**Recursion lineal**

En este tipo de recursión genra otra llamada recursiva. Se pueden distinguir dos tipos de recursión lineal.

**Recursión lineal no final.**

El resultado de la llamada recursiva se combina en una expresión para dar lugar al resultado de la función que llama. Un ejemplo típico de este tipo de recursión es el calculo de un numero factorial. Ya que la llamada de un numero n factorial es igual al producto de n por el factorial de n-1.

**Recursión lineal final**

El resultado que es devuelto es el resultado de la ejecución de la última llamada recursiva.

**Recursión múltiple**

Este tipo de recursión puede generar mas de una llamada a la función. Uno de los ejemplos mas comunes para este tipo de recursión es los números de Fibonacci.

**Recursión anidada**

Una función recursiva en la que el argumento pasado a la función es la función en sí misma.

**Recursión exponencial**

Recursión donde se realiza más de una llamada a la función desde dentro de sí misma. Esto lleva a un crecimiento exponencial en el número de llamadas recursivas.

**Recursión de cola**

Son funciones que realizan con una llamada recursiva que no crea ninguna operación deferida. Con un compilador que automáticamente optimiza las llamadas a las funciones recursivas de cola. Así es proceso que genera es iterativo y equivalente a usar ciclos for y while.

David S. Touretzky. (1990). *Common Lisp: A Gentle Introduction to Symbolic Computation*. California: The Benjamin/Cummings.

Mathias F. & Robert B.. (2001). How to Design Programs An Introduction to Computing and Programmin. Cambridge: MIT Press.