**Gramática de MiniJava**

*Program* → VarImportDecl *VarClassDecl* *MainClass*

VarImportDecl -> ImportDecl VarImportDecl| E

*ImportDecl* →**import***TypeName ImportDecl2***;**

**ImportDecl2 -> .\*** *|* ε

*VarClassDecl -> ClassDecl VarClassDecl | E*

*//ClassDecl* → **class** *id* ClassDecl2 **{** *BodyDecl* **}**

*ClassDecl* → ClassDecl2 **{** *BodyDecl* **}**

*ClassDecl2 -> class id ClassDecl3*

*//ClassDecl2 -> ClassDecl3* | ε

ClassDecl3 -> ClassDecl4 *id*

*ClassDecl4 ->* **extends** | **implements**

*TypeName* → *id TypeName2*

*//TypeName2 ->* **.***id TypeName2 | E*

*TypeName2 -> TypeName2* **.***id | E*

*MainClass* → **class** *id* **{** **public static void main** **(** **String[]** *id* **){** *Statement* **}}**

*BodyDecl* → *BodyDecl2 BodyDecl | E*

*BodyDecl2 -> VarDecl | ConstrDecl | MethodDecl | ClassDecl*

*VarDecl* → *Type id* ***;***

*ConstrDecl* → **public** *id* **(** *FormalList* **)**

**{** *VarVarDecl* *VarStatement* **}**

*VarVarDecl -> VarDecl VarVarDecl | E*

*VarStatement-> Statement VarStatement | E*

*MethodDecl* → **public** *TypeVoid id* **(** *FormalList* **)** **{** *VarVarDecl* *VarStatement* returnExp**}**

*TypeVoid -> Type |* **void**

*returnExp ->* **return** *Exp* **;** *|* ε

*FormalList* → *Type id VarTypeId* | ε

VarTypeId -> **,** *Type id VarTypeId |* ε

*Type* → **int** **[]**

→ **boolean**

→ **int**

→ *id*

*Statement* → **{** *VarStatement***}**

→ **if** **(** *Exp* **)** *Statement ElseStatement*

→ **while** **(** *Exp* **)** *Statement*

→ **System.out.println(** Exp **);**

→ **System.exit(INTEGER\_LITERAL);**

→ *id* CuadExpe **=** ExpType**;**

→ ThisId **.** *id* **(** *ExpList* **)**

*ElseStatement ->* **else** *Statement |* ε

*CuadExpe ->* **[** *Exp* **]** | ε

*ExpType ->*  *Exp* | **(** *Type* **) System.in.read()**

**ThisId -> this** | id

*Exp*  → *Exp\_Simpl* VarDistiExp

VarDistiExp -> DistiExp VarDistiExp | E

DistiExp -> *op Exp |***[** *Exp* **]**|**.** **length**| ***.*** *id* **(** *ExpList* **)**

*Exp\_Simpl* → ***INTEGER\_LITERAL***

→ **true**

→ **false**

→ *id*

→ **this**

→ **new int** **[** *Exp* **]**

→ **new** *id* **(***ExpList***)**

→ **!** *Exp*

→ **(** *Exp* **)**

→ ***STRING\_CONSTANT***

*ExpList* → *Exp VarComaExp* | ε

*VarComaExp -> , Exp VarComaExp | E*

*Op* → **+** | **-** | **\*** | **/** | **!=** | **==** | **<** | **<=** | **>=** | **>** | **||** | **&&**

**Gramática de MiniJava**

*Program* → VarImportDecl *VarClassDecl* *MainClass*

VarImportDecl -> ImportDecl VarImportDecl| E

*ImportDecl* →**import***TypeName ImportDecl2***;**

**ImportDecl2 -> .\*** *|* ε

*VarClassDecl -> ClassDecl VarClassDecl | E*

*//ClassDecl* → **class** *id* ClassDecl2 **{** *BodyDecl* **}**

*ClassDecl* → ClassDecl2 **{** *BodyDecl* **}**

*ClassDecl2 -> class id ClassDecl3*

*//ClassDecl2 -> ClassDecl3* | ε

ClassDecl3 -> ClassDecl4 *id*

*ClassDecl4 ->* **extends** | **implements**

*TypeName* → *id TypeName2*

*//TypeName2 ->* **.***id TypeName2 | E*

*TypeName2 -> TypeName2* **.***id | E*

*MainClass* → **class** *id* **{** **public static void main** **(** **String[]** *id* **){** *Statement* **}}**

*BodyDecl* → *BodyDecl2 BodyDecl | E*

*BodyDecl2 -> VarDecl | ConstrDecl | MethodDecl | ClassDecl*

*VarDecl* → *Type id* ***;***

*ConstrDecl* → **public** *id* **(** *FormalList* **)**

**{** *VarVarDecl* *VarStatement* **}**

*VarVarDecl -> VarDecl VarVarDecl | E*

*VarStatement-> Statement VarStatement | E*

*MethodDecl* → **public** *TypeVoid id* **(** *FormalList* **)** **{** *VarVarDecl* *VarStatement* returnExp**}**

*TypeVoid -> Type |* **void**

*returnExp ->* **return** *Exp* **;** *|* ε

*FormalList* → *Type id VarTypeId* | ε

VarTypeId -> **,** *Type id VarTypeId |* ε

*Type* → **int** **[]**

→ **boolean**

→ **int**

→ *id*

*Statement* → **{** *VarStatement***}**

→ **if** **(** *Exp* **)** *Statement ElseStatement*

→ **while** **(** *Exp* **)** *Statement*

→ **System.out.println(** Exp **);**

→ **System.exit(INTEGER\_LITERAL);**

→ *id* CuadExpe **=** ExpType**;**

→ ThisId **.** *id* **(** *ExpList* **)**

*ElseStatement ->* **else** *Statement |* ε

*CuadExpe ->* **[** *Exp* **]** | ε

*ExpType ->*  *Exp* | **(** *Type* **) System.in.read()**

**ThisId -> this** | id

*Exp*  → *Exp\_Simpl* VarDistiExp

VarDistiExp -> DistiExp VarDistiExp | E

DistiExp -> *op Exp |***[** *Exp* **]**|**.** **length**| ***.*** *id* **(** *ExpList* **)**

*Exp\_Simpl* → ***INTEGER\_LITERAL***

→ **true**

→ **false**

→ *id*

→ **this**

→ **new int** **[** *Exp* **]**

→ **new** *id* **(***ExpList***)**

→ **!** *Exp*

→ **(** *Exp* **)**

→ ***STRING\_CONSTANT***

*ExpList* → *Exp VarComaExp* | ε

*VarComaExp -> , Exp VarComaExp | E*

*Op* → **+** | **-** | **\*** | **/** | **!=** | **==** | **<** | **<=** | **>=** | **>** | **||** | **&&**