

# Ejercicio Expresiones Lambdas



## Organizador de Tareas en un Smartwatch

### Enunciado

Una compañía de tecnología está desarrollando un software para su nuevo smartwatch. Este reloj tiene una funcionalidad llamada "**Organizador de Tareas**", que permite gestionar tareas diarias de manera eficiente.

En este ejercicio, deberás implementar un sistema que permita:

1. **Filtrar tareas** según diferentes criterios (por ejemplo: tareas completadas, tareas de alta prioridad, o tareas que vencen hoy).
2. **Ordenar tareas** según diversos criterios (por ejemplo: por fecha de vencimiento o por duración estimada).
3. **Ejecutar acciones** sobre tareas seleccionadas (por ejemplo: enviar un recordatorio o marcar como completada).

Para ello, deberás usar **interfaces funcionales y expresiones lambda**.

### Requisitos

#### 1. Clase `Tarea`

- Atributos:
  - `String nombre`: el nombre de la tarea.
  - `boolean completada`: indica si la tarea está completada.
  - `LocalDate fechaVencimiento`: fecha límite para completar la tarea.
  - `int duracion`: duración estimada de la tarea en minutos.
  - `int prioridad`: nivel de prioridad (1 = alta, 2 = media, 3 = baja).
- Métodos:
  - Constructor para inicializar los atributos.
  - Getters y setters según sea necesario.
  - Método `toString()` para imprimir los detalles de la tarea.

#### 2. Interfaz funcional `FiltroTarea`

- Un método abstracto:

```
boolean test(Tarea tarea);
```

#### 3. Interfaz funcional `AccionTarea`

- Un método abstracto:

```
void ejecutar(Tarea tarea);
```

#### 4. Clase **Organizador**

Métodos estáticos:

`filtrarTareas(List<Tarea> tareas, FiltroTarea filtro)`: recibe una lista de tareas y un filtro, y devuelve una lista con las tareas que cumplen el criterio.

`ordenarTareas(List<Tarea> tareas, Comparator<Tarea> criterio)`: ordena la lista de tareas según el criterio dado.

`aplicarAccion(List<Tarea> tareas, AccionTarea accion)`: aplica una acción a cada tarea en la lista.

#### 5. Clase **Main**

- Crea una lista de tareas de ejemplo.
- Usa expresiones lambda para:
  - Filtrar tareas completadas.
  - Ordenar tareas por prioridad.
  - Enviar un recordatorio para tareas que vencen hoy.