

Tema 6. Recursión

6.0. Contenido y documentación

[6.0. Contenido y documentación](#)

[6.1. Algoritmo recursivo](#)

[6.2. Recursión de cola](#)

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/1e2e82e9-82ed-4e73-adcf-4ac64db19c2b/U6_Recursin.pdf

6.1. Algoritmo recursivo

Definición. Un **algoritmo recursivo** es un algoritmo que resuelve un problema llamándose a sí mismo sobre instancias más simples del problema.

Incluye uno o más **casos base** sobre los que se opera trivialmente.

Uno de los ejemplos más comunes de recursión es el cálculo de factoriales:

```
int factorial (int n){
    /*caso base*/
    if (n == 0)
        return 1;

    return n * factorial(n-1);
}
```

Otro ejemplo común que cuenta con más de un caso base es el cálculo de la sucesión de Fibonacci:

```
int fibonacci (int n){
    /*casos base*/
    if (n == 0)
        return 0;
    if (n == 1);
        return 1;

    return fibonacci(n-2) + fibonacci(n-1);
}
```

6.2. Recursión de cola

Definición. Una llamada **recursiva de cola** es una llamada recursiva tras la que no se realizan más operaciones.

Una función es recursiva de cola si la llamada recursiva es el último ejecutado en la función.

Las funciones recursivas de cola son sencillas de sustituir por funciones sin recursión. Para ello, basta con:

1. Reasignar los argumentos de la función.

2. Saltar al punto de inicio de la función.
3. Sustituir la llamada por un bucle usando el caso base de la recursión como condición de parada.