

**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE QUITO**

**CARRERA:  
INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de:  
INGENIEROS DE SISTEMAS**

**TEMA:  
DESARROLLO DE UN INTERFAZ GRÁFICA WEB QUE PERMITA LA  
VISUALIZACIÓN DEL MAPA DE UNA ZONA O ÁREA DETERMINADA Y  
GEORREFERENCIAR ENTIDADES MEDIANTE EL USO DE DISPOSITIVOS  
MÓVILES UTILIZANDO LA HERRAMIENTA GPS.**

**AUTORES:  
EDGAR PAÚL RAMÓN PILLAJO  
ALEJANDRO EFRAÍN ALTAMIRANO CEVALLOS**

**TUTORA:  
LINA PATRICIA ZAPATA MOLINA**

**Quito, marzo del 2018**

## CESIÓN DE DERECHOS DE AUTOR

Nosotros Alejandro Efraín Altamirano Cevallos, con documento de identificación N° 1718475971 y Edgar Paúl Ramón Pillajo con documento de identificación N° 1723348189, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación con el tema: **“DESARROLLO DE UN INTERFAZ GRÁFICA WEB QUE PERMITA LA VISUALIZACIÓN DEL MAPA DE UNA ZONA O ÁREA DETERMINADA Y GEORREFERENCIAR ENTIDADES MEDIANTE EL USO DE DISPOSITIVOS MÓVILES UTILIZANDO LA HERRAMIENTA GPS”**, mismo que ha sido desarrollado para optar por el título de: INGENIEROS DE SISTEMAS, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



.....  
ALEJANDRO EFRAÍN  
ALTAMIRANO CEVALLOS  
CI: 1718475971



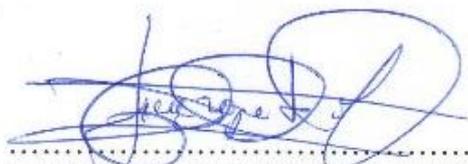
.....  
EDGAR PAÚL  
RAMÓN PILLAJO  
CI: 1723348189

Quito, marzo del 2018

## DECLARATORIA DE COAUTORÍA DE LA TUTORA

Yo declaro que bajo mi dirección y asesoría fue desarrollado el artículo académico, con el tema **“DESARROLLO DE UN INTERFAZ GRÁFICA WEB QUE PERMITA LA VISUALIZACIÓN DEL MAPA DE UNA ZONA O ÁREA DETERMINADA Y GEORREFERENCIAR ENTIDADES MEDIANTE EL USO DE DISPOSITIVOS MÓVILES UTILIZANDO LA HERRAMIENTA GPS”**, realizado por Alejandro Efraín Altamirano Cevallos y Edgar Paúl Ramón Pillajo, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, marzo del 2018



.....  
LINA PATRICIA ZAPATA MOLINA  
CI: 0501877278

## **Dedicatoria**

Dedico este trabajo en primer lugar a mi familia por haberme apoyado en todos estos años de estudios en la universidad y por tener la paciencia y soportar malos ratos que he tenido en mi vida

A mis padres por haber sido un pilar fundamental en toda mi vida de estudiante.

A mi hermana en especial ya que estando fuera del país, me ayudado en todo momento, me ha apoyado en los momentos más difíciles que he tenido en la vida, he seguido sus consejos y he salido adelante, también me ayudado en la parte económica para poder terminar mis estudios en la universidad

A mi hermano por sus consejos, motivación y ayuda en todo momento siempre ha estado presente en lo que he necesitado.

Alejandro Efraín Altamirano Cevallos.

Dedico este trabajo en primer lugar a Dios por darme el don de la vida, por darme fuerzas en los momentos más difíciles, por ser mi guía y darme el don de la sabiduría que me ha permitido concluir esta etapa de mi vida.

A mi queridos Mamá Miche y Papá Antonio quienes a pesar de no estar a mi lado físicamente me enseñaron que hay que luchar hasta el último momento y que nunca hay que perder la fe.

A mis padres Edgar y Mercedes por ser el pilar fundamental en mi vida, por enseñarme excelentes valores morales, por ser mis amigos y sobre todo por haberme permitido que pueda empezar este sueño que gracias a su esfuerzo, trabajo duro y lucha constante logre conseguir.

A mi querida hija Neith por ser mi fuente de motivación e inspiración, por sus sonrisas y amor incondicional, por esas palabras que en momentos fueron muy duras pero que me ayudaron a tomar conciencia y me permitieron levantarme, seguir adelante y lograr concluir con el desarrollo de este trabajo.

A mi amada esposa Zayra Cordones, mi compañera de vida por ser mi apoyo incondicional, por su sacrificio y esfuerzo, por todos los momentos difíciles que hemos pasado juntos y a pesar de todo eso nunca dejo de creer en mí, por su comprensión y motivación para que cada día sea un mejor persona, por toda su paciencia, apoyo y amor incondicional este trabajo es dedicado especialmente para ti.

Edgar Paúl Ramón Pillajo.

## **Agradecimiento**

Agradecemos de manera especial a la Ing. Lina Patricia Zapata, Msc., tutora de nuestro trabajo, por habernos motivado y orientarnos con sus conocimientos para poder realizar nuestro trabajo.

A la Universidad Politécnica Salesiana que nos abrió sus puertas para empezar y culminar con nuestros estudios, a los docentes que tuvimos la oportunidad de conocer, los cuales formaron parte de nuestra vida universitaria y que supieron compartir sus conocimientos para formar excelentes personas y profesionales.

Alejandro Efraín Altamirano Cevallos.  
Edgar Paúl Ramón Pillajo.

# DESARROLLO DE UN INTERFAZ GRÁFICA WEB QUE PERMITA LA VISUALIZACIÓN DEL MAPA DE UNA ZONA O ÁREA DETERMINADA Y GEORREFERENCIAR ENTIDADES MEDIANTE EL USO DE DISPOSITIVOS MÓVILES UTILIZANDO LA HERRAMIENTA GPS.

Paul Ramón<sup>1</sup>, Alejandro Altamirano<sup>2</sup>, Lina Zapata<sup>3</sup>

## Resumen

El presente artículo presenta el desarrollo y construcción de un interfaz gráfica Web que permite visualizar la ubicación de una entidad (persona, vehículo, etc.) en un mapa. Para lo cual se desarrolló una aplicación móvil la cual permite obtener las coordenadas geográficas mediante la utilización del GPS, las cuales son almacenadas y visualizadas en tiempo real en el mapa que se encuentra integrado en la interfaz web.

Adicional la interfaz web permitirá visualizar las trayectorias de una entidad en una fecha determinada.

**Palabras Clave:** Aplicación Web, Aplicación Móvil, base de datos, entidad, GPS, ubicación.

## Abstract

This article presents the development and construction of a Web graphical interface that allows to see the location of an entity (person, vehicle, etc.) on a map. For which a mobile application was developed which allows to obtain the geographical coordinates using the GPS, which are stored and visualized in real time on the map which is integrated into the web interface.

Additionally, the web interface will allow visualize the trajectories of an entity on a specific date.

**Keywords:** Web Application, Mobile Application, Database, entity, GPS, location.

---

<sup>1</sup> Docente de la carrera de Ingeniería de Sistemas, Universidad Politécnica Salesiana, sede Quito.

<sup>2</sup> Estudiante de la carrera de Ingeniería de sistemas, Universidad Politécnica Salesiana, sede Quito  
Autor por correspondencia: pramon@est.ups.edu.ec

<sup>3</sup> Estudiante de la carrera de Ingeniería de sistemas, Universidad Politécnica Salesiana, sede Quito  
Autor por correspondencia: aaltamirano@est.ups.edu.ec

## 1. Introducción

En los últimos años ha crecido la demanda de aplicaciones informáticas que permiten realizar monitoreo y seguimiento, a través del uso del GPS, de entidades como personas, autos, animales por motivos de seguridad.

Ante esta creciente demanda de las aplicaciones de georreferenciación, el desarrollo de un visualizador propio busca ampliar los conocimientos en esta área y disponer de una mayor cantidad de aplicaciones propias las cuales puedan ser configurables de acuerdo a las diferentes necesidades que requieran de un sistema de georreferenciación.

Esto es posible gracias al auge de los dispositivos móviles con característica cada día más avanzadas, una de esas es la integración del GPS, lo que ha permitido que cualquier ciudadano acceda a este tipo de sistemas ayudando con esto a ampliar el desarrollo de aplicaciones de georreferenciación.

Es por eso que el principal objetivo de este trabajo es desarrollar una interfaz gráfica web propia que permita visualizar la ubicación geográfica de una entidad que puede representar a una persona, vehículo, etc., sobre un mapa mediante la utilización del GPS de un dispositivo móvil.

Para este trabajo se consideró el desarrollo de dos aplicaciones. La primera es la aplicación web que permitirá la visualización de la trayectoria de los usuarios y la segunda es la aplicación móvil que permitirá utilizar el GPS del dispositivo y obtener las coordenadas de la ubicación del usuario.

Este trabajo está estructurado de la siguiente manera: La Sección 2 presenta los lenguajes y herramientas de programación. El análisis y diseño se encuentran en la Sección 3. La sección 4 presenta la implementación, las pruebas y resultados se muestran en la sección 5 y finalmente las conclusiones son presentadas en la Sección 6.

## 2. Lenguajes y Herramientas de Programación

### 2.1. Java

Es un lenguaje de programación orientado a objetos que fue diseñado específicamente para poseer pocas dependencias de implementación. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo.

Las aplicaciones de Java son generalmente compiladas a bytecode (clase Java) que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora subyacente («¿Qué es el lenguaje de programación JAVA? - Base de Conocimientos - ICTEA», 2016).

#### 2.1.1. Spring

Es un framework que proporciona un modelo de programación y configuración integral para aplicaciones empresariales modernas basadas en Java, en cualquier tipo de plataforma de implementación.

Un elemento clave de Spring es el soporte de infraestructura a nivel de aplicación: Spring se enfoca en la "fontanería" de aplicaciones empresariales para que los equipos puedan enfocarse en la lógica empresarial a nivel de aplicación, sin lazos innecesarios con entornos de implementación específicos («Spring Framework», 2017).

### 2.2. JavaScript

JavaScript es un lenguaje ligero e interpretado, orientado a objetos, conocido como el lenguaje de script para páginas web, pero también usado en muchos entornos sin navegador. Es un lenguaje script multi-paradigma, basado en prototipos, dinámico, soporta estilos de

programación funcional, orientada a objetos e imperativa.

JavaScript agrega mayor interactividad a la web, además se puede hacer uso de librerías y framework escritas sobre este lenguaje, que ayudan a crear una mejor experiencia de usuario en los sitios web. De igual manera JavaScript se puede utilizar en los servidores web («JavaScript | MDN», 2017).

### 2.2.1. *ReactJS*

React es una librería escrita en JavaScript, desarrollada para facilitar la creación de componentes interactivos, reutilizables, para interfaces de usuario.

Hace uso del paradigma denominado programación orientada a componentes. Dichos componentes se representan como clases que heredan de la clase "**Component**" cuyo único requerimiento especial es especificar el método render que define cuál será el contenido del mismo.

Una de las principales características de React es su velocidad de generación de vistas. Eso es posible ya que proporciona un Virtual DOM que genera React con cada componente creado y el algoritmo de Diff que lo que hace es marcar que elementos dentro del DOM Virtual tienen cambios para renderizar solo ellos y no tener que revisar y repintar el DOM entero de la página («¿Cómo funciona React.js?», 2017).

### 2.2.2. *React Native*

Es un framework que permite desarrollar apps nativas IOS y Android usando Javascript. Esto debido a que realiza una traducción del código React Native en Objective-C, para el caso de IOS y Java en Android.

Permite a los programadores desarrollar apps nativas sin tener que conocer los lenguajes de programación específicos para cada uno de los sistemas operativos de los dispositivos móviles.

La sintaxis es bastante clara y sencilla, utiliza el mismo diseño que React, permitiendo componer una interfaz de usuario móvil completa a partir de componentes declarativos, aportando flexibilidad y reaprovechamiento en el código («React Native · A framework for building native apps using React», 2017).

## 2.3. **PostgreSQL**

Es un potente sistema de base de datos objeto-relacional de código abierto. Cuyo desarrollo activo y una arquitectura que le ha permitido obtener sólida reputación de fiabilidad e integridad de datos.

PostgreSQL incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional, adicionalmente posee una gran escalabilidad. Lo que permite ajustar el número de CPUs y la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta («PostgreSQL: About», 1996).

## 2.4. **Firestore**

Es un servicio de que nos proporciona un backend en la nube con una fuente de datos en tiempo real listo para el desarrollo de aplicaciones web y apps para dispositivos móviles.

### 2.4.1. *Firestore Authentication*

Proporciona servicios de backend, SDK fáciles de usar y bibliotecas de IU ya elaboradas para autenticar a los usuarios en una aplicación. Admite la autenticación mediante contraseñas, números de teléfono, proveedores de identidad federados populares, como Google, Facebook y Twitter.

Además se integra estrechamente con otros servicios de Firebase y aprovecha los estándares de la industria como OAuth 2.0 y OpenID Connect, por lo que se puede integrar fácilmente con un backend personalizado («Firebase Authentication», 2017).

#### 2.4.2. *Firestore RealTime Database*

Firestore Realtime Database es una base de datos alojada en la nube. Los datos se almacenan en formato JSON y se sincronizan en tiempo real con cada usuario conectado.

Cuando se compilan aplicaciones multiplataforma, todos los usuarios comparten una instancia de Realtime Database y reciben actualizaciones automáticamente con los datos más recientes («Firestore Realtime Database | Firestore Realtime Database», 2017).

### 3. Análisis y Diseño

#### 3.1. Casos de Uso

La figura 1 muestra los diferentes casos de uso con los que va a interactuar el Usuario en la Aplicación Móvil. Los cuales se describen a continuación:

**Registrar App Móvil:** Permite registrar la cuenta de un usuario nuevo en el sistema.

**Ingresar App Móvil:** Permite que el usuario ingrese a la aplicación.

**Cerrar Sesión App Móvil:** Permite que el usuario salga de la aplicación.

**Validar Datos:** Valida que la información ingresada por el usuario sea la proporcionada al momento de registrar su cuenta.

**Marcar Posición y Trayecto:** Marca la posición actual del usuario en el mapa.

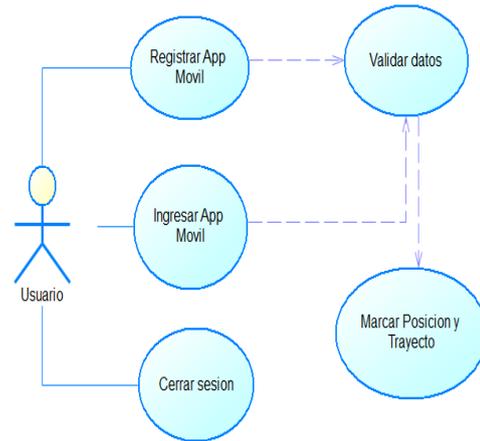


Figura 1. Caso de Uso Aplicación Móvil.

La figura 2 muestra los diferentes casos de uso con los que va a interactuar el Usuario en la Aplicación Web. Los cuales se describen a continuación:

**Visualizar Trayectoria:** Visualiza la trayectoria de un usuario en el mapa.

**Histórico de Trayectoria:** Permite visualizar la trayectoria de un usuario en una fecha determinada.

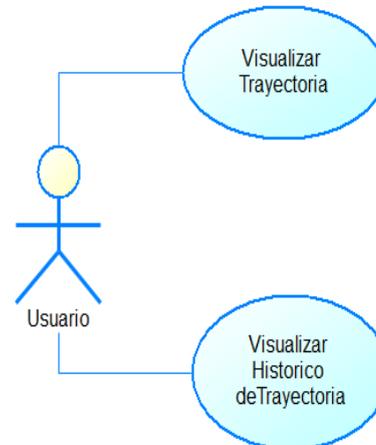


Figura 2. Caso de Uso Aplicación Web.

#### 3.2. Modelo Secuencial

La forma en que interactúa el usuario con la aplicación móvil se describe en el diagrama de secuencia de la figura 3.

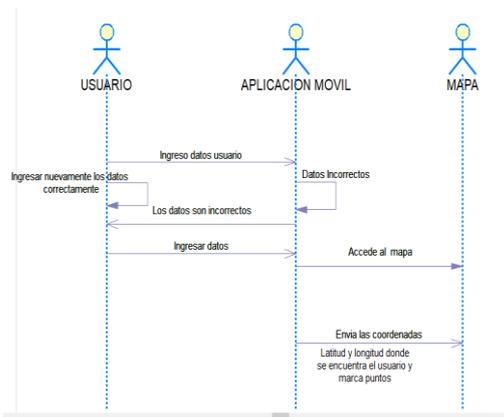


Figura 3. Modelo Secuencial Aplicación Móvil.

La forma en que interactúa el usuario con la aplicación web se describe en el diagrama de secuencia de la figura 4.

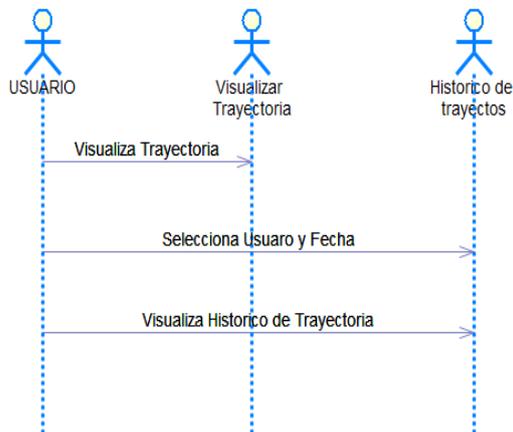


Figura 4. Modelo Secuencial Aplicación Web.

### 3.3. Modelo Entidad Relación

El Modelo Entidad Relación de la base de datos desarrollada para el funcionamiento de la aplicación, se describen en la figura 5 y en la tabla 1 la descripción de sus tablas.

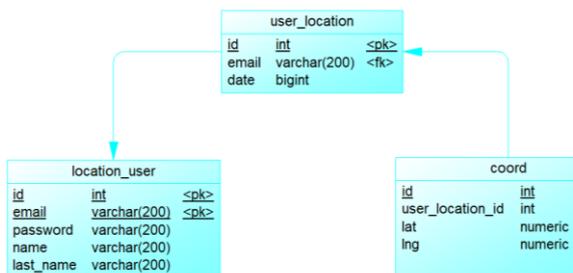


Figura 5. Diagrama Entidad Relación base de datos.

Tabla	Descripción
<b>user_location</b>	Almacena los datos del usuario registrados en la aplicación móvil.
<b>location_user</b>	Almacena los datos del usuario cada vez que un usuario ingresa a la aplicación móvil.
<b>coord</b>	Almacena las coordenadas de la ubicación actual de los usuarios conectados a la aplicación móvil.

Tabla 1. Tablas de la base de datos.

### 3.4. Aplicación Móvil

La aplicación móvil cuenta con dos vistas las cuales puede hacer uso el usuario. La primera vista permite realizar el registro del usuario al sistema y la segunda permite el ingreso del usuario a la aplicación.

La vista de registro permite crear un nuevo usuario en el sistema, lo que le permitirá al usuario obtener acceso al sistema de monitoreo. La vista de ingreso solicita al usuario las credenciales para acceder al sistema de monitoreo.

Una vez el usuario ingresó al sistema la aplicación comenzará a obtener las coordenadas geográficas de la posición del usuario a través del uso del GPS, las cuales serán mostradas en el mapa integrado en la aplicación así como también serán enviadas a la base de datos para que sean visualizadas en la interfaz web. Esta vista está compuesta por los campos que se especifican en la figura 6.

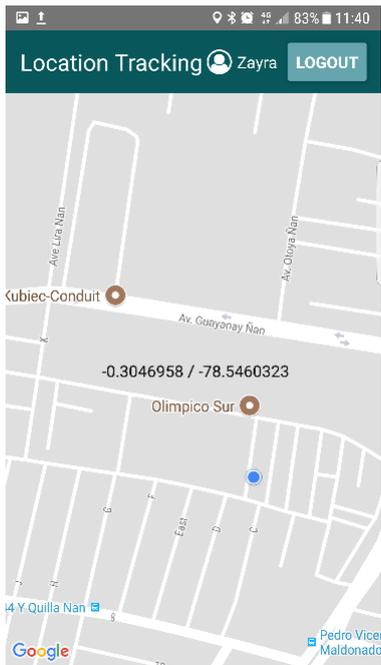


Figura 6. Interfaz Aplicación móvil.

### 3.5. Aplicación Web

La aplicación web está desarrollada para visualizar la trayectoria de los usuarios conectados al sistema través de la aplicación móvil.

Permite obtener los usuarios que se encuentran conectados mediante la aplicación móvil al sistema, si existen usuarios conectados por medio de la aplicación móvil, se puede visualizar en el mapa integrado la ruta o trayectoria que realice el usuario, como lo muestra la figura 7.

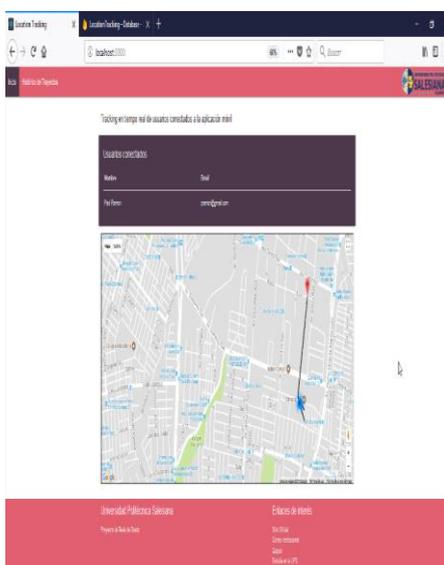


Figura 7. Interfaz Usuarios Conectados

La aplicación permite visualizar el histórico de las trayectorias que realizó un usuario en una fecha determinada y posterior a esto se mostrarán las diferentes trayectorias que realizó un usuario y las cuales se podrán visualizar en el mapa que se encuentra integrado en la interfaz web, como lo muestra la figura 8.

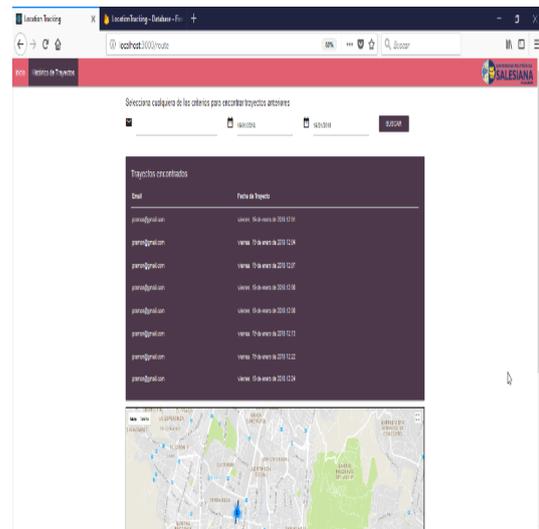


Figura 8. Interfaz Histórico de Trayectoria

## 4. Implementación

En el desarrollo de la aplicación web el Front-End se realizó en Javascript empleando la librería ReactJs, utiliza los servicios Realtime Database y Authentication de Firebase. La parte del Back-End se realizó en Java empleando el framework Spring Boot.

Como ya se mencionó anteriormente, para utilizar los servicios Realtime Database y Authentication de Firebase, es necesario instanciar una cuenta de Google y acceder a los servicios de Firebase, luego se debe acceder a la consola de administración de Firebase y proceder a crear un nuevo proyecto para poder asignar al nombre del proyecto en desarrollo y obtener la instancia que es generada por Firebase y que se debe incluir en la estructura del proyecto.

```

15 firebase.js x
1  import * as firebase from 'firebase';
2
3  export const firebaseApp = firebase.initializeApp({
4    apiKey: "AIzaSyBtb7nVPGj_NZk_wbHdOxgXN9X2iSggm-I",
5    authDomain: "locationtracking-59d6f.firebaseio.com",
6    databaseURL: "https://locationtracking-59d6f.firebaseio.com",
7    projectId: "locationtracking-59d6f",
8    storageBucket: "",
9    messagingSenderId: "112579139288"
10 });
11
12 export const db = firebaseApp.database();
13 export const auth = firebaseApp.auth();
14
15 export default firebase;

```

Figura 9. Código Fuente Instancia Servicios Firebase

La figura 9 muestra la instancia general del proyecto firebase creado así como la instancia del servicio Realtime Database y la instancia del servicio Authentication, los cuales se incluyen en el script `firebase.js` es el directorio *config* de nuestro proyecto.

La estructura de la aplicación web está basada en la estructura de un proyecto ReactJs y formada por el directorio *src* donde se incluyen los componentes, *Home.js*, *MapaLocation.js*, *HistoricalRoutes.js*, además del componente principal *App.js* y el punto de entrada *index.js*, en este se inicializa el componente principal *App.js*, a través del método *ReactDOM.Render*.

Este método recibe como primer parámetro el componente a renderizar y como segundo el elemento del DOM donde el componente va a ser renderizado, como se muestra en la figura 10.

```

15 index.js x
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3  import './index.css';
4  import App from './App';
5  import registerServiceWorker from './registerServiceWorker';
6
7  ReactDOM.render(<App />, document.getElementById('root'));
8  registerServiceWorker();
9

```

Figura 10. Código Fuente de iniciación de componentes

Los diferentes componentes de nuestra aplicación utilizan los servicios de

Firestore los cuales se configuran mediante la función de la figura 11, y permiten realizar la sincronización en tiempo real con la base de datos de Firestore.

```

db
  .ref("/users")
  .on("value", snapshot => {
    snapshot.forEach(user => {

```

Figura 11. Código Fuente de Sincronización con Firestore

El desarrollo de la aplicación móvil se realizó con el framework React Native, el cual permite desarrollar aplicaciones móviles nativas utilizando únicamente JavaScript, con el mismo diseño que ReactJS.

Es por eso que nuestra aplicación móvil está formada por el directorio *src* donde se incluyen los componentes, *Auth.js*, *LocationTracking.js*, *Login.js*, *MyLocationMapMarker.js*, *SignUp.js*, además del componente principal *App.js* y el punto de entrada *index.js*, en este se inicializa el componente principal *App.js*, a través del método *registerComponent.js*, como lo muestra la figura 12.

```

15 index.js x Bienvenido
1  import { AppRegistry } from 'react-native';
2  import App from './App';
3
4  AppRegistry.registerComponent('LocationTracking', () => App);
5

```

Figura 12. Código Fuente de iniciación de componentes.

Nuestro componente es *MyLocationMapMarker.js* cual se encarga de obtener las coordenadas de la ubicación del dispositivo móvil y las almacena en la base de datos.

```

export default class MyLocationMapMarker extends React.Component {
  constructor(props) {
    super(props);
    this.mounted = false;
    this.state = {
      myPosition: null
    };
  }
  componentDidMount() {
    this.mounted = true;
    navigator.geolocation.getCurrentPosition((position) => {
      const myPosition = { latitude: parseFloat(position.coords.latitude), longitude: parseFloat(position.coords.longitude) };
      console.warn("position", myPosition);
      this.setState({ myPosition });
    }, null);
    if (this.props.coordinate) return;
    this.watchLocation();
  }
  watchLocation() {
    this.watchID = navigator.geolocation.watchPosition(
      (position) => {
        const myLastPosition = this.state.myPosition;
        const myPosition = { latitude: parseFloat(position.coords.latitude), longitude: parseFloat(position.coords.longitude) };
        console.warn("position watch", myPosition);
        if (!isEqual(myPosition, myLastPosition)) {
          this.setState({ myPosition });
        }
      },
      null
    );
  }
}

```

Figura 13. Código Fuente MyLocationMapMarker.js

## 5. Pruebas y Resultados

Para la realización de las pruebas las aplicaciones se instalaron en un equipo con las siguientes características:

Marca: Dell  
 Modelo: Optiflex 960  
 Procesador: Intel Core 2 Duo E8400 3.00 Ghz  
 RAM: 4 GB  
 SO: Windows 10 Enterprise  
 Tipo de Sistema: 64-bit

### 5.1. Pruebas de Rendimiento

Las pruebas ejecutadas sobre la aplicación móvil corresponden a las de rendimiento. Para la ejecución de las pruebas se utilizaron dispositivos móviles Android de distintas marcas, en los cuales se midió el tiempo de respuesta en segundos de las distintas funcionalidades de la aplicación móvil utilizando la conexión WIFI y de Datos Móviles.

### Conexión Wifi

	Sony Xperia E5	Samsung J5 Pro	Samsung S6 Edge
Registrar de Usuario	11	8	5
Ingresar de Usuario	5	6	2
Cerrar Sesión	1.5	1.1	0.5

Tabla 2. Resultado Pruebas de Rendimiento

### Conexión Datos Móviles

	Sony Xperia E5	Samsung J5 Pro	Samsung S6 Edge
Registrar de Usuario	15	11	8.4
Ingresar de Usuario	10	6	4.2
Cerrar Sesión	3.5	2.2	1.2

Tabla 3. Resultado Pruebas de Rendimiento

Con las pruebas realizadas se puede apreciar que el tiempo de respuesta para cada una de las distintas funcionalidades de la aplicación móvil es mínimo y depende del tipo y de la velocidad de la conexión a Internet, lo que permite que el usuario acceda de manera ágil y rápida a las funcionalidades de la aplicación.

### 5.2. Pruebas de Consumo de Recursos

Se realizan pruebas para ver el consumo de recursos de la aplicación en el dispositivo móvil para esto se utilizó la aplicación App Tune-up Kit, la cual registro un consumo de CPU del 4,1% en el dispositivo, como se muestra en la figura 14. Este consumo de recursos es mínimo y no causa ningún tipo de afectación en el rendimiento y funcionalidad del dispositivo móvil.



Figura 14. Consumo de CPU

Se realizó otra prueba para verificar el consumo de Memoria RAM y CPU de la aplicación móvil, se utilizó la aplicación Simple System Monitor, la cual obtuvo un consumo del 0,42% en CPU y un consumo de 42,16 Mb de Memoria RAM. Estos datos de consumo son mínimos y no causan ningún tipo de afectación en el funcionamiento normal del dispositivo móvil, además esto permite que nuestra aplicación se pueda implementar en cualquier gama de dispositivos móviles.

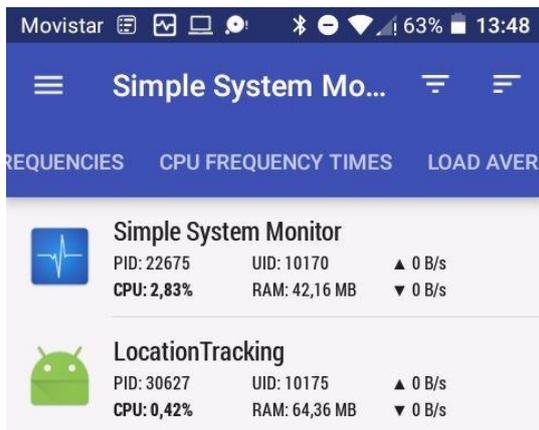


Figura 15. Consumo de CPU y Memoria Ram

### 5.3. Pruebas de Trazado de Trayectoria

En la figura 16 se puede apreciar la ruta que registró un usuario en un vehículo a través de las coordenadas obtenidas por la aplicación móvil desarrollada. Esta ruta tiene un margen de error del 10 % en comparación con la ruta que fue trazada en Google Maps.

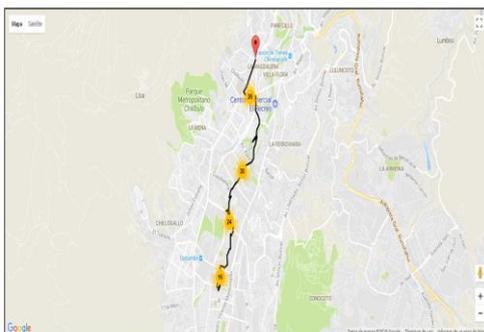


Figura 16. Prueba de Trayectoria 1

En la figura 17 se puede apreciar la ruta que registró un usuario a pie en las inmediaciones de la Universidad Politécnica Salesiana a través de las coordenadas obtenidas por la aplicación móvil desarrollada, se puede observar que la ruta es más corta y por tanto se disminuye el margen de error en comparación con la ruta que fue trazada en Google Maps.

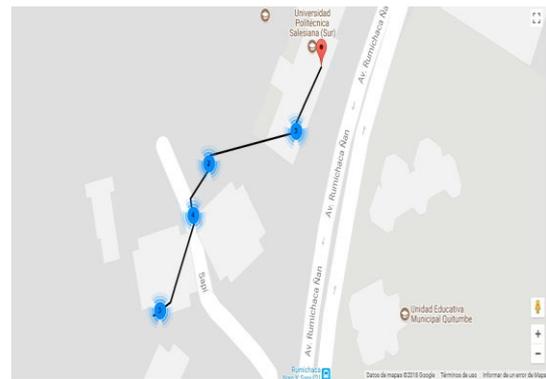


Figura 17. Prueba de Trayectoria 2

Considerando que la aplicación está expuesta a factores tales como el ancho de banda, intensidad de la señal y la calibración del GPS, se puede observar que el trazado de la trayectoria es aceptable y cumple con el objetivo de georreferenciar la ubicación de una entidad.

### 5.4. Pruebas de Consumo de Ancho de Banda

Se realizaron pruebas comparativas de consumo de ancho de banda entre la aplicación móvil desarrollada y la aplicación Google Maps.

Para la obtención de datos se realizaron 20 pruebas para una trayectoria de 10 metros, las medidas del consumo de ancho de banda se tomaron con la aplicación MyData Manager.



**Figura 18.** Consumo de Ancho de Banda Aplicación Móvil Desarrollada



**Figura 19.** Consumo de Ancho de Banda Aplicación Google Maps

La figura 18 describe el consumo de ancho de banda de la aplicación móvil desarrollada y la figura 19 describe el consumo de ancho de banda de la aplicación Google Maps, como se puede observar el consumo de ancho de banda en la aplicación Google Maps es mayor que la aplicación móvil desarrollada para el proyecto. Esto indica que aplicación móvil desarrollada no consume altas cantidades de ancho de banda esto permite el ahorro de costos al usuario en el servicio de Internet.

## 6. Conclusiones

El uso de Firebase en el desarrollo de nuestra aplicación permitió disminuir las perdidas en los tiempos de trasmisión de la información desde la aplicación móvil hacia la base de datos, debido a que Firebase mantiene un sincronismo en

tiempo real entre la aplicación y la base de datos.

El desarrollo de la interfaz web mediante la utilización de ReactJs permitió disminuir los tiempos de carga de nuestra aplicación web debido a que no realiza un refrescamiento total de la aplicación, al contrario solamente actualiza los componentes que presentaron cambios.

La integración de chips GPS en los dispositivos móviles a permitido desarrollar pruebas en las que se pudo observar que la marcación de los puntos de ubicación de los usuarios es aceptable sin tener que invertir en costosos dispositivos.

## Referencias

- ¿Cómo funciona React.js? (2017). Recuperado 28 de enero de 2018, a partir de <https://devcode.la/blog/como-funciona-reactjs/>
- Firebase Authentication. (2017, Octubre 11). Recuperado 28 de enero de 2018, a partir de <https://firebase.google.com/docs/auth/?hl=es-419>
- Firebase Realtime Database | Firebase Realtime Database. (2017, Octubre 11). Recuperado 28 de enero de 2018, a partir de <https://firebase.google.com/docs/database/?hl=es-419>
- JavaScript | MDN. (2017, Noviembre 22). Recuperado 28 de enero de 2018, a partir de <https://developer.mozilla.org/es/docs/Web/JavaScript>
- PostgreSQL: About. (1996). Recuperado 28 de enero de 2018, a partir de <https://www.postgresql.org/about/>
- ¿Qué es el lenguaje de programación JAVA? - Base de Conocimientos - ICTEA. (2016). Recuperado 28 de enero de 2018, a partir de <http://www.ictea.com/cs/knowledge>

base.php?action=displayarticle&id=8790

React Native · A framework for building native apps using React. (2017). Recuperado 28 de enero de 2018, a partir de <https://facebook.github.io/react-native/index.html>

Spring Framework. (2017). Recuperado 28 de enero de 2018, a partir de <https://projects.spring.io/spring-framework/>