

Taller de Análisis

Juan Pablo Landi Largo

Sucesión de Fibonacci

Código Mejorado

```
public class FibonacciRecursivo {

    // Función principal
    public static void main(String[] args) {
        int n = 10;

        System.out.println("=====");
        System.out.println("    SUCESIÓN DE FIBONACCI (RECURSIVO)    ");
        System.out.println("=====");
        System.out.println("Términos desde F(0) hasta F(" + n + "):\n");

        for (int i = 0; i <= n; i++) {
            System.out.printf("F(%d) = %d\n", i, calcularFibonacci(i));
        }

        System.out.println("=====");
    }

    /**
     * Calcula el término n de la sucesión de Fibonacci utilizando recursividad.
     * @param n índice del término a calcular
     * @return valor de F(n)
     */
    public static int calcularFibonacci(int n) {
        if (n <= 1) return n;
        return calcularFibonacci(n - 1) + calcularFibonacci(n - 2);
    }
}
```

Relación de recurrencia

La función utilizada se basa en la siguiente relación de recurrencia:

$$F(n) = F(n-1) + F(n-2), \text{ con } F(0) = 0 \text{ y } F(1) = 1$$

Verificación mediante prueba de escritorio

Iteración	Llamada	Desglose	Resultado
i = 0	fibonacci(0)	—	0
i = 1	fibonacci(1)	—	1
i = 2	fibonacci(2) → f(1)+f(0)	1 + 0	1
i = 3	fibonacci(3) → f(2)+f(1)	1 + 1	2
i = 4	fibonacci(4) → f(3)+f(2)	2 + 1	3
i = 5	fibonacci(5) → f(4)+f(3)	3 + 2	5
i = 6	fibonacci(6) → f(5)+f(4)	5 + 3	8
i = 7	fibonacci(7) → f(6)+f(5)	8 + 5	13
i = 8	fibonacci(8) → f(7)+f(6)	13 + 8	21
i = 9	fibonacci(9) → f(8)+f(7)	21 + 13	34
i = 10	fibonacci(10) → f(9)+f(8)	34 + 21	55

Cálculo con la fórmula de Binet

También es posible obtener los términos mediante la fórmula cerrada de Binet:

$$F(n) = (\varphi^n - \psi^n) / \sqrt{5}$$

Donde:

$$\varphi = (1 + \sqrt{5}) / 2 \approx 1.61803$$

$$\psi = (1 - \sqrt{5}) / 2 \approx -0.61803$$

n	F(n) (aproximado)
0	0
1	1
2	1
3	2
4	3
5	5
6	8
7	13
8	21

9	34
10	55