

# LECCIÓN 2: Funciones Recursivas

¿La recursividad existe en otros lenguajes de programación además de C?	Sí.
¿Qué es la recursividad?	La recursividad o recurrencia es la forma en la cual se especifica un proceso basado en su propia definición
¿De qué dos partes está formada una función recursiva?	Del caso base y de la llamada recurrente.
¿Para qué se utiliza una llamada recursiva?	Se utiliza para realizar una llamada a una función desde la misma función.
¿La recursividad y la iteración están relacionadas?	Sí,
¿Cualquier problema que se pueda resolver por recursividad, se puede resolver mediante la iteración?	Sí.
¿Cualquier problema que se pueda resolver por iteración, se puede resolver mediante la recursividad?	Sí.
¿Con qué se debe tener cuidado cuando se realiza una función recursiva?	Es fácil crear una función recursiva que no llegue a devolver nunca un resultado definitivo y no pueda llegar a un punto de finalización.
Di dos ejemplos de funciones recursivas	El factorial de un número y la sucesión de Fibonacci.
Implementa la función recursiva del factorial de un número	<pre>int factorial(int n) {     if (n == 0)         return 1;     else         return (n * factorial(n-1)); }</pre>
Implementa la función recursiva de la sucesión de Fibonacci	<pre>int fibonacci(int n) {     if ( n == 0    n == 1 )         return n;     else         return (fibonacci( n - 1 ) +                 fibonacci( n - 2 )); }</pre>
¿Cuántas propiedades debe cumplir una función recursiva?	Dos
Propiedad 1 de las funciones recursivas	Un algoritmo A que resuelve un problema P es recursivo si está basado

	<i>directa o indirectamente en sí mismo.</i>
Propiedad 2 de las funciones recursivas	Una función recursiva $f$ debe definirse en términos que no impliquen a $f$ al menos en un argumento o grupo de argumentos. Debe existir una "salida" de la secuencia de llamadas recursivas.
Ventajas de la recursividad	Nos proporciona soluciones simples, claras, elegantes y resuelven problemas complejos.
Desventaja de la recursividad	La ineficiencia
¿Por qué son ineficientes las funciones recursivas?	Porque una simple llamada puede generar un gran número de llamadas recursivas (sobrecarga de las funciones recursivas). Y por la ineficiencia inherente de algunos algoritmos recursivos.
¿Cuándo debemos usar las funciones recursivas?	La recursividad se debe usar cuando sea realmente necesaria, es decir, cuando no exista una solución iterativa simple
Tipos de recursividad	Recursividad lineal y múltiple
¿Cuándo una función recursiva es lineal?	Cuando cada llamada recursiva, genera, como mucho, otra llamada recursiva.
¿De qué tipos puede ser una función recursiva lineal?	Final o no final
¿Cuándo una función recursiva lineal es final?	Es final si la llamada recursiva es la última operación que se efectúa, devolviéndose como resultado lo que se haya obtenido de la llamada recursiva sin modificación alguna.
¿Cuándo una función recursiva lineal es no final?	Cuando el resultado obtenido de la llamada recursiva se combina para dar lugar al resultado de la función que realiza la llamada
¿Cuándo una función recursiva es múltiple?	La recursión es múltiple cuando cada llamada recursiva, genera más de una llamada recursiva.