



PROYECTO 1

06.04.2021

Carlos Benjamín Pac Flores

Centro Universitario de Occidente (CUNOC)

Ingeniería en Ciencias y Sistemas

Febrero 2022

Introducción	2
Plataforma de Desarrollo:	3
Alcances Tecnicos:	3
Requisitos De Software:	3
Detalle del Sistema:	4
Estructura del analisis Lexico:	4
Estructura del Análisis Sintáctico:	6
Símbolos Terminales:	6
Símbolos No Terminales:	6
Reglas de Producción:	6
Validación de los valores específicos (análisis semántico):	8
Código de Ejemplo:	9
Organización del Proyecto:	10
Estructura del Proyecto:	10

Introducción

El siguiente documento da presenta una descripción de la estructura del software de la aplicación servidor y aplicación cliente desarrollada en java 11 con multiplataforma, y que por medio de la aplicación de la app cliente se ubica y envía código de estilo java y que por medio del servidor lo procesamos mediante un analisis lexico y sintactico para obtener la cantidad de elementos repetidos en cada proyecto, y se envía un respuesta en formato pseudo Json para su uso en un lenguaje llamado COPY para generar un html con toda la información recopilada en el análisis.

Plataforma de Desarrollo:

- Sistema de gestión de proyecto Maven.
- NetBeans y Visual Studio Code
- Librería para análisis léxico JFLEX
- Librería para análisis sintáctico CUP
- Lenguaje Java versión 11

Alcances Tecnicos:

Utilizando el modelo de cliente servidor en implementación de sockets de aplicaciones java realizar la verificación de copia entre dos proyectos de java con esto también realizar el analisis lexico y sintactico de los mismos para poder verificar también la lógica de lo escrito en los archivos de java, ya realizado el análisis de los proyectos el servidor debe enviar una respuesta en un formato similar a json con todo lo encontrado y en la aplicación cliente poder recibir esta información y verificarla por medio de un análisis lexico y sintactico para establecer que la información recibida es correcta y recuperarla de igual manera; así también el diseño de un lenguaje muy similar a html donde se puede representar toda la información recibida del json a través de una variable global RESULT.

Requisitos De Software:

- Java 11
- Ram recomendada 2 GB

Detalle del Sistema:

Estructura del analisis Lexico JFLEX App Cliente:

- Lenguaje Json:

El lenguaje es muy similar al lenguaje Json estándar, pero implementado caseless es decir que no importa si el nombre de los parámetros esta en minúsculas o mayúsculas o una mezcla de ambos.

```
LineTerminator = \r|\n|\r\n
```

```
WhiteSpace      = {LineTerminator} | [ \t\f]
```

```
text = [\w]+
```

```
Number = [0-9]
```

```
Entero = {Number}+
```

```
<YYINITIAL> {
```

```
    "Score"                {Reconoce la propiedad Json Score}
```

```
    "Clases"               {Reconoce la propiedad Json Clases }
```

```
    "Variables"            {Reconoce la propiedad Json Variables}
```

```
    "Nombre"               {Reconoce la propiedad Json Nombre}
```

```
    "Tipo"                 {Reconoce la propiedad Json Tipo}
```

```
    "Funcion"              {Reconoce la propiedad Json Funcion}
```

```
    "Metodos"              {Reconoce la propiedad Json Metodos}
```

```
    "Parametros"           {Reconoce la propiedad Json Parametros}
```

```
    "Comentarios"          {Reconoce la propiedad Json Comentarios}
```

```
    "Texto"                {Reconoce la propiedad Json texto}
```

```
    {Entero}               {Reconoce el conjunto de dígitos que representa
```

```
un número entero}
```

```
    {text}                 {Reconoce una conjunto de símbolos}
```

```
    [\[ ]                 {Reconoce el símbolo [ ] }
```

```
    [\] ]                 {Reconoce el símbolo ] }
```

```
    [{ }                  {Reconoce el símbolo { }
```

```
    [ ] }                  {Reconoce el símbolo } }
```

```
    [, ]                  {Reconoce el símbolo , }
```

```
    [: ]                  {Reconoce el símbolo : }
```

```
    [\" ]                 {Reconoce el símbolo “ }
```

```
    {WhiteSpace}          {Reconoce los espacios y los ignora}
```

```
<STRING> {
```

```
    \"                    {Reconoce el símbolo “ }
```

```

    [^\n\r"\\]+    {Reconoce cualquier otro símbolo excepto: \n\r"}
    \\t            {Reconoce el símbolo \t }
    \\n            {Reconoce el símbolo \n }
    \\r            {Reconoce el símbolo \r }
    \\\\"          {Reconoce el símbolo \" }
    \\             {Reconoce el símbolo \ }
}

[^]              {Cualquier carácter fuera de las reglas anteriores}

```

- Lenguaje copy archivo .def

```

LineTerminator = \r|\n|\r\n
WhiteSpace     = {LineTerminator} | [ \t\f]
Comment        = {TraditionalComment}
TraditionalComment = "</" [^/] ~"/>" | "</" "/" + ">"
Number         = [0-9]
Entero         = {Number}+
Identifier     = [:jletter:] [:jletterdigit:]*
text          = [\w]+([\ ]+[\w]+)*

<YYINITIAL> {
    {Comment}          {Reconoce los comentarios y lo ignora}
    "Integer"          {Reconoce la palabra reservada Integer}
    "String"           {Reconoce la palabra reservada String}
    "html"             {Reconoce la etiqueta HTML html}
    "h1"               {Reconoce la etiqueta HTML h1}
    "h2"               {Reconoce la etiqueta HTML h2}
    "table"            {Reconoce la etiqueta HTML table}
    "for"              {Reconoce la etiqueta HTML especial for}
    "th"               {Reconoce la etiqueta HTML th}
    "tr"               {Reconoce la etiqueta HTML tr}
    "td"               {Reconoce la etiqueta HTML td}
    "br"               {Reconoce la etiqueta HTML br}
    "iterador"         {Reconoce la palabra reservada iterador}
    "hasta"            {Reconoce la palabra reservada hasta}
    "$$("              {Reconoce el símbolo de agrupación especial $$({}
    ")$$"              {Reconoce el símbolo de agrupación especial )$$}
    {Identifier}       {Reconoce los identificadores similares a JAVA}
    {Entero}           {Reconoce cadena de dígitos que forman un entero}
    [*]                {Reconoce el símbolo * }
    [/]                {Reconoce el símbolo / }
}

```

```

[+]          {Reconoce el símbolo + }
[-]          {Reconoce el símbolo - }
"="          {Reconoce el símbolo = }
"</"        {Reconoce la agrupación HTML </ }
"<"         {Reconoce la agrupación HTML < }
">"         {Reconoce la agrupación HTML > }
[(]          {Reconoce el símbolo ( }
[)]          {Reconoce el símbolo ) }
\[[]         {Reconoce el símbolo [ }
\]\]         {Reconoce el símbolo ] }
[.]          {Reconoce el símbolo . }
[,]          {Reconoce el símbolo , }
[:]          {Reconoce el símbolo : }
[;]          {Reconoce el símbolo ; }
\["]         {Reconoce el símbolo " }
{WhiteSpace} {Reconoce los espacios en blanco y los ignora }
}

<STRING>    {
    \"        {Reconoce el símbolo \" }
    [^\n\r\"\\]+ {Reconoce el conjunto de símbolos excepto \n\r\"\\ }
    \\t        {Reconoce el símbolo \\t }
    \\n        {Reconoce el símbolo \\n }
    \\r        {Reconoce el símbolo \\r }
    \\\\"      {Reconoce el símbolo \\" }
    \\         {Reconoce el símbolo \\ }
}

<INERTEXT>{
    {text}      {Reconoce las cadenas de símbolos que representen texto }
    [.]         {Reconoce el símbolo . }
    "</"        {Reconoce la agrupación HTML </ }
    "<"         {Reconoce la agrupación HTML < }
    "$$("       {Reconoce el símbolo de agrupación especial $$({ }
    {WhiteSpace} {Reconoce el símbolo : }
}

[^] {Cualquier carácter fuera de las reglas anteriores}

```

Estructura del Análisis Sintáctico App Cliente:

- Lenguaje Json

Símbolos Terminales:

CO_A, CO_C, LLA_A, LLA_C, PA_A, PA_C, COMA, PUNTO, DOSPUNTOS, PUNTOCOMA, STRING, ENTER
O, INT, STR, ITERATOR, HASTA, D_PA_A, PA_C_D, EQUAL, ME_Q, MA_Q, ME_QB, MAS, MENOS, DIV,
MUL, HTML, H1, H2, TABLE, FOR, TR, TH, TD, BR, TEXT, ID

Símbolos No Terminales:

Ini, asig, dec, decn, def, decp, exp ,t ,f ,html ,contHTML ,tablec
,tablectag ,trc ,paramhasta ,etiquetas ,varHTML2 ,varHTML4

Reglas De Producción:

```
def -> def html
def -> ini
def -> html
ini -> ini asig
ini -> ini dec
ini -> ini decn
ini -> asig
ini -> dec
ini -> decn
decn -> INT ID EQUAL exp PUNTOCOMA
decn -> STR ID EQUAL exp PUNTOCOMA
dec -> INT decp
dec -> STR decp
decp -> ID:id COMA decp
decp -> ID:id PUNTOCOMA
asig -> ID EQUAL exp PUNTOCOMA
exp -> exp:val1 MAS:s t:val2
exp -> exp:val1 MENOS:s t:val2
exp -> t
t -> t:val1 MUL:s f:val2
t -> t:val1 DIV:s f:val2
t -> f
f -> ID
f -> ID:id1 PUNTO ID:id2
f -> ID:id1 PUNTO ID CO_A:ref exp:index CO_C
f -> ID:id1 PUNTO ID CO_A:ref exp:index CO_C PUNTO ID:id3
f -> ENTERO
f -> STRING
f -> PA_A exp PA_C
html -> ME_Q HTML MA_Q contHTML ME_QB HTML MA_Q
```



```

html -> ME_Q HTML MA_Q ME_QB HTML MA_Q
contHTML -> contHTML ME_Q etiquetas
contHTML -> contHTML TEXT
contHTML -> contHTML D_PA_A varHTML2 PA_C_D
contHTML -> contHTML D_PA_A PA_C_D
contHTML -> ME_Q etiquetas
contHTML -> TEXT
contHTML -> D_PA_A varHTML2 PA_C_D
contHTML -> D_PA_A PA_C_D
etiquetas -> BR MA_Q
etiquetas -> H1 MA_Q contHTML ME_QB H1 MA_Q
etiquetas -> H1 MA_Q ME_QB H1 MA_Q
etiquetas -> H2 MA_Q contHTML ME_QB H2 MA_Q
etiquetas -> H2 MA_Q ME_QB H2 MA_Q
etiquetas -> TABLE MA_Q tablec ME_QB TABLE MA_Q
etiquetas -> TABLE MA_Q ME_QB TABLE MA_Q
etiquetas -> FOR ITERATOR DOSPUNTOS ID HASTA DOSPUNTOS paramhasta PUNTOCOMA MA_Q
contHTML ME_QB FOR MA_Q
tablec -> tablec ME_Q tablectag
tablec -> ME_Q tablectag
tablec -> tablec TEXT
tablec -> tablec D_PA_A varHTML2 PA_C_D
tablec -> tablec D_PA_A PA_C_D
tablec -> TEXT
tablec -> D_PA_A varHTML2 PA_C_D
tablec -> D_PA_A PA_C_D
tablectag -> TR MA_Q trc ME_QB TR MA_Q
tablectag -> TR MA_Q ME_QB TR MA_Q
tablectag -> BR:tag MA_Q
tablectag -> H1 MA_Q contHTML ME_QB H1 MA_Q
tablectag -> H1 MA_Q ME_QB H1 MA_Q
tablectag -> H2 MA_Q contHTML ME_QB H2 MA_Q
tablectag -> H2 MA_Q ME_QB H2 MA_Q
tablectag -> TABLE MA_Q tablec ME_QB TABLE MA_Q
tablectag -> TABLE MA_Q ME_QB TABLE MA_Q
tablectag -> FOR ITERATOR DOSPUNTOS ID HASTA DOSPUNTOS paramhasta PUNTOCOMA MA_Q
tablec ME_QB FOR MA_Q
trc -> trc ME_Q TH MA_Q contHTML ME_QB TH MA_Q
trc -> trc ME_Q TH MA_Q ME_QB TH MA_Q
trc -> trc ME_Q TD MA_Q contHTML ME_QB TD MA_Q
trc -> trc ME_Q TD MA_Q ME_QB TD MA_Q
trc -> ME_Q TH MA_Q contHTML ME_QB TH MA_Q
trc -> ME_Q TH MA_Q ME_QB TH MA_Q
trc -> ME_Q TD MA_Q contHTML ME_QB TD MA_Q
trc -> ME_Q TD MA_Q ME_QB TD MA_Q
varHTML2 -> ID:id1
varHTML2 -> ID:id1 PUNTO ID

```

```

varHTML2 -> ID:id1 PUNTO ID CO_A varHTML4:index CO_C
varHTML2 -> ID:id1 PUNTO ID CO_A varHTML4:index CO_C PUNTO ID:id3
varHTML4 -> ID
varHTML4 -> ENTERO
paramhasta -> ID
paramhasta -> ENTERO

```

- Lenguaje copy archivo .def

Símbolos Terminales:

CO_A, CO_C, LLA_A, LLA_C, COMA, DOSPUNTOS, STRING, ENTERO ,CLASS ,VAR ,NAME ,TYPE ,FUN ,METD ,COMENT ,PARA ,TEXT

Símbolos No Terminales:

Ini, reporte, reportep, contClass ,contClassp, contVar, contVarp, contMet, contMetp, contCom, contComp, expVar, expMet

Reglas De Producción:

```

ini -> LLA_A:ref reportep LLA_C
reportep -> reporte SCORE DOSPUNTOS STRING
reportep -> reporte CLASS DOSPUNTOS CO_A contClass CO_C
reportep -> reporte CLASS DOSPUNTOS CO_A CO_C
reportep -> reporte VAR DOSPUNTOS CO_A contVar CO_C
reportep -> reporte VAR DOSPUNTOS CO_A CO_C
reportep -> reporte METD DOSPUNTOS CO_A contMet CO_C
reportep -> reporte METD DOSPUNTOS CO_A CO_C
reportep -> reporte COMENT DOSPUNTOS CO_A contCom CO_C
reportep -> reporte COMENT DOSPUNTOS CO_A CO_C
reporte -> reporte SCORE DOSPUNTOS STRING COMA
reporte -> reporte CLASS DOSPUNTOS CO_A contClass CO_C COMA
reporte -> reporte CLASS DOSPUNTOS CO_A CO_C COMA
reporte -> reporte VAR DOSPUNTOS CO_A contVar CO_C COMA
reporte -> reporte VAR DOSPUNTOS CO_A CO_C COMA
reporte -> reporte METD DOSPUNTOS CO_A contMet CO_C COMA
reporte -> reporte METD DOSPUNTOS CO_A CO_C COMA
reporte -> reporte COMENT DOSPUNTOS CO_A contCom CO_C COMA
reporte -> reporte COMENT DOSPUNTOS CO_A CO_C COMA
reporte -> λ
contClass -> contClassp LLA_A NAME DOSPUNTOS STRING LLA_C
contClassp -> contClassp LLA_A NAME DOSPUNTOS STRING LLA_C COMA
contClassp -> λ
contVar -> contVarp LLA_A expVar LLA_C
contVarp -> contVarp LLA_A expVar LLA_C COMA
contVarp -> λ
contMet -> contMetp LLA_A expMet LLA_C
contMetp -> contMetp LLA_A expMet LLA_C COMA

```

```

contMetp -> λ
expVar -> NAME DOSPUNTOS STRING COMA TYPE DOSPUNTOS STRING COMA FUN DOSPUNTOS
STRING
expVar -> NAME DOSPUNTOS STRING COMA FUN DOSPUNTOS STRING COMA TYPE DOSPUNTOS
STRING
expVar -> FUN DOSPUNTOS STRING COMA NAME DOSPUNTOS STRING COMA TYPE DOSPUNTOS
STRING
expVar -> FUN DOSPUNTOS STRING COMA TYPE DOSPUNTOS STRING COMA NAME DOSPUNTOS
STRING
expVar -> TYPE DOSPUNTOS STRING COMA FUN DOSPUNTOS STRING COMA NAME DOSPUNTOS
STRING
expVar -> TYPE DOSPUNTOS STRING COMA NAME DOSPUNTOS STRING COMA FUN DOSPUNTOS
STRING
expMet -> NAME DOSPUNTOS STRING COMA TYPE DOSPUNTOS STRING COMA PARA DOSPUNTOS
ENTERO
expMet -> NAME DOSPUNTOS STRING COMA PARA DOSPUNTOS ENTERO COMA TYPE DOSPUNTOS
STRING
expMet -> TYPE DOSPUNTOS STRING COMA PARA DOSPUNTOS ENTERO COMA NAME DOSPUNTOS
STRING
expMet -> TYPE DOSPUNTOS STRING COMA NAME DOSPUNTOS STRING COMA PARA DOSPUNTOS
ENTERO
expMet -> PARA DOSPUNTOS ENTERO COMA TYPE DOSPUNTOS STRING COMA NAME DOSPUNTOS
STRING
expMet -> PARA DOSPUNTOS ENTERO COMA NAME DOSPUNTOS STRING COMA TYPE DOSPUNTOS
STRING
contCom -> contComp LLA_A TEXT DOSPUNTOS STRING LLA_C
contComp -> contComp LLA_A TEXT DOSPUNTOS STRING LLA_C COMA
contComp -> λ

```

Código de Ejemplo Json:

```

{
  Score: "0.9578",
  Clases: [
    {Nombre: "ejemplo_final"},
    {Nombre: "java2"},
    {Nombre: "CrearArchivos"},
    {Nombre: "java4"},
    {Nombre: "Cliente"},
    {Nombre: "Servidor"}
  ],
  Variables: [
    {Nombre: "tabla", Tipo: "Object", Funcion: "Metodo parse, Metodo parse"},
    {Nombre: "hola", Tipo: "int", Funcion: "Metodo Saludar, Metodo Saludar"},
    {Nombre: "var_char", Tipo: "char", Funcion: "Metodo Saludar, Metodo Saludar"},
    {Nombre: "val", Tipo: "String", Funcion: "Metodo Saludar, Metodo Saludar"},
    {Nombre: "valid", Tipo: "boolean", Funcion: "Clase ejemplo_final, Clase

```

```

ejemplo_final"},
    {Nombre: "value", Tipo: "String", Funcion: "Clase ejemplo_final, Clase
ejemplo_final"},
    {Nombre: "units", Tipo: "int", Funcion: "Metodo metodo, Metodo metodo"},
    {Nombre: "billpay", Tipo: "double", Funcion: "Metodo metodo, Metodo metodo"},
    {Nombre: "var1", Tipo: "Object", Funcion: "Metodo metodo, Metodo metodo"},
    {Nombre: "var45", Tipo: "String", Funcion: "Metodo metodo, Metodo metodo"},
    {Nombre: "var45345", Tipo: "String", Funcion: "Metodo metodo, Metodo metodo"},
    {Nombre: "ggdf", Tipo: "String", Funcion: "Clase java2, Clase java2"},
    {Nombre: "vart", Tipo: "String", Funcion: "Clase java2, Clase java2"},
    {Nombre: "code", Tipo: "String", Funcion: "Clase java2, Clase java2"},
    {Nombre: "generalPath", Tipo: "String", Funcion: "Metodo crear_guardar, Metodo
crear_guardar"},
    {Nombre: "carpeta", Tipo: "String", Funcion: "Metodo crear_guardar, Metodo
crear_guardar"},
    {Nombre: "pathCopy", Tipo: "String", Funcion: "Metodo crear_guardar, Metodo
crear_guardar"},
    {Nombre: "pathDef", Tipo: "String", Funcion: "Metodo crear_guardar, Metodo
crear_guardar"},
    {Nombre: "pathJson", Tipo: "String", Funcion: "Metodo crear_guardar, Metodo
crear_guardar"},
    {Nombre: "carpetaFile", Tipo: "Object", Funcion: "Metodo crear_guardar, Metodo
crear_guardar"},
    {Nombre: "archivoJson", Tipo: "Object", Funcion: "Metodo crear_guardar, Metodo
crear_guardar"},
    {Nombre: "archivoDef", Tipo: "Object", Funcion: "Metodo crear_guardar, Metodo
crear_guardar"},
    {Nombre: "fw", Tipo: "Object", Funcion: "Metodo crear_guardar, Metodo
crear_guardar"},
    {Nombre: "bw", Tipo: "Object", Funcion: "Metodo crear_guardar, Metodo
crear_guardar"},
    {Nombre: "oos", Tipo: "Object", Funcion: "Metodo crear_guardar, Metodo
crear_guardar"},
    {Nombre: "archivo", Tipo: "Object", Funcion: "Metodo crear_guardar, Metodo
crear_guardar"},
    {Nombre: "proyectoCopy", Tipo: "Object", Funcion: "Metodo crear_guardar, Metodo
crear_guardar"},
    {Nombre: "extencion", Tipo: "String", Funcion: "Metodo verificarArchivos, Metodo
verificarArchivos"},
    {Nombre: "archivos", Tipo: "Object", Funcion: "Metodo verificarArchivos, Metodo
verificarArchivos"},
    {Nombre: "MODEL_JSON", Tipo: "String", Funcion: "Clase java4, Clase java4"},
    {Nombre: "puerto", Tipo: "int", Funcion: "Constructor Cliente, Constructor Cliente,
Metodo setPuerto, Clase Cliente, Constructor Servidor, Metodo setPuerto, Clase Servidor"},
    {Nombre: "host", Tipo: "String", Funcion: "Constructor Cliente, Constructor Cliente,
Metodo setHost, Clase Cliente"},
    {Nombre: "puerto", Tipo: "int", Funcion: "Metodo setPuerto, Constructor Cliente,
Metodo setPuerto, Clase Cliente, Constructor Servidor, Metodo setPuerto, Clase Servidor"},
    {Nombre: "host", Tipo: "String", Funcion: "Metodo setHost, Constructor Cliente,
Metodo setHost, Clase Cliente"},

```

```

    {Nombre: "mensaje", Tipo: "Object", Funcion: "Metodo setMensaje, Metodo setMensaje,
Clase Cliente, Metodo run"},
    {Nombre: "socket", Tipo: "Object", Funcion: "Metodo run, Metodo run, Metodo run"},
    {Nombre: "out", Tipo: "Object", Funcion: "Metodo run, Metodo run, Metodo enviar"},
    {Nombre: "out", Tipo: "Object", Funcion: "Metodo enviar, Metodo run, Metodo enviar"},
    {Nombre: "data", Tipo: "Object", Funcion: "Metodo enviar, Metodo enviar"},
    {Nombre: "mensaje", Tipo: "Object", Funcion: "Clase Cliente, Metodo setMensaje, Clase
Cliente, Metodo run"},
    {Nombre: "host", Tipo: "String", Funcion: "Clase Cliente, Constructor Cliente, Metodo
setHost, Clase Cliente"},
    {Nombre: "puerto", Tipo: "int", Funcion: "Clase Cliente, Constructor Cliente, Metodo
setPuerto, Clase Cliente, Constructor Servidor, Metodo setPuerto, Clase Servidor"},
    {Nombre: "puerto", Tipo: "int", Funcion: "Constructor Servidor, Constructor Cliente,
Metodo setPuerto, Clase Cliente, Constructor Servidor, Metodo setPuerto, Clase Servidor"},
    {Nombre: "puerto", Tipo: "int", Funcion: "Metodo setPuerto, Constructor Cliente,
Metodo setPuerto, Clase Cliente, Constructor Servidor, Metodo setPuerto, Clase Servidor"},
    {Nombre: "serverSocket", Tipo: "Object", Funcion: "Metodo run, Metodo run"},
    {Nombre: "socket", Tipo: "Object", Funcion: "Metodo run, Metodo run, Metodo run"},
    {Nombre: "in", Tipo: "Object", Funcion: "Metodo run, Metodo run"},
    {Nombre: "mensaje", Tipo: "Object", Funcion: "Metodo run, Metodo setMensaje, Clase
Cliente, Metodo run"},
    {Nombre: "puerto", Tipo: "int", Funcion: "Clase Servidor, Constructor Cliente, Metodo
setPuerto, Clase Cliente, Constructor Servidor, Metodo setPuerto, Clase Servidor"}
],
Metodos: [
    {Nombre: "ejemplo_final", Tipo: "Constructor", Parametros: 0},
    {Nombre: "parse", Tipo: "boolean", Parametros: 0},
    {Nombre: "Saludar", Tipo: "String", Parametros: 1},
    {Nombre: "java2", Tipo: "Constructor", Parametros: 0},
    {Nombre: "metodo", Tipo: "String", Parametros: 0},
    {Nombre: "CrearArchivos", Tipo: "Constructor", Parametros: 0},
    {Nombre: "crear_guardar", Tipo: "void", Parametros: 2},
    {Nombre: "java4", Tipo: "Constructor", Parametros: 0},
    {Nombre: "verificarArchivos", Tipo: "Object", Parametros: 1},
    {Nombre: "Cliente", Tipo: "Constructor", Parametros: 0},
    {Nombre: "Cliente", Tipo: "Constructor", Parametros: 2},
    {Nombre: "getPuerto", Tipo: "int", Parametros: 0},
    {Nombre: "getPuerto", Tipo: "int", Parametros: 0},
    {Nombre: "setPuerto", Tipo: "void", Parametros: 1},
    {Nombre: "setPuerto", Tipo: "void", Parametros: 1},
    {Nombre: "getHost", Tipo: "String", Parametros: 0},
    {Nombre: "setHost", Tipo: "void", Parametros: 1},
    {Nombre: "getMensaje", Tipo: "Object", Parametros: 0},
    {Nombre: "setMensaje", Tipo: "void", Parametros: 1},
    {Nombre: "run", Tipo: "void", Parametros: 0},
    {Nombre: "run", Tipo: "void", Parametros: 0},
    {Nombre: "enviar", Tipo: "void", Parametros: 2},
    {Nombre: "Servidor", Tipo: "Constructor", Parametros: 0},
    {Nombre: "Servidor", Tipo: "Constructor", Parametros: 1},
    {Nombre: "getPuerto", Tipo: "int", Parametros: 0},

```

```

    {Nombre: "getPuerto", Tipo: "int", Parametros: 0},
    {Nombre: "setPuerto", Tipo: "void", Parametros: 1},
    {Nombre: "setPuerto", Tipo: "void", Parametros: 1},
    {Nombre: "run", Tipo: "void", Parametros: 0},
    {Nombre: "run", Tipo: "void", Parametros: 0}
  ],
  Comentarios: [
    {Texto: " comentario simple"},
    {Texto: " Statement,"},
    {Texto: ""},
    {Texto: " \"Comentarios\""},
    {Texto: "* * Comentarioooo * multilinea"},
    {Texto: " \"Hola aqui retorno\""},
    {Texto: "/ Envio de mensaje de respuesta al servidor"},
    {Texto: "*\n * Servidor\n "}
  ]
}

```

Código de Ejemplo copy.def:

```

</ iniciare a definir de alguna manera />
Integer Max1,Max2,Max3,Max4, i,j,k,l;
Max1=40;
Max2=5;
Max3=20;
Max4=6;
i=0;
j=0;
k=0;
l=0;
String texto="Su score fue de: "+RESULT.Score;
</ Aqui defino el html />
<html>
<H1>$( texto )$</h1>
<h2>Variables</h2>
  <table>
    <tr>
      <th>Numero</th>
      <th>Variable</th>
      <th>Tipo</th>
      <th>Funci'n</th>
    </tr>
    <for iterador:i hasta:Max1;>
      <tr>
        <td>$( i )$</td>
        <td>$( RESULT.Variables[i].Nombre )$</td>
        <td>$( RESULT.Variables[i].Tipo )$</td>

```

```

        <td> $$ ( RESULT.Variables[i].Funcion ) $$ </td>
    </tr>
</for>
</table>
<h2>Clases</h2>
<table>
    <tr>
        <th>Numero</th>
        <th>Nombre</th>
    </tr>
    <for iterador:j hasta:Max2;>
        <tr>
            <td> $$ ( j ) $$ </td>
            <td> $$ ( RESULT.Clases[j].Nombre ) $$ </td>
        </tr>
    </for>
</table>
<h2>Metodos</h2>
<table>
    <tr>
        <th>Numero</th>
        <th>Nombre</th>
        <th>Tipo</th>
        <th>Funci'on</th>
    </tr>
    <for iterador:k hasta:Max3;>
        <tr>
            <td> $$ ( k ) $$ </td>
            <td> $$ ( RESULT.Metodos[k].Nombre ) $$ </td>
            <td> $$ ( RESULT.Metodos[k].Tipo ) $$ </td>
            <td> $$ ( RESULT.Metodos[k].Parametros ) $$ </td>
        </tr>
    </for>
</table>
<h2>Comentarios</h2>
<table>
    <tr>
        <th>Numero</th>
        <th>Texto</th>
    </tr>
    <for iterador:l hasta:Max4;>
        <tr>
            <td> $$ ( l ) $$ </td>
            <td> $$ ( RESULT.Comentarios[l].Texto ) $$ </td>
        </tr>
    </for>
</table>

```

```
</html>
```

Estructura del analisis Lexico JFLEX Servidor:

```
LineTerminator = \r|\n|\r\n
InputCharacter = [^\r\n]
WhiteSpace      = {LineTerminator} | [ \t\f]
Identifier = [:jletter:] [:jletterdigit:]*
Number = [0-9]
Entero = {Number}+
Decimal = {Entero}[.]{Entero}
```

```
<YYINITIAL> {
    "true"          {Reconoce el valor lógico verdadero}
    "new"           {reconoce la palabra reservada new para objetos}
    "this"          {reconoce la palabra reservada this usada en métodos y
variables}
    "static"        {reconoce la propiedad estática de variables y metodos}
    "final"         {reconoce la propiedad final de las variables}
    "default"       {reconoce la palabra reservada exclusiva del switch}
    "false"         {reconoce el valor lógico falso}
    "import"        {reconoce la palabra reservada import}
    "public"        {reconoce la propiedad de visibilidad java public}
    "private"       {reconoce la propiedad de visibilidad java private}
    "protected"    {reconoce la propiedad de visibilidad java protected}
    "break"         {reconoce la palabra reservada break}
    "return"        {reconoce la palabra reservada return}
    "int"           {reconoce el tipo de variable int}
    "boolean"       {reconoce el tipo de variable boolean}
    "String"        {reconoce el tipo de variable String}
    "char"          {reconoce el tipo de variable char}
    "double"        {reconoce el tipo de variable double}
    "Object"        {reconoce el tipo de variable tipo objeto}
    "void"          {reconoce el tipo de función void}
    "if"            {reconoce la palabra reservada para el condicional if}
    "else"          {reconoce la palabra reservada para la continuación del
condicional if}
    "for"           {reconoce la palabra reservada para el ciclo for}
    "do"            {reconoce la palabra reservada para el ciclo do while}
    "while"         {reconoce la palabra reservada para el ciclo while y do
while}
    "switch"        {reconoce la palabra reservada para la sentencia
```


switch}	
"case"	{reconoce la palabra reservada case utilizada en la
sentencia switch}	
"class"	{reconoce la palabra reservada para la generacion de
una clase java}	
{Identifier}	{reconoce el nombre de un identificador de formato
java}	
"/*"	{ejecuta el sub estado -> COMENT_MULTII para comentarios
de múltiples líneas}	
"//"	{ejecuta el sub estado -> COMENTARIO para comentarios
simples de una sola línea}	
{Decimal}	{reconoce el conjunto de símbolos que conformen un
número decimal}	
{Entero}	{reconoce el conjunto de símbolos que conformen un
número entero}	
"!="	{Reconoce el operador lógico no igual}
"=="	{Reconoce el operador lógico igual}
"="	{Reconoce el símbolo de asignación igual a}
"<="	{Reconoce el símbolo lógico menor igual que}
">="	{Reconoce el símbolo lógico mayor igual que}
"<"	{Reconoce el símbolo lógico menor que}
">"	{Reconoce el símbolo lógico mayor que}
"&&"	{Reconoce el operador lógico AND}
" "	{Reconoce el operador lógico OR}
"!"	{Reconoce el operador lógico NOT}
"++"	{Reconoce el operador aritmético sumar uno}
"--"	{Reconoce el operador aritmético restar uno }
[*]	{Reconoce el operador aritmético multiplicar}
[/]	{Reconoce el operador aritmético dividir}
[+]	{Reconoce el operador aritmético sumar}
[-]	{Reconoce el operador aritmético restar}
[(]	{Reconoce el símbolo de agrupación paréntesis de
apertura}	
[)]	{Reconoce el símbolo de agrupación paréntesis de
cierre}	
[\\[{Reconoce el símbolo de agrupación corchete de
apertura}	
[\\]	{Reconoce el símbolo de agrupación corchete de cierre}
[{	{Reconoce el símbolo de agrupación llave de apertura}
}]	{Reconoce el símbolo de agrupación llave de cierre}
[.]	{Reconoce el símbolo punto}
[,]	{Reconoce el símbolo coma}

```

    [:]          {Reconoce el símbolo dos puntos}
    [;]          {Reconoce el símbolo punto y coma}
    ["]          {Ejecuta el sub estado -> STRING para detectar una
cadena de símbolos en java}
    '''          {Ejecuta el sub estado -> CHAR para detectar un valor
char de java}
    {WhiteSpace} {Detecta e ignora los espacios en blanco de java}
}

<STRING> {
    \"           {Termina el sub estado -> STRING}
    [^\\n\\r\\\"\\\\]+ {Detecta todos los símbolos excepto los saltos de
línea retorno de carro comillas y barra cruzada}
    \\t          {Detecta el macro tabulación}
    \\n          {Detecta el macro salto de línea}
    \\r          {detecta el macro de retorno de carro}
    \\\\\"         {Detecta el macro de la comillas internas}
    \\\\"         {Detecta el macro de la barra cruzada}
}

<CHAR> {
    '''          {Terminal el sub estado -> CHAR}
    [^'\\n\\r\\\"\\\\](''') {Detecta cualquier símbolo excepto la comilla simple,
salto de línea, retorno de carro, comilla, barra cruzada}
    \\t(''')      {Detecta el macro tabulación más una comilla de
cierre}
    \\n(''')      {Detecta el macro salto de línea más una comilla de
cierre}
    \\r(''')      {detecta el macro de retorno de carro más una
comilla de cierre}
    \\\\"(''')      {Detecta el macro de la comillas internas más una
comilla simple}
    \\\\'(''')      {Detecta el macro de la comilla simple interna más
otra comilla de cierre}
    \\'(''')      {Detecta el macro de la barra cruzada más una comilla
de cierre}
}

<COMENTARIO> {
    [\\n]          {Termina el subestado -> COMENTARIO}
    [^\\n\\"]+     {Detecta cualquier otros símbolos excepto el salto de
línea, comillas simples}

```

```

    \"          {Detecta las comillas}
}

<COMENT_MULTI> {
    \"/*\"      {Termina el sub estado -> COMENT_MULTI}
    [^\n*\" ]+  {Reconoce cualquier otros simbolos excepto el salto de
línea, el símbolo * y las comillas}
    [\\n]      {Reconoce el salto de línea}
    \"         {Reconoce las comillas}
    [*]        {Reconoce el símbolo *}
}

[^]    {Reconoce cualquier otro símbolo no conseguido con las reglas
anteriores y se presenta en forma de error}

```

Estructura del Análisis Sintáctico Servidor:

Símbolos Terminales:

IMPORT, PUBLIC, PRIVATE, PROTECTED, FINAL, BREAK, RETURN, INT, BOOL, STR, CHAR, DOUBLE, OBJ, CHAR_VAL, IF, ELSE, FOR, DO, WHILE, SWITCH, CASE, ID, DECIMAL, ENTERO, DIFERENTE, IGUAL, MENOR_IGUAL, THIS, MAYOR_IGUAL, MENOR_QUE, MAYOR_QUE, AND, OR, NOT, SUMAR, RESTAR, MUL, DIV, MAS, MENOS, PA_A, PA_C, CO_A, CO_C, LLA_A, LLA_C, PUNTO, COMA, DOSPUNTOS, PUNTOCOMA, STRING, CLASS, EQUAL, TRUE, FALSE, DEFAULT, STATIC, NEW, VOID

Símbolos No Terminales:

initJava, initJavaP, importJava, importJavaContenido, importJavaP, metodoP, sentC
onstru, parametrosLlamadaP, parametrosLlamada, classJava, sentencias, metodo, con
structorP, agregarfor, ciclowhile, condicionalif, constructor, cliclofor, cicloDo
While, sentanciaSalida, asignacionvariables, contenidoSwitch, contenidoSwitch2,
condicionalSwitch, llamadaFuncion, listaVariables, listaVariables2, declararVar
iables, contClass, parametros, parametrosP, declararVariables2, listaVariables3,
listaVariables4; visibilidad, tipo, asig, configMetodo, exp, t, f, g, h, i, j, k, l, llam
adaFuncionExp

Reglas De Producción:

```

initJava -> initJavaP classJava
initJava -> classJava

```

```


initJavaP -> importJavaP

```

```

importJavaP -> importJavaP importJava

```



```
importJavaP -> importJava
```

```
importJava -> IMPORT ID importJavaContenido
```

```
importJavaContenido -> PUNTO ID importJavaContenido
```

```
importJavaContenido -> PUNTO MUL PUNTOCOMA
```

```
importJavaContenido -> PUNTOCOMA
```

```
visibilidad -> PRIVATE
```

```
visibilidad -> PUBLIC
```

```
visibilidad -> PROTECTED
```

```
tipo -> INT
```

```
tipo -> BOOL
```

```
tipo -> STR
```

```
tipo -> CHAR
```

```
tipo -> DOUBLE
```

```
tipo -> OBJ
```

```
tipo -> ID
```

```
classJava -> visibilidad CLASS ID LLA_A contClass LLA_C
```

```
classJava -> visibilidad CLASS ID LLA_A LLA_C
```

```
classJava -> visibilidad FINAL CLASS ID LLA_A contClass LLA_C
```

```
classJava -> visibilidad FINAL CLASS ID LLA_A LLA_C
```

```
contClass -> contClass declararVariables
```

```
contClass -> contClass constructor
```

```
contClass -> contClass metodo
```

```
contClass -> declararVariables
```

```
contClass -> constructor
```

```
contClass -> metodo
```

```
constructor -> visibilidad ID:constructor PA_A parametros PA_C constructorP
```

```
constructor -> visibilidad ID:constructor PA_A PA_C constructorP
```

```
constructorP -> LLA_A LLA_C
```

```
constructorP -> LLA_A sentConstru LLA_C
```

```
parametros -> tipo ID
```

```
parametros -> tipo ID COMA parametrosP
```

parametrosP -> tipo ID COMA parametrosP

parametrosP -> tipo ID

metodo -> visibilidad configMetodo PA_A PA_C metodoP

metodo -> visibilidad STATIC configMetodo PA_A PA_C metodoP

metodo -> visibilidad configMetodo PA_A parametros PA_C metodoP

metodo -> visibilidad STATIC configMetodo PA_A parametros PA_C metodoP

configMetodo -> tipo ID

configMetodo -> VOID ID

metodoP -> LLA_A sentencias LLA_C

metodoP -> LLA_A LLA_C

asignacionvariables -> ID EQUAL asig PUNTOCOMA

asignacionvariables -> THIS PUNTO ID EQUAL asig PUNTOCOMA

asignacionvariables -> ID SUMAR PUNTOCOMA

asignacionvariables -> ID RESTAR PUNTOCOMA

declararVariables -> visibilidad STATIC FINAL tipo listaVariables PUNTOCOMA

declararVariables -> visibilidad FINAL STATIC tipo listaVariables PUNTOCOMA

declararVariables -> visibilidad STATIC tipo listaVariables PUNTOCOMA

declararVariables -> visibilidad FINAL tipo listaVariables PUNTOCOMA

declararVariables -> visibilidad tipo listaVariables PUNTOCOMA

declararVariables -> tipo listaVariables PUNTOCOMA

listaVariables -> ID

listaVariables -> ID EQUAL asig

listaVariables -> ID COMA listaVariables2

listaVariables -> ID EQUAL asig COMA listaVariables2

listaVariables2 -> ID EQUAL asig COMA listaVariables2

listaVariables2 -> ID COMA listaVariables2


listaVariables2 -> ID EQUAL asig

listaVariables2 -> ID

declararVariables2 -> FINAL tipo listaVariables3 PUNTOCOMA

declararVariables2 -> tipo listaVariables3 PUNTOCOMA

listaVariables3 -> ID




```
listaVariables3 -> ID EQUAL asig
listaVariables3 -> ID COMA listaVariables4
listaVariables3 -> ID EQUAL asig COMA listaVariables4
```

```
listaVariables4 -> ID EQUAL asig COMA listaVariables4
listaVariables4 -> ID COMA listaVariables4
listaVariables4 -> ID EQUAL asig
listaVariables4 -> ID
```

```
sentConstru -> sentConstru llamadaFuncion
sentConstru -> sentConstru condicionalif
sentConstru -> sentConstru ciclowhile
sentConstru -> sentConstru cicloDoWhile
sentConstru -> sentConstru condicionalSwitch
sentConstru -> sentConstru cliclofor
sentConstru -> sentConstru asignacionvariables
sentConstru -> sentConstru declararVariables2
sentConstru -> llamadaFuncion
sentConstru -> condicionalif
sentConstru -> ciclowhile
sentConstru -> cicloDoWhile
sentConstru -> condicionalSwitch
sentConstru -> cliclofor
sentConstru -> asignacionvariables
sentConstru -> declararVariables2
```

```
sentencias -> sentencias llamadaFuncion
sentencias -> sentencias condicionalif
sentencias -> sentencias ciclowhile
sentencias -> sentencias cicloDoWhile
sentencias -> sentencias condicionalSwitch
sentencias -> sentencias cliclofor
sentencias -> sentencias sentanciaSalida
sentencias -> sentencias asignacionvariables
sentencias -> sentencias declararVariables2
sentencias -> llamadaFuncion
sentencias -> condicionalif
sentencias -> ciclowhile
sentencias -> cicloDoWhile
sentencias -> condicionalSwitch
sentencias -> cliclofor
```



```
sentencias -> sentenciaSalida
sentencias -> asignacionvariables
sentencias -> declararVariables2
```

```
sentenciaSalida -> RETURN exp PUNTOCOMA
sentenciaSalida -> RETURN PUNTOCOMA
sentenciaSalida -> BREAK PUNTOCOMA
```

```
llamadaFuncion -> ID PA_A PA_C PUNTOCOMA
llamadaFuncion -> ID PA_A parametrosLlamada PA_C PUNTOCOMA
llamadaFuncion -> THIS PUNTO ID PA_A PA_C PUNTOCOMA
llamadaFuncion -> THIS PUNTO ID PA_A parametrosLlamada PA_C PUNTOCOMA
```

```
parametrosLlamada -> exp
parametrosLlamada -> exp COMA parametrosLlamadaP
```

```
parametrosLlamadaP -> exp COMA parametrosLlamadaP
parametrosLlamadaP -> exp
```

```
cliclofor -> FOR PA_A INT ID EQUAL exp PUNTOCOMA exp PUNTOCOMA ID agregarfor
PA_C LLA_A LLA_C
cliclofor -> FOR PA_A INT ID EQUAL exp PUNTOCOMA exp PUNTOCOMA ID agregarfor
PA_C LLA_A sentencias LLA_C
cliclofor -> FOR PA_A ID EQUAL exp PUNTOCOMA exp PUNTOCOMA ID agregarfor PA_C
LLA_A LLA_C
cliclofor -> FOR PA_A ID EQUAL exp PUNTOCOMA exp PUNTOCOMA ID agregarfor PA_C
LLA_A sentencias LLA_C
```

```
agregarfor -> SUMAR
agregarfor -> RESTAR
agregarfor -> EQUAL exp
```

```
ciclowhile -> WHILE PA_A exp PA_C LLA_A LLA_C
ciclowhile -> WHILE PA_A exp PA_C LLA_A sentencias LLA_C
```

```
cicloDoWhile -> DO LLA_A LLA_C WHILE PA_A exp PA_C PUNTOCOMA
cicloDoWhile -> DO LLA_A sentencias LLA_C WHILE PA_A exp PA_C PUNTOCOMA
```

```
condicionalif -> IF PA_A exp PA_C LLA_A sentencias LLA_C
condicionalif -> IF PA_A exp PA_C LLA_A LLA_C
```

```

condicionalif -> IF PA_A exp PA_C LLA_A sentencias LLA_C ELSE LLA_A LLA_C
condicionalif -> IF PA_A exp PA_C LLA_A sentencias LLA_C ELSE LLA_A
sentencias LLA_C
condicionalif -> IF PA_A exp PA_C LLA_A LLA_C ELSE LLA_A LLA_C
condicionalif -> IF PA_A exp PA_C LLA_A LLA_C ELSE LLA_A sentencias LLA_C

condicionalSwitch -> SWITCH PA_A exp PA_C LLA_A contenidoSwitch LLA_C
condicionalSwitch -> SWITCH PA_A exp PA_C LLA_A LLA_C

contenidoSwitch -> CASE exp DOSPUNTOS BREAK PUNTOCOMA
contenidoSwitch -> CASE exp DOSPUNTOS sentencias BREAK PUNTOCOMA
contenidoSwitch -> contenidoSwitch2 DEFAULT DOSPUNTOS sentencias
contenidoSwitch -> contenidoSwitch2 DEFAULT DOSPUNTOS
contenidoSwitch -> contenidoSwitch2 CASE exp DOSPUNTOS sentencias BREAK
PUNTOCOMA
contenidoSwitch -> contenidoSwitch2 CASE exp DOSPUNTOS BREAK PUNTOCOMA

contenidoSwitch2 -> contenidoSwitch2 CASE exp DOSPUNTOS sentencias BREAK
PUNTOCOMA
contenidoSwitch2 -> contenidoSwitch2 CASE exp DOSPUNTOS BREAK PUNTOCOMA
contenidoSwitch2 -> CASE exp DOSPUNTOS sentencias BREAK PUNTOCOMA
contenidoSwitch2 -> CASE exp DOSPUNTOS BREAK PUNTOCOMA

asig -> exp
asig -> NEW OBJ PA_A PA_C


exp -> exp OR t
exp -> t

t -> t AND f
t -> f

f -> f IGUAL g
f -> f DIFERENTE g
f -> g

g -> g MENOR_QUE h
g -> g MAYOR_QUE h
g -> g MAYOR_IGUAL h
g -> g MENOR_IGUAL h

```

g -> h

h -> h MAS i

h -> h MENOS i

h -> i

i -> i MUL j

i -> i DIV j

i -> j

j -> NOT k

j -> MENOS k

j -> MAS k

j -> k

k -> l SUMAR

k -> l RESTAR

k -> l

l -> ID

l -> THIS PUNTO ID

l -> llamadaFuncionExp

l -> TRUE

l -> FALSE

l -> DECIMAL

l -> ENTERO

l -> STRING

l -> CHAR_VAL

l -> PA_A exp PA_C

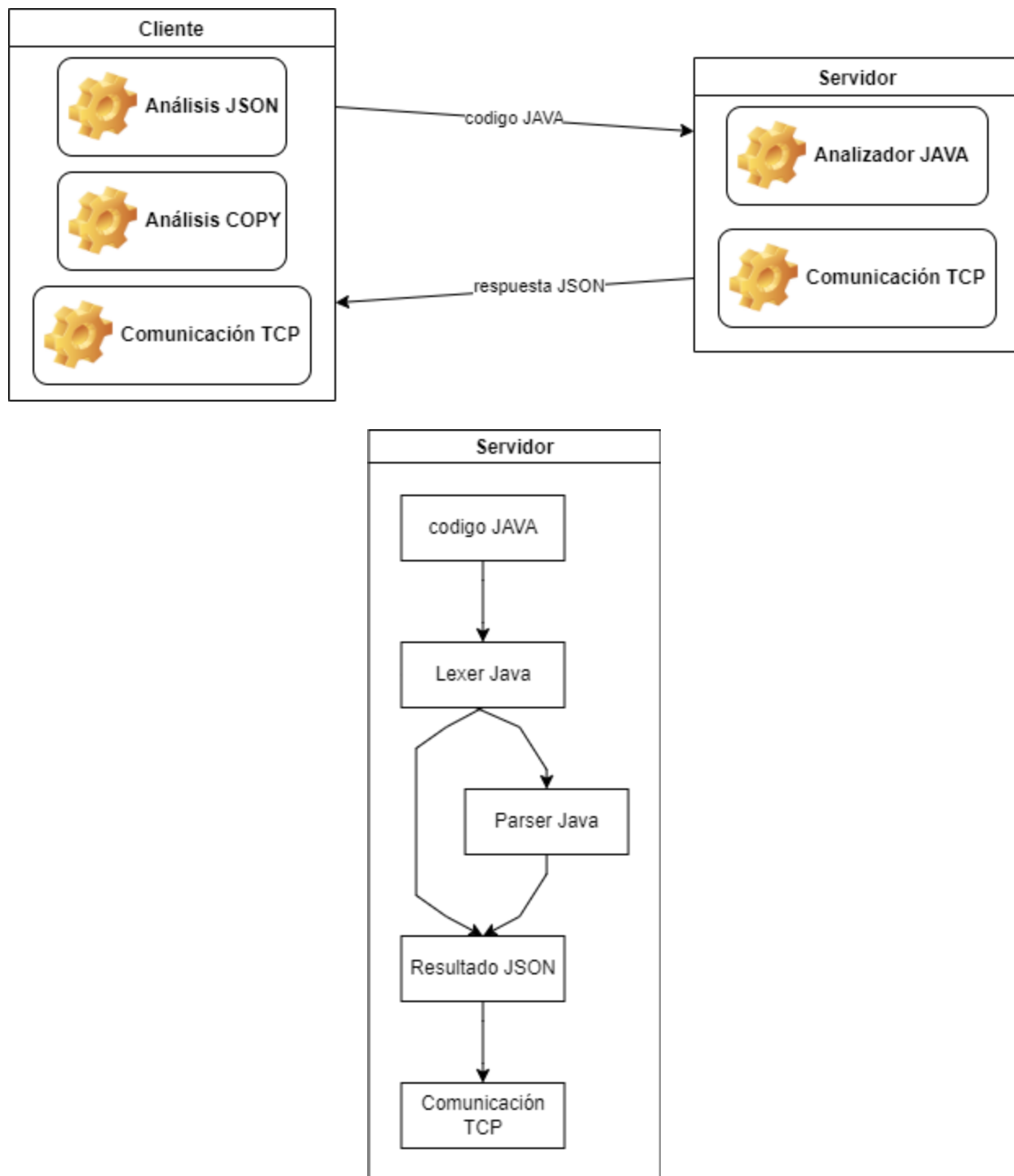
llamadaFuncionExp -> ID PA_A PA_C

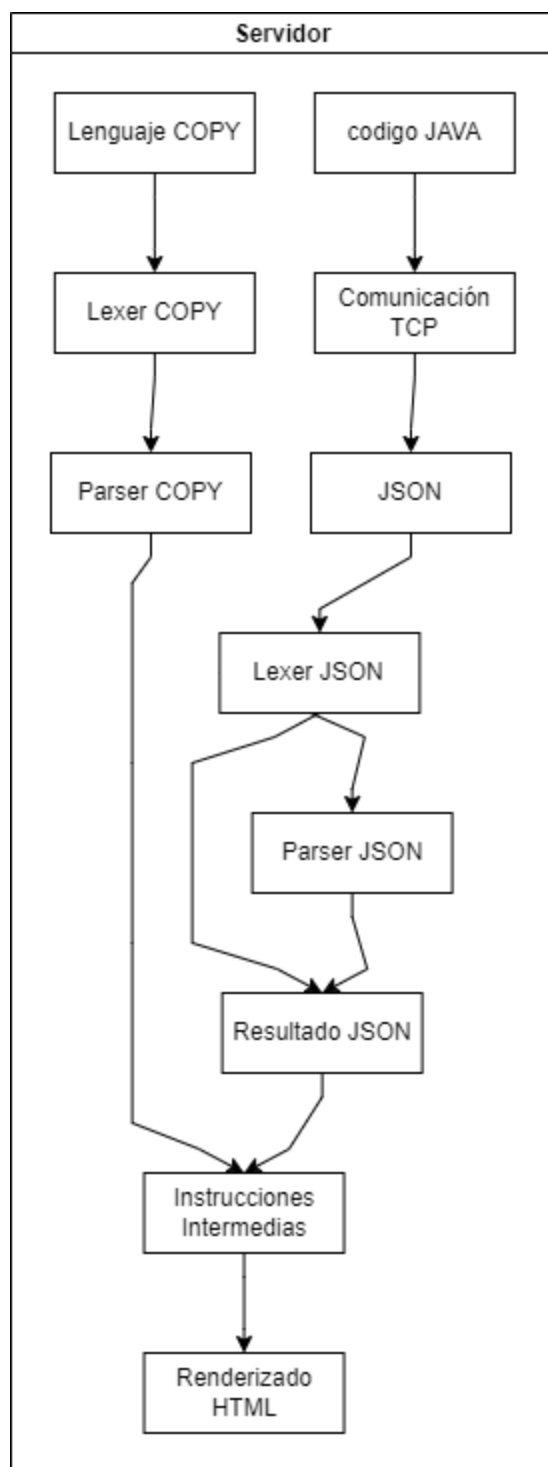
llamadaFuncionExp -> ID PA_A parametrosLlamada PA_C

llamadaFuncionExp -> THIS PUNTO ID PA_A PA_C

llamadaFuncionExp -> THIS PUNTO ID PA_A parametrosLlamada PA_C

Organización del Proyecto:





Estructura del Proyecto en App Cliente:

Modulos:

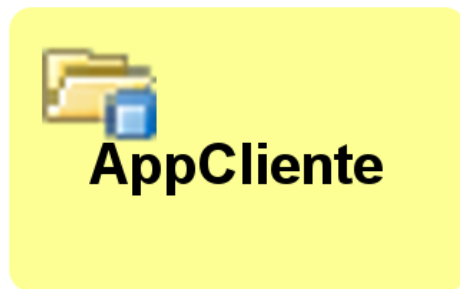
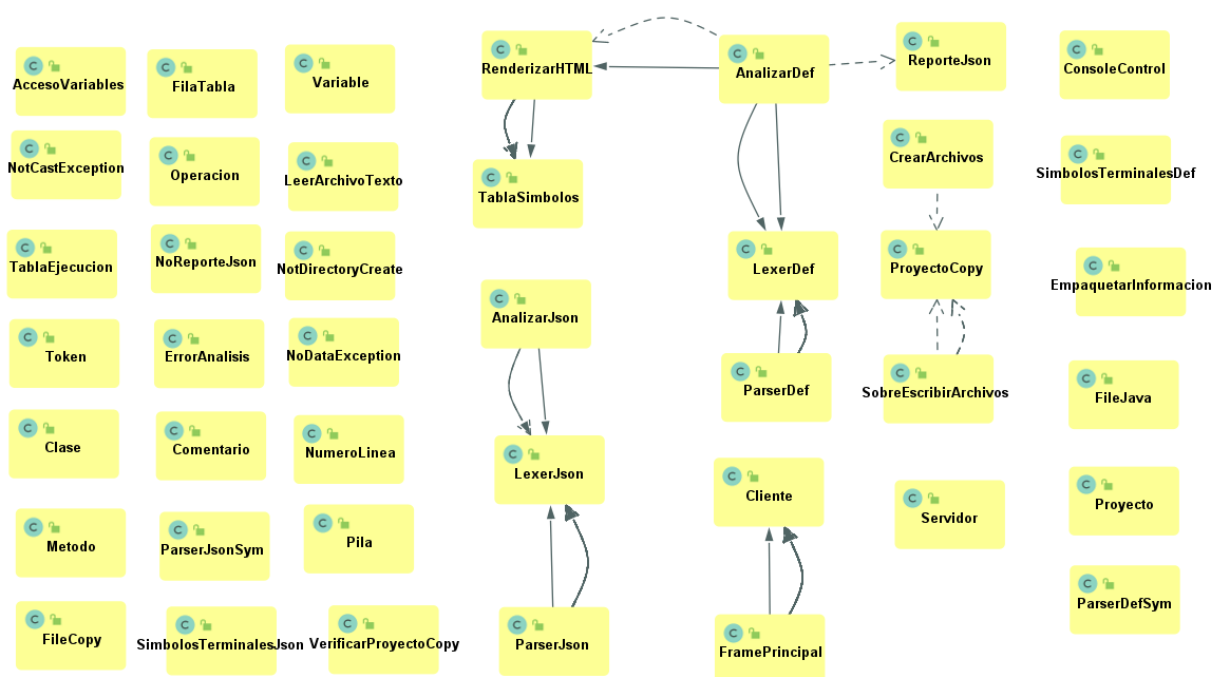


Diagrama de Clases:



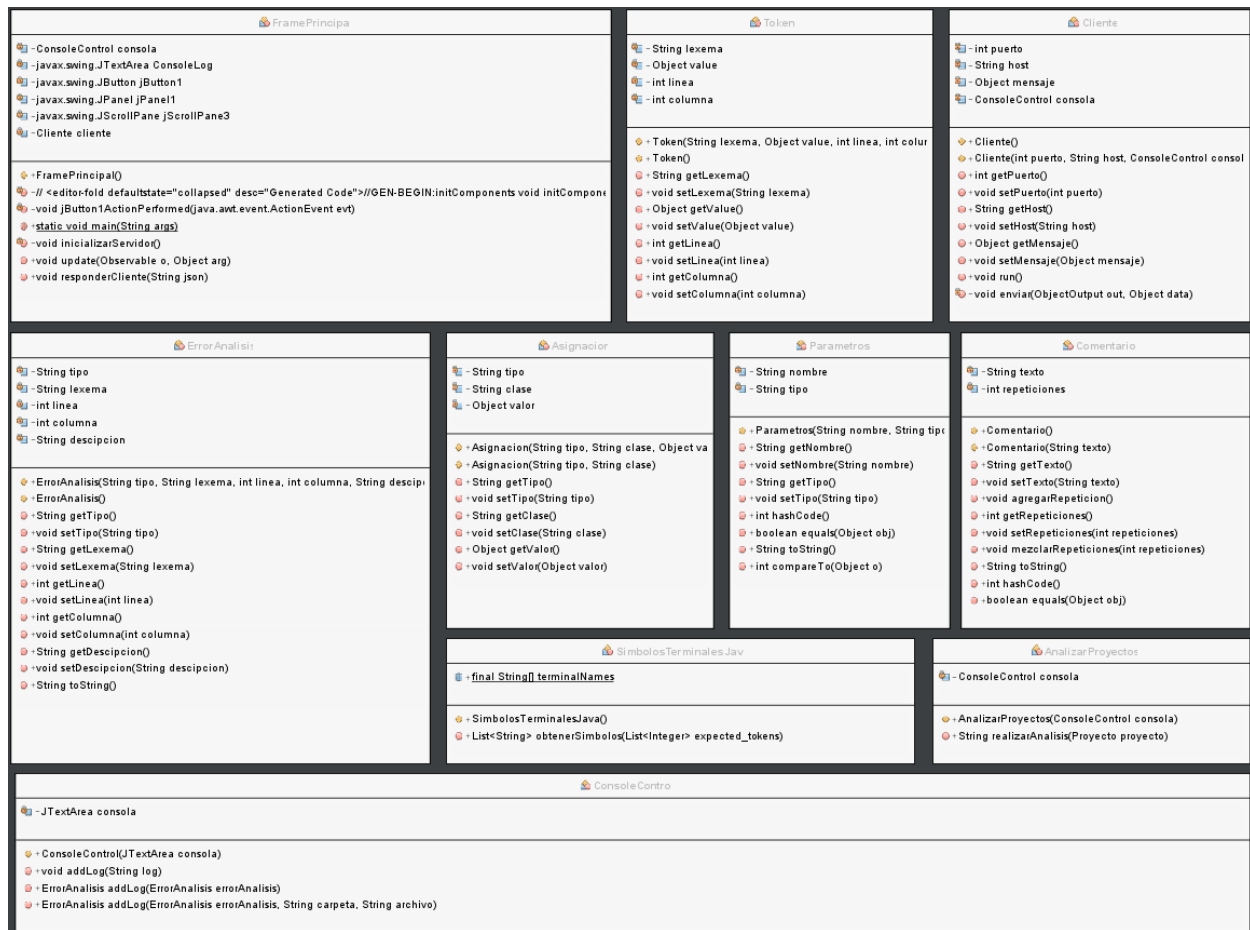
Es presentado de esta manera ya que su descripción general completa no permitirá visualizarlos de una manera correcta, por ende adelante de esta sección se adjunta su descripción completa.

Contenido de las clases utilizadas:

Análizar Java
<ul style="list-style-type: none"> - ArrayList<ErrorAnalysis> errores - LexerJava lexerJava - ParserJava parserJava - ConsoleControl consola
<ul style="list-style-type: none"> + AnalizarJava(ConsoleControl consola, ArrayList<ErrorAnalysis> errores) + void ejecutar(String texto, TablaSimbolos tablaSimbolos, ArrayList<Clase> clases, ArrayList<Metodo> metodos, ArrayList<Comentario> comentarios, String carpetaFuente, String archivoAnalizado) + ArrayList<ErrorAnalysis> getErrores()
Comparar Proyecto
<ul style="list-style-type: none"> - ResultadoAnálisis resultadoCarpeta1 - ResultadoAnálisis resultadoCarpeta2
<ul style="list-style-type: none"> + CompararProyectos(ResultadoAnálisis resultadoCarpeta1, ResultadoAnálisis resultadoCarpeta2) + ResultadoAnálisis getResultadoCarpeta1() + void setResultadoCarpeta1(ResultadoAnálisis resultadoCarpeta1) + ResultadoAnálisis getResultadoCarpeta2() + void setResultadoCarpeta2(ResultadoAnálisis resultadoCarpeta2) + String resultadosJson() - String calcularScore(int totalVariables, int varsRepetidas, int totalMetodos, int metodosRepetidos, int totalClases, int clasesRepetidas, int totalComen, int comenrepetidos) - String toJsonText(ReporteJson reporteJson) - void obtenerMetodosRepetidos(ArrayList<Metodo> metodos) - Metodo buscarListaMetodos(ArrayList<Metodo> metodos, Metodo metodo) - void obtenerComentariosRepetidos(ArrayList<Comentario> comentarios) - Comentario buscarListaComentario(ArrayList<Comentario> comentarios, Comentario comentario) - void obtenerClasesRepetidos(ArrayList<Clase> clases) - Clase buscarListaClases(ArrayList<Clase> clases, Clase clase) - void obtenerVariablesRepetidas(ArrayList<FilaTablaSimbolos> variables) - void convertirVariablesRepetidas(ArrayList<FilaTablaSimbolos> variables, ArrayList<Variable> varFinal) - String obtenerFuncionesVariables(ArrayList<String> funciones) - ArrayList<FilaTablaSimbolos> repitenciaAmbosProyectos(ArrayList<FilaTablaSimbolos> filas, ArrayList<FilaTablaSimbolos> filas0) - ArrayList<FilaTablaSimbolos> repitenciaTablaFinal(ArrayList<FilaTablaSimbolos> variables) - FilaTablaSimbolos buscarVariable(String nombre, String tipo, ArrayList<FilaTablaSimbolos> resultfinal) - int contarVariablesFinal(ArrayList<FilaTablaSimbolos> repitencia1, ArrayList<FilaTablaSimbolos> repitencia2) - ArrayList<Metodo> repitenciaMetodosProyectos(ArrayList<Metodo> metodos, ArrayList<Metodo> metodos0) - ArrayList<Metodo> repitenciaMetodosProyectos(ArrayList<Metodo> metodos) - int contarMetodosRepetidos(ArrayList<Metodo> repitenciaMetodo1, ArrayList<Metodo> repitenciaMetodo2) - ArrayList<Clase> repitenciaClasesProyecto(ArrayList<Clase> clases, ArrayList<Clase> clases0) - ArrayList<Clase> repitenciaClasesProyecto(ArrayList<Clase> clases) - int contarClasesRepetidas(ArrayList<Clase> repitenciaClase1, ArrayList<Clase> repitenciaClase2) - ArrayList<Comentario> repitenciaComentariosProyectos(ArrayList<Comentario> comentarios, ArrayList<Comentario> comentarios0) - ArrayList<Comentario> repitenciaComentariosProyectos(ArrayList<Comentario> comentarios) - int contarComentariosRepetidos(ArrayList<Comentario> repitenciaComentarios1, ArrayList<Comentario> repitenciaComentarios2)

<p>FilaTablaSymbolos</p> <ul style="list-style-type: none"> - String nombre - String tipo - ArrayList<String> funciones - int repeticiones <p> + FilaTablaSymbolos() + FilaTablaSymbolos(String nombre, String tipo, ArrayList<String> funciones) + FilaTablaSymbolos(String nombre, String tipo) + FilaTablaSymbolos(String nombre) + String getNombre() + void setNombre(String nombre) + String getTipo() + void setTipo(String tipo) + ArrayList<String> getFunciones() + void setFunciones(ArrayList<String> funciones) + void agregarFuncion(String nombreFuncion) + void mergeFunciones(ArrayList<String> nuevasFunciones) + void agregarRepeticion() + void mezclarRepeticiones(int repeticiones) + int hashCode() + boolean equals(Object obj) + boolean igualdadTipoNombre(Object obj) + String toString() + int getRepeticiones() + void setRepeticiones(int repeticiones) </p>	<p>Metodo</p> <ul style="list-style-type: none"> - String nombre - String tipo - ArrayList<Parametros> parametros - int repeticiones <p> + Metodo() + Metodo(String nombre, String tipo) + Metodo(String nombre, String tipo, ArrayList<Parametros> parametros) + String getNombre() + void setNombre(String nombre) + String getTipo() + void setTipo(String tipo) + ArrayList<Parametros> getParametros() + void setParametros(ArrayList<Parametros> parametros) + int hashCode() + boolean equals(Object obj) + boolean comparanMetodos(Metodo metodo) + void agregarRepeticion() + int getRepeticiones() + void setRepeticiones(int repeticiones) + void mezclarRepeticiones(int repeticiones) + String toString() </p>	<p>Tabla Simbolos</p> <ul style="list-style-type: none"> + final String INT + final String DOUBLE + final String CHAR + final String STRING + final String BOOLEAN + final String OBJECT + final String VOID + final String COMODIN + final String ERROR + ArrayList<FilaTablaSymbolos> filas <p> + TablaSimbolos() + ArrayList<FilaTablaSymbolos> buscarNombre(String nombre) + FilaTablaSymbolos buscarNombreTipo(String nombre, String tipo) + ArrayList<FilaTablaSymbolos> getFilas() + void resetTabla() + void imprimirTabla() + int numeroVariables() </p>
<p>Operacion Cas</p> <p> + OperacionCas() + static boolean validSumaResta(Asignacion type) + static String expectedSumaResta() + static boolean validUniarNegPosi(Asignacion type) + static String expectedUniarNegPosi() + static boolean validUniarBool(Asignacion type) + static String expectedUniarBool() + static boolean validArithmeticExpresion(Asignacion type) + static String newTypeSum(Asignacion left, Asignacion right) + static String expectedSum() + static String newTypeResMulDiv(Asignacion left, Asignacion right) + static String expectedResMulDiv() + static String newMayorMenor(Asignacion left, Asignacion right) + static String newComparacion(Asignacion left, Asignacion right) + static String newAndOr(Asignacion left, Asignacion right) </p>	<p>Proyecto</p> <ul style="list-style-type: none"> - String nombreCarpeta1 - ArrayList<FileJava> carpeta1 - String nombreCarpeta2 - ArrayList<FileJava> carpeta2 <p> + Proyecto(ArrayList<FileJava> carpeta1, ArrayList<FileJava> carpeta2) + Proyecto() + ArrayList<FileJava> getCarpeta1() + void setCarpeta1(ArrayList<FileJava> carpeta1) + ArrayList<FileJava> getCarpeta2() + void setCarpeta2(ArrayList<FileJava> carpeta2) + String getNombreCarpeta1() + void setNombreCarpeta1(String nombreCarpeta1) + String getNombreCarpeta2() + void setNombreCarpeta2(String nombreCarpeta2) </p>	<p>Variable</p> <ul style="list-style-type: none"> - String nombre - String tipo - String funcion <p> + Variable() + Variable(String nombre, String tipo, String funcion) + String getNombre() + void setNombre(String nombre) + String getTipo() + void setTipo(String tipo) + String getFuncion() + void setFuncion(String funcion) + String toString() + int hashCode() + boolean equals(Object obj) </p>

<p> Reporte Jsor</p> <ul style="list-style-type: none"> -String score -ArrayList<Clase> clases -ArrayList<Variable> variables -ArrayList<Metodo> metodos -ArrayList<Comentario> comentarios <p> +ReporteJsor() +ReporteJsor(String score, ArrayList<Clase> clases, ArrayList<Variable> variables, ArrayList<Metodo> metodos, ArrayList<Comentario> comentarios) +String getScore() +void setScore(String score) +ArrayList<Clase> getClases() +void setClases(ArrayList<Clase> clases) +ArrayList<Variable> getVariables() +void setVariables(ArrayList<Variable> variables) +ArrayList<Metodo> getMetodos() +void setMetodos(ArrayList<Metodo> metodos) +ArrayList<Comentario> getComentarios() +void setComentarios(ArrayList<Comentario> comentarios) +void imprimirReporte() +String toString() +int hashCode() +boolean equals(Object obj) </p>	<p> Clase</p> <ul style="list-style-type: none"> -String nombre -ArrayList<String> funciones -int repeticiones <p> +Clase() +Clase(String nombre) +Clase(String nombre, ArrayList<String> funciones) +String getNombre() +void setNombre(String nombre) +void agregarRepeticion() +int getRepeticiones() +void setRepeticiones(int repeticiones) +void mezclarRepeticiones(int repeticiones) +int hashCode() +boolean equals(Object obj) +boolean compararClases(Object obj) +ArrayList<String> getFunciones() +void setFunciones(ArrayList<String> funciones) +String toString() </p>
<p> Resultado Analisis</p> <ul style="list-style-type: none"> -ArrayList<Clase> clases -ArrayList<Metodo> metodos -ArrayList<Comentario> comentarios -TablaSimbolos tablaSimbolos <p> +ResultadoAnalisis(ArrayList<Clase> clases, ArrayList<Metodo> metodos, ArrayList<Comentario> comentarios, TablaSimbolos tablaSimbolos) +ArrayList<Clase> getClases() +void setClases(ArrayList<Clase> clases) +ArrayList<Metodo> getMetodos() +void setMetodos(ArrayList<Metodo> metodos) +ArrayList<Comentario> getComentarios() +void setComentarios(ArrayList<Comentario> comentarios) +TablaSimbolos getTablaSimbolos() +void setTablaSimbolos(TablaSimbolos tablaSimbolos) +int numeroDeVariables() +int numeroClases() +int numeroComentarios() +int numeroMetodos() +void imprimirClases() +void imprimirMetodos() +void imprimirComentarios() </p>	<p> File Java</p> <ul style="list-style-type: none"> -String code -String name -String extension <p> +FileJava() +FileJava(String code, String name, String extension) +String getText() +void setText(String code) +String getName() +void setName(String name) +String getExtension() +void setExtension(String extension) +void printCode() </p> <p> Servido</p> <ul style="list-style-type: none"> -int puerto -ConsoleControl consola <p> +Servidor() +Servidor(int puerto, ConsoleControl consola) +int getPuerto() +void setPuerto(int puerto) +void run() </p>



Estructura del Proyecto en Servidor:

Modulos:

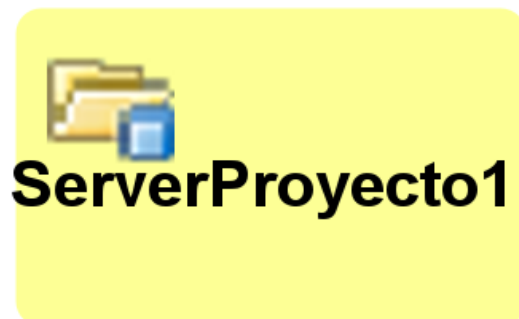
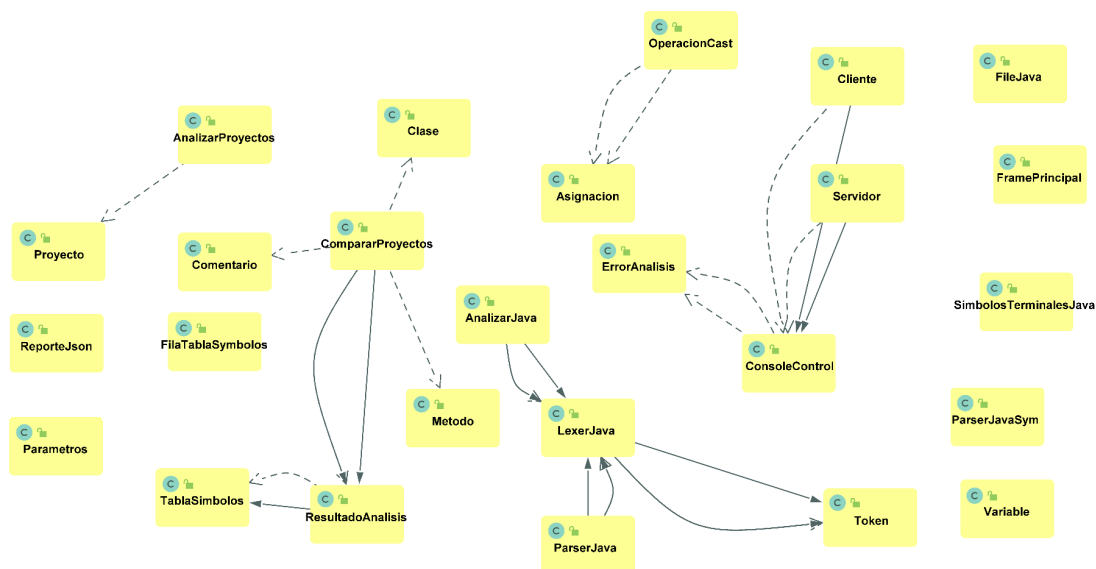
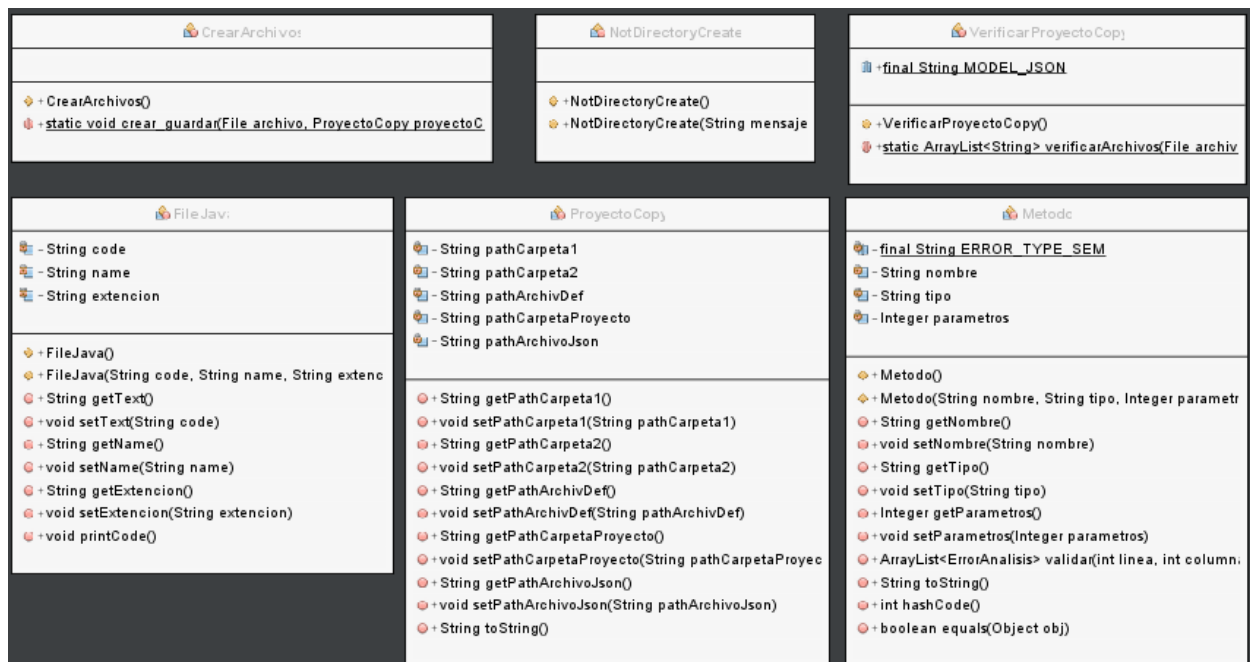
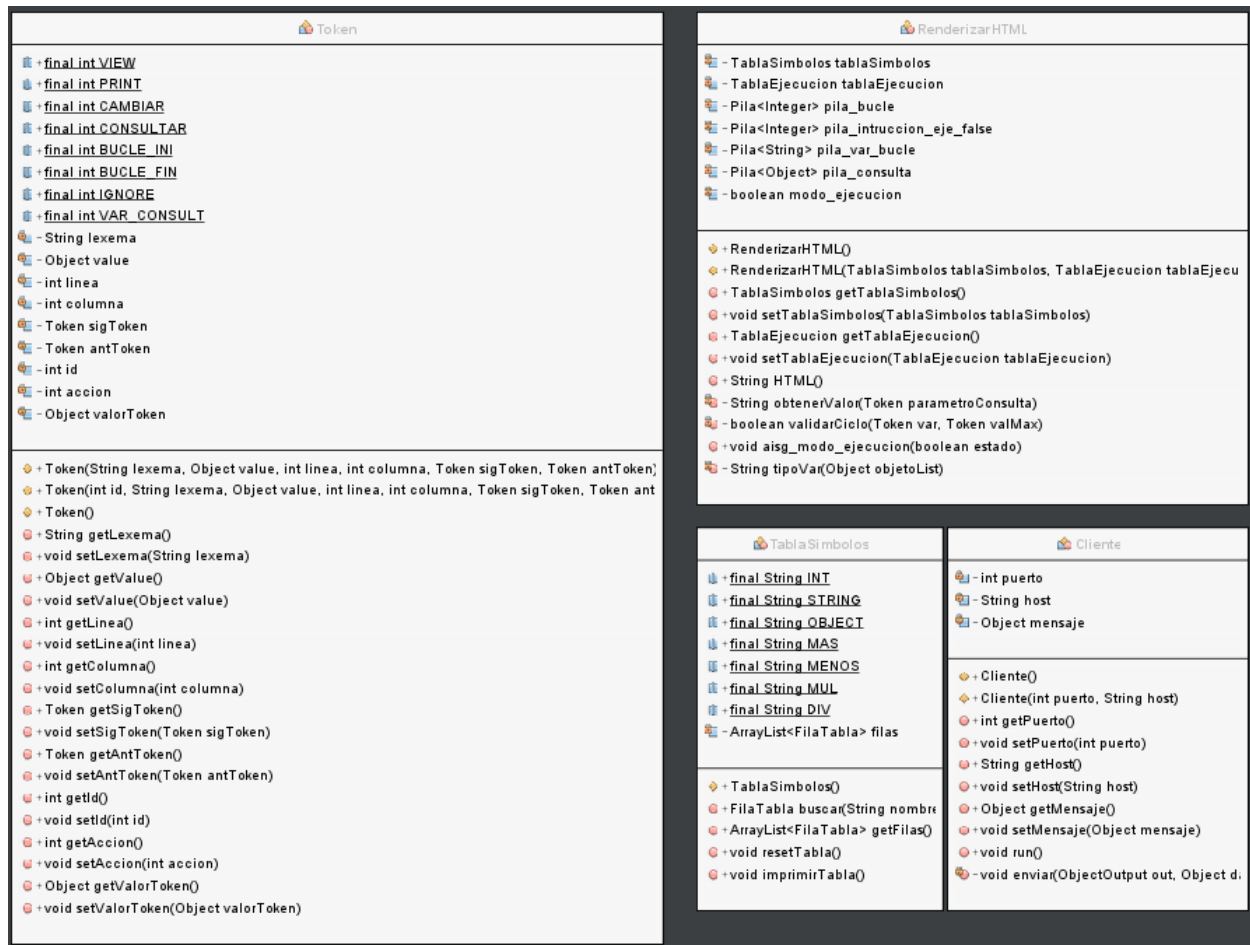


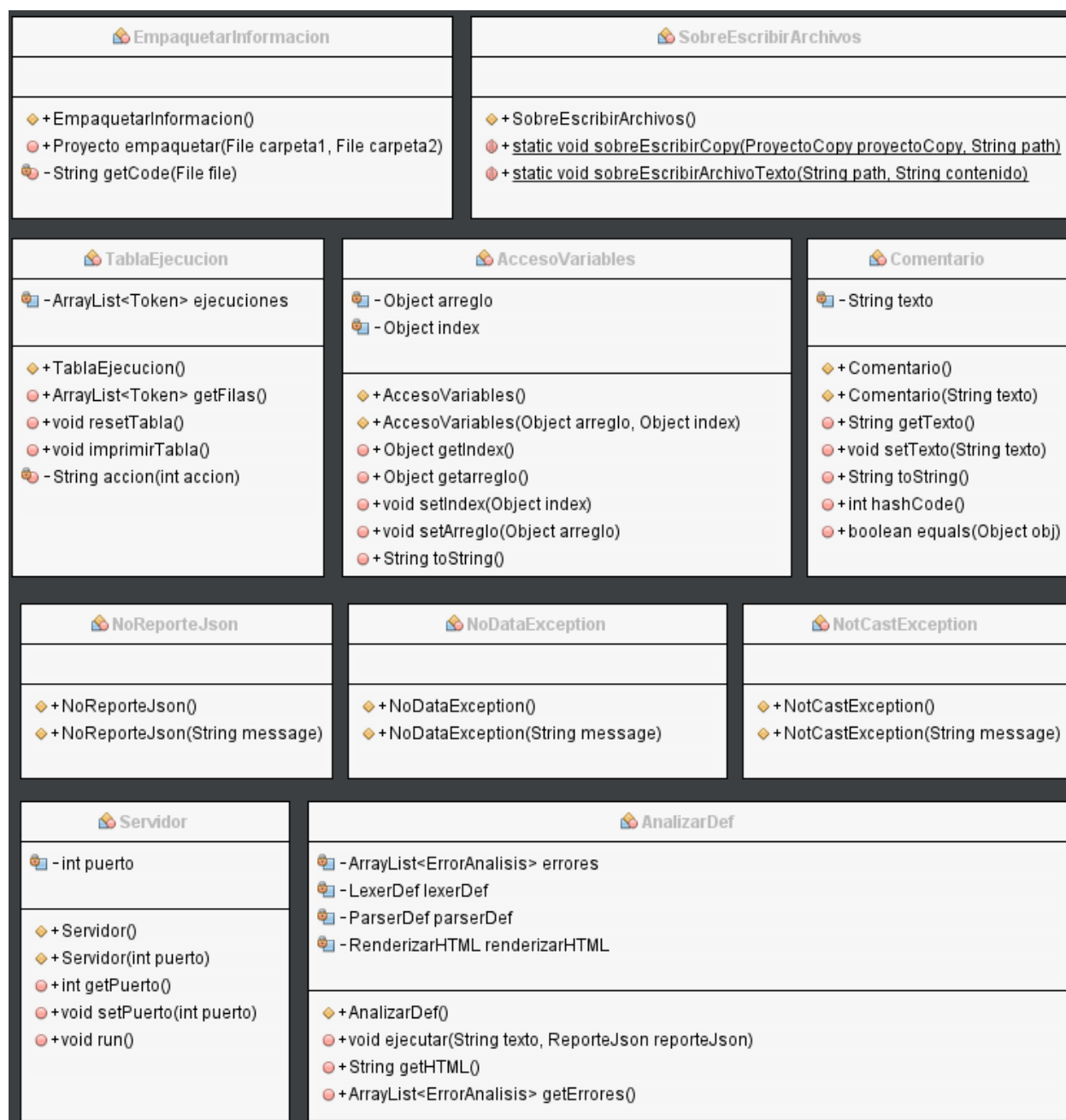
Diagrama de Clases:

Es presentado de esta manera ya que su descripción general completa no permitirá visualizarlos de una manera correcta, por ende adelante de esta sección se adjunta su descripción completa.



Contenido de las clases utilizadas:





FileCopy	NumeroLinea
<ul style="list-style-type: none"> -String pathProyecto1 -String pathProyecto2 -String pathFileDef -String pathFileReporteJson 	<ul style="list-style-type: none"> +final float LEFT +final float CENTER +final float RIGHT -Color color1 -final Border OUTER -final int HEIGHT -JTextComponent component -boolean updateFont -int borderGap -Color currentLineForeground -float digitAlignment -int minimumDisplayDigits -int lastDigits -int lastHeight -int lastLine -HashMap<String, FontMetrics> fonts
<ul style="list-style-type: none"> +FileCopy() +FileCopy(String pathProyecto1, String pathProyecto2, String pathFileDef, String pathFileReporteJson) +String getPathProyecto1() +void setPathProyecto1(String pathProyecto1) +String getPathProyecto2() +void setPathProyecto2(String pathProyecto2) +String getPathFileDef() +void setPathFileDef(String pathFileDef) +String getPathFileReporteJson() +void setPathFileReporteJson(String pathFileReporteJson) +String toString() +int hashCode() +boolean equals(Object obj) 	<ul style="list-style-type: none"> +NumeroLinea(JTextComponent component) +NumeroLinea(JTextComponent component, int minimumDisplayDigits) +boolean getUpdateFont() +void setUpdateFont(boolean updateFont) +int getBorderGap() +void setBorderGap(int borderGap) +Color getCurrentLineForeground() +void setCurrentLineForeground(Color currentLineForeground) +float getDigitAlignment() +void setDigitAlignment(float digitAlignment) +int getMinimumDisplayDigits() +void setMinimumDisplayDigits(int minimumDisplayDigits) -void setPreferredWidth() +void paintComponent(Graphics g) -boolean isCurrentLine(int rowStartOffset) +String getTextLineNumber(int rowStartOffset) -int getOffsetX(int availableWidth, int stringWidth) -int getOffsetY(int rowStartOffset, FontMetrics fontMetrics) +void caretUpdate(CaretEvent e) +void changedUpdate(DocumentEvent e) +void insertUpdate(DocumentEvent e) +void removeUpdate(DocumentEvent e) -void documentChanged() +void propertyChange(PropertyChangeEvent evt)
ErrorAnalysis	
<ul style="list-style-type: none"> -String tipo -String lexema -int linea -int columna -String descripcion 	
<ul style="list-style-type: none"> +ErrorAnalysis(String tipo, String lexema, int linea, int columna, String descripcion) +ErrorAnalysis() +String getTipo() +void setTipo(String tipo) +String getLexema() +void setLexema(String lexema) +int getLinea() +void setLinea(int linea) +int getColumna() +void setColumna(int columna) +String getDescripcion() +void setDescripcion(String descripcion) +String toString() 	

<p>Pila</p> <ul style="list-style-type: none"> - NodoPila<T> raiz - int size <ul style="list-style-type: none"> + Pila() + void push(T data) + T pop() + T peek() + boolean isEmpty() + int getSize() + ArrayList<T> toArrayList() + T[] toArray(Class<T> arrayType) + void imprimirPila() 	<p>Variable</p> <ul style="list-style-type: none"> - final String ERROR_TYPE_SEM - String nombre - String tipo - String funcion <ul style="list-style-type: none"> + Variable() + Variable(String nombre, String tipo, String funcion) + String getNombre() + void setNombre(String nombre) + String getTipo() + void setTipo(String tipo) + String getFuncion() + void setFuncion(String funcion) + ArrayList<ErrorAnalisis> validar(int linea, int column) + String toString() + int hashCode() + boolean equals(Object obj) 	<p>Operacion</p> <ul style="list-style-type: none"> - String tipo - Object valor <ul style="list-style-type: none"> + Operacion(String tipo, Object valor) + Operacion() + String getTipo() + void setTipo(String tipo) + Object getValor() + void setValor(Object valor) + static String autoCast(String tipoval1, String operacion, String tipo) + static Object sumaTerminos(Operacion val1, Operacion val2) + String toString() + int hashCode() + boolean equals(Object obj)
<p>FilaTabla</p> <ul style="list-style-type: none"> - String nombre - String tipo - Object valor <ul style="list-style-type: none"> + FilaTabla() + FilaTabla(String nombre, String tipo, Object va) + String getNombre() + void setNombre(String nombre) + String getTipo() + void setTipo(String tipo) + Object getValor() + void setValor(Object valor) + String toString() + int hashCode() + boolean equals(Object obj) 	<p>AnalizarJson</p> <ul style="list-style-type: none"> - ArrayList<ErrorAnalisis> errores - LexerJson lexerJson - ParserJson parserJson <ul style="list-style-type: none"> + AnalizarJson() + void ejecutar(String texto) + ArrayList<ErrorAnalisis> getErrores() + ReporteJson getReporteJson() <p>LeerArchivoTexto</p> <ul style="list-style-type: none"> + LeerArchivoTexto() + static String leerArchivo(String path) 	<p>Proyecto</p> <ul style="list-style-type: none"> - String nombreCarpeta1 - ArrayList<FileJava> carpeta1 - String nombreCarpeta2 - ArrayList<FileJava> carpeta2 <ul style="list-style-type: none"> + Proyecto(ArrayList<FileJava> carpeta1, ArrayList<FileJava> carpeta2) + Proyecto() + ArrayList<FileJava> getCarpeta1() + void setCarpeta1(ArrayList<FileJava> carpeta1) + ArrayList<FileJava> getCarpeta2() + void setCarpeta2(ArrayList<FileJava> carpeta2) + String getNombreCarpeta1() + void setNombreCarpeta1(String nombreCarpeta1) + String getNombreCarpeta2() + void setNombreCarpeta2(String nombreCarpeta2)

Reporte Jsor

- String score
- ArrayList<Clase> clases
- ArrayList<Variable> variables
- ArrayList<Metodo> metodos
- ArrayList<Comentario> comentarios

- +ReporteJson()
- +ReporteJson(String score, ArrayList<Clase> clases, ArrayList<Variable> variables, ArrayList<Metodo> metodos, ArrayList<Comentario> comenta
- +String getScore()
- +void setScore(String score)
- +ArrayList<Clase> getClases()
- +void setClases(ArrayList<Clase> clases)
- +ArrayList<Variable> getVariables()
- +void setVariables(ArrayList<Variable> variables)
- +ArrayList<Metodo> getMetodos()
- +void setMetodos(ArrayList<Metodo> metodos)
- +ArrayList<Comentario> getComentarios()
- +void setComentarios(ArrayList<Comentario> comentarios)
- +void imprimirReporte()
- +ArrayList<ErrorAnalisis> valida(int linea, int columna)
- +String toString()
- +int hashCode()
- +boolean equals(Object obj)