

AULA 2 - ALGORITMOS DISTINTOS PARA A RESOLUÇÃO DO MESMO PROBLEMA

Pretende-se comparar o número de operações e os tempos de execução de três métodos alternativos para o cálculo dos números de Fibonacci.

- Solução recursiva
$$F(n) = \begin{cases} 0, & \text{se } n = 0 \\ 1, & \text{se } n = 1 \\ F(n-1) + F(n-2), & \text{se } n \geq 2 \end{cases}$$

- Solução repetitiva $\text{Fib}(0) = 0; \text{Fib}(1) = 1; \text{Fib}(2) = \text{Fib}(1) + \text{Fib}(0); \text{Fib}(3) = \text{Fib}(2) + \text{Fib}(1); \dots$

Para efectuar o cálculo precisamos de três variáveis inteiras para representarem, respectivamente: o *valor actual* (inicialmente indefinido e que em cada iteração é igual à soma dos dois valores anteriores); o *valor anterior ao anterior* (que inicialmente é 0 e em cada iteração passa a ser o valor anterior); e o *valor anterior* (que inicialmente é 1 e em cada iteração passa a ser o valor actual acabado de calcular).

- Solução usando a fórmula fechada
$$F(n) = \frac{\phi^n - (1-\phi)^n}{\sqrt{5}}, \text{ com } \phi = \frac{1+\sqrt{5}}{2} \approx 1.618034$$

- Solução usando a fórmula fechada
$$F(n) = \text{round}\left(\frac{\phi^n}{\sqrt{5}}\right), \text{ que é equivalente a } F(n) = \left\lfloor \frac{\phi^n}{\sqrt{5}} + \frac{1}{2} \right\rfloor$$

- Solução usando a fórmula fechada
$$F(n) = \text{round}(c_1 \times e^{(n \times c_2)})$$

com $c_1 = 0.44721357549995773928$ e
com $c_2 = 0.48121182505960344750$

