Universidad de San Carlos de Guatemala
Centro Universitario de Occidente
División de Ciencias de la Ingeniería

Organización de Lenguajes y Compiladores I
Laboratorio

Proyecto 1
Manejador de Contenido Web
Manual Técnico

Jorge Anibal Bravo Rodríguez
202131782

1. HERRAMIENTAS USADAS

Java:
Java es un lenguaje de programación de propósito general que se ha vuelto muy popular debido a su portabilidad y capacidad para funcionar en diferentes plataformas. Se utiliza en una amplia gama de aplicaciones, desde aplicaciones empresariales hasta dispositivos móviles y sistemas integrados. Java es conocido por su sintaxis similar a C++ y su enfoque en la programación orientada a objetos.

NetBeans:
NetBeans es un entorno de desarrollo integrado (IDE) que proporciona herramientas para desarrollar aplicaciones en Java y otras tecnologías. Ofrece características como edición de código, depuración, compilación y gestión de proyectos. NetBeans es conocido por su facilidad de uso y su capacidad para admitir múltiples lenguajes de programación.

Visual Studio Code:
Visual Studio Code es un editor de código fuente desarrollado por Microsoft que es altamente personalizable y admite una amplia gama de lenguajes de programación. Ofrece características como resaltado de sintaxis, finalización de código, depuración y control de versiones integrado. Visual Studio Code es conocido por su rendimiento y su amplia gama de extensiones.

JFlex y Cup:
JFlex y Cup son herramientas utilizadas para crear analizadores léxicos y sintácticos en Java. JFlex se utiliza para generar analizadores léxicos basados en expresiones regulares, mientras que Cup se utiliza para generar analizadores sintácticos basados en gramáticas. Estas herramientas son útiles para el desarrollo de compiladores y analizadores de lenguaje.

HTML:
HTML (HyperText Markup Language) es el lenguaje estándar para crear páginas web. Se utiliza para estructurar el contenido de una página web y definir su presentación. HTML utiliza etiquetas para marcar diferentes tipos de contenido, como encabezados, párrafos, enlaces e imágenes. Es fundamental para el desarrollo web.

Script Bash:
Bash es un intérprete de comandos y un lenguaje de programación de shell para sistemas operativos tipo Unix. Se utiliza para automatizar tareas a través de scripts, como la administración del sistema, el procesamiento de archivos y la ejecución de programas. Los scripts Bash son comúnmente utilizados en sistemas Linux y Unix, usado en este proyecto para la compilación de los analizadores hechos con flex y cup.

Fedora Linux:
Fedora es una distribución de Linux patrocinada por Red Hat y conocida por su enfoque en la innovación y la adopción temprana de nuevas tecnologías. Es una distribución de propósito general que se utiliza tanto en entornos de escritorio como en servidores. Fedora es conocida por su enfoque en la comunidad y su compromiso con el software libre y de código abierto.

Apache HTTP:
Apache HTTP Server, comúnmente conocido como Apache, es un servidor web de código abierto ampliamente utilizado. Es conocido por su estabilidad, seguridad y flexibilidad, y es compatible con una amplia gama de sistemas operativos. Apache es altamente configurable y es utilizado por una gran cantidad de sitios web en todo el mundo.

Git:
Git es un sistema de control de versiones distribuido ampliamente utilizado para el seguimiento de cambios en el código fuente durante el desarrollo de software. Permite a los desarrolladores colaborar en proyectos, realizar un seguimiento de las modificaciones, fusionar ramas y revertir cambios. Git es conocido por su velocidad, escalabilidad y capacidad para manejar proyectos grandes.

GitHub:
GitHub es una plataforma de desarrollo colaborativo que utiliza Git para el control de versiones. Permite a los desarrolladores alojar y revisar el código, gestionar proyectos, realizar seguimiento de problemas y colaborar en equipos distribuidos. GitHub es conocido por su amplia comunidad, su integración con otras herramientas y su soporte para proyectos de código abierto.

2. GRAMÁTICAS USADAS

- SERVIDOR DE MANEJO DE CONTENIDO

Definición de los tokens (expresiones regulares)

```
// Uso general
tag_open      =\<
tag_close     =\>
tag_inclose   =\/
equals        =\=
comillas      =\"
nombre_def    =nombre
valor_def     =valor
val_open      =\[
val_close     =\]
LineTerminator = \r|\n|\r\n
WhiteSpace = {LineTerminator} | [ \t\f]
WhiteSpaceOp ={WhiteSpace}*
// valores parametros
identificador= [_\-$][a-zA-Z0-9_\-$]*
prm_val_id  = {val_open} {identificador} {val_close}
prm_val_fch = {val_open} ([0-9]{4})-([0-1][0-9]|2[0-3])-([0-9]{1,2}) {val_close}
prm_val_tit = {val_open} [A-Za-z0-9\s\.\,\:\/\-\_]+ {val_close}
```

```
<!-- - - - - - - - - - - - - - - DECLARACION ACCIONES - - - - - - - - - - - - - - - -->

<!--    LEXEMA                          DECLARACION LEXICA          SYMBOL -->

<!-- Sitios web -->
<accion nombre="NUEVO_SITIO_WEB">         (acc_add_wbst)     ->   ACC_ADD_WBST
<accion nombre="BORRAR_SITIO_WEB">        (acc_del_wbst)     ->   ACC_DEL_WBST

<!-- Paginas -->
<accion nombre="NUEVA_PAGINA">            (acc_add_pagw)     ->   ACC_ADD_PAGW
<accion nombre="BORRAR_PAGINA">           (acc_del_pagw)     ->   ACC_DEL_PAGW
<accion nombre="MODIFICAR_PAGINA">        (acc_mod_pagw)     ->   ACC_MOD_PAGW

<!-- Componentes -->
<accion nombre="AGREGAR_COMPONENTE">      (acc_add_comp)     ->   ACC_ADD_COMP
<accion nombre="BORRAR_COMPONENTE">       (acc_del_comp)     ->   ACC_DEL_COMP
<accion nombre="MODIFICAR_COMPONENTE">    (acc_mod_comp)     ->   ACC_MOD_COMP

<!-- Cerrar accion -->
</accion>                                  (accion_cl)       ->    ACCION_CL

<!-- Acciones -->
<acciones>                                 acciones_op       ->   ACCIONES_OP
</acciones>                                acciones_cl       ->   ACCIONES_CL



<!- - - - - - - - - - - - - - - DECLARACION PARAMETROS - - - - - - - - - - - - - - - -->
<!-- Son configuraciones de lo declarado en acciones -->

<!--    LEXEMA                          DECLARACION LEXICA            SYMBOL  -->
<parametro nombre="ID">                    param_type_id     ->    PARAM_TYPE_ID
<parametro nombre="TITULO">                param_type_tit    ->    PARAM_TYPE_TIT
<parametro nombre="SITIO">                 param_type_sit    ->    PARAM_TYPE_SIT
<parametro nombre="PADRE">                 param_type_pad    ->    PARAM_TYPE_PAD
<parametro nombre="USUARIO_CREACION">      param_type_usc    ->    PARAM_TYPE_USC
<parametro nombre="FECHA_CREACION">        param_type_fcr    ->    PARAM_TYPE_FCR
<parametro nombre="FECHA_MODIFICACION">    param_type_fmd    ->    PARAM_TYPE_FMD
<parametro nombre="USUARIO_MODIFICACION">  param_type_usm    ->    PARAM_TYPE_USM
<parametro nombre="PAGINA">                param_type_pag    ->    PARAM_TYPE_PAG
<parametro nombre="CLASE">                 param_type_cls    ->    PARAM_TYPE_CLS

</parametro>                                parametro_cl     ->     PARAMETRO_CL

<parametros>                               parametros_op     ->     PARAMETROS_OP
</parametros>                              parametros_cl     ->     PARAMETROS_CL



<!- - - - - - - - - - - - - - -  DECLARACION ETIQUETAS - - - - - - - - - - - - - - - -->
<!--    LEXEMA                          DECLARACION LEXICA            SYMBOL          VALOR-->
<etiquetas>                                etiquetas_op           ETIQUETAS_OP
</etiquetas>                               etiquetas_cl           ETIQUETAS_CL
```

```
<etiqueta valor="valor_a_retornar">          etiqueta               ETIQUETA          entre_""




<!-- - - - - - - - - - - - - - -  DECLARACION ATRIBUTOS - - - - - - - - - - - - - - - - - - - -->
<!--    LEXEMA                          DECLARACION LEXICA            SYMBOL -->
<atributo nombre="TEXTO">                 atr_typle_txt            ATR_TYPE_TXT
<atributo nombre="COLOR">                 atr_typle_clr            ATR_TYPE_CLR
<atributo nombre="ALINEACION">            atr_typle_aln            ATR_TYPE_ALN

<atributo nombre="ORIGEN">                atr_typle_ori            ATR_TYPE_ORI
<atributo nombre="ALTURA">                atr_typle_alt            ATR_TYPE_ALT
<atributo nombre="ANCHO">                 atr_typle_anc            ATR_TYPE_ANC

<atributos>                               atributos_op             ATRIBUTOS_OP
</atributos>                              atributos_cl             ATRIBUTOS_CL




<!-- - - - - - - - - - - - - - - - - - - - VALORES  - - - - - - - - - - - - - - - - - - - - - - -->
<!--    LEXEMA                          DECLARACION LEXICA            SYMBOL          VALOR-->
<!-- Atributos -->
[20301030]                                atr_val_intg             ATR_VAL_INTG       20301030
["CENTRAR"]                               atr_val_cent             ATR_VAL_CENT       "CENTRAR"
["IZQUIERDA"]                             atr_val_izqd             ATR_VAL_IZQD      "IZQUIERDA"
["DERECHA"]                               atr_val_dere             ATR_VAL_DERE       "DERECHA"
["JUSTIFICAR"]                            atr_val_just             ATR_VAL_JUST      "JUSTIFICAR"
[#FA8072]                                 atr_val_colh             ATR_VAL_COLH        #FA8072
[Te.XTO,: .]                              atr_val_text             ATR_VAL_TEXT       Te.XTO,: .

<!-- Clases -->
["TITULO"]                                class_val_tit          CLASS_TYPE_TIT
["PARRAFO"]                               class_val_par          CLASS_TYPE_PAR
["IMAGEN"]                                class_val_img          CLASS_TYPE_IMG
["VIDEO"]                                 class_val_vid          CLASS_TYPE_VID
["MENU"]                                  class_val_men          CLASS_TYPE_MEN

<!-- Parametros -->
[Esto: es un titulo.]                      prm_val_tit             PRM_VAL_TIT    Esto: es un titulo.
[1925-23-09]                               prm_val_fch             PRM_VAL_FCH        1925-23-09
[$IdEn-tiF_c]                              prm_val_id              PRM_VAL_ID         $IdEn-tiF_c
```

Definición gramatical

```
inicio:  instruccion

instruccion ::=  acciones | accion
accion      ::=  accion_add_wbst | accion_del_wbst | accion_add_pagw | accion_del_pagw | accion_mod_pagw |
accion_add_comp | accion_del_comp | accion_mod_comp
accion_rec  ::=  accion accion_rec | accion
acciones    ::=  ACCIONES_OP accion_rec ACCIONES_CL

// Parametros
prm_val_id  ::= PARAM_TYPE_ID    PRM_VAL_ID
prm_val_pad ::= PARAM_TYPE_PAD   PRM_VAL_ID
prm_val_uc  ::= PARAM_TYPE_USC   PRM_VAL_ID
prm_val_usm ::= PARAM_TYPE_USM   PRM_VAL_ID
prm_val_sit ::= PARAM_TYPE_SIT   PRM_VAL_ID
prm_val_fch ::= PARAM_TYPE_FCR   PRM_VAL_FCH
prm_val_fmd ::= PARAM_TYPE_FMD   PRM_VAL_FCH

prm_val_tit ::= PARAM_TYPE_TIT   PRM_VAL_TIT PARAMETRO_CL
prm_val_pag ::= PARAM_TYPE_PAG PRM_VAL_ID PARAMETRO_CL

//agregar componentes
class_val_tit ::= CLASS_TYPE_TIT;
class_val_par ::= CLASS_TYPE_PAR;
class_val_img ::= CLASS_TYPE_IMG;
class_val_vid ::= CLASS_TYPE_VID;
class_val_men ::= CLASS_TYPE_MEN;
class_type    ::= class_val_tit | class_val_par | class_val_img | class_val_vid | class_val_men
prm_val_cls   ::= PARAM_TYPE_CLS  class_type  PARAMETRO_CL

//Etiquetas
```

```
etiqueta    ::= ETIQUETA
etiqueta_op ::= etiqueta etiqueta_op | etiqueta
etiquetas   ::= ETIQUETAS_OP etiqueta_op ETIQUETAS_CL

//Atributos
atr_val_op_txt ::= PRM_VAL_TIT | ATR_VAL_TEXT
atr_val_txt ::= ATR_TYPE_TXT atr_val_op_txt ATRIBUTO_CL

aln_cent    ::= ATR_VAL_CENT
aln_izqd    ::= ATR_VAL_IZQD
aln_dere    ::= ATR_VAL_DERE
aln_just    ::= ATR_VAL_JUST
alineacion  ::= aln_cent  | aln_izqd | aln_dere | aln_just
atr_val_aln ::= ATR_TYPE_ALN alineacion ATRIBUTO_CL

atr_val_clr ::= ATR_TYPE_CLR ATR_VAL_COLH ATRIBUTO_CL
atr_val_ori ::= ATR_TYPE_ORI ATR_VAL_TEXT ATRIBUTO_CL |   ATR_TYPE_ORI PRM_VAL_TIT  ATRIBUTO_CL
atr_val_alt ::= ATR_TYPE_ALT ATR_VAL_INTG ATRIBUTO_CL
atr_val_anc ::= ATR_TYPE_ANC ATR_VAL_INTG ATRIBUTO_CL

// FALTA MENU

atributo    ::= atr_val_txt
atributo_op ::= atributo atributo_op | atributo;
atributos   ::= ATRIBUTOS_OP atributo_op ATRIBUTOS_CL;

// Acciones
accion_add_wbst ::=  ACC_ADD_WBST PARAMETROS_OP prm_val_id prm_val_uc prm_val_fch prm_val_fmd prm_val_usm
PARAMETROS_CL ACCION_CL
accion_del_wbst ::=  ACC_DEL_WBST prm_val_id ACCION_CL

prm_val_pad_op  ::=  prm_val_pad | /* epsilon */
accion_add_pagw  ::=    ACC_ADD_PAGW  PARAMETROS_OP        prm_val_id         prm_val_tit      prm_val_sit
prm_val_pad_op      prm_val_uc      prm_val_fch    prm_val_fmd     prm_val_usm     PARAMETROS_CL etiquetas
ACCIONES_CL
accion_del_pagw ::=  ACC_DEL_PAGW prm_val_id ACCIONES_CL

accion_mod_pagw ::=  ACC_MOD_PAGW  PARAMETROS_OP prm_val_id prm_val_tit PARAMETROS_CL etiquetas ACCION_CL

accion_add_comp  ::=   ACC_ADD_COMP   PARAMETROS_OP  prm_val_id   prm_val_pag   prm_val_cls   PARAMETROS_CL
atributos ACCION_CL

accion_del_comp ::=  ACC_DEL_COMP PARAMETROS_OP prm_val_id prm_val_pag  PARAMETROS_CL  ACCION_CL

accion_mod_comp ::=  ACC_MOD_COMP PARAMETROS_OP prm_val_id prm_val_pag prm_val_cls PARAMETROS_CL atributos
ACCION_CL
```

-

- **ANALIZADOR DEL LENGUAJE DE ESTADÍSTICAS**

Definición de los tokens (expresiones regulares)

```
// Uso general
consultar   =  [cC][oO][nN][sS][uU][lL][tT][aA][rR]

vsts_sitio  = [Vv][Ii][Ss][Ii][Tt][Aa][Ss][_][Ss][Ii][Tt][Ii][Oo]
vsts_pagina = [vV][iI][sS][iI][tT][aA][sS][_][pP][aA][gG][iI][nN][aA]
vsts_pgpops = [pP][aA][gG][iI][nN][aA][sS][_][pP][oO][pP][uU][lL][aA][rR][eE][sS]
componente  = [cC][oO][mM][pP][oO][nN][eE][nN][tT][eE]

// Componentes
tit = [tT][iI][tT][uU][lL][oO]
par = [pP][aA][rR][rR][aA][fF][oO]
img = [iI][mM][aA][gG][eE][nN]
vid = [vV][iI][dD][eE][oO]
men = [mM][eE][nN][uU]
tds = [tT][oO][dD][oO][sS]

comp_op = {tit} | {par} | {img} | {vid} | {men} | {tds}

//Simbolos generales
comillas =[\"] | [\“] | [\”]
coma =\,
eoi =\;
val_op =\<
val_cl =\>

inicio = "C:"
id = [a-zA-Z0-9$_.-]+
separador = "/" | "\\"
path = {inicio} {separador} ({id} {separador})+  {id} | {separador} ({id} {separador})+  {id}
```

Definición gramatical

```
inicia con consulta

consulta ::= CONSULTAR op FININSTR | /* nada */


cadena_id ::= COMILLAS   ID   COMILLAS   COMMA cadena_id | COMILLAS  ID   COMILLAS;
op        ::= visitas_sitio  | visitas_pagina  | paginas_populares | componentes;

visitas_sitio     ::= VISITAS_SITIO cadena_id
visitas_pagina    ::= VISITAS_PAGINA cadena_id
paginas_populares ::= PAGINAS_POPULARES   COMILLAS   ID    COMILLAS

dir         ::= PATH | ID SEPARADOR dir | ID
ruta        ::= PATH | ID
componentes ::= COMPONENTE  COMPONENTE_TYPE     COMILLAS ruta COMILLAS
```

3. MANEJO DE CONTENIDO - Clase WebManager.java

Esta clase es la encargada de gestionar todas las órdenes provenientes de la aplicación cliente que han pasado por el parser de manera satisfactoria. Para ello se vale de cuatro tipos de métodos, lo cuales se dividen en:

- Métodos de gestión de sitios web
- Métodos de gestión de páginas web
- Métodos de gestión de componentes web
- Métodos auxiliares internos.

Métodos para gestionar sitios web:

- **createWebsite(String websiteName):**
  - Crea una carpeta para el sitio web y un árbol que lo represente.
  - Agrega el nuevo sitio web a la lista de sitios web.
  - Devuelve una cadena con la respuesta de la operación.
- **deleteWebsite(String websiteName):**
  - Elimina la carpeta del sitio web y el árbol que lo representa de la lista de sitios web.
  - Devuelve una cadena con la respuesta de la operación.

Métodos para gestionar páginas web:

- **addPage(WebPage toAdd):**
  - Agrega una página a su respectivo sitio web.
  - Busca el árbol donde insertar la página.
  - Genera la página desde el nodo.
  - Devuelve una cadena con la respuesta de la operación.
- **deletePage(String idPag):**
  - Elimina una página en base a su ID.
  - Busca el nodo/página a eliminar.
  - Elimina sus hijos de manera recursiva.
  - Elimina la página.
  - Devuelve una cadena con la respuesta de la operación.
- **modifyPage(String idPagToModify, String title, LinkedList<WebComponent> tagslist):**
  - Modifica el título y las etiquetas de una página web.
  - Obtiene la página.
  - Elimina el archivo HTML actual.
  - Reemplaza las etiquetas.
  - Reemplaza el título.
  - Genera un nuevo archivo HTML.
  - Devuelve una cadena con la respuesta de la operación.

Métodos para gestionar componentes web:

- **addComponent(String idPagWhereAdd, WebComponent componentToAdd):**
    - Agrega un componente a una página en base a su ID.
    - Busca la página web.
    - Agrega el componente a la página.
    - Devuelve una cadena con la respuesta de la operación.
- **deleteComponent(String idPagWhereDelete, String idComponenrtToDelete):**
    - Elimina un componente de una página en base a los ID de la página y del componente.
    - Busca la página web.
    - Elimina el componente de la página.
    - Devuelve una cadena con la respuesta de la operación.
- **modifyComponent(String idPagWhereModify, WebComponent componentToAdd):**
    - Reemplaza un componente de una página web por otro.
    - Busca la página web.
    - Reemplaza el componente en la página.
    - Devuelve una cadena con la respuesta de la operación.

Método auxiliar:

- **getPageById(String id):**
    - Retorna un nodo según el ID de la página que se busca.
    - Recorre la lista de páginas.
    - Si encuentra la página, la devuelve.
    - Si no la encuentra, devuelve nulo.

Nota:

- Todos los métodos devuelven una cadena con la respuesta de la operación. Esta respuesta es usada por el parser para ser agregada a un string llamado Response, el cual contiene la respuesta del analizador con información útil para el usuario. Esta información se representa en por consola además de ser desplegada por la aplicación cliente en el apartado de respuesta del servidor.
- Los métodos para gestionar páginas web y componentes web utilizan el método auxiliar getPageById para obtener la página correspondiente.

4. ÁRBOL DE CONTENIDO - Clases Tree y Node

En programación, los árboles son estructuras de datos jerárquicas que se utilizan para representar relaciones entre elementos. En este caso, se presenta un árbol para modelar un sitio web, donde cada nodo representa una página web y las relaciones entre nodos representan la jerarquía del sitio.

1. **Node:** Esta clase representa un nodo individual en el árbol. Cada nodo contiene:
    - **id:** Identificador único para la página web.
    - **page:** Objeto WebPage que contiene información sobre la página web.
    - **childNodes:** Lista enlazada de nodos hijos.

    Los métodos de la clase Node permiten:

    - Obtener el identificador, la página web y la lista de nodos hijos.
    - Actualizar la página web asociada al nodo.
    - Agregar un nodo hijo.
    - Insertar un nodo hijo en la posición correcta dentro de la jerarquía del árbol.
    - Eliminar la página web y sus nodos hijos.

2. **Tree:** Esta clase representa el árbol completo del sitio web. Contiene:
    - **websiteName:** Nombre del sitio web.
    - **websitePath:** Ruta donde se almacena el sitio web.
    - **root:** Nodo raíz del árbol.

    Los métodos de la clase Tree permiten:

    - Obtener el nombre del sitio web.
    - Insertar una página web en el árbol, buscando la página padre y ubicándola en la posición correcta.

5. CLASES REPRESENTANTES DE ELEMENTOS WEB

Para poder manejar objetos web en el servidor opté por crear clases que me permitan representar elementos como páginas web, títulos html, imágenes, videos, etc. los cuales viven dentro de listas y nodos de árboles de acuerdo a los requerimientos de la aplicación. Estos objetos en java permiten una fácil manipulación sin necesidad de recurrir a un archivo html o xml cada vez que sea necesario realizar algún cambio.

Estos objetos están representados por las siguientes clases:

WebPage

proporciona una forma de administrar páginas web, incluido su contenido, estructura y generación de los archivos HTML correspondientes. Se basa en clases WebComponent separadas para representar los componentes básicos del contenido de la página web.

Atributos:

- id: Identificador único para la página web.
- title: Título de la página web que se muestra en la pestaña del navegador.
- site: Nombre del sitio web al que pertenece la página web.
- parentPage: ID de la página web padre si esta página web está anidada dentro de otra.
- creatorUser: Nombre de usuario de la persona que creó la página web.
- creationDate: Fecha y hora en que se creó la página web.
- modificationDate: Fecha y hora en que se modificó por última vez la página web.
- modifierUser: Nombre de usuario de la persona que modificó por última vez la página web.
- websitePath: Ruta en el servidor donde se almacenan los archivos del sitio web.
- tagsList: Cadena que contiene una lista separada por comas de las etiquetas de la página web. (Podría implementarse como una LinkedList separada para una mejor gestión).
- components: LinkedList que contiene instancias de la clase WebComponent, representando los componentes básicos del contenido de la página web (como títulos, párrafos, imágenes, etc.).
- tags: LinkedList que contiene instancias de la clase WebComponent específicamente para las etiquetas de la página web (la implementación parece estar comentada).

Métodos:

- **Constructores:**
    - WebPage(): Constructor vacío para crear un objeto de página web en blanco.
    - WebPage(String id, String title, ...): Constructor que toma varios parámetros para inicializar una página web con detalles.
- **Métodos Modificadores:**

    - setWebsitePath(String websitePath): Establece la ruta donde se generará el archivo HTML de la página web.
- **Métodos de Acceso (Getters):**
    - getId(), getParentPage(), getSite(): Métodos para acceder a la información de la página web.
- **Administración de Componentes:**
    - addHtmlComponent(WebComponent component): Agrega un nuevo componente web a la lista de contenido de la página web y activa la regeneración HTML.
    - removeHtmlComponent(String id): Elimina un componente web basado en su ID de la lista de contenido de la página web y activa la regeneración HTML.
    - replaceHtmlComponent(WebComponent newComponent): Reemplaza un componente web existente por uno nuevo o lo agrega si no se encuentra, y activa la regeneración HTML.
- **Administración de Etiquetas:**
    - replaceTags(LinkedList<WebComponent> tags): Reemplaza las etiquetas de la página web con una nueva lista y actualiza la cadena tagsList.
    - addTag(String tag): Agrega una nueva etiqueta a la cadena tagsList.
- **Administración de Título:**
    - replaceTitle(String title): Reemplaza el título de la página web y actualiza el atributo correspondiente.
- **Generación de HTML:**
    - getHtmlComponents(): Itera a través de los componentes de la página web y recupera su código HTML.
    - getHtmlTags(): Genera el código HTML para las metaetiquetas de la página web basado en la lista tagsList.
    - generateHtmlFile(): Crea el código HTML completo para la página web combinando las secciones de encabezado, componentes, etiquetas y pie de página. Luego, escribe este código en un archivo en el servidor basado en el websitePath y el ID de la página web.
- **Eliminación:**
    - deleteHtmlFile(): Elimina el archivo HTML asociado de la página web del servidor.

Componentes web

Son clases que representan un componente html de una página web.

Clase WebComponent:

- Esta es una clase genérica que representa un componente básico del contenido de una página web.
- Puede representar elementos como títulos, párrafos, imágenes, videos, etc. (tipos específicos se implementan en clases separadas).
- Tiene dos atributos:
    - htmlContent: Cadena que contiene el código HTML para el componente.
    - id: Cadena que contiene un identificador único para el componente.
- Tiene dos métodos para obtener información:
    - getHtml(): Devuelve el código HTML del componente.
    - getId(): Devuelve el ID del componente.

Clase VideoWebComponent:

- Esta clase hereda de WebComponent y representa específicamente un elemento de video HTML.
- Tiene un constructor que toma parámetros para el ID del video, la URL de origen, el ancho y el alto.
- Construye el código HTML para el elemento de video con esos parámetros y lo almacena en el atributo htmlContent.

Clase TituloWebComponent:

- Esta clase hereda de WebComponent y representa específicamente un elemento de título HTML h1.
- Tiene un constructor que toma parámetros para el ID del título, el contenido del texto, la alineación y el color.
- Construye el código HTML para el elemento h1 con esos parámetros y lo almacena en el atributo htmlContent.
    - Utiliza una instrucción switch para convertir las opciones de alineación ("CENTRAR", "IZQUIERDA", "DERECHA") en los estilos CSS correspondientes ("center", "left", "right").

Clase ParrafoWebComponent:

- Esta clase hereda de WebComponent y representa específicamente un elemento de párrafo HTML.
- Tiene un constructor que toma parámetros para el ID del párrafo, el contenido del texto, la alineación y el color.
- Construye el código HTML para el elemento de párrafo con esos parámetros y lo almacena en el atributo htmlContent.

    - Utiliza una instrucción switch para convertir las opciones de alineación ("CENTRAR", "IZQUIERDA", "DERECHA") en los estilos CSS correspondientes ("center", "left", "right").

Clase ImagenWebComponent:

- Esta clase hereda de WebComponent y representa específicamente un elemento de imagen HTML.
- Tiene un constructor que toma parámetros para el ID de la imagen, la URL de origen, la alineación, el alto y el ancho.
- Construye el código HTML para el elemento de imagen con esos parámetros y lo almacena en el atributo htmlContent.
    - Utiliza una instrucción switch para convertir las opciones de alineación ("CENTRAR", "IZQUIERDA", "DERECHA") en los estilos CSS correspondientes ("center", "left", "right").

Clase EtiquetaWebComponent:

- Esta clase hereda de WebComponent y representa una etiqueta HTML genérica.
- Tiene un constructor que toma una cadena que representa toda la etiqueta HTML como entrada.
- Asigna la misma cadena a los atributos htmlContent e id.