**Trabajo de Investigación**

**RECURSIVIDAD**

1. Se llama *recursividad* a un proceso mediante el que una función se llama a sí misma de forma repetida, hasta que se satisface alguna condición determinada. El proceso se utiliza para cálculos repetitivos en los que cada acción se determina en función de un resultado anterior.
2. El uso de la recursividad puede involucrar un compromiso entre simplicidad y rendimiento, dependiendo del problema en el cual se aplique. No es necesario definir la secuencia de pasos exacta para resolver el problema.
3. Cuando se ejecuta un programa recursivo, las llamadas recursivas no se ejecutan inmediatamente. Lo que se hace es colocarlas en una *pila* hasta que se encuentra la condición de terminación de la recursividad . Entonces se ejecutan las llamadas a la función en orden inverso a como se generaron, como si se fueran «sacando» de la pila. De lo contrario a la iteración que se van ejecutando las sentencias durante un numero determinado de veces, y generalmente  la salida de una iteración del proceso se utiliza como punto de inicio para la siguiente iteración.
4. El uso de la recursividad no es necesariamente la mejor aproximación a un problema, aunque la definición del problema sea recursiva. Una implementación no recursiva puede ser más eficiente, en cuanto a utilización de memoria y velocidad de ejecución.
5. Si una función recursiva contiene variables locales, se creará un conjunto *diferente* de variables locales durante cada llamada. Los nombres de las variables locales serán, por supuesto, siempre los mismos, como se hayan declarado en la función. Sin embargo, las variables representarán un conjunto diferente de valores cada vez que se ejecute la función. Cada conjunto de valores se almacenará en la pila; así se podrá disponer de ellas cuando el proceso recursivo se «deshaga», es decir, cuando las llamadas a la función se «saquen» de la pila y se ejecuten.

/\* calcular el factorial de una cantidad entera utilizando recursividad

\*/

#include <stdio.h>

long int factorial (int n);

int main (){

int n;

/\* leer la cantidad entera \*/

printf ("Ingrese valor a calcular factorial: ");

scanf ("%d", &n);

/\* calcular y visualizar el factorial \*/

printf( "n! = %ld \n", factorial (n)) ;

}

long int factorial(int n){

if (n <= 1){

return (1) ;}

else{

return(n \* factorial(n - 1));} la function se llama a si misma.

}

La función factorial se llama a sí misma recursivamente, con un argumento real *(n* - 1) que decrece en cada llamada sucesiva. Las llamadas recursivas terminan cuando el valor del argumento real se hace igual a 1.