

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

CENTRO UNIVERSITARIO DE OCCIDENTE

DIVISIÓN DE CIENCIAS DE LA INGENIERÍA



MANUAL TÉCNICO DE PRACTICA 2

**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

15-04-2021

Índice

Índice	2
Contenido	3
Información acerca del proyecto	3
Diagrama de clases	3
Detalle de organización de proyecto	3
Analizador creado con Jison	4
Analizador Léxico Wison	4
Analizador Sintáctico Wison	5

Contenido

Información acerca del proyecto

El proyecto fue creado con NodeJs y Angular, se utiliza javascript y typescript para comunicarse entre archivos al igual que html y scss para mostrar las páginas web.

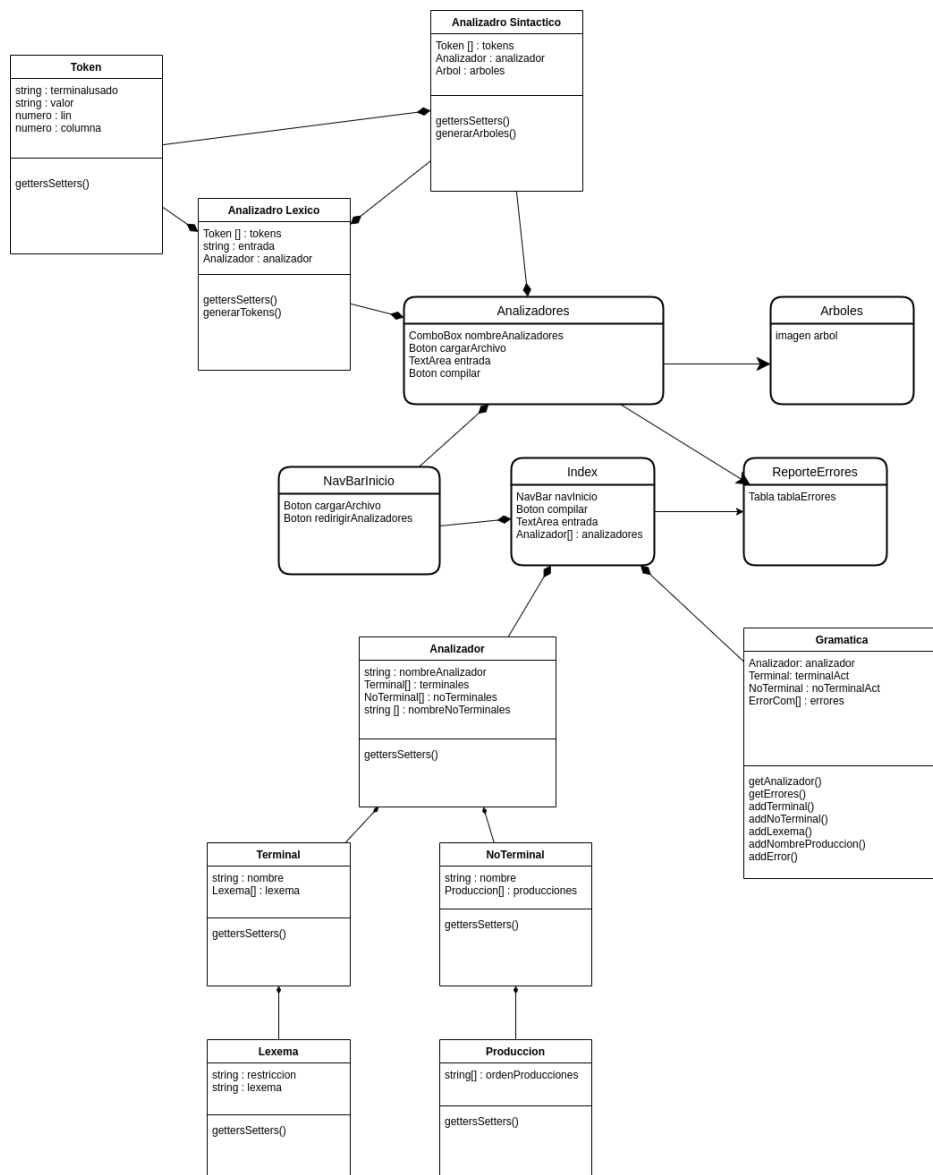
Las versiones con las que se trabajaron son las siguientes:

NodeJS : v10.19.0

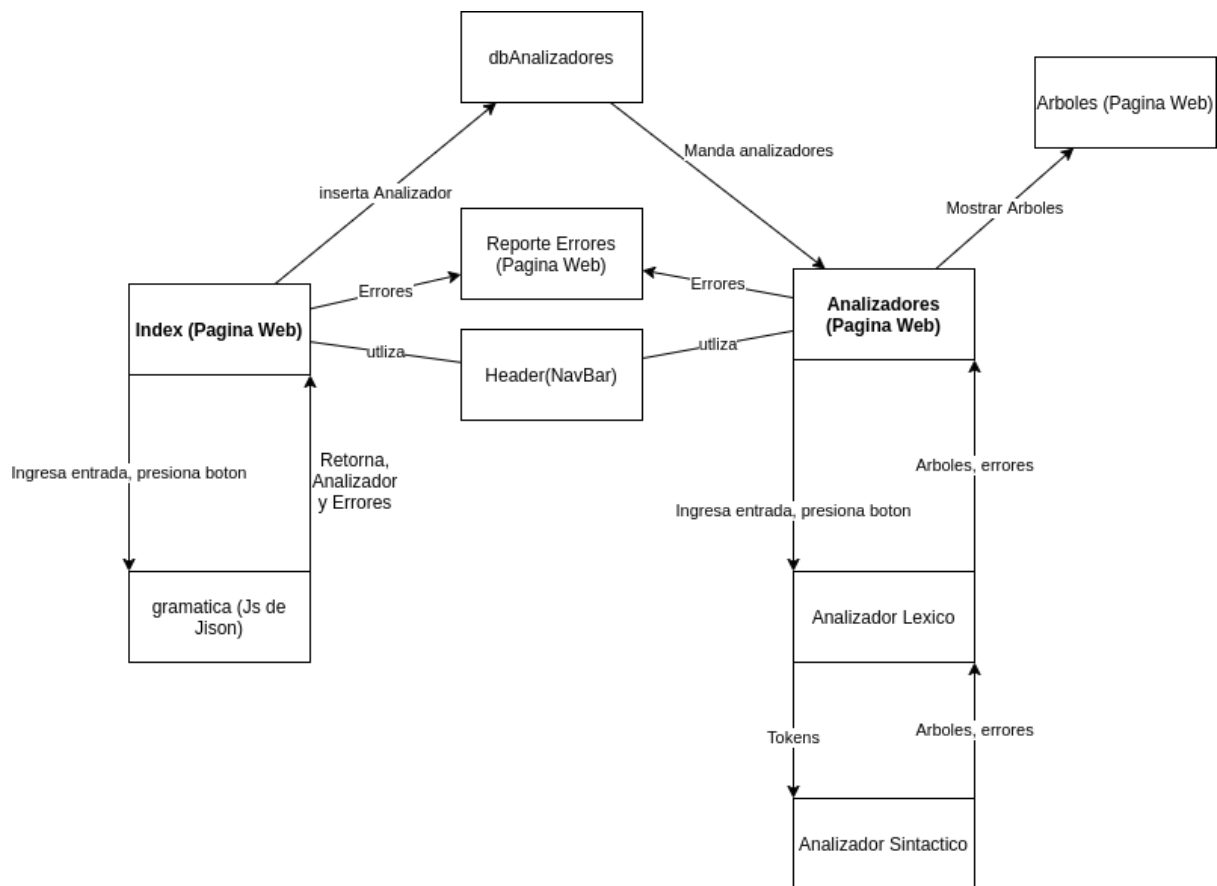
Angular : Angular CLI: 11.2.8

TypeScript : 4.2.3

Diagrama de clases



Detalle de organización de proyecto



Analizador creado con Jison

El analizador creado con jison es el que se utiliza para validar y obtener los valores que forma el lenguaje wison que se da en el enunciado del proyecto. A continuación se describen el analizador lexico y sintactico de este analizador completo.

Analizador Léxico Wison

Se utilizaron las siguientes reglas léxicas:

TERMINAL = \\$[a-zA-Z0-9_]+

NO_TERMINAL = %[a-zA-Z0-9_]+

ALFAB = "[aA-zZ]"

TODOS_NUM = "[0-9]"

CADENA = \'([a-zA-Z0-9\[\]\{\}\+\-*/:;¿?\\%()#][!@\$>\\^,.)+\\'

COMENTARIO_1 = \#([a-zA-Z0-9\[\]\{\}\+\-*/:;¿?\\%()#_<=>][!@\$>\\^,.)][]+

COMENTARIO_2 = **([a-zA-Z0-9\[\]\{\}\+\-*/:;¿?\\%()#_<=>][!@\$>\\^,.)[\\r\\n])**\\

FLECHA = "<-"

FLECHA_DOB = "<="

Tambien se utilizaron las siguientes palabras reservadas:

Wison, Terminal, No_Terminal, Syntax, Lex, Initial_Sim.

Al igual que se utilizaron los siguientes símbolos reservados:

¿, ?, {, }, (,), :, ;, *, +, |.

Analizador Sintáctico Wison

Para la creación del analizador sintáctico se utilizó la siguiente gramática {N,T,S,L}

N = {inicio,sig,sigLex,sigSyntax,cads,rex, prod,}

T = {Son todos los anteriormente descritos en el analizador léxico}

L = {inicio}

P =

inicio	Wison ¿ sig ?Wison
--------	--------------------

sig	Lex {: sigLex :} sig
	Syntax {{: sigSyntax:}} sig
	e

sigLex	Terminal terminal <- cads ; sigLex
	e

sigSyntax	No_Terminal noTerminal ; sigSyntax
	Initial_Sim noTerminal ; sigSyntax
	noTerminal <= prod ; sigSyntax
	e

cads	cadena rex
	alfab rex
	todosNum rex
	terminal
	(cads) cads
	e

prod	noTerminal prod
	terminal prod
	prod
	e

rex	*
	+
	?
	e