

## Ejercicio 5.1.1

Calcular el factorial de un número positivo  $n$ . Tener en cuenta la definición matemática de factorial:

el producto de todos los números enteros positivos desde 1 (es decir, los números naturales) hasta  $n$ . Por ejemplo:

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

*//vemos q es ir multiplicando cada numero por su anterior, hasta llegar al uno. en base a eso genero mi caso base y rceursivo*

**Funcion** factorial (n: entero) : entero **es**

**si**  $n = 0$  **entonces** *//caso base → es la condicion de parada. la ultima llamada que se abre y la primera que se ejecuta*

factorial := 1 *// necesario q sea 1 para que se pueda multiplicar*

**contrario** *//caso recursivo*

factorial :=  $n * \text{factorial}(n-1)$  *//queda pendiente con que numero se multiplica a n, hasta q eventualmente llega al 1*

## Algoritmo Recursivo

### CASO BASE

- Problema trivial que se **puede resolver sin cálculo**
- Representa una **solución simple** para un caso particular

### CASO RECURSIVO

Representa una solución que **involucra volver a utilizar la función original**, con parámetros que se acercan más al caso base

sumarHasta(x)

DEFINICIÓN FORMAL

#### CASO BASE

1 Si  $x = 1$

#### CASO RECURSIVO

sumarHasta(x-1) + x Si  $x > 1$

# Pasos para resolver un ALGORITMO RECURSIVO

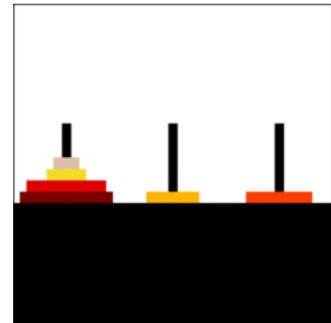
## PASO N° 1

El diseño debe responder a la pregunta

**¿Cómo se resuelve el caso más pequeño del problema?**

Esta respuesta debe ser **no-recursive**. Y define el **caso Base**.

Puede haber más de un caso base



## PASO N° 2

Organizar un **cambio de estado** que mueva el algoritmo hacia el caso base.

Un cambio de estado significa que se modifican algunos datos que el algoritmo está usando.

## PASO N° 3

**Establecer la llamada recursiva**

¿En qué punto el algoritmo se llamará a sí mismo y cómo?

## COMPONENTES DE LA SOLUCIÓN RECURSIVA



## DEFINICIÓN FORMAL

$$N! = \begin{cases} 1 & \text{si } N = 0 \text{ (caso base)} \\ N * (N - 1)! & \text{si } N > 0 \text{ (caso recursivo)} \end{cases}$$

# Método de las 3 preguntas

- **La pregunta Caso-Base**

*¿Existe una salida no recursiva o caso base del subalgoritmo? Además, ¿el subalgoritmo funciona correctamente para ella?*

- **La pregunta Más-pequeño**

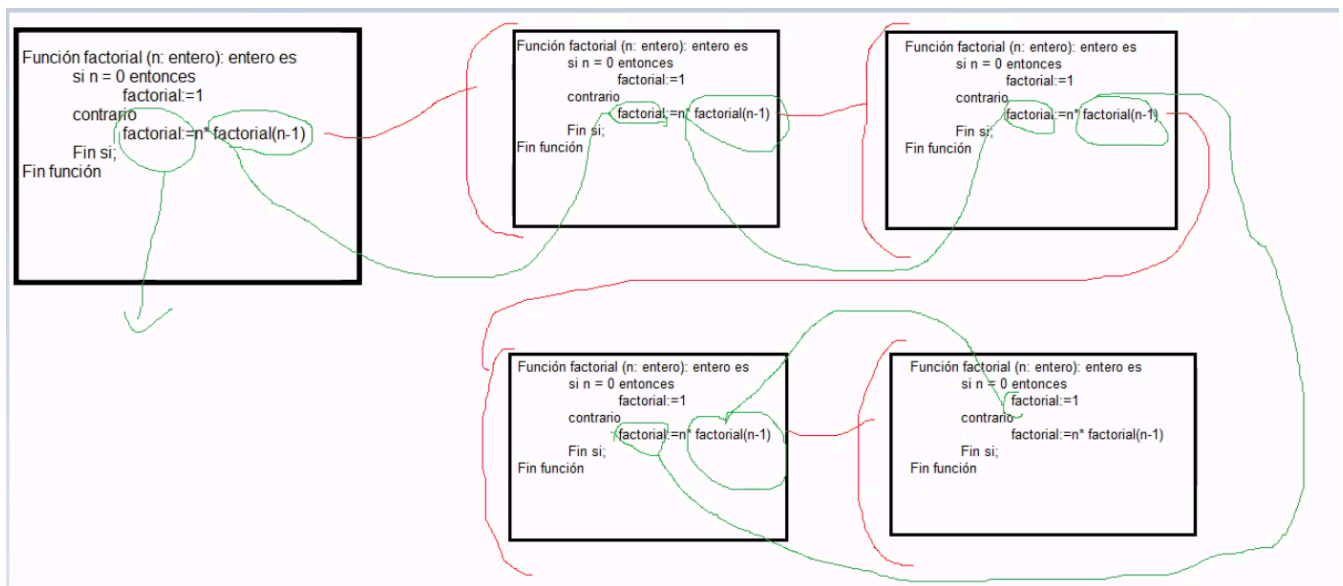
*¿Cada llamada recursiva se refiere a un caso más pequeño del problema original?*

- **La pregunta Caso-General**

*¿es correcta la solución en aquellos casos no base?*

resolución del factorial:

se hace una copia de esta llamada hasta llegar al caso base (línea roja) y después se hace desde la última copia abierta hasta la primera (línea verde)



**¿por qué ocupa mucha memoria?** cada variable usada en cada estado intermedio se guarda en memoria, en este caso el factorial