



# **Conoce Alcalá**

## **Memoria Trabajo Fin de Carrera**

José Antonio Mármol León

Ingeniería Técnica en Informática de Gestión

Consultor: Helena Boltà Torrell

Curso: 2013-2014

# Índice

1. Introducción.....	3
1.1. Términos y abreviaturas .....	3
1.2. Descripción del proyecto .....	5
1.3. Objetivos del proyecto .....	5
1.4. Calendario del proyecto.....	6
1.5. Dificultades encontradas en el desarrollo del proyecto.....	7
2. Diseño centrado en el usuario .....	8
2.1. Usuarios y contexto de uso .....	8
2.2. Diseño contextual.....	13
2.2.1 . Escenarios de uso .....	13
2.3. Prototipo.....	17
2.3.1 Sketches.....	17
2.3.2 Prototipos horizontal .....	20
2.4. Desarrollo pendiente a implementar.....	25
2.5. Modificación del diseño inicial .....	25
2.6. Posibles mejoras .....	25
3. Implementación.....	26
3.1. Estructura del proyecto.....	26
3.2. SQLITE .....	34
3.3. Pruebas.....	34
3.4. Capturas pantallas .....	37
4. Conclusión .....	41
5. Bibliografía .....	42

# 1. Introducción

Este documento se elabora como resultado del proyecto de Desarrollo en Android nativo realizado por José Antonio Mármol León como Trabajo Final de Carrera para la Universitat Oberta de Catalunya.

La aplicación desarrollada es una guía turística del municipio de Sevilla llamado “Alcalá de Guadaíra”.

El objetivo de este documento es plasmar lo que ha sido el desarrollo de una aplicación desde cero pasando por las diferentes fases del ciclo de vida del software.

El contenido del documento se estructura en capítulos de la siguiente manera:

- Introducción, términos y abreviaturas, descripción del proyecto, objetivos del proyecto, calendario y dificultades encontrados en el desarrollo.
- Diseño centrado en el usuario.
- Implementación.
- Conclusión.

## 1.1. Términos y abreviaturas

Abreviatura / Término	Descripción
POI	Punto de interés (del inglés <i>Point Of Interest</i> ), son puntos de localización GPS. Es un punto de ubicación específica que alguien puede encontrar útil o interesante. En español se puede encontrar como PDI.
GPS	Sistema de posicionamiento global por satélite (del inglés <i>Global Positioning System</i> ).
XML	Siglas en inglés de eXtensible Markup Language ('lenguaje de marcas extensible'), es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible.
RSS	Really Simple Syndication, un formato XML para difundir información actualizada frecuentemente a usuarios que se han suscrito a la fuente de contenidos. El formato permite distribuir contenidos sin necesidad de un navegador, utilizando un software diseñado para leer estos contenidos RSS
HTTP	<i>Hypertext Transfer Protocol</i> (protocolo de transferencia de hipertexto) es el protocolo utilizado en una transacción del a World Wide Web (www)
URL	<i>Uniform Resource Locator</i> (localizador de recursos uniforme). Es una secuencia de caracteres estandarizado que se usa para nombrar recursos en Internet para su localización. Dirección de Internet.

<b>API</b>	Interfaz de programación de aplicaciones (IPA) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas.
<b>ACTIVITY</b>	Componente de aplicación que proporciona una pantalla con la que los usuarios pueden interactuar con el fin de hacer algo, como marcar el teléfono, tomar una foto, envíe un correo electrónico, o ver un mapa. Cada actividad se le da una ventana en la que dibujar su interfaz de usuario.
<b>GOOGLE MAPS API</b>	Api de mapas perteneciente a Google. Ofrece imágenes de mapas desplazables, así como fotografías por satélite del mundo e incluso la ruta entre diferentes ubicaciones o imágenes a pie de calle Google Street View.
<b>SQLITE</b>	Sistema de gestión de bases de datos relacional que permite ser embebida (o incluida) en los propios programas.
<b>BEAN</b>	Componente software que tiene la particularidad de ser reutilizable ahorrando tiempo en el desarrollo. Un Bean puede representar desde un botón, un grid de resultados, un panel contenedor o un simple campo de texto, hasta otras soluciones mucho más complejas como conexiones a bases de datos, etc.
<b>IMPLEMENTAR</b>	Termino empleado en programación orientada a objetos para indicar que una clase (por ejemplo clase A) requiere tener una funcionalidad o funcionalidades concretas las cuales están presentes en otra clase (por ejemplo clase B). De esta forma si la clase A requiere de unas determinadas funcionalidades tendrá que utilizar (y codificar) los métodos de la clase B.
<b>EXTENDER</b>	Termino empleado en programación orientada a objetos para indicar que una clase
<b>LAYOUT</b>	Plantilla empleada para albergar los componentes (caja de textos, botones, etc) que forman una pantalla en Android.
<b>FRAMEWORK</b>	<p>Estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.</p> <p>Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio, y provee una estructura y una especial metodología de trabajo, la cual extiende o utiliza las aplicaciones del dominio.</p>

## 1.2. Descripción del proyecto

Investigando sobre las aplicaciones en Android y sobre los ámbitos que no están cubiertos o que lo están pero están incompletos se observa que sería posible y factible la realización de una guía turística.

El proyecto a desarrollar está orientado a la geolocalización de los monumentos y puntos de interés ubicados en la localidad sevillana de “Alcalá de Guadaíra”.

En la fase de análisis del proyecto se estudió y barajó la posibilidad de emplear un frameworks para realización de la aplicación. Más concretamente se pensó en usar [PhoneGap](#).

Las principales ventajas de usar phonegap son:

- Permite realizar aplicaciones multiplataforma: Permite exportar una aplicación a varios sistemas operativos (Android, IOs, BlackBerry, etc.).
- Independencia sobre el lenguaje nativo de las aplicaciones: No requiere aprender el lenguaje sobre el que están desarrollados las aplicaciones nativas.
- El desarrollo está basado en diseño web lo cual le otorga la flexibilidad en el diseño de pantallas.

No obstante Phonegap dispone de una serie de desventajas que al ser sopesadas han declinado su uso en favor del desarrollo en nativo.

- No explota todas las opciones disponibles de un dispositivo: Tiene limitaciones en cuanto al uso de sensores y sobre el uso de las configuraciones del dispositivo. Esto sería un problema de cara a posibles ampliaciones de funcionalidades de la aplicación.
- No es independiente del todo de la plataforma: Esto obliga a codificar ciertos aspectos en el lenguaje nativo del dispositivo. Un ejemplo se encuentra en el uso de SQLite.
- Al emplear Html5 para el diseño se producen problemas con los navegadores ya que Html5 no está estandarizado

En conclusión, se ha optado por el desarrollo en Android nativo para así tener mayor respaldo de cara a futuras actualizaciones.

## 1.3. Objetivos del proyecto

El objetivo principal del TFC es la realización desde cero de una aplicación nativa en Android basándonos en el diseño centrado en el usuario.

En todo proyecto, cuando se decide el empleo de un lenguaje y/o tecnología sin conocimientos previos de los mismos hay un periodo de investigación y análisis.

El motivo por el cual se ha escogido una guía turística es debido a la multitud de opciones y funcionalidades que suelen tener este tipo de aplicaciones, permitiendo abarcar más aspectos del API de Android. Otro factor decisivo a la hora de escoger la

realización de una guía turística ha sido por pertenecer al municipio y querer así potenciar el turismo en el mismo.

Los requisitos detectados son:

#### Requisitos funcionales

- Localizar enclaves.
- Rutas turísticas.
- Puntos de interés (POIs).
- Eventos
- Brújula
- Tiempo
- Conócenos
- Teléfonos interés
- Videos
- Noticias
- Multidioma
- Radio Online

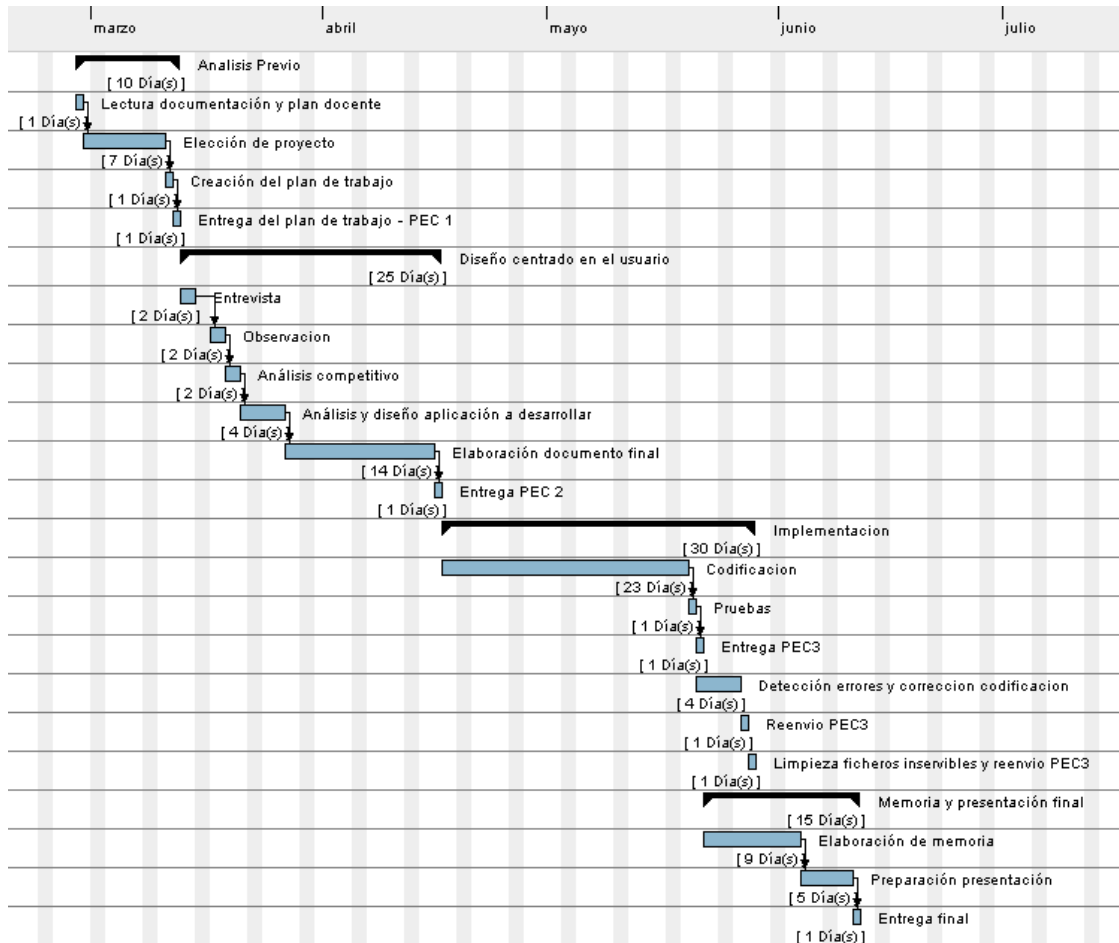
#### Requisitos no funcionales

- Interfaz amigable.
- Facilidad en el manejo de las opciones
- Pocos pasos a memorizar en cualquiera de las opciones.
- Respuesta visible ante acción del usuario (empleo de imagen de progreso).

En definitiva se ha intentado crear una aplicación extensa y que se asemejase a una guía turística profesional con el inconveniente del poco tiempo de desarrollo que se ha tenido.

## 1.4. Calendario del proyecto

Nombre	Fecha Inicio	Fecha Fin	Duración (días)	Duración (horas)
<b>Análisis Previo</b>	<b>27/02/14</b>	<b>12/03/14</b>	<b>10</b>	<b>240</b>
Lectura documentación y plan docente	27/02/14	27/02/14	1	24
Elección de proyecto	28/02/14	10/03/14	7	168
Creación del plan de trabajo	11/03/14	11/03/14	1	24
Entrega del plan de trabajo – PEC 1	12/03/14	12/03/14	1	24
<b>Diseño centrado en el usuario</b>	<b>13/03/14</b>	<b>16/04/14</b>	<b>25</b>	<b>600</b>
Entrevista	13/03/14	14/03/14	2	48
Observación	17/03/14	18/03/14	2	48
Análisis competitivo	19/03/14	20/03/14	2	48
Análisis y diseño aplicación a desarrollar	21/03/14	26/03/14	4	96
Elaboración documento final	27/03/14	15/04/14	14	336
Entrega PEC 2	16/04/14	16/04/14	1	24
<b>Implementación</b>	<b>17/04/14</b>	<b>28/05/14</b>	<b>30</b>	<b>720</b>
Codificación	17/04/14	19/05/14	23	552
Pruebas	20/05/14	20/05/14	1	24
Entrega PEC3	21/05/14	21/05/14	1	24
Detección errores y corrección codificación	21/05/14	26/05/14	4	96
Reenvió PEC3	27/05/14	27/05/14	1	24
Limpieza ficheros inservibles y reenvío PEC3	28/05/14	28/05/14	1	24
<b>Memoria y presentación final</b>	<b>22/05/14</b>	<b>11/06/14</b>	<b>15</b>	<b>360</b>
Elaboración de memoria	22/05/14	3/06/14	9	216
Preparación presentación	4/06/14	10/06/14	5	120
Entrega final	11/06/14	11/06/14	1	24
<b>TOTAL</b>			<b>161</b>	<b>3864</b>



**Observaciones:** Debido a errores detectados en la fase de implementación se ha producido un desajuste en la planificación de la entrega de la PEC3, la cual tendría que haber sido para el 21 de Mayo y se ha producido el 29 de Mayo.

## 1.5. Dificultades encontradas en el desarrollo del proyecto.

El principal problema ha sido la comprensión de la API de Android y en especial del empleo de Google Maps. Al no disponer de conocimientos sobre dicho API el poder realizar cualquier funcionalidad ha requerido un sobreesfuerzo en la implementación.

Indicar que aprender la comunicación entre los activitys ha sido la parte más compleja sobre todo en la opción de los puntos de interés, ya que se ha utilizado un único activity para diferentes pantallas, por tanto las redirecciones ha resultado complicado.

Otro obstáculo ha sido conseguir usar la radio en segundo plano, para ello ha sido necesario emplear servicios.

La maquetación de los componentes de los layouts también ha supuesto una dificultad.

La lectura de XML y RSS ha requerido estudiar como ejecutar procesos asíncronos.

El empleo de SQLITE también ha supuesto una dificultad por el desconocimiento de dicho sistema gestor de base de datos, así como su uso con Android.

Para terminar, dado que no se dispone de ningún XML ni RSS para los eventos se ha tenido que extraer los eventos por medio del propio HTML, para ello ha sido necesario leer línea a línea el código HTML de la página web de manera dinámica y obtener los datos.

## 2. Diseño centrado en el usuario

A continuación se describe el diseño centrado en el usuario final tras el desarrollo.

### 2.1. Usuarios y contexto de uso

Para la realización de la aplicación se escogió como métodos de indagación los siguientes:

- **Entrevistas**

Con el fin de conocer la opinión tanto de las personas del municipio como de aquellas personas que no lo son se optó por realizar entrevistas a personas de diferentes edades y de diferentes municipios. Esto proporcionaría una visión más amplia y neutral. A los entrevistados se les preguntó:

Personas que no son del municipio pero que han estado

- ¿Qué les motivó a conocer el municipio?
- ¿Qué visitaron?, ¿cómo se informaron de la existencia de lo visitado?
- En el tiempo que estuvieron visitando el municipio ¿Utilizaron algún tipo de guía?
- ¿Han realizado alguna ruta turística?, ¿qué les pareció las rutas turísticas?, ¿Estaban bien señalizadas?
- ¿Han estado en algún punto de información?
- Cuando han pasado por algún enclave ¿han observado carteles informativos?
- ¿Qué echaron en falta sobre la información del municipio?
- ¿Han estado en alguno de las fiestas del municipio?

Personas que son del municipio

- ¿Qué le gusta del municipio?
- ¿Alguna vez ha sido guía del municipio?, ¿qué ha enseñado del municipio?
- ¿Conoce los eventos organizados del municipio?, ¿y las noticias?, ¿cómo se informa de ello?

A todos se les formuló las siguientes preguntas:

- ¿Cree que la información que se muestra de los enclaves es suficiente y precisa?
- Cuando realiza un viaje ¿en qué se fija, cómo se organiza?
- Si tuviesen que utilizar una aplicación móvil como guía turística del municipio, ¿qué les gustaría que tuviese dicha aplicación?
- ¿Utilizan el móvil con frecuencia?
- ¿Qué móvil tiene?
- ¿Utilizarían una aplicación móvil como guía turística?



- **Observación.**

- Planteamiento

- Se estudió si la información disponible es suficiente y correcta para poder localizar los puntos de interés del municipio.

- Desarrollo

- Se realizó estudio sobre la información de los puntos de interés del municipio. Para ello se visitó a los puntos de interés tanto a pie como en vehículo.

- Resultados

- Se observó una carencia de información en determinadas zonas del municipio lo cual dificulta conocer dónde encontrar los puntos de interés. En determinadas zonas solo se puede acceder a pie lo cual no se indica en ningún cartel. Tampoco se indica la dificultad del terreno.

- Conclusiones obtenidas.

- Se detectó carencias informativas de determinadas zonas.

- **Análisis competitivo**

- Planteamiento

- Dado que ya existe una [aplicación en Android sobre el municipio](#) se decidió estudiarla con el objetivo de conocer su funcionalidad, sus interfaces y sus carencias para así potenciar la nueva aplicación.

- Desarrollo

- Testeo de la aplicación tras su descarga e instalación.

- Resultados

- Su principal carencia es que no hace uso de los dispositivos que incorporan los móviles tales como el acelerómetro o gps ni es multidioma

En cuanto al contenido, está estructurada en:

- Menú principal

- Muestra el Escudo del municipio y un menú en la parte inferior. El menú consta de los botones: Festejos, Sorteo, Mensajes, Guia Alcalá y Más



Figura 1: Pantalla Menú principal

- Festejos**  
 Pantalla referente a los festejos del municipio. El botón muestra el nombre de la fiesta (Carnavales, Semana Santa, Rocio, Feria).

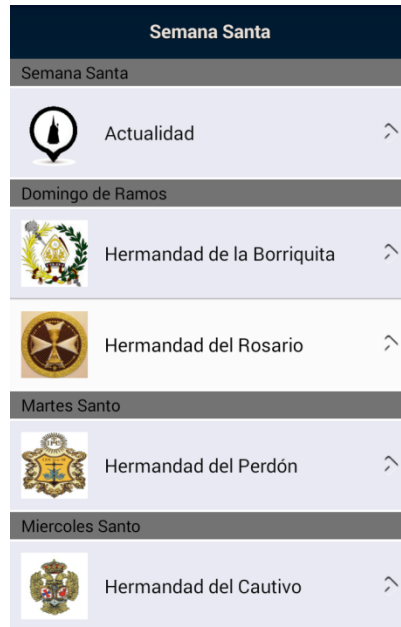


Figura 2: Pantalla Semana Santa

- Mensajes**  
 Pantalla donde se registran notificaciones (calles cortadas por obras, cambio linea de autobuses, etc).

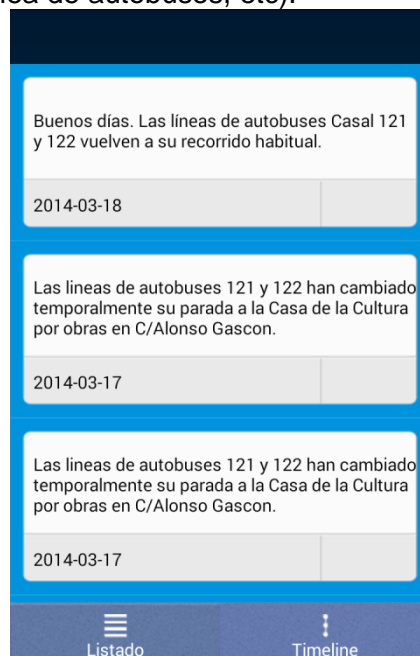


Figura 3: Pantalla Mensajes

- Guía Alcalá**  
 Pantalla donde se visualizan diferentes puntos de interés del municipio a modo de guía.



Figura 4: Pantalla Guía Alcalá

- Más**  
 Pantalla donde se visualizan puntos de interés como farmacias, El tiempo, noticias, eventos, etc.

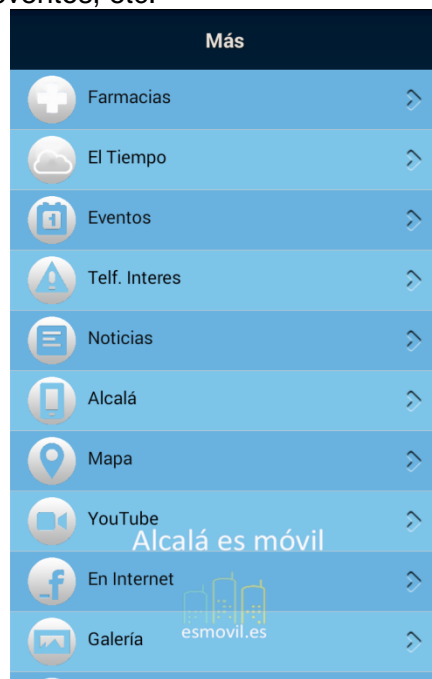


Figura 5: Pantalla Más

Conclusiones obtenidas.

La aplicación estudiada es completa en cuanto a información histórica sobre el municipio se refiere. No obstante, el objetivo de la aplicación a desarrollar se centra en la creación de una guía que permita localizar las zonas de interés y tener información de los mismos. Para ello es necesario emplear la geolocalización que permita conocer en qué punto se encuentra el usuario y que tiene a su alrededor. Además gracias a la geolocalización se podrá emplear los POIs de una manera más eficiente permitiendo al usuario llegar hasta ellos empleando Google Maps.

Como es una guía turística es necesario que la aplicación sea multidioma.

Otro punto a tener en cuenta es la gestión de los POIs, permitiendo la creación de POIs, y grupos.

Basándonos en las entrevistas y en la observación se obtuvieron dos perfiles para la aplicación: Los visitantes y los empadronados.

Perfil: Visitante
<b>Características</b>
Demográficas: Los no empadronados en el municipio, pudiendo ser incluso extranjeros.
Intereses: Conocer los rincones que dispone el municipio
Motivaciones: Orientarse en el municipio, estar informados de los eventos y fiestas que se produzcan, poder localizar puntos de interés
<b>Contexto de uso</b>
Dónde: En cualquier zona del municipio
Cuando: Para orientarse en el municipio, estar informados de los eventos y fiestas que se produzcan, poder localizar puntos de interés.
El uso de la aplicación será de periodos largos
Entorno: En cualquier zona del municipio, tanto en zonas al descubierto (empleando GPS, conexión a Internet) y zonas cerradas (conexión a Internet)

Perfil: Empadronado
<b>Características</b>
Demográficas: Los empadronados en el municipio
Intereses: Poder localizar puntos de interés, conocer las noticias y eventos del municipio.
Motivaciones: Estar informado en todo momento de lo que ocurra en el municipio.
<b>Contexto de uso</b>
Dónde: En cualquier zona del municipio
Cuando: Cuando requieran estar informados
El uso de la aplicación será de periodos cortos
Entorno: En cualquier zona del municipio, tanto en zonas al descubierto (empleando GPS, conexión a Internet) y zonas cerradas (empleando conexión a Internet)

A continuación se describirán los escenarios implementados.

## 2.2. Diseño contextual

### 2.2.1 . Escenarios de uso

#### Localizar enclaves.

Contexto: Usuario que visita el municipio desea conocer los enclaves

- Porqué: Para conocer los rincones del municipio.
- Cuando: a cualquier hora
- Dónde: En zona abiertas o cerradas. Será necesario el uso de GPS, conexión a Internet

#### Rutas turísticas.

Contexto: Usuario que visita el municipio desea realizar una de las rutas turísticas.

- Porqué: Para conocer las rutas turísticas que ofrece el municipio.
- Cuando: A cualquier hora
- Dónde: En zona abiertas o cerradas.

#### Puntos de interés (POIs).

Contexto: Usuario que visita el municipio y que desea conocer los puntos de interés.

- Porqué: Para encontrar un punto de interés
- Cuando: A cualquier hora
- Dónde: En zona abiertas o cerradas. En este caso concreto puede ser una mezcla de zona cerrada y zona abierta para usar Google Maps.

#### Crear Grupos POIs

Contexto: Usuario que desea crear sus propios grupos de puntos de interés.

- Porqué: Para tener sus propios grupos de POIs
- Cuando: En cualquier momento.  
Dónde: En zona abiertas o cerradas. No se requiere GPS ni conexión a Internet

#### Editar Grupos POIs

Contexto: Usuario que desea modificar los datos de los grupos POIs.

- Porqué: Para editar sus propios puntos de interés
- Cuando: En cualquier momento.
- Dónde: En zona abiertas o cerradas. No se requiere GPS ni conexión a Internet

### Eliminar Grupos POIs

Contexto: Usuario que desea eliminar un grupo

- Porqué: Para eliminar un grupo
- Cuando: En cualquier momento.
- Dónde: En zona abiertas o cerradas. No se requiere GPS ni conexión a Internet.

### Buscar Grupos

Contexto: Usuario que desea encontrar un Grupo.

- Porqué: Para editar sus propios grupos
- Cuando: En cualquier momento.
- Dónde: En zona abiertas o cerradas. No se requiere GPS ni conexión a Internet

### Crear POIs

Contexto: Usuario que desea crear sus propios puntos de interés.

- Porqué: Para tener sus propios puntos de interés
- Cuando: En cualquier momento.
- Dónde: En zona abiertas o cerradas. Se requiere GPS o conexión a Internet

### Editar POIs

Contexto: Usuario que desea modificar los datos de un POI.

- Porqué: Para editar sus propios puntos de interés
- Cuando: En cualquier momento.
- Dónde: En zona abiertas o cerradas. Se requiere GPS o conexión a Internet si desea modificar las coordenadas.

### Eliminar POIs

Contexto: Usuario que desea eliminar un POI.

- Porqué: Para editar sus propios puntos de interés
- Cuando: En cualquier momento.
- Dónde: En zona abiertas o cerradas. No se requiere GPS ni conexión a Internet

### Buscar POIs

Contexto: Usuario que desea encontrar un POI.

- Porqué: Para editar sus propios puntos de interés
- Cuando: En cualquier momento.
- Dónde: En zona abiertas o cerradas. No se requiere GPS ni conexión a Internet

### Visualizar eventos

Contexto: Usuario que visita el municipio y desea acudir a un evento

- Porqué: Para acudir a un evento
- Cuando: Cuando el usuario lo requiera
- Dónde: En zona abiertas o cerradas. En este caso concreto puede ser una mezcla de zona cerrada y zona abierta para usar Google Maps.

### Brújula

Contexto: Usuario que requiere de brújula para orientarse.

- Porqué: Para orientarse en zonas sin información
- Cuando: En cualquier momento.
- Dónde: En zona abiertas. No será necesario la geolocalización.

### Visualizar Tiempo

Contexto: Usuario que desea conocer el tiempo que hará en el municipio.

- Porqué: Para conocer el tiempo que hará en el municipio
- Cuando: En cualquier momento.
- Dónde: En zona abiertas o cerradas. Será necesario uso de conexión a Internet

### Conócenos

Contexto: Usuario que desea conocer datos de interés demográficos y cultural del municipio

- Porqué: Para informarse del municipio.
- Cuando: En cualquier momento.
- Dónde: En zona abiertas o cerradas. No hará falta conexión a internet.

### Teléfonos interés

Contexto: Usuario que desea llamar a uno de los teléfonos de interés.

- Porqué: Por algún tema el particular
- Cuando: En cualquier momento.
- Dónde: En zona abiertas o cerradas. No hará falta conexión a internet.

### Videos

Contexto: Usuario que desea visualizar los videos subidos a Youtube del municipio.

- Porqué: Para visualizar los videos subidos a Youtube del municipio.
- Cuando: En cualquier momento.
- Dónde: En zona abiertas o cerradas. Se requiere conexión a Internet

### Noticias

Contexto: Usuario que desea leer las noticias del municipio

- Porqué: Para estar informado de lo sucedido en el municipio.
- Cuando: En cualquier momento.
- Dónde: En zona abiertas o cerradas. Se requiere conexión a Internet.

### Radio Online

Contexto: Usuario que desea escuchar la radio del municipio

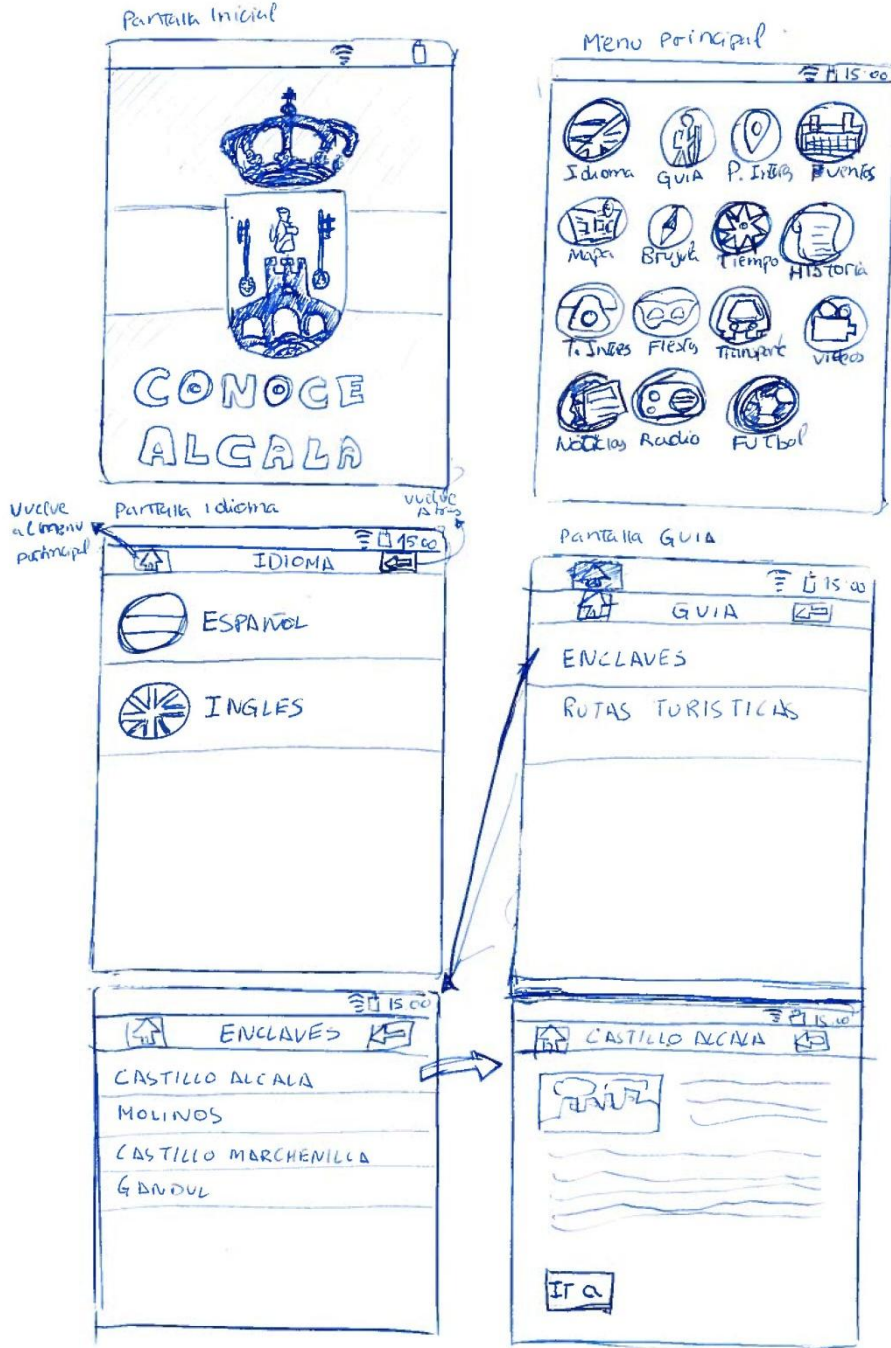
- Porqué: Por estar fuera del municipio o porque la señal analógica no es de calidad.
- Cuando: En cualquier momento.
- Dónde: En zona abiertas o cerradas. Se requiere conexión a Internet

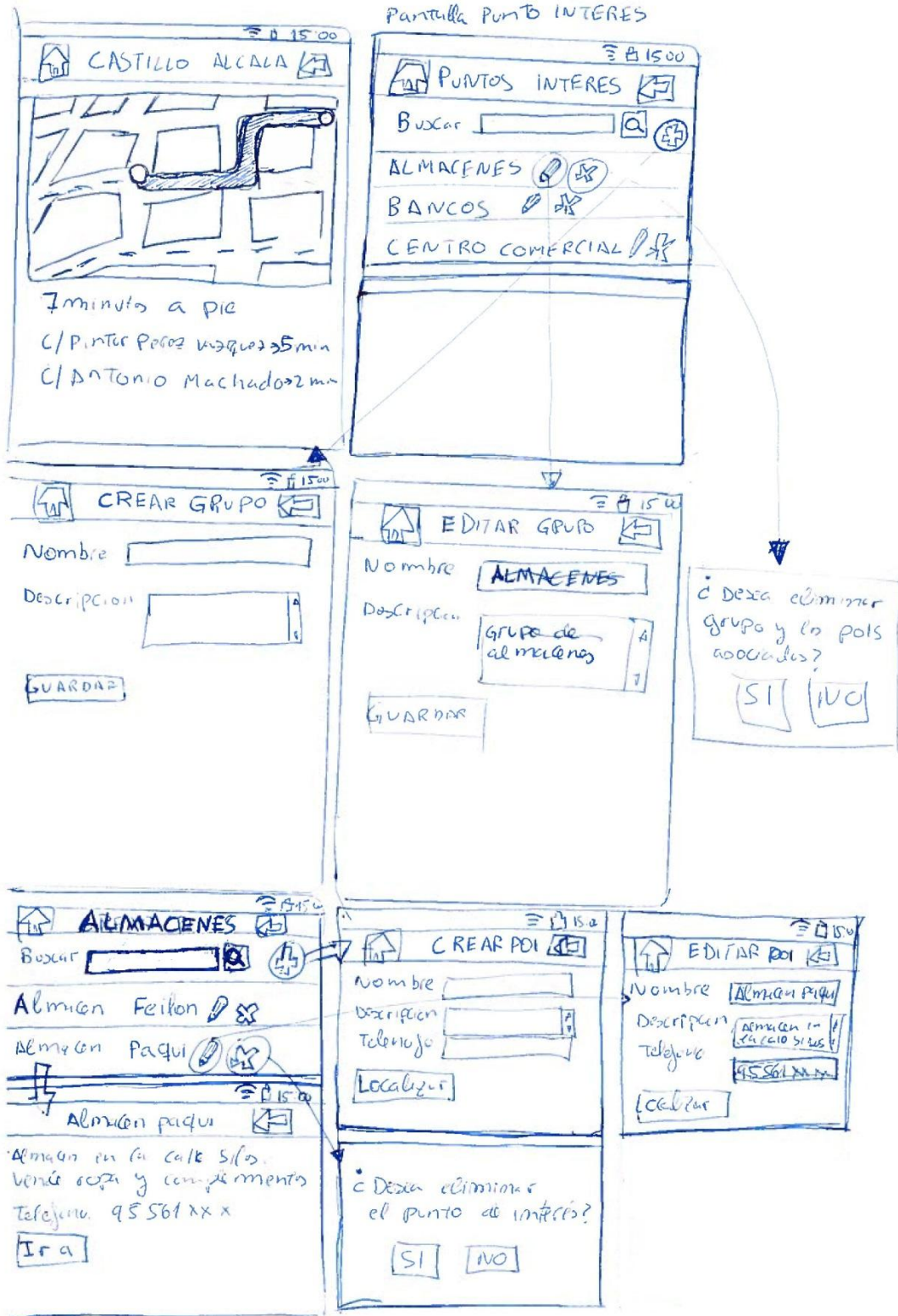


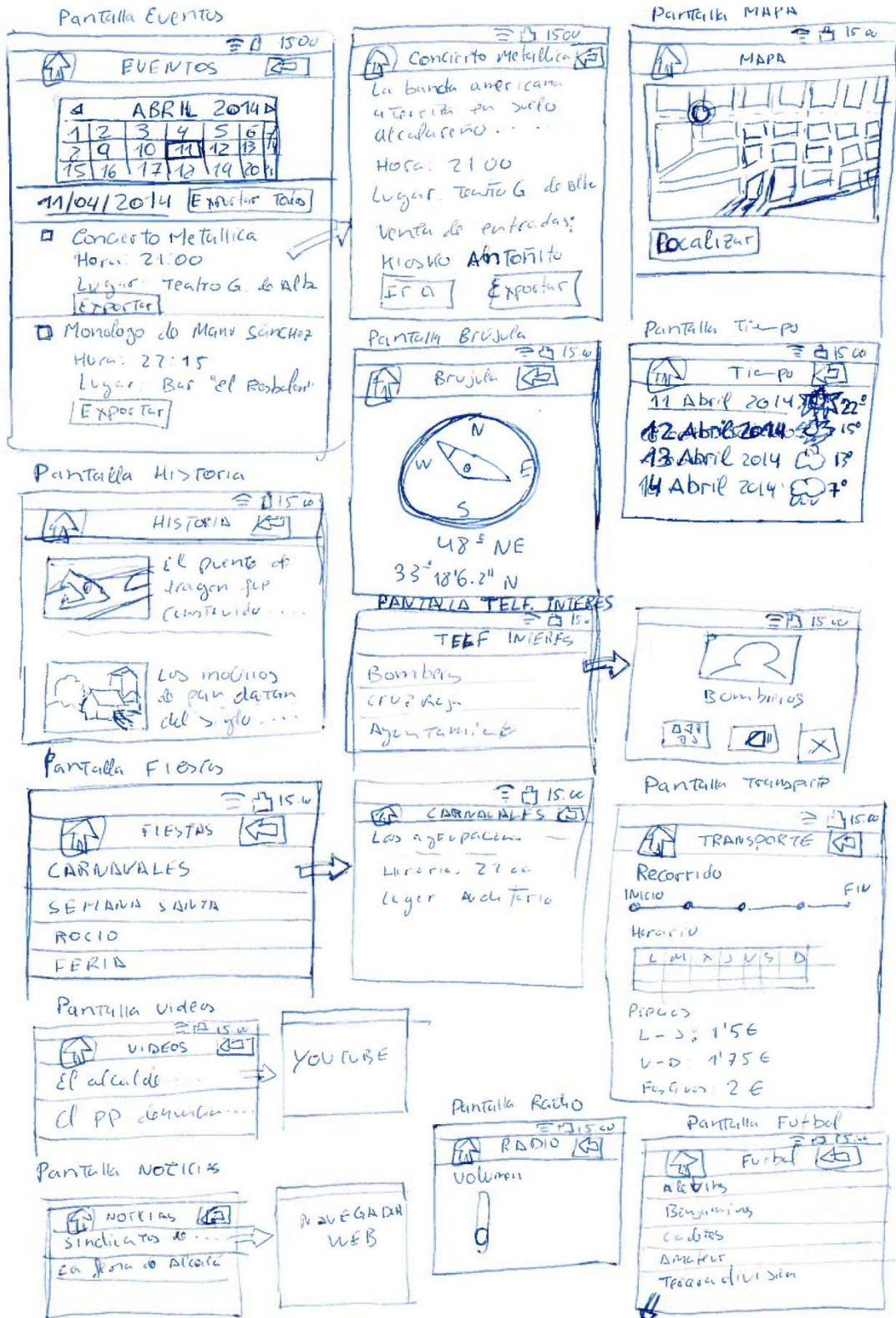
## 2.3. Prototipo

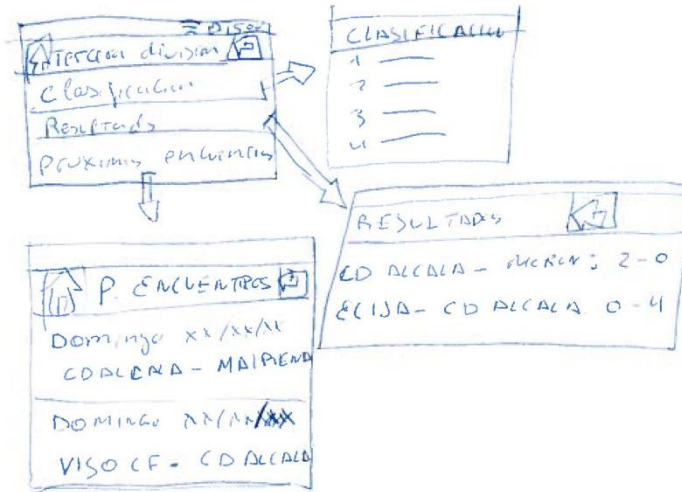
A continuación se describen el prototipo y el diseño inicial de la aplicación.

### 2.3.1 Sketches







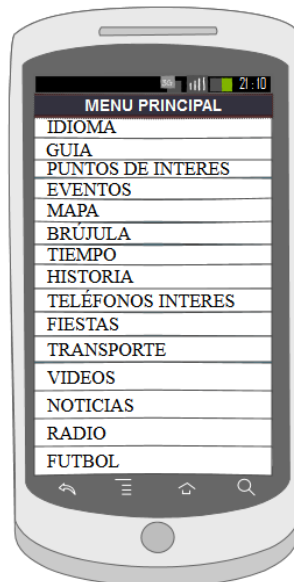


### 2.3.2 Prototipos horizontal

Pantalla Logo



Pantalla Menu Principal



Pantalla Idioma



Pantallas Guía: Enclaves y Rutas turísticas



Pantallas Puntos de Interés (creación, edición, eliminación de grupos y POIs)



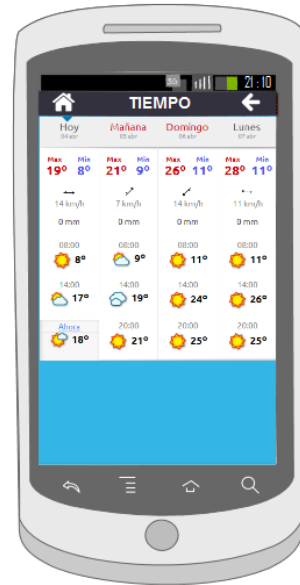
Pantallas Eventos



Pantalla Historia

Pantalla Brújula

Pantalla Tiempo



Pantalla Teléfonos Interés



Pantalla Fiestas



Pantalla Transporte



Pantalla videos



Pantalla Noticias



Pantalla Radio online



Pantallas Fútbol





## 2.4. Desarrollo pendiente a implementar.

Dado el limitado tiempo del que se ha dispuesto para el desarrollo de la aplicación se han dejado funcionalidades sin desarrollar las cuales son:

- Transporte
- Fútbol
- Fiestas.

Indicar sobre estas tres opciones que no existe ningún RSS ni ningún XML que permita obtener la información de una manera simple y eficiente. Una solución que se ha pensado es la de crear manualmente la información y guardarlo en XML y alojar dichos XML en algún servidor tipo Google Drive y hacer que la aplicación se conectase y a dicho servidor y leyese los XML. Nuevamente por falta de tiempo no se ha implementado pero es la única opción posible. Esta solución requiere de tener a alguien que vaya actualizando dichos XML.

## 2.5. Modificación del diseño inicial

- **Selección idioma:** Dado que Android permite dividir los textos por idiomas gracias al archivo “string.xml” (string\_es.xml, string\_fr.xml, etc) se ha optado por eliminar la opción de selección de idiomas y visualizar el idioma en la aplicación según la configuración que se tenga en el dispositivo móvil. De esta forma si el usuario tiene en su móvil el idioma en español la aplicación detectará dicho idioma y mostrará los textos en español. De igual forma pasará con los usuarios que tenga el dispositivo en inglés.
- **Eventos:** En esta opción se indicó de poder mostrar la ubicación del evento. Dado que la gran mayoría de los eventos carecen de dirección o la dirección que aparece no es posible vincularla de forma automática con Google Maps (por ejemplo una de las direcciones es “biblioteca pública”) se ha optado por suprimir dicha funcionalidad.  
Una solución sería crear un XML con la información de los eventos y buscar manualmente la dirección de cada evento y guardarla a modo de nombre de calle o a través de su latitud-longitud.  
Se ha incluido la búsqueda de grupos por ser algo que se ha podido desarrollar sin demasiadas complicaciones.
- **Historia:** La opción de “Historia” ha sido sustituida por “Conócenos” donde se visualiza información socio-demográfica del municipio de carácter turístico.

## 2.6. Posibles mejoras

De las entrevistas realizadas se obtuvo una serie de posibles funcionalidades que debido a la falta de tiempo no se ha podido incluir pero que se tendrá en cuenta en una futura versión del software. Dichas funcionalidades son:

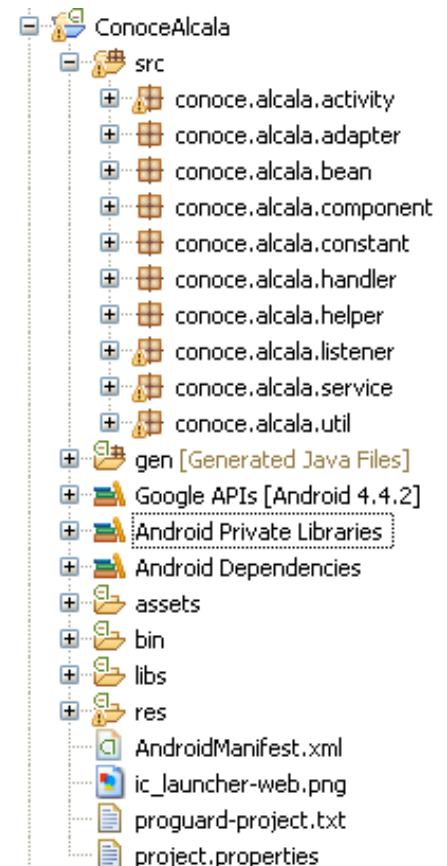
- Crear una visita virtual de los enclaves
- Poder chatear con gente de Alcalá
- Poder grabar los recorridos realizados por Alcalá semejante a lo que hacen las aplicaciones de running.
- Empleo de realidad aumentada (AR)

## 3. Implementación.

### 3.1. Estructura del proyecto

La estructura de carpetas (paquetes en Java) se divide en:

- *conoce.alcala.activity*: Paquete que incluye los activity usados en el proyecto.
- *conoce.alcala.adapter*: paquete que incluye los adapters.
- *conoce.alcala.bean*: paquete con los beans empleados.
- *conoce.alcala.component*: paquete que contiene las clases empleadas para visualizar ciertos componentes comunes en las pantallas (ventanas modales, ventana de progreso)
- *conoce.alcala.constant*: paquete que incluye clase con las constantes
- *conoce.alcala.handler*: paquete que incluye las clases que leen los RSS y XML
- *conoce.alcala.helper*: Paquete que incluye clase que interactúa con la BBDD.
- *conoce.alcala.listener*: paquete que incluye clase para la geolocalización.
- *conoce.alcala.service*: paquete que incluye clase para el servicio de la radio online
- *conoce.alcala.util*: paquete que incluye clase con los métodos útiles empleados en los activity (formateo de fecha, validación de conexión a internet, etc)



A continuación se pasará a describir las funcionalidades de las clases.

- **conoce.alcala.activity:**
  - *CompassActivity.java*: Activity empleado para visualizar brújula
  - *EventoActivity.java*: Activity empleado para visualizar los eventos
  - *EventoDetailActivity.java*: Activity empleado para visualizar el detalle de un evento.
  - *GoogleMapsActivity.java*: Activity empleado para visualizar mapas empleando Google Maps Api

- GuiaTuristicaActivity.java: Activity empleado para visualizar menú con las opciones de guía turística, es decir, las rutas de interés y los lugares de interés.
  - LugaresInteresActivity.java: Activity empleado para visualizar listado de los lugares de interés.
  - RutaInteresActivity.java: Activity empleado para visualizar listado de las rutas de interés.
  - MainActivity.java: Actitivity empleado para visualzar el menú principal de la aplicación.
  - NoticiaActivity.java: Activity empleado para visualizar las noticias.
  - PresentacionActivity.java: Activity empleado para visualizar escudo del municipio (antes de visualizar el menú principal).
  - PuntoInteresActivity.java: Activity empleado para visualizar, crear, borrar y editar los puntos de interés y los grupos.
  - RadioActivity.java: Activity empleado para visualizar radio online.
  - TelefonoActivity.java: Activity empleado para visualizar listado de teléfonos de interés.
  - TiempoActivity.java: Activity empleado para visualizar listado por días del tiempo del municipio.
  - TiempoDetailActivity.java: Activity empleado para visualizar el tiempo por horas del día seleccionado.
  - VideoActivity.java: Activity empleado para visualizar listado de videos en Youtube.
  - YoutubeActivity.java: Activity empleado para visualizar video embebiendo reproductor Youtube en la pantalla sin necesidad de ir directamente a la web. (ver YouTubeAndroidPlayerApi.jar)
  - WebActivity.java: Activity empleado para visualizar páginas web.
- **conoce.alcala.adapter**
- EventoAdapter.java: Clase empleada para cargar los valores extraídos de la página web en los campos que componen el listado de eventos.
  - EventoDetailAdapter.java: Clase empleada para cargar los valores extraídos de la página web en los campos que componen el listado de los detalles de los eventos.
  - GuiaAdapter.java: Clase empleada para cargar los valores extraídos del XML en los campos que componen el listado de los detalles de la guía.

- LugaresInteresAdapter.java: Clase empleada para cargar los valores extraídos del XML en los campos que componen el listado de los lugares de interés
  - MenuAdapter.java: Clase empleada para cargar los valores extraídos del XML en los campos que componen el listado del menú principal
  - NoticiaAdapter.java: Clase empleada para cargar los valores extraídos del XML en los campos que componen el listado de las noticias.
  - PuntoInteresAdapter.java: Clase empleada para cargar los valores extraídos del XML en los campos que componen el listado de los puntos de interés
  - RutaInteresAdapter.java: Clase empleada para cargar los valores extraídos del XML en los campos que componen el listado de las rutas de interés
  - TelefonoAdapter.java: Clase empleada para cargar los valores extraídos del XML en los campos que componen el listado de los teléfonos.
  - TiempoAdapter.java: Clase empleada para cargar los valores extraídos del XML en los campos que componen el listado del tiempo.
  - VideoAdapter.java: Clase empleada para cargar los valores extraídos del XML en los campos que componen el listado de los videos de Youtube
- 
- **conoce.alcala.bean**
    - Documento.java: Bean para guardar los documentos adjuntos de los eventos. Este bean se usa en Evanto.java
    - Evento.java: Bean para los eventos
    - Enclave.java: Bean para los enclaves
    - Guia.java: Bean para la guía turística
    - Ruta.java: Bean para las rutas de interés
    - Menu.java: Bean para el menú principal
    - Noticia.java: Bean para las noticias
    - PuntoInteres.java: Bean para los puntos de Interés
    - Telefono.java: Bean para los teléfonos
    - Tiempo.java: Bean para el tiempo
    - Video.java: Bean para los videos
  
  - **conoce.alcala.component**
    - DialogComponent.java: Clase que contiene componentes visuales tales como ventanas modales de información, de error y de confirmación.
    - ProgressDialogComponent.java: Clase que contiene el componente progreso para indicar al usuario que la aplicación está procesando la petición solicitada.

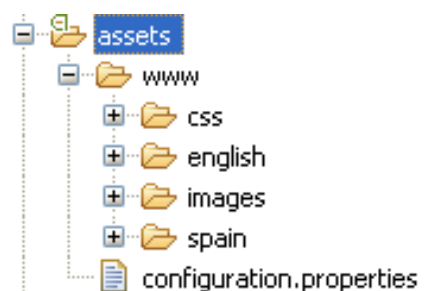
- **conoce.alcala.constant**  
Constant.java: clase que contiene las variables constantes.
- **conoce.alcala.handler**
  - TiempoHandler.java: Clase que lee y extrae los datos del XML externo relacionado con el tiempo.
  - VideoHandler.java: Clase que lee y extrae los datos del XML externo relacionado con los videos Youtube.
- **conoce.alcala.helper**  
ConoceAlcalaHelper.java: Clase que genera la base de datos en SQLITE así como las tablas. En el entregable se han incluido unos inserts a modo de ejemplo para permitir visualizar el funcionamiento de los puntos de interés.
- **conoce.alcala.listener**  
GeoLocationListener.java: Clase que implementa LocationListener y empleado para obtener la localización por medio de los proveedores de geolocalización del que disponga el dispositivo móvil (gps, wifi, etc).
- **conoce.alcala.service**  
RadioService.java: Clase que extiende de Services y que se emplea para crear una notificación y poder utilizar la radio en segundo plano.
- **conoce.alcala.util**  
Utils.java: Clase que contiene una serie de procedimientos y funciones útiles los cuales son empleados en la aplicación.

A continuación se pasa a explicar las carpetas propias de Android

- **assets**: carpeta que contiene datos internos tales como html y properties

Para el proyecto se ha creado una subcarpeta llamada “www” que contiene los html empleados para visualizar los lugares, rutas y los datos socios demográficos del municipio.

Indicar que debido a que la aplicación es multidioma se ha creado dos carpetas “english” y “spain” donde cada una alberga los html con los textos en los respectivos idiomas.



Destacar la importancia del fichero “configuration.properties” el cual contiene las rutas de los XML y RSS externos de los cuales se extrae la información, así como la ruta de archivos internos.

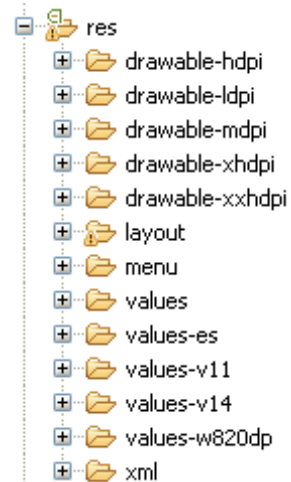
- **res**: carpeta que contiene los archivos usados para la visualización de las pantallas.

drawable-xxxx: carpetas que contienen las imágenes para las diferentes resoluciones de pantallas

menu: carpeta que contiene los XML referentes a los menú de las pantallas.

values: carpeta que contiene XML referentes al idioma, estilos y dimensiones de las pantallas.

Values-es: carpeta que contiene el idioma en español de los textos.

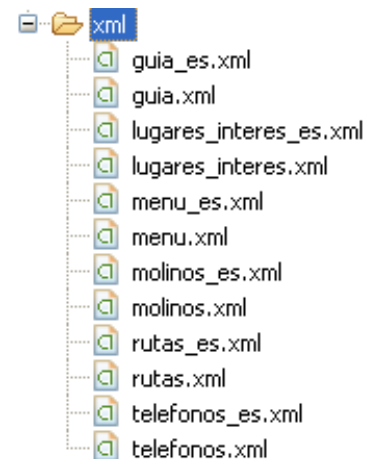


**Nota aclaratoria:** Debido a que Android proporciona la posibilidad de tener aplicaciones multidioma se optó por eliminar la opción de selección de idiomas y emplear la funcionalidad que proporciona Android. Si el usuario desea cambiar el idioma de la aplicación deberá cambiar el idioma del sistema operativo.

- **xml**: carpeta que contiene los XML empleados en las pantallas con listados.

Indicar que para visualizar los textos en el idioma del dispositivo móvil se ha creado dos versiones de los XML las versiones terminadas en “\_es” corresponde al idioma español. Para determinar el idioma se emplea la función “isEnglish()” incluida en la clase Utils.

Para la lectura de los XML se ha empleado SAX y XmlPullParser. Se ha empleado dichos parseadores por probar el comportamiento de los dos y para tener una visión más amplia del parseo de XML en Android.



### Lectura XML con SAX

```
public List<Tiempo> parse(String tag,String fecha){
    SAXParserFactory factory = SAXParserFactory.newInstance();
    try{
        URL url = Utils.getURL(prop.getProperty("url.weather"));
        SAXParser parser = factory.newSAXParser()
        TiempoHandler handler = new TiempoHandler(tag,fecha);
        parser.parse(Utils.getInputStream(url), handler);
        return handler.getTiempos();
    }catch (Exception e) {
        throw new RuntimeException(e);
    }
}
```

### Lectura XML con XmlPull

```

private List<Menu> parse() throws Exception{
    List<Menu> opciones = null;
    try{
        int xml = tils.isEnglish() ? R.xml.menu : R.xml.menu_es;
        XmlResourceParser xrp = etResources().getXml(xml);
        int evento = xrp.getEventType();
        Menu menuActual = null;
        while (evento != XmlPullParser.END_DOCUMENT) {
            String etiqueta = null;
            switch (evento) {
                case XmlPullParser.START_DOCUMENT:
                    opciones = new ArrayList<Menu>();
                    break;
                case XmlPullParser.START_TAG:
                    etiqueta = xrp.getName();
                    if (etiqueta.equals("Option")) {
                        menuActual = new Menu();
                    }else if (menuActual != null) {
                        if (etiqueta.equals("Texto")){
                            menuActual.setText(xrp.nextText());
                        }else if (etiqueta.equals("Id")){
                            menuActual.setId(new
                                Long(xrp.nextText()));
                        }
                    }
                    break;
                case XmlPullParser.END_TAG:
                    etiqueta = xrp.getName();
                    if (etiqueta.equals("Option") && menuActual
                        != null){
                        opciones.add(menuActual);
                    }
                    break;
            }
            evento = xrp.next();
        }
    }catch (Exception ex) {throw new Exception(ex);}
    return opciones;
}
    
```

- **layout:** carpeta que contiene los layout usados en las pantallas. A continuación se describe cada uno de los layouts
  - activity\_main.xml: Empleado para visualizar listado de opciones del menú principal (ver MainActivity.java)
  - compass.xml: Empleado para la opción de brújula (ver CompassActivity.java)
  - evento\_detail.xml: Empleado para la pantalla de detalle del evento seleccionado (ver EventoDetailActivity.java)
  - evento\_list\_body.xml: Empleado para visualizar los campos que componen cada elemento del listado de eventos. (ver EventoDetailAdapter.java y EventoAdapter.java)

- evento\_list.xml: Empleado para visualizar listado de eventos. (ver EventoDetailAdapter.java, EventoAdapter.java y EventoActivity.java)
- google\_maps.xml: Empleado para visualizar mapas de Google en las pantallas. (ver GoogleMapsActivity.java)
- grupo\_puntos\_interes\_body\_list.xml: Empleado para visualizar los campos que componen cada elemento del listado de grupos (ver PuntoInteresAdapter.java)
- grupo\_puntos\_interes\_edition.xml: Empleado para visualizar el detalle de un grupo. (ver PuntoInteresActivity.java)
- grupo\_puntos\_interes\_list.xml: Empleado para visualizar listado de grupos. (ver PuntoInteresActivity.java)
- guia\_list.xml: Empleado para visualizar listado con la guía de interés. (ver GuiaAdapter.java)
- guia\_lugares\_interes\_detail.xml: Empleado para visualizar listado con los lugares de interés. (ver LugaresInteresAdapter.java)
- list\_view\_simple.xml: Empleado para visualizar listado, es un layout genérico
- menu.xml: Empleado para visualizar campos empleados en el listado del menú principal (ver MenuAdapter.java)
- noticia.xml: Empleado para visualizar los campos empleados en el listado de noticias. (ver NoticiaAdapter.java)
- pantalla\_presentacion.xml: Empleado para visualizar el escudo del municipio. (ver PresentacionActivity.java)
- puntos\_interes\_edition.xml: Empleado para visualizar pantalla de edición de un punto de interés. (ver PuntoInteresActivity.java)
- radio\_stream.xml: Empleado para visualizar pantalla radio online (ver RadioActivity.java)
- ruta\_interes\_detail.xml
- telefono.xml: Empleado para visualizar listado de teléfonos (ver TelefonoAdaptar.java)
- tiempo\_list\_body.xml: Empleado para visualizar los campos empleados en el listado de tiempo por días. (ver TiempoAdapter.java)
- tiempo\_list\_detail.xml: Empleado para visualizar el tiempo por horas de para una fecha seleccionada. (ver TiempoAdapter.java)
- tiempo\_list.xml: Empleado para visualizar listado por días del tiempo del municipio (ver TiempoActivity.java, TiempoDetailActivity.java, TiempoAdapter.java)



- video.xml: Empleado para visualizar listado de videos (ver VideoActivity.java)
- web.xml: Empleado para visualizar navegador web (ver WebActivity.java)
- youtube.xml: Empleado para visualizar pantalla con reproductor de Youtube embebido. (ver YoutubeActivity.java).

A continuación se explicarán el uso de las librerías.

- saripaar-11242012.jar: Librería que permite realizar validaciones de campos empleando anotaciones. (ver PuntoInteresActivity.java)

```
public class PuntoInteresActivity extends Activity implements
Validator.ValidationListener {
...
@Required(order = 1, message = "El campo 'nombre' es
obligatorio")
@TextRule(order = 2, minLength = 1, maxLength = 90, message =
"Campo 'Nombre' entre 1 a 90 caracteres")
private EditText nombreEditText = null;

@TextRule(order = 3, minLength = 0, maxLength = 100, message =
"Campo 'Descripción' 0 a 100 caracteres")
private EditText descripcionEditText = null;
...
}
```

- yandex-translator-java.jar: Librería empleada para traducir textos de español a inglés. Se usa en las pantallas del tiempo, debido a que los textos que se visualizan son dinámicos y procedentes de un XML externo. (ver Utils.java)

```
public static String textTraductor(String text){
    try{
        Translate.setKey("trnsl.1.1.20140506T170003Z.71406b0d63622e
22.9c627363f047a90e0bb2a122c474a1456aa2f3ea");
        text = Translate.execute(text, Language.SPANISH,
Language.ENGLISH);
    }catch(Exception e){
        e.printStackTrace();
    }
    return text;
}
```

- YouTubeAndroidPlayerApi.jar: Librería para embeber reproductor de video de Youtube (ver YoutubeActivity.java).
- commons-lang3-3.3.2.jar: Se emplea para poder usar el escape de textos: StringEscapeUtils (ver Utils.java)

```
public static String unescapeHtml(String textHtml){
    return StringEscapeUtils.unescapeHtml4(textHtml);
}
```

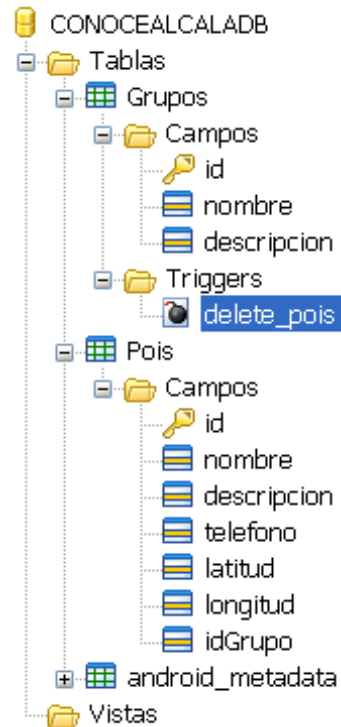
- google-play-services.jar: Librería empleada para poder integrar aplicaciones de Google como de terceros (Youtube, Google Maps, Google Street, etc).
- android-support-v4.jar: La librería de soporte contiene una variedad de APIs que o bien se añadieron en versiones más recientes de Android o bien no existen en la plataforma y sólo proporcionan soporte adicional cuando estás desarrollando funcionalidades específicas.

### 3.2. SQLITE

La aplicación utiliza SQLITE ya que viene integrada con Android y es un SGBD ligero. A continuación se muestra la estructura de las tablas.

A modo de ejemplo y de partida se ha incluido una serie de inserts los cuales serán creados junto con la base de datos. (ver ConoceAlcalaHelper.java)

```
@Override
public void onCreate(SQLiteDatabase db) throws
SQLException {
    try {
        db.execSQL(sqlCreateGrupos);
        db.execSQL(sqlCreatePois);
        db.execSQL("CREATE TRIGGER delete_pois
        BEFORE DELETE ON Grupos "
        + "FOR EACH ROW BEGIN"
        + " DELETE FROM Pois WHERE Pois.idGrupo =
        old.id;" + " END");
        db.execSQL("INSERT INTO Grupos (id, nombre,
        descripcion) VALUES(1, 'RESTAURANTES Y
        BARES', 'Restaurantes en Alcalá de Guadaíra)");
        ...
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```



### 3.3. Pruebas.

La aplicación ha sido probada en diferentes dispositivos:

- Motorola Motog: versión Android 4.4.2
- Samsung Galaxy Mini: version Android 2.3.2
- Samsung Galaxy Tablet II: versión Android 4.0.

Para las pruebas unitarias se ha utilizado el frameworks [Robotium](#) el cual se explicará a continuación:

- Las clases para los test tienen que heredar de la clase *ActivityInstrumentationTestCase2*.
- El objeto utilizado es del tipo *Solo* (*com.robotium.solo.Solo*)
- El método *testPreferencesSaved()* contiene la lógica de las pruebas.

```

package conoce.alcala.test;

import android.test.ActivityInstrumentationTestCase2;

import com.robotium.solo.Solo;

import conoce.alcala.R;
import conoce.alcala.activity.EventoActivity;

public class EventoTest extends
ActivityInstrumentationTestCase2<EventoActivity> {

    public EventoTest() {
        super(EventoActivity.class);
    }

    private Solo solo;

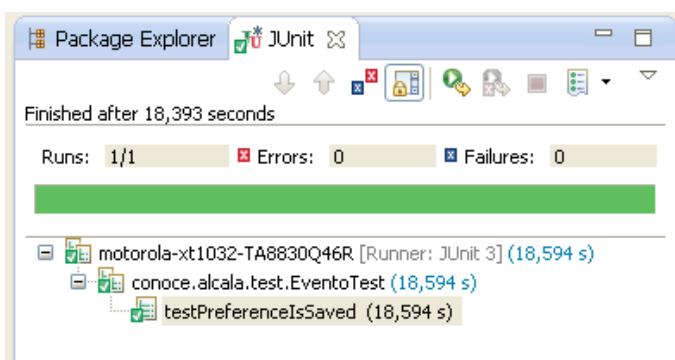
    public void setUp() throws Exception {
        solo = new Solo(getInstrumentation(), getActivity());
    }

    public void testPreferenceIsSaved() throws Exception {
        solo.assertCurrentActivity("wrong EventoActivity", EventoActivity.class);

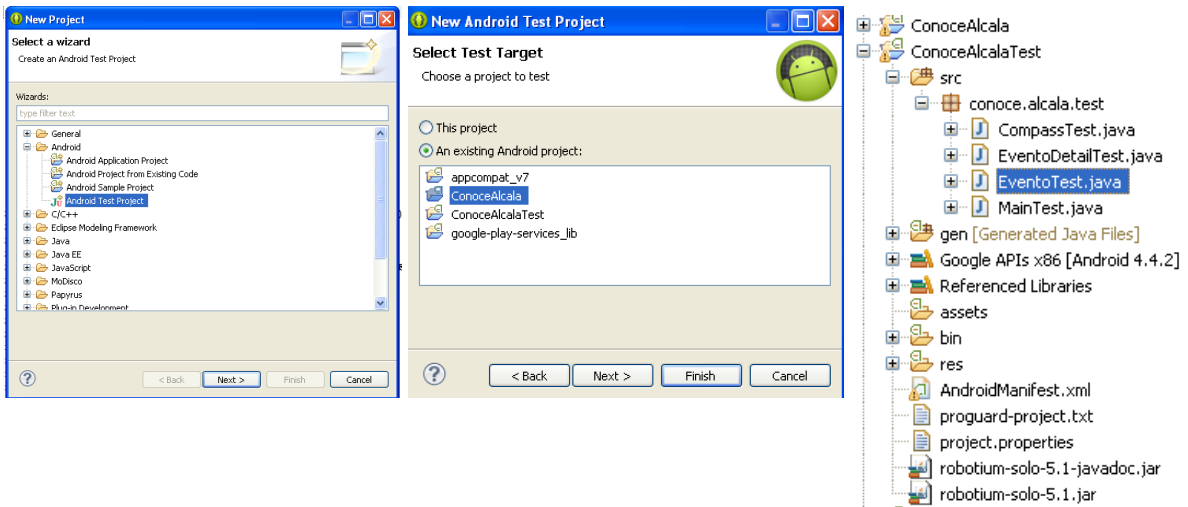
        solo.clickOnActionBarItem(R.id.MnuOpcCalendar);
        solo.setDatePicker(0, 2014, 3, 15);
        solo.clickOnText("Listo");
        solo.clickInList(1);
        solo.waitForDialogToClose();
        solo.clickOnActionBarItem(R.id.MnuOpcCreateEvent);
        solo.goBack();
    }

    @Override
    public void tearDown() throws Exception {
        solo.finishOpenedActivities();
    }
}
    
```

Resultado de la prueba:



Para la realización de las pruebas se ha creado un nuevo proyecto “ConoceAlcalaTest” de tipo “Android Test Project” el cual se ha vinculado con el proyecto “ConoceAlcala”.



Para usar robotium es necesario importar librería externa “robotium-solo” dentro del proyecto.



[Project Home](#) [Downloads](#) [Wiki](#) [Issues](#)

## Downloads

Google no longer accepts uploading of new binaries to the Google Code page. The files are now hosted at [Bintray](#).

### Current Direct Downloads

File	Description
<a href="#">robotium-solo-5.1.jar</a>	Robotium Solo 5.1
<a href="#">robotium-solo-5.1-javadoc.jar</a>	Robotium Solo 5.1 Javadoc
<a href="#">ExampleTestProject_v5.1.zip</a>	Example Test Project 5.1

### 3.4. Capturas pantallas

A continuación se muestran las pantallas finales desarrolladas.

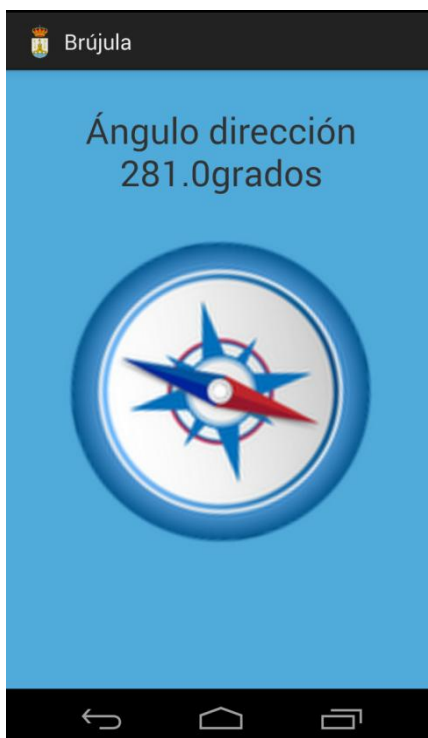
Pantalla Presentación



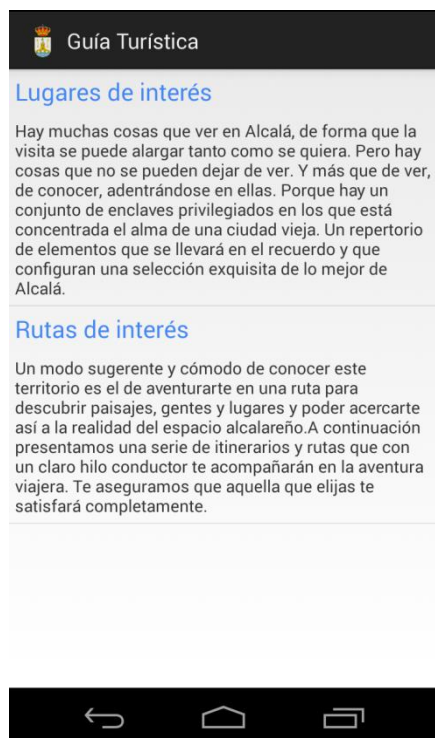
Pantalla Menú principal



Pantalla Brújula



Pantalla Guía turística



## Pantalla Lugares de Interés

**Lugares de interés**

**Recinto fortificado. Castillo medieval.**

Como todas las ciudades viejas, Alcalá ha ido acumulando historia desde que su suelo fuera ocupado por primera vez por los hombres prehistóricos que hallaron Gandul.

**Los molinos. Parque natural de Oromana.**

Por la importancia de su industria panadera la ciudad fue conocida como "Alcalá de los Panaderos". Los Molinos conservados datan de la época islámica y bajomedieval.

**Puente del Dragón. El Guardián del Castillo.**

Constituye una de las obras de ingeniería más importante de Andalucía y el primer puente figurativo de Europa. El puente representa la imagen de un dragón como guardián de la fortaleza situada justo detrás. La figura ha sido revestida con mosaico de azulejo.

**Castillo de Marchenilla.**

Edificado en dos fases constructivas; la imagen que llega a nuestros días es de traza renacentista. Posee una capilla de San Isidro Labrador, presidida por una valiosa pintura de la Virgen de Guadalupe.

## Pantalla Rutas de Interés

**Rutas de interés**

**La Ruta de Oromana**

Zona: Urbana  
 Tipo: Verde  
 Distancia: 2km  
 Dificultad: Baja  
 Realización: Apie o en bici  
 Duración: 1 hora a pie

**La Ruta de la Retama**

Zona: Urbana  
 Tipo: Verde  
 Distancia: 2km  
 Dificultad: Media-Baja  
 Realización: Apie o en bici  
 Duración: 2:30 horas a pie

**La Ruta de Marchenilla**

Zona: Guadaira y Escarpe del Alcor  
 Tipo: Verde  
 Distancia: 3.5km  
 Dificultad: Baja-Media  
 Realización: Apie o en bici  
 Duración: 1 hora a pie

**Ruta del Castillo**

Zona: Recinto fortificado  
 Tipo: Cultural  
 Distancia: 500m  
 Dificultad: Baja  
 Realización: A pie  
 Duración: 1h a pie

## Pantalla detalle lugar de interés

**Recinto fortificado. Castillo...**

**El Castillo**  
4500 AÑOS DE HISTORIA HUMANA SOBRE EL CERRO DEL CASTILLO

El Castillo de Alcalá es el resultado de cerca de 4.500 años de historia humana sobre el Cerro del Castillo. Los hallazgos arqueológicos más antiguos son los de la Edad del Bronce y la Edad del Hierro. Alcalá se fundó en un emplazamiento que ocupó ya el extremo norte del Cerro. Desde su emplazamiento defensivo la población se fue expandiendo hacia el sur y el este, ocupando el resto del cerro y el valle de Alcalá. En el siglo VIII, tras la caída del Imperio Carolingio, el Cerro del Castillo volvió a ser un asentamiento humano con un poblamiento magrija y islámico que terminó en un momento de esplendor en la época abisal (siglos XI y XII).

A partir del siglo IV y hasta el siglo XI, el Cerro del Castillo volvió a ser un emplazamiento al que la arqueología no aporta información. De hecho, aunque tenemos restos documentados del Castillo de Alá (siglo VIII) y parte del siglo IX en forma de los restos de "Torre", no existen restos arqueológicos de la fortificación antes de época almohade (siglos XII-XIII).

De la historia del Cerro del Castillo hasta el siglo XIV no podemos ver hoy nada, puesto que los indicios arqueológicos son escasos o no se hallan bajo superficie. Toda la edificación que podemos ver en la actualidad corresponde a la construcción de finales del siglo XV, cuando el Católico aragonés construyó una gran torre de alcazaba en el extremo norte del Cerro, como consecuencia del acuerdo de paz que se firmó entre el católico y el musulmán tras la batalla de Alcazar de San Juan.

La fortificación almohade del Cerro de los Siles es el núcleo del "Castillo", con el que hoy conocemos el conjunto de recintos (Plata de los Siles, Plata de la Sima y Alcazar Real) situados en el sector occidental del Cerro. Su fortificación actual responde a un desarrollo entre los siglos XI y XII, y su posición de muralla y almenas marca la imagen más conocida del "Castillo de Alcalá". En primer momento se amplió la fortificación original en paralelo a la muralla y en época almohade se le añadieron los almenares que se ven hoy en día. En el siglo XIII, cuando el Cerro de los Siles se amplió hacia el sur con un nuevo recinto del Cerro de la Sima, se empezó a construir el pequeño barrio "Barrio de San Isidro" que se desarrolló en Alcalá para abastecer a los soldados de la fortificación.

Tras la conquista castellana de 1347 se produjeron nuevas obras, dentro del proceso del cerco y conquista militar de Sevilla. Probablemente en momento se amplió el Cerro de los Siles hacia el este, con la construcción de dos torres adosadas que constituyen un elemento especial de arquitectura "peninsular" en Andalucía: el almenar de la Torre de la Plata, construido pocos años después en Sevilla. Tal y como narra la Primera Crónica General, las obras del Castillo de Alcalá serían promovidas por el papa Fernando IV, quien se instaló en Alcalá para abastecer a los soldados de la fortificación.

A finales del siglo XIII, Alcalá entra a formar parte de la línea interior del conjunto de fortificaciones de la "Bandera Morisca", espacio fortificado entre el Cerro de Sevilla y el Cerro de Granada. Durante varios decenios, el Cerro de los Siles y sus alrededores a través del Cerro del Castillo serían controlados, especialmente por los señores berberiscos que procedían de Sevilla y Alcazar de San Juan, quienes se asentaron en el Cerro del Castillo. Con esta tradición se constituye por primera vez un asentamiento urbano en el Cerro del Castillo, pero arqueológicamente sabemos que en época almohade nunca llegó a existir una ciudad fundada para o a la fortaleza. Por la situación militar del momento, la villa de Alcalá se amuralló rápidamente, cerrando la muralla con un circuito continuo de murallas que rodeaban al Cerro.

La villa de Alcalá durante el siglo XIV se desarrolló en la parte superior del Cerro del Castillo, entre la iglesia de Santa María del Real. Actualmente no podemos ver nada de su urbanismo, aunque las excavaciones han desarrollado la existencia de murallas de casas de grandes dimensiones, en torno a un patio central rodeado por un muro de mampolenas. La fortificación exterior del barrio (calles de Santa Bárbara, en redonda a través de las puertas, la de Santa María (barrio conocido como "Torre Morisca") y la de San Miguel al sur).

Juntamente, el Castillo de Alcalá dependía desde finales del siglo XIII del Convento (antiguo Ayuntamiento) de Sevilla, situación que se mantuvo hasta el año 2007, en que la titularidad de la fortaleza ha pasado definitivamente al Ayuntamiento de Alcalá de Guadaíra. Esta situación jurídica para Alcalá a principios del siglo XX, representa el momento de la independencia del Cerro del Castillo de Alcalá de Guadaíra y de Sevilla. En el caso de Alcalá, en el momento de la independencia de la ciudad, el Cerro del Castillo era un recinto fortificado que pertenecía a la jurisdicción de la Corona, y no a la jurisdicción de Alcalá. En el momento de la independencia de la ciudad, el Cerro del Castillo era un recinto fortificado que pertenecía a la jurisdicción de la Corona, y no a la jurisdicción de Alcalá.

El auge económico y demográfico que experimentó Alcalá durante la Baja Edad Media (siglos XIV-XV) explica que durante el siglo XIV se produjera un nuevo poblamiento en la parte superior del Cerro del Castillo, en torno a la iglesia de San Miguel. Este nuevo poblamiento se desarrolló en un espacio que hoy en día es un patio no contrastado. También se construyeron o restauraron completamente los muros de mampolenas que rodeaban al Cerro de los Siles y el Cerro de la Sima, convirtiéndolos en un recinto fortificado.

En la zona del Castillo, las transformaciones durante la Baja Edad Media son importantes. Un primer momento se fecha entre las épocas de Alfonso XI (1312-1350) y Pedro I (1350-1369), cuando probablemente se construyeron el Alcazar Real y la Torre del Homenaje. Posteriormente se empezó a edificar la torre del Cerro del Castillo de Alcalá por parte de Leonor de Guzmán, amante de Alfonso XI, lo que es un punto no contrastado. También se construyeron o restauraron completamente los muros de mampolenas que rodeaban al Cerro de los Siles y el Cerro de la Sima, convirtiéndolos en un recinto fortificado.

## Pantalla mapa lugar interés

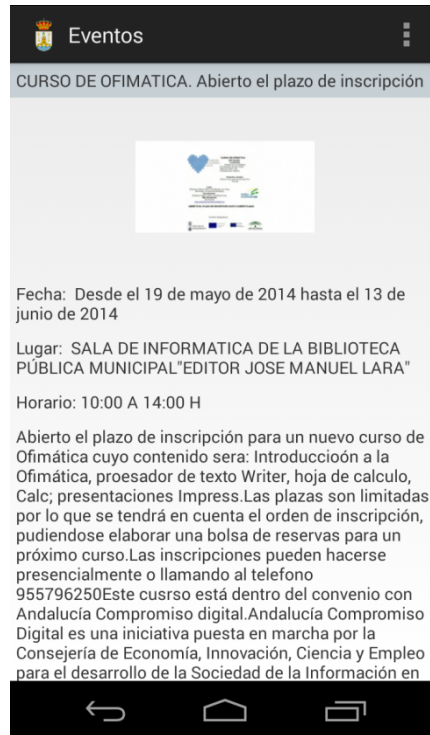
**Puntos de Interés**

**Recinto fortificado. Castillo medieval.**  
 Recinto fortificado. Castillo medieval.

Pantalla Evento



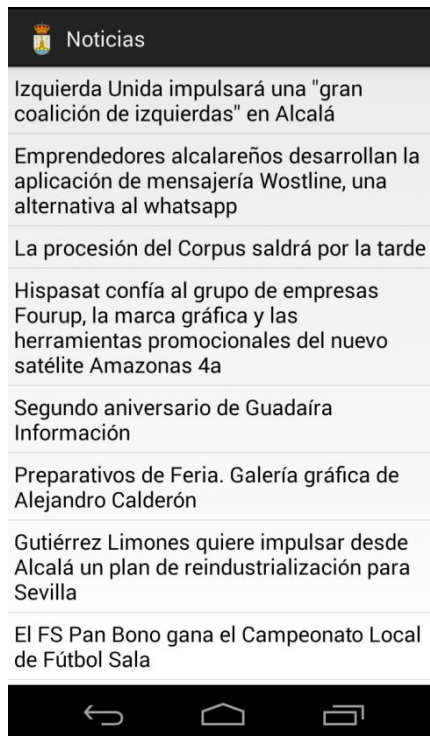
Pantalla detalle Evento



Pantalla Conócenos



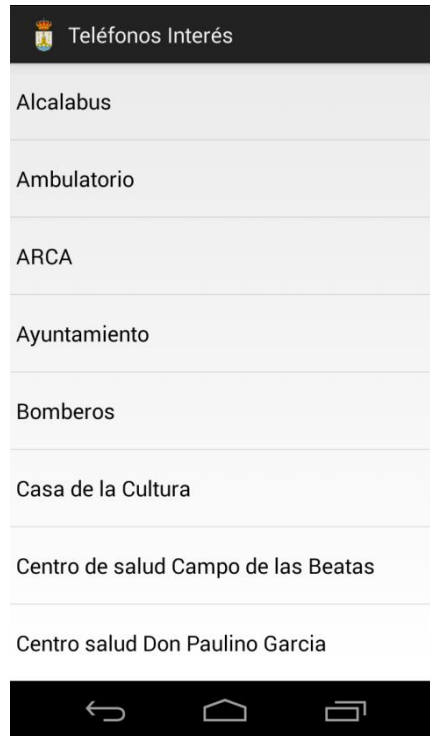
Pantalla Noticias



Pantalla Grupos puntos de interés



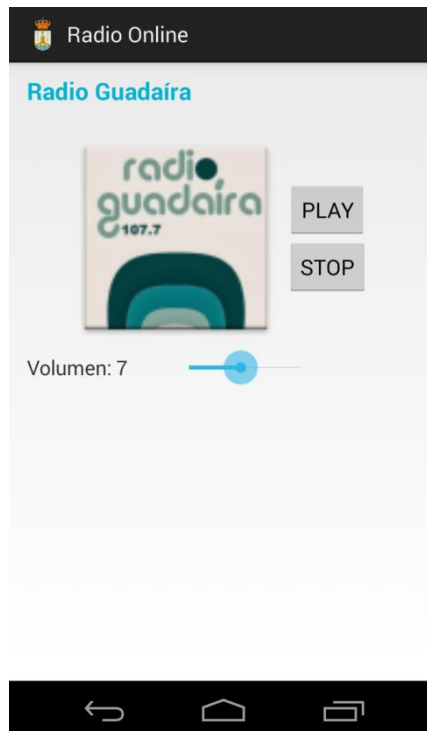
Pantalla Teléfonos interés



Pantalla Tiempo



Pantalla Radio online

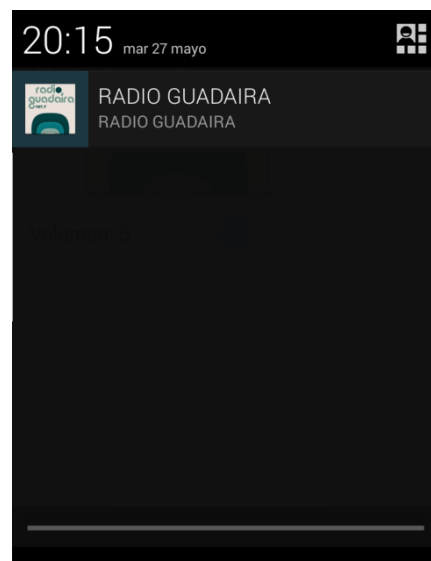




Pantalla Videos



Pantalla notificación radio online



## 4. Conclusión

La aplicación cumple con las funcionalidades que se describieron en la fase de diseño centrado en usuario. Es cierto que hay funcionalidades que no se han podido desarrollar y otras que se detectaron y de incluirse harían una aplicación más completa. No obstante, se ha tratado de respetar la compatibilidad con otras versiones de Android y de hacer un diseño amigable y fácil de manejar.

En definitiva, esta aplicación ha permitido adquirir unos conocimientos en cuanto a desarrollo de dispositivos móviles que era el principal objetivo del trabajo fin de carrera y además ha permitido generar una guía turística que permita potenciar visitas y dar a conocer el municipio.

## 5. Bibliografía

<https://developer.android.com/guide/index.html>

<http://pendientedemigracion.ucm.es/info/tecnomovil/documentos/android.pdf>

[http://www.sgoliver.net/blog/?page\\_id=3011](http://www.sgoliver.net/blog/?page_id=3011)

<http://www.androidcurso.com/index.php/curso-android-fundamentos/tutoriales-android-fundamentos/38-unidad-8-servicios-notificaciones-y-receptores-de-anuncios/288-un-servicio-para-ejecucion-en-segundo-plano>

<http://developer.android.com/reference/android/app/Activity.html>

<http://www.javaya.com.ar/androidya/detalleconcepto.php?codigo=140&inicio=>

<http://www.sgoliver.net/blog/?p=1542>

<http://www.sgoliver.net/blog/?p=1604>

<http://adf.ly/3103720/banner/http://rajeshvijayakumar.blogspot.com.es/2013/05/confirm-password-validation-using.html>

<http://api.yandex.com/translate/>

<https://developers.google.com/youtube/android/player/>

<http://commons.apache.org/proper/commons-lang/javadocs/api-release/index.html>

<http://phonegap.com/>

<http://www.phonegapspain.com/>

<https://code.google.com/p/robotium/>