

## Recursividad

Una función o procedimiento que se puede llamar a sí mismo se llama recursivo. La recursión (recursividad) es una herramienta muy potente en algunas aplicaciones, sobre todo de cálculo. La recursión puede ser utilizada como una alternativa a la repetición o estructura repetitiva. El uso de la recursión es particularmente idóneo para la solución de aquellos problemas que pueden definirse de modo natural en términos recursivos. La escritura de un procedimiento o función recursiva es similar a sus homónimos no recursivos; sin embargo, para evitar que la recursión continúe indefinidamente es preciso incluir una condición de terminación. La razón de que existan lenguajes que admiten la recursividad se debe a la existencia de estructuras específicas tipo pilas (stack, en inglés) para este tipo de procesos y memorias dinámicas. Las direcciones de retorno y el estado de cada subprograma se guardan en estructuras tipo pilas.

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.

### Tipos de recursividad:

- *Directa:* El Algoritmo recursivo se llama a sí mismo.
- *Indirecta:* El algoritmo recursivo llama a otro que provoca eventualmente llama al original
- *Final:* Al finalizar la llamada recursiva, no queda ninguna opción por ejectar salvo retornar.
- *No final:* Después de la llamada recursiva quedan acciones por realizar

## Bibliografía

- 1- Luisa Joyanes Aguilar (2008), Fundamentos de programación, mcgraw-hill, España. Pags 229
- 2- José Fager, W. Libardo Pantoja Yépez (2014), Estructuras de Datos, LATIn, Mexico. Pags 95-97