M3-UF2 –RECURSIVIDAD

http://codificando-sin-control.blogspot.com/2010/06/la-recursividad.html

UF2 –PROGRAMACION MODULAR

- ¿Qué es ?
- ¿Cómo diseñar un algoritmo recursivo?
- Ejemplo
- Ventajas
- Desventajas
- Conclusión
- Los guiños de Google

¿Qué es?

Alternativa diferente para implementar estructuras de repetición (ciclos). Se apoya en la modularidad, pues a través de los módulos se hacen llamadas recursivas.

Un módulo es recursivo si, como parte de su definición, incluye al menos una llamada a sí mismo

(Martínez, R. & Quiroga, E., 2001)

¿Qué es?

RECURSIVIDAD

- Es una técnica de programación que busca resolver un problema sustituyéndolo por otros problemas de la misma categoría, pero más simples.
- Se basa en que puedes resolver un problema, resolviendo el mismo problema en instancias más pequeñas.

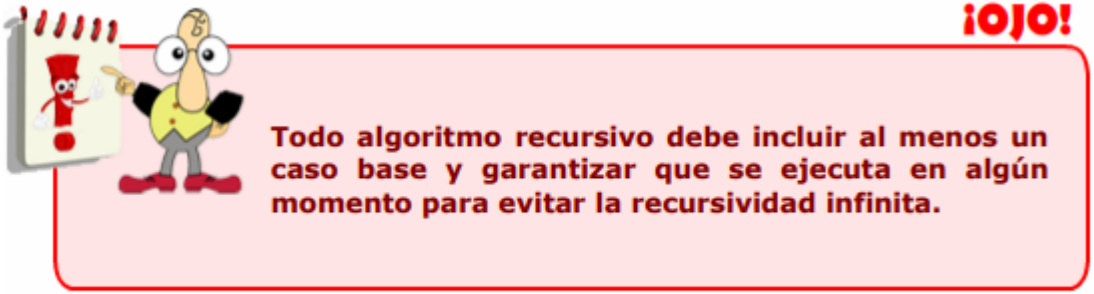
***Divide y Vencerás es una técnica algorítmica, la cual nos permite resolver un problema complejo, dividiendo en subproblemas más pequeños*

ALGORITMO RECURSIVO

- Se dice que un algoritmo es recursivo si dentro del cuerpo del algoritmo y de forma directa o indirecta se realiza una llamada a él mismo.
- **Todo algoritmo recursivo tiene su equivalente iterativo**

¿Cómo diseñar un algoritmo recursivo?

- Al escribir un algoritmo recursivo, debe establecerse de algún modo **cuando debe dejar de llamarse a sí mismo**, o de otra forma se repetiría indefinidamente.
- Para ello, se establece una **condición de salida** llamada caso base.
- Se llama **caso base** o condición de salida al caso trivial de un algoritmo recursivo, del cual **conocemos su solución**.



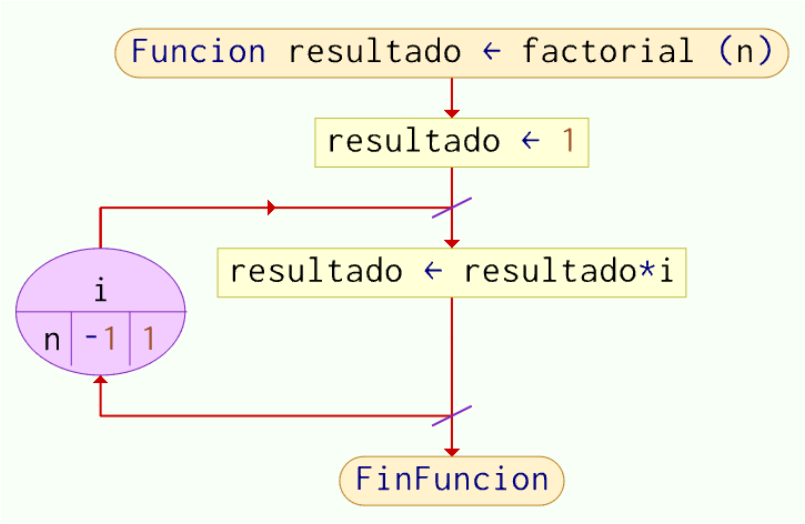
Ejemplo

- Ejemplo típico: Factorial (n!) de un número

  ¿Cómo se calcula el factorial de un número?

  Ejemplo: 3!

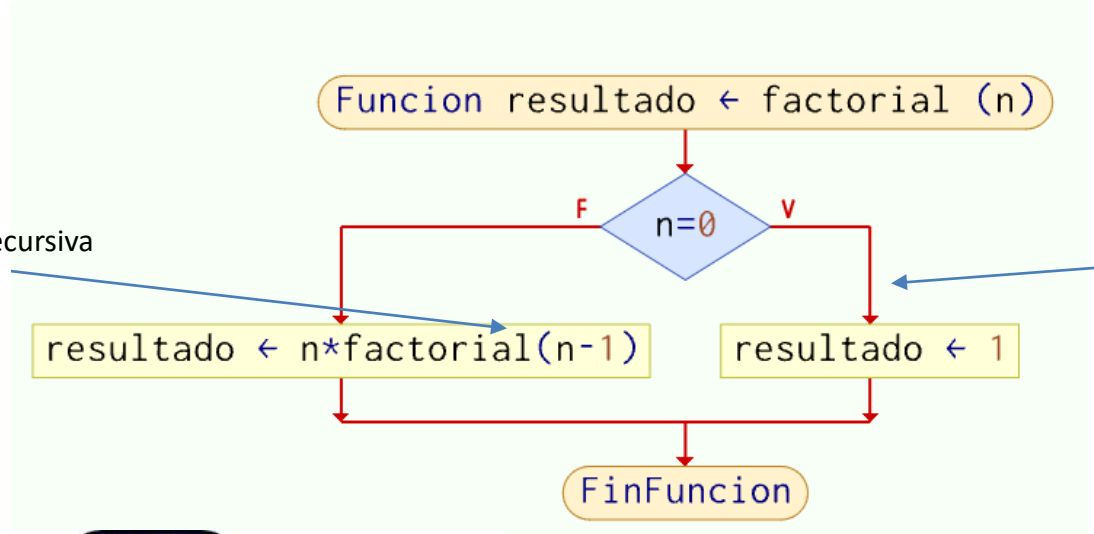
```
public static int factorial(int n){
    int resultado = 1;
    for (int i = n; i > 0; i--) {
        resultado *=i;
    }
    return (resultado);
}
```

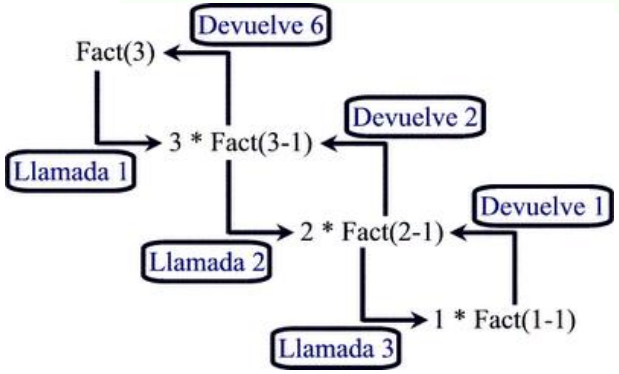


VERSION ITERATIVA

¿Qué es la recursividad?

VERSION RECURSIVA





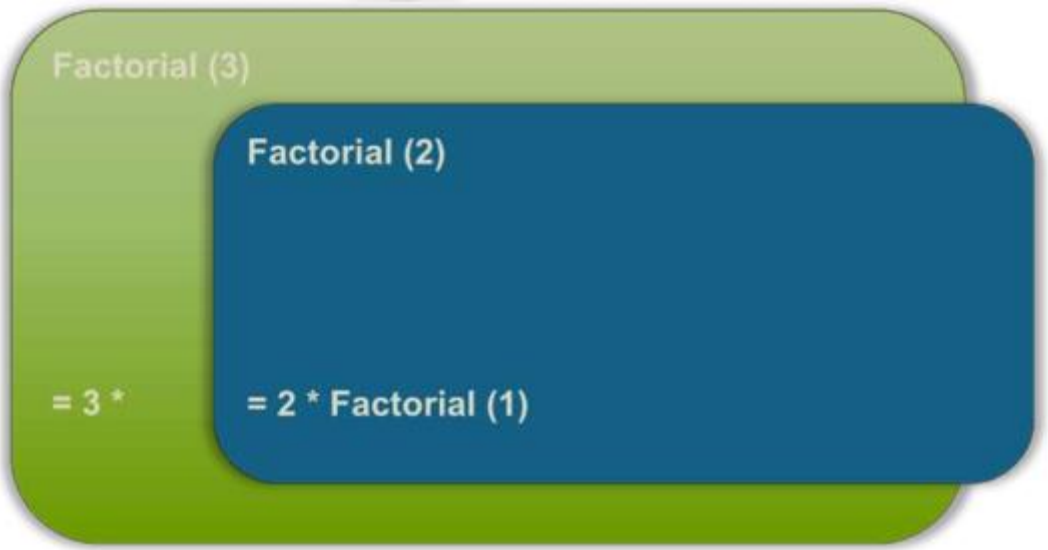

```
public static int factorial(int n) {
    int resultado;
    if (n == 0) {
        resultado = 1;
    } else {
        resultado = n*factorial(n - 1);
    }
    return resultado;
}
```

UF2 –PROGRAMACION MODULAR

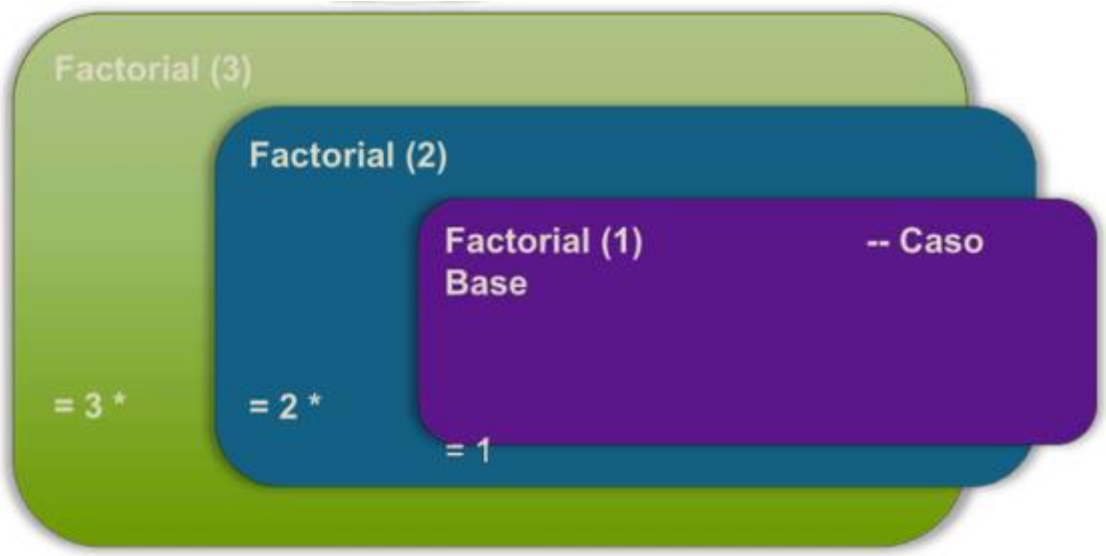
¿Qué es la recursividad?

UF2 –PROGRAMACION MODULAR

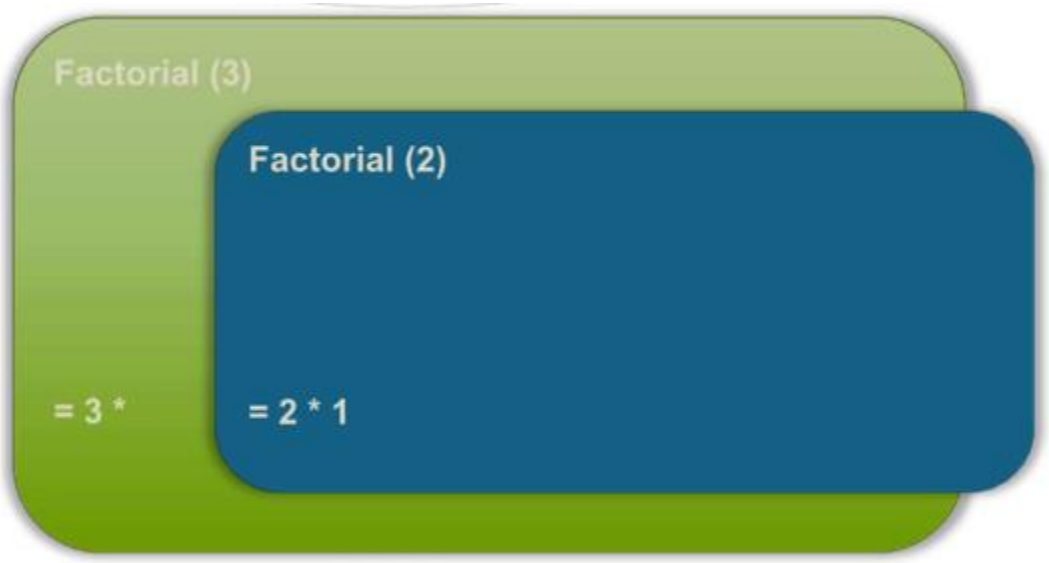
¿Qué es la recursividad?



¿Qué es la recursividad?



¿Qué es la recursividad?



¿Qué es la recursividad?

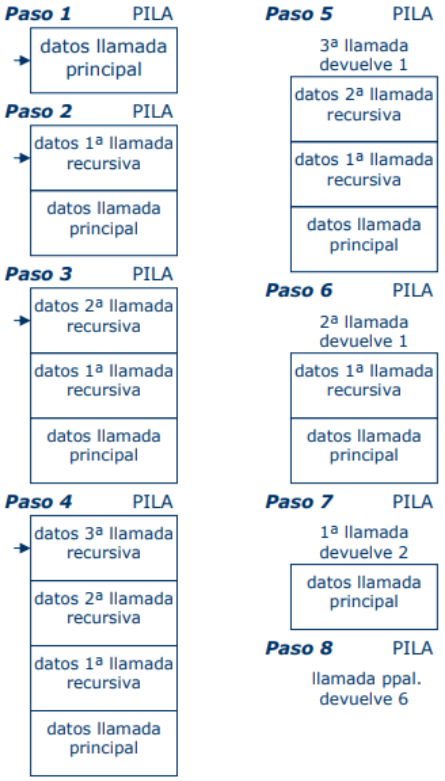
¿Qué es la recursividad?

Ventajas

- Menos líneas de código (más simples y elegantes).
  - Refleja el problema con más naturalidad.
  - Produce un programa más fácil de entender y depurar

Desventajas

- Tiempo de procesador
- Espacio en memoria, consume memoria adicional.
  - En cada llamada recursiva, se requiere espacio en la pila para almacenar dirección de retorno, parámetros y resultado de la llamada.
  - Conforme las llamadas devuelven su resultado, se libera el espacio asignado en la pila para dicha llamada



Conclusión

NOTA

De forma general, si la eficiencia es un factor importante y/o el algoritmo se va a ejecutar de forma frecuente, conviene escribir una solución iterativa.

Los guiños de Google

- Google es conocido por el espíritu bromista de sus desarrolladores, es por eso, que si buscáis en Google Recursion (que es recursividad en inglés, aunque también se puede decir así en castellano), el resultado que os devuelve es recursivo. ¡Probadlo!