



**TECNOLÓGICO  
NACIONAL DE MÉXICO®**



## **TECNOLÓGICO NACIONAL DE MÉXICO** **INSTITUTO TECNOLÓGICO DE TEPIC**

### **ING. EN SISTEMAS COMPUTACIONALES LENGUAJES Y AUTOMATAS 1**

SCD - 1015

Grupo: 5A

### **Unidad 6: Maquinas de Turing**

## **Actividad 5. Presentación y Entrega Final del Proyecto.**

**Docente:** Dr. Francisco Ibarra Carlos

### **Equipo 3**

#### **Integrantes:**

Barocio Contreras Alan	18401084
Cantabrana Esquivel José Maximiliano	18401092
Carrillo Sánchez Luis Sergio	18401095
Flores Ríos Manuel	18401115
Ramos Montes Christopher Jesús	18401180
Serrano Nuño Sergio Armando	18401083
Zurita Bizarrón Andrés Antonio	17401335

Video:

[https://drive.google.com/drive/folders/1DVXgUBkDu\\_wqKWwYPgVzoK7fBXUx9aH8?usp=sharing](https://drive.google.com/drive/folders/1DVXgUBkDu_wqKWwYPgVzoK7fBXUx9aH8?usp=sharing)

Repositorio: [https://github.com/SergioSerrano30/LyA\\_G5A\\_E3\\_Compilador](https://github.com/SergioSerrano30/LyA_G5A_E3_Compilador)

<b>Introducción e información general</b>	<b>3</b>
<b>1. Antecedentes</b>	<b>5</b>
1.1. Situación actual del problema	5
1.2 Elaboración de los antecedentes	6
<b>2. Justificación</b>	<b>6</b>
<b>3. Objetivos</b>	<b>7</b>
3.1. General	7
3.2. Específicos	7
<b>4. Metodología</b>	<b>7</b>
4.1. Funcionalidades	7
4.2. Descripción de la metodología del proyecto de investigación o desarrollo	9
Herramientas	9
Lenguaje	10
Tabla de tokens y lexemas.	12
Sintaxis del lenguaje.	12
<b>5. Cronograma de actividades</b>	<b>14</b>
<b>6. Resultados Esperados</b>	<b>15</b>
<b>Manual de usuario</b>	<b>16</b>
<b>Conclusiones</b>	<b>27</b>
<b>Glosario</b>	<b>28</b>
<b>7. Bibliografía</b>	<b>29</b>

## Introducción e información general

En este documento se hará la descripción completa del compilador para el control de acceso a empresas del sector privado llamado AICT (Asistente Inteligente Controlador Tecnológico), el compilador tiene la funcionalidad completa del analizador lexico y sintactico teniendo accesibilidad completa ya que están declaradas las funciones que necesitaremos para crear nuestro proyecto ayudándonos a realizar las funciones básicas de una manera más sencilla, además de que puede detectar errores y señalarnos en donde es que se encuentra el error de programación para que sea fácil de detectar para el usuario y también una breve leyenda de que tipo de problema se trata y en que linea de código se encuentra dicho error para la solución de el.

As seen throughout the course, the lexical analyzer is the first phase of a compiler. A lexical analyzer reads characters from the input file, where the string to analyze is found, recognizes lexemes and returns tokens. A compiler is a program that reads a program written in one language, the source language, and translates it into an equivalent program in another language, the target language.

<i>Componente Léxico</i>	<i>Lexemas de Ejemplo</i>	<i>Descripción Informal del Patrón</i>
<b>while</b>	while	while
<b>if</b>	if	if
<b>relación</b>	<, <=, =, <>, >, >=	< ó <= ó = ó <> ó > ó >=
<b>identificador</b>	nom, dirección, c4	letra seguida de letras y dígitos
<b>natural</b>	3, 4, 67, 98	cualquier constante natural
<b>real</b>	3.45, 12.78, 37.5	cualquier constante real
<b>literal</b>	“ejecución de una acción”	cualquier caracter entre “ ... ”, excepto ”

Para entender un poco mejor todo lo que se explicará a continuación un autómata que son de las partes más importantes en el compilador ya que este ejecutará la secuencia de código para llegar a un fin determinado, es una rama de la matemática conectada con la teoría algebraica de semigrupos y álgebras asociativas. Desde otro punto de vista es una rama del diseño de algoritmos para manipulación de cadenas y procesamiento de secuencias. Un autómata finito puede ser visto como un modelo matemático el cual es tan elemental como posible, en el sentido de que la máquina tiene un tamaño de memoria fijo y limitado, independientemente del tamaño de la entrada.

El autómata como tiene un espacio de memoria nos permite almacenar valores de variables para su posterior utilización dentro del programa ejecutando palabras reservadas, definiendo variables con nuestra gramática, evaluando tokens y determinará si el programa dentro de nuestro compilador es correcto o no estructuralmente.

Una vez explicado esto será más sencillo de comprender la parte estructural de nuestro compilador que es lo complicado, más adelante se detalla las palabras reservadas, cuales son las gramáticas que acepta nuestro lenguaje al trabajar con él y todo lo relacionado con esto.

El proyecto está pensado para las empresas del sector privado con el control de tiempos, control en la puertas de acceso, control de iluminación, control de temperatura, entre otras que más adelante se definirá detalladamente para un mejor conocimiento de las funcionalidades. Este proyecto se pensó para que dichas organizaciones tengan un control más cómodo, específico y funcional de todo lo que se puede realizar dentro de sus espacios de trabajo, como por ejemplo el tener un control de entradas y salidas nos ayuda a reducir los tiempos en que entran los trabajadores a la empresa ya que el que se tengan que parar a abrir una puerta puede ser un poco tardado, con este control de acceso se hará de forma automática ahorrando tiempo ya que no tienen que pararse completamente solo continuar caminando.

Otra funcionalidad por la cual se pensó es el controlar las luces de los espacios de trabajo, gracias a la actualidad y a los cambios de tecnología el tener un control de luces en un hogar ahorra muchísimo tiempo además que nos ayuda a controlar desde un dispositivo móvil que luces queremos tener encendidas para tener un ahorro de luz considerable, en lo que reflexionamos de esto es que si en un hogar es muy útil esta tecnología al aplicarlo en una organización sería de mucha ayuda, así que decidimos implementarlo en nuestras funcionalidades.

What is sought is to save time that could be seen as something insignificant but when it is applied it is seen that it is something that is useful for workers, clients or access to these control variables to be productive saving as much time as possible.



# **1. Antecedentes**

## **1.1. Situación actual del problema**

En el presente de estos días aún hay entidades administrativas de gobierno o negocios que no utilizan un método de registro de entradas y salidas del personal en sus instalaciones, por lo cual los empleados pueden ser susceptibles en ser difamados de culpables en situaciones que no les acontece, cuando suceden incidencias en las cuales pueden ser etiquetadas de robo, daño de materiales o inclusive peor caso como la situación de muerte.

Desde hace ya tiempo las organizaciones de mediano, o gran número, de empleados han tenido la necesidad de controlar el flujo de entrada y salida del personal para poder coordinar diversas actividades dentro de la operación y los procesos de trabajo. Este control siempre ha sido de mucha importancia para una empresa ya que de allí se calculaban las horas a cancelar por la jornada laboral, además quedaban los registros de asistencia o no de cada persona en particular.

De esta manera, los registros se llevaban de manera manual, asignando una tarea a una persona de anotar la hora de entrada y salida de cada empleado en una planilla que luego se tomaba como formato para los pagos de las horas laboradas y registros de asistencia. Sin embargo, esta práctica no era eficiente del todo cuando una organización contaba con una cantidad considerable de personal, turnos de trabajo además de que la persona que anotaba la entrada y salida representaba un gasto para la organización que no generaba ningún beneficio.

Es por esto que se realizan las primeras máquinas para realizar los registros de entrada y salida de manera automatizada, las cuales eran máquinas en sus inicios totalmente mecánicas llamada Reloj de Fichar y el método consistía en asignar una tarjeta personal a cada trabajador y la máquina realizaba un orificio a la tarjeta a la entrada y a la salida, quedando registrado en la tarjeta la asistencia. Sin embargo, este sistema tiene el problema de necesitar una persona para tomar la tarjeta de marcación de asistencia y de modo tener evidencia mediante perforaciones.

A raíz del problema que se presenta por las marcaciones por parte del personal, existe la necesidad de mejorar el método utilizado para los registros de entrada y salida. Es entonces que se generan dispositivos electrónicos que suplantaron las máquinas mecánicas, que permitirán al personal de las entidades laborales generar registros de entrada y salida automatizadas mediante tarjetas o detectores que les permita ser registrados en un sistema de control.



## 1.2 Elaboración de los antecedentes

With the evolution of electronics, more versatile methods of attendance verification are generated, thus developing the method of recognition of personnel through chips or codes in cell phones and cards, which will later be verified by sensors, which consist of identifying the presence of the code or chip by scanner sensor.

Nuestro proyecto a un inicio se planeó para un asistente de banco, para registrar ingreso, registrar salida, dar tickets para pasar a atención con el personal ejecutivo, al tener una platica sobre sus funcionalidades nos percatamos que sus funciones son mucho más completas y adaptables a cualquier ingreso a una institución o para alguna actividad ya que nos permite registrar asistencia, contar el personal, control de luces, control de temperatura y algunas otras funcionalidades muy prácticas para cualquier institución.

## 2. Justificación

El rendimiento de los trabajadores, así como el compromiso que estos tienen con la empresa, es muy importante para esta última, es por ello que el proyecto AICT pretende optimizar y agilizar el sistema de asistencia de la empresa, además de implementar innovaciones en él por lo que esto no sólo será un beneficio para la empresa, sino también, para los empleados. Entre los beneficios encontramos los siguientes:

- Registro de entrada y salida inteligente, pues nuestro proyecto contará con ciertas innovaciones para el registro de entrada y salida del empleado, este no perderá el tiempo en hacerlo manualmente.
- Registro de ubicaciones accedidas, al ser un proceso automatizado y con la ayuda de nuestros dispositivos, podríamos saber qué empleado accedió a cierta área específica.

- Optimización de costos, el sistema, a largo plazo, puede reducir los costos de la empresa, pues no necesitaría estar reemplazando su sistema actual cada cierto tiempo, además de que el uso de nuestros dispositivos abaratará el costo de mantenimiento.
- Mejora en la gestión de visitas en la empresa, pues ahora se podrá verificar el acceso de estas y evitar que ocurran problemas.
- Mejora en el acceso de los empleados, pues, en caso de no contar con ello, ahora se especificará qué empleado acaba de ingresar.

## 3. Objetivos

### 3.1. General

Desarrollar asistencia para empresas dedicadas a la atención del personal de las empresas, teniendo un control preciso de los trabajadores y visitantes a las instalaciones.

### 3.2. Específicos

1. Poner al alcance de los trabajadores y visitantes un sistema que permita hacer registro de entradas y salidas en las áreas de los edificios de empresas privadas.
2. Poner en uso un sistema digital de control de registros y entradas a habitaciones de un edificio que pertenezca al sector privado.
3. Permitir a los visitantes la visualización del tiempo en espera que hay actualmente en el establecimiento.
4. Automate the process to make an appointment based on a visitor's request, in addition to being able to cancel said appointment if necessary.
5. Automatizar el control de la temperatura en el establecimiento permitiendo aumentar o disminuir según sea necesario.
6. Digitalizar el control de acceso e iluminación del establecimiento.

## 4. Metodología

### 4.1. Funcionalidades

Aplicación de escritorio:

- Control de entradas y salidas
  - Se tendrá un control de entradas y salidas eficiente, funcionará con una tarjeta que tendrá el personal de la empresa, cuando pase la tarjeta la primera vez se tomará su asistencia y cuando pase su tarjeta 2 veces será su salida.

- Conteo de entradas y salidas

- Se podrá tener un conteo preciso del personal que está actualmente en la empresa, con ayuda del control de entradas y salidas del personal. Su función es que al momento de checar la entrada se tomará como 1 persona más en sistema, ya que es su entrada, esa persona hasta que no chequee su salida con su tarjeta permanecerá en sistema y se contará como una persona más. Al momento de que se pase su tarjeta para checar su salida en sistema, al ser la segunda vez que pasa su tarjeta, el sistema lo estará borrando de las personas en sistema actualizando el total de personas dentro.

- Generación de turno actual y turno siguiente

- Se podrán generar turnos mediante el asistente de voz agilizando el trámite para la creación de ellos, además se contará con la opción de decir en cuestión de atención cuál es el turno que actualmente se está atendiendo y cual es el próximo en pasar esto para una atención más personalizada y agradable.

- Tiempo de espera promedio

- Se podrá dar un tiempo de espera en promedio al momento de generar tu turno, eso se logrará en función de los clientes que estén antes y el tiempo promedio de atención que se calcule en sistema. Esta función ayudará a los clientes a saber cuanto es que estarán esperando para que se pueda atender, creando un escenario para que validen si pueden esperar el tiempo promedio.

- Agendar citas

- Se podrán agendar citas, en base a disponibilidad, horarios, días hábiles y tipo de atención. La generación de citas se tiene en cuenta para un tipo de atención más amplio no reduciéndose a solamente ir y esperar tu turno sino a que se pueda agendar la visita, para que como empresa mejorar su margen de beneficio ya que habrá más satisfacción por parte de nuestro cliente al tener esa accesibilidad pudiendo acudir un día que él pueda a una horario que se ajuste a sus necesidades.

- Cancelar citas

- Se podrán cancelar citas anteriormente generadas, en caso de que nuestro cliente haga llegar que no podrá llegar a su cita o no podrá, se contará con la opción de poder cancelar esa cita evitando una espera del cliente innecesaria.

- Temperatura del espacio

- Se podrá tener un control de temperatura amplio, con ayuda de un sensor de temperatura nos podrá decir cual es la temperatura del ambiente en el momento que preguntemos.

- Control de puertas

- Se podrá tener el control de puertas, pudiendo abrir y cerrar a como nosotros nos convenga, ya sea para la entrada de clientes y salida ó para alguna tarea en específico.

- Control de luces

- You will be able to have total control with the lighting of the place, we will be able to turn the lights on and off just by saying it to the voice assistant, saving time since it will not be necessary to go to the contact, we only say it and the areas that we want to turn off will be turned off.



- Dar hora y fecha
  - Se podrá visualizar la hora y fecha en una pantalla, debido a que atenderemos turnos y citas es necesario el tener el conocimiento de la hora así que se podrá visualizar la hora.

## 4.2. Descripción de la metodología del proyecto de investigación o desarrollo

Para poder cumplir con las expectativas de los objetivos anteriormente mencionados, se hará uso de las habilidades y conocimientos que hemos adquirido durante nuestro trayecto educativo. Por lo cual la metodología por usar será la siguiente:

Para realizar de forma exitosa los objetivos se utilizará una metodología ágil la cual sería SCRUM, ya que esta metodología está enfocada al trabajo en equipo entre cliente y proveedor, los integrantes colaboran con el único fin de avanzar gradualmente y lograr la entrega de un producto de calidad en tiempos y costos planeados, el SCRUM funciona de la manera siguiente:

Primero se debe de definir los objetivos y alcances esperados del producto a desarrollar, una vez establecido lo anterior el proyecto se divide en Sprint es decir en cada una de la fases del proyecto donde se presentan los avances de los clientes, donde el cliente prueba el producto y lo aprueba tal como está o sugiere cambios y una vez realizado todos los sprint necesarios finalmente se hace la entrega del producto final.

## Herramientas

- Lenguaje Java: se utilizará este lenguaje ya que este mismo ofrece herramientas para la elaboración de un analizador léxico y sintáctico, como lo sería JFlex y JCup. También se hará uso de este lenguaje por la experiencia que tienen los integrantes del equipo con este mismo. JFlex es una herramienta para generar escáneres: programas que reconocen patrones léxicos en un texto. flex lee los ficheros de entrada dados, o la entrada estándar si no se le ha indicado ningún nombre de fichero, con la descripción de un escáner a generar. La descripción se encuentra en forma de parejas de expresiones regulares y código C, denominadas reglas. flex genera como salida un fichero fuente en C, 'lex.yy.c', que define una rutina 'yylex()'. Este fichero se compila y se enlaza con la librería '-lfl' para producir un ejecutable. Cuando se arranca el fichero ejecutable, este analiza su entrada en busca de casos de las expresiones regulares. Siempre que encuentra uno, ejecuta el código C correspondiente.

If when the error is detected we have already passed through an acceptance state, the action corresponding to the acceptance state that has just been passed through and the rest of the characters are returned to the input and we position ourselves in the initial state, to start from there to recognize the next token.

- NetBeans: Es un IDE (Integrated Development Environment) o entorno de desarrollo integrado, que es gratuito y de código abierto. Si quieres saber qué es Netbeans, en primera instancia, se debe destacar que sirve para el desarrollo de aplicaciones web, corporativas, de escritorio y móviles que utilizan plataformas como Java y HTML5, entre otras.

Pero para saber en concreto qué es Netbeans, debemos aclarar cómo se define un IDE como Netbeans. Es sencillo. Un IDE es una aplicación que tiene como función principal la optimización del desarrollo de software por parte de los programadores.

Se trata de un editor de código fuente, junto con recursos de construcción automáticos y un depurador. Igualmente, brinda la función de autocompletado inteligente de código o IntelliSense. Y en el caso de la plataforma de Netbeans IDE, ofrece un compilador y un intérprete. Un IDE como Netbeans permite integrar los lenguajes de programación con las plataformas correspondientes a los sistemas operativos o entornos de programación.

Como este proyecto se estará utilizando la metodología de SCRUM es decir en trabajo en equipo se utilizara la herramienta siguiente:

- GitHub: se utilizara esta herramienta ya que es de los mejores sistemas de control de versiones, por lo que sería de ayuda para poder administrar los avances que realice el equipo de trabajo.

## Lenguaje

### Alfabeto

- Letras permitidas: { a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, A, B, C, D, E, F, G, H, I, J, K, L, M, N, Ñ, O, P, Q, R, S, T, U, V, W, X, Y, Z }
- Dígitos: { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }
- Operadores y símbolos especiales: { (, ), {, }, =, +, -, >, <, ., \*, ,, |, &, ^, @ }
- Separadores: { \n (Nueva línea), \r (Retorno de carro), \t (Tabulador horizontal), (espacio en blanco) }
- Palabras reservadas: { cita\_agendar, cita\_cancelar, turno\_nuevo, turno\_actual, turno\_proximo, turno\_tiempo, turno\_fecha, ilu\_encender, ilu\_apagar, temp\_establecer, temp\_subir, temp\_bajar, puerta\_abrir, puerta\_cerrar }

### Constantes del lenguaje

Una constante es una palabra reservada que le brinda al programador facilidades al momento de estar haciendo código. Estas constantes son llamadas anteponiendo en ellas el @ seguido de los elementos que dichas constantes requieren.

**Agendar:** La palabra reservada agendar permite reservar una cita para obtener una atención más amplia y personalizada. Esta palabra se incluye junto con los parámetros de hora, día y tipo de atención.

**Cancelar:** La palabra reservada cancelar permite cancelar una cita anteriormente reservada. Para cancelar la cita se requiere la palabra reservada cancelar junto con el identificador de la cita previa.

**Turno:** La palabra reservada turno permite obtener el turno más próximo disponible.

**Turno Actual:** La palabra reservada turno\_actual permite saber cual es el turno actual que puede ser atendido.

**Turno Próximo:** La palabra reservada turno\_proximo permite saber cual es el turno próximo va a ser atendido.

**Encender:** La palabra reservada encender permite encender las luces de varias áreas del inmueble, para realizar esto se hace con la palabra reservada encender y como parámetro la habitación del inmueble que se desea encender.

**Apagar:** La palabra reservada apagar permite apagar las luces de varias áreas del inmueble, para realizar esto se hace con la palabra reservada apagar y como parámetro la habitación del inmueble que se desea apagar.

**Tiempo:** La palabra reservada tiempo permite saber el tiempo aproximado de espera que se tendrá al solicitar un turno.

**Temperatura:** La palabra reservada temperatura permite saber la temperatura de varias áreas del inmueble, para realizar esto se hace con la palabra reservada temperatura y como parámetro la habitación del inmueble de la cual se desea conocer la temperatura.

**Subir:** La palabra reservada subir permite subir la temperatura de varias áreas del inmueble, para realizar esto se hace con la palabra reservada subir y como parámetros la habitación del inmueble de la cual se desea subir la temperatura y el número de grados que se desea incrementar.

**Bajar:** La palabra reservada bajar permite bajar la temperatura de varias áreas del inmueble, para realizar esto se hace con la palabra reservada bajar y como parámetros la habitación del inmueble de la cual se desea bajar la temperatura y el número de grados que se desea decrementar.

**Abrir:** La palabra reservada abrir permite abrir la puerta de varias áreas del inmueble, para realizar esto se hace con la palabra reservada abrir y como parámetros la habitación del inmueble de la cual se desea abrir la puerta y el identificador de la puerta en la habitación que se desea abrir.

**Cerrar:** The reserved word close allows you to close the door of several areas of the property. To do this, it is done with the reserved word close and as parameters the room of the property from which you want to close the door and the identifier of the door in the room that is closed. want to close.

**Fecha:** La palabra reservada fecha permite saber la fecha exacta incluyendo la hora, esto para saber las próximas citas agendadas que se deberán de atender.

Tabla de tokens y lexemas.

Tabla de símbolos		
Token (Componente léxico)	Lexema	Patrón
if	if	if
else	else	else
while	while	while
for	for	for
identificador	a, valor, b	Identificador = letra ( letra   dígito )*
integer	1, 2, 3, 5, 20, 900	[0-9]+[0-9]+[0-9]
operador relacional	<, >, ==, !=, <=, >=	<, >, ==, !=, <=, >=
operador aritmético	+, -, *, /	+, -, *, /
operador logico	&&,	&&,
float	float	[dígito]+". "[dígito]
end	end	end
return	return	return
string	string	string
Operador de agrupación	(, ), {, }, [, ]	(, ), {, }, [, ]
Operador de asignación	=	=
Operador incremental	++, --	++, --

## Sintaxis del lenguaje.

- **Instrucción class:** class «nombre»{«características e/o instrucciones»}
- **Instrucción return:** return(«objeto o valor a retornar»); (requiere una pre sentencia condicional con un parámetro de retorno válido).
- **Sentencia if:** si(«condición»){«instrucción»}
- **Sentencia else:** else{«instrucción»} (requiere una pre sentencia if)
- **Sentencia for:** for(«condición»){«instrucción»}
- **Sentencia while:** while(«valor de inicio», «condición del ciclo», «incremento»){«instrucción»}

- **Instrucción agendar:** agendar(«set hora», «set día», «tipo de atención») {get id de la cita agendada}
- **Instrucción turno:** turno{get turno más próximo disponible}
- **Instrucción turno\_actual:** turno\_actual{get turno actual}
- **Instrucción turno\_proximo:** turno\_proximo{get turno próximo}
- **Instrucción tiempo:** tiempo{get tiempo aproximado para ser atendido}
- **Instrucción cancelar:** cancelar(«set identificador de la cita agendada»)
- **Instrucción encender:** encender(«set identificador de la habitación»)
- **Instrucción apagar:** apagar(«set identificador de la habitación»)
- **Instrucción subir:** subir(«set identificador de la habitación», «set valor entero de grados a incrementar»)
- **Instrucción bajar:** bajar(«set identificador de la habitación», «set valor entero de grados a decrementar»)
- **Instrucción abrir:** abrir(«set identificador de la habitación», «set identificador de la puerta»)
- **Instrucción cerrar:** cerrar(«set identificador de la habitación», «set identificador de la puerta»)
- **Instrucción temperatura:** temperatura(«set identificador de la habitación») {get temperatura actual en la habitación}
- **Instrucción fecha:** {«get fecha actual»}

To determine the compiler rules, JFLEX, JCUP, regular expressions and automata tools were used. In addition to that, only the tokens that are included in the table based on the following data types have an effect

## 5. Cronograma de actividades

Tabla 1. Cronograma de Actividades

No.	Actividad	Descripción	Inicio	Fin	Sprint
1	Metodología SCRUM	Investigar a detalle el uso de la metodología SCRUM	11/03/2022	14/03/2022	No
2	Roles SCRUM	Definir los roles del equipo SCRUM	15/03/2022	17/03/2022	No
3	Control de versiones	Investigar y definir la herramienta para el control de versiones	18/03/2022	21/03/2022	No
<b><i>Etapas de Desarrollo del Proyecto</i></b>					
4	Estructura del Proyecto	Definir cómo va a estar estructurado el proyecto y creación de vistas	22/03/2022	28/03/2022	No
5	Sprint Planning  Desarrollo barra de tareas y barra de herramientas  Desarrollo analizador léxico	Reunión del equipo SCRUM para comenzar la fase de desarrollo del proyecto  Desarrollo de la barra de tareas y herramientas dentro del proyecto  Desarrollo del analizador léxico	29/03/2022	07/04/2022	1
6	Test de barra de tareas y herramientas  Test analizador léxico	Se hace la prueba que funcionen correctamente las funcionalidades de la barra de tareas y herramientas  Test del analizador léxico	08/04/2022	28/04/2022	1
7	Sprint review Sprint retrospective	Revisión del sprint terminado Retroalimentación del sprint terminado	29/04/2022	29/04/2022	1

8	Sprint planning  Desarrollo analizador sintáctico	Reunión del equipo SCRUM para comenzar el desarrollo para el analizador sintáctico Desarrollo del analizador sintáctico	02/05/2022	23/05/2022	2
9	Test analizador sintáctico	Test analizador sintáctico	24/05/2022	27/05/2022	2
10	Sprint review Sprint retrospective	Revisión del sprint terminado Retroalimentación del sprint terminado	30/05/2022	30/05/2022	2
11	Sprint planning  Desarrollar funcion para abrir puerta	Planear actividades para este sprint Desarrollar la función para abrir una puerta usando componentes reales	31/05/2022	04/06/2022	3
12	Test abrir puerta	Test función abrir puerta implementada en un objeto físico	04/06/2022	05/06/2022	3
13	Sprint review Sprint retrospective	Revisión del sprint terminado Retroalimentación del sprint terminado	06/06/2022	06/06/2022	3
<b><i>Etapas de Presentación del Proyecto</i></b>					
14	Presentación final	Presentar el proyecto elaborado	08/06/2022	08/06/2022	No

## 6. Resultados Esperados

Once the project is completed, there will be a software that will allow organizations to have a detailed control of everything, you will be able to control access to your organization, you will be able to easily schedule appointments avoiding time-consuming procedures, you will be able to control your temperature as desired we will be providing a unique and comforting experience when operating these tasks.

Asimismo gracias a su funcionalidad se podrá ver el turno actual en la atención al cliente, el turno siguiente para que nuestro cliente esté al tanto de que turno es el actual y que pueda tener conocimiento de cuantos turnos faltan para su atención permitiendo que el decida y analice si es viable que se quede a esperar el turno o realice una reprogramación de su turno con la función de cancelar y agendar nuevamente para un día que el desee con la función de Fecha nos ayudará a ver la fecha disponible para el siguiente turno, todo esto para ahorrar el tiempo del cliente para no hacer una mala experiencia y teniendo esa accesibilidad por parte del sistema para adaptarse a cualquier situación.

Un punto muy importante en las organizaciones es el tiempo, como dice el dicho el tiempo es oro es por eso que implementamos esta idea y se llevó a cabo para el proyecto final ya que el objetivo es ahorrar el mayor tiempo posible en actividades cotidianas automatizarlas para que su personal sea más productivo y el cliente se lleve una experiencia única ya que ayuda al servicio a cliente y da una buena impresión ya que deja ver que el personal únicamente se centra en realizar su trabajo.

## Manual de usuario

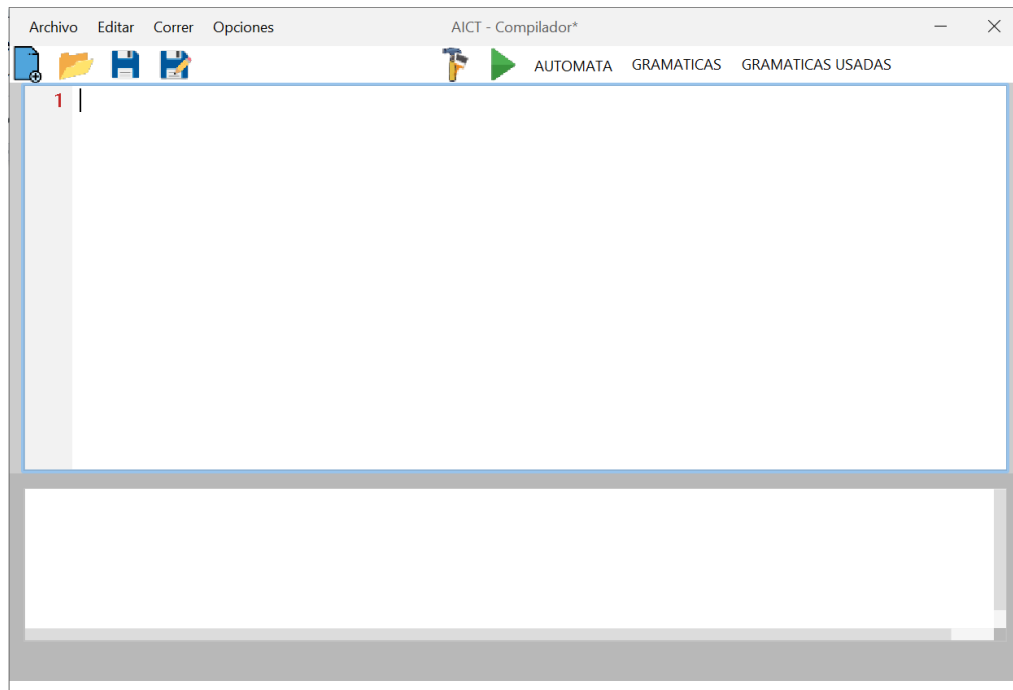
Un sistema es un conjunto de elementos relacionados entre sí que funciona como un todo. Si bien cada uno de los elementos de un sistema puede funcionar de manera independiente, siempre formará parte de una estructura mayor. Del mismo modo, un sistema puede ser, a su vez, un componente de otro sistema.

AICT es un asistente que nos permitirá realizar las funciones fundamentales de una empresa de forma sencilla aplicando una serie de comandos para su funcionamiento, para realizar este sistema primeramente tenemos que hacer el programa en el compilador que se explicará a continuación, dicho compilador está hecho 100% por estudiantes del Instituto Tecnológico de Tepic para en un futuro se implemente en el sector privado aumentando la productividad de las empresa.

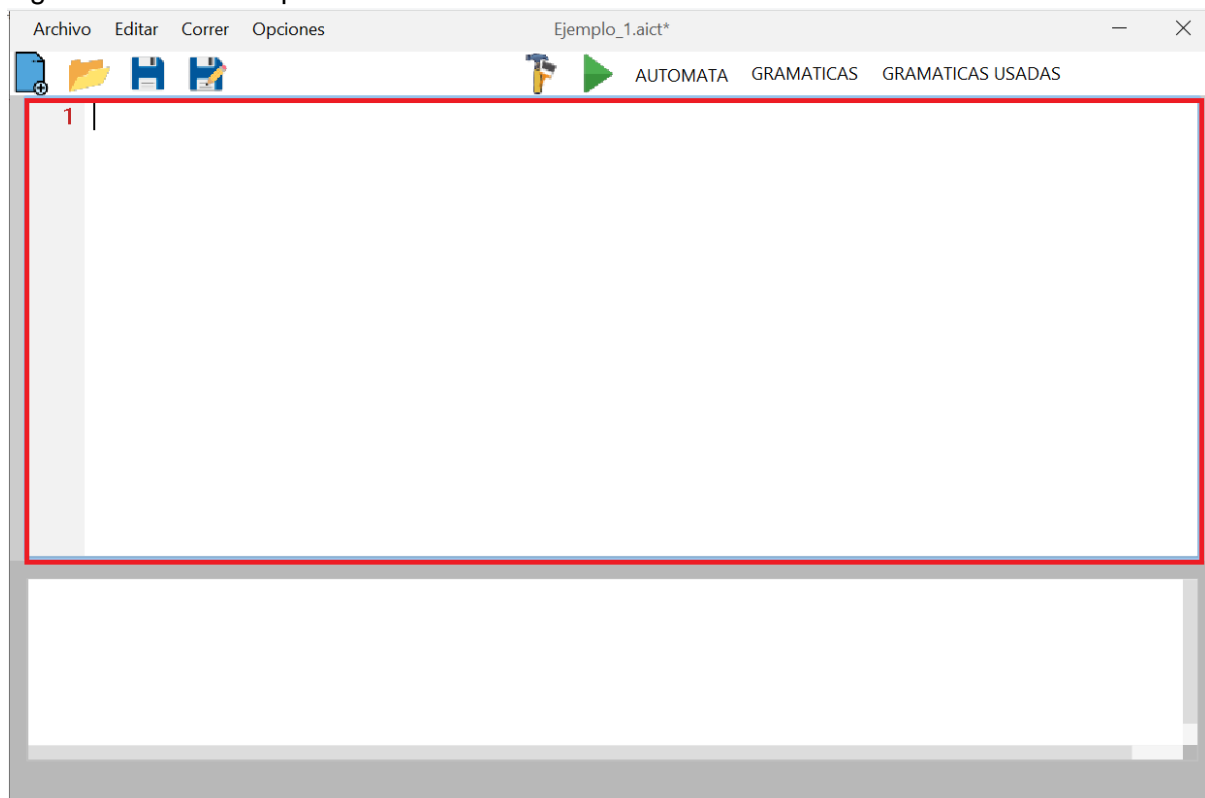
En este apartado del documento se verá el manual de usuario, osea el funcionamiento de cada una de las interfaz del compilador explicando detalladamente que es lo que realiza cada apartado para un mejor entendimiento para el usuario final.

La primera interfaz que nos mostrará es la general, en esta se verán todas las funcionalidades que permite hacer nuestro compilador, cada uno de los botones tiene una función en específico.

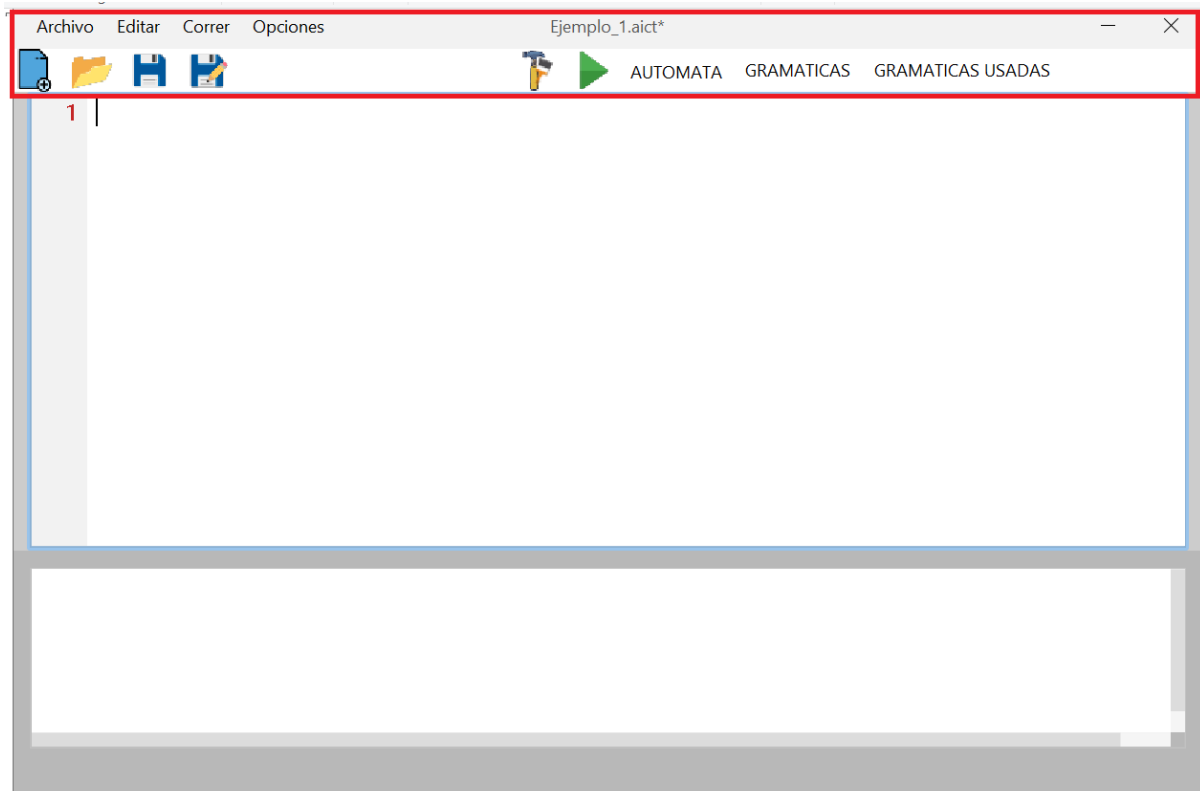




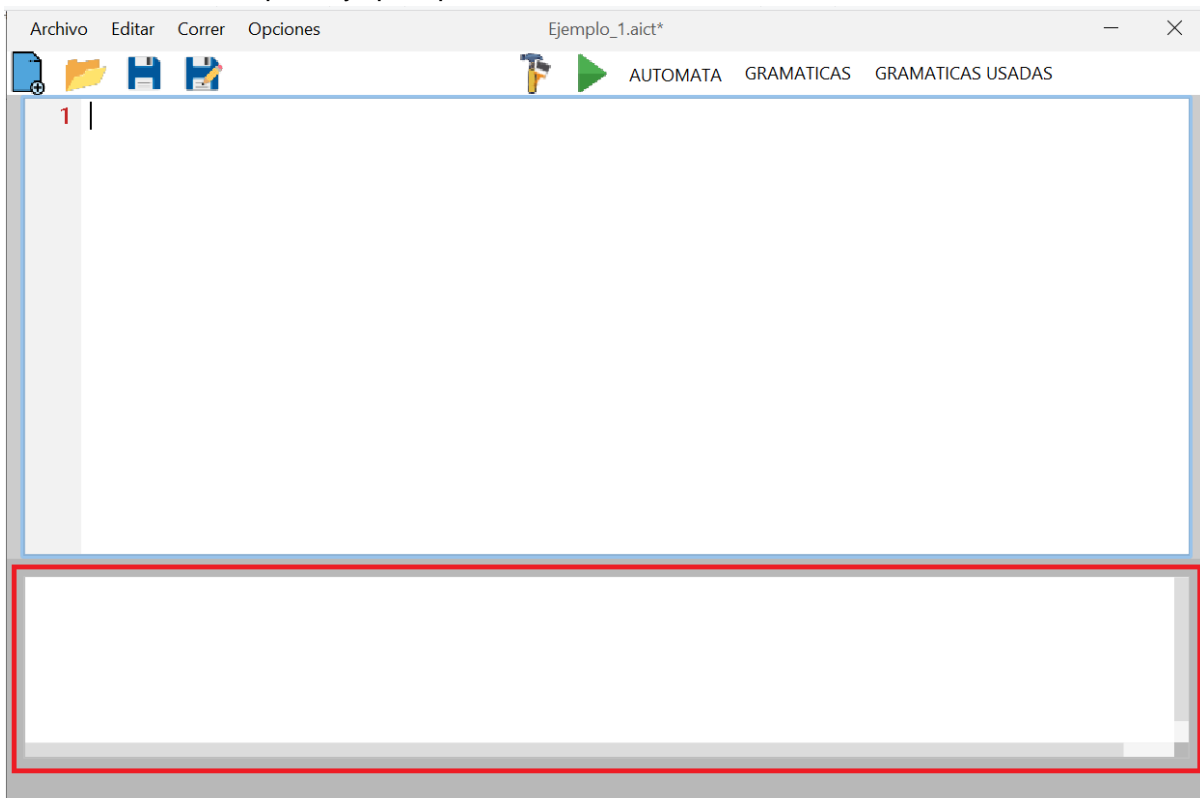
En el siguiente apartado de la pestaña principal está un cuadro de texto donde podremos realizar toda la programación que deseemos y la parte fundamental del compilador ya que aquí es donde escribiremos la línea de código para posteriormente compilar y ejecutar para lograr el resultado esperado.



In the upper part of the screen we have the buttons and the functionalities of the system, it will be possible to execute the program, compile, open a new file, see the automaton that is generated in the program, see the grammars, the grammars that were used in addition to that will come the name of the file that is being executed along with the extension with which it works, which is **aict** and other options that will be explained in detail later.

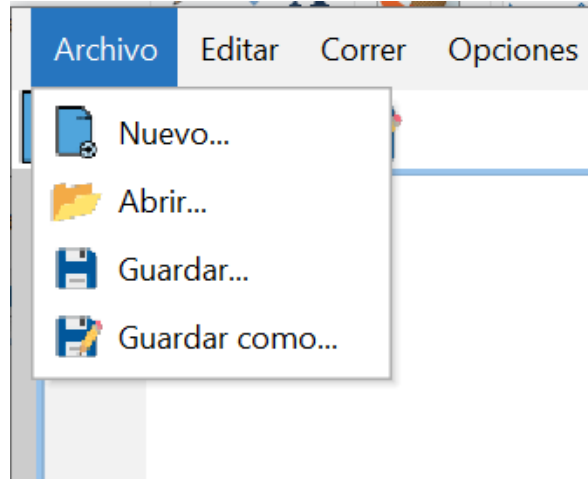


Por último de la pestaña principal tenemos el manejador de errores que se encuentra situado en la parte inferior de la pantalla que nos dirá que y nos explicará de manera detalla qué error se está generando en nuestro programa, nos ayudará a visualizar rápidamente el error con una descripción y que tipo de error se está cometiendo.

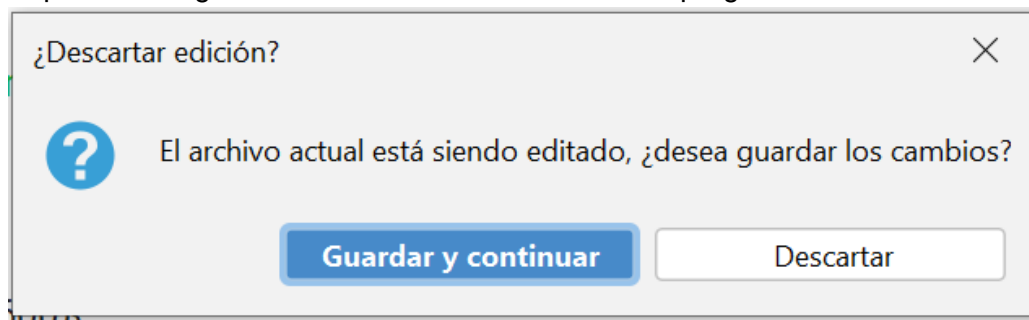


En la parte superior izquierda de la ventana principal tenemos un menú desplegable y en cada división del menú se hará algo distinto, en este caso el apartado **Archivo** podemos generar un archivo nuevo para que en caso de de querer realizar otro programa fácilmente lo hagamos, únicamente damos clic en Nuevo y listo podremos realizar otro programa fácilmente.

**Nota:** En caso de que estemos trabajando en un programa y damos nuevo en caso de no guardar el programa, se estará perdiendo el código que generamos anteriormente.

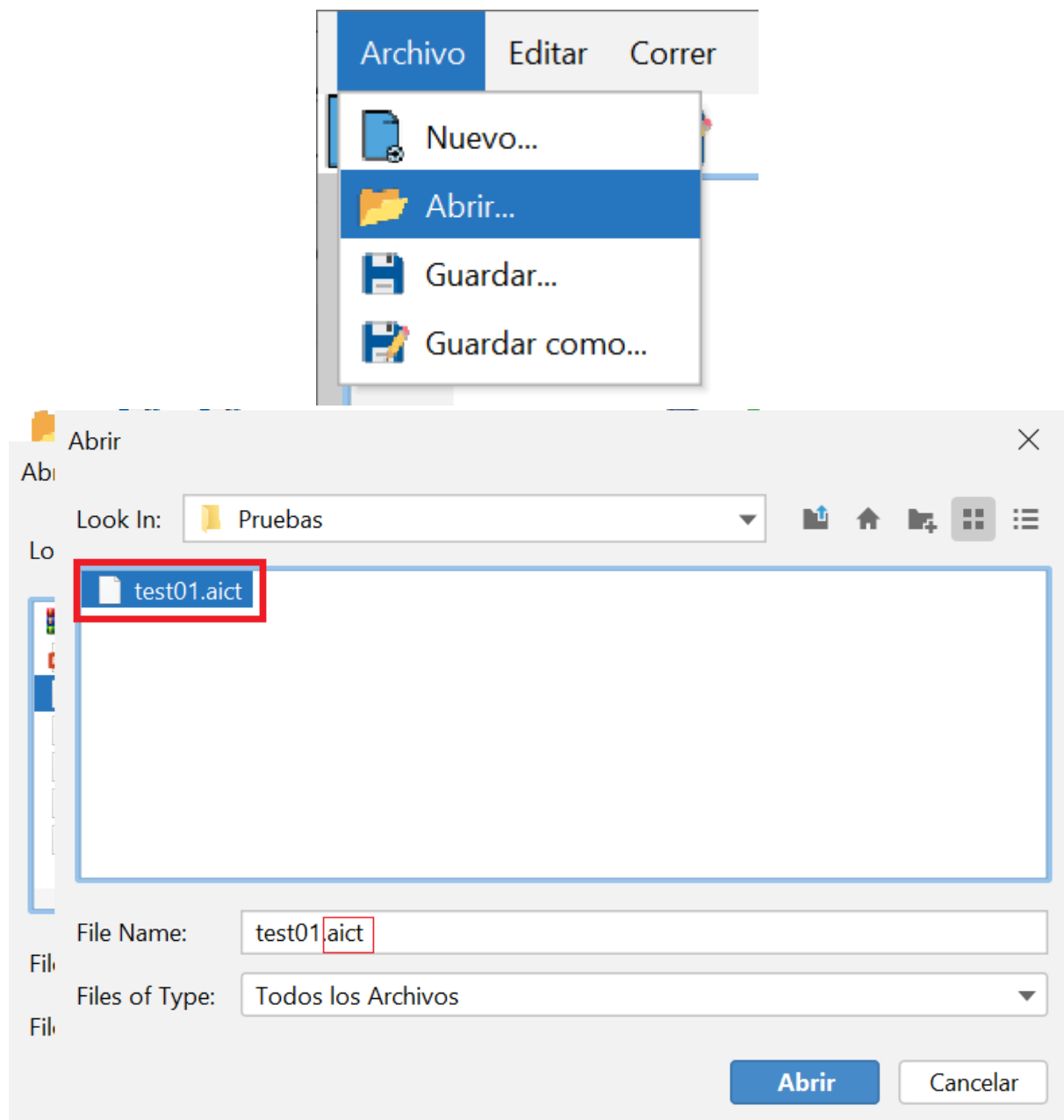


En caso de dar clic en Nuevo y nuestro programa no esté guardado nos mostrará el siguiente mensaje, este mensaje nos indicará que el archivo con el que actualmente estemos trabajando no está guardado y en caso de continuar se estará perdiendo el avance, nos dará la opción de **Guardar y continuar** que en caso de ser un archivo nuevo se abrirá el menú de navegación de Windows para guardar nuestro archivo y en caso de dar clic en **Descartar** se perderán todos los cambios y no se estará guardando el avance o en caso de que no esté guardado el archivo anteriormente el progreso.

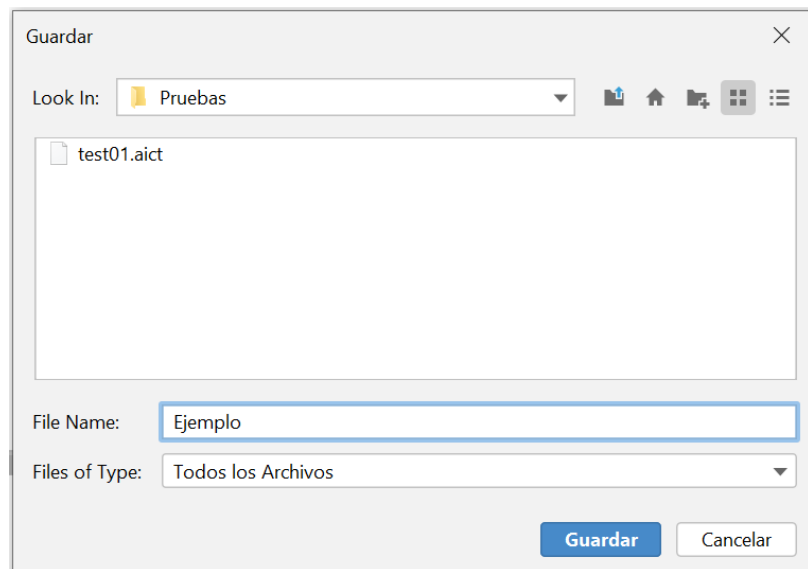
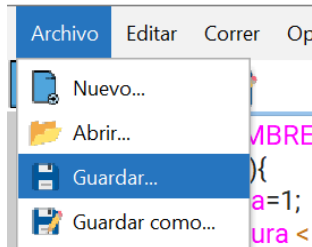


The following functionality within File gives us the option to open a previously saved program to continue working with it and complete what we want in it.

Note: The files that the compiler recognizes must have the extension .aict (Assistant Intelligent Technology Controller), other files with another extension will not be recognized.



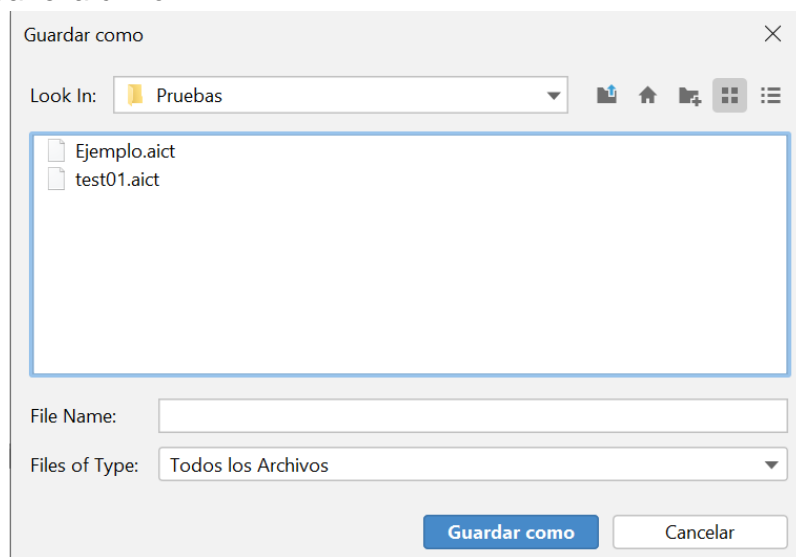
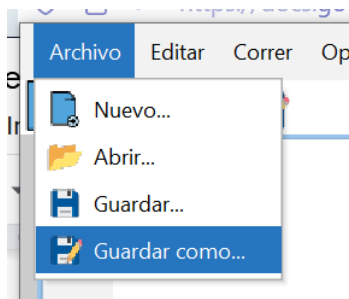
La siguiente funcionalidad dentro de **Archivo** es **Guardar** que nos permite guardar el archivo que estemos trabajando, en caso de no estar guardado por primera vez se habría el navegador de Windows para establecer la ruta de guardado y el nombre que deseemos utilizar para reconocer al archivo.



Al darle el nombre del archivo y dar clic en guardar se generará en la carpeta destino el archivo que nosotros nombramos con una extensión .aict que es la que genera el compilador y la que se tiene que usar para poder abrir archivos y seguir trabajando en el.

Nombre	Estado	Fecha de modificación	Tipo	Tamaño
Ejemplo.aict		6/8/2022 8:46 PM	Archivo AICT	0 KB
test01.aict		6/8/2022 6:54 PM	Archivo AICT	1 KB

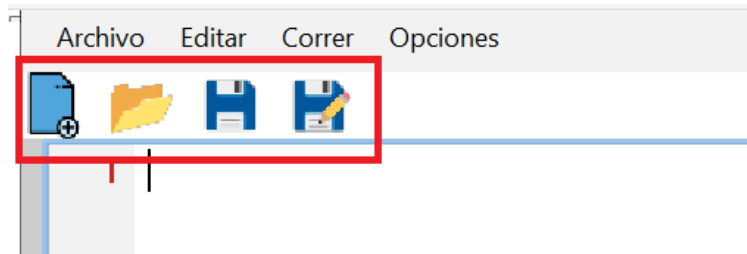
La funcionalidad siguiente en el apartado de **Archivo** es **Guardar** como que lo que realiza es que podemos renombrar el archivo sin que se realicen los cambios en el antiguo programa editado, al darle clic se abrirá el menú de navegación de Windos para nosotros decidir como queremos guardar el archivo.



When saving it in the path that we choose, they will be saved with the changes that we have made but without updating the previous file.

Ejemplo.aict		6/8/2022 8:46 PM	Archivo AICT	0 KB
test01.aict		6/8/2022 6:54 PM	Archivo AICT	1 KB
Ejemplo_1.aict		6/8/2022 8:53 PM	Archivo AICT	0 KB

En la parte de abajo del menú desplegable en la parte superior izquierda tenemos los botones con imágenes que realizan las mismas funciones que las de **Archivo**.



Nuevo



Abrir archivo



Guardar



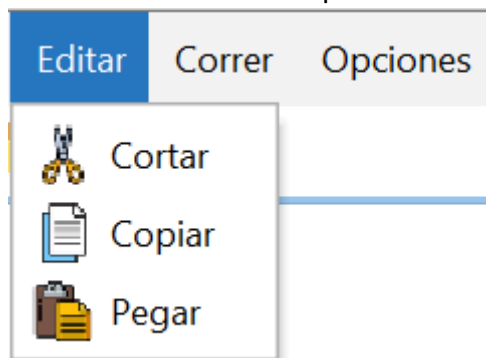
Guardar como

En el menú desplegable en la opción de **Editar** tenemos las funcionalidades de texto que nos ayudan a escribir rápidamente pegando código de algún archivo anterior, copiar texto o cortar.

**Cortar:** Nos permite borrar un texto que anteriormente seleccionamos.

**Copiar:** Nos permite copiar el texto seleccionado para reutilizar en donde queramos.

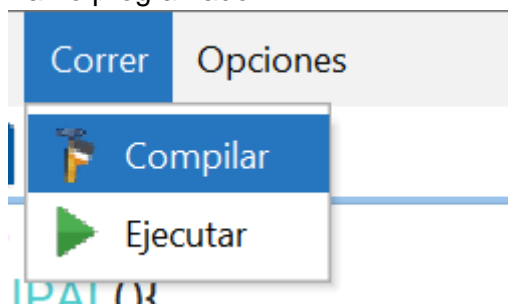
**Pegar:** Nos permite pegar un texto anteriormente copiado.



En el siguiente apartado del menú desplegable tenemos la opción de **Correr**, este apartado nos sirve ya que tenemos un programa desarrollado y queremos ver si tiene algunos errores y en caso de que no correrlo para ver qué resultado nos da.

**Compilar:** Esta función va a analizar línea de código por línea de código palabra por palabra para validar si nuestras palabras son válidas y si no se tiene ningún error tanto léxico como sintáctico. En caso de haber algún error se nos marcará en la parte de errores y nos dará una breve descripción del error presentado.

**Ejecutar:** La función ejecutar se puede realizar cuando nuestro programa no contenga ningún error y permite realizar lo programado.



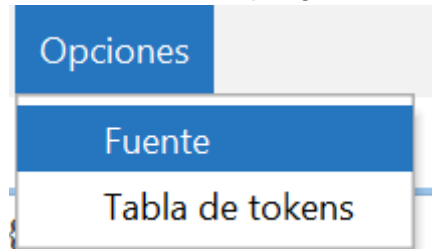


Compilar



Ejecutar

En el último apartado del menú desplegable tenemos la opción de **Opciones**, esta funcionalidad nos permite cambiar la letra y la fuente que utilizaremos para una experiencia más cómoda además de ver la tabla de tokens que genera nuestro programa fuente.



Al dar clic en Fuente se nos mostrará el siguiente menú donde podremos cambiar el tipo de letra, el tamaño de la letra, el estilo que queremos y para visualizar todas estas combinaciones se nos mostrará un ejemplo de como se verá la letra del programa en la parte de abajo para que configuremos a como nos guste más.

Elegir fuente

Font Name	Style	Size
Serif	Plain	12
Serif	Plain	12
Showcard Gothic	Bold	14
SimSun	Italic	16
SimSun-ExtB	Bold Italic	18
Silka Banner		20

Sample

AaBbYyZz

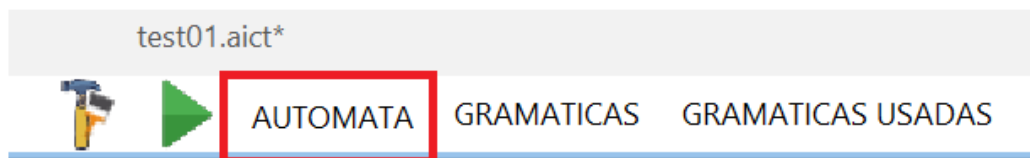
Aceptar

The functionality that the tokens table has is that it will show us all the tokens that were generated in the program once we have compiled it and that it has not generated errors, they will be shown in the form of a list to analyze each one of them , it will also tell us the line of code where it is and the lexeme it uses.

**Nota:** En caso de que no se haya compilado el sistema la tabla de tokens vendrá vacía.

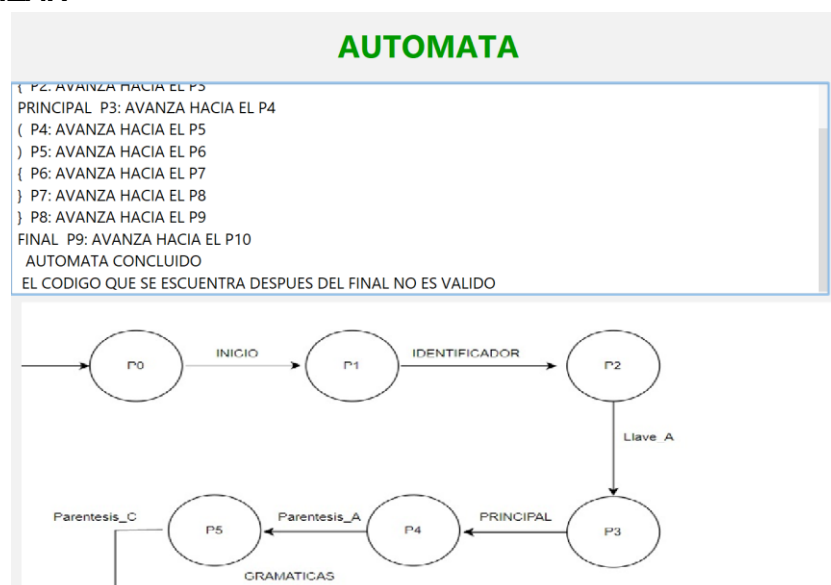
Componente léxico	Lexema	[Línea, Columna]
INICIO	INICIO	[1, 1]
Identificador	\$NOMBRE	[1, 8]
Llave_A	{	[1, 15]
PRINCIPAL	PRINCIPAL	[2, 1]
Parentesis_A	(	[2, 10]
Parentesis_C	)	[2, 11]
Llave_A	{	[2, 12]
Identificador	\$temperatura	[3, 1]
Op_Asignacion	=	[3, 13]
Numero	1	[3, 14]
Punto_Coma	;	[3, 15]
If	if	[4, 1]
Parentesis_A	(	[4, 3]
Identificador	\$temperatura	[4, 4]
Op_Relacional	<	[4, 17]
Numero	25	[4, 18]

La siguiente funcionalidad que tenemos es la de Autómata que lo que nos permite es ver el autómata que genera el programa, generar los estados que tiene e indicar que token avanza hacia qué estado en el programa.



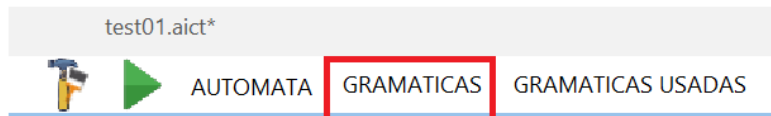
Se muestra de la siguiente manera:

**Nota:** En caso de que no se haya compilado aparecerá el mensaje **NO HAY CODIGO PARA ANALIZAR**



La siguiente funcionalidad que tenemos es la de Gramáticas que nos mostrará todas las gramáticas que utiliza el compilador y puede servirte como ayuda para ver cómo es que se escribe y que funcionalidad tiene dentro del compilador.



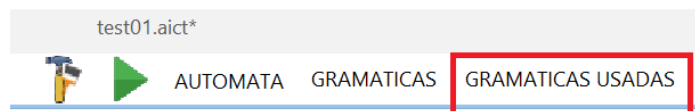


Se mostrará de la siguiente manera.

**Nota:** En caso de no compilar el programa la tabla vendrá vacía.

GRAMATICAS	
[3, 14]	VALOR ---> Numero
[3, 1]	VARIABLE ---> Identificador Op_Asignacion Numero
[3, 1]	VARIABLE_PC ---> Identificador Op_Asignacion Numero Punto_Coma
[4, 19]	VALOR ---> Numero
[4, 4]	EXP_LOGICA ---> Identificador Op_Relacional Numero
[4, 1]	IF ---> If Parentesis_A Identificador Op_Relacional Numero Parentesis_C
[5, 5]	FUNCION_COMPLETA ---> OP_Temperatura Parentesis_A Numero Parentesis_C
[5, 5]	FUNCION_COMPLETA_PC ---> OP_Temperatura Parentesis_A Numero Parentesis_C Punto_Coma
[5, 16]	VALOR ---> Numero
[5, 5]	FUNCION ---> OP_Temperatura
[7, 4]	EXP_LOGICA ---> Identificador Op_Relacional Cadena
[7, 20]	VALOR ---> Cadena
[7, 1]	IF ---> If Parentesis_A Identificador Op_Relacional Cadena Parentesis_C
[8, 16]	VALOR ---> Numero
[8, 5]	FUNCION_COMPLETA ---> OP_Temperatura Parentesis_A Numero Parentesis_C

In the following functionality is the **Used Grammars** and it will allow us to see all the grammar of the program that we carry out, it will indicate to us in which line of code it is located and what type of grammar.



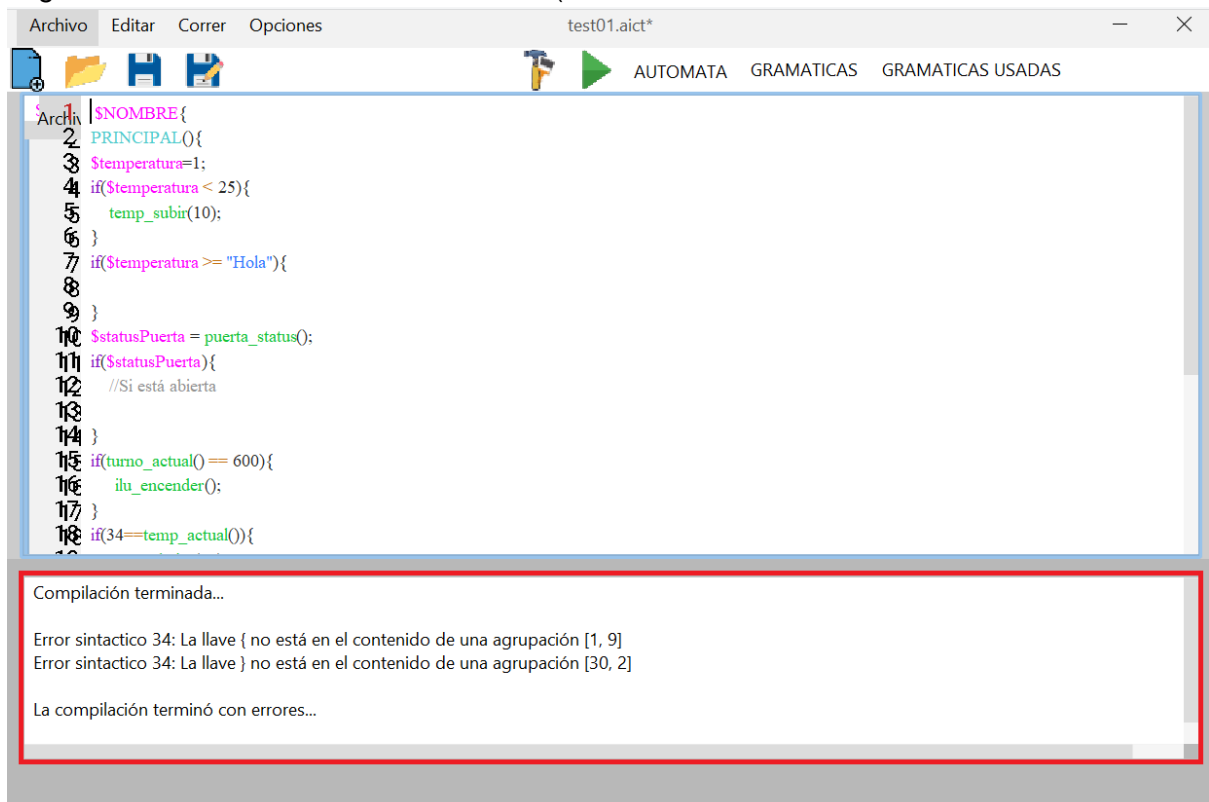
Se mostrará de la siguiente manera.

**Nota:** En caso de no compilar el programa la tabla vendrá vacía.

GRAMATICAS USADAS	
1	VALOR-> (Cadena   Numero   OP_RETORNO)
2	VARIABLE-> Identificador Op_Asignacion VALOR
3	VARIABLE_PC-> VARIABLE Punto_Coma
4	EXP_LOGICA-> (VALOR   Identificador) Op_Relacional (VALOR   Identificador)
5	IF-> If Parentesis_A (EXP_LOGICA   Identificador) Parentesis_C
6	FUNCION_COMPLETA-> FUNCION Parentesis_A (PARAMETROS   (VALOR   Identificador))
7	FUNCION_COMPLETA_PC-> FUNCION_COMPLETA Punto_Coma
8	FUNCION-> (OP_Cita   OP_Turno   OP_Iluminacion   OP_Temperatura   OP_Puerta)
9	OP_RETORNO-> OP_Retorno Parentesis_A Parentesis_C
10	CICLO_FOR-> For Parentesis_A (VALOR   Identificador) Parentesis_C
11	CICLO_WHILE-> While Parentesis_A (EXP_LOGICA   Identificador) Parentesis_C

Por último tenemos la sección de manejo de errores, en este apartado podremos ver cuáles problemas se presentan al programar y una pequeña descripción del error y una posible solución. Se describe que tipo de error es, por ejemplo Error Sintactico, mencionará el numero del error, se dará información del tipo de error y como solucionarlo y por último

vendrá la ubicación del error que será marcado entre [ ] el renglón es el primer número y el segundo numero son las lineas verticales (cadaara



```
1 | $NOMBRE{
2 | PRINCIPAL(){
3 | $temperatura=1;
4 | if($temperatura < 25){
5 |     temp_subir(10);
6 | }
7 | if($temperatura >= "Hola"){
8 |
9 | }
10 | $statusPuerta = puerta_status();
11 | if($statusPuerta){
12 |     //Si está abierta
13 | }
14 | }
15 | if(turno_actual() == 600){
16 |     ilu_encender();
17 | }
18 | if(34==temp_actual()){
19 |
```

Compilación terminada...

Error sintactico 34: La llave { no está en el contenido de una agrupación [1, 9]

Error sintactico 34: La llave } no está en el contenido de una agrupación [30, 2]

La compilación terminó con errores...

# Conclusiones

La construcción y diseño de un lenguaje y un compilador requiere de mucha atención al detalle, pues se tiene que realizar todo en base a nuestro anteproyecto. Tuvimos que realizar la definición de nuestro propio lenguaje, identificación de componentes léxicos, patrones y lexemas de dicho lenguaje. Gracias a la definición de todos estos componentes que construyen el analizador se nos facilitó de cierto modo la creación de este.

Con la realización de nuestros autómatas podemos hacer una identificación y comprobar el funcionamiento que tienen nuestras instrucciones, definidas previamente en nuestro lenguaje, lo que apoya a nuestro compilador y, a su vez, al analizador léxico. El uso de la herramienta de JFlex nos apoyó mucho, pues al ser un analizador facilita esa escritura del código.

Con el desarrollo de nuestro IDE vamos a poder verificar que el texto que ingresemos en nuestro compilador esté en un formato correcto y sea aceptado por nuestro lenguaje, pues también verificará que se tenga congruencia. Con la realización de dicho analizador, pudimos observar que en los diferentes compiladores el analizador léxico se comporta de una manera similar, pero no igual, lo mismo sucede con el que construimos nosotros como equipo.

# Glosario

**Gramática:** Conjunto de normas y reglas para hablar y escribir correctamente una lengua.

**GIC:** Gramáticas de contexto libre o libres de contexto. Sus reglas de producción tan sólo admiten tener un símbolo no terminal en su parte izquierda.

**VT:** Conjunto finito de símbolos terminales.

**VN:** Conjunto finito de símbolos no terminales.

**S:** es el símbolo inicial y pertenece a VN.

**P:** Conjunto de producciones o de reglas de derivación.

**Autómata:** Es un modelo matemático para una máquina de estado finito, en el que dada una entrada de símbolos, “salta” mediante una serie de estados de acuerdo a una función de transición (que puede ser expresada como una tabla).

**Estado:** Situación o modo de estar de una persona o cosa, en especial la situación temporal de las personas o cosas cuya condición está sujeta a cambios.

**Transición:** Paso o cambio de un estado, modo de ser, etc., a otro.

**Anomalía:** Cambio o desviación respecto de lo que es normal, regular, natural o previsible.

## 7. Bibliografía

---

Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D. (2008). COMPILADORES. PRINCIPIOS, TÉCNICAS Y HERRAMIENTAS. Segunda edición. México: PEARSON EDUCACIÓN.

Alfonseca, M. M., De la Cruz Echeandía, M., Ortega de la Puente, A., & Pulido Cañabate, E. (2006). Compiladores e interpretes: teoría y práctica. Madrid: PEARSON EDUCACIÓN, S.A.

Díaz, M. L. (15 de Marzo de 2021). Introducción a la construcción de compiladores. Obtenido de Introducción a la construcción de compiladores:  
<https://www.infor.uva.es/~mluisa/talf/docs/aula/A7.pdf>

Gálvez Rojas, S., & Mora Mata, M. Á. (2005). JAVA A TOPE: TRADUCTORES Y COMPILADORES CON LEX/YACC, JFLEX/CUP Y JAVACC. Málaga: Universidad de Málaga.

HOPCROFT, J. E., MOTWANI, R., & ULLMAN, J. D. (2007). Teoría de autómatas, lenguajes y computación. Madrid: PEARSON EDUCACIÓN S.A.

Robertson, C., Cai, S., & Pasic, A. (2019, 29 agosto). Preprocesador. Microsoft Docs.  
<https://docs.microsoft.com/es-es/cpp/preprocessor/preprocessor?view=msvc-160>  
[https://kataix.umag.cl/~jaguila/Compilers/T01\\_Fases\\_Compilador.pdf](https://kataix.umag.cl/~jaguila/Compilers/T01_Fases_Compilador.pdf) Lenguajes