Proyecto de Recetas

Nuevo proyecto

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

# Configuramos la aplicación

## Añadimos las dependencias necesarias

CONTROL + ALT + SHIFT + S

File – Proyect Structure

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Aplicamos los cambios

Texto

El contenido generado por IA puede ser incorrecto.

## Habilitamos el viewBinding,

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

No olvidar: Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

# Creación del modelo de datos

Vamos a crear una enumeración para determina las categorías de recetas

Primero creamos el paquete model

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Ahora voy a crear los Strings para que sea multiidioma

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Creo para los principales idiomas

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Y para china 😊

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Ahora para crear la enumeración

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

Asociamos cada elemento de la enumeración con un recurso definido en strings.xml

Texto

El contenido generado por IA puede ser incorrecto.

También creamos una enumeración para la Dificultad

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Creamos el data class para Receta.kt, como es una clase para almacenar datos la haremos como data classs y Kotlin nos proporciona automáticamente los métodos equals, hasCode, toString y copy. Y podemos añadir otros métodos para trabajar con los datos de las recetas.

Texto

El contenido generado por IA puede ser incorrecto.

Creamos la clase converters, Es **imprescindible** crear primero la clase Converters, porque Room necesita saber cómo guardar tus enum (Categoria y Dificultad) y el campo photo: ByteArray? en la base de datos.

Texto

El contenido generado por IA puede ser incorrecto.

Vamos a crear la clase RecetaDAO para decir como accedemos a los datos

Contiene funciones que sirve entre la aplicación y la base de datos, define como insertamos, borramos datos de la tabla recetas en Room.

Texto

El contenido generado por IA puede ser incorrecto.

Ahora creamos la base de datos local utilizando Room.

RecetaDatabase es como el **"motor de base de datos Room"**, y la app lo utiliza para obtener el DAO (recetaDao()) que contiene las funciones para acceder a las recetas almacenadas.

Texto

El contenido generado por IA puede ser incorrecto.

**Crear RecetaViewModel**

Actúa como puente entre la interfaz de usuario (Fragment) y el repositorio de datos (RecetaRepository)

Contiene un StateFlow que mantiene actualizada la lista de recetas en tiempo real

Observa los cambios de la base de datos a través del Repositorioo

Llama a las funciones del Repositorio para insertar, borrar, actualizar o recuperar las recetas.

Pero lo hace de un modo transparente.

Los fragment miran el StaeFlow del ViewModel y actualizan la interfaz automáticamente.

Texto

El contenido generado por IA puede ser incorrecto.

Si el usuario gira la pantalla o cambia entre fragmentos el ViewModel mantiene los datos.

**Crear el RecetaRepository**

El repositorio actúa como intermediario entre el ViewModel y la base de datos.

Texto

El contenido generado por IA puede ser incorrecto.

Primero creamos el item\_receta.xml que es un layout de una receta individual para mostrar en el RecyclerView.

Texto

El contenido generado por IA puede ser incorrecto.

Creamos el layout en res/layout/item\_receta.xml

Como tenemos el binding activado Android Studio generará automáticamente la clase ItemRecetaBinding ya que tenemos habilitado ViewBinding en el build.gradle. (lo pusimos al principio a true).

Utilizamos recursos para que los nombres aparezcan en el idioma correspondiente.

Texto

El contenido generado por IA puede ser incorrecto.

Ahora crearemos el RecetaAdapter

* Va a mostar una lista de recetas en un RecyclerView
* Va a detectar cuando se pulsa una receta

Texto

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Ahora debemos mostrar todas las recetas en una lista y hacer algo cuando pulsamoe en ellas.

Crearmos ListaRecetasFragment

**¿Qué debe hacer ListaRecetasFragment?**

1. **Inflar el layout** con FragmentListaRecetasBinding.
2. **Configurar el RecyclerView** y asociar el RecetaAdapter.
3. **Observar el StateFlow del ViewModel** con la lista de recetas.
4. **Reaccionar a los clics** (por ejemplo, navegar al detalle).

Primero deberemos hacer un layout llamado fragment\_lista\_recetas.xml que tenga al menos un RecyclerView

Solo necesitamos uno de momento ya que no vamos a mostrar varias listas independientes en la misma pantalla, como por ejemplo como aparece en Netflix o amazon, como recomendadas para ti, populares, etc. Así tendríamos por ejemplo varios RecyclerView dentro de un ScrollView

También si por ejemplo quisiera una lista de categorías y luego abajo filtradas las recetas tendría

<constraintLayout>

RecyclerView Horizontal 🡪 para las categorías

RecyclerView Vertical 🡪 para las recetas filtradas de cada categoría

<ConstraintLayout>

Antes de nada voy a subir el proyecto a git.

En la terminal miro si tengo git

Texto

El contenido generado por IA puede ser incorrecto.

Como hice un git init, lo elimino



Reviso la configuración de Android Studio

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Hacemos clic en Test para comprobar la versión y que AE lo detecta.

Creo el repositorio desde el Version Control que está arriba a la izquierda

Una captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.

Yo creo que esto es lo mismo que poner Git init en la terminal

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

Veo que estoy en la rama Master

Imagen de la pantalla de un celular con letras

El contenido generado por IA puede ser incorrecto.

Si hago un ls -Force en la terminal veo el .git. -Force para ver ocultos (force en minúsculas también funciona)

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Voy a crear un .gitignore para ignorar ficheros y carpetas en mi repositorio.

Los build contiene ficheros que se regeneran y ocupan mucho espacio. No tiene sentido versionarlos pues cambian cada vez que compilamos.

Los .idea , .iml son archivos de configuración de Android Studio y pueden cambiar entre equipos, no afectan al código fuente ni a la funcionalidad

Los de .gradle son temporales de configuración, y Gradle los puede generar automáticamente por lo que no es necesario almacenarlos.

Local.propierties contiene la ruta local al SDK de Android y como es distinta en cada equipo no debemos compartirla.

Los \*.jks o cualquier fichero de claves es privado y no debemos subirlo a ningún repositorio público pues contiene la clave de la firma de tu aplicación.

Puede que Android Studio lo cree al hacer el git

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

El contenido generado por IA puede ser incorrecto.

Lo abro y le pongo el resto.

**Hacemos el primer commit**

**Selecciono los ficheros**

Captura de pantalla de un celular

El contenido generado por IA puede ser incorrecto.