

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computación

Compiladores e Intérpretes

Profesor: Andrei Fuentes Leiva

Proyecto I Analizador Léxico para XHTML

Estudiantes: Daniel Cortés Sáenz, Isaac Ramírez Solano

I Semestre, 2013

Índice

Descripción del Problema.....	3
Diseño del Programa.....	3
Decisiones de Diseño.....	3
Algoritmos usados.....	3
Lista de Tokens.....	3
Librerías externas utilizadas.....	10
Análisis de Resultados.....	10
Objetivos alcanzados.....	10
Objetivos no alcanzados.....	10
Manual de usuario.....	11
Conclusión Personal.....	14

Descripción del Problema

El análisis léxico (scanner) es el primer paso que se realiza en el proceso de compilación. En este se verifica que todos los caracteres y símbolos ingresados se han escrito correctamente.

Para este curso se desarrollará un compilador para el lenguaje XHTML. La primera parte del proyecto constituye el desarrollo del analizador léxico para este lenguaje. XHTML es una variante de HTML pero un poco más estricto en cuanto a algunas de sus reglas. Estas reglas tuvieron que ser investigadas para determinar los tokens que retornará el scanner y que leerá el analizador sintáctico.

La especificación del analizador léxico tiene que ser desarrollada en Flex. Los errores tienen que ser manejados por el stderr y la entrada y salida de datos tiene que ser por medio del stdin y stdout respectivamente.

Diseño del Programa

Decisiones de Diseño

Para la implementación del analizador léxico se podía utilizar Flex o JFlex. Se decidió usar C porque en los ejemplos de clase se utilizó Flex. Además, se contaba con bastante documentación de Flex para consultar. La página oficial de Flex publicó una guía de cómo utilizarlo, desde lo más básico como lo es la instalación, estructura del archivo y compilación del programa; hasta lo más complejo como el manejo de memoria o la creación de tablas serializables.

Algoritmos usados

Lista de Tokens

Los tokens fueron definidos dentro del archivo Tokens.h

Token	Descripción
ERROR	Token para identificar los errores léxicos. Este se utiliza para los caracteres inválidos.
A	Token para la etiqueta <a>
_A	Token para la etiqueta
ACRONYM	Token para la etiqueta <acronym>
_ACRONYM	Token para la etiqueta </acronym>
ADDRESS	Token para la etiqueta <address>
_ADDRESS	Token para la etiqueta </address>
APPLET	Token para la etiqueta <applet>
_APPLET	Token para la etiqueta </applet>
AREA	Token para la etiqueta <area>

_AREA	Token para la etiqueta </area>
B	Token para la etiqueta
_B	Token para la etiqueta
BASE	Token para la etiqueta <base>
_BASE	Token para la etiqueta </base>
BASEFONT	Token para la etiqueta <basefont>
_BASEFONT	Token para la etiqueta </basefont>
BDO	Token para la etiqueta <bdo>
_BDO	Token para la etiqueta </bdo>
BIG	Token para la etiqueta <big>
_BIG	Token para la etiqueta </big>
BLOCKQUOTE	Token para la etiqueta <blockquote>
_BLOCKQUOTE	Token para la etiqueta </blockquote>
BODY	Token para la etiqueta <body>
_BODY	Token para la etiqueta </body>
BR	Token para la etiqueta
_BR	Token para la etiqueta </br>
BUTTON	Token para la etiqueta <button>
_BUTTON	Token para la etiqueta </button>
CAPTION	Token para la etiqueta <caption>
_CAPTION	Token para la etiqueta </caption>
CENTER	Token para la etiqueta <center>
_CENTER	Token para la etiqueta </center>
CITE	Token para la etiqueta <cite>
_CITE	Token para la etiqueta </cite>
CODE	Token para la etiqueta <code>
_CODE	Token para la etiqueta </code>
COL	Token para la etiqueta <col>
_COL	Token para la etiqueta </col>
COLGROUP	Token para la etiqueta <colgroup>
_COLGROUP	Token para la etiqueta </colgroup>
DD	Token para la etiqueta <dd>
_DD	Token para la etiqueta </dd>

DEL	Token para la etiqueta
_DEL	Token para la etiqueta
DIR	Token para la etiqueta <dir>
_DIR	Token para la etiqueta </dir>
DIV	Token para la etiqueta <div>
_DIV	Token para la etiqueta </div>
DFN	Token para la etiqueta <dfn>
_DFN	Token para la etiqueta </dfn>
DL	Token para la etiqueta <dl>
_DL	Token para la etiqueta </dl>
DT	Token para la etiqueta <dt>
_DT	Token para la etiqueta </dt>
EM	Token para la etiqueta
_EM	Token para la etiqueta
FIELDSET	Token para la etiqueta <fieldset>
_FIELDSET	Token para la etiqueta </fieldset>
FONT	Token para la etiqueta
_FONT	Token para la etiqueta
FORM	Token para la etiqueta <form>
_FORM	Token para la etiqueta </form>
FRAME	Token para la etiqueta <frame>
_FRAME	Token para la etiqueta </frame>
FRAMESET	Token para la etiqueta <frameset>
_FRAMESET	Token para la etiqueta </frameset>
H1	Token para la etiqueta <h1>
_H1	Token para la etiqueta </h1>
H2	Token para la etiqueta <h2>
_H2	Token para la etiqueta </h2>
H3	Token para la etiqueta <h3>
_H3	Token para la etiqueta </h3>
H4	Token para la etiqueta <h4>
_H4	Token para la etiqueta </h4>
H5	Token para la etiqueta <h5>

_H5	Token para la etiqueta </h5>
H6	Token para la etiqueta <h6>
_H6	Token para la etiqueta </h6>
HEAD	Token para la etiqueta <head>
_HEAD	Token para la etiqueta </head>
HR	Token para la etiqueta <hr>
_HR	Token para la etiqueta </hr>
HTML	Token para la etiqueta <html>
_HTML	Token para la etiqueta </html>
I	Token para la etiqueta <i>
_I	Token para la etiqueta </i>
IFRAME	Token para la etiqueta <iframe>
_IFRAME	Token para la etiqueta </iframe>
IMG	Token para la etiqueta
_IMG	Token para la etiqueta
INPUT	Token para la etiqueta <input>
_INPUT	Token para la etiqueta </input>
INS	Token para la etiqueta <ins>
_INS	Token para la etiqueta </ins>
ISINDEX	Token para la etiqueta <isindex>
_ISINDEX	Token para la etiqueta </isindex>
KBD	Token para la etiqueta <kbd>
_KBD	Token para la etiqueta </kbd>
LABEL	Token para la etiqueta <label>
_LABEL	Token para la etiqueta </label>
LEGEND	Token para la etiqueta <legend>
_LEGEND	Token para la etiqueta </legend>
LI	Token para la etiqueta
_LI	Token para la etiqueta
LINK	Token para la etiqueta <link>
_LINK	Token para la etiqueta </link>
MAP	Token para la etiqueta <map>
_MAP	Token para la etiqueta </map>

MENU	Token para la etiqueta <menu>
_MENU	Token para la etiqueta </menu>
META	Token para la etiqueta <meta>
_META	Token para la etiqueta </meta>
NOFRAMES	Token para la etiqueta <noframes>
_NOFRAMES	Token para la etiqueta </noframes>
NOSCRIPT	Token para la etiqueta <noscript>
_NOSCRIPT	Token para la etiqueta </noscript>
OBJECT	Token para la etiqueta <object>
_OBJECT	Token para la etiqueta </object>
OL	Token para la etiqueta
_OL	Token para la etiqueta
OPTGROUP	Token para la etiqueta <optgroup>
_OPTGROUP	Token para la etiqueta </optgroup>
OPTION	Token para la etiqueta <option>
_OPTION	Token para la etiqueta </option>
P	Token para la etiqueta <p>
_P	Token para la etiqueta </p>
PARAM	Token para la etiqueta <param>
_PARAM	Token para la etiqueta </param>
PRE	Token para la etiqueta <pre>
_PRE	Token para la etiqueta </pre>
Q	Token para la etiqueta <q>
_Q	Token para la etiqueta </q>
S	Token para la etiqueta <s>
_S	Token para la etiqueta </s>
SAMP	Token para la etiqueta <samp>
_SAMP	Token para la etiqueta </samp>
SCRIPT	Token para la etiqueta <script>
_SCRIPT	Token para la etiqueta </script>
SELECT	Token para la etiqueta <select>
_SELECT	Token para la etiqueta </select>
SMALL	Token para la etiqueta <small>

_SMALL	Token para la etiqueta </small>
SPAN	Token para la etiqueta
_SPAN	Token para la etiqueta
STRIKE	Token para la etiqueta <strike>
_STRIKE	Token para la etiqueta </strike>
STRONG	Token para la etiqueta
_STRONG	Token para la etiqueta
STYLE	Token para la etiqueta <style>
_STYLE	Token para la etiqueta </style>
SUB	Token para la etiqueta <sub>
_SUB	Token para la etiqueta </sub>
SUP	Token para la etiqueta <sup>
_SUP	Token para la etiqueta </sup>
TABLE	Token para la etiqueta <table>
_TABLE	Token para la etiqueta </table>
TBODY	Token para la etiqueta <tbody>
_TBODY	Token para la etiqueta </tbody>
TD	Token para la etiqueta <td>
_TD	Token para la etiqueta </td>
TEXTAREA	Token para la etiqueta <textarea>
_TEXTAREA	Token para la etiqueta </textarea>
TFOOT	Token para la etiqueta <tfoot>
_TFOOT	Token para la etiqueta </tfoot>
TH	Token para la etiqueta <th>
_TH	Token para la etiqueta </th>
THEAD	Token para la etiqueta <thead>
_THEAD	Token para la etiqueta </thead>
TITLE	Token para la etiqueta <title>
_TITLE	Token para la etiqueta </title>
TR	Token para la etiqueta <tr>
_TR	Token para la etiqueta </tr>
TT	Token para la etiqueta <tt>
_TT	Token para la etiqueta </tt>

U	Token para la etiqueta <u>
_U	Token para la etiqueta </u>
UL	Token para la etiqueta
_UL	Token para la etiqueta
VAR	Token para la etiqueta <var>
_VAR	Token para la etiqueta </var>

Librerías externas utilizadas

No se utilizaron librerías externas. Todas las librerías utilizadas están incluidas en C.

Análisis de Resultados

Objetivos alcanzados

En este primer proyecto se logró alcanzar todos los objetivos. Los objetivos alcanzados fueron los siguientes:

- Se delimitaron los caracteres no válidos dentro del lenguaje.
 - En caso de encontrarse uno, se imprime un mensaje de error mediante el stderr indicando la fila y la columna en donde se encontró el carácter no permitido.
- La información leída es almacenada y retornada de acuerdo al token leído.
- Se creó el archivo makefile que contiene las instrucciones para correr el programa.

Además, se tomaron en cuenta objetivos fuera del primer proyecto:

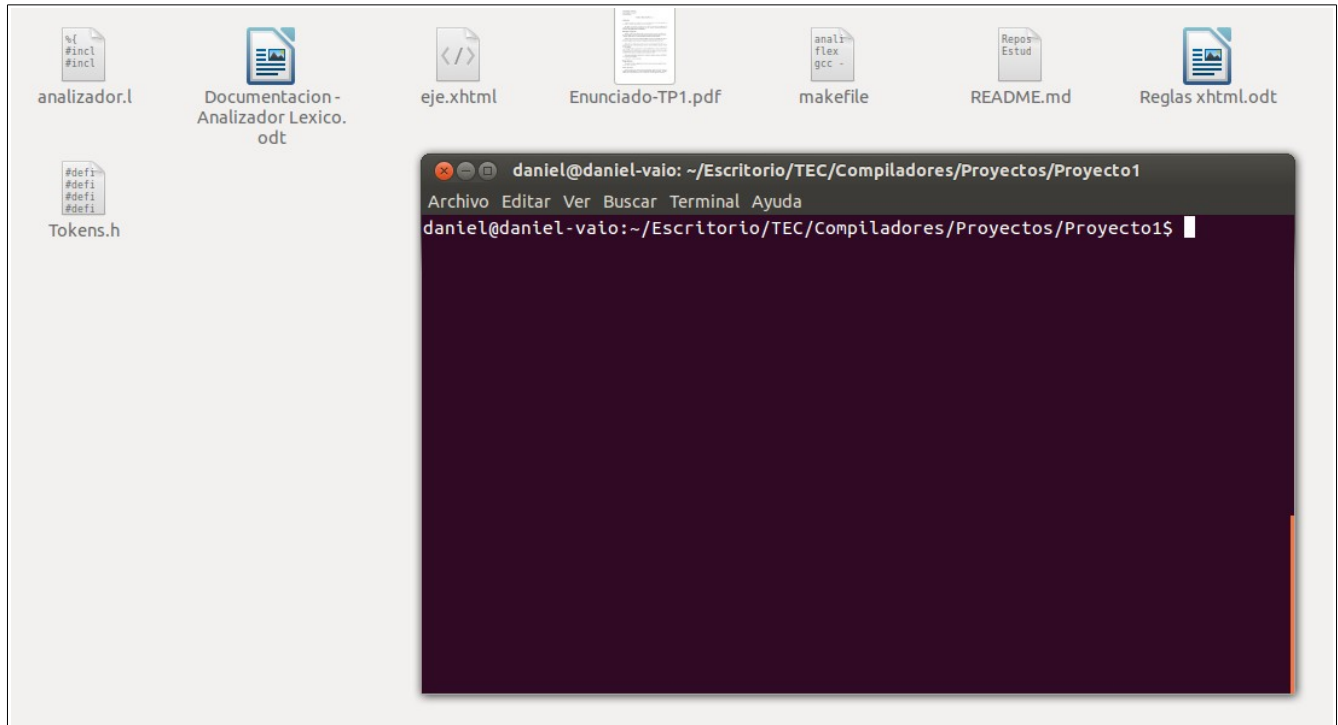
- Se definió la expresión regular que reconoce los comentarios dentro de XHTML (`<!-- →`)
 - En caso de no cerrarse la etiqueta de comentario se desplegará un error, indicando dónde se encontró un comentario sin terminar.
- Se definió la lista que almacenará los tags `<>` para determinar la correspondencia de las etiquetas. (Ejemplo: `<p><p> → error`).

Objetivos no alcanzados

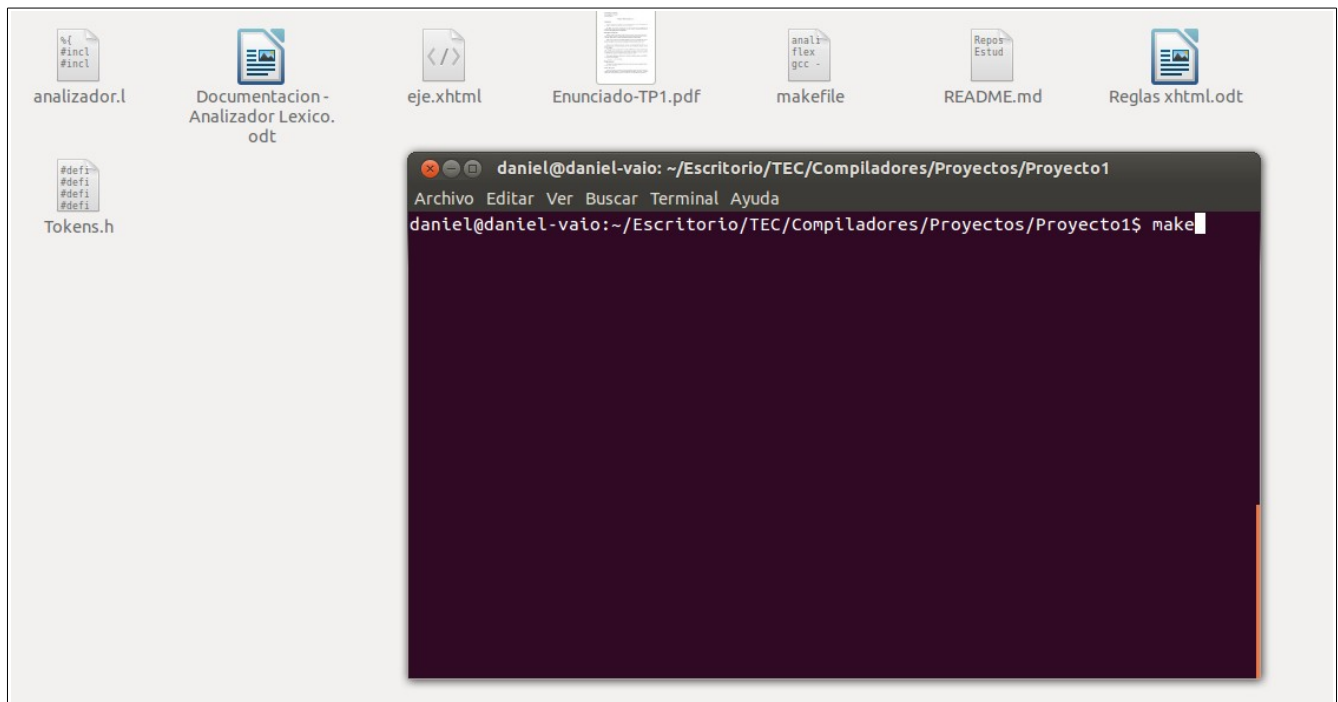
Debido a que se lograron alcanzar todos los objetivos del proyecto, no quedan objetivos no alcanzados en este primer proyecto.

Manual de usuario

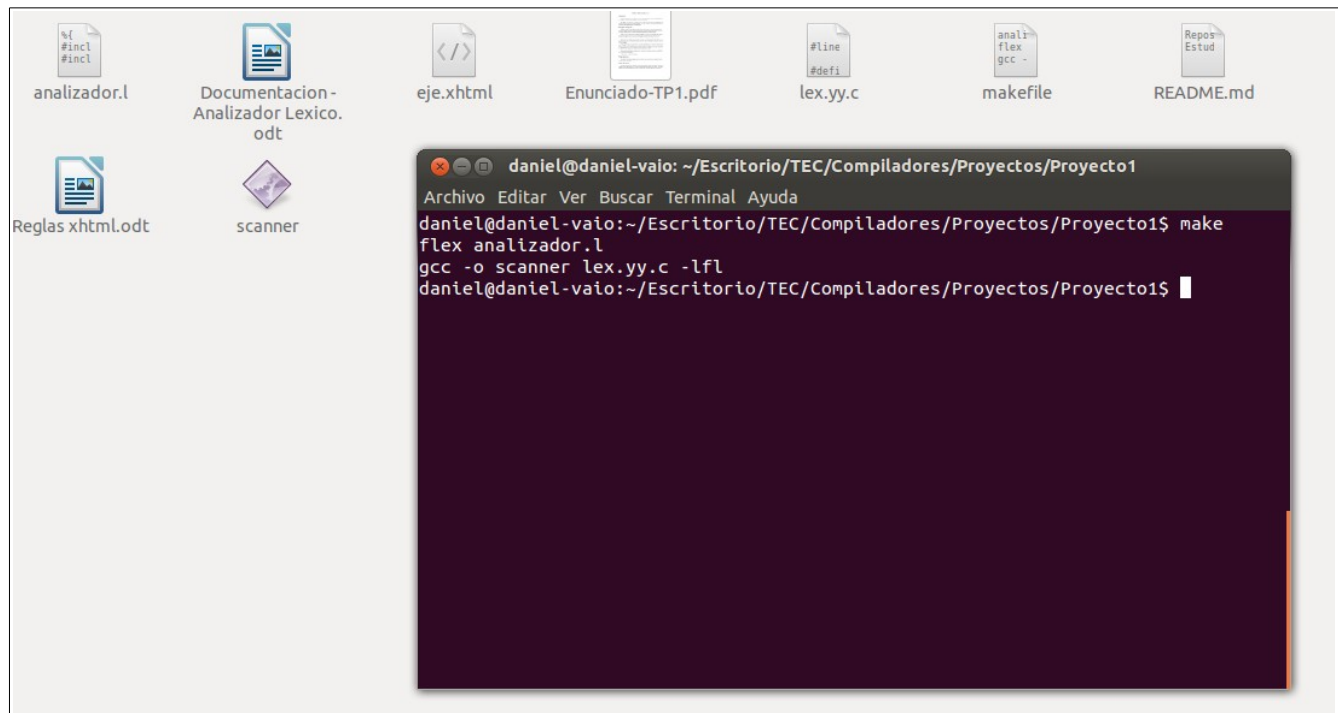
Para correr el programa primero se debe abrir una consola dentro del directorio del proyecto.



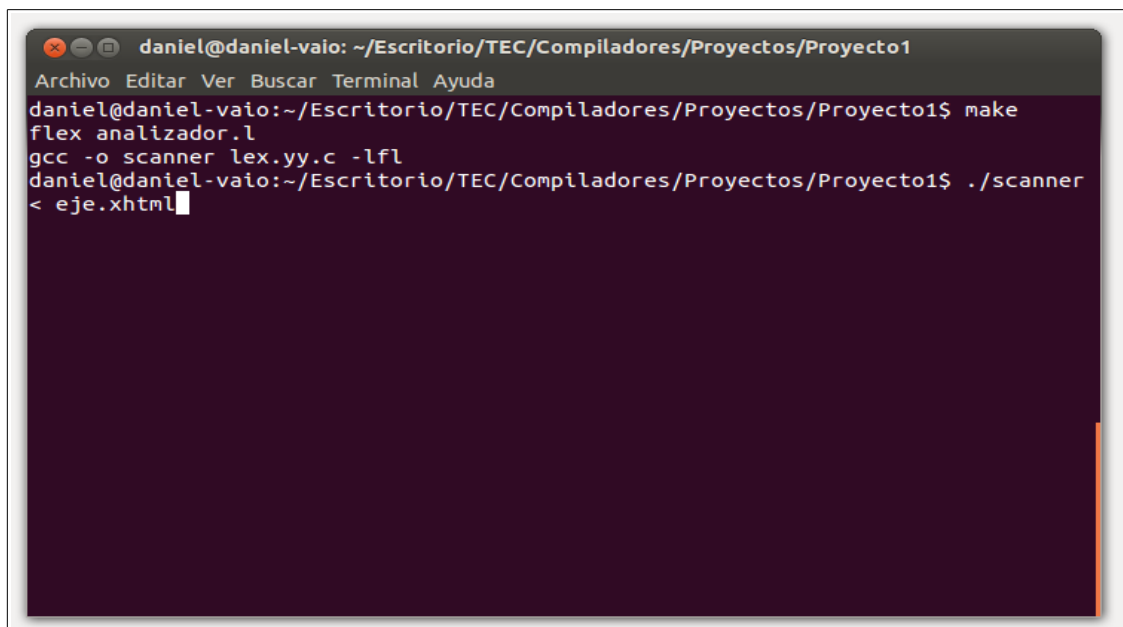
A continuación escribir el comando make el cual ejecutará las instrucciones para compilar el archivo analizador.l



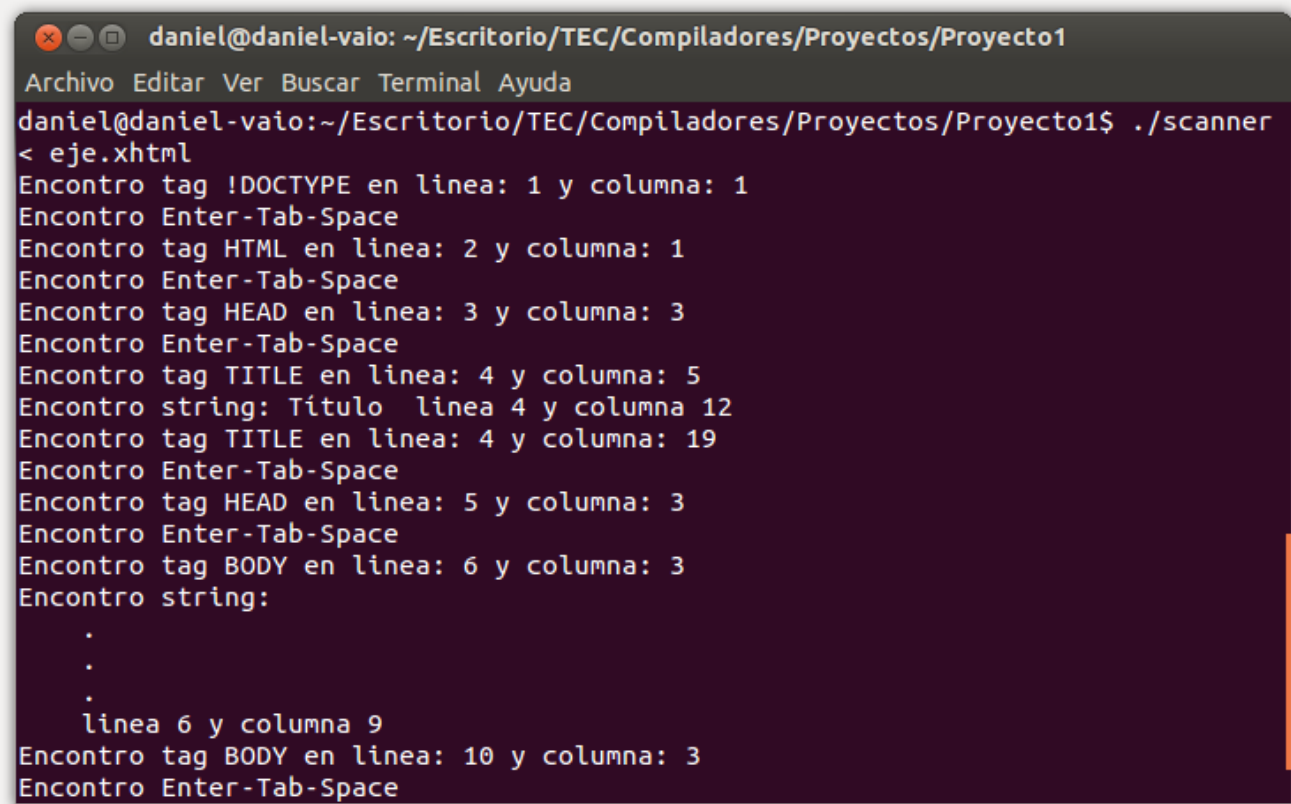
Al presionar enter se mostraran los comandos que se usaron para compilar el proyecto. Se podrá comprobar que no se utilizaron librerías externas. Se generarán los archivos scanner y lex.yy.c



Seguidamente se generará un .o llamado scanner. Para ejecutar el scanner con el archivo de ejemplo `eje.xhtml` se deberá ejecutar `./scanner < eje.xhtml`



Al ejecutar la instrucción se imprimirá por consola el resultado de la ejecución. Se imprime cada token encontrado con su línea y columna.



```
daniel@daniel-vaio: ~/Escritorio/TEC/Compiladores/Proyectos/Proyecto1
Archivo Editar Ver Buscar Terminal Ayuda
daniel@daniel-vaio:~/Escritorio/TEC/Compiladores/Proyectos/Proyecto1$ ./scanner
< eje.xhtml
Encontro tag !DOCTYPE en línea: 1 y columna: 1
Encontro Enter-Tab-Space
Encontro tag HTML en línea: 2 y columna: 1
Encontro Enter-Tab-Space
Encontro tag HEAD en línea: 3 y columna: 3
Encontro Enter-Tab-Space
Encontro tag TITLE en línea: 4 y columna: 5
Encontro string: Título línea 4 y columna 12
Encontro tag TITLE en línea: 4 y columna: 19
Encontro Enter-Tab-Space
Encontro tag HEAD en línea: 5 y columna: 3
Encontro Enter-Tab-Space
Encontro tag BODY en línea: 6 y columna: 3
Encontro string:
.
.
.
línea 6 y columna 9
Encontro tag BODY en línea: 10 y columna: 3
Encontro Enter-Tab-Space
```

Conclusión Personal

El primer proyecto sirvió para aclarar el cómo funciona un analizador léxico. Quedó claro que lo que se hace primero es definir expresiones regulares que capturen los caracteres y palabras permitidas dentro del lenguaje. Se deben definir también las expresiones regulares para determinar los caracteres inválidos.

Además, se dio a conocer la herramienta Flex más a fondo. La primera tarea corta ayudó a introducir la herramienta pero el proyecto amplió más la capacidad de Flex.

Otro aspecto importante fue la introducción de la herramienta de control de versiones Git y su uso en conjunto con GitHub. Al ser un requerimiento del proyecto, se forzó a investigar acerca los controladores de versiones. Su utilidad es impresionante, tal es así que se está usando para los proyectos de otros cursos.