

ALGORITMOS Y PROGRAMACIÓN II

Clase 2

Matrices (Continuación)

Temas de la clase

Arreglos bidimensionales (Matrices) (Continuación)



Actividades en Máquina

ACTIVIDAD 1: Crear un nuevo archivo **Catedra.pas**

Una cátedra necesita almacenar información de sus estudiantes (de los cuales se conoce nombre, apellido y nota (un entero)). La cátedra tiene 4 comisiones y en cada comisión hay 5 alumnos.

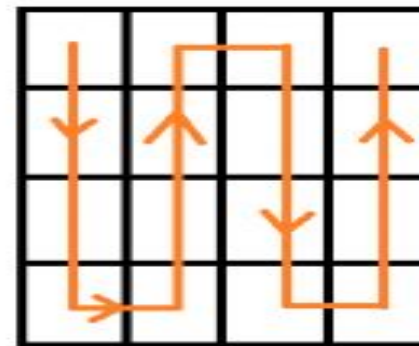
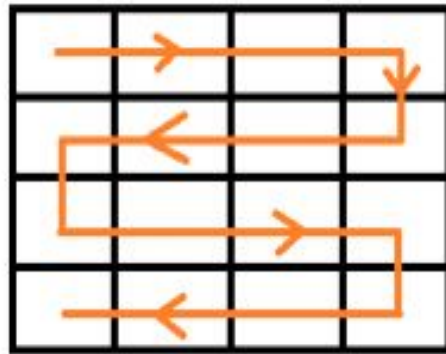
- Implemente un módulo que almacene la información de los 20 estudiantes.
- Implemente un módulo que reciba la información de los estudiantes y devuelva el estudiante con la nota máxima.
- Implemente un módulo que reciba la información de los estudiantes y una nota numérica y devuelva true si al menos un estudiante tiene dicha nota, y false en caso contrario.
- Invocar desde el programa principal a los módulos implementados y comprobar el correcto funcionamiento del mismo.



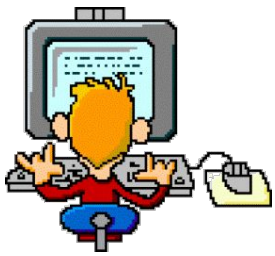
Actividades en Máquina

ACTIVIDAD 2: Crear un nuevo archivo **Catedra2.pas**

- a) Haga dos módulos que imprima los elementos de una matriz de $M \times N$ con los siguientes recorridos:



- b) Implemente dos módulos (cada uno usando un recorrido distinto) que reciba la información de los estudiantes y una nota numérica y devuelva el nombre y apellido del estudiante que tiene dicha nota, y 'Ninguno' en caso de no existir.



Actividades en Máquina

Generación aleatoria de inmuebles: descargar el archivo **Inmuelle.pas**

```
procedure cargarInmuelle(var inmu1:inmueble);
var
  vTipos:array[1..4] of string= ('Casa', 'Departamento', 'Duplex', 'Local Comercial');
  vLocal:array[1..10] of string= ('La Plata', 'Berisso', 'Ensenada',
                                'Quilmes', 'Avellaneda', 'Bernal', 'Berazategui', 'Azul', 'Tandil', 'Dolores');
begin

  inmu1.tipo:= vTipos[random(4)+1];
  inmu1.cantHab:=random(6);
  inmu1.cantBanos:=random(4)+1;
  inmu1.precio:=random(50000)/2+10000;
  inmu1.localidad:= vLocal[random(10)+1];

end;
```



Actividades en Máquina

ACTIVIDAD 3: Crear un nuevo archivo **ContandoInmuebles.pas**

Una inmobiliaria necesita un programa que procese la información de los inmuebles en alquiler del partido de la costa. Los inmuebles tienen de cero a cuatro habitaciones y de uno a tres baños. El ingreso de la información finaliza cuando se lee cantidad de baños cero (que no debe procesarse).

- a) Implemente un módulo que genere una nueva lista con inmuebles usando el procedimiento **cargarInmueble**.
- b) Implemente un módulo que reciba la lista de inmuebles y contabilice para cada par posible de cantHab-cantBaños cuantos inmuebles se poseen almacenados.
- c) Implemente un módulo que reciba la contabilidad y la imprima.
- d) Invocar desde el programa principal a los módulos implementados para probar el correcto funcionamiento del programa.

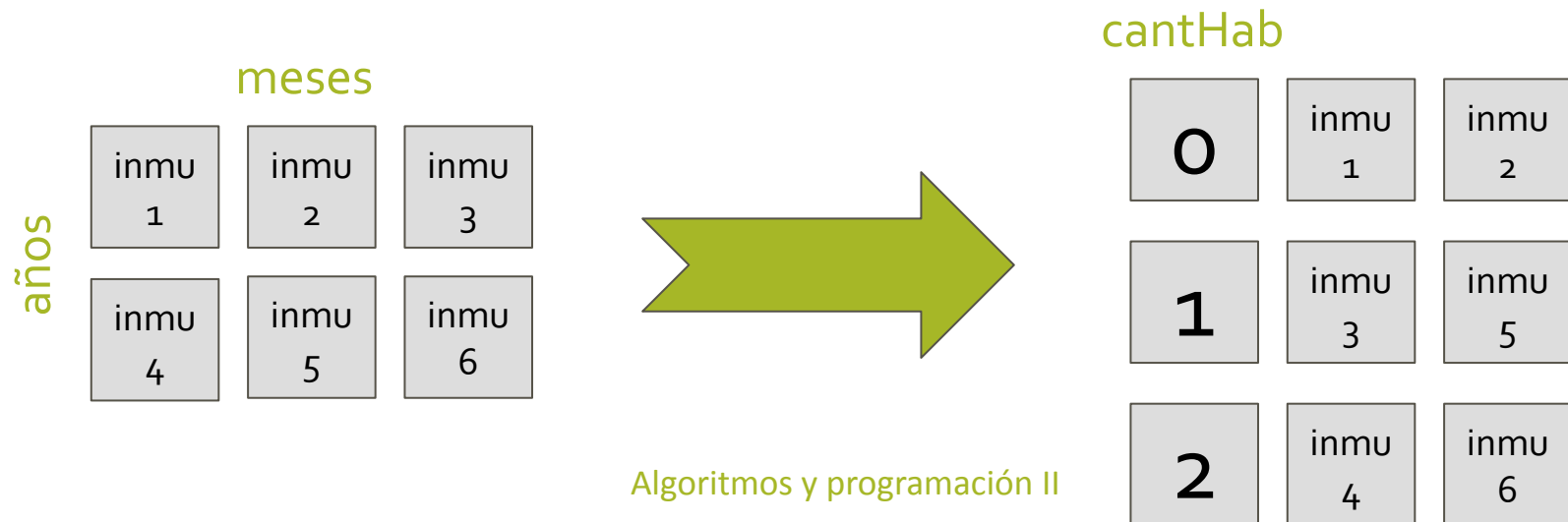


Actividades en Máquina

ACTIVIDAD 4: Crear un nuevo archivo **ReagrupandoInmuebles.pas**

Una inmobiliaria almacena información histórica sobre el inmueble vendido al precio más alto de cada mes de los años 2010 a 2023.

A la inmobiliaria le interesa procesar dicha información y agruparla por cantidad de habitaciones. Es decir, cuales son todos los inmuebles con cero habitaciones, cuales tiene una habitación, cuales dos, etc.





Actividades en Máquina

const

DIMF = 5;

type

inmueble = **record** . . . **end**;

listaInmuebles = **specialize** LinkedList<inmueble>;

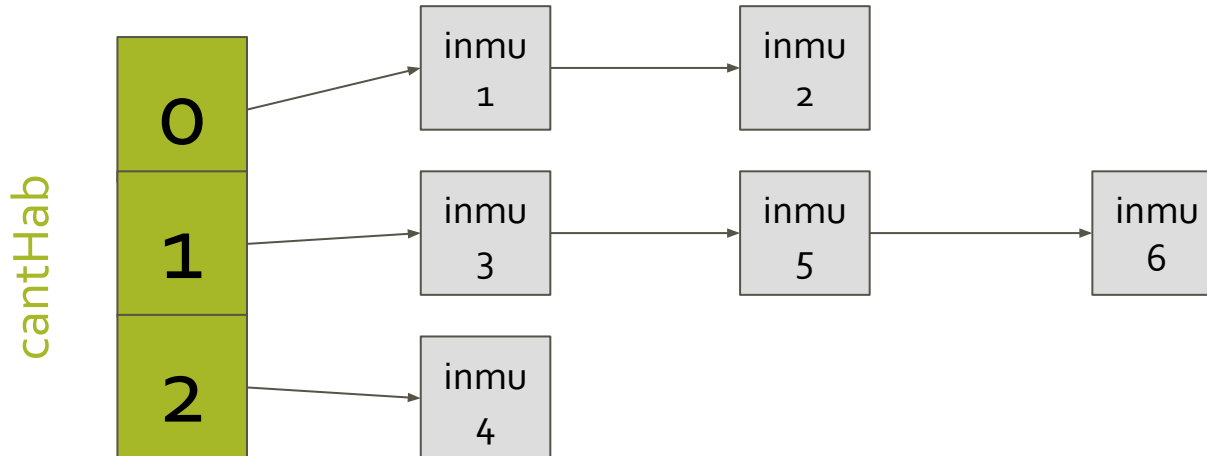
vector = **array** [1..DIMF] **of** listaInmuebles;

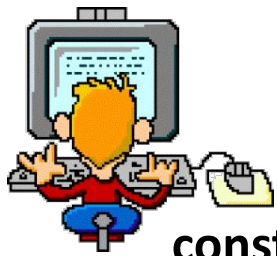
var v: Vector; i: inmueble; l: listaInmuebles;

begin

. . .

end.





Actividades en Máquina

const

DIMF = 5;

type

inmueble = **record** . . . **end**;

listaInmuebles = **specialize** LinkedList<inmueble>;

vector = **array** [1..DIMF] **of** listaInmuebles;

var v: Vector; i: inmueble; l: listaInmuebles;

begin

v[1]:= listaInmuebles.create(); { Crea una lista y la guarda en el vector }

leerInmueble(i);

v[1].add(i); { Agrega un elemento a la lista en la posición 1 }

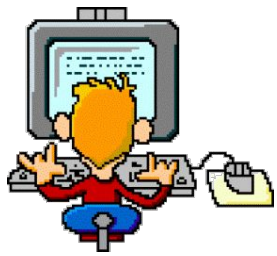
l:= listaInmuebles.create();

v[2]:= l; { Asigna una lista al vector }

v[3]:= v[2]; { Almacena la lista de la posición 2 a la posición 3 }

{ Ambas posiciones del vector **guardan la misma lista.** }

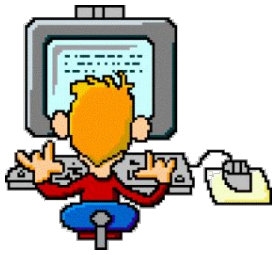
end.



Actividades en Máquina

ACTIVIDAD 4 (Continuación)

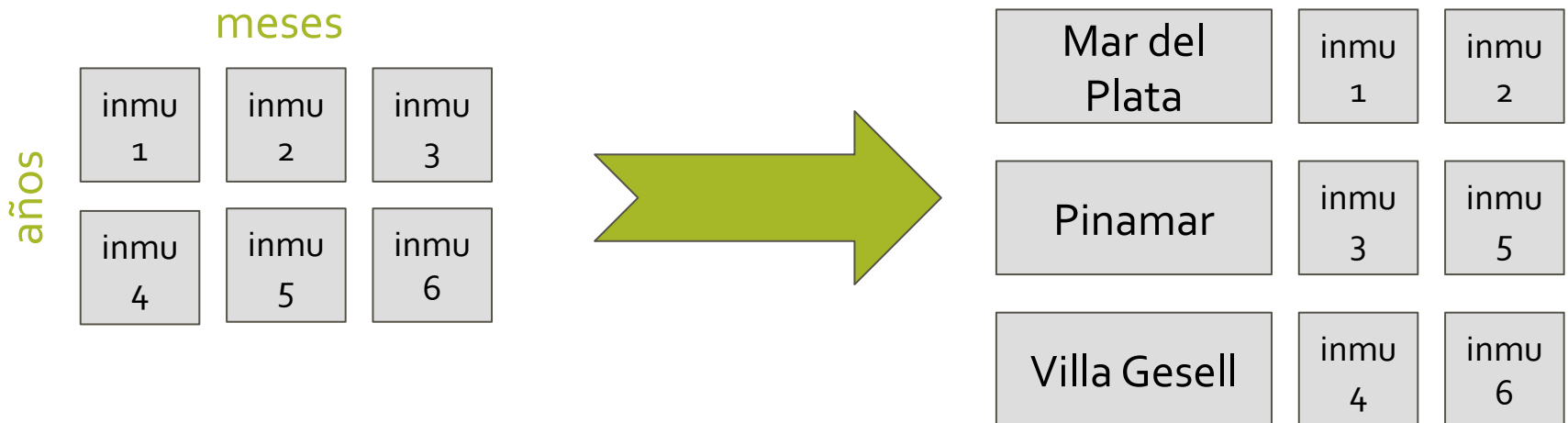
- a) Implemente un módulo que genere inmuebles con el procedimiento **cargarInmueble**. Se debe generar un inmueble para cada mes de los años 2010 a 2023. Los inmuebles generados representan el mejor vendido para cada mes-año. Los inmuebles tienen de cero a cuatro habitaciones.
- b) Implemente un módulo que reciba la estructura generada en a) y almacene a todos los inmuebles en otra estructura **agrupada** por cantidad de habitaciones.
- c) Implemente un módulo que imprima toda la información de los inmuebles agrupados por cantidad de habitaciones.
- d) Escriba un programa que invoque los módulos implementados y pruebe el correcto funcionamiento del mismo.

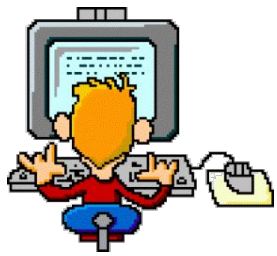


Actividades en Máquina

ACTIVIDAD 5: Crear un nuevo archivo **ReagrupandoInmuebles2.pas**

A partir de la actividad anterior, la inmobiliaria desea agrupar los inmuebles por localidad.





Actividades en Máquina

type

inmueble = **record** . . . **end**;

listaInmuebles = **specialize** LinkedList<inmueble>;

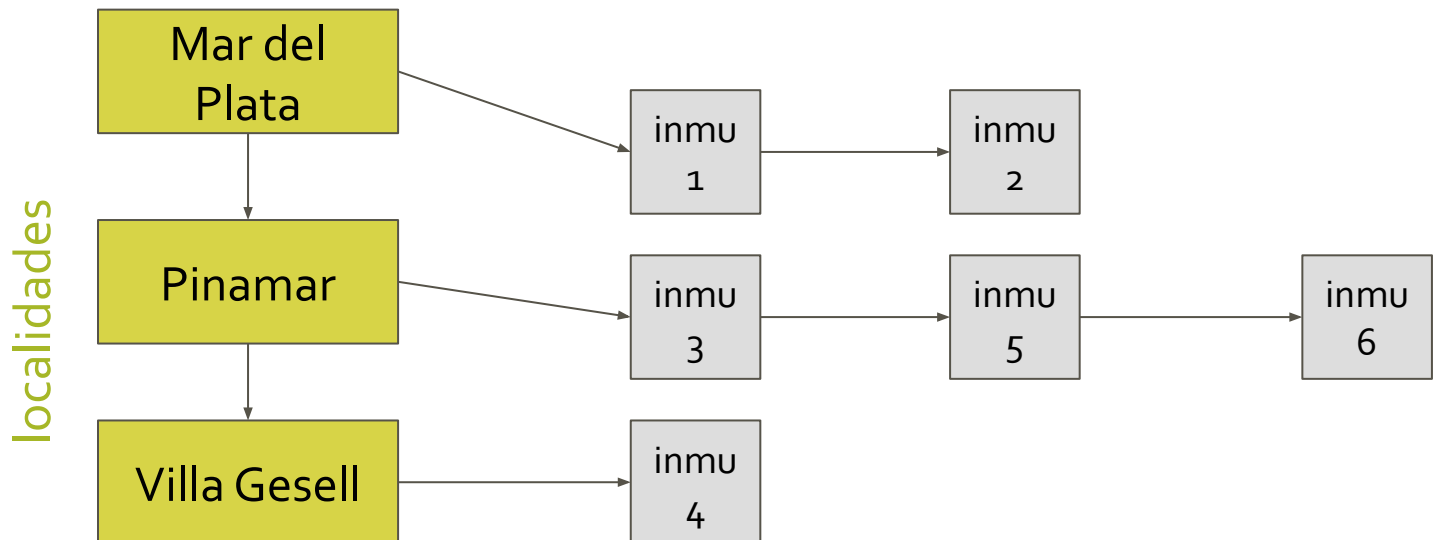
listaListas = **specialize** LinkedList<listaInmuebles>;

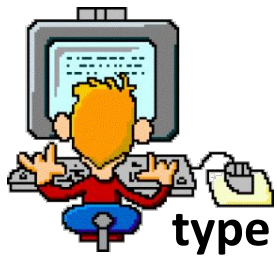
var ll: listaListas; i: inmueble; l: listaInmuebles;

begin

. . .

end.





Actividades en Máquina

```
inmueble = record . . . end;  
listaInmuebles = specialize LinkedList<inmueble>;  
listaListas = specialize LinkedList<listaInmuebles>;  
var ll: listaListas; i: inmueble; l: listaInmuebles;  
begin  
  ll:= listaListas.create(); { Crea una lista de listas }  
  l:= listaInmuebles.create(); { Crea una lista de inmuebles }  
  cargarInmuebles(l);  
  ll.add(l); { Agrega la lista de inmuebles a la lista de listas }  
  . . .  
  ll.reset();  
  while not ll.eol() do begin  
    l:= ll.current(); { current devuelve una lista de inmuebles }  
    recorrer(l);  
    ll.next(); { avanza al siguiente elemento de la lista de listas }  
  end;  
end.
```



Actividades en Máquina

ACTIVIDAD 5 (Continuación)

- a) Implemente un módulo que genere inmuebles con el procedimiento **cargarInmueble**. Se debe generar un inmueble para cada mes de los años 2010 a 2023. Los inmuebles generados representan el mejor vendido para cada mes-año.
- b) Implemente un módulo que reciba la estructura generada en a) y almacene todos los inmuebles en otra estructura **agrupada** por localidad de manera ordenada.
- c) Implemente un módulo que imprima toda la información de los inmuebles agrupados por localidad.
- d) Escriba un programa que invoque los módulos implementados y pruebe el correcto funcionamiento del mismo.



Arreglos tridimensionales

Un arreglo tridimensional, también conocido como tensor, es una estructura de datos organizada por filas, columnas y capas. Cada elemento es referenciado por tres índices.

- 9	4	2	5	7
3	0	1 2	8	6 1
1	2 3	- 6	4 5	2
2 2	3	- 1	7 2	6

	A	B	C	D
1	Filtrar			
2	Producto	- todo -		
3				
4	Datos	Vendedor	Región	
5	Suma - Ventas	Fernández	Norte	128.734 €
6			Sur	117.482 €
7		González	Norte	131.971 €
8			Sur	97.381 €
9		López	Norte	93.611 €
10			Sur	98.471 €
11	Cantidad - Ventas	Fernández	Norte	12
12			Sur	14
13		González	Norte	15
14			Sur	10
15		López	Norte	12
16			Sur	16
17	Promedio - Ventas	Fernández	Norte	28.733 €
18			Sur	
19		González	Norte	
20			Sur	
21		López	Norte	
22			Sur	
23	Total Suma - Ventas			66
24	Total Cantidad - Ventas			
25	Total Promedio - Ventas			
26				





Arreglos N-dimensionales

Un arreglo N-dimensional es una estructura de datos organizada en N dimensiones. Cada elemento es referenciado por N índices.

```
Program Arreglo_4_dimensiones;  
Const  
    Meses = 12;    Anios = 100;    Estaciones = 50;    Observaciones = 200;  
Type  
    multiD = array[1..Anios, 1..Meses, 1..Estaciones, 1..Observaciones]  
                of real;  
Var  
    temperaturas : multiD;  
Begin  
    temperaturas[34, 5, 21, 3] := 23.5;  
    ...  
End.
```

¿Cuántos for anidados se necesitan para la carga completa de este arreglo?

En general, para un arreglo de N dimensiones ¿cuántos for anidados se necesitan?



Arreglos N-dimensionales

Un arreglo N-dimensional es una estructura de datos organizada en N dimensiones. Cada elemento es referenciado por N índices.

```
Program Arreglo_4_dimensiones;  
Const  
    Meses = 12;    Anios = 100;    Estaciones = 50;    Observaciones = 200;  
Type  
    multiD = array[1..Anios, 1..Meses, 1..Estaciones, 1..Observaciones]  
                of real;  
Var  
    temperaturas : multiD;  
Begin  
    temperaturas[34, 5, 21, 3]:= 23.5;  
    ...  
End.
```

**¿Cuánta memoria estática
ocupa este arreglo?**