UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
SISTEMAS OPERATIVOS 1 A
ING. SERGIO MÉNDEZ
AUX. LEONEL AGUILAR
AUX. CARLOS RAMIREZ

PROYECTO 1

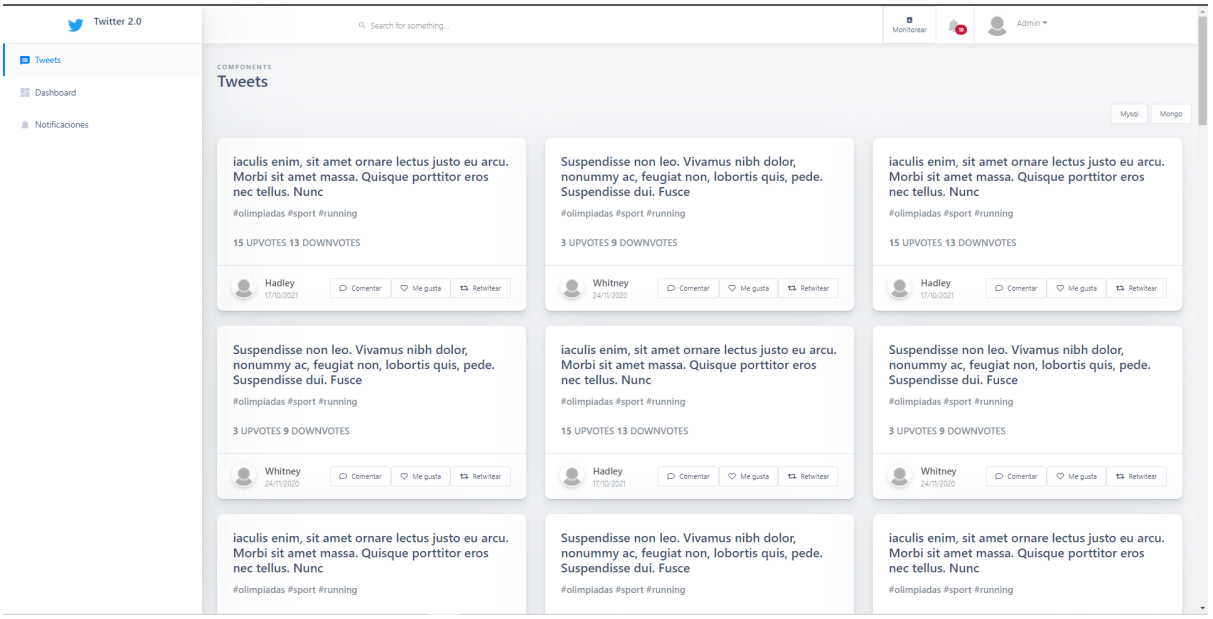
JUAN PABLO ARDON LOPEZ - 201700450
HECTOR AARON JUAREZ TAX - 201404288

LUNES 4 DE OCTUBRE DE 2021

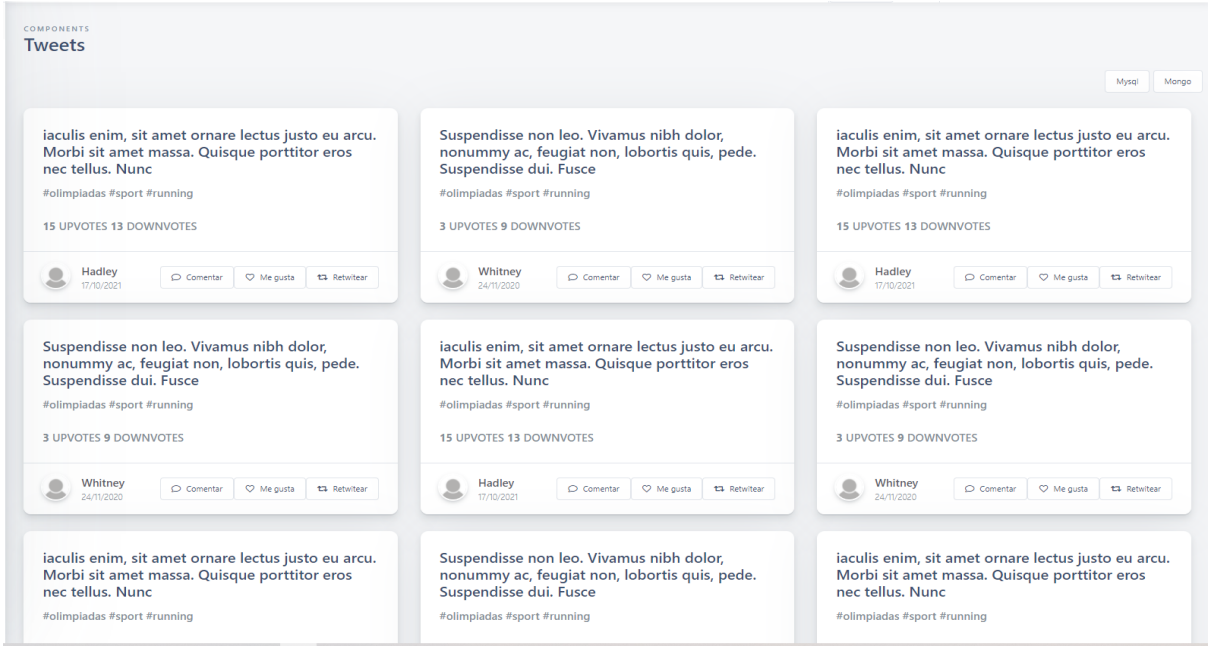
ÍNDICE

APLICACIÓN REACT

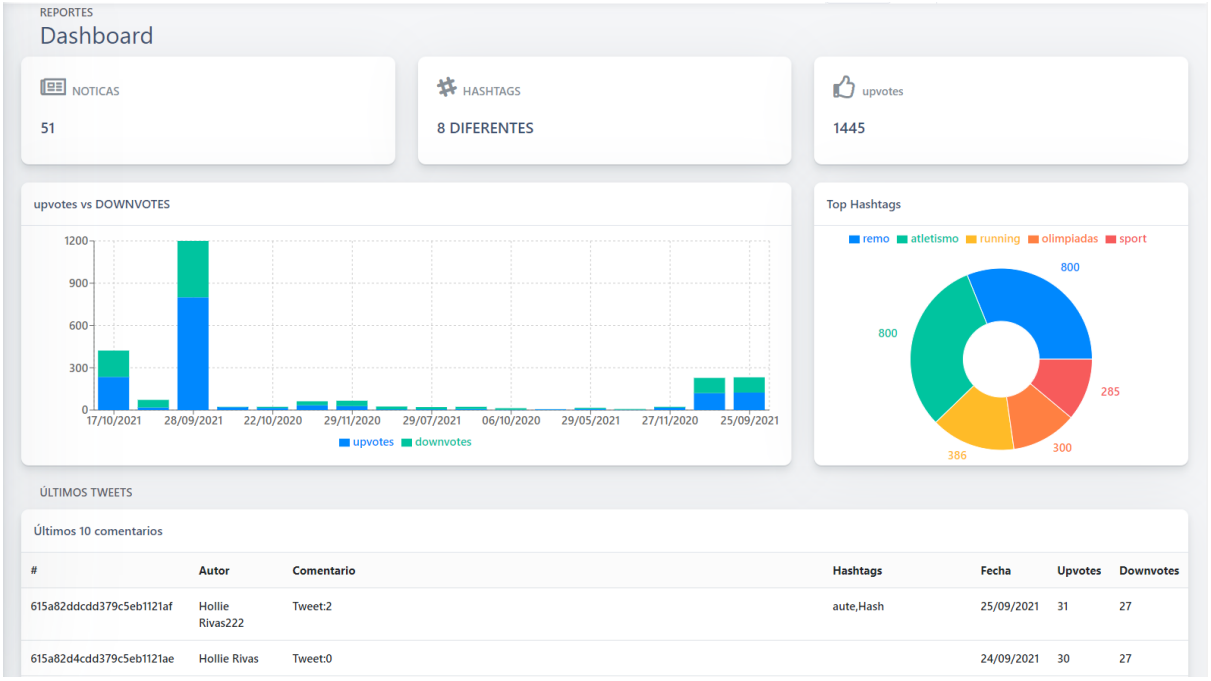
Se desarrolló una aplicación con el framework de react para que se pudiera observar todo el tráfico que se envía desde los load tester de una forma más limpia y ordenada.



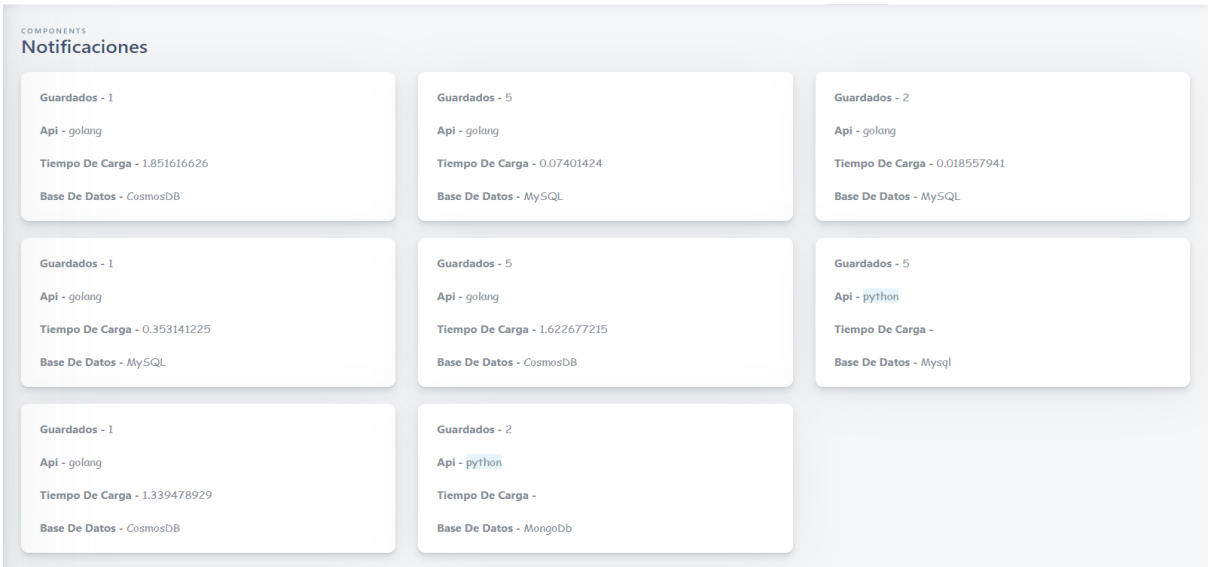
La aplicación cuenta con una sección dónde se pueden ver todos los tweets que han llegado desde el load tester, esta sección cuenta con un botón donde se puede filtrar la base de datos de donde queremos obtener los datos.



La otra sección es la de reportes. En esta parte se puede observar un conjunto de gráficas dónde se pueden observar de mejor manera la información analizada por medio de consultas. Se puede observar un conteo de todas los tweets que hay en la base de datos, la cantidad de hashtags que existen, y el número total de upvotes. En otra gráfica se puede observar la cantidad de upvotes y downvotes que hubieron durante una fecha específica. Y en la última gráfica de la derecha los top hashtags valorados por la cantidad de upvotes que estos tuvieron.



La última sección es la de notificaciones, donde se puede observar en tiempo real cuando llegan nuevos datos a la base de datos. La tarjeta muestra la cantidad de datos correctos e incorrectos que ingresaron a la base de datos, la api de la cuál fueron enviados y la base de datos a la que fueron ingresados.



LOAD TESTER

Los load tester nos permiten enviar un archivo JSON desde el cual existe información de un tweet. El autor, la cantidad de upvotes y downvotes, los hashtags que se utilizaron y el comentario como tal.

Todos los load tester siguen la misma lógica, al iniciar la aplicación se muestra un menú dónde se puede escoger por medio de números qué se quiere realizar.

```
###############-LOAD TESTER-###############
1. Cargar Archivo
2. Enviar Tráfico
3. Leer Archivo
4. Salir
```

Para enviar información primero se tiene que cargar el archivo JSON al programa para que pueda ser utilizado.

```
Nombre Archivo: prueba3.json
Archivo Aceptado
Presione enter para continuar...
```

Luego se escoge la segunda opción para enviar el tráfico

Se nos deplega un nuevo menú dónde elegimos si queremos mandar la data a una api específica o si preferimos que sea aleatorio

```
###############-Seleccionar Api-############
1. Python
2. Golang
3. Rust
4. Aleatorio
5. Regresar
```

Luego se nos muestra un mensaje de la ruta a la cuál fue enviada la data y se nos específica a qué base de datos se está ingresando la data.

```
======================================
Ruta: http://localhost:3000/function/rust/publicar
======================================
```

Cúando la información termina de ser cargada a la base de datos se muestra un mensaje que nos indica la cantidad de tiempo que tardó la transacción y el número total de datos correctos e incorrectos que se mandaron.

```
============Datos Mysql============
BD:
Correctos: 0
Incorrectos: 0
Tiempo: 0
====================================
```

```
=============Datos Mongo=============
BD:
Correctos: 0
Incorrectos: 0
Tiempo: 0
=====================================
```

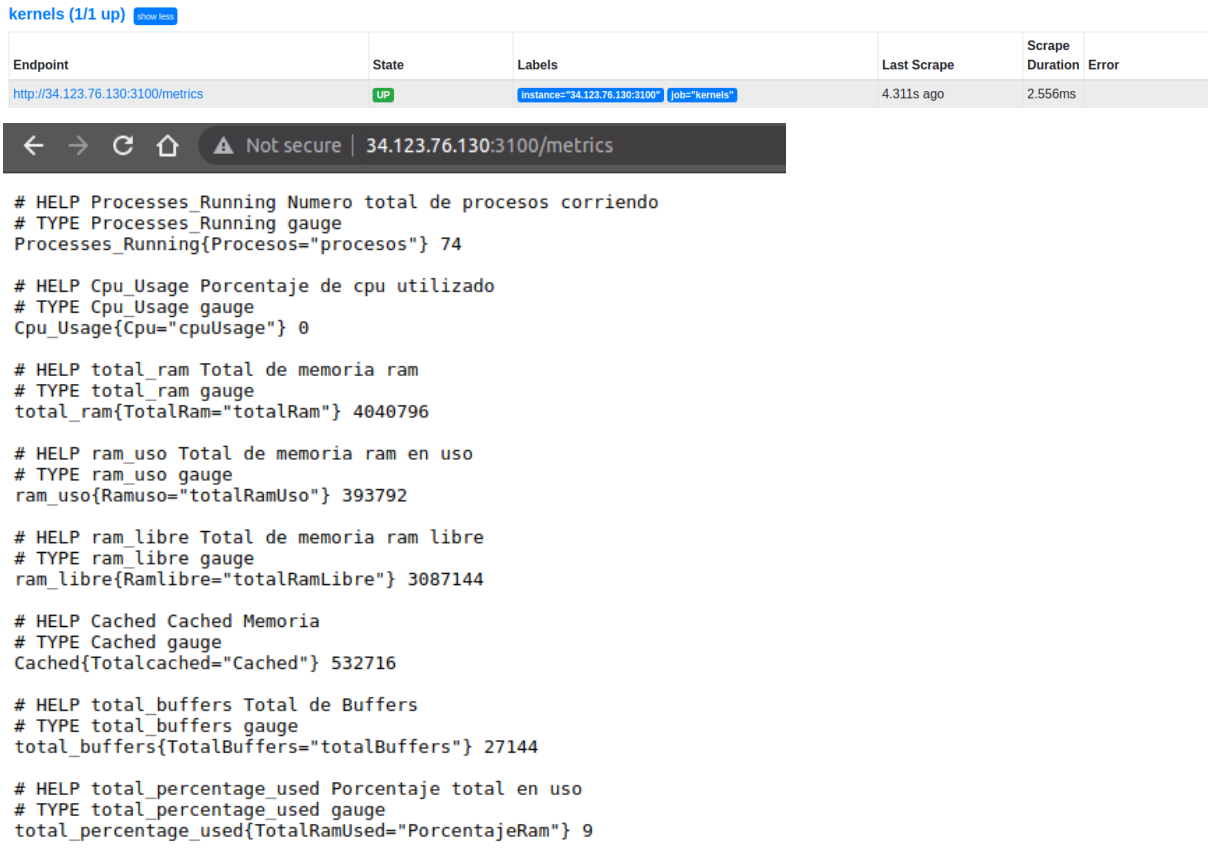
PROMETHEUS Y GRAFANA

PROMETHEUS

Prometheus es un sistema de monitoreo de código abierto que nos permite por medio de apis mostrar información relevante extraída del kernel del pc. Para este proyecto tuvimos que instalar dos módulos en las máquinas que queríamos monitorear, estos módulos nos permiten poder saber información importante tanto de la ram como del cpu.

Por medio de una api pudimos pasar estos datos a nuestro prometheus para que desde la api de prometheus pudiéramos realizar consultas y verificar ésta información de los kernels. Para exponer la data de nuestra api se tiene que habilitar el endpoint /metrics que es desde dónde prometheus extrae la información

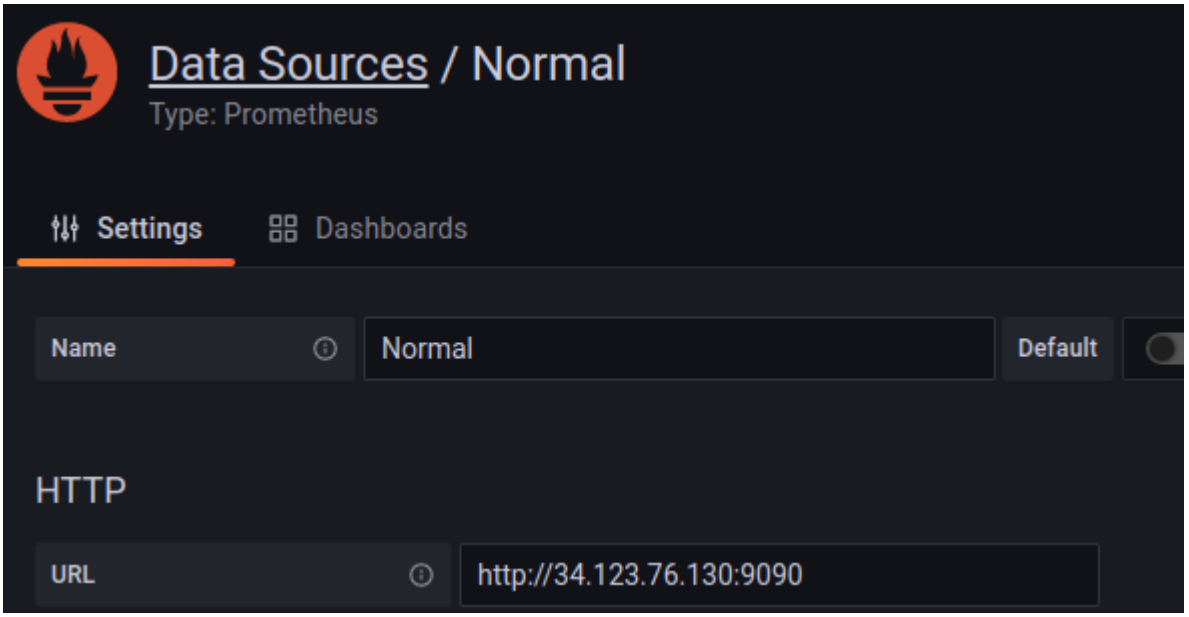
Ejemplo:



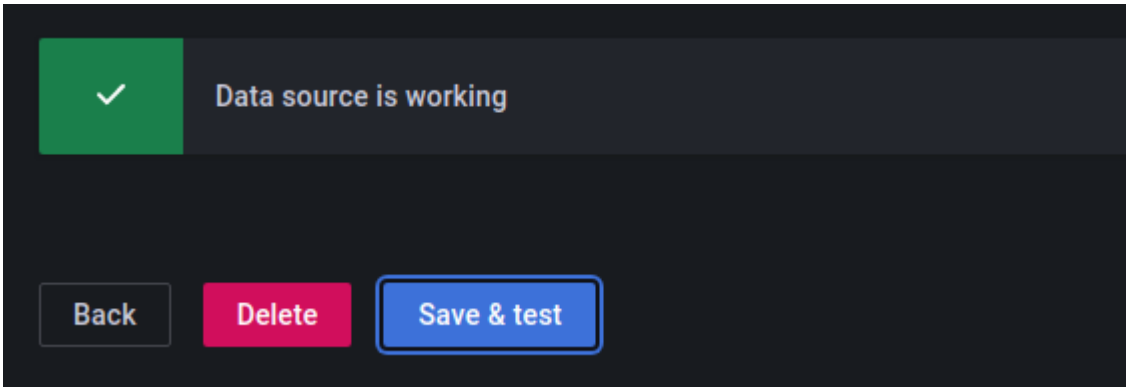
GRAFANA

Grafana es una herramienta de código abierto que nos permite generar gráficas a partir de consultas. Estas consultas pueden ser extraídas desde prometheus. Como se indicó anteriormente, se utilizó prometheus para exponer nuestros datos del módulo kernel, ahora con la ayuda de grafana, utilizaremos gráficas para que la información se pueda entender mejor.

Lo primero que se tiene que hacer es agregar una nueva fuente de datos, para eso nos dirigimos a configuración y agregar fuente de datos. Lo que necesitamos es la ruta dónde están los datos de prometheus y el puerto, que por defecto es 9090. Agregamos ésta información.

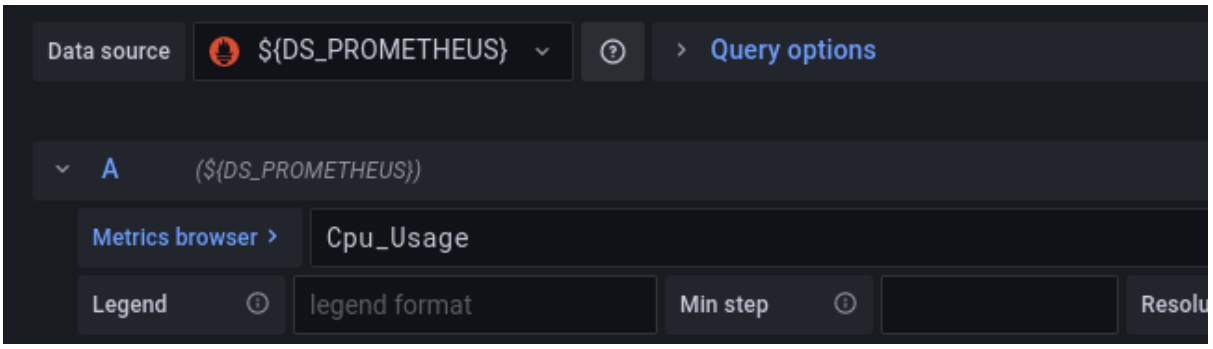


Luego presionamos el botón save & test, nos tiene que salir un mensaje que nos indica que si hay información de prometheus en esa ruta.



Cuando ya tengamos nuestros datos ya podemos crear nuestro dashboard personalizado para que se pueda ver la información.

Se recomienda que en las configuraciones de cada dashboard en la parte de Data source, se elija ${DS_PROMETHEUS} para que la fuente de datos se pueda cambiar desde la vista principal del dashboard y no por cada gráfica



Ya que tenemos todo configurado podemos utilizar dashboards que otras personas ya han hecho o crear nuestro propio dashboard. Este es un ejemplo de los dashboard que se pueden llegar a hacer en Grafana y Prometheus.