



# Code, Learn & Practice(Programación: Introducción a la Programación "Repaso de array´s y String en Java")

---

## Ejercicio 1: Suma de Elementos en un Array

- **Descripción:** Crea un método que tome un array de enteros y devuelva la suma de todos sus elementos.
- **Ejemplo:** `sumArray([1, 2, 3, 4]) -> 10`

### ► SOLUCIÓN:

```
public class Exercise1 {  
    public static void main(String[] args) {  
        int[] array = {1, 2, 3, 4};  
        System.out.println("Suma del array: " + sumArray(array));  
    }  
  
    public static int sumArray(int[] array) {  
        int sum = 0;  
        for (int num : array) {  
            sum += num;  
        }  
        return sum;  
    }  
}
```

## Ejercicio 2: Valor Mínimo y Máximo en un Array

- **Descripción:** Crea un método que tome un array de enteros y devuelva el valor mínimo y máximo en el array.
- **Ejemplo:** `minMaxArray([3, 7, 2, 8]) -> Min: 2, Max: 8`

### ► SOLUCIÓN:

```
public class Exercise2 {  
    public static void main(String[] args) {  
        int[] array = {3, 7, 2, 8};  
        int[] minMax = minMaxArray(array);  
        System.out.println("Valor mínimo: " + minMax[0] + ", Valor máximo:  
" + minMax[1]);  
    }  
  
    public static int[] minMaxArray(int[] array) {  
        int min = Integer.MAX_VALUE;
```

```
        int max = Integer.MIN_VALUE;
        for (int num : array) {
            if (num < min) min = num;
            if (num > max) max = num;
        }
        return new int[]{min, max};
    }
}
```

### Ejercicio 3: Media Aritmética

- **Descripción:** Implementa un método que calcule el promedio de los valores en un array de enteros.
- **Ejemplo:** `averageArray([4, 8, 10])` -> 7.33

► SOLUCIÓN:

```
public class Exercise3 {
    public static void main(String[] args) {
        int[] array = {4, 8, 10};
        System.out.println("Media del array: " + averageArray(array));
    }

    public static double averageArray(int[] array) {
        int sum = 0;
        for (int num : array) {
            sum += num;
        }
        return (double) sum / array.length;
    }
}
```

### Ejercicio 4: Reverso de un Array

- **Descripción:** Escribe un método que reciba un array de enteros y devuelva otro array con los elementos en orden inverso.
- **Ejemplo:** `reverseArray([1, 2, 3, 4])` -> [4, 3, 2, 1]

► SOLUCIÓN:

```
import java.util.Arrays;

public class Exercise4 {
    public static void main(String[] args) {
        int[] array = {1, 2, 3, 4};
        int[] reversed = reverseArray(array);
        System.out.println("Array invertido: " +
            Arrays.toString(reversed));
    }
}
```

```
public static int[] reverseArray(int[] array) {
    int[] reversed = new int[array.length];
    for (int i = 0; i < array.length; i++) {
        reversed[i] = array[array.length - 1 - i];
    }
    return reversed;
}
```

## Ejercicio 5: Búsqueda de un Elemento

- **Descripción:** Implementa un método que busque si un número específico está presente en un array.
- **Ejemplo:** `findInArray([1, 2, 3, 4], 3) -> true`

► SOLUCIÓN:

```
public class Exercise5 {
    public static void main(String[] args) {
        int[] array = {1, 2, 3, 4};
        int target = 3;
        System.out.println("¿Contiene el número " + target + "? " +
            findInArray(array, target));
    }

    public static boolean findInArray(int[] array, int target) {
        for (int num : array) {
            if (num == target) return true;
        }
        return false;
    }
}
```

## Ejercicio 6: Duplicados en un Array

- **Descripción:** Crea un método que tome un array y devuelva `true` si hay duplicados en el array, `false` en caso contrario.
- **Ejemplo:** `hasDuplicates([1, 2, 3, 2]) -> true`

► SOLUCIÓN:

```
import java.util.HashSet;

public class Exercise6 {
    public static void main(String[] args) {
        int[] array = {1, 2, 3, 2};
        System.out.println("¿Contiene duplicados? " +
            hasDuplicates(array));
    }
}
```

```
public static boolean hasDuplicates(int[] array) {
    HashSet<Integer> seen = new HashSet<>();
    for (int num : array) {
        if (!seen.add(num)) return true;
    }
    return false;
}
```

## Ejercicio 7: Ordenar un Array

- **Descripción:** Escribe un método que ordene un array de enteros de menor a mayor.
- **Ejemplo:** `sortArray([3, 1, 4, 2]) -> [1, 2, 3, 4]`

► SOLUCIÓN:

```
import java.util.Arrays;

public class Exercise7 {
    public static void main(String[] args) {
        int[] array = {3, 1, 4, 2};
        int[] sorted = sortArray(array);
        System.out.println("Array ordenado: " + Arrays.toString(sorted));
    }

    public static int[] sortArray(int[] array) {
        int[] sorted = Arrays.copyOf(array, array.length);
        Arrays.sort(sorted);
        return sorted;
    }
}
```

## Ejercicio 8: Concatenar Arrays

- **Descripción:** Implementa un método que tome dos arrays y devuelva un array resultante que sea la concatenación de los dos.
- **Ejemplo:** `concatArrays([1, 2], [3, 4]) -> [1, 2, 3, 4]`

► SOLUCIÓN:

## Ejercicio 9: Frecuencia de Elementos

- **Descripción:** Crea un método que calcule la frecuencia de cada número en un array.
- **Ejemplo:** `frequencyArray([1, 2, 2, 3]) -> {1: 1, 2: 2, 3: 1}`

► SOLUCIÓN:

## Ejercicio 10: Eliminar Duplicados

- **Descripción:** Escribe un método que tome un array y devuelva un nuevo array sin duplicados.
- **Ejemplo:** `removeDuplicates([1, 2, 2, 3])` -> `[1, 2, 3]`

► SOLUCIÓN:

## Ejercicio 11: Contar Vocales y Consonantes

- **Descripción:** Crea un método que reciba una cadena de texto y cuente cuántas vocales y consonantes contiene.
- **Ejemplo:** `countVowelsConsonants("hello")` -> `Vocales: 2, Consonantes: 3`

► SOLUCIÓN:

## Ejercicio 12: Palabra Más Larga

- **Descripción:** Escribe un método que encuentre la palabra más larga en una oración.
- **Ejemplo:** `longestWord("El perro come")` -> `perro`

► SOLUCIÓN:

## Ejercicio 13: Invertir una Cadena

- **Descripción:** Crea un método que invierta el contenido de una cadena.
- **Ejemplo:** `reverseString("hola")` -> `aloh`

► SOLUCIÓN:

## Ejercicio 14: Palíndromo

- **Descripción:** Implementa un método que determine si una cadena es un palíndromo (se lee igual al revés).
- **Ejemplo:** `isPalindrome("radar")` -> `true`

► SOLUCIÓN:

## Ejercicio 15: Contar Palabras en una Cadena

- **Descripción:** Crea un método que cuente el número de palabras en una cadena.
- **Ejemplo:** `countWords("El gato duerme")` -> `3`

► SOLUCIÓN:

## Ejercicio 16: Reemplazar Caracteres

- **Descripción:** Escribe un método que reemplace todas las ocurrencias de un carácter en una cadena con otro carácter.
- **Ejemplo:** `replaceChar("perro", 'r', 'l')` -> `pello`

► SOLUCIÓN:

## Ejercicio 17: Primera Letra en Mayúsculas

- **Descripción:** Crea un método que reciba una oración y devuelva la misma oración con la primera letra de cada palabra en mayúsculas.
- **Ejemplo:** `capitalizeWords("buenos dias")` -> `Buenos Dias`

► SOLUCIÓN:

## Ejercicio 18: Subcadena Común más Larga

- **Descripción:** Implementa un método que encuentre la subcadena más larga que dos cadenas tengan en común.

- **Ejemplo:** `longestCommonSubstring("abcdef", "zcdemf") -> cde`

► SOLUCIÓN:

## Ejercicio 19: Frecuencia de Palabras en una Cadena

- **Descripción:** Crea un método que reciba una cadena y devuelva la frecuencia de cada palabra en un mapa.
- **Ejemplo:** `wordFrequency("el gato el") -> {el: 2, gato: 1}`

► SOLUCIÓN:

## Ejercicio 20: Comparación de Anagramas

- **Descripción:** Escribe un método que determine si dos cadenas son anagramas (tienen las mismas letras en diferente orden).
- **Ejemplo:** `areAnagrams("amor", "roma") -> true`

► SOLUCIÓN:

## Licencia

Este proyecto está bajo la Licencia (Apache 2.0) - mira el archivo [LICENSE.md](#) para detalles.