INFORME SOBRE EL ARTÍCULO DE HILOS

FABIAN ESTEBAN VALENCIA ARBELAEZ C.C. 1.092.454.571

JHONNY ALEXANDER PEREA PEREA C.C 1.094.888.422

Programación 3

Profesor: LUDWIN AUGUSTO BUITRAGO

Universidad del Quindío

Facultad de Ingeniería

Ingeniería de Sistemas y Computación

Armenia, mayo 15 de 2024

INFORME

En el desarrollo de aplicaciones concurrentes en Java, es fundamental comprender y aplicar enfoques eficientes para gestionar la concurrencia de manera efectiva. En este informe, se analizan dos enfoques observados en el artículo:

- El enfoque tradicional con grupos de subprocesos
- El enfoque de hilos virtuales y futuros

ENFOQUE TRADICIONAL CON GRUPOS DE SUBPROCESOS:

El enfoque tradicional de programación concurrente en Java implica la creación y gestión manual de subprocesos para ejecutar tareas de forma paralela. En este enfoque, se utilizan grupos de subprocesos para controlar la ejecución de múltiples tareas de manera concurrente. Aunque este enfoque es muy utilizado, puede presentar varias dificultades en términos de eficiencia.

Características:
- Creación manual de subprocesos.
- Uso de “ExecutorService” para administrar grupos de subprocesos.
- Potencial creación incontrolada de hilos.
- Necesidad de sincronización para evitar condiciones de carrera.

Ejemplo de Código:

EjemploGrupoDeSubprocesos.java

```java
package practicaHilos;

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

public class EjemploGrupoDeSubprocesos {
    public static void main(String[] args) {
        ExecutorService executor = Executors.newFixedThreadPool(3);

        for (int i = 0; i < 3; i++) {
            executor.submit(() -> {
                System.out.println("Tarea en subproceso: " + Thread.currentThread().getName());
            });
        }

        executor.shutdown();
    }
}
```

Problems Javadoc Declaration Console

```
<terminated> EjemploGrupoDeSubprocesos [Java Application] /snap/eclipse/87/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86_64_17.0.10.v20240120-1143/jre/bin/java (14 may 202
Tarea en subproceso: pool-1-thread-2
Tarea en subproceso: pool-1-thread-1
Tarea en subproceso: pool-1-thread-3
```

ENFOQUE DE HILOS VIRTUALES Y FUTUROS:

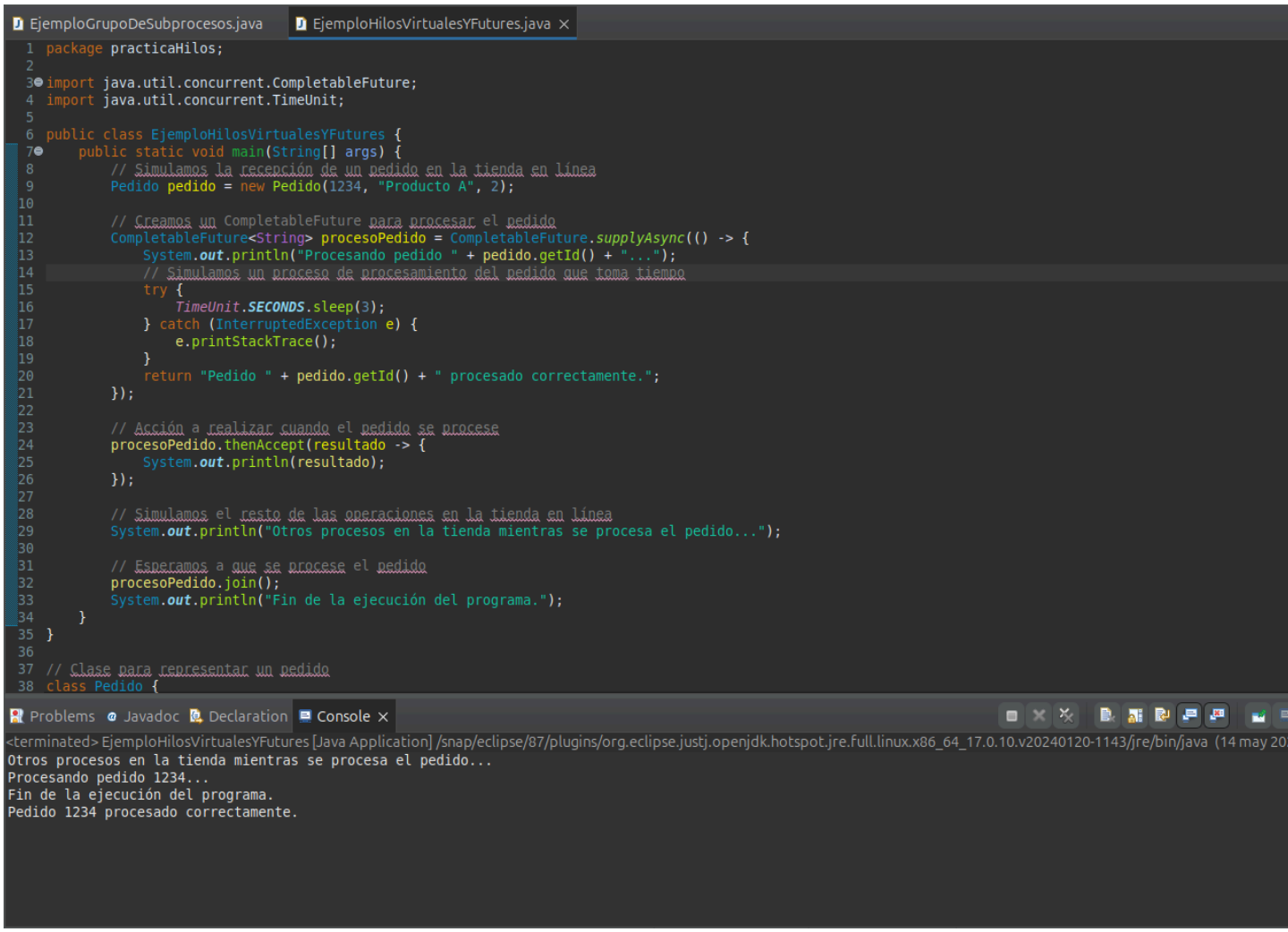
El enfoque de hilos virtuales y futuros introduce una forma más eficiente y optimizada de manejar la concurrencia en Java. Los hilos virtuales son hilos ligeros creados y gestionados por la JVM, lo que permite una ejecución más eficiente y optimizada. Por otro lado, los futuros eliminan el bloqueo y optimizan el rendimiento al centrarse en acciones realizadas por subprocesos en objetos compartidos.

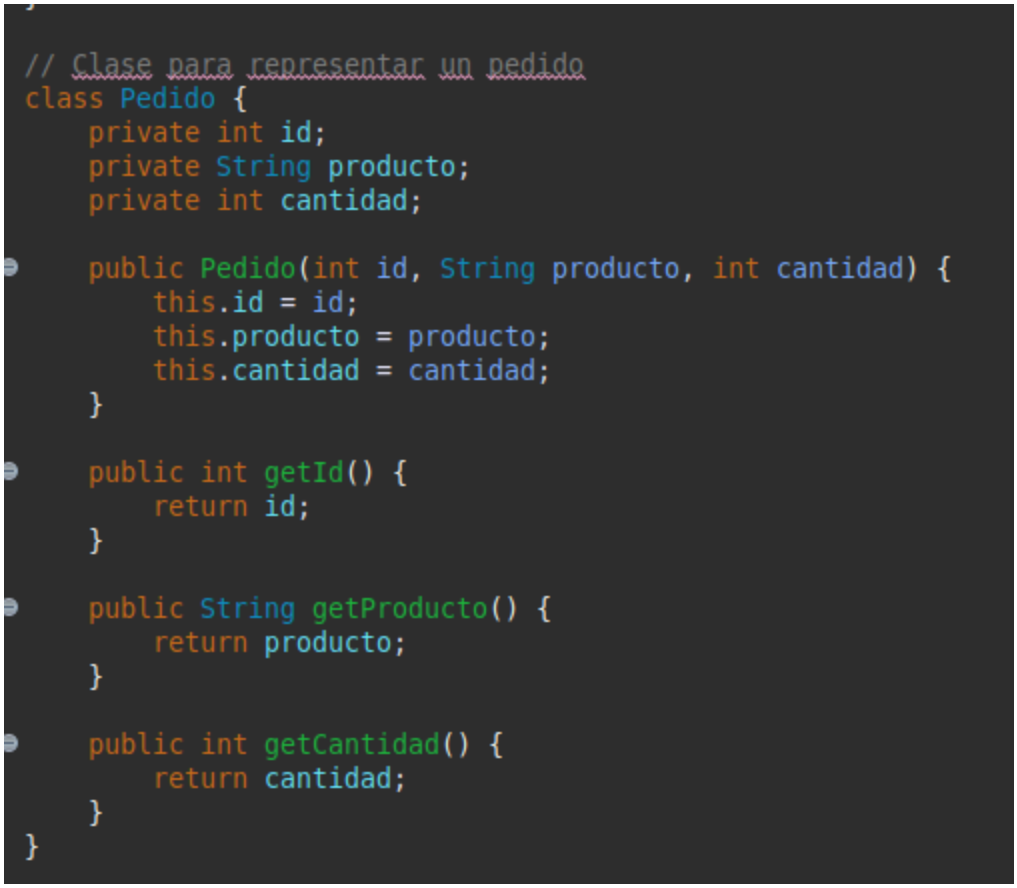
Características:

- Hilos virtuales creados por la JVM.
- Uso de “CompletableFuture” para trabajar con futuros.

- Eliminación del bloqueo a nivel de sistema operativo.
- Enfoque más funcional y centrado en acciones.

Ejemplo de Código:





```java
// Clase para representar un pedido
class Pedido {
    private int id;
    private String producto;
    private int cantidad;

    public Pedido(int id, String producto, int cantidad) {
        this.id = id;
        this.producto = producto;
        this.cantidad = cantidad;
    }

    public int getId() {
        return id;
    }

    public String getProducto() {
        return producto;
    }

    public int getCantidad() {
        return cantidad;
    }
}
```

CONCLUSIÓN:

Como conclusión, al comparar los enfoques tradicionales con grupos de subprocesos y los enfoques de hilos virtuales y futuros, se observa que la programación concurrente en Java ha evolucionado hacia métodos más eficientes y optimizados. Si bien el enfoque tradicional sigue siendo válido en ciertos contextos, el uso de hilos virtuales y futuros ofrece ventajas significativas en términos de rendimiento, escalabilidad y mantenibilidad del código.

En la elección del enfoque adecuado para el proyecto que estamos trabajando, es importante considerar los requisitos de concurrencia, la complejidad del código y los objetivos de rendimiento, por lo tanto se

consideró que la mejor forma de trabajar el proyecto es de la primera forma debido a que apenas nos estamos introduciendo en este tema tan basto y que más adelante es posible analizar cuál de las 2 es mejor utilizar dependiendo el proyecto.