



## Los 7 Fantásticos

# Lenguajes y Autómatas I

*Tema 6. Máquinas de Turing.*

*Actividad 5. Presentación y Entrega Final del Proyecto*

*Docente. Dr. Francisco Ibarra Carlos*

*Integrantes:*

Kevin Arturo Castellanos Amador – 19400547

Janeth Andrea Guillén de la Mora - 19400585

Axel Ulises Paredes Rubio - 19400642

Enciso Luna Cristian Gerardo – 19400563

Dominguez Herrera Denilson de Jesús - 19400559

Juan Carlos Nuñez Trigueros - 19400630

Axel Michell Ventura Almeida Montaña - 19400530

**Grupo 5A**

**Enlace del video:** <https://youtu.be/lMBLXX5aCwQ>

**Enlace del repositorio de GitHub:** [https://github.com/AxelUlises/LYA\\_CAFETERA\\_INT.git](https://github.com/AxelUlises/LYA_CAFETERA_INT.git)

Lenguajes y Autómatas I, Actividad 5, Unidad 6: Máquinas de Turing

Docente: Dr. Francisco Ibarra Carlos

ISC, Instituto Tecnológico de Tepic

## 1. Preliminares

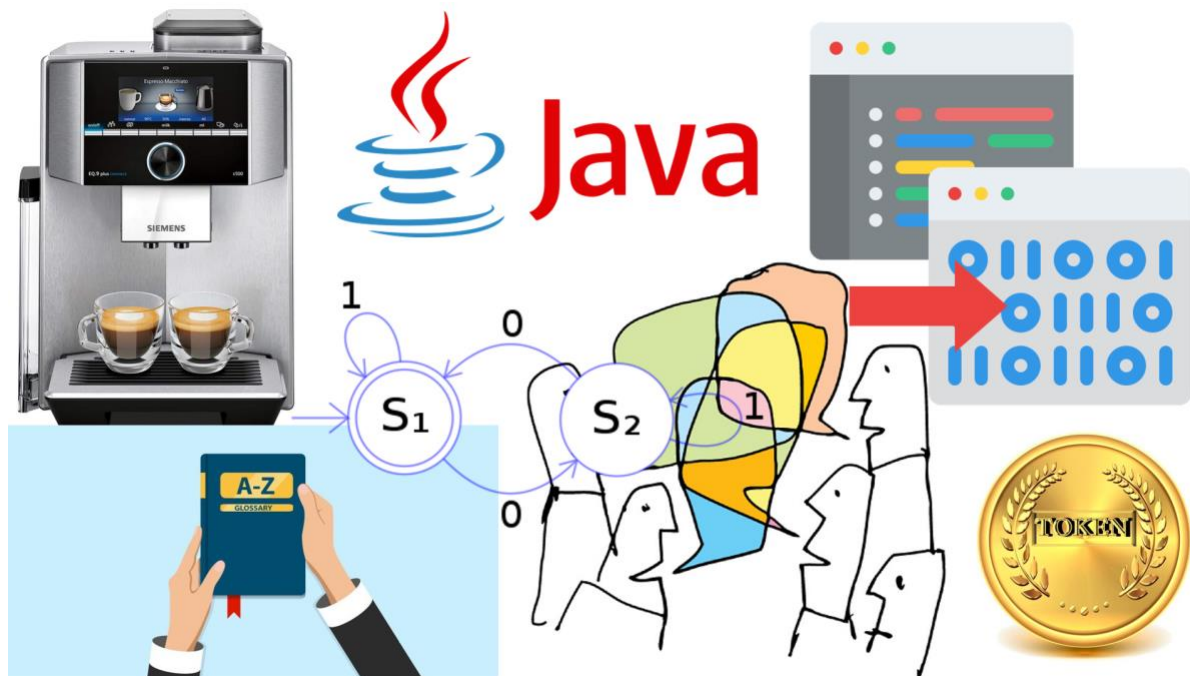
### 1.1 Agradecimientos

Agradecemos al Dr. Francisco Ibarra Carlos, por impartir al grupo la materia de Lenguajes y Autómatas I con la que con el aprendizaje pudimos realizar este proyecto, nos proporcionó información para implementarla en éste y contamos siempre con su apoyo. (metele mas labia papi)

### 1.2 Resumen

En el presente documento se explica el desarrollo de un lenguaje enfocado al tema del café, sobre todo proponer y solucionar problemáticas relacionadas con las tecnologías de las cafeteras de hoy en día dándole un toque más interactivo con el usuario.

### 1.3 Colash de imágenes



## 2. Generalidades del Proyecto

### 2.1 Introducción

This document presents the information on the development of the "Self Sufficient-Coffee Maker" project, this project is focused on the realization of a coffee maker which helps to manage processes of preparing beverages typical of establishments known as cafeterias, being examples of these (Lates, Coffee, Tea, Frappes, etc). Dicha gestión se enfoca en la

reducción de tiempos de preparación en un ámbito comercial y la automatización de la preparación de dichos productos en un ámbito personal, implementando algunas funciones que puedan facilitar el uso de la misma cafetera, creando instrucciones de voz que automaticen las funcionalidades de la cafetera y haga el uso de esta más intuitivo para el usuario final.

## 2.2 Antecedentes

Actualmente existen dos tipos de cafeteras, “máquinas de café expreso” y “máquinas para café americano”.

Within each of them there are several types, for example in espresso coffee machines there are semi-automatic, automatic and super-automatic ones that are used to make espresso-based drinks (espresso, cappuccino and coffee latte).

Mientras que en las de americano están las de termos, jarras y single serve pero sólo preparan café americano.

Esto antes mencionado se basa en la industria de los establecimientos de café, pero ¿qué hay de las cafeteras para los hogares?. Hoy en día existen cafeteras inteligentes, pero que se limitan a un par de tipos de café, las cuales alguna de sus únicas funciones son poder programar una hora exacta en la que tiene que preparar el café, sin embargo, el usuario tiene que estar en constante revisión de los niveles de agua y café. Lo que puede generar un problema o retraso en un día distraído.

Con lo cual implementando sensores podemos recordarle al usuario con tiempo sobre el relleno de los insumos principales. De esta forma hacemos de la practicidad una forma nueva de interactuar con el usuario.

## 2.3 Objetivos

### 2.3.1 Objetivo General

Develop a self-sufficient coffee maker through the implementation of various automation technologies and control by voice commands, to reduce time in the commercial sector and motorization in the personal sphere.

### 2.3.2 Objetivos Específicos

1. Proporcionar un proceso de preparación sencillo para el cliente final.
2. Reducir tiempos de producción y automatización de los procesos de creación.
3. Informar al usuario detalles sobre su bebida como la temperatura, tipo de café, complementos a agregar y tiempo que requerirá su preparación.
4. Proporcionar al usuario la opción de programar la preparación de de una bebida preferida de forma automática a una hora determinada del día.

## 2.4 Justificación

Uno de los problemas al seguir comprando café en establecimientos definidos es la gran acumulación de vasos de plástico, como tapas, bastidores y sobres de algún complemento como azúcar o cualquier otro endulzante, teniendo un impacto ambiental considerable. A eso sumándole el tiempo invertido al ir a un establecimiento por un café, lo que puede llegar a no ser muy factible para alguien con el tiempo limitado, como estudiantes, oficinistas, empresarios entre otras ramas más.

El uso de una cafetera personal puede ayudar a reducir la contaminación en vasos, tapas, sobres de algún complemento, puesto que el usuarios reutilizar diferentes contenedores para esta bebida, como el uso de termo, tazas de cerámica, entre otras opciones más que hagan de esta una experiencia más personalizada. Es por esto que la opción de tener una cafetera inteligente podría agilizar el día a día de muchas personas que necesitan de esta bebida para empezar su día productivo.

Since coffee is a drink with too much demand and attractiveness in the daily life of many people, it is included at breakfast, before going to work or during the academic day.

## 3. Marco Teórico

Las primera cafeteras:

Desde el siglo XV hasta el XVII, en Turquía, el café era tratado como una infusión más y por tanto, se preparaba en un hervidor con una jarra de metal. Se llamaba el hervidor de Bagdad. La primera cafetera como tal, se inventó en Francia el año 1802 y su precursor fue un farmacéutico llamado Francois Antoine Descroisilles que unió dos recipientes metálicos que se comunicaban con una chapa llena de agujeros, como un colador. Al ponerlo a hervir, el vapor que libera el agua subía hacia el depósito de arriba pasando por el filtro y a través del café molido, extrayendo su esencia, color y sabor.

Siglo XX: la evolución de las cafeteras

En el comienzo del siglo XX, concretamente en 1901, Luigi Bezzera, un empresario italiano, fabricó la primera cafetera industrial y cuatro años después, vendió la patente a Desiderio Pavoni, que la comercializó en bares y restaurantes. Así fue como se creó la cultura del café de bar que ha llegado hasta nuestros días.

La siguiente innovación en las cafeteras domésticas de la época surgió de Melitta Benz, una ama de casa alemana que en 1908, creó el primer filtro de café. Su idea tuvo tanto éxito que fundó una compañía y ésta llegó a sobrevivir a la primera y segunda guerra mundial siendo hoy en día una empresa con más de 3.000 empleados.

En 1992, De'Longhi desarrolló la cafetera Bar, su primera cafetera espresso, que sería la precursora de la marca para llegar hasta el día de hoy y convertirse en los líderes del mercado en cafeteras espresso.

## Entrega Final del Proyecto

La innovación de las cafeteras, hoy.

Desde que se empezaron a comercializar las cafeteras domésticas, éstas han evolucionado en diferentes tipos y formas de preparar el café. Cafeteras tradicionales italianas, cafeteras de cápsulas, de bomba tradicional, superautomáticas con molinillo de café integrado...

En poco más de 2 siglos, las cafeteras han evolucionado tanto que incluso hoy en día, existen modelos que pueden prepararte un café perfecto desde una app móvil.

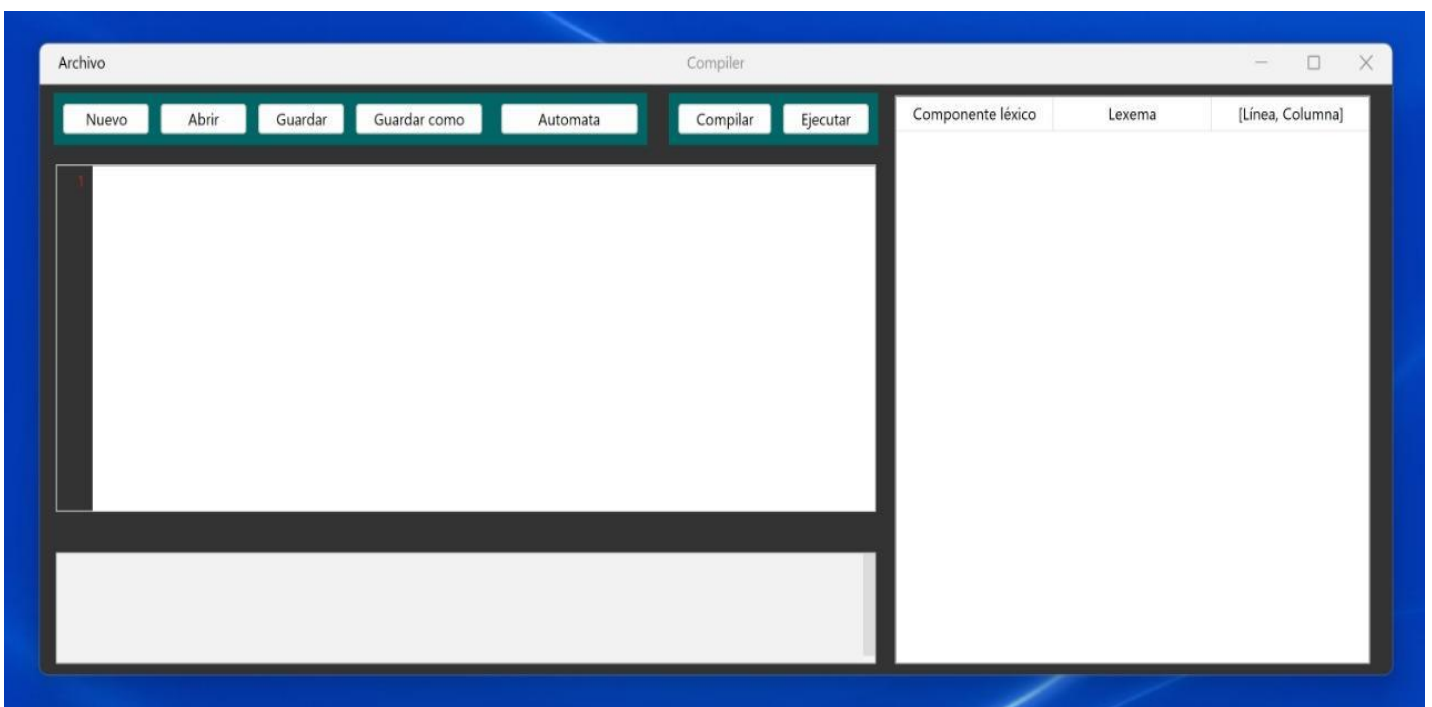
Pero nuestro proyecto se enfoca en llevar un paso más adelante a este tipo de electrodoméstico haciendo que encaje y vaya de la mano con el significado de autonomía y simplicidad ayudando en un enfoque comercial y personal donde no se requiera de intervención manual por parte de quien la maneja y pueda preparar diferentes tipos de bebida propias de establecimientos denominados (cafeterías).

## 4. Desarrollo

### 4.1. Manual de usuario

Pantalla principal:

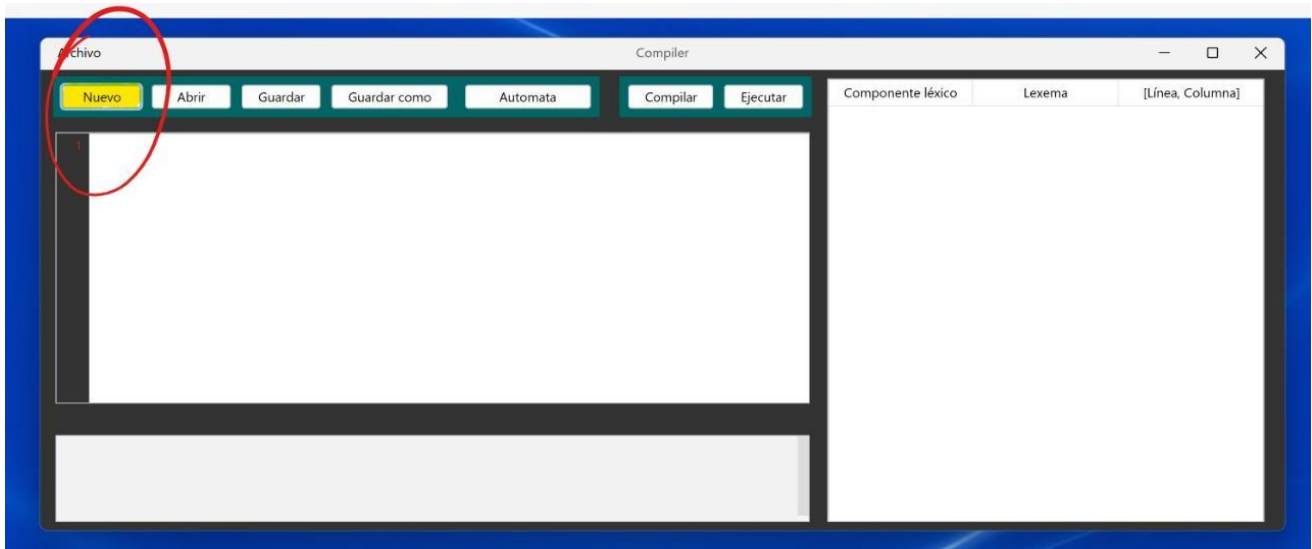
- Al ejecutar el programa nos aparecerá la pantalla principal del compilador, en la cual se verán todos los componentes que ayudan a utilizar nuestra herramienta de programación.



## Entrega Final del Proyecto

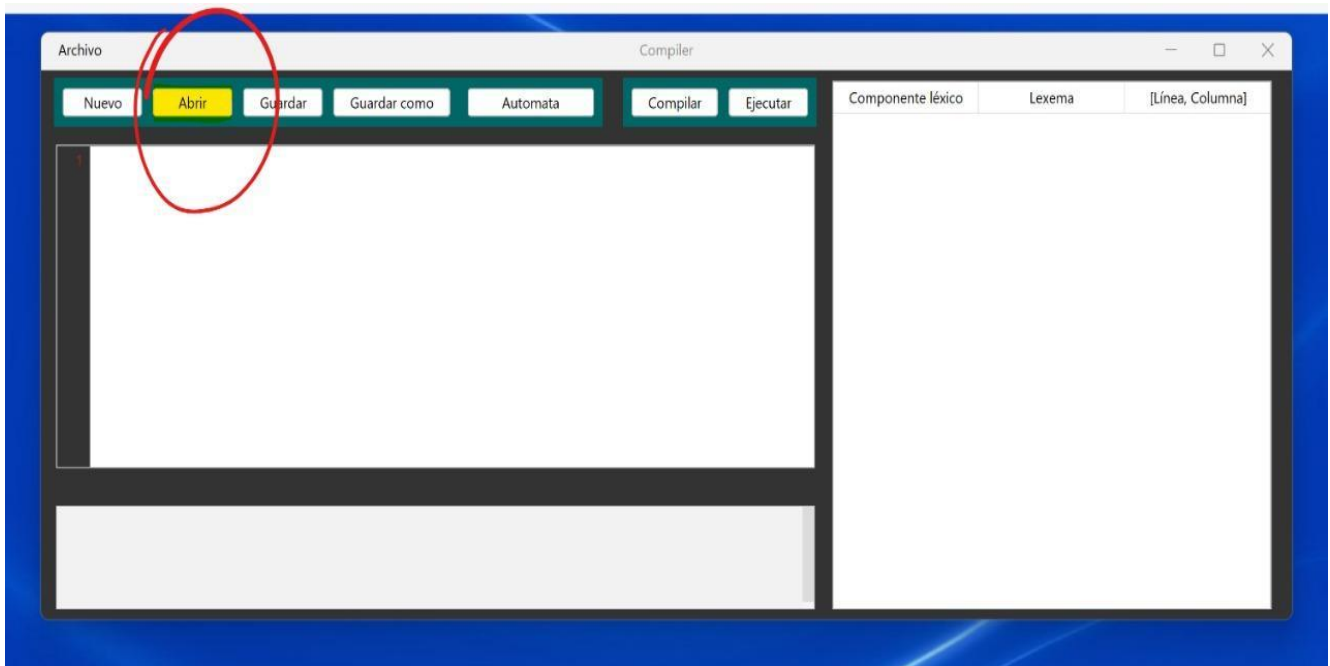
Botón nuevo:

- Este botón nos proporcionará una ventana en limpio para poder generar código dentro del compilador.



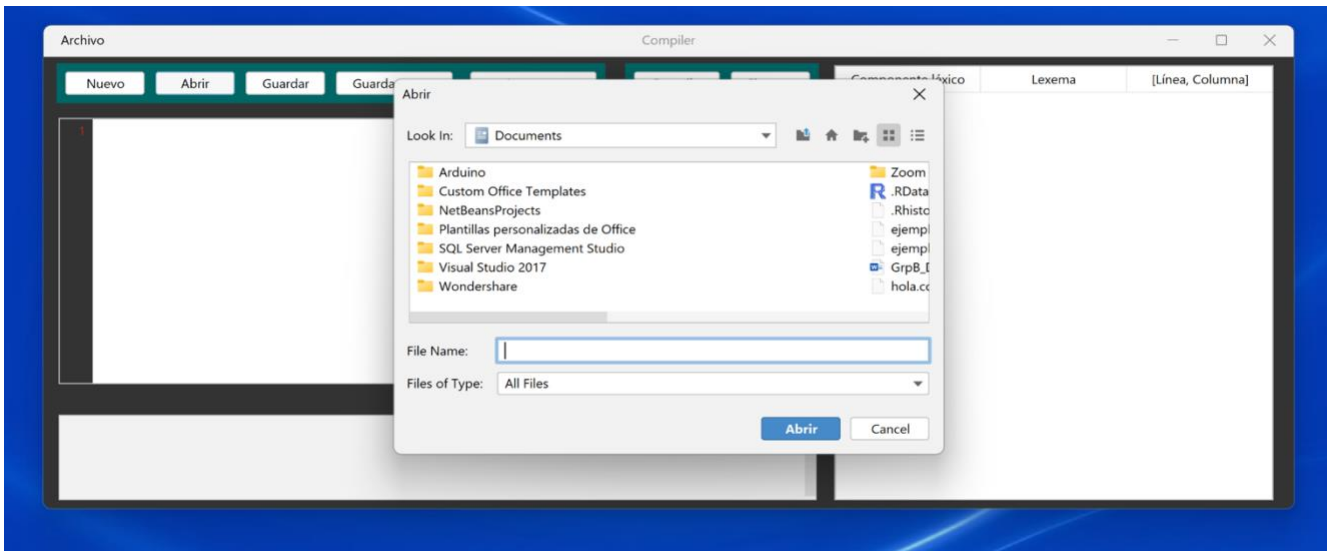
Botón abrir:

- Al dar clic en dicho botón nos mostrará los archivos que tengamos guardados dentro de nuestra computadora para así poder hacer uso de ellos.



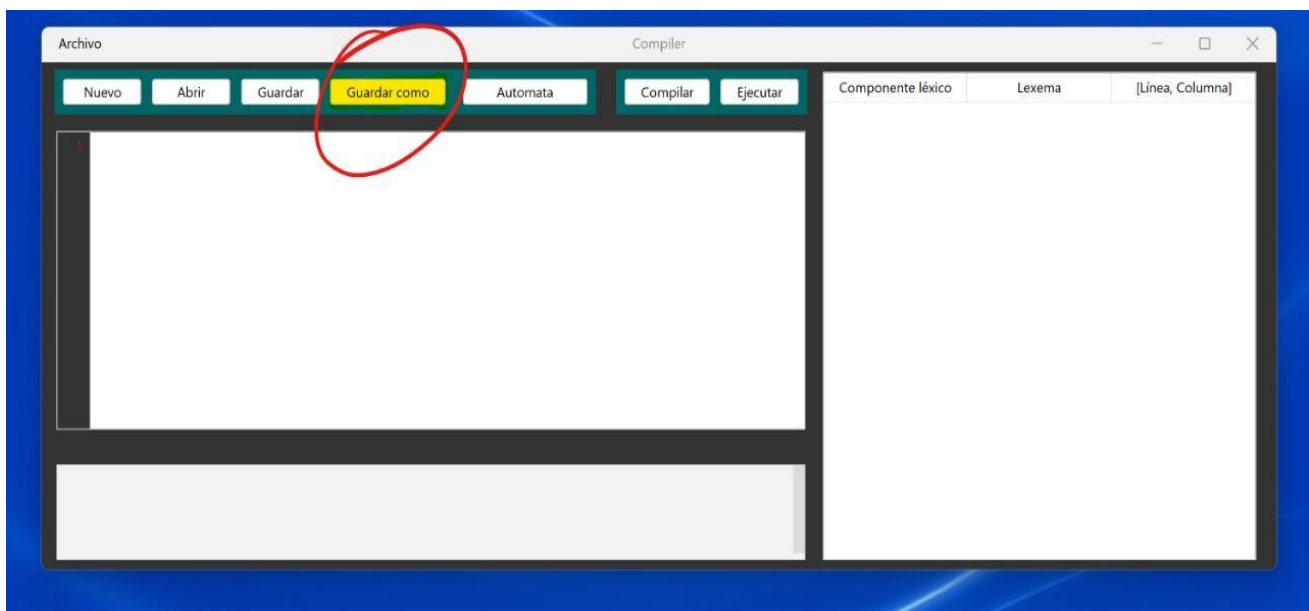
## Entrega Final del Proyecto

- En esta pestaña se accede directamente a los archivos del sistema, para seleccionar un archivo creado para editar en el compilador.



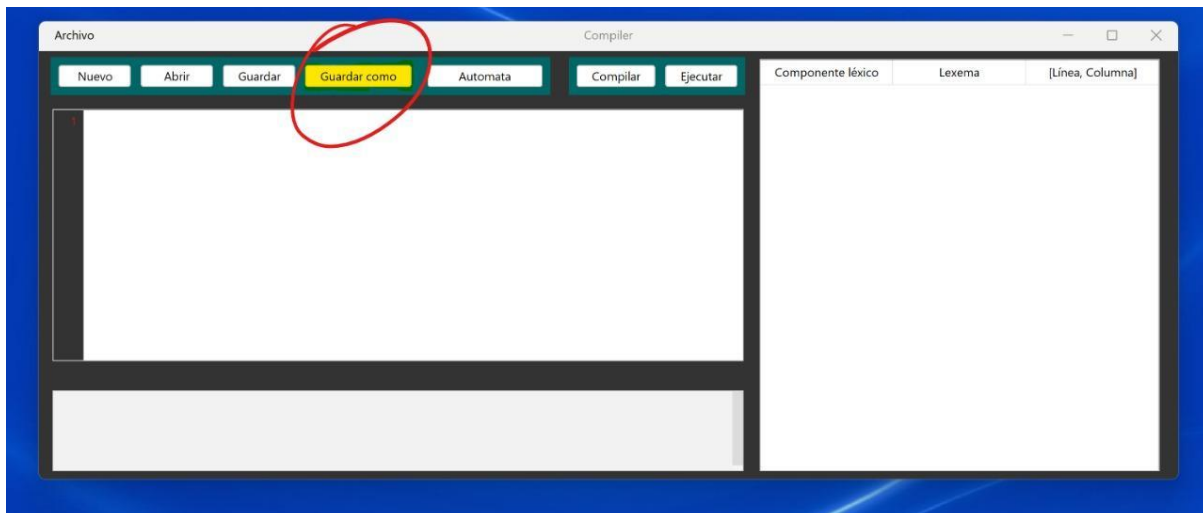
## Botón guardar/ guardar como:

- Al momento de seleccionar este botón nos guardará las modificaciones realizadas en el código y si dicho archivo no está guardado dentro de nuestra computadora, tendríamos que realizar la acción con el botón guardar como.

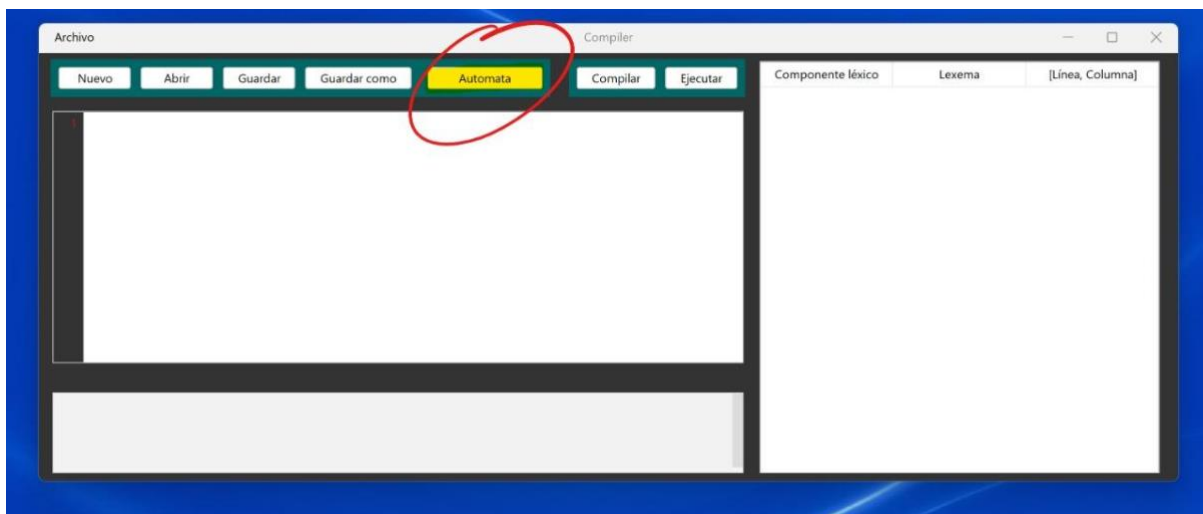




## Entrega Final del Proyecto



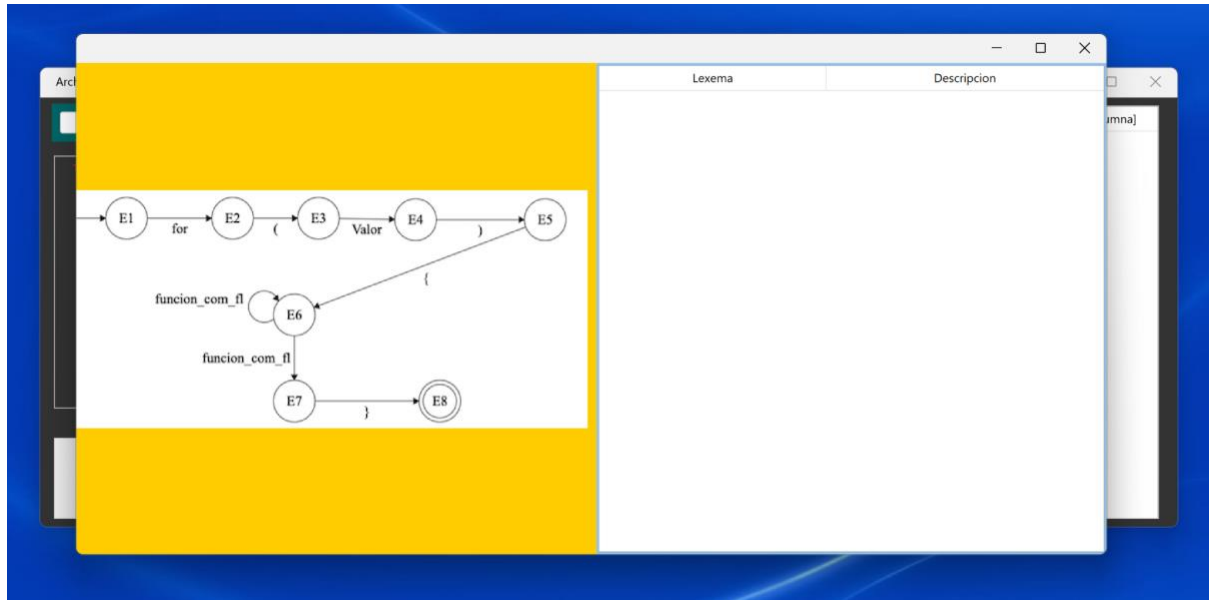
Botón autómeta:



- Al momento de seleccionar este botón se ejecutará un ventana emergente con la imagen de un autómeta arbitrario. también se muestra una tabla de explica el lexema de cada nodo y la descripción de la ruta que recorre dentro del autómeta expuesto.
- *NOTA: si no se ejecuta el codigo que esta predeterminado de manera arbitraria dentro del programa, la tabla aparecerá en blanco.*

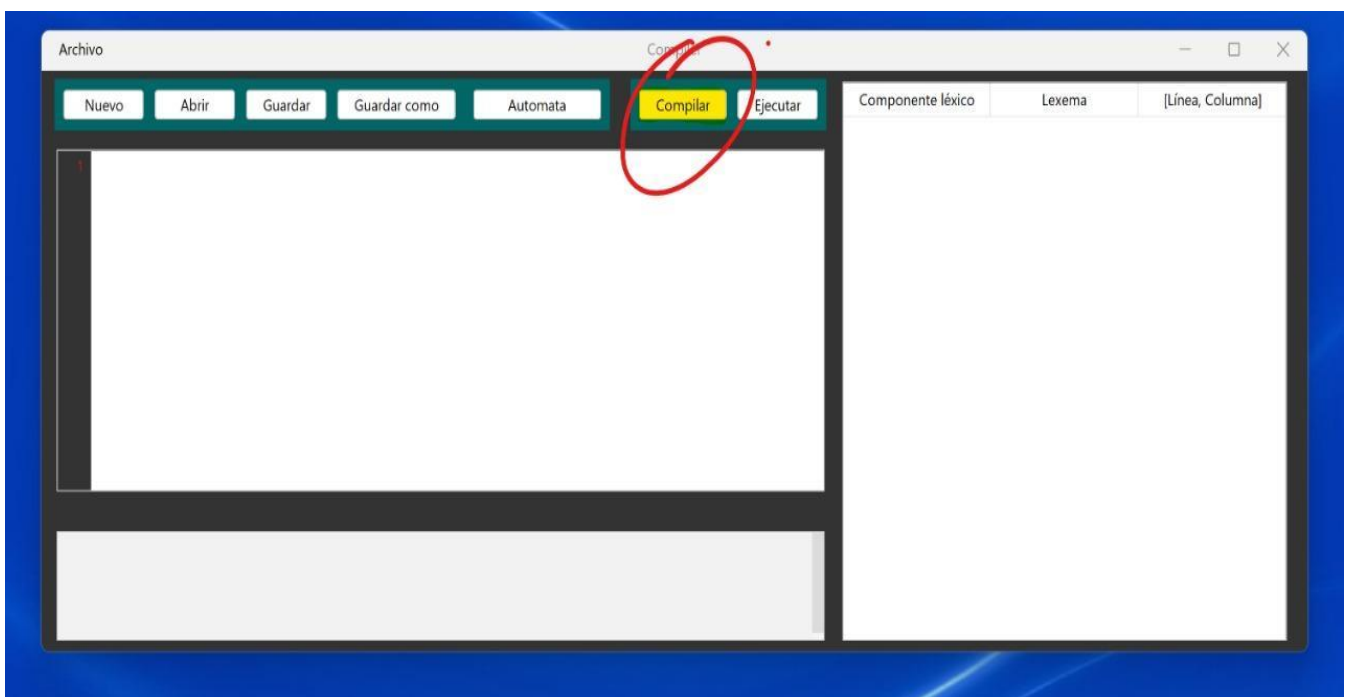


## Entrega Final del Proyecto



### Botón compilar:

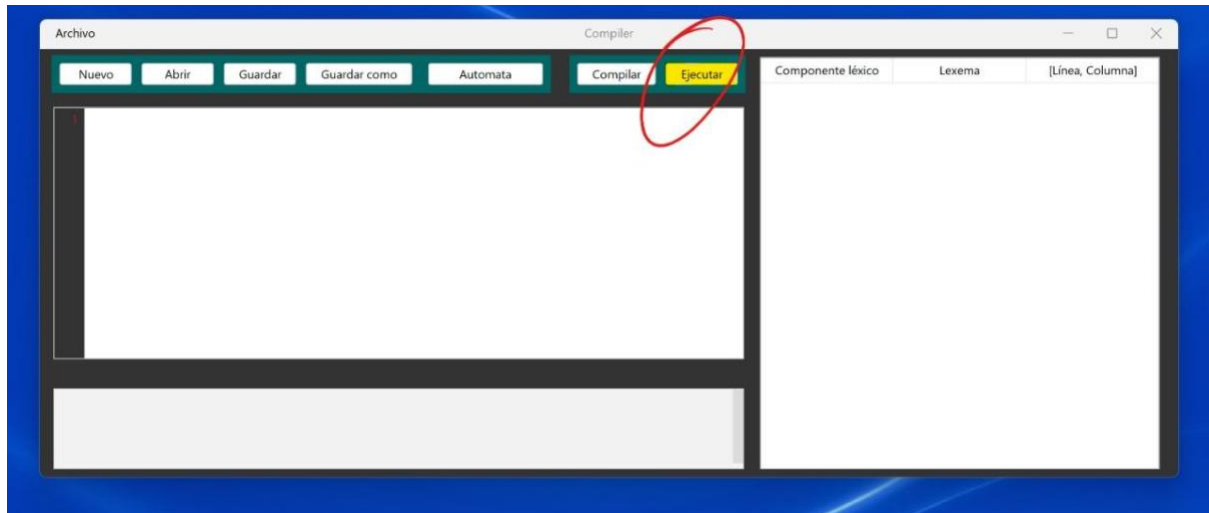
- Como su nombre lo dice al momento de dar clic en compilar el código que el usuario haya tecleado será compilado para posteriormente ejecutar y analizar si ocurrió algún error dentro de lo escrito.



## Entrega Final del Proyecto

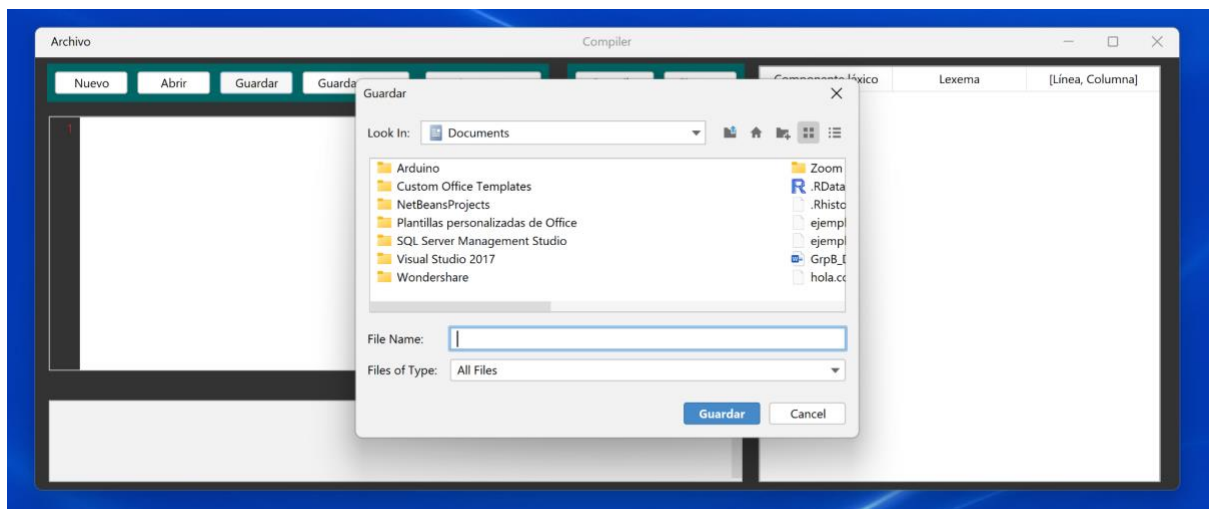
Botón ejecutar:

- Al momento de dar clic en ejecutar el código que el usuario haya tecleado será ejecutado para posteriormente analizar si ocurrió algún error dentro de la ejecución.



Dentro del botón “Guardar” o “Guardar Como” nos aparecerá la siguiente ventana emergente en la cual nos mandara a guardar nuestro archivo de código dentro de los archivos del sistemas y las carpetas existentes de la computadora en uso.

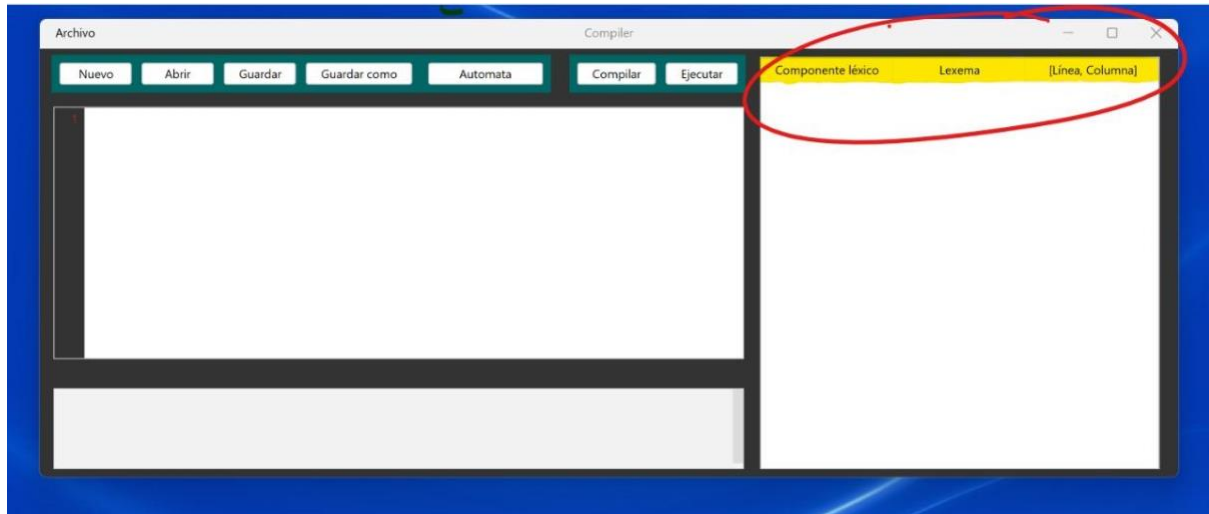
Esta ventana también puede aparecer al presionar los botones de “Compilar” o “Ejecutar” si el archivo que se está manipulando no se ha guardado previo a esto.



## Entrega Final del Proyecto

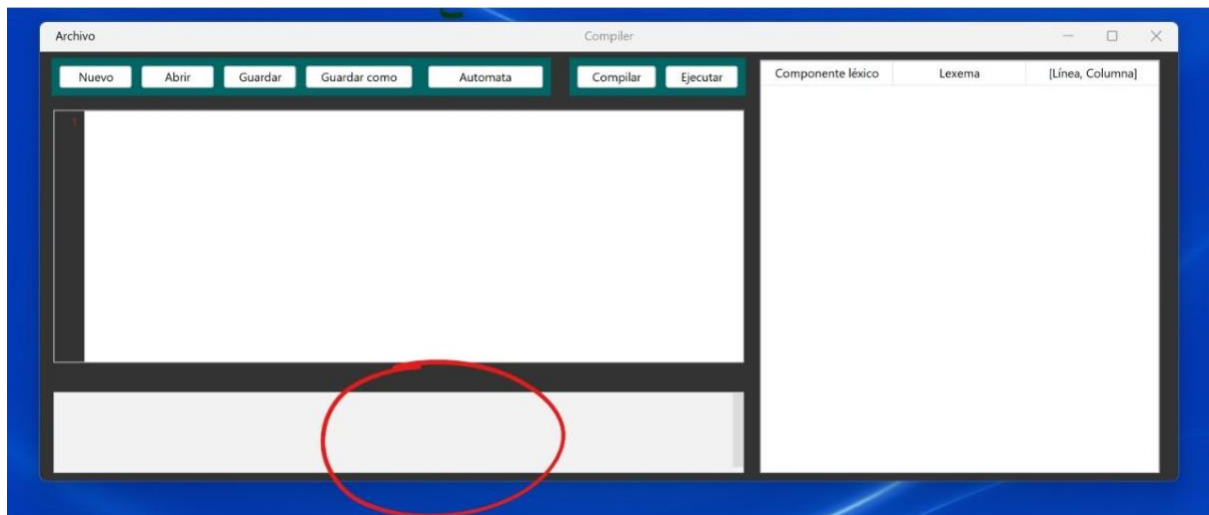
### Tabla de errores:

- En este apartado se mostrarán los errores de manera organizada, obteniendo su componente léxico, lexema y dentro de que línea y columna se dio dicho error.



### Consola de errores:

- En este apartado se mostrarán los errores que se realizaron al momento de ejecutarse el código y nos dará una breve descripción de ellos.



## 4.2. Manual técnico

### 4.2.1. Requerimientos técnicos

- Procesador Intel Pentium III
- Memoria RAM 4 GB
- Disco duro 2 GB
- Tarjeta gráfica GPU integrada en CPU

### 4.2.2. Requerimientos de software

- Sistema operativo Windows 10

### 4.2.3. Herramientas utilizadas

- Netbeans
- Java

### 4.2.4 Como Instalar

- Descargar netbeans, se recomienda la versión más actualizada
- Descargar la carpeta del proyecto que se encuentra en github (poner el enlace)
- Bajar las librerías necesarias para poder ejecutar el programa sin errores
- Descomprimir la carpeta del proyecto y guardar en un destino de fácil acceso o deseado por el usuario
- Abrir el proyecto desde netbeans, damos clic en archivo, abrir proyecto y posteriormente buscamos el proyecto donde lo hayamos guardado.

## 5. Resultados

Tenemos como resultados la creación de nuestro compilador basado en nuestro propio lenguaje de programación, así como la interfaz de este donde podemos programar y ver todo acerca de nuestro proyecto.

Además de la interfaz ya funcional, con sus gramáticas, autómatas, la posibilidad de la compilación y ejecución de algún programa previamente realizado y la posibilidad de ver los errores, tabla de tokens, etc.

## 6. Conclusión

Al terminar la realización de esta práctica nos queda como enseñanza los subtemas que vimos durante todo el semestre en un ámbito profesional, el cómo se implementan las gramáticas en una tecnología como ejemplo, como la creación de un autómata. Tener las bases teóricas de estas nos facilitaron el poder implementarlo en nuestro proyecto.

Nos deja como enseñanza el cómo es trabajar en equipo al momento de desarrollar algo, ya que el hacer fluir en armonía las dudas y métodos de trabajo de cada integrante es un gran reto, que en un mundo laboral es el día a día de cualquier ingeniero informático.

## 7. Glosario de términos

### Token:

Los tokens son las palabras reservadas de un lenguaje, secuencia de caracteres que representa una unidad de información en el programa fuente. En cada caso un token representa un cierto patrón de caracteres que el analizador léxico reconoce, o ajusta desde el inicio de los caracteres de entrada.

### Patrón:

Un patrón es una descripción de la forma que pueden tomar los lexemas de un token. En el caso de una palabra clave como token, el patrón es sólo la secuencia de caracteres que forman la palabra clave. Para los identificadores y algunos otros tokens, el patrón es una estructura más compleja que se relaciona mediante muchas cadenas.

### Lexema:

Un lexema es una secuencia de caracteres en el programa fuente, que coinciden con el patrón para un token y que el analizador léxico identifica como una instancia de ese token.

### Alfabeto:

Un alfabeto es un conjunto finito no vacío, cuyos elementos se denominan letras o símbolos. Denotamos un alfabeto arbitrario con la letra  $\Sigma$ .

### Símbolos:

Es una entidad abstracta que no se puede definir, ya que se dejaría como una axioma. Igual que se define un punto en la geometría. La cual normalmente los símbolos son letras (a, b, c,... z), dígitos (0,1,...9, caracteres (+, -, \*, /,>,< .....). los símbolos pueden estar formados por varias letras o caracteres.

### Cadena:

Es una secuencia ordenada de elementos que pertenecen a un cierto lenguaje formal o alfabeto análogas a una fórmula o a una oración.

### Concatenación de cadenas:

La concatenación de dos cadenas u y v, escrita uv, es "pegar" las dos cadenas para formar una nueva.

### Carácter:

Es una unidad de información que corresponde aproximadamente con un grafema o con una unidad o símbolo parecido, como los de un alfabeto o silabario de la forma escrita de un lenguaje

### Lenguaje:

Es un conjunto de cadenas, de todas las seleccionadas de un  $\Sigma^*$ . donde  $\Sigma$  determinado el alfabeto se denomina lenguaje. Si  $\Sigma$  es un alfabeto y  $L \subseteq \Sigma^*$ , entonces L es un lenguaje de  $\Sigma$ .

## Entrega Final del Proyecto

### Lenguaje fuente:

Es el código escrito en algún lenguaje de programación previo a ser interpretado o compilado.

### Lenguaje universal:

cualquier sistema de signos que permite componer e interpretar mensajes.

### Clase:

En informática, una clase es una plantilla para el objetivo de la creación de objetos de datos según un modelo predefinido. Las clases se utilizan para representar entidades o conceptos, como los sustantivos en el lenguaje.

### Jflex:

JFlex y CUP, son herramientas que reaccionan a una entrada de datos con una estructura y un lenguaje predeterminado, el cual se le da a conocer previamente para poder trabajar con los datos de entrada.

### Af:

Un autómata finito o máquina de estado finito es un modelo computacional que realiza cálculos en forma automática sobre una entrada para producir una salida.

### Gramática:

Una gramática desde el punto de vista de la teoría de autómatas es un conjunto finito de reglas que describen toda la secuencia de símbolos pertenecientes a un lenguaje específico L.

### Expresión regular:

Es una cadena de caracteres que es utilizada para describir o encontrar patrones dentro de otras cadenas, en base al uso de delimitadores y ciertas reglas de sintaxis.

### Operador:

Un operador es un signo o símbolo que especifica el tipo de cálculo que se realiza en una expresión.

### Operador aritmético:

Los operadores aritméticos realizan operaciones matemáticas, como sumas o restas con operandos.

### Operador lógico:

Los operadores lógicos permiten combinar más de una prueba relacionar en una comparación. Los operadores lógicos devuelven el valor TRUE (1) o FALSE (0). Los operadores lógicos tienen una menor precedencia que los operadores aritméticos.

### Operador relacional:

Los operadores relacionales son: < Menor que, > Mayor que, <= Menor o igual que, >= Mayor o igual que.