



Título del Proyecto:

AlcoiBTT

Autor:

Sánchez, Antonio

Escalona, Bruno

Director:

Pérez, Rubén

Silvestre, Javier

TESINA PARA LA OBTENCIÓN DEL TÍTULO DE:

**Máster en Desarrollo de
Aplicaciones sobre Dispositivos
Móviles**

Septiembre del 2016

Contenido

Título del Proyecto:	1
Autor:	1
Director:	1
Máster en Desarrollo de Aplicaciones sobre Dispositivos Móviles.....	1
Introducción	3
Descripción del problema	3
Objetivos	3
Motivación	4
Tecnologías utilizadas	5
Arquitectura de la aplicación	7
Esquema del diseño	7
Aplicación nativa Android	7
Modelo de datos	11
Vistas	13
Aplicación nativa Android	13
Panel de administrador	13
Servicio RESTful	18
Blog WordPress	20
Conclusiones	22
Anexos	22
Listado de fuentes entregadas / Código fuente en GitHub	22
Manual de usuario	22
Aplicación nativa Android	22

Introducción

Descripción del problema

Debido al aumento de usuarios amateurs de bicicletas de montaña en la región de Alcoy, se propuso al Ayuntamiento la realización de una aplicación para que los amantes de este deporte pudieran encontrar rutas que realizar en las cercanías de la ciudad.

Con esa idea en mente, se desea crear una aplicación en la que los usuarios puedan consultar rutas ciclistas en un mapa de Google, además de información adicional como número de kilómetros, dificultad, etc, incluyendo la posibilidad de seguir la ruta seleccionada

Como información adicional para que la aplicación sea más completa, se añade números de teléfono de emergencias, de tiendas de bicicletas, de lugares de interés como restaurantes u hoteles.

El ayuntamiento se encargará igualmente de administrar un blog en el que el usuario pueda consultar el estado de rutas, consejos o todo tipo de curiosidades relacionadas con el mundo del mountain-bike.

Inicialmente la aplicación será una aplicación nativa Android

Objetivos

Al finalizar la aplicación deberán de conseguirse los siguientes objetivos:

- Diseño de la interfaz de usuario de la aplicación nativa Android.
- Implementación de la interfaz de usuario, la cual deberá incluir los siguientes puntos principales:
 - Menú lateral.
 - Pantalla con información turística del ayuntamiento.
 - Pantalla con lector RSS para el Blog.
 - Pantalla con la información de interés para los usuarios.
 - Pantalla con las rutas.
- Creación de un Blog
- Creación de un servicio REST y base de datos del cual se alimentará la aplicación.
- Creación de un panel de administrador para alimentar la base de datos por parte del ayuntamiento.



Motivación

La motivación es Profesional.

El ayuntamiento de Alcoi acude a nosotros para cubrir su necesidad

Tecnologías utilizadas

La aplicación nativa está desarrollada en Java, utilizando las siguientes librerías externas:

GSON: Para facilitar la serialización / deserialización de los objetos JSON intercambiados en la comunicación con los Servicios Web REST

<https://sites.google.com/site/gson/gson-user-guide>

Los objetos necesarios para mapear los JSON intercambiados residen en el paquete **modelo**:

Http-Request: Para facilitar las llamadas http REST (GET, POST, PUT, DELETE...)

<https://github.com/kevinsawicki/http-request>

El servicio REST se ha realizado utilizando Node JS, seleccionado por:

- Ser una implementación de JavaScript en el servidor (y se ha usado JavaScript también para la parte de interfaz de usuario, por lo tanto evitando emplear otros lenguajes).
- Es extremadamente eficiente y está especializado para creación de aplicaciones escalables.
- Construir servicios REST es muy sencillo.

Para la persistencia de datos se ha empleado Mongo DB, debido a:

- Guarda BSON (representación binaria de JSON) en vez de datos en tablas relacionales por lo que se guardarán los datos de una forma mucho más sencilla al pasar datos JSON a través del servicio REST.
- Además se realizó en Mongo DB por deseo personal de indagar y profundizar en esta forma de persistencia.

Para la creación del panel de Administrador se ha usado HTML5, CSS y JavaScript, así como los frameworks Materializecss y backbone JS.

Materializecss se empleó para mantener una consistencia entre la aplicación nativa Android (material design) y el panel de administrador. Es un framework con un comportamiento similar a bootstrap que incorpora componentes y animaciones que proporcionan más información a los usuarios. Además, de un único sistema de respuesta subyacente en todas las plataformas que permite al usuario una experiencia más unificada.

Backbone JS ayuda a crear la estructura de la aplicación web proveyendo modelos y eventos personalizables, colecciones con una rica API con muchas funciones, vistas con controladores de eventos y conecta todo a nuestro servicio RESTful con una interface JSON de una forma muy sencilla.



Finalmente el desarrollo del Blog como complemento se desarrolla con Wordpress, empleando una plantilla desplegada en nuestro servidor.

Arquitectura de la aplicación

Esquema del diseño

Aplicación nativa Android

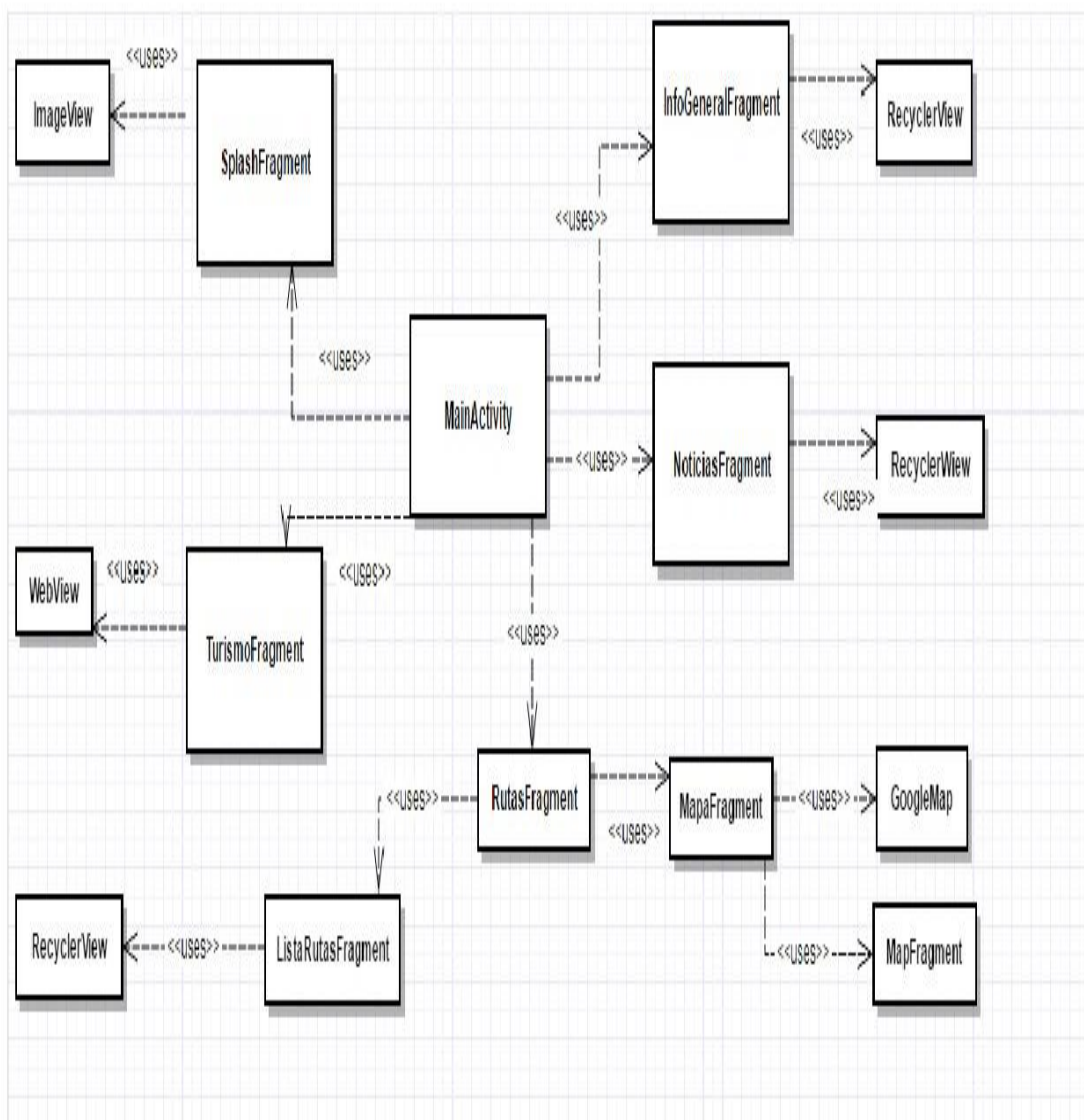


Figura 1 Esquema de diseño de la aplicación Android

La clase principal y parte central de la aplicación es la Activity **MainActivity**.

Dispone de un **NavigationDrawer** donde se ubican las diferentes funcionalidades.

Dispone de un **FrameLayout** llamado fragment.

Cada funcionalidad esta soportada por un fragment, el cual que se va cargando en dicho frameLayout según se va necesitando de forma dinámica.



Las funcionalidades a cubrir se implementan con fragmentos y son las siguientes:

- **SplashFragment**, es el fragmento inicial con la imagen corporativa, utilizando para ello un **ImageView**.

- **TurismoFragment**, es el fragmento encargado de cargar la web de turismo de Alcoi (www.alcoyturismo.com), utilizando para ello un **WebView**.

- **InfoGeneralFragment**, es el fragmento encargado de dar acceso a la parte de información general de Alcoi, lugares de interés general, tiendas...

La información la muestra en un **RecyclerView** y es cargada por una **AsyncTask** llamada **AsyncHttpTask**.

Como curiosidad destacar que esta tarea se trae el JSON y además hace la descarga de las imágenes asociadas, para evitar descargarlas a la hora de mostrarlas.

Esto lo hago a través de la clase de ayuda **Helper** en su método "downloadImage".

```
public static Bitmap downloadImage(String url) {
    Bitmap bitmap = null;
    InputStream stream = null;
    BitmapFactory.Options bmOptions = new BitmapFactory.Options();
    bmOptions.inSampleSize = 1;
    Log.d("Helper.downloadImage from:", url);
    try {
        stream = getHttpConnection(url);
        bitmap = BitmapFactory.decodeStream(stream, null, bmOptions);
        stream.close();
    }
    catch (IOException e1) {
        e1.printStackTrace();
        Log.e("Helper", e1.getMessage());
    }
    return bitmap;
}

// Makes HttpURLConnection and returns InputStream

private static InputStream getHttpConnection(String urlString)
throws IOException {
    Log.d("Helper.getHttpConnection from:", urlString);
    InputStream stream = null;
    URL url = new URL(urlString);
    URLConnection connection = url.openConnection();

    try {
        HttpURLConnection httpConnection = (HttpURLConnection)
connection;
        httpConnection.setRequestMethod("GET");
        httpConnection.connect();

        if (httpConnection.getResponseCode() ==
HttpURLConnection.HTTP_OK) {
            stream = httpConnection.getInputStream();
        }
    }
}
```




```
    }  
    catch (Exception ex) {  
        ex.printStackTrace();  
        Log.e("Helper.getHttpConnection from:", urlString);  
    }  
    return stream;  
}
```

Este fragmento hace uso de intenciones ACTION_DIAL para hacer llamadas

```
    private void llamar(String tlfo) {  
        Log.d(TAG+"<-->llamar", "a :"+tlfo);  
        Intent iCall=new  
Intent(Intent.ACTION_DIAL, Uri.parse("tel:"+tlfo));  
        startActivity(iCall);  
    }
```

y ACTION_VIEW con paquete com.google.android.apps.maps para navegar con Google Maps a una dirección

```
    private void navegar(String direccion) {  
        Log.d(TAG+"<-->navegar", "a :"+direccion);  
        Uri gmmIntentUri = Uri.parse("geo:0,0?q="+direccion);  
        Intent mapIntent = new Intent(Intent.ACTION_VIEW, gmmIntentUri);  
        mapIntent.setPackage("com.google.android.apps.maps");  
        if (mapIntent.resolveActivity(getActivity().getPackageManager())  
!= null) {  
            startActivity(mapIntent);  
        }  
    }  
}
```

- **NoticiasFragment**, es el fragmento encargado de dar acceso a las noticias de Alcoi (RSS).

Las noticias las muestra en un **RecyclerView** y es cargada por una **AsyncTask** llamada **AsyncHttpTask**.

Como curiosidad destacar que esta tarea se trae el JSON y además hace la descarga de las imágenes asociadas, para evitar descargarlas a la hora de mostrarlas

Esto lo hago a través de la clase de ayuda **Helper** en su método "downloadImage".

```
    public static Bitmap downloadImage(String url) {  
        Bitmap bitmap = null;  
        InputStream stream = null;  
        BitmapFactory.Options bmOptions = new BitmapFactory.Options();  
        bmOptions.inSampleSize = 1;  
        Log.d("Helper.downloadImage from:", url);  
        try {  
            stream = getHttpConnection(url);  
            bitmap = BitmapFactory.decodeStream(stream, null, bmOptions);  
            stream.close();  
        }  
    }
```



```
        catch (IOException e1) {
            e1.printStackTrace();
            Log.e("Helper", e1.getMessage());
        }
        return bitmap;
    }

    // Makes HttpURLConnection and returns InputStream

    private static InputStream getHttpConnection(String urlString)
    throws IOException {
        Log.d("Helper.getHttpConnection from:", urlString);
        InputStream stream = null;
        URL url = new URL(urlString);
        URLConnection connection = url.openConnection();

        try {
            HttpURLConnection httpConnection = (HttpURLConnection)
connection;
            httpConnection.setRequestMethod("GET");
            httpConnection.connect();

            if (httpConnection.getResponseCode() ==
HttpURLConnection.HTTP_OK) {
                stream = httpConnection.getInputStream();
            }
        }
        catch (Exception ex) {
            ex.printStackTrace();
            Log.e("Helper.getHttpConnection from:", urlString);
        }
        return stream;
    }
}
```

- **RutasFragment**, es el fragmento encargado de mostrar las rutas disponibles en un mapa, así como una lista con todas las rutas para permitir una selección por el usuario.

Para ello, dispone a su vez de 2 **FrameLayout** (fragmentMapa y fragmentListaRutas), donde, de forma dinámica se insertan los fragmentos que dan soporte al mapa (**MapaFragment**) y al **RecyclerView** con todas las rutas (**ListaRutasFragment**).

La carga de rutas es realizada por una **AsyncTask** llamada **AsyncObtenRutasTask**.

Como decisión de diseño, utilizo en este punto el patrón **Delegate**.

Creo una interface **RutasEventListener** con un único método llamado "rutasCargadas()".

En RutasFragment, tengo una Lista de elementos que implementan esta interfaz, y que serán:

- MapaFragment
- ListaRutasFragment

Cuando la tarea asíncrona termina de obtener las rutas, llama al método "setRutas", cuya implementación consiste en llamar a todos los elementos que implementan la interfaz delegada (MapaFragment y ListaRutasFragment) para que se carguen, al disponer ya de los datos necesarios para ello.

```
public void setRutas(List<Ruta> rutas) {  
    Log.d(TAG+"<-->setRutas", rutas.toString());  
    this.rutas=rutas;  
    for (RutasEventListener listener: escuchadoresRutas) {  
        listener.rutasCargadas();  
    }  
}
```

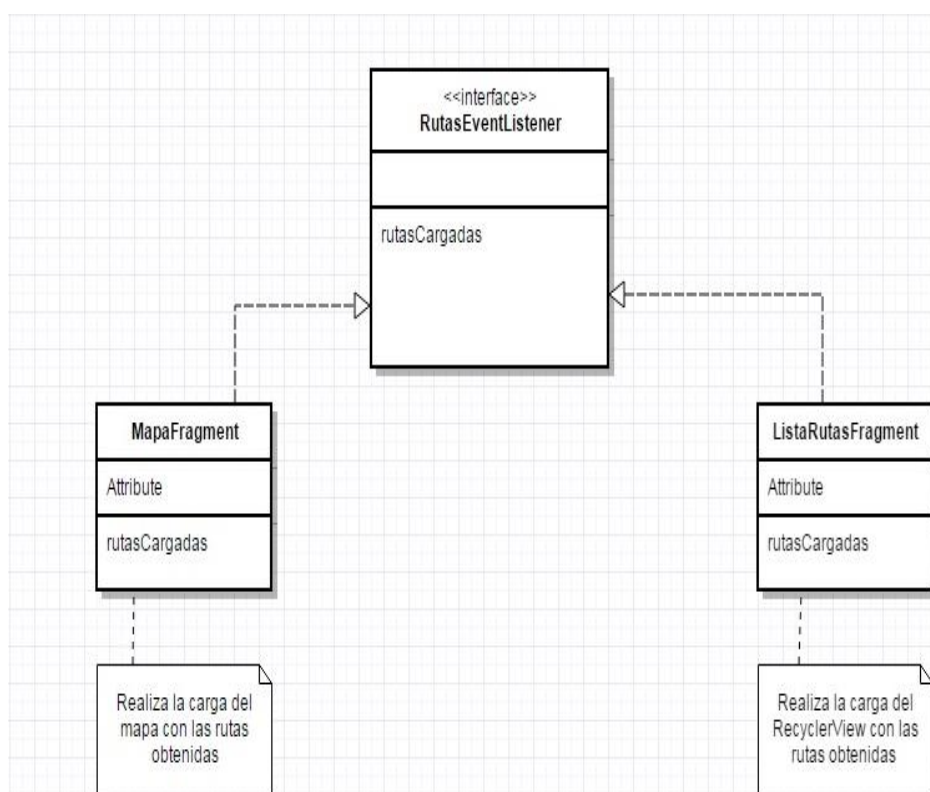


Figura 2 Esquema para la carga de rutas.

Modelo de datos

El esquema de datos empleado se explica más en detalle en el capítulo del servicio RESTful donde se dan todos los end points y el formato de datos que se emplean.

Los datos se han guardado en una base de datos Mongo DB por lo tanto no relacional ly basada en colecciones y datos JSON binarios.

Se guardarán dos colecciones una de ellas tendrá los datos de las rutas y la otra tendrá los datos de la información general. Cada uno de ellos tendrá un formato como el que se puede ver en las imágenes siguientes:

```
{
  "_id": ObjectId("57a629dc9f3ad87c23155c5d"),
  "coordenadas": [
    {
      "latitud": "38.669556000",
      "longitud": "-0.461557000",
      "altitud": "704.700000"
    },
    {
      "latitud": "38.669066000",
      "longitud": "-0.463280000",
      "altitud": "723.500000"
    },
    {
      "latitud": "38.667437000",
      "longitud": "-0.463984000",
      "altitud": "728.500000"
    },
    {
      "latitud": "38.666207000",
      "longitud": "-0.463667000",
      "altitud": "737.000000"
    },
    ...
  ],
  "descripcion": "Ruta larga para hacer en una mañana",
  "dificultad": "Muy Dificil",
  "distancia": 88.75,
  "esCircular": true,
  "nombre": "Alcoi y alrededores"
}
```

Figura 3 Modelo de datos para las rutas.

En el modelo de datos de las rutas podemos ver los siguientes campos:

- **_id**: donde se guarda el id del elemento creado en la base de datos y tener acceso al mismo de forma individual.
- **coordenadas**: que guarda un array con las coordenadas, **latitud**, **longitud** y **altitud**.
- **descripción**: se guarda la descripción de la ruta.
- **dificultad**: un string donde se guarda la dificultad (Fácil, Medio, Dificil y Muy Dificil).
- **distancia**: double con el valor en km de la distancia.
- **esCircular**: boolean con true o false dependiendo de si es circular o no.
- **nombre**: el nombre de la ruta.



```
{
  "_id": ObjectId("579de42d209212e80024df8f"),
  "direccion": "C/Casablanca, 3 y 5 - 03801, Alcoy",
  "icono": "http://rutas-bttalcoi.rhcloud.com/public/icons/ic_phone_black_36dp.png",
  "nombre": "Policia municipal Alcoi",
  "telf": NumberInt(965537145),
  "tipo": "telefonos_interes"
}
```

Figura 4 Modelo de datos para las informaciones.

En el modelo de datos de las informaciones podemos ver los siguientes campos:

- **_id**: donde se guarda el id del elemento creado en la base de datos y tener acceso al mismo de forma individual.
- **dirección**: String con la dirección.
- **icono**: String con el url donde está guardado el icono.
- **nombre**: Nombre del elemento guardado.
- **telf**: integer con el número de teléfono.
- **tipo**: tipo de teléfono creado.

Vistas

Las vistas del proyecto las podemos diferenciar claramente en dos, la vista e interfaz de usuario de la aplicación nativa Android y las vistas del panel de administrador para administrar todo el Back-End por parte del ayuntamiento.

Aplicación nativa Android

Se dispone de una actividad con fragmentos que cubren las funcionalidades objetivo.

Más detalle en el manual de usuario

Panel de administrador

El panel de administrador se compone de una single page subdividida mediante tabs en cuatro vistas diferenciadas según la funcionalidad de cada una de ellas:

- Vista para la creación de rutas.
- Vista para la modificación y eliminación de rutas.
- Vista para la creación de información general.
- Vista para la modificación y eliminación de información.

Vista para la creación de rutas

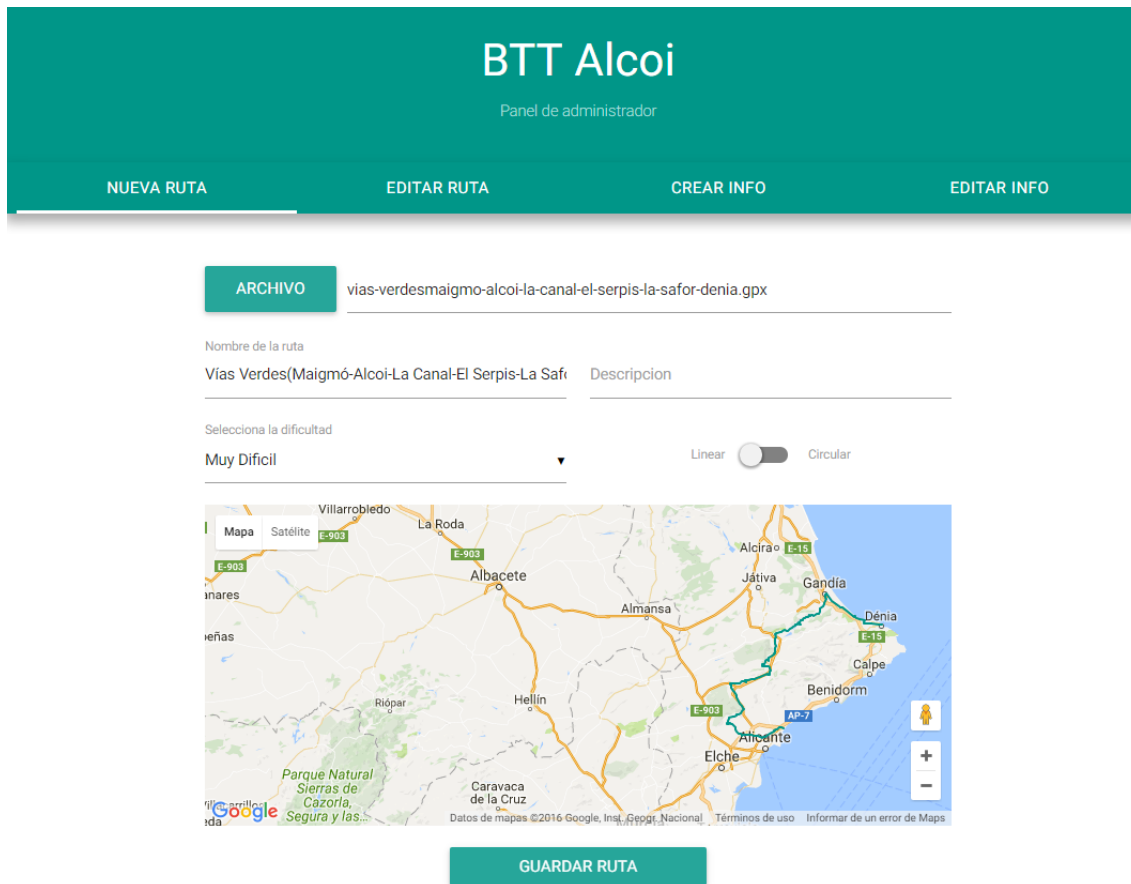


Figura 5 Vista para la creación de rutas.

El la vista para la creación de rutas el usuario puede seleccionar un fichero “.gpx”, el fichero es procesado mediante JavaScript y transformado a un objeto JSON como se describe en la parte de modelos de datos. Se obtienen las coordenadas y la distancia calculada con cada uno de los puntos de la ruta.

Además en el mapa de google se pre visualiza la ruta para comprobar de una forma sencilla si es la ruta deseada.

Al clicar el botón de **guardar ruta** se realiza un HTTP POST /rutas para guardarla en la base de datos.

Vista para la modificación y eliminación de rutas

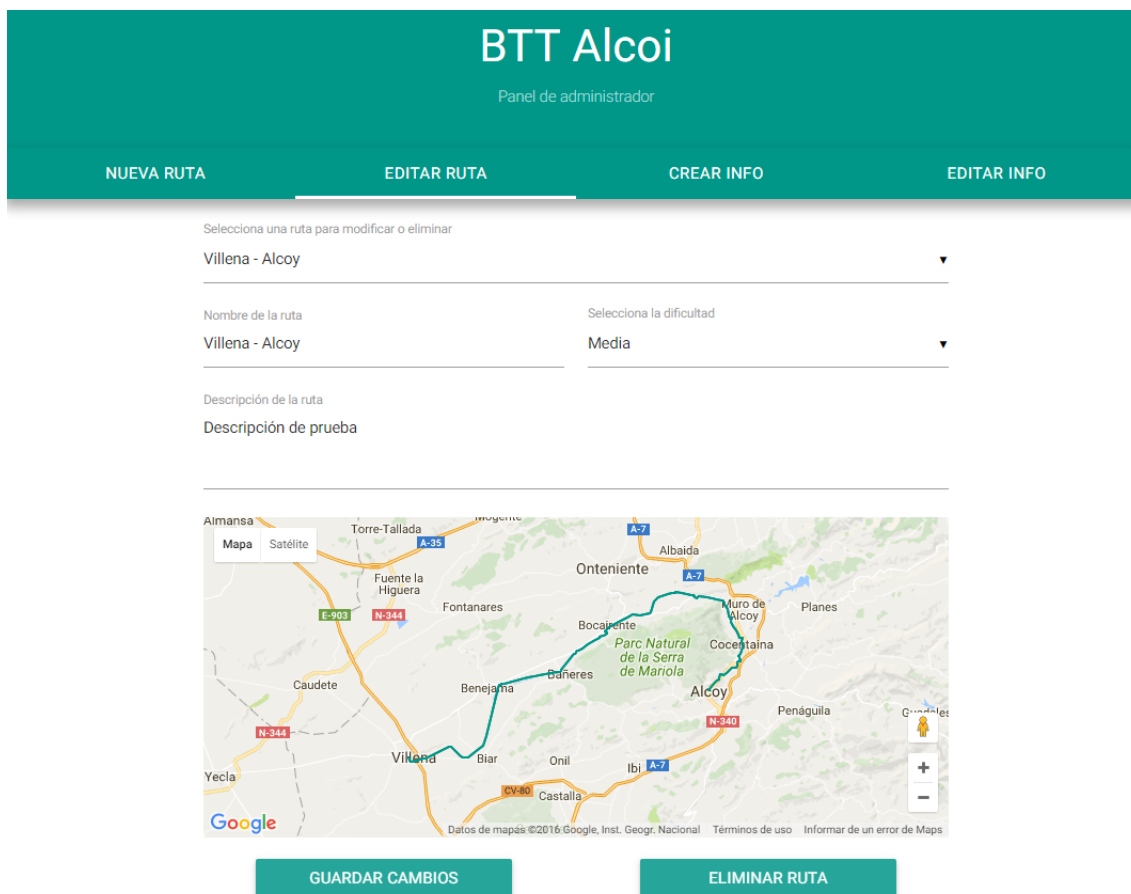


Figura 6 Vista para la modificación y eliminación de rutas.

En la vista de modificación y eliminación de rutas se pueden seleccionar todas las rutas disponibles que se muestran en un desplegable. Estas rutas se obtienen mediante un HTTP GET /rutas.

Se pueden modificar cada uno de los campos y clicar **guardar cambios** para realizar un HTTP PUT /rutas/:id.

Para eliminar la ruta seleccionada basta con clicar el botón de **eliminar ruta** y se realiza un HTTP DELETE /rutas/:id.

Vista para la creación de la información



The screenshot shows the 'BTT Alcoi' administrator panel. At the top, there is a teal header with the text 'BTT Alcoi' and 'Panel de administrador'. Below the header is a navigation bar with four buttons: 'NUEVA RUTA', 'EDITAR RUTA', 'CREAR INFO' (which is highlighted), and 'EDITAR INFO'. The main content area contains a form for creating information. It has four input fields: 'Nombre de la información', 'Teléfono', 'Dirección', and a dropdown menu labeled 'Selecciona tipo de información' with 'Tiendas' selected. A teal button labeled 'GUARDAR INFO' is positioned at the bottom of the form.

Figura 7 Vista para la creación de información.

El la vista para la creación de información general el usuario puede añadir datos que se transforman en JSON como se describe en la parte de modelos de datos.

Al clicar el botón de **guardar info** se realiza un HTTP POST /info para guardarla en la base de datos.



Vista para la modificación y eliminación de la información

The screenshot shows the 'BTT Alcoi' administrator panel. At the top, there is a green header with the title 'BTT Alcoi' and the subtitle 'Panel de administrador'. Below the header, there are four navigation buttons: 'NUEVA RUTA', 'EDITAR RUTA', 'CREAR INFO', and 'EDITAR INFO'. The 'EDITAR INFO' button is currently selected. The main content area is white and contains a form for editing information. At the top of the form, there is a dropdown menu labeled 'Selecciona los datos de información para modificar o eliminar' with the selected value 'Bicicletas Sanegre'. Below this, there are two columns of form fields. The left column has 'Nombre de la información' (Bicicletas Sanegre) and 'Teléfono' (966520019). The right column has 'Selecciona tipo de información' (Alquiler) and 'Dirección' (Pol. Ind. Cotes Baixes, C/ B,1 Alcoi). At the bottom of the form, there are two green buttons: 'GUARDAR CAMBIOS' and 'ELIMINAR INFO'.

Figura 8 Vista para la modificación y eliminación de la información.

En la vista de modificación y eliminación de información se pueden seleccionar todas las informaciones generales disponibles que se muestran en un desplegable. Estas informaciones se obtienen mediante un HTTP GET /info.

Se pueden modificar cada uno de los campos y clicar **guardar cambios** para realizar un HTTP PUT /info/:id.

Para eliminar la información seleccionada basta con clicar el botón de **eliminar info** y se realiza un HTTP DELETE /info/:id.

Servicio RESTful

El servicio RESTful se ha creado mediante Node JS como se describe en el apartado de tecnologías. Dando respuesta desde el END point <http://rutas-bttalcoi.rhcloud.com/> a las siguientes peticiones:

- **HTTP POST** para la creación de:
 - **POST /rutas** creación de las rutas. El formato de JSON es el siguiente:

```
{ nombre: String, //Nombre de la ruta
  descripcion: String, //Descripción de la ruta
  dificultad: String, //Dificultad de la ruta
  distancia: Number, //Distancia en km
  esCircular: Boolean, //Si es circular o no
  coordenadas:[{
    latitud: String, //Latitud
    longitud: String, //Longitud
    altitud: String, //Altitud
  }]
}
```

- **POST /info** creación de la información general. El formato de JSON es:

```
{ nombre: String, //Nombre de la información
  tipo: String, //Tipo de información
  telf: Number, //Teléfono de contacto
  icono: String, //url del icono
  direccion: String, //Dirección de contacto
}
```

- **HTTP GET** para la obtener:
 - **GET /rutas** devuelve una lista de todas las rutas disponibles en formato JSON:

```
{rutas:
  [{
    _id: Number, //Id de la ruta
    nombre: String, //Nombre de la ruta
    descripcion: String, //Descripción de la ruta
    dificultad: String, //Dificultad de la ruta
    distancia: Number, //Distancia en km
    esCircular: Boolean, //Si es circular o no
    coordenadas:[{
      latitud: String, //Latitud
      longitud: String, //Longitud
      altitud: String, //Altitud
    }]
  },
  {...}] //Se repite por cada ruta
}
```

- **GET /rutas/:id** devuelve una ruta en formato JSON donde el parámetro :id es el id de la ruta.

```
{rutas:
```



```
[{
  _id: Number,           //Id de la ruta
  nombre: String,       //Nombre de la ruta
  descripcion: String,  //Descripción de la ruta
  dificultad: String,   //Dificultad de la ruta
  distancia: Number,   //Distancia en km
  esCircular: Boolean, //Si es circular o no
  coordenadas:[{
    latitud: String,    //Latitud
    longitud: String,   //Longitud
    altitud: String,    //Altitud
  }]
}]
}
```

- **GET /rutas/:id/download** devuelve un fichero .gpx con la ruta de id indicada para poder ser utilizado en una aplicación de guiado.
- **GET /info** devuelve una lista de toda la información general en formato JSON:

```
{info:
  [{ _id: Number,           //Id de la información
    nombre: String,       //Nombre de la información
    tipo: String,         //Tipo de información
    telf: Number,         //Teléfono de contacto
    icono: String,        //url del icono
    direccion: String,    //Dirección de contacto
  },
  {...}]                  //Se repite por cada info
}
```

- **GET /info/:id** devuelve una información en formato JSON donde el parámetro :id es el id de la información.

```
{info:
  [{ _id: Number,           //Id de la información
    nombre: String,       //Nombre de la información
    tipo: String,         //Tipo de información
    telf: Number,         //Teléfono de contacto
    icono: String,        //url del icono
    direccion: String,    //Dirección de contacto
  },
  {...}]                  //Se repite por cada info
}
```

- **HTTP PUT** para la modificar:
 - **PUT /rutas/:id** modifica la ruta con id el indicado en el parámetro :id.
 - **PUT /info/:id** modifica la información con id el indicado en el parámetro :id.
- **HTTP DELETE** para la eliminar:
 - **DELETE /rutas/:id** elimina la ruta con id el indicado en el parámetro :id.
 - **DELETE /info/:id** elimina la información con id el indicado en el parámetro :id.

Blog WordPress

Debido a la necesidad del ayuntamiento de informar a los usuarios de nuevas rutas, de estado de las mismas, de curiosidades, etc. Se decidió crear un blog mediante WordPress para que mediante un lector RSS o directamente en la página del blog el usuario pudiera estar al tanto de las últimas informaciones.

Se creó un nuevo proyecto en OpenShift (para desplegar el blog de forma gratuita) PHP y MySQL y mediante la ayuda de OpenShift y su código de GitHub para la creación de un blog wordpress, se creó el blog. Para más información acudir a la siguiente documentación <https://github.com/openshift-quickstart/openshift-wordpress-developer-quickstart>.

Para facilitar la lectura de RSS del blog se instaló un plugin para poder hacer peticiones y que la respuesta fuera en JSON en vez de xml. Este plugin y toda la información para su instalación y documentación para acceder a cada uno de los end points está disponible en la siguiente dirección:

<https://es.wordpress.org/plugins/json-api/>.

OpenShift nos provee de todo el código que simplemente había que subirlo al proyecto PHP y MySQL de OpenShift.

Finalmente una vez creado el Blog se seleccionó una plantilla acorde con el proyecto y se crearon unos post de ejemplo para poder testear la aplicación.

El blog está disponible a través de la dirección:

<http://blog-bttalcoi.rhcloud.com/>



Alcoi BTT

Rutas Mountain Bike Alcoi

Alcoi BTT App



Nuevas rutas a partir de Agosto

July 28, 2016

Atentos a las nuevas rutas que publicaremos a partir de Agosto.

Vamos a mejorar los recorridos disponibles con más imágenes y zonas por las que pasar. Además añadiremos nuevas rutas por sitios que el Ayuntamiento a mejorado para el paso de las bicicletas.

Figura 9 Vista del Blog.

Conclusiones

El grado de cumplimiento de los objetivos que nos fijamos ha sido total.

Estaría muy bien desarrollar una aplicación para IOS que consumiera los mismos Servicios, y así, alcanzar mayor cuota de mercado.

Estaría muy bien poder mostrar elementos multimedia en la ruta (Fotos, notas, sonidos...).

Anexos

Listado de fuentes entregadas / Código fuente en GitHub

<https://bitbucket.org/brunoescalona/bttserviciosrest>

https://poliformat.upv.es/access/content/attachment/CFP_496_11730/Assignments/5560d9a6-c46b-4cb6-95fd-746f9e29dcb4/AlcoiBTT.rar

Manual de usuario

Aplicación nativa Android

La aplicación se muestra con la siguiente ventana:



Figura 10 Vista principal de la aplicación.

Pulsando sobre el icono superior izquierdo o arrastrando desde la izquierda a la derecha se muestra al menú de la aplicación, con las funcionalidades objetivo de la misma.

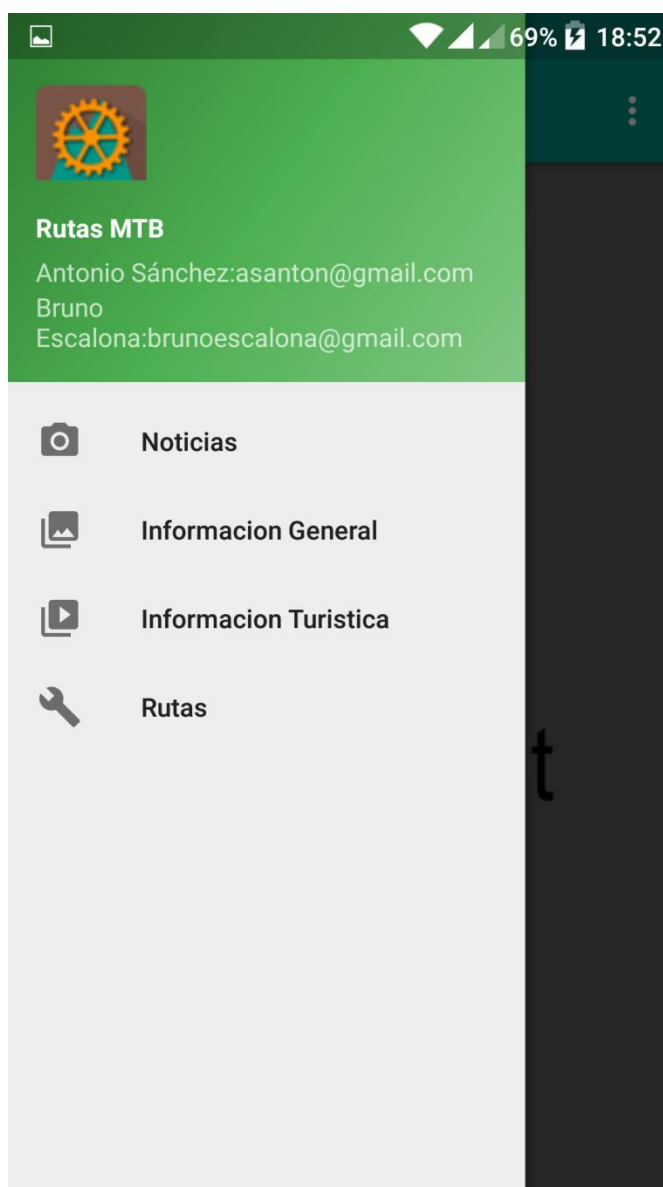


Figura 11 Menú lateral de la aplicación.

Si seleccionamos la 1ª funcionalidad "Noticias" se realiza la carga de noticias RSS, mostrándose una ventana de progreso, tras la cual aparece una lista de noticias descargadas del servidor.

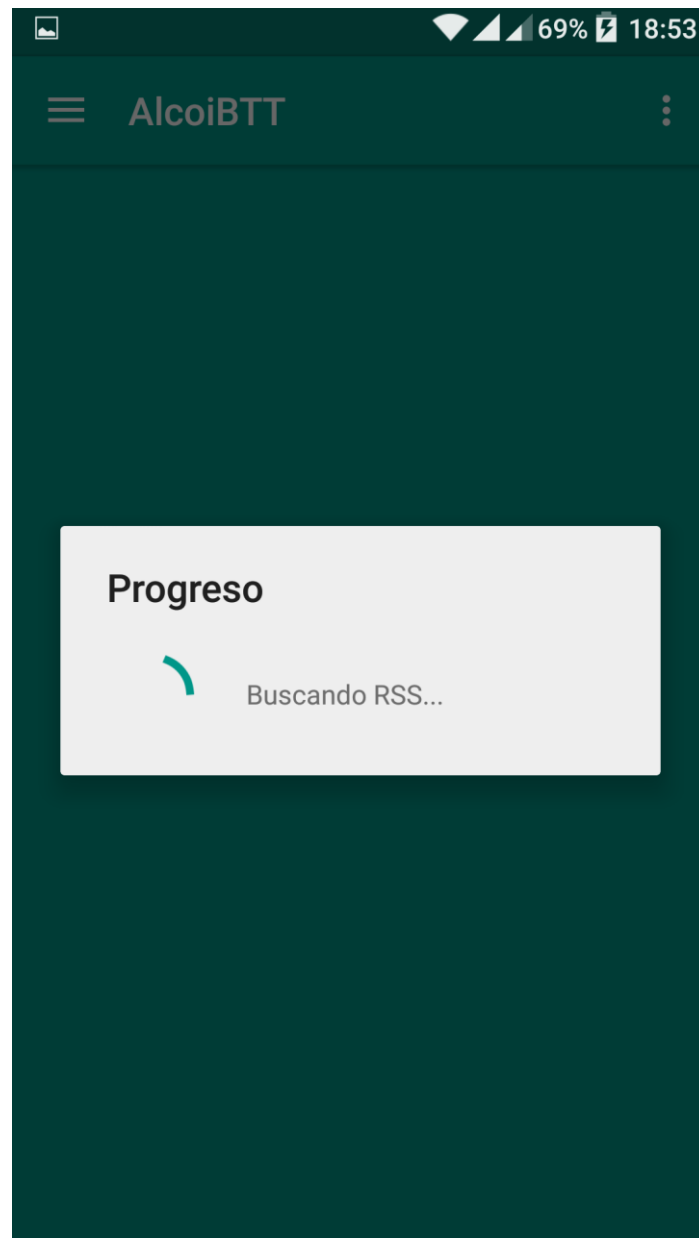


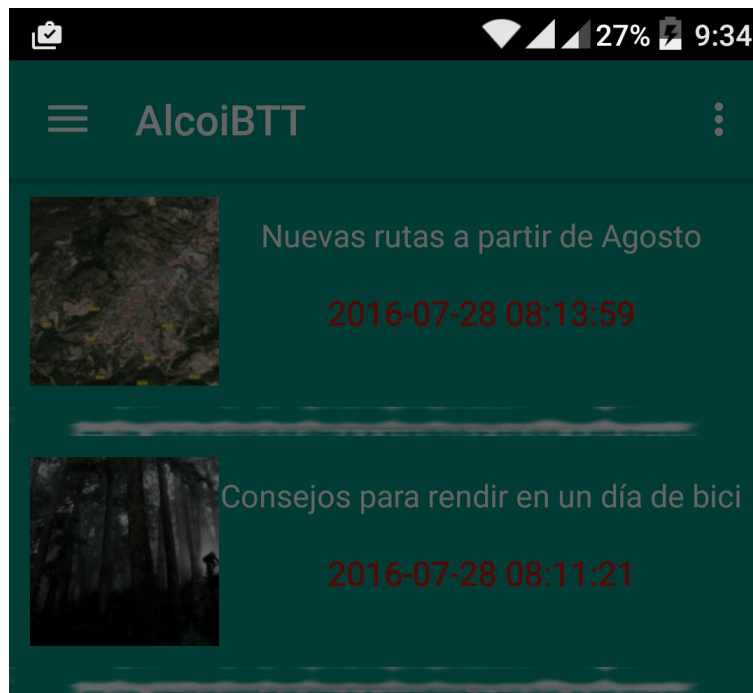
Figura 12 Pantalla de carga de RSS.



Figura 13 Vista de las noticias RSS del blog.

Si el usuario quiere ver más detalle sobre cualquiera de las noticias, es suficiente con pulsar sobre ella.

Se abre en el navegador web directamente la noticia como tal.



Completar acción utilizando



Chrome



Drive



Google+

Servicios de Google Play



Google+

SOLO UNA VEZ SIEMPRE

Figura 14 Abrir noticia RSS para más detalles.

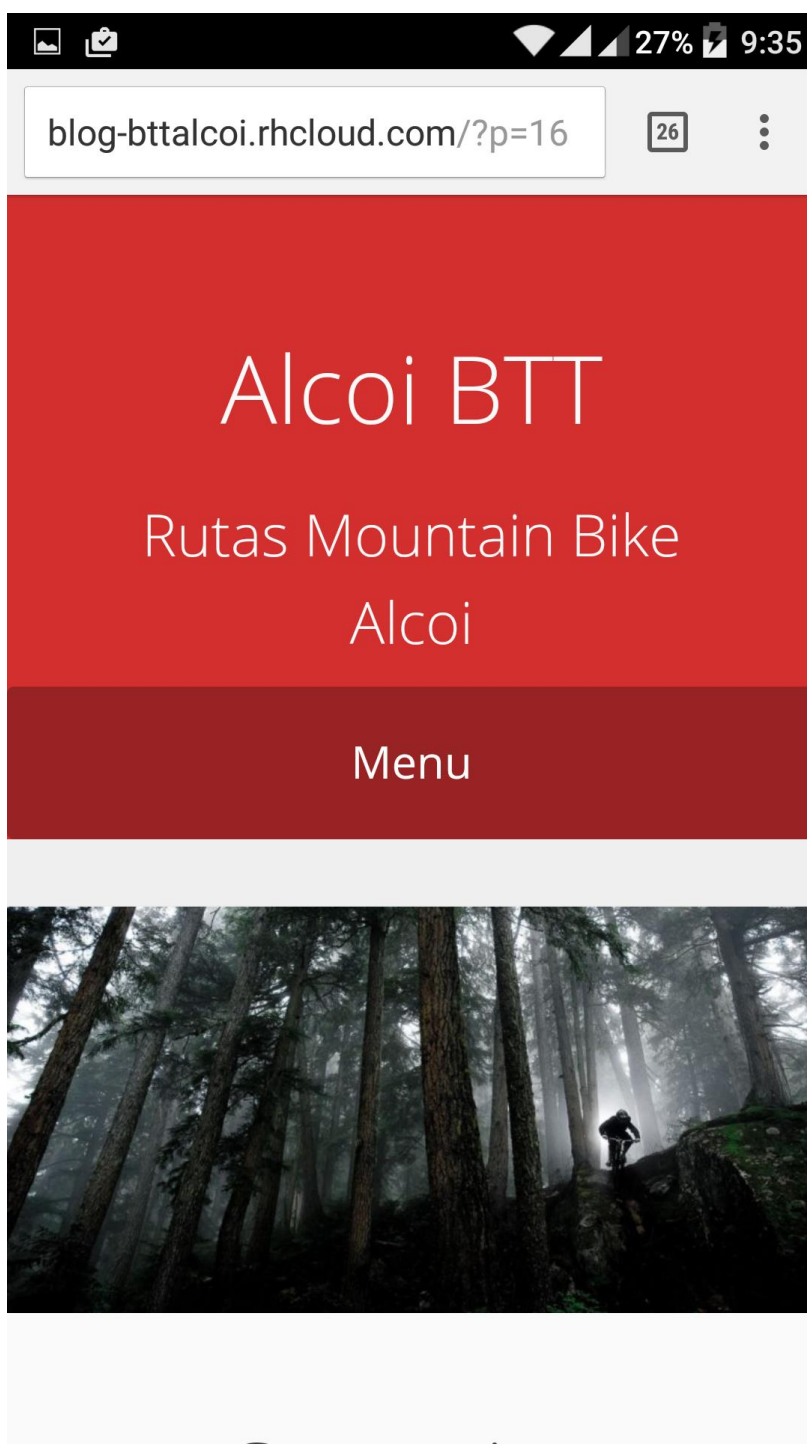


Figura 15 Vista del blog.

Si seleccionamos la 2ª funcionalidad "Información General " se realiza la información General de Alcoi, mostrándose una ventana de progreso, tras la cual aparece una lista elementos de información general descargados del servidor.

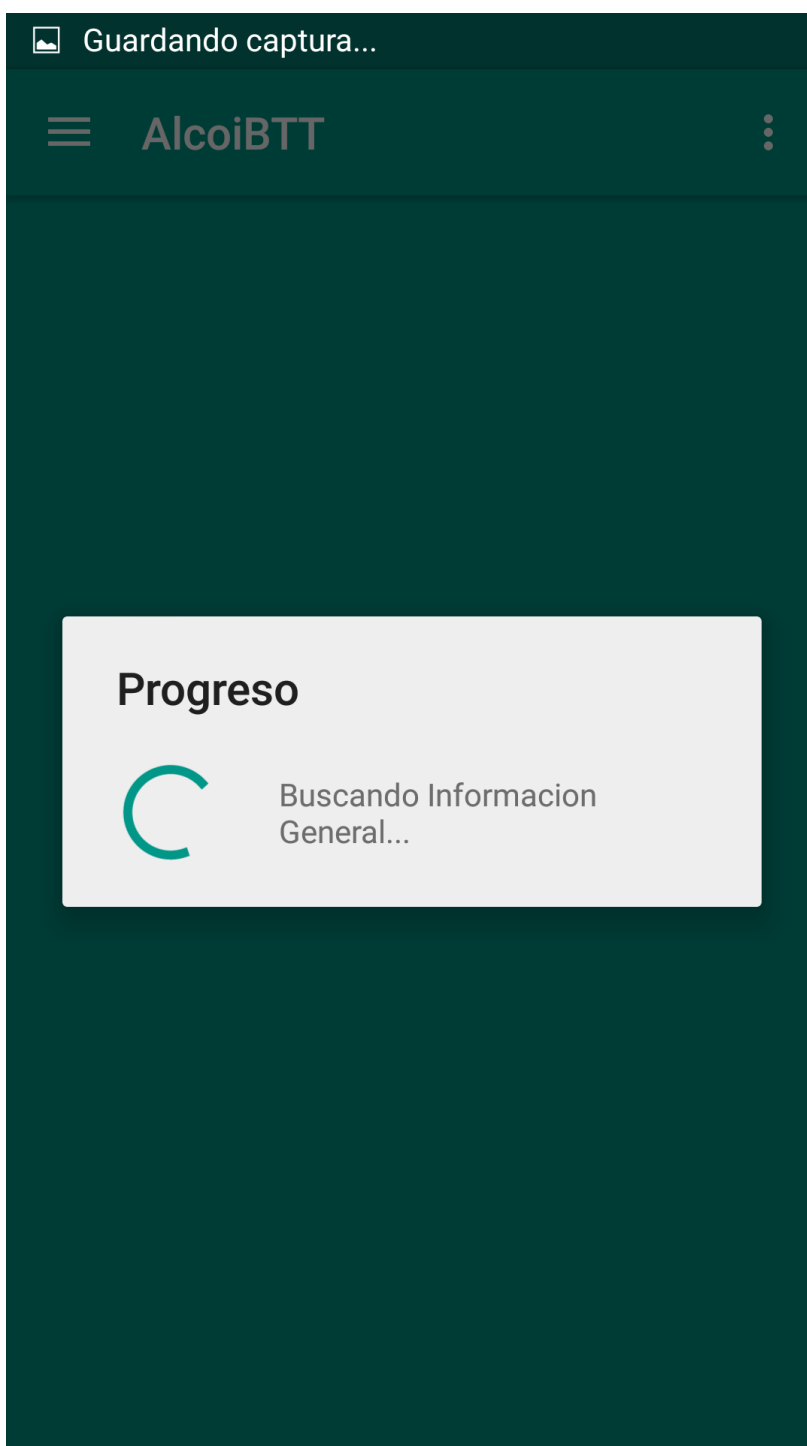


Figura 16 Pantalla de carga de la información.



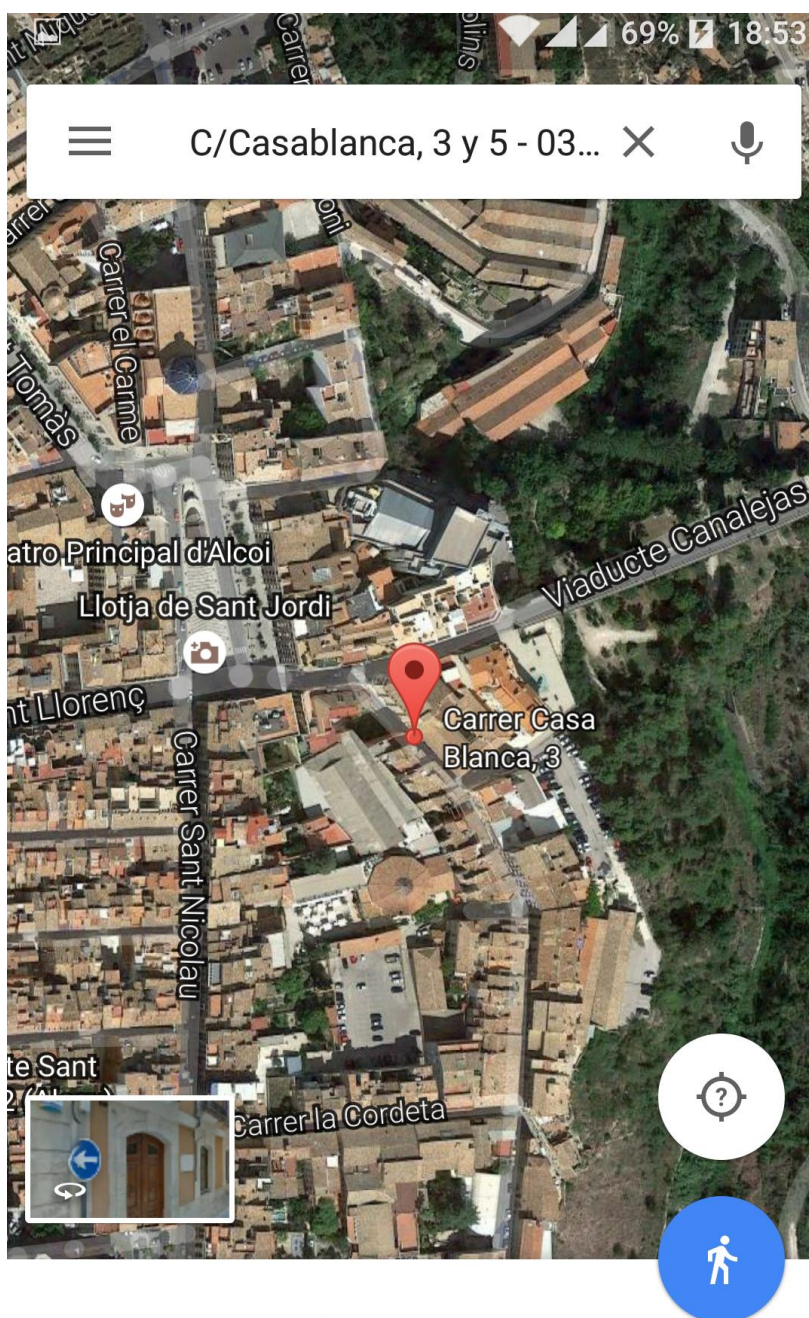
Figura 17 Pantalla de información general.

Sobre esta lista tenemos la opción de navegar, para lo cual es necesario pulsar sobre la dirección del lugar al que queremos navegar.



Google

Figura 18 Carga de Google Play para comprobar localización.



Carrer Casa Blanca, 3

03801 Alcoi, Alicante

Figura 19 Vista de localización del elemento de información seleccionado.



También tenemos la opción de llamar por teléfono, para lo cual es necesario pulsar sobre el teléfono del lugar al que queremos llamar.

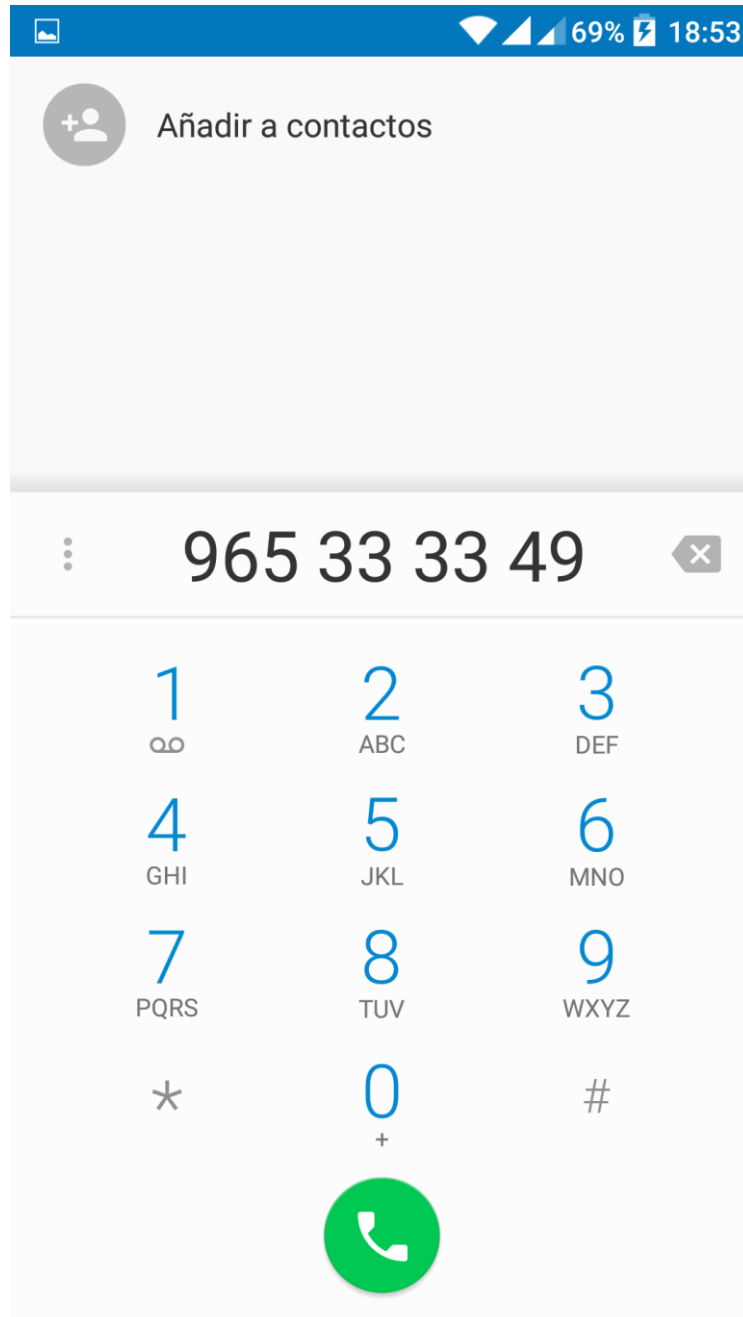


Figura 20 Pantalla de teléfono.

Si seleccionamos la 3ª funcionalidad "Información Turística" se carga la Web de turismo de Alcoi, mostrándose una ventana de progreso.

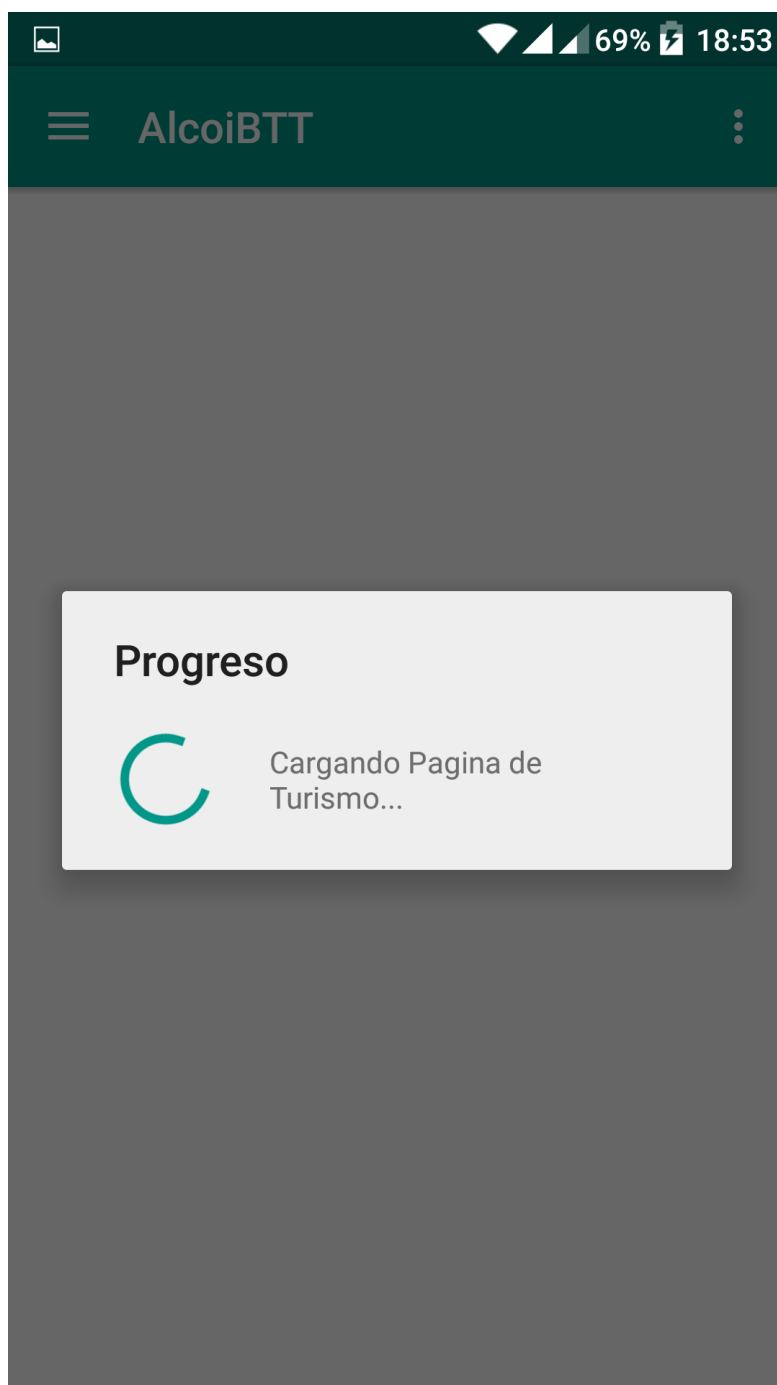
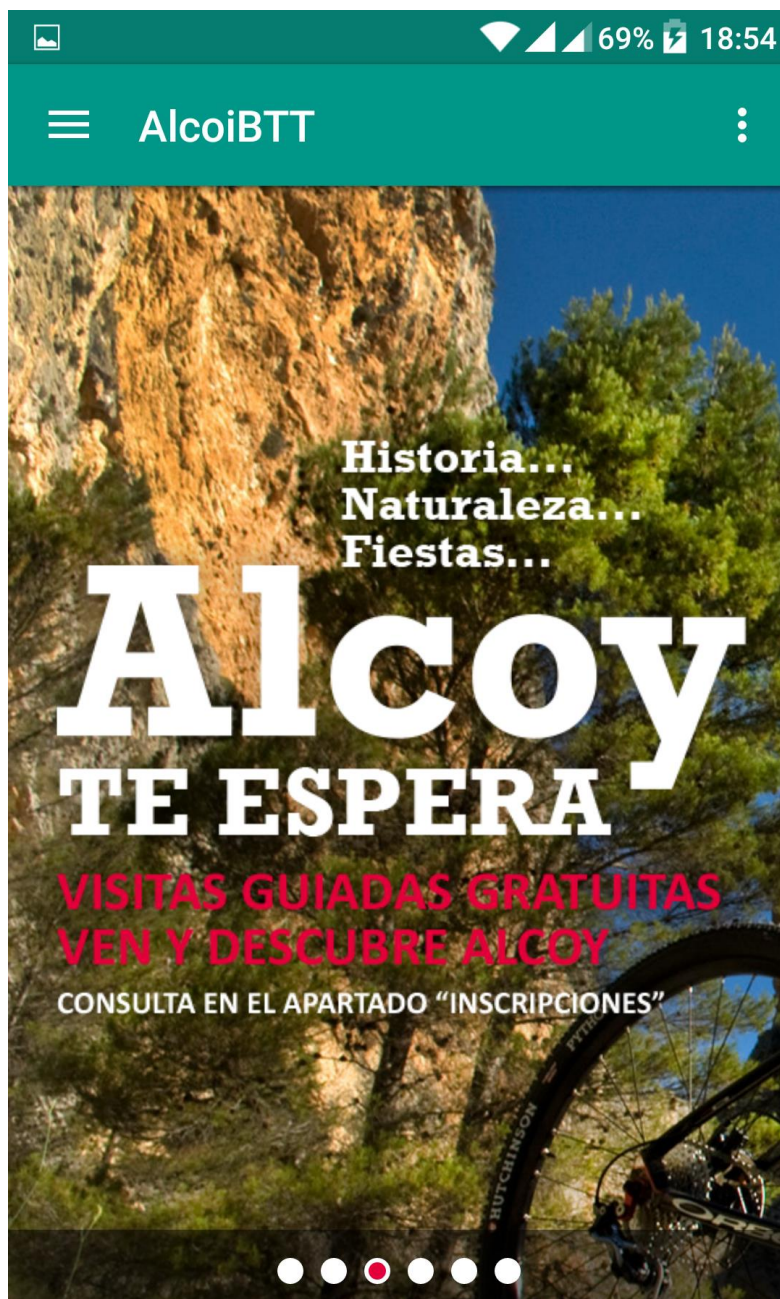


Figura 21 Vista de carga de la página de turismo.



∨ **MÁS INFORMACIÓN** ∨

Figura 22 Vista de página de turismo de Alcoi.

Si seleccionamos la 4ª funcionalidad "Rutas" se cargan las rutas que existan en la zona, tanto en un mapa representadas por colores en función de su dificultad (Verde = Sencilla, Azul=Media, Rojo=Difícil, Negro=Muy Difícil), como en una lista para facilitar su selección.

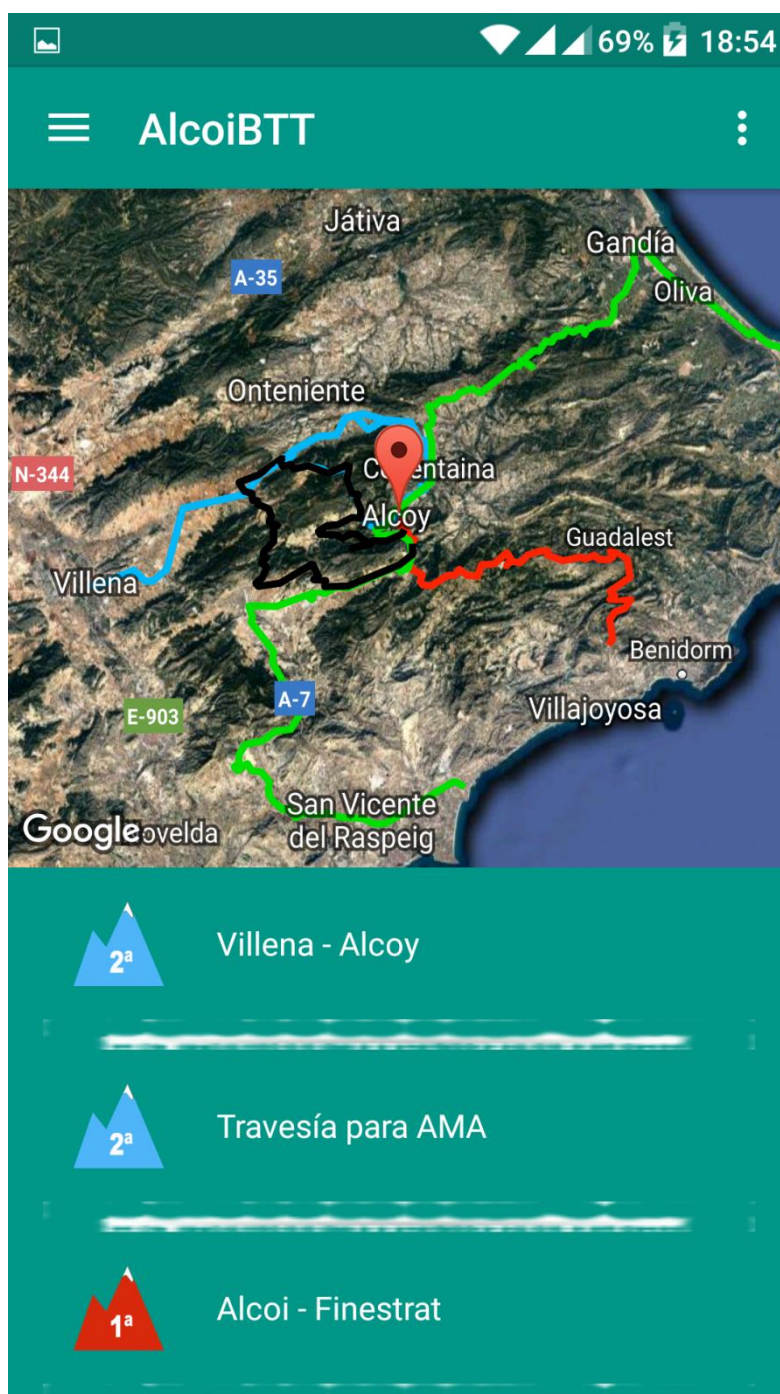


Figura 23 Vista de las rutas disponibles.

Si se quiere ver información referente a una ruta en concreto, se selecciona de la lista, apareciendo dicha ruta resaltada en el mapa, así como información referente a la misma (Distancia, si es o no es circular, e información que la describe).

Además, se da la opción de seguir dicha ruta, para lo cual es necesario disponer de una aplicación de seguimiento de tracks GPX, así como de permisos de escritura en SD.

De no disponer de permisos de escritura en SD no se podrá seguir la ruta.



De no disponer de ninguna aplicación de seguimiento de tracks, la aplicación, previa autorización del usuario, nos lleva a instalar OruxMaps (<https://play.google.com/store/apps/details?id=com.orux.oruxmaps&hl=es>).

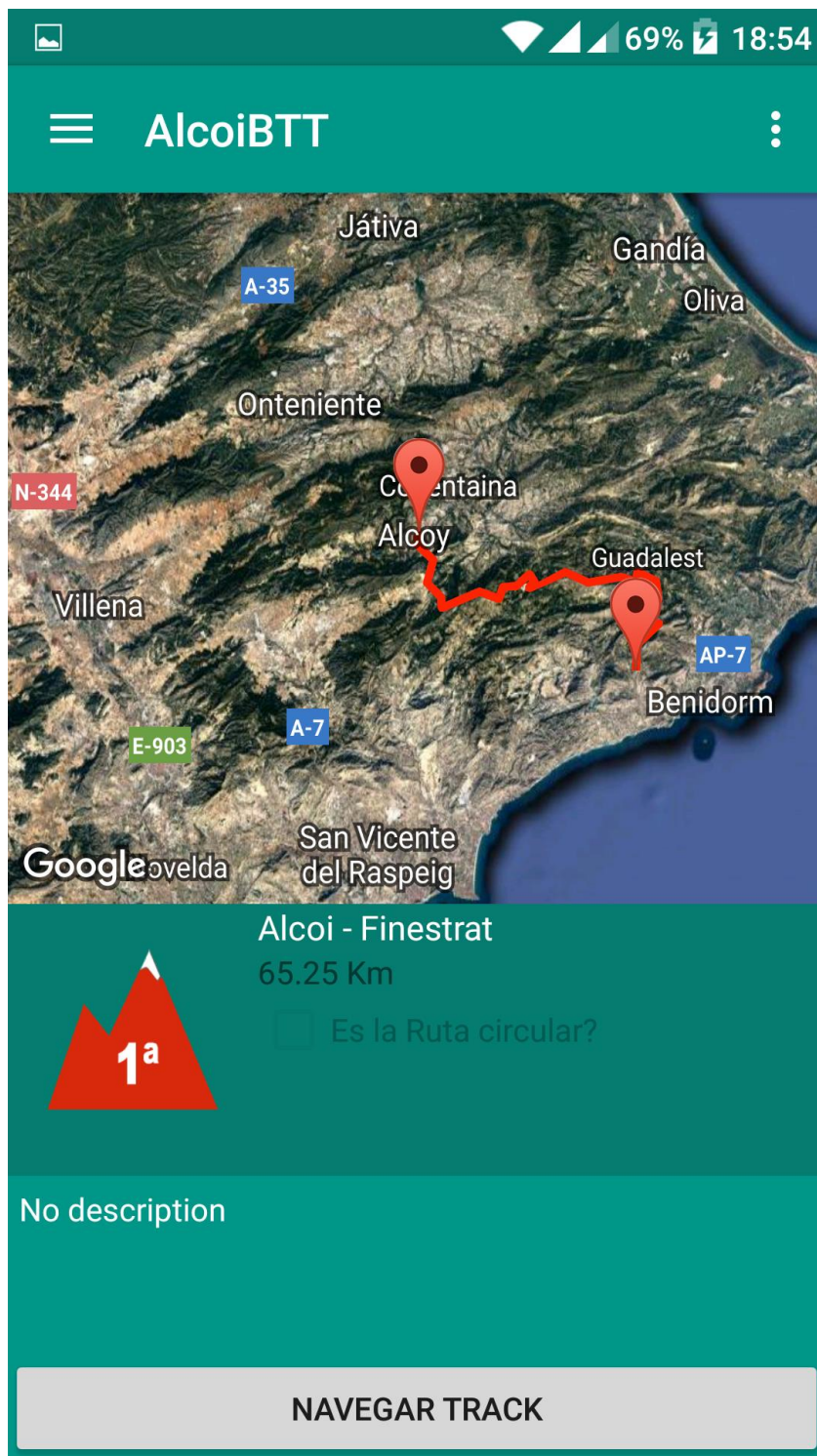


Figura 24 Vista de detalle de una ruta.

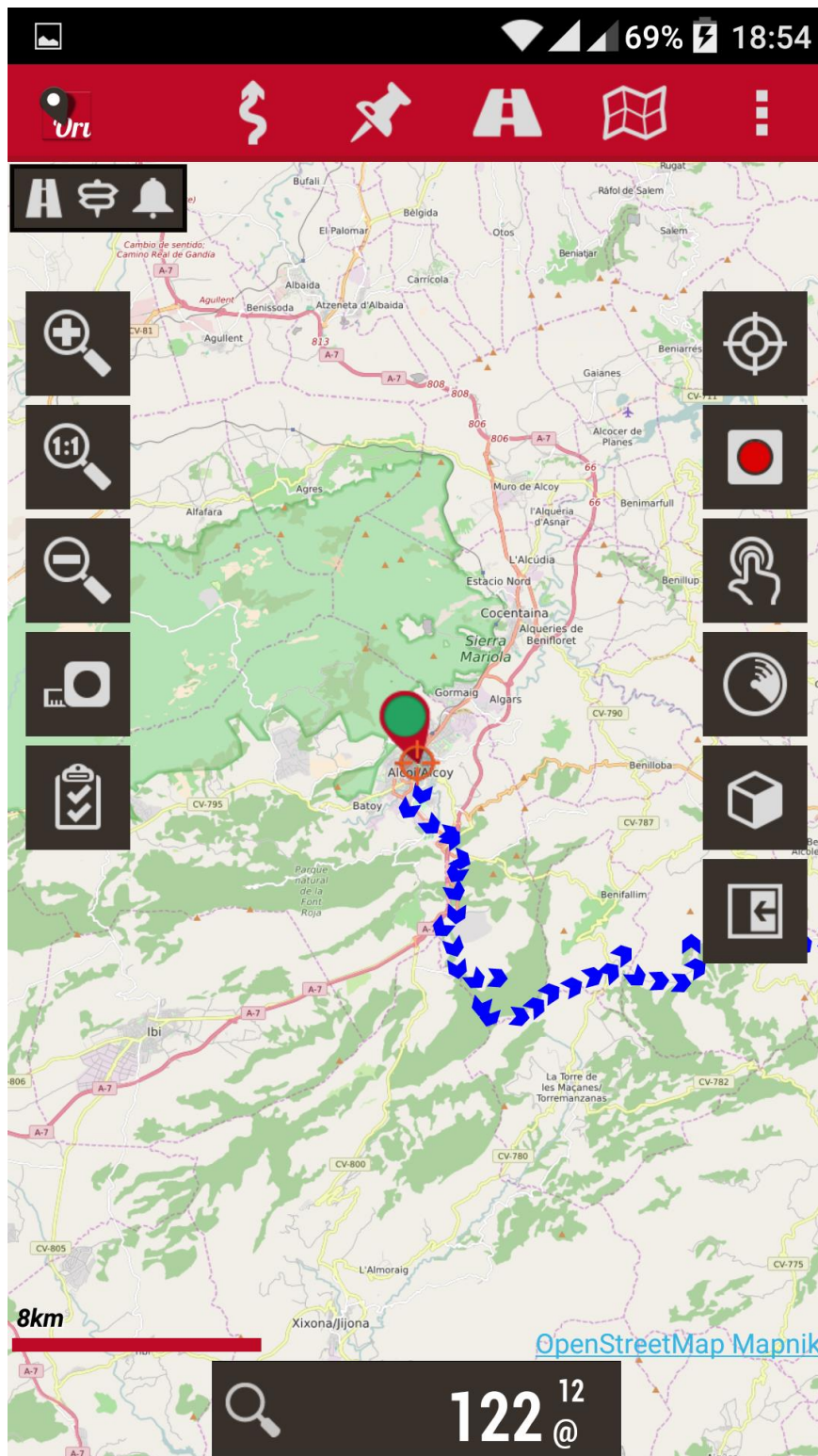


Figura 25 Vista de OruxMaps de la ruta cargada con el gpx.