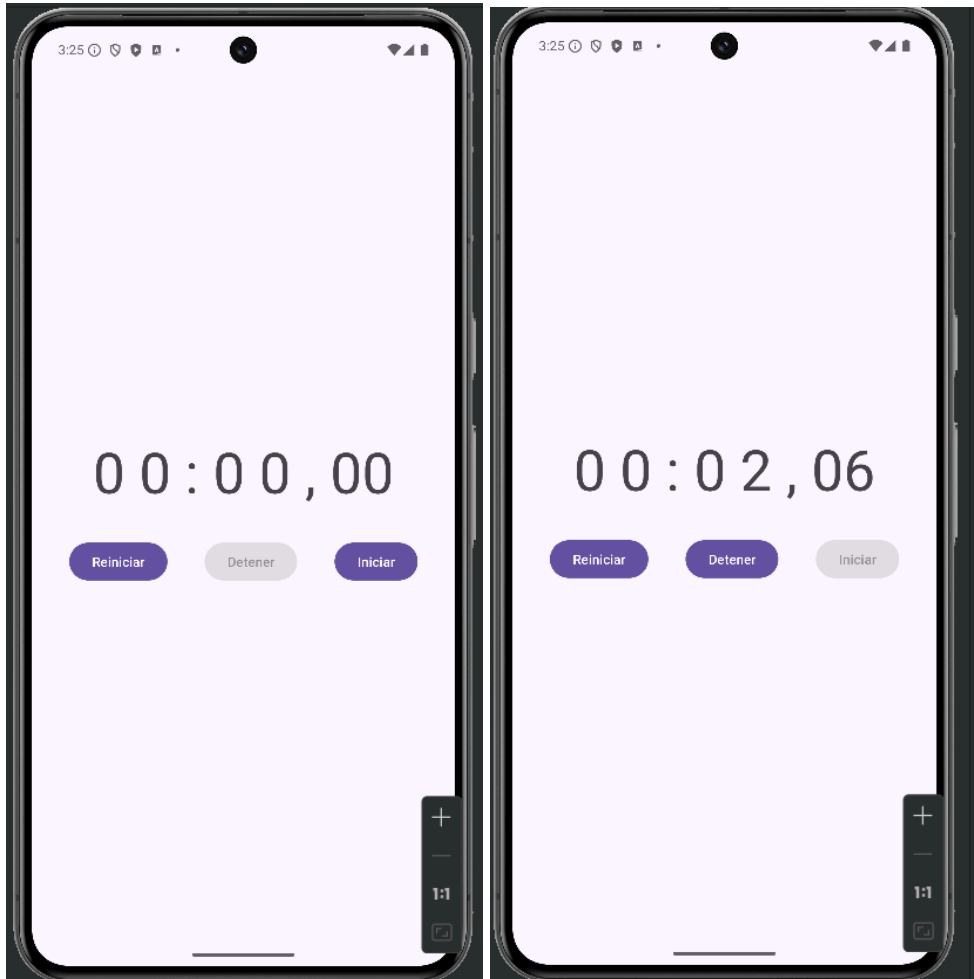


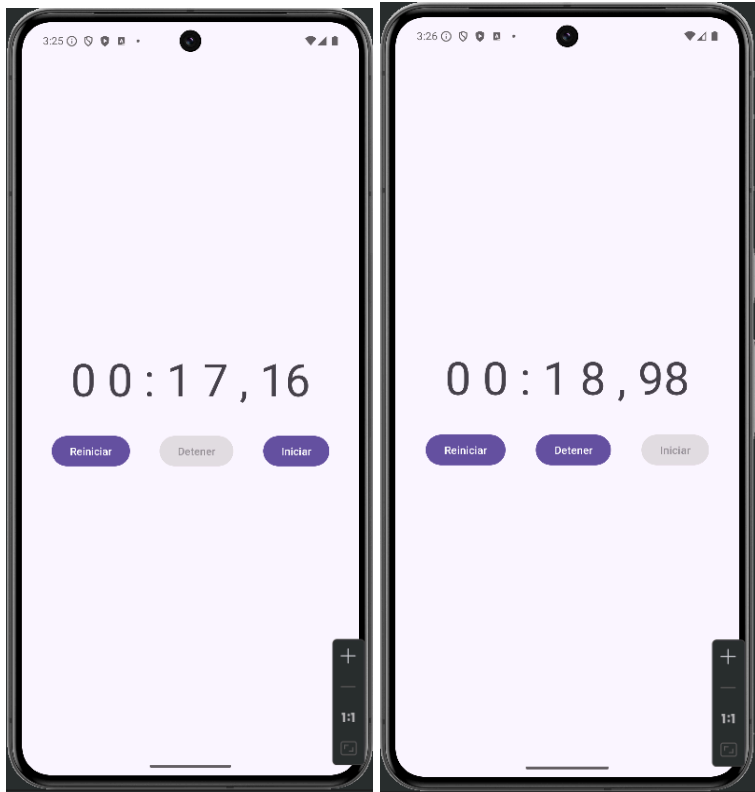
Tarea individual 4 - Cronometro

Inicio la aplicación que muestra un cronómetro con minutos, segundos, décimas de segundo y centésimas de segundo.

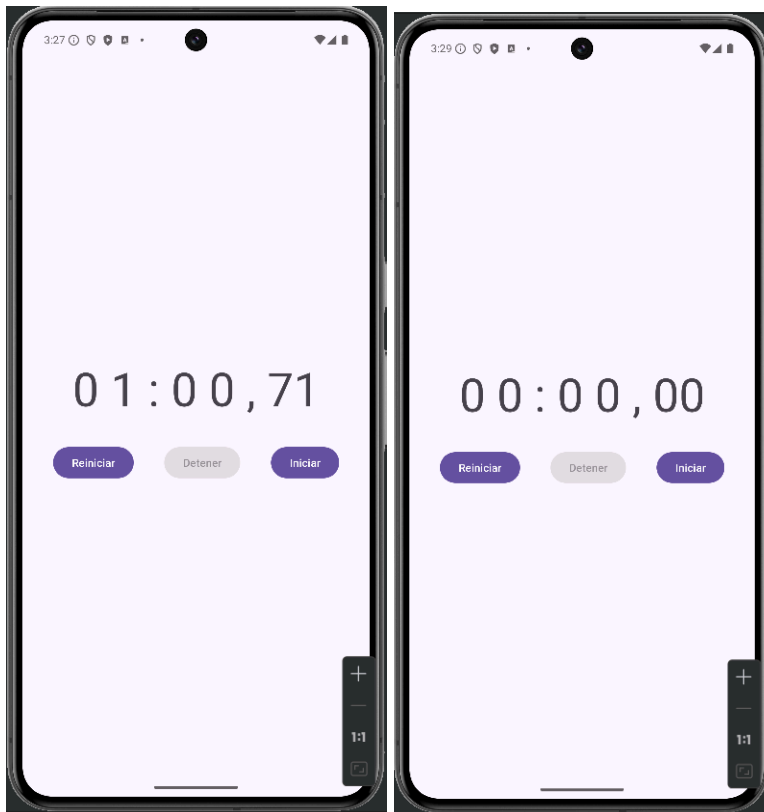
Inhabilita el botón de parar cuando no está iniciado, e inhabilita el botón de iniciar cuando sí esta iniciado.

He usado un hilo para el retardo en milisegundos, en una clase a parte.





El botón de reiniciar convierte a 0 los valores del cronómetro.



```

16 </> public class MainActivity extends AppCompatActivity {
17
18     private View mainView; no usages
19     private Button btnReiniciar, btnDetener, btnIniciar; 2 usages
20     public TextView cronometro; 2 usages
21
22     private int dmins, mins, dsecs, secs, dmilis, milis; 2 usages
23     private boolean parado; 2 usages
24     private boolean primero = true; no usages
25
26     private Object monitor; 4 usages
27
28     @Override
29     protected void onCreate(Bundle savedInstanceState) {
30
31         monitor = new Object();
32         dmins = 0; mins = 0; dsecs = 0; secs = 0; dmilis = 0; milis = 0;
33         parado = true;
34         super.onCreate(savedInstanceState);
35         EdgeToEdge.enable( $this$enableEdgeToEdge: this);
36         setContentView(R.layout.activity_main);
37         btnReiniciar = findViewById(R.id.btnReiniciar);
38         btnDetener = findViewById(R.id.btnDetener);
39         btnDetener.setEnabled(false);
40         btnIniciar = findViewById(R.id.btnIniciar);
41
42         cronometro = findViewById(R.id.cronometro);
43         hiloCronometro hc = new hiloCronometro(parado, cronometro, monitor, dmins, mins, dsecs, s
44         hc.start();
45
46         btnIniciar.setOnClickListener(new View.OnClickListener() {
47             @Override
48             public void onClick(View v) {
49                 synchronized (monitor) {
50                     hc.unsetParado();
51                     monitor.notify(); // Notifica al hilo que está esperando
52                 }
53                 btnIniciar.setEnabled(false);
54                 btnDetener.setEnabled(true);
55             }
56         });
57
58         btnDetener.setOnClickListener(new View.OnClickListener() {
59             @Override
60             public void onClick(View v) {
61                 hc.setParado();
62                 btnDetener.setEnabled(false);
63                 btnIniciar.setEnabled(true);
64             }
65         });
66     }
67 }

```

```

        btnReiniciar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                hc.reset();
            }
        });

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });
    }
}

```

Codigo de la clase 'hiloCronometro':

```

package com.example.ejer4;
import com.example.ejer4.R;
import android.util.Log;
import android.widget.TextView;

public class hiloCronometro extends Thread { 2 usages
    public TextView cronometro; 3 usages

    private int dmins, mins, dsecs, secs, dmilis, milis; 6 usages
    private boolean parado; 5 usages

    private final Object monitor; 5 usages

    public hiloCronometro(boolean parado, TextView cronometro, Object monitor, int dmins, int min
        this.parado = parado;
        this.cronometro = cronometro;
        this.monitor = monitor;
        this.dmins = dmins;
        this.mins = mins;
        this.dsecs = dsecs;
        this.secs = secs;
        this.dmilis = dmilis;
        this.milis = milis;
    }

    public void setParado(){ 1 usage
        this.parado = true;
    } //5,4 sin test

    public synchronized void unsetParado() { 1 usage
        this.parado = false;
        synchronized (monitor) {

```

```

31         this.parado = false;
32     synchronized (monitor) {
33         monitor.notify();
34     }
35 }
36
37 public void reset(){ 1 usage
38     this.dmins = 0; this.mins = 0; this.dsecs = 0; this.segs = 0;
39     this.dmilis = 0; this.milis = 0;
40     cronometro.setText("0 0 : 0 0 , 00");
41 }
42
43 public void run() {
44     synchronized (monitor){
45
46
47
48
49         dmins = 0; mins = 0; dsecs = 0; segs = 0; dmilis = 0; milis = 0;
50
51         while (true) { // Cambiamos el bucle para que sea infinito
52             while (this.parado) {
53                 try {
54                     monitor.wait(); // Espera en el monitor
55                 } catch (InterruptedException e) {
56                     e.printStackTrace();
57                 }
58             }
59

```

Almaceno los valores por separado para que se vean correctamente:

```

6         milis = milis + 10;
7
8         if(milis == 100) {
9             milis = 0;
10            dmilis = dmilis +1;
11        }
12
13        if(dmilis==10) {
14            dmilis = 0;
15            segs = segs +1;
16        }
17
18        if(segs == 10) {
19            segs = 0;
20            dsecs = dsecs +1;
21        }
22
23        if(dsecs == 6) {
24            dsecs = 0;
25            mins = mins + 1;
26        }
27
28        if(mins == 10 ) {
29            mins = 0;
30            dmins = dmins +1;
31        }
32
33        String todo = String.valueOf(dmins)+" " + String.valueOf(mins)+ " : " + String.valueOf(
34        cronometro.setText(todo);

```