

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

CENTRO UNIVERSITARIO DE OCCIDENTE

DIVISIÓN DE CIENCIAS DE LA INGENIERÍA



MANUAL TÉCNICO DE PROYECTO FINAL

**LABORATORIO DE
ORGANIZACIÓN DE LENGUAJES Y COMPILADORES 1**

PRESENTADO POR:

UMAÑA DE LEÓN, WILLIAM RODRIGO

(201931448)

DOCENTE:

Ing. JOSE MOISES GRANADOS GUEVARA

AUXILIAR

Bach. BRAYN MISAEL MONZON FUENTES

QUETZALTENANGO – QUETZALTENANGO – GUATEMALA

02-04-2021

Índice

Índice	2
Contenido	3
Información de la aplicación	3
Detalle proyecto	3
Analizadores	3
Léxico GCIC	3
Sintáctico GCIC	4

Contenido

Información de la aplicación

El proyecto fue escrito en el IDE Netbeans versión 12.0 y en Android Studio y contiene las siguientes especificaciones.

Lenguajes utilizados: java, jFlex, java Cup, kotlin.

Nombre del proyecto: proy1compi2

Versión del proyecto: 1.0-SNAPSHOT

Aunque el proyecto contiene 2 partes se toma a todo como uno solo.

Hay que tener en cuenta que para correr el proyecto es necesario tener Android Studio y Netbeans 12 de preferencia aunque se puede utilizar otro IDE para ejecutar el servidor. En el documento de instalación y desinstalación se explica más a detalle que se tiene que realizar para poder correrlo.

Analizadores

Léxico Compilar

Este analizador es el encargado de leer la entrada por el usuario en la página principal

Se utilizaron las siguientes reglas léxicas:

`/*-----Comentarios-----*/`

`comentariolin = \>\>[^\\n\\r]*`

`comentariobloc = \<\-[^->]*\\->`

`comentario = ({comentariolin}|{comentariobloc})`

`letra = [a-zA-Z]`

`digito = [0-9]`

`caracter = \'#\^[^\\]\'`

`cadena = \"^[^\"\\]*\"`

`numero = {digito}+([.]{digito}{1,6})?`

`espacio = \" \"`

`saltoLinea = \\n\\r\\r\\n`

`espacioblanco = ({espacio}|{saltoLinea}|[\\t\\n])+`

`identificador = ({letra}|\"_\")({letra}|{digito}|\"_\")*`

`notas = \"do#\"|\"re#\"|\"fa#\"|\"sol#\"|\"la#\"|\"do\"|\"re\"|\"mi\"|\"fa\"|\"sol\"|\"la\"|\"si\"`

La palabras reservadas utilizadas son:

true,false,extiende,keep,var,arreglo,si,sino,sino

si,switch,caso,default,para,mientras,hacer,continuar,retorna,Reproducir,Esperar,Sumarizar,Longitud,Mensaje,Principal,lista,nombre,random,circular,pistas,salir

Los simbolos reservados son los siguientes: :, !, <, >, {, }, [,] _, -, \$, ', ', (,), =

Sintáctico Compilar

$G = \{N, T, P, L\}$

$N = \{\text{inicio}\}$

$T = \{\text{Son todos los anteriormente descritos en el lexer indigo como palabras reservadas y reglas lexicas}\}$

$L = \{\text{inicio}\}$

Siguiente	Pista id ext salto
decl	Keep declp
	declp
declp	var tipoD id declp2 asgF salto
	var tipoD arreglo id declp2 [expr1] arr arrFin salto
declp2	, id declp2
	lambda
asg	id += expr1 salto
	id = expr1 salto
	id[expr1] arr2 = expr1 salto
condicional	si (condicion)
	sino si (condicion)
	sino
cambiar	switch (variable) salto
	caso expr1 salto
	default
	salir

ciclo1	para (asg ; condicion; id op2)
ciclo2	mientras(condicion)
	hacer
	continuar
funcion	Keep tipoD id (par)
	tipoD id (par)
	id(par)
funciones2	Ordenar(id, tipoOr)
	Sumarizar(arr3)
	Longitud(arr3)
	Mensaje(expr1)
funciones3	Reproducir(nota, expr1, expr1, expr1)
	Esperar(expr1, expr1)
	Principal()

ext	extiende id extp
	lambda
expr1	id fin1
	id[expr1] fin1
	num fin1
	cadena fin1
	caracter fin1
	true fin1
	false fin1
	(expr1) fin1
	!expr1
asgF	= expr1
	lambda
arr	[expr1] arr

	lambda
arrFin	= arrFinP
	lambda
arrFinP	{arrFinP2}
arrFinP2	arrFinP fin2
	expr1 fin3
arr2	[expr1] arr2
	lambda
condicion	opNul expr1 op3 opNul expr1 opLog
	opNul expr1
par	lambda
	tipoD id fin4
arr3	id
	arrFinP

extp	, id extp
	lambda
tipoD	entero
	doble
	boolean
	caracter
	cadena
fin1	op expr1
	op2 expr1
	op3 expr1
	op4 expr1

	lambda
fin2	, arrFinP
	lambda
fin3	, expr1 fin3
	lambda
opLog	op4 condicion
	e
variable	id
	id[expr1]
fin4	, par
	lambda
tipoOr	ascendente
	descendente
	pares
	impares
	primos

op	+
	-
	*
	/
	^
	%
op2	++

	--
op3	==
	!=
	>
	<
	>=
	<=
opNul	!!
	lambda
op4	&&
	!&&
	!
	&
nota	do
	re
	mi
	fa
	sol
	la
	si
	do#
	re#
	fa#
	sol#
	la#

Lexer Pintar

Este lexer pintar se utilizó para poder pintar las palabras dentro de la aplicación de escritorio que genera el servidor.

Para este lexer únicamente se utilizaron unas las reglas léxicas parecidas al Lexer Compilar

comentariolin = \>\>[^\\n\\r]*

si,switch,caso,default,para,mientras,hacer,continuar,retorna,Reproducir,Esperar,Sumarizar,Longitud,Mensaje,Principal,lista,nombre,random,circular,pistas,salir

