

# TC2020 Matemáticas Computacionales

## Programación en FLEX

### TAREA 5

**Forma de trabajo:** En equipos de 2 integrantes

**Forma de entrega:** Subir el programa en FLEX construido a BLACKBOARD.

**Fecha de entrega:** lunes 8 de abril de 2013

Una de las grandes dificultades al leer el código fuente de una página web construida con el lenguaje HTML, es que frecuentemente esté código no se encuentra indentado de forma adecuada.

El sistema FLEX puede ser usado como un poderoso filtro para procesar archivos texto. En esta tarea el trabajo consiste en desarrollar un "scanner" con FLEX, que funcione para analizar un archivo en HTML y regrese otro archivo en HTML indentado.

Las construcciones que deben contemplarse en la página HTML de entrada son las siguientes (**entrada**):

- El HTML es un lenguaje que basa su sintaxis en un elemento de base al que llamamos etiqueta. La etiqueta presenta frecuentemente dos partes:
  - Una apertura de forma general <etiqueta>
  - Un cierre de tipo </ etiqueta>
- Todo lo incluido en el interior de esa etiqueta sufrirá las modificaciones que caracterizan a esta etiqueta.
- Aunque las etiquetas se pueden escribir como cualquier combinación de mayúsculas y minúsculas, para este proyecto solo se manejarán etiquetas en minúsculas, para estar de acuerdo con los nuevos estándares.
- Un documento html esta delimitado por las etiquetas <html> ... </html>.
- Dentro de este documento, podemos asimismo distinguir dos partes principales:
  - Primero, el encabezado, delimitado por <head> y </head> donde colocaremos etiquetas de índole informativo como por ejemplo el titulo de nuestra página.
  - El cuerpo, flanqueado por las etiquetas <body> y </body>, que será donde colocaremos nuestro texto, tablas, listas e imágenes delimitados a su vez por otras etiquetas.
- Dentro del encabezado solo vamos a distinguir el título de la página, el cual es delimitado por las etiquetas <title> y </title>. El resto del contenido del encabezado indentarse al igual que las etiquetas del título.
- Dentro del cuerpo del documento vamos a distinguir:
  - <p> ... </p> para párrafos.
  - <font> ... </font> para estilos de texto.
  - <hn> ... </hn> para encabezados, donde la n puede ser un valor de 1 a 6.
  - <a> ... </a> para enlaces.
  - <table> ... </table> para tablas.
  - <ul> ... </ul> para listas.

- Una tabla se compone por renglones, los cuales son delimitados por las etiquetas `<tr>` y `</tr>`.
- A su vez, los renglones se componen de campos, los cuales son delimitados por las etiquetas `<td>` y `</td>`.
- Un campo de la tabla puede contener todos los elementos mencionados como parte del cuerpo del documento. De forma que una tabla puede tener anidadas otras tablas, listas, texto libre o formateado en párrafos, etc.
- Una tabla puede contener cualquier cantidad de renglones, y cada renglón puede contener cualquier cantidad de campos.
- Las listas especifican sus elementos mediante etiquetas `<li>`. El contenido del elemento se extiende hasta la siguiente etiqueta `<li>`. Después del último elemento se agrega una etiqueta `</li>`.
- Todas las etiquetas de apertura pueden tener propiedades definidas después de la etiqueta como por ejemplo en `<body BACKGROUND= "panorama.jpg">`.

Construcción de la salida del scanner (**salida**):

- Tu sistema debe ir revisando el archivo HTML de entrada e ir detectando las diferentes etiquetas del lenguaje.
- Una vez detectada, la etiqueta debe escribirse en una línea separada, e indentada adecuadamente.
- Cada línea deberá indentarse de acuerdo a su nivel de anidación en el código HTML, desplazándose 4 espacios por cada nivel de anidación.
- Lo contenido entre las etiquetas de apertura y cierre de cualquier tipo, se considera con un nivel de anidación adicional al de sus etiquetas.
- De esta forma, como las etiquetas `<html>` y `</html>` no se encuentran anidadas, estas deberán aparecer al inicio de la línea.
- Como el encabezado está anidado dentro de las etiquetas `<html>` y `</html>`, las etiquetas `<head>` y `</head>` deberán escribirse después de los primeros 4 espacios de la línea. Y así en adelante.
- De acuerdo a las reglas de construcción establecidas para las páginas HTML, la única forma de incrementar el nivel de anidación de las etiquetas y su contenido es por medio de la inclusión de tablas.
- Para no tener problemas con la anidación, deberán de eliminar todos los caracteres de tabulador que se encuentren (`\t`).

Tus fuentes deben compilar en la plataforma **cygwin**, usando exactamente los comandos vistos en clase. Se supone que el archivo HTML de entrada es entregado precisamente a la entrada de flex, o sea que para ejecutarlo es necesario redirigir la entrada, como en el comando siguiente:

```
>./identador.exe < test1.htm
```

La salida es simplemente enviada a la pantalla (salida estándar).

Solamente uno de los integrantes del equipo deberá subir el trabajo a blackboard, como un mensaje que contenga el **M\_Tarea5.I**, donde deberán sustituir la M por las matrículas de los

integrantes del equipo, separadas mediante caracteres de subrayado (\_). Indicar tanto en el mensaje, como dentro del archivo fuente en FLEX (como comentarios de C), el nombre de los integrantes del equipo y sus matrículas, ya que solamente los que aparezcan tendrán derecho a calificación.

## RECOMENDACIONES

- **No comenzar el trabajo unos cuantos días antes de la entrega.**

El dejar las cosas para lo último puede provocar problemas de organización, presión por la cercanía de la fecha de entrega, etc. A veces es necesario hacer pausas durante sesiones de trabajo para meditar una idea, tomar un poco de aire o simplemente dejar descansar al cerebro unos instantes.

- **Desarrollar la tarea en forma incremental.**

Siempre es malo desayunar todo lo de una semana en un sólo día o comerse el pastel de una sola mordida. Resuelvan primero las partes más fáciles de la tarea y a medida que esas partes queden resueltas podrán avanzar entonces a los casos más difíciles (por ejemplo, se puede comenzar generando un archivo de salida con la tabla en formato horizontal y sin traducciones, e integrar esos detalles posteriormente). Esto es importante pues se harán varias pruebas a la tarea, algunas fáciles, otras más difíciles, y cada prueba les hace ganar puntos. En caso de que su tarea esté incompleta es conveniente como sea que suban lo que alcanzaron a hacer, pues de lo contrario su tarea sería evaluada con 0 (cero).

- **Dividan el trabajo adecuadamente.**

Si la tarea se encarga en equipo no es para que los miembros del equipo estén pegados a la computadora al mismo tiempo tratando de resolver el problema. Es para que identifiquen de manera inteligente como pueden dividir la tarea en sub-tareas que puedan ser resueltas en forma independiente y que luego puedan ser ensambladas para generar el sistema final.

- **Investiguen lo que hace falta para comenzar la tarea antes de empezar a resolverla.**

Muchas veces por no invertir un poco de tiempo a actividades de investigación se reinventa el hilo negro y se programan cosas para las cuales los lenguajes o herramientas computacionales nos proveen soluciones predefinidas. Por ejemplo, no es necesario que investiguen como generar salidas a archivos por medio de estatutos de I/O pues Unix permite redirigir la salida estándar a un archivo.