Documentación aplicación APP_LOCATION

El objetivo de la aplicación es poder recibir las coordenadas en longitud y latitud de la ubicación de algún dispositivo (inicialmente pensado para un móvil) mediante solicitudes http tipo post y mostrar como se desplaza de manera dinámica en un mapa que es desplegado en una página HTML mediante la librería leaflet de JavaScript.

Frontend

Como se indico anteriormente el frontend es muy simple y se ha desarrollado con HTML, JavaScript y Bootstrap. Se constituye de una única interfaz grafica como se muestra a continuación:

En la interfaz se pueden observar un marcador en el mapa el cual da cuenta de la ubicación del dispositivo, en la parte superior se observa que se indica el valor numérico de la latitud y la longitud, también se indica la dirección (la cual es informada por el dispositivo). Los valores de temperatura, humedad y velocidad del viento se consultan a una api de clima desde el backend y luego se envían por websocket al frontend en JavaScript.

El botón con la etiqueta “Ver Ruta” permite ver la ruta que ha seguido el dispositivo durante su desplazamiento.

La comunicación por websocket implementada permite realizar las siguientes acciones:

- Enviar desde el backend los datos de longitud, latitud y dirección cada vez que son recibidos por este hacia el frontend, esto le da el dinamismo y/o funcionamiento en tiempo real.
- Enviar desde el backend los datos de temperatura, humedad y velocidad del viento hacia el frontend. Estos datos se consultan en el backend cada 5 segundos y se envían al frontend.
- Enviar desde el frontend una alerta cada vez que se actualiza la página para borrar los datos de ubicación almacenados.

Backend

El backend se implementó mediante la librería o framework Flask de Python, y su funcionamiento se divide en funciones que se explicaran a continuación:

Función para mostrar la página de inicio: esta función esta implementada mediante los decoradores de la librería flask y permite desplegar la pagina de inicio e interfaz con el usuario, pagina que se carga desde la carpeta template y se denomina index.html. La ruta para esta función es http://host_a_definir/.

Función para recibir los datos de longitud, latitud y dirección: esta función esta implementada mediante los decoradores de la librería flask y permite recibir los datos, mediante una solicitud post, de la longitud, latitud y dirección desde el dispositivo (móvil). Los datos se almacenan en una estructura de datos tipo Cola que se explicara más adelante. Luego de la recepción de los datos y su almacenamiento, dichos datos son enviados mediante websocket al frontend (JavaScript) para su despliegue dinámico mediante un marcador en el mapa como se explico anteriormente. La ruta para esta función es http://host_a_definir/recibir.

Función para enviar los datos de la ruta seguida: esta función esta implementada mediante los decoradores de la librería flask y rescata todos los puntos de ubicación (longitud y latitud) almacenados en la estructura de datos y los envían al frontend. La solicitud se realiza al dar click en el botón de la interfaz, lo cual genera una solicitud post mediante fetch de JavaScript al backend a esta función. El frontend recibe todos los puntos de ubicación mediante un JSON y los transforma en una ruta que se despliega en el mapa mediante la librería leaflet de JavaScript. La ruta para esta función es http://host_a_definir/ruta.

Función para actuar en caso de que se actualice la página de inicio: esta función esta implementada mediante los decoradores de la librería flask_socketio y se ejecuta cuando se envía un mensaje desde el frontend mediante websocket, en particular esta función se ejecuta cuando llega una alerta desde el frontend indicando que se actualizo la pagina de inicio y al ser ejecutada se borrar todos los datos de la estructura de datos tipo Cola que almacena los puntos de ubicación.

Función para consultar temperatura: esta función no se implementa con flask, en cambio se ejecuta en un threads o hilo, cuya ejecución se realiza en paralelo al hilo principal. Esta función consulta cada 5 segundos a un api la temperatura, humedad, velocidad del viento, entre otras cosas, y luego envía dichos datos al frontend mediante websocket. Esta función se pudiera haber ejecutado en el hilo principal y no utilizar threads, pero la ventaja es que su ejecución puede ser realizada con una periodicidad fija, ya que, si los datos que se puedan recibir desde el móvil llegan con una periodicidad muy baja, 1 segundo, por ejemplo, se tendrían que realizar llamadas al api de clima cada 1 segundo pudiendo recargar o bloquear las peticiones a dicha api. En esta función se utiliza el método filter de la programación funcional para filtrar los datos de respuesta que entrega la api de clima y solo obtener los datos de temperatura, humedad y velocidad del viento.

Estructura de datos tipo cola: esta estructura de datos se implemento mediante la creación de un paquete y un modulo que contiene la clase desde la cual se crea la estructura de datos. Para su creación se implemento una interfaz formal mediante el uso del módulo abc, con dicha interfaz formal se definieron los métodos básicos de la estructura de datos. Seguidamente se implementaron otros métodos. La idea inicial del modulo era crear una estructura tipo cola, con el primer elemento en

entrar es el primer elemento en salir, sin embargo, se implementaron otros métodos que para poder cumplir con algunas funcionalidades particulares de la aplicación, como por ejemplo se limita el número máximo de elementos que puede contener la aplicación, se fija ese valor a 100 elementos.

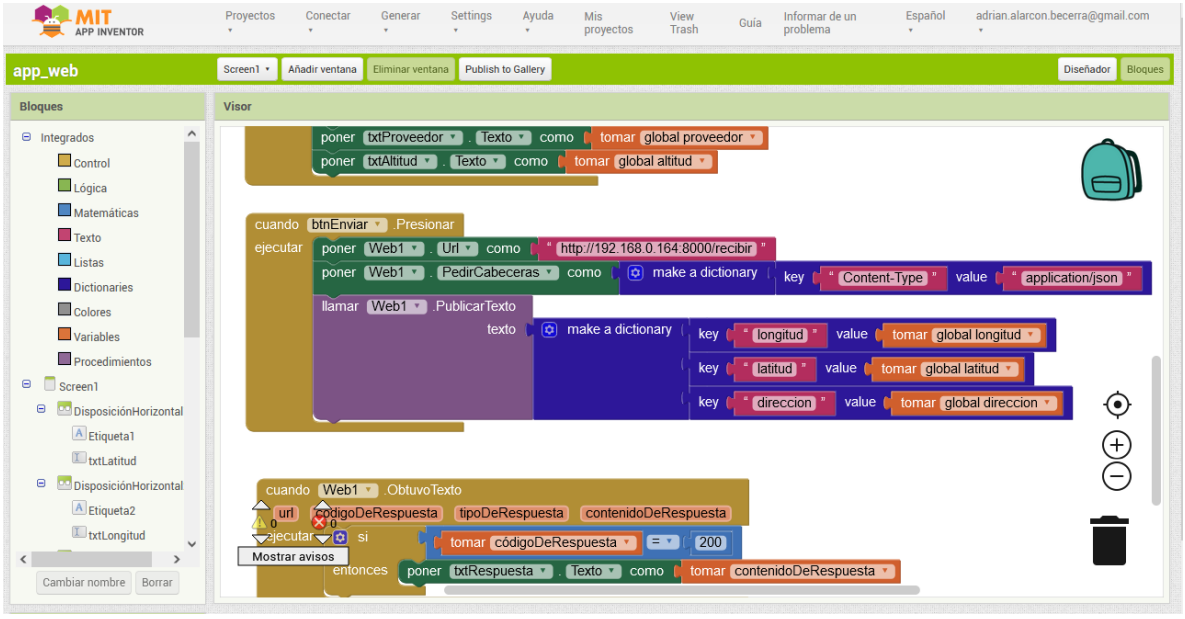
Archivos de pruebas

Mediante la librería pytest se implementaron 3 archivos de pruebas los cuales se describen a continuación:

- test_prueba1: este archivo de prueba permite verificar que la función implementada con decoradores de flask para mostrar la página de inicio index.htlm se despliegue correctamente y muestre efectivamente dicha vista. En particular se verifica que la vista desplegada contenga el tag <h1>Seguimiento Ubicación</h1>.
- test_prueba2: este archivo de prueba permite verificar la función implementada con decoradores de flask para recibir los datos desde el dispositivo móvil mediante una solicitud post.
- test_prueba3: este archivo de prueba permite verificar la función implementada con decoradores de flask para enviar desde el backend al frontend todos los puntos de ubicación almacenados y luego son desplegados en el frontend como una ruta en el mapa.

App en teléfono móvil

Para el envió de los datos desde el teléfono móvil se utilizo la aplicación app inventor para desarrollar una aplicación para Android, dicha aplicación permite programar app para Android mediante un lenguaje de programación en bloques como se muestra en la siguiente figura:



Como se muestra en la figura se implementa el envio de los datos de longitud, latitud y dirección mediante una solicitud post al host en donde se despliegue la aplicación desarrollada en flask descrita anteriormente.

Archivos para realizar envío de ubicaciones para prueba

Para probar el correcto despliegue de los puntos de ubicaciones en el mapa del fontend se desarrollo un script en Python denominado “petición.py” el cual realiza peticiones post enviando datos de ubicación (longitud, latitud y dirección) mediante la librería request. Dichos datos de ubicación se leen desde un archivo ubicaciones.csv que se incluye. El script envía un total de 14 puntos de ubicación cada 3 segundos. Para su uso se debe en primer lugar correr el servidor de prueba de flask en el archivo main.py, luego se debe abrir la dirección host que muestra el servidor de prueba de flask, con ellos se observara la pagina de inicio con el mapa, luego en otro terminal se debe ejecutar el script peticion.py y con ello se mostrara como dinámicamente se desplazan los puntos de ubicación en el mapa, si se presiona el botón se observara la ruta que sigue el dispositivo móvil simulado.