

## TAREA PROGRAMADA I

### Introducción

El objetivo de esta tarea es familiarizarse con los conceptos relativos a la fase de análisis léxico de un compilador; mediante el desarrollo de un scanner para XHTML.

Para efectos de esta tarea, se podrá usar Flex o JFlex para generar el analizador léxico. El programa deberá transformar el código XHTML a una serie de tokens que tengan información que será usada en las siguientes fases del procesamiento.

### Descripción del programa

Deberán escribir una especificación de Flex o JFlex para construir un scanner para XHTML. Los estudiantes deberán investigar la especificación estándar de XHTML, definir la lista de tokens que van a manejar; y generar las ER, y las acciones asociadas a esas ER, para procesar los tokens.

Deben tomar en cuenta que la información recopilada por el scanner en el análisis léxico va a ser usada por el analizador sintáctico, por lo que deben guardar toda la información relevante en las variables de Flex/JFlex, para que posteriormente puedan ser usadas por el parser (yylen, etc.)

El programa no deberá usar argumentos de línea de comandos, solamente deberá ser un ejecutable que reciba la entrada del **stdin**, escriba los tokens resultantes al **stdout**, y reporte errores al **stderr**.

Como parte de la entrega, deberán tener un archivo MAKEFILE (en el caso de java podrían usar ANT o MAVEN, de forma alternativa) que tenga todas las instrucciones necesarias para compilar/configurar el programa, de manera que con solo los archivos entregados, se pueda ejecutar el comando MAKE, y se generen automáticamente los ejecutables de la tarea.

Para invocar el programa, se deberá ejecutar el siguiente comando (asumiendo que el MAKEFILE construye el archivo **scanner**):

```
./scanner < archivoFuente
```

### Manejo de errores

El analizador léxico deberá registrar los errores léxicos que se presenten, y enviar la información de dichos errores al **stderr**.

### Pruebas funcionales

Como parte de la entrega de la tarea, tienen que desarrollar y entregar una serie de archivos de prueba que permitan verificar el correcto funcionamiento del scanner. Los archivos de prueba deberán incluir tanto casos válidos (i.e., HTML correcto), como casos inválidos (HTML incorrecto).

## Aspectos técnicos

El proyecto podrá ser escrito en C, C++ o Java, y deberá de funcionar en el sistema operativo Linux. En caso de requerir librerías adicionales para compilar y ejecutar el programa, deberán especificarlo en la documentación, ya que de lo contrario se descontarán puntos en la evaluación.

## Documentación

La documentación es un aspecto de gran importancia en el desarrollo de programas, especialmente en tareas relacionadas con el mantenimiento de los mismos.

La documentación externa deberá incluir:

- Tabla de contenidos
- Descripción del problema
- Diseño del programa: decisiones de diseño, algoritmos usados, lista de tokens que se van a manejar, y descripción de cada uno, razones de escogencia del generador de analizadores léxicos (Flex, JFlex)
- Librerías externas usadas: las librerías externas deben ser solamente de funcionalidades secundarias, y no de las funcionalidades claves de la tarea (hacer scanning del XHTML).
- Análisis de resultados: objetivos alcanzados, objetivos no alcanzados, razones por las cuales no se alcanzaron objetivos
- Manual de usuario (instrucciones uso, de ejecución y de compilación usando MAKEFILE). Es importante que especifiquen de manera clara todos los pasos que se tienen que efectuar para correr la tarea. Las instrucciones de compilación deben ser de línea de comandos (no de IDE). Deben explicar cómo instalar/configurar las librerías adicionales que usen en la tarea, en caso de haberlas (el MAKEFILE debería hacer eso automáticamente)
- Conclusión personal

## Evaluación

Documentación interna	2%
Documentación externa	13%
Funcionamiento correcto del analizador léxico	45%
Revisión de tarea	40%

## Aspectos administrativos

- La tarea vale un 10% de la nota del curso
- La tarea se hará en grupos de 2 personas.

- Fecha de entrega: Lunes 25 de marzo, 1:00 p.m. No se aceptan tareas entregadas después de esa fecha.

- Las dudas con respecto a las tareas deberán ser enviadas a los siguientes correos: [andreifu@gmail.com](mailto:andreifu@gmail.com) y [brivera@ic-itcr.ac.cr](mailto:brivera@ic-itcr.ac.cr)

- El medio de entrega será por medio de Github. Los estudiantes deberán enviar un correo a la dirección [brivera@ic-itcr.ac.cr](mailto:brivera@ic-itcr.ac.cr), con copia a [andreifu@gmail.com](mailto:andreifu@gmail.com), con el link del repositorio de Github en donde se encuentran todos los archivos de la tarea programada. Tanto el nombre del repositorio, como el asunto del correo enviado deberán ser **IC-5701 TP1-nombre1-nombre2-nombre3**

- El código que entreguen debe ser original. En caso contrario, se calificará con nota cero.