

**TALENTO
DIGITAL**
INTELIGENCIA
HUMANA

Talento Digital para Chile:

MÓDULO 1 PROGRAMACIÓN BÁSICA EN JAVA

UN PROYECTO DE:

DESARROLLADO POR:



MÓDULO 1 - PROGRAMACIÓN BÁSICA EN JAVA

1.2 EL ENTORNO JAVA PARA LA PROGRAMACIÓN

Semana 2 - Día 10

Objetivo de la jornada

- Entender el concepto de arreglos unidimensionales en Java y utilizarlos para dar solución a un problema según requerimientos establecidos.
 - Comprender y utilizar colecciones en Java para resolver problemas planteados.
-

Arreglos

Hace algunas unidades atrás, se definió lo que era un arreglo, zona de memoria que puede almacenar un conjunto de N datos del mismo tipo. Un array en Java puede considerarse un “objeto especial” o un tipo de dato diferente, ya que se crea con la sentencia new como el resto de objetos, pero sin embargo no hay una clase específica en Java que defina el tipo de los arrays.

La declaración de un arreglo es similar a la declaración de una variable, la sintaxis es la siguiente:

tipoDato nombreArreglo[];

Algunos ejemplos serían:

- int edad[]; = Declaración de un arreglo de enteros llamado edad.
- float nota[]; = Declaración de un arreglo de reales llamado nota.
- String nombre[]; = Declaración de un arreglo de cadenas llamado nombre.

Después de declarar un arreglo, este debe crearse, para ello se usa la palabra reservada `new` y se define la cantidad de posiciones que el arreglo puede contener para el almacenamiento de valores.

Arreglo = `new TipoDato[tamaño];`

Ejemplos:

- `edad = new int [10];` Creación del arreglo `edad` con 10 posiciones.
- `nota = new float [15];` Creación del arreglo `nota` con 15 posiciones.
- `nombre = new String [5];` Creación del arreglo `nombre` con 5 posiciones.

Antes de continuar, es importante detenerse y observa la siguiente sentencia:

```
int [ ] donacionesPorMes = new int [12];
```

En ella, se logran identificar errores habituales que cometen algunos desarrolladores, es bueno tener en consideración lo siguiente:

1. Pensar que los índices van de 0 a 12. Falso: van de 0 a 11. 2. Pensar que el número de elementos total es 11. Falso: son 12. 3. Usar `donacionesPorMes[12]`. El índice 12 no existe y el uso de esa expresión daría

lugar a un error en tiempo de ejecución del tipo “`ArrayIndexOutOfBoundsException`”. 4. La primera posición de un arreglo en Java es la posición 0 y la última es $n-1$, donde

n es el tamaño del arreglo que en este caso es $12-1 = 11$.

En Java también es posible declarar y crear un objeto en la misma línea de código, ejemplo:

- `int edad [] = new int [10];` = Declaración y creación de un arreglo llamado `edad` de 10 posiciones de tipo entero.
- `float nota [] = new float [15];` = Declaración y creación de un arreglo llamado `nota` de 15 posiciones de tipo `float`.
- `String nombre [] = new String [5];` = Declaración y creación de un arreglo llamado `nombre` de 5 posiciones de tipo `String`.

El siguiente ejemplo muestra la declaración, creación, inicialización y consulta de un arreglo llamado edad de 3 posiciones de tipo entero:

```
public class Programa2{
    public static void main(String arg[]){
        int edad[]= new int[3];
        edad[0]=18; edad[1]=20;
        edad[2]=15; for(int
        i=0;i<3;i++){
```

System.out.println("El valor del arreglo edad en la posicion "+i+" es "+edad[i]); } } } También es posible inicializar un arreglo en la misma línea donde es declarado y creado. El siguiente ejemplo crea e inicializa en la misma línea de código un arreglo de 3 posiciones de tipo cadena con los nombres de 3 personas diferentes, posteriormente se imprime el contenido del arreglo:

```
public class Programa3{
    public static void main(String arg[]){
        String nombre[] = ("Juan", "Pedro", "Diana"); for
        (int i = 0; i < 4; i++){
        System.out.println(nombre[i]); } }
```

} Si se desea saber el largo del arreglo, basta con usar la sintaxis nombreDelArray.length, la cual devuelve un entero (int) con el número de elementos que forman el array.

```
System.out.println ("El número de elementos en el array misNombres es de " +
misNombres.length );
```

Es posible usar ciclos for each con arreglos, lo cual tiene como beneficio que nunca saldrá de los límites del arreglo y que no es necesario conocer el tamaño del arreglo. Por ejemplo:

```
for (int tmpItem : cochesPorHora) {
    System.out.println ("Número: " + tmpItem); }

String [ ] misNombres = new String [10];

for (String tmpObjeto : misNombres) {
    System.out.println (tmpObjeto); }
```

Colecciones

Los arreglos son una excelente herramienta de programación, principalmente cuando la cantidad de posiciones del mismo se mantiene constante, sin embargo, al momento de incluir más elementos dentro de un arreglo o al momento de querer eliminar posiciones del mismo, los arreglos son ineficientes. En Java existen otras estructuras de almacenamiento conocidas como colecciones. Una colección permite almacenar información en memoria de manera dinámica, es decir, permite el aumento o disminución del tamaño de la colección de acuerdo con la necesidad de la aplicación.

Elemento Descripción

Collection Define un conjunto de métodos comunes a las demás interfaces o clases que se derivan de la interfaz Collection.

List Maneja de manera ordenada los elementos introducidos de acuerdo con el orden de inserción. Permite elementos duplicados.

Set Colección desordenada de objetos. No permite elementos duplicados.

ArrayList Clase concreta que implementa la interfaz List. Esta clase se basa en un arreglo dinámico. Permite un acceso rápido a los elementos de la colección pero es lenta para la inserción o eliminación de objetos.

LinkedList Clase concreta que implementa la interfaz List, esta clase se basa en una lista enlazada. Es muy útil cuando la cantidad de elementos de la colección es muy variable. El acceso a un elemento particular de la colección es más lento

HashSet Clase concreta que implementa la interfaz Set. Los elementos de esta clase no permiten duplicados y no se almacenan en orden no determinado

TreeSet Clase concreta que implementa la interfaz Set. Los elementos de esta clase no permiten duplicados y sus elementos se ordenan de manera ascendente

Uso de la clase ArrayList

La clase ArrayList en Java, es una clase que permite almacenar datos en memoria de forma similar a los Arrays pero con la ventaja de que el número de elementos que almacena, lo hace de forma dinámica, es decir, que no es necesario declarar su tamaño como pasa con los Arrays. Para comprender mejor el uso de la clase ArrayList, se presenta el siguiente ejemplo:

```
import java.util.ArrayList;

public class Ejemplo {
    public static void main(String arg[]) {
        ArrayList listaNums= new ArrayList();
        listaNums.add(8); listaNums.add(3);
        listaNums.add(5);

        System.out.println("Lista de numeros: " + listaNums);
        System.out.println("Numero posicion 2:" + listaNums.get(2));

        listaNums.set(0,15); System.out.println("Lista de numeros:
"+listaNums); } }
```

Recursos Bibliográficos: <http://puntocomnoesunlenguaje.blogspot.com/2012/07/arreglos-en-java-calcular-media.html>