


Departamento de Sistemas, Estadística e Informática



# ARREGLOS



# Agenda

- Definición de un Arreglo.
  - Clasificación de los arreglos.
  - Declaración y creación de arreglos.
  - Inicialización y acceso a los arreglos.
  - Ejercicios.
- 

# Arreglos (Array)

- Un arreglo es una secuencia de datos primitivos o de objetos, todos del mismo tipo, unidos bajo un identificador común.
- Todos los elementos de un arreglo tienen el mismo nombre pero se diferencian por la posición que ocupan en él.
- Son de tipo referencia
- Sinónimos:
  - Vector
  - Tabla
  - Matriz

# Clasificación

- **Unidimensionales(Vectores)**

También llamado vector, almacena datos indicando un índice.

- Ejm: `num[i];`

- **Bidimensionales(Matrices)**

Almacena datos en forma de una matriz. Para acceder a cada posición de un elemento dentro de un arreglo hay que indicar 2 índices.

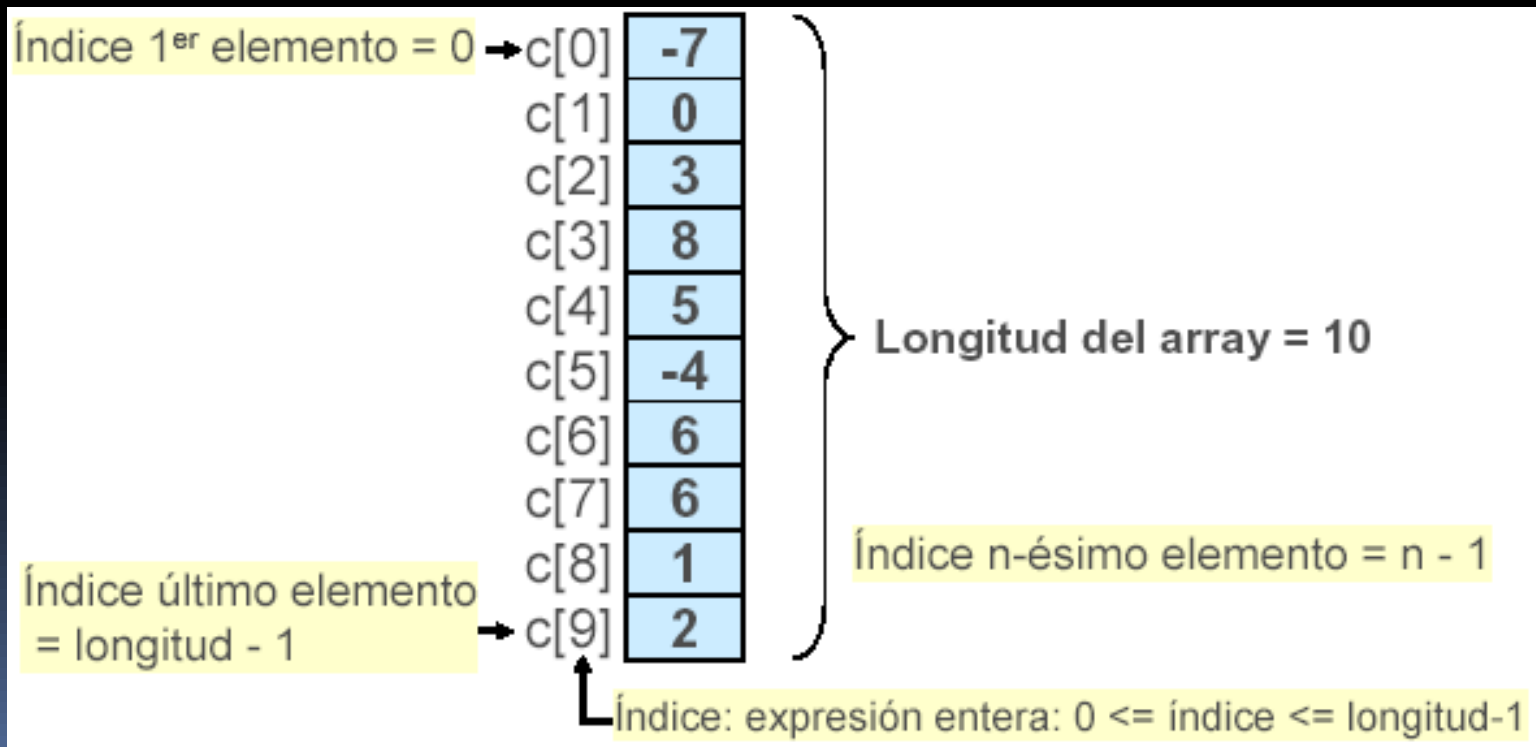
- Ejm: `m[i][j].`

- **Multidimensionales:**

Son arreglos que tienen más de dos dimensiones. Por cada dimensión del arreglo se tiene que utilizar un índice para ubicar exactamente a un elemento en particular.

# Arreglos Unidimensionales

- Cada elemento de un arreglo está asignado a una posición concreta del arreglo, designada por un índice.



# Declaración de Arreglos

- La declaración de un arreglo unidimensional se puede hacer de 2 formas:
  - 1. tipoDato **nombreArreglo[]**;
  - 2. **tipoDato[]** nombreArreglo;
  - Ejm: `int notas[ ];` ó `int[ ] notas;`
- En este ejemplo se está declarando un arreglo llamado notas que almacenará valores enteros (int).

# Creación de Arreglos

- Para crear un arreglo se usa el operador **new**, el cual reserva un espacio en memoria.
  - nombreArreglo = **new tipoDato[tamaño]**
    - Ejm: **notas = new int[4];**
      - Indica que el arreglo estará compuesto de 4 valores enteros.
- Tanto la declaración y creación de un arreglo se puede hacer en una sola línea:  

```
int notas[] = new int[4]; int[] notas = new int[4];
```
- Los elementos se inicializan a cero, false o null dependiendo del tipo de dato en el momento de su creación.

# Inicialización de Arreglos

- Los valores del arreglo se inicializan utilizando el índice del mismo entre corchetes:
  - Ejm: `notas[2]=8;`
- Los arreglos se pueden inicializar con los valores entre llaves {...} se parados por comas, en la propia declaración.

**Ejm:**

- `int notas[] = {8,7,9};` ó también
- `int notas[] = new int[] {8,7,9};`
- Esto significa que el arreglo se inicializará con los siguientes valores:  
`notas[0]=8, notas[1]=7, notas[2]=9`



# Acceso a los Arreglos

- Existe 2 formas de acceder a un arreglo:

## 1. Consulta de su longitud:

Se utiliza el método **length** que proporciona el tamaño de un arreglo.

**nombreArreglo.length**

Ejm:

**int notas[]=new notas[3]      -> notas.length=3**

## 2. Indexación:

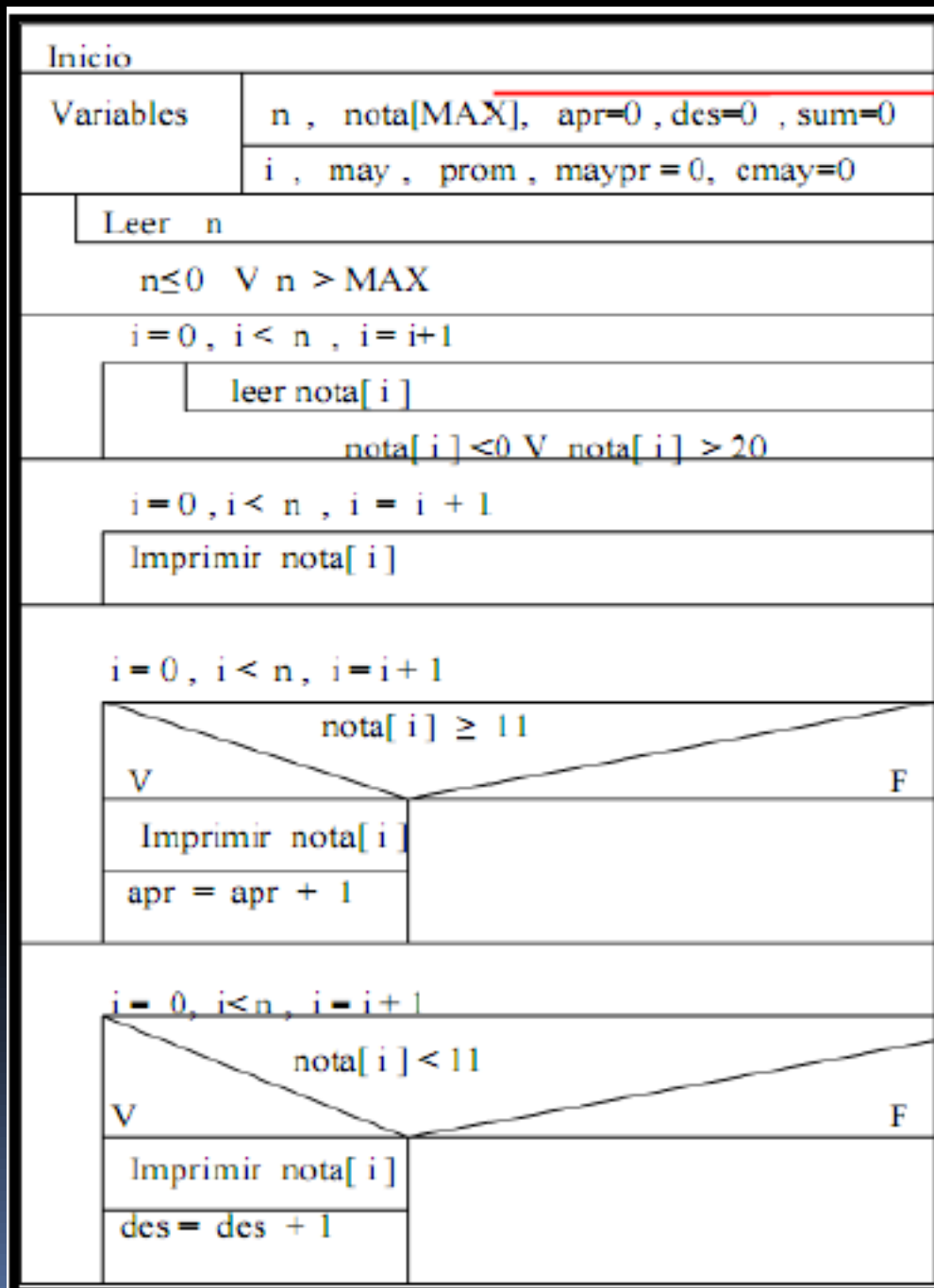
Accede a los elementos de un arreglo a través de su índice. El índice siempre tiene un rango entre 0 y N-1, siendo N el tamaño del arreglo (arreglo.length)

**nombreArreglo[i]**

# Ejercicio

- Escribir un programa que lea las notas de  $n$  alumnos y reporte lo siguiente:
  1. Todos los datos ingresados.
  2. La lista de alumnos aprobados, indicando cuantos son.
  3. La lista de alumnos desaprobados, indicando cuantos son.
  4. La nota promedio.
  5. La lista de alumnos con nota mayor o igual al promedio, indicando cuantos son.
  6. La nota mayor.
  7. La lista de alumnos con nota igual a la nota mayor, indicando cuantos son.

**Nota:** Utilizar una estructura **for** para cada caso



Max=100

Lee el nro de datos que se va a ingresar

Se ingresan las notas entre 0 y 20

Se imprime todas las notas

Se imprimen las notas aprobatorias y se lleva la cuenta de estas

Se imprimen las notas desaprobatorias y se lleva la cuenta de estas

$i = 0, i < n, i = i + 1$

$sum = sum + nota[i]$

$prom = sum / n$

$i = 0, i < n, i = i + 1$

V

$nota[i] \geq prom$

F

Imprimir  $nota[i]$

$maypr = maypr + 1$

$may = nota[0]$

$i = 0, i < n, i = i + 1$

V

$nota[i] > may$

F

$may = nota[i]$

$i = 0, i < n, i = i + 1$

V

$nota[i] = may$

F

$cmay = cmay + 1$

Imprimir:  $apr, des, prom, maypr, may, cmay$

Fin

Se suman las notas para calcular el Promedio.

Se determinan las notas mayores o iguales al promedio y se cuentan.

Se determinan la nota mayor.

Se buscan las notas iguales a la nota mayor y se lleva la cuenta de éstas