

Modificando Recetas

[Descargar estos apuntes](#)

Ejercicio

A partir del ejercicio de recetas ya implementado en este bloque, vamos a modificarlo de forma que ahora podremos visualizar más de una receta utilizando un componente específico para ello.

Primero deberemos modificar levemente el mock Recipe para que photo pase a ser un String? (que corresponderá al nombre de la imagen), deberemos de hacer el mismo cambio en el resto de elementos que representan a Recipe en las distintas partes de la arquitectura.

Lo siguiente será crear en RecipeDaoMock una lista de recetas en vez de una sola. Los datos para estas podrán ser los que se muestran a la derecha.

Como es de esperar, tendrás que cambiar RecipeViewModel de forma que ahora recipeState pasará a ser una lista de recetas extraídas del repositorio. Tendremos que modificar el método del evento, para que nos entre la receta sobre la que se ha hecho click, y en la funcionalidad deberemos trabajar sobre la receta pulsada de la lista.

```
private var recipes = mutableListOf<RecipeMock>(  
    RecipeMock(  
        id=1,  
        name = "Magdalenas de la abuela",  
        description = "Fabulosas magdalenas con pepitas",  
        chef = "Carlos Arguiñano",  
        photo = "magdalenas",  
        likes = 8,  
        isLiked = false  
    ),  
    RecipeMock(  
        id=2,  
        name = "Pan casero con tomates secos",  
        description = "Novedoso pan casero con un estim",  
        chef = "Carlos Arguiñano",  
        photo = "pan",  
        likes = 3,  
        isLiked = false  
    ),  
    RecipeMock(  
        id=3,  
        name = "Ensalada multicolor con nueces",  
        description = "Receta sencilla y rápida que dej",  
        chef = "Sonia de la Oz",  
        photo = "ensalada",  
        likes = 5,  
        isLiked = false  
    ),  
    RecipeMock(  
        id=4,  
        name = "Caldo de almendras, pasas y calabaza",  
        description = "Estupenda sopa vegetal de invier",  
        chef = "Mohamed Aldalay",  
        photo = null,  
        likes = 12,  
        isLiked = false  
    ),  
)
```

En cuanto a la parte de la vista, que es en la que vamos a realizar más cambios, vamos a crear una función composible nueva **CardRecipe** que tendrá el código que hasta ahora teníamos en RecipeScreen (ya que a esta función le llegará una sola receta y el evento sobre esta, que ya habíamos establecido en el anterior ejercicio). Como ahora las recetas tendrán distintas imágenes, tendrás que extraerlas de los **recursos** como hemos hecho en otros ejercicios y controlando si la imagen es null para asignar la imagen por defecto **Icons.Filled.Dining**.

En cambio a la función **RecipeScreen** le llegará la lista de las recetas y ese mismo evento, y será la que llame a CardRecipe con la receta correspondiente. Para realizar esto tendremos que añadir un componente **HorizontalPager** que nos permitirá la paginación entre las recetas con la llamada dentro de este a CardRecipe con la receta que corresponda a la página en transcurso (page 0 sería la primera). Conviene añadir este composible dentro de un Box que ocupe el máximo de alto.

Además añadiremos un **Indicador de página** que nos mostrará la página que estamos mostrando en cada momento como indicativo de las recetas que se pueden visualizar.



⚠️ Aviso: Cuando hayamos conseguido la funcionalidad de paginación correctamente, vamos a comprobar que el botón de Like no hace lo que esperamos. Esto es debido a aunque cambiemos un elemento del mutableList la lista no identifica que ha sido modificada, prueba a cambiar en el viewModel el tipo de tu lista a **MutableStateList** y verifica ahora el funcionamiento, en posteriores temas se presentará este tipo más profundamente.