

### Arreglos unidimensionales

Un arreglo es una estructura de datos que permite almacenar múltiples valores en un mismo identificador

Cada uno de los valores es identificado por un número natural iniciado en 0

Todos los valores almacenados deben ser del mismo tipo, el cual se indica en la declaración del arreglo

La cantidad de elementos que se pueden almacenar está definida por el tamaño del arreglo, el cual se asigna durante la creación del mismo y no puede cambiar durante la ejecución del programa

### Declaración

`tipo[] nombre;` Declara un vector de un tipo

`tipo nombre[];` Forma alternativa

Las dos formas son equivalentes, excepto si en la misma línea se declara otra variable.

`tipo []nombre1, nombre2;` ← nombre2 es un vector

`tipo nombre1[], nombre2;` ← nombre2 es una variable

### Creación

`nombre = new tipo[tamaño]`

Asigna al arreglo el tamaño indicado

El tamaño siempre tiene que ser un número entero mayor a 0.

Puede indicarse con una constante, una variable o una expresión.

Si el tamaño se indica con una variable, el arreglo es creado con tantos elementos como el valor de la variable.

### Acceso

`nombre [in dice]`

El operador de indexación (`[]`) recibe un número entero y devuelve el elemento almacenado en el arreglo en ese índice

Cada elemento se comporta como una variable

Por lo tanto puede ser asignado, obtenido para una expresión, impreso por pantalla, etc.

Si el índice indicado está fuera de los extremos (menor a cero o mayor al tamaño) el programa se interrumpe con una excepción `IndexOutOfBoundsException`

### Tamaño

`nombre.length`

El tamaño se obtiene con la propiedad `length`

Dado que los índices comienzan en 0, el último índice válido es `nombre.length-1`

### Recorrido

```
for (int i = 0; i < nombre.length; i++)  
{  
    bloque iterativo accediendo al arreglo con i  
}
```

Para recorrer un arreglo con un ciclo `for` se accede al mismo usando la variable contadora como índice del arreglo



By diegojserrano

Published 17th April, 2019.

Last updated 18th April, 2019.

Page 1 of 2.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

### Instrucción for mejorada

```
for (tipo variable: nombre) {  
    bloque iterativo accediendo a la variable  
}
```

`nombre` es el nombre del arreglo

`tipo` es el tipo con el que se definió el arreglo (no es siempre `int`)

`variable` es una variable local al ciclo

La variable es asignada automáticamente en cada vuelta a cada valor almacenado en el vector

### Limitaciones del for mejorado

No permite modificar el contenido del arreglo

Recorre todo el arreglo completo, no permite recorrer sólo una porción

Recorre desde el índice 0 hasta el último, no permite recorrer en forma inversa

### Arreglos de objetos

El tipo del arreglo puede ser un tipo primitivo o el nombre de una clase

Cuando el arreglo es de objetos, la creación con `new tipo[tamaño]` lo deja lleno de referencias nulas

Para que el arreglo contenga objetos deben crearse individualmente cada uno de ellos llamando a `new tipo(parámetros)` dentro del constructor

```
class [] nombre;  
nombre = new class[tamaño];  
  
for (int i = 0; i < nombre.length; i++)  
{  
    nombre[i] = new class(parámetros);  
}
```



By [diegojserrano](#)

[cheatography.com/diegojserrano/](https://cheatography.com/diegojserrano/)

Published 17th April, 2019.

Last updated 18th April, 2019.

Page 2 of 2.

Sponsored by [Readable.com](#)

Measure your website readability!

<https://readable.com>