

# Recursividad

- Relatos dentro de relatos.
- Películas dentro de películas.
- Muñecas rusas, dentro de muñecas rusas.
- Sueños dentro de sueños.



Ana Lilia LAUREANO-CRUCES  
UAM-A

Lenguajes de Programación



# Sacas, Metes = Pila

- **Meter**, significa suspender las operaciones relativas a la tarea que se tiene entre manos, sin olvidar en punto en el que se esta ubicado y emprender otra tarea.
  - La última tarea se dice que esta ubicada en un nivel más bajo, que las anteriores.
- **Sacar**, significa lo opuesto, completar las operaciones correspondientes al primer nivel, reasumiéndolas en el punto exacto donde fueron suspendidas
  - Y ascendiendo para ello un nivel.

# Cómo recordar con precisión en que punto se estaba en cada diferente nivel

- Mediante el almacenamiento de la información pertinente en una pila.
- Una pila es el equivalente a un tablero que nos indica:
  - Dónde quedó suspendida cada tarea no terminada (dirección de retorno).
  - Cuáles son los hechos que deben ser conocidos (variables).
- Cuando se hace un regreso hacia arriba, la pila se encarga de restablecer el contexto correspondiente, eliminando toda confusión u olvido posible.

# Funciones recursivas

- Una función es recursiva si su cuerpo contiene una aplicación de  $f$
- $f$  es recursiva si  $f$  puede activarse a sí misma
- Existen dos tipos de recursión
  - lineal
  - cola

# Recursión lineal

- Si la activación  $f(a)$  de  $f$  puede iniciar como máximo una nueva activación de  $f$ .
- ejemplo:
- **fun** *factorial* (n) =
  - **si**  $n = 0$  **entonces** 1 **otro**  $n * \textit{factorial} (n-1)$ ;

- La evaluación de una función recursiva lineal tiene dos fases:
  - una fase de activación, en la cual se inician las nuevas activaciones, y
  - una fase de solución, en la cual el control regresa de las activaciones con una modalidad *última entrada - primera salida*

# Función factorial lineal

- Ejemplo:

- $f(3) = 3 * f(2)$
- $= 3 * (2 * (f(1)))$
- $= 3 * (2 * (1 * f(0)))$
- $= 3 * (2 * (1 * 1))$
- $= 3 * (2 * 1)$
- $= 3 * 2$
- $= 6$

# Recursión de cola

- una función recursiva puede ser eficiente si se puede implementar *con recursión de cola*
- si devuelve un valor sin necesidad de recursión o si devuelve simplemente el resultado de una activación recursiva

- ejemplo:

- **fun**  $g(n, a) =$

- **si**  $n = 0$  **entonces**  $a$  **otro**  $g(n-1, n*a)$

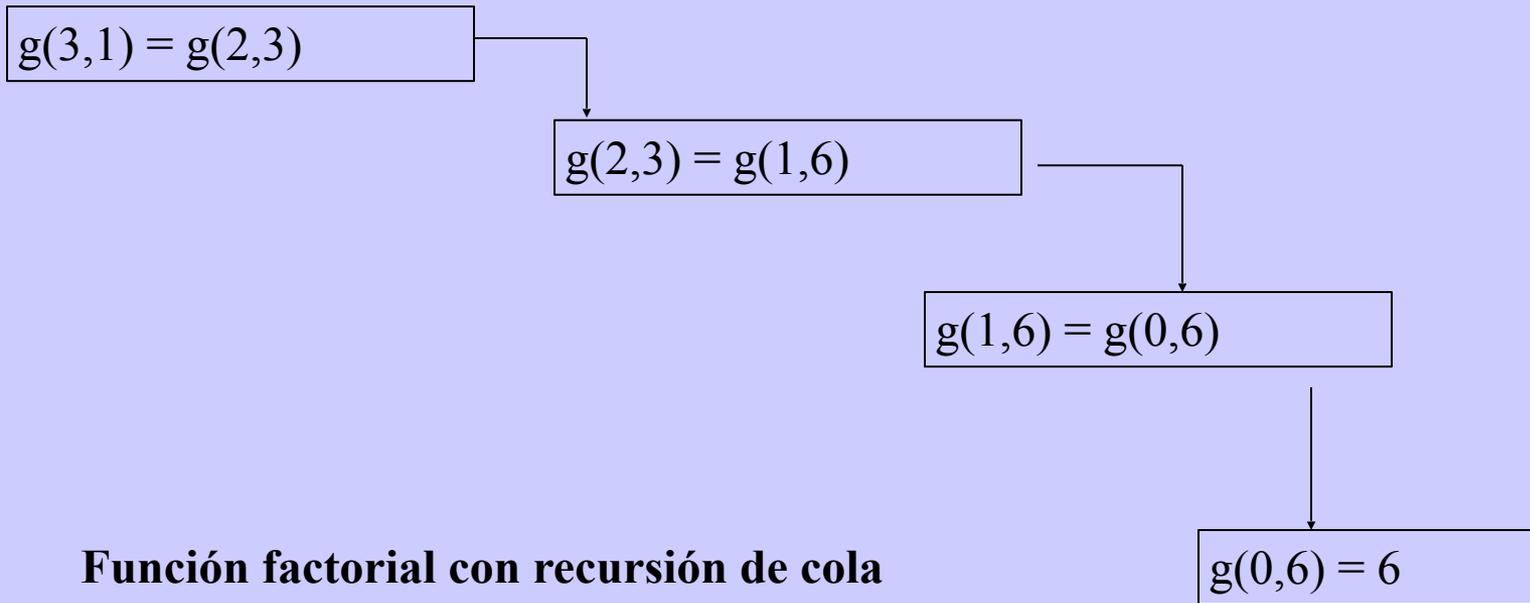
- **valor inicial de a=1**

- $a$  **si**  $n = 0$

- $g(n, a) =$

- $g(n-1, n*a)$  en caso contrario

- $g(3,1)$
- **si 3 entonces 1 otro**  $g(3-1, 3*1)$
- $g(3,1) = g(2,3) = g(1,6) = g(0,6) = 6$



- Todo el trabajo de una función lineal con recursión de cola se realiza en la fase de activación, cuando se inician las activaciones nuevas; siendo la fase de solución trivial debido a que el valor calculado por la activación final se convierte en el valor de toda la evaluación.

- En el caso de  $f(3) = 3 * f(2)$
- la multiplicación se realiza después de que el control regresa de la activación de  $f(2)$ .

