

# Proyecto Usuarios con Arquitectura Ampliado

[Descargar estos apuntes](#)

## Ejercicio

Vamos a realizar una ampliación del ejercicio de Usuarios, de forma que ahora se podrá acceder, una vez nos hayamos logeado, a la publicación de recetas. Para ello deberemos añadir una opción más en el menú que ya teníamos, quedando de la siguiente forma:

```
1. Sign in
2. Login
3. Change password
4. Logout
5. Recetas
6. Volver
Introduce opción:
```

Si se selecciona la opción 5 y no estamos logeados, se mostrará un mensaje indicando el problema. En caso de estar logeados se accederá a la screen de las recetas que mostrará el siguiente menú:

```
1. Seleccionar/Deseleccionar receta
2. Imprimir selección
3. Mostrar todas
4. Volver
Introduce opción:
```

- Con la primera opción se podrá añadir o quitar una receta, de una lista de `RecetasUiState` (tipo que además de lo tiene `RecetaMock`, que podemos ver más abajo en el aviso, tendrá un valor Boolean *seleccionada*) que habrá en el `ViewModel`, la selección se hará por el id de la receta.
- La opción Imprimir selección mostrará las recetas que tenemos seleccionada, por ejemplo de la siguiente forma:

```
Magdalenas de la abuela de Carlos Arguiñano (40)
Pan casero con tomates secos de Carlos Arguiñano (120)
```

- Mostrar todas, mostrará la lista de todas las recetas del sistema, crea un formado de salida curioso, aunque no se muestren todos los datos.
- Volver nos devolverá al programa principal, que tendrá un bucle preguntando si queremos salir

```
...
1. Volver
Introduce opcion: 4
Quiere Salir [S|s]?

```

Si le decimos que S, saldremos de la aplicación. Cualquier otra tecla nos mantendrá en la aplicación mostrándose otra vez el menú de login.

**⚠ Aviso:** Deberemos crear los paquetes, clases, interfaces y funciones necesarias para añadir la parte de las recetas cumpliendo con la arquitectura ya explicada. La clase de las recetas tendrá el siguiente código:

```
data class RecetaMock(
    val id:Int,
    val cookingTime: Int,
    val recipeName: String,
    var recipePicture: String?,
    val chefName: String,
    val chefMail: String,
    var chefPicture: String?)

```

Puedes crear la lista de recetas en el lugar correspondiente, de la siguiente manera:

```
private var receta = mutableListOf<RecetaMock>(
    RecetaMock(
        id=1,
        cookingTime = 40,
        recipeName = "Magdalenas de la abuela",
        recipePicture = "imagen1.png",
        chefName = "Carlos Arguiñano",
        chefMail = "carlos.arguiñano@gmail.com",
        chefPicture = null
    ),
    RecetaMock(
        id=2,
        cookingTime = 120,
        recipeName = "Pan casero con tomates secos",
        recipePicture = "imagen2.png",
        chefName = "Carlos Arguiñano",
        chefMail = "carlos.arguiñano@gmail.com",
        chefPicture = null
    ),
)

```



```
RecetaMock(  
    id=3,  
    cookingTime = 10,  
    recipeName = "Ensalada multicolor con cacahuete",  
    recipePicture = "imagen3.png",  
    chefName = "Sonia de la Oz",  
    chefMail = "soniaoz@gmail.com",  
    chefPicture = null  
),  
RecetaMock(  
    id=4,  
    cookingTime = 35,  
    recipeName = "Caldo de almendras pasas y calabaza",  
    recipePicture = "imagen4.png",  
    chefName = "Mohamed Aldalay",  
    chefMail = "mohaaldala@gmail.com",  
    chefPicture = null  
),  
)
```