

# Práctica de Programación en Java:

## Implementación del Problema de Parar

Hecho por: Alberto Valera

### **Introducción:**

El problema de parar, introducido por Alan Turing en 1936, es un problema fundamental en la teoría de la computación que demuestra la existencia de problemas indecidibles. En esta práctica de programación en Java, implementamos una simulación conceptual del problema de parar, explorando la indecidibilidad de ciertos problemas computacionales y la importancia de la teoría computacional.

### **Descripción de la Implementación:**

#### **1.Programa que Inevitablemente se Detiene:**

Implementé un programa en Java que cuenta hacia abajo desde el número 10 hasta cero. Una vez que el contador alcanza cero, el programa se detiene y se pone en bucle infinito.

#### **2.Programa que Nunca se Detiene:**

Implementé otro programa en Java que cuenta hacia arriba indefinidamente. Este programa se detiene ya que el contador siempre aumenta y nunca alcanza un límite y por eso el programa se para al ver un bucle indefinido.

#### **3.Función HaltChecker:**

Creé una función llamada haltChecker que simula la decisión teórica de si un programa se detiene o no con una entrada específica. Esta función no analiza realmente la ejecución del programa, sino que examina su estructura para tomar una decisión.

#### **4.Programa Reverser:**

Implementé un programa llamado Reverser que utiliza la función haltChecker. Este programa toma el código de un programa como entrada y determina si el programa se detendrá o no. Si haltChecker indica que el programa se detendrá, Reverser entra en un bucle infinito; de lo contrario, termina inmediatamente.

### **Discusión Reflexiva:**

#### **Desafíos y Límites de la Computación:**

Al simular problemas indecidibles como el problema de parar, he reflexionado sobre los desafíos inherentes a la computación. La imposibilidad de implementar un algoritmo perfecto para predecir paradas, resaltando los límites de lo que puede lograr la computación etc...

**Aprendizajes del Ejercicio:**

Este ejercicio me ha enseñado sobre la complejidad de ciertos problemas computacionales y la naturaleza de la indecibilidad. He comprendido mejor los límites de la computación y la importancia de la teoría computacional en nuestra comprensión del mundo digital.

Con esta práctica he logrado aprender más sobre el patrón de comportamiento de Chain of Responsibility, y me ha ayudado a hacer la práctica más amena y lograr aprender este patrón para el futuro ya que es muy importante y muy usado en la actualidad.

**Resultados Observados:**

La simulación de los programas de conteo hacia arriba y hacia abajo demuestra la diferencia entre un programa que se detiene y otro que nunca se detiene.

La función haltChecker simula la decisión teórica de si un programa se detiene o no.

El programa Reverser demuestra la paradoja del problema de parar al intentar determinar su propia terminación.

**Conclusiones:**

La simulación del problema de parar en Java me ha permitido explorar conceptos fundamentales en la teoría computacional, y también me ha permitido comprender la indecibilidad del problema de parar.

A través de la simulación de programas y el uso de la función haltChecker, pudimos visualizar la importancia de la estructura del programa en la determinación de su terminación.

La implementación del programa Reverser revela la contradicción paradójica al intentar determinar la terminación de un programa mediante un algoritmo.

En resumen, esta práctica me ha proporcionado una comprensión profunda de los límites de la computación y la existencia de problemas indecidibles en la teoría de la computación.