



Título del Proyecto:

AudioGuía de Valencia

Autor:

Sanz Monllor, Gorka

Director:

Carbonell, Vicente

TESINA PARA LA OBTENCIÓN DEL TÍTULO DE:

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

**Diploma de Especialización en
Desarrollo de Aplicaciones para
Android**

**Diploma de Especialización en
Computación Móvil y Ubicua**

(Quitar no que no proceda)



Contenido

Título del Proyecto:.....	1
Autor:.....	1
Director:.....	1
Máster en Desarrollo de Aplicaciones sobre Dispositivos Móviles.....	1
Diploma de Especialización en Desarrollo de Aplicaciones para Android.....	1
Diploma de Especialización en Computación Móvil y Ubicua.....	1
Introducción.....	3
Descripción del problema.....	3
Objetivos.....	3
Motivación.....	4
Tecnologías utilizadas.....	4
Arquitectura de la aplicación.....	5
Esquema del diseño.....	5
Modelo de datos.....	13
Listado de Servicios Web.....	15
Los datos a representar son la temperatura y la humedad.....	17
Vistas.....	17
Capítulos adicionales.....	21
Conclusiones.....	21
Anexos.....	21
Listado de fuentes entregadas / Código fuente en GitHub.....	21
Manual de usuario.....	21

Introducción

Descripción del problema

Las aplicaciones de software realizadas que intentaban simular Audioguías se aplicaban fundamentalmente a museos y para la localización no incluían ningún tipo de tecnología o algunas como Zigbee o Bluetooth. La aparición de los Smartphone y la inclusión de hardware especial como dispositivos GPS, Bluetooth o internet en tecnología 3G hace posible la mejora del desarrollo de aplicaciones como audioguías, de tal forma que se pueda detectar directamente el monumento a visitar a partir del software desarrollado y del GPS que se ha instalado.

Gracias al GPS también se puede representar en un mapa el punto actual en el que se encuentra el respectivo turista y todos los puntos cercanos a visitar.

Objetivos

Los principales objetivos que tiene que cubrir la aplicación móvil AudioGuía son los siguientes:

- 1- Explicación del objetivo del proyecto: se describe que es una audioguía en la que se han seleccionado los puntos turísticos más importantes de la ciudad de Valencia: Ciudad de las Artes y las Ciencias, Palau de la Música, Lonja de Valencia y Palacio del Marqués de Dos Aguas.
- 2- Cuando el turista que usa la aplicación se encuentra en una posición cercana a cada uno de los monumentos a visitar, se reproduce un audio relativo a cada uno de los monumentos. Concretamente, cuando el usuario se encuentra a menos de 300 m. del monumento respectivo se reproduce el audio.
- 3- Mostrar Distancias: es una tabla donde aparecen las distancias desde el punto en el que actualmente está el turista con respecto al resto de puntos turísticos.

- 4- Mapa: Se muestra un mapa en el que aparece la posición del usuario y los puntos turísticos más cercanos a la posición de dicho usuario.
- 5- El tiempo en Valencia: Se muestra las características del tiempo que va a hacer en Valencia en los próximos 7 días. Se muestran valores de temperatura y humedad.

Motivación

La motivación de realizar el proyecto se debe a la necesidad de tener una audioguía de Valencia donde se pueda obtener información de los principales monumentos de la ciudad, en función de la localización. Aparte de esto, al turista se le permite conocer en todo momento su posición y la posición respecto a los monumentos más cercanos, además de la información temporal diaria en la semana en que se ha hecho el viaje.

Tecnologías utilizadas

Las principales tecnologías utilizadas han sido:

- 1- **Java- Nativo:** Se ha utilizado para el desarrollo principal de la aplicación. Todo esto incluye las Activities y la parte de los mapas y localización dentro de estos, así como el acceso a las bases de datos.
- 2- **Angular JS:** es un framework MVC que permite optimizar el código de la parte de FrontEnd. Es una evolución optimizada de JavaScript. Está basado en la definición de controladores donde se escriben variables que se acceden desde el frontend. Es un lenguaje que optimiza el acceso a microservicios y está especializado en el tratamiento con tecnologías json. En el caso del proyecto se utiliza para la obtención de datos en formato JSON de un servicio Web que devuelve información meteorológica de un determinado lugar. Utilizando Angular el procesado de datos JSON es mucho más sencillo que utilizando otra tecnología como java nativo. Lo que hará falta ahora es una tecnología específica para representar los datos en formato json que llegan del servicio WEB en Angular.

3- **JQuery Mobile:** es una tecnología que permite representar en el front end los datos que llegan de un JSON.

Arquitectura de la aplicación

Esquema del diseño

La aplicación audioguía se ha desarrollado en base a una actividad principal, proyecto, desde la que se llaman a varias actividades: MapsActivity, Instrucciones, WeatherActivity, Location

Instrucciones: Muestra en un layout las instrucciones de la aplicación

Location: Muestra distancias

MapsActivity: muestra la localización actual y los puntos importantes más cercanos

WeatherActivity: muestra el tiempo en Valencia en un período temporal de 7 días.

ShowImage: muestra el audio del monumento cuando el usuario se ha acercado 300m.

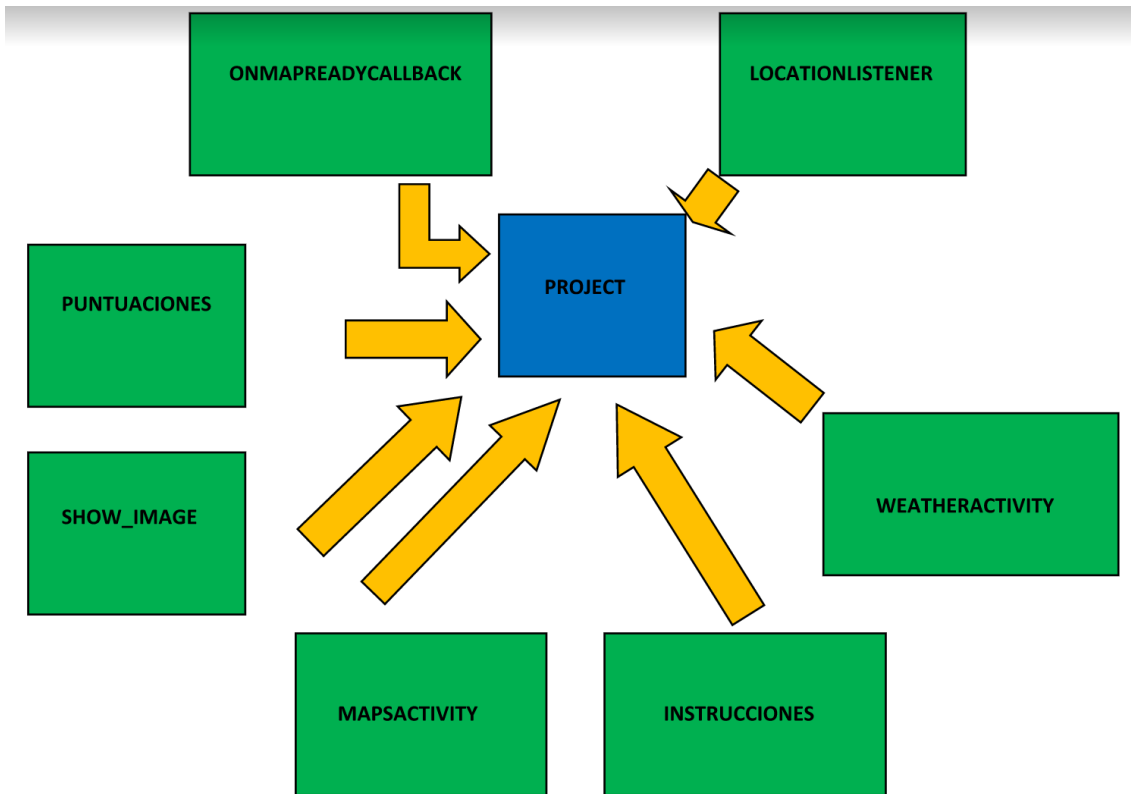


Ilustración 1

De la actividad temporal cuelgan todas las actividades. Implementa las interfaces `OnMapReadyCallback` y `LocationListener`. La utilidad de implementar `locationListener` se debe a que en la actividad principal cada cierto tiempo se monitoriza la posición del usuario, de tal manera que se leen las posiciones de los puntos característicos y se calcula la distancia respecto de ellos mediante los métodos que nos ofrece la clase `Location`.

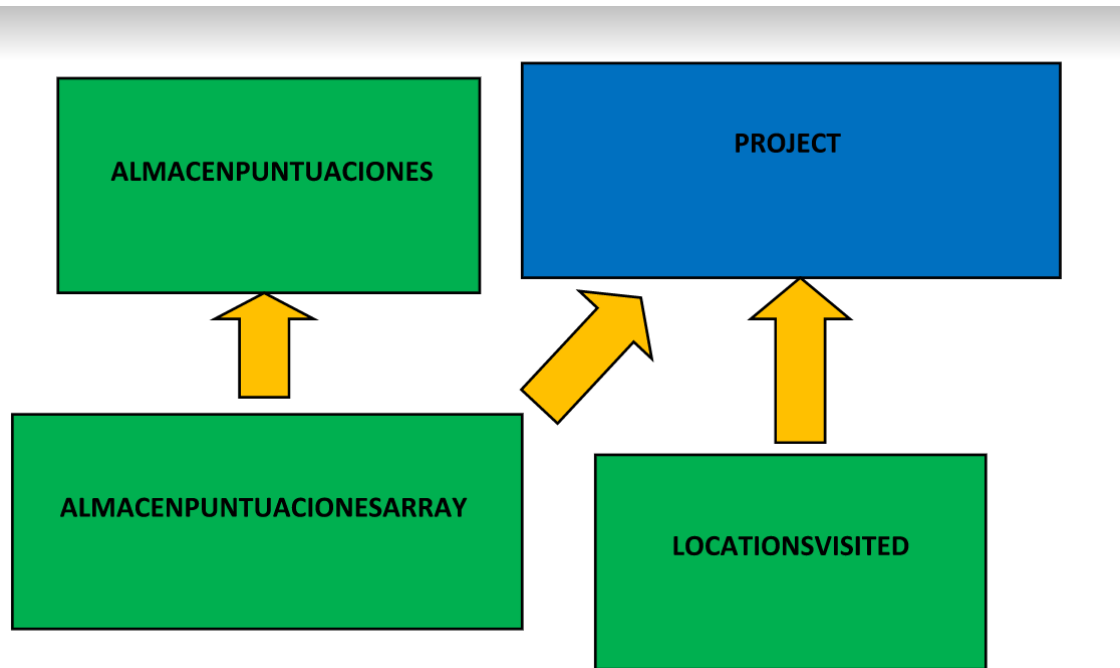


Ilustración 2

Toda esta información se almacena en la clase Almacen puntuaciones Array que es una clase que está formada por componentes de coordenadas (Longitud y Latitud) y que está preparada para ser insertada en un Adapter.

Instrucciones

Representa un layout donde aparecen las instrucciones de cómo utilizar la aplicación.

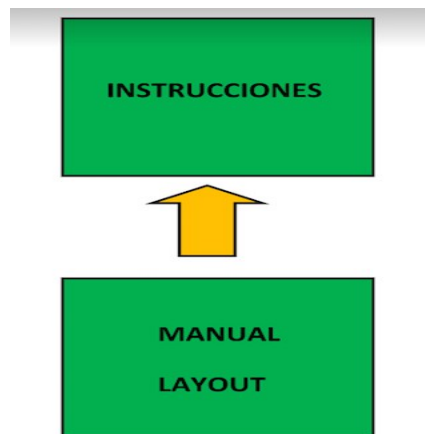


Ilustración 3

Puntuaciones

Es una clase que muestra las distancias de cada uno de los puntos a cada uno de los puntos de interés a partir de la información rellena en la actividad principal en la clase Almacén Puntuaciones. Están ordenadas por orden de menor distancia. Una vez se ha relleno el Adapter se utiliza el método `setListAdapter` para poder representar el respectivo monumento y la respectiva distancia al punto actual.

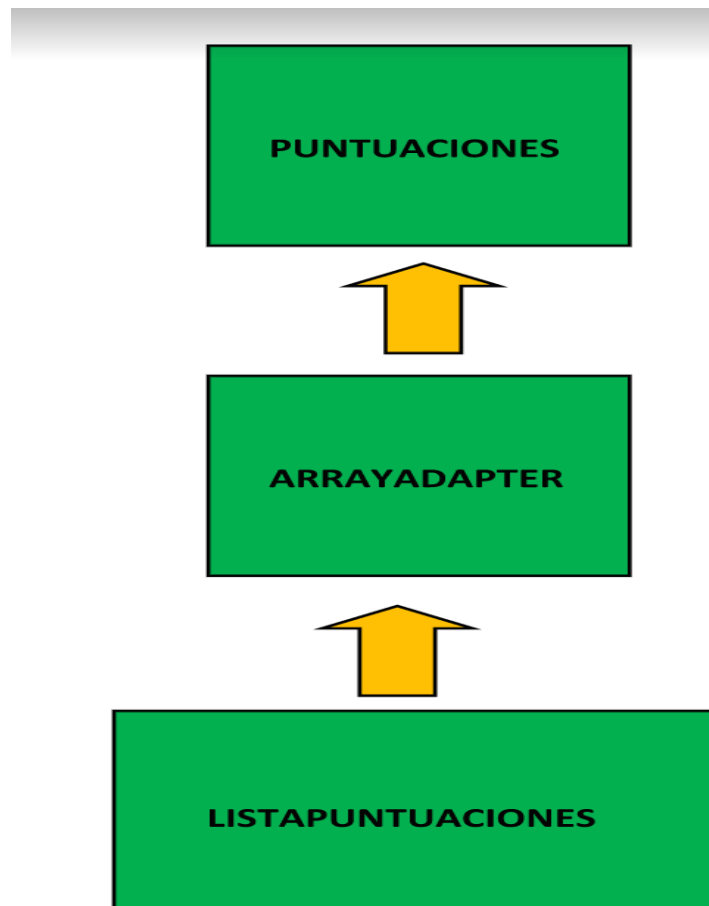


Ilustración 4

MapsActivity

Representa un mapa en código nativo donde se muestra la posición actual del turista y los puntos de interés más cercanos. Dichos puntos aparecen por el nombre. Básicamente al llamar a la actividad se pasan las coordenadas del usuario y se representan. Por otro lado se representa el resto de monumentos, de tal manera que en una escala se puede ver la distancia de todos los monumentos a la posición actual.

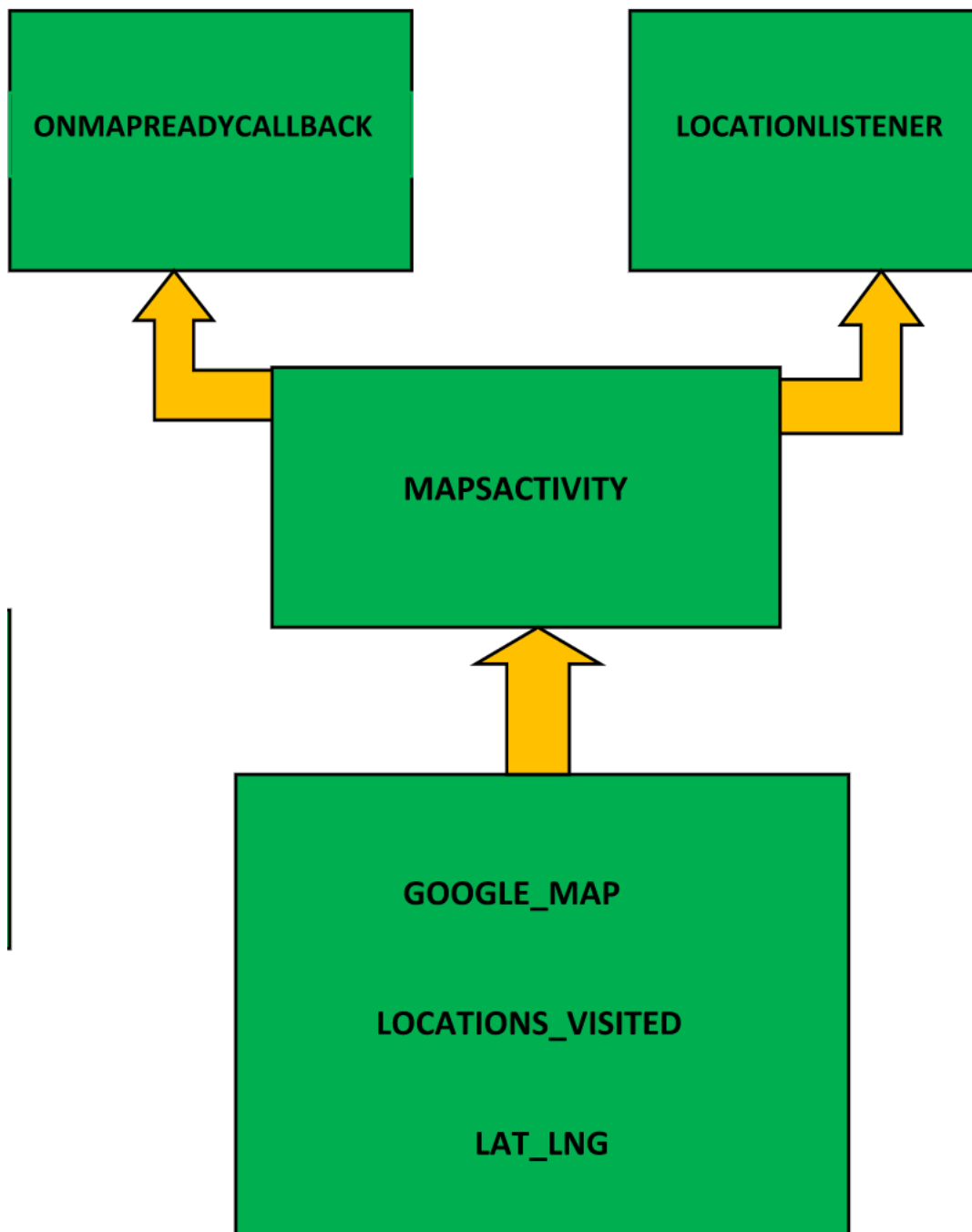


Ilustración 5

La actividad implementa las interfaces OnMapReadyCallBack y LocationListener. Utiliza las clases nativas de GoogleMap y las clases predefinidas LocationsVisited para representar la posición actual y la posición de todos los monumentos. A partir de ahí se representan en el mapa y a cada uno se le pone el nombre respectivo que tiene en la base

de datos, de tal manera que cuando se presiona sobre cada punto aparecerá el nombre de cada uno.

WeatherActivity

Para representar el tiempo se ha utilizado un servicio web que devuelve los datos en formato json, pasándole como parámetro de entrada el nombre de la ciudad de la que se quiere obtener los datos del tiempo, en este caso Valencia. Los datos que se obtienen van a ser la temperatura y la humedad. La tecnología para representar todos estos datos va a ser Angular.js por dos motivos: en primer lugar la facilidad para trabajar con formato json y de representar dichos datos en un frontend y en segundo lugar la posibilidad de comunicarse con microservicios.

El esquema utilizado sería el siguiente:

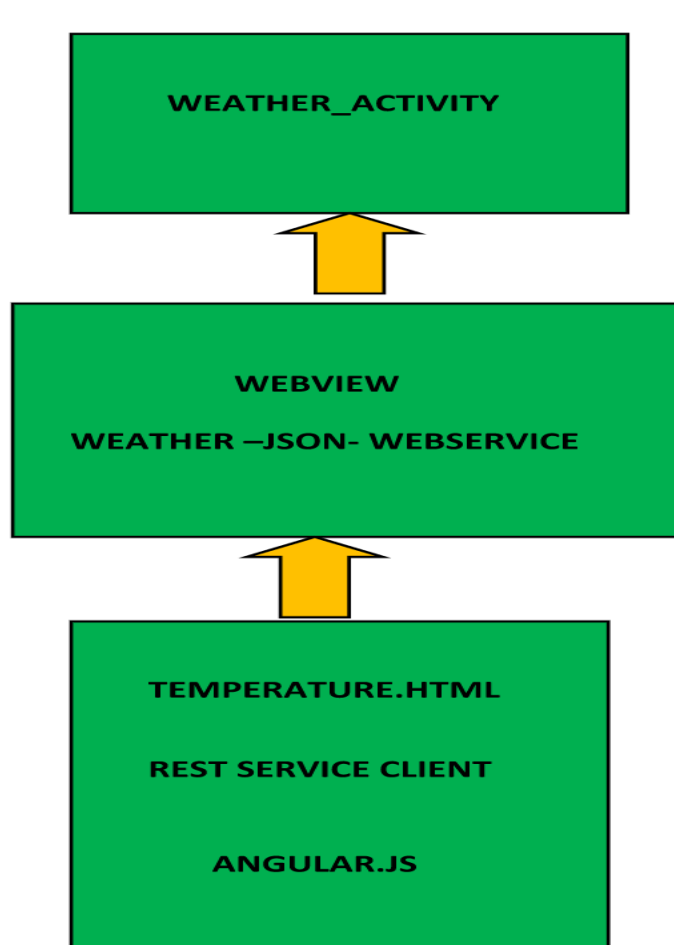


Ilustración 6

Se utiliza una actividad, WeatherActivity, y se usa la componente WebView para poder acceder al navegador y poder comunicarse con él, así como tratar todos los datos que devuelve en formato json. Desde el navegador es cuando se inicia la parte de Angular: procesado del json y representación de la información que llega del Webservice en el FrontEnd.

Show_image

Esta actividad aparece dentro del método onLocationChanged.

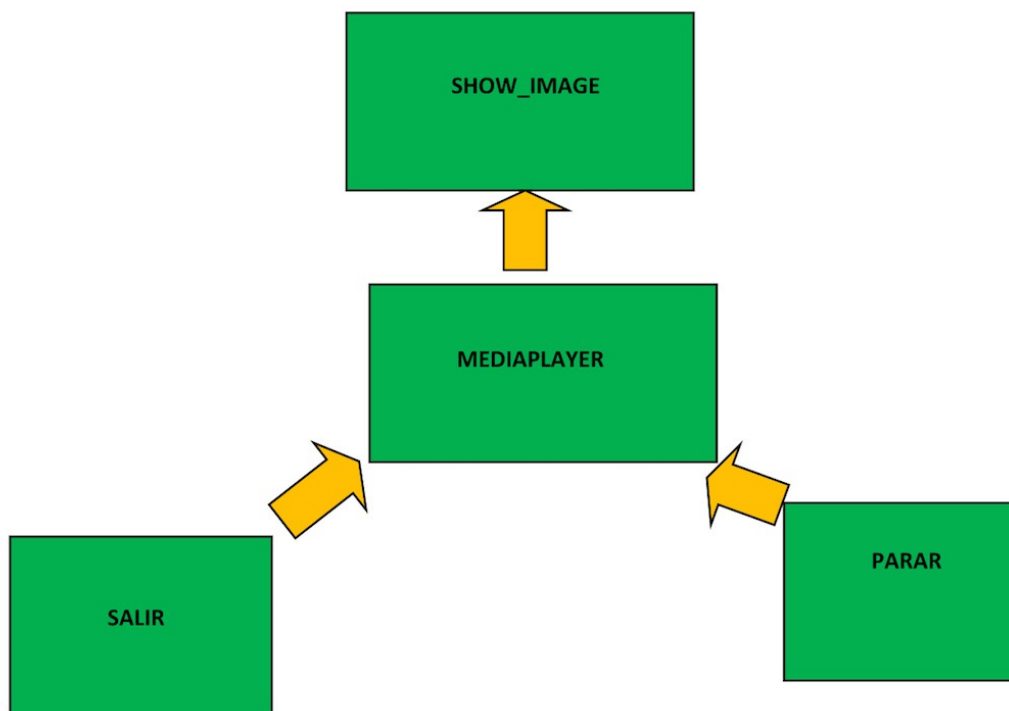


Ilustración 7

Básicamente se activa cuando se está en una localización en la que la distancia al punto de interés respectivo es inferior a 300 metros, aparece el respectivo audio del monumento más cercano. Está basado en un componente del tipo MediaPlayer, donde se han incluido dos botones adaptados para INICIAR/PARAR la actividad y otro para SALIR de la actividad.

El proceso que se sigue es el siguiente:

- 1- Cuando se inicia la actividad principal se lee de una base de datos todos los posibles puntos turísticos y de interés con sus respectivas coordenadas y se guardan en la clase LocationsVisited.
- 2- Se calcula la diferencia en distancia entre la localización actual y la localización de cada uno de dichos puntos.
- 3- Caso de que esa diferencia sea menor de 300 metros se abre una actividad en la que se reproduce el audio respectivo al punto de interés más cercano.

Modelo de datos

La base de datos que se va a utilizar es la siguiente:

ID	Latitude	Longitude	Image	Monument	Visited

Ilustración 8

Es una única tabla en la que se definen los siguientes campos:

ID: Primary Key. Es el identificador de la entrada de la Base de Datos

Latitude: TEXT. Es el valor de la Latitud del punto de interés

Longitude: TEXT. Es el valor de la longitud del punto de interés.

Image: TEXT. Es la ruta de la imagen asociada

Monument: TEXT. Es el nombre del monumento

Visited: Boolean. Es un parámetro booleano que indica si el monumento ha sido visitado o no.

Desde el punto de vista de las clases que se han desarrollado la base de datos se va a modelar con la clase LocationsVisited. Esta clase extiende de SQLiteHelper y sobrecarga varios métodos:

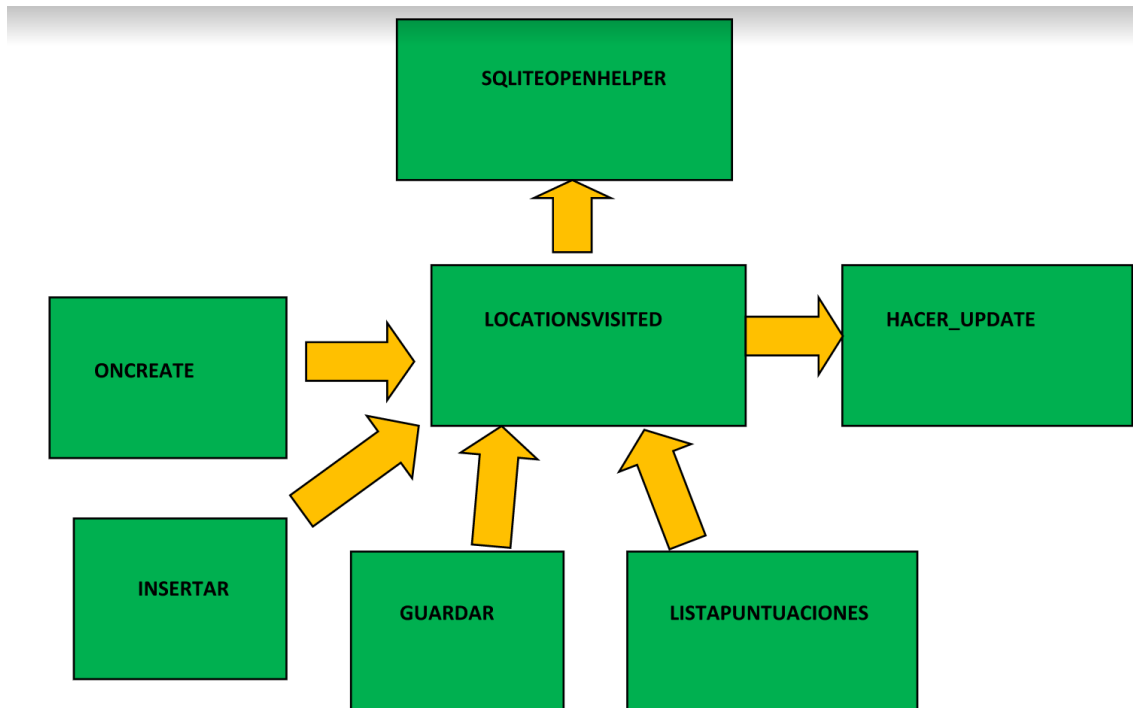


Ilustración 9

La clase Locations Visited hereda de SQLITEOPENHELPER y se utiliza para representar los accesos a la base de datos Turistic_Valencia. Los métodos implementados son los siguientes:

OnCreate: se crea la tabla Turistic Valencia con los siguientes campos: Latitude, Longitude, Image, Monument, Visited

Insertar : Inserta una entrada en la Base de Datos

Guardar: Realiza inserciones de todos los puntos turísticos a visitar en la base de datos. En este caso han sido: Ciudad de las Ciencias, Palacio del Marqués de Dos Aguas, Palau de la Musica y Lonja.

ListaPuntuaciones: selecciona todos los elementos de la base de datos para posteriormente poder mostrarlos en la parte de calcular la distancia.

Hacer_update: Hace un update en una fila de la base de datos. Modifica una entrada de la base de datos.

Listado de Servicios Web

Los datos acerca del tiempo se han obtenido a partir de la llamada al siguiente Webservice:

http://api.openweathermap.org/data/2.5/forecast/daily?q=NOMBRE_CIUADAD&cnt=NUMERO_DIAS&APPID=TOKEN

En nuestro caso, el nombre de la ciudad va a ser Valencia. El número de días va a ser 8 a partir del día actual y el token se obtiene a partir de registrarse en la Web. La url que se utiliza en el proyecto es la siguiente:

<http://api.openweathermap.org/data/2.5/forecast/daily?q=Valencia&cnt=8&APPID=be84d4a1241d45f7601b25bcf1cc984b>

que se devuelve al procesar la anterior url es el siguiente:

```
{"city":{"id":3625549,"name":"Valencia","coord":{"lon":-68.007652,"lat":10.16202},"country":"VE","population":0},"cod":"200","message":0.0132,"cnt":8,"list":[{"dt":1473350400,"temp":{"day":298.15,"min":296.48,"max":299.54,"night":296.48,"eve":299.54,"morn":298.15},"pressure":953.35,"humidity":100,"weather":[{"id":501,"main":"Rain","description":"moderate rain","icon":"10d"}],"speed":1.22,"deg":183,"clouds":92,"rain":5.61},"dt":1473436800,"temp":{"day":302.6,"min":295.24,"max":303.69,"night":296.1,"eve":303.69,"morn":295.24},"pressure":953.82,"humidity":76,"weather":[{"id":801,"main":"Clouds","description":"few clouds","icon":"02d"}],"speed":1.87,"deg":166,"clouds":20},"dt":1473523200,"temp":{"day":301.63,"min":294.9,"max":303.32,"night":294.9,"eve":303.32,"morn":295.32},"pressure":955.33,"humidity":67,"weather":[{"id":500,"main":"Rain","description":"light rain","icon":"10d"}],"speed":3.12,"deg":146,"clouds":56},"dt":1473609600,"temp":{"day":300.92,"min":292.58,"max":300.92,"night":295.08,"eve":299.86,"morn":292.58},"pressure":954.66,"humidity":74,"weather":[{"id":501,"main":"Rain","description":"moderate rain","icon":"10d"}],"speed":1.76,"deg":173,"clouds":8,"rain":8.83},"dt":1473696000,"temp":{"day":300.8,"min":294.94,"max":300.8,"night":294.94,"eve":296.64,"mor
```

```

n":296.7},"pressure":978.27,"humidity":0,"weather":
[{"id":501,"main":"Rain","description":"moderate
rain","icon":"10d"}],"speed":1.46,"deg":156,"clouds":16,"rain":11.5},
{"dt":1473782400,"temp":
{"day":300.68,"min":295.34,"max":300.68,"night":295.34,"eve":297.7,"mo
rn":296.86},"pressure":979.2,"humidity":0,"weather":
[{"id":501,"main":"Rain","description":"moderate
rain","icon":"10d"}],"speed":1.7,"deg":350,"clouds":32,"rain":4.96},
{"dt":1473868800,"temp":
{"day":300.36,"min":294.65,"max":300.36,"night":294.65,"eve":296.17,"m
orn":296.03},"pressure":979.77,"humidity":0,"weather":
[{"id":502,"main":"Rain","description":"heavy intensity
rain","icon":"10d"}],"speed":1.19,"deg":200,"clouds":40,"rain":13.35},
{"dt":1473955200,"temp":
{"day":300.87,"min":295.08,"max":300.87,"night":295.08,"eve":297.15,"m
orn":296.12},"pressure":980.38,"humidity":0,"weather":
[{"id":501,"main":"Rain","description":"moderate
rain","icon":"10d"}],"speed":1.72,"deg":131,"clouds":12,"rain":8.93}}]

```

Angular es la tecnología ideal para procesar la anterior información. La aplicación que se ha desarrollado está formada por un controlador desde el que se hace la petición GET a la anterior url. Como se ha observado, en el json está la fecha en milisegundos. En el controlador se ha incluido funciones que transforman la fecha en milisegundos en fecha a formato normal. Así se obtiene la respuesta en json y para cada una de las fechas en milisegundos se hace la anterior transformación. A partir de ahí desde el frontend se trata todo. El esquema sería el siguiente:

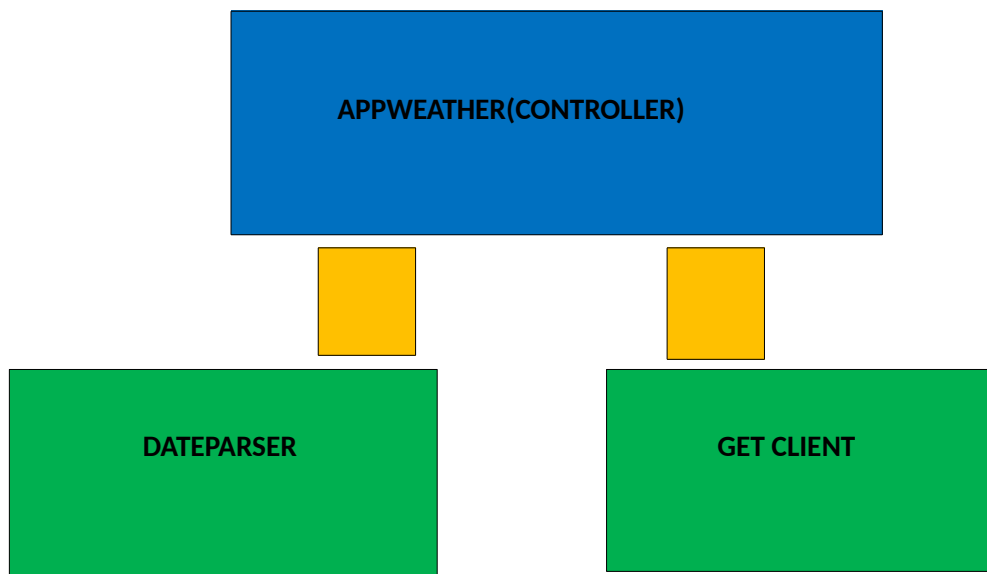


Ilustración 10

Los datos a representar son la temperatura y la humedad.

Vistas

Esquema que muestre las principales pantallas de la aplicación y el diagrama de navegación.



Ilustración 11 Pantalla principal

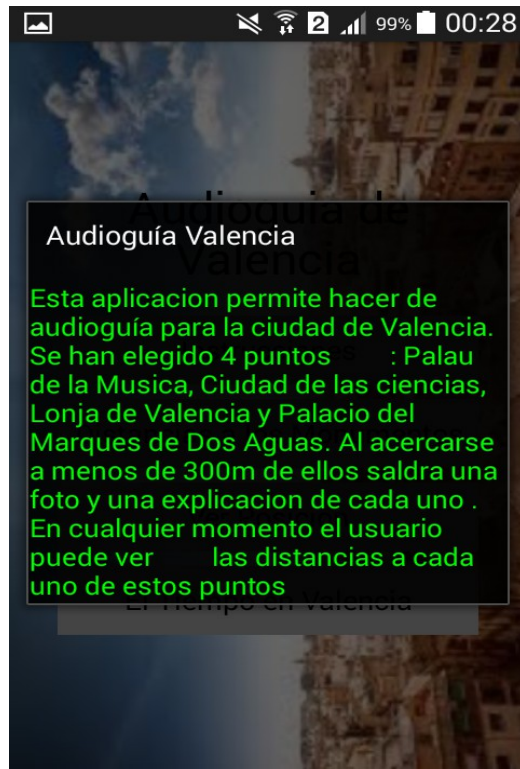


Ilustración 12 Instrucciones



Ilustración 13 Distancia a los monumentos

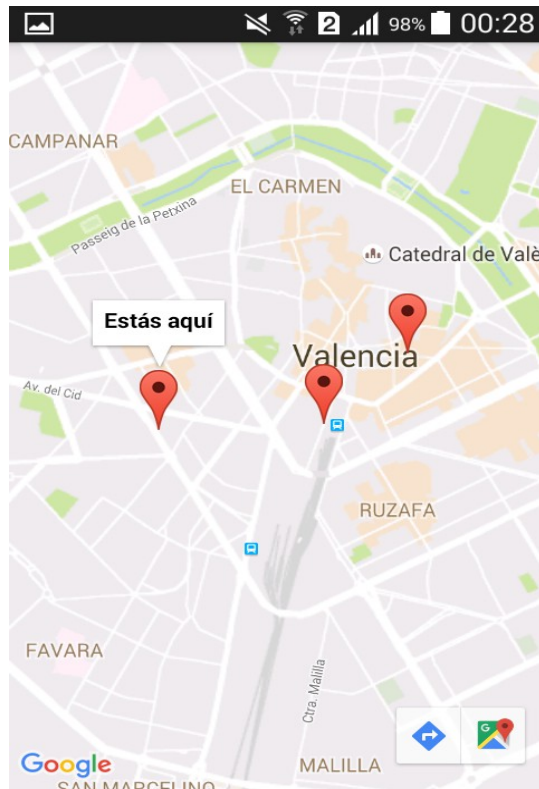


Ilustración 14 Ubicación en el mapa

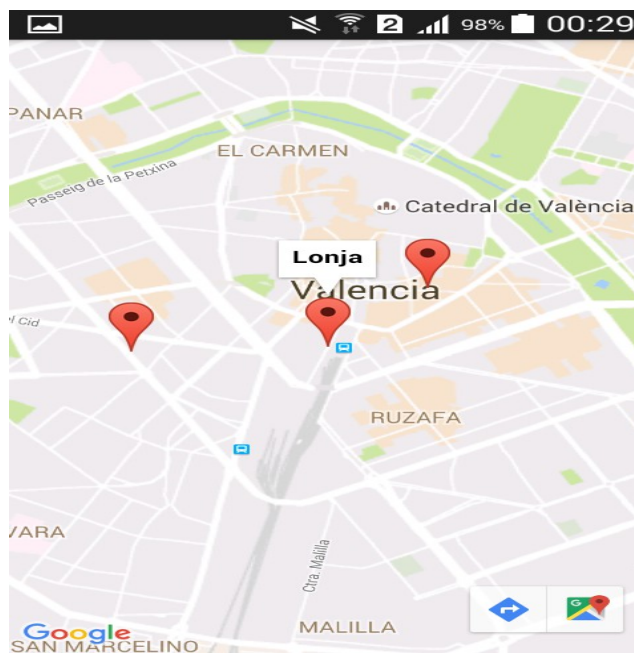


Ilustración 15 Ubicación el mapa



Ilustración 16 Audio de la Ciudad de las Ciencias

Capítulos adicionales

Conclusiones

Grado de cumplimiento de los objetivos planteados: en principio se han cumplido la totalidad de los objetivos que se han planteado. La idea es crear una aplicación que sirva como audioguía para la ciudad de Valencia para un turista. Posibles mejoras de la aplicación podrían ser las siguientes:

- 1- Mejorar la medida de distancias a todos los puntos turísticos
- 2- Incorporar mayor número de puntos turísticos en la base de datos
- 3- Incorporar un sistema inteligente que genere rutas para cada usuario en función de sus gustos y preferencias.
- 4- Desde el punto de vista del frontend migrar a una visualización de la aplicación más atractiva, planteando un entorno Web.

Líneas abiertas.

Consideraciones personales.

Anexos

Listado de fuentes entregadas / Código fuente en GitHub

El código de GitHub aparece en el siguiente Link:

<https://github.com/g sanz/FinalProjectDiploma>

Para descargarlo habría que usar el comando git clone:

git clone <https://github.com/g sanz/FinalProjectDiploma>

Manual de usuario

El manual de usuario es un documento que se entrega de forma anexa.