

RECURSIVIDAD

En el lenguaje de programación C se tienen la cualidad de la recursividad, o sea que C se puede emplear recursivamente.

A la recursividad se le puede denominar como función, una función es una subrutina o subprograma que forman un programa que realiza tareas bien definidas, la recursividad tiene la característica de poder llamarse a sí misma de forma directa o indirectamente cuando así se requiera, esto sin la necesidad de hacer algún uso de ciclos de repetición un buen ejemplo para poder tener contexto de la recursividad sería como las Matrushkas, que son muñecas rusas que contienen otra muñeca de menor tamaño dentro de sí misma.

La recursividad es un concepto fundamental en el área de las matemáticas y de la computación, es una alternativa diferente para implementar ciclos, además se puede usar en toda situación donde el resultado tenga que ser expresado con secuencias de movimientos, pasos o transformaciones gobernadas por un conjunto de reglas no ambiguas.

La función recursiva se compone por:

- *Caso base: una solución simple para un caso particular en el cual puede haber más de un caso base

- *Parte recursiva: Relaciona el resultado del algoritmo con resultados desglosados en casos más simples. Se hacen llamadas a la función, pero están más cercanas al caso base.

Los algoritmos recursivos son más simples que los iterativos, sin embargo, su ejecución en una computadora es más lenta y se requiere más recursos.

Hay cuatro tipos de recursividad las cuales son:

- *Recursividad simple: Es aquella en cuya definición solo aparece una llamada recursiva. Se puede cambiar a una función con ciclo iterativo.

- *Recursividad múltiple: Se da cuando hay más de una llamada a sí misma dentro del cuerpo de la función.

- *Recursividad anidada: En algunos de los argumentos de la llamada recursiva hay una nueva llamada a sí misma.

- *Recursividad cruzada o indirecta: Son algoritmos de los argumentos donde una función provoca una llamada a sí misma de forma indirecta, a través de otras funciones.

Un ejemplo que se suele usar en este tema es el tema factorial de un número.

se escribe un programa que calcule el factorial de un entero no negativo:

Este código será uno del tipo iterativo-repetitivo

```
public int factorial (int n) {  
  
    int fact = 1  
  
    for (int i = 1; i <= n; i++)  
  
        fact = i * fact;  
  
}
```

y este sería su forma recursiva

```
public int factorial (int n) {  
  
    if (n == 0) {  
        return 1;  
    } else {  
        return factorial(n-1) *n;  
    }  
  
}
```

para poder hacer que funcione bien la solución recursiva se tiene que saber bien cuál es el caso base para poder partir desde ahí

La ventaja de usar un programa recursivo es que son más cercanos a las matemáticas, generalmente se facilita el análisis del problema a resolver, se adapta de mejor manera a las estructuras de datos recursivos, y el algoritmo recursivo nos ofrece soluciones estructuradas, modulares y simples.

Aunque no siempre es bueno usar la recursividad todo depende en donde se requiera, es buena idea implementar esto para simplificar el código y cuando la estructura de datos es recursiva como lo son los árboles, cuando no se recomienda hacer uso de la recursividad sería cuando los métodos usen arreglos demasiado largos, cuando es impredecible su manera de comportarse y cuando la iteración resuelve de mejor manera el problema.

Recursión contra iteración

la iteración maneja un ciclo explícito o sea que se expresa de manera clara mientras que la recursión tiene repetidas invocaciones, por otro lado, en la iteración el ciclo termina o la condición del ciclo falla y la recursión debe de reconocer el caso base.

En ambos casos se pueden obtener ciclos infinitos.

Por último, se le conoce a dividir para vencer como recursividad, por las siguientes características que posee:

- *Muchas veces es posible dividir un problema en subproblemas con menor dificultad

- *es una técnica natural para las estructuras de datos, ya que por definición están compuestas por piezas.

Referencias:

O. Castillo. (04-2011). Análisis de algoritmo: Recursividad. 06-2020, de Universidad Veracruzana
Sitio web: <https://www.uv.mx/personal/ocastillo/files/2011/04/Recursividad.pdf>

Mónica E. García. (-). Funciones Tipos de funciones y Recursividad. 2020-06, de Universidad Tecnológica de la Mixteca Sitio web: [http://www.utm.mx/~mgarcia/PE4\(Funciones\).pdf](http://www.utm.mx/~mgarcia/PE4(Funciones).pdf)

BRIAN W. KERNIGHAN, DENNIS M. RITCHIE. (1991). EL LENGUAJE DE PROGRAMACION. Estados Unidos: Pearson Educación.