



Universidad Nacional de Jujuy Facultad de ingeniería Introducción a la informática

Clase 13

UNJu-FI-Introducción a la Informática
Samuel Franco-José Zapana

1

Recursividad

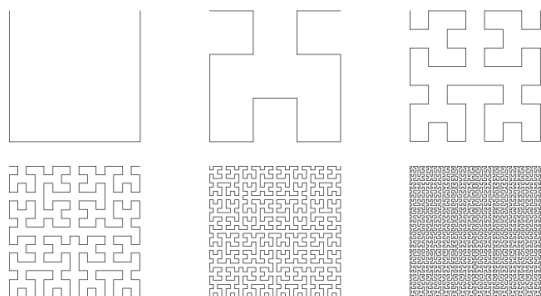
La recursividad es un modo de definir o especificar *algo* en términos de sí mismo.



UNJu-FI-Introducción a la Informática
Samuel Franco-José Zapana

2

Hilbert

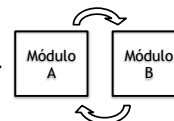
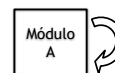


UNJu-FI-Introducción a la Informática
Samuel Franco-José Zapana

3

Definición

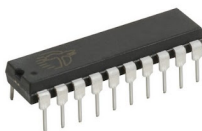
- Recursividad Directa
 - Cuando un módulo se llama a sí mismo.
 - Con una condición de salida
- Recursividad Indirecta
 - Cuando un módulo llama a otro módulo y éste llama al primero.
 - Con una condición de salida.



UNJu-FI-Introducción a la Informática
Samuel Franco-José Zapana

4

Recurso Stack



UNJu-FI-Introducción a la Informática
Samuel Franco-José Zapana

5

Recursividad

El diseño de un algoritmo recursivo consiste en identificar:

Los casos triviales

Las instancias recursivas

Cuando el problema está definido de manera recursiva, la implementación es directa.

UNJu-FI-Introducción a la Informática
Samuel Franco-José Zapana

6

Factorial

$$N! = \begin{cases} 1 & \text{si } N = 0 \text{ (base)} \\ N * (N - 1)! & \text{si } N > 0 \text{ (recursión)} \end{cases}$$

$n! = n * (n-1) * (n-2) * \dots * 1$

$4! = 4 * 3 * 2 * 1$

$3! = 3 * 2 * 1$

$2! = 2 * 1$

$1! = 1$

$0! = 1$

$4! = 4 * 3!$

$3! = 3 * 2!$

$2! = 2 * 1!$

$1! = 1$

$0! = 1$

Casos Base

UNJu-FI-Introducción a la informática
Samuel Franco-José Zapana

7

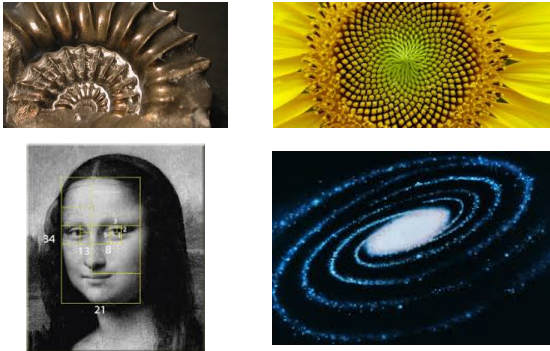
Recursividad

```
funcion FactRec (N): entero
[entero: N  parámetro por valor]
inicio
  si N = 0
    entonces
      FactRec ← 1
    sino
      FactRec ← N * FactRec (N-1)
  fin
fin [FactRec]
```

UNJu-FI-Introducción a la informática
Samuel Franco-José Zapana

8

Fibonacci



UNJu-FI-Introducción a la informática
Samuel Franco-José Zapana

9

Fibonacci

"Cierta hombre tenía una pareja de conejos juntos en un lugar cerrado y uno desea saber cuántos son creados a partir de este par en un año cuando es su naturaleza parir otro par en un simple mes, y en el segundo mes los nacidos parir también"

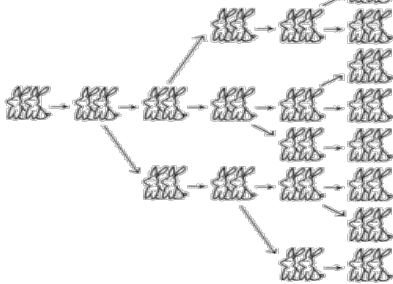


UNJu-FI-Introducción a la informática
Samuel Franco-José Zapana

10

Fibonacci

time = 012345



pairs = 112358

UNJu-FI-Introducción a la informática
Samuel Franco-José Zapana

11

Fibonacci

En la sucesión de Fibonacci los dos primeros términos son 1 y cada uno de los que siguen resultan de sumar los dos anteriores.

1 1 2 3 5 8...

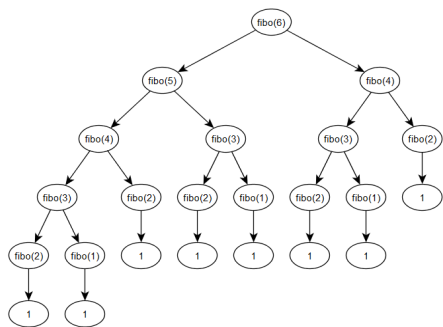
El nésimo término de la sucesión de Fibonacci se define como:

$$\text{fib}(n) = \begin{cases} 1 & n = 1 \text{ o } n = 2 \\ \text{fib}(n-1) + \text{fib}(n-2) & n > 2 \end{cases}$$

UNJu-FI-Introducción a la informática
Samuel Franco-José Zapana

12

Fibonacci



UNJu-FI-Introducción a la informática
Samuel Franco-José Zapana

13

Fibonacci

```
funcion fib (n): entero
[entero: n parámetro por valor]
inicio
  si (n = 1) O (n = 2)
    entonces
      fib ← 1
    sino
      fibo ← fib (n-2) + fib (n-1)
  fin
fin [fib]
```

UNJu-FI-Introducción a la informática
Samuel Franco-José Zapana

14

Revés recursivo

X	X/10	Resto (X, 10)
1936	193	6
193	19	3
19	1	9
1	0	1

UNJu-FI-Introducción a la informática
Samuel Franco-José Zapana

15

Revés recursivo

```
Procedimiento Reves (X)
[entero: X parámetro por valor]
inicio
  si X < 10
    entonces
      Escribir X
    sino
      Escribir Resto (X, 10)
      Reves (X/10)
  fin
fin [Reves]
```

UNJu-FI-Introducción a la informática
Samuel Franco-José Zapana

16

Binario recursivo

W	W/2	Resto (W, 2)
13	6	1
6	3	0
3	1	1
1	0	1

UNJu-FI-Introducción a la informática
Samuel Franco-José Zapana

17

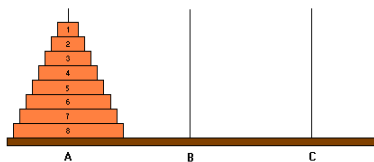
Binario recursivo

```
Procedimiento Bin (W)
[entero: W parámetro por valor]
inicio
  si W < 2
    entonces
      Escribir W
    sino
      Bin (W/2)
      Escribir Resto (W, 2)
  fin
fin [Bin]
```

UNJu-FI-Introducción a la informática
Samuel Franco-José Zapana

18

Torres de Hanoi



1. Sólo se puede mover un disco cada vez.
2. Un disco de mayor tamaño no puede descansar sobre uno más pequeño que él mismo.
3. Sólo puedes desplazar el disco que se encuentre arriba en cada varilla

UNJu-FI-Introducción a la informática
Samuel Franco-José Zapana

19

Torres de Hanoi

Algoritmo Torres de Hanói (Complejidad $\Theta(2^n)$)

Entrada: Tres pilas de números *origen*, *auxiliar*, *destino*, con la pila *origen* ordenada

Salida: La pila *destino*

1. si *origen* == {1} entonces
 1. mover el disco 1 de pila origen a la pila destino (insertarlo arriba de la pila destino)
 2. terminar
2. si no
 1. *hanoi*($\{1, \dots, n-1\}$, *origen*, *destino*, *auxiliar*) //mover todas las fichas menos la más grande (*n*) a la varilla auxiliar
 2. mover disco *n* a *destino* //mover la ficha grande hasta la varilla final
 3. *hanoi*(*auxiliar*, *origen*, *destino*) //mover todas las fichas restantes, $1 \dots n-1$, encima de la ficha grande (*n*)
3. terminar

UNJu-FI-Introducción a la informática
Samuel Franco-José Zapana

20

Torres de Hanoi

-->*hanoi*(3,a,b,c)

mover disco 1 desde la columna a hasta c.
mover disco 2 desde la columna a hasta b.
mover disco 1 desde la columna c hasta a.
mover disco 3 desde la columna a hasta c.
mover disco 1 desde la columna b hasta a.
mover disco 2 desde la columna b hasta c.
mover disco 1 desde la columna a hasta c.

UNJu-FI-Introducción a la informática
Samuel Franco-José Zapana

21

Torres de Hanoi

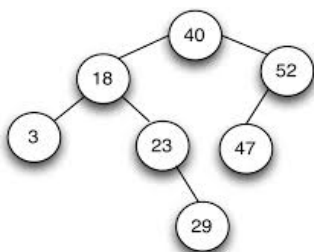
-->*hanoi*(4,a,b,c)

mover disco 1 desde la columna a hasta b.
mover disco 2 desde la columna a hasta c.
mover disco 1 desde la columna b hasta c.
mover disco 3 desde la columna a hasta b.
mover disco 1 desde la columna c hasta a.
mover disco 2 desde la columna c hasta b.
mover disco 1 desde la columna a hasta b.
mover disco 4 desde la columna a hasta c.
mover disco 1 desde la columna b hasta c.
mover disco 2 desde la columna b hasta a.
mover disco 1 desde la columna c hasta a.
mover disco 3 desde la columna b hasta c.
mover disco 1 desde la columna a hasta b.
mover disco 2 desde la columna a hasta c.
mover disco 1 desde la columna b hasta c.

UNJu-FI-Introducción a la informática
Samuel Franco-José Zapana

22

Ordenación



UNJu-FI-Introducción a la informática
Samuel Franco-José Zapana

23

Ventajas y desventajas

- **Cuando usar recursividad**
 - Cuando el problema es de naturaleza recursiva, por ejemplo árboles
- **Cuando NO usar recursividad**
 - Cuando las iteraciones sean la mejor opción
 - Cuando las llamadas recursivas no llenen el *stack*

UNJu-FI-Introducción a la informática
Samuel Franco-José Zapana

24