

Introducción.....	2
Desarrollo.....	3
1. GUI .....	3
2. Persistencia .....	3
3. Servicios .....	5
Visualización.....	6

## Introducción

---

El Presente trabajo consiste en una aplicación Android que lleve registro de las distancias recorridas por el usuario cada vez que sale a correr. El sistema deberá:

- Permitir al usuario cargar la información relevante a su configuración (setup inicial).
- Guardar una distancia ingresada en determinada fecha y hora a través del teclado.
- Obtener la distancia recorrida por el usuario a través del uso del GPS.
- Mostrar el recorrido realizado a través de la Google Maps Android API v2.
- Llevar un registro de las distancias ingresadas en la aplicación.
- Permitir al usuario compartir por mail los logros alcanzados.

## 1. GUI

Con el fin de utilizar las características que cada Activity posee, en el transcurso del desarrollo de la interfaz de usuario se utilizaron:

- **PreferenceActivity** para el setup del usuario, obteniendo así la persistencia de la configuración sin trabajo adicional.
- **Activity** para la interfaz en general.
- **ListActivity** para la visualización del historial de registros guardados.
- **FragmentActivity** para la visualización del recorrido realizado a través de Google Maps Android API.

Es importante mencionar en este punto que todos los layouts fueron generados en archivos XML para luego asociarlos a su respectiva Activity cuando ésta es creada, así como también todos los mensajes y textos que aparecen se encuentran en otro archivo XML (strings.xml).

Con lo descrito anteriormente, obtenemos un código desacoplado dejando solamente en la actividad la lógica asociada a dicha actividad.

## 2. Persistencia

Se optó por trabajar con una base de datos de sqlite3, con todos los beneficios que implica disponer de una base de datos en la aplicación (definición de tablas, atributos, queries, etc.).

Se definieron dos tablas:



```

run_logs
_id integer primary key autoincrement
length real not null
date varchar(20) not null    //Format: HH:MM
time varchar(20) not null    //Format: YYYY-MM-DD

```

```

run_locations
_id integer primary key autoincrement
latitude real not null
longitude real not null
run_id integer

```

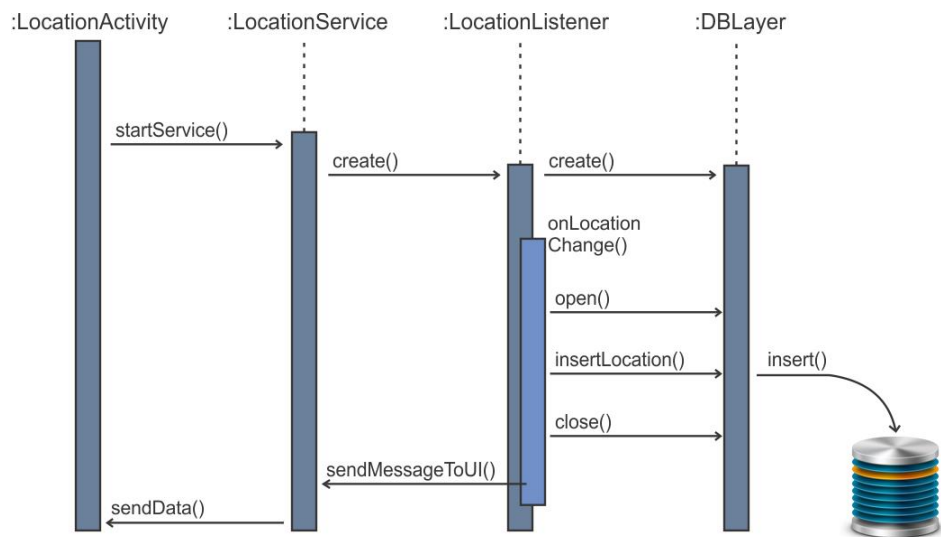
Los métodos para la interacción con la base de datos se listan a continuación con su javadoc:

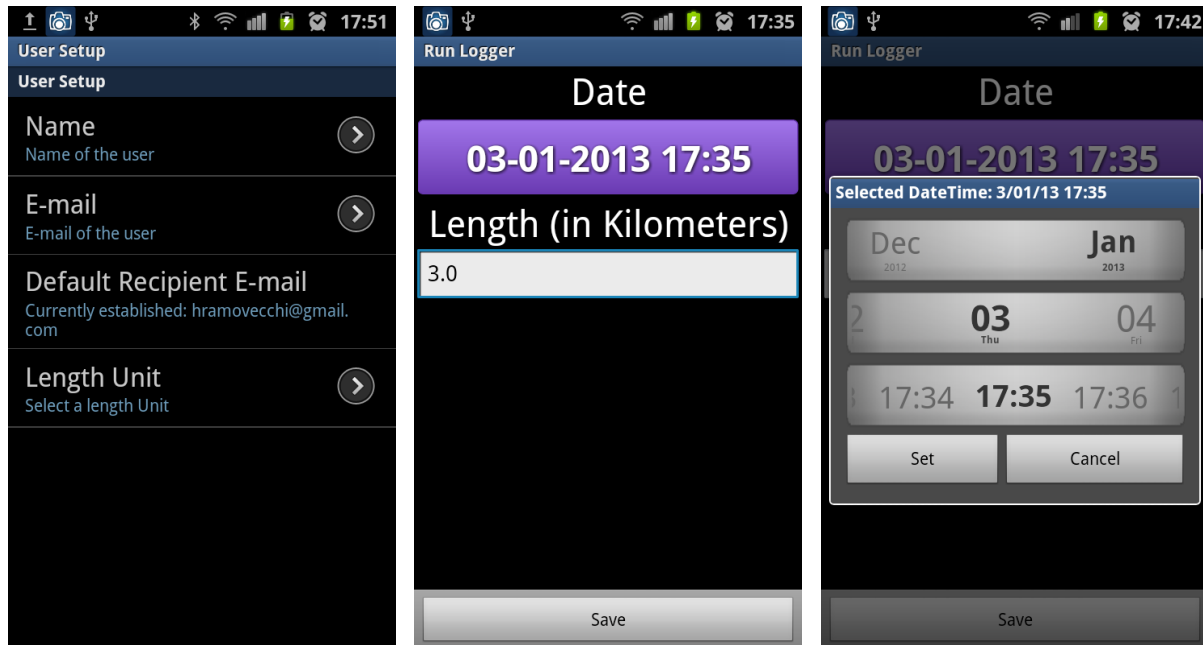
- `Cursor fetchAllLengths(String order)`
  - Fetches the length, date and \_id for all the registers in the run\_logs table
- `Cursor fetchAllLengthsWithTime()`
  - Fetches the length, date, time and \_id for all the registers in the run\_logs table
- `Cursor fetchLength(long rowId)`
  - Fetches the register identified by the ID passed as a parameter
- `Cursor fetchLengthAfterDate(String date, String order)`
  - Fetches the registers posterior to the date passed as a parameter
- `Cursor fetchLastlength()`
  - Fetches the last record registered
- `boolean updateLength(long rowId, Float length, String date)`
  - Updates a Length row
- `insertLength(String date, String time, Float length)`
  - Inserts a new record (date, time, length) into the DB
- `boolean deleteLength(long rowId)`
  - Deletes a length row matching the rowId param
- `void deleteAllRuns()`
  - Deletes all the run\_logs values
- `Cursor fetchOldestLength()`
  - Fetches the oldest record registered
- `Cursor getLocations(long run_id)`
  - Return the locations matching with the run\_id param
- `boolean deleteLocations(long run_Id)`
  - Delete the Locations with the run\_id param
- `boolean deleteUnusedLocations()`
  - Deletes all unused Locations
- `void deleteAllLocations()`
  - Deletes all the run\_location values
- `long insertLocation(double latitude, double longitude)`
  - Inserts a new Location row
- `boolean setRunIdToLastSavedLocations(long run_Id)`
  - Sets the run\_id to last saved locations
- `long countAllRunLocations()`
  - Returns the number of all the rows in the run\_locations Table
- `long countRunLocations(long run_id)`
  - Returns the number of rows with the given run\_id
- `boolean existLocationsToBeSaved()`
  - Returns if there are locations not saved with a valid run\_id value

### 3. Servicios

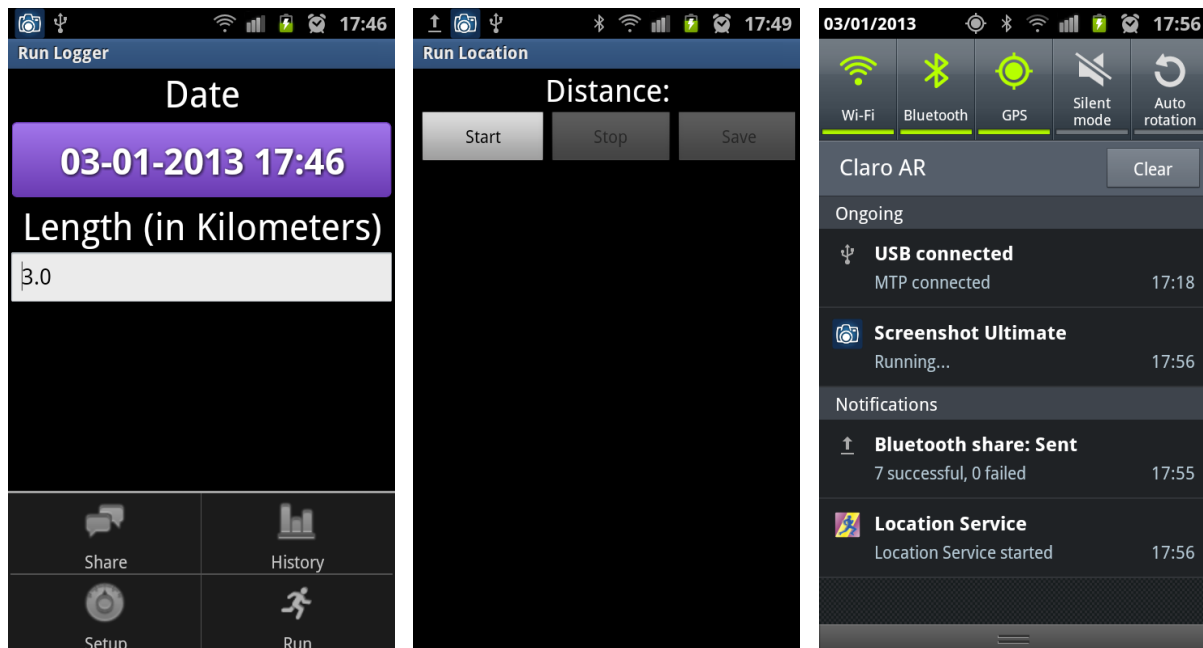
Los servicios en Android son utilizados con el fin de desacoplar grandes funcionalidades de código de la parte visual. Con dicha justificación se optó por crear un servicio que maneje la interacción con el servicio de locación de GPS, salvando dichas coordenadas en la base de datos, he informando a la interfaz visual a través de mensajes asincrónicos, información relevante al usuario.

A continuación se detalla el workflow de lo descrito con anterioridad:



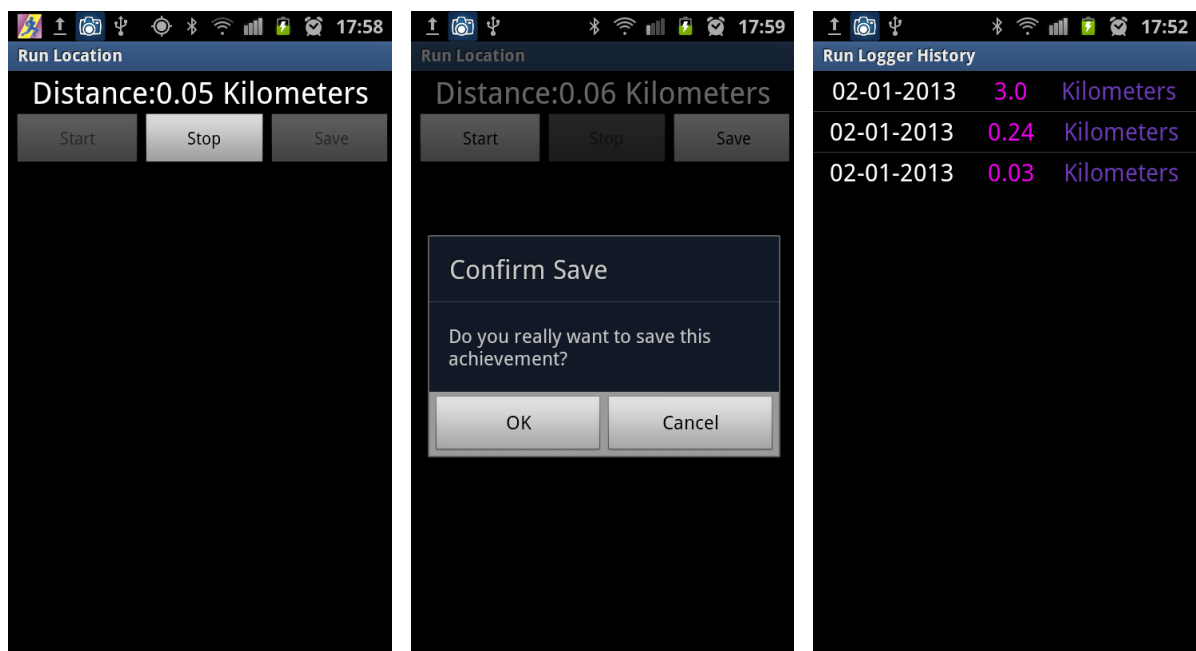


En la primera imagen puede apreciarse la pantalla correspondiente a la configuración de usuario. Las siguientes dos imágenes son correspondientes al ingreso de una distancia a través del teclado. El usuario elige una fecha y hora determinada para ingresar la distancia que recorrió.

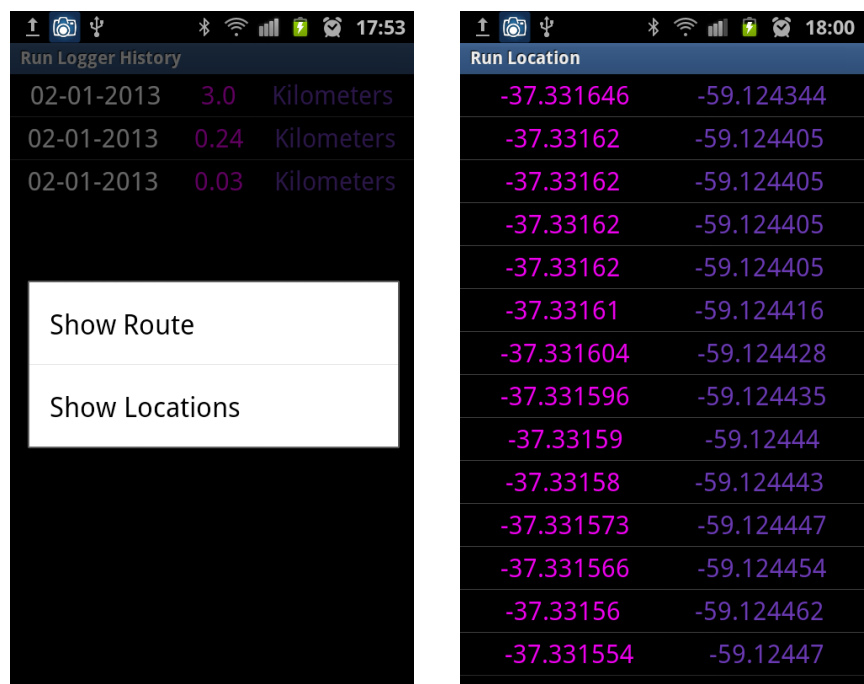


En la main page se encuentra el menú donde el usuario podrá seleccionar las diferentes acciones a realizar. Volver a la configuración, ver el historial de actividades, compartir los logros a través del correo electrónico y por último la opción de correr utilizando el servicio de GPS.

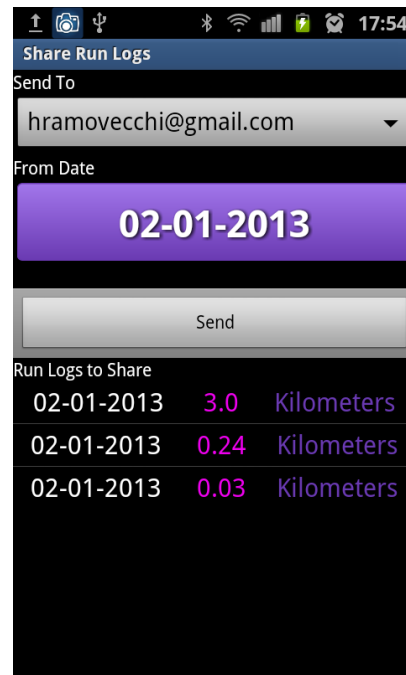
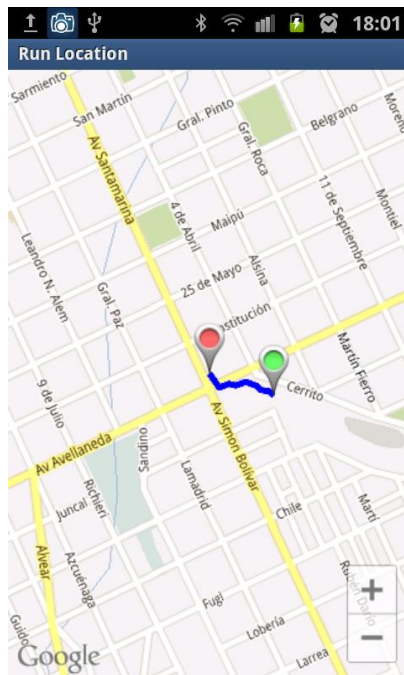
Al empezar a correr utilizando el servicio de GPS se iniciará el servicio para registrar los puntos recorridos, como puede apreciarse en el stack de actividades.



Cuando el usuario decida parar de realizar la actividad física, se confirmará a través de un Dialog si desea guardar la actividad realizada. Toda actividad que es guardada en la aplicación puede verse en un ListActivity con el formato que puede apreciarse en la imagen titulada Run Logger History



Al tocar en cada fila de la lista el usuario tendrá la posibilidad de eliminar dicho registro, modificar la fecha y distancia alcanzada, así como también poder ver las Locations que se guardaron para el recorrido realizado, esto último seguramente no será información interesante para un usuario normal, pero se realizó con un enfoque académico para obtener los valores guardados .



Y por último también se podrá visualizar el recorrido realizado al seleccionar la opción “Show Route”. La marca verde corresponde al punto de origen y el rojo al punto final de la actividad física.

La última imagen, correspondiente a la Activity relacionada a compartir los logros realizados permite al usuario, luego de seleccionar el correo electrónico del destinatario a través de las opciones del combobox y una fecha para el criterio de selección, a través del botón send se enviará la información por correo electrónico.