



Universidad Nacional Autónoma de México
Facultad de Contaduría y Administración
Sistema Universidad Abierta y Educación a Distancia

Alumno: López Acevedo Víctor Rafael

Grupo: 9696

Materia: INFORMATICA VI

Unidad: 5

Actividad: M5-01

Fecha: 1 de mayo 2025

Actividad: Estructuras con la sentencia For

Objetivo del aprendizaje: Comprender y practicar la estructura de control For

Descripción de la actividad:

1. Calcula la secuencia de fibonacci de cualquier número "n"
2. Deberá de emplear la estructura de control "For" o "Recursividad" para la ejecución de su programa.
3. Agregar dos botones en la vista, utilizando recursos XML.
 - a. Primer botón de avance

```
<Button
    android:id="@+id/botonavanzar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Avanzar"
    android:textSize="18sp" />
```

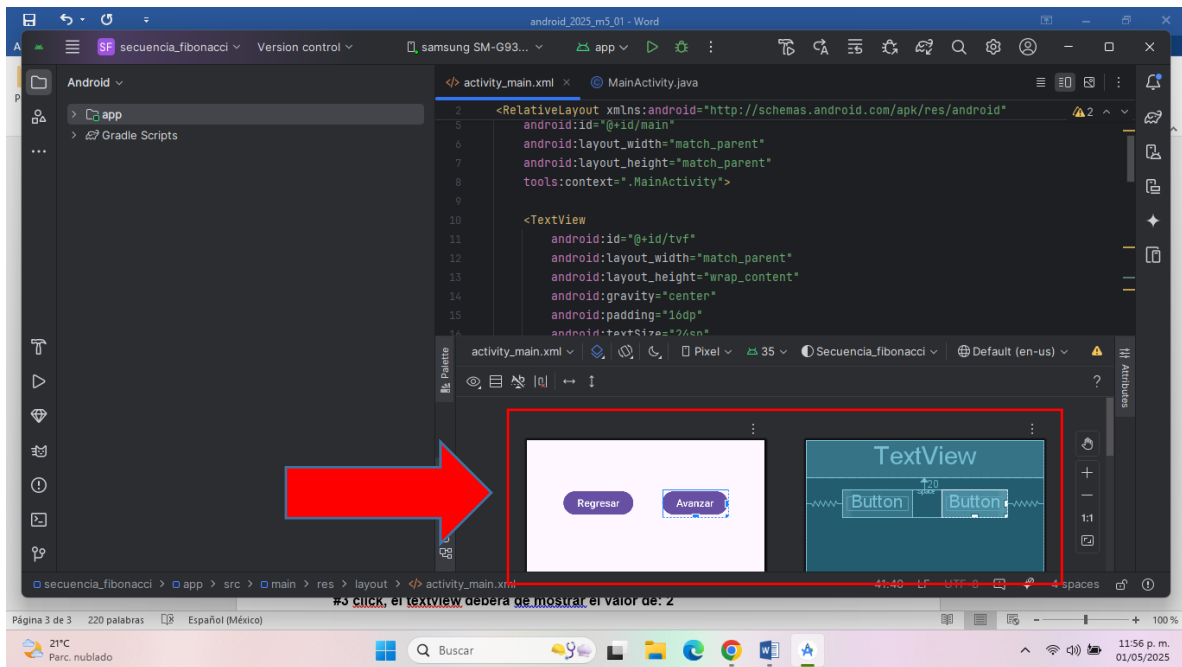
- b. Segundo botón de retroceso.

```
<Button
    android:id="@+id/botonregresar"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Regresar"
    android:textSize="18sp" />
```

4. Agregar un TextView en la vista utilizando recursos XML, éste textView mostrará el valor actual de la secuencia fibonacci.

```
<TextView
    android:id="@+id/tvf"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:padding="16dp"
    android:textSize="24sp"
    tools:layout_editor_absoluteX="93dp"
    tools:layout_editor_absoluteY="125dp" />
```

Al final, agregando los botones y el textview, así es como queda la aplicación



5. Al dar click en el botón deberá de realizar la secuencia fibonacci empezando con el número 1.

#1 click, el textview deberá de mostrar el valor de: 1

#2 click, el textview deberá de mostrar el valor de: 1

#3 click, el textview deberá de mostrar el valor de: 2

#4 click, el textview deberá de mostrar el valor de: 3

#5 click, el textview deberá de mostrar el valor de: 5

#6 click, el textview deberá de mostrar el valor de: 8

... consecutivamente ...

Incluir “memoization” en el cálculo de la secuencia fibonacci.

```
3 usages
Map<Integer, Long> memo = new HashMap<>();
```

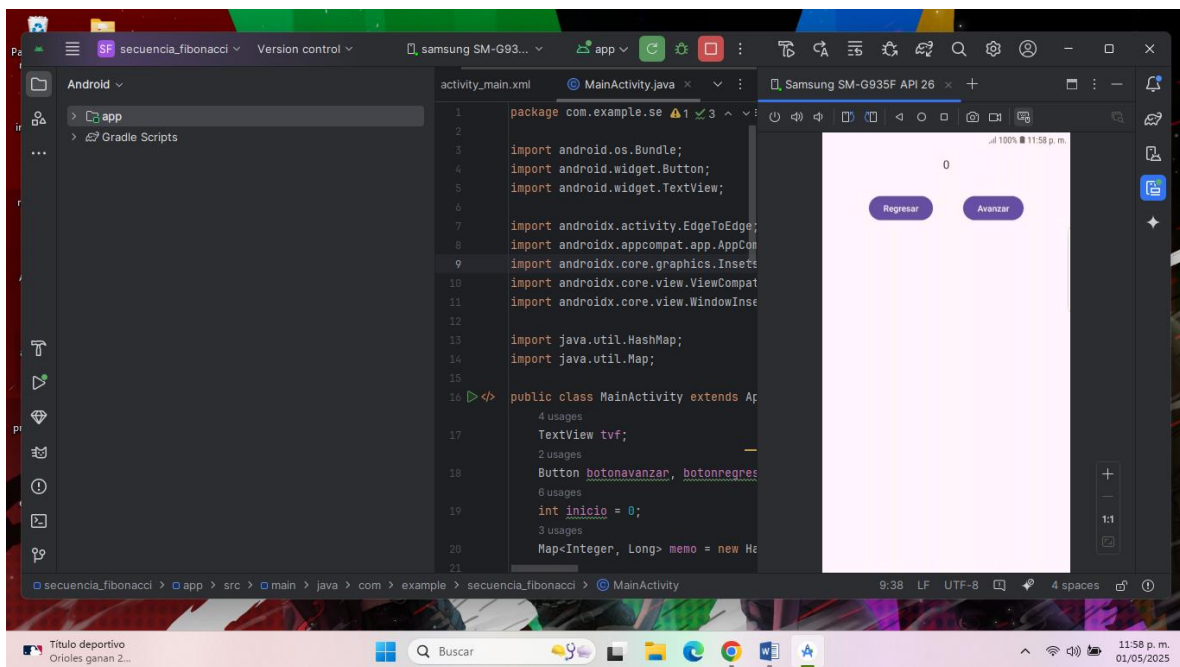
```

3 usages
private long fibonacci(int n) {
    if (memo.containsKey(n)) {
        return memo.get(n);
    }

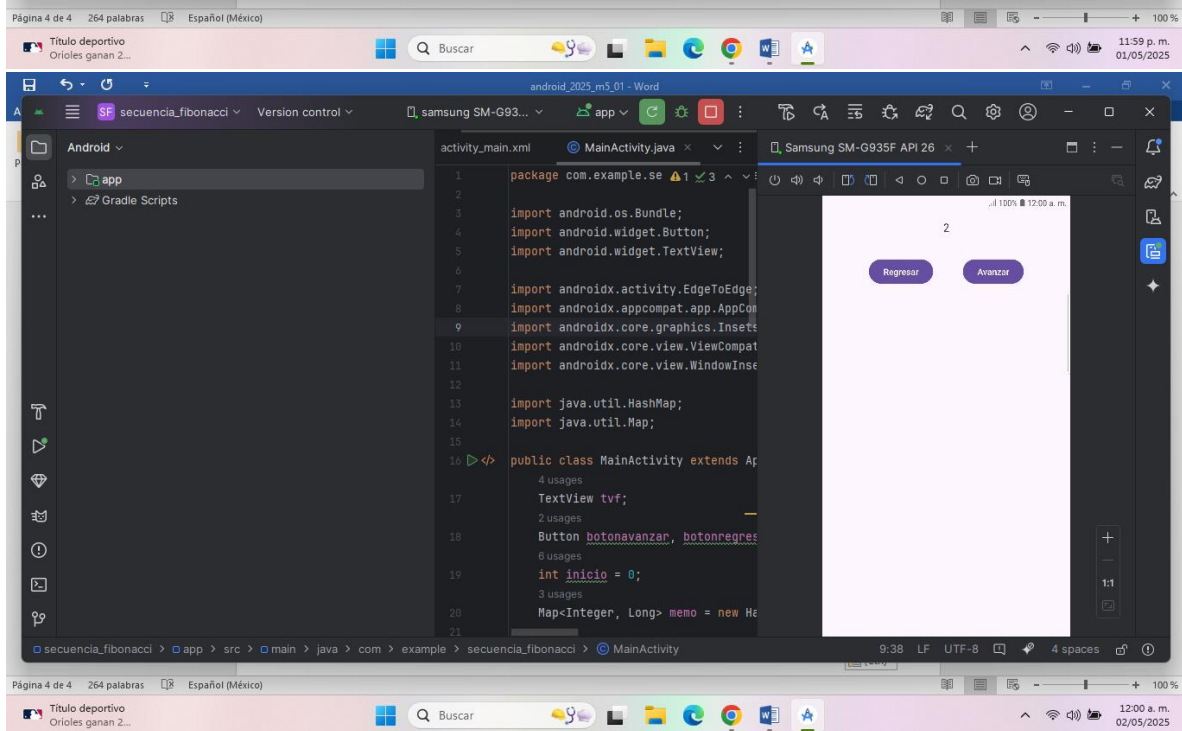
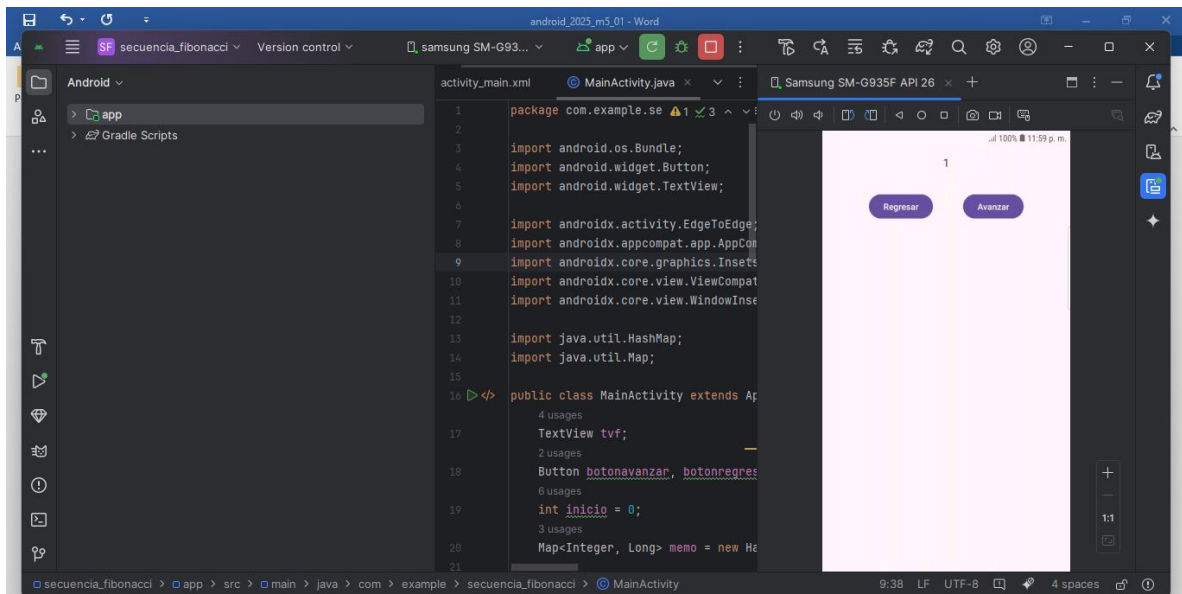
    long a = 0, b = 1, result = 0;
    for (int i = 0; i <= n; i++) {
        if (i == 0) result = 0;
        else if (i == 1) result = 1;
        else result = a + b;

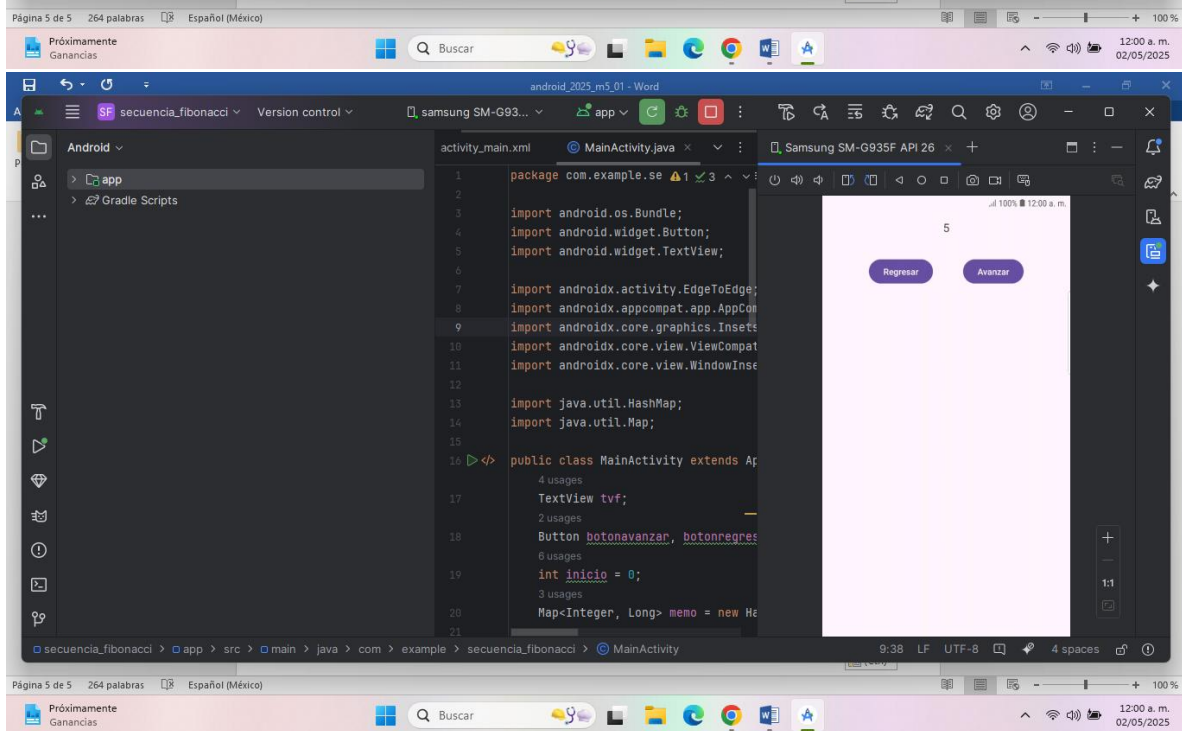
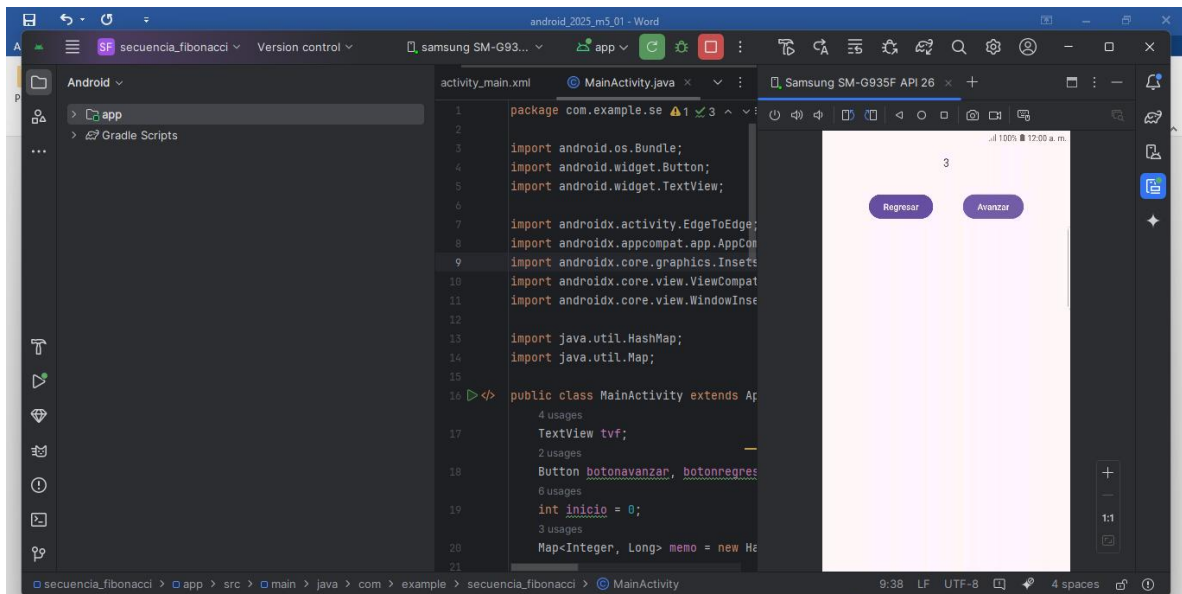
        memo.put(i, result);
        a = b;
        b = result;
    }
    return result;
}

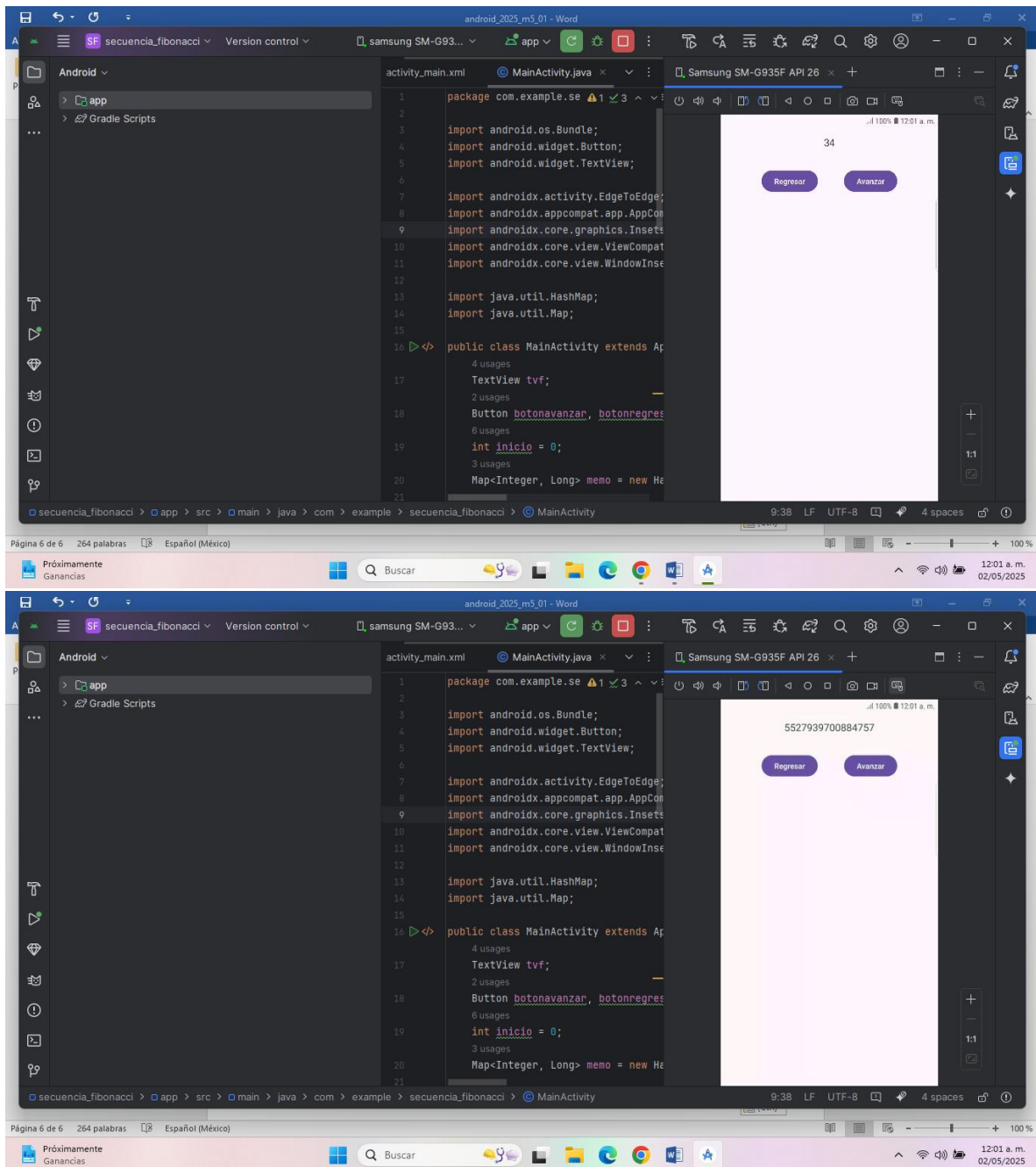
```



este es el resultado final de la aplicación, podemos observar que el textview se inicia en cero y conforme se va presionando el botón de avanzar la secuencia se incrementa







Y podemos llegar a números bastante altos, del mismo modo el botón regresar nos regresa a los primeros números, y se usa la memorization para que la aplicación no tenga que rehacer la operación, simplemente tomo los valores que se van almacenado conforme se van realizando operaciones cada que se avanza el numero

Este es el código utilizado en la aplicación

```
package com.example.secuencia_fibonacci;

import android.os.Bundle;
import android.widget.Button;
import android.widget.TextView;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import java.util.HashMap;
import java.util.Map;

public class MainActivity extends AppCompatActivity {
    TextView tvf;
    Button botonavanzar, botonregresar;
    int inicio = 0;
    Map<Integer, Long> memo = new HashMap<>();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);

        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;
        });

        tvf = findViewById(R.id.tvf);
        botonavanzar = findViewById(R.id.botonavanzar);
        botonregresar = findViewById(R.id.botonregresar);

        tvf.setText(String.valueOf(fibonacci(inicio)));

        botonavanzar.setOnClickListener(v -> {
            inicio++;
            tvf.setText(String.valueOf(fibonacci(inicio)));
        });

        botonregresar.setOnClickListener(v -> {
            if (inicio > 0) {
                inicio--;
                tvf.setText(String.valueOf(fibonacci(inicio)));
            }
        });
    }

    private long fibonacci(int n) {
```



```
    if (memo.containsKey(n)) {  
        return memo.get(n);  
    }  
  
    long a = 0, b = 1, result = 0;  
    for (int i = 0; i <= n; i++) {  
        if (i == 0) result = 0;  
        else if (i == 1) result = 1;  
        else result = a + b;  
  
        memo.put(i, result);  
        a = b;  
        b = result;  
    }  
    return result;  
}  
}
```