

**Actividad 6.1 Análisis de herramientas usadas para la solución de las situaciones problema**

Carlos Estrada Ceballos A01638214

Natalia Velasco García A01638047

Abigail Velasco García A01638095

Eduardo Esteva Camacho A01632202

28 de mayo del 2021

Implementación de métodos computacionales

TC2037.2

La situación problema planteada en este curso fue abordado en partes, el primer paso consiste en hacer un análisis, el segundo en diseñar un plan, el tercero en ejecutarlo y por último en la etapa cuatro se valida el resultado. Para cada etapa de la solución decidimos usar ciertas herramientas de acuerdo con distintos criterios tales como accesibilidad, facilidad, eficiencia y funcionalidad.

En la etapa de análisis las herramientas que usamos tienen como fin analizar el lenguaje, puesto que la situación problema se trata del procesamiento del lenguaje, en esta categoría se encuentran los autómatas finitos deterministas, expresiones regulares y JFlap.

La siguiente etapa es la de diseño de la solución, en esta etapa es necesario documentar el plan, para esto se usaron aplicaciones para hacer diagramas, otras herramientas que se consideraron dentro de esta categoría son las que están relacionadas con paralelismo y concurrencia, pensando en mejorar la eficiencia de la solución propuesta.

Finalmente, las etapas tres y cuatro están dentro de la categoría de implementación, aquí se toman en cuenta las herramientas que nos fueron más útiles al ejecutar lo que planeamos en las etapas pasadas.

A continuación, se presentan todas las herramientas utilizadas más a detalle y las razones por las que consideramos que eran necesarias para resolver la situación problema.

**Análisis del lenguaje**

El procesamiento del lenguaje nos permite crear herramientas que facilitan muchas tareas en la vida cotidiana, por ejemplo, aplicaciones como traductores, auto correctores, buscadores, editores, compiladores, intérpretes, etc. Esto es posible gracias al estudio del lenguaje, para entenderlo es necesario hacer un análisis léxico, sintáctico y semántico. (Olmos, 2018)

Existen varios medios para procesar el lenguaje, el que se utilizó en clase fue un generador de análisis léxico, justo en eso consiste una parte de la situación problema, en hacer un programa que reconoce e identifica los tipos de tokens de un conjunto de símbolos (La entrada) y determina si pertenecen o no al lenguaje para después crear como salida un archivo que muestra cada token y su tipo.

La solución propuesta en el equipo fue determinar la categoría léxica de cada token, leer la entrada de símbolos lexema por lexema y agruparlos hasta encontrarse con un token de otra categoría que no puede unirse al grupo. Las herramientas que se usaron para resolver esa tarea son expresiones regulares, para la especificación de tokens y autómatas finitos deterministas para estructurarlos.

Las expresiones regulares son un metalenguaje, un metalenguaje sirve para describir un lenguaje, las expresiones regulares describen los elementos léxicos; orden y precedencia, repeticiones, y agrupaciones. Esto permite agrupar símbolos, concatenar cadenas y elegir alternativas, en resumen, encontrar palabras. (Cueva, 2000)

“Una máquina abstracta o autómata es un dispositivo teórico capaz de recibir y transmitir información” (Isasi, Martínez, Borrajo, 1997, p. 2). Está formado por un conjunto de estados, un estado inicial y subconjuntos de estados definidos y finales; los estados son representados por nodos; también tiene un conjunto de tokens aceptados, y una función de transición representada por arcos. (Olmos, 2018)

El nodo en el que está la cinta al leer una entrada es el estado inicial, según el símbolo de entrada ocurren las transiciones entre estados y cuando llega a un estado final se puede terminar la palabra o cambiar de estado.

Los autómatas finitos y las expresiones regulares se relacionan de forma directa, los autómatas reconocen o validan palabras, aceptan secuencias de transiciones de acuerdo con las reglas del lenguaje descritas por las expresiones regulares. (Isasi, Martínez, Borrajo, 1997).

Estas herramientas son muy importantes, más allá de ayudarnos a implementar un identificador y resaltador léxico, fueron de gran ayuda para entender cómo funcionan los lenguajes naturales y formales y cómo funcionan las herramientas de procesamiento de lenguaje que usamos todos los días.

Conocerlas es de gran utilidad en el desempeño de nuestra carrera, puesto que nos permite tener un mejor criterio para evaluar y comparar herramientas de procesamiento de lenguaje para tomar mejores decisiones y facilita adaptarse a un nuevo lenguaje de programación.

**JFlap**

JFlap es una herramienta que facilita el aprendizaje de la teoría de autómatas, esta te permite “experimentar con temas de lenguajes formales que incluyen autómatas finitos no deterministas, autómatas de pila no deterministas, máquinas de Turing de cintas múltiples, varios tipos de gramáticas, análisis sintáctico y sistemas L.” (Rodger, 2005)

Esta aplicación fue desarrollada con el objetivo de mejorar la experiencia de aprendizaje de estudiantes universitarios y de preparatoria; otorga feedback inmediato, permite a los estudiantes aplicar la teoría por lo que la complementa, al representar gráficamente los conceptos los hace más entendible, además permite realizar pruebas con diferentes inputs para conocer las limitantes de la solución y si es correcta.

Está aplicación es gratuita además su página web contiene toda la información necesaria para usar el software. Esta cuenta con un libro, módulos, ejercicios, presentaciones, documentación, videos, guías, tutoriales, artículos, etc.

**Diseño de desarrollo de software**

**Diagramas UML**

El Lenguaje unificado de modelado (UML) es indispensable para lograr un buen desarrollo de software, ya que te permite documentar el diseño y las decisiones tomadas sobre este, abordar un problema en partes, definir requerimientos, representar información para los diferentes stakeholders y aumentar la calidad del software.

Dos de las herramientas más comunes para realizar estos diagramas son Lucidchart y app.diagrams.net. Ambas satisfacen necesidades diferentes y dependiendo del proyecto una es mejor que la otra; para estudiantes app.diagrams.net sería la aplicación más conveniente por su accesibilidad, en cambio Lucidchart es una mejor opción para empresas por todas sus herramientas de colaboración y sus integraciones. Las diferencias entre ambas aplicaciones están descritas en la siguiente tabla:

|  |  |  |
| --- | --- | --- |
|  | **Lucidchart** | **app.diagrams.net** |
| Accesibilidad | - Es necesario pagar por el servicio.  - Cuenta con una versión gratuita, sin embargo, limita el total de elementos que puedes insertar en todos tus documentos.  - Puede ser utilizada en dispositivos móviles Android y iPhone. | - Es una aplicación gratuita.  - Para realizar un documento no es necesario tener una cuenta.  - Tiene una versión de escritorio disponible para Windows, macOS y Linux, esto te permite realizar diagramas sin internet. |
| Facilidad de implementación | -Cuenta con plantillas, guías y una herramienta para encontrar funciones.  - Tiene más plantillas con diseños más variados sin embargo la mayoría son para cuentas premium.  - Vincula datos de otros documentos que son actualizados automáticamente en el diagrama.  - Puede importar diagramas de otras aplicaciones. | - Todas las plantillas son gratuitas.  - Puedes agregar figuras individuales.  - La aplicación tiene un diseño intuitivo.  - Puede trabajar con diagramas realizados desde 2005 e importar diagramas de otras aplicaciones. |
| Colaboración | - Permite comentar cada elemento y editar un documento al mismo tiempo.  - Registra los cambios y los usuarios que los realizaron. | - Se colabora por medio de Google Drive.  - No registra los cambios que hace cada usuario.  - Permite comentarios. |
| Seguridad | - Cuenta con bloqueo de dominio, soporte técnico y asignación de licencias | - La empresa no puede acceder a los datos.  - Los documentos solo se guardan en plataformas de almacenamiento en la nube o en la memoria local del dispositivo |
| Integraciones | G Suite, Microsoft Office, Slack, Jira, Confluence, Salesforce, Slack, AWS, Azure, GCP, Zapier, GitHub, BambooHR, Quip, LeanIX, Asana, AuditBoard, Nextpond, UVexplorer, Zoho Connect, Nuclino y Dropbox Paper. | Atlassian Confluence, Jira, Google Drive, OneDrive, GitHub, Aha!, Atom, BookStack, Docstell, FOSWiki, Grafana, Growi, JupyterLab, LumApps, Nextcloud, Nuclino, ownCloud, Redmine, Samepage, Sandstorm, Tiki Wiki CMS Groupware, VS Code y XWiki. |