

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
ORGANIZACIÓN DE LENGUAJES Y COMPILADORES 1
SEGUNDO SEMESTRE 2021
CATEDRATICO: ING. KEVIN LAJPOP
TUTOR ACADEMICO: SANDRA JIMÉNEZ

PROYECTO No.2
[MANUAL DE USUARIO]



Lester Efrain Ajucum Santos 201504510

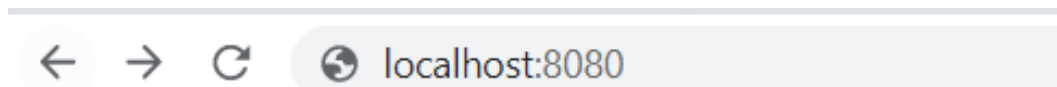
MANUAL DE USUARIO

PROYECTO SYSCOMPILER

El presente proyecto fue desarrollado para que los alumnos del curso de Introducción a la Programación y Computación 1, aprendan a programar y tener conocimiento de todas las generalidades de un lenguaje de programación.

UTILIZANDO LA WEB APP

Para poder utilizar la aplicación web, es necesario que el usuario se dirija al navegador web como Google Chrome, Mozilla Firefox, Microsoft Edge y escribir la siguiente dirección localhost:8080



Al ingresar a la dirección web se mostrará la pantalla de inicio de la aplicación.



La aplicación está dividida en diferentes áreas:

En la parte superior se encuentra la barra de herramientas.

Al centro se encuentra el editor de texto, la consola de la aplicación.

En la parte inferior se encuentra el área de reportes en la cual el usuario puede visualizar con más detalle los resultados obtenidos al ejecutar el código fuente.

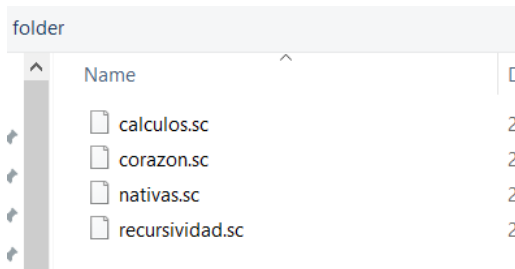
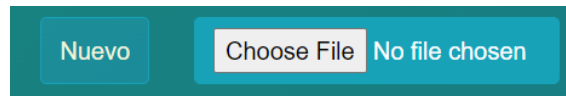
EDITOR DE TEXTO

Una vez iniciada la aplicación el usuario podrá acceder a las opciones que proporciona el editor de texto de la aplicación.

OPCIONES DEL EDITOR DE TEXTO

Abrir Archivo

El editor tiene la capacidad de abrir únicamente archivos **.sc**

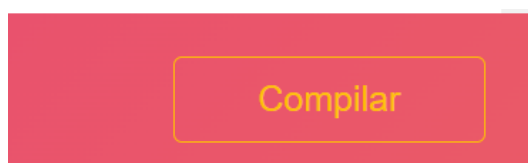


```
void metodo1(){
    //llamada del metodo
    figural(10);
}
void figural(int n) {
    String cadenaFigura = "";
    double i;
    i=-3*n/2;
    //iniciando dibujo
    while(i<=n){
        cadenaFigura = "";
        double j;
        j=-3*n/2;
        while(j<=3*n){
            double absolutoi;
            absolutoi = i;
            double absolutoj;
            absolutoj = j;
            if(i < 0)
            {
                absolutoi = i * -1;
            }
            if(j < 0)
            {
                absolutoj = j * -1;
            }
            if((absolutoi + absolutoj < n)
            || ((-n / 2 - i) * (-n / 2 - i) + (n / 2 - j) * (n / 2 - j) <= n * n / 2)
            || ((-n / 2 - i) * (-n / 2 - i) + (-n / 2 - j) * (-n / 2 - j) <= n * n / 2)) {
                cadenaFigura = cadenaFigura + " ";
            }
            else
            {
                cadenaFigura = cadenaFigura + ". ";
            }
            j=j+1;
        }
        WriteLine(cadenaFigura);
        i=i+1;
    }
    WriteLine("Si la figura es un corazón, te aseguro que tendrás un 100 :3");
}
Start WITH metodo1();
```

HERRAMIENTAS

Ejecutar

Se encarga de ejecutar la lógica del programa, generando el análisis léxico, sintáctico y semántico, además es el encargado de ejecutar cada una de las sentencias.



Consola

Esta área sirve para controlar el flujo del programa, permite visualizar errores dentro del código fuente, mensajes y todo lo que sea indicado dentro del lenguaje.

```
i = 3
i = 4
i = 5
estamos en el caso = 0
estamos en el caso = 1
estamos en el caso = 2
0
2
Funcion de Ackerman (3, 4) = 125
***ERROR: La variable x no existe en la tabla de simbolos. En la linea 87 y columna 1
***ERROR: No se puede ejecutar la sentencia break dentro de la sentencia if. En la linea 88 y columna 2
```

```
Funcion de Ackerman (3, 4) = 125
***ERROR: La variable x no existe en la tabla de simbolos. En la
***ERROR: No se puede ejecutar la sentencia break dentro de la
```

Reportes de errores

Se muestran los errores semánticos, léxicos y sintácticos detectados durante el análisis del código fuente.

Tabla de Símbolos

AST

Reporte Errores

Tabla de Errores

Tipo de Error	Descripción	Línea	Columna
Lexico	El carácter @ no forma parte del lenguaje	79	78
Lexico	El carácter # no forma parte del lenguaje	90	3
Semantico	La variable x no existe en la tabla de símbolos	87	1
Semantico	No se puede ejecutar la sentencia break dentro de la sentencia if.	88	2

Semantico	La variable x no existe en la tabla de simbolos
Semantico	No se puede ejecutar la sentencia break dentro de la sentencia

En la primera columna se detalla el rol del símbolo, este puede ser de tipo variable, función, método, vector o lista.

En la segunda columna se muestra el nombre del símbolo.

En la tercera columna se muestra el tipo de dato que el símbolo posee, en este caso puede ser de tipo entero, doble, cadena, carácter o booleano.

En la cuarta columna se detalla el ámbito en el que el símbolo se encuentra declarado. Los símbolos pueden tener un ámbito global o local, dependiendo donde se encuentren ubicados dentro de las sentencias del programa.

En la quinta columna se despliega el valor que el símbolo guarda, este debe concordar con el tipo de dato con el que el símbolo fue declarado.

Y por último en la sexta columna se muestra la cantidad de parámetros, este atributo solo aplica para aquellas sentencias con un rol de función o método.

Tabla de Símbolos

AST

Reporte Errores

Tabla de Símbolos

Rol	Nombre	Tipo	Ambito	Valor	Parametros
funcion	ackerman	entero	global	...	2
variable	i	entero	global	3	...
variable	dob1	decimal	global	60.65	...
variable	dob2	decimal	global	60.65	...
variable	dob3	decimal	global	309.6	...
variable	dob4	decimal	global	60.65	...
variable	cadena	cadena	global	Hola MUNDO	...
variable	ternario	booleano	global	false	...