

# App Companion

La siguiente práctica pretende que el alumno desarrolle un App Companion para demostrar los conocimientos adquiridos en la asignatura.

Para la realización de la práctica se requiere que el alumno use:

- Kotlin como lenguaje de programación.
- Android Studio como IDE de desarrollo.
- Se permite el uso de IA y código externo siempre y cuando se haga constar las partes exactas en las que se ha aplicado código no propio. El alumno es responsable del código generado por lo que errores, malas prácticas y problemas de rendimiento podrán afectar negativamente a la nota.

## **En caso de entrega no funcional la nota máxima será un 4.**

Se entiende como no funcional aquella entrega que:

- En su ejecución presenta fallos que finalicen la ejecución del programa de forma inesperada y de forma reiterada.
- Aquella entrega que no es posible ejecutar.
- No se obtiene al menos un 4 en cada uno de los dos apartados genéricos.
- No se ha realizado o no es funcional el 50% de cada apartado.
- Code review no aplica a los 2 puntos anteriores.

## **Contenidos Android (5 puntos)**

### **Navegación (3 puntos)**

La aplicación deberá de contener mínimo 6 actividades diferentes y distintos sistemas de navegación. Debe de permitir una navegación completa e intuitiva y una forma de cerrar la aplicación que no requiera de forzar su cierre.

#### **Algunos ejemplos de activities:**

- Splash Screen.
- Configuración.
- Login/Registro.
- Perfiles.
- Settings.

#### **Algunos ejemplos de navegación:**

- Abrir nueva activity.
- Navigation bar.
- Back button en la top bar.

### UI (3 puntos)

La UI debe de presentar una buena experiencia de usuario y seguir unos buenos patrones de diseño, por lo que se pide:

- Desarrollar un diseño propio que no sea el default de todas las aplicaciones y que ofrezca feedback al usuario.
- Usar las views correctas según la necesidad de la aplicación, así como usar unas buenas constraints.
- La UI debe de funcionar correctamente tanto en posición *portrait* como *landscape*.
- La aplicación debe de ofrecer mínimo dos diseños diferentes elegibles por el usuario.

### **API - Confirmar la API por correo (4 puntos)**

La aplicación debe de consumir los datos de una API que requiera el uso de un token de seguridad para ser consumida. Se tendrá en cuenta la usabilidad de la aplicación y el control de errores con la API. La aplicación mínima debe de permitir hacer consulta sobre un elemento en concreto, obtener resultados que coincidan con algún parámetro y mostrar imágenes obtenidos de la API.

- La aplicación ofrece la posibilidad de buscar información de acuerdo a un parámetro genérico de búsqueda, por ejemplos un historial de todas las partidas que se realizaron durante una semana en concreto.
- La aplicación ofrece la posibilidad de buscar sobre un elemento en concreto dado su identificador, por ejemplo en el caso anterior seleccionar una partida y a través de su ID consultar los jugadores que participaron en ella.
- La aplicación muestra imágenes ofrecidas por la API, por ejemplo en el caso anterior al lado del nick de cada jugador se muestra su imagen de perfil.

## Contenidos Firebase (5 puntos)

### Autenticación (4 puntos)

La aplicación debe de ofrecer un login convencional de Usuario y contraseña y al menos un método de autenticación a través de algún proveedor. Además, debe de permitirle modificar su perfil.

- El usuario puede registrarse y loguearse mediante un Usuario y Contraseña.
- El usuario puede registrarse y loguearse mediante un proveedor.
- El usuario puede desloguearse de la aplicación.
- Si el usuario no se desloguea manualmente la aplicación debe de mantenerle el login cuando se vuelva a abrir.
- La aplicación tiene alguna activity que muestra al usuario su perfil y le permite editar algunos valores.

**Database (4 puntos)**

Se almacena información en una base de datos para ser usada posteriormente por la aplicación. Se requiere usar al menos en 1 ocasión la base de datos para consumir la información en tiempo real y otra ocasión para consultas ocasionales.

- Se puede insertar información a la base de datos.
- Se da un uso a la base de datos en tiempo real.
- Se da un uso ocasional a la base de datos.
- La aplicación sigue siendo funcional en caso de que la información tarde en obtenerse y da feedback de ello.

**Analytics/Crashlytics (2 puntos)**

La aplicación debe de obtener información relevante para ser analizada, así como los crasheos que haya. El día de la entrevista se pedirá justificación de como la información obtenida es relevante para mejorar la usabilidad de la aplicación.

- La aplicación lleva una analítica de al menos 2 elementos diferentes que permiten mejorar la usabilidad de la aplicación.
- La aplicación lleva un control de todos los crasheos que se dan.

### **Code Review - Penaliza sin límite**

El no aplicar las buenas prácticas de programación supondrá una penalización sobre la nota final.

- No usar el tipo correcto de variable: **-0.2 por variable.**
- No mantener constancia en la nomenclatura de variables: **-0.2 por variable.**
- No comentar el código o mal comentar el código: **-1 punto.**
- Existencia de magic numbers: **-0.2 por magic number.**
- Existencia de variables sin uso: **-0.2 por variable.**
- Presentar graves errores de rendimiento: **-0.5 puntos.**
- No usar correctamente el repositorio git o sin git flow. **-1 punto.**
- No aplicar programación modular ni encapsular el código. **-1 punto**



## **Entrega**

- Para la entrega el alumno deberá de subir un archivo .txt en el campus con un enlace al repositorio git usado.
- Se permite el uso de github y gitlab. Dar acceso a [alex.perez@enti.cat](mailto:alex.perez@enti.cat)
- Opcionalmente el proyecto podrá contener un documento llamado Readme.txt en el que el alumno podrá añadir las observaciones que vea oportunas para la corrección del trabajo.
- En caso de entrega en grupo es imprescindible que entreguen ambos y que en la raíz del proyecto haya un .txt llamado grupo con el nombre y apellidos de todos los miembros que lo forman.

**ES RESPONSABILIDAD DEL ALUMNO QUE EL PROYECTO FUNCIONE Y SEA ACCESIBLE.**