

Universidad de San Carlos de Guatemala
Centro Universitario de Occidente
Division de Ciencias de la Ingeniería
Ingeniería en Ciencias y Sistemas

Manua tecnico proyecto 1: “DWforms”

David Rodolfo Martinez Miranda
20632145

Manual técnico

En este manual se proporciona la información para que se puedan hacer correcciones o simplemente saber como funcionan los comandos principales para la aplicación pueda funcionar de manera adecuada. Se comenzaran por las gramaticas que se encuentran del lado del servidor para que se pueda procesar las solicitudes que se hacen en el servidor, lo principal para reconocer las solicitudes es la siguiente:

Reconocedores lexicos (Forma expresion regular)

La siguiente expresion regular reconoce la palabra reservada solicitud “solicitud”, se especifico al usuario que solo usara minusculas, pero es tipo ignore case.

`(s|S)(o|O)(l|L)(i|I)(c|C)(i|I)(t|T)(u|U)(d|D)`

La siguiente expresion regular reconoce la palabra reservada inicio de solicitud “ini_solicitud”, tambien sigue la restriccion anterior.

`(i|I)(n|N)(i|I)_(s|S)(o|O)(l|L)(i|I)(c|C)(i|I)(t|T)(u|U)(d|D)`

La siguiente expresion regular reconoce la palabra reservada fin solicitud “fin_solicitud”, tambien sigue la restriccion de solicitud.

`(f|F)(i|I)(n|N)_(s|S)(o|O)(l|L)(i|I)(c|C)(i|I)(t|T)(u|U)(d|D)`

La siguiente expresion regular reconoce la palabra reservada inicio de solicitudes “ini_solicitudes”, tambien sigue la restriccion de solicitud.

`(i|I)(n|N)(i|I)_(s|S)(o|O)(l|L)(i|I)(c|C)(i|I)(t|T)(u|U)(d|D)(e|E)(s|S)`

La siguiente expresion regular reconoce la palabra reservada fin solicitudes “fin_solicitudes”

`(f|F)(i|I)(n|N)_(s|S)(o|O)(l|L)(i|I)(c|C)(i|I)(t|T)(u|U)(d|D)(e|E)(s|S)`

Para reconocer algunos simbolos reservados son los siguientes, los cuales se usan para el analisis de la estructura.

`<, >, {, }, (,), [,], :, !, ‘, ’`

{espacioNormal} se especifica en tokens.

La siguiente expresion regular reconoce la palabra reservada credenciales usuario

`\">{espacioNormal}CREDENCIALES_USUARIO{espacioNormal}\"`

La siguiente expresion regular reconoce la palabra reservada crear usuario.

`\">{espacioNormal}CREAR_USUARIO{espacioNormal}\"`

La siguiente expresion regular reconoce la palabra reservada usuario.

`\{espacioNormal\}USUARIO{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada password.

`\{espacioNormal\}PASSWORD{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada fecha creacion.

`\{espacioNormal\}FECHA_CREACION{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada modificar usuario.

`\{espacioNormal\}MODIFICAR_USUARIO{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada usuario antiguo.

`\{espacioNormal\}USUARIO_ANTIGUO{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada usuario nuevo.

`\{espacioNormal\}USUARIO_NUEVO{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada nuevo password.

`\{espacioNormal\}NUEVO_PASSWORD{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada fecha modificacion.

`\{espacioNormal\}FECHA_MODIFICACION{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada eliminacion de usuario.

`\{espacioNormal\}ELIMINAR_USUARIO{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada login usuario.

`\{espacioNormal\}LOGIN_USUARIO{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada parameros formulario.

`\{espacioNormal\}PARAMETROS_FORMULARIO{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada nuevo formulario.

`\{espacioNormal\}NUEVO_FORMULARIO{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada id.

`\{espacioNormal\}ID{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada titulo.
`\{espacioNormal\}TITULO{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada nombre.
`\{espacioNormal\}NOMBRE{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada tema.
`\{espacioNormal\}TEMA{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada usuario creacion.
`\{espacioNormal\}USUARIO_CREACION{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada eliminar formulario.
`\{espacioNormal\}ELIMINAR_FORMULARIO{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada modificar formulario.
`\{espacioNormal\}MODIFICAR_FORMULARIO{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada parametros componente.
`\{espacioNormal\}PARAMETROS_COMPONENTE{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada agregar componente.
`\{espacioNormal\}AGREGAR_COMPONENTE{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada nombre campo.
`\{espacioNormal\}NOMBRE_CAMPO{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada formulario.
`\{espacioNormal\}FORMULARIO{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada clase.
`\{espacioNormal\}CLASE{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada indice.
`\{espacioNormal\}INDICE{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada texto visible.
`\{espacioNormal\}TEXTO_VISIBLE{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada alineacion.
`\{espacioNormal\}ALINEACION{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada requerido.
`\{espacioNormal}REQUERIDO{espacioNormal}`

La siguiente expresion regular reconoce la palabra reservada opciones.
`\{espacioNormal}OPCIONES{espacioNormal}`

La siguiente expresion regular reconoce la palabra reservada filas.
`\{espacioNormal}FILAS{espacioNormal}`

La siguiente expresion regular reconoce la palabra reservada columnas.
`\{espacioNormal}COLUMNAS{espacioNormal}`

La siguiente expresion regular reconoce la palabra reservada url.
`\{espacioNormal}URL{espacioNormal}`

La siguiente expresion regular reconoce la palabra reservada eliminar componente.
`\{espacioNormal}ELIMINAR_COMPONENTE{espacioNormal}`

La siguiente expresion regular reconoce la palabra reservada modificar componente.
`\{espacioNormal}MODIFICAR_COMPONENTE{espacioNormal}`

La siguiente expresion regular reconoce la palabra reservada consultas.
`\{espacioNormal}CONSULTAS{espacioNormal}`

La siguiente expresion regular reconoce la palabra reservada consultar datos.
`\{espacioNormal}CONSULTAR_DATOS{espacioNormal}`

La siguiente expresion regular reconoce la palabra reservada consulta-n.
`\{espacioNormal}CONSULTA-{digito}+{espacioNormal}`

La siguiente expresion regular reconoce la palabra reservada select.
`\{espacioNormal}SELECT`

La siguiente expresion regular reconoce la palabra reservada to.
`TO`

La siguiente expresion regular reconoce la palabra reservada form.
`FORM`

La siguiente expresion regular reconoce la palabra reservada where.
`WHERE`

Fin reservadas

Inicio patrones para tokens con lexemas

La siguiente expresio reconoce a espacios en las palabras reservadas anteriores, se utilizara como **espacioNormal**

“ “

La siguiente expresion regular reconoce el patron para id.

`(_|-\$)([a-zA-Z]|[0-9]|(|-\$))+`

La siguiente expresion regular reconoce el patron para digito.

`[0-9]`

La siguiente expresion regular reconoce el patron para fecha que es un parametro, con formato aaaa-mm-dd.

`{digito} {digito} {digito} {digito} \- {digito} {digito} \- {digito} {digito}`

La siguiente expresion regular reconoce el patron para el tema, en este caso solo se incluyen dos WHITE y DARK.

`(DARK|WHITE)`

La siguiente expresion regular reconoce el patron para la clase que es un parametro.

`(CAMPO_TEXTO|AREA_TEXTO|CHECKBOX|RADIO|FICHERO|IMAGEN|COMBO|BOTON)`

La siguiente expresion regular reconoce el patron para la alineacion que es un parametro.

`(CENTRO|IZQUIERDA|DERECHA|JUSTIFICAR)`

La siguiente expresion regular reconoce el patron para requerimento, es que un parametro.

`(SI|NO)`

La siguiente expresion regular reconoce el patron para operador logico.

`(AND|OR|NOT)`

La siguiente expresion regular reconoce el patron para las opciones en algunas de las clases, este tiene un estilo que separa a las opciones con operador logico “|”

`\("[^\r\n|<|>|!|\{|\}|\[|\]"*)*\|?\)+\"`

La siguiente expresion regular reconoce el patron para reconocer comilla, esto para los reportes, los cuales se esepificican mas adelante.

"

fin expresiones regulares

La siguiente parte acontinuacion es la gramatica que reconoce la estructura que mas adelante sera definida y mostrada con ejemplos.

Gramatica definida por $G = \{N, T, D, S\}$

N =

{S0 ,S1 ,S2 ,S7 ,S8 ,S12 ,S13 ,S14 ,S15 ,S16 ,S17 ,S18 ,S19 ,S20 ,S21 ,S22 ,S23 ,S24 ,S28 ,S29 ,S30 ,S31 ,S32 ,S33 ,S34 ,S35 ,S36 ,S37 ,S38 ,S39 ,S40 ,S41 ,S50 ,S51 ,S52 ,S53 ,S54 ,S55 ,S56 ,S57 ,S58 ,S59 ,S60 ,S61 ,S62 ,S63 ,S64 ,S65 ,S66 ,S67 ,S68 ,S69 ,S70 ,S71 ,S72 ,S73 ,S74 ,S75 ,S76 ,S77 ,S78 ,S79 ,S9 ,S10 ,S11 ,S3 ,S27 ,reservadas ,cosasSinEspacios ,cosasConEspacios}

T =

{COMILLA ,MENOR_QUE ,MAYOR_QUE ,SIGNO_INTERROGACION ,INICIO_SOLICITUD ,FIN_SOLICITUD ,DOS_PUNTOS ,COMA ,INICIO_SOLICITUDES ,FIN_SOLICITUDES ,CREAR_USUARIO ,LLAVE_ABIERTA ,LLAVE_CERRADA ,CREACIONES_USUARIO ,CORCHETE_ABIERTO ,CORCHETE_CERRADO ,MODIFICAR_USUARIO ,ELIMINAR_USUARIO ,LOGIN_USUARIO ,PARAMETROS_FORMULARIO ,NUEVO_FORMULARIO ,ELIMINAR_FORMULARIO ,MODIFICAR_FORMULARIO ,PARAMETROS_COMPONENTE ,AGREGAR_COMPONENTE ,ELIMINAR_COMPONENTE ,MODIFICAR_COMPONENTE ,CONSULTAS ,CONSULTAR_DATOS ,FECHA_CREACION ,USUARIO ,PASSWORD ,ID_PARAMETRO ,FECHA ,USUARIO_ANTIGUO ,USUARIO_NUEVO ,NUEVO_PASSWORD ,FECHA_MODIFICACION ,ID ,TITULO ,NOMBRE ,TEMA ,USUARIO_CREACION ,CADENA ,TEMA_PARAMETRO ,NOMBRE_CAMPO ,FORMULARIO ,CLASE ,CLASE_PARAMETRO ,INDICE ,TEXTO_VISIBLE ,ALINEACION ,ALINEACION_PARAMETRO ,REQUERIDO ,REQUERIDO_PARAMETRO ,OPCIONES ,OPERADOR_OR ,FILAS ,COLUMNAS ,URL ,SELECT ,TO ,FORM ,WHERE ,IGUAL ,OPERADOR_LOGICO ,OPERADOR_REPORTES ,GION ,CONSULTA ,CUALQUIER_COSA_SIN_E

SPACIOS ,CUALQUIER_COSA_CON_ESPACIOS,OPCIONES_PARAMETRO,
CUALQUIER_COSA_SIN_ESPACIOS_SIN_COMILLAS,
CUALQUIER_COSA_REPORTE ,DIGITO }

S → S0

D:

S0 ->

MENOR_QUE SIGNO_INTERROGACION INICIO_SOLICITUDES
MAYOR_QUE S1 MENOR_QUE SIGNO_INTERROGACION
FIN_SOLICITUDES MAYOR_QUE

|S2

S1 -> S2

|S1 S2

S2 -> MENOR_QUE SIGNO_INTERROGACION INICIO_SOLICITUD
DOS_PUNTOS S9

|error S8 S2

S9 -> S3

|S10

|S11

|S12

S3 -> CREAR_USUARIO MAYOR_QUE S50 S8
|MODIFICAR_USUARIO MAYOR_QUE S53 S8
|ELIMINAR_USUARIO MAYOR_QUE S56 S8
|LOGIN_USUARIO MAYOR_QUE S59 S8

S50 -> LLAVE_ABIERTA CREDENCIALES_USUARIO DOS_PUNTOS
CORCHETE_ABIERTO S52 CORCHETE_CERRADO LLAVE_CERRADA

S52 -> S51

|S51 COMA S52

S51 -> LLAVE_ABIERTA S13 LLAVE_CERRADA

S53 -> LLAVE_ABIERTA CREDENCIALES_USUARIO DOS_PUNTOS
CORCHETE_ABIERTO S55 CORCHETE_CERRADO LLAVE_CERRADA

S55 -> S54
|S54 COMA S55

S54 -> LLAVE_ABIERTA S15 LLAVE_CERRADA

S56 -> LLAVE_ABIERTA CREDENCIALES_USUARIO DOS_PUNTOS
CORCHETE_ABIERTO S58 CORCHETE_CERRADO LLAVE_CERRADA

S58 -> S57
|S57 COMA S58

S57 -> LLAVE_ABIERTA S17 LLAVE_CERRADA

S59 -> LLAVE_ABIERTA CREDENCIALES_USUARIO DOS_PUNTOS
CORCHETE_ABIERTO

S61 CORCHETE_CERRADO LLAVE_CERRADA

S61 -> S60
S61 -> S60 COMA S61

S60 -> LLAVE_ABIERTA S13 LLAVE_CERRADA

S10 -> NUEVO_FORMULARIO MAYOR_QUE S62 S8
S10 -> ELIMINAR_FORMULARIO MAYOR_QUE S65 S8
S10 -> MODIFICAR_FORMULARIO MAYOR_QUE S68 S8

S62 -> LLAVE_ABIERTA PARAMETROS_FORMULARIO DOS_PUNTOS
CORCHETE_ABIERTO S64 CORCHETE_CERRADO LLAVE_CERRADA

S64 -> S63
S64 -> S63 COMA S64

S63 -> LLAVE_ABIERTA S18 LLAVE_CERRADA

S65 -> LLAVE_ABIERTA PARAMETROS_FORMULARIO DOS_PUNTOS
CORCHETE_ABIERTO S67 CORCHETE_CERRADO LLAVE_CERRADA

S67 -> S66
S67 -> S66 COMA S67

S66 -> LLAVE_ABIERTA S20 LLAVE_CERRADA

S68 -> LLAVE_ABIERTA PARAMETROS_FORMULARIO DOS_PUNTOS
CORCHETE_ABIERTO S70 CORCHETE_CERRADO LLAVE_CERRADA

S70 -> S69

S70 -> S69 COMA S70

S69 -> LLAVE_ABIERTA S21 LLAVE_CERRADA

S11 -> AGREGAR_COMPONENTE MAYOR_QUE S71 S8

S11 -> ELIMINAR_COMPONENTE MAYOR_QUE S74 S8

S11 -> MODIFICAR_COMPONENTE MAYOR_QUE S77 S8

S71 -> LLAVE_ABIERTA PARAMETROS_COMPONENTE DOS_PUNTOS
CORCHETE_ABIERTO S73 CORCHETE_CERRADO LLAVE_CERRADA

S73 -> S72

S73 -> S72 COMA S73

S72 -> LLAVE_ABIERTA S23 LLAVE_CERRADA

S74 -> LLAVE_ABIERTA PARAMETROS_COMPONENTE DOS_PUNTOS
CORCHETE_ABIERTO S76 CORCHETE_CERRADO LLAVE_CERRADA

S76 -> S75

S76 -> S75 COMA S76

S75 -> LLAVE_ABIERTA S23 LLAVE_CERRADA

S77 -> LLAVE_ABIERTA PARAMETROS_COMPONENTE DOS_PUNTOS
CORCHETE_ABIERTO S79 CORCHETE_CERRADO LLAVE_CERRADA

S79 -> S78

S79 -> S78 COMA S79

S78 -> LLAVE_ABIERTA S23 LLAVE_CERRADA

S12 -> CONSULTAR_DATOS MAYOR_QUE S7 S29 S41 S8

S7 -> LLAVE_ABIERTA CONSULTAS DOS_PUNTOS CORCHETE_ABIERTO
LLAVE_ABIERTA

S41 -> LLAVE_CERRADA CORCHETE_CERRADO LLAVE_CERRADA

S8 -> MENOR_QUE FIN_SOLICITUD SIGNO_INTERROGACION MAYOR_QUE

S13 -> USUARIO DOS_PUNTOS cosasSinEspacios S14

S13 -> PASSWORD DOS_PUNTOS cosasSinEspacios S14

S13 -> S28 S14

S13 -> error COMA S13

S14 ->

S14 -> COMA S13

S15 -> USUARIO_ANTIGUO DOS_PUNTOS cosasSinEspacios S16

S15 -> USUARIO_NUEVO DOS_PUNTOS cosasSinEspacios S16

S15 -> NUEVO_PASSWORD DOS_PUNTOS cosasSinEspacios S16

S15 -> FECHA_MODIFICACION DOS_PUNTOS FECHA S16

S15 -> error COMA S15

S16 ->

S16 -> COMA S15

S17 -> USUARIO DOS_PUNTOS cosasSinEspacios

S18 -> S27 S19

S18 -> TITULO DOS_PUNTOS S40 S19

S18 -> NOMBRE DOS_PUNTOS cosasSinEspacios S19

S18 -> TEMA DOS_PUNTOS TEMA_PARAMETRO S19

S18 -> USUARIO_CREACION DOS_PUNTOS cosasSinEspacios S19

S18 -> S28 S19

S18 -> error COMA S18

S19 ->

S19 -> COMA S18

S20 -> ID DOS_PUNTOS ID_PARAMETRO

S21 -> ID DOS_PUNTOS ID_PARAMETRO S22

S21 -> TITULO DOS_PUNTOS S40 S22

S21 -> NOMBRE DOS_PUNTOS cosasSinEspacios S22

S21 -> TEMA DOS_PUNTOS TEMA_PARAMETRO S22

S22 ->

S22 -> COMA S21

S23 -> S27 S24

S23 -> NOMBRE_CAMPO DOS_PUNTOS S40 S24

S23 -> FORMULARIO DOS_PUNTOS ID_PARAMETRO S24

S23 -> CLASE DOS_PUNTOS CLASE_PARAMETRO S24

S23 -> TEXTO_VISIBLE DOS_PUNTOS S40 S24

S23 -> ALINEACION DOS_PUNTOS ALINEACION_PARAMETRO S24

S23 -> REQUERIDO DOS_PUNTOS REQUERIDO_PARAMETRO S24

S23 -> OPCIONES DOS_PUNTOS OPCIONES_PARAMETRO S24

S23 -> FILAS DOS_PUNTOS DIGITO S24

S23 -> COLUMNAS DOS_PUNTOS DIGITO S24

S23 -> URL DOS_PUNTOS cosasSinEspacios S24

S23 -> INDICE DOS_PUNTOS DIGITO S24

S23 -> error COMA S23

S27 -> ID DOS_PUNTOS ID_PARAMETRO

S28 -> FECHA_CREACION DOS_PUNTOS FECHA

S24 ->

S24 -> COMA S23

S29 -> CONSULTA DOS_PUNTOS S30 S39

S30 -> SELECT TO FORM OPERADOR_REPORTES ID_PARAMETRO
CORCHETE_ABIERTO

S32 CORCHETE_CERRADO S34

S31 ->

S31 -> CUALQUIER_COSA_SIN_ESPACIOS_SIN_COMILLAS

S32 -> S31 S33

S33 ->

S33 -> COMA S32

S34 -> COMILLA

S34 -> WHERE CORCHETE_ABIERTO S35 CORCHETE_CERRADO COMILLA

S35 -> CUALQUIER_COSA_SIN_ESPACIOS_SIN_COMILLAS S38 S36 S37

S36 -> DIGITO

S36 -> CUALQUIER_COSA_REPORTE

S37 ->

S37 -> OPERADOR_LOGICO S35

S38 -> MAYOR_QUE

S38 -> MENOR_QUE

S38 -> MENOR_QUE MAYOR_QUE

S38 -> IGUAL

S38 -> MAYOR_QUE IGUAL

S38 -> MENOR_QUE IGUAL

S39 ->

S39 -> COMA S29

S40 -> cosasSinEspacios

S40 -> CUALQUIER_COSA_CON_ESPACIOS

cosasSinEspacios -> CUALQUIER_COSA_SIN_ESPACIOS

cosasSinEspacios -> reservadas

reservadas -> MENOR_QUE

reservadas -> MAYOR_QUE

reservadas -> SIGNO_INTERROGACION

reservadas -> INICIO_SOLICITUD

reservadas -> FIN_SOLICITUD

reservadas -> DOS_PUNTOS

reservadas -> COMA

reservadas -> INICIO_SOLICITUDES

reservadas -> FIN_SOLICITUDES

reservadas -> CREAR_USUARIO

reservadas -> LLAVE_ABIERTA

reservadas -> LLAVE_CERRADA

reservadas -> CREDENCIALES_USUARIO

reservadas -> CORCHETE_ABIERTO

reservadas -> CORCHETE_CERRADO

reservadas -> MODIFICAR_USUARIO

reservadas -> ELIMINAR_USUARIO

reservadas -> LOGIN_USUARIO
reservadas -> PARAMETROS_FORMULARIO
reservadas -> NUEVO_FORMULARIO
reservadas -> ELIMINAR_FORMULARIO
reservadas -> MODIFICAR_FORMULARIO
reservadas -> PARAMETROS_COMPONENTE
reservadas -> AGREGAR_COMPONENTE
reservadas -> ELIMINAR_COMPONENTE
reservadas -> MODIFICAR_COMPONENTE
reservadas -> CONSULTAS
reservadas -> CONSULTAR_DATOS
reservadas -> FECHA_CREACION
reservadas -> USUARIO
reservadas -> PASSWORD
reservadas -> ID_PARAMETRO
reservadas -> FECHA
reservadas -> USUARIO_ANTIGUO
reservadas -> USUARIO_NUEVO
reservadas -> NUEVO_PASSWORD
reservadas -> FECHA_MODIFICACION
reservadas -> ID
reservadas -> TITULO
reservadas -> NOMBRE
reservadas -> TEMA
reservadas -> USUARIO_CREACION
reservadas -> TEMA_PARAMETRO
reservadas -> NOMBRE_CAMPO
reservadas -> FORMULARIO
reservadas -> CLASE
reservadas -> CLASE_PARAMETRO
reservadas -> INDICE
reservadas -> TEXTO_VISIBLE
reservadas -> ALINEACION
reservadas -> ALINEACION_PARAMETRO
reservadas -> REQUERIDO
reservadas -> REQUERIDO_PARAMETRO
reservadas -> OPCIONES
reservadas -> OPERADOR_O
reservadas -> FILAS
reservadas -> COLUMNAS
reservadas -> URL
reservadas -> SELECT

reservadas -> TO
reservadas -> FORM
reservadas -> WHERE
reservadas -> IGUAL
reservadas -> OPERADOR_LOGICO
reservadas -> OPERADOR_REPORTE
reservadas -> GION
reservadas -> CONSULTA
reservadas -> OPCIONES_PARAMETRO
reservadas -> DIGITO

La gramatica presentada permite reconocer el mensaje de entrada que vendra desde la aplicación cliente de escritorio, la gramatica acepta este tipo de estructuras:

```
<!ini_solicitud:"NOMBRE_DE_LA_SOLICITUD">  
  { "bloque_parametros":[{  
    "parametro_1_serie1": "valor_parametro_1_serie1",  
    "parametro_2_serie1": "valor_parametro_2_serie1",  
    "parametro_N_serie1": "valor_parametro_N_serie1",  
  },  
  {"parametro_1_serie2": "valor_parametro_1_serie2",  
   "parametro_2_serie2": "valor_parametro_1_serie2",  
   "parametro_N_serie2": "valor_parametro_N_serie2",  
  },  
  { "parametro_1_serieM": "valor_parametro_1_serieM",  
    "parametro_2_serieM": "valor_parametro_2_serieM",  
    "parametro_N_serieM": "valor_parametro_N_serieM",  
  }  
  ]  
}  
<fin_solicitud!>
```

Esto en ser caso de una sola solicitud, en caso de ser mas solicitudes podra aceptar la siguiente instrucción.

```
<!ini_solicitudes>  
  <!ini_solicitud:"NOMBRE_DE_LA_SOLICITUD_1">  
    { "bloque_parametros":[{
```

```

        ....
    },
    ....
]
}
<fin_solicitud!>
<!--ini_solicitud:"NOMBRE_DE_LA_SOLICITUD_2">
    { "bloque_parametros":[{
        ....
    },
    ....
    ]
    }
<fin_solicitud!>
....
....
<!--fin_solicitudes>

```

Para logeo de usuarios

```

<!--ini_solicitud:"LOGIN_USUARIO">
    { "CREDENCIALES_USUARIO":[{
        "USUARIO": "juanito619lopez",
        "PASSWORD": "A12345678910"
    }
    ]
    }
<fin_solicitud!>

```

Para crear usuarios

```

<!--ini_solicitud:"CREAR_USUARIO">
    { "CREDENCIALES_USUARIO":[{
        "USUARIO": "juanito619",
        "PASSWORD": "q12345678910"
    }
    ]
    }
<fin_solicitud!>

```



```
    }  
  ]  
}  
<fin_solicitud!>
```

Para modificar usuarios

```
<!ini_solicitud:"MODIFICAR_USUARIO">  
  { "CREDENCIALES_USUARIO":[{  
    "USUARIO_ANTIGUO": "juanito619",  
    "USUARIO_NUEVO": "juanito619lopez",  
    "NUEVO_PASSWORD": "A12345678910"  
  }  
]  
}  
<fin_solicitud!>
```

Para eliminar usuarios

```
<!ini_solicitud:"ELIMINAR_USUARIO">  
  { "CREDENCIALES_USUARIO":[{  
    "USUARIO": "juanito619"  
  }  
]  
}  
<fin_solicitud!>
```

Crear formularios

```
<!ini_solicitud:"NUEVO_FORMULARIO">  
  { "PARAMETROS_FORMULARIO":[{  
    "ID": "$form3",  
    "TITULO": "Formulario para encuesta 1",  
  }  
]  
}
```

```
        "NOMBRE": "formulario_encuesta_1",
        "TEMA": "DARK"
    }
]
}
<fin_solicitud!>
```

Para modificar formularios

```
<!ini_solicitud:"MODIFICAR_FORMULARIO">
    { "PARAMETROS_FORMULARIO":[{
        "ID": "$form3",
        "TITULO": "Formulario Modificado para encuesta 1",
        "NOMBRE": "formulario_encuesta_1_v2",
        "TEMA": "BLUE"
    }
    ]
    }
<fin_solicitud!>
```

Para eliminar formularios

```
<!ini_solicitud:"ELIMINAR_FORMULARIO">  
  { "PARAMETROS_FORMULARIO":[{  
    "ID": "$form3"  
  }  
]  
}  
<fin_solicitud!>
```

Agregar un componente

```
<!ini_solicitud:"AGREGAR_COMPONENTE">  
  { "PARAMETROS_COMPONENTE":[{  
    "ID": "$_grupo_paises",  
    "NOMBRE_CAMPO": "Pais",  
    "FORMULARIO": "$form3",  
    "CLASE": "COMBO",  
    "TEXTO_VISIBLE": "Pais de Origen ",  
    "ALINEACION": "CENTRO",  
    "REQUERIDO": "SI",  
    "OPCIONES": "Guatemala|El Salvador|Honduras|OTRO"  
  }  
]  
}
```

<fin_solicitud!>

Para eliminar componentes

```
<!ini_solicitud:"ELIMINAR_COMPONENTE">
  { "PARAMETROS_COMPONENTE":[{
    "ID": "$_grupo_paises",
    "FORMULARIO": "$form3"
  }
  ]
}
<fin_solicitud!>
```

Para modificar componentes

```
<!ini_solicitud:"MODIFICAR_COMPONENTE">
  { "PARAMETROS_COMPONENTE":[{
    "ID": "$_grupo_paises",
    "FORMULARIO": "$form3",
    "CLASE": "CHECKBOX",
    "INDICE": "1",
    "ALINEACION": "DERECHA",
    "OPCIONES": "Guatemala|El Salvador|Honduras|OTRO"
  }
  ]
}
<fin_solicitud!>
```

Al momento de procesar estas solicitudes se cuenta con una funcion y una clase que ayuda a verificar los parametros que seran necesario, asi como tambien que no se repitan los parametros, se presentan acontinuacion.

Funcion que verifica los parametros repetidos

```
private void agregarDatos(String tipo,String dato)
```

- tipo hace referencia al tipo de parametros en si.
- Dato es el valor de dicho parametro

Estos datos son almacenados en listados que son utilizados para poder guardar las instrucciones que va a procesar. Dentro de esta funcion tambien se realiza una verificacion si el parametro, (no el valor del parametro) esta repetido activa un booleano que este se usa cuando se quiere agregar la instrucción a un objeto que colecciona las instrucciones a procesar, la cual se describe a continuacion.

Objeto que guarda las instrucciones

Al momento de que se cumple con cierta estructura de datos esta es almacenada en un objeto que su clase es la de Solicitudes. La clase solicitudes contiene lo siguiente:

Clase Solicitudes

- Un listado tipo String que almacena a los parametros.
- Un listado tipo String que almacena los datos de los parametros.
- Un valor tipo String que almacena el tipo de instrucción que se debe procesar.
- Un arreglo tipo int, por si se producen errores, este indica y contiene la linea y columna de dicho error.

Cuando se esta analizando y se encuentra con una reduccion que esta creara un match para la produccion o un error y se instancia a:

- `public void report_error(String message, Object info)`
- `public void syntax_error(Symbol cur_token)`
- `public void report_fatal_error(String message, Object info)`

Entonces entra en accion el siguiente objeto **reporte** que es instanceado por la clase **Reportes**.

Esta clase contiene lo siguiente.

Clase Reportes

- Contiene un listado tipo arreglo String de errores
- Contiene un listado tipo arreglo String de solicitudes procesadas

El listado para los errores almacenan datos de los errores que pueden ser dos tipos en este caso son **semanticos o sintacticos**, los semanticos hacen referencias a errores de parametros repetidos o cuando no son los parametros suficientes según sea la instruccion. Los errores sintacticos se producen cuando se hacen faltas a en la estructura de entrada.

Lo indicado anterior es utilizado al momento de ejecutar el parser.

Como se especifico anteriormente en las solicitudes, cuando no existen errores sintacticos o semanticos primarios, no se ejecutan las ordenes, en este caso se utiliza una clase que es la encargada de procesar las solicitudes, en esta clase se verifican si existen los errores mencionados anteriormente. La clase contiene lo siguiente:

Clase analizadorSolicitudes

- Una funcion estatica, que se encarga de ejecutar el analizador lexico, el parser, y los reportes, asi mismo implementa una funcion que se encarga de hacer una salida para la aplicación cliente que tambien tiene un parser que ejecuta las respuestas madadas desde el servidor.

Cuando se procesan las solicitudes en una clase llamada procesador solicitudes, esta recibe los parametros que fueron ejecutados en su momento en el parser.

Clase procesarSolicitudes

Esta clase contiene lo siguiente:

- Un listado de solicitudes.
- Un boolean para saber si esta logueado.
- Un String que contiene el usuario de salida.
- Un String que contiene el usuario de entrada.
- Un boolean que verifica si el usuario logueado es eliminado.

Al ejecutar la funcion publica

```
public void procesarSolicitudes(Reportes reporteEntrada)
```

Este procesa las entradas de las solicitudes, recibe un objeto de tipo Reportes, que ya fue instanceado anteriormente para poder agregar mas reportes o errores en tiempo de ejecucion de las instrucciones.

En esta funcion se recorre el listado de solicitudes y con el atributo de tipo de instruccion este entra a la opcion que se solicito, siendo estas:

- CRUD_USUARIO
- LOGIN_USUARIO
- CRUD_FORMULARIO
- CRUD_COMPONENTE

Estas funciones llaman a funciones estaticas de cada tipo, en este caso manejadores según el tipo, manejador para usuario, formulario, componente. Estas se encargan de hacer los procesos necesarios.

Clase manejadorUsuario

Esta clase contiene:

public static void consultaUsuario()

Esta funcion se encarga de procesar las solicitudes crud correspondientes al usuario, crear, editar e eliminar, recibe los parametros correspondientes para poder ejecutar la orden deseada.

String tipo, List<String> datos, List<String> parametros, Reportes reporteEntrada, int[] posXY, String usuarioEntrada, StringBuilder usuarioSalida

- String tipo, contiene el tipo de CRUD.
- List<String> datos, contiene los valores asociados a los parametros.
- List<String> parametros, contiene el listado de parametros asociados a los datos correspondientes.
- Reportes, para agregar reportes.
- Pos xy, para indicar informacion al usuario, de una posicion especifica en el area de entrada.
- String Usuario entrada se utiliza al momento de eliminarse.
- String Usuario salida para poder setear el usuario de salida.

Para los procesos de recuperacion de datos se utiliza un parser para cualquier tipo de CRUD y login en el usuario que se almacena en formato indigo, mas adelante se especifica esto.

- **public static editarUsuario()**
 - Se utilizar para editar los datos necesarios en el usuario.
- **public static boolean parametrosVerificador()** es utilizada para verificar que venga los parametros requeridos al momento de querer agregar una orden.
- **public static boolean login()** sirve para verificar los datos del usuario y decidir si se loguea o no.
- **public static Usuario recuperacionUsuario()**, setea un usuario para su uso que necesite darse.
- **public static String parametrosObligatorios()**, para especificar al usuario que parametros son obligatorios al violar una regla se los parametros obligatorios.

Clase ManejadorFormulario

Esta clase contiene:

public static void consultaFormulario()

Esta funcion se encarga de procesar las solicitudes crud correspondientes al formulario, crear, editar e eliminar, recibe los parametros correspondientes para poder ejecutar la orden deseada.

String tipo, List<String> datos, List<String> parametros, Reportes
reporteEntrada, int[] posXY

- String tipo, contiene el tipo de CRUD.
- List<String> datos, contiene los valores asociados a los parametros.
- List<String> parametros, contiene el listado de parametros asociados a los datos correspondientes.
- String Usuario para poder setear el usuario en el formulario si no viene especificado en los parametros.
- Reportes, para agregar reportes.
- Pos xy, para indicar informacion al usuario, de una posicion especifica en el area de entrada.

Para los procesos de recuperacion de datos se utiliza un parser para cualquier tipo de CRUD en el formulario que se almacena en formato indigo, mas adelante se especifica esto.

- **public static boolean parametrosVerificador()**, esta funcion verifica los parametros obligados en la solicitud, antes de agregarse.
- **public static String parametrosObligatorios()**, para indicarle al usuario que parametros son requeridos.
- **public static Formulario recuperacionFormulario()**, sirve para cuando se hace el parser este retorne el formulario recuperado.
- **public static void modificarUsuario()**, al modificar al usuario en una solicitud, este tendra que verse relfejados en los formularios que el creo.
- **public static void elimiarFormulariosUsuarioEleminacion()**, en el caso de eliminarse un usuario, se deben eliminar los formularios creados.

Clase ManejadorComponente

Esta funcion se encarga de procesar las solicitudes crud correspondientes al componente, crear, editar e eliminar, recibe los parametros correspondientes para poder ejecutar la orden deseada.

String tipo, List<String> datos, List<String> parametros, Reportes

- String tipo, contiene el tipo de CRUD.
- List<String> datos, contiene los valores asociados a los parametros.
- List<String> parametros, contiene el listado de parametros asociados a los datos correspondientes.
- Reportes, para agregar reportes.

Para los procesos de recuperacion de datos se utiliza un parser para cualquier tipo de CRUD en el componente que se almacena en formato indigo, mas adelante se especifica esto.

- **public static boolean parametrosVerificador()**, esta funcion verifica los parametros obligados en la solicitud, antes de agregarse.
- **public static String parametrosObligatorios()**, para indicarle al usuario que parametros son requeridos.
- **public static Formulario recuperacionComponente()**, sirve para cuando se hace el parser este retorne el formulario recuperado.
- **public static void modificarConClase()**, al modificar al componente, este verifica que la clase pertenezca a los datos que se quieran modificar.

- **public static void agregarFaltantes()**, agreaga datos correspondientes a la clase, si no se especifican en los parametros entonces este los agrega del ya existente.
- **private static boolean parametrosVerificadorObjeto_O_Entrada()**, este verifica que los parametros coincidan con los datos de entrada sino es asi entonces no podra hacer modificaciones al componente.
- **private static void modificarIndiceComponente()**, modifica a la posicion que se quiera indicar, este tambien valida que el indice ingresado no tenga problemas, al querer setear una posicion no valida.
- **private static void modificadorComponente()**, encargado de poder modificar lo que se desee en el componente.
- **private static Formulario existenciaFormulario()**, si existe el fomulario al cual se quiere hacer la solicitud del componente entonces retorna el formulario sino retorna nulo.
- **private static void reescribir()**, reescribe el formluario con los componentes que se agregaron o modificaro o eliminaron.
- **public static Componente parserCompoente()**, al momento de recuperar el componente este se encarga de parsearlo en objeto.

Gramatica para recuperar un usuario

Lexico:

Para reconocer algunos simbolos reservados son los siguientes, los cuales se usan para el analisis de la estructura.

{, }, :, ' , '

{espacioNormal} se especifica en tokens

La siguiente expresion regular reconoce la palabra reservada password.

`\{espacioNormal\}PASSWORD{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada fecha creacion.

`\{espacioNormal\}FECHA_CREACION{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada fecha modificacion.

`\{espacioNormal\}FECHA_MODIFICACION{espacioNormal}\`

Fin reservadas

Inicio patrones para tokens con lexemas

La siguiente expresio reconoce a espacios en las palabras reservadas anteriores, se utilizara como espacioNormal

“ “

La siguiente expresion regular reconoce el patron para digito.

[0-9]

Analizador sintactico

La siguiente parte acontinuacion es la gramatica que reconoce la estructura que mas adelante sera definida y mostrada con ejemplos.

Gramatica definida por $G = \{N, T, D, S\}$

$N = \{S0, S1, S2, cualquierCosaSinEspacios\}$

$T = \{DOS_PUNTOS, COMA, LLAVE_ABIERTA, LLAVE_CERRADA, PASSWORD, FECHA_CREACION, FECHA_MODIFICACION, CUALQUIER_COSA_SIN_ESPACIOS, FECHA\}$

$S = \{S0\}$

D :

S1 ->

S1 -> PASSWORD DOS_PUNTOS cualquierCosaSinEspacios S2

S1 -> FECHA_CREACION DOS_PUNTOS FECHA S2

S1 -> FECHA_MODIFICACION DOS_PUNTOS FECHA S2

S1 -> error COMA S1

S2 ->

S2 -> COMA S1

cualquierCosaSinEspacios -> PASSWORD

cualquierCosaSinEspacios -> FECHA_CREACION

cualquierCosaSinEspacios -> FECHA_MODIFICACION

cualquierCosaSinEspacios -> CUALQUIER_COSA_SIN_ESPACIOS

cualquierCosaSinEspacios -> FECHA

Esta gramatica utiliza funciones que estan incluidas en la clase manejador usuario que ya se especifico anteriormente en este documento.

Gramatica para analizar los formularios

Lexico:

Para reconocer algunos simbolos reservados son los siguientes, los cuales se usan para el analisis de la estructura.

(,), {, }, :, ',

{espacioNormal} se especifica en tokens

La siguiente expresion regular reconoce el patron para fecha que es un parametro, con formato aaaa-mm-dd.

{digito} {digito} {digito} {digito} \- {digito} {digito} \- {digito} {digito}

La siguiente expresion regular reconoce el patron para el tema, en este caso solo se incluyen dos WHITE y DARK.

(DARK|WHITE)

La siguiente expresion regular reconoce el patron para la clase que es un parametro.

(CAMPO_TEXTO|AREA_TEXTO|CHECKBOX|RADIO|FICHERO|IMAGEN|
COMBO|BOTON)

La siguiente expresion regular reconoce el patron para la alineacion que es un parametro.

(CENTRO|IZQUIERDA|DERECHA|JUSTIFICAR)

La siguiente expresion regular reconoce el patron para requerimiento, es que un parametro.

(SI|NO)

Analizador sintactico para recuperar formulario y componente

La siguiente parte acontinuacion es la gramatica que reconoce la estructura que mas adelante sera definida y mostrada con ejemplos.

Gramatica definida por $G = \{N, T, D, S\}$

$N = \{S0, S1, S2, S3, S4, S5, S6, S7, S8, \text{cosasSinEspacios}, \text{reservadas}\}$

$T =$

{DOS_PUNTOS, COMA, ID_FORMULARIO, TITULO, NOMBRE, TEMA, USUARIO_CREACION, ID, ID_PARAMETRO, TEMA_PARAMETRO, LLAVE_ABIERTA, LLAVE_CERRADA, NOMBRE_CAMPO, FORMULARIO, CLASE, CLASE_PARAMETRO, INDICE, TEXTO_VISIBLE, ALINEACION, ALINEACION_PARAMETRO, REQUERIDO, REQUERIDO_PARAMETRO, OPCIONES, ESTRUCTURA, PARENTESIS_ABIERTO, PARENTESIS_CERRADO, OPCIONES_PARAMETRO, FILAS, COLUMNAS, URL, DIGITO, FECHA_CREACION, FECHA, CUALQUIER_COSA_SIN_ESPACIOS, CUALQUIER_COSA_CON_ESPACIOS }

$S = \{S0\}$

D:

S0 -> LLAVE_ABIERTA S1 LLAVE_CERRADA
S1 ->
S1 -> S2
S2 -> TITULO DOS_PUNTOS S8 S3
S2 -> NOMBRE DOS_PUNTOS cosasSinEspacios S3
S2 -> TEMA DOS_PUNTOS TEMA_PARAMETRO S3
S2 -> USUARIO_CREACION DOS_PUNTOS cosasSinEspacios S3
S2 -> FECHA_CREACION DOS_PUNTOS FECHA S3
S2 -> ESTRUCTURA DOS_PUNTOS PARENTESIS_ABIERTO S4
PARENTESIS_CERRADO S3
S2 -> error COMA S1
S3 ->
S3 -> COMA S2
S4 -> LLAVE_ABIERTA S5 LLAVE_CERRADA
S4 -> S4 COMA LLAVE_ABIERTA S5 LLAVE_CERRADA
S6 ->
S6 -> COMA S4
S5 -> ID DOS_PUNTOS ID_PARAMETRO S7
S5 -> NOMBRE_CAMPO DOS_PUNTOS S8 S7
S5 -> FORMULARIO DOS_PUNTOS ID_PARAMETRO S7
S5 -> CLASE DOS_PUNTOS CLASE_PARAMETRO S7
S5 -> TEXTO_VISIBLE DOS_PUNTOS S8 S7
S5 -> ALINEACION DOS_PUNTOS ALINEACION_PARAMETRO S7
S5 -> REQUERIDO DOS_PUNTOS REQUERIDO_PARAMETRO S7
S5 -> OPCIONES DOS_PUNTOS OPCIONES_PARAMETRO S7

S5 -> FILAS DOS_PUNTOS DIGITO S7
S5 -> COLUMNAS DOS_PUNTOS DIGITO S7
S5 -> URL DOS_PUNTOS cosasSinEspacios S7
S5 -> INDICE DOS_PUNTOS DIGITO S7
S5 -> error COMA S5
S7 ->
S7 -> COMA S5
S8 -> CUALQUIER_COSA_CON_ESPACIOS
S8 -> cosasSinEspacios
cosasSinEspacios -> CUALQUIER_COSA_SIN_ESPACIOS
cosasSinEspacios -> reservadas
reservadas -> DOS_PUNTOS
reservadas -> COMA
reservadas -> ID_FORMULARIO
reservadas -> TITULO
reservadas -> NOMBRE
reservadas -> TEMA
reservadas -> USUARIO_CREACION
reservadas -> ID
reservadas -> ID_PARAMETRO
reservadas -> TEMA_PARAMETRO
reservadas -> LLAVE_ABIERTA
reservadas -> LLAVE_CERRADA
reservadas -> NOMBRE_CAMPO
reservadas -> FORMULARIO
reservadas -> CLASE
reservadas -> CLASE_PARAMETRO
reservadas -> INDICE
reservadas -> TEXTO_VISIBLE
reservadas -> ALINEACION
reservadas -> ALINEACION_PARAMETRO
reservadas -> REQUERIDO
reservadas -> REQUERIDO_PARAMETRO
reservadas -> OPCIONES
reservadas -> ESTRUCTURA
reservadas -> PARENTESIS_ABIERTO
reservadas -> PARENTESIS_CERRADO
reservadas -> OPCIONES_PARAMETRO
reservadas -> FILAS
reservadas -> COLUMNAS
reservadas -> URL
reservadas -> DIGITO

reservadas -> FECHA_CREACION
reservadas -> FECHA

Para recuperar un formulario se utilizan funciones que fueron especificadas en manejadorFormulario asi como tambien para recuperar componentes las que fueron usadas en manejadorComponente.

Al momento de ejecutar las instrucciones, se genera un texto con un formato especial para salida la cual esta mostrada en parserRespuestas.

- **Public String parserRespuestas()**
- **public String parseadorUsuario()**
- **public String parseadorFormularioSalida()**

Estas funciones se encargan de generar los formatos para almacenar en la base de datos del sistema.

Para ver el funcionamiento de la aplicación escritorio sigue mas adelante.

Manual tecnico para aplicación escritorio

De este lado solo se tienen como clases principales para el analisis un JFrame que permite la edicion de texto en un area de texto, tambien maneja tablas para mostrar informacion necesaria para cada proceso, tambien se cuenta con un analizador para respuestas, las cuales son interpretadas que vienen desde el servidor.

Gramatica para recibir respuestas

Lexico

Para reconocer algunos simbolos reservados son los siguientes, los cuales se usan para el analisis de la estructura.

<, >, !, (,), {, }, :, ' ,

La siguiente expresion regular reconoce la palabra reservada inicio respuesta "ini_respuesta".

ini_respuesta

La siguiente expresion regular reconoce la palabra reservada fin respuesta "fin_respuesta".

fin_respuesta

La siguiente expresion regular reconoce la palabra reservada inicio respuestas "ini_respuestas".

ini_respuesta

La siguiente expresion regular reconoce la palabra reservada fin respuestas "fin_respuestas".

fin_respuesta

La siguiente expresion regular reconoce la palabra reservada info.

\{espacioNormal\}info{espacioNormal}\

La siguiente expresion regular reconoce la palabra reservada info.

\{espacioNormal\}error{espacioNormal}\

La siguiente expresion regular reconoce la palabra reservada crear usuario.

\{espacioNormal\}CREAR_USUARIO{espacioNormal}\

La siguiente expresion regular reconoce la palabra reservada modificar usuario.
`\{espacioNormal\}MODIFICAR_USUARIO{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada eliminacion de usuario.
`\{espacioNormal\}ELIMINAR_USUARIO{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada login usuario.
`\{espacioNormal\}LOGIN_USUARIO{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada nuevo formulario.
`\{espacioNormal\}NUEVO_FORMULARIO{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada eliminar formulario.
`\{espacioNormal\}ELIMINAR_FORMULARIO{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada modificar formulario.
`\{espacioNormal\}MODIFICAR_FORMULARIO{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada agregar componente.
`\{espacioNormal\}AGREGAR_COMPONENTE{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada eliminar componente.
`\{espacioNormal\}ELIMINAR_COMPONENTE{espacioNormal}\`

La siguiente expresion regular reconoce la palabra reservada modificar componente.
`\{espacioNormal\}MODIFICAR_COMPONENTE{espacioNormal}\`

Analisis sintactico

La siguiente parte acontinuacion es la gramatica que reconoce la estructura que mas adelante sera definida y mostrada con ejemplos.

Gramatica definida por $G = \{N, T, D, S\}$

$N = \{S_0, S_1, S_2, S_5, S_7, S_{13}, S_3, S_4, S_6, S_8, S_9, S_{10}, S_{11}, S_{12}\}$

T =

{MENOR_QUE , MAYOR_QUE , SIGNO_INTERROGACION , DOS_PUNTOS ,
COMA , LLAVE_ABIERTA , LLAVE_CERRADA , INICIO_RESPUESTA ,
FIN_RESPUESTA , INICIO_RESPUESTAS , FIN_RESPUESTAS , INFO ,
CREAR_USUARIO , MODIFICAR_USUARIO , ELIMINAR_USUARIO ,
LOGIN_USUARIO , NUEVO_FORMULARIO, ELIMINAR_FORMULARIO ,
MODIFICAR_FORMULARIO , AGREGAR_COMPONENTE ,
ELIMINAR_COMPONENTE , MODIFICAR_COMPONENTE ,
ERROR_PARAMETRO , CUALQUIER_COSA_CON_ESPACIOS ,
CUALQUIER_COSA_SIN_ESPACIOS }

S = {S0}

D:

S0 -> MENOR_QUE SIGNO_INTERROGACION INICIO_RESPUESTAS
MAYOR_QUE S1 MENOR_QUE SIGNO_INTERROGACION
FIN_RESPUESTAS MAYOR_QUE

S0 -> S2

S1 -> S2

S1 -> S1 S2

S2 -> MENOR_QUE SIGNO_INTERROGACION INICIO_RESPUESTA
DOS_PUNTOS S3

S2 -> error S13

S3 -> S4

S3 -> S9

S3 -> S11

S4 -> S8 MAYOR_QUE LLAVE_ABIERTA S5 LLAVE_CERRADA S13

S5 -> INFO DOS_PUNTOS S6 S7

S6 -> CUALQUIER_COSA_CON_ESPACIOS

S6 -> CUALQUIER_COSA_SIN_ESPACIOS

S7 ->

S7 -> COMA S5

S8 -> CREAR_USUARIO

S8 -> MODIFICAR_USUARIO

S8 -> ELIMINAR_USUARIO

S8 -> LOGIN_USUARIO

S8 -> ERROR_PARAMETRO

S9 -> S10 MAYOR_QUE LLAVE_ABIERTA S5 LLAVE_CERRADA S13

S10 -> NUEVO_FORMULARIO

S10 -> ELIMINAR_FORMULARIO

S10 -> MODIFICAR_FORMULARIO

S11 -> S12 MAYOR_QUE LLAVE_ABIERTA S5 LLAVE_CERRADA S13
S12 -> AGREGAR_COMPONENTE
S12 -> ELIMINAR_COMPONENTE
S12 -> MODIFICAR_COMPONENTE
S13 -> MENOR_QUE SIGNO_INTERROGACION FIN_RESPUESTA
MAYOR_QUE

Clase manejadorRespuestas

Esta clase contiene:

public static void analizarRespuestas()

Esta funcion se encarga de procesar las respuestas correspondientes, recibe los parametros correspondientes para poder ejecutar la orden deseada.

Luego los agrega a una tabla para que estos sean agregados, la estructura que se maneja es como la siguiente:

```
<!ini_respuestas>
  <!ini_respuesta:"NOMBRE_DEL_TIPO_SOLICITUD_PROCESADA">
    { "parametros":[{
      "tipoProceso": "datosProcesos",
      "info":"informacion1".
      "info":"informacion2".
      "info":"informacion3"
    }],
  ]
}
<fin_respuesta!>
<!ini_respuesta:"error">
  { "parametros":[{
    "tipoError": "tipoError",
    "info":"informacion1".
    "info":"informacion2".
    "info":"informacion3"
  }],
}
]
```

<fin_respuesta!>
<!fin_respuestas>

Tambien el usuario al importar un formulario, podra hacerlo desde un fileChooser y luego este sera procesado en el servidor e igual forma mandara una respuesta.