

# Entrega Desarrollo de Aplicaciones Android

**Nombre de la aplicación:** Recipe Manager

**Autor:** Jorge Sanzo Hernando

**Link de descarga:** <https://play.google.com/apps/testing/com.isanzo97.recipemanager>

## Descripción funcional:

La aplicación desarrollada para esta entrega consiste en la creación de un gestor de recetas, el usuario podrá registrarse en el sistema, iniciar sesión, consultar, eliminar recetas y crear nuevas recetas, también podrá cambiar distintos ajustes de la aplicación.

Primero vemos la pantalla de inicio de sesión, donde se pedirán permisos de ubicación y que el usuario active su GPS. Para registrarse en el sistema hay que rellenar los campos requeridos y presionar el botón de registrarse, si no se produce ningún error veremos un mensaje confirmando el registro.

Una vez registrados podremos iniciar sesión presionando el botón de iniciar sesión tras haber introducido nuestro nombre de usuario y contraseña, si todo ha salido bien el móvil mostrará una notificación en la barra de notificaciones dando la bienvenida al usuario a la aplicación y mostrando la ubicación geográfica desde la cual está iniciando sesión, si no se conceden permisos o no se activa el GPS no se mostrará dicha notificación.

Después de iniciar sesión llegamos a la pantalla principal donde tendremos nuestra lista de recetas, al principio aparecerá vacía si el usuario es nuevo, pero si ya hemos creado alguna receta previamente, aparecerá aquí con las opciones de ver más detalles sobre ella o de eliminarla.

Arriba a la derecha tenemos un botón en la barra de herramientas para cerrar nuestra sesión y a la izquierda un botón para abrir el menú desplegable.

En este menú desplegable podremos acceder a otras pantallas como la de crear nueva receta y los ajustes, también tenemos un botón para cerrar sesión desde aquí

En la pantalla de creación de receta deberemos rellenar todos los campos con la información necesaria y tendremos dos botones uno para borrar todos los campos y volver a empezar en caso de que hayamos puesto alguna información equivocada y otro para crear nuestra nueva receta.

Por último, tenemos la pantalla de ajustes donde podemos configurar dos parámetros de la aplicación, por un lado, el uso de datos, donde podremos elegir si actualizar los datos de la aplicación con llamadas al servidor a través de internet siempre que queramos ver nuestras recetas o hacerlo solo una vez cada 24h y utilizar la base de datos local del teléfono para evitar el gasto de datos. El otro parámetro que podremos configurar es el tiempo que permanecerá nuestra sesión abierta, pudiendo seleccionar periodos de 12h, 24h y hasta 48h, también podremos elegir nunca pero tendremos que iniciar sesión cada vez que cerremos la aplicación

## Descripción técnica:

Se ha utilizado el patrón MVVM, Clean Architecture y el patrón Repository para implementar el código de la aplicación, también se ha dividido la en varios módulos para aislar y encapsular el código, los módulos son:

- **App:** Contiene las vistas, organizadas cada una en su paquete, los modelos y los viewmodels asociados a cada una de las vistas y todo lo relacionado con la inyección de dependencias, que ha sido gestionado con la librería Koin.

- **BuildSrc:** Contiene un único fichero con todas las librerías y sus versiones utilizadas en la aplicación.
- **Common:** Contiene las interfaces de uso común en toda la aplicación, en este caso solo hay una única interfaz para implementar un Logger personalizado y utilizarlo para hacer debug durante el desarrollo.
- **Common-Android:** Implementa las interfaces definidas en el módulo Common, contiene todas las extensiones definidas para los distintos tipos de datos y define un viewmodel especial para gestionar la barra de herramientas de la parte superior de la aplicación en las diferentes vistas.
- **Domain:** Define todos los modelos de datos que se utilizan, los casos de uso que desarrolla la aplicación y también define las interfaces del patrón Repository para el acceso a datos tanto locales como remotos.
- **Data:** Implementa las interfaces definidas en el módulo Domain para poder utilizar el patrón Repository.
- **Local y Remote:** Ambos módulos implementan el patrón DAO de acceso a datos, el módulo Local gestiona el acceso a la base de datos local del teléfono móvil y el módulo Remote gestiona el acceso a los datos que se encuentran en el servidor.

## Librerías utilizadas:

- **io.arrow-kt:arrow-core-data (versión 0.10.0):** esta librería nos permite crear objetos de tipo Option y Either, los objetos de tipo Option solo pueden tener un valor, del tipo A o None y los objetos de tipo Either pueden tener un valor de tipo A o de tipo B. Es muy útil para las llamadas de red, por ejemplo, si pedimos las recetas del usuario la respuesta podría ser un error o una lista de recetas, así que usaríamos Either. En el caso de una llamada de red que no devuelva nada, por ejemplo, al borrar una receta, usaríamos Option para gestionar la respuesta, siendo sus posibles valores None si todo ha ido bien o un objeto de tipo error si no.
- **org.koin:koin-core (versión 2.0.1):** esta librería se ha utilizado para gestionar las dependencias dentro de la aplicación, se utilizó como alternativa a Dagger 2 ya que es algo más simple y sencilla y se organiza por módulos, donde se declaran las dependencias de cada uno, lo cual encaja muy bien con la arquitectura de la aplicación. La principal diferencia es que Koin monta las dependencias en tiempo de ejecución y Dagger 2 en tiempo de compilación, aunque esto hace que Koin tenga un tiempo de compilación más rápido al no usar elementos de tipo annotation, y que Dagger 2 tenga mejor tiempo de ejecución ya que todo ha sido preprocesado al compilar.
- **com.jakewharton.timber:timber (versión 4.7.1):** esta librería es una alternativa para escribir logs por consola, es equivalente a utilizar la función Log.d(tag, message) que existe por defecto en Kotlin pero nos ahorra el tener que poner un tag en cada log pudiendo hacer Timber.d(message)
- **com.jakewharton.threetenabp:threetenabp (versión 1.2.1):** esta librería se ha utilizado como alternativa a java.time, aunque es prácticamente igual muchas de sus funciones son más accesibles e intuitivas e incluye algunas funciones nuevas sobre todo para formatear fechas, o para conseguir un timestamp y pasarlo a segundos (Instant.now().epochSeconds()) que ha sido para lo que más se ha utilizado en este desarrollo.

No se han utilizado ninguna APIs Deprectaed en el desarrollo de la aplicación, tampoco está basada en ninguna otra aplicación previamente existente en internet (salvo los iconos e imágenes incluidos que si han sido descargados de <https://www.flaticon.es/> , de <https://iconos8.es/> y de Google Imágenes) y tampoco se han utilizado frameworks externos, para la base de datos se ha utilizado la librería de Room y para las llamadas de red la librería de Retrofit

La aplicación cuenta con dos tipos de build diferentes, la de release que sería la de distribución al público y la de debug o desarrollo que incluye dos librerías muy útiles a la hora de comprobar las llamadas de red y el estado de la base de datos, estas librerías son:

- **com.github.ChuckerTeam.Chucker:library (versión 3.0.1):** esta librería muestra en forma de notificación todas las llamadas de red según se realizan, donde podemos ver el código de la llamada, la request realizada y la response obtenida con gran cantidad de detalles
- **com.amitshekhar.android:debug-db (versión 1.0.6):** esta librería nos permite abrir una página web en el navegador donde podemos observar el estado de todas las tablas y columnas de la base de datos del móvil mientras ejecutamos la aplicación

## Motivación del desarrollo:

La motivación del desarrollo de esta aplicación ha sido la de reutilizar el servidor previamente creado en la asignatura Tecnologías del Lado Servidor: Cloud Computing de este mismo master para darle un posible uso real teniendo una aplicación móvil como cliente, todas las llamadas de red de la aplicación son realizadas a ese servidor que actualmente se encuentra desplegado en Amazon Web Service.

## Puntos de mejora:

El siguiente paso en el desarrollo de esta aplicación podría ser el de incluir la opción de editar una receta previamente creada, permitir cambiar los colores de la aplicación o por ejemplo añadir la posibilidad de establecer una hora o fecha en cada receta a modo recordatorio para el usuario, para poder programar cierta receta para determinada fecha y recibir una notificación el día antes, por ejemplo.