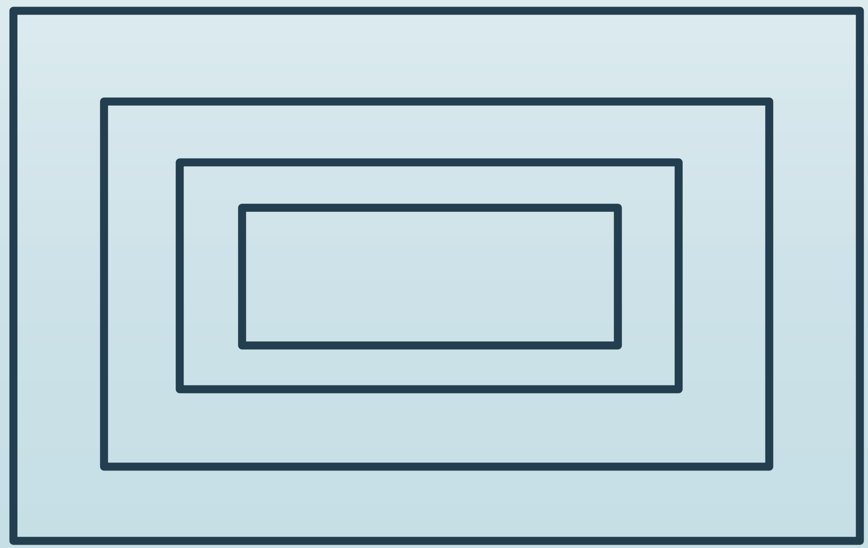
Unidad 2

Recursividad



Competencia específica

- Aplica la recursividad en la solución de problemas valorando su pertinencia en el uso eficaz de los recursos.

Unidad 2 contenido

Tema	Subtema
Recursividad	2.1 Definición 2.2 Procedimientos recursivos 2.3 Ejemplos de casos recursivos

Recursividad

Es la definición de un objeto en términos de sí mismo.

Ejemplo Matrushka

- La Matrushka es una artesanía tradicional rusa. Es una muñeca de madera que contiene otra muñeca más pequeña dentro de sí. Esta muñeca, también contiene otra muñeca dentro. Y así, una dentro de otra.

Métodos Recursivos

- Se dice que un proceso es recursivo, si forma parte de sí mismo, o sea que se define en función de sí mismo.

  - Problemas matemáticos.
  - Estructuras de datos.

- El factorial de un número
- La serie de Fibonacci
- Máximo común divisor con el algoritmo de Euclides
- La transformada de Fourier
- Cálculo de la suma de todos los números desde 1 a *n*
- 2 a la potencia de un número entero positivo
- Cualquier número a la potencia de un número entero positivo
- Etc.

Métodos recursivos en Java

- Se dice que un método es recursivo si se llama así mismo.

- El compilador Java permite cualquier número de llamadas recursivas a un método. Cada vez que el método es llamado, sus parámetros y variables locales son iniciadas.

- La recursión es un proceso extremadamente potente pero consume muchos recursos.

- Aunque un problema por definición sea recursivo, no siempre será el método de solución mas adecuado.

- En las aplicaciones prácticas, antes de poner en marcha un proceso recursivo es necesario demostrar que el nivel máximo de recursión, esto es, el número de veces que se va a llamar a sí mismo no solo es finito, si no realmente pequeño.

- La razón es que se requiere cierta cantidad de memoria.

Recursividad

Directa

Cuando una función se llama a sí misma una o más veces directamente.

Indirecta o mutua

Cuando una función llama a otra y ésta llama a su vez a la que la llamó.

- Calculo del factorial (ciclo for).

- 4!=1x2x3x4
- 5!=1x2x3x4x5
- 8!=1x2x3x4x5x6x7x8
- 100!=1x2x3x…x100
- n!=1x2x3x4x…x(n-1)x(n)
- **n! =(n-1)! *n**

```
int factorial( int n) {
 int fac=1;
 for(int i=1; i<=n; i++)
  {
     fac= fac * i;
  }
 return fac;
}
```

- ¿Cuándo es eficaz escribir un método recursivo?

R= Cuando el proceso del programa sea por definición recursivo. Por ej. El cálculo del factorial de un número.

- ¿Cómo se plantean soluciones recursivas a los problemas de repetición?

Respondiendo a las siguientes preguntas:

1.- ¿Cómo se resuelve el caso mas pequeño del problema?

2.- ¿Cómo se resuelve un caso general del problema, suponiendo que ya se tiene la solución al siguiente caso mas pequeño del problema?

- Ejercicios
- Análisis y solución de problemas recursivos

Ejercicios

- Cálculo del factorial de n
- Escribir una función recursiva que devuelva la suma de los primeros N enteros positivos.
- Calcular v^n.
- Obtener el enésimo número de Fibonacci
- Implementa un método recursivo para sumar los dígitos de un número entero positivo.
- Escribir un método recursivo que muestre la serie de los primeros n números enteros positivos
- Multiplicación por sumas sucesivas recursivo

Ejemplo 1Cálculo del factorial de n solución recursiva

Recordemos:

- Calculo del factorial (ciclo for).

- 4!=1x2x3x4

- 5!=1x2x3x4x5
- 8!=1x2x3x4x5x6x7x8
- 100!=1x2x3x…x100
- n!=1x2x3x4x…x(n-1)x(n)
- **n! =n(n-1)!**

Cálculo del factorial de n solución recursiva

1.- ¿Cómo se resuelve el caso mas pequeño del problema?

R: $0! = 1$

2.- ¿Cómo se resuelve un caso general del problema, suponiendo que ya se tiene la solución al siguiente caso mas pequeño del problema?

R: $n! = n\underbrace{(n-1)!}$

mensaje recursivo

```
public int factorial (int n) {
      if (n==0)
         return 1;
      else
         return n* factorial(n-1);

  }
```

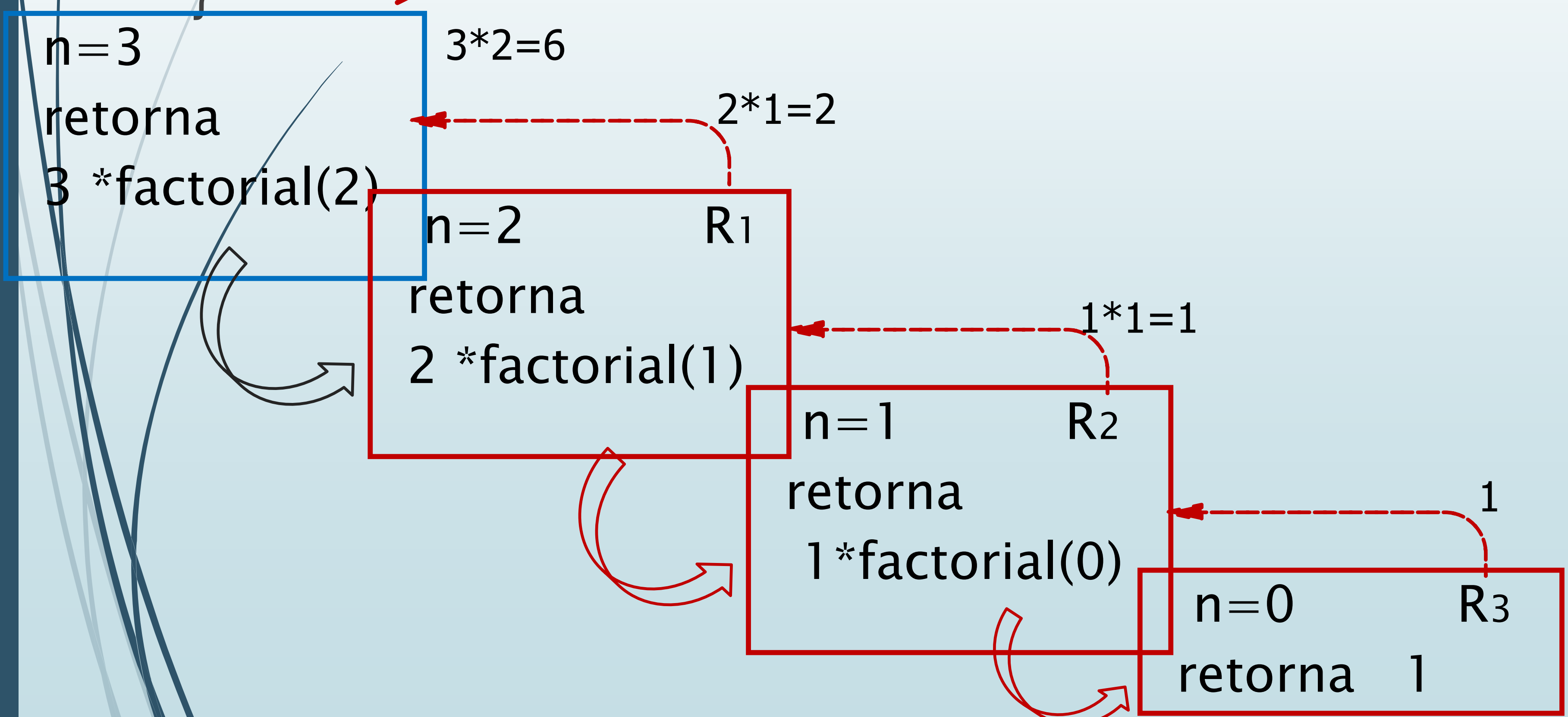
Si llamas por ejemplo al factorial de 3 el proceso es:

- R=ob.factorial(3);

```java
public int factorial (int n) {
    if (n==0)
        return 1;
    else
        return n* factorial(n-1);
}
```

R=ob.factorial(3);

R= 6;

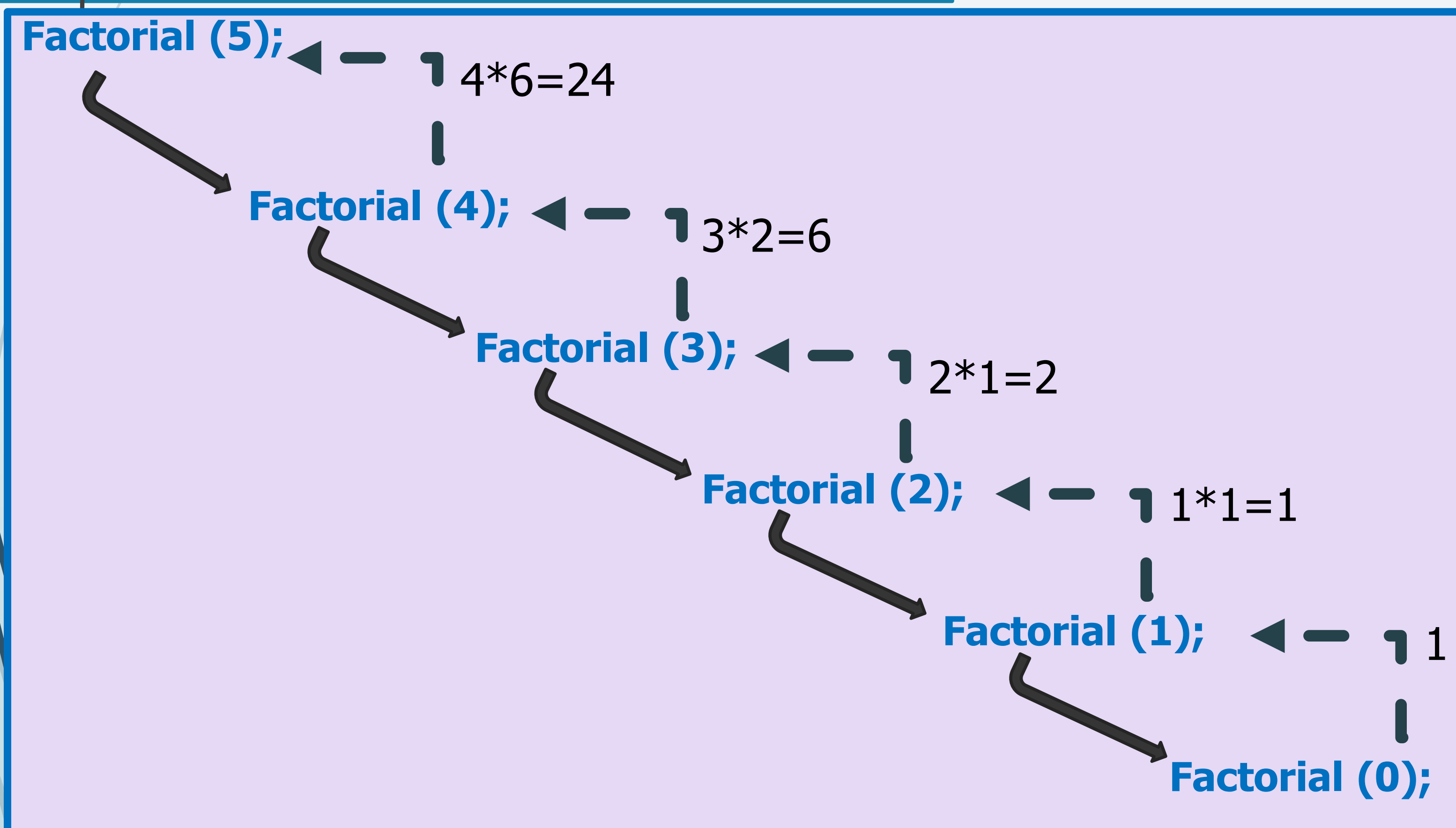


```
public int factorial (int n) {
    if (n==0)
        return 1;
    else
        return n* factorial(n-1);
}
```

5*24=120

R=ob.factorial(5);

R=120;



Implementación en java

- **Extra clase**

Ejemplo 2
Escribir un método recursivo que devuelva la suma de los primeros N enteros positivos.

$S_1 = 1$ $\qquad S_2 = S_1 + 2$

$S4 = 1+2+3 + 4$

$S_4 = S_3 + 4$

$S5 = \underbrace{1+2+3+4}_{S_4} + 5$

$S_5 = S_4 + 5$

$S9 = 1+2+\ldots+8+9$

$S_n = 1+2+\ldots+ (n-1)+ n$

$Sn = S(n-1)+ n$

Mensaje recursivo

Cálculo de la suma de los primeros N enteros positivos.

1.- ¿Cómo se resuelve el caso mas pequeño del problema?

R: **$S1 = 1$**

2.- ¿Cómo se resuelve un caso general del problema, suponiendo que ya se tiene la solución al siguiente caso mas pequeño del problema?

R:

$$Sn = \mathbf{S(n-1) + n}$$

mensaje recursivo

Código java

```
public int suma_n(int n) {
 if(n==1)
   return 1;
   else
     return suma_n(n-1) + n;
}
```

R=ob.suma_n(3);   R → $\boxed{6}$

=====================

n=3
Retorna **suma_n(2)**+3
= 3+3
=6

===================

n=2
Retorna **suma_n(1)**+2
= 1+2
=3

===================

n=1
Retorna 1

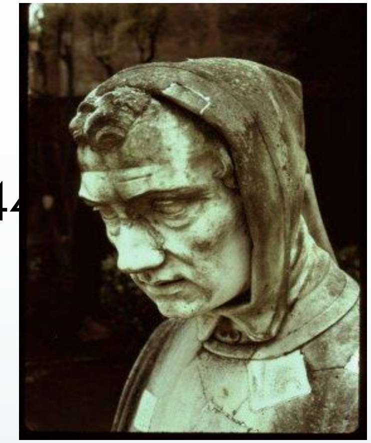
Actividad 2.1 Implementación en Java

- Implementa un proyecto en Java que mediante un objeto permita acceder a los métodos recursivos de los ejemplos revisados en clase.
- (Factorial de un número, Suma de los primeros n números, número de Fibonacci, suma dígitos)

Serie Fibonacci

La sucesión de Fibonacci

0,1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144



- A finales del siglo XII, la república de Pisa es una gran potencia comercial, con delegaciones en todo el norte de Africa. En una de estas delegaciones, en la ciudad argelina de Bugía, uno de los hijos de Bonaccio, el responsable de la oficina de aduanas en la ciudad, Leonardo, es educado por un tutor árabe en los secretos del cálculo posicional hindú y tiene su primer contacto con lo que acabaría convirtiéndose, gracias a él, en uno de los más magníficos regalos del mundo árabe a la cultura occidental: nuestro actual sistema de numeración posicional.

http://www.ite.educacion.es/formacion/enred/web_espiral/naturaleza/vegetal/fibonacci/fibonacci.htm

- **Leonardo de Pisa, Fibonacci**, nombre con el que pasará a la Historia. Fibonacci es más conocido entre los matemáticos por una curiosa sucesión de números:

  0; 1; 1; 2; 3, 5; 8; 13; 21; 34; 55; 89....

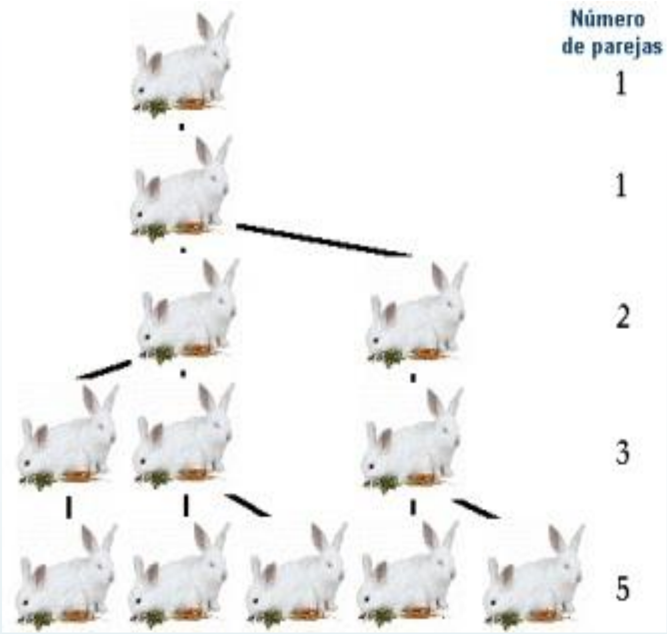
que colocó en el margen de su Liber abaci junto al conocido "problema de los conejos" que más que un problema parece un acertijo de matemáticas recreativas. El problema en lenguaje actual diría:

- ***"Una pareja de conejos tarda un mes en alcanzar la edad fértil, a partir de ese momento cada vez engendra una pareja de conejos, que a su vez, tras ser fértiles engendrarán cada mes una pareja de conejos. ¿Cuántos conejos habrá al cabo de un determinado número de meses?."***

http://www.ite.educacion.es/formacion/enred/web_espiral/naturaleza/vegetal/fibonacci/fibonacci.htm

- En este gráfico vemos que el número de parejas a lo largo de los meses coincide con los términos de la sucesión. Veamos con detalle estos números.

  1; 1; 2; 3, 5; 8; 13; 21; 34; 55; 89, 144....



- Es fácil ver que cada término es la suma de los dos anteriores.

- Pero existe entre ellos otra relación curiosa, el cociente entre cada término y el anterior se va acercando cada vez más a un número muy especial, ya conocido por los griegos y aplicado en sus esculturas y sus templos:

  el número áureo. =1.618039....

-

Ejemplo 3 (en clase)

- Escriba una definición recursiva de un método que tiene un parámetro n de tipo entero y que devuelve el n-ésimo número de **Fibonacci.**

- Los números de **Fibonacci** se definen de la siguiente manera:

$F_0 = 0$

$F_1 = 1$

$F_2 = 1$

$F_3 = 2$

$F_4 = 3$

$F_5 = 5$

. . .

$F_n =$ ¿

El método tiene un parámetro n de tipo entero y que devuelve el n-ésimo número de **Fibonacci.**

números de **Fibonacci**:

$F_0 = 0$

$F_1 = 1$

$F_2 = 1 = F_0 + F_1$

$F_3 = 2$

$F_4 = 3$

$= F_2 + F_3$

$F_5 = 5$

$F_5 = F_3 + F_4$

. . .

$\mathbf{F_n = ? = F_{n-1} + F_{n-2}}$

$F_{i+2} = F_i + F_{i+1}$

Metodología de solución

Caso más pequeño?

$F_0 = 0$

$F_1 = 1$

Caso general?

$F_n = F_{n-1} + F_{n-2}$

```
public int Fibonacci(int n) {
  if (n==0 || n==1)
       return n;

  else
       return Fibonacci(n-1) + Fibonacci(n-2);

}
```

Ejemplo 4

Implementa un método recursivo para sumar los dígitos de un número entero positivo.

- Entrada:12348

proceso: 1+2+3+4+8=18

Salida 18

15 1+5=6

10 1+0=1

9 9 =9

...

2 2=2

1 1=1

n n= ?

12348

12348	? = 8
1234	? = 4
123	? = 3
12	? = 2
1	? = 1

Ejemplo 4

Implementa un método recursivo para para sumar los dígitos de un número entero positivo.

- Entrada:12348

proceso:
1+2+3+4+8=18

Salida 18

15 1+5=6

10 1+0=1

9 9 =9

...

2 2=2

1 1=1

n n= ?

cociente

1234

10 | 12348
23
34
48
8

residuo

12348

12348 % ? = 8

12348 % **10** = 8
1234 % 10 = 4
123 % 10 =3
12 % 10 =2
1 % 10 =1

123

10 | 1234
23
34
4

12

10 | 123
23
3

1

10 | 12
2

0

10 | 1
1

sumadigitos n= sumadigitos(n/10) + (n%10)

Cálculo de la suma de lo dígitos de un número entero positivo.

SOLUCION RECURSIVA

1.- ¿Cómo se resuelve el caso mas pequeño del problema?

R: $sumadigitos_0 = 0$ Si n es 0, resultado es 0

2.- ¿Cómo se resuelve un caso general del problema, suponiendo que ya se tiene la solución al siguiente caso mas pequeño del problema?

R:

$sumadigitos_n =$ **sumadigitos(n/10) + (n%10)**

mensaje recursivo

```
Int sumadigitos (int n)  {
  if (n==0)
        return n;
  else
        return sumadigitos ( n/10 ) +  ( n % 10 );
}
```

Actividad 2.2
Resuelve el siguiente ejercicio y envia

- Se publicará en Teams.

Recursividad en las estructuras de datos

- Muchas estructuras de datos tienen una definición recursiva y por tanto, la recursividad es una buena opción para implementar algunas de sus operaciones.

- La recursividad puede aplicarse tanto en estructuras lineales, como en estructuras jerárquicas, principalmente en operaciones que implican búsquedas y recorridos.

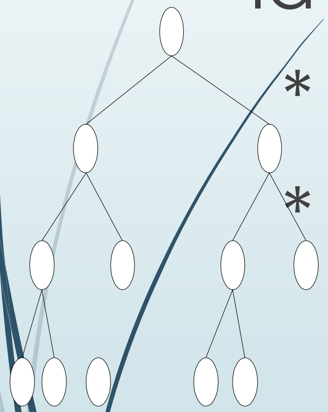
- Problema:
  - Escribe un método para contar los nodos de una lista de enlace simple.
  - Escribe un método recursivo para contar los nodos de una lista de enlace simple.

ejercicio

- Escribe un método recursivo para imprimir los nodos de una lista de enlace simple.

Recursividad en estructuras de datos Jerárquicas

- La recursividad en árboles Binarios generalmente puede ser planteada de la siguiente forma:



* Caso mas pequeño: árbol vacío

* Caso general: se aplica la operación recursiva sobre el subárbol izquierdo(raíz.izq) y sobre el subárbo derecho (raíz.der)

Ejercicios

- Imprimir el contenido de un árbol binario.