**CONCURRENCY & PARALLELISM**

Charles Aevedo Díaz

Álvaro José Ortega Vargas

Hugo José Pérez Guerrero

Ingeniería de Sistemas

Universidad Tecnológica de Bolívar

Cartagena, Colombia

# abstract

# ***In this paper, we are going to see two important topics in programming, such as concurrency and parallelism. We are going to see a brief definition about them, in order to know why it’s important to have knowledge in these topics, and then we are going to give some examples, so we can take a look on how it works.***

# ***keywords:*** Concurrency, parallelism, programming language, programs, javascript, scripts, node, multi core

# Introducción

# Para este proyecto, tendremos que ver algunas definiciones para continuar y practicar con los temas, por lo tanto, necesitamos hablar sobre la concurrencia, qué es y cuáles son los beneficios de usarlo durante la programación de scripts complejos.

# Vamos a aplicar esto en dos códigos diferentes usando javascript como lenguaje de programación principal, vamos a mostrar cómo usar hilos y usar diferentes núcleos en javascript, también necesitamos usar un entorno (node) para usar los núcleos, como Javascript es un lenguaje interpretado, el nodo permite interactuar con el servidor y hacer uso de los núcleos.

# Teoría

## Node.js:

Node es un entorno de ejecución de Javascript que incluye todo lo necesario para ejecutar un programa que está escrito en Javascript. Llegó a existir cuando se extendió JS para permitirle ejecutarlo en máquinas como aplicaciones independientes, como otros lenguajes de scripting que utilizan el motor de máquina en tiempo de ejecución V8 JS. Este motor toma su código JS y lo convierte en un código de máquina (nivel bajo) más rápido.

*Concurrencia:*

La concurrencia es la capacidad de un algoritmo para ejecutar más de una tarea a la vez. Escribir programas concurrentes puede ser difícil, ya que se necesita una buena abstracción para dividir el código en tareas separadas e independientes.

La concurrencia tiene diferentes niveles:

Nivel de instrucción, nivel de instrucción, nivel de unidad y nivel de programa.

*Paralelismo:*

El término paralelismo se refiere a las técnicas para hacer que los programas sean más rápidos al realizar varios cálculos al mismo tiempo. Esto requiere hardware con múltiples unidades de procesamiento. En muchos casos, los cálculos secundarios son de la misma estructura, pero esto no es necesario.

# Código

## En esta sección, hablaremos brevemente sobre los códigos que se adjuntarán junto con este documento; Hemos escrito dos códigos diferentes para mostrar los usos de la concurrencia y el paralelismo.

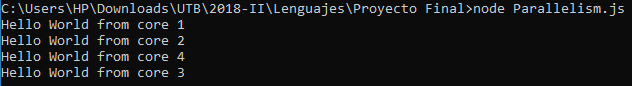
## Concurrencia:

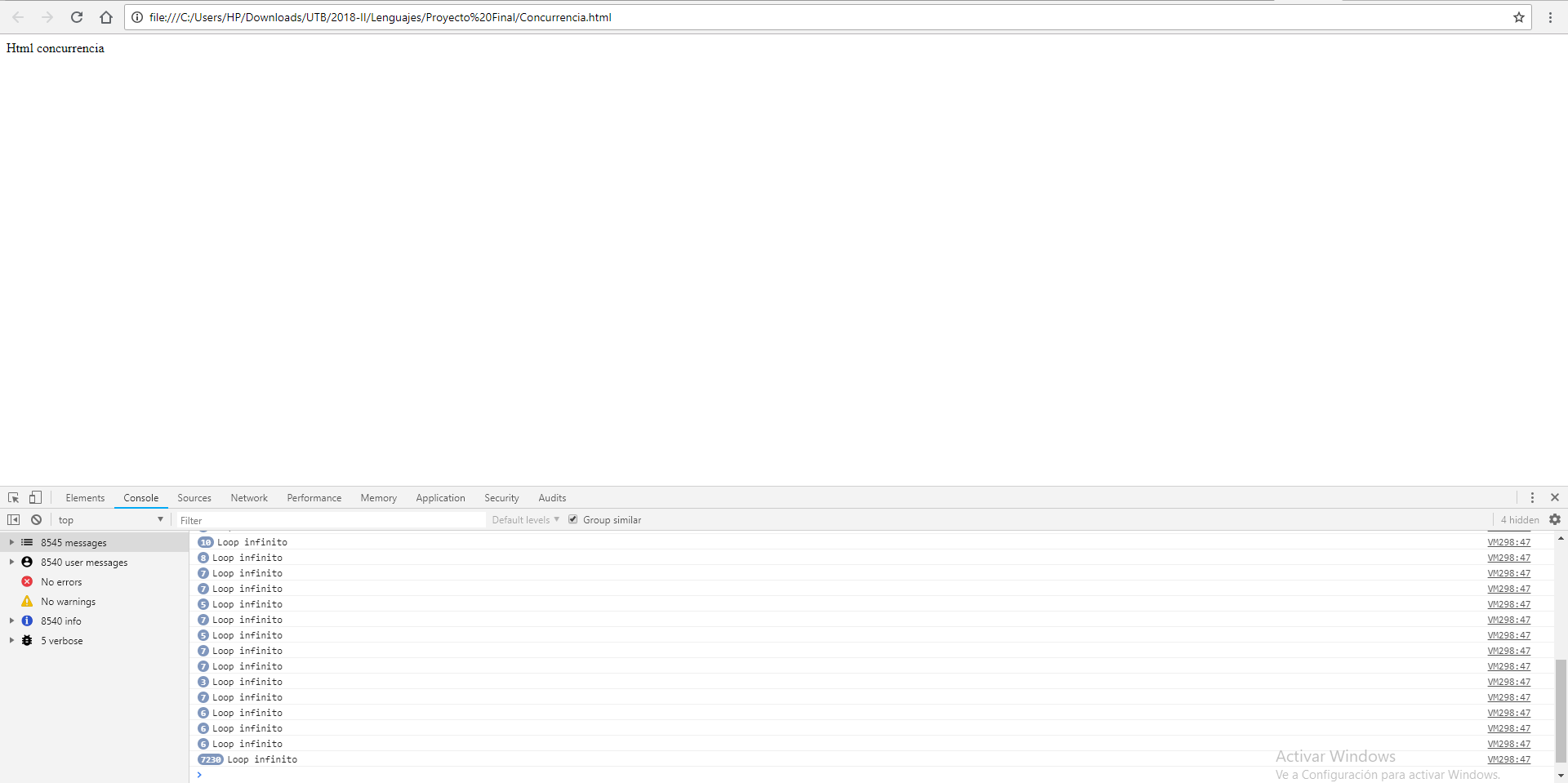
En el método de concurrencia, hemos escrito un script en un archivo HTML, que ejecutará un bucle para siempre, por lo que nunca obtendríamos el texto que está en el cuerpo del documento HTML. Para evitar esta situación, se ha utilizado un hilo para realizar esta función, mientras que el cuerpo se mostrará por separado.

*Paralelismo:*

Este código es bastante simple, pero muestra con buena precisión el uso del paralelismo y cómo funciona. Utilizará la cantidad de núcleos disponibles en la PC y registrará en la consola un mensaje de "Hello World" junto con el núcleo que se usó para hacerlo.

# Salidas





# CONCLUSIóN

# La concurrencia y el paralelismo pueden ser muy útiles para hacer programas más rápidos durante la ejecución, y aunque pueden verse como iguales, pero son diferentes, ya que la concurrencia es la capacidad de ejecutar dos (o más) tareas diferentes por el mismo programa en el mismo. Al mismo tiempo, sin dependencias, mientras que el paralelismo es la técnica de usar múltiples núcleos para ejecutar varios cálculos, esta depende del hardware (unidades de procesamiento).

# Referencias

1. https://wiki.haskell.org/Parallelism\_vs.\_Concurrency
2. Concepts of Programming Languages; Robert W. Sebesta; 10th Ed.
3. https://www.techopedia.com/definition/25146/concurrency-programming