

## Recursividad

Es una técnica de programación en la cual un método puede llamarse a sí mismo, en la mayoría de casos un algoritmo iterativo es más eficiente que uno recursivo si de recursos de la computadora se trata, pero un algoritmo recursivo en muchos casos permite realizar problemas muy complejos de una manera más sencilla.

### **Reglas de la recursividad:**

**Regla 1:** Por lo menos debe tener un caso base y una parte recursiva.

**Regla 2:** Toda parte recursiva debe tender a un caso base.

**Regla 3:** El trabajo nunca se debe duplicar resolviendo el mismo ejemplar de un problema en llamadas recursivas separadas.

### **Recursividad Directa**

Cuando una función realiza un número determinado de llamadas a si misma hasta encontrar la condición de terminación de la recursión.

### **Recursividad Indirecta**

Cuando una función hace un llamado a otra, la que en un momento determinado hará un nuevo llamado a la función que la llamo en un principio.

### ***Procedimientos recursivos***

- Fibonacci: La secuencia de Fibonacci es aquella secuencia de enteros donde cada elemento es la suma de los dos anteriores.
- Torres de Hanoi: Se tienen tres torres y un conjunto de n discos de diferentes tamaños. Inicialmente los discos están ordenados de mayor a menor en la primera torre. El problema consiste en pasar los discos a la tercera torre utilizando la segunda como auxiliar.
  - En cada movimiento solo puede moverse un disco.
  - No puede quedar un disco mayor sobre uno menor.

## Casos de recursividad.

### -Factorial

La definición recursiva del factorial de un número  $n > 0$ , considerando que  $0! = 1$  por definición, se obtiene como consecuencia de aplicar la propiedad asociativa de la multiplicación.

$$n! = n * (n - 1)!$$

### -Multiplicación entera

La definición recursiva de la multiplicación de dos números  $a$  y  $b$ , se deriva de la definición de la multiplicación como una suma abreviada y la aplicación de la propiedad asociativa de la suma, es decir:

$$a * b = \begin{cases} a + (a * (b-1)) & \text{si } b > 0 \\ 0 & \text{si } b = 0 \end{cases}$$

### -Exponenciación entera

La definición recursiva de la operación exponenciación entera, es decir, calcular la potencia  $a^b$ , se deriva de la definición de la potencia como una multiplicación abreviada de la propiedad asociativa de la multiplicación, es decir:

$$A^b = \begin{cases} a * a^{b-1} & \text{si } b > 0 \\ 0 & \text{si } b = 0 \end{cases}$$

### -Suma de los elementos de un vector

Las sumas de los elementos de un vector pueden expresarse, también, recursivamente, se representa:

$$\sum_{i=1}^{n-1} V_i + V_n$$