



# INSTITUTO POLITECNICO NACIONAL

## ESCUELA SUPERIOR DE CÓMPUTO (ESCOM)



### COMPILADORES

---

#### PROYECTO:

- DEMO "LOGOS" MANUAL DE USUARIO

#### INTEGRANTES:

- CASTAÑEDA YESCAS LUIS CARLOS
- LOREDO CORTÉS LUIS JOSUE
- SANTOS MÉNDEZ ULISES JESÚS
- VEGA ALVAREZ BRYAN ALBERTO

#### NOMBRE DEL MAESTRO:

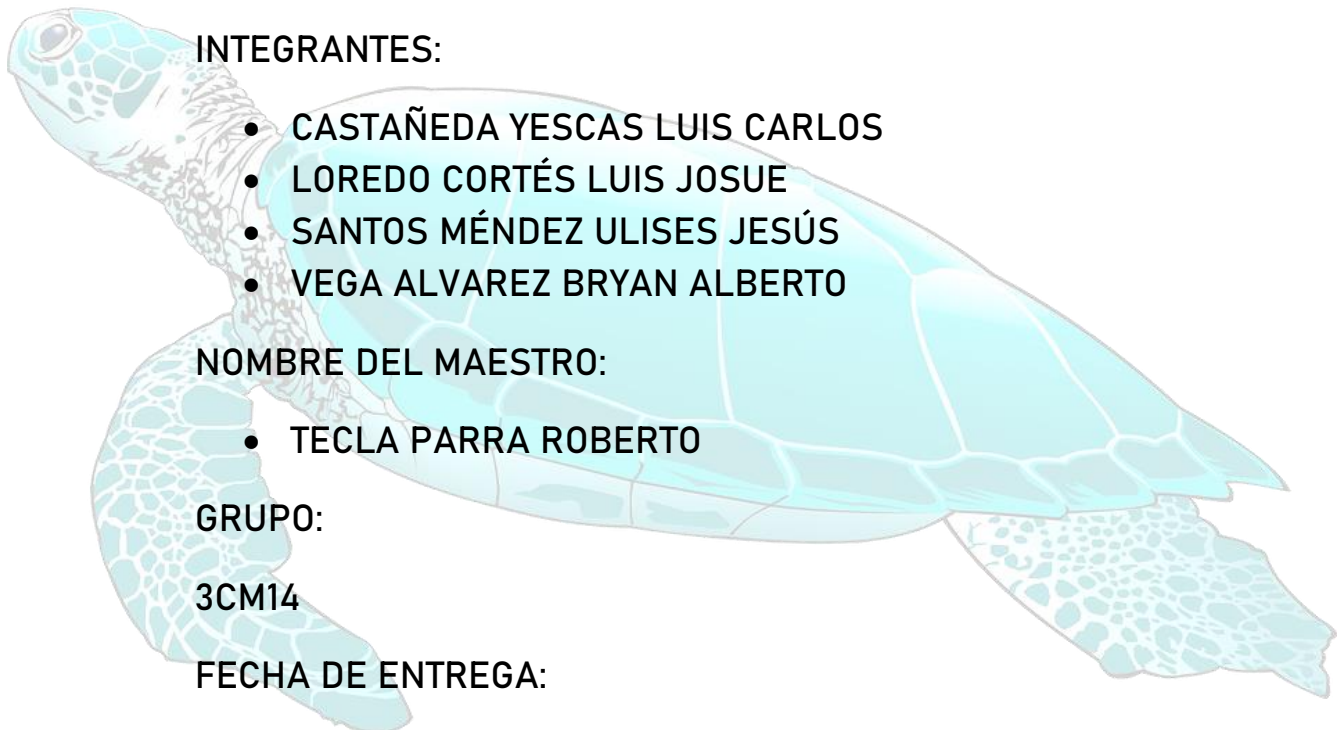
- TECLA PARRA ROBERTO

#### GRUPO:

3CM14

#### FECHA DE ENTREGA:

- 21/06/2023



## INTRODUCCIÓN:

El lenguaje de programación Logo, desarrollado en la década de 1960 en el Instituto de Tecnología de Massachusetts (MIT), se diseñó específicamente para enseñar a programar a niños y jóvenes. Fue creado por Seymour Papert, Wally Feurzeig y Cynthia Solomon, quienes buscaban desarrollar un entorno de programación accesible y amigable para los niños.

El enfoque principal de Logo era brindar a los niños una herramienta para explorar conceptos de programación y desarrollar habilidades de pensamiento computacional a través de la resolución de problemas prácticos. Logo utilizaba una interfaz gráfica en la que los niños podían dar instrucciones a una "tortuga" virtual que se movía en la pantalla y dibujaba formas.

La idea central de Logo era fomentar la creatividad y la experimentación. Los niños podían programar la tortuga para que dibujara figuras geométricas, patrones y diseños, lo que les permitía visualizar y comprender los conceptos de programación, como el control de flujo, la iteración y las estructuras de datos.

Logo se basaba en un lenguaje de programación de alto nivel con una sintaxis sencilla y comprensible. Los comandos eran expresados en forma de palabras y frases en inglés, lo que facilitaba su comprensión y uso para los niños.

Además del aspecto gráfico, Logo también permitía a los niños escribir programas utilizando instrucciones de texto. Esto les brindaba una mayor flexibilidad y capacidad de expresión en la programación.

Logo se convirtió en una herramienta popular en las escuelas y se utilizó para enseñar una variedad de conceptos, desde lógica y matemáticas hasta resolución de problemas y pensamiento algorítmico. Su enfoque en la experimentación y la creatividad ayudó a los niños a desarrollar habilidades cognitivas y de razonamiento, así como a fomentar su interés en la informática y la programación.

El desarrollo del presente proyecto está enfocado en que sea posible enseñar a programar a personas con pocos conocimientos de una forma divertida, esto será con ayuda de una interfaz gráfica amigable que le permita ver las instrucciones, así como meter cualquier comando que sea válido y que le permita dibujar cualquier figura que se le pueda ocurrir.

## OBJETIVO:

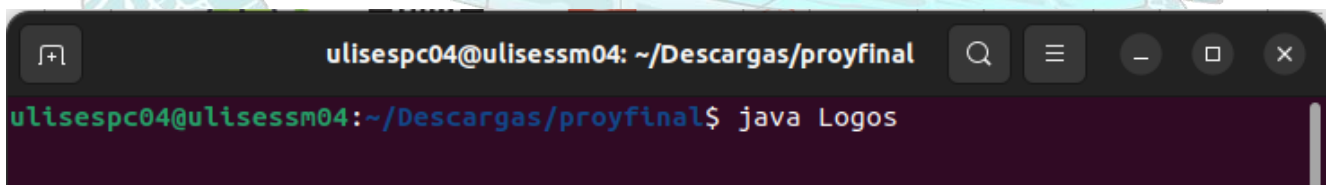
- El objetivo de este trabajo es dar referencia técnica acerca del proyecto para que pueda ser entendido por alguna persona con los conocimientos técnicos aptos y así poder ser modificado o incluso optimizado.
- Tener una GUI que permita al usuario manejarla de forma intuitiva y fácil, lo suficiente para aprender de una forma rápida.

## REQUERIMIENTOS TÉCNICOS:

- Java.
- Java Development Kit (JDK).
- Java Runtime Edition (JRE).
- Byacc.
- 4 GB RAM.
- Disco duro de cualquier capacidad.
- Windows/Linux

## EJECUCIÓN:

Lo primero que se hará será abrir una terminal en Linux donde se encuentren los archivos correspondientes al proyecto, una vez que se ha abierto la terminal se ingresa el comando Java Logos, solo se deberá de ingresar un comando por el mismo motivo se les esta entregando una carpeta ya compilada aún así si hubiera algún cambio a futuro basta con ejecutar `Javac *.java` para obtener las clases y finalmente ejecutamos Java Logos



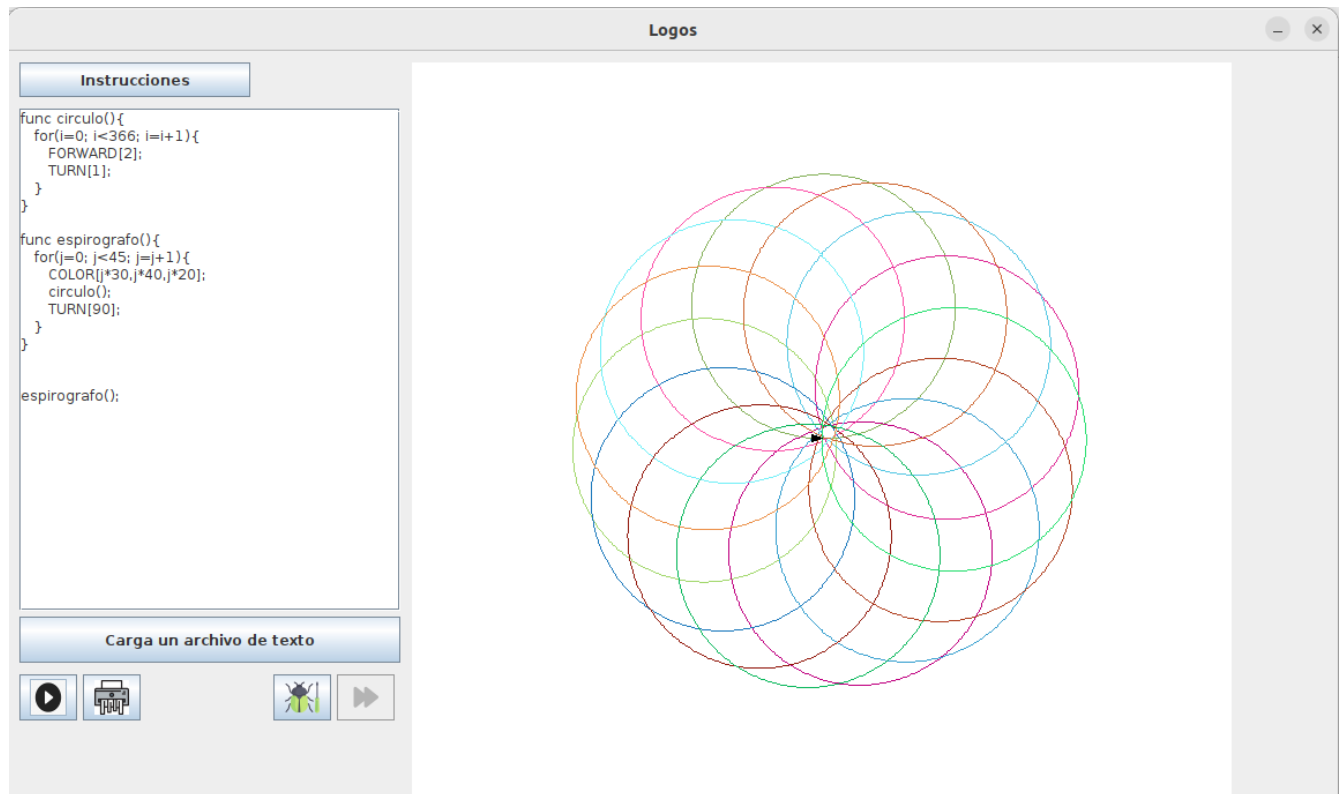
Ahora que está en ejecución el programa, y principalmente consta de un botón para consultar las instrucciones, un área para pegar o ingresar tu propio código, un botón para realizar la inserción de un archivo .txt donde venga tu código, un botón de Inicio, Borrar, Debug y cuando se active el debug te dejará verlo paso a paso con el botón de avance.

Una vez que se ejecuta el programa tendremos la interfaz gráfica mostrada en la Imagen 5, en donde

tenemos los siguientes componentes:

- Un área de texto, donde se ingresan los comandos del lenguaje.
- Un botón que permite dibujar las acciones descritas por los comandos.

- Un botón que permite borrar lo que está dibujado en el panel y nos permite realizar el botón del siguiente punto, si fue seleccionado anteriormente y se le vuelve a seleccionar regresamos se nos deshabilita la opción del botón siguiente y guarda el estado en el panel lo que hicimos mientras lo seleccionamos por primera vez.
- Un botón que dibuja paso a paso la figura.
- Un panel, donde se visualiza el dibujo.



## COMANDOS BÁSICOS :

Las siguientes son expresiones regulares:

- FORWARD[n];
- TURN[n];
- COLOR[R, G, B];
- PenUP[];
- PenDOWN[];
- for(exploración inicial; condición; expresión final){}
- while(condición){ declaracion(es)}

- `if (condición){instrucciones}else{instrucciones} proc [nombre](){Bloque de instrucciones}`  
\*\*Para usar parámetros usar \$n, donde n es el parámetro para usar\*\*, en las llamadas los parámetros se separan con una ','.
- `func [nombre](){ Bloque de instrucciones return valor; }` \*\*Para usar parámetros usar \$n, donde n es el parámetro para usar\*\*, en las llamadas los parámetros se separan con una ','.

### Condiciones disponibles:

- `==`, comparación, entre enteros
- `!=`, diferente, entre enteros
- `<=`, operación menor o igual, entre enteros.
- `>`, operación mayor, entre enteros.
- `>=`, operación mayor o igual, entre enteros.
- `&&`, operación "and", entre enteros. `||`, operación "or", entre enteros.

### EXPLICACION DE ALTO NIVEL

El lenguaje proporciona instrucciones secuenciales, decisiones, ciclos, funciones y procedimientos. A través de esas instrucciones y junto con los comandos básicos anteriormente descritos se pueden realizar dibujos más complejos, tales como espirales, reglas, árboles, fractales, entre otros.

