

PROGRAMACIÓN I

TEORÍA – CECILIA SANZ

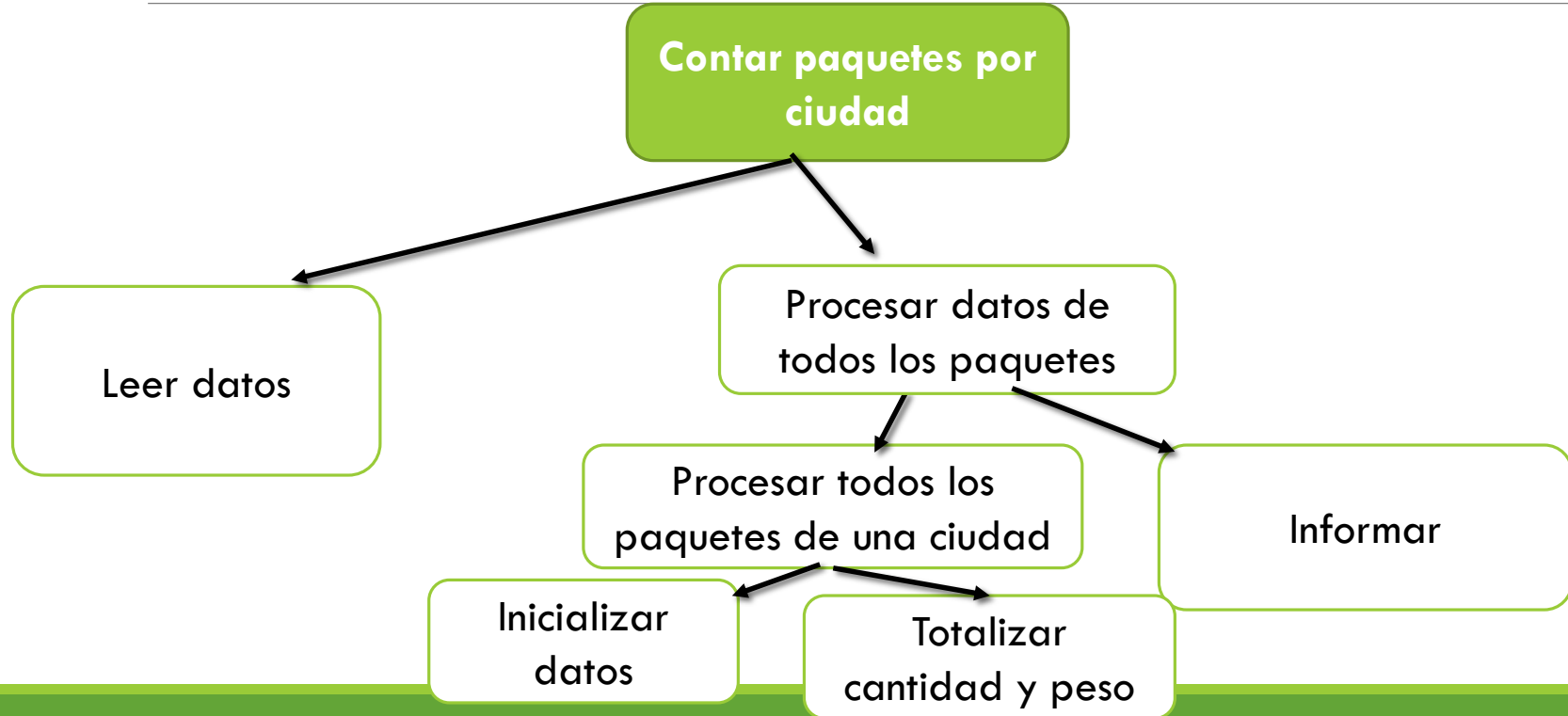
Temas

- ✓ Estructura de datos arreglo
- ✓ Definición
- ✓ Operaciones
- ✓ Ejemplos

REPASO

Supongamos que en un despacho se están recibiendo paquetes de Mercado Libre para diferentes ciudades. Cada paquete indica a qué ciudad destino se está enviando entre 100 (códigos de 1 a 100) que se manejan en dicho despacho y por otra parte se tiene el peso del paquete. **Los paquetes vienen ordenados por código de destino.** Informar la cantidad de paquetes recibidos por ciudad y el peso promedio de los paquetes de cada una. **¿Cómo lo resolvemos?**

EJERCICIO de REPASO



```
Program votación;  
Const maxCiudad= 100; fin=0;  
Type codigosCiudad= 0.. maxCiudad;  
      paquete= record  
                codCiudad: codigosCiudad;  
                peso: real;  
      end;
```

{Aquí va el proceso de Leer y el de ProcesarUnaCiudad}

```
Var p: paquete; actual: codigosCiudad; totalxCiudad: integer; promPeso:real;
```

```
Begin  
  Leer(p);  
  While (p.codCiudad <> fin) do  
    begin  
      procesarUnaCiudad (p, actual, totalxCiudad, promPeso);  
      Writeln('Ciudad ', actual, 'tiene: ', totalxCiudad, ' y peso promedio ', promPeso);  
    End;  
  End.
```

Procedure Leer (VAR p:paquete);

Begin

readln(p.codCiudad);

if (p.codCiudad <> fin)

then readln(p.peso);

End;

**Procedure procesarUnaCiudad (VAR p: paquete; VAR actual: codigosCiudad; VAR
cantPaquetes:integer; VAR promPeso:real);**

Begin

actual:= p.codCiudad;

cantPaquetes:=0; promPeso:=0;

While (actual = p.codCiudad) do

Begin

cantPaquetes:= cantPaquetes+1;

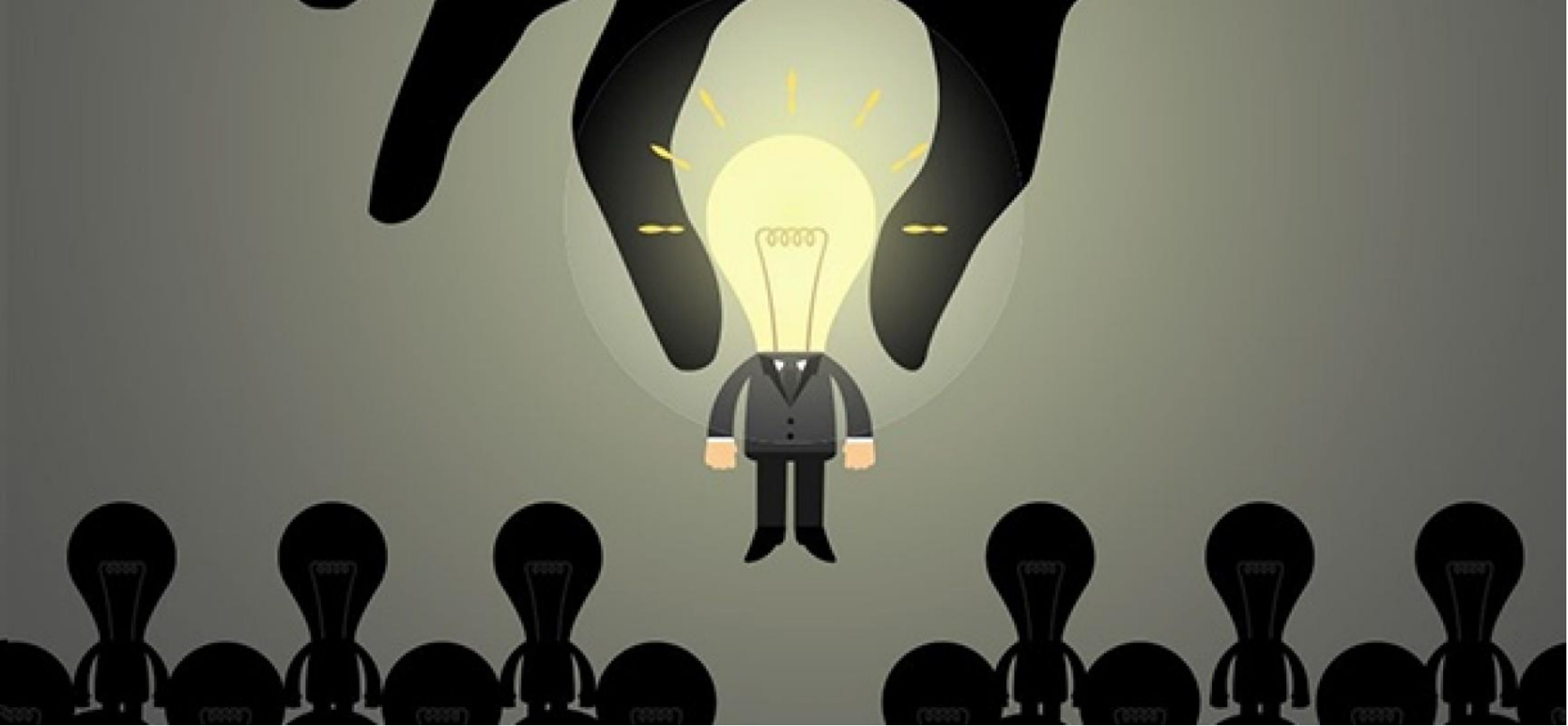
promPeso:= promPeso + p.peso;

Leer(p);

end;

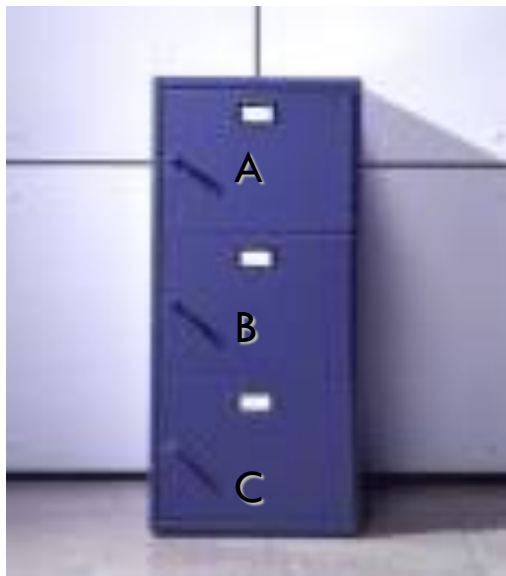
promPeso:= promPeso/cantPaquetes;

End;



Motivación

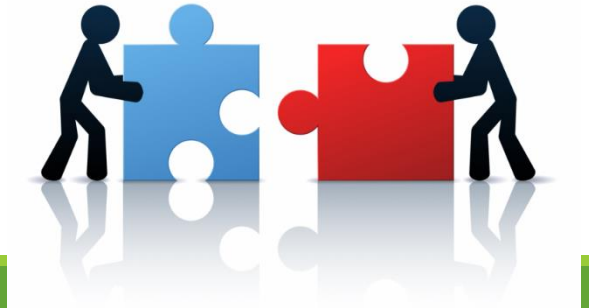
Motivación



Motivación

Supongamos que estamos recibiendo votaciones para diferentes concursantes de un programa de TV. Cada voto indica a qué participante se está votando a través de su código que va de 1 a 20. Debo informar el total de votos por cada participante.

¿Cómo lo resolvemos?



Motivación

- 
- Para resolver estos problemas podemos usar una estructura de datos tipo **arreglo**.

Un tipo de dato *Arreglo* es una colección de elementos a los que se accede mediante un índice.

SI TIENE UN SOLO INDICE SE DENOMINA VECTOR

Un arreglo (**ARRAY**) es una estructura de datos que permite acceder a cada componente a través de una **variable índice**, que da la **posición** de la componente dentro de la estructura de datos.

Conceptos iniciales



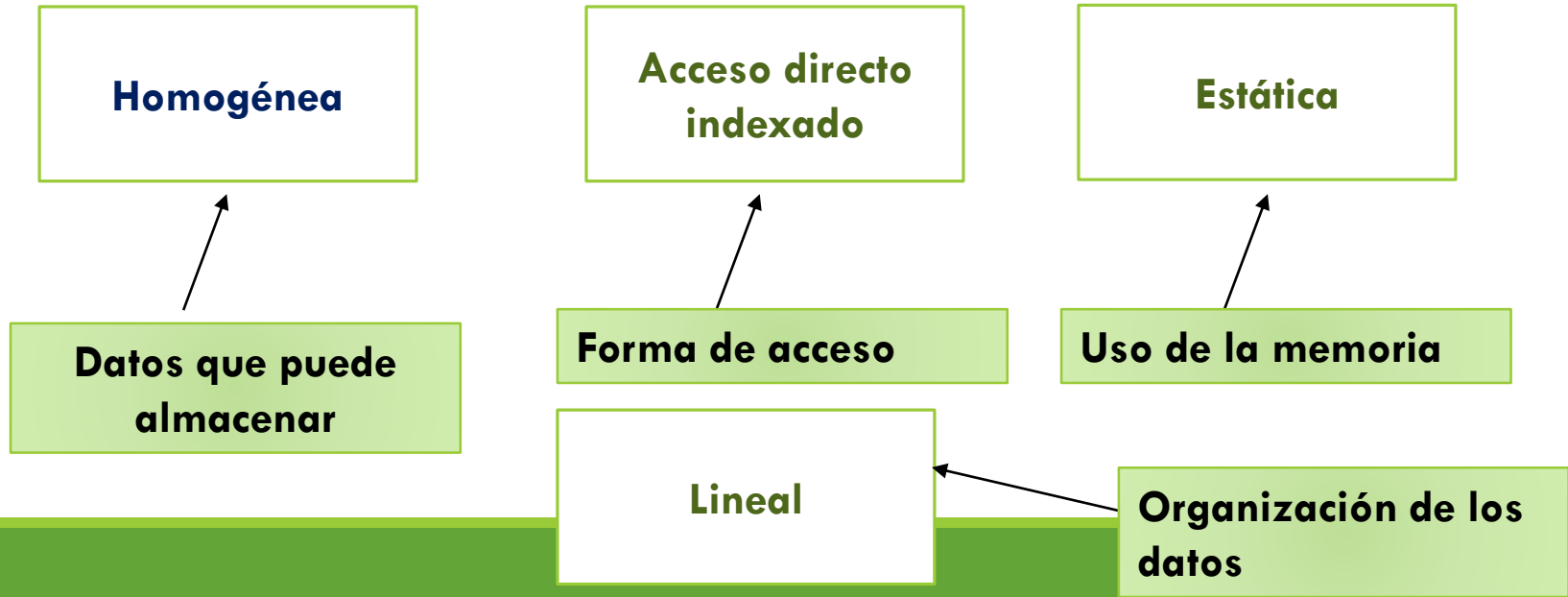
Estructura de datos-ARREGLO-Concepto

Un tipo de dato *Arreglo* es una colección *indexada* de elementos.



Estructura de datos-ARREGLO-Concepto

- Un tipo de dato *Arreglo* es una colección *indexada* de elementos.



Estructura de datos-ARREGLO-Declaración

Program uno;

type concursantes= **array**[indice] **of** tipo;

Subrangos de tipos ordinales

integer

char

boolean

enumerativo

Los **elementos** de un arreglo pueden pertenecer a **cualquier tipo de datos de asignación estática:**

- Enteros
- Reales
- Caracteres
- Registros, Otro arreglo

Estructura de datos-ARREGLO-Concepto



¿Qué es el índice de un arreglo?

Es una variable de tipo ordinal.

Permite acceder a cualquier elemento de manera directa.

Representa un desplazamiento desde la posición inicial del arreglo

Estructura de datos-VECTOR-Concepto

¿Cómo declaramos un vector en Pascal?

```
TYPE vector = array [1..10] of integer;
```

Dim física

Tipo de elemento
que almacenaré

Estructura de datos-VECTOR-Concepto

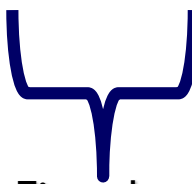
En el ejemplo

```
TYPE vector= array [1..20] of integer;
```



Dim física

A blue bracket is drawn under the range '[1..20]' in the code above, pointing down to the text 'Dim física'.



Tipo de elemento
que almacenaré

A blue bracket is drawn under the text 'of integer;' in the code above, pointing down to the text 'Tipo de elemento que almacenaré'.

Declaraciones en Pascal - Ejemplos

Type

Sucursales = Array [1..100] of real;

Temperaturas = Array [1..30] of real;

AñosAutos = Array [2000..2012] of integer;

Categorias = Array ['A' .. 'D'] of real;

Frecuencia = Array [char] of integer;

Var

Autos : añosAutos; {este vector tiene 13 componentes de tipo entero}

Frec : frecuencia; {Este vector tendrá 255 componentes de tipo entero}

Sucursal : Sucursales; {Este vector tendrá 100 componentes de tipo real}

TotalCategorias: categorias;{ este vector tiene 4 componentes de tipo real}

TempMensual: Temperaturas; { este vector tiene 30 componentes de tipo real}

VECTORES



Estructura de datos-ARREGLO-Operaciones

- Asignar elementos.
- Lectura/escritura.
- Recorridos.
- Agregar elementos.
- Insertar elementos.
- Borrar elementos.
- Búsqueda de un elemento.
- Ordenación de un arreglo.

OPERACIÓN DE ASIGNACIÓN

Estructura de datos-VECTOR-Operaciones

Program uno;

type

concurantes=array [1..20] of integer;

var

c: concurantes;

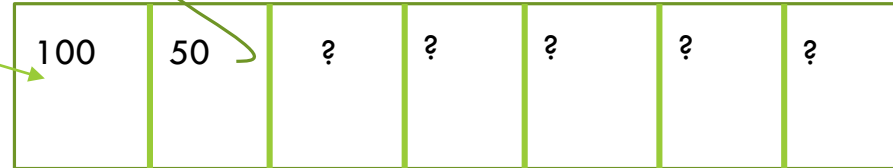
begin

c[1]:= 100;

c[2]:= 50;

.....

end.



OPERACIÓN DE CARGAR UN VECTOR (LECTURA DESDE TECLADO)

Estructura de datos-VECTOR-Operaciones

Program uno;

Const max=20;

type

concursantes= array [1..max] of integer;

var

c: concursantes;

x,i:integer;

begin

for i:= 1 to max do

begin

readln (x);

c[i]:= x;

End;

.....

for i:= 1 to max do

readln (c[i]);

?	?	?	?	?	?	?
---	---	---	---	---	---	---

100	500	200	1000	90	900	1003
-----	-----	-----	------	----	-----	------

Ejemplo

Suponga que se hay un concurso de poesía y se registran la cantidad de inscriptos por día durante el mes de septiembre. Informar los días que superan el promedio de inscriptos del mes.

Estructura de datos-VECTOR-Ejercicio

```
Program Ejemplo1;  
  CONST max=30;  
  type  
    rango: 1..max;  
    inscriptos= array [1..max] of integer;  
  var insc: inscriptos; i: rango; prom: real;  
begin  
  CargarInscriptos (insc);  
  prom: = promedio(insc);  
  For i:= 1 to max do  
    If (insc [i] > prom) then writeln ('El día ', i, 'supera el promedio del mes');  
end.
```

Estructura de datos-ARREGLO-Ejercicio

```
Procedure CargarInscriptos (var insc: inscriptos);
```

```
Var i: rango;
```

```
Begin
```

```
  For i:= 1 to max do
```

```
    readln(insc[i]);
```

```
  end.
```

```
Function Promedio (insc:inscriptos): real;
```

```
Var i: rango; total: integer;
```

```
Begin
```

```
  total:=0;
```

```
  for i:=1 to max do
```

```
    total:=total+ insc[i];
```

```
  promedio:= total/max;
```

```
End;
```

