

## “Fase # 2”

➤ Gramática Original:

- Program  $\rightarrow$  Decl<sup>+</sup>
- Decl  $\rightarrow$  VariableDecl | FunctionDecl | ConstDecl | ClassDecl | InterfaceDecl
- VariableDecl  $\rightarrow$  Variable ;
- Variable  $\rightarrow$  Type ident
- ConstDecl  $\rightarrow$  const ConstType ident;
- ConstType  $\rightarrow$  int | double | bool | string
- Type  $\rightarrow$  int | double | bool | string | ident | Type[]
- FunctionDecl  $\rightarrow$  Type ident ( Formals ) StmtBlock | void ident ( Formals ) StmtBlock
- Formals  $\rightarrow$  Variable , Formals | Variable
- ClassDecl  $\rightarrow$  class ident < : ident > < , ident<sup>+</sup> , > { Field\* }
- Field  $\rightarrow$  VariableDecl | FuncionDecl | ConstDecl
- interfaceDecl  $\rightarrow$  interface ident { Prototype\* }
- Prototype  $\rightarrow$  Type ident ( Formals ) ; | void ident ( Formals ) ;
- StmtBlock  $\rightarrow$  { VariableDecl\* ConstDecl\* Stmt\* }
- Stmt  $\rightarrow$  < Expr > ; | IfStmt | WhileStmt | ForStmt | BreakStmt | ReturnStmt | PrintStmt | StmtBlock
- IfStmt  $\rightarrow$  if ( Expr ) Stmt < else Stmt >
- WhileStmt  $\rightarrow$  while ( Expr ) Stmt
- ForStmt  $\rightarrow$  for ( Expr ; Expr ; Expr ) Stmt
- ReturnStmt  $\rightarrow$  return Expr ;
- BreakStmt  $\rightarrow$  break ;
- PrintStmt  $\rightarrow$  Console.WriteLine( Expr\* , ) ;
- Expr  $\rightarrow$  Lvalue = Expr | Constant | LValue | this | ( Expr ) | Expr + Expr | Expr \* Expr | Expr % Expr | - Expr | Expr < Expr | Expr <= Expr | Expr == Expr | Expr && Expr | ! Expr | New ( ident )
- LValue  $\rightarrow$  ident | Expr . ident
- Constant  $\rightarrow$  intConstant | doubleConstant | boolConstant | stringConstant | null

➤ Gramática Final Implementada:

0.  $\text{Program}' \rightarrow \text{Program}$
1.  $\text{Program} \rightarrow \text{Decl Decl}'$
2.  $\text{Decl}' \rightarrow \text{eps}$
3.  $\text{Decl}' \rightarrow \text{Decl Decl}'$
4.  $\text{Decl} \rightarrow \text{VariableDecl}$
5.  $\text{Decl} \rightarrow \text{FunctionDecl}$
6.  $\text{Decl} \rightarrow \text{ConstDecl}$
7.  $\text{Decl} \rightarrow \text{ClassDecl}$
8.  $\text{Decl} \rightarrow \text{InterfaceDecl}$
9.  $\text{VariableDecl} \rightarrow \text{Variable ;}$
10.  $\text{Variable} \rightarrow \text{Type ident}$
11.  $\text{ConstDecl} \rightarrow \text{const ConstType ident ;}$
12.  $\text{ConstType} \rightarrow \text{int}$
13.  $\text{ConstType} \rightarrow \text{double}$
14.  $\text{ConstType} \rightarrow \text{bool}$
15.  $\text{ConstType} \rightarrow \text{string}$
16.  $\text{Type} \rightarrow \text{int}$
17.  $\text{Type} \rightarrow \text{double}$
18.  $\text{Type} \rightarrow \text{bool}$
19.  $\text{Type} \rightarrow \text{string}$
20.  $\text{Type} \rightarrow \text{ident}$
21.  $\text{Type} \rightarrow \text{Type}[]$
22.  $\text{FunctionDecl} \rightarrow \text{Type ident ( Formals ) StmtBlock}$
23.  $\text{FunctionDecl} \rightarrow \text{void ident ( Formals ) StmtBlock}$
24.  $\text{Formals} \rightarrow \text{Variable , Formals}$
25.  $\text{Formals} \rightarrow \text{Variable}$
26.  $\text{ClassDecl} \rightarrow \text{class ident ClassDecl2 ClassDecl3 \{ Field' \}}$
27.  $\text{ClassDecl2} \rightarrow \text{: ident}$
28.  $\text{ClassDecl2} \rightarrow \text{eps}$
29.  $\text{ClassDecl3} \rightarrow \text{, ident ident' ,}$
30.  $\text{ClassDecl3} \rightarrow \text{eps}$
31.  $\text{ident}' \rightarrow \text{ident ident'}$
32.  $\text{ident}' \rightarrow \text{eps}$
33.  $\text{Field}' \rightarrow \text{Field Field'}$
34.  $\text{Field}' \rightarrow \text{eps}$
35.  $\text{Field} \rightarrow \text{VariableDecl}$
36.  $\text{Field} \rightarrow \text{FuncionDecl}$
37.  $\text{Field} \rightarrow \text{ConstDecl}$
38.  $\text{interfaceDecl} \rightarrow \text{interface ident \{ Prototype' \}}$
39.  $\text{Prototype}' \rightarrow \text{Prototype Prototype'}$
40.  $\text{Prototype}' \rightarrow \text{eps}$

Walter Orozco  
Ricardo Chian

- 41. Prototype  $\rightarrow$  Type ident ( Formals ) ;
- 42. Prototype  $\rightarrow$  void ident ( Formals ) ;
- 43. StmtBlock  $\rightarrow$  { VariableDecl' ConstDecl' Stmt' }
- 44. VariableDecl'  $\rightarrow$  VariableDecl VariableDecl'
- 45. VariableDecl'  $\rightarrow$  eps
- 46. ConstDecl'  $\rightarrow$  ConstDecl ConstDecl'
- 47. ConstDecl'  $\rightarrow$  eps
- 48. Stmt'  $\rightarrow$  Stmt Stmt'
- 49. Stmt'  $\rightarrow$  eps
- 50. Stmt  $\rightarrow$  Expr' ;
- 51. Stmt  $\rightarrow$  IfStmt
- 52. Stmt  $\rightarrow$  WhileStmt
- 53. Stmt  $\rightarrow$  ForStmt
- 54. Stmt  $\rightarrow$  BreakStmt
- 55. Stmt  $\rightarrow$  ReturnStmt
- 56. Stmt  $\rightarrow$  PrintStmt
- 57. Stmt  $\rightarrow$  StmtBlock
- 58. Expr'  $\rightarrow$  Expr
- 59. Expr'  $\rightarrow$  eps
- 60. IfStmt  $\rightarrow$  if ( Expr ) Stmt IfStmt'
- 61. IfStmt'  $\rightarrow$  else Stmt
- 62. IfStmt'  $\rightarrow$  eps
- 63. WhileStmt  $\rightarrow$  while ( Expr ) Stmt
- 64. ForStmt  $\rightarrow$  for ( Expr ; Expr ; Expr ) Stmt
- 65. ReturnStmt  $\rightarrow$  return Expr ;
- 66. BreakStmt  $\rightarrow$  break ;
- 67. PrintStmt  $\rightarrow$  Console.WriteLine( Expr Expr2 , ) ;
- 68. Expr2  $\rightarrow$  Expr Expr2
- 69. Expr2  $\rightarrow$  eps
- 70. Expr  $\rightarrow$  Lvalue = Expr
- 71. Expr  $\rightarrow$  Constant
- 72. Expr  $\rightarrow$  LValue
- 73. Expr  $\rightarrow$  this
- 74. Expr  $\rightarrow$  ( Expr )
- 75. Expr  $\rightarrow$  Expr + Expr
- 76. Expr  $\rightarrow$  Expr \* Expr
- 77. Expr  $\rightarrow$  Expr % Expr
- 78. Expr  $\rightarrow$  - Expr
- 79. Expr  $\rightarrow$  Expr < Expr
- 80. Expr  $\rightarrow$  Expr <= Expr
- 81. Expr  $\rightarrow$  Expr == Expr
- 82. Expr  $\rightarrow$  Expr && Expr
- 83. Expr  $\rightarrow$  ! Expr

Walter Orozco  
Ricardo Chian

- 84. Expr  $\rightarrow$  New ( ident )
- 85. LValue  $\rightarrow$  ident
- 86. LValue  $\rightarrow$  Expr . ident
- 87. Constant  $\rightarrow$  intConstant
- 88. Constant  $\rightarrow$  doubleConstant
- 89. Constant  $\rightarrow$  boolConstant
- 90. Constant  $\rightarrow$  stringConstant
- 91. Constant  $\rightarrow$  null

➤ Tabla de First:

<b>No terminal</b>	<b>First</b>
Program'	FunctionDec,InterfaceDecl,const,class,int,double,bool,string,ident
Program	FunctionDec,InterfaceDecl,const,class,int,double,bool,string,ident
Decl'	eps,FunctionDec,InterfaceDecl,const,class,int,double,bool,string,ident
Decl	FunctionDec,InterfaceDecl,const,class,int,double,bool,string,ident
VariableDecl	int,double,bool,string,ident
Variable	int,double,bool,string,ident
ConstDecl	const
ConstType	int,double,bool,string
Type	int,double,bool,string,ident
FunctionDec	int,double,bool,string,ident,void
Formals	int,double,bool,string,ident
ClassDecl	class
ClassDecl2	:,eps
ClassDecl3	„eps
ident'	ident,eps
Field'	eps,FuncionDecl,const,int,double,bool,string,ident
Field	FuncionDecl,const,int,double,bool,string,ident
interfaceDecl	interface
Prototype'	eps,int,double,bool,string,ident,void
Prototype	int,double,bool,string,ident,void
StmtBlock	{
VariableDecl'	eps,int,double,bool,string,ident
ConstDecl'	const,eps
Stmt'	eps,{,;,if,while,for,break,return,Console.WriteLine(Lvalue,this,(!,New,intConstant, doubleConstant,boolConstant,stringConstant,ident,+,*,%,<,<=,==,&&,.-
Stmt	{,;,if,while,for,break,return,Console.WriteLine(Lvalue,this,(!,New,intConstant, doubleConstant,boolConstant,stringConstant,ident,+,*,%,<,<=,==,&&,.-
Expr'	",Lvalue,this,(!,New,intConstant,doubleConstant,boolConstant,stringConstant, ident,+,*,%,<,<=,==,&&,.-
IfStmt	if
IfStmt'	else,eps
WhileStmt	while

Walter Orozco  
Ricardo Chian

ForStmt	For
ReturnStmt	return
BreakStmt	break
PrintStmt	Console.WriteLine(
Expr2	eps,Lvalue,this,(!,New,intConstant,doubleConstant,boolConstant,stringConstant, ident,+,*,%,<,<=,==,&&,-
Expr	Lvalue,this,(!,New,intConstant,doubleConstant,boolConstant,stringConstant, eps,ident,+,*,%,<,<=,==,&&,-
LValue	ident,Lvalue,this,(!,New,intConstant,doubleConstant,boolConstant,stringConstant, eps,+,*,%,<,<=,==,&&,-
Constant	intConstant,doubleConstant,boolConstant,stringConstant,eps