



Universidad Nacional Autónoma de  
México



Facultad de Contaduría y Administración

Desarrollo de Aplicaciones móviles

Profesor: Cristian Cardoso Arellano

Alumno: Ortega Maldonado Diego Daniel

Actividad M5 01

Clase que se encarga de la lógica de la sucesión:

```
public class GestorFibonacci {  
  
    6 usages  
    int index = 1;  
  
    3 usages  
    public int getCurrent() { return fibonacci(index); }  
  
    1 usage  
    public int next() {  
        index++;  
        return fibonacci(index);  
    }  
  
    1 usage  
    public int previous() {  
        if (index > 1) {  
            index--;  
        }  
        return fibonacci(index);  
    }  
}
```

```
3 usages  
private int fibonacci(int n) {  
    if (n == 1 || n == 2) return 1;  
  
    int a = 1, b = 1, result = 1;  
    for (int i = 3; i <= n; i++) {  
        result = a + b;  
        a = b;  
        b = result;  
    }  
    return result;  
}
```

Clase que ejecuta la serie:

```

public class MainActivity extends AppCompatActivity {

    6 usages
    private GestorFibonacci gestorFibonacci;
    4 usages
    private TextView txtFibonacci;

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

        gestorFibonacci = new GestorFibonacci();
        txtFibonacci = findViewById(R.id.txtFibonacci);

        Button btnNext = findViewById(R.id.btnNext);
        Button btnPrevious = findViewById(R.id.btnPrevious);

        // Mostrar el valor inicial
        txtFibonacci.setText(String.valueOf(gestorFibonacci.getCurrent()));

        btnNext.setOnClickListener(v -> {
            gestorFibonacci.next();
            txtFibonacci.setText(String.valueOf(gestorFibonacci.getCurrent()));
        });

        btnPrevious.setOnClickListener(v -> {
            gestorFibonacci.previous();
            txtFibonacci.setText(String.valueOf(gestorFibonacci.getCurrent()));
        });
    }
}

```

Resultados (orden actual – anterior – siguiente)

34

Avanzar

Retroceder

21

Avanzar

Retroceder

55

Avanzar

Retroceder

## Conclusión.

El uso de la estructura de iteración for, en conjunción con la estructura condicional if, permite el cálculo de la serie de Fibonacci de una forma relativamente eficiente. La lógica de generación requiere la iteración para poder avanzar, y la evaluación de una condición lógica para impedir que la secuencia intente calcular números menores a la unidad, cosa que es tanto ilógica como matemáticamente imposible. Esta actividad muestra cómo se requiere un buen conocimiento de las estructuras de control de flujo tanto iterativas como condicionales para lograr que un programa no tenga fallos inesperados ni valores aberrantes en su ejecución.