



Facultad de Ciencias  
de la **Administración**

**Licenciatura en Sistemas**



## *Algoritmos y Programación*

# *Estructura de Datos*

## *Análisis de Casos*

### *Parte 1*



## **Unidad IV**

**Prof. Miguel Fernández**

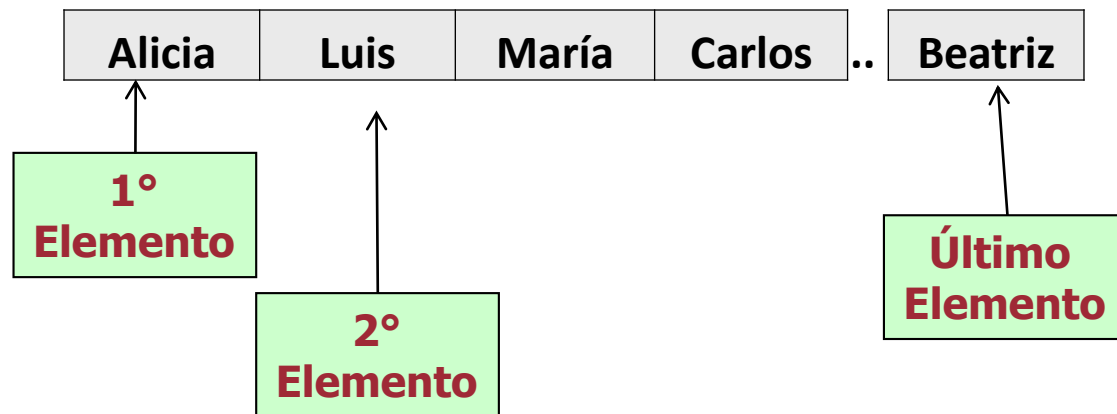
# Estructura de Datos

## Arreglos Unidimensionales Simples:

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

### Ejemplo:

Para almacenar el nombre de los 180 alumnos de la materia Algoritmos:



# Estructura de Datos

---

Cómo lo definíamos:

**Program Prueba**

**Var**

**Alumnos : array [ 1..180] of string;**

Para asignar valor cero al elemento ubicado en la posición 50:

**Alumnos [ 50] := 0**

Para imprimir el dato del elemento ubicado en la posición 77:

**Imprimir Alumnos [ 77]**

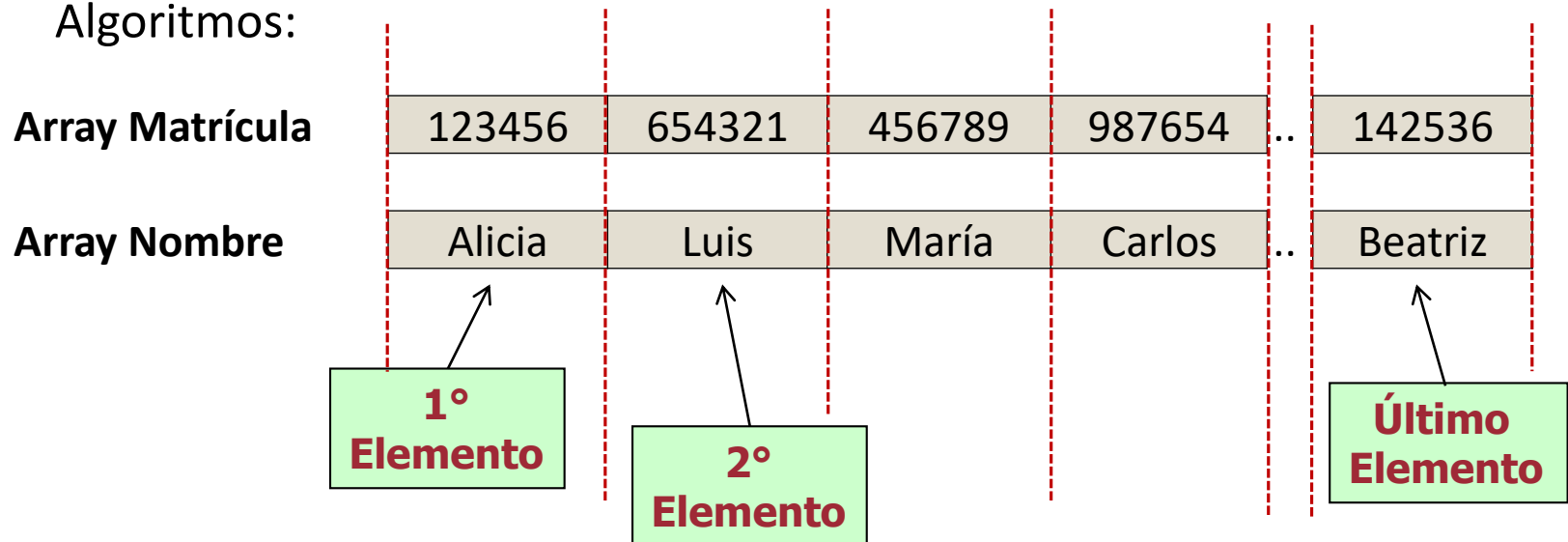
# Estructura de Datos

## 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 180 alumnos de Algoritmos:



# Estructura de Datos

Cómo lo definimos:

**Program Prueba**

**Var**

**Matrícula : array [ 1..180] of integer;**

**Nombre : array [ 1..180] of string;**

**Ejemplos:**

- a) Ingresar el nombre y la matrícula de un alumno y grabarlos en la primera posición

**Ingresar Matricula(1), Nombre(1)**

- b) Recorrer el array e imprimir sus contenidos

**Para i = 1, 180, 1**

**Imprimir Matricula(i), Nombre(i)**

**FinPara**

# Estructura de Datos

## Ejemplo 1:

Dado el ejemplo anterior, ingresar los datos de los 180 alumnos y luego de finalizado el ingreso codificar un algoritmo de consulta tal que ingresando el número de matrícula se imprima el nombre del alumno.

### ¿ Cuántos arreglos se necesitan ?

**Arreglo 1** – Para almacenar los nombres de cada uno de los alumnos

- 180 elementos
- tipo string

**Arreglo 2** – Para almacenar los números de matrícula de cada alumno

- 180 elementos
- tipo entero

# Estructura de Datos

Program Prueba

Var

Matricula : array [1..180] of integer;

Nombre : array [1..180] of string;

i, Mat : integer;

Encontrado: Boolean;

Inicio

Mostrar "Ingrese N° Matrícula (0-Fin)

Mostrar "luego ingrese Nombre"

Ingresar Mat

Mientras Mat  $\neq$  0

Si Mat  $<$  0 or Mat  $>$  180

Mostrar "Fuera de Rango

Sino

Ingresar Matricula(Mat)

Ingresar Nombre(Mat)

FinSi

Ingresar Mat

FinMientras

Mostrar "Ingrese consulta (0-Fin)"

Ingresar Mat

Mientras Mat  $\neq$  0

Encontrado := False

i := 0

Mientras i  $<$  180 and

Encontrado = False

i := i + 1

Si Matricula(i) = Mat

Mostrar Nombre(i)

Encontrado := True

Finsi

FinMientras

Si Encontrado = False

Mostrar "No Encontrado"

FinSi

Ingresar MAT

FinMientras

**Fin**

# Estructura de Datos

## Ejemplo 2:

Se debe procesar los gastos que realizaron durante un mes cada una de las distintas dependencias de un Ministerio. Se requiere:

- a) Ingresar los nombres de las 14 dependencias del Ministerio
- b) Ingresar de cada factura de gasto los siguientes datos:
  - Nro – Número de dependencia (rango de 1 a 14)
  - Impo - Importe del gasto realizado

Al finalizar el ingreso de datos se debe imprimir un listado que contenga los totales de gastos discriminados por dependencia

**Nombre de la Dependencia – Total de gastos del mes**



# Estructura de Datos

## Ejemplo 2:

### ¿ Cuántos arreglos se necesitan ?

**Arreglo 1** – Para almacenar los nombres de cada dependencia

- 14 elementos
- tipo string

1	2	3	4	...	14
Administración	Archivo	Contable	Liquidaciones	.....	Despacho

**Arreglo 2** – Para acumular los gastos de cada dependencia

- 14 elementos
- tipo real
- Inicializar al inicio

1	2	3	4	...	14
0	0	0	0	.....	0

# Estructura de Datos

Program libros

Var

DepNom : array [1..14] of string;

DepAcu : array [1..14] of real;

Impo : real;

i, Nro : integer;

Inicio

Mostrar "Ingrese N° Dep. (0-Fin)

Ingresar Nro

Mientras Nro <> 0

Si Nro < 0 or Nro > 14

Mostrar "Fuera de Rango"

Sino

Ingresar DepNom(Nro)

DepAcu(Nro) := 0

FinSi

Ingresar Nro

FinMientras

Se definen ambos arreglos

Se ingresan los nombres de cada dependencia y se almacena en el Array "DepNom".

Se inicializa al array "DepAcu" para Luego acumular los gastos de cada Dependencia.

# Estructura de Datos

## Continúa Ejercicio



Se acumula en el array "DepAcu"  
Utilizando el número de depen-  
dencia (Nro) como subíndice.

Se imprime listado con los nombres  
de las dependencias (DepNom) y los  
Importes acumulados (DepAcu).

```
Mostrar "Ingrese datos Gastos (0-Fin)"
Ingresar Nro
Mientras Nro <> 0
    Ingresar Impo
    DepAcu(Nro) := DepAcu(Nro) + Impo
    Ingresar Nro
FinMientras
Mostrar "Impresión de Totales"
Para i = 1, 14, 1
    Imprimir DepNom(i), DepAcu(i)
FinPara
Fin
```

# Estructura de Datos

## Ejercicio Completo

Program libros

Var

DepNom : array [1..14] of string;

DepAcu : array [1..14] of real;

Impo : real;

i, Nro : integer;

Inicio

Mostrar "Ingrese N° Dep. (0-Fin)"

Ingresar Nro

Mientras Nro  $\neq$  0

Si Nro  $<$  0 or Nro  $>$  14

Mostrar "Fuera de Rango"

Sino

Ingresar DepNom(Nro)

DepAcu(Nro) := 0

FinSi

Ingresar Nro

FinMientras

Mostrar "Ingrese datos Gastos (0-Fin)"

Ingresar Nro

Mientras Nro  $\neq$  0

Ingresar Impo

DepAcu(Nro) := DepAcu(Nro) + Impo

Ingresar Nro

FinMientras

Mostrar "Impresión de Totales"

Para i = 1, 14, 1

Imprimir DepNom(i), DepAcu(i)

FinPara

**Fin**

**...Y Ahora ...  
¡ A Trabajar !**



# Estructura de Datos

## Ejercicio 1

Se deben procesar los datos de las ventas de libros en una Feria de Libros. En la feria existe 15 stands de distintas editoriales, y se deben codificar los siguientes procesos:

- a) Ingresar el nombre de cada una de las 15 editoriales que tienen stand en la feria
- b) Ingresar los siguientes datos de cada libro que se vende:

Nro : Número de stand de la editorial donde se vendió (rango de 1 a 15)

TitLib : Título del libro

Impo : Importe de la venta

De cada venta imprimir:

**Nombre de la Editorial - Título del libro – Importe**

- c) Al final del proceso imprimir los importes totales de la venta discriminado por editorial

**Nombre de la Editorial - Importe de la venta**

# Estructura de Datos

Program libros

Var

StaNom : array [1..15] of string;

StaAcu : array [1..15] of real;

TitLib : string;

Impo : real;

i, Nro : integer;

Inicio

Mostrar "Ingrese N° Stand (0-Fin)

Ingresar Nro

**Mientras** Nro  $\neq$  0

Si Nro  $<$  0 or Nro  $>$  15

Mostrar "Fuera de Rango"

Sino

Ingresar StaNom(Nro)

StaAcu(Nro) := 0

FinSi

Ingresar Nro

**FinMientras**

Mostrar "Ingrese datos Ventas (0-Fin)"

Ingresar Nro

**Mientras** Nro  $\neq$  0

Ingresar TitLib, Impo

Imprimir StaNom(Nro), TitLib, Impo

StaAcu(Nro) := StaAcu(Nro) + Impo

Ingresar Nro

**FinMientras**

Mostrar "Impresión de Totales"

**Para** i = 1, 15, 1

Imprimir StaNom(i), StaAcu(i)

**FinPara**

**Fin**

## Ejercicio 2

Se deben procesar las ventas de una estación de servicio de un día, realizadas por cada uno de los 18 empleados de playa. Se requiere procesar:

- a) Ingresar los nombres de los 18 empleados de playa
- b) Ingresar los siguientes datos de cada venta:
  - Nro : Número de empleado (rango de 1 a 18)
  - CodCom : Código de combustible (1-Nafta, 2-GasOil, 3-querosene)
  - CanLit : Cantidad de litros vendidos
  - Impo : Importe de la venta

Al finalizar el proceso imprimir:

- a) Importes totales de ventas discriminadas por empleado:

**Nombre del Empleado - Importe total vendido en el día**

- b) Cantidad total de litros vendidos discriminados por código de combustible.





Facultad de Ciencias  
**UNER** de la **Administración**

## *Algoritmos y Programación*



***Estructura de Datos***  
***Análisis de Casos***  
***Parte 1***

**FIN DE LA  
CLASE**