

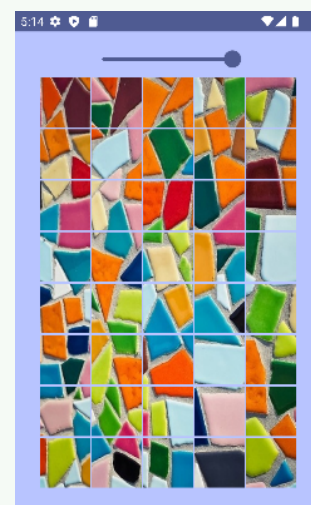
# Iniciandonos con ViewModel con ejercicio Mosaico

[Descargar estos apuntes](#)

## Ejercicio

Vamos a crear un ejercicio sencillo que nos sirva para iniciarnos con el concepto de ViewModel a la vez que practiquemos [FlowLayout](#) y [Slider](#). Para ello nos descargaremos las siguientes [imagenes](#) que añadiremos a la carpeta drawable de recursos.

El ejercicio trata de crear un composable **MosaicoScreen** que muestre la disposición de las imágenes usando un layout **FlowLayout**, este tipo de contenedor permite distribuir el contenido de forma fluida dependiendo del ancho de la pantalla o de la disposición del dispositivo (giraremos el dispositivo para ver su comportamiento con diferentes anchos), hasta un máximo de elementos establecidos. Para fingir los distintos tamaños de ancho, vamos a disponer de un **Slider** que modificará la variable del máximo elementos por fila.



El siguiente código permitirá crear una lista con el nombre de las imágenes que tendremos en drawable y que se deberán mostrar en el FlowRow.

```
val mosaico = mutableListOf<String>()

for (i in 1..8) {
    for (j in 1..5) {
        mosaico.add("fila_${i}_columna_${j}")
    }
}
```

Recuerda que para convertir el nombre en recurso tendremos que usar el siguiente código:

```

val contexto = LocalContext.current
val imageResource =
    contexto.resources.getIdentifier(urlImagen, "drawable", contexto.packageName)
...
Image( ...
    painter = painterResource(id = imageResource), contentDescription = "")
)

```

Para el Slider usaremos un mínimo de 1 y un máximo de ocho y convertiremos el float a int para poder usarlo en el FloRow.

Los estados deberán estar contenidos en el ViewModel y MosaicoScreen será llamada de dos formas para ver el alcance del ViewModel desde las dos perspectivas.

- Desde la Activity
- Desde un Preview

Para ver la diferencia, establece una cantidad de columnas y gira la pantalla en las dos formas de ejecución.

