

Universidad de San Carlos de Guatemala USAC.
División de Ciencias de la Ingeniería.
Centro Universitario de Occidente CUNOC.
Compiladores 2



Estudiante

Elvis Lizandro Aguilar Tax

Carnet

201930304

Herramientas de Desarrollo

Desarrollo backend:

- Lenguaje :
 java
 versión: 1.8 — 11.0.2
 Spring-Boot
 versión: v2.7.18
- IDE
 - Netbeans version 12.0

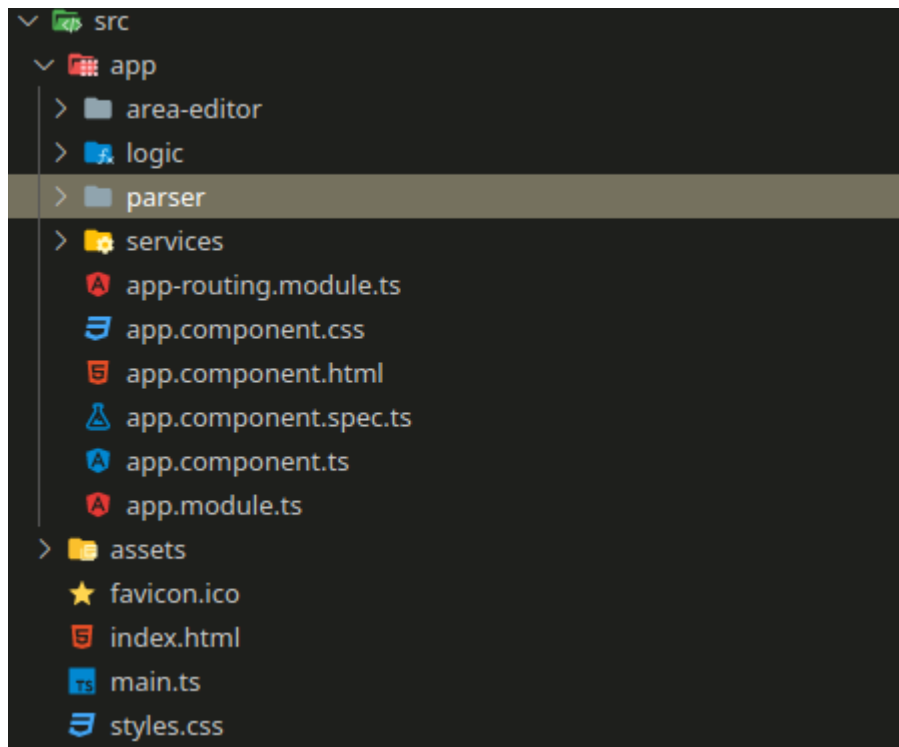
Desarrollo frontend:

- Lenguaje :
 Typescript
 versión: –
 Angular
 Angular CLI: 16.2.3
 Node: 18.18.0
 Package Manager: npm 9.8.1
- Analizador Léxico y sintáctico
 - **Jison**
 - version: 1.8.2
 - documentacion oficial: <https://gerhobbelt.github.io/jison/docs/>
 -
- IDE
 - Visual Studio code

Sistema operativo

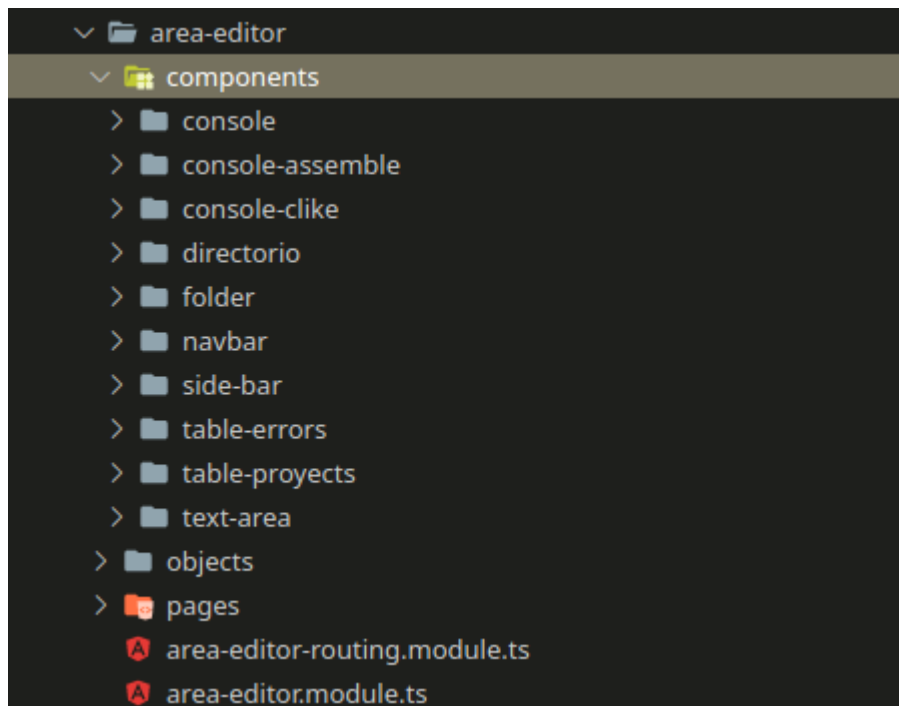
- OS: Manjaro Linux X86_64

Organización del proyecto (Paquetes)

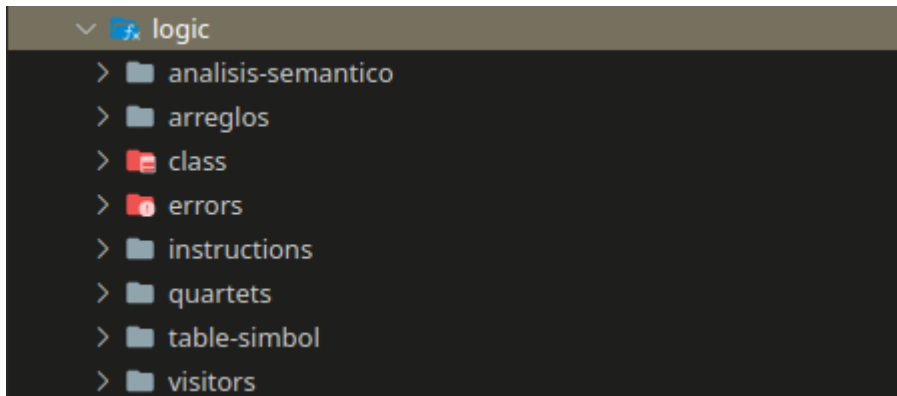


src: es el paquete principal del proyecto

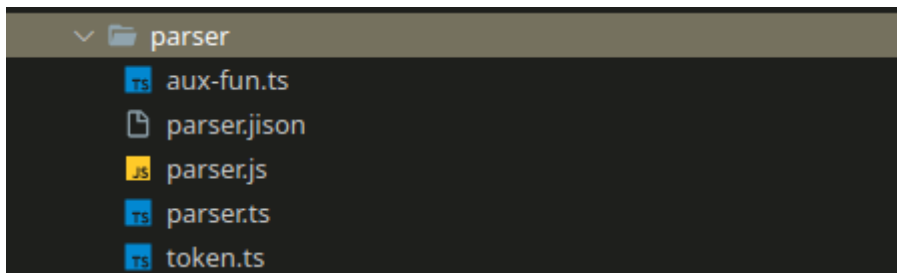
área-editor: paquete encargado de la parte IU, lógica y páginas (frontend puro)



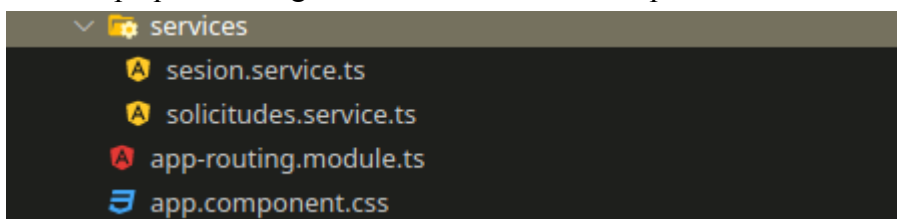
logic: Paquete encargado de la lógica del parser, es decir, representación de instrucciones, AST, generador de código 3 direcciones, validaciones semánticas.



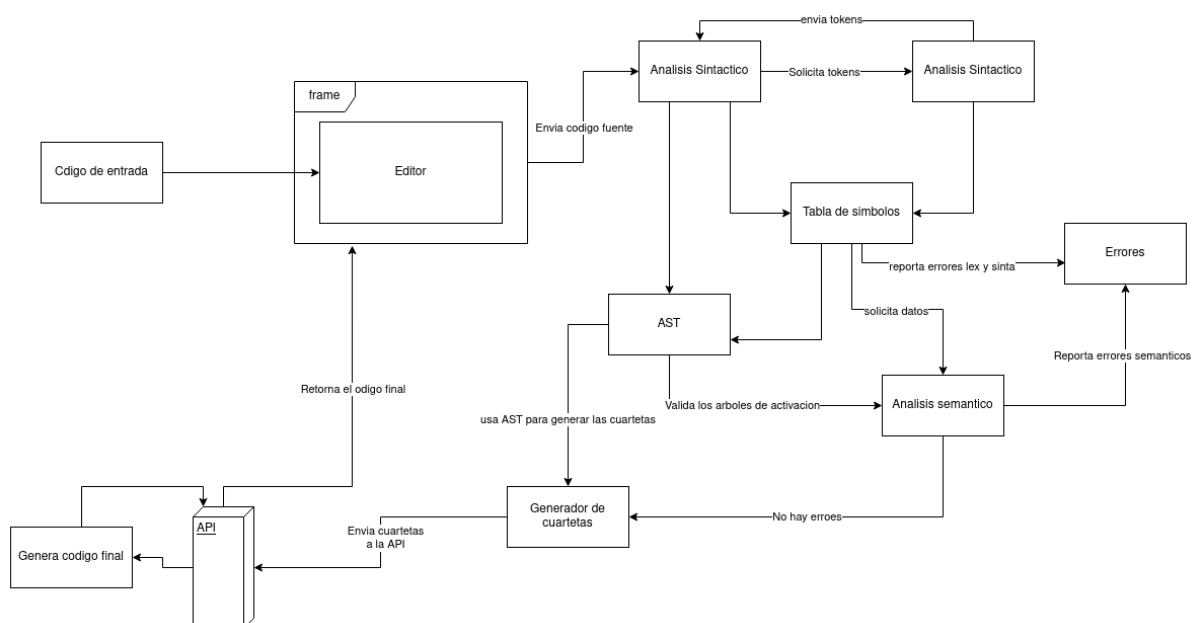
parser: Paquete encargado del parser, archivos jison, parser.js, parser.ts



service: paquete encargado de los servicios de la aplicación



Organización project



Analisis Lexico

*Expresiones regulares para tokens

1. Saltos de línea y comentarios

lineTerminator	\r\n\r\n
whitespace	{lineTerminator} [\t\f]
comenLine	"/"/".*\n
comenMultiLine	\\/*[s\S]*?*/

2. Palabras reservadas

int	"int"
float	"float"
string	"String"
char	"char"
boolean	"boolean"
import	"import"
public	"public"
private	"private"
class	"class"
final	"final"
static	"static"
void	"void"
main	"main"
package	"package"
protected	"protected"
getter	"@Getter"
setter	"@Setter"
println	"System.out.println"
print	"System.out.print"
true	"true"
false	"false"
new	"new"
continue	"continue"
return	"return"
this	"this"
null	"null"
extends	"extends"

—operadores de clase Math

mathabs	"Math.abs"
mathceil	"Math.ceil"
mathfloor	"Math.floor"

mathround	"Math.round"
mathmax	"Math.max "
mathmin	"Math.min"
mathpow	"Math.pow"
mathsqrt	"Math.sqrt"
mathrandom	"Math.random"
mathtoRadians	"Math.toRadians"
mathacos	"Math.acos "
mathsin	"Math.sin"
mathatan	"Math.atan"
mathexp	"Math.exp"

–bifurcaciones

if	"if"
else	"else"
switch	"switch"
case	"case"
break	"break"
default	"default"
while	"while"
do	"do"
for	"for"

–lectura por consola

readfloat	"readFloat"
readint	"readInt"
readchar	"readChar"
readboolean	"readBoolean"
readstring	"readString"

3. Operadores

–incremento decremento

incre	"++"
decre	"--"
masigul	"+="

–aritméticos

mas	"+"
menos	"_"
por	"*"
division	"/"

modulo	"%"
--------	-----

–relacionales

mayorOI	">="
menorOI	"<="
mayorQ	">"
menorQ	"<"
equals	"=="
diferente	"!="

//logicos

and	"&&"
or	" "
not	"!"

4. Signos

coma	","
parentesisA	"("
parentesisC	")"
puntoComa	";"
dosPunto	":"
igual	"="
punto	."
llaveA	"{"
llaveC	"}"
corchetA	"["
corchetC	"]"

5. ID

id	[a-zA-Z_][a-zA-Z_0-9]*
----	------------------------

Analisis Sintactico

Gramaticas

–configuracion de precedencia

%left OR

%left AND

%right NOT

%left EQUALS DIFERENTE MAYORQ MAYOROI MENORQ MENOROI

%left MAS MENOS

%left POR DIVISION MODULO

–producción inicial

%start initial

–Gramática para paquetes

pack

: PACKAGE ids_imprt PUNTOCOMA

|

;

–Gramática para imports

imprts

: imprts imprt

|

;

imprt

: IMPORT ids_imprt term_imprt

;

ids_imprt

: ids_imprt PUNTO ID

| ID

;

term_imprt

: PUNTO POR PUNTOCOMA

| PUNTOCOMA

;

–Gramática para clase

clase

: visi_class clas_name LLAVEA sente_glos LLAVEC

| getSet visi_class clas_name LLAVEA sente_glos LLAVEC

exti

: EXTENDS ID

|

;

clas_name

: CLASS ID exti

;

visi_class

: PUBLIC FINAL

| PUBLIC

