



Taller de Programación



AGENDA



Recursión



RECURSIÓN - Características



La recursión **es una metodología** para resolver problemas.

Permite resolver un problema P por resolución de instancias más pequeñas $P_1, P_2, \dots P_n$ del mismo problema.

El problema P_i es de la misma naturaleza que el problema original, pero en algún sentido es más simple.

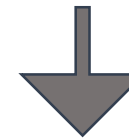


RECURSIÓN - Características



Suponga que debe realizar un módulo que devuelva el factorial de un número

El problema es siempre el mismo, sólo que se va achicando



Hasta cuando?

**CASO
BASE**

4 x 3 x 2 x 1

24

Factorial 3

3 x 2 x 1

6

Factorial 2

2 x 1

2

Factorial 1

1

Si el numero
es 4

$$\text{fac}(n) \begin{cases} 1 & \text{si } n=0 \text{ o } n=1 \\ n \times \text{fac}(n-1) & \end{cases}$$



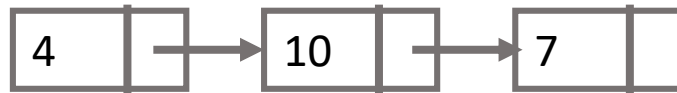
RECURSIÓN - Características



Suponga que debe realizar un módulo que imprima los elementos de una lista

SOLUCIÓN ITERATIVA

```
Procedure imprimir (L:lista);  
Begin  
  while (L <> nil) do  
    begin  
      write (L^.dato);  
      L:= L^.sig;  
    end;  
End;
```



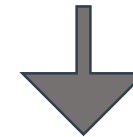
L

4

10

7

El problema es siempre el mismo, sólo que se va achicando



Hasta cuando?

**CASO
BASE**



RECURSIÓN - Características



El problema es siempre el mismo, pero debe ir achicándose.

Siempre tienen al menos un caso base, en el cual el código a implementar no es recursivo (a veces no debe escribirse código en el caso base, pero siempre existe). Pueden haber más casos base.

El caso base es el que determina el final de la solución recursiva



Suponga que debe realizar un módulo que devuelva la potencia de un número x elevado a la n (x^n). Cómo lo pensaría? Cuántos casos base hay?



RECURSIÓN - Características



Suponga que debe realizar un módulo que imprima el contenido de una lista.

SOLUCIÓN ITERATIVA

```
Procedure imprimir (L:lista);
Begin
  while (L <> nil) do
    begin
      write (L^.dato);
      L:= L^.sig;
    end;
  End;
```

Y
recursiva?

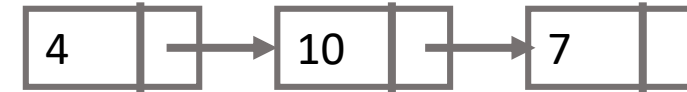
SOLUCIÓN RECURSIVA

```
Procedure imprimir (L:lista);
Begin
  IF (L <> nil) then
    begin
      write (L^.dato);
      L:= L^.sig;
      imprimir (L);
    end;
  End;
```

Cómo funciona?



RECUSIÓN - Características



```
Procedure imprimir (L:lista);  
Begin  
  IF (L <> nil) then  
    begin  
      write (L^.dato);  
      L:= L^.sig;  
      imprimir (L);  
    end;  
End;
```



Procedimiento imprimir	L= 4	4	3
Procedimiento imprimir	L= 10	10	3
Procedimiento imprimir	L= 7	7	3
Procedimiento imprimir	L= nil	En este caso no se hace nada	

Variables del programa

Programa principal

**Diferencia con la
solución secuencial**



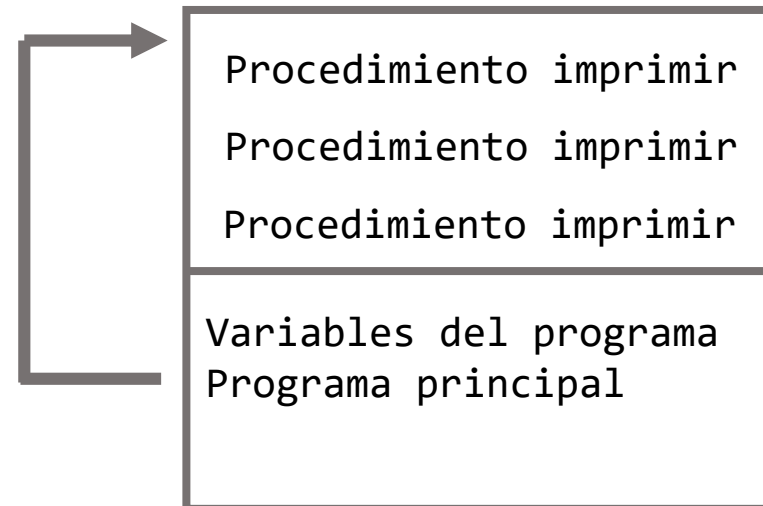
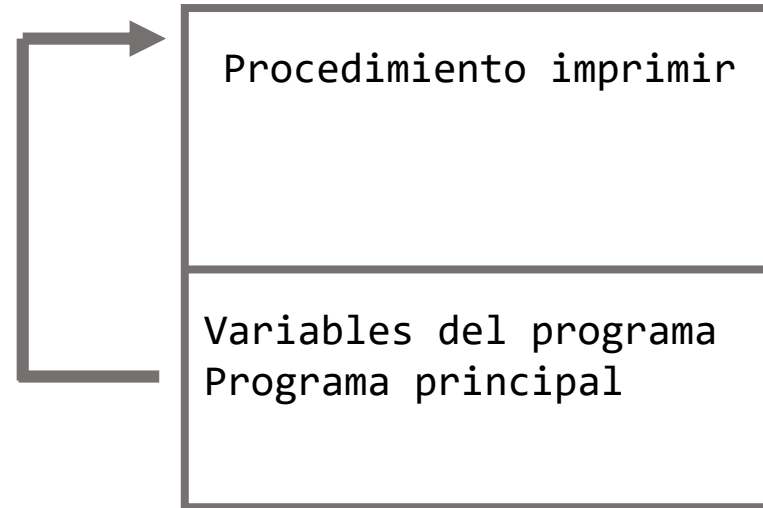
RECURSIÓN - Características

SOLUCIÓN ITERATIVA

```
Procedure imprimir (L:lista);  
Begin  
  while (L <> nil) do  
    begin  
      write (L^.dato);  
      L:= L^.sig;  
    end;  
End;
```

SOLUCIÓN RECURSIVA

```
Procedure imprimir (L:lista);  
Begin  
  IF (L <> nil) then  
    begin  
      write (L^.dato);  
      L:= L^.sig;  
      imprimir (L);  
    end;  
End;
```



Suponga que la lista tiene 2 elementos?



Cuál cree que es más eficiente?

Qué pasa con los parámetros?



RECURSIÓN - Características

SOLUCIÓN RECURSIVA

```
Procedure imprimir (x:integer);  
Begin  
  IF (x > 0) then  
    begin  
      write (x);  
      x:= x - 1;  
      imprimir (x);  
    end;  
End;
```

Supongamos $x = 4$



Imprimir $x = 4$, **imprime 4**
 $x := 3$

Imprimir $x = 3$, **imprime 3**
 $x := 2$

Imprimir $x = 2$, **imprime 2**
 $x := 1$

Imprimir $x = 1$, **imprime 1**
 $x := 0$

Imprimir $x = 0$, **no hace nada**



RECURSIÓN - Características

SOLUCIÓN RECURSIVA

```
Procedure imprimir (var x:integer);  
Begin  
  IF (x > 0) then  
    begin  
      write (x);  
      x:= x - 1;  
      imprimir (x);  
    end;  
End;
```

Supongamos $x = 4$



Imprimir $x = 4$, **imprime 4**
 $x := 3$

Imprimir $x = 3$, **imprime 3**
 $x := 2$

Imprimir $x = 2$, **imprime 2**
 $x := 1$

Imprimir $x = 1$, **imprime 1**
 $x := 0$

Imprimir $x = 0$, **no hace nada** **$x := 0$**



RECURSIÓN - Características



Suponga que debe realizar un módulo que dado un número entero y un valor n (entero positivo) se calcule x^n

SOLUCIÓN ITERATIVA

```
Procedure potencia (num,n:integer;var pot:integer);
Begin
  if (n = 0) then pot:= 1
  else
    begin
      pot:= 1;
      for i:- 1 to n do
        pot:= pot * num;
      end;
    end;
End;
```

SOLUCIÓN RECURSIVA

```
Procedure potencia (num,n:integer; var pot:integer);
Begin
  if (n = 0) then pot:= 1
  else
    begin
      potencia (num, (n-1), pot);
      pot:= pot * num;
    end;
End;
```

**Y con una
función?**



RECURSIÓN - Características



Suponga que debe realizar un módulo que dado un número entero y un valor n (entero positivo) se calcule x^n

SOLUCIÓN RECURSIVA

```
Function potencia (num,n:integer): integer;  
Var  
    pot:integer;  
  
Begin  
    if (n = 0) then potencia:= 1  
    else  
        potencia:= num * potencia(num, n-1));  
    end;  
End;
```

**Y como
funciona?**



RECURSIÓN - Características

SOLUCIÓN RECURSIVA

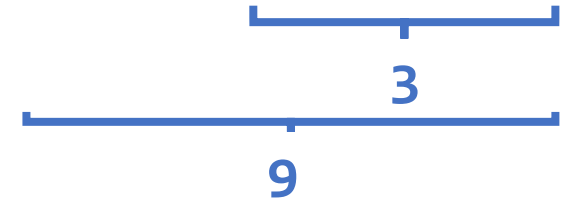
```
Function potencia (num,n:integer): integer;  
Var  
    pot:integer;  
  
Begin  
    if (n = 0) then potencia:= 1  
    else  
        potencia:= num * potencia(num, n-1));  
    end;  
End;
```

Supongamos num = 3 n=2

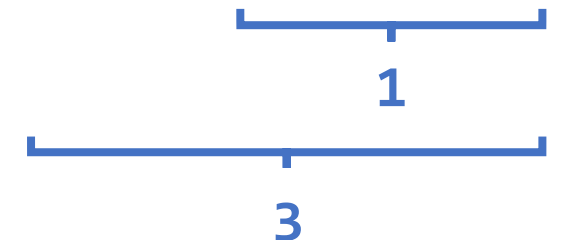
9



potencia num= 3,n=2, **3 * potencia (3,1)**



potencia num= 3,n=1, **3 * potencia (3,0)**



potencia num= 3,n=0, **potencia = 1**