

# UNA ESTRATEGIA DIDÁCTICA PARA EL DICTADO DE INFORMÁTICA TEÓRICA

(X Congreso edutec 2007 Buenos Aires - Argentina)

Ing. Marciszack, Marcelo Martín – [mmarciszack@istemas.frc.utn.edu.ar](mailto:mmarciszack@istemas.frc.utn.edu.ar)

Ing. Vázquez, Juan Carlos J. – [jvazquez@bbs.frc.utn.edu.ar](mailto:jvazquez@bbs.frc.utn.edu.ar)

Ing. Meloni, Brenda Elizabeth – [bmeloni@hotmail.com](mailto:bmeloni@hotmail.com)

Ing. Paz Menvielle, María Alejandra [pazmenvielle@yahoo.com.ar](mailto:pazmenvielle@yahoo.com.ar)

Departamento de Ingeniería en Sistemas de Información  
Facultad Córdoba – Universidad Tecnológica Nacional  
Ciudad Universitaria, Córdoba 5016, Argentina

## RESUMEN

Para poder asimilar los conocimientos de informática teórica dictados en las diferentes carreras de Informática, es necesario contar con un conjunto integrado de herramientas didácticas que permitan, por un lado, aportar sólidas bases teóricas y por otro, disponer de claros ejemplos de aplicación a la resolución de problemas reales, como las herramientas para simulación, las implementaciones sobre casos concretos y el desarrollo de algoritmos, entre otros.

Los estudiantes presentan dificultad para asimilar estos contenidos ya que los mismos están muy cercanos a la matemática y son fuertemente abstractos. Concientes de que estamos preparando futuros profesionales, nuestra preocupación es la de mostrar claramente la aplicabilidad de estas herramientas conceptuales, no sólo a la construcción de compiladores (enfoque tradicional de la teoría de los autómatas y lenguajes) sino a la resolución de problemas en general.

Por otro lado, es necesario lograr la integración de conocimientos de Teoría de la Computación, Programación e Ingeniería de Software, permitiendo a los educandos probar

sus modelos teóricos, inspeccionar, modificar las herramientas, optimizarlas y analizar su complejidad, alcanzando de esta forma un aprendizaje significativo.

Por lo tanto el presente trabajo está orientado al desarrollo de estas herramientas y materiales de estudio, sentando bases sólidas de conocimientos teóricos, con una gran ejercitación y aplicabilidad de los mismos.

**Palabras Claves:** Informática Teórica, Máquinas abstractas, Gramáticas Formales, Autómatas, Compiladores

## 1. INTRODUCCION

Los contenidos de informática Teórica, incluidos en el núcleo común de las carreras con informática (lenguajes y gramáticas formales, teoría de autómatas, traducción de lenguajes y otros) son difíciles de relacionar por los estudiantes, por su naturaleza matemática y fuertemente abstracta.

Además los estudiantes efectúan prácticas de diseño de máquinas abstractas, sin contar con herramientas para la experimentación en computadora y exploración de contenidos, que les permita en forma clara aplicarlos en

la resolución de problemas reales. Esta dificultad manifiesta de relacionar la teoría y sus posibles aplicaciones, se ha advertido en los rendimientos durante las evaluaciones de los mismos.

Para asimilar en forma completa estos contenidos, consideramos que se deben realizar prácticas efectivas de los mismos utilizando generadores de analizadores lexicográficos y sintácticos, simuladores de máquinas abstractas, conversores de gramáticas a autómatas y otros, estudiando además su funcionamiento y los algoritmos involucrados.

De igual manera para producir un aprendizaje significativo de la teoría que da sustento a la informática teórica, es necesario que el educando pueda explorar estos conocimientos y mediante un evaluador de contenidos teóricos (que incluye en ambiente web la generación de un instrumento de evaluación automatizado basado en un repositorio de preguntas y respuestas) le permita en forma aleatoria generar un test de autoevaluación.

La importancia de una completa asimilación de estos contenidos, radica en que los mismos no sólo tienen efectiva aplicabilidad en la construcción de compiladores, sino en temas tan diversos dentro de la disciplina informática como son la ingeniería de requerimientos o programación de dispositivos móviles y embebidos.

Existe una gran cantidad de bibliografía y herramientas de software para el estudio de estos temas, pero las mismas los abordan en forma parcializada, sin integración adecuada, con diferentes nomenclaturas y con definiciones dispares, por lo que no resultan didácticamente adecuadas para llevar adelante el proceso de enseñanza-aprendizaje, sin permitir la correcta transposición de los contenidos teóricos en aplicaciones de la realidad.

Este proyecto tiene como objetivos lograr profundo conocimiento teórico y práctico, de

las técnicas y herramientas conceptuales de la Informática Teórica relacionadas con la construcción de compiladores, e involucrar a alumnos en el estudio y la investigación de temas fundamentales de su carrera.

Los aspectos teóricos, para el abordaje de la temática planteada, esta muy difundido y es bastante generalizado: sobre la teoría de autómatas y lenguajes formales, es completo lo publicado en [1] [2] [3] [4] [8] [10], con respecto específicamente a la aplicabilidad en el modelo abstracto sobre la máquina RAM [5] [6]; respecto a la utilización de técnicas y herramientas para la construcción de compiladores [7] [8] [9], es variado y extenso, pero los mismos no se ajustan en lo didáctico, para ser transferido en forma directa a los alumnos en una carrera de grado en informática.

En la web, existen varios sitios en donde se pueden acceder a simuladores, generalmente la mayoría de ellos son para máquinas de Turing en donde, de acuerdo a la bibliografía de origen, varían en la definición formal de sus componentes. Por lo tanto, para lograr un aprendizaje significativo de los contenidos curriculares de teoría de autómatas, gramáticas formales, traducción de lenguajes y compiladores impartidos en la asignatura Sintaxis y Semántica del Lenguaje, se debe elaborar una estrategia didáctica para la enseñanza de los contenidos, en base a una nomenclatura uniforme que permita la representación coherente.

En este trabajo se propone construir un conjunto de herramientas disponibles para ejecutar desde la web: simuladores de máquinas abstractas (autómatas finitos, con pila, linealmente acotados, máquinas de Turing) y generadores de analizadores, brindando los fuentes para que estudiándolos y comprendiendo su funcionamiento se vea la aplicación de los contenidos teóricos y, en todo momento además, poder autoevaluar sus aprendizajes.

## 2. OBJETIVO GENERAL

El objetivo de este trabajo es lograr que los estudiantes fijen los contenidos curriculares de teoría de autómatas, gramáticas formales, traducción de lenguajes y compiladores, realizando práctica efectiva de los mismos en simuladores de máquinas abstractas, estudiando el funcionamiento y los programas fuente de los mismo, permitiendo modificaciones que agreguen nuevas funcionalidades y para optimizar su funcionamiento.

### Objetivos Específicos

**Cognitivos:** lograr profundo conocimiento teórico y práctico, de las técnicas y herramientas conceptuales de la informática teórica, relacionadas con la construcción de compiladores, con el reconocimiento general de patrones, la programación automática desde una especificación formal, la traducción de códigos y los modelos de computación lineal e introducirnos a los modelos paralelos.

**Académicos:** Generar un marco, para la enseñanza-aprendizaje de la informática teórica y propender a la actualización constante de contenidos involucrando a alumnos en el estudio serio y la investigación de temas fundamentales de su carrera.

## 3. METODOLOGÍA

El trabajo en general está guiado por el método científico, ya que detectado el problema (baja comprensión de la conexión entre teoría y práctica) se realizó la recolección de datos sobre el rendimiento académico de los alumnos en los últimos años, encuestas de dificultades declaradas por los educandos y el estudio del marco teórico, se planteó una hipótesis (la falta de visión sobre aplicaciones reales) y se desarrolló una

solución (nuestro proyecto) que incluye una herramienta de medición para contrastar resultados obtenidos.

Dentro del proyecto, el modelado se realizó con orientación a objetos, utilizando para la especificación UML y para el desarrollo se utilizó la técnica de programación extrema (XP); en la construcción de los simuladores adoptamos un lenguaje de programación de última generación, Java, utilizado por la mayoría de los estudiantes de las carreras de informática.

Para integrar los contenidos y facilitar el aprendizaje, se construyó el sitio Web donde se encuentran disponibles los simuladores para la realización de prácticas, conjuntamente con material teórico, práctico y de evaluación sobre los contenidos de la asignatura.

## 4. DESARROLLO

El trabajo comenzó con la recolección de datos: a) sobre el rendimiento de alumnos de la asignatura de Sintaxis y Semántica del Lenguaje y b) del material de las temáticas involucradas que se encuentra publicado, continuando con la selección y clasificación del mismo.

Posteriormente, se procedió a la unificación de la nomenclatura y simbología ha utilizar.

Se estableció la estrategia didáctica para abordar los contenidos de manera de facilitar el aprendizaje de los alumnos incluyendo todo en la estructura del sitio web.

Se han desarrollado simuladores como herramientas didácticas para poder ejemplificar y visualizar los conceptos teóricos involucrados en informática teórica. Estas herramientas en ambiente web, permiten ver los resultados gráficamente en un applet o expresado como XML.

En una primera etapa se desarrolló un modelo general de Conjuntos, concepto

base de la teórica de lenguajes, como de otros tantos modelos matemáticos; sobre el que se basan los modelos correspondientes a las gramáticas y a los autómatas.

Los simuladores de gramáticas permiten definir las mediante los alfabetos de símbolos terminales y no terminales, el axioma y el conjunto de producciones; los usuarios podrán cambiar los componentes que define la gramática, y visualizarlas en un grafo, en un XML o en notación formal.

El simulador permite identificar el tipo de gramática, identifica símbolos superfluos terminales y no terminales, símbolos inaccesibles, permitiendo limpiar la gramática o indicar si se encuentra en una Forma Normal. En los casos de gramáticas regulares y dependientes del contexto, se generará el autómata correspondiente.

Respecto de los simuladores de autómatas, en esta primera etapa, nos enfocamos en los autómatas finitos y los autómatas a pila, que son los de más relevancia en la fase de análisis dentro de un compilador.

Estos simuladores permiten explorar, pudiendo ingresar y modificar los componentes que los definen, los alfabetos (de entrada, salida y pila), el conjunto de estados, el conjunto de estados finales, el estado inicial y las transiciones.

En ambos casos se reconoce si una palabra pertenece a un lenguaje generado por una gramática y si puede ser aceptada por un autómata.

También mediante conversores se puede pasar un autómata a una gramática, de acuerdo a su tipo, y también realizar el proceso inverso; de ésta manera se entiende el isomorfismo entre las gramáticas y los autómatas.

Al poder los alumnos visualizar una aplicación práctica de los contenidos, y

poder simular la ejecución de autómatas finitos, con pila, gramáticas formales y su isomorfismo, de manera que ellos mismos evalúen las soluciones a los problemas planteados en las clases prácticas, es de esperar que los conocimientos y habilidades adquiridas, sean realmente significativas y eleven su desempeño en el cursado de Informática Teórica

## 5. PRODUCTOS CONSTRUIDOS

Página Web, con los contenidos teóricos de la asignatura, prácticos resueltos y propuestos y el generador de test automatizados para la autoevaluación de los alumnos.

Para la ejercitación práctica de los contenidos, se desarrollaron las siguientes herramientas:

- Simuladores para la ejercitación práctica de la teoría de Conjuntos.
- Reconocedor de cadenas generadas por gramáticas formales interpretadas.
- Simuladores de autómatas finitos, con pila, linealmente acotados y máquina de Turing, para que los alumnos evalúen sus soluciones a los problemas planteados en las clases prácticas.
- Generador de analizadores léxicos a partir de una gramática regular que muestre la generación automática de código desde una especificación formal y las distintas formas de tratar con el no determinismo del autómata finito obtenido: conversión a autómata finito determinista, búsqueda en árboles generados por estados posibles y procesamiento en paralelo de los distintos caminos deterministas generados por el no determinismo del autómata.
- Generador de analizadores sintácticos LL(k) y LR(k) desde la especificación de un lenguaje según su gramática libre de contexto.

## **6. IMPACTO DEL TRABAJO**

Además de facilitar la enseñanza-aprendizaje de los contenidos de informática teórica, se pretende que las buenas prácticas adquiridas en los primeros momentos de la formación, impacten a futuro en el desarrollo de la industria de software de la región, industria intelecto-intensiva.

Todo el conocimiento que se pueda lograr sobre estos temas, contribuye directa o indirectamente al desarrollo tecnológico de software, al desarrollar en nuestros alumnos y docentes, mejores prácticas en lo que se refiere a la construcción de software, fomentando su utilización, desde los inicios en la formación académica, con la utilización de métodos formales, y la aplicación de técnicas de calidad en la construcción del mismo.

## **7. RESULTADOS**

Durante el ciclo 2007 los contenidos teóricos, ejercitación práctica, las herramientas construidas y el test de autoevaluación de contenidos, están a disposición de los alumnos en sitio web alojados en la intranet de la Facultad Regional Córdoba, a modo de prueba piloto para que sean efectivamente puestos a prueba.

Si bien se encuentra en pleno dictado la asignatura, ya se evidencia que el resultado obtenido ha sido satisfactorio, desde las siguientes dimensiones de análisis: la primera, que resulta subjetiva ya que no existen parámetros que de medición que la justifiquen, es que se ha reflejado con respecto a ciclos anteriores que los alumnos se encuentran mas motivados al poder verificar con los simuladores sus ejercitaciones, y la segunda que si es verificable, ya que aplicar la misma

metodología de evaluación, durante el mismo período y sobre idénticos contenidos al efectuar una comparación de los rendimientos académicos con ciclos lectivos anteriores, el promedio de las calificaciones obtenidas por los estudiantes se ha visto incrementado en algo superior a un punto es una escala de diez.

Estas mediciones se harán efectivas, sobre el final de dictado de la asignatura, y serán ratificadas o rectificadas cuando en el ciclo lectivo 2008, estén disponibles para todos los alumnos de todas las comisiones en el dictado de la asignatura.

## **8. CONCLUSION**

Las herramientas desarrolladas que se encuentran en etapa de prueba piloto, junto con los contenidos teóricos y prácticos de la asignatura, puesto a disposición de los alumnos en un sitio web, se han comenzado a utilizar en forma satisfactoria ya que los alumnos han evidenciado aplicación práctica de los contenidos teóricos, han podido verificar los resultados de la ejecución de los prácticos, y han podido autoevaluar sus aprendizajes previamente a la instancia de evaluación.

La prueba piloto realizada durante el 2007 permitirá introducir los cambios necesarios para facilitar la comprensión por parte de los alumnos. A partir del próximo ciclo estará a disposición para los alumnos en general, en donde serán tenidos en cuenta los resultados observados durante el desarrollo de la primera etapa durante el presente ciclo lectivo.

Para completar la utilización de este conjunto de herramientas, y como futuro trabajo, durante el ciclo lectivo 2008, se realizará una medición adicional, en donde se aplicarán los

test de evaluaciones como instrumento predictivo, permitiendo detectar tópicos de dificultad de manera de efectuar correcciones en el dictado y producir de esta manera una mejora en el proceso de enseñanza-aprendizaje.

### **Agradecimientos**

Agradecemos a la Agencia Córdoba Ciencia y a las autoridades de la Facultad Regional Córdoba de la Universidad tecnológica Nacional, el subsidio y apoyo para la realización del proyecto.

### **9. BIBLIOGRAFIA**

[1] J. E. Hopcroft / Rajeev Motwani / Jeffrey D. Ullman (2002), Introducción a la Teoría de Autómatas Lenguajes y Computación, Addison-Wesley P. C.

[2] J. Glenn. Brookshear (1993), Teoría de la Computación, Addison-Wesley Iberoamericana

[3] Dean Kelley (1995), Teoría de Autómatas y Lenguajes Formales, Prentice Hall

[4] P. Isasi / P. Martínez / D. Borrajo (1997), Lenguajes, Gramáticas y Autómatas, Addison-Wesley

[5] Alberto Bertoni / Giancarlo Mauri / Nicoleta Sabadini (1981) A Characterization of the class of Functions computable in Polynomial time on Random Acces Machines, ACM pp. 168-176

[6] Nader Bshouty (1993), On the Complexity of Functions for Random Access Machines. Journal of the Association for Computing Machinery, Vol 40 N° 2 pp. 211-223

[7] V. Aho / R. Sethi / J. D. Ullman ( 1997 ), Compiladores: principios, técnicas y herramientas, Addison-Wesley

[8] Martin, John C. (1991) Introduction to Languages and the Theory of Computation. Mac. Graw-Hill

[9] Karen A. Lemone (1996), Fundamentos de Compiladores, CECSA

[10] Fernández, G.; Sáez Vacas, F. (1995) Fundamentos de Informática, Lógica, Autómatas, Algoritmos y Lenguajes", Anaya Multimedia, Madrid.