

Objetivos generales

- Aplicar conocimientos de análisis léxico y sintáctico, así como el manejo de errores para crear herramientas útiles.

Objetivos específicos

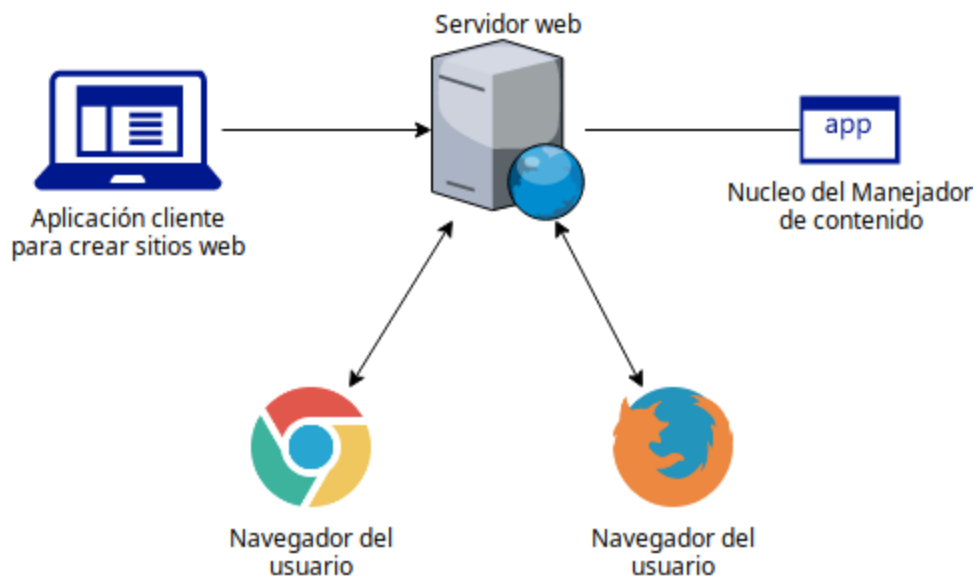
- Crear expresiones regulares y gramaticales libres de contexto complejas.
- Implementar las fases de análisis léxico y sintáctico de un compilador.
- Combinar la funcionalidad de JFlex y Cup en aplicaciones reales.
- Entender una arquitectura Cliente-servidor.
- Adquirir conocimientos sobre el lenguaje XML

Descripción de la actividad

En la actualidad, los sitios web crecen en número de páginas web para brindar a los usuarios contenido como lo pueden ser noticias, videos, etc. Un ejemplo de ello son los sitios web de noticieros en donde cada noticia está contenida en una página web. Lo anterior supone un problema para el proceso común de creación de páginas web, donde una persona con conocimientos de programación debe escribir código html y ubicar la nueva página web en el servidor Http.

Para solucionar el problema anterior se desarrolló la idea de los manejadores de contenido, que son sistemas informáticos que automatizan la creación de sitios web y su contenido utilizando componentes que se agregan a las páginas web de manera gráfica permitiendo que personas no programadores (editores) puedan crear sitios web rápidamente.

Se le solicita la creación de un manejador de contenidos liviano que permita la comunicación entre una aplicación cliente y una aplicación en el servidor para despachar y construir sitios web.



Funciones del servidor

Las páginas web creadas deben ser servidas por medio de un servidor web HTTP (Apache Http o similar) a través del puerto 80.

Los usuarios finales de la aplicación podrán conectarse al servidor por medio de navegadores web y así navegar a través del sitio web deseado.

En el servidor también se ejecuta la aplicación núcleo del manejador de contenidos para escuchar los mensajes que envíe la aplicación cliente.

Los mensajes de la aplicación cliente incluyen una o más acciones estructuradas en código XML de la siguiente forma:

Una sola acción:

```
<accion nombre="NOMBRE_DE_LA_ACCION">
  <parametros>
    <parametro nombre="NOMBRE_PARAMETRO">
      [VALOR DEL PARAMETRO]
    </parametro>
  </parametros>
</accion>
```

Más de una acción:

```
<acciones>
  <accion nombre="...">
    .....
  </accion>
  <accion nombre="...">
    .....
  </accion>
  .....
</acciones>
```

El valor dentro de cada parámetro debe estar entre corchetes []

Aplicación cliente

La aplicación cliente debe ser una aplicación de escritorio usada por los editores para ingresar al sistema manejador de contenido y ejecutar acciones y consultas de estadísticas.

La aplicación cliente debe estar compuesta por un área de texto donde el usuario escribe el código XML de la o las acciones (las acciones se describen más adelante). También se debe contar con un área de respuesta donde se visualizan:

- Errores léxicos o sintácticos que se generan al ejecutar las acciones XML. Se debe incluir una descripción completa del error (línea, columna, token o símbolo de error)
- Mensajes de respuesta del servidor: el servidor responde al cliente por medio de mensajes XML que deben ser entendidos por la aplicación cliente y desplegados en el área de respuesta.

Sitios web

La aplicación debe ser capaz de manejar varios sitios web. El servidor http es el encargado de despachar las páginas web al puerto 80 pero el núcleo del manejador de contenidos debe ser el encargado de almacenar (estilo base de datos) la información de cada sitio web y su estructura, nombre, páginas, etc. Se recomienda usar xml para almacenar dicha información (ver un ejemplo al final).

La idea de los sitios web es que estén diseñados como un árbol donde la página **index** sea el nodo padre de todo el sitio web. Debido a lo anterior cada página web puede contener páginas hijas y hermanas, y dicha estructura se debe almacenar en la estructura propuesta en el párrafo anterior..

Acción para crear sitios web

La aplicación cliente permite crear varios sitios web los cuales deben funcionar sobre el mismo servidor http. Cada sitio web debe tener siempre una página **index**

Valor del atributo nombre: **NUEVO_SITIO_WEB**

Parámetros:

- **ID***: el identificador del sitio web
- **USUARIO_CREACION**: el ID del usuario que ejecutó la acción.
- **FECHA_CREACION**: Fecha en que se crea el sitio web, en formato yyyy-MM-dd
- **FECHA_MODIFICACION**: Fecha de la última modificación
- **USUARIO_MODIFICACION**: Id del ultimo usuario que realizó modificaciones en el sitio.

Si no se incluye explícitamente alguno de los parametros **USUARIO_CREACION**, **FECHA_CREACION**, **FECHA_MODIFICACION**, **USUARIO_MODIFICACION** la aplicación cliente los debe agregar al mensaje con los valores correctos.

Acción para eliminar sitios web

Al eliminarse un sitio web, se deben eliminar también todas sus páginas web.

Valor del atributo nombre: **BORRAR_SITIO_WEB**

Parámetros:

- **ID***: Nombre del sitio web a eliminar.

Páginas web

Los editores son los encargados de la creación de páginas web usando componentes. Los componentes se agregan a cada página y se pueden configurar de manera que el contenido tenga las propiedades deseadas por el editor.

Acción para crear página

Valor del atributo nombre: **NUEVA_PAGINA**

Parámetros:

- **ID***: el identificador de la página
- **TITULO**: El título de la página mostrado en el navegador web, puede contener cualquier.
- **SITIO***: El identificador del sitio web de la página
- **PADRE***: El identificador de la página web padre. Si éste parámetro no se incluye en la acción entonces el núcleo del manejador de contenidos entiende que la página es hermana del index.
- **USUARIO_CREACION**: el ID del usuario que ejecutó la acción.
- **FECHA_CREACION**: Fecha en que se crea la página, en formato yyyy-MM-dd
- **FECHA_MODIFICACION**: Fecha de la última modificación
- **USUARIO_MODIFICACION**: Id del último usuario que realizó modificaciones a la página.

Si no se incluye explícitamente alguno de los parámetros **USUARIO_CREACION**, **FECHA_CREACION**, **FECHA_MODIFICACION**, **USUARIO_MODIFICACION** la aplicación cliente los debe agregar al mensaje con los valores correctos.

Cada página web puede tener etiquetas que no son más que un conjunto palabras clave (mismo formato que los ID) que sirven para relacionar varias páginas web del mismo sitio. Para agregar etiquetas a una página web se debe usar la estructura del ejemplo siguiente:

```
<accion nombre="NUEVA_PAGINA">
  <parametros>
    .....
    <parametro nombre="ID">
      [productos]
    </parametro>
    .....
  </parametros>
  <etiquetas>
    <etiqueta valor="etiqueta1"/>
    <etiqueta valor="etiqueta2"/>
  </etiquetas>
</accion>
```

La estructura **<etiquetas>** solo puede aparecer en las acciones **NUEVA_PAGINA** y **MODIFICAR_PAGINA**



Acción para eliminar páginas

Al eliminarse una página, se deben eliminar también todas sus páginas web hijas.

Valor del atributo nombre: **BORRAR_PAGINA**

Parámetros:

- **ID***: Nombre de la página a eliminar.

Acción para modificar parámetros de páginas

Esta acción permite reemplazar los valores de los parámetros TITULO y ETIQUETAS si son definidos en ésta acción.

Además del parámetro **ID**, es obligatorio que alguno de los parámetros anteriores esté definido.

Valor del atributo nombre: **MODIFICAR_PAGINA**

```
<accion nombre="MODIFICAR_PAGINA">
  <parametros>
    <parametro nombre="ID">
      [producto]
    </parametro>
    <parametro nombre="TITULO">
      [nuevo titulo]
    </parametro>
  </parametros>
  <etiquetas>
    <etiqueta valor="nueva_etiqueta1"/>
    <etiqueta valor="nueva_etiqueta2"/>
  </etiquetas>
</accion>
```

Acción para agregar componentes a páginas web

Esta acción permite agregar un componente a una pagina especifica.

Valor del atributo nombre: **AGREGAR_COMPONENTE**

Parámetros:

- **ID***: Identificador del componente. Este ID debe ser único en el contexto de la página web.
- **PAGINA***: Identificador de la página en donde se va a crear el componente.
- **CLASE***: la clase de componente. Puede ser TITULO, PARRAFO, IMAGEN, VIDEO, MENU

Además de parámetros, los atributos de cada componente se deben definir usando la estructura

<atributos>

Ejemplo:



```
<accion nombre="AGREGAR_COMPONENTE">
  <parametros>
    <parametro nombre="ID">
      [comp-1]
    </parametro>
    <parametro nombre="PAGINA">
      [productos]
    </parametro>
    <parametro nombre="CLASE">
      [TITULO]
    </parametro>
  </parametros>
  <atributos>
    <atributo nombre="TEXTO">
      [Este es el texto que aparece en el titulo :) ]
    </atributo>
    <atributo nombre="COLOR">
      [#5A5A5A]
    </atributo>
  </atributos>
</accion>
```

Acción para eliminar componentes de paginas web

Esta acción permite eliminar un componente a una pagina especifica.

Valor del atributo nombre: **BORRAR_COMPONENTE**

Parámetros:

- **ID***: Identificador del componente.
- **PAGINA***: Identificador de la página en donde se encuentra el componente a eliminar.

Acción para modificar componentes de paginas web

Esta acción prácticamente reemplaza un componente existe en una página por la nueva configuración enviada en esta acción. Sigue la misma estructura de la acción **AGREGAR_COMPONENTE**

Valor del atributo nombre: **MODIFICAR_COMPONENTE**

Parámetros:

- **ID***: Identificador del componente a reemplazar.
- **PAGINA***: Identificador de la pagina en donde se encuentra el componente.
- **CLASE***: la clase de componente. Puede ser TITULO, PARRAFO, IMAGEN, VIDEO, MENU



Además de parámetros, los atributos de cada componente se deben definir usando la estructura

`<atributos>`

Ejemplo:

```
<accion nombre="MODIFICAR_COMPONENTE">
  <parametros>
    <parametro nombre="ID">
      [comp-1]
    </parametro>
    <parametro nombre="PAGINA">
      [productos]
    </parametro>
    <parametro nombre="CLASE">
      [PARRAFO]
    </parametro>
  </parametros>
  <atributos>
    <atributo nombre="TEXTO">
      [Este es el texto que aparece en el parrafo :) ]
    </atributo>
  </atributos>
</accion>
```

Componentes

Las clases de componentes permitidos son las siguientes:

TITULO

Genera la etiqueta `<h1>` en la página web resultante.

Atributos:

- **TEXTO***: contiene el texto del título
- **ALINEACION**: indica la alineación del título, el valor puede ser CENTRAR, IZQUIERDA, DERECHA, JUSTIFICAR
- **COLOR**: indica el valor del texto del párrafo. el valor debe ser en formato hexadecimal. Ej: #5A5A5A

PARRAFO

Genera la etiqueta `<p>` en la página web resultante.

Atributos:

- **TEXTO***: contiene el texto del párrafo
- **ALINEACION**: indica la alineación del título, el valor puede ser CENTRAR, IZQUIERDA, DERECHA, JUSTIFICAR
- **COLOR**: indica el valor del texto del párrafo. el valor debe ser en formato hexadecimal. Ej: #5A5A5A

IMAGEN

Genera la etiqueta en la página web resultante.

Atributos:

- **ORIGEN***: contiene la dirección url de la imagen a mostrar.
- **ALINEACION**: indica la alineación del título, el valor puede ser CENTRAR, IZQUIERDA, DERECHA, JUSTIFICAR
- **ALTURA***: indica el valor de la altura de la imagen. Debe ser un entero
- **ANCHO***: indica el valor del ancho de la imagen. Debe ser un entero

VIDEO

Genera la etiqueta <video> en la página web resultante.

Atributos:

- **ORIGEN***: contiene la dirección url del video a mostrar.
- **ALTURA***: indica el valor de la altura de la imagen. Debe ser un entero
- **ANCHO***: indica el valor del ancho de la imagen. Debe ser un entero

MENU

Genera un listado de links (etiquetas <a>) basado en la configuración deseada.

Atributos:

- **PADRE**: El menú es creado a partir de las páginas hijas del identificador especificado en este atributo.
- **ETIQUETAS**: etiquetas separadas por el carácter barra |. El Menú es creado a partir de todas las páginas que incluyan alguna de las etiquetas definidas en este atributo.

Los atributos de este componente se pueden combinar para filtrar las páginas que se incluyen en el menú. por ejemplo se puede crear un menu de las paginas hijas de 'productos' que contengan la etiqueta 'tenis'

Se debe especificar alguno de los atributos.

Consulta de estadísticas

Los editores pueden consultar estadísticas de cada sitio web, página web y componentes usando un lenguaje de consulta especial llamado SQcms. Este lenguaje no debe ser case sensitive por lo que los comandos pueden estar escritos en mayúsculas, minúsculas o con combinaciones.

Consulta de visitas a sitio web

Este comando permite conocer la cantidad de visitas que un sitio web ha tenido, equivalente a la suma de visitas de cada página del sitio. Se pueden indicar varios sitios separados por coma.

```
CONSULTAR VISITAS_SITIO <id de sitio web>;  
CONSULTAR VISITAS_SITIO <id de sitio web>, <id de sitio web>, <id de sitio web>;
```

ejemplo:

```
Consultar visitas_sitio "misitio";  
Consultar visitas_sitio "misitio","otrositio";
```

Consulta de visitas a página web

Este comando permite conocer la cantidad de visitas que una página web ha tenido. Se pueden indicar varios sitios separados por coma.

```
CONSULTAR VISITAS_PAGINA <path_de_la_pagina>;  
CONSULTAR VISITAS_PAGINA <path_de_la_pagina>,<path_de_la_pagina>;
```

Ejemplos:

```
Consultar visitas_pagina "misitio.mipagina";  
Consultar visitas_pagina "misitio.mipagina", "otrositio.pagina2";
```

Consulta de páginas populares

Este comando permite listar las 10 páginas más populares (con más cantidad de visitas) en un sitio web.

```
CONSULTAR PAGINAS_POPULARES <id de sitio web>;
```

ejemplo:

```
Consultar PAGINAS_POPULARES "misitio";
```

Consulta de uso de componentes

Este comando permite mostrar la cantidad de componentes por tipo usado en alguna pagina en particular.

CONSULTAR COMPONENTE <tipo de componente> <path_de_la_pagina>;

El tipo de componente puede ser el nombre de algun componente (TITULO, MENU, etc) o la palabra TODOS que muestra la cantidad de cada componente.

ejemplo:

Consultar COMPONENTE TODOS "misitio.mipagina";

Tomar en cuenta

- Todos los ID deben ser un conjunto de caracteres alfanuméricos incluyendo e iniciando con `_` , `-` o `$`. **`_id $id -id -titulo4 $parrafo45`**
- No se pueden repetir los ID en cada grupo de entidades diferentes (páginas, sitios web, etc.)
- Los nombres de las acciones, parámetros, atributos son case sensitive, en cambio los nombres de las etiquetas y atributos XML no lo son.
- El orden de cada etiqueta XML hija dentro de una etiqueta padre no importa. sin embargo, no se deben repetir estructuras con el mismo nombre.
- Todos los parámetros y atributos marcados con `*` son obligatorios.

Importante

- Usar lenguaje de programación JAVA
- Usar herramientas Jflex y Cup para cualquier tipo de análisis lexico y sintáctico.
- Práctica obligatoria para tener derecho al siguiente proyecto.
- Las copias obtendrán nota de cero y se notificará a coordinación.
- Si se va a utilizar código de internet, entender la funcionalidad para que se tome como válido.

Entrega

La fecha de entrega es el día viernes 19 de abril a las 09:00. Los componentes a entregar son:

- Código fuente (enlace del repo de github)
- Ejecutables (jar)
- Manual técnico incluyendo una sección con las expresiones regulares y las gramáticas usadas.

- Manual de usuario.

Calificación

Se calificará en la computadora de cada alumno por lo que es necesario que tengan instalado y configurado todo lo necesario.