

### Algoritmos recursivos

La 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:

Para que un problema pueda resolverse de forma recursiva debe cumplir las siguientes 3 reglas:

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.

Ejemplo: Calcular el factorial de un número.

### FACTORIAL DE UN NÚMERO N

$$8! = 8 * 7!$$

$$7! = 7 * 6!$$

$$6! = 6 * 5!$$

.

En general,

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

Veamos un caso particular, calculemos el factorial de 5 ( 5! ):

factorial de 5 =  $5 * 4!$  ———> “factorial de 5 es igual 5 multiplicado por factorial de 4”

factorial de 4 =  $4 * 3!$  ———> “factorial de 4 es igual 4 multiplicado por factorial de 3”

factorial de 3 =  $3 * 2!$  ———> “factorial de 3 es igual 3 multiplicado por factorial de 2”

factorial de 2 =  $2 * 1!$  ———> “factorial de 2 es igual 2 multiplicado por factorial de 1”

factorial de 1 = 1 ———> “factorial de 1 es 1” ———> “caso base”

Una implementación en java seria:

```
public long factorial (int n){  
if (n == 0 || n==1) //Caso Base  
return 1;  
else  
return n * factorial (n - 1); //Parte Recursiva  
}
```

### Bibliografía

Jose Fager, W. Libardo Pantoja Yepez (2014), Estructuras de Datos, LATIn, Mexico. Pags 105-106