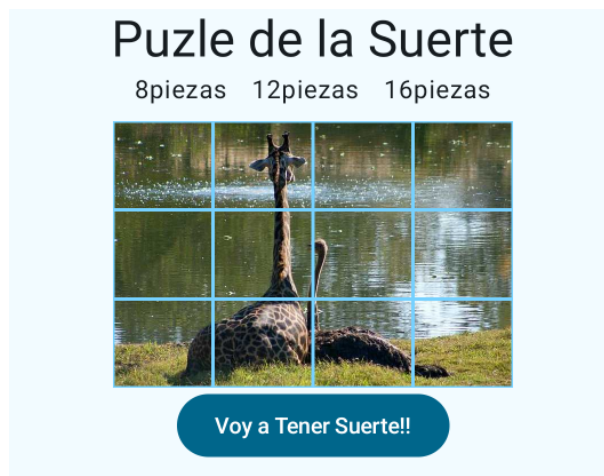


# FlowRow

[Descargar estos apuntes](#)

## Ejercicio

Vamos a comenzar una pequeña aplicación que muestre unas imágenes que componen un puzzle, podrán ser de diferentes tamaños (8, 12 y 16 piezas) y en un principio solo mostraremos el puzzle que corresponda con la selección de piezas. Para ello vamos a crear una pantalla con el siguiente aspecto:



Comenzaremos por crear el paquete features con el fichero **JuegoScreen.kt**, en el que construiremos todas nuestras funciones Composables. Que podrían ser las siguientes:

- **RectanguloImagen**, a la que le llega una string con el nombre de la imagen y un color y usando los componentes `Box` e `Image` deberás de conseguir el siguiente aspecto.

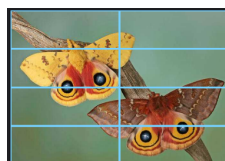


💡 **Tips:** la image la recuperaremos de los recursos con `painterResource`, para conseguir el id del recurso asociado al nombre de la imagen lo haremos con la siguiente línea

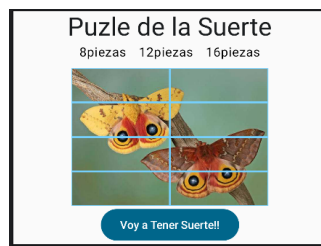
```
val imageResource = contexto.resources.getIdentifier(nombreImagen, "drawable", contexto.packageName)
```

el nombre de la imagen no llevará la extensión del tipo de archivo.

- **DistribuidorImágenes**, será la función que distribuya las imágenes que forman el puzzle, en forma de parrilla, dependiendo del número de piezas por fila con las que se quiera jugar. Para ello se usará un `FlowRow`, a esta función le llegará, como mínimo, la lista de los nombres de las imágenes, el color del borde y el número de filas en las que queremos distribuir el puzzle.



- **BasePantalla**, función a la que le llega la Lista de los nombres de las imágenes y algunos parámetros más que deberás de deducir para conseguir la funcionalidad y que mediante un contenedor `Column`, `Text`, `Row` y `Spacer` consiga el siguiente aspecto.



Para el botón, del que todavía no hemos visto el componente, podéis usar el siguiente código

```
Button(modifier = Modifier.align(Alignment.CenterHorizontally), onClick = { }) { Text(text = "Voy a Tener Suerte!!") }
```

, por ahora no le pondremos funcionalidad.