

Santiago Navas 1551619
Paolo Veliz 1543219
Alison Ramos 1510819
Jeffrey Reyes 1603719

1. Especificaciones léxicas del lenguaje

/* Palabras reservadas */

CLASE = "clase"
EXTIENDE = "extiende"
PROPIEDADES = "propiedades"
METODOS = "metodos"
PUBLICAS = "publicas"
PUBLICOS = "publicos"
PROTEGIDOS = "protegidos"
PROTEGIDAS = "protegidas"
PRIVADAS = "privadas"
PRIVADOS = "privados"
SI = "si"
DEVOLVER = "devolver"
CONSTRUCTOR = "constructor"
DESTRUCTOR = "destructor"
HACER = "hacer"
MIENTRAS = "mientras"
INCREMENTAR = "incrementar"
DESDE = "desde"
DECREMENTAR = "decrementar"
INSTANCIAR = "instanciar"
INCLUIR = "incluir"
SINO = "sino"
ENTONCES = "entonces"
PRINCIPAL = "Principal"
ESTATICO = "estatico"

/* Tipos de datos */

CADENA = "cadena"
BOLEANO = "booleano"
REAL = "real"
ENTERO = "entero"
NULO = "nulo"

/* Funciones */

CONSTRUCTOR = "constructor"

LEER = "leer"
ESCRIBIR = "escribir"
ELIMINAR = "eliminar"
CADENA_ENTERO = "cadenaAEntero"
CADENA_REAL = "cadenaAReal"
CADENA_BOLEANO =
"cadenaABoleano"
SENO = "seno"
COSENO = "coseno"
TANGENTE = "tangente"
LOGARITMO = "logaritmo"
RAIZ = "raiz"

/* Operadores */

AND = "AND"
OR = "OR"
MAYOR = ">"
MENOR = "<"
MENOROIGUAL = ">=" "
MAYOROIGUAL = "<=" "
DISTINTO = "!=" "
VERDADERO = "verdadero"
FALSO = "falso"
IGUALQUE = "==" "

/* Símbolos */

DOSPUNTOS = ":" "
PUNTO = "." "
PUNTOYCOMA = "," "
COMA = "," "
IGUAL = "=" "
INICIOCORCHETE = "[" "
FINALCORCHETE = "]" "
INICIO LLAVE = "{" "
FINAL LLAVE = "}" "
INICIOPARENTESIS = "(" "
FINALPARENTESIS = ")" "
GUION_BAJO = "_" "

/* Operadores básicos */

SUMA = "+"	{IGUAL}
RESTA = "-"	{INICIOCORCHETE}
DIVISION = "/" + ![/]	{FINALCORCHETE}
MULTIPLICACION = "**"	{INICIOLLAVE}
MODULO = "%"	{FINALLLAVE}
INCREMENTO = "++"	{INICIOPARENTESIS}
DECREMENTO = "--"	{FINALPARENTESIS}
	{GUION_BAJO}
	{SUMA}
/* Expresiones regulares */	{RESTA}
	{DIVISION}
Simbolos = ({DOSPUNTOS}	{MULTIPLICACION}
{PUNTO}	{MODULO}
{PUNTOYCOMA}	{MAYOR}
{COMA}	{MENOR})
FINLINEA = \r \n \r\n	
EspaciosBlancos = {FINLINEA} [\t]	
NUMEROS = [0-9][0-9]*	
LetrasMayus = [A-Z] "Á" "É" "Í" "Ó" "Ú" "Ñ"	
LetrasMinus = [a-z] "ñ" "á" "é" "í" "ó" "ú"	
Letras = {LetrasMinus} {LetrasMayus}	
VARIABLE = {LetrasMinus}({Letras}{NUMEROS}{GUION_BAJO})*	
VARIABLECLASE = {LetrasMayus}({Letras}{NUMEROS}{GUION_BAJO})*	
Texto = ({Letras}{NUMEROS})[\t]({Simbolos})+	
CadenaString = "\"" + {Texto} + "\""	
/* Errores */	
ERVARIABLE = {NUMEROS}({Letras}{NUMEROS}{GUION_BAJO})+	
/* No tokens */	
Comentario = "/" {Texto}* {FINLINEA}?	
ComentarioLargo = "/"* [^*] ~"/" "/"* "*" + "/"	

2. Producciones

```
package com.mycompany.jflex.test;
```

```
terminal CLASE;
terminal EXTIENDE;
terminal PROPIEDADES;
terminal METODOS;
terminal PUBLICAS;
terminal PUBLICOS;
terminal PROTEGIDOS;
terminal PROTEGIDAS;
terminal PRIVADAS;
```

terminal PRIVADOS;
terminal SI;
terminal DEVOLVER;
terminal CONSTRUCTOR;
terminal DESTRUCTOR;
terminal HACER;
terminal MIENTRAS;
terminal INCREMENTAR;
terminal DECREMENTAR;
terminal DESDE;
terminal INSTANCIAR;
terminal INCLUIR;
terminal SINO;
terminal ENTONCES;
terminal PRINCIPAL;
terminal ESTATICO;
terminal DOSPUNTOS;
terminal PUNTO;
terminal PUNTOYCOMA;
terminal COMA;
terminal IGUAL;
terminal INICIOCORCHETE;
terminal FINALCORCHETE;
terminal INICIOLLAVE;
terminal FINALLLAVE;
terminal INICIOPARENTESIS;
terminal FINALPARENTESIS;
terminal AND;
terminal OR;
terminal MAYOR;
terminal MENOR;
terminal MENOROIGUAL;
terminal MAYOROIGUAL;
terminal DISTINTO;
terminal VERDADERO;
terminal FALSO;
terminal IGUALQUE;
terminal CADENA;
terminal BOLEANO;
terminal REAL;
terminal ENTERO;
terminal NULO;
terminal LEER;
terminal ESCRIBIR;
terminal ELIMINAR;

terminal CADENA_ENTERO;
terminal CADENA_REAL;

```

terminal CADENA_BOLEANO;
terminal SENO;
terminal COSENO;
terminal TANGENTE;
terminal LOGARITMO;
terminal RAIZ;
terminal SUMA;
terminal RESTA;
terminal DIVISION;
terminal MULTIPLICACION;
terminal MODULO;
terminal INCREMENTO;
terminal DECREMENTO;
terminal ID;
terminal IDCLASE;
terminal NUMEROS;
terminal FINLINEA;

```

```

terminal STRING;

```

```

non terminal S, Asignar, LineasCodigo, Identificadores, Parametros, TipoDato, NumReal;
non terminal Operadores, Operacion, OpcionesEscribir, OpcionesPrivados,
OpcionesPublicos, OpcionOperacion;
non terminal FuncionesSueltas, FuncionesEspeciales, FuncionesMatematicas,
FuncionMatematica, FuncionLeer, FuncionEscribir;
non terminal Declaraciones, BibliotecaExterna;
non terminal Clases, Constructor, Destructor, AsignarConstructor, MetodoDestructor,
MetodosProtegidos, MetodosPrivados, MetodosPublicos, Metodos, Repetir;
non terminal PropPrivadas, PropProtegidas, PropPublicas;
non terminal Instancia, MetodoInstancia, ErrorInstancia, OpcionesProtegidos, LineaCodigo,
FuncionPrincipal, ContenidoPrincipal;
S ::= LineasCodigo;

```

```

NumReal ::= NUMEROS PUNTO NUMEROS;

```

```

TipoDato ::= ENTERO                                     {:
System.out.println("SIM: TipoDato -> ENTERO "); :}|
    CADENA                                             {:
System.out.println("SIM: TipoDato -> CADENA "); :}|
    BOLEANO                                           {:
System.out.println("SIM: TipoDato -> BOLEANO "); :}|
    NULO                                              {:
System.out.println("SIM: TipoDato -> NULO "); :}|
    REAL                                              {: System.out.println("SIM:
TipoDato -> REAL "); :};

```

```

Operadores ::= SUMA                                     {:
System.out.println("SIM: Operadores -> SUMA "); :}|

```

```

        RESTA                                                                    {:
System.out.println("SIM: Operadores -> RESTA "); :}|
        MULTIPLICACION                                                            {:
System.out.println("SIM: Operadores -> MULTIPLICACION "); :}|
        DIVISION                                                                    {:
System.out.println("SIM: Operadores -> DIVISION "); :}|
        MODULO                                                                      {:
System.out.println("SIM: Operadores -> MODULO "); :};

Operacion ::= ID Operadores Operacion                                           {:
System.out.println("SIM: Operacion -> ID Operadores Operacion "); :}|
        ID                                                                        {: System.out.println("SIM:
Operacion -> ID "); :}|
        NUMEROS Operadores Operacion                                             {:
System.out.println("SIM: Operacion -> NUMEROS Operadores Operacion "); :}|
        NUMEROS                                                                    {:
System.out.println("SIM: Operacion -> NUMEROS "); :};

Identificadores ::= ID                                                            {:
System.out.println("SIM: Identificadores -> ID "); :}|
        ID COMA Identificadores                                                  {:
System.out.println("SIM: Identificadores -> ID COMA Identificadores "); :};

Asignar ::= ENTERO Identificadores IGUAL NUMEROS PUNTOYCOMA
{: System.out.println("SIM: Asignar -> ENTERO Identificadores IGUAL NUMEROS
PUNTOYCOMApoDato "); :}|
        CADENA Identificadores IGUAL STRING PUNTOYCOMA
{: System.out.println("SIM: Asignar -> CADENA Identificadores IGUAL STRING
PUNTOYCOMA "); :}|
        BOLEANO Identificadores IGUAL FALSO PUNTOYCOMA
{: System.out.println("SIM: Asignar -> BOLEANO Identificadores IGUAL FALSO
PUNTOYCOMA "); :}|
        BOLEANO Identificadores IGUAL VERDADERO PUNTOYCOMA
{: System.out.println("SIM: Asignar -> BOLEANO Identificadores IGUAL VERDADERO
PUNTOYCOMA "); :}|
        NULO Identificadores PUNTOYCOMA                                         {:
System.out.println("SIM: Asignar -> NULO Identificadores PUNTOYCOMA "); :}|
        REAL Identificadores NumReal                                             {:
System.out.println("SIM: Asignar -> REAL Identificadores NumReal "); :}|
        ENTERO Identificadores IGUAL NUMEROS PUNTOYCOMA FINLINEA
{: System.out.println("SIM: Asignar -> ENTERO Identificadores IGUAL NUMEROS
PUNTOYCOMA FINLINEA "); :}|
        CADENA Identificadores IGUAL STRING PUNTOYCOMA FINLINEA
{: System.out.println("SIM: Asignar -> CADENA Identificadores IGUAL STRING
PUNTOYCOMA FINLINEA "); :}|
        BOLEANO Identificadores IGUAL FALSO PUNTOYCOMA FINLINEA
{: System.out.println("SIM: Asignar -> BOLEANO Identificadores IGUAL FALSO
PUNTOYCOMA FINLINEA "); :}|

```

```

        BOLEANO Identificadores IGUAL VERDADERO PUNTOYCOMA FINLINEA
{: System.out.println("SIM: Asignar -> BOLEANO Identificadores IGUAL VERDADERO
PUNTOYCOMA FINLINEA "); :}|
        NULO Identificadores PUNTOYCOMA FINLINEA                                {:
System.out.println("SIM: Asignar -> NULO Identificadores PUNTOYCOMA FINLINEA "); :}|
        REAL Identificadores NumReal FINLINEA                                {:
System.out.println("SIM: Asignar -> REAL Identificadores NumReal FINLINEA "); :};

Parametros ::= TipoDato ID                                                    {:
System.out.println("SIM: Parametros -> TipoDato ID "); :}|
        TipoDato ID COMA Parametros                                           {:
System.out.println("SIM: Parametros -> TipoDato ID COMA Parametros "); :};

FuncionesSueltas ::= ENTERO ID INICIOPARENTESIS Parametros FINALPARENTESIS
FINLINEA DEVOLVER Operacion FINLINEA {: System.out.println("SIM: FuncionesSueltas
-> ENTERO ID INICIOPARENTESIS Parametros FINALPARENTESIS FINLINEA
DEVOLVER Operacion FINLINEA "); :}|
        ENTERO ID INICIOPARENTESIS FINALPARENTESIS FINLINEA DEVOLVER
Operacion FINLINEA      {: System.out.println("SIM: FuncionesSueltas -> ENTERO ID
INICIOPARENTESIS FINALPARENTESIS FINLINEA DEVOLVER Operacion FINLINEA ");
:}|
        REAL ID INICIOPARENTESIS Parametros FINALPARENTESIS FINLINEA
DEVOLVER Operacion FINLINEA  {: System.out.println("SIM: FuncionesSueltas -> REAL
ID INICIOPARENTESIS Parametros FINALPARENTESIS FINLINEA DEVOLVER Operacion
FINLINEA "); :}|
        REAL ID INICIOPARENTESIS FINALPARENTESIS FINLINEA DEVOLVER
Operacion FINLINEA      {: System.out.println("SIM: FuncionesSueltas -> REAL ID
INICIOPARENTESIS FINALPARENTESIS FINLINEA DEVOLVER Operacion FINLINEA ");
:}|
        BOLEANO ID INICIOPARENTESIS Parametros FINALPARENTESIS
FINLINEA DEVOLVER Operacion FINLINEA {: System.out.println("SIM: FuncionesSueltas
-> BOLEANO ID INICIOPARENTESIS Parametros FINALPARENTESIS FINLINEA
DEVOLVER Operacion FINLINEA "); :}|
        BOLEANO ID INICIOPARENTESIS FINALPARENTESIS FINLINEA
DEVOLVER Operacion FINLINEA      {: System.out.println("SIM: FuncionesSueltas ->
BOLEANO ID INICIOPARENTESIS FINALPARENTESIS FINLINEA DEVOLVER Operacion
FINLINEA "); :}|
        NULO ID INICIOPARENTESIS Parametros FINALPARENTESIS FINLINEA
DEVOLVER Operacion FINLINEA  {: System.out.println("SIM: FuncionesSueltas -> NULO
ID INICIOPARENTESIS Parametros FINALPARENTESIS FINLINEA DEVOLVER Operacion
FINLINEA "); :}|
        NULO ID INICIOPARENTESIS FINALPARENTESIS FINLINEA DEVOLVER
Operacion FINLINEA      {: System.out.println("SIM: FuncionesSueltas -> NULO ID
INICIOPARENTESIS FINALPARENTESIS FINLINEA DEVOLVER Operacion FINLINEA ");
:}|
        CADENA ID INICIOPARENTESIS Parametros FINALPARENTESIS FINLINEA
DEVOLVER Operacion FINLINEA  {: System.out.println("SIM: FuncionesSueltas ->

```

```

CADENA ID INICIOPARENTESIS Parametros FINALPARENTESIS FINLINEA DEVOLVER
Operacion FINLINEA "); :}|
    CADENA ID INICIOPARENTESIS FINALPARENTESIS FINLINEA DEVOLVER
Operacion FINLINEA      {: System.out.println("SIM: FuncionesSueltas -> CADENA ID
INICIOPARENTESIS FINALPARENTESIS FINLINEA DEVOLVER Operacion FINLINEA ");
:};

FuncionesEspeciales ::= REAL CADENA_REAL INICIOPARENTESIS CADENA ID
FINALPARENTESIS FINLINEA      {: System.out.println("SIM: FuncionesEspeciales ->
REAL CADENA_REAL INICIOPARENTESIS CADENA ID FINALPARENTESIS FINLINEA ");
:}|
    ENTERO CADENA_ENTERO INICIOPARENTESIS CADENA ID
FINALPARENTESIS FINLINEA      {: System.out.println("SIM: FuncionesEspeciales ->
ENTERO CADENA_ENTERO INICIOPARENTESIS CADENA ID FINALPARENTESIS
FINLINEA "); :}|
    BOLEANO CADENA_BOLEANO INICIOPARENTESIS CADENA ID
FINALPARENTESIS FINLINEA      {: System.out.println("SIM: FuncionesEspeciales ->
BOLEANO CADENA_BOLEANO INICIOPARENTESIS CADENA ID FINALPARENTESIS
FINLINEA "); :};

OpcionesEscribir ::= ID                                     {:
System.out.println("SIM: OpcionesEscribir -> ID "); :}|
    NUMEROS                                               {: System.out.println("SIM:
OpcionesEscribir -> NUMEROS "); :}|
    STRING                                               {: System.out.println("SIM:
OpcionesEscribir -> STRING "); :};

FuncionEscribir ::= ESCRIBIR OpcionesEscribir PUNTOYCOMA
{: System.out.println("SIM: FuncionEscribir -> OpcionesEscribir PUNTOYCOMA "); :};

FuncionLeer ::= LEER ID PUNTOYCOMA                        {:
System.out.println("SIM: FuncionLeer -> LEER ID PUNTOYCOMA "); :};

FuncionesMatematicas ::= REAL SENO INICIOPARENTESIS REAL ID FINALPARENTESIS
FINLINEA      {: System.out.println("SIM: FuncionesMatematicas -> REAL SENO
INICIOPARENTESIS REAL ID FINALPARENTESIS FINLINEA "); :}|
    REAL COSENO INICIOPARENTESIS REAL ID FINALPARENTESIS
FINLINEA      {: System.out.println("SIM: FuncionesMatematicas -> REAL COSENO
INICIOPARENTESIS REAL ID FINALPARENTESIS FINLINEA "); :}|
    REAL TANGENTE INICIOPARENTESIS REAL ID FINALPARENTESIS
FINLINEA      {: System.out.println("SIM: FuncionesMatematicas -> REAL TANGENTE
INICIOPARENTESIS REAL ID FINALPARENTESIS FINLINEA "); :}|
    REAL LOGARITMO INICIOPARENTESIS REAL ID FINALPARENTESIS
FINLINEA      {: System.out.println("SIM: FuncionesMatematicas -> REAL LOGARITMO
INICIOPARENTESIS REAL ID FINALPARENTESIS FINLINEA "); :}|
    REAL RAIZ INICIOPARENTESIS REAL ID FINALPARENTESIS FINLINEA
{: System.out.println("SIM: FuncionesMatematicas -> REAL RAIZ INICIOPARENTESIS
REAL ID FINALPARENTESIS FINLINEA "); :};

```

/* CLASES */

```
Declaraciones ::= TipoDato Identificadores FINLINEA Declaraciones      {:  
System.out.println("SIM: Declaraciones -> TipoDato Identificadores FINLINEA  
Declaraciones"); :};
```

```
        TipoDato Identificadores FINLINEA      {:  
System.out.println("SIM: Declaraciones -> TipoDato Identificadores FINLINEA"); :};
```

```
PropPublicas ::= PROPIEDADES PUBLICAS DOSPUNTOS FINLINEA Declaraciones  
PropPrivadas   {: System.out.println("SIM: PropPublicas -> PROPIEDADES PUBLICAS  
DOSPUNTOS FINLINEA Declaraciones PropPrivadas"); :};
```

```
        PropPrivadas                               {: System.out.println("SIM:  
PropPublicas -> PropPrivadas"); :};
```

```
PropPrivadas ::= PROPIEDADES PRIVADAS DOSPUNTOS FINLINEA Declaraciones  
PropProtegidas   {: System.out.println("SIM: PropPrivadas -> PROPIEDADES PRIVADAS  
DOSPUNTOS FINLINEA Declaraciones PropProtegidas"); :};
```

```
        PropProtegidas                               {: System.out.println("SIM:  
PropPrivadas -> PropProtegidas"); :};
```

```
PropProtegidas ::= PROPIEDADES PROTEGIDAS DOSPUNTOS FINLINEA Declaraciones  
{: System.out.println("SIM: PropProtegidas -> PROPIEDADES PROTEGIDAS DOSPUNTOS  
FINLINEA Declaraciones FINLINEA"); :};
```

```
        FINLINEA                                     {: System.out.println("SIM:  
PropProtegidas -> FINLINEA"); :};
```

```
MetodosPublicos ::= METODOS PUBLICOS DOSPUNTOS FINLINEA OpcionesPublicos  
MetodosPrivados   {: System.out.println("SIM: MetodosPublicos -> METODOS PUBLICOS  
DOSPUNTOS FINLINEA OpcionesPublicos MetodosPrivados"); :};
```

```
        MetodosPrivados                               {: System.out.println("SIM:  
MetodosPublicos -> MetodosPrivados"); :};
```

```
MetodosPrivados ::= METODOS PRIVADOS DOSPUNTOS FINLINEA OpcionesPrivados  
MetodosProtegidos   {: System.out.println("SIM: MetodosPrivados -> METODOS PRIVADOS  
DOSPUNTOS FINLINEA OpcionesPrivados MetodosProtegidos"); :};
```

```
        MetodosProtegidos                               {:  
System.out.println("SIM: MetodosPrivados -> MetodosProtegidos"); :};
```

```
MetodosProtegidos ::= METODOS PROTEGIDOS DOSPUNTOS FINLINEA  
OpcionesProtegidos   {: System.out.println("SIM: MetodosProtegidos -> METODOS  
PROTEGIDOS DOSPUNTOS FINLINEA OpcionesProtegidos FINLINEA "); :};
```

```
        FINLINEA                                     {: System.out.println("SIM:  
MetodosProtegidos -> FINLINEA"); :};
```

```
OpcionesPublicos ::= FuncionesSueltas OpcionesPublicos      {:  
System.out.println("SIM: OpcionesPublicos -> FuncionesSueltas OpcionesPublicos"); :};
```

```
        FuncionesSueltas                               {: System.out.println("SIM:  
OpcionesPublicos -> FuncionesSueltas"); :};
```



```

        Constructor OpcionesPublicos                                {:
System.out.println("SIM: OpcionesPublicos -> Constructor OpcionesPublicos"); :}]
        Constructor                                                {: System.out.println("SIM:
OpcionesPublicos -> Constructor"); :}]
        Destructor                                                  {: System.out.println("SIM:
OpcionesPublicos -> Destructor"); :};

OpcionesPrivados ::= FuncionesSueltas OpcionesPrivados            {:
System.out.println("SIM: OpcionesPrivados -> FuncionesSueltas OpcionesPrivados"); :}]
        FuncionesSueltas                                           {: System.out.println("SIM:
OpcionesPrivados -> FuncionesSueltas"); :}]
        Constructor OpcionesPrivados                                {:
System.out.println("SIM: OpcionesPrivados -> Constructor OpcionesPublicos"); :}]
        Constructor                                                {: System.out.println("SIM:
OpcionesPrivados -> Constructor"); :};

OpcionesProtegidos ::= FuncionesSueltas                            {:
System.out.println("SIM: OpcionesProtegidos -> FuncionesSueltas"); :};

Clases ::= CLASE IDCLASE FINLINEA PropPublicas MetodosPublicos    {:
System.out.println("SIM: Clases -> CLASE IDCLASE FINLINEA PropPublicas
MetodosPublicos"); :}]
        CLASE IDCLASE EXTIENDE IDCLASE FINLINEA PropPublicas MetodosPublicos
{: System.out.println("SIM: Clases -> CLASE IDCLASE EXTIENDE IDCLASE FINLINEA
PropPublicas MetodosPublicos"); :};

/* CONSTRUCTOR & DESTRUCTOR */
Constructor ::= CONSTRUCTOR INICIOPARENTESIS Parametros FINALPARENTESIS
FINLINEA AsignarConstructor{: System.out.println("SIM: Constructor -> CONSTRUCTOR
INICIOPARENTESIS Parametros FINALPARENTESIS FINLINEA AsignarConstructor"); :}]
        CONSTRUCTOR INICIOPARENTESIS FINALPARENTESIS FINLINEA
{: System.out.println("SIM: Constructor -> CONSTRUCTOR INICIOPARENTESIS
FINALPARENTESIS FINLINEA"); :};

Destructor ::= DESTRUCTOR INICIOPARENTESIS FINALPARENTESIS FINLINEA
MetodoDestructor          {: System.out.println("SIM: Destructor -> DESTRUCTOR
INICIOPARENTESIS FINALPARENTESIS FINLINEA MetodoDestructor"); :};

MetodoDestructor ::= ELIMINAR ID FINLINEA MetodoDestructor        {:
System.out.println("SIM: MetodoDestructor -> ELIMINAR ID FINLINEA MetodoDestructor");
:}]
        ELIMINAR ID FINLINEA                                       {:
System.out.println("SIM: MetodoDestructor -> ELIMINAR ID FINLINEA"); :};

AsignarConstructor ::= ID IGUAL ID FINLINEA AsignarConstructor     {:
System.out.println("SIM: AsignarConstructor -> ID IGUAL ID FINLINEA
AsignarConstructor"); :}]

```

```

ID IGUAL ID FINLINEA
System.out.println("SIM: AsignarConstructor -> ID IGUAL ID FINLINEA"); :};

/* INSTANCIACION */
Instancia ::= IDCLASE ID IGUAL INSTANCIAR IDCLASE INICIOPARENTESIS
FINALPARENTESIS FINLINEA {; System.out.println("SIM: Instancia -> IDCLASE ID IGUAL
INSTANCIAR IDCLASE INICIOPARENTESIS FINALPARENTESIS FINLINEA
MetodoInstancia"); :};

MetodoInstancia ::= ESCRIBIR ID PUNTO ID FINLINEA
System.out.println("SIM: MetodoInstancia -> ESCRIBIR ID PUNTO ID FINLINEA"); :}|
ELIMINAR IDCLASE FINLINEA
System.out.println("SIM: MetodoInstancia -> ELIMINAR IDCLASE FINLINEA"); :};

ErrorInstancia ::= ELIMINAR IDCLASE FINLINEA ESCRIBIR ID PUNTO ID FINLINEA
{; System.out.println("SIM: ErrorInstancia -> ELIMINAR IDCLASE FINLINEA ESCRIBIR ID
PUNTO ID FINLINEA"); :};

/* CONDICIONAL */
/*
FuncionSi ::= SI Condicionante ENTONCES FINLINEA Respuesta;

FuncionSino ::= SINO Condicionante ENTONCES FINLINEA Respuesta FincionSino|
SINO FINLINEA Respuesta;

FuncionCompuesta ::= FuncionSi FuncionSino|
FincionSi;

Respuesta ::= DEVOLVER OpcionOperacion|
ESCRIBIR STRING;

Condicionante ::= ID Condicion NUMEROS|
NUMEROS Condicion ID|
ID Ccondicion ID|
NUMEROS Condicion NUMEROS;

Condicion ::= MAYOR|
MENOR|
MENOROIGUAL|
MAYOROIGUAL|
DISTINTO|
IGUALQUE;
*/

LineasCodigo ::= LineaCodigo LineasCodigo {; System.out.println("SIM: LineasCodigo ->
LineasCodigo LineaCodigo"); :}|
LineaCodigo {; System.out.println("SIM: LineasCodigo -> LineaCodigo");
:};

```

```

LineaCodigo ::= BibliotecaExterna          {: System.out.println("SIM: LineaCodigo ->
Bibliotecas"); :}
                Clases                      {: System.out.println("SIM: LineaCodigo -> Clases"); :}
                FuncionPrincipal             {: System.out.println("SIM: LineaCodigo ->
FuncionPrincipal"); :};

FuncionPrincipal ::= PRINCIPAL INICIOPARENTESIS FINALPARENTESIS FINLINEA
Repetir  {: System.out.println("SIM: FuncionPrincipal -> ENTERO PRINCIPAL
INICIOPARENTESIS Parametros FINALPARENTESIS FINLINEA ContenidoPrincipal"); :};

Repetir ::= ContenidoPrincipal Repetir  {: System.out.println("SIM: Repetir ->
ContenidoPrincipal Repetir"); :}
                ContenidoPrincipal          {: System.out.println("SIM: Repetir -> ContenidoPrincipal");
:};

ContenidoPrincipal ::= FuncionesSueltas                                     {:
System.out.println("SIM: ContenidoPrincipal -> FuncionesSueltas"); :}|
                Instancia                                                    {: System.out.println("SIM:
ContenidoPrincipal -> Instancia"); :}|
                FuncionesMatematicas                                         {:
System.out.println("sim: LineaCodigo -> FuncionesMatematicas"); :}|
                FuncionesEspeciales                                         {:
System.out.println("sim: LineaCodigo -> FuncionesEspeciales"); :}|
                FuncionEscribir                                              {: System.out.println("SIM:
ContenidoPrincipal -> FuncionEscribir"); :}|
                FuncionLeer                                                  {: System.out.println("SIM:
ContenidoPrincipal -> FuncionLeer"); :}|
                Asignar;

// LineasCodigo ::= LineaCodigo LineasCodigo|
//          LineaCodigo;

// LineaCodigo ::= Asignar          {: System.out.println("sim: LineaCodigo ->
Asignar"); :}|
//          FuncionesSueltas        {: System.out.println("sim: LineaCodigo ->
FuncionesSueltas"); :}|
//          Clases                  {: System.out.println("sim: LineaCodigo -> Clases"); :}|
//          FuncionesMatematicas    {: System.out.println("sim: LineaCodigo ->
FuncionesMatematicas"); :}|
//          FuncionesEspeciales     {: System.out.println("sim: LineaCodigo ->
FuncionesEspeciales"); :}|
//          FuncionEscribir         {: System.out.println("sim: LineaCodigo ->
FuncionEscribir"); :}|
//          FuncionLeer             {: System.out.println("sim: LineaCodigo ->
FuncionLeer"); :}|
//          Instancia               {: System.out.println("sim: LineaCodigo -> Instancia"); :}|

```

```
//      BibliotecaExterna      ;
```

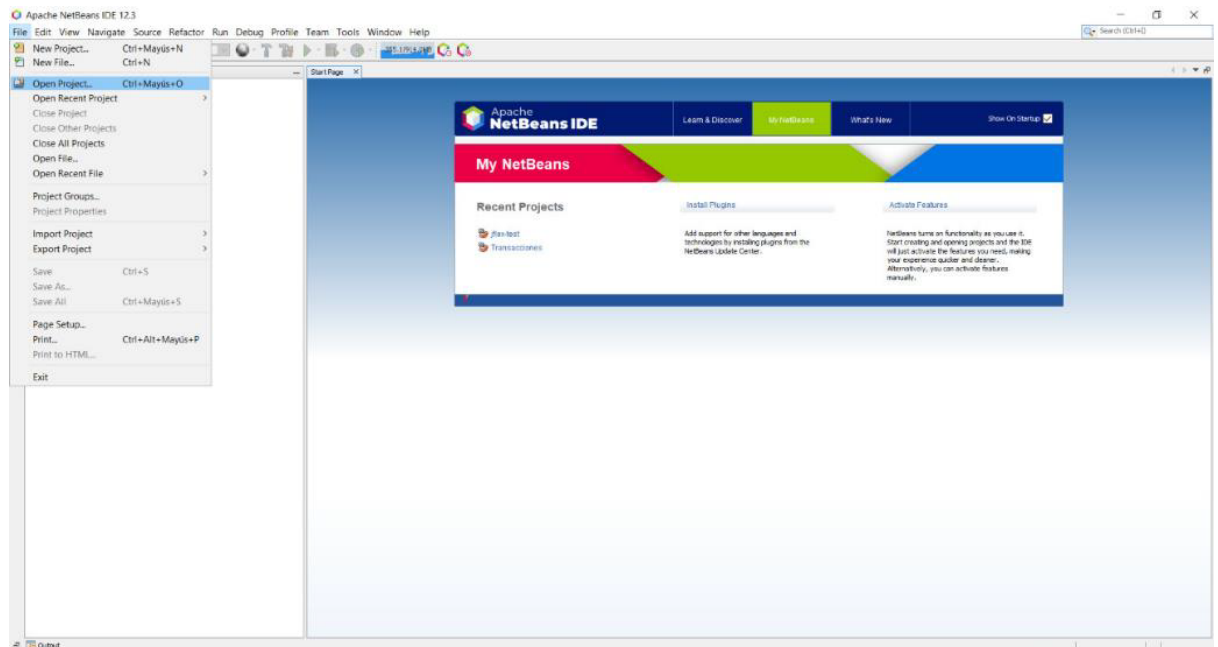
```
/* BIBLIOTECAS */
```

```
BibliotecaExterna ::= INCLUIR STRING FINLINEA { : System.out.println("SIM:
```

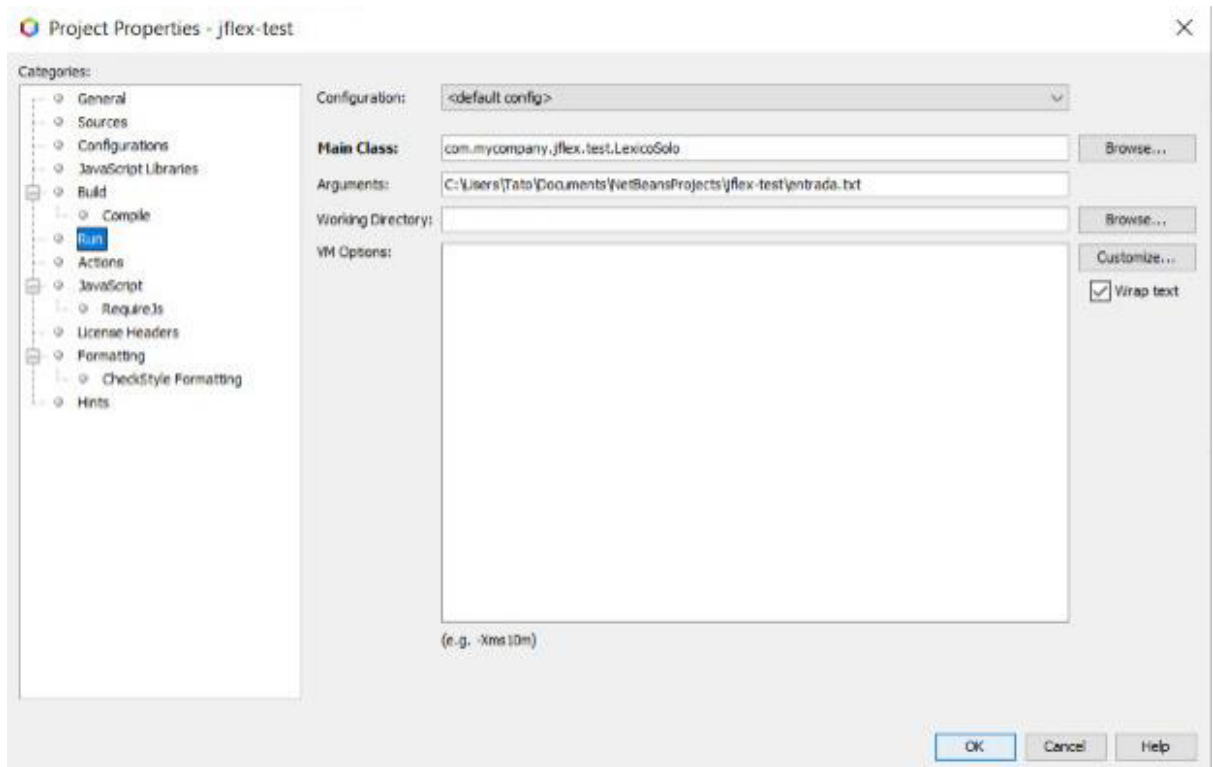
```
BibliotecaExterna -> INCLUIR STRING FINLINEA"); :};
```

3. Manual de usuario

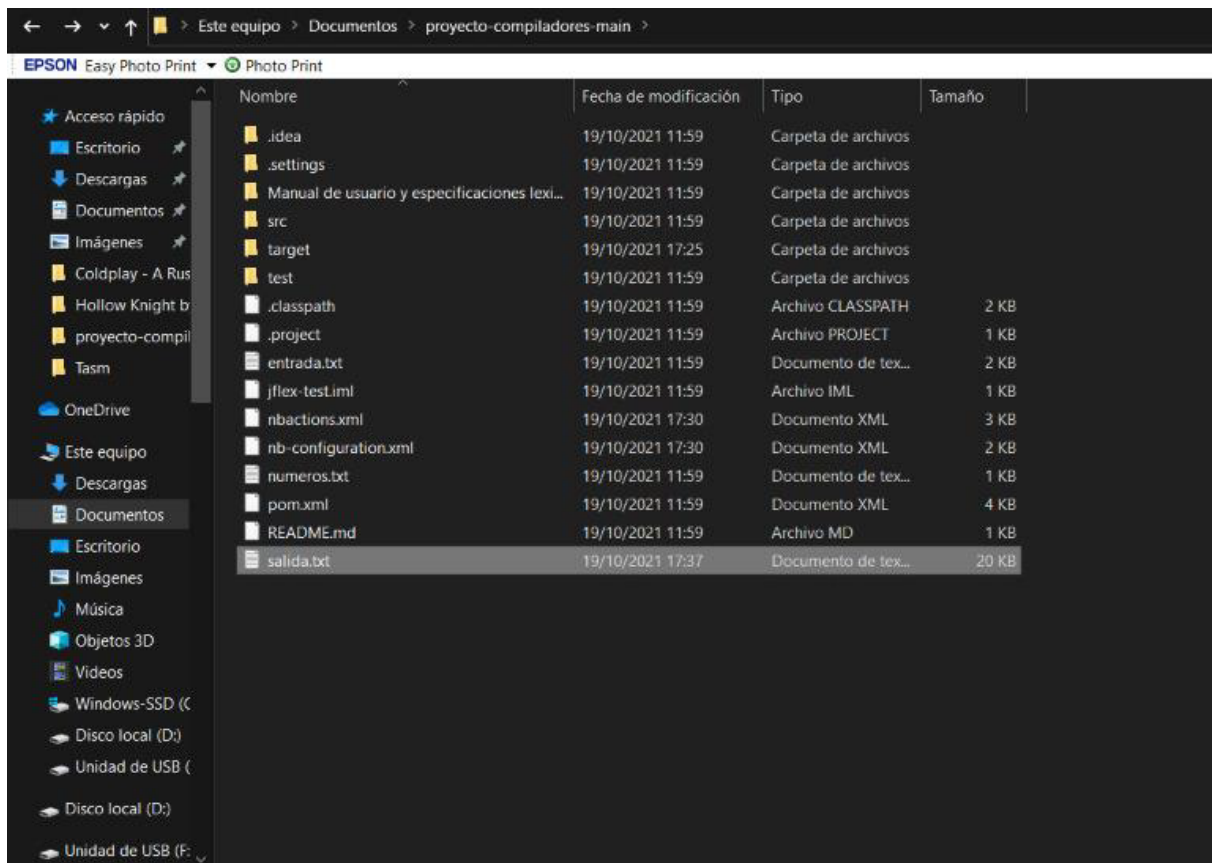
- A. Después de haber descargado el archivo se necesitará se necesitará como requerimiento java 11.
- B. Para ejecutar este debe de tener algún IDE, nosotros sugerimos Apache NetBeans pues es el más sencillo de configurar.
- C. Abrimos NetBeans nos ubicamos en la parte superior izquierda y daremos click en File y se nos desplegara una barra de tareas en la cual haremos click en OpenProject o ya sea directamente con el teclado, tecleamos Ctrl+Mayús+O,



- D. Luego se nos desplegará una ventana en la cual tendremos que buscar el archivo descargado y darle click para que este ya esté entre los archivos de NetBeans el cual se mostrará en la parte izquierda.



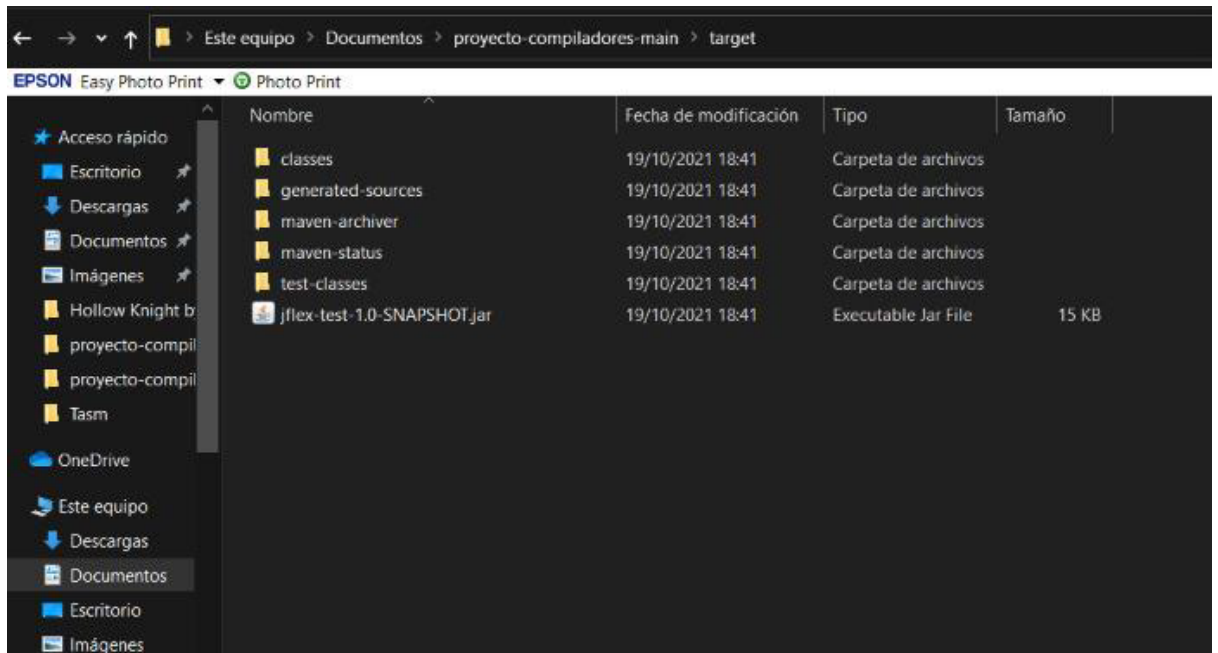
- F. En el archivo entrada.txt se va a ingresar la toda la línea de código que se va a manejar, una vez terminado se guarda y se pasa al programa.
- G. Al ejecutar va a leer el archivo entrada.txt y se va a generar un archivo ".txt" en la carpeta del proyecto con el nombre de salida.



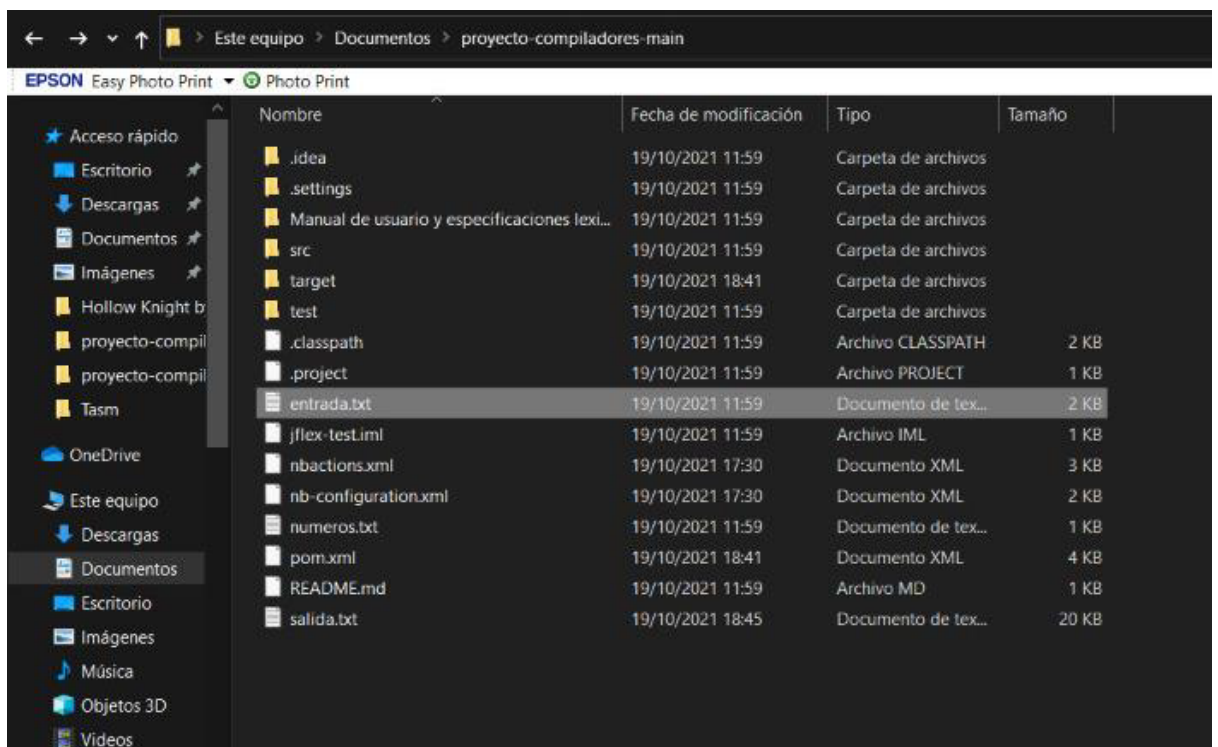
- H. En este archivo se encontrará todos los tokens que se encontraran mostrándonos la palabra encontrada, el lexema con el que coincide y la línea y columna en la que se encuentra.

```
salida.txt X
C: > Users > Alison > Documents > proyecto-compiladores-main > salida.txt
1 Comentario - Línea 1 | Columna: 1
2 Fin de línea encontrado - Línea 2 | Columna: 1
3 Palabra encontrada: 'entero' coincide con lexema: ENTERO - Línea 3 | Columna: 1
4 Palabra encontrada: 'multiplicar' coincide con lexema: VARIABLE - Línea 3 | Columna: 8
5 Palabra encontrada: '(' coincide con lexema: INICIOPARENTESIS - Línea 3 | Columna: 20
6 Palabra encontrada: 'entero' coincide con lexema: ENTERO - Línea 3 | Columna: 21
7 Palabra encontrada: 'a' coincide con lexema: VARIABLE - Línea 3 | Columna: 28
8 Palabra encontrada: ',' coincide con lexema: COMA - Línea 3 | Columna: 29
9 Palabra encontrada: 'entero' coincide con lexema: ENTERO - Línea 3 | Columna: 31
10 Palabra encontrada: 'b' coincide con lexema: VARIABLE - Línea 3 | Columna: 38
11 Palabra encontrada: ')' coincide con lexema: FINALPARENTESIS - Línea 3 | Columna: 39
12 Fin de línea encontrado - Línea 3 | Columna: 40
13 Palabra encontrada: 'devolver' coincide con lexema: DEVOLVER - Línea 4 | Columna: 3
14 Palabra encontrada: 'a' coincide con lexema: VARIABLE - Línea 4 | Columna: 12
15 Palabra encontrada: '*' coincide con lexema: MULTIPLICACION - Línea 4 | Columna: 14
16 Palabra encontrada: 'b' coincide con lexema: VARIABLE - Línea 4 | Columna: 16
17 Fin de línea encontrado - Línea 4 | Columna: 17
18 Fin de línea encontrado - Línea 5 | Columna: 1
19 Palabra encontrada: 'clase' coincide con lexema: CLASE - Línea 6 | Columna: 1
20 Palabra encontrada: 'Rectangulo' coincide con lexema: VARIABLECLASE - Línea 6 | Columna: 7
21 Fin de línea encontrado - Línea 6 | Columna: 17
22 Palabra encontrada: 'propiedades' coincide con lexema: PROPIEDADES - Línea 7 | Columna: 3
23 Palabra encontrada: 'publicas' coincide con lexema: PUBLICAS - Línea 7 | Columna: 15
24 Palabra encontrada: ':' coincide con lexema: DOSPUNTOS - Línea 7 | Columna: 23
25 Fin de línea encontrado - Línea 7 | Columna: 24
26 Palabra encontrada: 'entero' coincide con lexema: ENTERO - Línea 8 | Columna: 3
27 Palabra encontrada: 'ancho' coincide con lexema: VARIABLE - Línea 8 | Columna: 10
28 Fin de línea encontrado - Línea 8 | Columna: 15
29 Palabra encontrada: 'entero' coincide con lexema: ENTERO - Línea 9 | Columna: 3
30 Palabra encontrada: 'largo' coincide con lexema: VARIABLE - Línea 9 | Columna: 10
31 Fin de línea encontrado - Línea 9 | Columna: 17
32 Palabra encontrada: 'metodos' coincide con lexema: METODOS - Línea 10 | Columna: 3
33 Palabra encontrada: 'publicos' coincide con lexema: PUBLICOS - Línea 10 | Columna: 11
34 Palabra encontrada: ':' coincide con lexema: DOSPUNTOS - Línea 10 | Columna: 19
35 Fin de línea encontrado - Línea 10 | Columna: 20
36 Palabra encontrada: 'constructor' coincide con lexema: CONSTRUCTOR - Línea 11 | Columna: 3
37 Palabra encontrada: '(' coincide con lexema: INICIOPARENTESIS - Línea 11 | Columna: 15
38 Palabra encontrada: 'entero' coincide con lexema: ENTERO - Línea 11 | Columna: 16
```

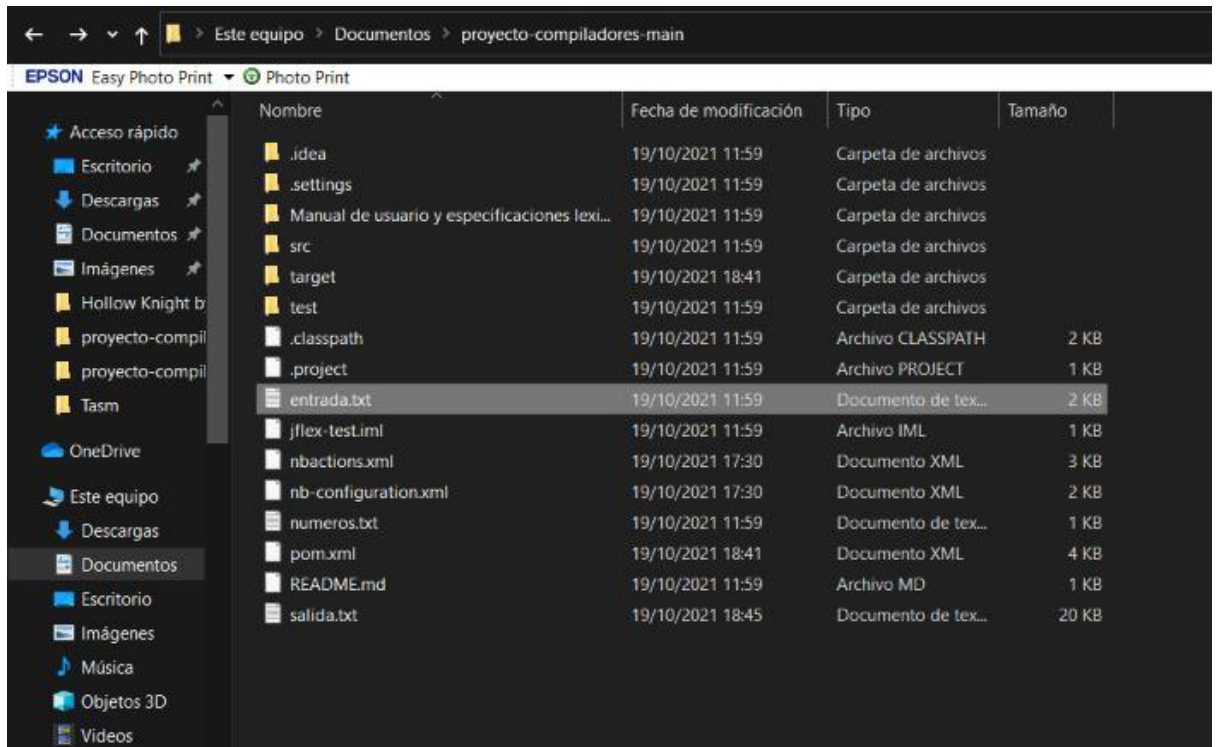
- I. En caso que sea desde consola abrimos el CMD, una vez dentro usamos el comando cd para dirigirnos al archivo del proyecto y en esta dirigirnos a la carpeta target en la cual encontraremos un archivo jar.



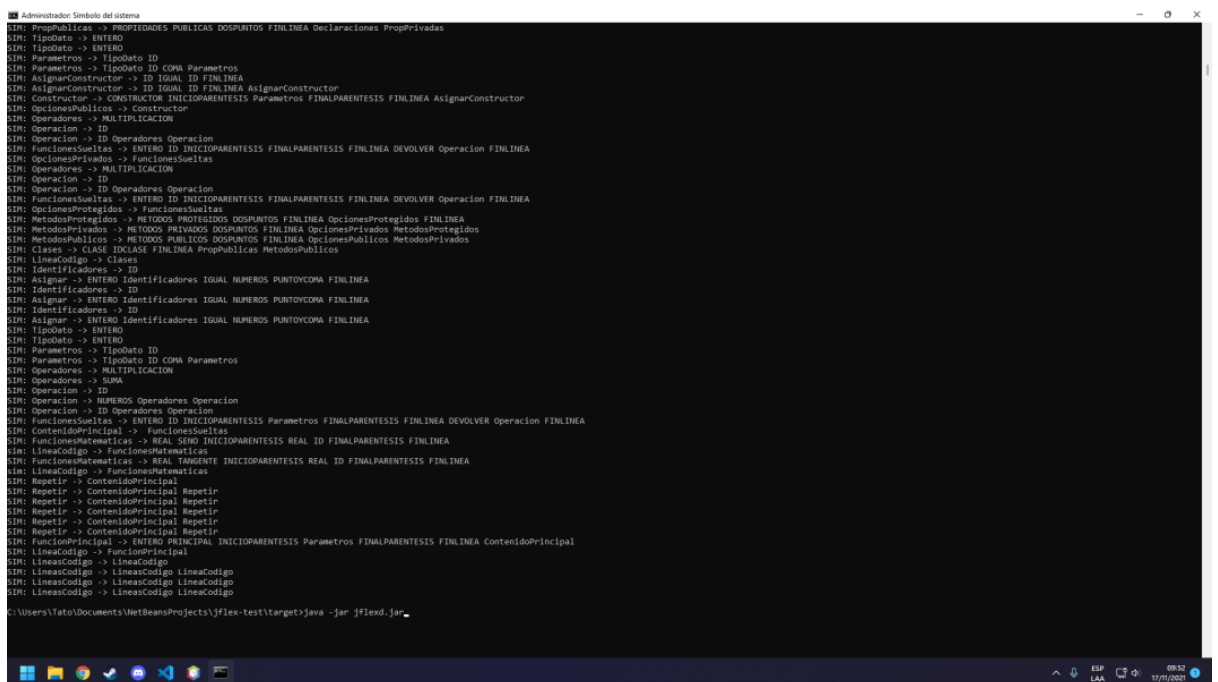
- J. Una vez estemos dentro de este archivo colocaremos el siguiente comando: `java -jar jflex-test-1.0-SNAPSHOT.jar ""` Entre estas comillas está la dirección del archivo que se necesita leer en este caso sería el archivo `entrada.txt`.



- K. Una vez ejecutado, se nos mostrará todo el resultado como se vio anteriormente y nos creará el archivo `salida.txt`.



- L. Además en consola se mostrarán las reducciones de las producciones que se encuentren dentro del archivo “entrada.txt”



- M. Se puede ejecutar el programa desde consola, con el siguiente comando “-jar jflexed.jar” ubicándose desde donde se encuentra el archivo inicialmente.