

Universidad de San Carlos de Guatemala
Facultad de Ingeniería

PROYECTO 2

Nathan Antonio Valdez²

202001568

OBJETIVOS

1.1 Objetivos Generales

Aplicar los conocimientos sobre análisis léxico y sintáctico de un compilador para desarrollar un intérprete sencillo funcional para el lenguaje especificado.

1.2 Objetivos Específicos

- Que el estudiante aprenda a generar un analizadores léxicos y sintácticos utilizando la herramienta JISON.
- Qué el estudiante comprenda el manejo de errores léxicos y sintácticos correctamente durante la interpretación.
- Que el estudiante realicé la implementación del patrón de diseño intérprete para el manejo de acciones gramaticales en javascript.

Requisitos del Sistema

Requisitos del Sistema Java 8

La información detallada sobre los requisitos de Java 8 está disponible en [Configuraciones de Java 8 soportadas.](#)

Windows

- Windows 10 (8u51 y superiores)
- Windows 8.x (escritorio)
- Windows 7 SP1
- Windows Vista SP2
- Windows Server 2008 R2 SP1 (64 bits)
- Windows Server 2012 y 2012 R2 (64 bits)
- RAM: 128 MB
- Espacio en disco: 124 MB para JRE; 2 MB para Java Update
- Procesador: Mínimo Pentium 2 a 266 MHz
- Exploradores: Internet Explorer 9 y superior, Firefox

Mac OS X

- Mac con Intel que ejecuta Mac OS X 10.8.3+, 10.9+
- Privilegios de administrador para la instalación
- Explorador de 64 bits

Se requiere un explorador de 64 bits (Safari, por ejemplo) para ejecutar Oracle Java en Mac.

Linux

- Oracle Linux 5.5+¹
- Oracle Linux 6.x (32 bits), 6.x (64 bits)²
- Oracle Linux 7.x (64 bits)² (8u20 y superiores)
- Red Hat Enterprise Linux 5.5+¹ 6.x (32 bits), 6.x (64 bits)²
- Red Hat Enterprise Linux 7.x (64 bits)² (8u20 y superiores)
- Suse Linux Enterprise Server 10 SP2+, 11.x
- Suse Linux Enterprise Server 12.x (64 bits)² (8u31 y superiores)
- Ubuntu Linux 12.04 LTS, 13.x
- Ubuntu Linux 14.x (8u25 y superiores)
- Ubuntu Linux 15.04 (8u45 y superiores)
- Ubuntu Linux 15.10 (8u65 y superiores)
- Exploradores: Firefox

Requisitos del sistema Solaris

Consulte las [configuraciones del sistema Java 8](#) soportadas para obtener información sobre las plataformas, sistemas operativos, administradores de escritorio y exploradores compatibles.

Flujo del Sistema

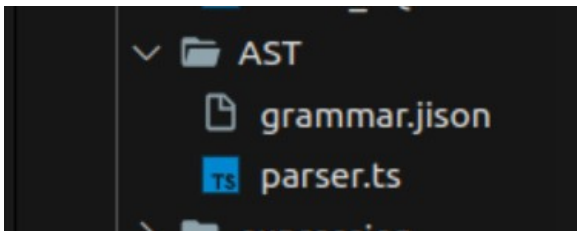
Primeramente cabe recalcar que se trabajo por medio de un servidor de typescript

creado con node js

Esto es únicamente el servidor de typescript, luego para el frontend se utilizo un el framework REACT

Luego en el servidor de typescript se crearon diferentes rutas para realizar las peticiones y al lograr tener acceso, se creó el código del interprete

AST

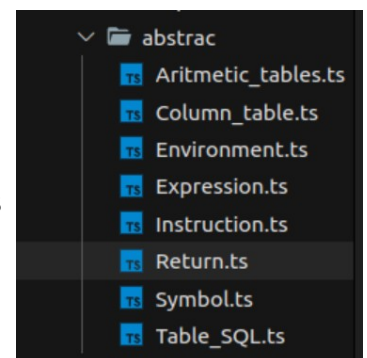


Para comenzar se creó un archivo json donde se realizó la gramática para nuestro proyecto, para mayor detalle puede leer el archivo de gramática que se subió junto con este proyecto.

Abstract:

Aquí se creo todas las clases abstractas de nuestro proyecto:

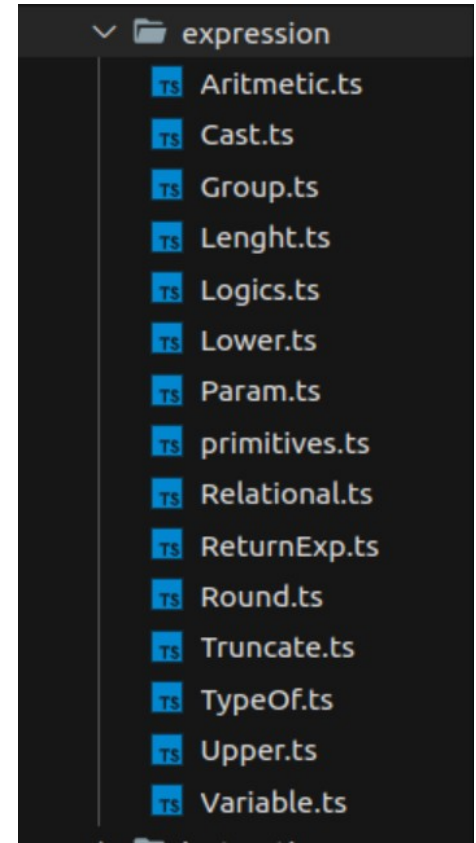
- **instrucción:** Este archivo fue creado con la finalidad de ser una clase padre para todas las ejecuciones de tipo Instuccion de nuestro codigo.
- **Enviroment:** Este maneja todos nuestro entornos creados para nuestro código
- **Expression:** Este fue creado con la finalidad de ser una clase padre para todas las ejecuciones de tipo Expression de nuestro código.
- **Return:** Aquí se defininio una enumeración para guardar el tipo de dato que se este ingresando y retornar el valor y tipo.



Expression:

Aquí se crearon todas las expresiones de nuestro código:

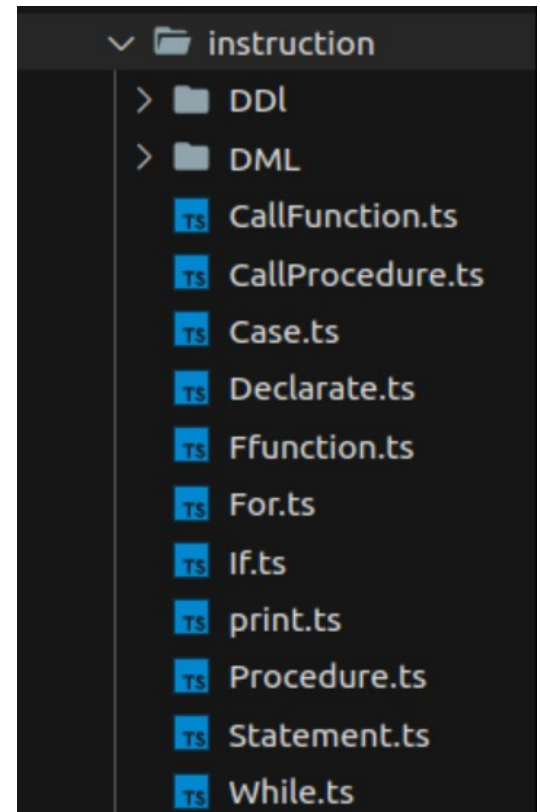
- **Variable:** Se encarga de acceder a las variables
- **Lenght:** Se encarga de realizar la función nativa length
- **Operación Aritmética:** Se encarga de realizar todas las operaciones aritméticas
- **Operación lógica:** Se encarga de realizar todas las operaciones lógicas.
- **Operación relacional:** Se encarga de realizar todas las operaciones relacionales
- **Parámetro:** se encarga de definir los parámetros de las funciones
- **Primitivo:** Se encarga de definir los valores primitivos como int, string, boolean, char, double.
- **Round:** Se encarga de realizar la función nativa rounda
- **Tolower:** Se encarga de realizar la función nativa tolower.
- **ToUpper:** Se encarga de realizar la función nativa Toupper
- **Truncate:** Se encarga de realizar la función nativa Truncate
- **Typeof:** Se encarga de realizar la función nativa typeof



INSTRUCCIONES

En esta carpeta están todos los archivos que realizan unas ejecuciones de tipo instrucción

- **CallFunction:** Se encarga de llamar a las funciones
- **Declaration:** declara las variables del lenguaje
- **For :** realiza el for del lenguaje
- **Función:** realiza las funciones las declara y guarda
- **If:** realiza la ejecución if del lenguaje
- **Print:** realiza la ejecución print del lenguaje
- **Statement:** Aquí se realizan las instrucciones que están dentro de las funciones, if, else, while
- **While** Aquí se realiza la ejecución while del lenguaje



Gramatica Utilizada:

<start> ::= <instrucciones> EOF

<instrucciones> ::= <instrucciones> <instruccion>
| <instruccion>

<instruccion> ::= <print>
| <declare>
| <set>
| <select>
| <ddl>
| <dml>
| <if>
| <case>
| <while>
| <for>
| <function>
| <call_function>
| <procedure>
| <call_procedure>
| RBREAK PUNTOCOMA
| RCONTINUE PUNTOCOMA
| RRETURN PUNTOCOMA
| RRETURN <expresion> PUNTOCOMA

<print> ::= RPRINT <expresion> PUNTOCOMA

<declare> ::= RDECLARE <variable_l> PUNTOCOMA

| RDECLARE VARIABLE_NAME <data_type> RDEFAULT <primitivo>
PUNTOCOMA

<variable_l> ::= <variable_l> COMMA VARIABLE_NAME <data_type>
| VARIABLE_NAME <data_type>

<set> ::= RSET VARIABLE_NAME EQUAL <expresion> PUNTOCOMA

<select> ::= RSELECT <expresion> PUNTOCOMA

<ddl> ::= <create>
| <alter>
| <drop>

<create> ::= RCREATE RTABLE ID PARENIZQ <column_list> PARENDER
PUNTOCOMA

<column_list> ::= <column_list> COMMA ID <data_type>
| ID <data_type>

<alter> ::= RALTER RTABLE ID RADD ID <data_type> PUNTOCOMA
| RALTER RTABLE ID RDROP RCOLUMN ID PUNTOCOMA
| RALTER RTABLE ID RRENAME RTO ID PUNTOCOMA
| RALTER RTABLE ID RRENAME RCOLUMN ID RTO ID PUNTOCOMA

<drop> ::= RDROP RTABLE ID PUNTOCOMA

<dml> ::= <dml_insert>
| <dml_select>
| <dml_update>

| <dml_truncate>

| <dml_delete>

<dml_insert> ::= RINSERT RINTO ID PARENIZQ <id_list> PARENDER
RVALÜES PARENIZQ <values_list> PARENDER PUNTOCOMA

<id_list> ::= <id_list> COMMA ID
| ID

<values_list> ::= <values_list> COMMA <primitivo>
| <primitivo>

<dml_select> ::= RSELECT <id_list> RFROM ID PUNTOCOMA
| RSELECT BY RFROM ID PUNTOCOMA
| RSELECT <id_list> RFROM ID RWHERE <where_conds>
PUNTOCOMA
| RSELECT BY RFROM ID RWHERE <where_conds> PUNTOCOMA

<dml_update> ::= RUPDATE ID RSET <set_columns> RWHERE
<where_conds> PUNTOCOMA

<set_columns> ::= ID EQUAL <primitivo>
| <set_columns> COMMA ID EQUAL <primitivo>

<dml_truncate> ::= RTRUNCATE RTABLE ID PUNTOCOMA

<dml_delete> ::= RDELETE RFROM ID RWHERE <where_conds>
PUNTOCOMA

<where_conds> ::= <where_conds> AND <where_conds>
| <where_conds> OR <where_conds>

| <where_conds> NOT <where_conds>
| <condition>

<condition> ::= ID EQUAL <arit_cond>
| ID DIFERENT <arit_cond>
| ID MINOR <arit_cond>
| ID MINOREQUAL <arit_cond>
| ID GREATER <arit_cond>
| ID GREATEREQUAL <arit_cond>

<arit_cond> ::= <arit_cond> PLUS <arit_cond>
| <arit_cond> LESS <arit_cond>
| <arit_cond> DIVIDED <arit_cond>
| <arit_cond> MODUL <arit_cond>
| LESS <arit_cond> %prec NEG
| <arit_cond> BY <arit_cond>
| <primitivo>

<if> ::= RIF <expresion> RTHEN <newStatement> REND RIF
PUNTOCOMA
| RIF <expresion> RTHEN <newStatement> RELSE <newStatement>
REND RIF PUNTOCOMA

<case> ::= RCASE <expresion> <case_list> RELSE <expresion> REND
PUNTOCOMA
| RCASE <expresion> <case_list> RELSE <expresion> REND <a_s>
PUNTOCOMA
| RCASE <case_list> RELSE <expresion> REND PUNTOCOMA
| RCASE <case_list> RELSE <expresion> REND <a_s> PUNTOCOMA

<a_s> ::= RAS <expresion>

<case_list> ::= <case_list> RWHEN <expresion> RTHEN <expresion>
| RWHEN <expresion> RTHEN <expresion>

<newStatement> ::= <instrucciones>

<while> ::= RWHILE <expresion> RBEGIN <newStatement> REND
PUNTOCOMA

<for> ::= RFOR VARIABLE_NAME RIN ENTERO RANGO ENTERO RBEGIN
<newStatement> REND PUNTOCOMA

<function> ::= RCREATE RFUNCTION ID PARENIZQ <params> PARENDER
RRETURN <data_type> RBEGIN <newStatement> REND PUNTOCOMA
| RCREATE RFUNCTION ID PARENIZQ <params> PARENDER
RRETURNS <data_type> RBEGIN <newStatement> REND PUNTOCOMA

<call_function> ::= RSELECT ID PARENIZQ <args> PARENDER PUNTOCOMA

<args> ::= <args> COMMA <expresion>
| <expresion>
| <empty>

<procedure> ::= RCREATE RPROCEDURE ID <params> RAS RBEGIN
<newStatement> REND PUNTOCOMA

<call_procedure> ::= RCALL ID PARENIZQ <args> PARENDER PUNTOCOMA

<params> ::= <params> COMMA VARIABLE_NAME <data_type>
| VARIABLE_NAME <data_type>
| <empty>

<expresion> ::= <primitivo>
 | <aritmetica>
 | <relacional>
 | <logica>
 | <cast>
 | <natives>
 | PARENIZQ <expresion> PARENDER
 | VARIABLE_NAME
 | ID
 | ID PARENIZQ <args> PARENDER

<aritmetica> ::= <expresion> PLUS <expresion>
 | <expresion> LESS <expresion>
 | <expresion> BY <expresion>
 | <expresion> DIVIDED <expresion>
 | <expresion> MODUL <expresion>
 | LESS <expresion> %prec NEG

<relacional> ::= <expresion> EQUAL <expresion>
 | <expresion> DIFERENT <expresion>
 | <expresion> MINOR <expresion>
 | <expresion> MINOREQUAL <expresion>
 | <expresion> GREATER <expresion>
 | <expresion> GREATEREQUAL <expresion>

<logica> ::= <expresion> AND <expresion>
 | <expresion> OR <expresion>
 | NOT <expresion>

<primitivo> ::= ENTERO

| DECIMAL

| DATE

| CADENA

| RTRUE

| RFALSE

| RNULL

<cast> ::= RCAST PARENIZQ <expresion> RAS <data_type> PARENDER

<natives> ::= RLOWER PARENIZQ <expresion> PARENDER

| RUPPER PARENIZQ <expresion> PARENDER

| RROUND PARENIZQ <expresion> COMMA ENTERO PARENDER

| RROUND PARENIZQ <expresion> PARENDER

| RLENGTH PARENIZQ <expresion> PARENDER

| RTRUNCATE PARENIZQ <expresion> COMMA ENTERO PARENDER

| RTRUNCATE PARENIZQ <expresion> PARENDER

| RTYPEOF PARENIZQ <expresion> PARENDER

<data_type> ::= RINT

| RDOUBLE

| RDATE

| RVARCHAR

| RBOOLEAN

| error