

# **Proyecto Desarrollo Avanzado de Software**

segunda parte

# Requisitos:

- 1 Uso de Listviews personalizados o de RecyclerView + CardView para mostrar listados de elementos con diferentes características.
- 2 Usar una base de datos local para listar y añadir elementos
- 3 Uso de diálogos.
- 4 Usar notificaciones locales.
- 5 Controlar que la aplicación se comporte correctamente aunque en medio de la ejecución recibamos una llamada telefónica, giremos el móvil, etc. No se puede impedir que la aplicación se gire.
- 6 Documentar la aplicación mediante comentarios en el propio código.
- 7 Permitir que una misma funcionalidad se comporte de manera distinta dependiendo de la orientación (o del tamaño) del dispositivo mediante el uso de Fragments
- 8 Hacer la aplicación multiidioma
- 9 Uso de ficheros de texto
- 10 Uso de Preferencias, para guardar las preferencias del usuario en cuanto a mostrar/esconder cierta información, elegir colores para la aplicación, o cualquier otra cosa relacionada con la visualización de la aplicación
- 11 Crear estilos y temas propios, para personalizar fondos, botones, etc
- 12 Usar intents implícitos para abrir otras aplicaciones, contactos, etc.
- 13 Pantalla de login (y registro), para guardar credenciales de usuario (nombre de usuario – contraseña)
- 14 Uso de una base de datos remota para el registro y la identificación de usuarios.
- 15 Uso de mensajería FCM.
- 16 Captar imágenes desde la cámara, guardarlas en el servidor y mostrarlas en la aplicación.

Nota: en esta entrega se han depurado los errores reportados en la anterior entrega y añadido las funcionalidades mínimas de esta segunda entrega (14, 15 y 16)

# Introducción:

La App propuesta como solución es Ofertea, esta cuenta con las siguientes características para satisfacer los requisitos propuestos:

Muestra ofertas desde un fichero JSON local en su apartado catalogo, dicho catálogo constituye un 'fragment' con diferentes formas de visualizarse en función del dispositivo y su orientación. Esta lista es personalizada ya que cuenta con iconos para indicar que son objetos pinchables los que se muestran, por tanto con esta característica se espera cumplir los requisitos (9, 1, 7 y 11).

La aplicación cuenta con un punto de acceso / registro de usuarios el cual se realiza contra una BD remota, todo acceso se gestiona contra esta mediante el servicio 'gestorUsuarios.php' y permite insertar /consultar usuarios, por tanto con esta característica se espera cumplir los requisitos (2, 13 y 14).

Para el control de los términos de uso de la App se utilizan diálogos, una de sus opciones (la ,neutral) mostrará en el navegador mi Curriculum , por tanto con esta característica se espera cumplir el requisito (3 y 12).

Para la gestión de productos favoritos se utilizan notificaciones para retro alimentar al usuario de las acciones realizadas, posteriormente se trabajará un sistema de referencias cruzadas con dicha información, asimismo estas están gestionadas mediante una llamada al servicio 'gestorMensajeria.php' que hace uso de Google firebase, por tanto se espera cumplir los requisitos (4 y 15) .

Para la gestión de imagenes relacionadas se utiliza el servicio 'gestorImagenes.php' , este almacena imagenes en el servidor y guarda sus rutas en la BD remota, al iniciar cualquier producto se carga una miniatura por defecto de una de dichas imagenes remotas para mostrarla en la parte superior de la vista individual de producto (solo implementado en modo NO 'land'), con esto se espera cumplir el requisito (16).

En cuanto a la navegación se gestiona tanto el botón 'Back' del móvil como los distintos iconos dispuestos para esta tarea, y cada interfaz está preparada para interactuar con los distintos eventos del S.O, por tanto con esta característica se espera cumplir el requisito (5).

En cuanto a preferencias, la aplicación cuenta con una actividad que la gestiona, permite cambiar el idioma y el estilo visual de la misma, para la gestión interna de estas características se utilizan preferencias, por tanto con esta característica se espera cumplir los requisitos (8 y 10).

En cuanto a estilos se proveen dos estilos personalizados para la aplicación AppTheme y AppTheme1, posteriormente se añadirán otros seleccionables, por tanto con esta característica se espera cumplir el requisito (11).

En cuanto al código de la aplicación se encuentra disponible de manera pública en gitHub totalmente comentado y dispuesto de manera consistente para su entendimiento, por tanto con esta ,característica se espera cumplir el requisito (6).

# Clases

La aplicación cuenta con las siguientes clases, la indentación denota las clases internas:

- DummyContent: Esta clase gestiona la carga la lista de ofertas desde un formato JSON.
- Item: Esta clase representa el objeto oferta dentro de la aplicación y gestiona su información .
- ItemDetailActivity: Esta clase gestiona la vista de los detalles de un artículo en modo individual.
- ItemDetailFragment: Esta clase gestiona el comportamiento de los componentes de la listview y el View Holder en modo pantalla dividida.
- ItemListActivity: Esta clase gestiona de manera general la vista del catálogo de ofertas , delegará el la clase ‘SimpleItemRecyclerViewAdapter’ la gestión del contenido de la lista y el modo de mostrarla.
- SimpleItemRecyclerViewAdapter: Esta clase decide si la interacción con la lista debe ser en pantalla dividida o en vista simple.
- ViewHolder: Esta clase gestiona el contenido de un item en pantalla dividida.
- LoginActivity: Esta clase gestiona el acceso y registro de los usuarios a la aplicación.
- UserLoginTask: Esta clase soporta un método asíncrono de conexión a la BD para la autenticación de usuarios.
- MainActivity: Esta clase constituye el menú principal de la App y dirige la navegación.
- MyBD: Esta clase gestiona la interacción con la BD local de la App.
- SettingsActivity: Esta clase gestiona las preferencias de usuario en la App.
- FCM: Esta clase gestiona la recepción de mensajes FCM desde Google firebase.
- GeneradorConexionesSeguras: Esta clase gestiona las peticiones remotas al servicio web.
- ImagedownloadTask: Esta clase gestiona la descarga asíncrona de una imagen del servidor.
- ImageUploadTask: Esta clase gestiona la subida asíncrona de una imagen al servicio web.
- UserMessageTask: Esta clase gestiona la notificación remota asíncrona de un nuevo producto favorito.

Nota: El código de la App <https://github.com/EdgarAndresSantamaria/Ofertea> cuenta con comentarios mediante los que entender en profundidad cada clase.

# Manual de usuario

Para acceder a la funcionalidad de la aplicación deberá primeramente aceptar los términos de uso y seleccionar su configuración, posteriormente desde el menú desplegable situado en la parte superior izquierda de la pantalla principal podrá acceder a toda la funcionalidad de la App. Actualmente existen las funcionalidades de perfil, catalogo y preferencias, que permiten acceder / registrarse, visualizar las ofertas existentes y configurar la aplicación respectivamente.

Para hacer uso de funciones avanzadas (favorito y cámara) deberá accederse a la vista individual de un producto del catálogo en modo NO 'land' y hacer uso de los iconos de aquella vista, la cámara nos permite acceder a la galería del móvil para subir una foto al servidor y el corazón (también disponible en modo 'land') notificará de manera remota el evento de añadir un producto a favoritos, distinto desenlace en función de haber entrado o no en la App.

Nota: la pantalla principal también cuenta con explicaciones de los distintos componentes que componen la App (aún en desarrollo)