

Tarea obligatoria de PMDM. Mapas

Contenido

1	Introducción.....	1
2	Duración	1
3	Objetivos.....	2
4	Recursos	2
5	Tarea.....	4
5.1	MainActivity.java	5
5.2	VistaLugarActivity.java.....	6
5.3	EdicionLugarActivity.java	8
5.4	MapaActivity	8
5.5	PreferenceFragment.....	9
6	Criterios de entrega.....	10
6.1	Apartado1. Interfaces	10
6.2	Mapas	11
6.3	Sistema de notificaciones	11
6.4	Base de datos SQLite	13
6.5	Refactorización	12
6.6	Opcionalmente para mejorar nota	15

1 Introducción

En esta tarea vamos a realizar el proyecto MisLugares, realizado hasta ahora a lo largo del curso

2 Duración

Tres semanas

3 Objetivos

Realización de una aplicación Android con:

1. Manejo de actividades, interfaces y estilos.
2. Manejo de colecciones Java.
3. Manejo de adaptadores
4. RecyclerView
5. Permisos
6. Preferencias
7. Mapas
8. Sistema de notificaciones
9. Base de datos SQLite
10. Refactorización
11. JDBC o Api REST

4 Recursos

Para realizar la tarea utilizaremos:

1. El gran libro de Android 7ª edición



2. Los apuntes del curso que están en el aula virtual:

Tema 1 <ul style="list-style-type: none">VocabularioIntroducción AndroidArquitecturaVeron claves e instaladorTutorial android. DirecciónConfiguración depuración hardwareEstructura de proyectos androidTemas/PantallasPlantilla de corrección examen	Tema 5 <ul style="list-style-type: none">Tema 5. Recycler view, fotos, almacenamiento, intenciones
Tema 2 <ul style="list-style-type: none">ANDROID PDILayoutsUnidades de MedidaEjemploLoginConstraintLayoutEjemploLoginConstraintCoordinatorLayoutConstraintLayout apuntesNuevo LoginIAIZAActividades Con Scrolling En AndroidVistas	Tema 6 <ul style="list-style-type: none">Los mapas en android
Tema 3 <ul style="list-style-type: none">Tema MiLugaresImágenesEntrega LayoutsMiLugares	Tema 7 <ul style="list-style-type: none">Base de datos SQLite en MiLugares
Tema 4 <ul style="list-style-type: none">Tema 4. Apoyo a tutorialmaterial para mi lugaresEjercicio Spinners	

3. Y el tutorial de la UPV sobre Android studio. Aquí tenéis el enlace:

<http://www.androidcurso.com/index.php/mooc/programa-mooc>

5 Tarea

Usaremos los apuntes del tutorial de la UPV sobre Android studio a partir del tema 3.
Obviaremos el tema 6 del tutorial.

Para la realización de la aplicación no necesitáis leer el tema 1.

Se deben entregar obligatoriamente las siguientes clases. Añadir casos de uso u otra metodología MVC no es obligatorio pero se felicita.

5.1 MainActivity.java

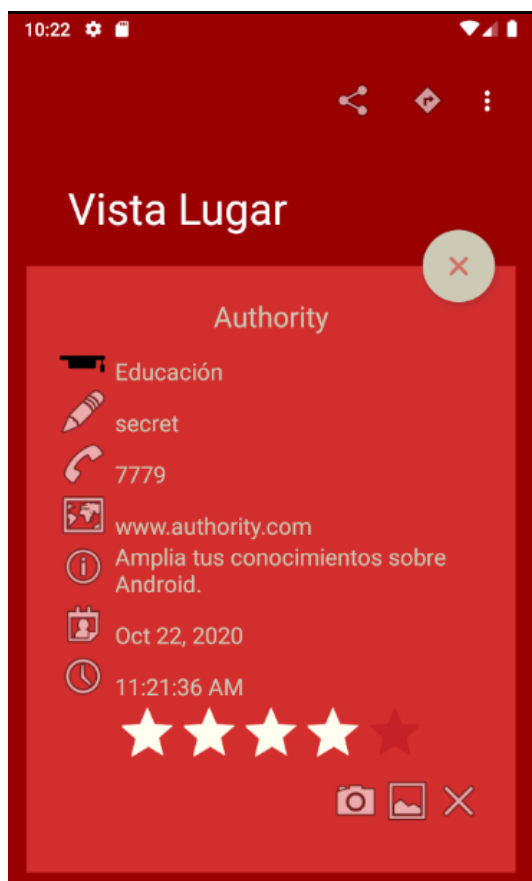
Esta **actividad nos permite visualizar el número total de lugares que tenemos**. Podemos acceder en el icono de la persona a los mapas. Opcionalmente nos permitirá sincronizar con Firebase en el icono upload. Pulsando sobre el lugar nos lleva a VistaLugarActivity

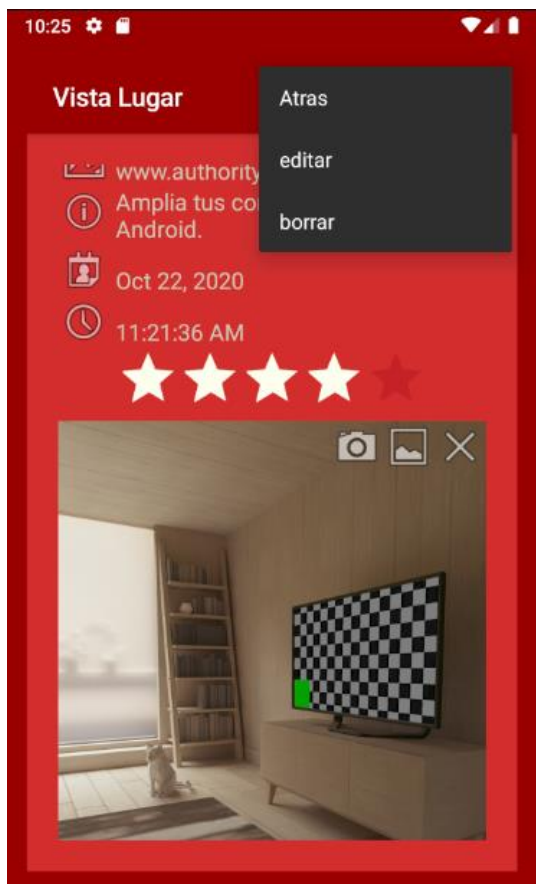
Pulsando sobre el botón flotante podremos añadir un lugar nuevo desde EdicionLugarActivity.



5.2 VistaLugarActivity.java

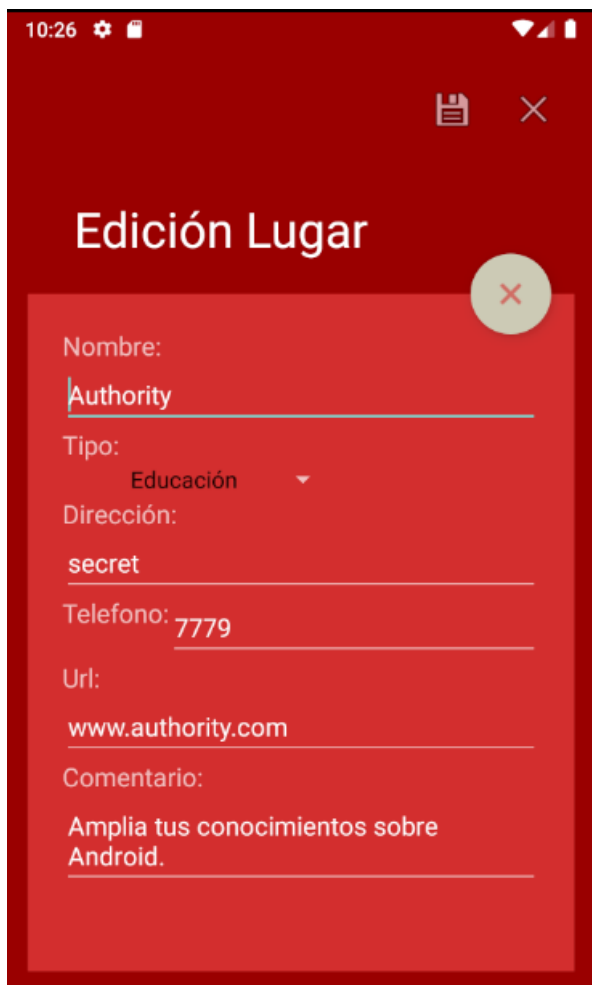
Esta **actividad** nos permite **visualizar los datos personales del lugar**. Pulsando sobre el botón X de cancel volvemos a la actividad principal. Pulsando en editar navegamos a EdicionLugarActivity. Pulsando en el icono de galería o de fotografía podemos añadir una foto.





5.3 EdicionLugarActivity.java

Nos **permite editar los lugares, modificarlos y guardarlos**. Igualmente nos permite añadiendo un nuevo lugar pulsando en el botón + desde la ventana principal. Pulsando en el floating buton cancel volvemos atrás.



10:26

Edición Lugar

Nombre: Authority

Tipo: Educación

Dirección: secret

Telefono: 7779

Url: www.authority.com

Comentario: Amplia tus conocimientos sobre Android.

5.4 MapaActivity

Nos **permite acceder a nuestros lugares a partir del mapa** de la API de Google maps. Al hacer doble click sobre el icono muestra la información relativa al lugar.



5.5 PreferenceFragment

Nos permite cambiar la configuración del lugar. Se accede desde el menú configuración de **MainActivity.java**.

Mandar notificaciones Notificar si estamos cerca de un lugar	<input type="checkbox"/>
Máximo de lugares a mostrar Limita en número de valores que se muestran en la lista	
Criterio de ordenación Que lugares quieres que aparezcan antes	

6 Criterios de entrega

6.1 Apartado 1. Interfaces

Elegid estilos y colores propios a vuestro gusto.

MainActivity: Ventana de entrada a la aplicación con **RecyclerView**, **menú**, **scroll** y **botón flotante**.

VistaLugar: Ventana de visión del lugar, **menú**, **scroll** y **botón flotante**.

EdicionLugar: Ventana de edición del lugar, **menú**, **scroll** y **botón flotante**.

Preferencias: Ventana de preferencias, **un fragment**

AcercaDe: Ventana de ayuda, **tipo diálogo**.

MapaActivity: para mostrar nuestros lugares en el mapa **con la API de Google Maps**.

6.2 Mapas

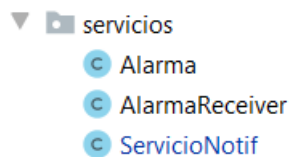
Controlaremos nuestra actividad en el mapa para mostrar nuestro programa usando la API de Google en **MapaActivity.java**

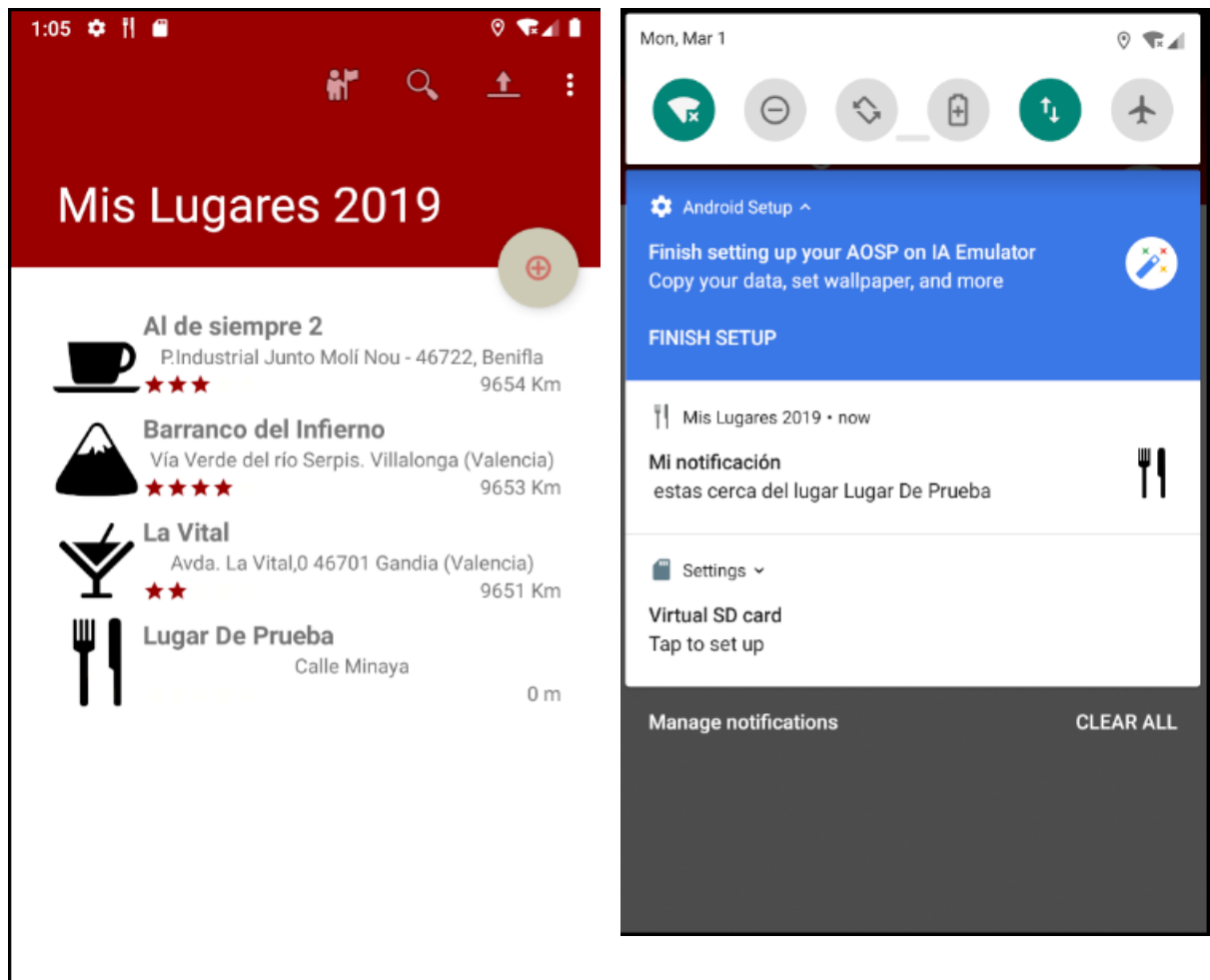
6.3 Sistema de notificaciones

Añadimos conforme a lo realizado en el curso un sistema de notificaciones, con una alarma que se dispara a los 10 segundos de arrancar la aplicación y que se activa cada 30 segundos.

Esta Alarma arranca un Broadcast receiver, que lanza un servicio.

El servicio mandará una notificación por cada lugar que este a menos de 100 metros de distancia de nuestra ubicación actual.





6.4 Refactorización

Aplicamos los principios sólidos de programación orientada a objetos aprendidos en clase para refactorizar el código obtenido.

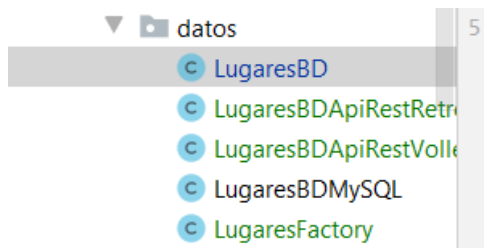
Seguimos las instrucciones dadas en clase.

1. La **clase adaptador no puede depender de una implementación concreta** que pertenece a otra clase, en este caso un Cursor de base de datos. Quitar el cursor del constructor y mandar los métodos que dependen de cursor a LugaresBD.
2. Eliminar una concreción dependiente del framework de Android de nuestro POJO.
 - a. **R.drawable ha de ser sacado de TipoLugar a una clase aparte. Yo la he llamado AdaptadorTipoLugarDrawable.**
3. *Introducir Casos de Uso o controladores.*
 - a. Introducir un paquete casosdeuso, con las clases CasosDeUso que serán nuestros controladores de aplicación.

- i. CasoDeUsoNavegacion: con métodos estáticos controlaremos toda la navegación de la aplicación. Los métodos onActivityResult se deben mantener en cada Actividad. Se resuelve pasando la actividad como parámetro.
- ii. CasoDeUsoLugar: con métodos estáticos realizaremos todas las operaciones sobre, usuario, guardado, modificación borrado, nuevo, y sobre las fotografías.
- iii. CasoDeUsoPermisos: crearemos métodos estáticos para manejar la petición de permisos. Pasad cada actividad como un Listener para el método onPermissionsResult que sea.
- iv. CasoDeUsoLocalización: para la gestión de ubicaciones y proveedor de Localización, usaremos este CasoDeUso, está en el libro. La gestión de permisos llevarla a CasoDeUsoPermisos.

6.5 Base de datos SQLite

Para mantener **nuestra persistencia de datos** añadimos una **base de datos SQLite** sobre **nuestro interfaz lugares**, de **manera transparente para el resto de elementos de la aplicación**. A la clase que maneja la base de datos e implementa el interfaz Lugares, le llamaremos LugaresBD, conforme al libro.



Cambiar el programa para que el interfaz lugares se adapte a la refactorización, usando el id para buscar los elementos siempre que sea posible.

```
public interface Lugares {
    final static String TAG = "MisLugares";
    static GeoPunto posicionActual = new GeoPunto(0,0);
    Lugar elemento(int id); // Devuelve el elemento dado su id
    void anyade(Lugar lugar); // Añade el elemento indicado
    int nuevo(); //Añade un elemento en blanco y devuelve su id
    void borrar(Lugar lugar); //Elimina el elemento con el id indicado
    int tamaño(); //Devuelve el número de elementos
    void actualiza(int id, Lugar lugar); //Reemplaza un elemento
```

```
public List<Lugar> listaLugares();  
Lugar getElemento(int pos );  
  
}
```

7 Mejoras

1. *Llevarlo a versión Android X. Haced una copia del proyecto para esta parte.*
2. Cambios en el modelo y SQLite
 - a. **Introduce un id en lugares. Aunque la clave se genera de manera automática recupera la clave conforme insertes.**
 - b. **Cambiar el TipoLugar en la base de datos SQLite a String.**

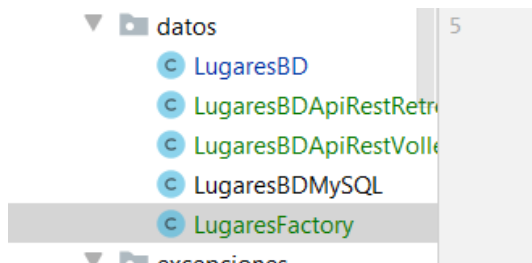
Librerías versión 30.

```
implementation 'androidx.activity:activity:1.1.0'  
implementation 'androidx.appcompat:appcompat:1.1.0'  
implementation 'androidx.constraintlayout:constraintlayout:1.1.3'  
  
implementation 'com.google.android.material:material:1.2.1'  
implementation 'androidx.recyclerview:recyclerview:1.1.0'  
implementation 'com.google.android.gms:play-services-maps:17.0.0'  
implementation 'com.google.api:api-common:1.10.1'  
  
implementation 'com.android.support:multidex:1.0.3'
```

7.1 Opcionalmente para mejorar nota

Añadimos un *acceso JDBC a una base de datos MySQL* para mantener nuestros datos. Intentamos que sea lo más transparente posible respecto al resto de nuestra aplicación

O con la REST API que os proporcione sobre un servidor Tomcat 9 realizamos un acceso a datos a MYSQL para nuestra aplicación usando la librería retrofit. El proyecto de API Rest se adjuntará junto a este documento.



Base de datos MYSQL. Valida tanto para MySQL como para MariaDB,

```
CREATE DATABASE IF NOT EXISTS `mislugares` ;  
ALTER DATABASE `mislugares` CHARACTER SET utf8 COLLATE utf8_general_ci;  
use mislugares;  
CREATE USER `lugares`@`%` IDENTIFIED BY 'lugares';  
  
GRANT ALL PRIVILEGES ON mislugares.* TO 'lugares'@`%` IDENTIFIED BY 'lugares';  
  
DROP TABLE IF EXISTS `lugares`;  
  
CREATE TABLE lugares (  
    _id INT(10) PRIMARY KEY AUTO_INCREMENT,  
    nombre VARCHAR(100),
```

direccion VARCHAR(200),
longitud FLOAT,
latitud REAL,
tipo VARCHAR(50),
foto VARCHAR(200),
telefono VARCHAR(20),
url VARCHAR(200),
comentario TEXT,
fecha BIGINT,
valoracion FLOAT);