

Ejercicio # 2

La recursividad es un concepto fundamental en la programación, donde una función se llama a sí misma en su propia definición.

En términos de matemáticas, el factorial de un número entero no negativo (n), denotado como ($n!$), es el producto de todos los números enteros positivos desde 1 hasta n . Se puede expresar de la forma:

$$n! = n \times (n - 1) \times (n - 2) \times \dots \times 2 \times 1$$

Está definido de la siguiente forma:

$$n! = \begin{cases} 1 & \text{si } n = 0 \\ n \times (n - 1)! & \text{si } n > 0 \end{cases}$$

Por ejemplo, el factorial de 5 se calcula como:

$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

Además, por definición, $0! = 1$.

P1. Escribir en pseudocódigo el algoritmo que calcule el factorial de un número ingresado por el usuario.

1. Defina el caso base: Es la condición que detiene la recursión. Sin esto, la función seguirá llamándose a sí misma indefinidamente, lo que lleva a un desbordamiento de pila que generalmente resulta en un error de programa.
2. Caso recursivo o caso general: En esta parte, la función se llama a sí misma.