```
                <Programa> ::= <Cabecera_programa> <bloque>
         <Cabecera_programa> ::= 'main'

                   <bloque> ::= <Inicio_de_bloque>
                                <Declar_de_variables_locales>
                                <Declar_de_subprogs>
                                <Sentencias>
                                <Fin_de_bloque>

        <Declar_de_subprogs> ::= <Declar_de_subprogs> <Declar_subprog>
                                 |
            <Declar_subprog> ::= <Cabecera_subprograma> <bloque>
      <Cabecera_subprograma> ::= <tipo> <identificador> '(' (<decl_parametros>)? ')'
          <decl_parametros> ::= <decl_parametros> ',' <tipo> <idenfiticador> |
                                <tipo> <identificador>

<Declar_de_variables_locales> ::= <Marca_ini_declar_variables>
                                  <Variables_locales>
                                  <Marca_fin_declar_variables>
                                  |
 <Marca_ini_declar_variables> ::= Variables <Inicio_de_bloque>
         <Variables_locales> ::= <Variables_locales> <Cuerpo_declar_variables>
                                 |
    <Cuerpo_declar_variables> ::= <declar_variable> | <declar_contenedor>
           <declar_variable> ::= <tipo> <lista_variables> ;
         <declar_contenedor> ::= list of <tipo> <lista_variables> ;
                     <tipo> ::= 'int' | 'char' | 'bool' | 'float'

 <Marca_fin_declar_variables> ::= <Fin_de_bloque>
           <Inicio_de_bloque> ::= {
              <Fin_de_bloque> ::= }

                <Sentencias> ::= <Sentencias> <Sentencia>
                                 | <Sentencia>
                 <Sentencia> ::= <bloque>
                                 | <sentencia_asignacion>
                                 | <sentencia_if>
                                 | <sentencia_while>
                                 | <sentencia switch>
                                 | <sentencia_entrada>
                                 | <sentencia_salida>
                                 | <sentencia_return>
                                 | <sentencia_break>
                                 | <sentencia_avanza>
                                 | <sentencia_retroceder>
                                 | <sentencia_comienzo>

        <sentencia_asignacion> ::= <identificador> = <expresion> ;
                <sentencia_if> ::= if '(' <expresion> ')' <bloque> (else '(' <expresion> ')' <bloque>)?
            <sentencia_while> ::= while '(' <expresion> ')' <bloque>

             <setencia_switch> ::= switch '(' <identificador> ')' <bloque_switch>
               <bloque_switch> ::= <Inicio_de_bloque> <cuerpo_switch> (<sentencia_default>)? <Fin_de_bloque>
               <cuerpo_switch> ::= <cuerpo_switch> case <constante>: <sentencias>
                                 | case <constante>: <sentencias>
           <sentencia_default> ::= default: <sentencias>

           <sentencia_entrada> ::= scanf <lista_variables> ;
            <sentencia_salida> ::= print <lista_expr_o_cadena> ;
            <sentencia_return> ::= return <expresion> ;

             <sentencia_break> ::= break ;

            <sentencia_avanza> ::= <identificador> '>>'
        <sentencia_retroceder> ::= <identificador> '<<'
          <sentencia_comienzo> ::= '$' <identificador>
```

```
          <expresion> ::= ( <expresion> )
                        | <op_unario> <expresion>
                        | <expresion> <op_binario> <expresion>
                        | <identificador>
                        | <constante>
                        | <lista_explicita>
                        | <funcion>
                        | <expresion> '++' <expresion> '@' <expresion>

    <lista_explicita> ::= '[' <lista_constantes> ']'
   <lista_constantes> ::= <lista_constantes> <constante>
                        | <constante>

           <funcion> ::= <identificador> '(' (<lista_variables>)? ')'
    <lista_variables> ::= <lista_variables> , <identificador>
                        | <identificador>

         <op_unario> ::= '!' | '#' | '?'
        <op_binario> ::= '+' | '-' | '*' | '/' | '<' | '>' | '==' | '!=' | '<=' | '>='
                           | '&&' | '||' | '^' | '@' | '--' | '%' | '**'

      <identificador> ::=  _<cadena>
                        | <letra> <cadena>
         <constante> ::= <real> | "true" | "false" | <letra>

<lista_expr_o_cadena> ::= <lista_expr_o_cadena> , <lista_expr_o_cadena>
                        | <expresion>
                        | <cadena>
            <cadena> ::= <cadena><cadena>
                        | <letra>
                        | <real>
                        |

           <natural> ::= <digito> <natural>
                        | <digito>
            <entero> ::= <natural>
                        | <signo> <entero>
              <real> ::= <entero>
                        | <entero> '.' <natural>




             <letra> ::= [a-zA-Z]
            <digito> ::= [0-9]
             <signo> ::= '+' | '-'
```