











**GUIA DE PROGRAMACION RECURSIVA**

Proponemos una estrategia formada por varias etapas. Se establecen reglas generales (que, por ser generales, están sujetas a excepciones)

Desarrollamos las etapas en paralelo con un ejemplo de uso:

Ejemplo >> Se pide una función recursiva para juntar dos números en uno.

**joint(351, 23) ---> 35123**

**joint(34, 987) ---> 34987**

//pre: m >= 0 y n >= 0

//post: retorna el numero cuyas primeras cifras son las de m y las ultimas las de n

int joint(int m, int n)

Etapas:

**Etapa 0**: (peliminar) El esqueleto de una funcion (o accion) recusiva es una estructura alternativa:

**if (...) .......**

**else if (...) ...**

**................**

**else if (...)**

**ese ...**

los casos sin llamada recursiva son los "casos directos". El resto de casos son los "casos recursivos"

**Etapa 1**: Los casos directos de obtienen respondiendo a "para que valores de los parametros es facil calcular el resultado?".

joint(m,n) es facil de calcular cuando n solo tiene un digito (n < 10). En este caso el resultado es

10\*m + n

Por ejemplo, joint(89, 7) es 10\*89 + 7 (= 897)

**if (n < 10) ....//caso directo**

**else .....//caso recursivo**

**Etapa 3**: resolvemos el caso recursivo en tres pasos

**3.a)** Proponemos una llamada recursiva. El argumento con el que se hace la llamada

\*\*\*ha de acercarse a uno de los casos directos\*\*\*\*

En nuestro ejemplo, una propuesta de llamada recursiva puede ser **joint(m, n/10).**

La propuesta de llamada joint(m, 10\*n) no tiene sentido porque no se acerca al caso base

**3.b)** Punto \*\*magico\*\*: suponemos que la llamada recursiva que proponemos funciona correctamente. Documentamos el resultado que obtenemos de la llamada en un comentario. (La \*\*magia\*\* equivale a la frase "por hipotesis de induccion" en una demostracion por induccion)

En nuestro ejemplo:

**if (n < 10) return 10\*m + n;**

**else {**

**int z = joint(m, n/10)**

**//z es el numero cuyas primeras cifras son las de m y sus ultimas son las de n/10**

**....**

**3.c)** Completamos el caso recursivo con el codigo necesario para que se cumpla la postconcicion. Si no se consigue, volvemos al paso 3.a)

En nuestro ejemplo, necesitamos completar el caso recursivo

**//z es el numero cuyas primeras cifras son las de m y sus ultimas son las de n/10**

**....**

**....**

**//post: retorna el numero cuyas primeras cifras son las de m y las ultimas las de n**

**return 10\*z + n%10**

Codigo completo:

**int joint(int m, int n) {**

**if (n < 10) return 10\*m + n;**

**else {**

**int z = joint(m, n/10);**

**//z es el numero cuyas primeras cifras son las de m y sus ultimas son las de n/10**

**return 10\*z + n%10;**

**}**





























