

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

PROYECTO No.2
[MANUAL TÉCNICO]



Lester Efrain Ajucum Santos 201504510

PROYECTO SYSCOMPILER

Es un proyecto realizado para la Escuela de Ciencias y Sistemas de la Facultad de Ingeniería, que consiste en un lenguaje para los estudiantes del curso de Introducción a la programación y computación 1.

ARQUITECTURA GENERAL DEL PROYECTO

El siguiente proyecto se maneja una arquitectura Cliente-Servidor, con el objetivo de que pueda separar los servicios administrados por el intérprete, de la aplicación cliente que se mostrará al usuario final.



ENTORNO DE TRABAJO

CLIENTE

Se desarrolla la siguiente web app, cuya finalidad es proporcionar ciertas funcionalidades, características y herramientas de utilidad al usuario.

La función principal es proveer de un editor para el ingreso de código fuente en el lenguaje antes mencionado y analizar cada una de sus instrucciones.

```
79 writeline("Funcion de Ackerman (" + m + ", " + n + ") = " + ackerman(m, n));@
80
81 int ackerman(int m, int n) {
82     return (m == 0 ? n + 1 : (m > 0 && n == 0 ? ackerman(m - 1, 1) : ackerman(m - 1, ackerman(m, n - 1))));
83 }
84
85
86 //-- ERRORES SEMANTICOS
87 x = 20;
88 if(true){
89     break;
90 }#
```

El código se muestra en un editor de texto con una barra lateral de línea de números de 79 a 90. En la parte inferior derecha del editor, hay un botón de color naranja con el texto "Compilar".

Sección de reportes

En la sección de reportes se muestra un resumen del análisis realizado al código fuente.

Reportes de errores

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

Tabla de Símbolos

AST

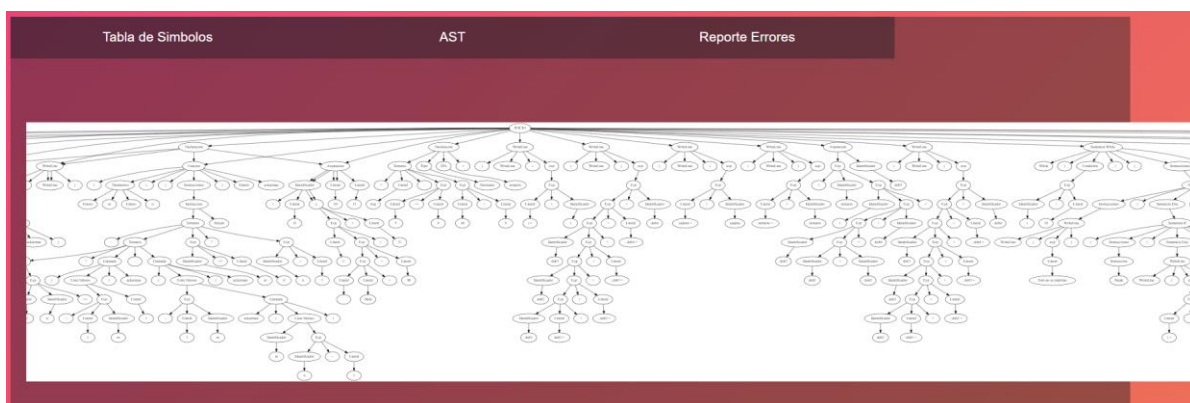
Reporte Errores

Tabla de Errores

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

Reportes de árbol de análisis sintáctico

El siguiente reporte despliega una imagen detallada del árbol de análisis sintáctico para el código fuente. Está es una representación abstracta de la estructura del código en el lenguaje syscompiler.



Reportes de tabla de simbolos

El siguiente reporte es un resumen de todos los símbolos como variables, métodos y funciones que han sido declaradas dentro del flujo del programa.

Tabla de Simbolos					
AST					
Reporte Errores					
Tabla de Simbolos					
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	...

Área de 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
```

SERVIDOR

NodeJs web server

Web server desarrollado con NodeJS en su versión 13.7.0

v13.7.0

Package manager NPM

Npm en su versión 6.13.6

6.13.6

Typescript version

Typescript en su versión 4.0.3

Version 4.0.3

HERRAMIENTAS DE ANALISIS LEXICO Y SINTACTICO

Jison

Generador de analizador sintáctico y léxico en lenguaje Javascript.

SERVICIO DE COINTEINERIZACIÓN

Docker Desktop es una aplicación desarrollada para ambientes Mac o Windows, que permite la creación y despliegue de aplicaciones contenerizadas y microservicios.



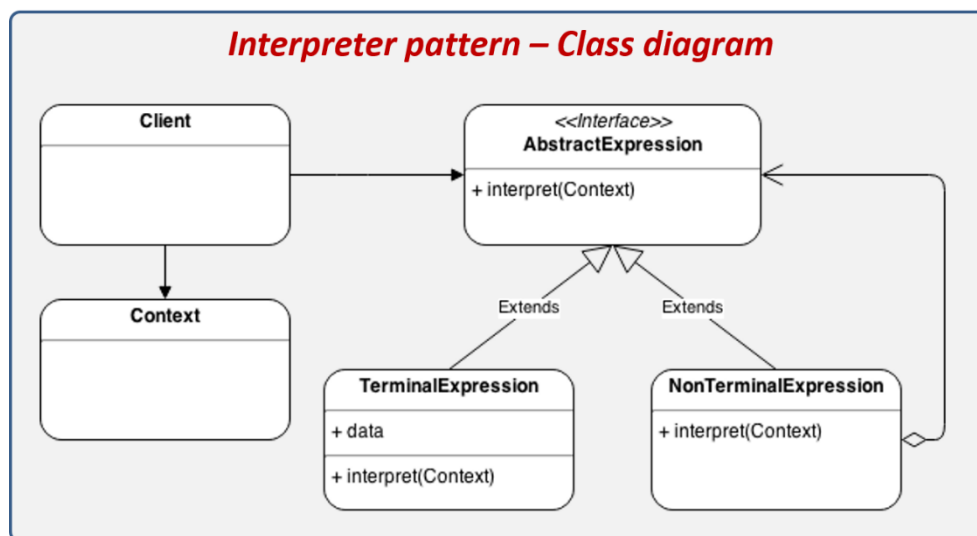
REQUERIMIENTOS DEL SISTEMA PARA INSTALACIÓN DE DOCKER EN WINDOWS

- Windows 11 64-bit: Home o Pro version 21 H2 en adelante, o la version Enterprise o Education version 21H2 en adelante.
- Windows 10 64-bit Home o Pro 2004 (build 19041) o mayor.
- Habilitar la característica WSL2 en Windows.
- Procesador 64 bit
- 4 GB de RAM
- Virtualización por hardware a nivel de BIOS.
- Descargar e instalar el paquete de actualización de Kernel para Linux.

PATRÓN INTERPRETE

El patrón interprete se utiliza para desarrollar un intérprete para un lenguaje.

Uno de los requisitos para implementar el patrón interprete es el conocimiento de gramáticas formales. Además, para su funcionalidad se deben combinar técnicas de programación orientada a objetos avanzada.



LENGUAJE PARA PROYECTO 2 (SYSCOMPILER)

PALABRAS RESERVADAS (KEYWORDS)

TIPOS DE DATOS

Int
Double
Boolean
Char
String
True
False

ESTRUCTURAS DE DATOS

Operador new

new

Listas dinamicas

DynamicList
append
getValue
setValue

SENTENCIAS DE CONTROL

if
else
switch
case
default

SENTECIAS CICLICAS

while
for
Do

SENTENCIAS DE TRANSFERENCIA

sentencia break

break
continue
return

*****METODOS & FUNCIONES*****

void
WriteLine
toLowerCase
toUpperCase
length
truncate
round
typeof
toString
toCharArray
start
with

SIMBOLOS ACEPTADOS

. punto - dot
, coma - comma
; punto y coma - semicolon
: dos puntos - colon
[corchete que abre - left square bracket
] corchete que cierra - right square bracket
(parentesis que abre - left parentheses
) parentesis que cierra - right parentheses
{ llave que abre - left brace - left bracket
} llave que cierra - right brace - right bracket
= igual - equal (Asignacion/Assignment)

Operadores Relacionales

<= Menor o igual - less than or equal to
< Menor que - less than
>= Mayor o igual - greater than or equal to
> Mayor que - greater than
== Igualacion - Equal to
!= Diferencia - not equal

Operador Ternario

?

Operadores lógicos

&& And - and

|| Or - or

! not - not

Operadores Aritméticos

-- menosmenos - decrement

++ masmas - increment

+ suma - plus

- resta - subtraction

* multiplicacion - multiplication

/ division - division

^ potencia - exponentiation

% Modulo - modulus/remainder

- Unario - unary negation

PRECEDENCIA DE OPREADORES (DE MAYOR A MENOR)		
OPERADOR		ASOCIATIVIDAD
masmas/incremento	++	n/a
menosmenos/decre	--	n/a
negacion/not	!	right to left
unario/unary	-	
masmas/prefijo	++	
menosmenos/prefijo	--	
potencia	^	
Multiplicacion	*	left to right
division	/	
modulo	%	
suma	+	left to right
resta	-	
menor que	<	left to right
menor igual que	<=	
mayor que	>	
mayor igual que	>=	
igualdad	==	left to right
diferencia	!=	
y/and	&&	left to right
o/or		left to right