



# **Centro de Enseñanza Técnica Industrial**

## **Desarrollo de Software**

### **Programación de ExecutorService**

**Jesús Alberto Aréchiga Carrillo**

**22310439      5N**

**Profesor**

**José Francisco Pérez Reyes**

**Octubre 2024**

**Guadalajara, Jalisco**

# Introducción

En esta práctica se analizan los códigos de la implementación de `ExecutorService`, uno con el uso de lambdas y otro sin el uso de lambdas. Se documentan las diferencias tanto en código como en funcionalidad.

## Desarrollo

Los códigos para analizar son:

Sin uso de lambdas:

```
public class ThreadExample implements Runnable{
    private final String name;
    public ThreadExample(String name) {
        this.name = name;
    }

    @Override
    public void run(){
        for(int i = 1; i <= 10 ; i++)
            System.out.println("Running " + this.name + ": " + i);
    }
}
```

```
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
public class ExecutorServiceExample {
    public static void main(String[] args) {
        ExecutorService executor = Executors.newFixedThreadPool(3);
        executor.execute(new ThreadExample("Hugo"));
        executor.execute(new ThreadExample("Paco"));
        executor.execute(new ThreadExample("Luis"));
        executor.shutdown();
        // Esto intentará detener todas las tareas en ejecución de
        inmediato, no se dan
        // garantías para detener las tareas, es un intento de mejor
        esfuerzo
        //executor.shutdownNow();
        // Bloqueará el hilo que lo llamó hasta que se haya terminado por
        completo o hasta que
        // se agote el tiempo de espera.
        // Normalmente se llama después de shutdown() o shutdownNow().
        //executor.awaitTermination(10_000, TimeUnit.MILLISECONDS);
    }
}
```

Es necesario la implementación de la clase hilo, que será lo que haga el proceso. Después se implementa dicho hilo en el executor y se empieza a correr el hilo. Esto mantiene la misma estructura que usar los hilos por sí solos (sin el executor) pero la diferencia es que ya no se crea el hilo y se corre con el método start(), ahora el executor se encarga de ese trabajo.

|                  |                  |                  |
|------------------|------------------|------------------|
| Running Hugo: 1  | Running Hugo: 1  | Running Hugo: 1  |
| Running Luis: 1  | Running Paco: 1  | Running Luis: 1  |
| Running Paco: 1  | Running Luis: 1  | Running Paco: 1  |
| Running Luis: 2  | Running Paco: 2  | Running Luis: 2  |
| Running Hugo: 2  | Running Hugo: 2  | Running Hugo: 2  |
| Running Luis: 3  | Running Paco: 3  | Running Luis: 3  |
| Running Paco: 2  | Running Luis: 2  | Running Paco: 2  |
| Running Luis: 4  | Running Paco: 4  | Running Luis: 4  |
| Running Hugo: 3  | Running Hugo: 3  | Running Hugo: 3  |
| Running Luis: 5  | Running Paco: 5  | Running Luis: 5  |
| Running Paco: 3  | Running Luis: 3  | Running Paco: 3  |
| Running Luis: 6  | Running Paco: 6  | Running Luis: 6  |
| Running Hugo: 4  | Running Hugo: 4  | Running Hugo: 4  |
| Running Luis: 7  | Running Paco: 7  | Running Luis: 7  |
| Running Paco: 4  | Running Luis: 4  | Running Paco: 4  |
| Running Luis: 8  | Running Paco: 8  | Running Luis: 8  |
| Running Hugo: 5  | Running Hugo: 5  | Running Hugo: 5  |
| Running Luis: 9  | Running Paco: 9  | Running Luis: 9  |
| Running Paco: 5  | Running Luis: 5  | Running Paco: 5  |
| Running Luis: 10 | Running Paco: 10 | Running Luis: 10 |
| Running Hugo: 6  | Running Hugo: 6  | Running Hugo: 6  |
| Running Paco: 6  | Running Luis: 6  | Running Paco: 6  |
| Running Hugo: 7  | Running Hugo: 7  | Running Hugo: 7  |
| Running Paco: 7  | Running Luis: 7  | Running Paco: 7  |
| Running Hugo: 8  | Running Hugo: 8  | Running Hugo: 8  |
| Running Paco: 8  | Running Luis: 8  | Running Hugo: 9  |
| Running Hugo: 9  | Running Hugo: 9  | Running Paco: 8  |
| Running Paco: 9  | Running Luis: 9  | Running Hugo: 10 |
| Running Hugo: 10 | Running Hugo: 10 | Running Paco: 9  |
| Running Paco: 10 | Running Luis: 10 | Running Paco: 10 |

Uso de lambdas:

```
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;

public class ExecutorServiceExample {
    public static void main(String[] args) {
        ExecutorService executor = Executors.newFixedThreadPool(3);
        executor.execute(() -> doLongWork("Hugo"));
        executor.execute(() -> doLongWork("Paco"));
        executor.execute(() -> doLongWork("Luis"));
        executor.shutdown();
    }
}
```

```

private static void doLongWork(String msg) {
    for(int i = 1; i <= 10; i++) {
        System.out.println("Running " + msg + ": " + i);
    }
}
}

```

Es un código más compacto y fácil de leer. Al utilizar lambdas se deja de usar la clase de un hilo. No debe confundirse con que ya no se usen hilos, si se usan, pero ya no es necesario hacer la clase que herede de la clase hilo o que implemente la interfaz Runnable. Todo ya lo implementa el ExecutorService, simplemente es indicar qué función es la que va a hacer (función lambda).

|                  |                  |                  |
|------------------|------------------|------------------|
| Running Luis: 1  | Running Hugo: 1  | Running Paco: 1  |
| Running Hugo: 1  | Running Luis: 1  | Running Luis: 1  |
| Running Paco: 1  | Running Paco: 1  | Running Hugo: 1  |
| Running Hugo: 2  | Running Luis: 2  | Running Luis: 2  |
| Running Luis: 2  | Running Hugo: 2  | Running Paco: 2  |
| Running Hugo: 3  | Running Luis: 3  | Running Luis: 3  |
| Running Paco: 2  | Running Paco: 2  | Running Hugo: 2  |
| Running Hugo: 4  | Running Luis: 4  | Running Luis: 4  |
| Running Luis: 3  | Running Hugo: 3  | Running Paco: 3  |
| Running Hugo: 5  | Running Luis: 5  | Running Paco: 4  |
| Running Paco: 3  | Running Paco: 3  | Running Paco: 5  |
| Running Hugo: 6  | Running Luis: 6  | Running Luis: 5  |
| Running Luis: 4  | Running Hugo: 4  | Running Hugo: 3  |
| Running Hugo: 7  | Running Luis: 7  | Running Luis: 6  |
| Running Paco: 4  | Running Paco: 4  | Running Paco: 6  |
| Running Hugo: 8  | Running Luis: 8  | Running Luis: 7  |
| Running Luis: 5  | Running Hugo: 5  | Running Hugo: 4  |
| Running Hugo: 9  | Running Luis: 9  | Running Luis: 8  |
| Running Paco: 5  | Running Paco: 5  | Running Paco: 7  |
| Running Hugo: 10 | Running Luis: 10 | Running Paco: 8  |
| Running Luis: 6  | Running Hugo: 6  | Running Luis: 9  |
| Running Paco: 6  | Running Paco: 6  | Running Hugo: 5  |
| Running Luis: 7  | Running Hugo: 7  | Running Luis: 10 |
| Running Paco: 7  | Running Paco: 7  | Running Paco: 9  |
| Running Luis: 8  | Running Hugo: 8  | Running Hugo: 6  |
| Running Paco: 8  | Running Paco: 8  | Running Paco: 10 |
| Running Luis: 9  | Running Paco: 9  | Running Hugo: 7  |
| Running Paco: 9  | Running Hugo: 9  | Running Hugo: 8  |
| Running Luis: 10 | Running Paco: 10 | Running Hugo: 9  |
| Running Paco: 10 | Running Hugo: 10 | Running Hugo: 10 |

Un dato curioso en esta implementación es que hay un hilo que acaba mucho antes que los otros dos en ambas maneras de implementarlo. Hubo una excepción en el uso de lambdas, pero sigue teniendo un comportamiento parecido

En cuanto a la funcionalidad, son iguales, tienen la misma manera de ejecutarse, aunque puede que el uso de lambdas sea ligeramente más eficiente porque no tiene que acceder a otro archivo donde está la clase que implementa el hilo, en el caso de la implementación de lambdas todo está en el mismo archivo.

## **Conclusiones:**

El uso de `ExecutorService` es muy eficiente y ayuda mucho tanto a escribir código como leerlo. Se puede seguir implementando el hilo manualmente al indicar explícitamente cuál es la clase que se va a utilizar. En el caso del uso de lambdas es más sencillo hacer la implementación, sólo es necesario saber que se están utilizando hilos pero ya no es necesario indicarlos ni trabajar con ellos, simplemente se hace una función dentro de la clase (como un método normal) y `ExecutorService` crea el hilo utilizando la función que se le haya indicado como el método `run()`.

Personalmente me agrada más la implementación de lambdas, si tuviera que analizar un código en producción o de algún proyecto que no conozco y utiliza `ExecutorService`, ayudaría mucho más que implemente lambdas que la clase `Threads` manualmente.