



**Instituto Tecnológico y de Estudios Superiores de Monterrey**

Unidad de formación TC 2037.1. Implementación de métodos computacionales.

**Actividad 6.1 Análisis de herramientas usadas para la solución de las situaciones problema  
(evidencia de competencia).**

*Actividad colaborativa*

Grupo 1

David Alejandro Velázquez Valdéz A01632648

Diego Alberto Ortiz Mariscal A01552000

Felix David De Haro Soto A01637589

28 de Mayo del 2021

### **Actividad 6.1 Análisis de herramientas usadas para la solución de las situaciones problema (evidencia de competencia).**

Un resaltador léxico es uno de los bloques fundamentales para un programador ya que es una de las primeras formas de debuggear y organizar código, pero no solo es eso, si no también implementa varios de los fundamentos de programación como lo son las máquinas de estados y las expresiones regulares. Para poder resolver la situación problema de manera que se hiciera escalable y fácil de adaptar, usamos diferentes herramientas que nos apoyaron en esta tarea de diferentes maneras. Esto va desde los editores de texto en donde escribimos el código, hasta diferentes herramientas de diagramas que nos permitieron planear el procesamiento de cada uno de los tokens.

Para poder facilitar la explicación de las herramientas que usamos durante el proceso de desarrollo del proyecto, se separaron en: Teoría de Autómatas, Programación y trabajo en equipo. Cada una de estas áreas tiene las herramientas que se relacionan entre sí y que se usaron en el proyecto en alguna etapa.

#### **Teoria de Automatas**

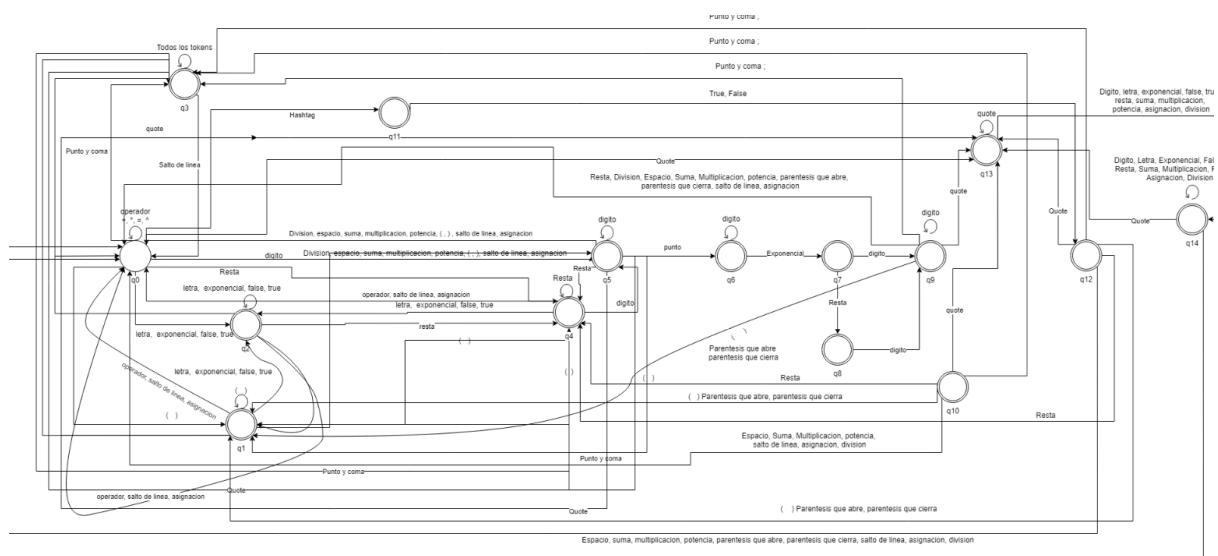
Para esta primera área se van a tocar todos los apoyos que se usaron durante la planeación temprana del proyecto, ya que Teoria de automatas es el fundamento de nuestra solución de la situación problema. El enfoque de esta etapa temprana fue con el tema de AFD (Autómata Finito Determinista) y cómo sería usado para empezar a construir el lexer.

Un Autómata Finito Determinista es, en sus raíces, una máquina de estados y las máquinas de estados son grafos dirigidos. El funcionamiento de una AFD comienza con una cadena de caracteres que se acepta como entrada si cumple con el requisito de pertenecer al alfabeto que se está analizando, para después transicionar del estado inicial a otro estado dentro de la máquina. La característica especial de una AFD es que cada transición tiene solamente un estado de destino. Esta característica en específica permite realizar un lexer, ya que los tokens que se analizan (la mayor parte de las veces) pertenecen a un solo estado.

Una de las herramientas que se usó para comprender mejor las máquinas de estado fue JFlap, un programa realizado en Java que te permite recrear las máquinas y simular un alfabeto y

una cadena de caracteres que se usará para simular el paso de los caracteres por cada transición y estado.

Los AFD pueden ser representados de diferentes maneras, teóricamente y visualmente. Para la situación problema se necesitaba visualizar la máquina de estados para poder trasladar el concepto a código, por lo que se usó la herramienta de “app.diagrams.net”. En esta herramienta se representó en forma de un diagrama UML, ya que esta es el estándar para realizar diagramas relacionados con software y su planeación. Dentro de esta herramienta se encuentran los estados y las transiciones marcadas con formas específicas para facilitar la comprensión de la máquina de estados.



**Imagen 1.** Diagrama UML del AFD usado para resolver la situación problema.

En la **Imagen 1** podemos observar el estándar UML para la representación de la máquina de estados, siendo el círculo un estado, y cada una de las líneas una transición con un requisito específico dentro del alfabeto.

## Programación

- Expresiones regulares

Las expresiones regulares son inmensamente útiles en el mundo de la computación y no es sorpresa alguna que sean algunos de los fundamentos y bloques de construcción de los resaltadores de texto, las expresiones regulares nos permiten buscar, manejar y modificar texto. Por lo que nos sirvió en nuestro resaltador léxico para localizar

palabras y formatearlas según sea necesario, sin embargo en vez de usar regex de alto nivel, en cierta forma implementamos nuestra regex mediante la máquina de estados.

- C++

C++ es una de los lenguajes de programación más optimizados y utilizados en la industria, tiene distintos usos y es ampliamente utilizado en distintas aplicaciones, decidimos usar C++ no solo por su rapidez sino también por sus herramientas para programación orientada a objetos, lo que nos permite abstraer distintas partes del programa y por lo tanto hacerlo escalable, modificable y fácil de testear. Esto nos fue muy útil desde el principio ya que el proyecto fue cambiando en cuanto escala, requerimientos y complejidad, por lo que tener distintas partes abstractas y como objetos nos permite modificar rápidamente el programa y enfocarnos únicamente en los cambios necesarios. Así mismo C++ es un lenguaje ampliamente utilizado para la programación paralela y concurrente lo que nos permite hacer el programa multithreaded y al incorporar objetos únicamente nos enfocamos en la lógica de concurrencia productor - consumidor.

- HTML-CSS

El objetivo principal de la segunda entrega de la situación problema es optimizar el tiempo de ejecución del programa con un paradigma de programación llamado Paralelismo. Este paradigma viene de la idea de “Divide y vencerás” aplicado a los problemas de programación y computación; la diferencia aquí es que las partes divididas del problema se resuelven al mismo tiempo para poder realizar muchas operaciones en un periodo corto de tiempo. Este paralelismo en computadoras modernas se obtiene al usar los hilos que tiene tu procesador que usualmente no se utilizan al momento de computar un problema, por esta razón es que la programación paralela depende mucho del hardware que se esté usando. El objetivo de aplicarlo en nuestro proyecto es el de acelerar el proceso de análisis de los tokens para poder leer varios archivos de texto a la vez de forma rápida y eficiente.

- Java como lenguaje para el manejo de Concurrencia/Paralelismo y generación de GUIs

Java es un lenguaje de programación que permite el manejo de concurrencia/paralelismo en la compilación del programa que provoca que el tiempo de ejecución del programa sea más rápido.

Paralelismo se define como la ejecución simultánea de uno o más tareas, lo cual va relacionado con el hardware de la computadora que está ejecutando el programa, ya que cada computadora cuenta con un procesador diferente, cada uno de estos cuenta con un diferente número de cores y threads que se encuentran en el procesador, siendo los threads la capacidad de trabajar de manera paralela la ejecución de un cierto programa, lo cual aumenta el rendimiento del sistema que se está ejecutando. El entorno de Java tiene la capacidad de poder hacer paralelismo, teniendo diversas librerías que permiten el manejo de cada uno de los threads que se utilizan para la ejecución de algún software, una de las funciones disponibles de Java para usar paralelismo mediante threads es la siguiente:

- `start();`
- `Thread.sleep();`
- `wait();`
- `notify();`
- `join();`

Las funciones anteriores son de las más usadas para el uso de threads ya que permiten diversas acciones que ayudan a obtener los resultados que se requieren al momento de utilizar los threads como herramienta para aumentar el rendimiento del sistema en la ejecución del programa.

Como se mencionó anteriormente, Java tiene la capacidad de utilizar paralelismo y también como en muchos lenguajes, la ejecución en serie, lo que distingue entre los 2 es la rapidez de ejecución entre cada tipo de método para la ejecución del programa.

Para la cuestión del GUI, se utilizó IntelliJ IDEA para generar interfaces gráficas y de esa manera facilitar el uso de un programa.

Debido a que este lenguaje es muy importante y frecuentado hoy en día, es posible que el apareamiento de nuevas tecnologías vaya incrementando debido a que muchas personas utilizan este lenguaje como medio para la creación de aplicaciones que ayuden a un fin en específico, por lo que el surgimiento de nuevas tecnologías seguirá destacando con este lenguaje.

- IDE(jetbrains -> CLion, IntelliJ IDEA)

Jetbrains es una empresa de desarrollo de software el cual hace diversos entornos de desarrollo integrado para diferentes usos y lenguajes, sus diferentes IDE son muy completos y facilitan el trabajo al momento de desarrollar software, dando la facilidad de brindar herramientas para el desarrollo de software, herramientas para el trabajo en equipo, entre otros. Los principales programas que vienen de la misma empresa Jetbrains para la realización de todos los trabajos fueron CLion para el lenguaje c++, e IntelliJ IDEA para el lenguaje Java con la posibilidad de hacer interfaces gráficas. Lo que destaca de Jetbrains ante los demás IDE para los lenguajes utilizados, es que facilita la creación de archivos que te ayuden a compilar el programa de manera rápida y sencilla, además de que te instala las librerías esenciales y te autocompleta funciones de una manera muy inteligente por parte del IDE, lo que en otros IDEs a veces se requieren de métodos diferentes y que toman más tiempo para desarrollar algún software, por lo que se vuelve una compañía muy popular gracias a la versatilidad de sus programas y la comodidad de desarrollar software en ellos.

Jetbrains al ser una compañía que busca el desarrollo de nuevas tecnologías para ofrecer herramientas nuevas para los programadores, se irán teniendo mejores versiones de sus productos e incluso más productos que ayuden a otros desarrolladores a trabajar de una manera más rápida y eficiente.

### **Trabajo en equipo**

- Github

Github es una herramienta que beneficia el poder alojar proyectos de software a dicha plataforma a la nube, la cual tiene control de versiones, siendo utilizado principalmente para la creación de código fuente, github es una herramienta donde se pueden compartir otros espacios de trabajo a los demás con fines de que apoyar a otros desarrolladores de software, lo cual lo vuelve una herramienta esencial para el control del proyecto y la colaboración. Esta herramienta es la principal para realizar trabajos con las características anteriores, ya que hoy en día no existen herramientas igual de profesionales, lo cual se vuelve la de máxima prioridad para desalojar todos los trabajos que se hicieron a lo largo del curso y ver el historial de cambios en caso de requerirse. En cuanto a la aparición de nuevas tecnologías, esta plataforma ha ido mejorando y brindando más herramientas para su uso, volviéndolo más eficiente y una herramienta que se actualiza y se hace más útil a lo largo del tiempo como desarrolladores.

**Reflexión - Participación del pensamiento formal en la aparición de nuevas tecnologías:**

Desde que existen civilizaciones, se ha dado a ver que el humano es un ser con capacidades cognitivas muy grandes, el cual tiene la capacidad de idear nuevas herramientas o formas de poder resolver problemas de manera más sencilla, lo cual se ha ido fomentando la cuestión de que el lenguaje y la lógica son los principales participantes para lograr nuevos aprendizajes. Esto conlleva al desarrollo de la ciencia y la tecnología hoy en día, desde el inicio de la revolución industrial, se han encontrado maneras de realizar acciones humanas muy complicadas y que requieren mucho tiempo con el uso de tecnologías que facilitan todo y realizan procesos mucho más complejos, rápidos y eficientes, se puede ver cómo el pensamiento formal tiene que ver con el desarrollo y evolución de las nuevas tecnologías que presentamos hoy en día.

El pensamiento formal se desarrolla a partir de los once a los quince años aproximadamente, donde el individuo tiene la capacidad de idear para poder resolver problemas, pudiéndose prescindir de la situación, del contenido completo y palpable de las cosas que nos rodean, situándonos en un ambiente abstracto que nos conlleva a una resolución de problemas. Cada 5 años parece que la tecnología va avanzando de forma impresionante cada día se desarrollan capacidades tecnológicas más rápidas más eficientes y menos costosas, todo esto lo debemos a la alta competitividad entre las empresas globales lo que lleva a que la tecnología.

El ser humano al ir evolucionando y creciendo de tal manera, la aparición de nuevas tecnologías conlleva una serie de procesos abstractos que tienen que ser originados de una idea que busque la solución de un problema en específico, tomándose como parte del proceso la participación del pensamiento formal, lo cual conlleva a que el humano tenga la curiosidad de emprender y de idear una solución ante el problema destacado; esto ha provocado que el surgimiento de nuevas tecnologías se den a ver mediante el pensamiento formal, tomando una serie de elementos tan abstractos para la creación de estar, viniendo desde hipótesis formales y de diversos estudios científicos, físicos, matemáticos, entre otros para la resolución de los nuevos problemas a enfrentar en nuestra vida diaria. Gracias a que se tiene la capacidad de analizar con detenimiento el problema a resolver, existen maneras de siempre llegar a una resolución del problema planteado, se puede visualizar que los problemas que se han ido resolviendo a lo largo del curso, se ha involucrado la participación del llamado pensamiento formal, donde se requiere encontrar diferentes soluciones para la resolución de los problemas que se encuentran adelante.

## Referencias:

- Anaya, C., Gómez, M., Paba, C. (2018). Eficacia de un programa para el desarrollo del pensamiento formal en estudiantes de 9° de un colegio público del departamento del Magdalena. *Redalyc*. Recuperado de <https://www.redalyc.org/jatsRepo/4975/497563255007/html/index.html>
- Aquino, F. (2003). El pensamiento formal y la educación científica en la enseñanza superior. *Redalyc*. Recuperado de <https://www.redalyc.org/pdf/311/31100704.pdf>
- Autómata Finito Determinista. (2013). *Ecured*. Recuperado de [www.ecured.cu/Autómata\\_finito\\_determinista](http://www.ecured.cu/Autómata_finito_determinista)
- Herramientas esenciales para desarrolladores de software y equipos. (s. f.). *Jetbrains*. Recuperado de <https://www.jetbrains.com/es-es/>
- Programación paralela. (s. f.). *Ferestrepoca.github.io*. Recuperado de [http://ferestrepoca.github.io/paradigmas-de-programacion/paralela/paralela\\_teoría/index.html](http://ferestrepoca.github.io/paradigmas-de-programacion/paralela/paralela_teoría/index.html)
- Ricardo. (2017). Autómatas finitos deterministas y no deterministas. *RicardoGeek*. Recuperado de <https://ricardogeek.com/automatas-finitos-deterministas-y-no-deterministas/>
- Roales, N. (2015). El paralelismo en java y el framework Fork/Join. *Adictos al trabajo*. Recuperado de <https://www.adictosaltrabajo.com/2015/08/21/el-paralelismo-en-java-y-el-framework-forkjoin/#:~:text=Paralelismo%20NO%20es%20concurrency&text=Ambos%20se%20basan%20en%20la,solapados%20C%20pero%20no%20necesariamente%20simult%C3%A1neos>