El BackEnd y el Front End son ejecutados en docker-compose y están montados en una instancia de amazon EC2. Estos se comunican a través de una Amazon API Gateway. Los datos se almacenan en una base de datos MongoDB, la cual está dentro de la nube Atlas de MongoDB.

Para desarrollar el pipe CI, se utilizó Github Actions para crear un Workflow que monta la aplicación con Docker-Compose con el fin de validar el funcionamiento luego de hacer cambios a las ramas principales. En particular, para activar este Workflow basta con hacer *push* o *pull request* de *Master* o *Develop* y la acción de Github se ejecutará sin problemas, indicando el estado final de su ejecución.

Para correr la aplicación en un ambiente local es necesario cambiar las variables de entorno “REACT_APP_URL_BACK” y “MONGO_URI” dentro del archivo docker-compose.yml por “http://localhost:9000” y “mongodb://mongo:27017/db” respectivamente. Luego con docker-compose up --build, la aplicación debería estar corriendo en localhost:80.

RDOC1 - Entrega 2

Se implementaron varias funcionalidades nuevas a la aplicación web, que se detallan a continuación:

Servicio de cálculo de índices:

Se agregó un botón con la funcionalidad de hacer ping a otros usuarios. Este envía una solicitud al usuario al cual se le hace el ping, para que pueda aceptar o rechazar este ping. De aceptarlo, se mandan los datos de cantidad de *Tags* y coordenadas de las ubicaciones de ambos usuarios a una cola RabbitMQ. Esta luego es vaciada por workers, los cuales calculan los índices sidi, siin y dindin; guardando este último dentro del modelo de Ping en la base de datos. Finalmente, din din es mostrado en el perfil de ambos usuarios.

Tags

Ahora existe la opción de asociar etiquetas a las localizaciones agregadas, por el momento existen 10 etiquetas predeterminadas y al crear la ubicación, el usuario selecciona qué *tags* quiere asociar, estos tienen una relación n-n con las ubicaciones, de esta forma, no se crean tags repetidos en la base de datos.

OAuth:

Se usa el servicio de autenticación de google para poder registrarse e iniciar sesión mediante un agente externo.

CronJob

Al hacer ping a un usuario, se calcula la distancia que hay entre los últimos puntos registrados por ambos usuarios, y este número se muestra junto con la información del ping y los índices. Actualmente el dato de la distancia se está programado para actualizarse cada minuto.

RDOC2 - Entrega 2

La documentación de la API se encuentra en el siguiente enlace:

https://documenter.getpostman.com/view/11793429/Uz5CLxjB

E3

Se implemente

El sistema de chats funciona de la siguiente manera: al aceptar un ping recibido, automáticamente el cliente en cuestión creará una sala para conversar, donde simultáneamente invita al emisor del ping a la sala. Luego, ambos usuarios pueden ver las salas en que están invitados desde la vista '/chatList'. Si presionan el nombre de la sala podrán ingresar a ella y conversar mutuamente sin problemas.