Recursión

Temas de la clase

1. Recursión. Definición y características
2. Ejemplos de Recursión
3. Método de búsqueda dicotómica en vectores. Una aplicación

Recursión.

Es una metodología para resolver problemas.

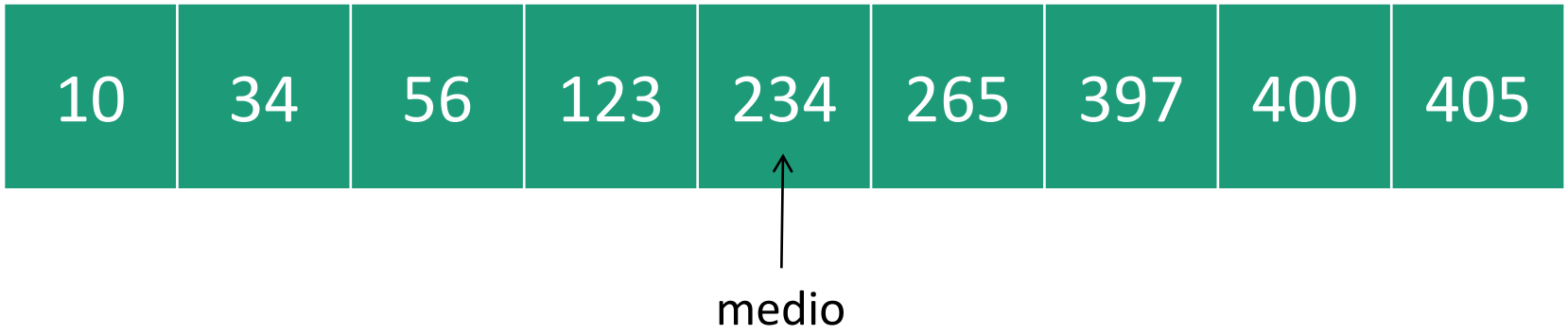
Permite resolver un problema P por resolución de instancias más pequeñas P_1, P_2, … P_n del mismo problema.

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

Veamos un ejemplo …

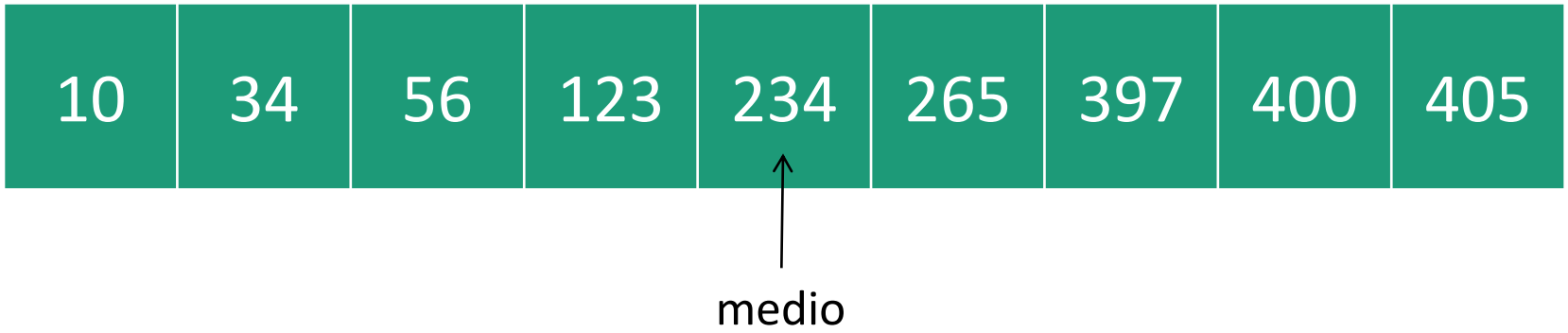
Motivación. Búsqueda dicotómica.

Busco el 56



Motivación. Búsqueda dicotómica.

Busco el 56

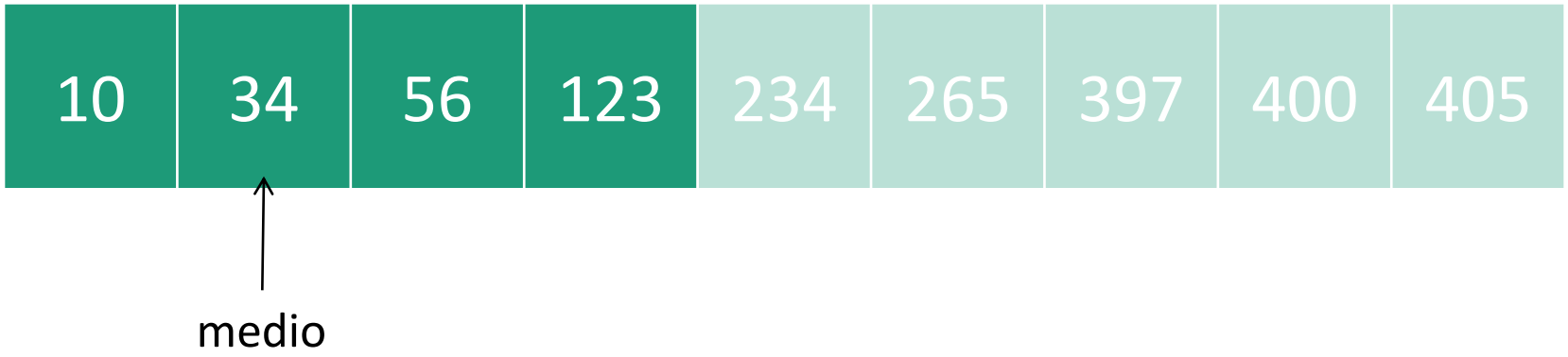


¿Cómo es 56 con respecto a v[medio]?

1. Si es = terminé
2. Si es < busco en la mitad inferior
3. Si es > busco en la mitad superior

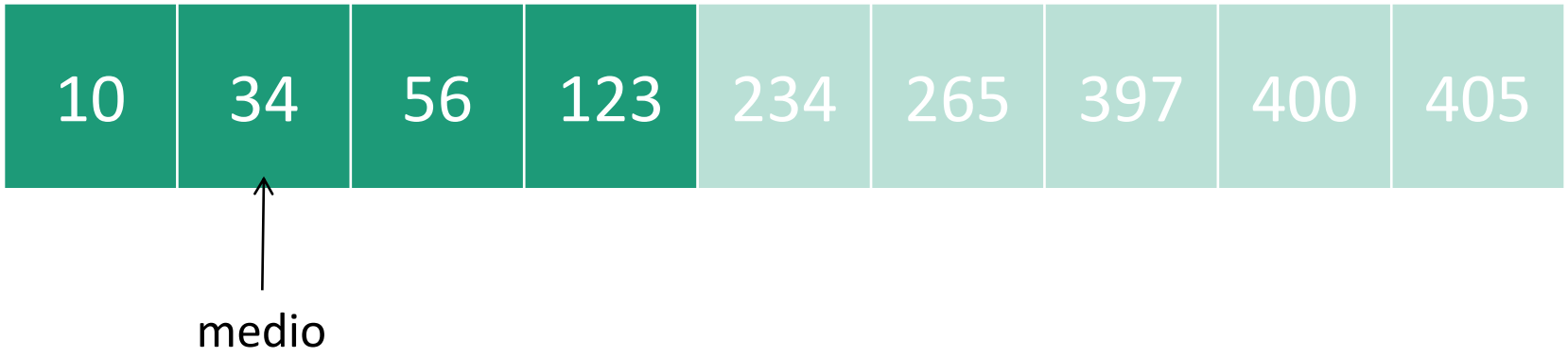
Motivación. Búsqueda dicotómica.

Busco el 56



Motivación. Búsqueda dicotómica.

Busco el 56

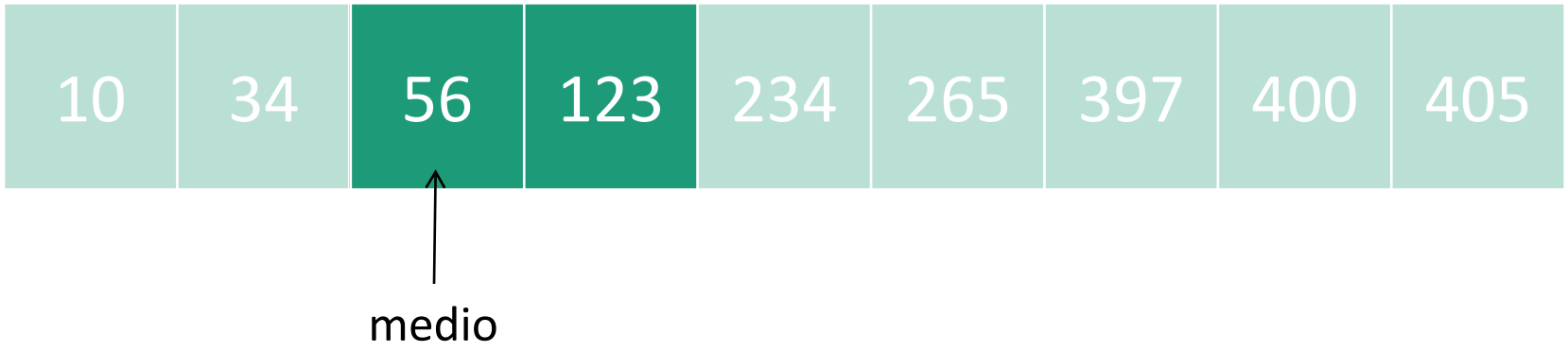


¿Cómo es 56 con respecto a v[medio]?

1. Si es = terminé
2. Si es < busco en la mitad inferior
3. Si es > busco en la mitad superior

Motivación. Búsqueda dicotómica.

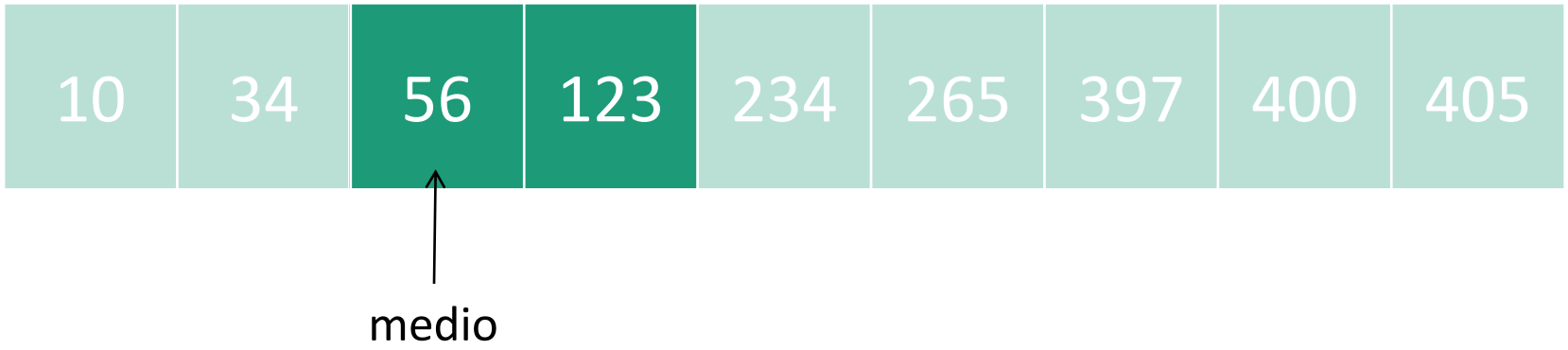
Busco el 56



Motivación. Búsqueda dicotómica.

Busco el 56

Encontré el 56!!



¿Cómo es 56 con respecto a v[medio]?

1. Si es = terminé
2. Si es < busco en la mitad inferior
3. Si es > busco en la mitad superior

Motivación. Búsqueda dicotómica.

Medio??

Primera mitad??

Segunda mitad??

Observemos que :

1. La primera vez se trabaja con el vector completo para determinar el punto medio

2. La siguiente vez, el vector se reduce a la mitad

3. La siguiente vez, el vector se reduce a la mitad de la mitad

Motivación. Búsqueda dicotómica.

```
Buscar (vector, datoABuscar)
 si el vector "no tiene elementos"  entonces
     No lo encontré y termino la búsqueda
 sino
     Determinar el punto medio del vector
     Comparar datoABuscar con el contenido del punto medio
     si coincide entonces "Lo encontré"
     sino
        si datoABuscar < contenido del punto medio entonces
            Buscar (1era mitad del vector, datoABuscar)
        sino
            Buscar (2da mitad del vector, datoABuscar)
```

Motivación. Búsqueda dicotómica.

```
Buscar (vector, datoABuscar)
 si el vector "no tiene elementos"  entonces
    No lo encontré y termino la búsqueda
 sino
    Determinar el punto medio del vector
    Comparar datoABuscar con el contenido del punto medio
    si coincide entonces "Lo encontré"
    sino
       si datoABuscar < contenido del punto medio entonces
          Buscar (1era mitad del vector, datoABuscar)
       sino
          Buscar (2da mitad del vector, datoABuscar)
```

Observaciones importantes de esta solución :

1) El módulo realiza invocaciones a si mismo.
2) En cada llamada, el tamaño del vector se reduce a la mitad.
3) Existen 2 casos distintos que se resuelven de manera particular y directa (casos base):

a) Cuando el vector "no contiene elementos"
b) Cuando encuentro el datoABuscar

Recursión. Definición y características

Una solución recursiva resuelve un problema por resolución de instancias más pequeñas del mismo problema.

Un algoritmo recursivo involucra:

- Alguna condición de terminación (implícita/explícita)
- Una autoinvocación (llamada recursiva). Se debe garantizar que en un nro finito de autoinvocaciones se alcanza la condición de terminación.

```
Buscar (vector, datoABuscar)
 si el vector "no tiene elementos"  entonces
    No lo encontré y termino la búsqueda
 sino
    Determinar el punto medio del vector
    Comparar datoABuscar con el contenido del punto medio
    si coincide entonces "Lo encontré"
    sino
       si datoABuscar < contenido del punto medio entonces
          Buscar (1era mitad del vector, datoABuscar)
       sino
          Buscar (2da mitad del vector, datoABuscar)
```

Ejemplo de Recursión

Potencia de un número

Planteo de solución recursiva. Tener en cuenta:

1. ¿Cómo defino el problema en términos de problemas más simples del mismo tipo?
2. ¿Cómo achico el problema en cada llamado recursivo?
3. ¿Qué instancia/s del problema son caso/s base?

$$X^n \begin{cases} 1 & \text{si } n = 0 \\ X * X^{n-1} & \text{si } n >= 1 \end{cases}$$

Caso base

Recursión

Ejemplo:

$2^3 = 2 * 2^2$

$= 2 * 2 * 2^1$ (2^2)

$= 2 * 2 * 2 * 2^0$ (2^1)

$= 2 * 2 * 2 * 1 = 8$ (2^0)

Ejemplo de Recursión

Potencia de un número

$$X^n \begin{cases} 1 & si\ n=0 \\ X * X^{n-1} & si\ n>=1 \end{cases}$$

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

Actividades en

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

- **Descargar el programa** `CalculoDePotencia`

- **ACTIVIDAD 1**

a) Repasar la función `potencia` vista.

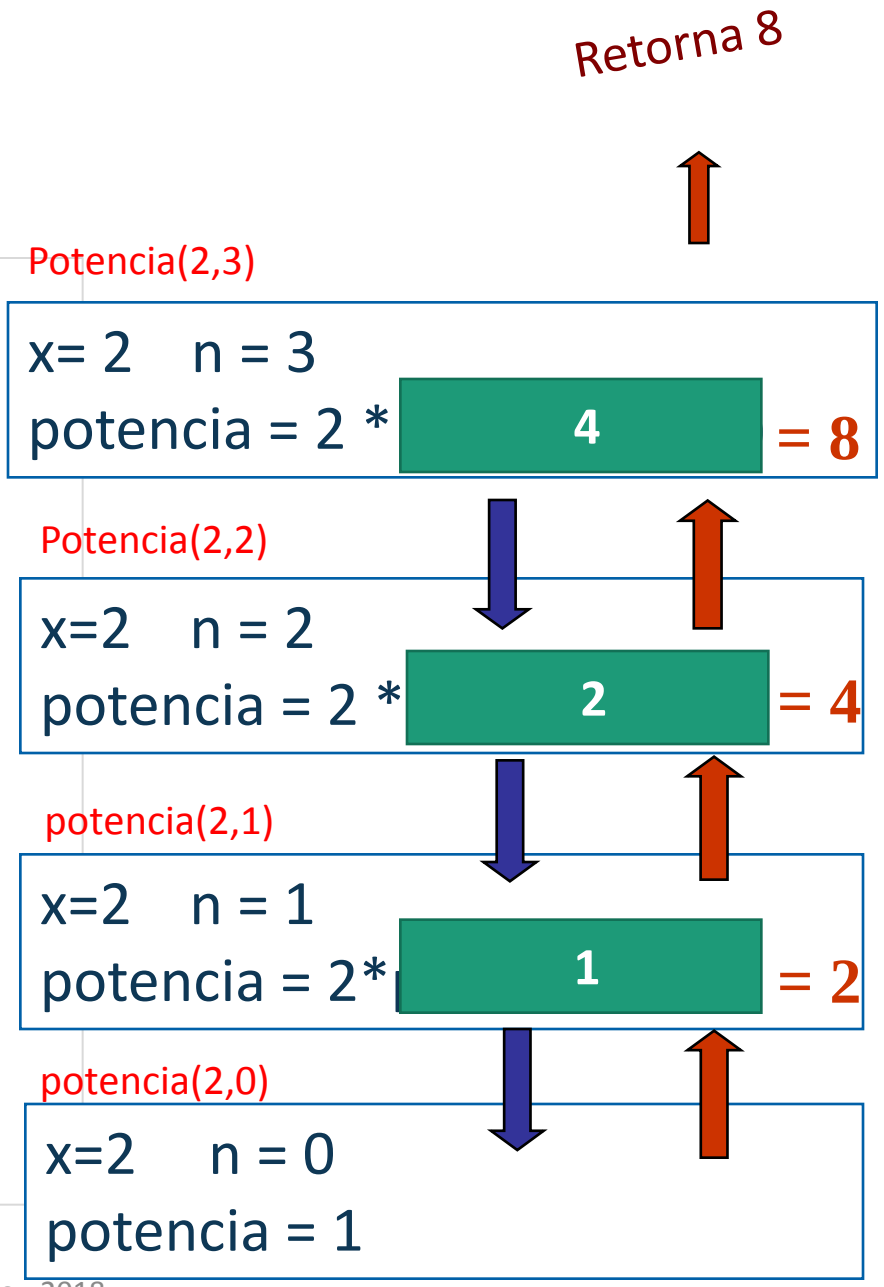
b) Completar el programa `CalculoDePotencia` para que lea dos valores X y n, invoque a la función `potencia` para calcular X^n y muestre el resultado.

¿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;

var
  x,n:integer;
begin
  read (x,n);
  write(potencia(x,n));
end.
```



Actividades en Máquina

- **ACTIVIDAD 2**

a) Implementar en el programa **CalculoDePotencia**, la función **potencia1**

```
Function potencia1 (x,n: integer): real;
begin
  potencia1 := x * potencia1(x,n-1);
end;
```

b)Invocar a la función **potencia1** para calcular 5^3 .

c) Compilar y ejecutar. ¿Qué ocurre? ¿Por qué?

Actividades en Máquina

- **ACTIVIDAD 3**

a) Implementar en el programa **CalculoDePotencia**, la función **potencia2**

```
Function potencia2 (x,n: integer): real;
begin
  if (n = 0) then
      potencia2:= 1
  else
      potencia2 := x * potencia2(x,n);
end;
```

b) Invocar a la función **potencia2** para calcular 5^3 .

c) Compilar y ejecutar. ¿Qué ocurre? ¿Por qué?

Recordemos el cálculo del dígito máximo de un número entero

```
program CalculoDigitoMaximo;
type digito=-1..9;

var num: integer;

function digitomaximo (n:integer):digito;
var
  max, dig: digito;

begin
  max:= -1;
  while (n<>0) do begin
    dig:= n mod 10;
    if dig > max then max:= dig;
    n:= n div 10;
  end;
  digitomaximo:= max;
  end;
begin
  read (num);
  write ('El digito maximo de ', num, ' es: ', digitomaximo(num));
end.
```

Pensemos una solución recursiva….

Cálculo del dígito máximo de un número entero (Sol. Recursiva)

```
program CalculoDigitoMaximo;
type digito=-1..9;

var num: integer;

function digitomaximo (n:integer):digito;
var
  max, dig: digito;

begin
  max:= -1;
  while (n<>0) do begin
    dig:= n mod 10;
    if dig > max then max:= dig;
    n:= n div 10;
  end;
  digitomaximo:= max;
  end;
begin
  read (num);
  write ('El digito maximo de ', num, ' es: ', digitomaximo(num));
end.
```

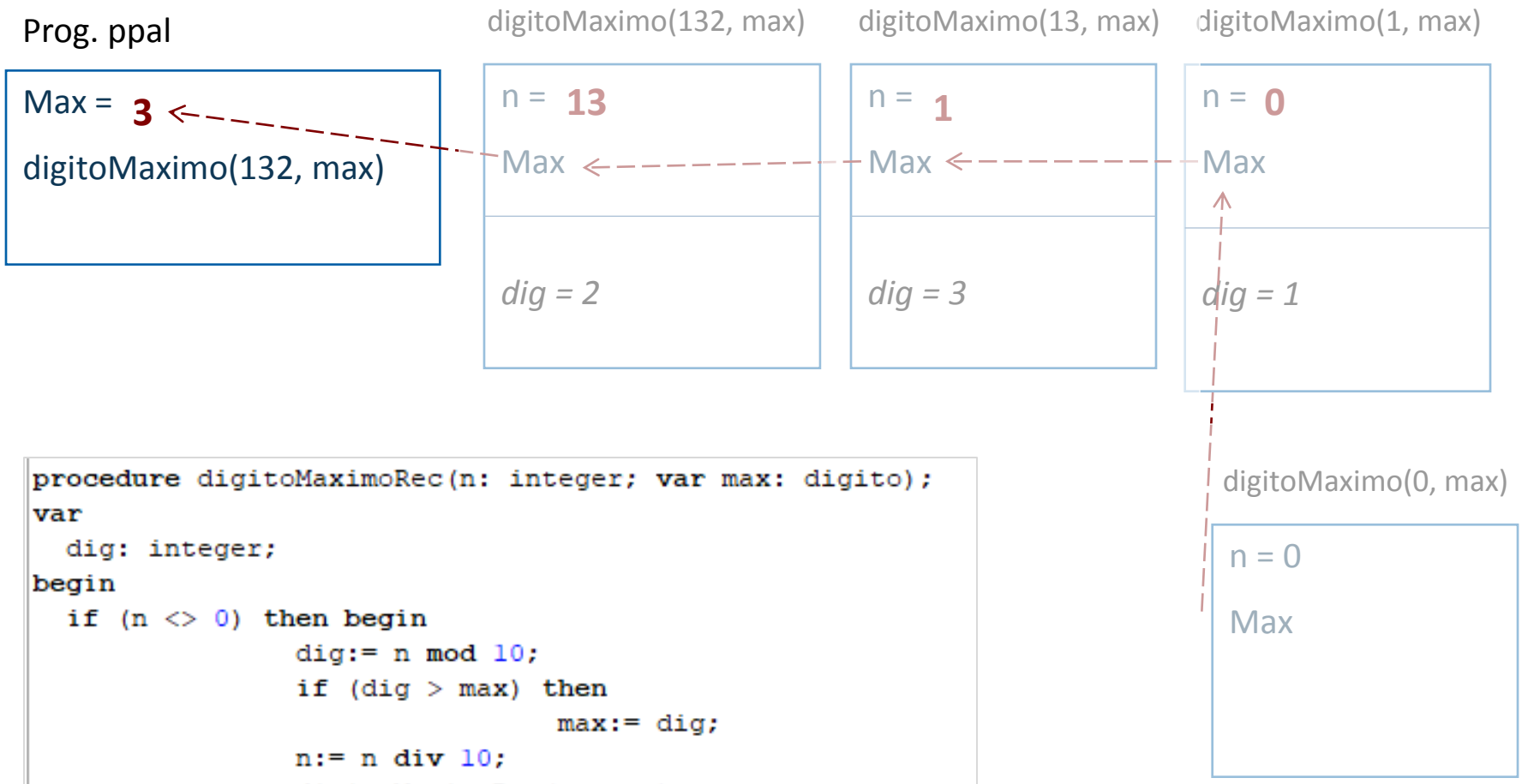
```
program CalculoDigitoMaximoRec;
type digito=-1..9;

var num: integer;
    max: digito;

procedure digitoMaximoRec(n: integer; var max: digito);
var
  dig: integer;
begin
  if (n <> 0) then begin
                dig:= n mod 10;
                if (dig > max) then
                                max:= dig;
                n:= n div 10;
                digitoMaximoRec(n, max);
              end;
end;

begin
  read (num);
  max := -1;
  digitoMaximoRec(num, max);
  write ('El digito maximo de ', num, ' es: ', max);
end.
```

¿Cómo funciona?




```
procedure digitoMaximoRec(n: integer; var max: digito);
var
  dig: integer;
begin
  if (n <> 0) then begin
                dig:= n mod 10;
                if (dig > max) then
                                max:= dig;
                n:= n div 10;
                digitoMaximoRec(n, max);
              end;
end;
```

Actividades en Máquina

ACTIVIDAD 4

Descargar el programa `Recursion`

a) Repase el procedimiento `digitoMaximoRec.`

¿Cuál es el caso base?

¿Cómo se acerca al caso base?

b) Compile, ejecute y compruebe el resultado.

Actividades en Máquina

Utilizando el programa **`Recursion`** realice las siguientes actividades:

ACTIVIDAD 5

a)Modificar el procedimiento digitoMaximoRec. Debe colocarse la instrucción `writeln ('max: ', max);` después de la invocación al procedimiento.

b)Responder:

- ¿Qué valor se muestra antes de finalizar cada instancia recursiva?
- ¿Qué valor se muestra en el programa principal?

Actividades en Máquina

Utilizando el programa **`Recursion`** realice las siguientes actividades:

ACTIVIDAD 6

a)Modificar el procedimiento digitoMaximoRec. Debe colocarse la instrucción `writeln ('max: ', max);` antes de la invocación al procedimiento.

b) Responder:

- ¿Qué valor se muestra antes de cada llamada recursiva? ¿Por qué?
- ¿Qué valores se imprimen si el parámetro max es pasado por valor? ¿Qué imprime en el programa?¿Funciona?

Actividades en Máquina

ACTIVIDAD 7

```
5236
ImprimirDigitos1
6
3
2
5
```

```
5236
ImprimirDigitos2
5
2
3
6
```

En el programa **`Recursion`**

a. Implementar el procedimiento recursivo **`ImprimirDigitos1`** que imprime los dígitos de un número dado, empezando por la unidad.

b. Implementar el procedimiento recursivo **`ImprimirDigitos2`** que imprime los dígitos de un número dado, finalizando con la unidad.

Nota: el planteo de la solución recursiva es similar a la del procedure digitoMaximoRec visto

Actividades en Máquina

ACTIVIDAD 8

Implementar un programa **`ListaConRecursion`** que:

a) Genere una lista de números enteros y muestre los valores guardados (utilizar los módulos del **`ProgramaLista`** ya visto)

b) Invoque a un módulo recursivo **`ImprimirEnOrden`** que imprima los valores contenidos en la lista en el orden en que se guardaron.

c) Invoque a un módulo recursivo **`ImprimirOrdenInverso`** que imprima los valores contenidos en la lista desde el último dato guardado al primero.

Actividades en Máquina

ACTIVIDAD 9

Implementar el programa **`BusquedaDicotomica`** que:

a) Utilizando los módulos del programa vectorOrdenado de la clase anterior, genere un vector ordenado.

b) Implemente el método de búsqueda dicotómica recursivo con el siguiente encabezado:

```
Procedure busquedaDicotomica( v: vector; ini,fin: indice; dato:integer; var pos: indice);
```

c) Utilizando el método implementado, lea un valor de teclado e imprima el resultado de la búsqueda.

Enviar a través de la Mensajería de Ideas, **`BusquedaDicotomica.pas`** al docente asignado al grupo.

Actividades en Máquina

ACTIVIDAD 10

Utilizando **`ProgramaVectores`** (medioteca Ideas) realice las siguientes actividades:

a) Implementar un módulo recursivo **`Máximo`** que devuelva el máximo valor del vector.

b) Implementar un módulo recursivo **`Suma`** que devuelva la suma de los valores contenidos en el vector

c) Utilizar los módulos implementados para mostrar el máximo y la suma.

Actividades en Máquina

ACTIVIDAD 11

En el programa **`ListaConRecursion:`**

a) Implementar un módulo recursivo **`Máximo`** que devuelva el máximo valor de la lista.

b) Implementar un módulo recursivo **`Minimo`** que devuelva el mínimo valor de la lista.

c) Implementar un módulo recursivo **`Buscar`** que devuelva verdadero si un valor determinado se encuentra en la lista o falso en caso contrario.

Actividades en Máquina

ACTIVIDAD 12

Implementar un programa que informe si una lista de caracteres representa una palabra palíndrome

(Palíndrome: se lee igual de izquierda a derecha que de derecha a izquierda)

Sugerencia: revisar el módulo recursivo **`ImprimirOrdenInverso`**