

# Ejercicios Java 8-11-17

## Parte 1: Optionals

### Ejercicio 1.1

Crea una función que reciba un Optional y devuelva la longitud del string si está presente, o 0 si está vacío.

```
public static int getStringLength(Optional<String> input) {  
    // tu código aquí  
}
```

### Ejercicio 1.2

Crea una función que reciba un String y devuelva un Optional que contenga la conversión a número, o un Optional.empty() si no es un número válido.

```
public static Optional<Integer> parseInteger(String input) {  
    // tu código aquí  
}
```

## Parte 2: Streams

### Ejercicio 2.1

Dada una lista de nombres:

```
List<String> names = List.of("Ana", "Luis", "Pedro", "María", "Lucía");
```

Filtra los nombres que tienen más de 4 letras y conviértelos a mayúsculas usando Streams.

### Ejercicio 2.2

A partir de una lista de enteros:

```
List<Integer> numbers = List.of(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
```

- Filtra los pares.
- Eleva cada número al cuadrado.
- Suma todos los resultados.

## Parte 3: Funciones Lambda

### Ejercicio 3.1

Crea una Comparator usando una lambda que ordene las palabras por su longitud de

menor a mayor.

### Ejercicio 3.2

Escribe una función que acepte dos parámetros:

- Una lista de enteros
- Un predicado (`Predicate<Integer>`)

Y devuelva una lista de los elementos que cumplen el predicado.

## Parte 4: Métodos Referenciados

### Ejercicio 4.1

Convierte esta lambda en un método referenciado:

```
list.forEach(s -> System.out.println(s));
```

### Ejercicio 4.2

Usa un método referenciado para ordenar una lista de strings ignorando mayúsculas/minúsculas.

```
List<String> words = List.of("Apple", "banana", "apricot", "Blueberry");
```

## Parte 5: Strings Multilínea

### Ejercicio 5.1

Crea un String multilínea que contenga un poema de 4 versos.

### Ejercicio 5.2

Dado este texto multilínea, cuenta cuántas líneas tiene:

```
String text = ""  
    Uno  
    Dos  
    Tres  
    Cuatro  
    "";
```

## Parte 6: Pattern Matching (instanceof)

### Ejercicio 6.1

Completa este método usando Pattern Matching para evitar el casting manual:

```
public static void printStringLength(Object obj) {  
    // tu código aquí  
}
```

## Ejercicio 6.2

Completa esta función para sumar dos objetos si ambos son Integer, o imprimir un error si no:

```
public static void addIfIntegers(Object a, Object b) {  
    // tu código aquí  
}
```

# Parte 7: Ejercicios Extra

## Ejercicio 7.1 – Filtrado y Agrupado

Tienes una lista de personas:

```
record Person(String name, int age) {}  
  
List<Person> people = List.of(  
    new Person("Ana", 23),  
    new Person("Luis", 31),  
    new Person("Pedro", 23),  
    new Person("Lucía", 28),  
    new Person("Marta", 31)  
);
```

Agrupar las personas por edad usando `Streams (Map<Integer, List<Person>>)`.

Después, para cada edad, muestra los nombres separados por comas.

Ejemplo de salida:

```
Edad: 23 -> Ana, Pedro  
Edad: 28 -> Lucía  
Edad: 31 -> Luis, Marta
```

## Ejercicio 7.2 – Optional avanzado

Implementa una función que:

- Reciba una lista de Optional.
- Devuelva la **media** (promedio) de los valores presentes.
- Si no hay valores, que devuelva Optional.empty().

```
public static Optional<Double> averageOptionals(List<Optional<Integer>>  
optionals) {  
    // tu código aquí  
}
```

## Ejercicio 7.3 – Stream + Pattern Matching + String multilínea

Dada una lista de Object que puede contener:

- String
- Integer
- Otros objetos (por ejemplo: List, LocalDate, Hashmap, ...)

Imprime un resumen multilínea como este:

```
Found a String: "Hola mundo"  
Found an Integer: 42  
Unknown type: class java.time.LocalDate
```

Utiliza

- ``instanceof`` con pattern matching
- String multilínea para el resumen.

## Ejercicio 7.4 – Map-Reduce Complejo

Tienes una lista de frases:

```
List<String> phrases = List.of(  
    "Hola mundo",  
    "Aprendiendo Java",  
    "Streams son poderosos",  
    "Lambdas y Optionals"  
);
```

Queremos calcular:

- Cuántas palabras totales hay.
- Cuántas palabras únicas hay.

Reglas:

- Las palabras se separan por espacio.
- No distinguir mayúsculas/minúsculas ("Hola" = "hola").