



Practica 2.1 Prueba de Fibonacci Recursivo y Iterativo

Alumno: Diego Emmanuel Del Angel Flores

Carrera: Ingeniería en Sistemas Computacionales

Materia: Estructura De Datos

Maestra: Clara Alicia Vela Ortega

Objetivo:

Calcule la secuencia de Fibonacci de forma iterativa y recursiva, probando ambos algoritmos con valores de $n = 20, 30, 40$ y 60 . Indique el tiempo que tarda cada uno en la tabla de comparación a continuación. Utilice el método Java `nanotime()` para realizar esta medición en nanosegundos, para ello debe definir una variable `time1` antes de llamar a la ejecución del método, y definir otra variable `time2` cuando la ejecución del método se complete, luego restar `time2` de `time1` para obtener el tiempo de ejecución el tiempo es ($\text{tiempoR} = \text{tiempo2} - \text{tiempo1}$).

Implementacion de codigo

```
Main.java

1 public class Main {
2     public static void main(String[] args) {
3         System.out.println("===== Fibonacci Iterativo =====");
4
5         for (int i =20; i≤60; i = i+10){
6             long inicio1 = System.nanoTime();
7             long result = fibo(i);
8             long fin1 = System.nanoTime();
9             long ttl =fin1 -inicio1;
10
11             System.out.println("El valor obtenido es:"+fibo(i));
12             System.out.println("El calculo de la serie fibonacci "+ i +" tomo:"+ttl);
13             System.out.println(" \n");
14         }
15     }
16 }
```

```
Main.java

1 System.out.println("===== Fibonacci Recursivo =====");
2     for (int i=20; i≤60; i = i +10 ){
3         long inicio = System.nanoTime();
4         long result = fib(i);
5         long fin = System.nanoTime();
6         long tt = fin -inicio;
7
8
9         System.out.println("El valor obtenido es:"+fib(i));
10        System.out.println("El calculo de la serie fibonacci " + i + " tomo:" + tt);
11        System.out.println(" \n");
12    }
13 }
```

```

Main.java

1  public static long fib(int n){
2      if( n ≤ 1){
3          return n;
4      }else {
5          return fib(n-1) + fib(n-2);
6      }
7  }
8  public static long fibo(int n){
9      int previous = 1 -2;
10     int result =1;
11     for(int i =0; i≤n; ++i){
12         int sum = result + previous;
13         previous = result;
14         result = sum;
15     }
16     return result;
17 }
18 }

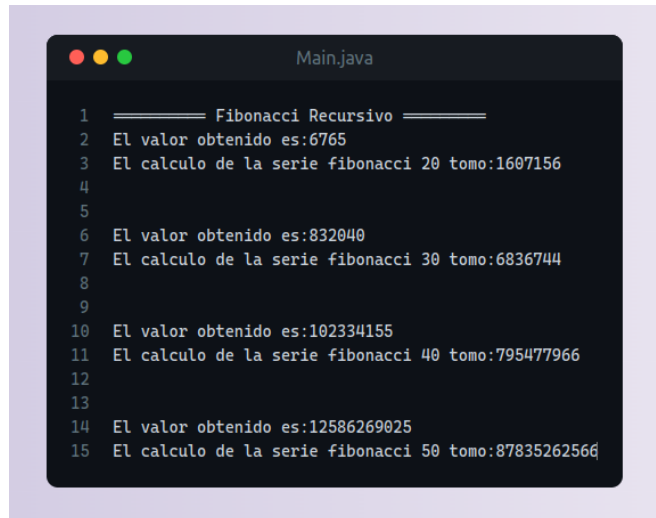
```

```

Main.java

1  ===== Fibonacci Iterativo =====
2  El valor obtenido es:6765
3  El calculo de la serie fibonacci 20 tomo:3437
4
5
6  El valor obtenido es:832040
7  El calculo de la serie fibonacci 30 tomo:1623
8
9
10 El valor obtenido es:102334155
11 El calculo de la serie fibonacci 40 tomo:1884
12
13
14 El valor obtenido es:-298632863
15 El calculo de la serie fibonacci 50 tomo:3226
16
17
18 El valor obtenido es:1820529360
19 El calculo de la serie fibonacci 60 tomo:2215

```



```
1  ===== Fibonacci Recursivo =====
2  El valor obtenido es:6765
3  El calculo de la serie fibonacci 20 tomo:1607156
4
5
6  El valor obtenido es:832040
7  El calculo de la serie fibonacci 30 tomo:6836744
8
9
10 El valor obtenido es:102334155
11 El calculo de la serie fibonacci 40 tomo:795477966
12
13
14 El valor obtenido es:12586269025
15 El calculo de la serie fibonacci 50 tomo:87835262566
```

N	Valor Obtenido	Tiempo Iterativo	Tiempo Recursivo
20	6775	3437	1607156
30	832040	1623	6836744
40	102334155	1884	795477966
60	1820529360	2215	-

Conclusión:

Sin duda alguna, el enfoque iterativo tiene complejidad y velocidad, ni hablar cuando es un $O(n)$ y por eso, una vez ejecutado la programación de formularios iterativos lleva mucho tiempo, En cambio el enfoque recursivo es lento.