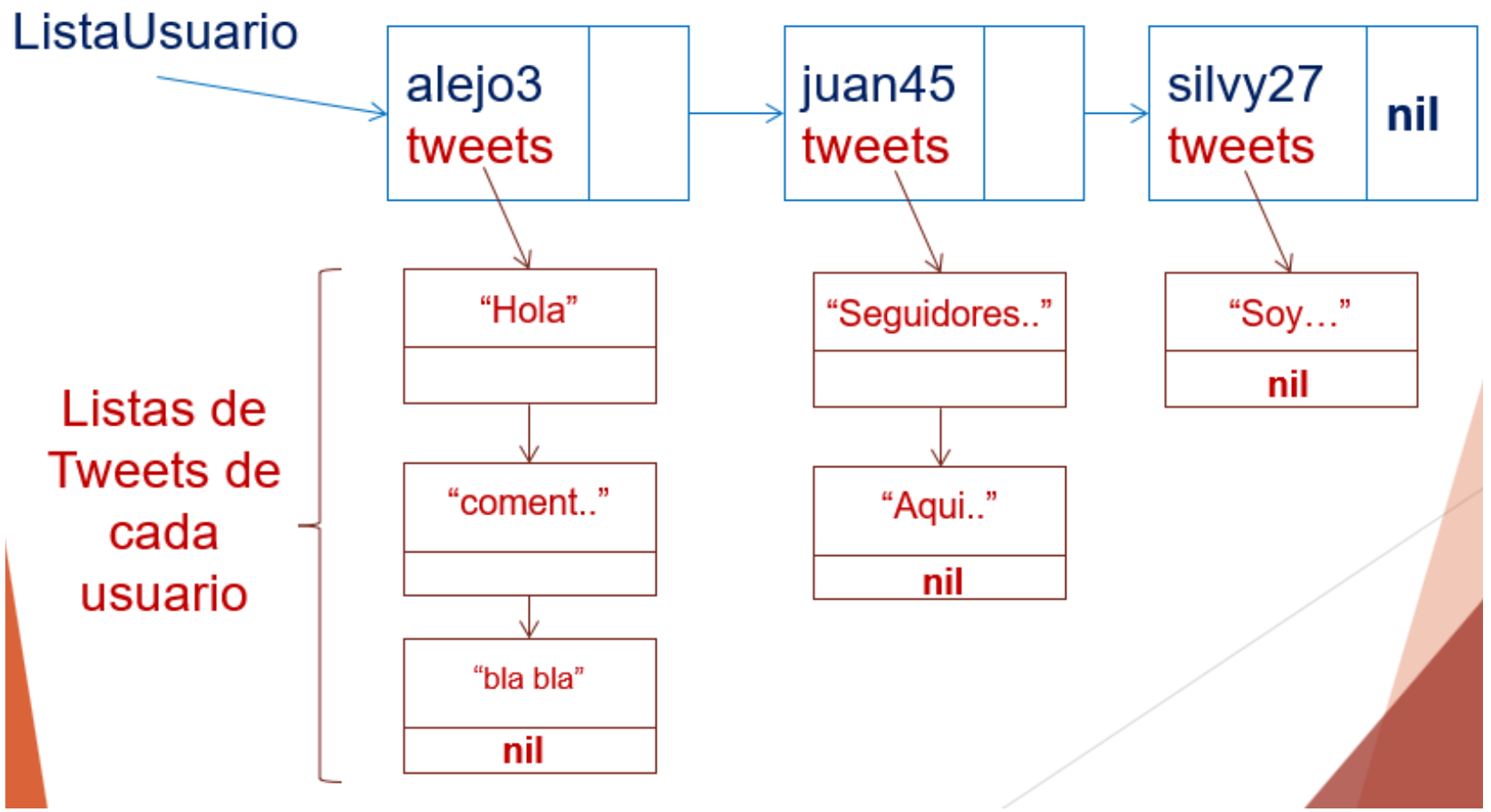
Módulo de Programación Imperativa

Clase de Repaso

Listas y Punteros



Recursión

Pasos para pensar la solución

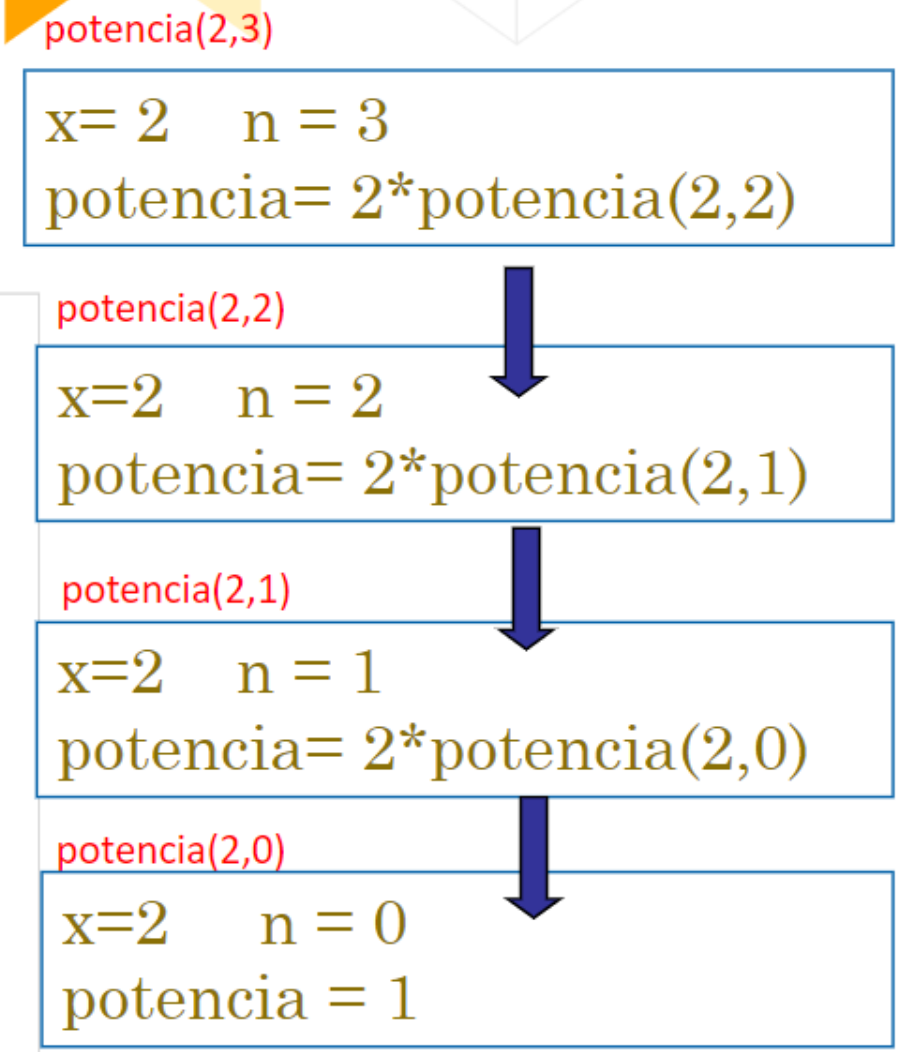
1) Búsqueda del caso o los casos base

2) ¿Cómo se reduce el problema?

3) ¿Qué tipo de módulo utilizar?
   a) Función
   b) Procedimiento

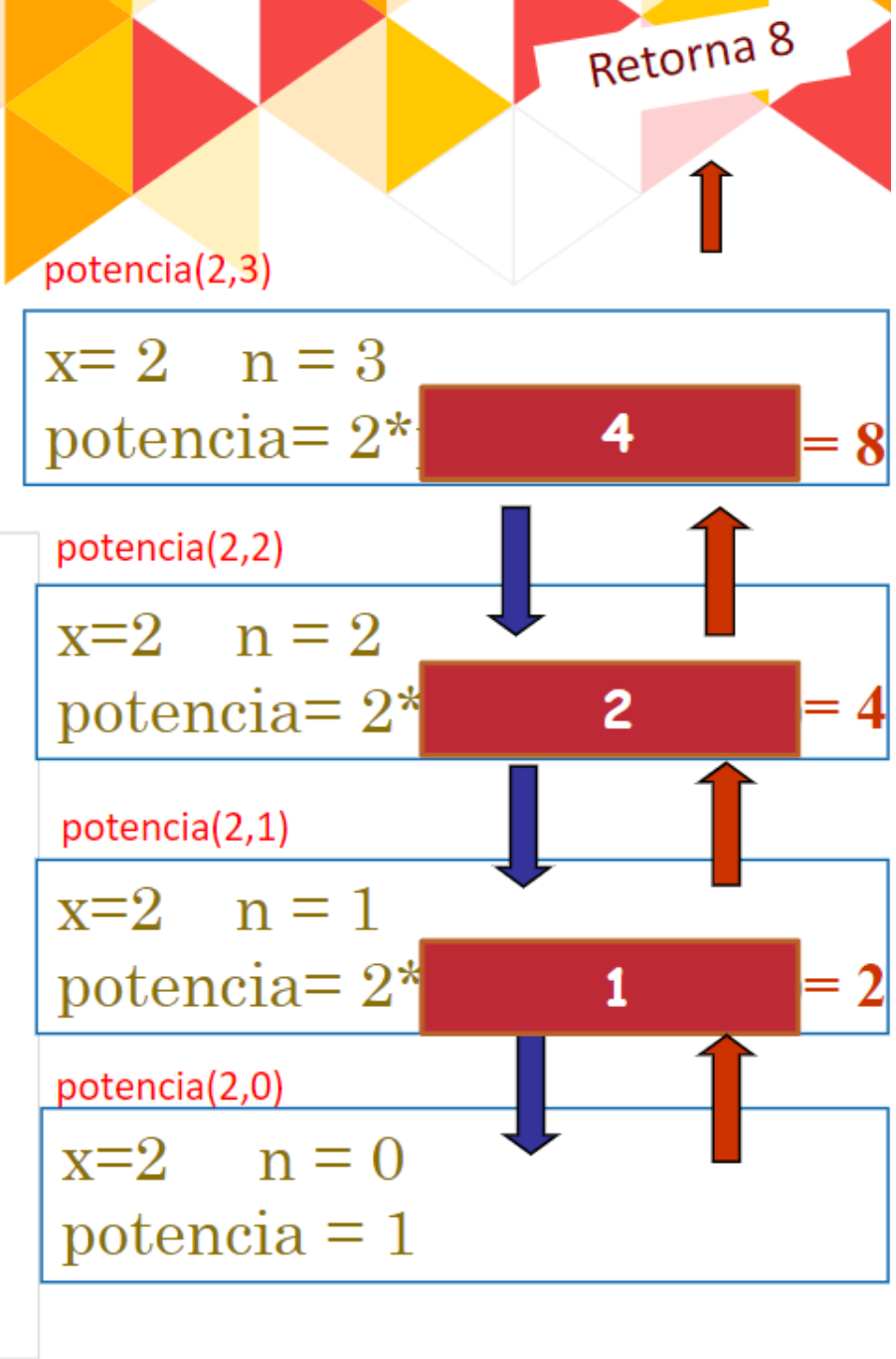
¿Cómo funciona?

```
Program ejemplo

Function potencia (x,n:integer): real;
begin
  if (n = 0) then
    potencia:= 1;
  else
    potencia := x * potencia(x,n-1)
end;

Begin
  read (x,n);
  write(potencia(x,n))
End.
```



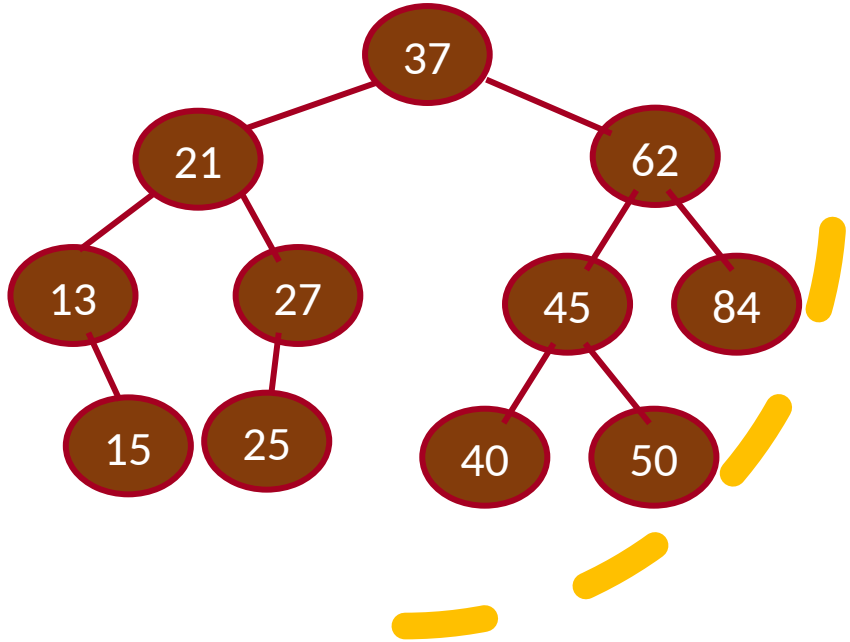
¿Cómo funciona?

```
Program ejemplo

Function potencia (x,n:integer): real;
begin
  if (n = 0) then
    potencia:= 1;
  else
    potencia := x * potencia(x,n-1)
end;

Begin
  read (x,n);
  write(potencia(x,n))
End.
```

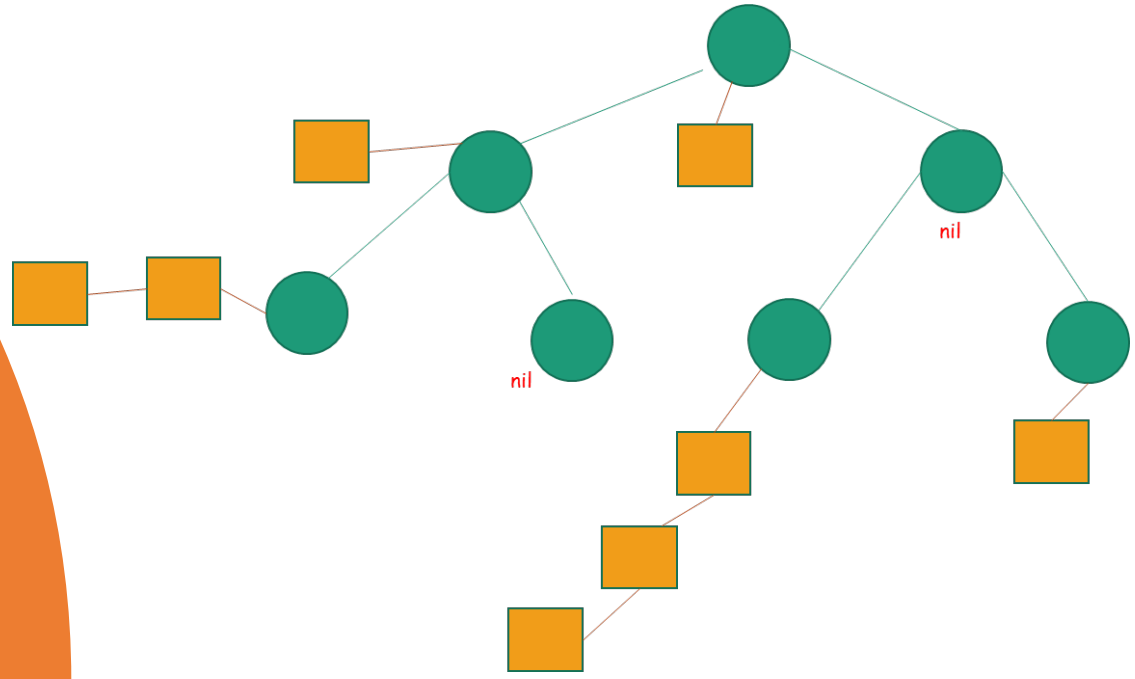


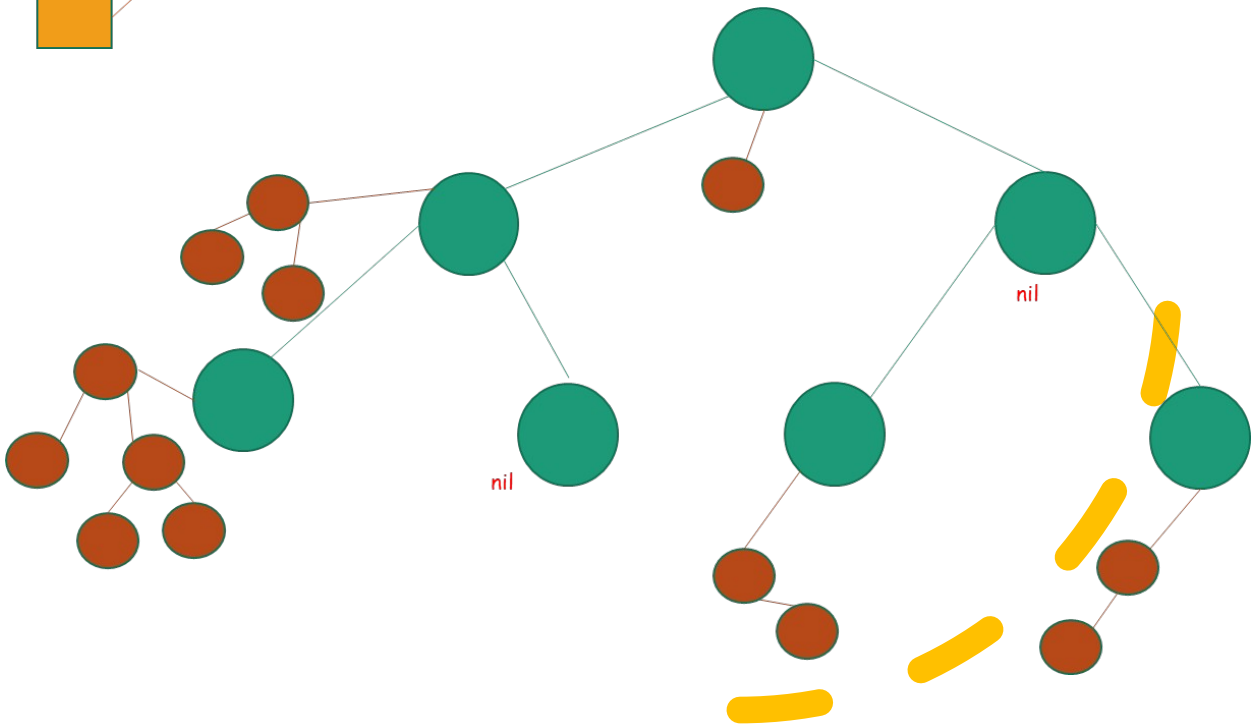
Árboles

1) Eficiencia en la búsqueda (altura)
2) Tener siempre presente el criterio de ordenación
3) Observar qué datos interesa almacenar
4) Tipos de recorridos
   a) Pre Orden
   b) En Orden
   c) Pos Orden
   d) Por niveles
   e) Acotados



Árboles y punteros





Resolución de Problemas

Interrogantes

1) Estructura de datos
   a) ¿Tiene algún orden?
   b) ¿Es eficiente mi decisión? Memoria/tiempo

2) ¿Qué hacer?
   a) ¿Cómo es el ingreso de datos?
   b) ¿Cómo se generan los datos?

3) ¿Hay recorridos?
   a) ¿Dependen del orden o no?
   b) ¿Se ven afectada la salida por la forma del recorrido?

4) Módulos
   a) ¿Qué funcionalidades principales se diferencian?

5) Programa
   a) Variables y secuencia

Ejercicio 1

X dispone de una lista simple con los tweets realizados durante los últimos 5 segundos. De cada tweet se conoce: el código y nombre de usuario que lo generó, el contenido del mensaje y si el mismo es o no un retweet. Esta información no tiene ningún orden y se debe tener en cuenta que podrían existir en la lista varios tweets del mismo usuario.

Se pide:

a) Realice un módulo que reciba la lista con los tweets y genere una nueva estructura donde para cada usuario se almacene la cantidad de mensajes publicados. Esta estructura debe estar ordenada por código de usuario y debe ser eficiente para la búsqueda por dicho criterio.

Una vez generado el árbol:

b) Informar la cantidad de tweets de los usuarios con código entre 100 y 700.
c) Informar el nombre del usuario con mayor cantidad de tweets.
d) ¿Qué cambiaría del ejercicio implementado si la lista inicial fuera una lista de listas? (Del usuario y sus tweets)

Ejercicio 2

Una biblioteca quiere tener un mejor acceso a sus libros. Para ello, nos piden generar un árbol binario de búsqueda con los datos de todos sus libros. De cada libro se conoce: título, ISBN y clasificador bibliográfico (código alfanumérico que permite clasificar el tema del ejemplar), que se leen desde teclado. La lectura finaliza con el ISBN 0 (cero). Interesa poder buscar los libros eficientemente por ISBN.

Se pide:

a) Generar árbol binario de búsqueda según el enunciado.

Una vez generado el árbol:

b) Realice un módulo que reciba el árbol y un ISBN de libro, y retorne verdadero si existe dicho libro en el árbol o falso en caso contrario.
c) Realizar un módulo que reciba el árbol y un código clasificador, y devuelva la cantidad de veces que aparece en el árbol (el módulo debe tener en cuenta que puede no existir).
d) Realice un módulo que reciba el árbol y un título de libro, y retorne verdadero si existe dicho libro en el árbol o falso en caso contrario.
e) Realizar un programa que invoque a los módulos realizados e informe desde el programa principal los datos correspondientes.