

Arreglos unidimensionales (Vectores)

Definir una variable en forma de arreglo permite que esta pueda almacenar multiples valores de un mismo tipo de dato.

La variable **num** es de tipo entero y puede almacenar un unico valor

int num = 245;

num

245

La variable **num** es de tipo entero y puede almacenar multiple valores en forma de una estructura de una dimensión.

int[] num = {45, -67, 71, 66, -21};

num

0	1	2	3	4
45	-67	71	66	-21

1 fila con 5 columnas = 5 valores

Declaración

Forma 1:

Esta forma de declaración se utiliza solo cuando no se conoce la longitud, tamaño o cantidad de valores que va a almacenar la variable. Esta se puede realizar de dos posibles:

- a) La siguiente instrucción declara la variable **num** y le reserva espacio en memoria, por lo que es posible obtener el valor del atributo length (num.length) sin que se genere un error en tiempo de ejecución.

$$\text{int}[] \text{ num} = \{\};$$

- b) La siguiente instrucción declara la variable **num** pero no reserva espacio en memoria por lo que no es posible obtener el valor del atributo length (num.length) ya que se generará un error en tiempo de ejecución.

$$\text{int}[] \text{ num} = \text{null};$$

Forma 2:

Esta forma de declaración se utiliza cuando se conoce la longitud o tamaño (cantidad de elementos) que va a tener la variable pero no se conoce cuales son los valores.

$$\text{int}[] \text{ num} = \text{new int}[3];$$

La anterior instrucción declara la variable **num** y reserva un espacio en memoria suficiente para almacenar 3 valores de tipo int. Como no se saben cuales son los valores, por defecto se les asigna 0.

num		
0	1	2
0	0	0

Forma 3:

Esta forma de declaración se utiliza cuando se conoce cuales son los valores que va a tener la variable.

int[] num = {245, -67, 871};

La anterior instrucción declara la variable **num** y reserva un espacio en memoria suficiente para almacenar 3 valores de tipo int: **245**, **-67** y **871**.

num		
0	1	2
245	-67	871

Reservar el espacio en memoria para la variable

Cuando ya se ha declarado la variables pero no se le reservó un espacio en memoria para almacenar los valores (forma 1b) y se le quiere reservar por primera vez, o cuando se le reservó un espacio en memoria (formas 1a, 2 y 3) pero se le quiere reservar un nuevo espacio, se utiliza la siguiente instrucción:

num = new int[5];

La anterior instrucción reserva a la variable **num** un espacio en memoria suficiente para

almacenar 5 valores de tipo int.

```
num = new int[n];
```

La anterior instrucción reserva a la variable **num** un espacio en memoria suficiente para almacenar n valores de tipo int, donde n debe ser una variable con un valor entero mayor o igual a 0.

Acceder a los valores

Para acceder a los valores del arreglo unidimensional es necesario hacerlo de forma individual a través de un índice que determina la posición en la cual se encuentra almacenado. En algunos lenguajes de programación como C, C++, Java, C# y Swift los índices se numeran comenzando desde 0 pero en otros lenguajes este comportamiento puede variar.

```
int[] num = {245, -67, 871};
```

```
int x = 0, columna = 2;
```

Obtener un valor

La siguiente instrucción obtiene el segundo valor (índice o posición 1) almacenado en la variable **num** asigna a la variable **x**.

```
x = num[1];
```

La siguiente instrucción obtiene el tercer valor (índice o posición 2) almacenado en la variable **num** y lo asigna a la variable **x**.

```
x = num[columna];
```

Asignar un valor

La siguiente instrucción asigna el número **911** como primer valor (índice o posición 0) de la variable **num**.

```
num[0] = 911;
```

La siguiente instrucción asigna el número **875** como tercer valor (índice o posición 2) de la variable **num**.

```
num[columna] = 875;
```

La siguiente instrucción asigna el número **545** a cada una de las posiciones de la variable **num**.

```
for (int columna = 0; columna < num.length; ++columna){  
    num[columna] = 545;  
}
```

Obtener la longitud

Se puede obtener la capacidad, longitud o cantidad máxima de elementos que puede almacenar un arreglo unidimensional leyendo el valor de su atributo `length`.

```
int numColumns = num.length;
```