Prácticas para Carlos Daniel de 2º DAM 23/24 para Multimedia.

Profesor: Santiago Rodenas Herráiz.
Correo electrónico: srodher115@g.educaand.es

Programación multimedia y dispositivos móviles. (2.º DAM-A)

Prácticas

Te dejo una relación de prácticas que deberás ir entregando a lo largo de este trimestre.

1. **Desarrollar** una App para Android Studio con kotlin, donde te permita acceder desde un Activity con dos botones, a su respectivo Activity, del cual deberás pasar para el primer botón tu nombre junto a tus apellidos y para el segundo botón con destino al segundo Activity un número de teléfono. Desde ambos Activities, podrás volver mediante un botón de Atrás al primero. Deberás utilizar un diseño xml lo más sencillo posible. Para el paso de datos, tendrás que pasarle un Bundle con los valores. Utiliza una fuente de datos de tamaño grande.

2. **Desarrollar** una App para Android Studio con kotlin, con un botón que al pulsarlo, deberá informar del número de teléfono que aparece junto a cualquier otro texto que desees, por ejemplo “llamaremos al teléfono 666-666-666”, utilizando el sintetizador de voz utilizado en clase. Aparte de informar, te deberá de llamar por teléfono. Tienes que aceptar los permisos de llamada telefónica. Toda la información, la tienes en la plataforma dentro del primer tema.

3. **Desarrollar** una App para Android Studio con kotlin, donde incorpores dentro de una lista mutable un conjunto de objetos del tipo **película**. Dicha película estará formada por el título, el año, una breve descripción y una imagen que corresponderá con una url

de una imagen capturada de internet. Dicha lista, deberás mostrarla previamente por consola, utilizando **Logs**. En esta aplicación, te pido que utilices un **RecyclerView** junto a un **AdaptadorPelicula** y **ViewHolderPelicula.** Tu aplicación, aparte de mostrar en pantalla los datos, deberá mostrarte un listado dentro de tu Activity. Para las imágenes, utiliza **Glide**.

4. **Amplía** la aplicación anterior, para que permitas la **eliminación** de cualquier película de tu RecyclerView. Para ello, añade un botón de **borrado** dentro de tu **ViewHolderPelicula** y mediante paso de funciones y llamadas de orden superior, al seleccionar una película para eliminar, deberás borrar dicha película. Te aconsejo que de momento, todas las acciones a los botones, las hagas dentro de tu Activity. Cuando te crees tu Adaptador y después tu ViewHolder, recuerda pasarle la función borrar y desde el mismo ViewHolder, debes invocar al método cuyo cuerpo de la función deberás desarrollar desde el mismo Activity. Llamadas a funciones de orden superior.

5. **Amplía** tu aplicación, para incorporar un segundo Activity con el **login**. Dicho login, serán dos EditText y un botón validar. De momento, para validar deberás comprobar en código tu usuario y contraseña cuyo valor deberás tener dentro de tu fichero de recursos String. Si la validación es acertada, deberás lanzar el Activity anterior mediante un Intent.

6. **Desarrolla** una aplicación Android Studio con kotlin, en el que crees **5 fragmentos** cualesquiera. Necesitarás un Activity con un contenedor de fragmentos y un botón siguiente | anterior. Por defecto, te deberá mostrar el primer fragmento. Los 4 primeros fragmentos, deberán tener un botón siguiente y los últimos cuatro fragmentos un botón anterior. Deberás utilizar un Navigation Component, para que mediante un gráfico de navegación, puedas representar la navegación de tus 5 fragmentos. Deberás asociar rutas que enlazan los destinos de la forma que desde el fragmento 1 puedas navegar al fragmento 2 con el botón siguiente y viceversa con el botón anterior. Desde el fragmento 2, podrás ir al fragmento 3 con su botón de siguiente y al fragmento 1 con su botón anterior. Así sucesivamente hasta el fragmento 5, que sólo podrás navegar con su botón anterior al fragmento 4. En cada fragmento, deberás representar una imagen y un texto correspondiente a cualquier historia, por ejemplo **escenas del Quijote**. Recuerda utilizar la opción de navigate mediante el NavController. Dichas acciones, las tienes a partir del gráfico de navegación.

7. **Modifica** tu aplicación del **punto 5**, haciendo que todo tu recyclerview en la UI lo incorpores dentro de un fragmento. En vez de desarrollar toda la lógica en un Activity, lo harás dentro de un Fragmento. Tu Activity sólo tendrá un contenedor de fragmentos, que de manera estática lanzará el fragmento que te muestre el listado de películas.

8. **Modifica** tu aplicación del **punto 7**, para crear un **DialogFragmen, en donde puedas** insertar una **nueva película.** Dicho diálogo, deberá contener cada uno de los campos de texto a introducir correspondiente al título de la película, la descripción, el año y un campo de texto para insertar la url de la imagen. Te aconsejo que incorpores un botón flotante en el fragmento donde tienes tu RecyclerView. Dicho botón, deberá crearte un DialogFragment en el que debes de recoger los datos. Cuando detectes el botón de añadir una película, créate el diálogo al que deberás pasar una función lambda, que sea invocada desde el mismo diálogo en caso afirmativo y que dicha invocación corresponda con una llamada de orden superior que deberás tratar en el mismo fragmento. Dicha invocación a la función pasada como argumento en el Diálogo, deberá mandar la nueva película. Repito, desde el fragmento, al invocar al botón y antes de crear el Diálogo, deberás hacer algo como **val dialog-> DialogNewFilm() { film -> okNewFilm(film)}** y desde el mismo Diálogo, cuya función **onNewFilmDialog: (Film) → Unit** recibimos en su constructor, en la parte **setPositiveButtom**, deberás invocar a dicha función recibida en el constructor del Diálogo, de la forma **onNewFilm(newFilm)**. Recuerda que **newFilm**, debes crearlo dentro del Diálogo con los datos que captures.

9. **Modifica** tu aplicación del punto 8, para adaptarte a un **patrón de diseño mvvm**.

10. **Modifica** tu aplicación del punto 9, para incorporar una persistencia mediante **Room**. Tu Dao, deberá listar, insertar y eliminar.