<Program> 🡪 <declara-vars> <declara-función> model : <Declara-vars-data> <bloque>  
<declara-vars> 🡪 <vars>  
<declara-vars> 🡪 E  
<declara-función> 🡪 <función>  
<declara-función> 🡪 E  
<Declara-vars-data> 🡪 <vars> <data>  
<Declara-vars-data> 🡪 E

<vars> 🡪 vars { <lista-vars> }  
<lista-vars> 🡪 Declaración : <lista-ids> ; <l-vars>  
<l-vars> 🡪 E  
<l-vars> 🡪 <lista-vars>  
<lista-ids> 🡪 id <l-id>  
<l-id> 🡪 , <lista-ids>  
<l-id> 🡪 E

<Declaración> 🡪 <arreglo>  
<Declaración> 🡪 <matriz>  
<Declaración> 🡪 <Tipo>

<Arreglo> 🡪 [ Cte\_i ]

<Matriz> 🡪 [ Cte\_i ] [ Cte\_i ]

<Tipo> 🡪 int  
<Tipo> 🡪 float

<var-cte> 🡪 id  
<var-cte> 🡪 cte\_i  
<var-cte> 🡪 cte\_f  
<var-cte> 🡪 <llamada-a-función>  
<var-cte> 🡪 <acceso-a-arreglo>

<Data> 🡪 data { <lista-asignación> }  
<lista-asignación> 🡪 <asignación> ; <l-asignación>  
<l-asignación> 🡪 E  
<l-asignación> 🡪 <lista-asignación>

<Función-optimización> 🡪 min = <restricciones>  
<Función-optimización> 🡪 max = <restricciones>

<restricciones> 🡪 where <restricciones’>  
<restricciones’> 🡪 <expresión>  
<restricciones’> 🡪 <suma> ; <lista-for>  
<lista-for> 🡪 <for> ; <li-for>  
<li-for> 🡪 E  
<li-for 🡪 <lista-for>

<suma> 🡪 sum ( id : <expresión> )

<for> 🡪 for ( id : <suma> <for’>  
<for’> 🡪 < = id )  
<for’> 🡪 > = id )

<exp> 🡪 <término> <lista-terminos>  
<lista-terminos> 🡪 E  
<lista-terminos> 🡪 + <exp>  
<lista-terminos> 🡪 - <exp>

<término> 🡪 <factor> <lista-factores>  
<lista-factores> 🡪 E  
<lista-factores> 🡪 \* <término>  
<lista-factores> 🡪 / <término>

<expresión> 🡪 <exp> <lista-exp>  
<lista-exp> 🡪 E  
<lista-exp> 🡪 > <exp>  
<lista-exp> 🡪 < <exp>  
<lista-exp> 🡪 <> <exp>

<factor> 🡪 ( <expresión> )  
<factor> 🡪 <var-cte>  
<factor> 🡪 + <var-cte>  
<factor> 🡪 - <var-cte>

<asignación> 🡪 <var> = <expresión> ;

<var> 🡪 id <var’>  
<var’> 🡪 E  
<var’> 🡪 [ <expresión> ]

<escritura> 🡪 print ( <lista-strings> ) ;  
<lista-strings> 🡪 <expresión> <li-strings>  
<lista-strings> 🡪 cte.string <li-strings>  
<li-strings> 🡪 E  
<li-strings> 🡪 + <lista-strings>

<condición> 🡪 if ( <expresión> ) <bloque> <bloque-else> ;  
<bloque-else> 🡪 E  
<bloque-else> 🡪 else <bloque>

<bloque> 🡪 { <lista-estatutos> }  
<lista-estatutos> 🡪 E  
<lista-estatutos> 🡪 <estatuto> <li-estatuto>  
<li-estatuto> 🡪 E  
<li-estatuto> 🡪 <lista-estatutos>

<estatuto> 🡪 <asignación>  
<estatuto> 🡪 <condición>  
<estatuto> 🡪 <función-optimización>  
<estatuto> 🡪 <escritura>  
<estatuto> 🡪 <ciclo>  
<estatuto> 🡪 <return>

<función> 🡪 func id ( <lista-args> ) : <Declara-vars-data> <bloque>   
<lista-args> 🡪 E  
<lista-args> 🡪 <tipo> id <li-args>  
<li-args> 🡪 E  
<li-args> 🡪 , <lista-args>