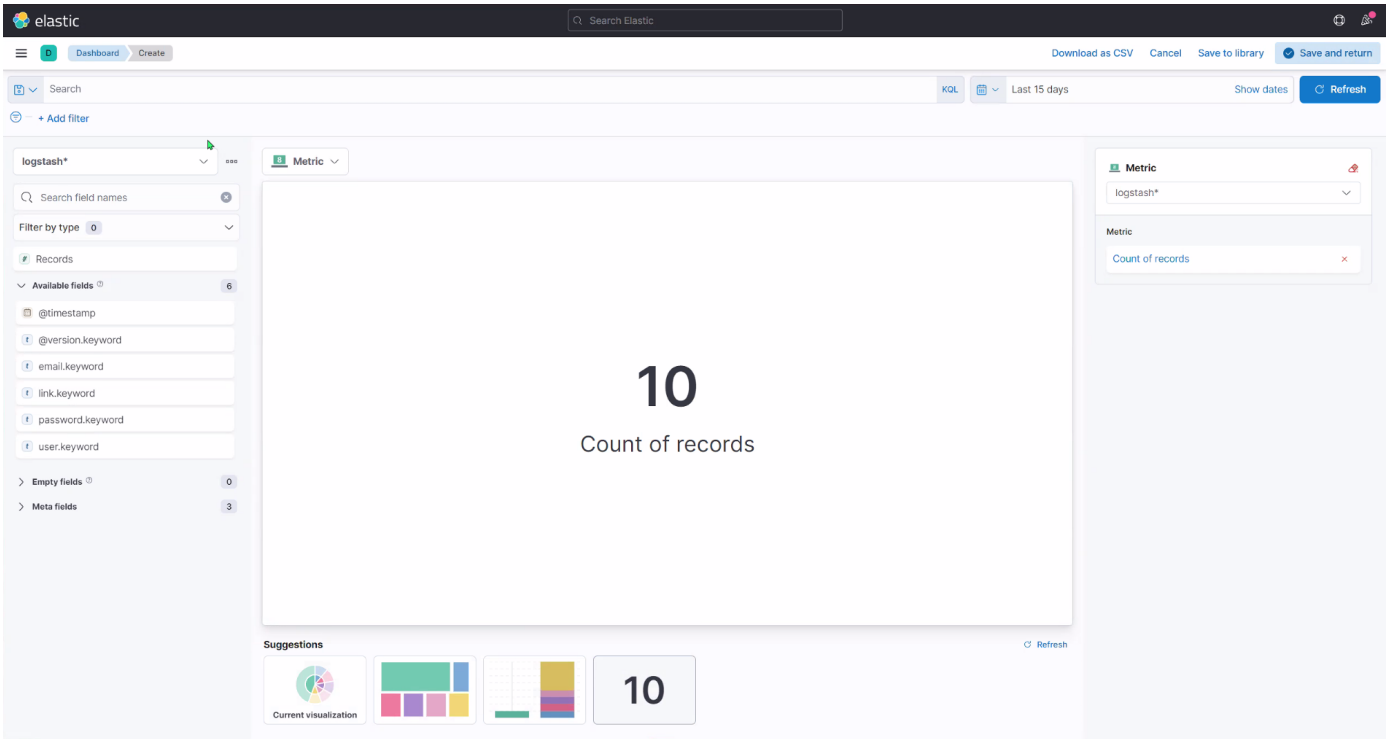


Kibana

Las siguientes capturas de pantalla son ejemplos del dashboard de Kibana de BoardLab:

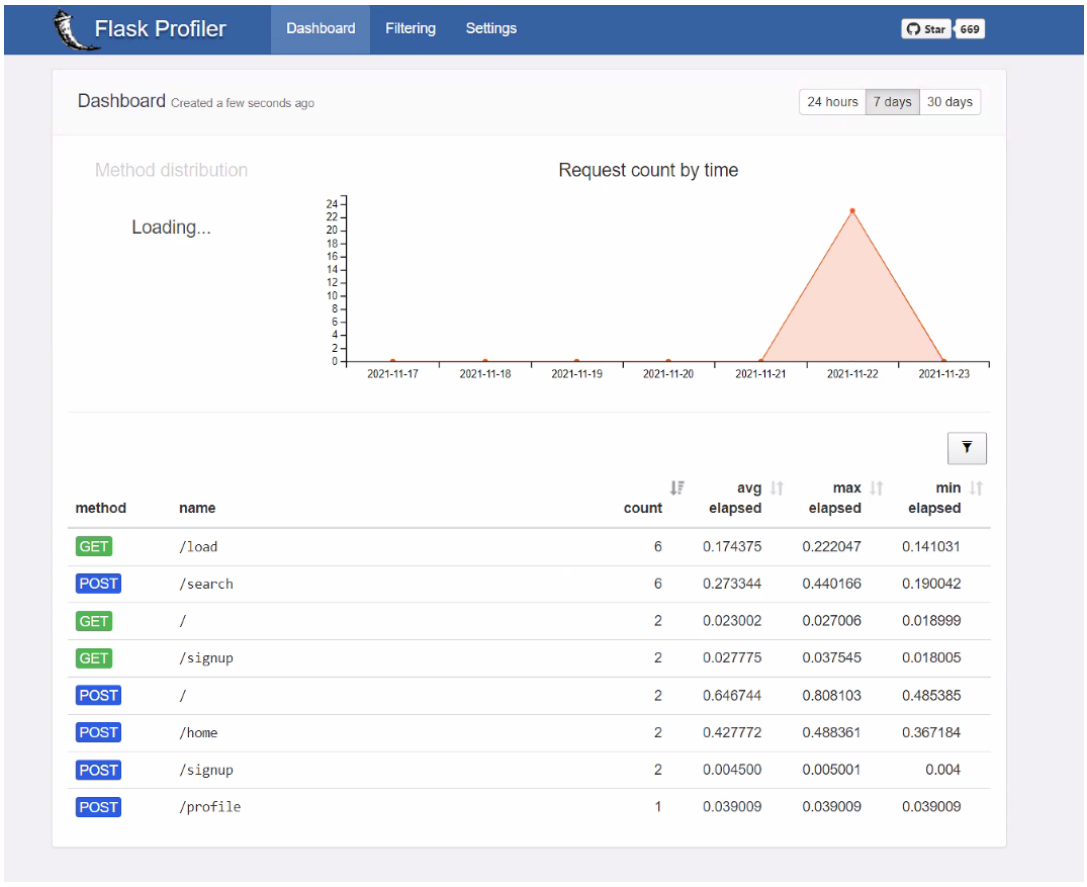






Profiler

Para realizar profiling al proyecto se utilizó flask-profiler. Después de ejecutar todas las opciones de la plataforma, se accedió a la información que genera flask-profiler para visualizar las gráficas obtenidas:

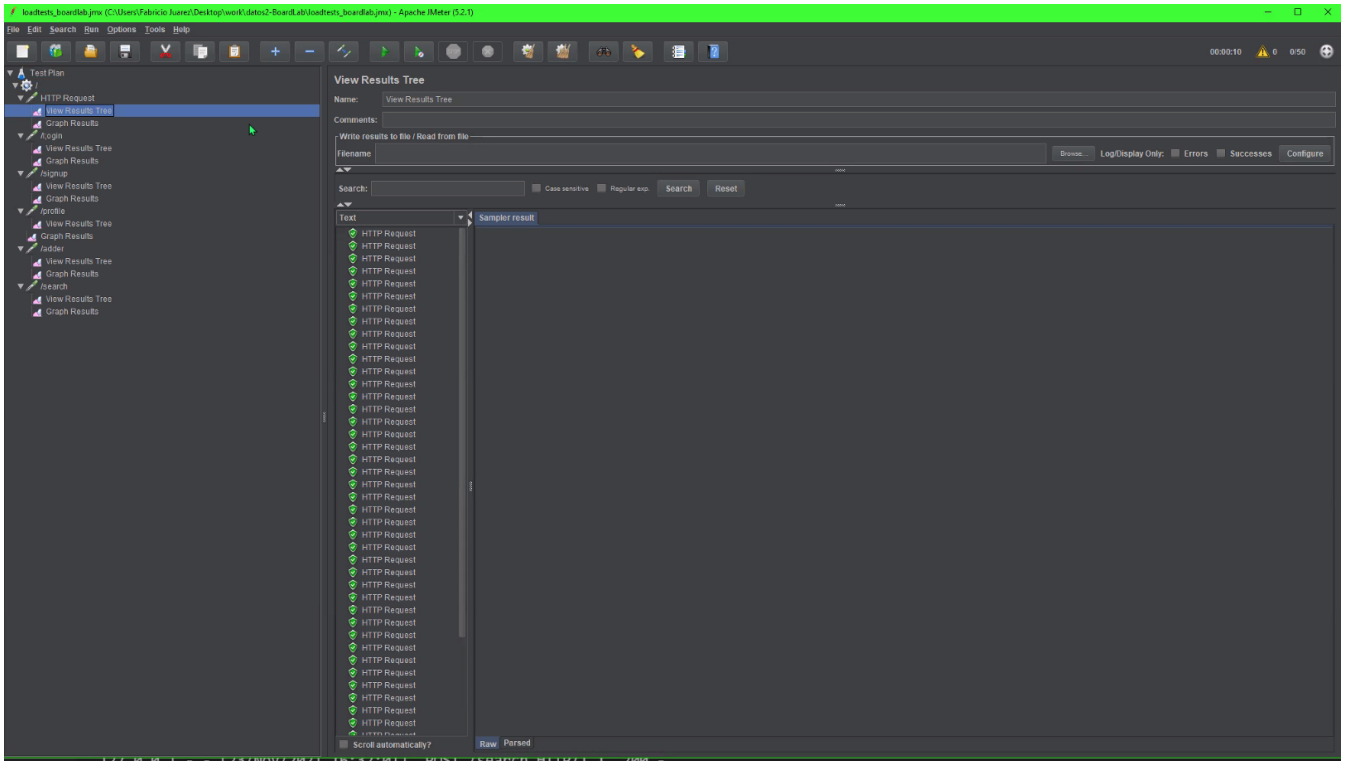


method	name	count	avg elapsed	max elapsed	min elapsed
GET	/load	6	0.174375	0.222047	0.141031
POST	/search	6	0.273344	0.440166	0.190042
GET	/	2	0.023002	0.027006	0.018999
GET	/signup	2	0.027775	0.037545	0.018005
POST	/	2	0.646744	0.808103	0.485385
POST	/home	2	0.427772	0.488361	0.367184
POST	/signup	2	0.004500	0.005001	0.004
POST	/profile	1	0.039009	0.039009	0.039009

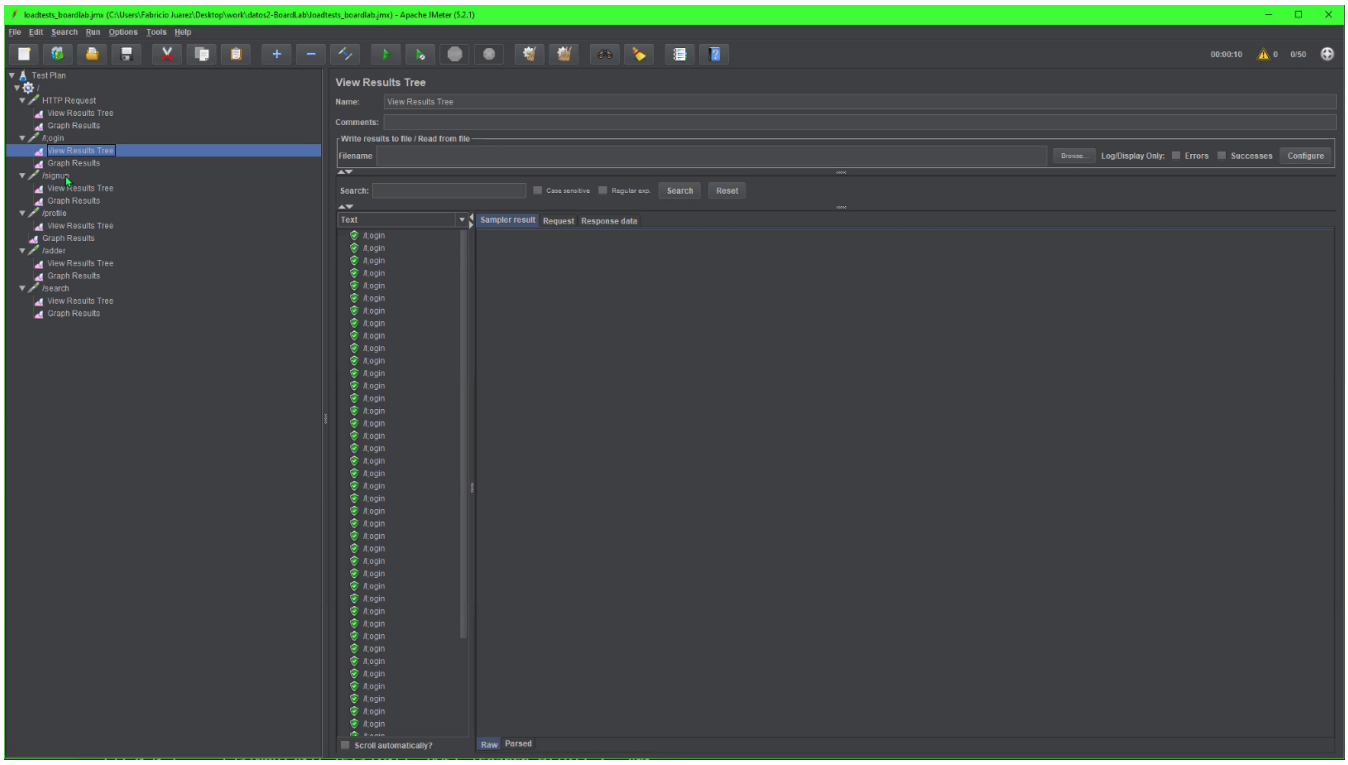
Pruebas de carga

Las pruebas de carga del proyecto se realizaron con Jmeter, con las cuales se obtuvieron los siguientes resultados:

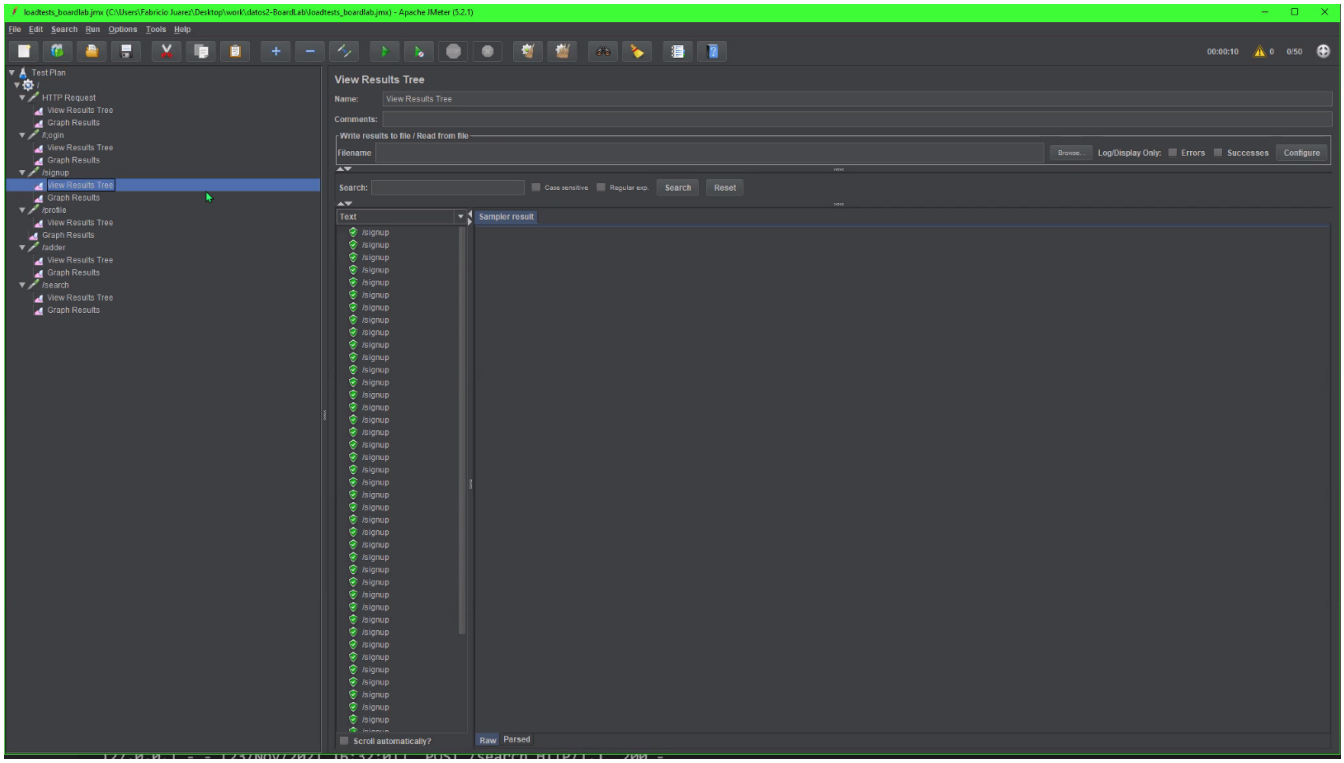
/



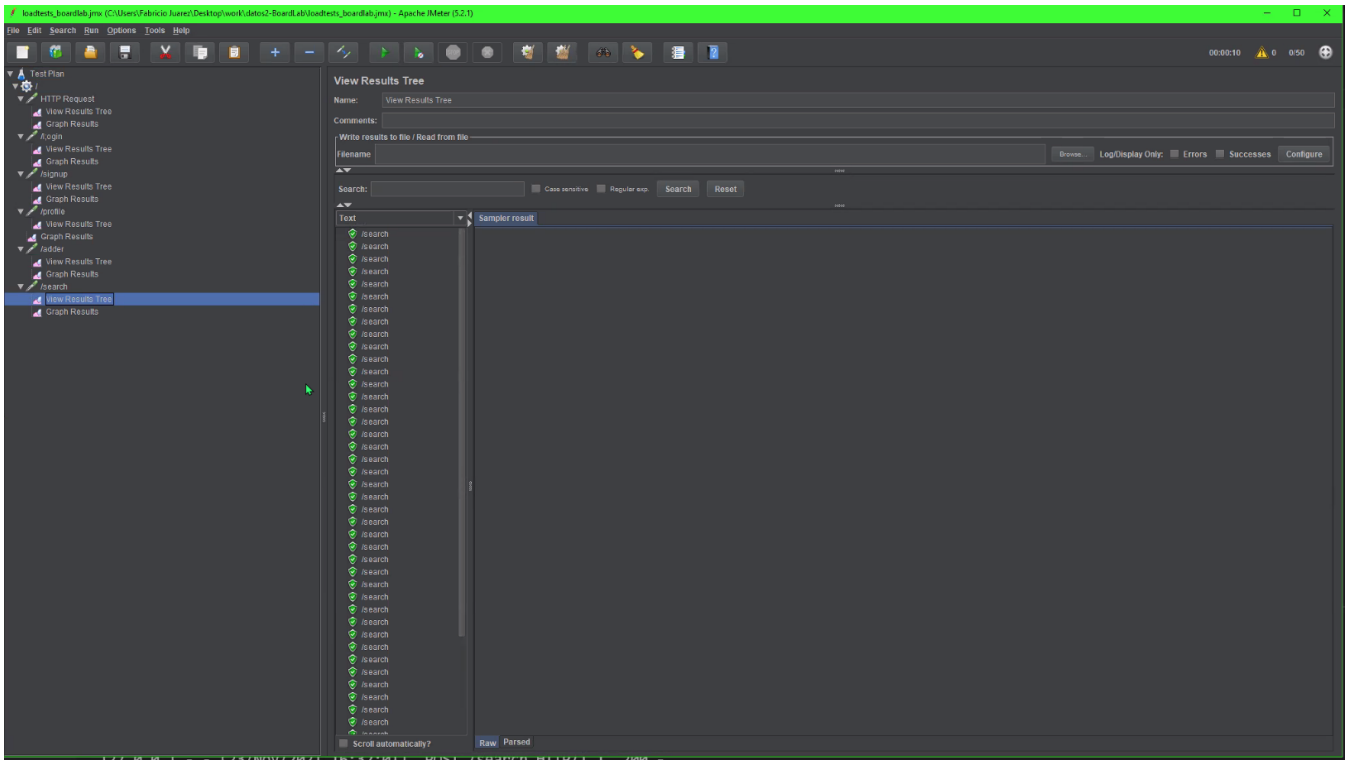
Login



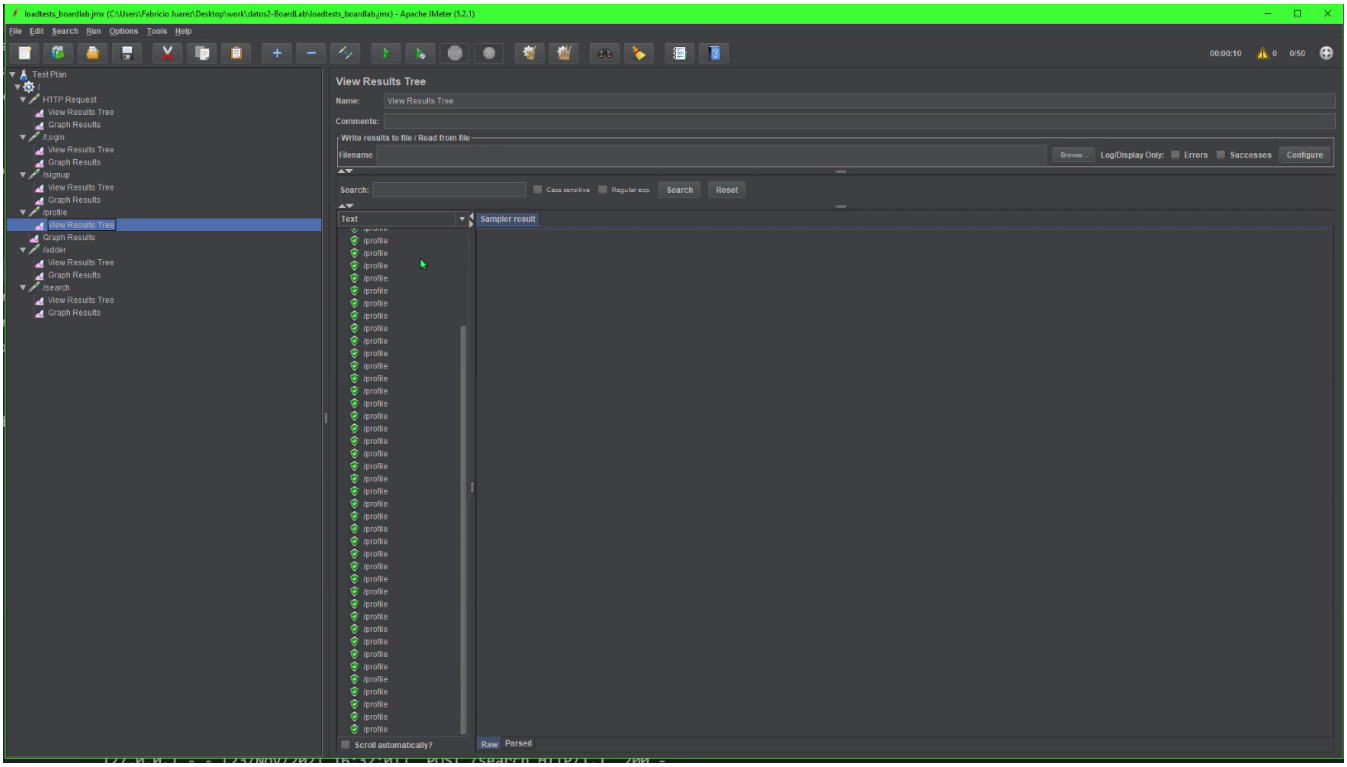
Signup



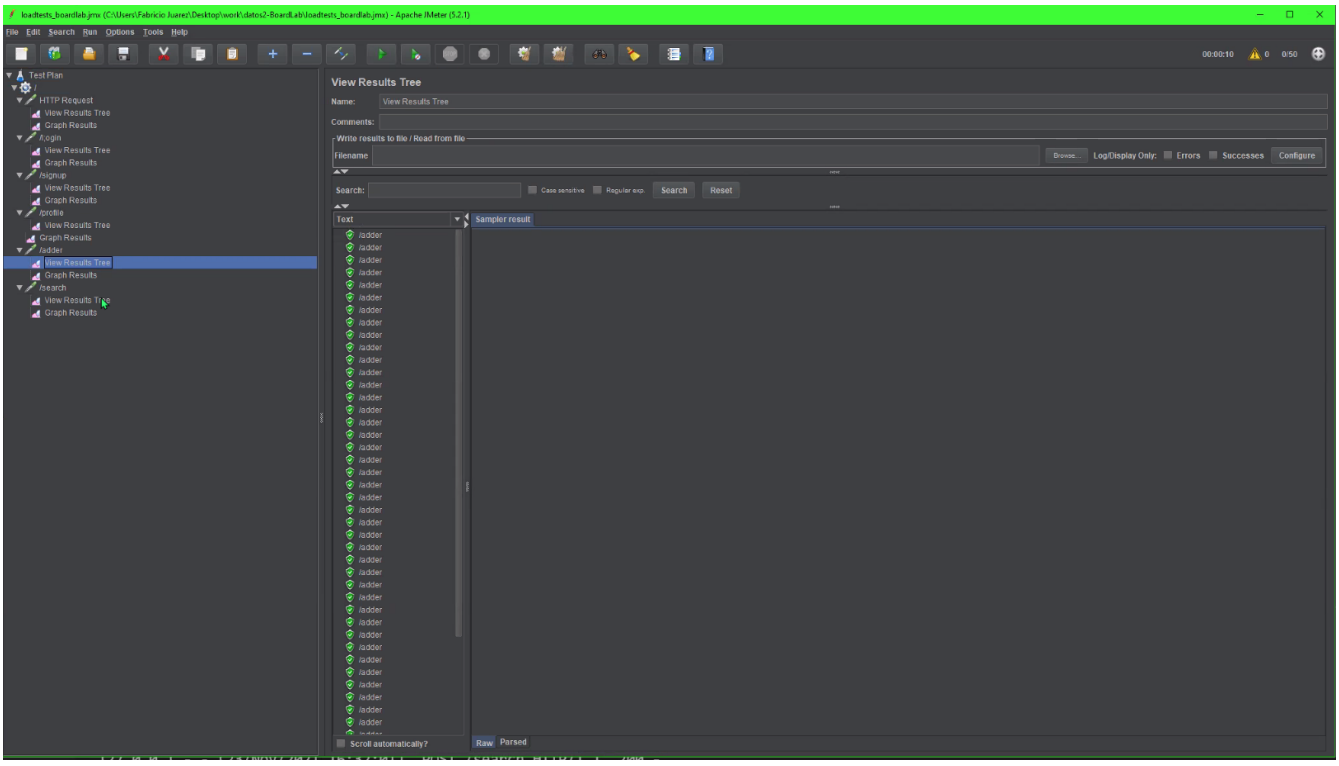
Search



Profile



Adder



Requerimientos - Librerías para el proyecto

- pip install Flask
- pip install peewee
- pip install python-memcached
- pip install flask-profiler
- pip install elasticsearch
- pip install kafka-python
- pip install python-logstash