Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Escuela de Ciencias y Sistemas

Organización de Lenguajes y Compiladores 2

Catedrático: M.Sc. Luis Espino

Auxiliares: Juan Carlos Maeda, Pavel Vásquez

**MinorC**

**Manual de Usuario**

Kairi Itzmar Alexander Franco Medina

201222591

3 de julio de 2020

# ÍNDICE

[ÍNDICE 2](#_Toc44608635)

[INTRODUCCIÓN 2](#_Toc44608636)

[Objetivo general 2](#_Toc44608637)

[Objetivo específico 3](#_Toc44608638)

[REQUERIMIENTOS DEL SISTEMA 3](#_Toc44608639)

[CONTENIDO TÉCNICO 3](#_Toc44608640)

[IDE 3](#_Toc44608641)

[Consola 3](#_Toc44608642)

[Flujo 3](#_Toc44608643)

[Resaltado de sintaxis 3](#_Toc44608644)

[Sintaxis de MinorC 4](#_Toc44608645)

[Tabla de Símbolos 4](#_Toc44608646)

[Reportes 4](#_Toc44608647)

[Reportes de errores 4](#_Toc44608648)

[Reportes de la tabla de símbolos 4](#_Toc44608649)

[Reportes del AST y del GDA 4](#_Toc44608650)

[Reportes gramaticales 4](#_Toc44608651)

[RESPONSABLES 4](#_Toc44608652)

# INTRODUCCIÓN

MinorC es un subconjunto del lenguaje C, creado con el fin de poner en práctica los conceptos del proceso de compilación.

El siguiente manual le será útil principalmente a los a los usuarios finales que deseen utilizar MinorC como lenguaje de programación

## Objetivo general

Analizar y sintetizar un compilador para un lenguaje de programación MinorC, utilizando PLY.

## Objetivo específico

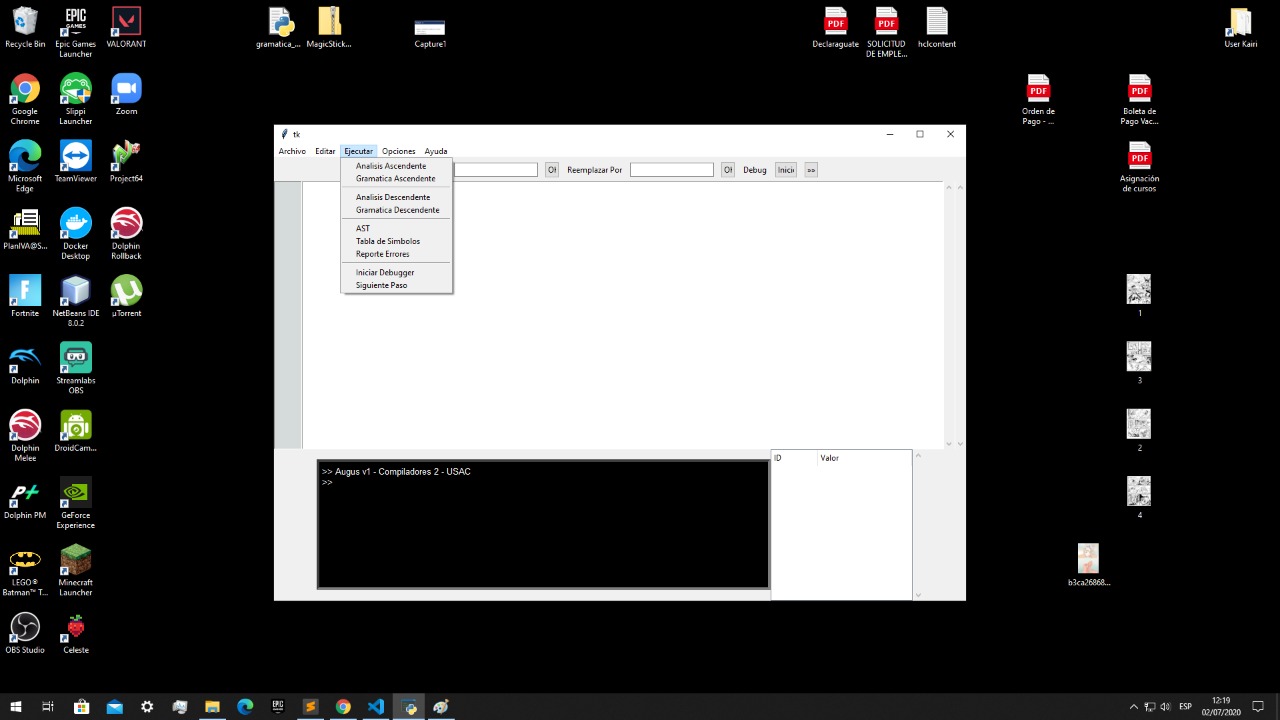
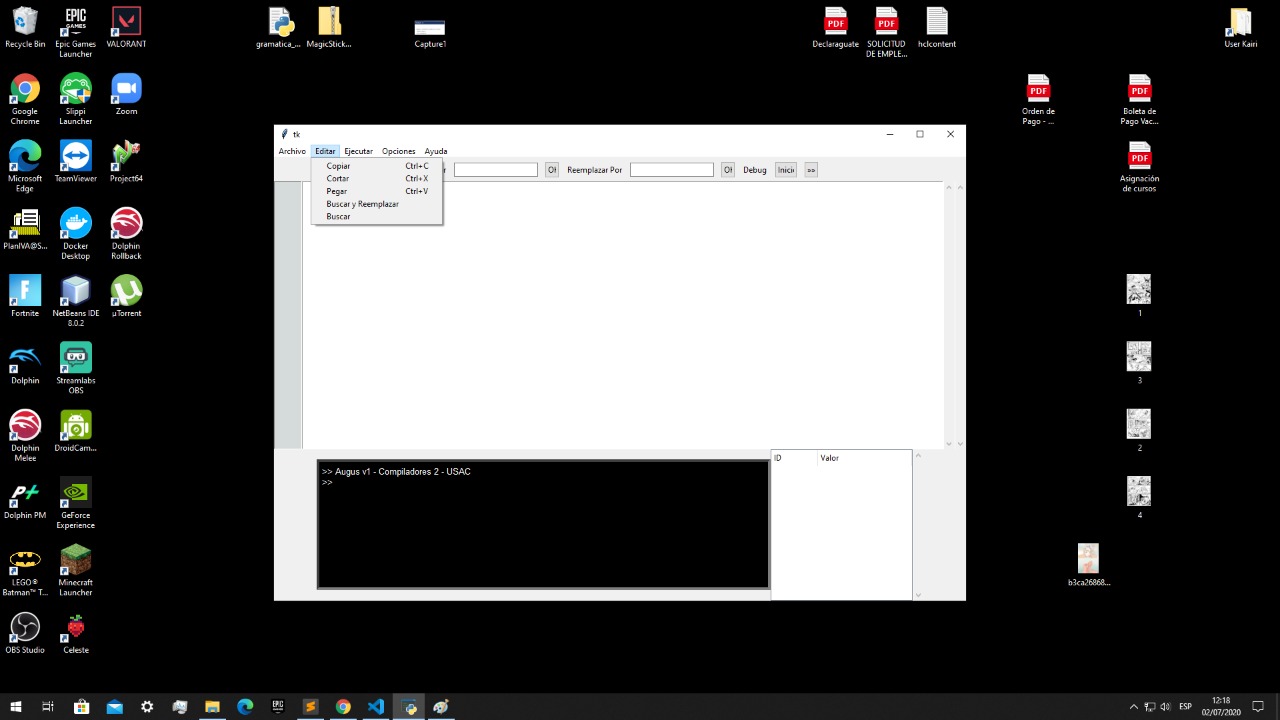
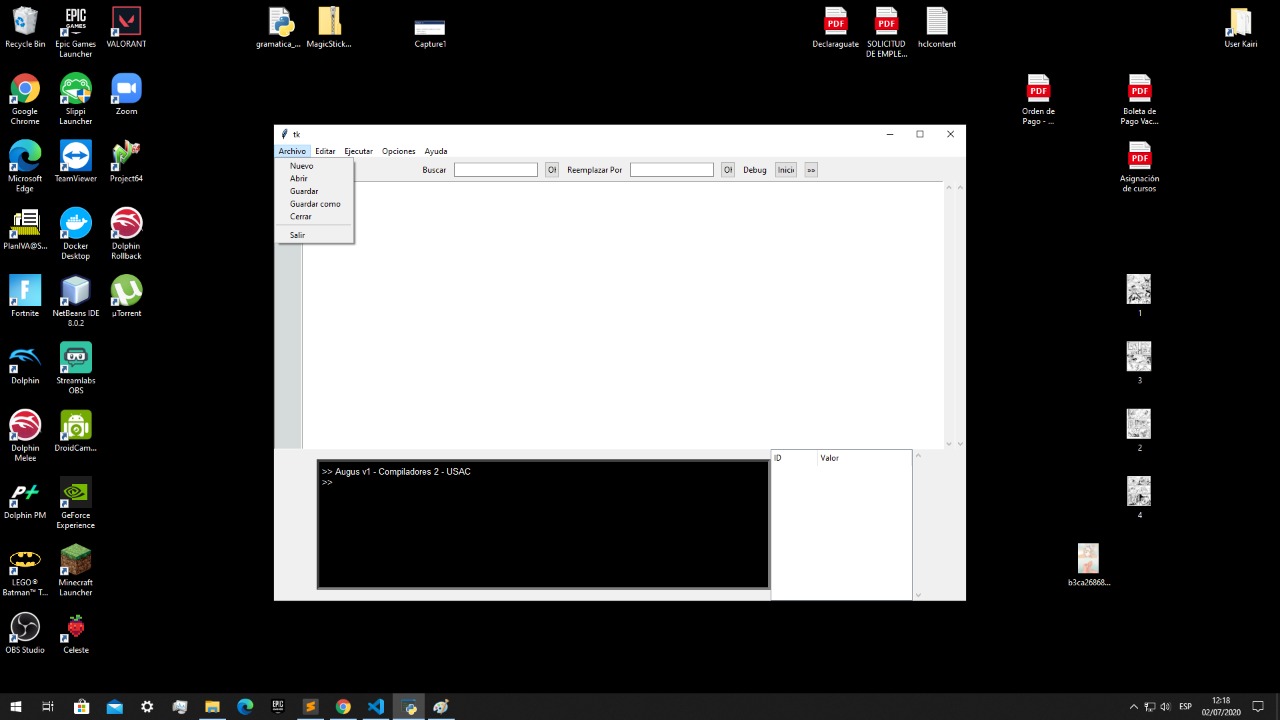
Construir un analizador léxico y sintáctico, implementando la traducción dirigida por la sintaxis y reglas semánticas con atributos sintetizados, que permita la traducción de un código de alto nivel (MinorC) a un código de tres direcciones con la sintaxis Augus.

# REQUERIMIENTOS DEL SISTEMA

* Python3.8
* Ply
* Graphviz

# CONTENIDO

A continuación, se presenta la explicación de las características de lo que se puede realizar con el traductor de MinorC, utilizando el siguiente ejemplo.



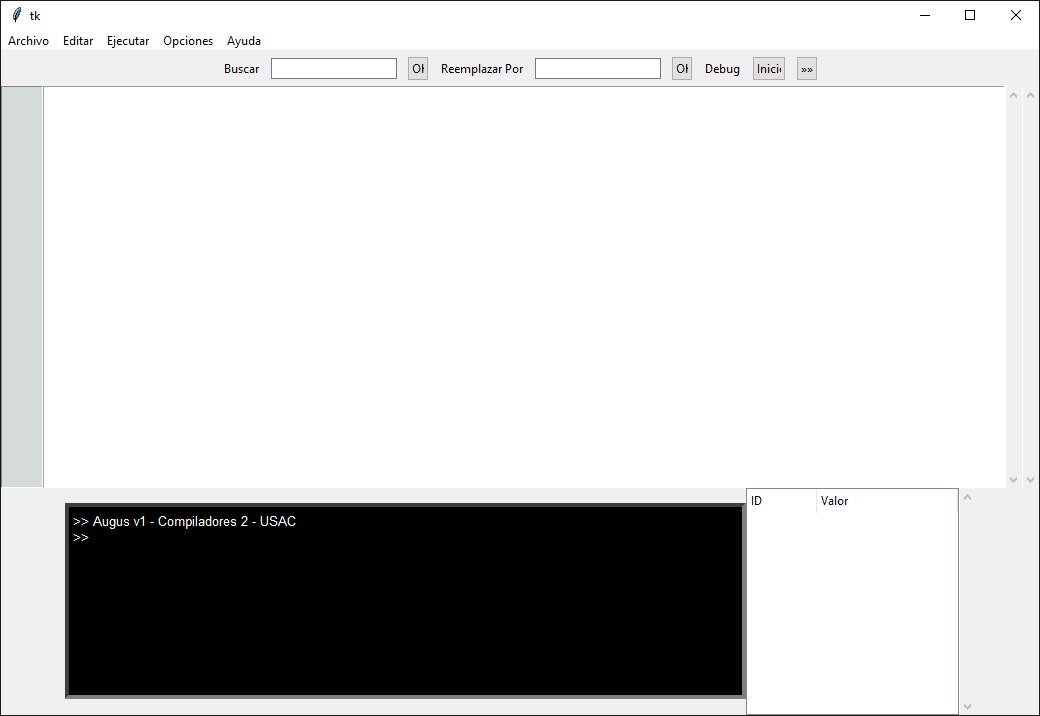
## IDE

En la pantalla IDE encontrará las categorías y opciones:

* Archivo: nuevo, abrir, guardar como, cerrar y salir.
* Editar: copiar, pegar, cortar, buscar, reemplazar.
* Ejecutar: analizador sintáctico ascendente y descendente.
* Opciones: cambiar de color de fondo, quitar los números de línea.
* Ayuda: ayuda y acerca de.

## Consola

En la pantalla IDE se encuentra la consola para ejecutar la interpretación del código.



## Flujo

El procedimiento para poder realizar la interpretación es el siguiente:

1. Ingresar el código en MinorC ya sea por archivo o editado.
2. Ejecutar el código con el menú y/o con un ícono (o paso a paso).
3. Observar el resultado de la ejecución que se muestra en la consola.
4. Transformar el código de MinorC a código de Augus.
5. Optimizar el código de Augus.
6. Mostrar el resultado de la ejecución en consola.
7. Opcionalmente, acceder a los reportes.

## Resaltado de sintaxis

Durante el ingreso de código se resaltará la sintaxis de.

## Sintaxis de MinorC

MinorC es un subconjunto de C.

Puede ver la sintaxis de MinorC en:

## Reportes

Los reportes se generan después del a ejecución, ya sea correcta o incorrecta y se muestran en x; estos reportes pueden ser:

* Reporte de errores.
* Reporte de tabla de símbolos-
* Reporte del AST y del GDA.
* Reporte gramatical.

### Reportes de errores

En el reporte de error se les mostrará el tipo (error de léxico, error sintáctico, error semántico), el número de línea y la descripción.

### Reportes de la tabla de símbolos

El reporte le mostrará la tabla de símbolos después de la ejecución de un archivo, incluyendo las variables, funciones y procedimientos, especificando: identificador, tipo, dimensión, valor, declarada en (ámbito), y referencias (estructura interna).

### Reportes del AST y del GDA

Este reporte le presenta en una pestaña nueva, el árbol sintáctico abstracto como una imagen o documento. No aplica para el árbol sintáctico.

### Reportes gramaticales

Este reporte se genera automáticamente, en el encontrará una explicación, indicando que expresiones se utilizaron, precedencia, símbolos terminales y no terminales, y las reglas semánticas.

# RESPONSABLES

Asignación del proyecto: M.Sc. Luis Espino

Programador: Kairi Itzmar Alexander Franco Medina

Encargados de revisión: Juan Carlos Maeda, Pavel Vásquez