# Arreglos en JAVA

Para saber **cómo funcionan los arrays en Java** debes tener presente que un arreglo es una estructura, cuyo tamaño no se puede modificar, que permite almacenar en su interior una serie de datos del mismo tipo.

Los arrays pueden tener una o más dimensiones y, aunque tengan varios elementos, para identificarlos solo se usa el nombre que le has asignado al momento de declarar el arreglo como tal.

## ¿Cómo se declaran los arrays en Java?

La **sintaxis para declarar un array** sigue la siguiente forma general:

```
tipodedato[] nombredelarray=new tipodedato[cantidad de datos a almacenar];
```

Para comprenderlo mejor revisa esta línea de código:

```
int [] arreglo=new int[3];
```

El arreglo declarado únicamente podrá contener tres datos de tipo entero en su interior. Ahora bien, **si el array tiene más de una dimensión**, la forma de declararlo será un poco diferente:

```
tipodedato[][] nombredelarray=new tipodedato[cantidad de datos a almacenar en la 1ra dimensión][cantidad de datos a almacenar
```

Para verlo mejor:

```
int[][] arreglo = new int[3][3];
```

Este array tendrá dos dimensiones y albergará un total de nueve datos de tipo entero. Si es de más de dos dimensiones, la forma de declararlo se mantiene similar, aumentando en uno el número de corchetes usados por cada dimensión.

Cuando usas arreglos o matrices en tu código es necesario introducir los datos y llevar a cabo diversas actividades con ellos.

## ¿Cómo escribir datos en un array en Java?

Escribir datos en un arreglo en Java puede hacerse de la siguiente manera:

```
tipodedato[] nombredelarreglo = new tipodedato[extensióndelarreglo] | ;
nombredelarreglo[posiciónqueocuparáeldato] = dato;
```

Si lo vemos en un ejemplo sería algo como esto:

```
int[] arreglo = new int[3];
arreglo[0] = 6;
arreglo[] = 2;
arreglo[] = 1;
```

También podría hacerse de la siguiente manera, ambas son correctas:

```
int[] arreglo = {
    6,
    2,
    1
};
```

## ¿Cómo recorrer los elementos de un array en Java?

Cuando un array ya tiene información en su interior y necesitas revisarla o trabajar con ella, toma en cuenta estas opciones:

### Bucle for en Java

Este bucle permite que revisemos los datos que están dentro del array. Será necesario tener una constante que incremente su valor de uno en uno hasta alcanzar la longitud del array. Recuerda que **los índices del array comienzan en 0**, por lo tanto si el arreglo tiene, por ejemplo 3 posiciones, sus índices irán desde el 0 al 2.

```
public static void main(String[] arg) {
    int[] arreglo = {
        1,
        2,
        6
    };
```

```
    for (int i = 0; i < 3; i++) {
        System.out.println("El número que está en la posición" + i + "es: "+arreglo[i]);
    }
}
```

Es posible que, por diversas razones, desconozcas la cantidad de elementos que componen un array, en ese caso es útil la función que aporta ***length*** y que va a permitir recorrer el arreglo aunque no tengamos claro cuántos elementos tiene.
Su funcionamiento es como sigue:

```
public static void main(String[] args) {
    int[] arreglo = {
        1,
        3,
        5,
        2
    };
    for (int i = 0; i < arreglo.length; i++) {
        System.out.println("El número es: "+arreglo[i]);
    }
}
```

### Bucle for each en Java

Se trata de un bucle de fácil construcción y manejo que permite recorrer el array sin necesidad de usar una variable que se incrementa. Veamos un ejemplo sencillo para ilustrarlo mejor:

```
public static void main(String[] args) {
    int[] arreglo = {
        1,
        3,
        5,
        2
    };
    for (int dato: arreglo) {
        System.out.println("El número es: "+(dato));
    }
}
```

Se declara una variable local que contendrá sucesivamente cada uno de los elementos que están dentro del array y los irá imprimiendo desde el primero hasta el último.

### Algunos métodos que puedes usar con los array en Java

Los **array en Java** tienen algunos métodos que permiten trabajar sobre los datos guardados, algunos son:

- **Public static int bynarySearch(Object[] a, Object key)**

Hace una búsqueda en el array para identificar el lugar que ocupa un elemento dentro del mismo.

- **Public static void sort(Objec[] a)**

Sirve para ordenar los elementos que están dentro del array.

- **Public static boolean equals (long[] a, long[] b)**

Permite comparar dos arrays para ver si estos son iguales entre sí.
En esencia, es así **cómo funcionan los arrays en Java**, es importante tener en cuenta que cada array:

- Contiene un tipo único de elementos.
- Solo puede tener un número fijo de elementos que no se modifican.
- Cada elemento de un array ocupa un índice específico, estos se comienzan a numerar desde el 0.
- Para leer o revisar los elementos que están dentro de un array deberás usar bucles.
- La clase arrays tiene una serie de métodos que puedes usar para trabajar con su contenido.

## Bibliografía:


- Extraído de <<Como funcionan los arreglos en Java>>. Consultado el 28 de noviembre del 2022