Investigación: Arreglos en Java

1. ¿Cómo se declara un arreglo en Java?

Un arreglo o array es una estructura que permite guardar varios valores del mismo tipo en una sola variable.

Es como una caja con varios espacios donde se guardan datos.

Hay varias formas de declarar e inicializar como: Declaración usando valores calculados, Declarar e inicializar usando new con valores. Pero describiré solamente 2 formas.

Forma 1: Declarar el arreglo y definir el tamaño

// Se declara un arreglo de tipo entero con 5 espacios.

int[] numeros = new int[5];

// Se asignan valores manualmente.

numeros[0] = 10;

numeros[1] = 20;

numeros[2] = 30;

numeros[3] = 40;

numeros[4] = 50;

Explicación:

- int[] → tipo de datos.
- números → nombre del arreglo.
- new int[5] → crea el arreglo con 5 espacios.

Forma 2: Declarar e inicializar directamente

// Se crea el arreglo con valores desde el inicio.

int[] valores = {1, 2, 3, 4, 5};

Explicación:

- Java crea automáticamente el arreglo.
- Ya contiene los valores indicados.

2. Métodos principales de la clase Arrays

La clase Arrays ayuda a trabajar con arreglos más fácilmente.

Primero se importa: import java.util.Arrays;

a) Arrays.sort()

Ordena los valores del arreglo.

```
int[] numeros = {5, 2, 8, 1, 3};
Arrays.sort(numeros);
System.out.println(Arrays.toString(numeros));
```

Resultado:

[1, 2, 3, 5, 8]

b) Arrays.binarySearch()

Busca un valor dentro del arreglo.

```
int[] numeros = {1, 2, 3, 4, 5};
int posicion = Arrays.binarySearch(numeros, 3);
System.out.println("Posición: " + posicion);
```

Resultado:

Posición: 2

c) Arrays.copyOf()

Copia un arreglo.

```
int[] original = {10, 20, 30};
int[] copia = Arrays.copyOf(original, 3);
System.out.println(Arrays.toString(copia));
```

Resultado:

[10, 20, 30]

d) Arrays.fill()

Llena el arreglo con un valor.

```
int[] numeros = new int[5];
Arrays.fill(numeros, 100);
```

System.out.println(Arrays.toString(numeros));

Resultado:

[100, 100, 100, 100, 100]

e) Arrays.equals()

Compara dos arreglos.

int[] a = {1, 2, 3};

int[] b = {1, 2, 3};

boolean iguales = Arrays.equals(a, b);

System.out.println(iguales);

Resultado:

true

3. Método extra útil: Arrays.toString()

Muestra el arreglo como texto.

int[] numeros = {1, 2, 3};

System.out.println(Arrays.toString(numeros));

Resultado:

[1, 2, 3]

4. ¿Cómo se recorren los arreglos en Java?

Los arreglos en Java se recorren para poder ver o usar todos sus valores. La forma más común es usando un ciclo for, que va posición por posición. También se puede usar for-each, que es más simple y automático.

Recorrer significa ver todos los valores del arreglo.

Forma 1: For tradicional

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

for (int i = 0; i < numeros.length; i++) {

   System.out.println(numeros[i]);  }
```

Explicación:

- i es el índice
- length es el tamaño del arreglo

Forma 2: For-each

Es más simple.

```
int[] numeros = {10, 20, 30};
for (int numero : numeros) {
   System.out.println(numero);
}
```

Explicación:

- No usa índice
- Recorre automáticamente

Forma 3: Usando Streams

Forma más moderna.

```
import java.util.Arrays;
int[] numeros = {10, 20, 30};
Arrays.stream(numeros).forEach(System.out::println);
```

5. Diferencias entre Array y ArrayList

1. Tamaño

Array:

- Tamaño fijo
- No se puede cambiar

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

ArrayList:

- Tamaño dinámico
- Puede crecer

```
import java.util.ArrayList;
```

```
ArrayList<Integer> lista = new ArrayList<>();
lista.add(10);
lista.add(20);
```

6. Tipos de datos

Array:

- Usa tipos primitivos

```
int[] numeros = {1, 2, 3};
```

ArrayList:

- Usa clases envolventes

```
ArrayList<Integer> lista = new ArrayList<>();
```

7. Métodos disponibles

Array:

- No tiene muchos métodos

ArrayList:

- Tiene métodos útiles

```
lista.add(10);
lista.remove(0);
lista.size();
```

8. Rendimiento

Array:

- Más rápido
- Usa menos memoria

ArrayList:

- Un poco más lento
- Usa más memoria

9. ¿Cuándo usar cada uno?

Usar Array cuando:

- El tamaño es fijo
- Se necesita más velocidad

Usar ArrayList cuando:

- El tamaño cambia
- Se agregan o eliminan datos