*Program:*

**empty**

*Element* *Program*

*Element:*

**function** *Identifier* **(** *ParameterListOpt* **)** *CompoundStatement*

*Statement*

*ParameterListOpt:*

**empty**

*ParameterList*

*ParameterList:*

*Identifier ParameterList’*

*ParameterList’:*

**,** *Identifier* *ParameterList´*

**empty**

*CompoundStatement*:

**{** *StatementList* **}**

*StatementList*:

**empty**

*Statement* *StatementList*

*Statement*:

**;**

**If** **(***AssignmentExpression***)** *Statement’* *IfNot*

**while** **(***AssignmentExpression***)***Statement’*

**for** **(***ForConditions***)** *Statement’*

**break** **;**

**continue** **;**

*ThrowStatement* **; /\* posiblemente no necesario(Produccion)\*/**

**switch (***AssignmentExpression***) {***CaseBlock***}**

**do** *Statement’* **while (***AssignmentExpression***)**

**return** *ExpressionOpt* **;**

**try** *CompoundStatement CatchBlock**FinallyBlock*

*VariablesOrExpression* **;**

*Statement’*:

*Statement*

*CompoundStatement*

*IfNot:*

**else** *Statement’*

**empty**

*ForConditions:*

**var** *VariablesList* **;** *ExpressionOpt* **;** *ExpressionOpt*

*ExpressionOpt* **;** *ExpressionOpt* **;** *ExpressionOpt*

*CaseBlock:*

*CaseClauseList DefaultClause*

*DefaultClause:*

**empty**

**default:** *Statement’ CaseClauseList*

*CaseClauseList:*

*CaseClause CaseClauseList*

**empty**

*CaseClause:*

**case** *AssignmentExpression***:** *StatementList*

*CatchBlock:*

**empty**

**catch (***identifier***)** *CompoundStatement*

*FinallyBlock:*

**finally** *CompoundStatement*

**empty**

*ThrowStatement:*

**throw** *ThrowStatement´*

**empty /\* si es vacio el statement aceptaria vacio tambien\*/**

*ThrowStatement’:*

*Identifier*

**new** *Constructor*

**empty**

*VariablesOrExpression*:

**var** *VariablesList*

*Expression*

*VarOpt:*

**var**

**empty**

*VariablesList*:

*Variable VariablesList’*

*VariblesList’:*

**,** *Variable VariablesList´*

**empty**

*Variable*:

**Identifier** *Variable’*

*Variable’:*

**=** *AssignmentExpression*

**empty**

*ExpressionOpt*:

*Expression*

**empty**

*Expression*:

*AssignmentExpression Expression’*

*Expression’:*

**empty**

**,** *AssignmentExpression* *Expression´*

*AssignmentExpression*:

*ConditionalExpression* *AssignmentExpression*’

*AssignmentExpression’*:

*AssignmentOperator* *ConditionalExpression* AssignmentExpression´

**empty**

*ConditionalExpression*:

*OrExpression ConditionalExpression´*

*ConditionalExpression´:*

? *AssignmentExpression* : *AssignmentExpression*

**empty**

*OrExpression*:

*AndExpression* *OrExpression*´

*OrExpression*´:

|| *AndExpression* *OrExpression*´

**empty**

*AndExpression*:

*BitwiseOrExpression* *AndExpression*´

*AndExpression*´:

&& *BitwiseOrExpression* *AndExpression*´

**empty**

*BitwiseOrExpression*:

*BitwiseXorExpression* *BitwiseOrExpression*´

*BitwiseOrExpression´:*

| *BitwiseXorExpression* *BitwiseOrExpression*´

**empty**

*BitwiseXorExpression*:

*BitwiseAndExpression* *BitwiseXorExpression*´

*BitwiseXorExpression*´:

^ *BitwiseAndExpression* *BitwiseXorExpression*´

**empty**

*BitwiseAndExpression*:

*RelationalExpression BitwiseAndExpression*´

*BitwiseAndExpression*´:

& *RelationalExpression BitwiseAndExpression*´

**empty**

*RelationalExpression*:

*ShiftExpression* *RelationalExpression*´

*RelationalExpression*´:

*RelationalOperator ShiftExpression* *RelationalExpression*´

**empty**

*ShiftExpression*:

*AdditiveExpression* *ShiftExpression*´

*ShiftExpression*´:

*ShiftOperator* *AdditiveExpression* *ShiftExpression*´

**empty**

*AdditiveExpression*:

*MultiplicativeExpression* *AdditiveExpression*´

*AdditiveExpression*´:

*AdditiveOperator* *MultiplicativeExpression* *AdditiveExpression*´

**empty**

*MultiplicativeExpression*:

*UnaryExpression* *MultiplicativeExpression*´

*MultiplicativeExpression*´:

*MultiplicativeOperator* *UnaryExpression* *MultiplicativeExpression*´

**empty**

*UnaryExpression*:

*MemberExpression UnaryExpression’*

*PrimaryExpression*

*UnaryOperator* *UnaryExpression*

*IncrementOperator* *MemberExpression*

*DecrementOperator MemberExpression*

**new** *Constructor*

*UnaryExpression’:*

**empty**

*IncrementOperator*

*DecrementOperator*

*Constructor*: /\*me parece no necesario\*/

*ConstructorCall*

*ConstructorCall*:

Identifier *ConstructorCall*´

*ConstructorCall*´:

**(** *ExpressionOpt* **)**

. Identifier *ConstructorCall*´

**empty**/\*Esto no estaba\*/

*MemberExpression*:

Identifier *AccesorList*

*AccesorList*:

**.** Identifier *AccesorList*

**[***AssignmentExpression***]** *AccesorList*

**(** *ExpressionOpt* **)** *AccesorList*

*PrimaryExpression*:

**(***AssignmentExpression***)**

**[** *ExpressionOpt* **]**

*IntegerLiteral*

*FloatingPointLiteral*

*StringLiteral*

*BooleanLiteral*

**null**

*BooleanLiteral:*

**true**

**false**

*AssignmentOperator:*

**=** | ***\*=*** | ***/=*** | ***%=*** | ***+=*** | ***-=*** | ***<<=*** |***>>=*** | ***>>>=*** | ***&=*** | ***^=*** | ***|=***

*RelationalOperator:*

**>**  | **>=** | **<** | **<=**

*ShiftOperator:*

**>>** | **<<** |  **>>>**

*AdditiveOperator*:

**+**  | **-**

*MultiplicativeOperator:*

**\***  |  **/** | **%**

*UnaryOperator:*

**!**  |  **~** | **- /\*Falta ~ en el lexico\*/**

*IncrementOperator:*

**++**

*DecrementOperator:*

**--**