

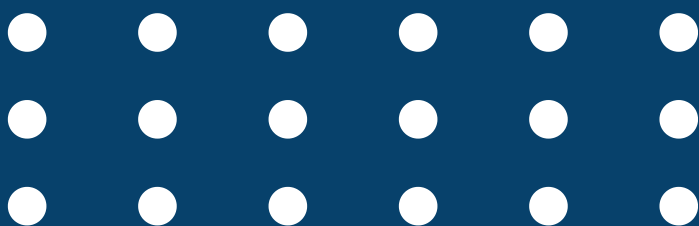
Universidad De San Carlos de Guatemala
Centro Universitario de Occidente CUNOC



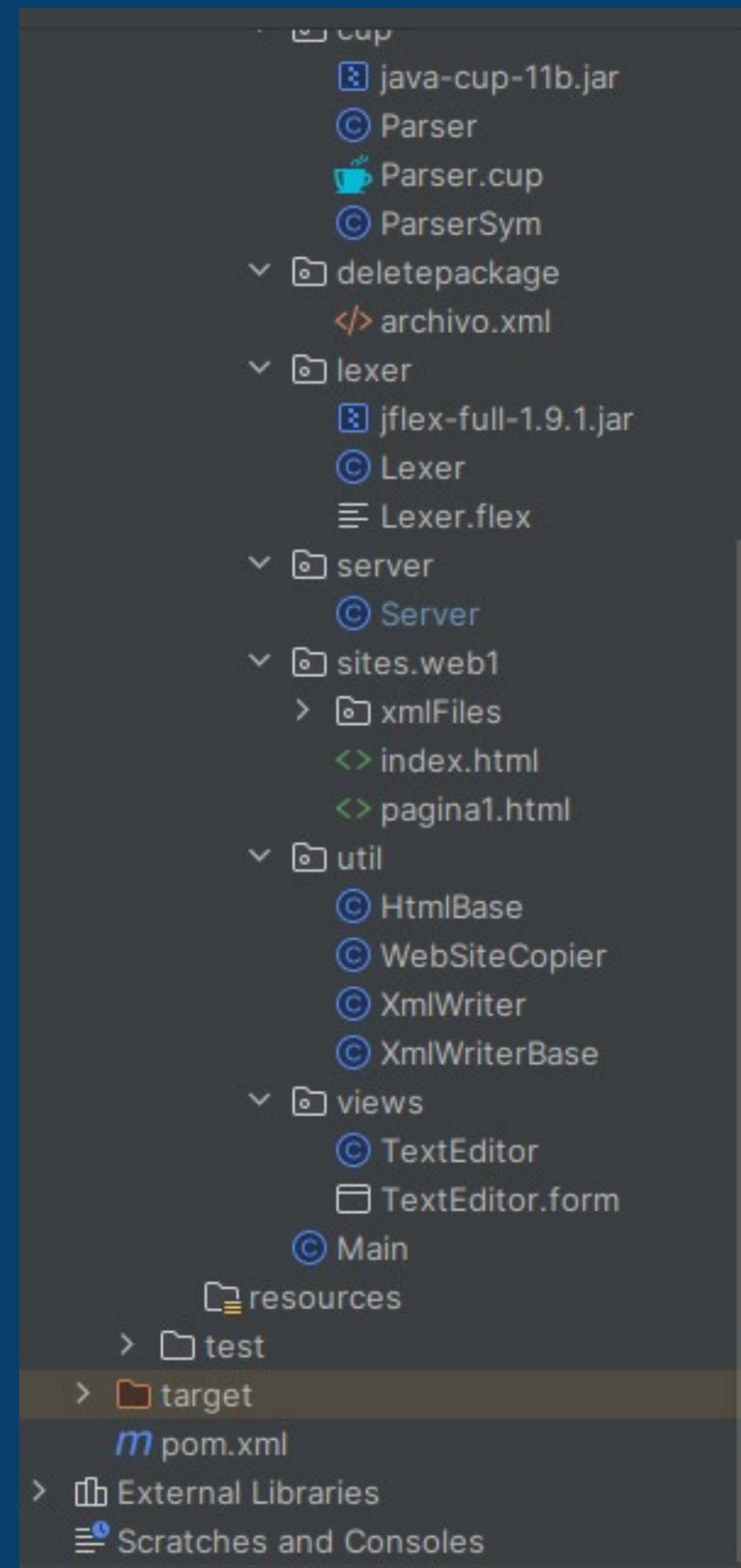
MANUAL TÉCNICO

202031773 Enmer Oswaldo Sandoval Mazariegos

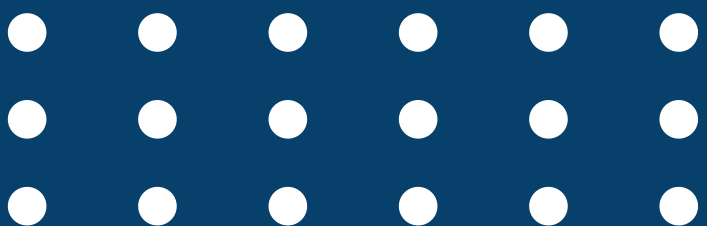
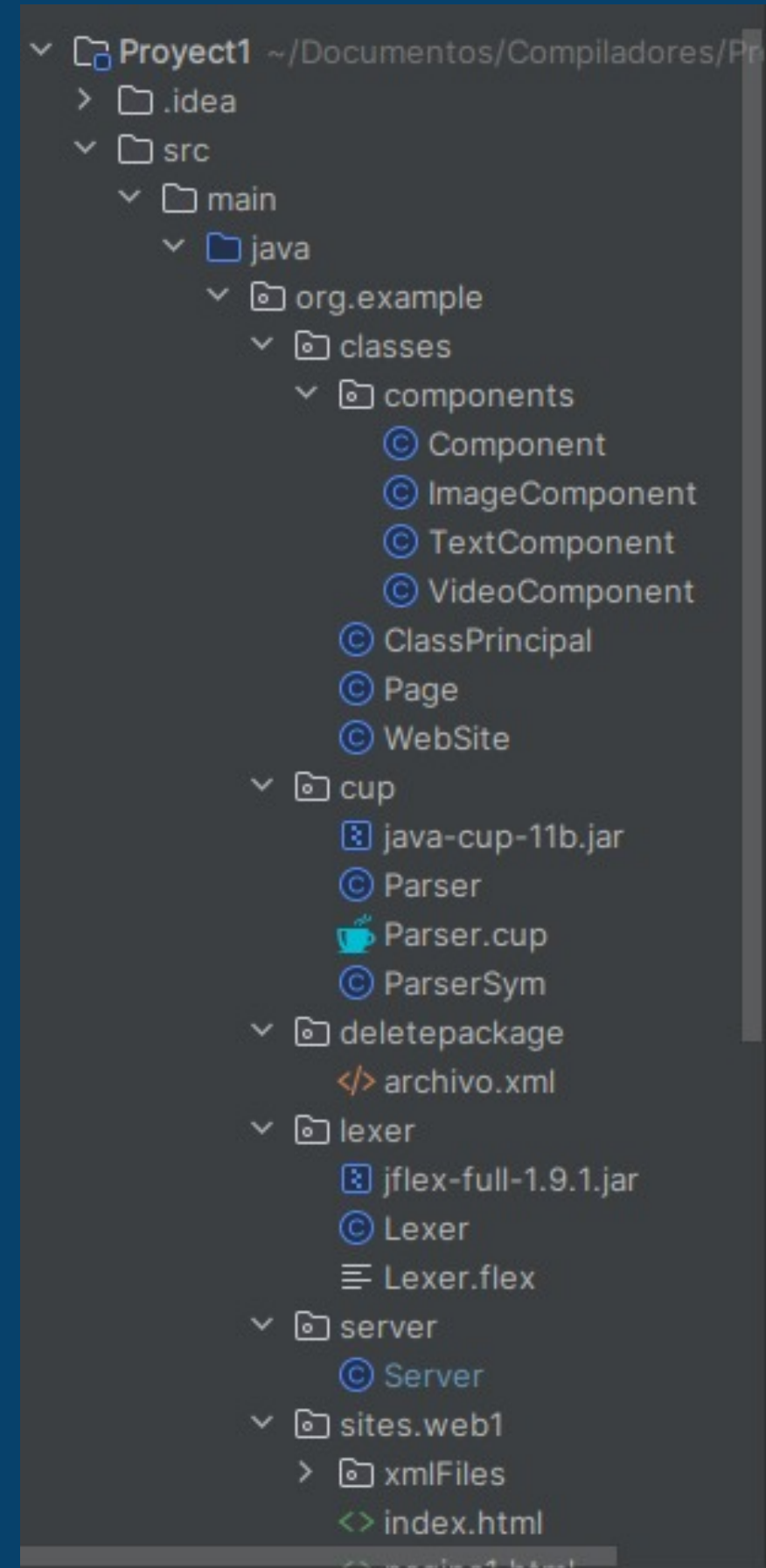
Este manual proporciona una guía detallada sobre el uso, la implementación y el funcionamiento del Compilador de XML a HTML. El objetivo principal del compilador es facilitar la conversión precisa de documentos XML a HTML estándar, permitiendo a los desarrolladores generar contenido web dinámico a partir de datos XML. Las características clave incluyen una conversión precisa, cumplimiento de estándares, flexibilidad y optimización de rendimiento. El manual cubre la instalación, configuración, sintaxis, ejemplos de uso, mejores prácticas y solución de problemas comunes. Está dirigido a desarrolladores con conocimientos básicos de XML y HTML, así como experiencia en desarrollo web.



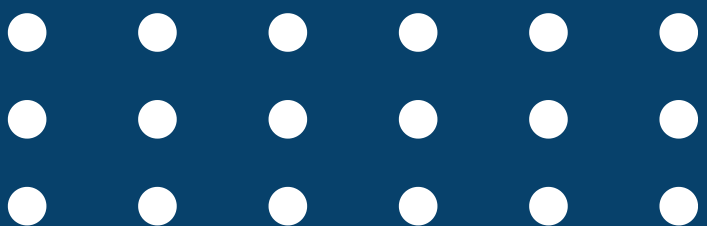
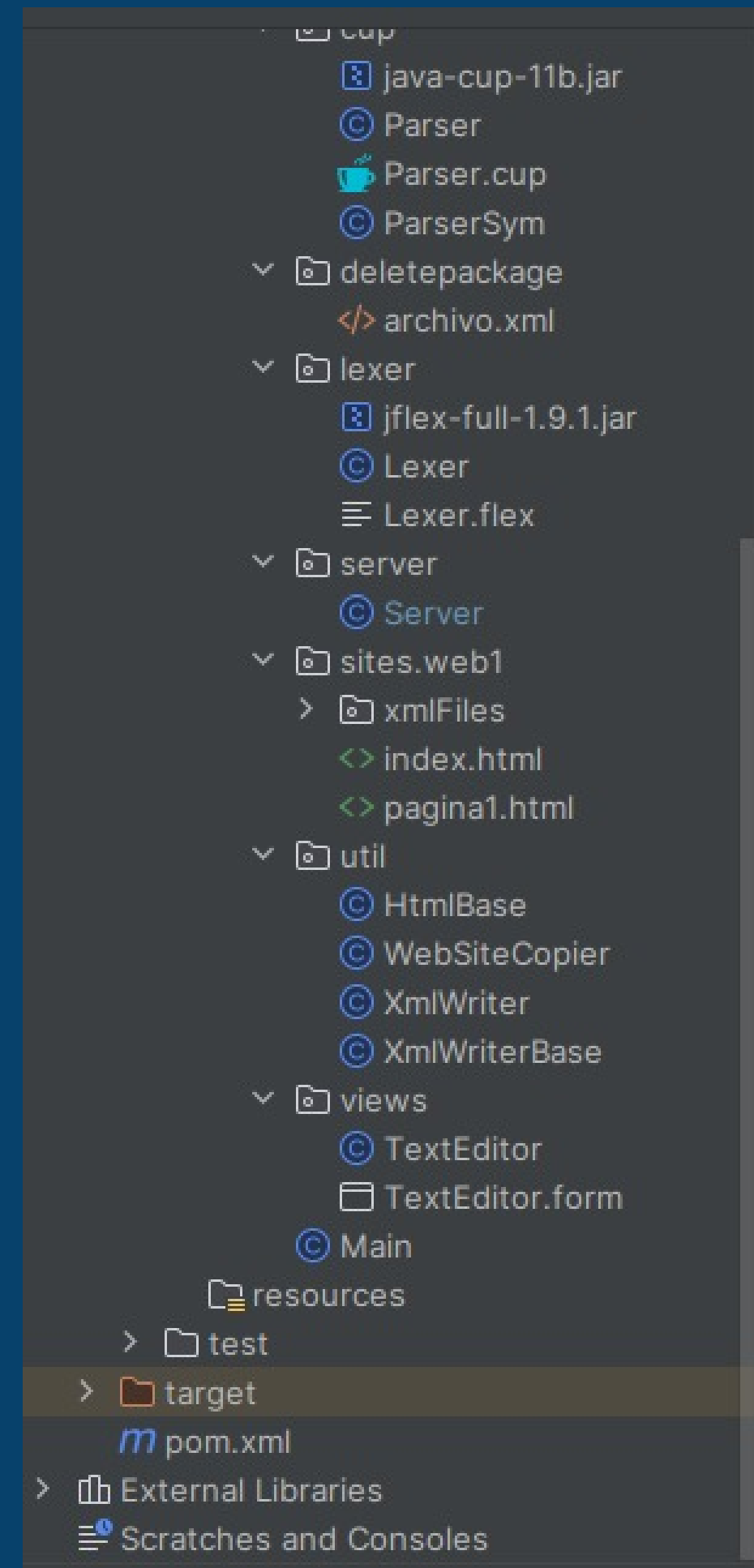
Distribución del proyecto.



Distribución del proyecto.




Distribución del proyecto.



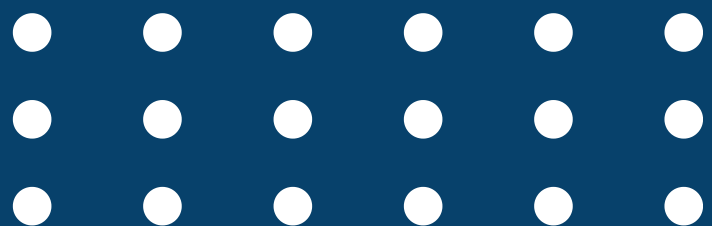
Clase main

```
import org.example.server.Server;

To Run code, press F6 or click the  icon in the gutter.

Enmer
public class Main {
    Enmer
    public static void main(String[] args) {
        Server server = new Server();
        server.threadApp();
    }
}
```

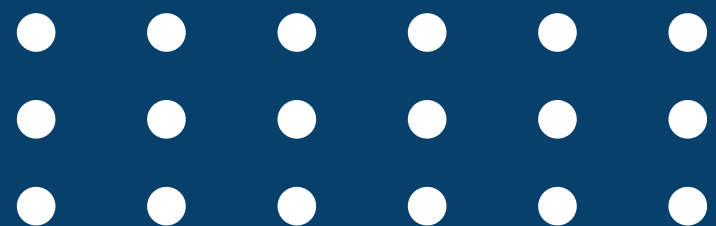
Desde aca inicializamos el
servidor



Descripción

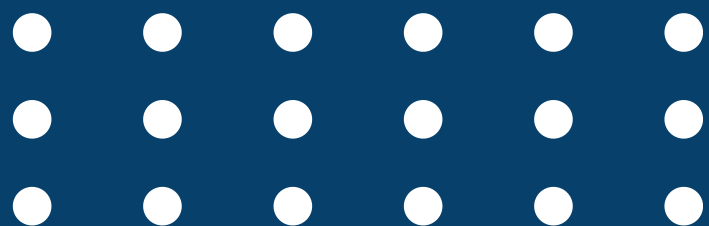
Este código implementa un servidor web básico que escucha en el puerto 8080 y ejecuta acciones según los mensajes recibidos. Aquí está el resumen más conciso:

- **Clase Server:** Implementa un servidor socket en el puerto 8080 y acepta conexiones entrantes. Tiene un método runParser que analiza los mensajes recibidos y ejecuta acciones correspondientes.
- **Método runParser(String text):** Analiza el texto recibido, extrae información sobre sitios web, páginas y componentes, y ejecuta acciones según la información obtenida.
- **Métodos createSites, verificationComponents, readXmlExist, modifyComponente, modifyTitle:** Realizan acciones específicas como crear, modificar o eliminar sitios web, páginas y componentes según las instrucciones recibidas.



Este codigo fue implementado bajo una gramatica muy escriptaa para poder crear paginas web en base a codigo xml y se pueden generar paginas web dinamicas sin necesidad de saber programar es muy eficiente para el usuario.

por lo tanto usamos 6 cadenas de produccion que se desarrollan en mas cadenas de subproduccion usando simbolos terminales (palabras reservadas) como tambien los non terminales que son las cadenas de produccion que utilizamos para interpretar el codigo html que el usuario nos proporciona



Simbolos terminales y no terminales

```
private String quitarEspacios(String space){
    return space.replace(" ", "");
}

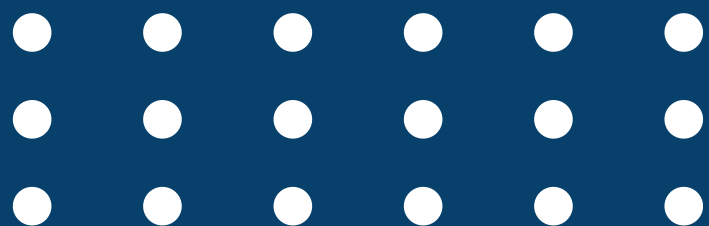
:}

terminal NUEVO_SITIO_WEB, ID, USUARIO_CREACION, FECHA_CREACION, FECHA_MODIFICACION, USUARIO_MODIFICACION,
BORRAR_SITIO_WEB, NUEVA_PAGINA, TITULO, TITULOCORCHETES, SITIO, PADRE, MODIFICAR_PAGINA, BORRAR_PAGINA,
PAGINA, CLASE, AGREGAR_COMPONENTE, BORRAR_COMPONENTE, NOMBRE, IGUAL, MAYORQUE, COMILLAS,
PARRAFO, IMAGEN, VIDEO, MENU, TEXTO, ALINEACION, COLOR, ORIGEN, ALTURA, ANCHO, ETIQUETAS, INITACCION,
INITACCION, ENDACCION, INITPARAMETRO, ENDPARAMETRO, INITATRIBUT, ENDATRIBUT, INITPARAMETROS, ENDPARAMETROS,
INITATRIBUTOS, ENDATRIBUTOS, IDENTIFICADOR;

//INITETIQUETAS ENDETIQUETAS INITETIQUETA ENDETIQUETA VALOR

//Strings
non terminal String principalAttribut, wideAttribut, heightAttribut, originAttribut,
colorAttribut, lineUpAttribut, textAttribut, pageComponent, principal, site, title, modificationUser,
modificationDate, creationDate, userCreator, id, prePrincipal, classComponent, atributClass;

//Components
non terminal VideoComponent declaredVideoAttribut;
non terminal ImageComponent declaredImageAttribut;
non terminal TextComponent declaredTextAttribut;
```



Utilizando StrinBuilder generamos el
codigo html y xml que necesitamos
para su funcionamiento

```
        return false;
    }

1 usage  👤 Enmer
public void createIndex(String path){
    StringBuilder htmlBuilder = new StringBuilder();
    htmlBuilder.append("<html>\n");
    htmlBuilder.append("<head>\n");
    htmlBuilder.append("</head>\n");
    htmlBuilder.append("<body>\n");
    htmlBuilder.append("</body>\n");
    htmlBuilder.append("</html>\n");
    writeIndex(htmlBuilder, path);
}

1 usage  👤 Enmer
public void writeIndex(StringBuilder htmlBuilder, String path){
    try {
        BufferedWriter writer = new BufferedWriter(new FileWriter( fileName: path + "/index.html"
        writer.write(htmlBuilder.toString());
        writer.close();
    }catch (Exception e){
        e.printStackTrace();
    }
}

}
```

2 usages 👤 Enmer

Texto xml valido siendo este una demostracion

```
<accion nombre="NUEVA_PAGINA">
  <parametros>
    <parametro nombre="ID">
      [hola]
    </parametro>
    <parametro nombre="TITULO">
      [El titulo de la]
    </parametro>
    <parametro nombre="SITIO">
      [El identificador del sitio web ]
    </parametro>
    <parametro nombre="PADRE">
      [identificador de la pagina padre]
    </parametro>
    <parametro nombre="USUARIO_CREACION">
      [ID del usuario que ejecuto la accion]
    </parametro>
    <parametro nombre="FECHA_CREACION">
      [4444-02-16]
    </parametro>
    <parametro nombre="USUARIO_MODIFICACION">
      [id del ultimo usuario que realizo modificaciones a pagina]
    </parametro>
    <parametro nombre="FECHA_MODIFICACION">
      [1444-54-66]
    </parametro>
  </parametros>
</accion>
```

h > parametros > parametro

**ESPEREMOS Y SEA DE TU UTILIDAD AGRADECIENDO EL TIEMPO Y ESPACIO PARA LEER
ESTE MANUAL DE ENTREGA**

ENMER SANDOVAL

202031773

INGENIERIA EN CIENCIAS Y SISTEMAS CUNOC

