



UNIVERSIDAD DE BUENOS AIRES

Departamento de Computación

Facultad de Ciencias Exactas y Naturales

Computación Móvil

Segundo cuatrimestre de 2013

PathFinder - Aplicación para Android

16 de mayo de 2013

Integrante	Correo electrónico
Almansi, Emilio Guido	ealmansi@gmail.com
Castro, Alan	alancastro90@gmail.com
Vilerío, Silvio	svilerino@gmail.com

# Índice

1. Descripción y requerimientos	3
2. Arquitectura	4
3. Dependencias	5

# 1. Descripción y requerimientos

La aplicación permite realizar al usuario principalmente tres acciones: activar su dispositivo para comenzar a guardar su ubicación de forma periódica, visualizar la sucesión de puntos guardados unidos mediante rectas en un mapa dentro de la aplicación y compartir su posición más reciente.

Para realizar el proceso de activación del dispositivo y el almacenamiento de los datos, la aplicación se comunica con un servidor dedicado que cumple una función similar a la de una RESTful API, atendiendo tres tipos de pedidos por parte de la aplicación:

- Activación: registra el dispositivo bajo un email y contraseña, que luego son necesarios para acceder a los datos.
- Actualización: agrega un nuevo punto de tipo (latitud, longitud, fecha y hora) en la secuencia de puntos de un dispositivo.
- Descarga: devuelve el historial de actualizaciones de un dispositivo particular.

La visualización del recorrido se realiza sobre un mapa dentro de la aplicación, permitiendo desde la misma interfaz compartir la ubicación más reciente por email mediante el explorador del dispositivo o la aplicación Gmail, presente por defecto en Android.

Alternativamente, la última actualización se puede compartir a través de un tweet, mediante un método de la API provista por Twitter<sup>1</sup>. Esta opción al igual que la anterior, comparte la ubicación mediante un link a Google Maps con un marcador en las coordenadas correspondientes.

---

<sup>1</sup>Documentación de Web Intents: <https://dev.twitter.com/docs/intents#tweet-intent>.

## 2. Arquitectura

La aplicación esta formada por cuatro Activities: Inicio, Ingresar, Activar y Mapa.

InicioActivity consta sencillamente del logo y título de la aplicación, y dos botones redireccionando a Ingresar y Activar (esta última solo es accesible cuando el dispositivo aún no fue activado).

Las otras tres Activities siguen el esquema MVC, con una clase de modelo y una de vista, atribuyendo el rol de controlador íntegramente a la clase de tipo Activity. Se puede ver en la figura 1 que efectivamente toda la comunicación entre el modelo y la vista se genera mediante el controlador.

IngresarActivity y ActivarActivity son similares ya que ambas constan de un formulario y representan una acción puntual que necesita comunicación con el servidor. Por eso es que ambas se relacionan en el diagrama con clases de tipo AsyncTask (Descargar y Activar, respectivamente), ya que a través de ellas se realizan los pedidos al servidor. La Activity conteniendo el mapa no realiza ninguna comunicación con el servidor ya que simplemente consume los datos obtenidos mediante DescargarRecorridoAsyncTask.

Por último, la actualización de las sucesivas posiciones del dispositivo debe ser independiente de la aplicación; es decir, debe comunicarse con el servidor independientemente de si el usuario está utilizando Pathfinder. Para ello se utilizó el recurso de servicios Android en la clase GPSBackgroundService, enviando los datos al servidor mediante la clase EnviarCoordenadasAsyncTask. Para garantizar que el servicio se lance al encenderse el dispositivo, se implementó la clase GPSBackgroundServiceLauncher que extiende BroadcastReceiver y se subscribe al intent BOOT\_COMPLETED, inicializando el servicio al ser llamado.

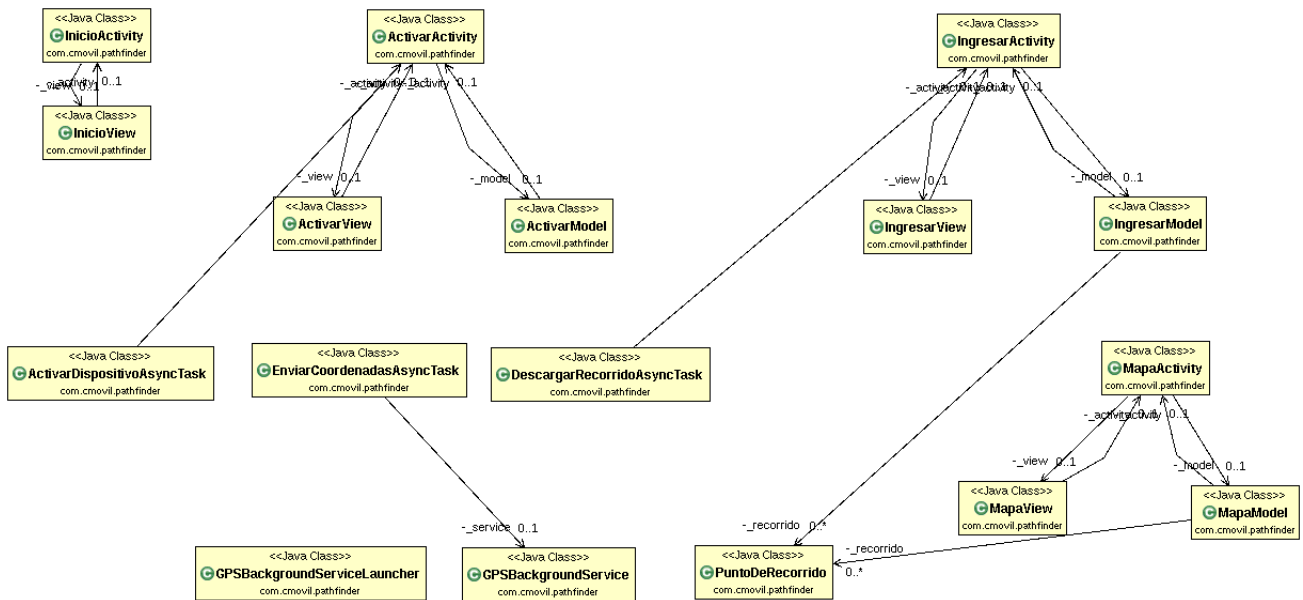


Figura 1: Diagrama generado de forma automática con las relaciones entre las diferentes clases que componen la aplicación.

### 3. Dependencias

Para visualizar un mapa dentro de la aplicación se utilizó Google Maps Android API v2, por lo cual para correr la aplicación es necesario instalar Google Play Services desde el SDK Manager y configurar las dependencias del proyecto y el Manifest.<sup>2</sup>

El servidor dedicado se desarrolló en Node.js; para ejecutarlo solo es necesario instalar la plataforma<sup>3</sup> y lanzar la aplicación JavaScript provista con el código fuente de PathFinder.

---

<sup>2</sup>Instrucciones completas para utilizar Google Maps Android API v2: <https://developers.google.com/maps/documentation/android/start>.

<sup>3</sup>Descarga e instalación de Node.js: <http://nodejs.org/>.