

Manual Técnico

Indice

Manual Técnico.....	1
Ambiente de desarrollo:.....	1
Gramáticas.....	2
Gramática de las solicitudes(lenguaje Índigo).....	2
Gramática de las respuestas del servidor al cliente(lenguaje igual al Índigo).....	14
Gramática de los reportes(lenguaje SQForm).....	19
Gramática de almacenamiento de los usuarios(lenguaje Usuario)	24
Gramática de almacenamiento de los formularios(lenguaje Formulario).....	27
Diagrama de Clases de la aplicación cliente:.....	35
Diagrama de Clases de la aplicación Web:.....	36

Ambiente de desarrollo:

JVM: JDK 14

Entorno de desarrollo: Apache NetBeans IDE 11.3

Sistema operativo: Windows 10

Versión del sistema operativo: 20H2

Memoria Ram : 12Gb

CPU: Intel Core i7-7700HQ

Servidor: Apache Tomcat 9

Dependencias:

- java-cup-runtime
- jackson-databind
- Json
- javax-web-api 7.0
- Jstl 1.2

Gramáticas

Gramática de las solicitudes(lenguaje Índigo)

Gramática del analizador léxico:

decimales

decimal=[1-9][0-9]*[.]{entero}|[0][.]{entero}

Número enteros

entero=[0-9]+

number = ({entero}|{decimal})

letra = [a-zA-Z]

Palabra sin espacios

string = ({letra}|{number}|[_]|[-]|[\$]|[\.])+

Palabra con espacios

stringSpace = ({letra}|{number})+({string}|{whiteSpace})*

Consultas

cont_consulta = ("\"SELECT" | "\"" + {whiteSpace} + "SELECT") {consulta} ("]" | "]" + {whiteSpace} + "\"")

consulta = ([^]] | "]" + [^ \f\t\r\n\]" | "]" + ({whiteSpace}) [^]\")*

consulta_N = ("CONSULTA")[-]({entero})

Los ID de componentes y formularios

CONT_ID = ([\${}][_][-])({letra}|{number}|[\${}][-][_])+

El contenido del parámetro opciones de los componentes

cont_opciones = (({stringSpace}|{string})("")({whiteSpace})?)+({stringSpace}|{string})

Para la URL

cont_url = ({path_relativo}|{http}|{https})?([\\V])?(({stringSpace})[\\V])+({stringSpace})?

http = ("http:/")

https = ("https:/")

path_relativo = (({letra})(":"))

Palabras reservadas sin importar las mayúsculas o minúsculas

ini_solicitudes = {ini_solicitud}[eE][sS]

ini_solicitud = [iI][nN][iI][_][sS][oO][lL][iI][cC][iI][tT][uU][dD]

fin_solicitudes = {fin_solicitud}[eE][sS]

fin_solicitud = [fF][iI][nN][_][sS][oO][lL][iI][cC][iI][tT][uU][dD]

Todo tipo de espacios en blanco

lineTerminator = \r\n\r\n

whiteSpace = {lineTerminator} | [\t\f]

Palabras reservadas:

CREAR_USUARIO
MODIFICAR_USUARIO
ELIMINAR_USUARIO
LOGIN_USUARIO
NUEVO_FORMULARIO
ELIMINAR_FORMULARIO
MODIFICAR_FORMULARIO
AGREGAR_COMPONENTE
ELIMINAR_COMPONENTE
MODIFICAR_COMPONENTE
CONSULTAR_DATOS
CREDENCIALES_USUARIO
PARAMETROS_FORMULARIO
PARAMETROS_COMPONENTE
CONSULTAS
ID
TITULO
TEXTO_VISIBLE
OPCIONES
FORMULARIO
URL

Símbolos:

"

{
}
:
[
]
,
<
>
!

Gramática para el analizador sintáctico:

Símbolo terminales:

MENQ,ADMIRACION,INI_SOLICITUDES,MAQ, FIN_SOLICITUDES,INI_SOLICITUD, DOS_PUNTOS,
CONSULTAR_DATOS,FIN_SOLICITUD,CREAR_USUARIO, MODIFICAR_USUARIO, ELIMINAR_USUARIO,
LOGIN_USUARIO, NUEVO_FORMULARIO, ELIMINAR_FORMULARIO,MODIFICAR_FORMULARIO,
AGREGAR_COMPONENTE, ELIMINAR_COMPONENTE, MODIFICAR_COMPONENTE,
CREDENCIALES_USUARIO, PARAMETROS_FORMULARIO,PARAMETROS_COMPONENTE, COMILLAS,
STRING,CONSULTAS, LLAI, LLAD, CORI, CORD, COMA,ID, CONT_ID, STRING_SPACE,TITULO,
FORMULARIO,TEXTO_VISIBLE,OPCIONES,CONT_OPCIONES,URL,CONT_URL, CONSULTA_N,
CONT_CONSULTA

Símbolos no terminales:

inicio,sig_inicio,solicitudes,fin_solicitudes,solicitud, sig_solicitud,fin_solicitud,
nombre_solicitud_usuario,nombre_solicitud_formulario,nombre_solicitud_componente,contenido_usuario,

contenido_formulario, contenido_componente, contenido_consulta,
bloque_parametro_comillas_credenciales_usuario, bloque_parametro_comillas_parametros_formulario,
bloque_parametro_comillas_parametros_componente, paquetes_usuario, paquetes_formulario, paquetes_componente,
paquetes_consulta, bloques_usuario, bloques_formulario, bloques_componente, bloques_consulta, bloque_usuario,
bloque_formulario, bloque_componente, bloque_consulta, parametros_usuario, parametros_formulario,
parametros_componente, parametros_consulta, parametro_usuario, parametro_formulario, parametro_componente,
datos, ids, cont_ids, titulos, cont_string_spaces, formulario, texto_visible, opciones, cont_opciones, url, cont_url,
parametro_consulta, consulta_n, con_consulta

Gramática:

```
inicio ::=  MENQ ADMIRACION sig_inicio  
          |error sig_inicio  
          ;  
sig_inicio ::=  INI_SOLICITUDES MAQ solicitudes fin_solicitudes  
               |INI_SOLICITUD DOS_PUNTOS COMILLAS sig_solicitud  
               |error  
               ;  
solicitudes ::= solicitudes solicitud  
               |solicitud  
               |error sig_solicitud  
               ;  
fin_solicitudes ::= MENQ ADMIRACION FIN_SOLICITUDES MAQ  
                  |error
```

;

solicitud ::= MENQ ADMIRACION INI_SOLICITUD DOS_PUNTOS COMILLAS sig_solicitud

;

sig_solicitud ::= nombre_solicitud_usuario COMILLAS MAQ contenido_usuario fin_solicitud
| nombre_solicitud_formulario COMILLAS MAQ contenido_formulario fin_solicitud
| nombre_solicitud_componente COMILLAS MAQ contenido_componente fin_solicitud
| CONSULTAR_DATOS COMILLAS MAQ contenido_consulta fin_solicitud
| error fin_solicitud

;

fin_solicitud ::= MENQ FIN_SOLICITUD ADMIRACION MAQ

| error

;

nombre_solicitud_usuario ::= CREAR_USUARIO

| MODIFICAR_USUARIO

| ELIMINAR_USUARIO

| LOGIN_USUARIO

;

nombre_solicitud_formulario ::= NUEVO_FORMULARIO

| ELIMINAR_FORMULARIO

| MODIFICAR_FORMULARIO

;

nombre_solicitud_componente ::= AGREGAR_COMPONENTE
| ELIMINAR_COMPONENTE
| MODIFICAR_COMPONENTE;

contenido_usuario ::= LLAI bloque_parametro_comillas_credenciales_usuario DOS_PUNTOS paquetes_usuario LLAD
| error paquetes_usuario LLAD;

contenido_formulario ::= LLAI bloque_parametro_comillas_parametros_formulario DOS_PUNTOS paquetes_formulario
LLAD

| error

;

contenido_componente ::= LLAI bloque_parametro_comillas_parametros_componente DOS_PUNTOS
paquetes_componente LLAD

| error

;

contenido_consulta ::= LLAI COMILLAS CONSULTAS COMILLAS DOS_PUNTOS paquetes_consulta LLAD

| error

;

bloque_parametro_comillas_credenciales_usuario ::= COMILLAS CREDENCIALES_USUARIO COMILLAS

| error

;

bloque_parametro_comillas_parametros_formulario ::= COMILLAS PARAMETROS_FORMULARIO COMILLAS

| error


```

;

bloque_parametro_comillas_parametros_componente ::=  COMILLAS PARAMETROS_COMPONENTE COMILLAS
|error
;

paquetes_usuario ::= CORI bloques_usuario CORD
|error
;

paquetes_formulario ::= CORI bloques_formulario CORD
|error
;

paquetes_componente ::= CORI bloques_componente CORD
|error
;

paquetes_consulta ::= CORI bloques_consulta CORD
|error
;

bloques_usuario ::= bloques_usuario COMA bloque_usuario
|bloque_usuario
|error
;

bloques_formulario ::= bloques_formulario COMA bloque_formulario
|bloque_formulario
|error

```

;

bloques_componente ::= bloques_componente COMA bloque_componente
 |bloque_componente
 |error
 ;

bloques_consulta ::= bloques_consulta COMA bloque_consulta
 |bloque_consulta
 |error
 ;

bloque_usuario ::= LLAI parametros_usuario LLAD
 ;

bloque_formulario ::= LLAI parametros_formulario LLAD
 ;

bloque_componente ::= LLAI parametros_componente LLAD
 ;

bloque_consulta ::= LLAI parametros_consulta LLAD
 ;

parametros_usuario ::= parametros_usuario COMA parametro_usuario
 |parametro_usuario
 |error

```

;

parametros_formulario ::= parametros_formulario COMA parametro_formulario
                        | parametro_formulario
                        | error
;

parametros_componente ::= parametros_componente COMA parametro_componente
                        | parametro_componente
                        | error
;

parametros_consulta ::= parametros_consulta COMA parametro_consulta
                       | parametro_consulta
                       | error
;

parametro_usuario ::= datos DOS_PUNTOS datos
;

parametro_formulario ::= datos DOS_PUNTOS datos
                       | ids DOS_PUNTOS cont_ids
                       | titulosDOS_PUNTOS cont_string_spaces
                       ;

parametro_componente ::= datos DOS_PUNTOS datos
                       | ids DOS_PUNTOS cont_ids

```

```

        |formulario DOS_PUNTOS cont_ids
        |texto_visible DOS_PUNTOS cont_string_spaces
        |opciones DOS_PUNTOS cont_opciones
        |url DOS_PUNTOS cont_url
    ;
datos ::=      COMILLAS STRING COMILLAS
            |cont_ids
            |error
            ;

ids ::= COMILLAS ID COMILLAS
    ;

cont_ids ::= COMILLAS CONT_ID COMILLAS
    ;

titulos ::= COMILLAS TITULO COMILLAS
    ;

cont_string_spaces ::= COMILLAS STRING_SPACE COMILLAS
                    |datos
                    ;
formulario ::=      COMILLAS FORMULARIO COMILLAS
                    ;
texto_visible ::=   COMILLAS TEXTO_VISIBLE COMILLAS

```

```

;

opciones ::=    COMILLAS OPCIONES COMILLAS

;
cont_opciones ::= COMILLAS CONT_OPCIONES COMILLAS
                  |cont_string_spaces
;

url ::=          COMILLAS URL COMILLAS
;
cont_url ::=     COMILLAS CONT_URL COMILLAS
                  |cont_string_spaces
;
parametro_consulta ::= consulta_n DOS_PUNTOS con_consulta
;
consulta_n ::=   COMILLAS CONSULTA_N COMILLAS
                  |error
;
con_consulta ::= COMILLAS CONT_CONSULTA COMILLAS
                  |error
;

```

Gramática de las respuestas del servidor al cliente(lenguaje igual al Índigo)

Gramática del analizador léxico:

decimales

decimal=[1-9][0-9]*[.]{entero}|[0][.]{entero}

Número enteros

entero=[0-9]+

number = ({entero}|{decimal})

letra = [a-zA-Z]

Palabra sin espacios

string = ({letra}|{number}|[_]|[-]|[\$]|[\.])+

Palabra con espacios

stringSpace = ({letra}|{number})+({string}|{whiteSpace})*

Los ID de componentes y formularios

CONT_ID = ([\$_]|[_]|[-])({letra}|{number}|[\$]|[-]|[_])+

El contenido del parámetro opciones de los componentes

cont_opciones = (({stringSpace}|{string})("")({whiteSpace}))+({stringSpace}|{string})

Para la URL

cont_url = ({path_relativo}|{http}|{https})?([\W])?(({stringSpace})[\W])+({stringSpace})?

http = ("http:/")

https = ("https:/")

path_relativo = (({letra})(":"))

Lectura de todos los tipos de cadenas

allStrings = ({string}|{stringSpace}|{CONT_ID}|{cont_opciones}|{cont_url})

Palabras reservadas sin importar las mayúsculas o minúsculas

ini_respuestas = {ini_respuesta}[sS]

ini_respuesta = [iI][nN][iI][_][rR][eE][sS][pP][uU][eE][sS][tT][aA]

fin_respuestas = {fin_respuesta}[sS]

fin_respuesta = [fF][iI][nN][_][rR][eE][sS][pP][uU][eE][sS][tT][aA]

Todo tipo de espacios en blanco

lineTerminator = \r|\n|\r\n

whiteSpace = {lineTerminator} | [\t\f]

Palabras reservadas:

USUARIO

RESPUESTA_SERVIDOR

RESPUESTA

LOGIN_USUARIO

BLOQUE

Símbolos:

"

{

}

:

[

]

,

<

>
!

Gramática para el analizador sintáctico:

Símbolos terminales:

MENQ,ADMIRACION,INI_RESPUESTAS,INI_RESPUESTA,MAQ, FIN_RESPUESTAS,
DOS_PUNTOS,FIN_RESPUESTA, LOGIN_USUARIO,RESPUESTA_SERVIDOR,BLOQUE,
COMILLAS,USUARIO,RESPUESTA, STRING, LLAI, LLAD, CORI, CORD, COMA

Símbolos no terminales:

inicio,sig_inicio,solicitudes, fin_respuestas,solicitud,sig_solicitud,fin_solicitud, contenido_usuario,contenido_respuesta,
bloque,paquetes_usuario, paquetes_respuesta,bloques_usuario,bloques_respuesta,bloque_usuario, bloque_respuesta,
parametros_usuario,parametros_respuesta,parametro_usuario, parametro_respuesta,usuario, respuesta,datos

Gramática:

inicio ::= MENQ ADMIRACION sig_inicio
 |error sig_inicio
 ;

sig_inicio ::= INI_RESPUESTAS MAQ solicitudes fin_respuestas
 |INI_RESPUESTA DOS_PUNTOS COMILLAS sig_solicitud


```

        |error
        ;
solicitudes ::= solicitudes solicitud
        |solicitud
        |error sig_solicitud
        ;

fin_respuestas ::= MENQ ADMIRACION FIN_RESPUESTAS MAQ
        |error
        ;
solicitud ::=  MENQ ADMIRACION INI_RESPUESTA DOS_PUNTOS COMILLAS sig_solicitud
        ;
sig_solicitud ::=  LOGIN_USUARIO COMILLAS MAQ contenido_usuario fin_solicitud
        |RESPUESTA_SERVIDOR COMILLAS MAQ contenido_respuesta fin_solicitud
        |error fin_solicitud
        ;
fin_solicitud ::=  MENQ ADMIRACION FIN_RESPUESTA MAQ
        |error
        ;
contenido_usuario ::= LLAI bloque DOS_PUNTOS paquetes_usuario LLAD
        |error paquetes_usuario LLAD
        ;
contenido_respuesta ::=LLAI bloque DOS_PUNTOS paquetes_respuesta LLAD
        |error paquetes_respuesta LLAD
        ;

```

```

bloque ::= COMILLAS BLOQUE COMILLAS
        |error
        ;
paquetes_usuario ::= CORI bloques_usuario CORD
        |error
        ;
paquetes_respuesta ::= CORI bloques_respuesta CORD
        |error
        ;
bloques_usuario ::= bloques_usuario COMA bloque_usuario
        |bloque_usuario
        |error
        ;
bloques_respuesta ::= bloques_respuesta COMA bloque_respuesta
        |bloque_respuesta
        |error
        ;
bloque_usuario ::= LLAI parametros_usuario LLAD
        ;
bloque_respuesta ::= LLAI parametros_respuesta LLAD
        ;

parametros_usuario ::= parametros_usuario COMA parametro_usuario
        |parametro_usuario
        |error

```

```

;
parametros_respuesta ::= parametros_respuesta COMA parametro_respuesta
                        | parametro_respuesta
                        | error ;
parametro_usuario ::= usuario DOS_PUNTOS datos
;
parametro_respuesta ::= respuesta DOS_PUNTOS datos
;
usuario ::= COMILLAS USUARIO COMILLAS
;
respuesta ::= COMILLAS RESPUESTA COMILLAS
;
datos ::= COMILLAS STRING COMILLAS
        | error
;

```

Gramática de los reportes(lenguaje SQForm)

Gramática del analizador léxico:

Números decimales:

decimal=[1-9][0-9]*[.]{number}| [0][.]{number}

Números enteros:

number=[0-9]+

Los id de los formularios:

id_form = ([\$_][_][-])(\{letra\}|\{number\}|[_][-])*

Cadena de caracteres alfanuméricos:

nombre = (\{letra\}|\{number\}|[_][-])+

Cadena con espacios

string = (\{letra\}|\{number\})+(\{letra\}|\{number\}|\{whiteSpace\})*

Espacios en blanco

lineTerminator = \r|\n|\r\n

whiteSpace = {\lineTerminator} | [\t\f]

Palabras reservadas:

SELECT

TO

FORM

WHERE

NOT

AND

OR

Símbolos:

-

>

<

=

[
]
,
"

Gramática para el analizador sintáctico:

Símbolos terminales:

COMILLA,SELECT,TO,FORM,GUION,MAQ,ID_FORM,NOMBRE,CORI,CORD,COMA,WHERE,MENQ,IGUAL,
AND,OR,NOT,NUM,STRING

Símbolos no terminales:

inicio,flecha,nombre_form,nombre_forms,nombre,campos,camposs,nombres_campos,clausulas,restrigciones,restrigcion,
operador_relacional,operador_logico,comparacion,string

Gramática:

inicio ::= COMILLA SELECT TO FORM flecha nombre_form
 |error
 ;
flecha ::= GUION MAQ
 |error
 ;

```

nombre_form ::= nombre nombre_forms
              |error
              ;
nombre_forms ::= campos clausulas COMILLA
              |campos COMILLA ;

nombre ::= ID_FORM
         |NOMBRE
         ;
campos ::= CORI camposs
        |error
        ;
camposs ::= nombres_campos CORD
          |CORD
          ;
nombres_campos ::= nombres_campos COMA nombre_Campos
                 |NOMBRE
                 |error
                 ;
clausulas ::= WHERE CORI restrigciones CORD
            |error
            ;
restrigciones ::= restrigciones operador_logico restrigciones
                |restrigcion
                |error

```

```

;
restrigcion ::= nombre operador_relacional comparacion
|error
;

```

```

operador_relacional ::=  MENQ
                        |MAQ
                        |MENQ MAQ
                        |IGUAL
                        |MAQ IGUAL
                        |MENQ IGUAL
;

```

```

operador_logico ::= AND
                |OR
                |NOT
;

```

```

comparacion ::= NUM
            |string
            |error
;

```

```

string ::= COMILLA STRING COMILLA
        |error
;

```

Gramática de almacenamiento de los usuarios(lenguaje Usuario)

Gramática del analizador léxico:

Número decimal:

decimal=[1-9][0-9]*[.]{entero}|[0][.]{entero}

Número entero:

entero=[0-9]+

La unión de los números enteros y decimales

number = ({entero}|{decimal})

Letras

letra = [a-zA-Z]

Cadena de letras

string = ({letra}|{number}|[_]|[-]|[\$]|[\.])+

Espacios en blanco

lineTerminator = \r|\n|\r\n

whiteSpace = {lineTerminator} | [\t\f]

Palabras reservadas:

db.users

USUARIO

USER

PASS

FECHA_CREACION

FECHA_MODIFICACION

Símbolos:

"

{

}

:

[

]

,

Gramática del analizar sintáctico:

Símbolos terminales:

DB_USERS, LLAI, LLAD, DOS_PUNTOS, COMILLA, USUARIO, CORI, CORD, COMA, USER, PASS,
FECHA_CREACION, FECHA_MODIFICACION, STRING, ERROR

Símbolos no terminales:

inicio, usuarios, usuario, usuario_comillas, contenido, user, user_comillas, pass, pass_comillas, fecha_creacion,
fecha_creacion_comillas, fecha_modificacion, fecha_modificacion_comillas, dato

Gramática:

```
inicio ::= DB_USERS LLAI usuarios LLAD  
        |error  
        ;
```

```
usuarios ::= usuarios usuario  
          |usuario  
          ;
```

```
usuario ::= usuario_comillas DOS_PUNTOS CORI contenido CORD  
         |error contenido ;
```

```
usuario_comillas ::= COMILLA USUARIO COMILLA  
                  |error ;
```

```
contenido ::= user COMA pass COMA fecha_creacion COMA fecha_modificacion  
           ;
```

```
user ::= LLAI user_comillas DOS_PUNTOS dato LLAD  
       |error ;
```

```
user_comillas ::= COMILLA USER COMILLA  
               |error ;
```

```
pass ::= LLAI pass_comillas DOS_PUNTOS dato LLAD  
        |error ;
```

pass_comillas ::= COMILLA PASS COMILLA
|error ;

fecha_creacion ::= LLAI fecha_creacion_comillas DOS_PUNTOS dato LLAD
|error ;

fecha_creacion_comillas ::= COMILLA FECHA_CREACION COMILLA
|error ;

fecha_modificacion ::= LLAI fecha_modificacion_comillas DOS_PUNTOS dato LLAD
|error ;

fecha_modificacion_comillas ::= COMILLA FECHA_MODIFICACION COMILLA
|error ;

dato ::= COMILLA STRING COMILLA
|error ;

Gramática de almacenamiento de los formularios(lenguaje Formulario)

Gramática del analizador léxico:

Número decimal:

decimal=[1-9][0-9]*[.]{entero}| [0][.]{entero}

Número entero:

entero=[0-9]+

La unión de los números enteros y los decimales

number = ({entero}|{decimal})

Letra alfanumérico:

letra = [a-zA-Z]

Cadena de letras:

string = ({letra}|{number}|[_]|[-]|[\$]|[\.])+

Cadena de letras con espacios:

stringSpace = ({letra}|{number})+ ({string}|{whiteSpace})*

Id:

CONT_ID = ([\$_]|[_]|[-])({letra}|{number}|[\$]|[-]|[_])+

Opciones de los componentes:

cont_opciones = (({stringSpace}|{string})("")({whiteSpace})?)+ ({stringSpace}|{string})

URL:

cont_url = ({path_relativo}|{http}|{https})?([\\V])?(({stringSpace})[\\V])+ ({stringSpace})?

http = ("http:/")

https = ("https:/")

path_relativo = (({letra})(":"))

La unión de todas las cadenas de letras y números

allStrings = ({string}|{stringSpace}|{CONT_ID}|{cont_opciones}|{cont_url})

Espacios en blanco:

lineTerminator = \r|\n|\r\n

whiteSpace = {lineTerminator} | [\t\f]

Palabras reservadas:

db.formularios
FORMULARIO
ID_FORMULARIO
NOMBRE
TITULO
TEMA
USUARIO_CREACION
FECHA_CREACION
ESTRUCTURA
DATOS_RECOPILADOS
ID
NOMBRE_CAMPO
CLASE
INDICE
TEXTO_VISIBLE
ALINEACION
OPCIONES
REQUERIDO
FILAS
COLUMNAS
URL
REGISTRO

Símbolos:

"
{
}
:
[
]
,
(
)

Gramática del analizar sintáctico:

Símbolos terminales:

DB_FORMULARIOS, PAI, PAD, FORMULARIO, DOS_PUNTOS, CORI, CORD, COMA, COMILLA, ID_FORM, TITULO, NOMBRE, TEMA, USER_CREACION, FECHA_CREACION, ESTRUCTURA, LLAI, LLAD, ID_COMP, NOMBRE_CAMPO, CLASE, INDICE, TEXTO_VISIBLE, ALINEACION, REQUERIDO, OPCIONES, FILAS, COLUMNAS, URL, DATOS_RECOPIADOS, REGISTRO, STRING, ERROR

Símbolos no terminales:

inicio, formularios, formulario, contenido_formulario, id_form, titulo, nombre, tema, user_creacion, fecha_creacion, estructura, sig_estructura, componentes, componente, valores_componente, id_comp, nombre_campo, lase, indice, texto_visible, alienacion,

requerido,opciones,filas,columnas,url,datos_recopilados,sig_datos_recopilados,registros,registro,bloque_registro,
nombre_campo_registro,cont_registros,cont_registro,dato

Gramática:

inicio ::= DB_FORMULARIOS PAI formularios PAD
 |error
 ;

formularios ::= formularios formulario
 |formulario
 |error
 ;

formulario ::= COMILLA FORMULARIO COMILLA DOS_PUNTOS CORI contenido_formulario CORD
 ;

contenido_formulario ::= id_form COMA titulo COMA nombre COMA tema COMA user_creacion COMA fecha_creacion
 COMA estructura datos_recopilados
 ;

id_form ::= COMILLA ID_FORM COMILLA DOS_PUNTOS dato
 |error ;

titulo ::= COMILLA TITULO COMILLA DOS_PUNTOS dato
 |error ;

nombre ::= COMILLA NOMBRE COMILLA DOS_PUNTOS dato
 |error ;

```

tema ::= COMILLA TEMA COMILLA DOS_PUNTOS dato
      |error ;
user_creacion ::= COMILLA USER_CREACION COMILLA DOS_PUNTOS dato
      |error ;
fecha_creacion ::= COMILLA FECHA_CREACION COMILLA DOS_PUNTOS dato
      |error ;

estructura ::= COMILLA ESTRUCTURA COMILLA DOS_PUNTOS PAI sig_estructura
      |error ;

sig_estructura ::= componentes PAD
      |PAD
      ;
componentes ::= componentes componente
      |componente
      ;
componente ::= LLAI valores_componente LLAD
      |error ;

valores_componente ::= id_comp COMA nombre_campo COMA clase COMA indice COMA texto_visible COMA
alienacion COMA requerido COMA opciones COMA filas COMA columnas COMA url
      ;
id_comp ::= COMILLA ID_COMP COMILLA DOS_PUNTOS dato
      |error ;
nombre_campo ::= COMILLA NOMBRE_CAMPO COMILLA DOS_PUNTOS dato

```



```

|error ;
clase ::= COMILLA CLASE COMILLA DOS_PUNTOS dato
|error ;
indice ::= COMILLA INDICE COMILLA DOS_PUNTOS dato
|error ;
texto_visible ::= COMILLA TEXTO_VISIBLE COMILLA DOS_PUNTOS dato
|error ;
alienacion ::= COMILLA ALINEACION COMILLA DOS_PUNTOS dato
|error ;
requerido ::= COMILLA REQUERIDO COMILLA DOS_PUNTOS dato
|error ;
opciones ::= COMILLA OPCIONES COMILLA DOS_PUNTOS dato
|error ;
filas ::= COMILLA FILAS COMILLA DOS_PUNTOS dato
|error ;
columnas ::= COMILLA COLUMNAS COMILLA DOS_PUNTOS dato
|error ;
url ::= COMILLA URL COMILLA DOS_PUNTOS dato
|error ;

datos_recopilados ::= COMILLA DATOS_RECOPILADOS COMILLA DOS_PUNTOS PAI sig_datos_recopilados
|error ;

sig_datos_recopilados ::= registros PAD
|PAD

```

|error
;

registros ::= registros registro
|registro
;

registro ::= LLAI bloque_registro LLAD
|error ;

bloque_registro ::= nombre_campo_registro COMA cont_registros
|error ;

nombre_campo_registro ::= COMILLA NOMBRE_CAMPO COMILLA DOS_PUNTOS dato
|error ;

cont_registros ::= cont_registros cont_registro
|cont_registro
;

cont_registro ::= COMILLA REGISTRO COMILLA DOS_PUNTOS dato
|error ;

dato ::= COMILLA STRING COMILLA
|error ;

Diagrama de Clases de la aplicación cliente:

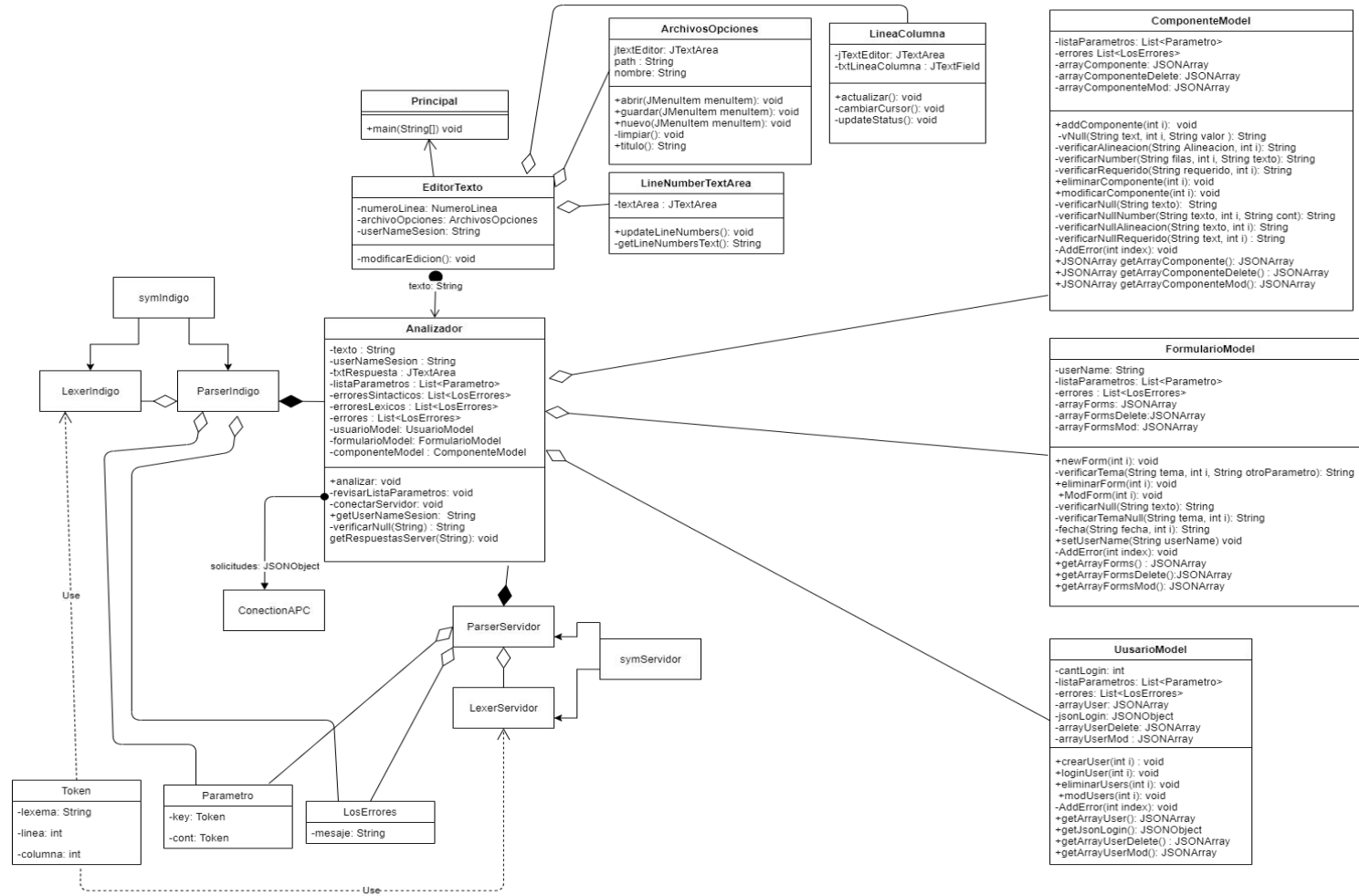


Diagrama de Clases de la aplicación Web:

