

Tarea Programada #1

- La tarea debe entregarse al profesor el día y a la hora convenida.
- La tarea debe ser enviada en un tar.gz que contenga todos los archivos, el nombre de este archivo debe ser APELLIDO y NOMBRE, ejemplo HernandezErick.tar.gz
- La tarea debe contener lo siguiente:
 - a. Fuentes, todo el código necesario para ejecutar la tarea. El código debe estar debidamente documentado.
 - b. Documentación
 - i. Descripción técnica (algoritmos, interfaces, estructuras, etc.)
 - ii. Manual de Usuario
- Toda tarea debe ser defendida ante el profesor, de tal manera todos los estudiantes deben poder explicar la solución satisfactoriamente.
- ¡Buena Suerte!

A Evaluar	Puntos	Nota
Documentación	5	
Validación de XML	40	
Validación de HTML	40	
Indentación y Salida	15	
Total	100	

Validación de Documentos

La tarea consiste en validar 2 clases de documentos, páginas HTML y documentos XML, además, de darles formato en caso de que estos estén correctos.

Ambos tipos de documentos permiten definir tags en los cuales se puede incluir información, esta información puede contenido dentro del tag o atributos que describen o dan algún comportamiento al mismo.

Por ejemplo se puede tener el siguiente tag:

```
<a id='name' name='name' href='#'>Text</a>
```

En este caso A es el nombre del tag, tiene tres atributos, id, name y href y el contenido del Tag es "Text".

Un tag puede tener otros tags dentro lo que permite anidarlos en una estructura de árbol, no es necesario generar esta estructura, sino solo validar que los tags estén correctos.

No hay que validar el contenido de un atributo, así por ejemplo si alguno de los atributos contuviera código por ejemplo este se trata como texto, pero si hay que tener en cuenta que el código podría contener comillas y/u otros tags, por lo que se espera que su programa no se pierda.

Tampoco se debe validar el contenido de los tags "style" o "script" que pueden contener otros formatos.

Dada la estructura de árbol pondremos un máximo a la cantidad de tags anidados que el programa puede manejar, siendo este de 1000; además, delimitaremos el tamaño del archivo a 4K.

El programa tiene que poder determinar todos los errores de un documento y escribirlos a la pantalla con su número de línea y número de columna. Por ejemplo, errores como que un tag no tenga cierre, que no se cierren las comillas de un atributo, o que el documento este mal formado en el caso de un html.

Note que el XML es genérico y los tags pueden tener cualquier nombre, sin embargo el html no, hay algunos tags que deben estar presentes y deben tener una estructura específica, por ejemplo, debe haber un tag html, uno body, uno head, el title solo debe estar dentro del head, y así sucesivamente.

También debe de ignorar los comentarios de HTML aún si estos contienen texto, código, tags, etc.

Así se recomienda hacer el validador para el XML primero y luego para el HTML que sería una especificación del primero.

También el programa debe indentar a 3 espacios el documento a salida estándar si este pasa la validación.

El programa debe tomar la entrada de la entrada estándar y dar su resultado en la salida estándar. No es necesario abrir o cerrar archivos.