



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

Escola Tècnica Superior d'Enginyeria Informàtica



# Tema 1. Recursividad

Programación (PRG)

Jorge González Mollá

Departamento de Sistemas Informáticos y Computación



# Índice

1. Introducción
2. Pila de Registros de Activación
3. Arrays
4. Recorrido
5. Búsqueda
6. Conclusiones

# Recursividad VS Iteratividad

- Un **algoritmo** es **recursivo** si obtiene la solución de un problema en base a los resultados que él mismo proporciona para casos más sencillos del mismo problema, esto es, si **se invoca a sí mismo** ...
- Así, para resolver un problema complejo mediante un algoritmo recursivo se descompone el problema en una serie de problemas más simples que se resuelven **empleando el mismo algoritmo**, para luego componer la solución del problema complejo en base a las soluciones de los problemas más simples.
- Tanto recursividad como iteratividad hacen uso de **estructuras de control**: **recursividad** usa como instrucción principal una **instrucción de selección** (condicional) e **iteratividad** usa como instrucción principal una **instrucción de repetición** (bucle).
- Ambas requieren una **condición de terminación**: la **condición del caso base** en **recursividad** y la **condición de la guarda del bucle** en **iteratividad**.
- Ambas se **aproximan** gradualmente **a la terminación**: **iteratividad** conforme se acerca al cumplimiento de la guarda y **recursividad** conforme se divide el problema en otros más pequeños, acercándose también al cumplimiento de una condición, en este caso la del Caso Base.
- Ambas pueden tener (por error) una **ejecución potencialmente infinita**.

# Recursividad VS Iteratividad

- Se puede demostrar que la solución algorítmica de cualquier problema algorítmicamente resoluble, se puede expresar recursivamente y también iterativamente.
- En este sentido, se dice que **recursividad** e **iteratividad** son **equivalentes**, y por ello, **alternativas**.
- En general, no es posible afirmar qué es lo más conveniente o sencillo.
- Es frecuente encontrar problemas para los que la solución **iterativa** es **más sencilla de estructurar** que la **recursiva**. Además, **recursividad** supone, en general, **más carga computacional** (espacio en memoria) que **iteratividad**.
- En otros casos, la versión **recursiva** refleja de manera **más natural**, **concisa** y **elegante** la solución al problema que la versión **iterativa**, lo que hace que sea más fácil de depurar y entender.
- Se puede concluir que **recursividad** e **iteratividad**, además de **alternativas**, son **complementarias**.