

# Manual técnico

## Herramientas de desarrollo

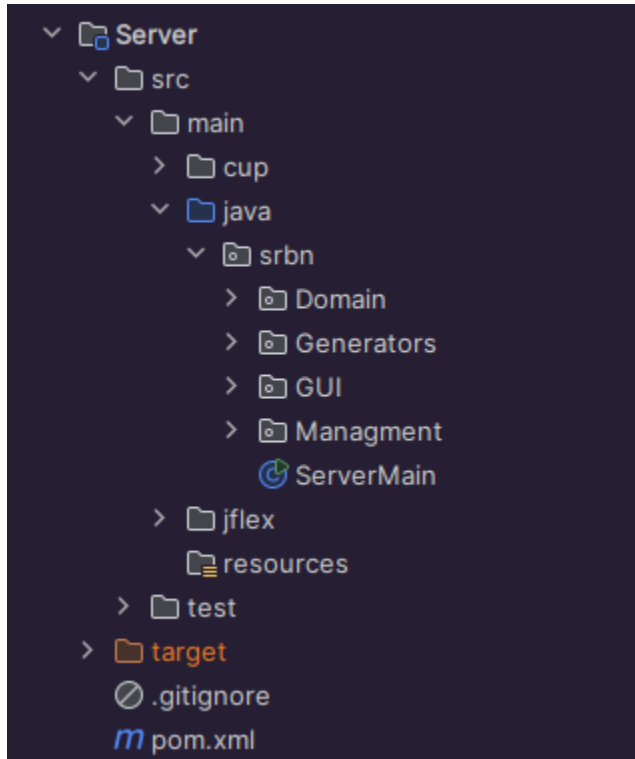
Las herramientas (IDEs, lenguaje de programación, librerías, servidor, etc) usadas para el desarrollo de la aplicación (servidor y cliente) se listan a continuación:

- Lenguaje:
  - Java:
    - versión 17.0.8
- Analizador léxico:
  - JFlex:
    - versión 1.9.1
- Analizador sintáctico:
  - JCup Runtime:
    - versión 11b
- Implementación del servidor
  - Apache Server:
    - versión 2.4
    - desplegado localmente en el puerto 80
- Librerías utilizadas:
  - Jackson – core:
    - versión 2.17.0
  - Jackson – datatype:
    - versión 2.17.0
- Entorno de desarrollo (IDE);
  - IntelliJ IDEA:
    - versión 2024.1 (Ultimate edition)
- Sistema operativo de desarrollo
  - Windows 11 – home
    - versión 10.0.22621

## Organización del código fuente

### Aplicación servidor

La aplicación del servidor, presenta la siguiente estructura de paquetes y directorios:



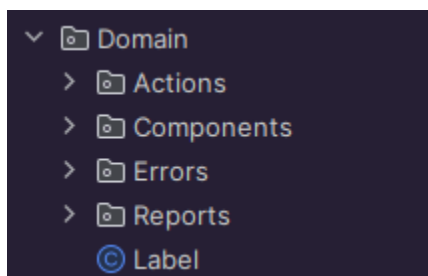
#### Directorio: **Java**

A demás de contar con los paquetes:

- Domain
- Generators
- GUI
- Management\*

Tenemos la clase **ServerMain** que seria la clase principal con la cual ejecutamos todo el código

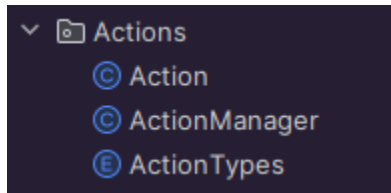
#### Paquete: **Domain**



En este paquete, básicamente se manejan todos los objetos de la aplicación, conformado por los paquetes;

- Actions
- Components
- Errors
- Reports

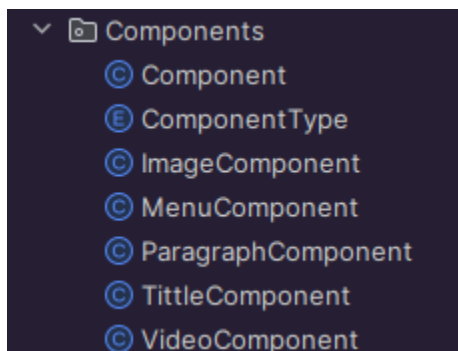
#### Paquete: Domain – Actions



Formado por tres clases

- Action: es el objeto acción, este es utilizado como la base de la aplicación, son objetos contruidos de forma ascendente por medio de la **sintaxis el proyecto**.
- ActionManager: es una clase con la cual manejamos los diferentes tipos de acciones (objetos Action) que podemos tener por medio de la gramática, esos tienen funciones diferentes en el programa y en nuestro equipo, lo cual es controlado por mencionada clase.
- ActionTypes: clase tipo **Enum** utilizado para manejo y control de los diferentes tipos de acciones que podemos tener en la aplicación.

#### Paquete: Domain – Components



Formado por siete clases, las cuales manejan una herencia, siendo el objeto padre **Component**, la cual forma parte de nuestro objeto **Action** y este hereda sus atributos principales a la siguiente lista:

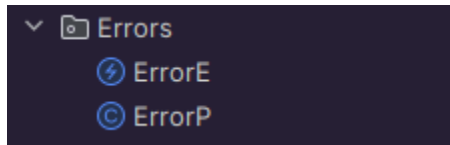
- ImageComponent
- MenuComponent
- ParagraphComponent
- TittleComponent

- VideoComponent

Que son diferentes tipos de componentes, los cuales tienen atributos propios y distintos cada uno

- ComponentType: Clase tipo **Enum** la cual nos ayuda para llevar un mejor control de los diferentes tipos de componentes (**Component**) y tener un mejor manejo de los mismos.

Paquete: Domain – Errors

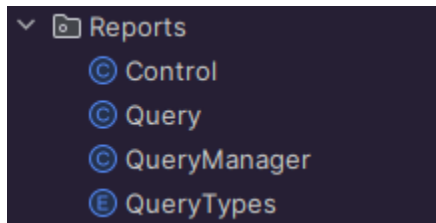


Formado por dos clases

- ErrorE: Es una clase, abreviando su nombre **ErrorException**, que extiende de la clase perteneciente a java **Exception** y esta hecha para llevar un mejor control de los errores y excepciones surgidas en la aplicación.
- ErrorP: Es una clase, abreviando su nombre **ErrorProduction**, que nos ayuda a tener un mejor manejo de los errores obtenidos en nuestra producción gramatical.

Es un paquete dedicado al manejo de errores, para que la aplicación no detenga su ejecución al encontrar este tipo de errores.

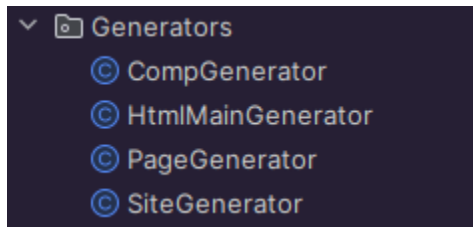
Paquete: Domain - Reports



Formado por 4 clases

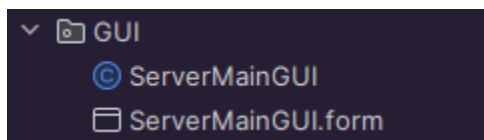
- Control: Un objeto dedicado para el control de visitas en la página, este guarda los identificadores de la página y lleva un conteo que incrementa en medida que las páginas o sitios son visitados.
- Query: Un objeto dedicado para almacenar las consultas capturadas en la gramática y tener un mejor manejo de ellas en nuestra aplicación.
- QueryManager: Es una clase con la cual manejamos los diferentes tipos de consultas (**Query**) que podemos obtener en nuestra gramática, ya que cada una funciona y solicita diferentes datos que han ido grabándose durante la ejecución de nuestra aplicación.
- QueryTypes: Clase tipo **Enum** que nos ayuda para llevar un mejor manejo de los diferentes tipos de consultas que se pueden realizar.

## Paquete: **Generators**



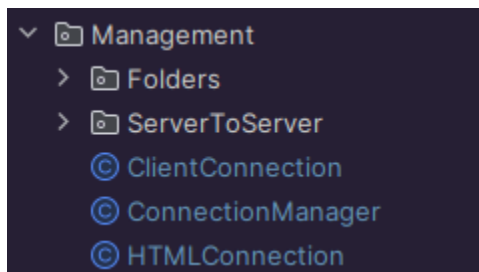
Formado por cuatro clases, las cuales sirven para generar código html según se solicite en el xml de entrada

## Paquete: **GUI**



Formado por dos clases, un formulario de java swing y el controlador java del formulario, son clases que pertenecen a la interfaz grafica del servidor

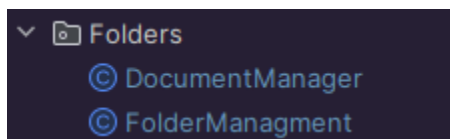
## Paquete: **Management**



Formado por dos paquetes y tres clases

- ClientConnection: Clase en donde se maneja la conexión con la aplicación del cliente por el puerto 81
- HTMLConnection: Clase en donde se maneja la escucha a las peticiones hechas por el html para el manejo de reportes, por el puerto 82
- ConnectionManager: Clase dedicada al manejo de las peticiones de ambas conexiones, peticiones xml y html

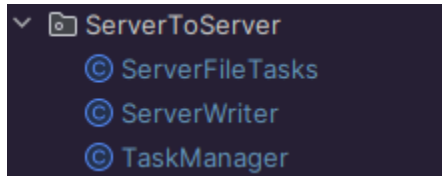
## Paquete: Management – Folders



Formado por dos clases, paquete dedicado para la escritura y lectura, de archivos y de directorios

- DocumenetManger: Clase dedicada a la creación de archivos que utilizara nuestra aplicación, tanto los html generados como archivos utilizados por la aplicación para el manejo de peticiones
- FolderManagement\*: Clase dedicada al manejo y control de directorios utilizados por la aplicación.

#### Paquete: ServerToServer

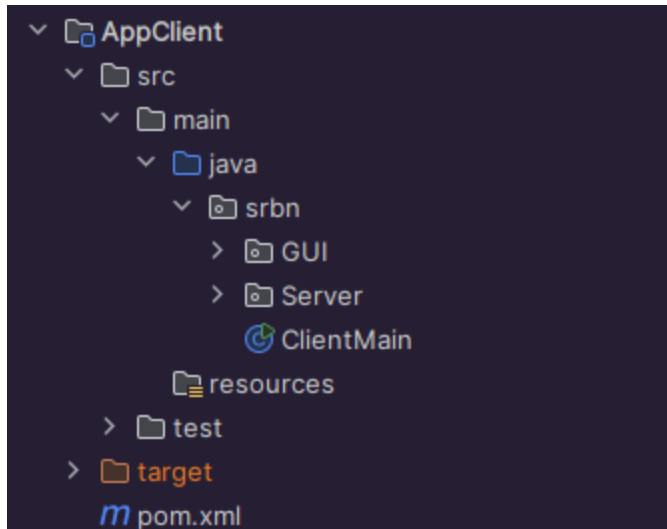


Formado por tres clases, las cuales son utilizadas para configuración del servidor (apache) y manejo de este

- ServerFileTasks (deprecated): clase creada para edicion y escritura de archivos de configuracion del servidor.
- ServerWriter (deprecated): clase creada para la escritura de archivos de configuración del servidor.
- TaskManager: Clase encargada del manejo de las tareas del servidor, guardar dominios o escribir reportes.

## Aplicación Cliente

La aplicación del servidor, presenta la siguiente estructura de paquetes y directorios:



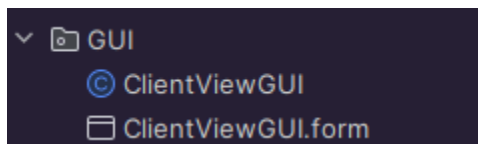
### Directorio: Java

A demás de contar con los paquetes

- GUI
- Server

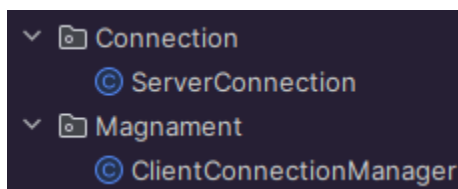
Tenemos la clase ***ClientMain***, que sería la clase principal con la cual ejecutamos todo el código

### Paquete: GUI



Formado por dos clases, un formulario de java swing y el controlador java del formulario, son clases que pertenecen a la interfaz gráfica de la aplicación del cliente

### Paquete: Server



Formado por dos paquetes sencillos

### Paquete: Server – Connection

- ServerConnection: Clase creada para el manejo de las solicitudes enviadas al socket que se conecta al servidor

Paquete: Server – Management\*

- ClientConnectionManager: Clase dedicada a la conexión al puerto 81, puerto con el cual se comunica con el servidor de la aplicación.



## Análisis léxico

La parte del análisis léxico de las peticiones que la aplicación cliente puede realizar están a cargo de la clase ***Lexer*** que como ya se describió, fue generada por la herramienta ***JFlex***, utilizando las siguientes expresiones regulares

Saltos de línea y espacios en blanco, fue necesario declarar dichas expresiones para poder incluirlas u obviarlas en algunas expresiones que lo requieran

```
WHITESPCS = ([\s\t\r\n]+)
```

Secuencia de dígitos, para utilizarlo en números enteros

```
DIGIT = [0-9]
```

Apertura de etiquetas, para usos prácticos y mejor manejo en la gramática, las etiquetas de apertura de un bloque de etiquetas, atributos y parámetros, se manejaron como un token único que cumplieran con la estructura representada en la siguiente imagen.

```
//openTags  
OPENLABELSTAG = "<etiquetas>"  
OPENATTRIBUTE = "<atributos>"  
OPENPARAMSTAG = "<parametros>"
```

Cierre de etiqueta, para usos prácticos y mejor manejo en la gramática, las etiquetas de cierre de bloques de parámetros, etiquetas, atributos y acciones, se manejaron como un token único que cumplen con las estructuras representadas en las siguientes imágenes, a esto también se agrega el cierre de acción y el cierre de un tag de línea (</>)

```
//closeTags  
CLOSEACTIONTAG = "</accion>"  
CLOSEPARAMTAG = "</parametro>"  
CLOSELABELTAG = "</>"  
CLOSELABELSTAG = "</etiquetas>"  
CLOSEATTRIBUTE = "</atributo>"  
CLOSEATTRIBUTE = "</atributos>"  
CLOSEPARAMSTAG = "</parametros>"  
CLOSEACTIONSTAG = "</acciones>"
```

Tenemos expresiones donde detallamos la secuencia esperada de los tipos de justificación y, en este caso, **TITULOID** puede ser la declaración de un componente tipo título o de un atributo tipo título, por ende, son dos valores distintos diferenciados por el uso de los corchetes, de igual forma, internamente se ignoran mencionados corchetes

```
TITULOID = "[TITULO]" | "TITULO"  
CENTER_LOC = "[CENTRAR]"  
LEFT_LOC = "[IZQUIERDA]"  
RIGHT_LOC = "[DERECHA]"  
JUSTIFY_LOC = "[JUSTIFICAR]"
```

Declaramos la expresión regular de IDs, en un caso están encerrados por corchetes, los cuales sirven para identificar los identificadores o nombres de los elementos que se crean en el programa, a diferencia de **TAGID**, que están encerrados en comillas dobles (") y estos son utilizados para definir valores tipo texto o similar, en ambos casos se ignoran los corchetes y comillas para el uso del texto que almacenan.

```
ID = ("["[^\n\r]*"]")  
TAGID = ("\["[^\n\r]*\"")
```

Se agregaron también símbolos específicos que forman parte del alfabeto de la gramática

```
DBLEQUOTES = "\"\""  
EQUAL = "="  
MAYORQ = ">"  
COMMA = ","  
SEMICOLON = ";"
```

Expresiones regulares utilizadas para fechas y también para códigos hexadecimales para el manejo de colores.

```
DATE      = ("["[0-9]{4}-[0-9]{2}-[0-9]{2}"]")
HEXCODE    = ("#[0-9a-fA-F]{6}"]")
```

El uso de las palabras reservadas fue definido en la declaración de los símbolos que forman parte del alfabeto, formado por las palabras reservadas con las que se abre la etiqueta de acciones, etiqueta de parámetro, nombre, etiqueta, atributo y la apertura de acción, esto debido a su secuencia de caracteres queda abierto (sin el cierre de >)

Las palabras reservadas para el uso de consultas, están definidas en este bloque, también las del tipo de acción que se busca ejecutar.

Así mismo, se encuentran las palabras reservadas del tipo de componente que se busca colocar y las que forman parte del tipo de atributo que se desea colocar

Los tipos de alineación también forman parte de las palabras reservadas

```
//reservadas
(<acciones>)           {return symbol(ParserSym.ACTIONS, yytext());}
(<parametro>)          {return symbol(ParserSym.PARAM, yytext());}
(nombre="\")           {return symbol(ParserSym.NAME, yytext());}
(<etiqueta>)           {return symbol(ParserSym.OPENLABELTAG, yytext());}
(<atributo>)            {return symbol(ParserSym.ATTRIBUTE, yytext());}
(<accion>)              {return symbol(ParserSym.ACTION, yytext());}
//consultas
(CONSULTAR)             {return symbol(ParserSym.CONSULT, yytext());}
(VISITAS_PAGINA)        {return symbol(ParserSym.VISITS_PAGE, yytext());}
(VISITAS_SITIO)         {return symbol(ParserSym.VISITS_SITE, yytext());}
(PAGINAS_POPULARES)     {return symbol(ParserSym.POPULAR_PAGES, yytext());}
(COMPONENTE)            {return symbol(ParserSym.COMPONENT, yytext());}
(TODO)                  {return symbol(ParserSym.ALL, yytext());}
//tipos de acciones
(NUEVO_SITIO_WEB)       {return symbol(ParserSym.NEW_SITE, yytext());}
(BORRAR_SITIO_WEB)      {return symbol(ParserSym.DELETE_SITE, yytext());}
(NUEVA_PAGINA)          {return symbol(ParserSym.NEW_PAGE, yytext());}
(BORRAR_PAGINA)         {return symbol(ParserSym.DELETE_PAGE, yytext());}
(MODIFICAR_PAGINA)      {return symbol(ParserSym.MODIFY_PAGE, yytext());}
(AGREGAR_COMPONENTE)    {return symbol(ParserSym.ADD_COMPONENT, yytext());}
(MODIFICAR_COMPONENTE)  {return symbol(ParserSym.MODIFY_COMPONENT, yytext());}
(BORRAR_COMPONENTE)     {return symbol(ParserSym.DELETE_COMPONENT, yytext());}
```

```
//clases de componentes
(PARRAFO)                {return symbol(ParserSym.PARAGRAPH, yytext());}
(IMAGEN)                  {return symbol(ParserSym.IMAGE, yytext());}
(VIDEO)                   {return symbol(ParserSym.VIDEO, yytext());}
(MENU)                    {return symbol(ParserSym.MENU, yytext());}
//ATRIBUTOS
(TEXTO)                   {return symbol(ParserSym.TEXT, yytext());}
(ALINEACION)              {return symbol(ParserSym.ALIGNMENT, yytext());}
(COLOR)                   {return symbol(ParserSym.COLOR, yytext());}
(ORIGEN)                  {return symbol(ParserSym.SOURCE, yytext());}
(ALTURA)                 {return symbol(ParserSym.HEIGHT, yytext());}
(ANCHO)                   {return symbol(ParserSym.WIDTH, yytext());}
//ALINEACIONES
{CENTER_LOC}              {return symbol(ParserSym.CENTER_LOC, yytext());}
{LEFT_LOC}                {return symbol(ParserSym.LEFT_LOC, yytext());}
{RIGHT_LOC}               {return symbol(ParserSym.RIGHT_LOC, yytext());}
{JUSTIFY_LOC}             {return symbol(ParserSym.JUSTIFY_LOC, yytext());}
```

## Análisis sintáctico

La parte del análisis sintáctico, como ya se mencionó anteriormente, está a cargo de la clase ***Parser***, clase que ha sido generada con la herramienta JCup, a continuación se describe la gramática, que hace posible ejecutar las peticiones de la aplicación cliente.

### Descripción de la gramática

Se tienen declarados los siguientes no terminales

```
non terminal String    parentAttribute, sourceAttribute, colorAttribute, textAttribute, pageParam, modDateParam,
                        userModParam, createDateParam, userCrtParam, parentParam, siteParam, tittleParam, idParam,
                        alignmentLoc, widthAttribute, heightAttribute, alignmentAttribute, compTypeDecl;

non terminal Component  tittleCompDef, paragCompDef, imageCompDef, videoCompDef, menuCompDef, classParamDef;

non terminal Action     newSiteActDef, deleteSiteActDef, newPageActDef, deletePageActDef, modifyPageActDef, addCompActDef,
                        modifyCompActDef, deleteCompActDef, defActionsDecl, opnActionDecl, actionDecl;

non terminal Label      labelDef;

non terminal ArrayList<Label>  labelDecl, addLabels;

non terminal ArrayList<Control> pathDef;

non terminal Query      querysTypeDef, pagesVisitsDef, siteVisitsDef, popularPagesDef, componentDef;

non terminal            xmlWebCreation, clsTags, querysDecl ;
```

Se tienen declarados los siguientes terminales

terminal String	NUMBER_VALUE, CENTER_LOC, LEFT_LOC, RIGHT_LOC, JUSTIFY_LOC, HEIGHT, WIDTH, ID_VALUE, TAGID, COLOR_VALUE, DATE_VALUE;
terminal	DBLEQUOTES, MAYORQ, OPENLABELSTAG, CLOSEACTIONTAG, CLOSEPARAMTAG, CLOSELABELTAG, CLOSELABELSTAG, ACTION, PARAM, NAME, OPENPARAMSTAG, OPENATTRIBUTESTAG, ATTRIBUTE, OPENLABELTAG, SEMICOLON, CLOSEATTRIBUTESTAG, CLOSEATTRIBUTETAG, CLOSEPARAMSTAG, CLOSEACTIONTAG, DELETE_COMPONENT, NEW_SITE, DELETE_SITE, NEW_PAGE, DELETE_PAGE, MODIFY_PAGE, ADD_COMPONENT, MODIFY_COMPONENT, ID_PARAM, TITTLE, SITE, PAGE, CLASSTYPE, USER_CREATION, DATE_CREATION, USER_MODIFICATION, DATE_MODIFICATION, PARENT, PARAGRAPH, IMAGE, VIDEO, MENU, TEXT, ALIGNMENT, COLOR, SOURCE, COMMA, ACTIONS, CONSULT, VISITS_PAGE, VISITS_SITE, POPULAR_PAGES, COMPONENT, ALL;

Iniciando la gramática de esta forma, teniendo como principal tres tipos de declaraciones

***defActionsDecl***: utilizada para definir un bloque de acciones, es decir, muchas acciones seguidas encerradas dentro de las etiquetas <acciones> </acciones>

***actionDecl***: utilizada para declarar una acción en sus distintos tipos

***querysDecl***: utilizada para declarar consultas de reportes

```

start with xmlWebCreation;

xmlWebCreation ::=  actionDecl:newAction {: addAction(newAction);;}
                  |  ACTIONS defActionsDecl:actions CLOSEACTIONSTAG {: System.out.println("actions det");;}
                  |  querysDecl:queries {: System.out.println("queries det");;}
                  ;

defActionsDecl ::=  actionDecl:newAction {: addAction(newAction);;}
                  |  defActionsDecl:_newAction actionDecl:newAction {: addAction(newAction);;}
                  ;

actionDecl ::=     ACTION NAME opnActionDecl:newAction CLOSEACTIONTAG {: RESULT = newAction;;}
                  |  error      {:;}
                  ;

querysDecl ::=     CONSULT querysTypeDef:query SEMICOLON {: addQuery(query);;}
                  |  querysDecl:_type CONSULT querysTypeDef:query SEMICOLON {: addQuery(query);;}
                  ;

```

Las consultas divididas en sus diferentes tipos, tenemos una producción por cada tipo según se requiera

```

querysTypeDef ::=  pagesVisitsDef:pagesVisits      {: RESULT = pagesVisits; ;}
                  |  siteVisitsDef:siteVisits      {: RESULT = siteVisits; ;}
                  |  popularPagesDef:popularPages  {: RESULT = popularPages; ;}
                  |  componentDef:component        {: RESULT = component; ;}
                  ;

siteVisitsDef ::=  VISITS_SITE pathDef:paths      {:
                                                    Query query = new Query("GET_VISITS_SITE", paths);
                                                    RESULT = query;
                                                    :}
                  ;

pagesVisitsDef ::=  VISITS_PAGE pathDef:paths      {:
                                                    Query query = new Query("GET_VISITS_PAGE", paths);
                                                    RESULT = query;
                                                    :}
                  ;

popularPagesDef ::=  POPULAR_PAGES pathDef:paths  {:
                                                    Query query = new Query("GET_POPULAR_PAGES", paths);
                                                    RESULT = query;
                                                    :}
                  ;

componentDef ::=   COMPONENT compTypeDef:compType pathDef:paths
                  {:
                    Query query = new Query("GET_COMPONENTS", getTypeComp(compType), paths);
                    RESULT = query;
                  :}

```

A diferencia de las acciones, que al ser diferentes de forma tan compleja, esta descrita por las siguientes producciones

```

opnActionDecl ::=  newSiteActDef:actionnewSite      {: RESULT = actionnewSite; System.out.println("new Site act");;}
                  |  deleteSiteActDef:actiondeleteSite {: RESULT = actiondeleteSite; System.out.println("delete site act"); ;}
                  |  newPageActDef:actionnewPage      {: RESULT = actionnewPage; System.out.println("new Page act"); ;}
                  |  deletePageActDef:actiondeletePage {: RESULT = actiondeletePage; System.out.println("del Page act"); ;}
                  |  modifyPageActDef:actionmodifyPage {: RESULT = actionmodifyPage; System.out.println("mod Page act"); ;}
                  |  addCompActDef:actionaddComp      {: RESULT = actionaddComp; System.out.println("add comp act");;}
                  |  modifyCompActDef:actionmodifyComp {: RESULT = actionmodifyComp; System.out.println("mod comp act"); ;}
                  |  deleteCompActDef:actiondeleteComp {: RESULT = actiondeleteComp; System.out.println("del comp act"); ;}
                  |  error:err                        {: ;}
                  ;

```

Produccion para creación de nuevo sitio, en esta se pueden obviar un o mas valores de los parámetros de creacion

```
newSiteActDef ::=      NEW_SITE:actionType DBLEQUOTES MAYORQ
                        OPENPARAMSTAG
                        idParam:idP
                        userCrtParam:userCrtP
                        createDateParam:creationDateP
                        modDateParam:modDateP
                        userModParam:userModParam
                        CLOSEPARAMSTAG

                    |      NEW_SITE:actionType DBLEQUOTES MAYORQ
                        OPENPARAMSTAG
                        idParam:idP
                        userCrtParam:userCrtP
                        createDateParam:creationDateP
                        CLOSEPARAMSTAG
```

```
|      NEW_SITE:actionType DBLEQUOTES MAYORQ
      OPENPARAMSTAG
      idParam:idP
      userCrtParam:userCrtP
      CLOSEPARAMSTAG

|      NEW_SITE:actionType DBLEQUOTES MAYORQ
      OPENPARAMSTAG
      idParam:idP
      CLOSEPARAMSTAG
```

Creacion de paginas web, siguiendo la misma secuencia que la creación de sitios, en ella pueden obviarse uno o mas atributos

```
newPageActDef ::= NEW_PAGE:actionType DBLEQUOTES MAYORQ
                  OPENPARAMSTAG
                  idParam:idP
                  tittleParam:tittleP
                  siteParam:siteP
                  parentParam:parentP
                  userCrtParam:userCrtP
                  createDateParam:creationDateP
                  userModParam:userModP
                  modDateParam:modDatP
                  CLOSEPARAMSTAG
                  addLabels:lbls
```

```
| NEW_PAGE:actionType DBLEQUOTES MAYORQ
  OPENPARAMSTAG
  idParam:idP
  tittleParam:tittleP
  siteParam:siteP
  parentParam:parentP
  userCrtParam:userCrtP
  CLOSEPARAMSTAG
```

```
| NEW_PAGE:actionType DBLEQUOTES MAYORQ
  OPENPARAMSTAG
  idParam:idP
  tittleParam:tittleP
  siteParam:siteP
  parentParam:parentP
  CLOSEPARAMSTAG
```



```
| NEW_PAGE:actionType DBLEQUOTES MAYORQ  
OPENPARAMSTAG  
idParam:idP  
tittleParam:tittleP  
siteParam:siteP  
CLOSEPARAMSTAG
```

```
| NEW_PAGE:actionType DBLEQUOTES MAYORQ  
OPENPARAMSTAG  
idParam:idP  
tittleParam:tittleP  
siteParam:siteP  
userCrtParam:userCrtP  
createDateParam:creationDateP  
CLOSEPARAMSTAG
```

```
| NEW_PAGE:actionType DBLEQUOTES MAYORQ  
OPENPARAMSTAG  
idParam:idP  
tittleParam:tittleP  
siteParam:siteP  
userCrtParam:userCrtP  
CLOSEPARAMSTAG
```

Producciones para la eliminación de paginas y sitios creados anteriormente

```
deleteSiteActDef ::= DELETE_SITE DBLEQUOTES MAYORQ  
                  OPENPARAMSTAG  
                  idParam:idP  
                  CLOSEPARAMSTAG
```

```
deletePageActDef ::= DELETE_PAGE:actionType DBLEQUOTES MAYORQ  
                   OPENPARAMSTAG  
                   idParam:id_value  
                   CLOSEPARAMSTAG
```

Produccion para la edición de valores de paginas o sitios

```
modifyPageActDef ::= MODIFY_PAGE:actionType DBLEQUOTES MAYORQ  
                  OPENPARAMSTAG  
                  idParam:id_value  
                  tittleParam:tittle_value  
                  CLOSEPARAMSTAG  
                  addLabels:lbls
```

Producciones para manejo de componentes en una pagina

Agregar un componente a una pagina

```
addCompActDef ::= ADD_COMPONENT DBLEQUOTES MAYORQ  
                 OPENPARAMSTAG  
                 idParam:id_value  
                 pageParam:page_value  
                 PARAM NAME CLASSTYPE DBLEQUOTES MAYORQ  
                 classParamDef:comp
```

Modificar un componente en una pagina

```

modifyCompActDef ::= MODIFY_COMPONENT:actionType DBLEQUOTES MAYORQ
                    OPENPARAMSTAG
                    idParam:id_value
                    pageParam:page_value
                    PARAM NAME CLASSTYPE DBLEQUOTES MAYORQ
                    classParamDef:comp

```

Borrar un componente en una pagina

```

deleteCompActDef ::= DELETE_COMPONENT:actionType DBLEQUOTES MAYORQ
                    OPENPARAMSTAG
                    idParam:id_value
                    pageParam:page_value
                    CLOSEPARAMSTAG

```

Debido al tipo de componentes que podemos manejar, se decidio trabajar en una subproduccion mas pequeña y especifica, la cual da lugar a

```

classParamDef ::=  tittleCompDef:tittleComp
                   |  paragCompDef:paragComp
                   |  imageCompDef:imageComp
                   |  videoCompDef:videoComp
                   |  menuCompDef:menuComp
                   ;

```

Compueta por las producciones

```

tittleCompDef ::=  TITTLE:tittle
                   CLOSEPARAMTAG
                   CLOSEPARAMSTAG
                   OPENATTRIBUTESTAG
                   textAttribute:textAttribute
                   alignmentAttribute:alignmentAttribute
                   colorAttribute:colorAttribute
                   CLOSEATTRIBUTESTAG

```

✓		TITTLE:tittle CLOSEPARAMTAG CLOSEPARAMSTAG OPENATTRIBUTE textAttribute:textAttribute alignmentAttribute:alignmentAttribute CLOSEATTRIBUTESTAG
---	--	---

✓		TITTLE:tittle CLOSEPARAMTAG CLOSEPARAMSTAG OPENATTRIBUTE textAttribute:textAttribute colorAttribute:colorAttribute CLOSEATTRIBUTESTAG
---	--	---

✓		TITTLE:tittle CLOSEPARAMTAG CLOSEPARAMSTAG OPENATTRIBUTE textAttribute:textAttribute CLOSEATTRIBUTESTAG
---	--	--

```
paragCompDef::=
```

```
    PARAGRAPH:tittle  
    CLOSEPARAMTAG  
    CLOSEPARAMSTAG  
    OPENATTRIBUTESTAG  
    textAttribute:textAttribute  
    alignmentAttribute:alignmentAttribute  
    colorAttribute:colorAttribute  
    CLOSEATTRIBUTESTAG
```

```
|  
    PARAGRAPH:tittle  
    CLOSEPARAMTAG  
    CLOSEPARAMSTAG  
    OPENATTRIBUTESTAG  
    textAttribute:textAttribute  
    alignmentAttribute:alignmentAttribute  
    CLOSEATTRIBUTESTAG
```

```
|  
    PARAGRAPH:tittle  
    CLOSEPARAMTAG  
    CLOSEPARAMSTAG  
    OPENATTRIBUTESTAG  
    textAttribute:textAttribute  
    colorAttribute:colorAttribute  
    CLOSEATTRIBUTESTAG
```

```
| PARAGRAPH:tittle
CLOSEPARAMTAG
CLOSEPARAMSTAG
OPENATTRIBUTESTAG
textAttribute:textAttribute
CLOSEATTRIBUTESTAG
```

```
imageCompDef::=
```

```
IMAGE:image
CLOSEPARAMTAG
CLOSEPARAMSTAG
OPENATTRIBUTESTAG
sourceAttribute:sourceAttribute
alignmentAttribute:alignmentAttribute
heightAttribute:heightAttribute
widthAttribute:widthAttribute
CLOSEATTRIBUTESTAG
```

```
| IMAGE:image
CLOSEPARAMTAG
CLOSEPARAMSTAG
OPENATTRIBUTESTAG
sourceAttribute:sourceAttribute
heightAttribute:heightAttribute
widthAttribute:widthAttribute
CLOSEATTRIBUTESTAG
```

```

videoCompDef ::=
    VIDEO:video
    CLOSEPARAMTAG
    CLOSEPARAMSTAG
    OPENATTRIBUTETAG
    sourceAttribute:sourceAttribute
    heightAttribute:heightAttribute
    widthAttribute:widthAttribute
    CLOSEATTRIBUTETAG

```

```

menuCompDef ::=
    MENU:menu
    CLOSEPARAMTAG
    CLOSEPARAMSTAG
    OPENATTRIBUTETAG
    parentAttribute:parentAttribute
    CLOSEATTRIBUTETAG
    clsTags

```

Dichas producciones van según el tipo de atributo o parámetro que necesiten los componentes en su creación, a continuación se describen las estructuras de los componentes utilizados tanto en las producciones de creación y edición de sitios, como en las de manejo de componentes

```

idParam ::=
    PARAM NAME ID_PARAM DBLEQUOTES MAYORQ
    ID_VALUE:id_value
    CLOSEPARAMTAG
    ;

tittleParam ::=
    PARAM NAME TITTLE DBLEQUOTES MAYORQ
    ID_VALUE:id_value
    CLOSEPARAMTAG
    ;

```

```
siteParam::=      PARAM NAME  SITE DBLEQUOTES MAYORQ
                  ID_VALUE:id_value
                  CLOSEPARAMTAG
                  ;

parentParam::=    PARAM NAME PARENT DBLEQUOTES MAYORQ
                  ID_VALUE:id_value
                  CLOSEPARAMTAG
                  ;

userCrtParam::=   PARAM NAME  USER_CREATION DBLEQUOTES MAYORQ
                  ID_VALUE:id_value
                  CLOSEPARAMTAG
                  ;

createDateParam::= PARAM NAME DATE_CREATION DBLEQUOTES MAYORQ
                   DATE_VALUE:date_value
                   CLOSEPARAMTAG
                   ;
```



```
createDateParam::=  PARAM NAME DATE_CREATION DBLEQUOTES MAYORQ
                    DATE_VALUE:date_value
                    CLOSEPARAMTAG
                    ;

userModParam::=    PARAM NAME USER_MODIFICATION DBLEQUOTES MAYORQ
                    ID_VALUE:id_value
                    CLOSEPARAMTAG
                    ;

modDateParam::=    PARAM NAME DATE_MODIFICATION DBLEQUOTES MAYORQ
                    DATE_VALUE:date_value
                    CLOSEPARAMTAG
                    ;

pageParam::=       PARAM NAME PAGE DBLEQUOTES MAYORQ
                    ID_VALUE:id_value
                    CLOSEPARAMTAG
                    ;

textAttribute::=   ATTRIBUTE NAME TEXT DBLEQUOTES MAYORQ
                    ID_VALUE:id_value
                    CLOSEATTRIBUTETAG
                    ;

alignmentAttribute::=ATTRIBUTE NAME ALIGNMENT DBLEQUOTES MAYORQ
                    alignmentLoc:alignment_loc
                    CLOSEATTRIBUTETAG
                    ;
```

```

colorAttribute ::=  ATTRIBUTE NAME COLOR DBLEQUOTES MAYORQ
                   COLOR_VALUE:id_value
                   CLOSEATTRIBUTETAG
                   ;

sourceAttribute ::=  ATTRIBUTE NAME SOURCE DBLEQUOTES MAYORQ
                    TAGID:id_value
                    CLOSEATTRIBUTETAG
                    ;

heightAttribute ::=  ATTRIBUTE NAME HEIGHT DBLEQUOTES MAYORQ
                    NUMBER_VALUE:num_value
                    CLOSEATTRIBUTETAG
                    ;

widthAttribute ::=  ATTRIBUTE NAME WIDTH DBLEQUOTES MAYORQ
                   NUMBER_VALUE:num_value
                   CLOSEATTRIBUTETAG
                   ;

parentAttribute ::=  ATTRIBUTE NAME PARENT DBLEQUOTES MAYORQ
                   ID_VALUE:id_value
                   CLOSEATTRIBUTETAG
                   ;

alignmentLoc ::=  CENTER_LOC:center_loc
                  |  LEFT_LOC:left_loc
                  |  RIGHT_LOC:right_loc
                  |  JUSTIFY_LOC:justify_loc
                  ;

```

Y por ultimo, una producción dedicada al manejo de etiquetas

```
addLabels ::= OPENLABELSTAG labelDecl:labels CLOSELABELSTAG
            |
            ;

labelDecl ::= labelDef:tag

            | labelDecl:tagDef labelDef:tag
            ;

labelDef ::= OPENLABELTAG TAGID:id_val CLOSELABELTAG
            ;
```