Programación Multimedia y Dispositivos Móviles
PROYECTO BLOQUE 2
Obligatorio

1. Introducción

En este documento se presentarán las opciones de proyecto para finalizar el **bloque 2** del módulo de **Programación Multimedia y Dispositivos Móviles**. Deberás seleccionar una de las siguientes opciones disponibles e indicarlo en la encuesta que se abrirá en el aula virtual para indicar la opción elegida. Habrá un número máximo por cada una de las opciones, por lo que deberás plantearte una alternativa.

2. Requisitos del proyecto

Todas las opciones de proyecto planteadas en este documento tienen la misma (más o menos) dificultad, realizar una aplicación móvil que cumpla con los apartados vistos durante el curso. A continuación se detallan los puntos a tratar.

- Widgets vistos (*EditText*, *ViewText*, *Toast*, *SnackBar*, *RecyclerView*, etc).
- *Intents* (implícitos y explícitos).
- Diálogos.
- Uso de más de una *activity* y/o *fragments*.
- Gestión de permisos.
- Implementación de menús.
- Persistencia de datos (*SQLite* y *Firebase*).
- Mapas y posicionamiento.
- Diseño de las UI sencillas y comprensibles.

Se busca conseguir una aplicación funcional y sin errores, **recuerda que una aplicación que no compile, será evaluada con una nota de cero**. A continuación, se detallarán las opciones disponibles y una breve descripción de las funcionalidades de cada una de las aplicaciones. Habrá un número máximo de alumnos que podrán realizar la misma aplicación.

3. Opciones de proyecto

A continuación, se detallan los posibles proyectos a presentar, en base a la descripción dada de cada una de las opciones, desarrollarás la aplicación libremente, cumpliendo los requisitos indicados.

- ***MyAmazingPlaces***: el objetivo de esta aplicación es almacenar tus lugar favoritos, mostrando un listado completo de todos ellos. La aplicación deberá ofrecer las opciones necesarias para añadir, modificar y eliminar, así como ver los detalles y la ubicación en el mapa.

  Entre los datos a guardar estará el tipo de sitio (restaurante, monumento, museo, etc), el nombre, teléfono si tiene, ubicación y/o dirección y la posibilidad de añadir una foto, ya bien sea desde la cámara o desde la galería, una valoración mediante estrellas (de 1 a 5) y su página web si tiene. Tanto el número de teléfono, la página web como la dirección, permitirán lanzar un *intent* con la tarea a realizar.

Añade al menos un menú con opciones para ordenar el listado mostrado. Por orden alfabético mediante el nombre y otra ordenación por valoración de las estrellas.

- ***MyFavouritesPets***: esta aplicación se encargará de guardar todas tus mascotas favoritas, permitiendo añadir, modificar o eliminar mascotas.

  De cada mascota se almacenará su nombre, nombre científico, tipo de pelaje (corto, largo, lanudo, etc), su clase (ave, mamífero, anfibio, etc), una imagen, ya bien sea mediante cámara o añadida desde la galería, y su nivel, del 1 al 5, de *"amorosidad"*. También deberás añadir un enlace a una página web que amplíe información sobre la mascota (Wikipedia por ejemplo), permitiendo abrir el enlace desde la propia aplicación.

  En el propio listado deberás dar la opción para marcar mascotas como favoritas y añadir un menú para ordenar el listado de diferentes formas, alfabéticamente por el nombre, otra por el nivel de *"amorosidad"* y una última opción que permita ver solo las mascotas marcadas como favoritas o todas.

- ***MyLittleShoppingList***: esta aplicación permite guardar no solo la lista de la compra, si no que deberá guardar también el supermercado, tienda, mercado… donde comprar, del cual se guardará el nombre, la dirección y la página web y teléfono si tiene. Desde el detalle del establecimiento se podrá visitar la página web si tiene, llamar por teléfono si dispone y consultar su ubicación en el mapa.

  Cuando se cree una lista de la compra, deberá indicarse para que establecimiento es y un nombre de lista. Dicha lista contendrá todos los productos a comprar, indicando su nombre, precio si se conoce, y cantidad. La lista deberá mostrar en todo momento el importe conocido de la lista, es decir, de aquellos productos que tengan precio. Además, cada uno de los elementos de la lista deberá contener un *check* para saber si ya lo has cogido o no en el momento de la compra.

  Una vez finalizada la lista, deberá existir una opción que permita finalizar la lista, pasando a un histórico. Deberás añadir un menú a la aplicación que permita ver los establecimientos, las listas activas y el histórico. Tanto los establecimientos como las listas, se añadirán y se podrán eliminar.

- ***MyDeliciousCandies***: esta aplicación se encargará de guardar todas tus golosinas favoritas, permitiendo añadir, modificar o eliminar las golosinas.

  De cada caramelo se almacenará su nombre, fabricante, tipo de dulce (caramelo, chicle, masticable, etc), su formato de venta (individual, empaquetado, etc), una imagen, ya bien sea mediante cámara o añadida desde la galería, y su nivel, del 1 al 5, de dulzor. Además, deberás añadir la página web del fabricante, permitiendo que se pueda visitar la página web desde la aplicación.

  En el propio listado deberás dar la opción para marcar gominolas como favoritas y añadir un menú para ordenar el listado de diferentes formas, alfabéticamente por el nombre, otra por el nivel de dulzor, otra opción que permita ver las gominolas por fabricante y una última opción que permita ver solo las gominolas marcadas como favoritas o todas.

- ***MyFabulousRecipes***: esta aplicación tiene como objetivo almacenar recetas de cocina. Deberá permitir al usuario añadir, modificar o eliminar recetas.

  De las recetas se guardará el nombre de la receta, una breve descripción y una foto del plato. Además, deberán añadirse los ingredientes con sus cantidades y los pasos de su elaboración, tanto los ingredientes como los pasos se deben poder eliminar y/o modificar. También deberás añadir información sobre el nivel de dificultad de elaboración de la receta y el tiempo total de elaboración (aproximado). Además de toda esta información, deberá poder añadirse un enlace a una página web que haga referencia a la receta o similar.

  Añade un menú a la aplicación que permita ordenar el listado de diferentes formas, alfabéticamente por el nombre de la receta, otra por el nivel de dificultada y una más por el tiempo que se tarda en elaborar la receta.

- ***MyWonderfulMoneyControl***: con esta aplicación se pretende llevar un control de gastos diario. Desde la pantalla inicial se podrán añadir movimientos a un listado, cada movimiento tendrá un título (compra pan, almuerzo, etc), deberá llevar la fecha (por defecto la del momento de creación, pero puede cambiarse), el tipo de movimiento (ingreso o gasto), una categoría (alimentación, bares y restaurantes, ocio, museos, etc) y el importe. También se debe poder añadir una foto, para guardar el ticket, así como la posibilidad de añadir la ubicación de ese gasto o ingreso.

  La aplicación deberá mostrar un menú donde se pueda ver el listado diario, que será la pantalla de entrada, una opción para ver un resumen de gastos por meses/año (listado, por ejemplo enero-2020, febrero-2020, …, enero-2021, …) y otra opción para ver el resumen de gastos total por categoría.

  Las categorías deberán añadirse desde la aplicación. Deberá existir una opción para acceder a un listado de categorías, desde donde se podrá añadir, modificar o eliminar categorías

- ***MyParkingControl***: esta aplicación te permitirá registrar la ubicación del aparcamiento de tu coche. Mediante la opción de registro de la posición, cogerá tu posición actual y guardará las coordenadas, la fecha y hora de aparcamiento.

  La aplicación mostrará en un listado todos los aparcamientos, indicando, mediante colores, si el coche está aparcado en ese sitio o no, solo podrá existir un aparcamiento activo. Al seleccionar un elemento del listado, se pasará al detalle, donde se podrá indicar que ya no estás aparcado, registrando la fecha y hora de salida (también se podrá ejecutar esta acción mediante una pulsación larga del elemento de la lista), así como la opción de ver en el mapa la posición y añadir una foto del aparcamiento. Los aparcamientos se podrán eliminar si se desea.

  La aplicación controlará que si registras un nuevo aparcamiento, pero no has desaparcado de otro, marcará como desaparcado el último registrado.

  Además, deberá tener las opciones correspondientes para mostrar el listado de aparcamientos y otra (opcional) en la que se muestre sobre el mapa todos los puntos en los qua se haya aparcado.

Por último, dentro de la vista detalle de un aparcamiento, tendrás la opción de compartir la ubicación con alguien mediante correo electrónico.

Analiza bien los requisitos de las aplicaciones para realizar un planteamiento eficiente de la base de datos, algunas de las aplicaciones tendrán un esquema más complicado que otras. No se utilizarán ficheros para almacenar la información principal de la aplicación, aquella relacionada con la temática de la aplicación.

4. Configuración del proyecto

Se creará un proyecto en **Android Studio** con API mínima **21** utilizando **Kotlin** como lenguaje de programación, cuyo nombre será el título de la aplicación seleccionada. Una vez creado el proyecto, dentro de la sección ***java***, crea la siguiente estructura:

- El paquete principal *com.example.myfantasticapp*, contendrá la clase principal y los controladores de las vistas.
- El paquete *com.example.myfantasticapp.**model***, para almacenar el modelo, o modelos aplicados a la aplicación.
- El paquete *com.example.myfantasticapp.**utils***, para almacenar todas aquellas clases útiles que puedan ser necesarias.
- El paquete *com.example.myfantasticapp.**adapters***, para almacenar todos los *adapters* necesarios para la aplicación.

> **NOTA**: sustituye *example* por tu nombre y primer apellido, y *myfantasticapp* por el nombre de tu aplicación.

5. Opciones optativas de mejoras

Todas las aplicaciones que hacen uso de mapas, pueden resolverse simplemente abriendo la ubicación en una aplicación de mapas. Como parte opcional, puedes integrar la API de ***Google Maps*** o ***Mapbox*** en la aplicación, añadiendo posicionamiento, para visualizar las opciones.

Como segunda mejora opcional, añade una opción de *backup* sobre ***Firebase***, de forma que cada vez que se aplique, vuelque la base de datos local sobre *Firebase*. Para simplificar el proceso, cada vez que se use la opción, se creará una colección, cuyo nombre será el nombre de la aplicación, fecha (ddmmaaaa) y hora (hhmmss), con todos los datos de la base de datos, por ejemplo *MyFabulousRecipes-23112021114856*.

Deberás tener en cuenta que *Firebase* es una base de datos documental, por tanto, cada registro (documento) deberá contener toda la información.

6. Reglas de evaluación

6.1. Parte obligatoria

Para obtener la calificación final del proyecto se aplicarán los siguientes criterios de evaluación:

- Estructura y organización del proyecto, esto implica la creación de los *packages* necesarios para la organización del código fuente del proyecto. Mínimo deberán existir los *packages model*, *utils* y *adapters*. *0,75 puntos*.
- Creación y organización de los *widgets* vistos durante el curso. Se evaluará la simplicidad del diseño, funcionalidad y buena organización de las UI. *0,5 puntos*.
- El uso de listados para mostrar información, teniendo mayor consideración al uso del *RecyclerView* para mostrar la información. *1 punto*.
- Uso de *intents* implícitos y gestión de permisos. *2 puntos*.
- Navegabilidad, o usabilidad. La aplicación debe ser de uso fácil, estableciendo un sistema de navegación entre pantallas claro, ya bien sea mediante el uso de *activities*, *fragments* o *Navigation*, valorándose de menos a más. *1,5 puntos*.
- Implementación de menús, valorándose el uso inteligente de los mismos y la utilidad. *1,5 puntos*.
- Persistencia de datos mediante la aplicación de bases de datos *SQLite*. Se tendrá en cuenta la consistencia y adaptación de la BD a los requerimientos de la aplicación. *2,5 puntos*.
- Mostrar mensajes de error y/o información cuando no se pueda hacer algo (por ejemplo, guardar un registro sin toda la información necesaria, cancelar un proceso sin guardar, etc). *1 punto*.
- Uso de diálogos para confirmar y/o cancelar acciones (por ejemplo, eliminar un registro). *1,5 puntos*.
- Documentación, limpieza y eficiencia del código. *1 punto*.

6.2. Parte optativa

La implementación de las partes optativas podrán mejorar la nota final del proyecto con hasta 2 puntos:

- Si obtienes menos de 9,5 puntos en la parte obligatoria, la nota máxima será de 10.
- Si obtienes al menos 9,5 puntos en la parte obligatoria, la nota máxima será de 11.

7. Entrega

Se entregará el proyecto realizado en **Android Studio** utilizando el lenguaje de programación **Kotlin** y API mínima 21. El formato del paquete deberá ser **edu.nombreapellido.proyecto**.

Una vez terminado el proyecto, deberás enviar el directorio completo del proyecto Android Studio comprimido, antes de ello, limpia el proyecto para que ocupe menos espacio, utiliza la opción ***Build > Clean Project***. El fichero deberá estar en formato .ZIP.