

Tarea Tema 3

En esta tarea trabajaremos sobre lo aprendido en los tres temas de Android Studio. Vamos a crear una aplicación nueva llamada Alumnos que tendrá las características que os indicaré adelante a lo largo de la tarea

1 Duración

2 horas.

2 Objetivos

1. Aplica tecnologías de desarrollo para dispositivos móviles evaluando sus características y capacidades.
2. Desarrolla aplicaciones para dispositivos móviles analizando y empleando las tecnologías y librerías específicas.
3. Desarrolla programas que integran contenidos multimedia analizando y empleando las tecnologías y librerías específicas.

3 Tarea.

3.1 *Introducción*

Vamos a crear una aplicación llamada Alumnos, para la gestión de alumnos en el centro. De nombre al proyecto le llamareis ApellidosNombreAlumnos. La estructura del proyecto se detallará a lo largo de la tarea.

3.2 *Materiales*

Clases Java:

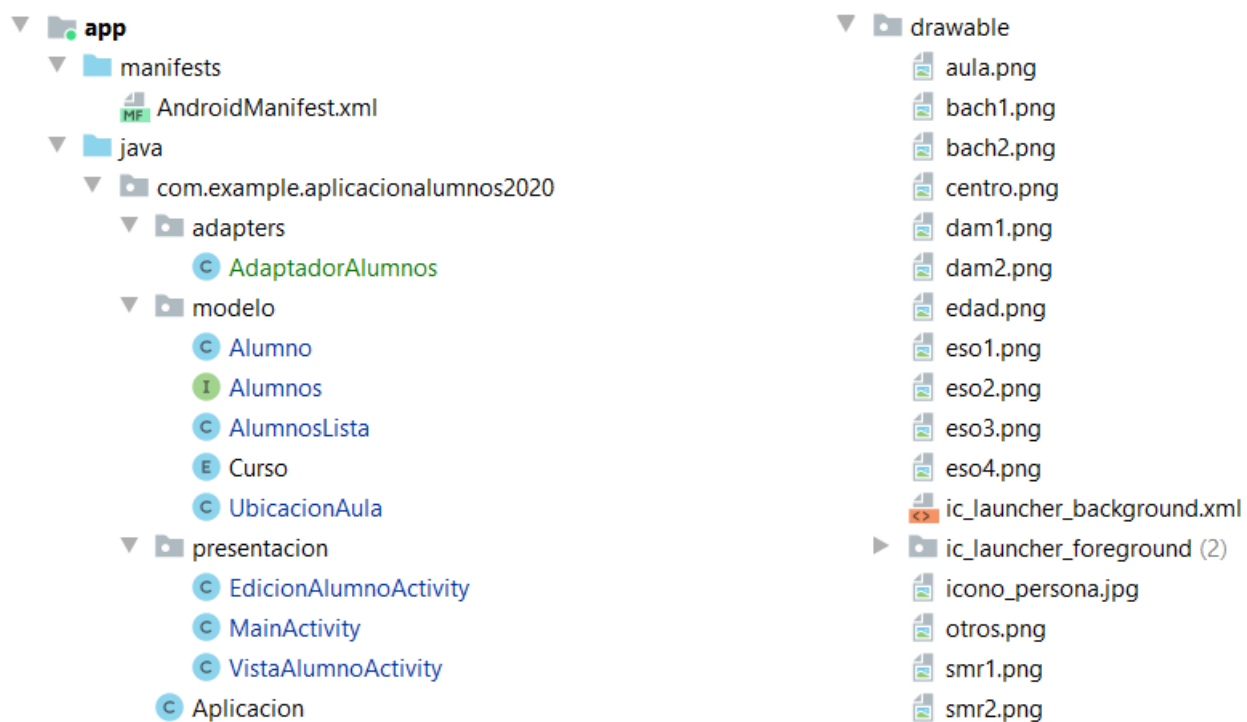
Se proporciona dos clases **Alumno.java** y **Aula.java**. **Curso.java**, de tipo enumerado y un interfaz **Alumnos.java**

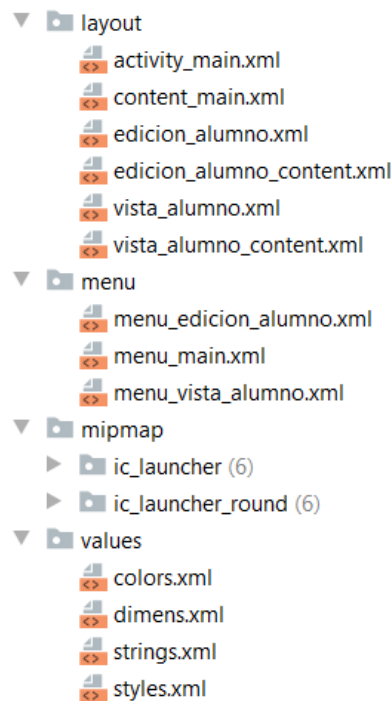
Imágenes:

Se proporcionará un zip, **iconos.zip** con todos los recursos que necesitareis para drawable.

3.3 Estructura del proyecto

En las siguientes imágenes podeis apreciar tanto la estructura de clases, de imágenes y layouts XML. Vuestro proyecto tiene que quedar exactamente igual que el que vereis a continuación.





3.4 Librerías

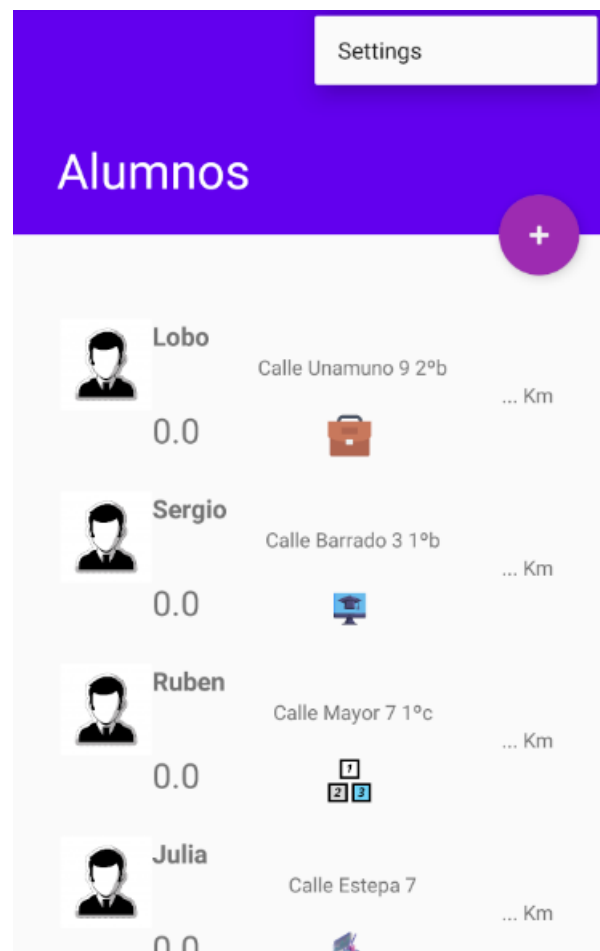
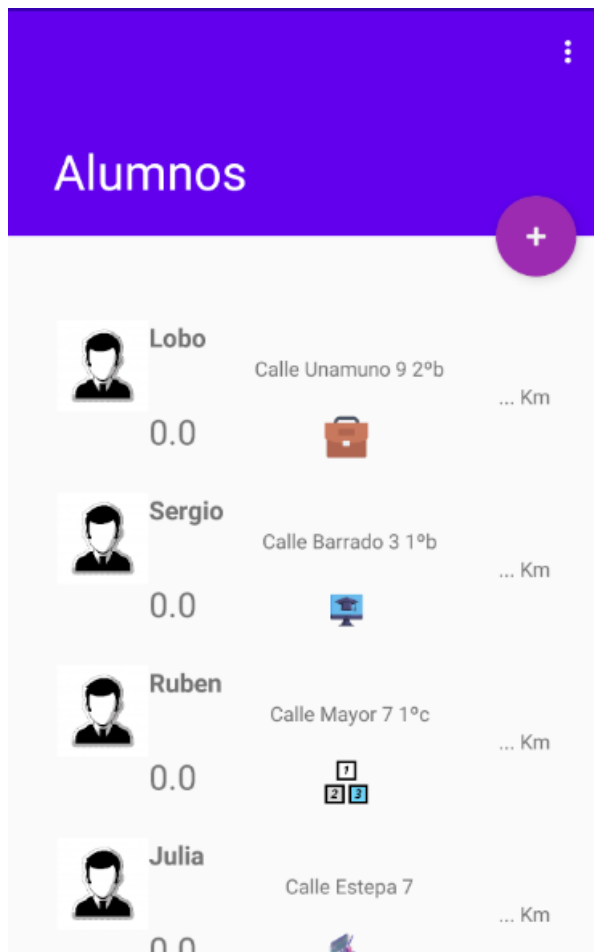
Esta versión de la aplicación la realizaremos en versión 26 como hemos visto en clase. Esta son las librerías base, necesitareis añadir el resto a necesidad de los componentes que se os vayan pidiendo.

```
implementation 'com.android.support:appcompat-v7:26.0.0'  
implementation 'com.android.support.constraint:constraint-layout:1.0.2'
```

3.5 Layouts

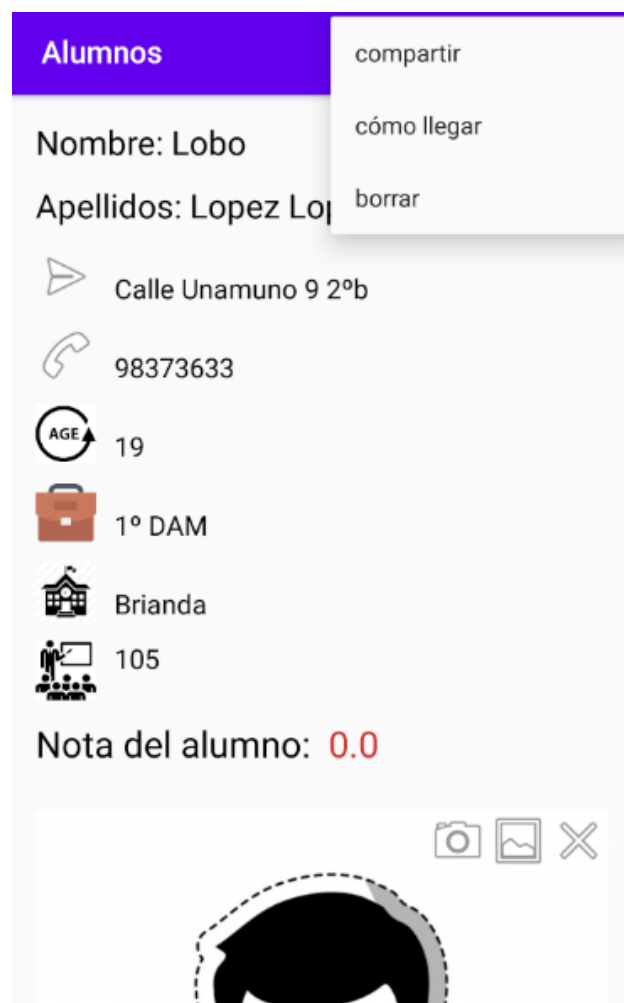
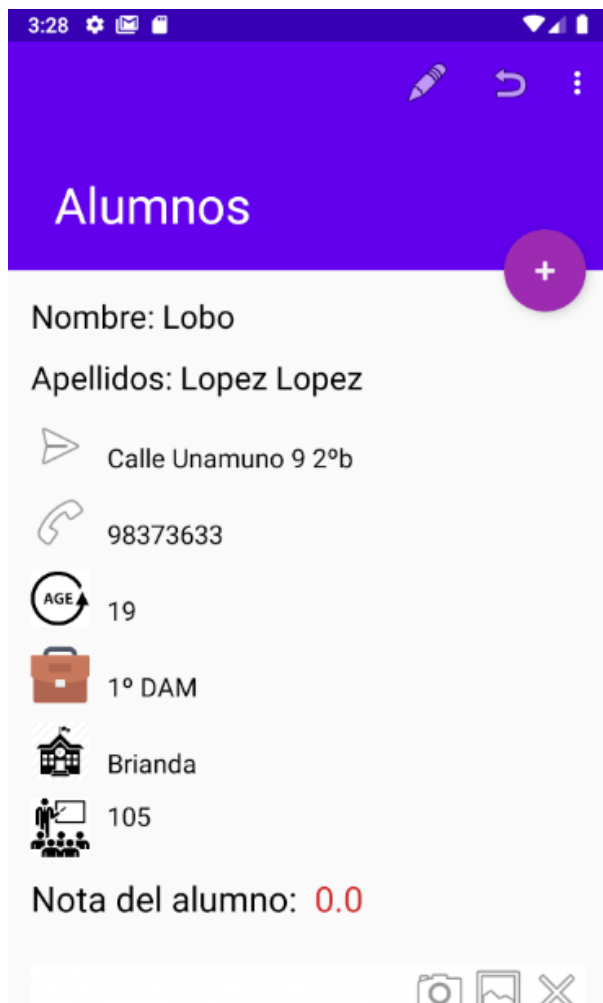
Definiremos el el layout de activity_main.xml como un Coordinator Layout con una barra de navegación colapsable, y un fab Button. El contenido será un recycler view como el que podéis ver a continuación. Cada elemento de la lista debe estar definido en alumno_lista.xml. Tenéis la opción de usar ConstraintLayout o RelativeLayout para realizar cada elemento de la lista.

activity_main.xml



Pulsando en cualquier alumno de la lista, la aplicación se dirigirá a visualizar los datos del alumno en VistaAlumnoActivity

vista_alumno.xml



Cuando se pulse al icono  volvemos a MainActivity. (Icono revert).

Colores: podéis elegir vuestra propia paleta. Respetar el rojo de la nota.

3.6 Criterios de calificación en la lógica del programa

Clases

Aplicacion.java: de tipo Application: mantiene los datos de los alumnos y permite compartirlos por todas las actividades. Nos proporcionará las herramientas para obtener una referencia a la Colección AlumnosLista.

AlumnosLista.java: Mantiene una estructura de datos de tipo colección con una lista de objetos de la clase Alumno. Implementa el Interfaz alumnos.

MainActivity.java: carga el recycler view con los datos de AlumnosLista. Cuando pulsamos en un elemento de la lista, nos redirige a VistaAlumnoActivity. Pasaremos como parámetro a VistaLugarActivity la posición en la colección del alumno. Activity de tipo AppCompatActivity.

VistaAlumnoActivity.java: muestra los datos para un alumno. Recoge la posición en la lista del alumno y obtiene el alumno de la colección de alumnos por posición. Activity de tipo AppCompatActivity.

EdicionAlumnoActivity.java: No se realiza en este examen, pero dejar la clase vacía. Activity de tipo AppCompatActivity.

1. Debéis crear una **estructura de almacenamiento de tipo colección a elegir para los alumnos** y añadir al menos 3 de ellos. Esta estructura de datos se compartirá por todas las actividades a través de la clase Aplicación. La clase aplicación será de tipo Application **(2 puntos)**
2. Cargar el RecyclerView **(2 puntos)**
3. Mostrar los datos del alumno en AlumnoActivity **(2 puntos)**.
4. Incluir y pedir permisos de almacenamiento externo en VistaAlumnoActivity.java. **(1 punto)**
5. Layouts y vistas correctamente diseñadas y que se ajusten a lo que os muestro **(2 puntos)**.
6. Limpieza y claridad en el código. Uso de Orientado a Objetos de manera correcta **(1 punto)**

Es obligatorio que el programa funcione hasta donde lleguéis para obtener la nota mínima 5.

4 Entrega

Entregareis en un **apellidosnombreTareaTema3.zip** con el proyecto y un fichero **apellidosnombreTareaTema3.pdf** con las pantallas.