

5. CONCLUSIONES

En este trabajo se presentó el desarrollo de un intérprete denominado CXML que permite verificar si un documento XML está bien formado y emitir un informe de salida para dicho documento. La realización de este intérprete permitió, por un lado, profundizar los conceptos teóricos y prácticos adquiridos durante el año en las asignaturas Teoría de la Computación y Lenguaje de Programación y Compiladores. Por otro lado, adquirir habilidades y destrezas en el uso de los generadores automáticos como Flex y Bison y, en programación con lenguaje C. Pero además, se logró llevar los conceptos teóricos a la práctica computacional, para obtener como resultado una herramienta de software aplicable a problemas del mundo real.

Por último, ante el desafío presentado por la cátedra, se desarrollaron un conjunto de expresiones regulares, se codificaron en el fichero de entrada al Flex, se diseñó la gramática para el lenguaje XML que fue posteriormente codificada en Bison y, por último, se implementaron las rutinas y estructuras de datos necesarias para emitir el informe de salida. Es así que, se obtuvo un intérprete válido y completo, donde se verifica la sintaxis y se emite mensajes de errores si el archivo de entrada no es correcto.

Los trabajos futuros se orientan a realizar un editor de documentos XML que facilite las tareas de edición a los usuarios. Entre las funciones que se proponen es que posibilite la inserción de etiquetas de fin en forma automática luego de que el usuario haya ingresado una etiqueta de inicio. También se pretende generar un archivo con la estructura del documento XML ingresado, un archivo que contenga los elementos y atributos a definir, es decir, el esquema estructural del documento XML. Para esto se utilizará el lenguaje de definición de esquema DTD (Document Type Definition).

6. BIBLIOGRAFÍA

1. Aho, Sethi, Ullman. Principios, Compiladores: técnicas y herramientas. Addison Wesley, 2010.
2. Barchini de Gimenez, Graciela y Alvarez de Benitez Margarita. Fundamentos Teóricos de la Ciencia de la Computación, Departamento de Informática. FCEyT, 2009.
3. Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E., & Yergeau, F. Extensible markup language (XML). World Wide Web Consortium Recommendation REC-xml-19980210. <http://www.w3.org/TR/1998/REC-xml-19980210>.
4. Louden, K. C. Construcción de compiladores: principios y práctica. Cengage Learning Editores, 2004.
5. Part, X. S. World Wide Web Consortium (W3C), Recommendation (October 28, 2004), <http://www.w3.org/TR/xmlschema-1>.
6. Pisabarro Marrón, Alma María. "El generador de analizadores sintácticos YACC- Teoría de autómatas y lenguajes formales", Universidad de Valladolid
7. Simmross Wattenberg, Federico. "El generador de analizadores léxicos lex – Teoría de Autómatas y Lenguajes formales", Universidad de Valladolid.

Otros Sitios de Consulta:

8. http://es.wikipedia.org/wiki/Extensible_Markup_Language
9. https://sites.google.com/site/emersalva/documentacion_a40235_tarea7#_Toc265082561
10. <http://www.euskonews.com/0471zkb/gaia47101es.html>