Entrega 1

Dominio: https://www.e0carlosgarces.tk

Se creó una aplicación web con estructura back y front end. El back end se realizó con ayuda de ruby on rails, un sistema de subscripción a un broker MQTT, además de un servicio de workers. Todo lo anterior se puede encontrar en la carpeta de backend. Para el servicio de workers, se implementó una api con Koa, la cual recibe peticiones de la aplicación en rails, y envía datos a los workers, para que estos calculen un índice de complejidad de ciertos eventos. Los workers están implementados en gran parte, sin embargo no se lograron usar de la manera óptima en la aplicación debido a que faltó tiempo de desarrollo.

Por otro lado, el frontend se construyó con ReactJs. Se realizó una app con vistas simples y con una ruta que contiene todos los eventos ordenados con un componente de paginación. Al lado de cada evento hay un botón para calcular el índice de complejidad mediante los workers. El sistema de registro e inicio de sesión solo funciona el backend pero en el front no fue implementado, por lo tanto, para ver los eventos se tiene que ir directamente a estos.

La aplicación se encuentra ubicada en distintos contenedores, donde todos estos se coordinan con la ayuda de docker. El backend y el frontend se comunican entre sí, debido a la api que ofrece la aplicación de Ruby on Rails.

Respecto al pipeline CI, este fué implementado con github actions en la carpeta .github.

Para correr la aplicación de forma local, se debe ubicar en la raíz del repositorio y usar el comando ‘sudo docker compose up -d’, ahí en el ‘localhost:3002’ se encontrará la vista de la aplicación, la cuál está conectada con el resto de la aplicación.

Todo lo anterior es lo precente en la branch `master`, sin embargo, en la branch `pipeline_ci` se hizo el bonus, es decir, aparte de lo anterior se creó una sección llamada ***Docker compose*** la cual arma las imagenes del proyecto y corre los containers en el fondo bajo el comando `docker compose up -d`, de esta manera la siguiente sección, llamada ***Run tests*** ejecuta el comando `docker compose exec rails-app rspec spec`, el cual corre todos los tests de la aplicacion dentro del contenedor con nombre ***rails-app***. Se decidió dejar esto en la branch mencionada al principio debido a que se prefirió evitar que estos cambios generaran problemas con el proyecto principal. Se dejaron 0 tests debido a que se tenian que construir los containers denuevo para cada test implementado y tomaban mucho tiempo, pero como se puede ver el codigo funciona.

Por ultimo decir que el código está basado del sigiente: `https://github.com/HoussemDellai/github-actions-course/blob/main/.github/workflows/050-docker-build-workflow.yml`