

Práctica 2.

Analizar y modificar aplicación
existente.

Resultados de aprendizaje y criterios de evaluación.

CE1f. Se ha analizado la estructura de aplicaciones existentes para dispositivos móviles identificando las clases utilizadas.

CE1g. Se han realizado modificaciones sobre aplicaciones existentes.

1. Analizando una aplicación.

Se proporciona una aplicación sencilla que gestiona una simple agenda telefónica, en la que se dispone de una pantalla principal con las entradas de las personas, un simple buscador y un botón para añadir nuevas entradas.

El repositorio en GitHub es el siguiente:

https://github.com/pass1enator/PMDM_T1_P2_template

Responder a las siguientes preguntas:

1. ¿Qué clases se encuentran en la carpeta o paquete “model”? ¿De qué se encarga cada clase?
2. ¿Qué se puede encontrar en la carpeta res? ¿Y dentro de la carpeta layout? Abrir por ejemplo activity_main.xml
3. ¿Qué clase se encuentra relacionada con activity_detail.xml? ¿En qué parte del código de la clase se relacionan ambos?
4. Existe un ListView modificado para mostrar la lista de contactos, investigar que clase permite personalizar las entradas del listado, ¿de quién hereda? ¿Qué métodos se han tenido que implementar para poder usar la clase? ¿Cómo se definen los métodos en la clase padre?
5. ¿Qué método de la clase anterior se encarga de crear una nueva entrada en el listado?
6. En algunos puntos del código se tiene métodos del tipo setOn..., normalmente setOnClickListener. Explicar qué son este tipo de métodos y qué patrón UML implementan.
7. Existe una carpeta o paquete llamada DI, de Denpendecy Injection o inyección de dependencias en castellano. ¿Qué problema resuelve la inyección de dependencias? Un ejemplo es el código de la clase MainActivity donde se tiene:

```
@Inject  
lateinit var personasrepositor: PersonasRepository
```

8. ¿Cómo se obtiene/crea el objeto personasrepository?
9. En el código se usan palabras iniciadas por @ ¿Qué son? ¿En qué fases se utilizan?

2. Modificando el código.

Las modificaciones del código son sencillas, no se han de crear clases, ni conocer la API de Android.

10. La búsqueda no funciona correctamente, se ha de modificar el código para que al introducir un texto se filtre las personas cuyo nombre o apellido o teléfono o mail contenga la palabra introducida.
11. Si se borra el texto de búsqueda, se ha de volver a la situación anterior.
12. Además, se ha de poder borrar sobre los datos filtrados, actualizándose en el listado.
13. También se tiene el problema de que se pueden crear nuevas entradas, pero al seleccionar editar, la actividad aparece sin los datos, idear alguna solución sencilla (modificar la clase PersonareRepository) para que al pulsar editar, se puedan ver los datos.
14. Cambiar el color primario por verde, se ve entre otros elementos en los botones.
15. Por último, al modificar un contacto el programa no funciona bien y en vez de modificar el existente, añade un nuevo, solucionar el problema.

3. Entrega.

Se realizará en Aules

16. Parte 1: Respuesta a las preguntas con tus palabras, justificando y razonando las mismas.
17. Parte 2: Explicación del problema encontrado, posibles soluciones, solución aportada y fragmento de código.

Proyecto comprimido y sin compilar en Aules (cuando esté disponible)