



# Curso de Java Standard



Ing. Octavio Robleto



octavio.robleto@gmail.com



<https://octaviorobleto.com>



# Introducción

En muchas ocasiones nos encontraremos con la necesidad de utilizar muchas variables que pueden tener el mismo sentido; para esto Java nos proporciona un tipo de objeto llamado arreglos.

Consiste en almacenar bajo un mismo identificador un conjunto de variables que pertenecen al mismo tipo.

Al tener todos los valores bajo un mismo nombre accedemos a cada elemento a través de un índice, este índice comienza con el número cero, por lo tanto el último elemento se encuentra en el índice ( $\text{longitudArreglo} - 1$ )

		nombres			
Índice	Valor	0	1	2	3
		Octavio	Mariana	Analía	Ariel

  

		numeros		
Índice	Valor	0	1	2
		23.3	24.6	16.0

# Arreglos Unidimensionales

En java podemos crear arreglos unidimensionales y también de dos dimensiones (Matrices) para este curso solo veremos los de una sola dimensión, pero tengamos en cuenta que son muy parecidos.

## Hay dos formas de declarar un arreglo en Java:

Las dos formas son correctas pero se recomienda la primera ya que es mas utilizada en diferentes lenguajes

```
tipo[] identificador;  
tipo identificador[];
```

## También hay dos formas de inicializarlos:

1. La primera forma la utilizamos cuando tenemos los valores iniciales definidos.
2. La segunda vemos que utiliza una nueva sentencia “**new**” seguido del tipo de dato y entre corchetes va la longitud del mismo, esto le indica a java que construya un nuevo arreglo y después se le asignan los valores correspondientes en el índice indicado.

```
identificador = { valor1 , valor2 , valorn };  
identificador2 = new tipo[n];  
identificador2[0] = valor1;  
identificador2[1] = valor2;  
identificador2[n] = valorn;
```

# Acceder a los datos

Para acceder a cada uno de los elementos que posee el arreglo debemos indicar que posición a través del índice.

```
System.out.println("Valor: " + (identificadorArreglo[0]));
```

Ahora bien, si el arreglo hay que accederlo elemento por elemento tenemos unas estructuras mucho mas cómodas para hacer eso y son los bucles, muy particularmente contamos con uno donde debemos saber la cantidad de iteraciones a realizar.

Aunque podemos colocar el numero en el ciclo directamente, esto no es recomendable, para evitar eso podemos acceder a un atributo que poseen los arreglos que nos indican la longitud del mismo, lo accedemos colocando el identificador del arreglo seguido de **“.length”**.

```
System.out.println("Longitud del Arreglo: " + (identificadorArreglo.length));
```

```
for (int i = 0; i < identificadorArreglo.length; i++) {  
    System.out.println("Valor del Arreglo " + identificadorArreglo[i] );  
}
```

# Utilidades para los arreglos

Hay unos métodos que nos ayudan a trabajar con los arreglos en Java, para eso debemos primero que nada importar la biblioteca `java.util.Arrays` antes de la clase.

```
// paquete al que pertenece la clase
package identificadorPaquete;

// bibliotecas importadas
import java.util.Arrays;

// declaracion de la clase
public class IdentificadorClase {
```

Nombre	Descripción
<code>copyOf</code>	Copia un arreglo y lo devuelve en un nuevo arreglo.
<code>copyOfRange</code>	Copia un arreglo dado un rango y lo devuelve en un nuevo arreglo
<code>equals</code>	Indica si dos arreglos son iguales.
<code>sort</code>	Ordena el arreglo de forma natural.
<code>toString</code>	Muestra el contenido del arreglo

# Utilidades para los arreglos

```
int numerosOriginal[] = { 8, 9, 5, 12, 14, 50, 60, 16};

// Copia el arreglo numerosOriginal hasta la quinta posicion(este ultimo no incluido) devuelve un arreglo
int numeros2[] = arreglos.copyOf(numerosOriginal, 4);

// Copia el arreglo numerosOriginal de la tercera hasta la octava posicion, devuelve un arreglo
int numeros3[] = arreglos.copyOfRange(numerosOriginal, 2, 6);

// Compara si los arreglos numeros2 y numeros3 son iguales
System.out.println("Metodo equals: " + arreglos.equals(numeros2, numeros3));

// Ordena los valores del arreglo numerosOriginal
arreglos.sort(numerosOriginal);

// Muestra los valores que contiene el arreglo numerosOriginal
System.out.println(arreglos.toString(numerosOriginal));
```

# Valores por defectos

Que pasa si inicializamos el arreglo con un tamaño n y decidimos antes de asignarle valores acceder a ellos?

```
identificador2 = new tipo[n];
```

Java para evitar errores inicializa los arreglos de la siguiente manera:

1. Para números el valor cero “0” o “0.0”.
2. Para caracteres el espacio en blanco ‘ ’.
3. Para booleanos el valor false.
4. Para los Objetos “String” la ausencia de valor y se mostrara la palabra “null”, que es algo distinto de vacío.