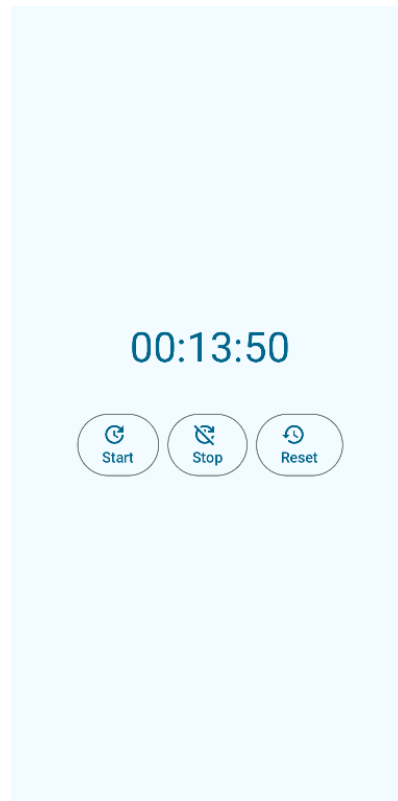


Ejercicio Cronómetro

[Descargar estos apuntes](#)

El ejercicio tratará de crear un nuevo proyecto Cronometro, en el cual la pantalla inicial tendrá el siguiente aspecto:

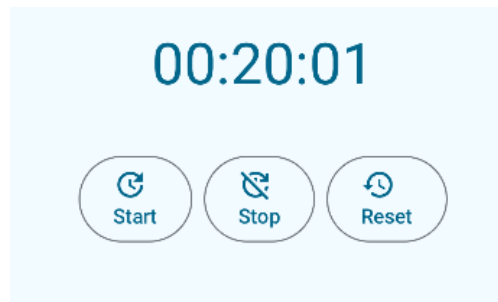


Cada botón tendrá el evento correspondiente a las acciones deseadas:

- Start, iniciará la visualización de los segundos, minutos y horas transcurridos desde su pulsación. Lanzando una corrutina que incrementará en un segundo un contador. Se pasa el código del método que transforma los segundos en el string de salida.

```
fun formateaSegundos(segundosEntrada: Int): String {  
    val hora = Integer.toString(segundosEntrada / 3600)  
    val minutos = Integer.toString(segundosEntrada % 3600 / 60)  
    val segundos = Integer.toString(segundosEntrada % 3600 % 60)  
    var stringBuilder = StringBuilder()  
    stringBuilder.append((if (hora.toInt() < 10) "0$hora" else hora).toString())  
    stringBuilder.append(":")  
    stringBuilder.append((if (minutos.toInt() < 10) "0$minutos" else minutos).toString())  
    stringBuilder.append(":")  
    stringBuilder.append((if (segundos.toInt() < 10) "0$segundos" else segundos).toString())  
    return stringBuilder.toString()  
}
```

- Stop pausará el cronómetro (cancelaremos la corrutina). Si pulsamos start una vez parado el cronómetro, seguirá el conteo desde el segundo en el que se había pausado.
- Reset cancela la corrutina y reiniciará los contadores a sus valores por defecto.



👉 **Importante:** Para este ejercicio crearemos un `CronometroScreen.kt` con los composables necesarios para construir la vista que se muestra en las imágenes, los estados se crearán en este mismo fichero (stateful), y las corrutinas se gestionarán dentro del `onClick` de los botones. Por lo que no será necesario crear el `ViewModel`.