

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas

MANUAL TECNICO
Proyecto 1

Nombre: Cristian Alexander Azurdia Ajú
Carnet: 201020331

Gramáticas

- Gramática de GXML:

start with INICIO;

INICIO ::= CUERPO_GXML:n

```
{:  
  if(importar == true && traducir == true)  
  {  
    RESULT = new Sentencia_Inicio(n,entrada,importar);  
    salida = RESULT.ejecutar(entorno_global,"");  
  }  
  else if(importar == true && traducir == false)  
  {  
    RESULT = new Sentencia_Inicio(n,entrada,importar);  
    salida = RESULT.ejecutar(entorno_global, new FS_Arreglo(), entrada);  
  }  
  else if( importar == false && traducir == true)  
  {  
    RESULT = new Sentencia_Inicio(n,entrada,importar);  
    salida = RESULT.ejecutar(null,"");  
  }  
  else if( importar == false && traducir == false)  
  {  
    RESULT = new Sentencia_Inicio(n,entrada,importar);  
    salida = RESULT.ejecutar(null, new FS_Arreglo(), entrada);  
  }  
:};
```

CUERPO_GXML ::= CUERPO_GXML:n1 SENTENCIA_IMPORTAR:n2

```
{:
```

```

        RESULT = n1;
        RESULT.add(n2);
    :}
|CUERPO_GXML:n1 TAG_VENTANA:n2
{:
    RESULT = n1;
    RESULT.add(n2);
:}
|SENTENCIA_IMPORTAR:n
{:
    RESULT = new ArrayList<Instruccion>();
    RESULT.add(n);
:}
|TAG_VENTANA:n
{:
    RESULT = new ArrayList<Instruccion>();
    RESULT.add(n);
:};

```

```

/*****SENTENCIA
IMPORTAR*****/

SENTENCIA_IMPORTAR ::= s_less r_importar:n1 s_greather contenido_tag:n2
s_less s_div r_importar s_greather

{:
    RESULT = new Sentencia_Importar(n2,n1right,n1left);
:};

```

```

/*****CREAR
VENTANAS*****/

TAG_VENTANA ::= s_less:n1 r_ventana LISTA_ELEMENTOS:n2 s_greather
LISTA_CONTENIDO_VENTANA:n3 s_less s_div r_ventana s_greather

{:
    RESULT = new Sentencia_Ventana(n2,n3,n1right,n1left);
:};

```

```

    :}

    |s_less:n1 r_ventana LISTA_ELEMENTOS:n2 s_greather s_less s_div
r_ventana s_greather

    {:
        RESULT = new Sentencia_Ventana(n2,n1right,n1left);

    :};

```

```

/*****CONTENIDO
VENTANAS*****/

```

```

LISTA_CONTENIDO_VENTANA ::= LISTA_CONTENIDO_VENTANA:n1
CONTENIDO_VENTANA:n2

```

```

    {:
        RESULT = n1;
        RESULT.add(n2);

    :}

|CONTENIDO_VENTANA:n

    {:
        RESULT = new ArrayList<Instruccion>();
        RESULT.add(n);

    :};

```

```

CONTENIDO_VENTANA ::= TAG_CONTENEDOR:n

```

```

    {:
        RESULT = n;

    :}

|TAG_TEXTO:n

    {:
        RESULT = n;

    :}

|TAG_CONTROL:n

    {:
        RESULT = n;

    :}

```

```

|TAG_BOTON:n
{:
    RESULT = n;
:}
|TAG_ENVIAR:n
{:
    RESULT = n;
:}
|TAG_MULTIMEDIA:n
{:
    RESULT = n;
:};

```

TAG_CONTENEDOR ::= s_less:n1 r_contenedor LISTA_ELEMENTOS:n2 s_greather
 LISTA_CONTENIDO_VENTANA:n3 s_less s_div r_contenedor s_greather

```

{:
    RESULT = new Sentencia_Contenedor(n2,n3,n1right,n1left);
:}

```

|s_less:n1 r_contenedor LISTA_ELEMENTOS:n2 s_greather s_less s_div
 r_contenedor s_greather

```

{:
    RESULT = new Sentencia_Contenedor(n2,n1right,n1left);
:};

```

TAG_TEXTO ::= s_less:n1 r_texto LISTA_ELEMENTOS:n2 s_greather
 contenido_tag:n3 s_less s_div r_texto s_greather

```

{:
    RESULT = new Sentencia_Texto(n2,n3,n1right,n1left);
:}

```

|s_less:n1 r_texto LISTA_ELEMENTOS:n2 s_greather s_less s_div r_texto
 s_greather

```

{:
    RESULT = new Sentencia_Texto(n2,n1right,n1left);
:};

```

TAG_CONTROL ::= s_less:n1 r_control LISTA_ELEMENTOS:n2 s_greater
CONTENIDO_CONTROL:n3 s_less s_div r_control s_greater

```
{:  
    RESULT = new Sentencia_Controlador(n2,n3,n1right,n1left);  
:}  
|s_less:n1 r_control LISTA_ELEMENTOS:n2 s_greater s_less s_div  
r_control s_greater  
{:  
    RESULT = new Sentencia_Controlador(n2,n1right,n1left);  
:};
```

TAG_BOTON ::= s_less:n1 r_boton LISTA_ELEMENTOS:n2 s_greater
contenido_tag:n3 s_less s_div r_boton s_greater

```
{:  
    RESULT = new Sentencia_Boton(n2,n3,n1right,n1left,false);  
:}  
|s_less:n1 r_boton LISTA_ELEMENTOS:n2 s_greater s_less r_texto  
ELEMENTO_NOMBRE s_greater contenido_tag:n3 s_less s_div r_texto s_greater  
s_less s_div r_boton s_greater  
{:  
    RESULT = new Sentencia_Boton(n2,n3,n1right,n1left,false);  
:}  
|s_less:n1 r_boton LISTA_ELEMENTOS:n2 s_greater s_less s_div  
r_boton s_greater  
{:  
    RESULT = new Sentencia_Boton(n2,n1right,n1left,false);  
:};
```

TAG_ENVIAR ::= s_less:n1 r_enviar LISTA_ELEMENTOS:n2 s_greater
contenido_tag:n3 s_less s_div r_enviar s_greater

```
{:  
    RESULT = new Sentencia_Boton(n2,n3,n1right,n1left,true);  
:}
```

```

|s_less:n1 r_enviar LISTA_ELEMENTOS:n2 s_greather s_less r_texto
ELEMENTO_NOMBRE s_greather contenido_tag:n3 s_less s_div r_texto s_greather
s_less s_div r_enviar s_greather

```

```

{:
    RESULT = new Sentencia_Boton(n2,n3,n1right,n1left,true);
:}

```

```

|s_less:n1 r_enviar LISTA_ELEMENTOS:n2 s_greather s_less s_div
r_enviar s_greather

```

```

{:
    RESULT = new Sentencia_Boton(n2,n1right,n1left,true);
:};

```

```

TAG_MULTIMEDIA ::= s_less:n1 r_multimedia LISTA_ELEMENTOS:n2 s_greather
s_less s_div r_multimedia s_greather

```

```

{:
    RESULT = null;
:};

```

```

CONTENIDO_CONTROL ::= CONTENIDO_CONTROL:n2 TAG_DEFECTO:n1

```

```

{:
    RESULT = n2;
    RESULT.add(n1);
:}

```

```

|CONTENIDO_CONTROL:n1 TAG_LISTA_DATOS:n2

```

```

{:
    RESULT = n1;
    RESULT.addAll(n2);
:}

```

```

|TAG_LISTA_DATOS:n

```

```

{:
    RESULT = n;
:}

```

```

|TAG_DEFECTO:n

```

```

{:
    RESULT = new ArrayList<GXML_Elemento>();
    RESULT.add(n);
:};

```

TAG_DEFECTO ::= s_less:n1 r_defecto s_greater contenido_tag:n2 s_less s_div
r_defecto s_greater

```

{:
    RESULT = new GXML_Elemento("defecto",
    Tabla_Enums.tipo_primitivo_Simbolo.cadena, n2, n1right, n1left);
:};

```

TAG_LISTA_DATOS ::= s_less:n1 r_listadatos s_greater LISTA_DATOS:n2 s_less
s_div r_listadatos s_greater

```

{:
    RESULT = n2;
:}
|s_less:n1 r_listadatos s_greater s_less s_div r_listadatos s_greater
{:
    RESULT = new ArrayList<GXML_Elemento>();
:};

```

LISTA_DATOS ::= LISTA_DATOS:n1 TAG_DATO:n2

```

{:
    RESULT = n1;
    RESULT.add(n2);
:}
|TAG_DATO:n
{:
    RESULT = new ArrayList<GXML_Elemento>();
    RESULT.add(n);
:};

```


TAG_DATO ::= s_less:n1 r_dato s_greather contenido_tag:n2 s_less s_div r_dato
s_greather

```
{:  
    RESULT = new GXML_Elemento("dato",  
    Tabla_Enums.tipo_primitivo_Simbolo.cadena, n2, n1right, n1left);  
};
```

/*****ELEMENTOS DE LAS
ETIQUETAS*****/

LISTA_ELEMENTOS ::= LISTA_ELEMENTOS:n1 ELEMENTO:n2

```
{:  
    RESULT = n1;  
    RESULT.add(n2);  
:  
|ELEMENTO:n  
{:  
    RESULT = new ArrayList<GXML_Elemento>();  
    RESULT.add(n);  
};
```

ELEMENTO ::= ELEMENTO_ID:n

```
{:  
    RESULT = n;  
:  
|ELEMENTO_NOMBRE:n  
{:  
    RESULT = n;  
:  
|ELEMENTO_TIPO:n  
{:  
    RESULT = n;
```

```
:}  
|ELEMENTO_COLOR:n  
{:  
    RESULT = n;  
:}  
|ELEMENTO_ACCIONINICIAL:n  
{:  
    RESULT = n;  
:}  
|ELEMENTO_ACCIONFINAL:n  
{:  
    RESULT = n;  
:}  
|ELEMENTO_POSX:n  
{:  
    RESULT = n;  
:}  
|ELEMENTO_POSY:n  
{:  
    RESULT = n;  
:}  
|ELEMENTO_ALTO:n  
{:  
    RESULT = n;  
:}  
|ELEMENTO_ANCHO:n  
{:  
    RESULT = n;  
:}  
|ELEMENTO_BORDE:n  
{:  
    RESULT = n;
```

```
:}  
|ELEMENTO_FUENTE:n  
{:  
    RESULT = n;  
:}  
|ELEMENTO_TAM:n  
{:  
    RESULT = n;  
:}  
|ELEMENTO_NEGRITA:n  
{:  
    RESULT = n;  
:}  
|ELEMENTO_CURSIVA:n  
{:  
    RESULT = n;  
:}  
|ELEMENTO_TEXTO:n  
{:  
    RESULT = n;  
:}  
|ELEMENTO_NUMERICO:n  
{:  
    RESULT = n;  
:}  
|ELEMENTO_TEXTOAREA:n  
{:  
    RESULT = n;  
:}  
|ELEMENTO_DESPLEGABLE:n  
{:  
    RESULT = n;
```

```

:}
|ELEMENTO_MAXIMO:n
{:
    RESULT = n;
:}
|ELEMENTO_MINIMO:n
{:
    RESULT = n;
:}
|ELEMENTO_ACCION:n
{:
    RESULT = n;
:}
|ELEMENTO_REFERENCIA:n
{:
    RESULT = n;
:}
|ELEMENTO_PATH:n
{:
    RESULT = n;
:}
|ELEMENTO_AUTOREPRODUCCION:n
{:
    RESULT = n;
:};

```

ELEMENTO_ID ::= r_id:n1 s_equal cadena:n2

```

{:
    RESULT = new GXML_Elemento(n1,
    Tabla_Enums.tipo_primitivo_Simbolo.cadena, n2.substring(1,n2.length()-1), n1right,
    n1left);
:};

```

ELEMENTO_NOMBRE ::= r_nombre:n1 s_equal cadena:n2

```
{:  
    RESULT = new GXML_Elemento(n1,  
    Tabla_Enums.tipo_primitivo_Simbolo.cadena, n2.substring(1,n2.length()-1), n1right,  
    n1left);  
};
```

ELEMENTO_TIPO ::= r_tipo:n1 s_equal cadena:n2

```
{:  
    RESULT = new GXML_Elemento(n1,  
    Tabla_Enums.tipo_primitivo_Simbolo.cadena, n2.substring(1,n2.length()-1), n1right,  
    n1left);  
};
```

ELEMENTO_POSX ::= r_posx:n1 s_equal entero:n2

```
{:  
    RESULT = new GXML_Elemento(n1,  
    Tabla_Enums.tipo_primitivo_Simbolo.entero, Integer.parseInt(n2), n1right, n1left);  
};  
|r_posx:n1 s_equal decimal:n2  
{:  
    RESULT = new GXML_Elemento(n1,  
    Tabla_Enums.tipo_primitivo_Simbolo.decimal, Double.parseDouble(n2), n1right,  
    n1left);  
};
```

ELEMENTO_POSY ::= r_posy:n1 s_equal entero:n2

```
{:  
    RESULT = new GXML_Elemento(n1,  
    Tabla_Enums.tipo_primitivo_Simbolo.entero, Integer.parseInt(n2), n1right, n1left);  
};  
|r_posy:n1 s_equal decimal:n2  
{:
```

```
        RESULT = new GXML_Elemento(n1,  
Tabla_Enums.tipo_primitivo_Simbolo.decimal, Double.parseDouble(n2), n1right,  
n1left);
```

```
    :};
```

```
ELEMENTO_ANCHO ::= r_alto:n1 s_equal entero:n2
```

```
{:
```

```
        RESULT = new GXML_Elemento(n1,  
Tabla_Enums.tipo_primitivo_Simbolo.entero, Integer.parseInt(n2), n1right, n1left);
```

```
    :}
```

```
|r_alto:n1 s_equal decimal:n2
```

```
{:
```

```
        RESULT = new GXML_Elemento(n1,  
Tabla_Enums.tipo_primitivo_Simbolo.decimal, Double.parseDouble(n2), n1right,  
n1left);
```

```
    :};
```

```
ELEMENTO_ALTO ::= r_ancho:n1 s_equal entero:n2
```

```
{:
```

```
        RESULT = new GXML_Elemento(n1,  
Tabla_Enums.tipo_primitivo_Simbolo.entero, Integer.parseInt(n2), n1right, n1left);
```

```
    :}
```

```
|r_ancho:n1 s_equal decimal:n2
```

```
{:
```

```
        RESULT = new GXML_Elemento(n1,  
Tabla_Enums.tipo_primitivo_Simbolo.decimal, Double.parseDouble(n2), n1right,  
n1left);
```

```
    :};
```

```
ELEMENTO_BORDE ::= r_borde:n1 s_equal booleano:n2
```

```
{:
```

```
        RESULT = new GXML_Elemento(n1,  
Tabla_Enums.tipo_primitivo_Simbolo.booleano, n2.equalsIgnoreCase("verdadero") ?  
true : false, n1right, n1left);
```

```
    :};
```

ELEMENTO_COLOR ::= r_color:n1 s_equal cadena:n2

```
{:  
    RESULT = new GXML_Elemento(n1,  
    Tabla_Enums.tipo_primitivo_Simbolo.cadena, n2.substring(1,n2.length()-1), n1right,  
    n1left);  
};
```

ELEMENTO_FUENTE ::= r_fuente:n1 s_equal cadena:n2

```
{:  
    RESULT = new GXML_Elemento(n1,  
    Tabla_Enums.tipo_primitivo_Simbolo.cadena, n2.substring(1,n2.length()-1), n1right,  
    n1left);  
};
```

ELEMENTO_TAM ::= r_tamano:n1 s_equal entero:n2

```
{:  
    RESULT = new GXML_Elemento(n1,  
    Tabla_Enums.tipo_primitivo_Simbolo.entero, Integer.parseInt(n2), n1right, n1left);  
};  
|r_tamano:n1 s_equal decimal:n2  
{:  
    RESULT = new GXML_Elemento(n1,  
    Tabla_Enums.tipo_primitivo_Simbolo.entero, Double.parseDouble(n2), n1right, n1left);  
};
```

ELEMENTO_NEGRITA ::= r_negrita:n1 s_equal booleano:n2

```
{:  
    RESULT = new GXML_Elemento(n1,  
    Tabla_Enums.tipo_primitivo_Simbolo.booleano, n2.equalsIgnoreCase("verdadero") ?  
    true : false, n1right, n1left);  
};
```

ELEMENTO_CURSIVA ::= r_cursiva:n1 s_equal booleano:n2

```
{:
```

```

        RESULT = new GXML_Elemento(n1,
        Tabla_Enums.tipo_primitivo_Simbolo.booleano, n2.equalsIgnoreCase("verdadero") ?
        true : false, n1right, n1left);

        :};

```

ELEMENTO_MAXIMO ::= r_maximo:n1 s_equal entero:n2

```

        {:
        RESULT = new GXML_Elemento(n1,
        Tabla_Enums.tipo_primitivo_Simbolo.entero, Integer.parseInt(n2), n1right, n1left);

        :}

|r_maximo:n1 s_equal decimal:n2

        {:
        RESULT = new GXML_Elemento(n1,
        Tabla_Enums.tipo_primitivo_Simbolo.entero, Double.parseDouble(n2), n1right, n1left);

        :};

```

ELEMENTO_MINIMO ::= r_minimo:n1 s_equal entero:n2

```

        {:
        RESULT = new GXML_Elemento(n1,
        Tabla_Enums.tipo_primitivo_Simbolo.booleano, Integer.parseInt(n2), n1right, n1left);

        :}

|r_minimo:n1 s_equal decimal:n2

        {:
        RESULT = new GXML_Elemento(n1,
        Tabla_Enums.tipo_primitivo_Simbolo.booleano, Double.parseDouble(n2), n1right,
        n1left);

        :};

```

ELEMENTO_REFERENCIA ::= r_referencia:n1 s_equal cadena:n2

```

        {:
        RESULT = new GXML_Elemento(n1,
        Tabla_Enums.tipo_primitivo_Simbolo.cadena, n2.substring(1,n2.length()-1), n1right,
        n1left);

        :};

```


ELEMENTO_ACCIONINICIAL ::= r_accioninicial:n1 s_equal s_key_open
SENTENCIA_LLAMADA:n2 s_key_close

{:

RESULT = new GXML_Elemento(n1,
Tabla_Enums.tipo_primitivo_Simbolo.identificador, n2, n1right, n1left);

:};

ELEMENTO_ACCIONFINAL ::= r_accionfinal:n1 s_equal s_key_open
SENTENCIA_LLAMADA:n2 s_key_close

{:

RESULT = new GXML_Elemento(n1,
Tabla_Enums.tipo_primitivo_Simbolo.identificador, n2, n1right, n1left);

:};

ELEMENTO_ACCION ::= r_accion:n1 s_equal s_key_open
SENTENCIA_LLAMADA:n2 s_key_close

{:

RESULT = new GXML_Elemento(n1,
Tabla_Enums.tipo_primitivo_Simbolo.identificador, n2, n1right, n1left);

:};

ELEMENTO_PATH ::= r_path:n1 s_equal cadena:n2

{:

RESULT = new GXML_Elemento(n1,
Tabla_Enums.tipo_primitivo_Simbolo.cadena, n2, n1right, n1left);

:};

ELEMENTO_AUTOREPRODUCCION ::= r_autoreproduccion:n1 s_equal
booleano:n2

{:

RESULT = new GXML_Elemento(n1,
Tabla_Enums.tipo_primitivo_Simbolo.booleano, n2, n1right, n1left);

:};

/*****EXPRESIONES*****/
*****/

SENTENCIA_LLAMADA ::= identificador:n1 s_par_open LISTA_EXPRESIONES:n2
s_par_close

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta("SENTENCIA_LLAMADA");  
    RESULT.setValor(n1);  
    RESULT.setFila(String.valueOf(n1right));  
    RESULT.setColumna(String.valueOf(n1left));  
    RESULT.getHijos().add(n2);  
:}
```

|identificador:n1 s_par_open s_par_close

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta("SENTENCIA_LLAMADA");  
    RESULT.setValor(n1);  
    RESULT.setFila(String.valueOf(n1right));  
    RESULT.setColumna(String.valueOf(n1left));  
:};
```

/*****EXPRESIONES*****/
*****/

LISTA_EXPRESIONES ::= LISTA_EXPRESIONES:n1 s_coma EXPRESION:n2

```
{:  
    Nodo_AST_FS nodo_expresion = new Nodo_AST_FS();  
    nodo_expresion.setEtiqueta("EXPRESION");  
    nodo_expresion.setValor("");  
    nodo_expresion.setFila(String.valueOf(n2right));  
    nodo_expresion.setColumna(String.valueOf(n2left));  
    nodo_expresion.getHijos().add(n2);  
  
    RESULT = n1;  
    RESULT.getHijos().add(nodo_expresion);
```

```

:}
|EXPRESION:n
{:
    Nodo_AST_FS nodo_expresion = new Nodo_AST_FS();
    nodo_expresion.setEtiqueta("EXPRESION");
    nodo_expresion.setValor("");
    nodo_expresion.setFila(String.valueOf(nright));
    nodo_expresion.setColumna(String.valueOf(nleft));
    nodo_expresion.getHijos().add(n);

    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("LISTA_EXPRESIONES");
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(nright));
    RESULT.setColumna(String.valueOf(nleft));
    RESULT.getHijos().add(nodo_expresion);
:};

```

EXPRESION ::= EXPRESION_ARITMETICA:n

```

{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("EXPRESION_ARITMETICA");
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(nright));
    RESULT.setColumna(String.valueOf(nleft));
    RESULT.getHijos().add(n);
:}
|EXPRESION_RELACIONAL:n
{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("EXPRESION_RELACIONAL");

```

```

    RESULT.setValor("");
    RESULT.setFila(String.valueOf(nright));
    RESULT.setColumna(String.valueOf(nleft));
    RESULT.getHijos().add(n);
:}

|EXPRESION_UNARIA:n
{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("EXPRESION_UNARIA");
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(nright));
    RESULT.setColumna(String.valueOf(nleft));
    RESULT.getHijos().add(n);
:}

|EXPRESION_LOGICA:n
{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("EXPRESION_LOGICA");
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(nright));
    RESULT.setColumna(String.valueOf(nleft));
    RESULT.getHijos().add(n);
:}

|s_par_open EXPRESION:n s_par_close
{:
    RESULT = n;
:}

|DATO_PRIMITIVO:n
{:
    RESULT = n;
:};

```

EXPRESION_ARITMETICA ::= EXPRESION:n1 s_plus:o EXPRESION:n2

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta(o);  
    RESULT.setValor("");  
    RESULT.setFila(String.valueOf(n1 right));  
    RESULT.setColumna(String.valueOf(n1 left));  
    RESULT.getHijos().add(n1);  
    RESULT.getHijos().add(n2);
```

:{}

|EXPRESION:n1 s_minus:o EXPRESION:n2

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta(o);  
    RESULT.setValor("");  
    RESULT.setFila(String.valueOf(n1 right));  
    RESULT.setColumna(String.valueOf(n1 left));  
    RESULT.getHijos().add(n1);  
    RESULT.getHijos().add(n2);
```

:{}

|EXPRESION:n1 s_mul:o EXPRESION:n2

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta(o);  
    RESULT.setValor("");  
    RESULT.setFila(String.valueOf(n1 right));  
    RESULT.setColumna(String.valueOf(n1 left));  
    RESULT.getHijos().add(n1);  
    RESULT.getHijos().add(n2);
```

:{}

|EXPRESION:n1 s_div:o EXPRESION:n2

```
{:
```

```

    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta(o);
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(n1right));
    RESULT.setColumna(String.valueOf(n1left));
    RESULT.getHijos().add(n1);
    RESULT.getHijos().add(n2);
:}
|EXPRESION:n1 s_mod:o EXPRESION:n2
{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta(o);
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(n1right));
    RESULT.setColumna(String.valueOf(n1left));
    RESULT.getHijos().add(n1);
    RESULT.getHijos().add(n2);
:}
|EXPRESION:n1 s_to:o EXPRESION:n2
{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta(o);
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(n1right));
    RESULT.setColumna(String.valueOf(n1left));
    RESULT.getHijos().add(n1);
    RESULT.getHijos().add(n2);
:}
|EXPRESION:n1 s_increment:o
{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta(o);

```

```

    RESULT.setValor("");
    RESULT.setFila(String.valueOf(n1 right));
    RESULT.setColumna(String.valueOf(n1 left));
    RESULT.getHijos().add(n1);
  :}
|EXPRESION:n1 s_decrement:o
{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta(o);
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(n1 right));
    RESULT.setColumna(String.valueOf(n1 left));
    RESULT.getHijos().add(n1);
  :};

```

EXPRESION_UNARIA ::= s_minus:o EXPRESION:n1

```

{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta(o);
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(n1 right));
    RESULT.setColumna(String.valueOf(n1 left));
    RESULT.getHijos().add(n1);
  :}
|s_plus:o EXPRESION:n1
{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta(o);
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(n1 right));
    RESULT.setColumna(String.valueOf(n1 left));
    RESULT.getHijos().add(n1);
  :}

```

```
};
```

```
EXPRESION_LOGICA ::= EXPRESION:n1 s_or:o EXPRESION:n2
```

```
{:
```

```
    RESULT = new Nodo_AST_FS();
```

```
    RESULT.setEtiqueta(o);
```

```
    RESULT.setValor("");
```

```
    RESULT.setFila(String.valueOf(n1right));
```

```
    RESULT.setColumna(String.valueOf(n1left));
```

```
    RESULT.getHijos().add(n1);
```

```
    RESULT.getHijos().add(n2);
```

```
};
```

```
|EXPRESION:n1 s_and:o EXPRESION:n2
```

```
{:
```

```
    RESULT = new Nodo_AST_FS();
```

```
    RESULT.setEtiqueta(o);
```

```
    RESULT.setValor("");
```

```
    RESULT.setFila(String.valueOf(n1right));
```

```
    RESULT.setColumna(String.valueOf(n1left));
```

```
    RESULT.getHijos().add(n1);
```

```
    RESULT.getHijos().add(n2);
```

```
};
```

```
|s_not:o EXPRESION:n1
```

```
{:
```

```
    RESULT = new Nodo_AST_FS();
```

```
    RESULT.setEtiqueta(o);
```

```
    RESULT.setValor("");
```

```
    RESULT.setFila(String.valueOf(n1right));
```

```
    RESULT.setColumna(String.valueOf(n1left));
```

```
    RESULT.getHijos().add(n1);
```

```
};
```


EXPRESION_RELACIONAL ::= EXPRESION:n1 s_greater:o EXPRESION:n2

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta(o);  
    RESULT.setValor("");  
    RESULT.setFila(String.valueOf(n1 right));  
    RESULT.setColumna(String.valueOf(n1 left));  
    RESULT.getHijos().add(n1);  
    RESULT.getHijos().add(n2);  
    :}
```

|EXPRESION:n1 s_less:o EXPRESION:n2

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta(o);  
    RESULT.setValor("");  
    RESULT.setFila(String.valueOf(n1 right));  
    RESULT.setColumna(String.valueOf(n1 left));  
    RESULT.getHijos().add(n1);  
    RESULT.getHijos().add(n2);  
    :}
```

|EXPRESION:n1 s_greater_equal:o EXPRESION:n2

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta(o);  
    RESULT.setValor("");  
    RESULT.setFila(String.valueOf(n1 right));  
    RESULT.setColumna(String.valueOf(n1 left));  
    RESULT.getHijos().add(n1);  
    RESULT.getHijos().add(n2);  
    :}
```

|EXPRESION:n1 s_less_equal:o EXPRESION:n2

```
{:
```

```

    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta(o);
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(n1right));
    RESULT.setColumna(String.valueOf(n1left));
    RESULT.getHijos().add(n1);
    RESULT.getHijos().add(n2);
  :}
|EXPRESION:n1 s_compare:o EXPRESION:n2
{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta(o);
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(n1right));
    RESULT.setColumna(String.valueOf(n1left));
    RESULT.getHijos().add(n1);
    RESULT.getHijos().add(n2);
  :}
|EXPRESION:n1 s_diferent:o EXPRESION:n2
{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta(o);
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(n1right));
    RESULT.setColumna(String.valueOf(n1left));
    RESULT.getHijos().add(n1);
    RESULT.getHijos().add(n2);
  :};

```

DATO_PRIMITIVO ::= r_nulo:v

```

{:
    RESULT = new Nodo_AST_FS();

```

```
RESULT.setEtiqueta(Tabla_Enums.tipo_primitivo_Simbolo.nulo.toString());
```

```
    RESULT.setValor(v);
```

```
    RESULT.setFila(String.valueOf(vright));
```

```
    RESULT.setColumna(String.valueOf(vleft));
```

```
    :}
```

```
|booleano:v
```

```
{:
```

```
    RESULT = new Nodo_AST_FS();
```

```
RESULT.setEtiqueta(Tabla_Enums.tipo_primitivo_Simbolo.booleano.toString());
```

```
    RESULT.setValor(v);
```

```
    RESULT.setFila(String.valueOf(vright));
```

```
    RESULT.setColumna(String.valueOf(vleft));
```

```
    :}
```

```
|entero:v
```

```
{:
```

```
    RESULT = new Nodo_AST_FS();
```

```
RESULT.setEtiqueta(Tabla_Enums.tipo_primitivo_Simbolo.entero.toString());
```

```
    RESULT.setValor(v);
```

```
    RESULT.setFila(String.valueOf(vright));
```

```
    RESULT.setColumna(String.valueOf(vleft));
```

```
    :}
```

```
|decimal:v
```

```
{:
```

```
    RESULT = new Nodo_AST_FS();
```

```
RESULT.setEtiqueta(Tabla_Enums.tipo_primitivo_Simbolo.decimal.toString());
```

```
    RESULT.setValor(v);
```

```
    RESULT.setFila(String.valueOf(vright));
```

```
    RESULT.setColumna(String.valueOf(vleft));
```

```
    :}
```

```
|caracter:v
```

```

{:
    RESULT = new Nodo_AST_FS();

RESULT.setEtiqueta(Tabla_Enums.tipo_primitivo_Simbolo.caracter.toString());
    RESULT.setValor(v);
    RESULT.setFila(String.valueOf(vright));
    RESULT.setColumna(String.valueOf(vleft));
:}
|cadena:v
{:
    RESULT = new Nodo_AST_FS();

RESULT.setEtiqueta(Tabla_Enums.tipo_primitivo_Simbolo.cadena.toString());
    RESULT.setValor(v);
    RESULT.setFila(String.valueOf(vright));
    RESULT.setColumna(String.valueOf(vleft));
:}
|identificador:v
{:
    RESULT = new Nodo_AST_FS();

RESULT.setEtiqueta(Tabla_Enums.tipo_primitivo_Simbolo.identificador.toString());
    RESULT.setValor(v);
    RESULT.setFila(String.valueOf(vright));
    RESULT.setColumna(String.valueOf(vleft));
:};

```

- Gramática de FuncionScript:

start with INICIO;

INICIO ::= CUERPO_FS:n

```

{:
    raiz = new Nodo_AST_FS();

```

```

    raiz.setEtiqueta("AST");
    raiz.setValor("");
    raiz.setFila(String.valueOf(nright));
    raiz.setColumna(String.valueOf(nleft));
    raiz.getHijos().add(n);
    ejecutar_AST();
};

```

CUERPO_FS ::= CUERPO_FS:n1 SENTENCIA_IMPORTAR:n2

```

{:
    RESULT = n1;
    RESULT.getHijos().add(n2);
:}
|CUERPO_FS:n1 FUNCION:n2
{:
    RESULT = n1;
    RESULT.getHijos().add(n2);
:}
|CUERPO_FS:n1 SENTENCIA:n2
{:
    RESULT = n1;
    RESULT.getHijos().add(n2);
:}
|SENTENCIA_IMPORTAR:n
{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("CUERPO_FS");
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(nright));
    RESULT.setColumna(String.valueOf(nleft));
    RESULT.getHijos().add(n);
:}

```

|FUNCION:n

{:

```
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("CUERPO_FS");
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(nright));
    RESULT.setColumna(String.valueOf(nleft));
    RESULT.getHijos().add(n);
```

};

|SENTENCIA:n

{:

```
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("CUERPO_FS");
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(nright));
    RESULT.setColumna(String.valueOf(nleft));
    RESULT.getHijos().add(n);
```

};

SENTENCIA_IMPORTAR ::= r_importar:n1 s_par_open cadena:n2 s_par_close
s_dotcoma

{:

```
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("SENTENCIA_IMPORTAR");
    RESULT.setValor(n2);
    RESULT.setFila(String.valueOf(n1right));
    RESULT.setColumna(String.valueOf(n1left));
```

};

FUNCION ::= r_funcion:n identificador:n1 s_par_open LISTA_PARAMETROS:n2
s_par_close s_key_open LISTA_SENTENCIAS:n3 s_key_close

{:

```

    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("FUNCION");
    RESULT.setValor(n1);
    RESULT.setFila(String.valueOf(nright));
    RESULT.setColumna(String.valueOf(nleft));
    RESULT.getHijos().add(n3);
    RESULT.getHijos().add(n2);
    :}

    |r_funcion:n identificador:n1 s_par_open s_par_close s_key_open
LISTA_SENTENCIAS:n2 s_key_close
    {:
        RESULT = new Nodo_AST_FS();
        RESULT.setEtiqueta("FUNCION");
        RESULT.setValor(n1);
        RESULT.setFila(String.valueOf(nright));
        RESULT.setColumna(String.valueOf(nleft));
        RESULT.getHijos().add(n2);
    :};

LISTA_SENTENCIAS ::=  LISTA_SENTENCIAS:n1 SENTENCIA:n2
    {:
        RESULT = n1;
        RESULT.getHijos().add(n2);
    :}
    | SENTENCIA:n
    {:
        RESULT = new Nodo_AST_FS();
        RESULT.setEtiqueta("LISTA_SENTENCIAS");
        RESULT.setValor("");
        RESULT.setFila(String.valueOf(nright));
        RESULT.setColumna(String.valueOf(nleft));
        RESULT.getHijos().add(n);
    :};

```

SENTENCIA ::= SENTENCIA_DECLARACION:n s_dotcoma

{:

RESULT = n;

:}

|SENTENCIA_ASIGNACION:n s_dotcoma

{:

RESULT = n;

:}

|SENTENCIA_IMPRIMIR:n s_dotcoma

{:

RESULT = n;

:}

|SENTENCIA_SI:n

{:

RESULT = n;

:}

|SENTENCIA_SELECCIONA:n

{:

RESULT = n;

:}

|SENTENCIA_DETENER:n s_dotcoma

{:

RESULT = n;

:}

|SENTENCIA_LLAMADA:n s_dotcoma

{:

RESULT = n;

:}

|SENTENCIA_RETORNAR:n s_dotcoma

{:

RESULT = n;


```
:}  
|SENTENCIA_LLAMADA_FUNCIONES_ARREGLO:n s_dotcoma  
{:  
    RESULT = n;  
:}  
|SENTENCIA_CREAR_VENTANA:n s_dotcoma  
{:  
    RESULT = n;  
:}  
|SENTENCIA_LEER_GXML:n s_dotcoma  
{:  
    RESULT = n;  
:}  
|SENTENCIA_CREAR_CONTENEDOR:n s_dotcoma  
{:  
    RESULT = n;  
:}  
|SENTENCIA_CREAR_TEXTO:n s_dotcoma  
{:  
    RESULT = n;  
:}  
|SENTENCIA_CREAR_CAJA_TEXTO:n s_dotcoma  
{:  
    RESULT = n;  
:}  
|SENTENCIA_CREAR_AREA_TEXTO:n s_dotcoma  
{:  
    RESULT = n;  
:}  
|SENTENCIA_CREAR_CONTROL_NUMERICO:n s_dotcoma  
{:  
    RESULT = n;
```

```
:}  
|SENTENCIA_CREAR_DESPLEGABLE:n s_dotcoma  
{:  
    RESULT = n;  
:}  
|SENTENCIA_CREAR_BOTON:n s_dotcoma  
{:  
    RESULT = n;  
:}  
|SENTENCIA_CREAR_IMAGEN:n s_dotcoma  
{:  
    RESULT = n;  
:}  
|SENTENCIA_CREAR_REPRODUCTOR:n s_dotcoma  
{:  
    RESULT = n;  
:}  
|SENTENCIA_CREAR_VIDEO:n s_dotcoma  
{:  
    RESULT = n;  
:}  
|SENTENCIA_EVENTO_ALCLIC:n s_dotcoma  
{:  
    RESULT = n;  
:}  
|SENTENCIA_EVENTO_ALCARGAR:n s_dotcoma  
{:  
    RESULT = n;  
:}  
|SENTENCIA_EVENTO_ALCERRAR:n s_dotcoma  
{:  
    RESULT = n;
```

```

:}

|SENTENCIA_CREAR_DESDE_ARCHIVO:n s_dotcoma
{:
    RESULT = n;
:}

|SENTENCIA_INCREMENTO:n s_dotcoma
{:
    RESULT = n;
:}

|SENTENCIA_DECREMENTO:n s_dotcoma
{:
    RESULT = n;
:};

```

```

/*****INSTRUCCIONES*****/
*****/

```

```

//DECLACION

```

```

SENTENCIA_DECLARACION ::= r_var:n1 LISTA_IDENTIFICADORES:n2 s_equal
VALOR:n3

```

```

{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("SENTENCIA_DECLARACION");
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(n1 right));
    RESULT.setColumna(String.valueOf(n1 left));
    RESULT.getHijos().add(n2);
    RESULT.getHijos().add(n3);
:}

|r_var:n1 LISTA_IDENTIFICADORES:n2 s_dotcoma
{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("SENTENCIA_DECLARACION");

```

```

        RESULT.setValor("");
        RESULT.setFila(String.valueOf(n1 right));
        RESULT.setColumna(String.valueOf(n1 left));
        RESULT.getHijos().add(n2);
    };

//ASIGNACION
SENTENCIA_ASIGNACION ::=  LISTA_IDENTIFICADORES:n1 s_equal VALOR:n2
    {
        RESULT = new Nodo_AST_FS();
        RESULT.setEtiqueta("SENTENCIA_ASIGNACION");
        RESULT.setValor("");
        RESULT.setFila(String.valueOf(n1 right));
        RESULT.setColumna(String.valueOf(n1 left));
        RESULT.getHijos().add(n1);
        RESULT.getHijos().add(n2);
    }
|SENTENCIA_ACCESO:n1 s_equal VALOR:n2
    {
        RESULT = new Nodo_AST_FS();
        RESULT.setEtiqueta("SENTENCIA_ASIGNACION");
        RESULT.setValor("");
        RESULT.setFila(String.valueOf(n1 right));
        RESULT.setColumna(String.valueOf(n1 left));
        RESULT.getHijos().add(n1);
        RESULT.getHijos().add(n2);
    };

VALOR ::= EXPRESION:n
    {
        RESULT = new Nodo_AST_FS();
        RESULT.setEtiqueta("EXPRESION");
        RESULT.setValor("");
    }

```

```

    RESULT.setFila(String.valueOf(nright));
    RESULT.setColumna(String.valueOf(nleft));
    RESULT.getHijos().add(n);
  :}
|ARREGLO:n
{:
    RESULT = n;
  :}
|OBJETO:n
{:
    RESULT = n;
  :};

```

SENTENCIA_IMPRIMIR ::= r_imprimir:n s_par_open EXPRESION:n2 s_par_close

```

{:
    Nodo_AST_FS nodo_expresion = new Nodo_AST_FS();
    nodo_expresion.setEtiqueta("EXPRESION");
    nodo_expresion.setValor("");
    nodo_expresion.setFila(String.valueOf(n2right));
    nodo_expresion.setColumna(String.valueOf(n2left));
    nodo_expresion.getHijos().add(n2);

    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("SENTENCIA_IMPRIMIR");
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(nright));
    RESULT.setColumna(String.valueOf(nleft));
    RESULT.getHijos().add(nodo_expresion);
  :};

```

SENTENCIA_SELECCION ::= EXPRESION:n1 s_ternario EXPRESION:n2
s_dobledot EXPRESION:n3

```

{:

```

```

        Nodo_AST_FS nodo_expresion_relacional = new Nodo_AST_FS();

nodo_expresion_relacional.setEtiqueta("EXPRESION_RELACIONAL");

        nodo_expresion_relacional.setValor("");
        nodo_expresion_relacional.setFila(String.valueOf(n1right));
        nodo_expresion_relacional.setColumna(String.valueOf(n1left));
        nodo_expresion_relacional.getHijos().add(n1);

        RESULT = new Nodo_AST_FS();
        RESULT.setEtiqueta("SENTENCIA_SELECCION");
        RESULT.setValor("");
        RESULT.setFila(String.valueOf(n1right));
        RESULT.setColumna(String.valueOf(n1left));
        RESULT.getHijos().add(n1);
        RESULT.getHijos().add(n2);
        RESULT.getHijos().add(n3);

};

```

SENTENCIA_SI ::= r_si:i s_par_open EXPRESION:n1 s_par_close s_key_open
LISTA_SENTENCIAS:n2 s_key_close

```

{
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("SENTENCIA_SI");
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(iright));
    RESULT.setColumna(String.valueOf(ileft));
    RESULT.getHijos().add(n1);
    RESULT.getHijos().add(n2);

}

|r_si:i s_par_open EXPRESION:n1 s_par_close s_key_open
LISTA_SENTENCIAS:n2 s_key_close r_sino s_key_open LISTA_SENTENCIAS:n3
s_key_close

{
    RESULT = new Nodo_AST_FS();

```

```

        RESULT.setEtiqueta("SENTENCIA_SI");
        RESULT.setValor("");
        RESULT.setFila(String.valueOf(iright));
        RESULT.setColumna(String.valueOf(ileft));
        RESULT.getHijos().add(n1);
        RESULT.getHijos().add(n2);
        RESULT.getHijos().add(n3);
    :}

    |r_si:i s_par_open EXPRESION:n1 s_par_close s_key_open
LISTA_SENTENCIAS:n2 s_key_close LISTA_SENTENCIAS_SINO_SI:n3
    {:
        RESULT = new Nodo_AST_FS();
        RESULT.setEtiqueta("SENTENCIA_SI");
        RESULT.setValor("");
        RESULT.setFila(String.valueOf(iright));
        RESULT.setColumna(String.valueOf(ileft));
        RESULT.getHijos().add(n1);
        RESULT.getHijos().add(n2);
        RESULT.getHijos().add(n3);
    :}

    |r_si:i s_par_open EXPRESION:n1 s_par_close s_key_open
LISTA_SENTENCIAS:n2 s_key_close LISTA_SENTENCIAS_SINO_SI:n3 r_sino
s_key_open LISTA_SENTENCIAS:n4 s_key_close
    {:
        RESULT = new Nodo_AST_FS();
        RESULT.setEtiqueta("SENTENCIA_SI");
        RESULT.setValor("");
        RESULT.setFila(String.valueOf(iright));
        RESULT.setColumna(String.valueOf(ileft));
        RESULT.getHijos().add(n1);
        RESULT.getHijos().add(n2);
        RESULT.getHijos().add(n3);
        RESULT.getHijos().add(n4);
    }

```

```
};
```

```
SENTENCIA_SELECCIONA ::= r_selecciona:n s_par_open EXPRESION:n1  
s_par_close s_key_open LISTA_SENTENCIAS_CASOS:n2 r_defecto s_dobledot  
s_key_open LISTA_SENTENCIAS:n3 s_key_close s_key_close
```

```
{:
```

```
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta("SENTENCIA_SELECCIONA");  
    RESULT.setValor("");  
    RESULT.setFila(String.valueOf(nright));  
    RESULT.setColumna(String.valueOf(nleft));  
    RESULT.getHijos().add(n1);  
    RESULT.getHijos().add(n2);  
    RESULT.getHijos().add(n3);
```

```
};
```

```
|r_selecciona:n s_par_open EXPRESION:n1 s_par_close s_key_open  
LISTA_SENTENCIAS_CASOS:n2 s_key_close
```

```
{:
```

```
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta("SENTENCIA_SELECCIONA");  
    RESULT.setValor("");  
    RESULT.setFila(String.valueOf(nright));  
    RESULT.setColumna(String.valueOf(nleft));  
    RESULT.getHijos().add(n1);  
    RESULT.getHijos().add(n2);
```

```
};
```

```
SENTENCIA_CASO ::= r_caso:n EXPRESION:n1 s_dobledot s_key_open  
LISTA_SENTENCIAS:n2 s_key_close
```

```
{:
```

```
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta("SENTENCIA_CASOS");  
    RESULT.setValor("");  
    RESULT.setFila(String.valueOf(nright));
```



```

        RESULT.setColumna(String.valueOf(nleft));
        RESULT.getHijos().add(n1);
        RESULT.getHijos().add(n2);
    };

```

SENTENCIA_SINO_SI ::= r_sino r_si: i s_par_open EXPRESION: n1 s_par_close
s_key_open LISTA_SENTENCIAS: n2 s_key_close

```

    {
        RESULT = new Nodo_AST_FS();
        RESULT.setEtiqueta("SENTENCIA_SINO_SI");
        RESULT.setValor("");
        RESULT.setFila(String.valueOf(iright));
        RESULT.setColumna(String.valueOf(ileft));
        RESULT.getHijos().add(n1);
        RESULT.getHijos().add(n2);
    };

```

SENTENCIA_DETENER ::= r_detener: n

```

    {
        RESULT = new Nodo_AST_FS();
        RESULT.setEtiqueta("SENTENCIA_DETENER");
        RESULT.setValor("");
        RESULT.setFila(String.valueOf(nright));
        RESULT.setColumna(String.valueOf(nleft));
    };

```

SENTENCIA_LLAMADA ::= identificador: n1 s_par_open LISTA_EXPRESIONES: n2
s_par_close

```

    {
        RESULT = new Nodo_AST_FS();
        RESULT.setEtiqueta("SENTENCIA_LLAMADA");
        RESULT.setValor(n1);
        RESULT.setFila(String.valueOf(n1right));
    };

```

```

        RESULT.setColumna(String.valueOf(n1left));
        RESULT.getHijos().add(n2);
    :}
|identificador:n1 s_par_open s_par_close
{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("SENTENCIA_LLAMADA");
    RESULT.setValor(n1);
    RESULT.setFila(String.valueOf(n1right));
    RESULT.setColumna(String.valueOf(n1left));
:};

```

SENTENCIA_RETORNAR ::= r_retornar:n EXPRESION:n1

```

{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("SENTENCIA_RETORNAR");
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(nright));
    RESULT.setColumna(String.valueOf(nleft));
    RESULT.getHijos().add(n1);
:};

```

SENTENCIA_ACCESO ::= identificador:n1 s_dot identificador:n2

```

{:
    Nodo_AST_FS nodo_id = new Nodo_AST_FS();
    nodo_id.setEtiqueta("IDENTIFICADOR");
    nodo_id.setValor(n2);
    nodo_id.setFila(String.valueOf(n2right));
    nodo_id.setColumna(String.valueOf(n2left));

    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("SENTENCIA_ACCESO");

```

```

        RESULT.setValor(n1);
        RESULT.setFila(String.valueOf(n1right));
        RESULT.setColumna(String.valueOf(n1left));
        RESULT.getHijos().add(nodo_id);
    :}

|identificador:n1 s_cor_open EXPRESION:n2 s_cor_close
{:
    Nodo_AST_FS nodo_expresion = new Nodo_AST_FS();
    nodo_expresion.setEtiqueta("EXPRESION");
    nodo_expresion.setValor("");
    nodo_expresion.setFila(String.valueOf(n2right));
    nodo_expresion.setColumna(String.valueOf(n2left));
    nodo_expresion.getHijos().add(n2);

    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("SENTENCIA_ACCESO");
    RESULT.setValor(n1);
    RESULT.setFila(String.valueOf(n1right));
    RESULT.setColumna(String.valueOf(n1left));
    RESULT.getHijos().add(nodo_expresion);
:};

```

SENTENCIA_LLAMADA_FUNCIONES_ARREGLO ::= identificador:n1
LISTA_LLAMADAS_FUNCIONES_ARREGLO:n2

```

    {:
        RESULT = new Nodo_AST_FS();

RESULT.setEtiqueta("SENTENCIA_LLAMADA_FUNCIONES_ARREGLO");
        RESULT.setValor(n1);
        RESULT.setFila(String.valueOf(n1right));
        RESULT.setColumna(String.valueOf(n1left));
        RESULT.getHijos().add(n2);
    :};

```

```
LISTA_LLAMADAS_FUNCIONES_ARREGLO ::=
LISTA_LLAMADAS_FUNCIONES_ARREGLO:n1 s_dot FUNCION_ARREGLO:n2
```

```
{:
    RESULT = n1;
    RESULT.getHijos().add(n2);
:}
| s_dot FUNCION_ARREGLO:n
{:
    RESULT = new Nodo_AST_FS();
```

```
RESULT.setEtiqueta("LISTA_LLAMADAS_FUNCIONES_ARREGLO");
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(nright));
    RESULT.setColumna(String.valueOf(nleft));
    RESULT.getHijos().add(n);
:};
```

```
FUNCION_ARREGLO ::= FUNCION_DESCENDENTE:n
```

```
{:
    RESULT = n;
:}
|FUNCION_ASCENDENTE:n
{:
    RESULT = n;
:}
|FUNCION_INVERTIR:n
{:
    RESULT = n;
:}
|FUNCION_MAXIMO:n
{:
    RESULT = n;
```

:{

|FUNCION_MINIMO:n

{:

 RESULT = n;

:{

|FUNCION_FILTRAR:n

{:

 RESULT = n;

:{

|FUNCION_BUSCAR:n

{:

 RESULT = n;

:{

|FUNCION_MAP:n

{:

 RESULT = n;

:{

|FUNCION_REDUCE:n

{:

 RESULT = n;

:{

|FUNCION_TODOS:n

{:

 RESULT = n;

:{

|FUNCION_ALGUNO:n

{:

 RESULT = n;

:{

|FUNCION_OBTENER_POR_ETIQUETA:n

{:

 RESULT = n;

```

:}
|FUNCION_OBTENER_POR_ID:n
{:
    RESULT = n;
:}
|FUNCION_OBTENER_POR_NOMBRE:n
{:
    RESULT = n;
:};

```

FUNCION_DESCENDENTE ::= r_descendente:n s_par_open s_par_close

```

{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("FUNCION_DESCENDENTE");
    RESULT.setValor(n);
    RESULT.setFila(String.valueOf(nright));
    RESULT.setColumna(String.valueOf(nleft));
:};

```

FUNCION_ASCENDENTE ::= r_ascendente:n s_par_open s_par_close

```

{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("FUNCION_ASCENDENTE");
    RESULT.setValor(n);
    RESULT.setFila(String.valueOf(nright));
    RESULT.setColumna(String.valueOf(nleft));
:};

```

FUNCION_INVERTIR ::= r_invertir:n s_par_open s_par_close

```

{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("FUNCION_INVERTIR");

```

```

        RESULT.setValor(n);
        RESULT.setFila(String.valueOf(nright));
        RESULT.setColumna(String.valueOf(nleft));
    :};

```

FUNCION_MAXIMO ::= r_maximo:n s_par_open s_par_close

```

    {:
        RESULT = new Nodo_AST_FS();
        RESULT.setEtiqueta("FUNCION_MAXIMO");
        RESULT.setValor(n);
        RESULT.setFila(String.valueOf(nright));
        RESULT.setColumna(String.valueOf(nleft));
    :};

```

FUNCION_MINIMO ::= r_minimo:n s_par_open s_par_close

```

    {:
        RESULT = new Nodo_AST_FS();
        RESULT.setEtiqueta("FUNCION_MINIMO");
        RESULT.setValor(n);
        RESULT.setFila(String.valueOf(nright));
        RESULT.setColumna(String.valueOf(nleft));
    :};

```

FUNCION_FILTRAR ::= r_filtrar:n1 s_par_open identificador:n2 s_par_close

```

    {:
        RESULT = new Nodo_AST_FS();
        RESULT.setEtiqueta("FUNCION_FILTRAR");
        RESULT.setValor(n2);
        RESULT.setFila(String.valueOf(n1right));
        RESULT.setColumna(String.valueOf(n1left));
    :};

```

FUNCION_BUSCAR ::= r_buscar:n1 s_par_open identificador:n2 s_par_close

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta("FUNCION_BUSCAR");  
    RESULT.setValor(n2);  
    RESULT.setFila(String.valueOf(n1right));  
    RESULT.setColumna(String.valueOf(n1left));  
};
```

FUNCION_MAP ::= r_map:n1 s_par_open identificador:n2 s_par_close

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta("FUNCION_MAP");  
    RESULT.setValor(n2);  
    RESULT.setFila(String.valueOf(n1right));  
    RESULT.setColumna(String.valueOf(n1left));  
};
```

FUNCION_REDUCE ::= r_reduce:n1 s_par_open identificador:n2 s_par_close

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta("FUNCION_REDUCE");  
    RESULT.setValor(n2);  
    RESULT.setFila(String.valueOf(n1right));  
    RESULT.setColumna(String.valueOf(n1left));  
};
```

FUNCION_TODOS ::= r_todos:n1 s_par_open identificador:n2 s_par_close

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta("FUNCION_TODOS");  
    RESULT.setValor(n2);
```



```

        RESULT.setFila(String.valueOf(n1right));
        RESULT.setColumna(String.valueOf(n1left));
    };

```

```

FUNCION_ALGUNO ::=    r_alguno:n1 s_par_open identificador:n2 s_par_close
    {
        RESULT = new Nodo_AST_FS();
        RESULT.setEtiqueta("FUNCION_ALGUNO");
        RESULT.setValor(n2);
        RESULT.setFila(String.valueOf(n1right));
        RESULT.setColumna(String.valueOf(n1left));
    };

```

```

FUNCION_OBTENER_POR_ETIQUETA ::= r_obteneretiqueta:n1 s_par_open
cadena:n2 s_par_close
    {
        RESULT = new Nodo_AST_FS();

        RESULT.setEtiqueta("FUNCION_OBTENER_POR_ETIQUETA");
        RESULT.setValor(n2);
        RESULT.setFila(String.valueOf(n1right));
        RESULT.setColumna(String.valueOf(n1left));
    };

```

```

FUNCION_OBTENER_POR_ID ::= r_obtenerid:n1 s_par_open cadena:n2
s_par_close
    {
        RESULT = new Nodo_AST_FS();
        RESULT.setEtiqueta("FUNCION_OBTENER_POR_ID");
        RESULT.setValor(n2);
        RESULT.setFila(String.valueOf(n1right));
        RESULT.setColumna(String.valueOf(n1left));
    };

```

FUNCION_OBTENER_POR_NOMBRE ::= r_obtenernombre:n1 s_par_open
cadena:n2 s_par_close

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta("FUNCION_OBTENER_POR_NOMBRE");  
    RESULT.setValor(n2);  
    RESULT.setFila(String.valueOf(n1right));  
    RESULT.setColumna(String.valueOf(n1left));  
};
```

SENTENCIA_LEER_GXML ::= r_leergxml:n1 s_par_open cadena:n2 s_par_close

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta("SENTENCIA_LEER_GXML");  
    RESULT.setValor(n2);  
    RESULT.setFila(String.valueOf(n1right));  
    RESULT.setColumna(String.valueOf(n1left));  
};
```

SENTENCIA_CREAR_VENTANA ::= r_crearventana:n1 s_par_open cadena:n2
s_coma EXPRESION:n3 s_coma EXPRESION:n4 s_coma EXPRESION:n5
s_par_close

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta("SENTENCIA_CREAR_VENTANA");  
    RESULT.setValor(n2);  
    RESULT.setFila(String.valueOf(n1right));  
    RESULT.setColumna(String.valueOf(n1left));  
    RESULT.getHijos().add(n5);  
    RESULT.getHijos().add(n3);  
    RESULT.getHijos().add(n4);  
};
```

SENTENCIA_CREAR_CONTENEDOR ::= identificador:n1 s_dot r_crearcontenedor
s_par_open EXPRESION:n2 s_coma EXPRESION:n3 s_coma cadena:n4 s_coma
EXPRESION:n5 s_coma EXPRESION:n6 s_coma EXPRESION:n7 s_par_close

```
{:  
    Nodo_AST_FS nodo_color = new Nodo_AST_FS();  
    nodo_color.setEtiqueta("COLOR");  
    nodo_color.setValor(n4);  
    nodo_color.setFila(String.valueOf(n4right));  
    nodo_color.setColumna(String.valueOf(n4left));  
  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta("SENTENCIA_CREAR_CONTENEDOR");  
    RESULT.setValor(n1);  
    RESULT.setFila(String.valueOf(n1right));  
    RESULT.setColumna(String.valueOf(n1left));  
    RESULT.getHijos().add(nodo_color);  
    RESULT.getHijos().add(n2);  
    RESULT.getHijos().add(n3);  
    RESULT.getHijos().add(n5);  
    RESULT.getHijos().add(n6);  
    RESULT.getHijos().add(n7);  
};
```

SENTENCIA_CREAR_TEXTO ::= identificador:n1 s_dot r_creartexto s_par_open
EXPRESION:n2 s_coma EXPRESION:n3 s_coma cadena:n4 s_coma EXPRESION:n5
s_coma EXPRESION:n6 s_coma EXPRESION:n7 s_coma EXPRESION:n8 s_coma
EXPRESION:n9 s_par_close

```
{:  
    Nodo_AST_FS nodo_color = new Nodo_AST_FS();  
    nodo_color.setEtiqueta("COLOR");  
    nodo_color.setValor(n4);  
    nodo_color.setFila(String.valueOf(n4right));  
    nodo_color.setColumna(String.valueOf(n4left));
```

```

RESULT = new Nodo_AST_FS();
RESULT.setEtiqueta("SENTENCIA_CREAR_TEXTO");
RESULT.setValor(n1);
RESULT.setFila(String.valueOf(n1right));
RESULT.setColumna(String.valueOf(n1left));
RESULT.getHijos().add(nodo_color);
RESULT.getHijos().add(n2);
RESULT.getHijos().add(n3);
RESULT.getHijos().add(n5);
RESULT.getHijos().add(n6);
RESULT.getHijos().add(n7);
RESULT.getHijos().add(n8);
RESULT.getHijos().add(n9);
:};

```

SENTENCIA_CREAR_CAJA_TEXTO ::= identificador:n1 s_dot r_crearcajatexto
s_par_open EXPRESION:n2 s_coma EXPRESION:n3 s_coma EXPRESION:n4
s_coma EXPRESION:n5 s_coma cadena:n6 s_coma EXPRESION:n7 s_coma
EXPRESION:n8 s_coma EXPRESION:n9 s_coma EXPRESION:n10 s_coma
EXPRESION:n11 s_coma EXPRESION:n12 s_par_close

```

{:
    Nodo_AST_FS nodo_color = new Nodo_AST_FS();
    nodo_color.setEtiqueta("COLOR");
    nodo_color.setValor(n6);
    nodo_color.setFila(String.valueOf(n6right));
    nodo_color.setColumna(String.valueOf(n6left));

    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("SENTENCIA_CREAR_CAJA_TEXTO");
    RESULT.setValor(n1);
    RESULT.setFila(String.valueOf(n1right));
    RESULT.setColumna(String.valueOf(n1left));
    RESULT.getHijos().add(nodo_color);

```

```

        RESULT.getHijos().add(n2);
        RESULT.getHijos().add(n3);
        RESULT.getHijos().add(n4);
        RESULT.getHijos().add(n5);
        RESULT.getHijos().add(n7);
        RESULT.getHijos().add(n8);
        RESULT.getHijos().add(n9);
        RESULT.getHijos().add(n10);
        RESULT.getHijos().add(n11);
        RESULT.getHijos().add(n12);
    };

```

SENTENCIA_CREAR_AREA_TEXTO ::= identificador:n1 s_dot r_crearareatexto
s_par_open EXPRESION:n2 s_coma EXPRESION:n3 s_coma EXPRESION:n4
s_coma EXPRESION:n5 s_coma cadena:n6 s_coma EXPRESION:n7 s_coma
EXPRESION:n8 s_coma EXPRESION:n9 s_coma EXPRESION:n10 s_coma
EXPRESION:n11 s_coma EXPRESION:n12 s_par_close

```

{
    Nodo_AST_FS nodo_color = new Nodo_AST_FS();
    nodo_color.setEtiqueta("COLOR");
    nodo_color.setValor(n6);
    nodo_color.setFila(String.valueOf(n6right));
    nodo_color.setColumna(String.valueOf(n6left));

    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("SENTENCIA_CREAR_AREA_TEXTO");
    RESULT.setValor(n1);
    RESULT.setFila(String.valueOf(n1right));
    RESULT.setColumna(String.valueOf(n1left));
    RESULT.getHijos().add(nodo_color);
    RESULT.getHijos().add(n2);
    RESULT.getHijos().add(n3);
    RESULT.getHijos().add(n4);
    RESULT.getHijos().add(n5);
}

```

```

        RESULT.getHijos().add(n7);
        RESULT.getHijos().add(n8);
        RESULT.getHijos().add(n9);
        RESULT.getHijos().add(n10);
        RESULT.getHijos().add(n11);
        RESULT.getHijos().add(n12);
    };

```

SENTENCIA_CREAR_CONTROL_NUMERICO ::= identificador:n1 s_dot
 r_crearcontrolnumerico s_par_open EXPRESION:n2 s_coma EXPRESION:n3 s_coma
 EXPRESION:n4 s_coma EXPRESION:n5 s_coma EXPRESION:n6 s_coma
 EXPRESION:n7 s_coma EXPRESION:n8 s_coma EXPRESION:n9 s_par_close

```

    {
        RESULT = new Nodo_AST_FS();

```

```

    RESULT.setEtiqueta("SENTENCIA_CREAR_CONTROL_NUMERICO");
        RESULT.setValor(n1);
        RESULT.setFila(String.valueOf(n1 right));
        RESULT.setColumna(String.valueOf(n1 left));
        RESULT.getHijos().add(n2);
        RESULT.getHijos().add(n3);
        RESULT.getHijos().add(n4);
        RESULT.getHijos().add(n5);
        RESULT.getHijos().add(n6);
        RESULT.getHijos().add(n7);
        RESULT.getHijos().add(n8);
        RESULT.getHijos().add(n9);
    };

```

SENTENCIA_CREAR_DESPLEGABLE ::= identificador:n1 s_dot
 r_creardesplegable s_par_open EXPRESION:n2 s_coma EXPRESION:n3 s_coma
 EXPRESION:n4 s_coma EXPRESION:n5 s_coma EXPRESION:n6 s_coma
 EXPRESION:n7 s_coma EXPRESION:n8 s_par_close

```

    {

```

```

        RESULT = new Nodo_AST_FS();

RESULT.setEtiqueta("SENTENCIA_CREAR_DESPLEGABLE");

        RESULT.setValor(n1);
        RESULT.setFila(String.valueOf(n1right));
        RESULT.setColumna(String.valueOf(n1left));
        RESULT.getHijos().add(n2);
        RESULT.getHijos().add(n3);
        RESULT.getHijos().add(n4);
        RESULT.getHijos().add(n5);
        RESULT.getHijos().add(n6);
        RESULT.getHijos().add(n7);
        RESULT.getHijos().add(n8);

};

SENTENCIA_CREAR_BOTON ::= identificador:n1 s_dot r_crearboton s_par_open
EXPRESION:n2 s_coma EXPRESION:n3 s_coma cadena:n4 s_coma EXPRESION:n5
s_coma EXPRESION:n6 s_coma EXPRESION:n7 s_coma EXPRESION:n8 s_coma
EXPRESION:n9 s_coma EXPRESION:n10 s_par_close

{
    Nodo_AST_FS nodo_color = new Nodo_AST_FS();
    nodo_color.setEtiqueta("COLOR");
    nodo_color.setValor(n4);
    nodo_color.setFila(String.valueOf(n4right));
    nodo_color.setColumna(String.valueOf(n4left));
    RESULT = new Nodo_AST_FS();

    RESULT.setEtiqueta("SENTENCIA_CREAR_BOTON");
    RESULT.setValor(n1);
    RESULT.setFila(String.valueOf(n1right));
    RESULT.setColumna(String.valueOf(n1left));
    RESULT.getHijos().add(nodo_color);
    RESULT.getHijos().add(n2);
    RESULT.getHijos().add(n3);

```

```

        RESULT.getHijos().add(n5);
        RESULT.getHijos().add(n6);
        RESULT.getHijos().add(n7);
        RESULT.getHijos().add(n8);
        RESULT.getHijos().add(n9);
        RESULT.getHijos().add(n10);
    :}

|identificador:n1 s_dot r_crearboton s_par_open EXPRESION:n2
s_coma EXPRESION:n3 s_coma cadena:n4 s_coma EXPRESION:n5 s_coma
EXPRESION:n6 s_coma EXPRESION:n7 s_coma EXPRESION:n8 s_coma
EXPRESION:n9 s_par_close

{:
    Nodo_AST_FS nodo_color = new Nodo_AST_FS();
    nodo_color.setEtiqueta("COLOR");
    nodo_color.setValor(n4);
    nodo_color.setFila(String.valueOf(n4right));
    nodo_color.setColumna(String.valueOf(n4left));

    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("SENTENCIA_CREAR_BOTON");
    RESULT.setValor(n1);
    RESULT.setFila(String.valueOf(n1right));
    RESULT.setColumna(String.valueOf(n1left));
    RESULT.getHijos().add(nodo_color);
    RESULT.getHijos().add(n2);
    RESULT.getHijos().add(n3);
    RESULT.getHijos().add(n5);
    RESULT.getHijos().add(n6);
    RESULT.getHijos().add(n7);
    RESULT.getHijos().add(n8);
    RESULT.getHijos().add(n9);
:};

```


SENTENCIA_CREAR_IMAGEN ::= identificador:n1 s_dot r_crearimagen s_par_open
EXPRESION:n2 s_coma EXPRESION:n3 s_coma EXPRESION:n4 s_coma
EXPRESION:n5 s_coma EXPRESION:n6 s_par_close

```
{:  
    RESULT.setEtiqueta("SENTENCIA_CREAR_IMAGEN");  
    RESULT.setValor(n1);  
    RESULT.setFila(String.valueOf(n1 right));  
    RESULT.setColumna(String.valueOf(n1 left));  
    RESULT.getHijos().add(n2);  
    RESULT.getHijos().add(n3);  
    RESULT.getHijos().add(n4);  
    RESULT.getHijos().add(n5);  
    RESULT.getHijos().add(n6);  
};
```

SENTENCIA_CREAR_REPRODUCTOR ::= identificador:n1 s_dot
r_crearreproductor s_par_open EXPRESION:n2 s_coma EXPRESION:n3 s_coma
EXPRESION:n4 s_coma EXPRESION:n5 s_coma EXPRESION:n6 s_coma
EXPRESION:n7 s_par_close

```
{:  
  
    RESULT.setEtiqueta("SENTENCIA_CREAR_REPRODUCTOR");  
    RESULT.setValor(n1);  
    RESULT.setFila(String.valueOf(n1 right));  
    RESULT.setColumna(String.valueOf(n1 left));  
    RESULT.getHijos().add(n2);  
    RESULT.getHijos().add(n3);  
    RESULT.getHijos().add(n4);  
    RESULT.getHijos().add(n5);  
    RESULT.getHijos().add(n6);  
    RESULT.getHijos().add(n7);  
};
```

SENTENCIA_CREAR_VIDEO ::= identificador:n1 s_dot r_crearvideo s_par_open
EXPRESION:n2 s_coma EXPRESION:n3 s_coma EXPRESION:n4 s_coma
EXPRESION:n5 s_coma EXPRESION:n6 s_coma EXPRESION:n7 s_par_close

```
{:  
    RESULT.setEtiqueta("SENTENCIA_CREAR_VIDEO");  
    RESULT.setValor(n1);  
    RESULT.setFila(String.valueOf(n1right));  
    RESULT.setColumna(String.valueOf(n1left));  
    RESULT.getHijos().add(n2);  
    RESULT.getHijos().add(n3);  
    RESULT.getHijos().add(n4);  
    RESULT.getHijos().add(n5);  
    RESULT.getHijos().add(n6);  
    RESULT.getHijos().add(n7);  
:};
```

SENTENCIA_EVENTO_ALCLIC ::= identificador:n1 s_dot r_clic s_par_open
SENTENCIA_LLAMADA:n2 s_par_close

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta("SENTENCIA_EVENTO_ALCLIC");  
    RESULT.setValor(n1);  
    RESULT.setFila(String.valueOf(n1right));  
    RESULT.setColumna(String.valueOf(n1left));  
    RESULT.getHijos().add(n2);  
:};
```

SENTENCIA_EVENTO_ALCARGAR ::= identificador:n s_dot r_cargar s_par_open
s_par_close

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta("SENTENCIA_EVENTO_ALCARGAR");  
    RESULT.setValor(n);  
    RESULT.setFila(String.valueOf(nright));
```

```

        RESULT.setColumna(String.valueOf(nleft));
    :}
    | identificador:n1 s_dot r_cargar s_par_open
SENTENCIA_LLAMADA:n2 s_par_close
    {:
        RESULT = new Nodo_AST_FS();
        RESULT.setEtiqueta("SENTENCIA_EVENTO_ALCARGAR");
        RESULT.setValor(n1);
        RESULT.setFila(String.valueOf(n1right));
        RESULT.setColumna(String.valueOf(n1left));
        RESULT.getHijos().add(n2);
    :};

SENTENCIA_EVENTO_ALCERRAR ::= identificador:n s_dot r_cerrar s_par_open
s_par_close
    {:
        RESULT = new Nodo_AST_FS();
        RESULT.setEtiqueta("SENTENCIA_EVENTO_ALCERRAR");
        RESULT.setValor(n);
        RESULT.setFila(String.valueOf(nright));
        RESULT.setColumna(String.valueOf(nleft));
    :}
    | identificador:n1 s_dot r_cerrar s_par_open
SENTENCIA_LLAMADA:n2 s_par_close
    {:
        RESULT = new Nodo_AST_FS();
        RESULT.setEtiqueta("SENTENCIA_EVENTO_ALCERRAR");
        RESULT.setValor(n1);
        RESULT.setFila(String.valueOf(n1right));
        RESULT.setColumna(String.valueOf(n1left));
        RESULT.getHijos().add(n2);
    :};

```

```
SENTENCIA_CREAR_DESDE_ARCHIVO ::= identificador:n1 s_dot r_creararray  
s_par_open s_par_close
```

```
{:  
    RESULT = new Nodo_AST_FS();  
  
RESULT.setEtiqueta("SENTENCIA_CREAR_DESDE_ARCHIVO");  
    RESULT.setValor(n1);  
    RESULT.setFila(String.valueOf(n1right));  
    RESULT.setColumna(String.valueOf(n1left));  
:}  
|r_creararray:n1 s_par_open cadena:n2 s_par_close  
{:  
    Nodo_AST_FS nodo_archivo = new Nodo_AST_FS();  
    nodo_archivo.setEtiqueta("PATH");  
    nodo_archivo.setValor(n2);  
    nodo_archivo.setFila(String.valueOf(n2right));  
    nodo_archivo.setColumna(String.valueOf(n2left));  
  
    RESULT = new Nodo_AST_FS();  
  
RESULT.setEtiqueta("SENTENCIA_CREAR_DESDE_ARCHIVO");  
    RESULT.setValor("");  
    RESULT.setFila(String.valueOf(n1right));  
    RESULT.setColumna(String.valueOf(n1left));  
    RESULT.getHijos().add(nodo_archivo);  
:};
```

//PRODUCCIONES AUXILIARES

```
LISTA_PARAMETROS ::= LISTA_PARAMETROS:n1 s_coma r_var identificador:n2  
{:  
    Nodo_AST_FS nodo_identificador = new Nodo_AST_FS();  
    nodo_identificador.setEtiqueta("IDENTIFICADOR");
```

```

    nodo_identificador.setValor(n2);
    nodo_identificador.setFila(String.valueOf(n2right));
    nodo_identificador.setColumna(String.valueOf(n2left));

    n1.getHijos().add(nodo_identificador);
    RESULT = n1;
:}
|r_var identificador:n
{:
    Nodo_AST_FS nodo_identificador = new Nodo_AST_FS();
    nodo_identificador.setEtiqueta("IDENTIFICADOR");
    nodo_identificador.setValor(n);
    nodo_identificador.setFila(String.valueOf(nright));
    nodo_identificador.setColumna(String.valueOf(nleft));

    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("LISTA_PARAMETROS");
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(nright));
    RESULT.setColumna(String.valueOf(nleft));
    RESULT.getHijos().add(nodo_identificador);
:};

```

LISTA_IDENTIFICADORES ::= LISTA_IDENTIFICADORES:n1 s_coma
 identificador:n2

```

{:
    Nodo_AST_FS nodo_identificador = new Nodo_AST_FS();
    nodo_identificador.setEtiqueta("IDENTIFICADOR");
    nodo_identificador.setValor(n2);
    nodo_identificador.setFila(String.valueOf(n2right));
    nodo_identificador.setColumna(String.valueOf(n2left));

    n1.getHijos().add(nodo_identificador);

```

```

        RESULT = n1;
    :}
|identificador:n
{
    Nodo_AST_FS nodo_identificador = new Nodo_AST_FS();
    nodo_identificador.setEtiqueta("IDENTIFICADOR");
    nodo_identificador.setValor(n);
    nodo_identificador.setFila(String.valueOf(nright));
    nodo_identificador.setColumna(String.valueOf(nleft));

    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("LISTA_IDENTIFICADORES");
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(nright));
    RESULT.setColumna(String.valueOf(nleft));
    RESULT.getHijos().add(nodo_identificador);
};

```

LISTA_SENTENCIAS_SINO_SI ::= LISTA_SENTENCIAS_SINO_SI:n1
SENTENCIA_SINO_SI:n2

```

{
    RESULT = n1;
    RESULT.getHijos().add(n2);
}
|SENTENCIA_SINO_SI:n
{
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("LISTA_SENTENCIA_SINO_SI");
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(nright));
    RESULT.setColumna(String.valueOf(nleft));
    RESULT.getHijos().add(n);
};

```

LISTA_SENTENCIAS_CASOS ::= LISTA_SENTENCIAS_CASOS:n1
SENTENCIA_CASO:n2

```
{:  
    RESULT = n1;  
    RESULT.getHijos().add(n2);  
:}  
|SENTENCIA_CASO:n  
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta("LISTA_SENTENCIA_CASOS");  
    RESULT.setValor("");  
    RESULT.setFila(String.valueOf(nright));  
    RESULT.setColumna(String.valueOf(nleft));  
    RESULT.getHijos().add(n);  
:};
```

SENTENCIA_INCREMENTO ::= identificador:n1 s_increment

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta("SENTENCIA_INCREMENTO");  
    RESULT.setValor(n1);  
    RESULT.setFila(String.valueOf(n1right));  
    RESULT.setColumna(String.valueOf(n1left));  
:}  
|SENTENCIA_ACCESO:n1 s_increment  
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta("SENTENCIA_INCREMENTO");  
    RESULT.setValor("");  
    RESULT.setFila(String.valueOf(n1right));  
    RESULT.setColumna(String.valueOf(n1left));  
    RESULT.getHijos().add(n1);
```

```
};
```

```
SENTENCIA_DECREMENTO ::= identificador:n1 s_decrement
```

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta("SENTENCIA_DECREMENTO");  
    RESULT.setValor(n1);  
    RESULT.setFila(String.valueOf(n1 right));  
    RESULT.setColumna(String.valueOf(n1 left));  
};
```

```
|SENTENCIA_ACCESO:n1 s_decrement
```

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta("SENTENCIA_DECREMENTO");  
    RESULT.setValor("");  
    RESULT.setFila(String.valueOf(n1 right));  
    RESULT.setColumna(String.valueOf(n1 left));  
    RESULT.getHijos().add(n1);  
};
```

```
/******EXPRESIONES*****  
*****/
```

```
LISTA_EXPRESIONES ::= LISTA_EXPRESIONES:n1 s_coma EXPRESION:n2
```

```
{:  
    Nodo_AST_FS nodo_expresion = new Nodo_AST_FS();  
    nodo_expresion.setEtiqueta("EXPRESION");  
    nodo_expresion.setValor("");  
    nodo_expresion.setFila(String.valueOf(n2 right));  
    nodo_expresion.setColumna(String.valueOf(n2 left));  
    nodo_expresion.getHijos().add(n2);  
  
    RESULT = n1;
```



```

        RESULT.getHijos().add(nodo_expresion);
    :}
|EXPRESION:n
{:
    Nodo_AST_FS nodo_expresion = new Nodo_AST_FS();
    nodo_expresion.setEtiqueta("EXPRESION");
    nodo_expresion.setValor("");
    nodo_expresion.setFila(String.valueOf(nright));
    nodo_expresion.setColumna(String.valueOf(nleft));
    nodo_expresion.getHijos().add(n);

    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("LISTA_EXPRESIONES");
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(nright));
    RESULT.setColumna(String.valueOf(nleft));
    RESULT.getHijos().add(nodo_expresion);
:};

```

EXPRESION ::= EXPRESION_ARITMETICA:n

```

{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("EXPRESION_ARITMETICA");
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(nright));
    RESULT.setColumna(String.valueOf(nleft));
    RESULT.getHijos().add(n);
:}
|EXPRESION_RELACIONAL:n
{:
    RESULT = new Nodo_AST_FS();

```

```

    RESULT.setEtiqueta("EXPRESION_RELACIONAL");
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(nright));
    RESULT.setColumna(String.valueOf(nleft));
    RESULT.getHijos().add(n);
:}
|EXPRESION_UNARIA:n
{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("EXPRESION_UNARIA");
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(nright));
    RESULT.setColumna(String.valueOf(nleft));
    RESULT.getHijos().add(n);
:}
|EXPRESION_LOGICA:n
{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("EXPRESION_LOGICA");
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(nright));
    RESULT.setColumna(String.valueOf(nleft));
    RESULT.getHijos().add(n);
:}
|s_par_open EXPRESION:n s_par_close
{:
    RESULT = n;
:}
|DATO_PRIMITIVO:n
{:
    RESULT = n;
:}

```

|SENTENCIA_SELECCION:n

{:

 RESULT = n;

:.}

|SENTENCIA_LLAMADA:n

{:

 RESULT = n;

:.}

|SENTENCIA_ACCESO:n

{:

 RESULT = n;

:.}

|SENTENCIA_LLAMADA_FUNCIONES_ARREGLO:n

{:

 RESULT = n;

:.}

|SENTENCIA_LEER_GXML:n

{:

 RESULT = n;

:.}

|SENTENCIA_CREAR_VENTANA:n

{:

 RESULT = n;

:.}

|SENTENCIA_CREAR_CONTENEDOR:n

{:

 RESULT = n;

:.}

|SENTENCIA_CREAR_BOTON:n

{:

 RESULT = n;

:.};

EXPRESION_ARITMETICA ::= EXPRESION:n1 s_plus:o EXPRESION:n2

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta(o);  
    RESULT.setValor("");  
    RESULT.setFila(String.valueOf(n1 right));  
    RESULT.setColumna(String.valueOf(n1 left));  
    RESULT.getHijos().add(n1);  
    RESULT.getHijos().add(n2);  
:}
```

|EXPRESION:n1 s_minus:o EXPRESION:n2

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta(o);  
    RESULT.setValor("");  
    RESULT.setFila(String.valueOf(n1 right));  
    RESULT.setColumna(String.valueOf(n1 left));  
    RESULT.getHijos().add(n1);  
    RESULT.getHijos().add(n2);  
:}
```

|EXPRESION:n1 s_mul:o EXPRESION:n2

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta(o);  
    RESULT.setValor("");  
    RESULT.setFila(String.valueOf(n1 right));  
    RESULT.setColumna(String.valueOf(n1 left));  
    RESULT.getHijos().add(n1);  
    RESULT.getHijos().add(n2);  
:}
```

|EXPRESION:n1 s_div:o EXPRESION:n2

```

{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta(o);
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(n1 right));
    RESULT.setColumna(String.valueOf(n1 left));
    RESULT.getHijos().add(n1);
    RESULT.getHijos().add(n2);

```

```

:}

```

```

|EXPRESION:n1 s_mod:o EXPRESION:n2

```

```

{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta(o);
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(n1 right));
    RESULT.setColumna(String.valueOf(n1 left));
    RESULT.getHijos().add(n1);
    RESULT.getHijos().add(n2);

```

```

:}

```

```

|EXPRESION:n1 s_to:o EXPRESION:n2

```

```

{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta(o);
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(n1 right));
    RESULT.setColumna(String.valueOf(n1 left));
    RESULT.getHijos().add(n1);
    RESULT.getHijos().add(n2);

```

```

:}

```

```

|EXPRESION:n1 s_increment:o

```

```

{:
    RESULT = new Nodo_AST_FS();

```

```

    RESULT.setEtiqueta(o);
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(n1 right));
    RESULT.setColumna(String.valueOf(n1 left));
    RESULT.getHijos().add(n1);
  :}
|EXPRESION:n1 s_decrement:o
{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta(o);
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(n1 right));
    RESULT.setColumna(String.valueOf(n1 left));
    RESULT.getHijos().add(n1);
  :};

```

EXPRESION_UNARIA ::= s_minus:o EXPRESION:n1

```

{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta(o);
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(n1 right));
    RESULT.setColumna(String.valueOf(n1 left));
    RESULT.getHijos().add(n1);
  :}
|s_plus:o EXPRESION:n1
{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta(o);
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(n1 right));
    RESULT.setColumna(String.valueOf(n1 left));

```

```
        RESULT.getHijos().add(n1);  
    :};
```

EXPRESION_LOGICA ::= EXPRESION:n1 s_or:o EXPRESION:n2

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta(o);  
    RESULT.setValor("");  
    RESULT.setFila(String.valueOf(n1right));  
    RESULT.setColumna(String.valueOf(n1left));  
    RESULT.getHijos().add(n1);  
    RESULT.getHijos().add(n2);  
:}
```

|EXPRESION:n1 s_and:o EXPRESION:n2

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta(o);  
    RESULT.setValor("");  
    RESULT.setFila(String.valueOf(n1right));  
    RESULT.setColumna(String.valueOf(n1left));  
    RESULT.getHijos().add(n1);  
    RESULT.getHijos().add(n2);  
:}
```

|s_not:o EXPRESION:n1

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta(o);  
    RESULT.setValor("");  
    RESULT.setFila(String.valueOf(n1right));  
    RESULT.setColumna(String.valueOf(n1left));  
    RESULT.getHijos().add(n1);  
:};
```

EXPRESION_RELACIONAL ::= EXPRESION:n1 s_greather:o EXPRESION:n2

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta(o);  
    RESULT.setValor("");  
    RESULT.setFila(String.valueOf(n1 right));  
    RESULT.setColumna(String.valueOf(n1 left));  
    RESULT.getHijos().add(n1);  
    RESULT.getHijos().add(n2);  
:}
```

|EXPRESION:n1 s_less:o EXPRESION:n2

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta(o);  
    RESULT.setValor("");  
    RESULT.setFila(String.valueOf(n1 right));  
    RESULT.setColumna(String.valueOf(n1 left));  
    RESULT.getHijos().add(n1);  
    RESULT.getHijos().add(n2);  
:}
```

|EXPRESION:n1 s_greather_equal:o EXPRESION:n2

```
{:  
    RESULT = new Nodo_AST_FS();  
    RESULT.setEtiqueta(o);  
    RESULT.setValor("");  
    RESULT.setFila(String.valueOf(n1 right));  
    RESULT.setColumna(String.valueOf(n1 left));  
    RESULT.getHijos().add(n1);  
    RESULT.getHijos().add(n2);  
:}
```

|EXPRESION:n1 s_less_equal:o EXPRESION:n2


```

{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta(o);
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(n1 right));
    RESULT.setColumna(String.valueOf(n1 left));
    RESULT.getHijos().add(n1);
    RESULT.getHijos().add(n2);
:}
|EXPRESION:n1 s_compare:o EXPRESION:n2
{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta(o);
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(n1 right));
    RESULT.setColumna(String.valueOf(n1 left));
    RESULT.getHijos().add(n1);
    RESULT.getHijos().add(n2);
:}
|EXPRESION:n1 s_diferent:o EXPRESION:n2
{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta(o);
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(n1 right));
    RESULT.setColumna(String.valueOf(n1 left));
    RESULT.getHijos().add(n1);
    RESULT.getHijos().add(n2);
:};

```

DATO_PRIMITIVO ::= r_nulo:v

```

{:

```

```

        RESULT = new Nodo_AST_FS();

RESULT.setEtiqueta(Tabla_Enums.tipo_primitivo_Simbolo.nulo.toString());
        RESULT.setValor(v);
        RESULT.setFila(String.valueOf(vright));
        RESULT.setColumna(String.valueOf(vleft));
    :}
|booleano:v
{:
        RESULT = new Nodo_AST_FS();

RESULT.setEtiqueta(Tabla_Enums.tipo_primitivo_Simbolo.booleano.toString());
        RESULT.setValor(v);
        RESULT.setFila(String.valueOf(vright));
        RESULT.setColumna(String.valueOf(vleft));
    :}
|entero:v
{:
        RESULT = new Nodo_AST_FS();

RESULT.setEtiqueta(Tabla_Enums.tipo_primitivo_Simbolo.entero.toString());
        RESULT.setValor(v);
        RESULT.setFila(String.valueOf(vright));
        RESULT.setColumna(String.valueOf(vleft));
    :}
|decimal:v
{:
        RESULT = new Nodo_AST_FS();

RESULT.setEtiqueta(Tabla_Enums.tipo_primitivo_Simbolo.decimal.toString());
        RESULT.setValor(v);
        RESULT.setFila(String.valueOf(vright));
        RESULT.setColumna(String.valueOf(vleft));
    :}

```

```

|caracter:v
{
    RESULT = new Nodo_AST_FS();

RESULT.setEtiqueta(Tabla_Enums.tipo_primitivo_Simbolo.caracter.toString());
    RESULT.setValor(v);
    RESULT.setFila(String.valueOf(vright));
    RESULT.setColumna(String.valueOf(vleft));
}
|cadena:v
{
    RESULT = new Nodo_AST_FS();

RESULT.setEtiqueta(Tabla_Enums.tipo_primitivo_Simbolo.cadena.toString());
    RESULT.setValor(v);
    RESULT.setFila(String.valueOf(vright));
    RESULT.setColumna(String.valueOf(vleft));
}
|identificador:v
{
    RESULT = new Nodo_AST_FS();

RESULT.setEtiqueta(Tabla_Enums.tipo_primitivo_Simbolo.identificador.toString());
    RESULT.setValor(v);
    RESULT.setFila(String.valueOf(vright));
    RESULT.setColumna(String.valueOf(vleft));
};

ARREGLO ::= s_cor_open LISTA_EXPRESIONES:n s_cor_close
{
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("ARREGLO");
    RESULT.setValor("");

```

```

    RESULT.setFila(String.valueOf(nright));
    RESULT.setColumna(String.valueOf(nleft));
    RESULT.getHijos().add(n);
};

```

OBJETO ::= s_key_open LISTA_ATRIBUTOS:n s_key_close

```

{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("OBJETO");
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(nright));
    RESULT.setColumna(String.valueOf(nleft));
    RESULT.getHijos().add(n);
};

```

LISTA_ATRIBUTOS ::= LISTA_ATRIBUTOS:n1 s_coma ATRIBUTO:n2

```

{:
    RESULT = n1;
    RESULT.getHijos().add(n2);
:}
|ATRIBUTO:n
{:
    RESULT = new Nodo_AST_FS();
    RESULT.setEtiqueta("LISTA_ATRIBUTOS");
    RESULT.setValor("");
    RESULT.setFila(String.valueOf(nright));
    RESULT.setColumna(String.valueOf(nleft));
    RESULT.getHijos().add(n);
};

```

ATRIBUTO ::= identificador:n1 s_dobledot EXPRESION:n2

```

{:

```

```
Nodo_AST_FS nodo_expresion = new Nodo_AST_FS();  
nodo_expresion.setEtiqueta("EXPRESION");  
nodo_expresion.setValor("");  
nodo_expresion.setFila(String.valueOf(n2right));  
nodo_expresion.setColumna(String.valueOf(n2left));  
nodo_expresion.getHijos().add(n2);
```

```
RESULT = new Nodo_AST_FS();  
RESULT.setEtiqueta("ATRIBUTO");  
RESULT.setValor(n1);  
RESULT.setFila(String.valueOf(n1right));  
RESULT.setColumna(String.valueOf(n1left));  
RESULT.getHijos().add(nodo_expresion);
```

```
};
```