

## Serie de Fibonacci (Con memoización)

**0, 1, 1, 2, 3, 5, 8, 13, 21, 34 .....**

La serie de Fibonacci nos dice que se suman los dos números previos para sacar un tercero, así hasta el infinito.

Por regla, se inicia con 0 y 1. Haciendo el tercer número el "1". Y sumando el anterior a ese "1" da como resultado "2".

¿Cómo podemos atacar ese problema?

Podemos almacenar los números de fibonacci calculados hasta ahora y almacenar todos los valores en una matriz.

En mi programa de Python lo hago de Arriba hacia abajo y guardando las variables en el arreglo

0	1	2	3	4
0	1			
0	1	1		
0	1	1	2	
0	1	1	2	3

**Complejidad de tiempo:  $O(N)$**

**Complejidad espacial:  $O(N)$**

## CÓDIGO:

```
1  # Usando memoización (De arriba hacia abajo)
2
3  #Podemos almacenar los números de fibboonacci calculados hasta ahora y
4  # almacenar todos los valores en una matriz.
5
6
7  # Iniciar el array de dp
8  dp = [-1 for i in range(10)]
9
10 def fibbo(n):
11     if (n <= 1):
12         return n;
13     global dp;
14
15     # Variables temporales a guardar
16     # Valores de fibbo(n-1) y de fibbo(n-2)
17     primero = 0;
18     segundo = 0;
19
20     if (dp[n - 1] != -1):
21         primero = dp[n - 1];
22     else:
23         primero = fibbo(n - 1);
24     if (dp[n - 2] != -1):
25         segundo = dp[n - 2];
26     else:
27         segundo = fibbo(n - 2);
28     dp[n] = primero + segundo;
29
30     # Memoización
31     return dp[n] ;
32
33 # Main
34 if __name__ == '__main__':
35     n = 7;
36     print(fibbo(n));
```