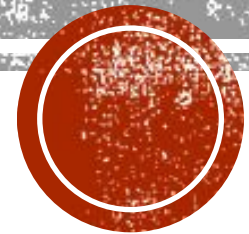


# PROGRAMACIÓN

## UNIDAD 4: Estructuras de almacenamiento



Curso / Ciclo Formativo: 1º Desarrollo de aplicaciones web

Profesora: María Navarro Elbal

# Arrays

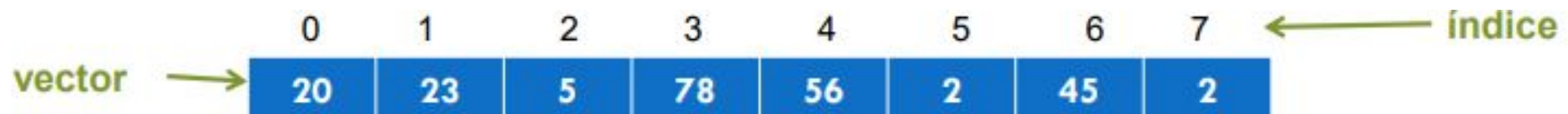
## Arrays

- **Arrays** → conjunto de datos del mismo tipo que se almacenan en posiciones consecutivas de memoria.
- **Nombre o identificador** del array → se asocia con la dirección de memoria donde comienza de la estructura.
- **Índice** → representa el desplazamiento que hay que aplicar sobre esa dirección para llegar a la dirección que ocupa el elemento que tiene ese índice



# Arrays lineales o vectores

- ❑ Estructura de almacenamiento homogénea lineal estática-
  - ❑ **Homogénea.** Todos los elementos son del mismo tipo
  - ❑ **Lineal.** Cada elemento tiene un elemento anterior y otro posterior, excepto el primero y el último .
  - ❑ **Estática.** Tamaño definido que no se puede incrementar o decrementar en tiempo de ejecución.
- ❑ Serie de posiciones consecutivas en memoria
- ❑ **Acceso:** mediante un índice (de tipo int)
- ❑ El **nombre** o identificador se asocia con la dirección de memoria donde comienza la estructura





# Arrays o vectores

## Arrays o vectores

### □ Declaración de vectores

```
<tipo> [ ] <nombre>  
ó  
<tipo> <nombre>[ ];
```

```
byte[] temperaturas;  
byte temperaturas[];
```

Declaración de la referencia  
que apuntará al objeto de  
tipo vector unidimensional de  
enteros

### □ Creación de vectores.

```
<nombre> = new <tipo> [<tamaño>];
```

```
temperaturas = new byte [100];  
// "temperaturas" apunta al vector
```

Referencia apuntando a un nuevo  
array de 100 elementos de tipo  
entero

### ■ Posiciones

- 0 .. (N-1)
- Sólo valores enteros
- Primer valor siempre cero.

```
// 2 Opciones de declaración y creación del vector  
int n=100;  
byte[] temperaturas;  
temperaturas = new byte[n];  
  
int n=100;  
byte[] temperaturas = new byte[n];
```



# Arrays o vectores

- ❑ Inicialización
  - ❑ A valores predeterminados
    - ❑ numéricas: 0
    - ❑ objetos: null (indica que no apunta a ningún elemento)
    - ❑ boolean: false
    - ❑ caracteres: '\u0000'
  - ❑ A valores concretos

```
byte [ ] vector = {10, 11, 12, 11, 10, 9, 10, 12, 18, 15}  
//creación de un vector de tamaño 10 con los valores especificados.
```

- ❑ Puede hacer uso de lenght, que indica la longitud de array

```
<nombre_vector>.length;
```



# Arrays o vectores

- ❑ Acceso a los elementos

`<nombre_matriz> [<índice>]`

```
System.out.println ("El quinto día de la semana es: " + diaSemana[4]);
```

- ❑ Se intenta acceder a un elemento que no existe → [ArrayIndexOutOfBoundsException](#)
- ❑ Recorrido de un array → [for](#)

```
//Recorrido de todos los elementos de un array con un for  
for (i=0; i<10; i++) { //Es preferible el uso de length  
    System.out.println (vector[i]);  
}
```

