

Manual Técnico

Para la realización del proyecto se hizo uso de varios lenguajes de programación y herramientas para la generación de el parser. También se utilizó un Cliente y Servidor independiente para poder probar peticiones REST a un servidor.

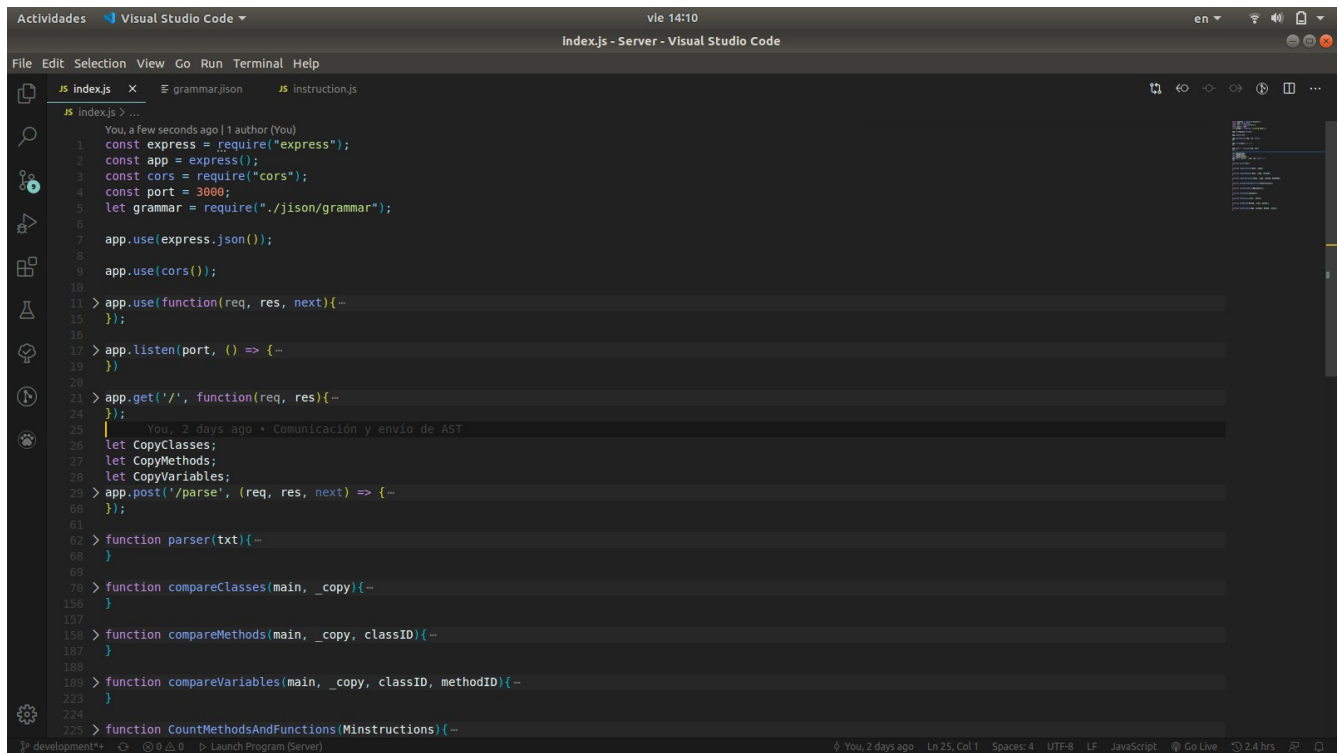
El Cliente:

El cliente usa un servidor Go para poder estar en un localhost. Este tiene parte de html, css y javascript. Las peticiones al servidor se realizan mediante javascript. Se muestran algunos pantallazos de como esta programado el cliente.

El Servidor:

El servidor cuenta con varios archivos que ayudan en varios aspectos. Esta la parte de index.js que es el archivo principal del servidor donde se encuentra la petición REST y algunos métodos para comparar AST para ver las copias. Esta la sección de la gramática que esta el archivo jison con el que se genero la gramática y el archivo js que generó jison. También en esta sección de gramática se encuentra un archivo “auxiliar” que se encarga de tener funciones que generan el AST. Se muestran algunos screenshots de como esta programado el servidor.

- index.js



```
1  const express = require("express");
2  const app = express();
3  const cors = require("cors");
4  const port = 3000;
5  let grammar = require("../jison/grammar");
6
7  app.use(express.json());
8
9  app.use(cors());
10
11 > app.use(function(req, res, next){--
15   });
16
17 > app.listen(port, () => {--
19   });
20
21 > app.get('/', function(req, res){--
24   });
25   You, 2 days ago + Comunicación y envío de AST
26   let CopyClasses;
27   let CopyMethods;
28   let CopyVariables;
29 > app.post('/parse', (req, res, next) => {--
60   });
61
62 > function parser(txt){--
68   }
69
70 > function compareClasses(main, _copy){--
156   }
157
158 > function compareMethods(main, _copy, classID){--
187   }
188
189 > function compareVariables(main, _copy, classID, methodID){--
223   }
224
225 > function CountMethodsAndFunctions(Minstructions){--
```

Actividades Visual Studio Code vie 14:10 en

index.js - Server - Visual Studio Code

File Edit Selection View Go Run Terminal Help

JS index.js X grammar.json JS instruction.js

JS index.js > ...

```
224
225 > function CountMethodsAndFunctions(Minstructions){--
226 }
227
228 > function CountParameters(Mparameters){--
229 }
230
231 > function IsFunOrMet(sentence){--
232 }
233
234 > function PushClass(_class, _count){--
235 }
236
237 > function PushMethod(method, _class, params){--
238 }
239
240 > function PushVariable(type, variable, method, _class){--
241 }
242
243
244
```

development++ 0 0 0 Launch Program (Server) You, 2 days ago Ln 25, Col 1 Spaces: 4 UTF-8 LF JavaScript Go Live 2.4 hrs

- grammar.json

Actividades Visual Studio Code vie 14:11 en

grammar.json - Server - Visual Studio Code

File Edit Selection View Go Run Terminal Help

JS index.js X grammar.json JS instruction.js

json > grammar.json

```
1 You, a few seconds ago | 1 author (You)
2
3 /* LEXICAL ANALYSIS */
4 %lex
5 %options case-sensitive
6
7 integer [0-9]+
8 double {integer}("."{integer})?
9 stringL (\("[^"]*"|'[^']*'|\\[^\n\r"]*\))
10 charL (\('\'|'\'|\\[^\n\r"]*\))
11 id ([a-zA-Z_][a-zA-Z0-9_]*)
12
13 %%
14 \s+ /* skip whitespace */
15 You, 16 days ago • análisis Léxico
16 /* COMMENTS */
17 /*- */ return
18 [/[ ]+([*]*)+([/*]*)+([*]*)+[/] return
19
20 /* NATIVE VALUES */
21 {double} return 'Double_Number'
22 {integer} return 'Integer_Number'
23 {stringL} return 'String_Literal'
24 {charL} return 'Char_Literal'
25 "true" return 'true'
26 "false" return 'false'
27
28 /* SYMBOLS */
29
30 /* ARITHMETIC SYMBOLS */
31 "++" return 'S_PLUSPLUS'
32 "--" return 'S_MINUSMINUS'
33 "^" return 'S_POTENCY'
34 "*" return 'S_MULTIPLY'
35 "/" return 'S_DIVISION'
36 "%" return 'S_MODULE'
37 "+" return 'S_PLUS'
38 "-" return 'S_MINUS'
```

development++ 0 0 0 Launch Program (Server) You, 16 days ago Ln 14, Col 1 Spaces: 4 UTF-8 LF json Go Live 2.4 hrs

Visual Studio Code - vie 14:11

grammar.json - Server - Visual Studio Code

```
File Edit Selection View Go Run Terminal Help

JS index.js grammar.json X JS instruction.js

jison > grammar.json

39 /* RELATIONAL SYMBOLS */
40 "<="      return 'S_MINOREQUALS'
41 ">="      return 'S_MAJOREQUALS'
42 "=="      return 'S_EQUALEQUALS'
43 "!="      return 'S_DIFFERENT'
44 "<"       return 'S_MINOR'
45 ">"       return 'S_MAJOR'
46
47 /* LOGICAL SYMBOLS */
48 "||"      return 'S_OR'
49 "&&"      return 'S_AND'
50 "!"       return 'S_NOT'
51
52 /* GENERAL SYMBOLS */
53 "=="      return 'S_EQUALS'
54 ";"       return 'S_SEMICOLON'
55 ":::"     return 'S_TWOPPOINTS'
56 "{"       return 'S_OPEN_KEY'
57 "}"       return 'S_CLOSE_KEY'
58 "("       return 'S_OPEN_PARENTHESIS'
59 ")"       return 'S_CLOSE_PARENTHESIS'
60 ","       return 'S_COMMA'
61
62 /* RESERVED WORDS */
63 /* TYPES */
64 "int"      return 'WR_INT'
65 "double"   return 'WR_DOUBLE'
66 "boolean"  return 'WR_BOOLEAN'
67 "char"     return 'WR_CHAR'
68 "String"   return 'WR_STRING'
69
70 /* GENERAL RESERVED WORDS */
71 "class"    return 'class'
72 "import"   return 'import'
73
74 /* IFELSE SENTENCE */
75 "if"       return 'if'
76 "else"     return 'else'
77
```

development++ ↻ 0 0 0 Launch Program (Server) You, 16 days ago Ln 14, Col 1 Spaces: 4 UTF-8 LF jison Go Live 2.4 hrs

Visual Studio Code - vie 14:11

grammar.json - Server - Visual Studio Code

```
File Edit Selection View Go Run Terminal Help

JS index.js grammar.json X JS instruction.js

jison > grammar.json

78 /* SWITCH SENTENCE */
79 "switch"   return 'switch'
80 "case"     return 'case'
81 "default"  return 'default'
82 "break"    return 'break'
83
84 /* WHILE SENTENCE */
85 "while"    return 'while'
86
87 /* DO WHILE SENTENCE */
88 "do"       return 'do'
89
90 /* FOR SENTENCE */
91 "for"      return 'for'
92
93 /* CONTINUE */
94 "continue" return 'continue'
95
96 /* RETURN */
97 "return"   return 'return'
98
99 /* MAIN */
100 "void"     return 'void'
101 "main"     return 'main'
102
103 /* PRINT SENTENCE */
104 "System"   return 'System'
105 "."        return 'S_POINT'
106 "out"      return 'out'
107 "print"    return 'print'
108 "println"  return 'println'
109
110 /* ID */
111 {id}       return 'id'
112 <<EOF>>    return 'EOF'
113           { ARRAYMANAGEMENT.pushAr(LexicalErrors, { ERROR: "Caracter invalido no perteneciente al lenguaje: " + yytext, LINE: yylloc.first_line, C
114
115 /lex
116
```

development++ ↻ 0 0 0 0 Launch Program (Server) You, 16 days ago Ln 14, Col 1 Spaces: 4 UTF-8 LF jison Go Live 2.4 hrs

Visual Studio Code interface showing the grammar.js file for grammarjson-server. The code defines the grammar rules and the parser logic. The current view shows the START rule and the associated error handling logic.

```
117  const OPERATION_TYPE      = require("../Instructions/instruction").OPERATIONS;
118  const TYPES               = require("../Instructions/instruction").TYPES;
119  const APIInstructions      = require("../Instructions/instruction").APIInstructions;
120
121  const ARRAYMANAGEMENT     = require("../Instructions/instruction").ARRAYMANAGEMENT;
122
123  ARRAYMANAGEMENT.DeleteArrays();
124
125  let LexicalErrors         = ARRAYMANAGEMENT.getLex();
126  let SyntacticalErrors     = ARRAYMANAGEMENT.getSyn();
127
128  %}
129
130  /* PRECEDENCE */
131  %left 'else'
132  %left 'S_OR'
133  %left 'S_AND'
134  %left 'S_EQUALSEQUALS', 'S_DIFFERENT'
135  %left 'S_MAJOREQUALS', 'S_MINOREQUALS', 'S_MAJOR', 'S_MINOR'
136  %left 'S_PLUS', 'S_MINUS'
137  %left 'S_MULTIPLY', 'S_DIVISION', 'S_MODULE'
138  %left 'S_POTENCY'
139  %right 'S_NOT'
140  %left UMINUS
141
142  /* SYNTACTIC ANALYSIS */
143  %start START
144
145  %%
146
147  START : IMPORTLIST CLASSLIST EOF      { return { AST: APIInstructions.root($1, $2), LEXICAL_ERRORS: LexicalErrors, SINTACTICAL_ERRORS: SyntacticalErrors}; }
148      | IMPORTLIST EOF                  { return { AST: APIInstructions.root($1, undefined), LEXICAL_ERRORS: LexicalErrors, SINTACTICAL_ERRORS: SyntacticalError
149      | CLASSLIST EOF                    { return { AST: APIInstructions.root(undefined, $1), LEXICAL_ERRORS: LexicalErrors, SINTACTICAL_ERRORS: SyntacticalError
150      | ERRORSENTENCE SEMICOLON START {
151          ARRAYMANAGEMENT.pushAr( SyntacticalErrors,
152          {
153              ERROR: "Se esperaba import o clase, se encontro "" + $1 + """,
154              LINE: this._first_line
155          }
156      );
157      }
158      ;
```

Visual Studio Code interface showing the grammar.js file for grammarjson-server. The code defines the grammar rules and the parser logic. The current view shows the START rule and the associated error handling logic.

```
148  START : IMPORTLIST CLASSLIST EOF      { return { AST: APIInstructions.root($1, $2), LEXICAL_ERRORS: LexicalErrors, SINTACTICAL_ERRORS: SyntacticalErrors}; }
149      | IMPORTLIST EOF                  { return { AST: APIInstructions.root($1, undefined), LEXICAL_ERRORS: LexicalErrors, SINTACTICAL_ERRORS: SyntacticalError
150      | CLASSLIST EOF                    { return { AST: APIInstructions.root(undefined, $1), LEXICAL_ERRORS: LexicalErrors, SINTACTICAL_ERRORS: SyntacticalError
151      | ERRORSENTENCE SEMICOLON START {
152          ARRAYMANAGEMENT.pushAr( SyntacticalErrors,
153          {
154              ERROR: "Se esperaba import o clase, se encontro "" + $1 + """,
155              LINE: this._first_line,
156              COLUMN: this._first_column
157          }
158      );
159      }
160      | ERRORSENTENCE S_OPEN_KEY ERRORSENTENCE S_CLOSE_KEY START {
161          ARRAYMANAGEMENT.pushAr( SyntacticalErrors,
162          {
163              ERROR: "Se esperaba import o clase, se encontro "" + $1 + """,
164              LINE: this._first_line,
165              COLUMN: this._first_column
166          }
167      );
168      }
169      | EOF                                { return { AST: 'Vacio', LEXICAL_ERRORS: undefined, SINTACTICAL_ERRORS: undefined }; }
170      ;
171
172  IMPORTLIST : FINALIMPORT IMPORTLIST { $$ = APIInstructions.newImportList($2, $1); }
173              | FINALIMPORT           { $$ = APIInstructions.newImportList(undefined, $1); }
174              ;
175
176  FINALIMPORT: import id SEMICOLON { $$ = APIInstructions.newImport($2); }
177              | import ERRORSENTENCE SEMICOLON
178              {
179                  ARRAYMANAGEMENT.pushAr( SyntacticalErrors,
180                  {
181                      ERROR: "Se esperaba ID1, se encontro "" + $2 + """,
182                      LINE: this._first_line,
183                      COLUMN: this._first_column
184                  }
185              );
186              $$ = undefined;
187          }
```


Visual Studio Code - vie 14:11

grammar.json - Server - Visual Studio Code

```
187
188
189
190 CLASSLIST : FINALCLASS CLASSLIST { $$ = APIInstructions.newClassList($2, $1); }
191 | FINALCLASS { $$ = APIInstructions.newClassList(undefine, $1); }
192
193
194 FINALCLASS : class id $ OPEN KEY INSIDECLASS $ CLOSE KEY { $$ = APIInstructions.newClass($2, $4); }
195 | class id $ OPEN KEY $ CLOSE KEY { $$ = APIInstructions.newClass($2, undefine); }
196 | class ERRORSSENTENCE $ OPEN KEY INSIDECLASS $ CLOSE KEY {
197   ARRAYMANAGEMENT.pushAr( SyntacticalErrors,
198     {
199       ERROR: "Se esperaba ID, se encontro '" + $2 + "'",
200       LINE: this._$.first_line,
201       COLUMN: this._$.first_column
202     }
203   );
204   $$ = undefine;
205 }
206 | class $ OPEN KEY INSIDECLASS $ CLOSE KEY {
207   ARRAYMANAGEMENT.pushAr( SyntacticalErrors,
208     {
209       ERROR: "Se esperaba ID, se encontro nada",
210       LINE: this._$.first_line,
211       COLUMN: this._$.first_column
212     }
213   );
214   $$ = undefine;
215 }
216
217
218 INSIDECLASS : FINALINSIDECLASS INSIDECLASS { $$ = APIInstructions.newListInsideClass($2, $1); }
219 | FINALINSIDECLASS { $$ = APIInstructions.newListInsideClass(undefine, $1); }
220
221 FINALINSIDECLASS : DECLARATIONSENTENCE { $$ = $1; }
222 | METHODSFUN { $$ = $1; }
223 | ERRORSSENTENCE SEMICOLON {
224   ARRAYMANAGEMENT.pushAr( SyntacticalErrors,
225     {
```

Visual Studio Code - vie 14:11

grammar.json - Server - Visual Studio Code

```
219 | FINALINSIDECLASS { $$ = APIInstructions.newListInsideClass(undefine, $1); }
220
221 FINALINSIDECLASS : DECLARATIONSENTENCE { $$ = $1; }
222 | METHODSFUN { $$ = $1; }
223 | ERRORSSENTENCE SEMICOLON {
224   ARRAYMANAGEMENT.pushAr( SyntacticalErrors,
225     {
226       ERROR: "Se esperaba declaracion o metodo, se encontro '" + $1 + "'",
227       LINE: this._$.first_line,
228       COLUMN: this._$.first_column
229     }
230   );
231   $$ = undefine;
232 }
233 | ERRORSSENTENCE $ OPEN KEY ERRORSSENTENCE $ CLOSE KEY {
234   ARRAYMANAGEMENT.pushAr( SyntacticalErrors,
235     {
236       ERROR: "Se esperaba declaracion o metodo, se encontro '" + $1 + "'",
237       LINE: this._$.first_line,
238       COLUMN: this._$.first_column
239     }
240   );
241   $$ = undefine;
242 }
243
244
245 METHODSFUN : TYPE id $ OPEN PARENTHESIS PARAMETERDECLARATION $ CLOSE PARENTHESIS $ OPEN KEY SENTENCESLIST $ CLOSE KEY { $$ = APIInstructions.newFunction($1, $2
246 | TYPE id $ OPEN PARENTHESIS $ CLOSE PARENTHESIS $ OPEN KEY SENTENCESLIST $ CLOSE KEY { $$ = APIInstructions.newFunction($1, $2, undefine, $6) }
247 | void id $ OPEN PARENTHESIS PARAMETERDECLARATION $ CLOSE PARENTHESIS $ OPEN KEY SENTENCESLIST $ CLOSE KEY { $$ = APIInstructions.newMethod($2, $4,
248 | void id $ OPEN PARENTHESIS $ CLOSE PARENTHESIS $ OPEN KEY SENTENCESLIST $ CLOSE KEY { $$ = APIInstructions.newMethod($2, undefine, $6); }
249 | void main $ OPEN PARENTHESIS $ CLOSE PARENTHESIS $ OPEN KEY SENTENCESLIST $ CLOSE KEY { $$ = APIInstructions.newMethod($2, undefine, $6); }
250
251
252 TYPE : WR_INT { $$ = $1; } | WR_DOUBLE { $$ = $1; } | WR_CHAR { $$ = $1; } | WR_STRING { $$ = $1; } | WR_BOOLEAN { $$ = $1; }
253
254
255 PARAMETERDECLARATION : PARAMETER $ COMMA PARAMETERDECLARATION { $$ = APIInstructions.newParamsList($3, $1); }
256 | PARAMETER { $$ = APIInstructions.newParamsList(undefine, $1); }
257 | ERRORSSENTENCE $ COMMA PARAMETERDECLARATION {
```

Visual Studio Code - vie 14:11

grammar.json - Server - Visual Studio Code

```
File Edit Selection View Go Run Terminal Help

js index.js grammar.json instruction.js

json > grammar.json

253 ;
254
255 PARAMETERDECLARATION : PARAMETER S COMMA PARAMETERDECLARATION { $$ = APIInstructions.newParamsList($3, $1); }
256 | PARAMETER { $$ = APIInstructions.newParamsList(undefined, $1); }
257 | ERRORSSENTENCE S COMMA PARAMETERDECLARATION {
258 |   ARRAYMANAGEMENT.pushAr( SyntacticalErrors,
259 |   {
260 |     ERROR: "Se esperaba parametro, se encontro: '" + $1 + "'",
261 |     LINE: this._$.first_line,
262 |     COLUMN: this._$.first_column
263 |   }
264 | );
265 | $$ = undefined;
266 | }
267 | ERRORSSENTENCE {
268 |   ARRAYMANAGEMENT.pushAr( SyntacticalErrors,
269 |   {
270 |     ERROR: "Se esperaba parametros, se encontro: '" + $1 + "'",
271 |     LINE: this._$.first_line,
272 |     COLUMN: this._$.first_column
273 |   }
274 | );
275 | $$ = undefined;
276 | }
277 ;
278
279 PARAMETER : TYPE id { $$ = APIInstructions.newParam($1, $2); }
280 ;
281
282 SENTENCESLIST : DECLARATIONSENTENCE SENTENCESLIST { $$ = APIInstructions.newSentenceList($2, $1); }
283 | ASSIGNMENTORCALLSENTENCE SENTENCESLIST { $$ = APIInstructions.newSentenceList($2, $1); }
284 | PRINTSENTENCE SENTENCESLIST { $$ = APIInstructions.newSentenceList($2, $1); }
285 | IFELSESENTENCE SENTENCESLIST { $$ = APIInstructions.newSentenceList($2, $1); }
286 | SWITCHSENTENCE SENTENCESLIST { $$ = APIInstructions.newSentenceList($2, $1); }
287 | FORSENTENCE SENTENCESLIST { $$ = APIInstructions.newSentenceList($2, $1); }
288 | WHILESENTENCE SENTENCESLIST { $$ = APIInstructions.newSentenceList($2, $1); }
289 | DOWHILESENTENCE SENTENCESLIST { $$ = APIInstructions.newSentenceList($2, $1); }
290 | CONTINUESENTENCE SENTENCESLIST { $$ = APIInstructions.newSentenceList($2, $1); }
291 | BREAKSENTENCE SENTENCESLIST { $$ = APIInstructions.newSentenceList($2, $1); }
```

Visual Studio Code - vie 14:11

grammar.json - Server - Visual Studio Code

```
File Edit Selection View Go Run Terminal Help

js index.js grammar.json instruction.js

json > grammar.json

279 PARAMETER : TYPE id { $$ = APIInstructions.newParam($1, $2); }
280 ;
281
282 SENTENCESLIST : DECLARATIONSENTENCE SENTENCESLIST { $$ = APIInstructions.newSentenceList($2, $1); }
283 | ASSIGNMENTORCALLSENTENCE SENTENCESLIST { $$ = APIInstructions.newSentenceList($2, $1); }
284 | PRINTSENTENCE SENTENCESLIST { $$ = APIInstructions.newSentenceList($2, $1); }
285 | IFELSESENTENCE SENTENCESLIST { $$ = APIInstructions.newSentenceList($2, $1); }
286 | SWITCHSENTENCE SENTENCESLIST { $$ = APIInstructions.newSentenceList($2, $1); }
287 | FORSENTENCE SENTENCESLIST { $$ = APIInstructions.newSentenceList($2, $1); }
288 | WHILESENTENCE SENTENCESLIST { $$ = APIInstructions.newSentenceList($2, $1); }
289 | DOWHILESENTENCE SENTENCESLIST { $$ = APIInstructions.newSentenceList($2, $1); }
290 | CONTINUESENTENCE SENTENCESLIST { $$ = APIInstructions.newSentenceList($2, $1); }
291 | BREAKSENTENCE SENTENCESLIST { $$ = APIInstructions.newSentenceList($2, $1); }
292 | RETURNSENTENCE SENTENCESLIST { $$ = APIInstructions.newSentenceList($2, $1); }
293 | { $$ = undefined; }
294 | ERRORSSENTENCE SEMICOLON SENTENCESLIST {
295 |   ARRAYMANAGEMENT.pushAr( SyntacticalErrors,
296 |   {
297 |     ERROR: "Se esperaba sentencia, se encontro '" + $1 + "'",
298 |     LINE: this._$.first_line,
299 |     COLUMN: this._$.first_column
300 |   }
301 | );
302 | $$ = undefined;
303 | }
304 | ERRORSSENTENCE S_OPEN_KEY SENTENCESLIST S_CLOSE_KEY SENTENCESLIST {
305 |   ARRAYMANAGEMENT.pushAr( SyntacticalErrors,
306 |   {
307 |     ERROR: "Se esperaba sentencia, se encontro '" + $1 + "'",
308 |     LINE: this._$.first_line,
309 |     COLUMN: this._$.first_column
310 |   }
311 | );
312 | }
313 ;
314
315 CONTINUESENTENCE : continue SEMICOLON { $$ = APIInstructions.newContinue(); }
316 ;
317 BREAKSENTENCE : break SEMICOLON { $$ = APIInstructions.newBreak(); }
```

```
Actividades Visual Studio Code vie 14:11 en grammarjison - Server - Visual Studio Code

File Edit Selection View Go Run Terminal Help

jison > grammarjison
311
312
313
314
315 CONTINUESENTENCE : continue SEMICOLON { $$ = APIInstructions.newContinue(); }
316
317
318
319 RETURNSENTENCE : return EXPRESSION SEMICOLON { $$ = APIInstructions.newReturn($2); }
320 | return SEMICOLON { $$ = APIInstructions.newReturn(undefined); }
321
322
323 DECLARATIONSENTENCE : TYPE IDLIST SEMICOLON { $$ = APIInstructions.newDeclaration($1, $2); }
324 | TYPE IDLIST S_EQUALS EXPRESSION SEMICOLON { $$ = APIInstructions.newDecAs($1, $2, $4); }
325
326
327 IDLIST : IDLIST S_COMMA id { $$ = APIInstructions.newListIDs($1, $3); }
328 | id { $$ = APIInstructions.newListIDs(undefined, $1); }
329
330
331 ASSIGNMENTORCALLSENTENCE : id S_EQUALS EXPRESSION SEMICOLON { $$ = APIInstructions.newAssignment($1, $3); }
332 | id S_OPEN_PARENTHESIS S_CLOSE_PARENTHESIS SEMICOLON { $$ = APIInstructions.newCallFun($1, undefined); }
333 | id S_OPEN_PARENTHESIS EXPRESSIONLIST S_CLOSE_PARENTHESIS SEMICOLON { $$ = APIInstructions.newCallFun($1, $3); }
334
335
336 /* PRINT STATEMENTS */
337 PRINTSENTENCE : System S_POINT out S_POINT PRINTOPT SEMICOLON { $$ = $5; }
338
339
340 PRINTOPT : print S_OPEN_PARENTHESIS EXPRESSION S_CLOSE_PARENTHESIS { $$ = APIInstructions.newPrint("System.out.print", $3); }
341 | print S_OPEN_PARENTHESIS S_CLOSE_PARENTHESIS { $$ = APIInstructions.newPrint("System.out.print", undefined); }
342 | println S_OPEN_PARENTHESIS EXPRESSION S_CLOSE_PARENTHESIS { $$ = APIInstructions.newPrint("System.out.println", $3); }
343 | println S_OPEN_PARENTHESIS S_CLOSE_PARENTHESIS { $$ = APIInstructions.newPrint("System.out.println", undefined); }
344
345
346 /* CONDITIONAL STATEMENTS */
347 IFELSESENTENCE : if S_OPEN_PARENTHESIS EXPRESSION S_CLOSE_PARENTHESIS S_OPEN_KEY SENTENCESLIST S_CLOSE_KEY { $$ = APIInstructions.newIf($3, $6, undefined, unde
348 | if S_OPEN_PARENTHESIS EXPRESSION S_CLOSE_PARENTHESIS S_OPEN_KEY SENTENCESLIST S_CLOSE_KEY else S_OPEN_KEY SENTENCESLIST S_CLOSE_KEY { $$ = A
349 | if S_OPEN_PARENTHESIS EXPRESSION S_CLOSE_PARENTHESIS S_OPEN_KEY SENTENCESLIST S_CLOSE_KEY else IFELSESENTENCE { $$ = APIInstructions.newIf($3,
350
351
352 SWITCHSENTENCE : switch S_OPEN_PARENTHESIS EXPRESSION S_CLOSE_PARENTHESIS S_OPEN_KEY CASELIST OPTDEFAULT S_CLOSE_KEY { $$ = APIInstructions.newSwitch($3, $6,
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

```
Actividades Visual Studio Code vie 14:11 en grammarjison - Server - Visual Studio Code

File Edit Selection View Go Run Terminal Help

jison > grammarjison
311
312
313
314
315 CONTINUESENTENCE : continue SEMICOLON { $$ = APIInstructions.newContinue(); }
316
317
318
319 RETURNSENTENCE : return EXPRESSION SEMICOLON { $$ = APIInstructions.newReturn($2); }
320 | return SEMICOLON { $$ = APIInstructions.newReturn(undefined); }
321
322
323 DECLARATIONSENTENCE : TYPE IDLIST SEMICOLON { $$ = APIInstructions.newDeclaration($1, $2); }
324 | TYPE IDLIST S_EQUALS EXPRESSION SEMICOLON { $$ = APIInstructions.newDecAs($1, $2, $4); }
325
326
327 IDLIST : IDLIST S_COMMA id { $$ = APIInstructions.newListIDs($1, $3); }
328 | id { $$ = APIInstructions.newListIDs(undefined, $1); }
329
330
331 ASSIGNMENTORCALLSENTENCE : id S_EQUALS EXPRESSION SEMICOLON { $$ = APIInstructions.newAssignment($1, $3); }
332 | id S_OPEN_PARENTHESIS S_CLOSE_PARENTHESIS SEMICOLON { $$ = APIInstructions.newCallFun($1, undefined); }
333 | id S_OPEN_PARENTHESIS EXPRESSIONLIST S_CLOSE_PARENTHESIS SEMICOLON { $$ = APIInstructions.newCallFun($1, $3); }
334
335
336 /* PRINT STATEMENTS */
337 PRINTSENTENCE : System S_POINT out S_POINT PRINTOPT SEMICOLON { $$ = $5; }
338
339
340 PRINTOPT : print S_OPEN_PARENTHESIS EXPRESSION S_CLOSE_PARENTHESIS { $$ = APIInstructions.newPrint("System.out.print", $3); }
341 | print S_OPEN_PARENTHESIS S_CLOSE_PARENTHESIS { $$ = APIInstructions.newPrint("System.out.print", undefined); }
342 | println S_OPEN_PARENTHESIS EXPRESSION S_CLOSE_PARENTHESIS { $$ = APIInstructions.newPrint("System.out.println", $3); }
343 | println S_OPEN_PARENTHESIS S_CLOSE_PARENTHESIS { $$ = APIInstructions.newPrint("System.out.println", undefined); }
344
345
346 /* CONDITIONAL STATEMENTS */
347 IFELSESENTENCE : if S_OPEN_PARENTHESIS EXPRESSION S_CLOSE_PARENTHESIS S_OPEN_KEY SENTENCESLIST S_CLOSE_KEY { $$ = APIInstructions.newIf($3, $6, undefined, unde
348 | if S_OPEN_PARENTHESIS EXPRESSION S_CLOSE_PARENTHESIS S_OPEN_KEY SENTENCESLIST S_CLOSE_KEY else S_OPEN_KEY SENTENCESLIST S_CLOSE_KEY { $$ = A
349 | if S_OPEN_PARENTHESIS EXPRESSION S_CLOSE_PARENTHESIS S_OPEN_KEY SENTENCESLIST S_CLOSE_KEY else IFELSESENTENCE { $$ = APIInstructions.newIf($3,
350
351
352 SWITCHSENTENCE : switch S_OPEN_PARENTHESIS EXPRESSION S_CLOSE_PARENTHESIS S_OPEN_KEY CASELIST OPTDEFAULT S_CLOSE_KEY { $$ = APIInstructions.newSwitch($3, $6,
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```


Visual Studio Code - vie 14:11

grammar.json - Server - Visual Studio Code

```
File Edit Selection View Go Run Terminal Help

js index.js grammar.json X js instruction.js

json > grammar.json
374 WHILESENTENCE : while S_OPEN_PARENTHESIS EXPRESSION S_CLOSE_PARENTHESIS S_OPEN_KEY SENTENCESLIST S_CLOSE_KEY { $$ = APIInstructions.newWhile($3, $6); }
375 ;
376
377 DOWHILESENTENCE : do S_OPEN_KEY SENTENCESLIST S_CLOSE_KEY while S_OPEN_PARENTHESIS EXPRESSION S_CLOSE_PARENTHESIS SEMICOLON { $$ = APIInstructions.newDowhile($7
378 ;
379
380 EXPRESSION : S_MINUS EXPRESSION %prec UMINUS { $$ = APIInstructions.newUnaryOP($2, OPERATION_TYPE.NEGATE); }
381 | S_NOT EXPRESSION { $$ = APIInstructions.newUnaryOP($2, OPERATION_TYPE.NOT); }
382 | EXPRESSION S_PLUS EXPRESSION { $$ = APIInstructions.newBinaryOP($1, $3, OPERATION_TYPE.ADDITION); }
383 | EXPRESSION S_MINUS EXPRESSION { $$ = APIInstructions.newBinaryOP($1, $3, OPERATION_TYPE.SUBTRACTION); }
384 | EXPRESSION S_MULTIPLY EXPRESSION { $$ = APIInstructions.newBinaryOP($1, $3, OPERATION_TYPE.MULTIPLICATION); }
385 | EXPRESSION S_DIVISION EXPRESSION { $$ = APIInstructions.newBinaryOP($1, $3, OPERATION_TYPE.DIVISION); }
386 | EXPRESSION S_MODULE EXPRESSION { $$ = APIInstructions.newBinaryOP($1, $3, OPERATION_TYPE.MODULO); }
387 | EXPRESSION S_POTENCY EXPRESSION { $$ = APIInstructions.newBinaryOP($1, $3, OPERATION_TYPE.POTENCY); }
388 | EXPRESSION S_MAJOR EXPRESSION { $$ = APIInstructions.newBinaryOP($1, $3, OPERATION_TYPE.MAJOR_THAN); }
389 | EXPRESSION S_MINOR EXPRESSION { $$ = APIInstructions.newBinaryOP($1, $3, OPERATION_TYPE.LESS_THAN); }
390 | EXPRESSION S_MAJOREQUALS EXPRESSION { $$ = APIInstructions.newBinaryOP($1, $3, OPERATION_TYPE.MAJOR_EQUALS_THAN); }
391 | EXPRESSION S_MINOREQUALS EXPRESSION { $$ = APIInstructions.newBinaryOP($1, $3, OPERATION_TYPE.LESS_EQUALS_THAN); }
392 | EXPRESSION S_EQUALSEQUALS EXPRESSION { $$ = APIInstructions.newBinaryOP($1, $3, OPERATION_TYPE.EQUALS_EQUALS); }
393 | EXPRESSION S_DIFFERENT EXPRESSION { $$ = APIInstructions.newBinaryOP($1, $3, OPERATION_TYPE.DIFFERENT); }
394 | EXPRESSION S_OR EXPRESSION { $$ = APIInstructions.newBinaryOP($1, $3, OPERATION_TYPE.OR); }
395 | EXPRESSION S_AND EXPRESSION { $$ = APIInstructions.newBinaryOP($1, $3, OPERATION_TYPE.AND); }
396 | S_OPEN_PARENTHESIS EXPRESSION S_CLOSE_PARENTHESIS { $$ = $2; }
397 | id S_OPEN_PARENTHESIS EXPRESSIONLIST S_CLOSE_PARENTHESIS { $$ = APIInstructions.newCallFun($1, $3); }
398 | id S_OPEN_PARENTHESIS S_CLOSE_PARENTHESIS { $$ = APIInstructions.newCallFun($1, undefined); }
399 | id { $$ = APIInstructions.newValue($1, TYPES.ID); }
400 | Double_Number { $$ = APIInstructions.newValue($1, TYPES.DOUBLE); }
401 | Integer_Number { $$ = APIInstructions.newValue($1, TYPES.INTEGER); }
402 | String_Literal { $$ = APIInstructions.newValue($1, TYPES.STRING); }
403 | Char_Literal { $$ = APIInstructions.newValue($1, TYPES.CHAR); }
404 | true { $$ = APIInstructions.newValue($1, TYPES.BOOLEAN); }
405 | false { $$ = APIInstructions.newValue($1, TYPES.BOOLEAN); }
406 ;
407
408 EXPRESSIONLIST : EXPRESSION S_COMMA EXPRESSIONLIST { $$ = APIInstructions.newListExpression($3, $1); }
409 | EXPRESSION { $$ = APIInstructions.newListExpression(undefined, $1); }
410 ;
411
412 ERRORSENTENCE : ERRORSENTENCE FINALERROR { $$ = $1 + " " + $2; }
```

Visual Studio Code - vie 14:11

grammar.json - Server - Visual Studio Code

```
File Edit Selection View Go Run Terminal Help

js index.js grammar.json X js instruction.js

json > grammar.json
395 | EXPRESSION S_AND EXPRESSION { $$ = APIInstructions.newBinaryOP($1, $3, OPERATION_TYPE.AND); }
396 | S_OPEN_PARENTHESIS EXPRESSION S_CLOSE_PARENTHESIS { $$ = $2; }
397 | id S_OPEN_PARENTHESIS EXPRESSIONLIST S_CLOSE_PARENTHESIS { $$ = APIInstructions.newCallFun($1, $3); }
398 | id S_OPEN_PARENTHESIS S_CLOSE_PARENTHESIS { $$ = APIInstructions.newCallFun($1, undefined); }
399 | id { $$ = APIInstructions.newValue($1, TYPES.ID); }
400 | Double_Number { $$ = APIInstructions.newValue($1, TYPES.DOUBLE); }
401 | Integer_Number { $$ = APIInstructions.newValue($1, TYPES.INTEGER); }
402 | String_Literal { $$ = APIInstructions.newValue($1, TYPES.STRING); }
403 | Char_Literal { $$ = APIInstructions.newValue($1, TYPES.CHAR); }
404 | true { $$ = APIInstructions.newValue($1, TYPES.BOOLEAN); }
405 | false { $$ = APIInstructions.newValue($1, TYPES.BOOLEAN); }
406 ;
407
408 EXPRESSIONLIST : EXPRESSION S_COMMA EXPRESSIONLIST { $$ = APIInstructions.newListExpression($3, $1); }
409 | EXPRESSION { $$ = APIInstructions.newListExpression(undefined, $1); }
410 ;
411
412 ERRORSENTENCE : ERRORSENTENCE FINALERROR { $$ = $1 + " " + $2; }
413 | FINALERROR { $$ = $1; }
414 ;
415 FINALERROR : error
416 ;
```


Actividades Visual Studio Code vie 14:11 en

instruction.js - Server - Visual Studio Code

File Edit Selection View Go Run Terminal Help

JS index.js JS grammar.json JS instruction.js X

JSON > Instructions > JS instruction.js > ...
You, an hour ago | 1 author (You)

1 > const TYPES = {--
8 }
9
10 > const OPERATIONS = {--
37 }
38
39 > const INSTRUCTIONS = {--
59 }
60
61 > const OPT_SWITCH = {--
64 }
65
66 > function newOp(left, right, type){--
72 }
73
74 > const ARRAYMANAGEMENT = {--
90 }
91
92 > const APIInstructions = {--
420 }
421
422 module.exports.OPERATIONS = OPERATIONS;
423 module.exports.TYPES = TYPES;
424 module.exports.APIInstructions = APIInstructions;
425 module.exports.ARRAYMANAGEMENT = ARRAYMANAGEMENT; You, an hour ago • Uncommitted changes

development** You, an hour ago Ln 425, Col 50 Spaces: 4 UTF-8 LF JavaScript Go Live 2.4 hrs