



Recibe una cálida:

¡Bienvenida!

Te estábamos esperando 😊 

➤ El Entorno Java para la programación

Plan formativo: Desarrollo de Aplicaciones Full Stack Java Trainee V2.0

HOJA DE RUTA

¿Cuáles **skill** conforman el programa?



REPASO CLASE ANTERIOR

En la clase anterior trabajamos :

- ✓ *Bucle while*
- ✓ *Bucle do-while*
- ✓ *Bucle for*

LEARNING PATHWAY

4.3

Start! 🏁

El Entorno Java para
la programación

Arreglos e introducción a
Colecciones

Arreglos en Java

Arreglando

OBJETIVOS DE APRENDIZAJE

¿Qué aprenderemos?



Conocer y manipular arreglos en Java



Comprender el concepto de las colecciones





Rompehielo 🥶



En fila: 🙌

Veamos la siguiente imagen, y en base a ella repondan en el chat o levantando la mano:

Consigna: ✍️

1. ¿Cuántas personas hay en la fila?
2. ¿Cuántas posiciones hay en la fila?
3. ¿Quién está en la posición 3 de la fila?
4. ¿Qué posición ocupa la persona de camisa azul en la fila?



➤ Arreglos en Java



Arreglos

¿Qué es un arreglo?:



Un arreglo, también conocido como **array**, es una estructura de datos que puede **almacenar una colección de elementos del mismo tipo**. En Java, los arrays tienen un **tamaño fijo** que se define en el momento de su creación. Cada elemento del array se puede acceder mediante un **índice numérico** que representa su posición en el array.



Para declarar un array en Java, debes especificar su tipo y su tamaño:

```
int[] numeros = new int[4];
```

Los corchetes indican que esta variable es del tipo arreglo y el prefijo int informa que almacenará números enteros. Además, especificamos su tamaño en 4 espacios.





Arreglos



Índices de un arreglo:



```
int[] numeros = new int[4];
```

Este array está vacío, es como una variable normal, pero con cuatro espacios. Para rellenarlos, hay que asignar datos a cada parte del contenedor.



Para asignar valores, tienes que utilizar el **índice** de los arrays. Se cuenta **desde la posición 0** hasta la que tenga el array. En este caso, tiene cuatro espacios y, por lo tanto, el índice va del 0 al 3 (4 valores en total).



```
int[] numeros = new int[4];  
numeros[0] = 10;  
numeros[1] = 20;  
numeros[2] = 30;  
numeros[3] = 40;
```



Arreglos

Acceder a una posición del arreglo:



Después de almacenar valores, llega el momento de utilizarlos. Para usar uno de estos cuatro valores, lo haremos llamando al array y la posición que queramos:

```
7      int[] numeros = new int[4];  
8      numeros[0] = 10;  
9      numeros[1] = 20;  
10     numeros[2] = 30;  
11     numeros[3] = 40;  
12  
13     System.out.print("El valor de la primera posición del array es: " + numeros[0] + ".");  
14
```

Console X

<terminated> main [Java Application] C:\Users\marie\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-1502\jre\bin\javaw.exe (15 a
El valor de la primera posición del array es: 10.



Arreglos



Declarar e inicializar array en Java:



También puedes inicializar un array con valores específicos en el momento de su creación. Por ejemplo:

```
int[] numeros = {10, 20, 30, 40};
```



De esta forma, no hay que indicar la longitud, ya que los propios elementos la especifican. **Si hay cuatro elementos, el array toma una longitud de índice de 4.**





Arreglos



Acceder con bucle for a todos los elementos de un arreglo:



Se puede automatizar el acceso a todas las partes de un array con un bucle. En esta ocasión, vamos a utilizar el bucle for, que es el más usado para recorrer arreglos.

Truco: la función **.length** devuelve el tamaño del arreglo.



```
7      int[] numeros = {10, 20, 30, 40};
8
9      for (int i = 0; i < numeros.length; i++) {
10         System.out.println("El valor es: " + numeros[i] + ".");
11     }
12
13
14
```

<

Console X

<terminated> main [Java Application] C:\Users\marie\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64

El valor es: 10.
El valor es: 20.
El valor es: 30.
El valor es: 40.





Arreglos



El bucle for-each de Java

Desde hace bastante tiempo se añadió una herramienta muy especial para este tipo de tareas tan frecuentes. Se trata del bucle **for-each o enhanced loop**.

El resultado es el mismo, y Java lo entiende correctamente. Solo hay que darle un **nombre al iterador, dos puntos y el nombre del array que se quiere iterar** de arriba a abajo. No hay que usar ni length ni incrementos. Es más cómodo, limpio y fácil de escribir.



```
7      int[] numeros = {10, 20, 30, 40};
8
9      for (int i : numeros) {
10         System.out.println("El valor es: " + i + ".");
11     }
12
13
14
```

< Console X

<terminated> main [Java Application] C:\Users\marie\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre:

El valor es: 10.
El valor es: 20.
El valor es: 30.
El valor es: 40.



Arreglos bidimensionales



Matrices:

Los arreglos bidimensionales o matrices tienen el mismo tratamiento que los arrays. La diferencia de sintaxis se centra en la creación:



Declaración y creación de una Matriz

```
int[][] arregloM = new int [Filas][Columnas];
```

De esta manera podemos crear dos dimensiones diferentes para el mismo arreglo, es por esto que lleva dos corchetes.





Arreglos

Acceder con dos bucle for a todos los elementos de una matriz:

Para recorrer una matriz, es necesario que ingresemos tanto a filas como a columnas. Para esto vamos a utilizar un bucle for anidado por cada dimensión de la matriz.



```
7      int[][] matriz = new int [2][2];
8
9      matriz[0][0] = 10;
10     matriz[0][1] = 20;
11     matriz[1][0] = 30;
12     matriz[1][1] = 40;
13
14
15     for (int i = 0; i < matriz.length; i++) {
16         for (int j = 0; j < matriz.length; j++) {
17             System.out.println("fila " + i + " columna " + j + " tiene el valor: " + matriz[i][j]);
18         }
19     }
```

Console X

<terminated> main [Java Application] C:\Users\marie\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-1502\jre\bin\javaw.exe (15 ago. 2023)

```
fila 0 columna 0 tiene el valor: 10
fila 0 columna 1 tiene el valor: 20
fila 1 columna 0 tiene el valor: 30
fila 1 columna 1 tiene el valor: 40
```



➤ Introducción a Colecciones



Colecciones



¿Qué son las colecciones?:

Una colección **representa un grupo de objetos**, conocidos como elementos. Podemos crear una colección con cualquier tipo de objeto, incluso los creados por nosotros. La única restricción es que **no podremos crear una colección sólo de tipo primitivo**, para eso usaremos los tipos de objetos equivalentes a los primitivos. Por ejemplo: **en vez de int, hay que utilizar Integer**. A continuación veremos los objetos equivalentes a los primitivos de Java.



Tipos de datos	
Primitivos	Objetos
int	Integer
double	Double
long	Long
char	Character
boolean	Boolean
String ya es un objeto, por lo que no tiene tipo primitivo	

Arreglos

Arreglos:

✕ **Tamaño fijo:** Los arreglos tienen un tamaño fijo que se define al crearlos.

Tipo de datos único: Los arreglos almacenan elementos de un único tipo de dato.

Acceso por índice: En los arreglos se accede a elementos por su índice numérico.

Almacenamiento continuo: Los arreglos son continuos en memoria,

Colecciones

Colecciones:

Tamaño variable: Las colecciones se expanden conforme se agregan elementos.

Múltiples tipos: Las colecciones pueden almacenar objetos de diferentes tipos.

✕ **Acceso no por índice:** En las colecciones se accede por iteración o claves.

○ **Almacenamiento no continuo:** Las colecciones pueden fragmentarse.



LIVE CODING

Ejemplo en vivo

Recorriendo Arrays:

Vamos a ver un ejemplo de manipulación de cadenas en arreglos. Esto es una práctica útil para guardar datos como por ejemplo contraseñas

- 1. Declarar y recorrer un arreglo que almacene el nombre de 5 países ingresados por el usuario. Mostrar el contenido del arreglo por pantalla.*

 **Tiempo: 20 minutos**





Ejercicio N° 1

Arreglando



Arreglando

Pongamos en práctica: 🙌

Los arreglos son estructuras que nos permiten almacenar varios datos con un tipo en común. Esto sirve para gestionar una lista de supermercado o para organizar a los jugadores titulares de tu equipo favorito. Son ampliamente utilizados y es por esto que es necesario ponerlo en práctica.

Consigna: 📝

Crear un arreglo para almacenar 3 números reales ingresados por el usuario y mostrar el mayor elemento.

Tiempo 🕒: 25 minutos

Paso a paso: ⚙️

1. declara el arreglo con tamaño 3
2. recorrer el arreglo con un bucle for
3. pedir el numero al usuario dentro del bucle
4. asignar el valor a cada elemento utilizando el iterador i
5. luego de almacenar, recorre el arreglo con otro bucle for dónde validarás cuál es el mayor (condicionales)



○

¿Alguna consulta?

+



RESUMEN

¿Qué logramos en esta clase?

- ✓ **Comprender la sintaxis e implementación de los arreglos en Java**
- ✓ **Aprender el concepto de colecciones y las principales diferencias que tienen con los arreglos**



#WorkingTime

Continuemos ejercitando

¡Antes de cerrar la clase! Te invitamos a: 🙌 🙌 🙌

1. Repasar nuevamente la grabación de esta clase
2. Revisar el material compartido en la plataforma de Moodle (lo que se vio en clase y algún ejercicio adicional)
 - a. Material 1 (Foro)
 - b. *Lectura Módulo 4, Lección 3: páginas 11 - 12*
3. Traer al próximo encuentro, todas tus dudas y consultas para verlas antes de iniciar nuevo tema.

¡Muchas Gracias!

Nos vemos en la próxima clase 🙌



Momento: ✚

Time-out!

🕒 5 min.

