<Clase>=clase<<id>><Herencia><Implementación>{<Body>}<Clase>|λ

<Herencia>=extiende<<id>>|λ

<Implementación>=implementa<<id>><AUX7>|λ

<Body>=<<Acceso>><Atributo o Método>|λ

<Atributo o Método>=<Declaración>;<Body>|<<Tipodato>><<id>>(<AUX10>){<Cuerpo><Retorno>}<Body>|<<id>>(<AUX10>){<Cuerpo>}<Body>|principal(){<Cuerpo>}<Body>| λ

<Retorno>=retornar<AUX2>;| λ

<Cuerpo>=<Sentencia><Cuerpo>| λ

<Sentencia>=si(<Condición>)<Sentencia><AUX1>|mientras(<Condición>)<Sentencia>|

Sea(<<id>>){<AUX3>}<Sentencia>|hacer<sentencia>mientras(<Condición>);|{<Cuerpo>}|

para(<<id>>=<AUX2>;<Condición>;<<id>><AUX9>)<Sentencia>|

<Declaración>;|<<id>><AUX9>;|leer(<<id>>);|escribir(<Operación>);|;

<Declaración>=<<Tipodato>><<id>><AUX7>

<Condición>=<AUX2><AUX8>

<AUX1>=sino<Sentencia>| λ

<AUX2>=<<id>><Llamada>|<<num>>|<<cad>>|<<char>>|

nuevo<<id>><Mat1>(<AUX5>)|!<<id>><Llamada>|<<bool>>

<AUX3>=caso<<num>>:<Sentencia>cambio;<AUX3>|defecto:<Sentencia>cambio;

<AUX4>=<<opbool2>><Condición>| λ

<AUX5>=<AUX2><AUX6>| λ

<AUX6>=,<AUX2><AUX6>| λ

<AUX7>=,<<id>><AUX7>|=<AUX2><AUX7>| λ

<AUX8>=<<opbool1>><AUX2><AUX4>| λ

<AUX9>= =<Operación>|<<opm3>><AUX2>

<AUX10>=<<Tipodato>><<id>><AUX11>| λ

<AUX11>=,<AUX10>| λ

<Operación>=<Operación><<opm1>><O1>|<O1>

<O1>=<O1><<opm2>><O2>|<O2>

<O2>=<O2>\*\*<O3>|<O3>

<O3>=<AUX2>|(<Operación>)

<Llamada>=.<<id>>(<AUX5>)| λ

<<Acceso>>=publico|privado

<<Tipodato>>=bool|entero|real|caracter|cadena|<<id>>

<<opm1>>=+|-

<<opm2>>=\*|/|%

<<opm3>>=+=|-=|\*=|/=

<<opbool1>>=<|>|!=|<=|>=|==

<<opbool2>>=&&| || | ^

Restricciones:

El <<Acceso>> nunca es vacío y solo puede ser público o privado.

Solo se pueden usar operadores matemáticos con otro <<id>>.Ej: a+=1,b\*=3

No se pueden: a++,b--.

Solo puede haber una llamada por <<id>>.No se puede: id.id.id().

En el para no se puede declarar un <<id>>.

El sea() siempre lleva un defecto: y los casos solo pueden ser números.(No caracteres)

No hay matrices ,constantes y atributos o métodos de clases.

No hay Excepciones ni clases internas.

Error: Se dan operaciones matemáticas con cadenas y caracteres y los procedimientos pueden ser parámetros de funciones y valores de condiciones (Se soluciona con análisis semántico). Se puede implementar interfaces pero no se pueden crear interfaces(Ah ahí la maté).