



**Taller de Programación Web**

Ejercicios Complementarios



## Criterios de Calificación:

- Se deberá subir el link del repo de actividades anteriores. Solo se agregará una carpeta con el nombre "EjerciciosComplementariosLevel3".
- Los Ejercicios deben tener nombre según la numeración de las consignas. Ejemplo: Ejercicio1.java, Ejercicio2.java, etc.
- Convenciones de nombre para variables (camelCase), Clases (UpperCamelCase), funciones (camelCase).
- Prolijidad y fácil lectura. No deberán existir saltos de líneas consecutivos.
- Entregas fuera de término disminuyen la calificación, como así también devoluciones para rehacer si algún ejercicio no cumple las consignas.
- **Los ejercicios deben ser resueltos con Optional y/o Streams.**

## EJERCICIOS - LEVEL 3

1. Dada una lista que contiene valores de tipo String, se deberá filtrar todos los valores que NO sean null o vacío ("").

### Input (Entrada):

```
List<String> palabras = new ArrayList<>();  
palabras.add("Hola");  
palabras.add(null);  
palabras.add("Informatorio");  
palabras.add("");
```

### Output (Salida):

```
["Hola", "Informatorio"]
```

2. Dada una lista que contiene números enteros, se deberá generar otra lista que contendrá el resultado de aplicar la operación de potencia a 2.

### Input (Entrada):

```
List<Integer> palabras = List.of(1, 2, 3, 4, 5);
```



## Output (Salida):

```
[1, 4, 9, 16, 25]
```

3. Se dispone de una lista de Strings, de la cual saber la cantidad de valores que comiencen con la letra B (tanto mayúscula como minúscula, en ese caso decimos que será ignoreCase).

## Input (Entrada):

```
List<String> palabras = List.of("Batman", "Aquaman", "Flash", "Batgirl",  
"Wonder Woman", "brainiac");
```

## Output (Salida):

```
3
```

4. Se dispone de una lista de Integer, de la cual queremos obtener otra lista aplicando la operación de factorial pero no se desean valores repetidos.

## Input (Entrada)

```
List<Integer> palabras = List.of(1, 2, 4, 4, 4);
```

## Output (Salida):

```
[1, 2, 24]
```

5. Se posee una Lista con objetos de clase Alumno con los atributos: apellido, nombre y fechaDeNacimiento (con tipos: String, String y LocalDate). Se desea generar un Map<String, Integer> donde la clave de Map será el apellido concatenado con el nombre (con separador de espacio en blanco) y el value la edad del alumno.
- La lista de entrada debe estar cargada con varios alumnos (al menos 5) para subir el ejemplo y demostrar su funcionamiento.
  - En ejemplo se muestra solo con 1 Alumno a modo de abreviar



- En el ejemplo también se usa `LocalDate.now().minusYears(30)`, en el ejercicio a presentar se deberá usar otra técnica de construcción para la fecha (no usar `.now()`). Pueden ver métodos `.parse()`, etc)

## Input (Entrada)

```
List<Alumno> alumnos = List.of(new Alumno("Homero", "Simpson",  
LocalDate.now().minusYears(30)));
```

## Output (Salida):

```
{"Simpson Homero"=30}
```

