

## Parte 1: Teoria

### 1. Definicion y concepto

La recursividad es cuando una función o un método se llama a sí mismo.

### 2. Caso base y caso recursivo

Para la funcion fibonacci recursiva, identifica cual es el caso base y cual es el caso recursivo:

```
int fibonacci(int n) {  
  if (n == 0) return 0;  
  else if (n == 1) return 1;  
  else return fibonacci(n - 1) + fibonacci(n - 2);  
}
```

Caso base: El caso que evita bucles infinitos

Caso recursivo : Donde la función se llama a si misma.

### 3. Ventajas y desventajas de la recursion

Escribe al menos una ventaja y una desventaja de usar recursion en programacion

El código recursivo es fácil de escribir y mantener, pero cada llamada consume memoria de las variables.

### 4. Recursion vs Iteracion

Explica cuando podria ser mejor usar recursion en lugar de un bucle (iteracion) y viceversa

Sería mejor utilizar la recursión en un caso que se puede dividir en problemas idénticos más simples, en cambio, es mejor usar una secuencia cuando se actúa sobre distintos elementos(ej. Arrays, listas, etc.)