

CORPORACIÓN UNIVERSITARIA AUTÓNOMA DEL CAUCA



PROGRAMACION II

ENTREGABLE EJERCICIOS CLASE 2

PRESENTADO POR:

SALOMÓN MONTILLA LUNA

PRESENTADO A:

CRISTHIAN ALEJANDRO CAÑAR MUÑOZ

POPAYÁN, CAUCA

14 DE FEBRERO DEL 2025

CAPTURAS DE PANTALLA

Ejercicio no. 1: Defina una función llamada **listaPotencia** que reciba un parámetro de tipo número entero (int) y una lista (list), y devuelva una nueva lista con todos los elementos de la lista elevados a la potencia del número que se pasó por parámetro.

```
//Ejercicio 1
public static List<Integer> listaPotencia(int exponente, List<Integer> numeros){ 1 usage  salo
    List<Integer> nuevaLista = new ArrayList<>();
    for (int i = 0; i < numeros.size(); i++) {
        nuevaLista.add(i, (int) Math.pow(numeros.get(i), exponente));
    }
    return nuevaLista;
}
```

Ejecución:

```
Ejercicio 1: -----
Lista original: [2, 3, 4]
Resultado de la potencia: [4, 9, 16]
```

Ejercicio no. 2: Defina una función llamada **listasMultiplicacion** que reciba por parámetro dos listas (list) y devuelva una nueva lista con la multiplicación elemento a elemento de ambas de listasFunción:

```
//Ejercicio 2
public static List<Integer> listasMultiplicacion(List<Integer> lista1, List<Integer> lista2) { 1 usage  salo
    List<Integer> resultado = new ArrayList<>();

    if (lista1.size() != lista2.size()) {
        System.out.println("Las listas deben tener el mismo tamaño!");
        return null;
    }

    for (int i = 0; i < lista1.size(); i++) {
        resultado.add(lista1.get(i) * lista2.get(i));
    }

    return resultado;
}
```

Ejecución:

```
Ejercicio 2: -----  
Lista 1: [2, 4, 6]  
Lista 2: [3, 5, 7]  
Resultado de la multiplicación: [6, 20, 42]
```

Ejercicio no. 3: Defina una función llamada **calcular Promedio** que reciba por parámetro una lista y devuelva el promedio de todos los elementos de la lista.

```
//ejercicio 3  
public static double calcularPromedio(List<Integer> lista) { 1 usage salo  
    int suma = 0;  
    for (int num : lista) {  
        suma += num;  
    }  
  
    return (double) suma / lista.size();  
}
```

Ejecución:

```
Ejercicio 3: -----  
Lista original: [2, 4, 6, 8, 10]  
Resultado del promedio: 6.0
```

Ejercicio no. 4: Defina una función llamada **invertirLista** que reciba por parámetro una lista y devuelva una nueva lista con los elementos en orden inverso.

```
//ejercicio 4  
public static List<Integer> invertirLista(List<Integer> lista) { 1 usage salo  
    List<Integer> listaInvertida = new ArrayList<>(lista);  
    for (int i = 0; i < lista.size(); i++) {  
        listaInvertida.set(i, lista.get(lista.size() - 1 - i));  
    }  
    return listaInvertida;  
}
```

Ejecución: (Al final del documento)

Ejercicio no. 5: Defina una función llamada **eliminar Duplicados** que reciba por parámetro una lista y devuelva una nueva lista con los elementos únicos de la lista original, sin duplicados.

```
//ejercicio 5

public static List<Integer> eliminarDuplicados(List<Integer> lista) { 1 usage  🧑 salo
    List<Integer> listaSinDuplicados = new ArrayList<>();

    for (int num : lista) {
        if (!listaSinDuplicados.contains(num)) {
            listaSinDuplicados.add(num);
        }
    }

    return listaSinDuplicados;
}
```

Ejecución: (Al final del documento)

Ejercicio no. 6: Defina una función llamada **estaOrdenada** que reciba por parámetro una lista de números enteros y devuelva **'true'** si la lista está ordenada de manera ascendente, y **'false'** en caso contrario.

```
//ejercicio 6

public static boolean estaOrdenada(List<Integer> lista) { 1 usage  🧑 salo
    for (int i = 0; i < lista.size() - 1; i++) {
        if (lista.get(i) > lista.get(i + 1)) {
            return false;
        }
    }
    return true;
}
```

Ejecución: (Al final del documento)

Ejercicio no. 7: Defina una función llamada **combinarListas** que reciba por parámetros dos listas de números enteros y devuelva una nueva lista combinando ambas listas.

```
//ejercicio 7
private static List<Integer> combinarListas(List<Integer> lista, List<Integer> lista2) { 1 usage salo
    List<Integer> listaCombinada = new ArrayList<>();
    for (int num : lista) {
        listaCombinada.add(num);
    }

    for (int num : lista2) {
        listaCombinada.add(num);
    }
    return listaCombinada;
}
```

Ejecución: (Al final del documento)

Ejecución ejercicios desde el 4 hasta el 7:

Listas originales:

```
Lista 1 original: [3, 1, 4, 1, 5, 9, 2]
Lista 2 original: [10, 20, 30]
```

```
Ejercicio 4: -----
Lista invertida: [2, 9, 5, 1, 4, 1, 3]
Ejercicio 5: -----
Lista sin duplicados: [3, 1, 4, 5, 9, 2]
Ejercicio 6: -----
¿Lista ordenada? false
Ejercicio 7: -----
Lista combinada: [3, 1, 4, 1, 5, 9, 2, 10, 20, 30]
```

Enlace al repositorio: https://github.com/salomontillaUni/ejercicios_clase_2