


# Recetas con login en Room


[Descargar estos apuntes](#)

## Ejercicio

A partir del ejercicio de recetas ya implementado en el [bloque 3 parte ViewModel](#), vamos a modificarlo de forma que ahora guardaremos las recetas en una Base de Datos gestionada mediante Room. La BD contendrá la tabla Recetas, **RecipeEntity** con su correspondiente interface **RecipeDao** para gestionar las funciones CRUD.

 **Aviso:** Tendremos que modificar levemente la entidad recipe para que ahora el string de la imagen sea de tipo Blob (revisa el Codelab de AgendaRoom para ayudarte en este ejercicio). Además le quitaremos el elemento **isLiked** que no tiene sentido asociado a una receta, para gestionar los likes correctamente nos crearemos otra tabla.

La otra tabla que crearemos será la asociada a la parte de logear un usuarios, que ya hemos usado en otros ejercicios y que nos servirá para entrar en la aplicación. El código base de usuarios lo puedes encontrar en [este enlace](#). El usuario estará representado, a nivel de room, por las clases **UsuarioDao** y **UsuarioEntity**. A la tabla usuario le añadiremos una lista de enteros para gestionar los likeds.

 **Aviso:** Room no gestiona directamente listas, por lo que se le tiene que aplicar un conversor de tipo lista a string, el código junto a otros necesarios se pasa en el siguiente [enlace](#) es el archivo *RoomConverters.kt* y que debes alojar en el paquete room, especificaciones para su uso las encontrarás en este [enlace](#). Tendrás que añadir el código que te convierta de lista a cadena y a la inversa.

## LoginScreen

La aplicación se iniciará con una pantalla de login que solo permitirá la opción de logearse, (por lo que deberás añadir datos de usuarios de ejemplo en la base de datos).

## RecipeScreen

Una vez logeados se pasará a la pantalla de recetas, que mostrará las recetas usando un HorizontalPager que teníamos hecho con anterioridad, aunque ahora le añadiremos una BottomAppBar y

una TopAppBar con los iconos que se ven en la imagen para gestionar las acciones de la App. Para inicializar la tabla recetas con valores, se usará el compain object que se pasa en la MainActivity.

## Gestión de los Like

Como nos habremos logeado con un usuario para entrar a la app, podremos hacer Like para ese usuario de manera que lo que haremos será añadir el id de la receta a la lista de enteros que tenemos en la tabla usuarios. De forma que los like serán personalizados para cada usuario, así que tendremos que realizar el código necesario para que esto se refleje al logearse con distintos usuarios.

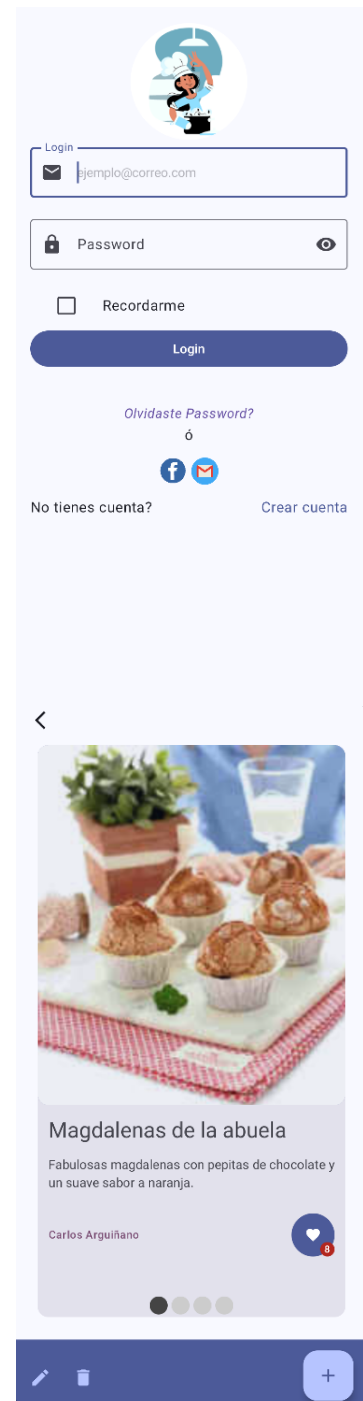
## Eliminar recetas

Al clicar sobre el icono de la papera alojado en la BottomAppBar, se eliminará de la Base de datos la receta que se está mostrando en ese momento, para controlar que elemento se está mostrando, puedes guardar en una variable de estado el valor de la receta que está activa en ese momento usando `currentPage` para indicar la posición.

## Añadir y modificar recetas

Para esta funcionalidad deberemos crear una nueva pantalla **NewRecipeScreen** con el aspecto de la última imagen de la derecha, tendremos una TopAppBar, un FloatingActionButton y el contenido principal con la imagen. En esta pantalla podremos gestionar tanto la inserción de nueva receta, como la eliminación de una receta existente. La única diferencia es que si se carga la screen mediante el icono de edición, tendremos que pasar el id de la receta para poder acceder a ella y mostrar los datos de esta.

**⚠ Aviso:** Para esta última pantalla deberemos crear tanto la clase Event como la clase ViewModel. Por otra parte los iconos de las dos TopAppBar, tendrán la funcionalidad de navegar atrás. **Tendremos que gestionar los id de las nuevas recetas introducidas, para conseguir un buen funcionamiento, no podremos crear el id como autogerado.**





Por último tendremos la funcionalidad de seleccionar una imagen de la galería al pulsar sobre la imagen de la cuchara y el tenedor, sustituyendo la imagen por defecto por la seleccionada.

