

# Ingeniería en Computación

# Sede San Carlos

# Desarrollo de Aplicaciones para Dispositivos Móviles

# Grupo 50

# Módulo de Mapas

# Integrantes:

# Yorbi Gerardo Méndez Soto

# Carlos Jimenez Gonzalez

# Juan Miguel Arce Rodriguez

# Daryn Soto Soto

# Martes 07 de Junio del 2016

Contenido

[Introducción 3](#_Toc453112139)

[Análisis del problema 4](#_Toc453112140)

[Solución del problema 5](#_Toc453112141)

[Análisis de resultados 7](#_Toc453112142)

[Conclusiones 8](#_Toc453112143)

[Referencias 9](#_Toc453112144)

# Introducción

Los celulares han llegado a formar parte de la vida cotidiana que además, ha llegado a ser un electrónico indispensable para los seres humanos. Los celulares Smartphone, en la actualidad, es una escogencia regular de productos para los consumidores y, debido a esto, el ecosistema de Android se ha ido expandiendo en varios ámbitos y ha llegado a mostrar su capacidad en otros dispositivos tales como las tabletas y televisores inteligentes.

Las aplicaciones para dispositivos móviles es de los campos con más demanda en el mercado actual, debido a que está al alcance de la mayoría de la población por lo tanto es un mercado factible para las empresas que quieren crecer exponencialmente.

Al utilizar Google Play para su distribución, los desarrolladores han tomado ventaja en este mercado libre, sin proceso de revisión, para la distribución de sus aplicaciones Android compatibles con muchos dispositivos sin necesidad de pago alguno, sin incluir aquellas aplicaciones que tienen un costo de compra. El desarrollo para aplicaciones Android está fuertemente ligado con el consumo de recursos externos a la aplicación, como al usar los servicios de google para mostrar un mapa o para consumir información de una base de datos externa. Además de tener acceso a recursos externos del celular, también tiene acceso a recursos del mismo dispositivo que rondan desde elementos del el hardware, incluyendo la cámara, la base de datos local y el GPS hasta sensores del celular para que la aplicación tenga una mayor aplicación en diferentes mercados.

# Análisis del problema

En general, la aplicación es una plataforma móvil en la cual se publican eventos de carreras. La función principal es permitir a los usuario es ver su progreso en tiempo real y el de sus amigos, sobre una ruta. Los usuarios de esta aplicación son capaces de registrarte y agregar amigos, así como publicar eventos para que otros usuarios tengan la opción de que participen y ligar su cuenta con otras redes sociales como Facebook.

El propósito del proyecto es crear el módulo que maneja el despliegue de las rutas sobre el mapa. Los elementos que se despliegan varían dependiendo sobre el contexto en el que utilice la aplicación; los cuales se listan a continuación.

* Desplegar el proceso del usuario en tiempo real sobre la ruta de una carrera que se está llevando a cabo (un evento).
* Desplegar el proceso de un amigo en tiempo real sobre la ruta de una carrera que se está llevando a cabo (un evento).
* Desplegar el proceso del usuario en tiempo real sobre una ruta que se haya almacenado en el perfil.
* Desplegar el proceso del usuario en tiempo real sobre una ruta que haya almacenado un amigo en su perfil.
* Desplegar una ruta almacenada en el perfil.
* Desplegar las rutas almacenadas de los amigos.

Además de desplegar una ruta ya almacenada, los usuarios son capaces de crear sus propias rutas. Estas rutas se dibujan mientras el usuario corre, y al final del recorrido se puede elegir almacenarla o no.

Se debe de crear un servicio web para almacenar y consumir la información relevante a los usuarios y las rutas.

# Solución del problema

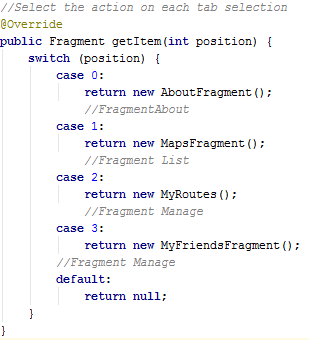
Como contenedor principal de los módulos de la aplicación se dio uso al componente ViewPager, que utiliza fragments para desplegar el contenido de la aplicación en diferentes tabs, o secciones. El ViewPager cambia de página o contenido cuando el usuario selecciona un Tab o cuando desliza el dedo, ya sea para la izquierda o para la derecha, a través de la pantalla. Para manejar los fragmentos que se utilizan en el ViewPager, se debe de dar uso a un FragmentPagerAdapter y los mismos son encargados de contener el código donde se indica que se va mostrar. En la siguiente figura se muestra el método getItem, que se encarga de devolver el fragment que se debe de mostrar dependiendo de la posición actual en que se encuentre el usuario, esto en base al tab seleccionado o en base a la posición de la acción slide. El método es de la clase FragmentPagerAdapter, por ende se debe de sobrescribir.

Ilustración 1: Codigo del FragmentPagerAdapter que obtiene el fragment dado una posición.

Claramente se muestra los fragments que se utilizaron durante el desarrollo de la aplicación, entre ellos, el AboutFragment que se encarga de mostrar información acerca de la aplicación, tal como donde fue creada, para que curso y por quienes fue creada la aplicación, el MapsFragment que se encarga de mostrar datos relacionados con el Mapa tal como la ubicación del usuario, una ruta o camino o bien la ruta que se está grabando en el momento, el fragment MyRoutes que se encarga de mostrar las rutas que tiene almacenado el api, el fragment MyFriendsFragment que despliega la lista de mis amigos para poder acceder a la rutas que ese amigo tiene para luego desplegarlas en el mapa.

Para el despliegue de las rutas se utilizó un fragment que implementa GoogleMap por medio de un API\_KEY, de esta forma se puede dibujar en el mapa la ruta especificada. La ruta se logró dibujar gracias a la función de Android PolylineOptions la cual consiste en trazar una línea recta de punto a punto, estos puntos son la localización exacta del dispositivo móvil, para tener una mejor perspectiva de la ruta dichos puntos se generan de forma automática cada 30 segundos por medio de un Handler y un Runnable

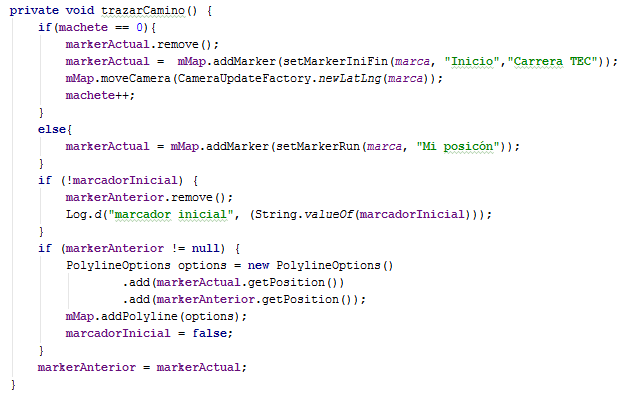


Ilustración 2: Función encargada de trazar camino.

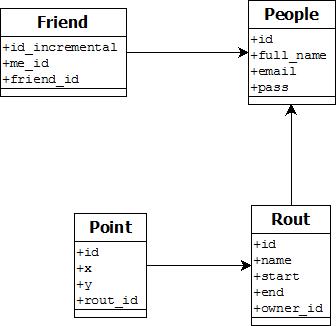
El backend de la aplicación no fue desarrollado nativo en Android, se desarrolló en Laravel y es consumido desde Android.

Ilustración 3: Diseño de la base de datos.

Los datos provienen de las tablas: people, friends, points y rout.

La tabla people almacena todas las personas de la aplicación, con datos personales como nombre completo, email, contraseña, etc. La tabla rout contiene las rutas de las personas, con los datos del nombre de la ruta, un id a quién pertenece la ruta, nombre de la ruta, etc. La tabla points almacena todos las puntos de una ruta y contiene los datos como: punto x, punto y, también un id a la ruta a que pertenece esa localización, La tabla friends es un tabla que relaciona una persona con otras, así se puede saber cuales son los amigos de la persona correspondiente y guarda los id de una persona con los id de los amigos.

Para consumir los datos de la base de datos se montó el api en un servidor, en este caso en xampp. Para el acceso se requiere la direccion ip de la maquina donde está el servidor y en cual puerto, por ejemplo: http://172.24.42.201:8080, y el nombre de la carpeta donde se encuetra el api, por ejemplo: /api\_shopadvisor/public/, de aqui en adelante se pone el nombre de la instancia de la que se va hacer la consulta y el metodo por ejemplo:rout/all, de esta forma el URL queda asi: http://172.24.42.201:8080/api\_shopadvisor/public/rout/all, de esta forma se obtiene todas las rutas.

# Análisis de resultados

|  |  |  |
| --- | --- | --- |
| **Tarea** | **Estado** | **Observaciones** |
| Definir el UI del recorrido actual, recorrido de cada uno de los corredores seguidos, históricos de mis recorridos, históricos de corredores seguidos. | Incompleto | No se implementaron históricos. |
| Construir los servicios web necesarios para conectar con el servidor  principal la información solicitada | Completo |  |
| Actualizar en tiempo real los datos obtenidos del módulo de tracking cuando un  Usuario activa el inicio y final de un recorrido. | Completo |  |
| Pintar en el mapa la ruta definida por el administrador, en caso de que estemos participando en alguna competición. | Completo |  |
| Utilizar el sistema de GPS del sistema operativo. | Completo |  |
| Actualizar la ubicación cada 30 segundos. | Completo |  |

# Conclusiones

Para poder programar una aplicación en Android es importante tener conocimiento acerca del paradigma orientado a objetos, ya que Android maneja la parte de lógica y del controlador de las vistas con clases y objetos de clases.

En la elaboración de la base de datos no hubo mayor problema, en gran medida de que se trata de un módulo de un sistema y las tablas son pocas.

La aplicación cuenta con backend, desarrollado en el framework de PHP llamado Laravel, que facilita mucho trabajo y ahorra tiempo a la hora de generar el código necesario para el funcionamiento de la aplicación, además que el código es entendible, escalable y mantenible.

# Referencias

*Android Google Maps Tutorial*. (2016). *www.tutorialspoint.com*. Retrieved 8 June 2016, from http://www.tutorialspoint.com/android/android\_google\_maps.htm

*Introducción a Laravel para API REST | El blog de Zarovich*. (2014). *Blog.zarovich.org*. Retrieved 8 June 2016, from http://blog.zarovich.org/index.php/2014/06/introduccion-a-laravel-para-api-rest/

J. Arroyo, F. (2016). *Adictosaltrabajo.com*. Retrieved 5 June 2016, from https://www.adictosaltrabajo.com/tutoriales/android-viewpager/

*ViewPager | Android Developers*. (2016). *Developer.android.com*. Retrieved 8 June 2016, from https://developer.android.com/reference/android/support/v4/view/ViewPager.html?hl=es