

HERRAMIENTAS RECOMENDADAS



DIAGRAMS.NET

- Google Drive y Google Workspace
- GitHub, Gitlab y Dropbox
- Variedad de diagramas



JFLAP

- Varios temas de lenguajes formales
- Construcción y prueba de ejemplos
- Experimentación con pruebas de construcción



AUTÓMATAS FINITOS DETERMINISTAS

- Fácil de comprender (lenguaje universal)
- Sencillo de elaborar



GITHUB

- Búsqueda muy rápida en estructura de los repos
- Comunidad amplia
- Fácil integración con servicios de terceros.



HTML + CSS

- Fácil aprendizaje
- Documentos más limpios mediante la estructuración de bloques lógicos
- Compatible con múltiples navegadores



LENGUAJE C++

- Alto desempeño
- Fácil manejo de memoria
- Soportado en IDEs más populares



VISUAL STUDIO CODE

- Soporte nativo para múltiples de programación
- Biblioteca de extensiones
- Configuración de interfaz

INTEGRANTES

Abraham Mendoza Pérez	A01274857
Aldo Alejandro Degollado Padilla	A01638391
Luis Alonso Martínez García	A01636255

Link de video: <https://youtu.be/lzTI6JivDPk>

Guión:

Abraham Mendoza Pérez primera parte:

- Diagrams y JSFlap -

Primero empezamos por diagrams.net, el cual solo lo usamos solo para crear el diagrama de autómatas finitos, el cual fue que optamos para poder llegar a la resolución de la situación problema aunque es importante recalcar que no solo se queda en hacer este tipo de diagrama, sino que como su nombre lo indica, diagrams.net nos ofrece una variedad. Bastante grande, de distintos diagramas, entre los cuales puedo mencionar diagramas UML, entre los cuales pueden estar diagramas de estado, diagramas de casos, actividades, muchos de los cuales son para un aspecto más técnico. Esto se puede ver al momento de estar trabajando en códigos que ya estén terminados o incluso que estén en proceso, los cuales nos permiten analizar de mejor manera cómo es que se comportan estos. Y esto también facilita mucho el trabajo en equipo, ya que se puede exponer de la mejor manera y como es que este propio código funciona. Y esto no solo se queda así, ya que uno de las cosas que ofrece esta herramienta es poder trabajar con herramientas como Github, Gitlab o Dropbox, ya que en estas ofrece actualizaciones de los templates de la página, por ejemplo en Github cada vez que hay un nuevo diagrama llegan al repositorio de la herramienta donde se van lanzando ideas, actualizaciones entonces creemos que es algo de bastante ayuda, ya que Github es algo con lo que hemos estado trabajando en estos últimos años de nuestra carrera.

De igual manera cuenta con otras ventajas las cuales son tener Google Drive y Google Workspace, ya que el poder estar trabajando de manera colaborativa al no estar en el mismo lugar creemos que es de bastante ayuda y por ejemplo al estar en esta modalidad en línea pues resulta bastante ventajoso para todos los miembros del equipo. Eso sería por aparte de diagrams.net.

Las siguientes herramientas que utilizamos fue JFLAP. Como tal, este consiste en un programa que se corre en Java en el cual nos ofrece el crear nuestros propios lenguajes formales. Entre estos lenguajes formales vamos a encontrar que son expresiones regulares máquinas de Turing, máquinas de Moore y obviamente lo que nosotros necesitamos es el autómata finito. Y lo interesante es que aparte de construir, se nos ofrece la opción de que podamos revisar entradas en este mismo autómata finito y de acuerdo a las condiciones que se les haya dado por proceso para poder pasar las siguientes transiciones. Es que va funcionando. Entonces esto nos ayudó muchísimo para ir probando si nuestro autómata finito

estaba realizando correctamente la clasificación que en primer lugar fueron tokens, pero a la última versión ya fue para categorías léxicas. Entonces eso fue de muchísima ayuda para poder tener un autómata finito que estuviera en forma y cumpliera las expectativas de la situación promedio.

Luis Alonso Martínez:

- Automatas Finitos Deterministas y Github Desktop -

Como lo mencionó mi compañero Abraham, para la situación problema estuvimos trabajando con el concepto de Autómatas Finitos Deterministas para realizar el analizador léxico, que consiste en recibir una cadena de entrada y va procesando los símbolos de entrada de uno a la vez. Su funcionamiento comienza en la posición del estado inicial, y desde este estado comienza su ejecución. Entonces utilizamos todo este concepto dónde empieza en este estado inicial y va pasando por los estados dependiendo del carácter con el que se encuentra hasta llegar a un estado final que representa una categoría del lenguaje que se está analizando.

Entonces terminamos utilizando este método porque creemos que es fácil de comprender ya que es un lenguaje universal, además de que creemos que su elaboración o implementación es sencilla, el trasladar este concepto que ya platicamos a código realmente no es algo muy difícil, ya que las definiciones y conceptos son muy claros.

Para la parte de trabajo colaborativo nosotros usamos y recomendamos ampliamente el uso de GitHub, principalmente aprovechando su herramienta de GitHub Desktop porque te brinda una interfaz que la verdad es que es muy sencillo de utilizar, y la visualización de las modificaciones que realizas es bastante útil a la hora de realizar cualquier proyecto en equipo, ya que además de ver los cambios que han realizado los demás colaboradores, puedes ver cuáles son los cambios que vas a subir en el commit, lo que da una mejor visibilidad de la evolución de los proyectos. Además de que tiene una búsqueda muy rápida en estructura de los repos, tiene una comunidad amplia, por lo que es muy fácil obtener ayuda, y tiene una fácil integración con servicios de terceros.

Aldo Alejandro Degollado Padilla

- C++, HTML, CSS y Visual Studio Code

Hola, soy Aldo Degollado, como ya me presentó mi compañero Abraham y en este caso me toca platicarles de 4 herramientas que son 3 lenguajes de programación y un IDE que nos ayudaron en nuestra realización de la situación problema 1. ¿Por qué hablo solamente de la situación problema 1? Bueno, aunque en esta materia sí existen dos situaciones problemas, hasta el día de hoy: 28 de mayo de 2021. Solamente hemos entregado la primera y por ende no podemos hablar de la que todavía no hacemos.

Y si hablamos de las actividades normales del curso, por así decirlo, nos damos cuenta de que estas en su gran mayoría fueron realizadas con lenguajes formales como lo son Skin o Prolog o con DFA's. Ahora les voy a hablar un poco de nuestro primer lenguaje de programación y el principal en nuestras actividades que fueron el analizador sintáctico y el resaltador sintáctico que fue C++, nuestro lenguaje columna, nuestro lenguaje principal, por así decirlo. Este lenguaje como pueden ver, las ventajas que tienes que es de alto desempeño, tienen un fácil manejo de memoria y es soportado en los IDE's más populares.

Nosotros utilizamos Visual Studio Code, pero si hubiéramos utilizado CLion o Atom, también ahí lo hubiéramos podido editar. Es un lenguaje bastante amigable en ese sentido. ¿Por qué escogimos ahora C++ más personalmente? Nosotros al principio estábamos pensando entre C++, Python y Java. A Java la verdad es que nunca lo habíamos tratado, pero como nos dijeron que en algún punto de la materia íbamos a tener que traducir de nuestro lenguaje, si es que no lo hacíamos con Java, a Java decidimos que a lo mejor sería una buena opción.

Al final, hace poco nos enteramos de que no iba a ser necesario traducirlo. Entonces, pues nos decidimos, por excluir a Java, porque realmente no es un lenguaje del que tengamos tanto conocimiento. De igual forma, Python sí es un lenguaje que hemos trabajado tanto mi equipo como yo, pero no lo habíamos tocado desde el primer semestre y C++ lo veníamos tocando en todos los semestres y lo seguimos tocando ahorita todavía. Y fue por eso que escogimos este. También fue porque nosotros ya tenemos un conocimiento amplio de las bibliotecas que tiene C++ y sabíamos más o menos cuáles íbamos a tener que utilizar.

Ahora les voy a hablar un poco de HTML y CSS, que comúnmente se ven como un solo lenguaje, por así decirlo, aunque no, o sea, es un conjunto que se complementan muy bien. Y pues las cualidades de estos lenguajes es que son de fácil aprendizaje, los documentos son más limpios porque puedes dividir tanto el contenido de la página web que quieres hacer en HTML con el estilo que quieres que tenga la página web en el CSS y que son compatibles con múltiples navegadores.

Esto lo primero es lo que acabo de mencionar. Nos permitió tener un código más limpio, más estructurado, mejor dividido. Y lo mejor de todo fue que, para el resaltador de sintaxis, que fue para el único que utilizamos estos dos lenguajes, nos ayudó a ver de una manera mucho más visual las palabras coloreadas que fue lo que hacía el resaltador. A las palabras clave las ponía de cierto color, errores de otro y cosas por el estilo. Y con el CSS eso era bastante sencillo, entonces nos facilitó mucho la tarea. Y por último, les voy a hablar del IDE que nosotros utilizamos, que es Visual Studio Code. Este IDE tiene soporte nativo para múltiples lenguajes de programación. Tienen una biblioteca de extensiones que nos fue de mucha utilidad y tienen una configuración de interfaz que también es buena, aunque no es algo que nos haya decantado por este IDE. Ahora, ¿por qué hablaba de la biblioteca de extensiones?

Bueno, en particular hubo dos extensiones que a nosotros nos funcionaron bastante bien, que fue la llamada Code Runner. En lo que nos ayudó fue que podíamos ejecutar el documento CPP sin tener que meternos al JSON para ejecutarlo en nosotros mismos y era con la ayuda de un simple botoncito que nos permitía ver el resultado de nuestro código. También utilizamos otra que se llamaba, o se llama más bien, Git Lens. ¿Para qué te sirve Git Lens? En cada línea de código, te pone después donde termina el texto del código.

Quién fue la última persona de ese repositorio que editó esa línea del código y cuándo fue. Entonces así nosotros podíamos ir viendo ah, mira, fulanito de tal editó esta parte del código y ya le metió tal cosa. Ah muy bien. Entonces eso ya no lo tengo que hacer. O en su defecto al revés, ¿no? Y eso nos ayudó bastante y pues bueno, estas fueron tres de las herramientas más importantes que nosotros utilizamos. Espero les haya gustado y eso fue todo. Muchas gracias.