**Actividad**

**UF 1-4. Hilos**

**Ivan Nuñez Rodriguez**

2º DAM

Programación de Procesos y Servicios

**Índice**

1. Código.
2. Resultado.
3. Enlace de Github

**Código**

**Hilo Primo**

import lombok.AllArgsConstructor;  
import java.time.Instant;  
  
@AllArgsConstructor  
public class HiloPrimo extends Thread {  
 private final int numero;  
 private final boolean guardarEnArchivo;  
  
 @Override  
 public void run() {  
 Instant inicio = Instant.*now*();  
 boolean esPrimo = esNumeroPrimo(numero);  
 Instant fin = Instant.*now*();  
  
 NumeroPrimo resultado = NumeroPrimo.*crear*(numero, esPrimo, getName(), inicio, fin);  
  
 // Mostrar resultado en pantalla  
 System.*out*.println(resultado.getResultado());  
  
 // Guardar en archivo si el usuario lo ha solicitado  
 if (guardarEnArchivo) {  
 Resultado.*guardarResultado*(resultado);  
 }  
 }  
  
 private boolean esNumeroPrimo(int num) {  
 if (num < 2) return false;  
 for (int i = 2; i <= Math.*sqrt*(num); i++) {  
 if (num % i == 0) return false;  
 }  
 return true;  
 }  
}

**Main**

import java.util.Scanner;  
  
public class Main {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 int[] numeros = new int[4];  
  
 System.*out*.println("Introduce 4 números para verificar si son primos:");  
 for (int i = 0; i < 4; i++) {  
 System.*out*.print("Número " + (i + 1) + ": ");  
 while (!scanner.hasNextInt()) {  
 System.*out*.println("Error: Introduce un número válido.");  
 scanner.next();  
 }  
 numeros[i] = scanner.nextInt();  
 }  
  
 System.*out*.print("¿Quieres guardar los resultados en un archivo? (S/N): ");  
 boolean guardarEnArchivo = scanner.next().equalsIgnoreCase("S");  
  
 System.*out*.println("\nProcesando números...\n");  
  
 HiloPrimo[] hilos = new HiloPrimo[4];  
 for (int i = 0; i < 4; i++) {  
 hilos[i] = new HiloPrimo(numeros[i], guardarEnArchivo);  
 hilos[i].start();  
 }  
  
 // Esperar a que todos los hilos terminen  
 for (HiloPrimo hilo : hilos) {  
 try {  
 hilo.join();  
 } catch (InterruptedException e) {  
 System.*err*.println("Error esperando a los hilos: " + e.getMessage());  
 }  
 }  
  
 System.*out*.println("\nProcesamiento terminado.");  
 }  
}

**Numero primo**

import lombok.AllArgsConstructor;  
import lombok.Data;  
import java.time.Duration;  
import java.time.Instant;  
  
@Data  
@AllArgsConstructor  
public class NumeroPrimo {  
 private int numero;  
 private boolean esPrimo;  
 private String nombreHilo;  
 private long tiempoProcesamiento;  
  
 public static NumeroPrimo crear(int numero, boolean esPrimo, String nombreHilo, Instant inicio, Instant fin) {  
 long tiempo = Duration.*between*(inicio, fin).toMillis();  
 return new NumeroPrimo(numero, esPrimo, nombreHilo, tiempo);  
 }  
  
 public String getResultado() {  
 return String.*format*("Número: %d | Procesado por: %s | Tiempo: %d ms | %s",  
 numero, nombreHilo, tiempoProcesamiento, (esPrimo ? "Es primo" : "No es primo"));  
 }  
  
 public String toFileString() {  
 return numero + "," + nombreHilo + "," + tiempoProcesamiento + "," + (esPrimo ? "Es primo" : "No es primo");  
 }  
}

**Resultado**

import java.io.FileWriter;  
import java.io.IOException;  
import java.io.PrintWriter;  
import java.time.LocalDateTime;  
import java.time.format.DateTimeFormatter;  
  
public class Resultado {  
 private static final String *NOMBRE\_ARCHIVO* = "resultados\_primos.txt";  
  
 public static synchronized void guardarResultado(NumeroPrimo resultado) {  
 try (FileWriter fw = new FileWriter(*NOMBRE\_ARCHIVO*, true);  
 PrintWriter pw = new PrintWriter(fw)) {  
  
 String timestamp = LocalDateTime.*now*().format(DateTimeFormatter.*ofPattern*("yyyy-MM-dd HH:mm:ss"));  
 pw.println(timestamp + " -> " + resultado.toFileString());  
  
 } catch (IOException e) {  
 System.*err*.println("Error al escribir en el archivo: " + e.getMessage());  
 }  
 }  
}

**Ejecución**

**Se indtroucen números**

**Texto

El contenido generado por IA puede ser incorrecto.**

**Se confirma la creación de texto**

****

**Se muestra en pantalla el resultado**

**Pantalla de computadora con letras

El contenido generado por IA puede ser incorrecto.**

**Se genera txt**

****

**Se muestran datos del txt**

**Texto

El contenido generado por IA puede ser incorrecto.**

**Enalce a Github**

[**https://github.com/Ivannunezrodriguez/Programacion\_de\_Servicios\_y\_Procesos\_24-25/tree/cd948895b43ec1d64a3a48ec8c152092474fa5d7/Actividades/tema\_2/Actividad%20UF1-4**](https://github.com/Ivannunezrodriguez/Programacion_de_Servicios_y_Procesos_24-25/tree/cd948895b43ec1d64a3a48ec8c152092474fa5d7/Actividades/tema_2/Actividad%20UF1-4)