



Facultad de Ciencias
de la **Administración**

Algoritmos y Programación

Estructuras de Datos

Unidad IV Registros

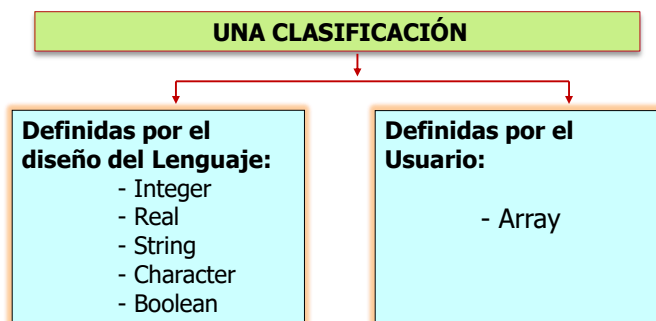
Prof. Miguel A. Fernández



Algoritmos y Programación

Repasando:

Tipo de Dato: Atributo de un objeto de dato que especifica el rango de valores y las operaciones posibles.



Arreglos Compuestos - Registros

Arreglos Unidimensionales Simples:

- Se caracterizan por el procesamiento de repeticiones de un solo tipo de dato.

Ejemplo:

Para almacenar el nombre de los 80 alumnos de la materia Algoritmos:



Arreglos Compuestos - Registros

Cómo lo definíamos:

```
Program Prueba
Var
  Alumno : array [1..80] of string;
```

IMPORTANTE
El tipo implícito DEBE tomar como referencia un tipo definido por el diseño del lenguaje (TIPO PRIMITIVO).

Pero también sabemos:

```
Program Prueba
Type
  T-Alumnos = array [1..80] of string;
Var
  Alumno : T-Alumnos;
```

IMPORTANTE
El tipo definido por el usuario (T-Alumnos) DEBE tomar como referencia un tipo definido por el diseño del lenguaje (TIPO PRIMITIVO).

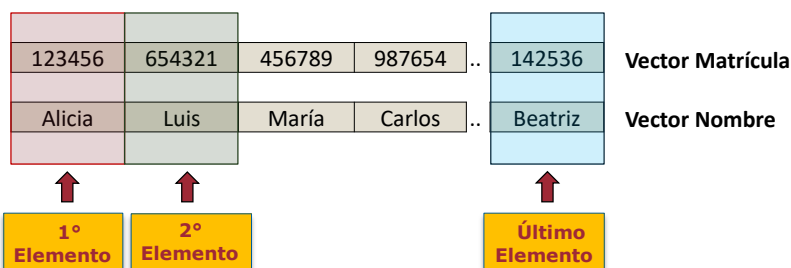
Arreglos Compuestos - Registros

Arreglos Unidimensionales Simples Encadenados:

- Son varios arreglos que tienen la misma dimensión y que almacenan datos que tienen alguna relación entre sí.

Ejemplo:

Para almacenar el Nro.de matrícula y el nombre, de los 80 alumnos de Algoritmos:



Arreglos Compuestos - Registros

OBJETO DE DATOS

A un lugar donde se almacenan datos, en tiempos de ejecución se los suele denominar "Objeto de Datos".

¿ Qué almacenan los "objetos de datos" ?

- Los objetos de datos simples o escalares pueden almacenar un solo dato (variables).
- Los objetos de datos tipo array pueden almacenar conjuntos de datos pero del mismo tipo (integer, real, string, etc.).

¿ Y cuántos datos de un mismo tipo se puede almacenar en un elemento de un arreglo ?



¡ SOLO UNO !

Arreglos Compuestos - Registros

TODO BIEN, PERO.....

¿ Será posible representar varios datos en un mismo elemento de un solo arreglo ?

Y si resulta posible, ¿ podrán esos datos tener distintos tipos ?



Como respuesta a estos interrogantes, los lenguajes de programación:

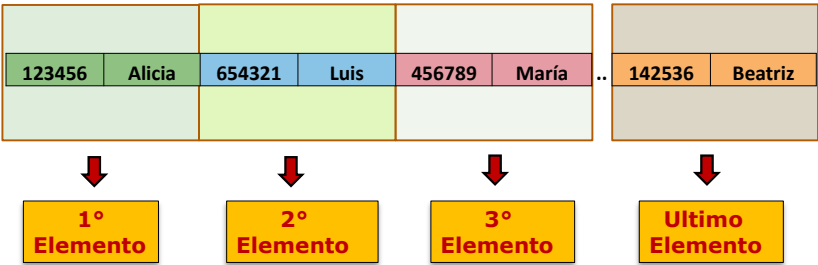
Proveen mecanismos que permiten agrupar datos en un MISMO elemento, incluso de ¡¡¡ distintos tipos !!!



Arreglos Compuestos - Registros

En el caso del ejemplo de los datos de los alumnos, podríamos agrupar todos los datos en un solo arreglo.

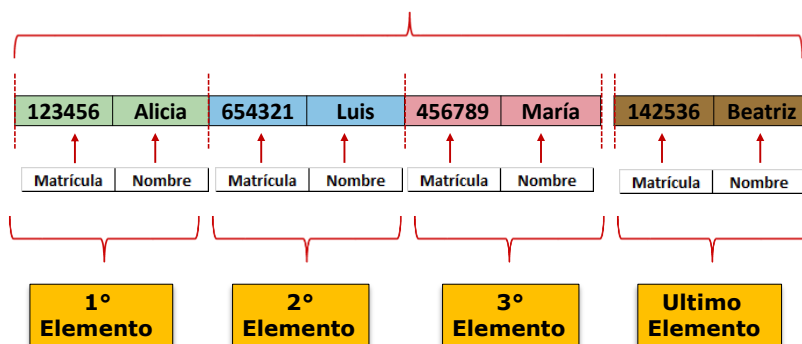
El arreglo almacenará los dos datos de cada alumno (matrícula y nombre) en un mismo elemento.





Arreglos Compuestos - Registros

Array "Alumno"



Cada elemento del vector "Alumno" tiene dos **CAMPOS**:

- Matricula
- Nombre

Arreglos Compuestos - Registros

Formas de referenciamiento:

- a) Algunos lenguajes aceptan referenciamiento por nombre de campo solamente:

Imprimir Matricula(10)

Ingresar Nombre(15)

- b) Pascal requiere la siguiente sintaxis:

Nombre Arreglo [subíndice] . Nombre de Campo

Ejemplo:

Imprimir Alumno[10].Matricula

Ingresar Alumno[15].Nombre



Arreglos Compuestos - Registros

En Pascal: ESPECIFICACIÓN

Pascal proporciona un tipo nuevo de datos llamado “**Record**” que permite definir estructuras que agrupan datos de distintos tipos.

Ejemplo: Se deben procesar los datos de los 56 postulantes a un empleo:

- Apellido y Nombre
- Nro.de Documento
- Dirección
- Edad



```

Type
  R-Postula = Record
    APNO : string;
    DOCU : integer;
    DIRE : string;
    EDAD : integer;
  end;
Var
  V-Postula : array [ 1..56] of R-Postula;
  
```

Arreglos Compuestos - Registros

En Resumen:



Se ha definido:

- Una objeto de nombre: “V-Postula”
- El objeto “V-Postula” es un array de 56 elementos donde cada elemento es del tipo “R-Postula”
- El tipo “R-Postula” es un Record compuesto por los siguientes campos:

“APNO” del tipo primitivo string (cadena).

“DOCU” del tipo primitivo entero.

“DIRE” del tipo primitivo string.

“EDAD” del tipo primitivo entero.



```

Type
  R-Postula = Record
    APNO : string;
    DOCU : integer;
    DIRE : string;
    EDAD : integer;
  end;
Var
  V-Postula : array [ 1..56] of R-Postula;
  
```



Arreglos Compuestos - Registros

Para una solicitud de empleo que ha publicado una empresa, se han presentado 5 postulantes. Se requiere:

- a) Ingresar los siguientes datos de cada postulante y almacenarlos en una estructura de datos:
 - Apellido y Nombre
 - Número de Documento
- b) Imprimir un listado que contenga:

Apellido y Nombre – Nro.de Documento



Planilla de aspirantes inscriptos

Nombre	Documento
María	111122
José	212122
Laura	543216
Roberto	123456
Inés	987654



Arreglos Compuestos - Registros

Program Aspirantes

Type

```
R-Aspira = record
  R-Nom : string;
  R-Doc : integer;
end;
```

Var

```
VecAsp = array [1..5] of R-Aspira;
```

Cómo
definimos el
registro

Representación gráfica

Elemento 1		Elemento 2		Elemento 3		Elemento 4		Elemento 5	
R-Nom	R-Doc	R-Nom	R-Doc	R-Nom	R-Doc	R-Nom	R-Doc	R-Nom	R-Doc

También se puede representar:

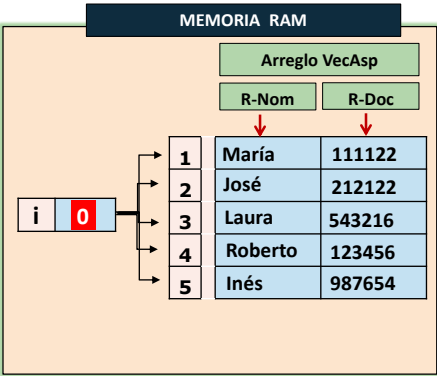
Elemento 1	R-Nom	R-Doc
Elemento 2	R-Nom	R-Doc
Elemento 3	R-Nom	R-Doc
Elemento 4	R-Nom	R-Doc
Elemento 5	R-Nom	R-Doc



Arreglos Compuestos - Registros

```
Program Aspirantes
Type
  R-Aspira = record
    R-Nom : string;
    R-Doc : integer
  end;
Var
  VecAsp = array [1..5] of R-Aspira;
  i : integer

Inicio
  Ingresar i
  Mientras i <> 0
    Ingresar VecAsp[i].R-Nom
    Ingresar VecAsp[i].R-Doc
  Ingresar i
  FinMientras
FIN
```



Fin de Proceso

```
Program Aspirantes
Type
  R-Aspira = record
    R-Nom : string;
    R-Doc : integer
  end;
Var
  VecAsp = array [1..5] of R-Aspira;
  i : integer;

Inicio
  Ingresar i
  Mientras i <> 0
    Si i < 0 or i > 5
      Mostrar "Error"
    Sino
      Ingresar VecAsp[i].R-Nom
      Ingresar VecAsp[i].R-Doc
    FinSi
  Ingresar i
  FinMientras
FIN
```

Arreglos Compuestos - Registros

Ingreso con control



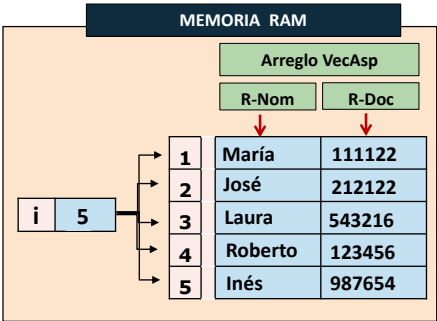


Arreglos Compuestos - Registros

```
Program Aspirantes
Type
  R-Aspira = record
    R-Nom : string;
    R-Doc : integer;
  end;
Var
  VecAsp = array [1..5] of R-Aspira;
  i : integer

Inicio
  Para i = 1, 5, 1
    Imprimir VecAsp[i].R-Nom
    Imprimir VecAsp[i].R-Doc
  FinPara
FIN
```

Fin de Proceso



Impresora

María	111122
José	212122
Laura	543216
Roberto	123456
Inés	987654



Arreglos Compuestos - Registros

EJERCICIO

Para una solicitud de empleo que ha publicado una empresa, se han Presentado 56 postulantes. Se requiere:

- a) Ingresar los siguientes datos de cada postulante:
 - Apellido y Nombre
 - Número de Documento
 - Dirección de Contacto
 - Edad

- b) Finalizado el ingreso de datos, imprimir un listado:

Apellido y Nombre – Nro.de Documento – Dirección - Edad

Al finalizar el listado imprimir el apellido y nombre del postulante de mayor edad.

```

Begin
  Ingresar i
  Mientras i <> 0
    Si i < 0 or > 56
      Mostrar "Fuera de Rango"
    Sino
      Ingresar V-Pos[i].ApNo
      Ingresar V-Pos[i].Docu
      Ingresar V-Pos[i].Direc
      Ingresar V-Pos[i].Edad
    FinSi
    Ingresar i
  FinMientras

  MAX := Min-Valor
  Para i = 1, 56, 1
    Imprimir V-Pos[i].ApNo, V-Pos[i].Docu
      V-Pos[i].Direc, V-Pos[i].Edad
    Si V-Pos[i].Edad > MAX
      MAX      := V-Pos[i].Edad
      May-Nom := V-Pos[i].ApNo
    FinSi
  FinPara
  Imprimir "Postulante > Edad: " May-Nom
END

```

Arreglos Compuestos - Registros

SOLUCIÓN

```

Program Postulante
Type
  R-Postula = Record
    ApNo : string;
    Docu : integer;
    Direc : string;
    Edad : integer
  end;
Var
  V-Pos: array [ 1..56] of R-Postula;
  MAX, i : integer;
  May-Nom : string;

```



Arreglos Compuestos - Registros

EJERCICIO

Dado el ejercicio anterior codificar los siguientes procesos:

- Imprimir la cantidad de postulantes de acuerdo con el siguiente rango de edades:
 - Menores de 30 años
 - Entre 31 y 40 años
 - Entre 41 y 50 años
 - Mas de 50 años
- Mostrar todos los datos de un aspirante cuyo número de documento ingresa el operador.



Facultad de Ciencias
de la **Administración**

Algoritmos y Programación



Registros

**FIN DE LA
CLASE**