

1. Datos Generales de la asignatura

Nombre de la asignatura:	Lenguajes y Autómatas II.
Clave de la asignatura:	SCD - 1016
SATCA¹:	2 - 3 - 5
Carrera:	Ingeniería en Sistemas Computacionales.

2. Presentación

Caracterización de la asignatura

En ésta asignatura se debe desarrollar el análisis semántico, la generación de código, la optimización y la generación del código objeto para obtener el funcionamiento de un compilador.

También se busca proveer al estudiante de herramientas, conocimientos y habilidades necesarias para desarrollar un compilador con base en los conocimientos previos de la asignatura Lenguajes y Autómatas I. La aportación de ésta asignatura es relevante en el ámbito del desarrollo de software de sistemas.

Es indispensable distinguir que la carrera de Ingeniería en Sistemas Computacionales se basa, no sólo en el desarrollo de software comercial y administrativo, sino también en el desarrollo de software científico y para el desarrollo tecnológico. Ésta asignatura se ubica en la segunda categoría y es indispensable desarrollar software en estos campos para preparar a los egresados y tengan la posibilidad de cursar posgrados de alto nivel.

La asignatura trata de concretar un traductor iniciado en la asignatura previa para que el estudiante comprenda que es capaz, mediante técnicas bien definidas, de crear su propio lenguaje de programación.

La aportación de la asignatura al perfil del egresado será específicamente la siguiente:

- Implementa aplicaciones computacionales para solucionar problemas de diversos contextos, integrando diferentes tecnologías, plataformas o dispositivos.
- Diseña, desarrolla y aplica modelos computacionales para solucionar problemas, mediante la selección y uso de herramientas matemáticas.
- Diseña e implementa interfaces para la automatización de sistemas de hardware y desarrollo del software asociado.

Intención didáctica

La asignatura consta de cuatro bloques estructurados y definidos que abarcan la última etapa de la fase de análisis y síntesis. Al término del semestre se debe obtener un compilador o traductor completo, funcionando de acuerdo a ciertas restricciones y requisitos.

La primera unidad se centra totalmente en el analizador semántico, por lo que el analizador sintáctico debió ser concluido en la asignatura de lenguajes y autómatas I, ya que servirá de base en esta unidad.

¹ Sistema de Asignación y Transferencia de Créditos Académicos

En la segunda unidad se analizan las técnicas para generar código intermedio, para incluirse en su proyecto.

La tercera unidad se centra en la optimización del código. Es importante hacer notar que de ésta fase depende la buena y eficiente ejecución del código objeto.

En el último bloque se aborda el tema de la generación de código objeto. Como paso final, es importante que el código resultante sea eficiente y pueda correr directamente sobre la computadora en lenguaje ensamblador o basándose en microinstrucciones.

3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Evento
<p>Instituto Tecnológico de Saltillo del 5 al 9 de octubre de 2009.</p>	<p>Representantes de los Institutos Tecnológicos de:</p> <p>Alvarado, Arandas, Campeche, Celaya, Centla, Cerro Azul, Champotón, Ciudad Acuña, Ciudad Cuauhtémoc, Ciudad Juárez, Ciudad Madero, Ciudad Valles, Coatzacoalcos, Cocula, Colima, Comitán, Durango, El Istmo, Huetamo, La Laguna, La Paz, Lázaro Cárdenas, Lerdo, Libres, Linares, Macuspana, Matamoros, Mérida, Mexicali, Morelia, Nuevo Laredo, Nuevo León, Occidente del Estado de Hidalgo, Ocotlán, Orizaba, Oriente del Estado de Hidalgo, Parral, Piedras Negras, Pinotepa, Saltillo, San Luis Potosí, Sur de Guanajuato, Sur del Estado de Yucatán, Tapachula, Tepexi de Rodríguez, Teziutlán, Tijuana, Toluca, Tuxtepec, Veracruz, Villahermosa, Xalapa, Zacatecas y Zacatepec.</p>	<p>Reunión Nacional de Diseño e Innovación Curricular para el Desarrollo y Formación de Competencias Profesionales de las Carreras de Ingeniería en Sistemas Computacionales, Ingeniería Informática e Ingeniería en Geociencias.</p>

<p>Instituto Tecnológico Superior de Poza Rica del 22 al 26 de febrero de 2010.</p>	<p>Representantes de los Institutos Tecnológicos de:</p> <p>Alvarado, Arandas, Campeche, Celaya, Centla, Cerro Azul, Champotón, Ciudad Acuña, Ciudad Cuauhtémoc, Ciudad Juárez, Ciudad Madero, Ciudad Valles, Coatzacoalcos, Cocula, Colima, Comitán, Durango, El Istmo, Huetamo, La Laguna, La Paz, Lázaro Cárdenas, Lerdo, Libres, Macuspana, Matamoros, Mérida, Mexicali, Morelia, Nuevo Laredo, Nuevo León, Occidente del Estado de Hidalgo, Orizaba, Oriente del Estado de Hidalgo, Parral, Piedras Negras, Pinotepa, Saltillo, San Luis Potosí, Sur de Guanajuato, Sur del Estado de Yucatán, Tapachula, Tepexi de Rodríguez, Teziutlán, Tijuana, Toluca, Tuxtepec, Veracruz, Villahermosa, Xalapa, Zacatecas y Zacatepec.</p>	<p>Reunión Nacional de Consolidación de los Programas en Competencias Profesionales de las Carreras de Ingeniería en Sistemas Computacionales, Ingeniería Informática e Ingeniería Petrolera del SNEST.</p>
<p>Instituto Tecnológico de Querétaro del 22 al 25 de octubre de 2012.</p>	<p>Representantes de los Institutos Tecnológicos de:</p> <p>Acayucan, Altamira, Cajeme, Campeche, Cananea, Cd. Acuña, Cd. Cuauhtémoc, Cd. Juárez, Cd. Madero, Cd. Valles, Celaya, Centla, Cerro Azul, Chetumal, Chihuahua II, Chilpancingo, Coalcomán, Coatzacoalcos, Cocula, Colima, Comalcalco, Delicias, Durango, Ébano, Escárcega, Huixquilucan, La Paz, León, Lerdo, Los Ríos, Macuspana, Mante, Milpa Alta, Minatitlán, Morelia, Nuevo Laredo, Nuevo León, Oaxaca, Oriente del Estado de México, Oriente del Estado de Hidalgo, Pachuca, Piedras Negras, Progreso, Puerto Vallarta, Purhepecha, Tacámbaro, Tehuacán,</p>	<p>Reunión Nacional de Seguimiento Curricular de los Programas en Competencias Profesionales de las Carreras de Ingeniería en Sistemas Computacionales, Ingeniería Informática e Ingeniería en Tecnologías de la Información y Comunicaciones.</p>

	Tepexi de Rodríguez, Tepic, Teposcolula, Teziutlán, Tierra Blanca, Tijuana, Tlaxiaco, Toluca, Tuxtepec, Uruapan, Valladolid, Veracruz, Villahermosa, Zacatecas, Zacatecas Norte, Zacatepec, Zapopan, Zitácuaro y Zongólica.	
Instituto Tecnológico de Toluca, del 10 al 13 de febrero de 2014.	Representantes de los Institutos Tecnológicos de: Cerro Azul, Colima, Lerdo, Toluca y Veracruz.	Reunión de Seguimiento Curricular de los Programas Educativos de Ingenierías, Licenciaturas y Asignaturas Comunes del SNIT.

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura
Implementa un compilador para un lenguaje específico considerando las etapas del mismo.

5. Competencias previas

Define, diseña y programa las fases del analizador léxico y sintáctico de un traductor o compilador para preámbulo de la construcción de un compilador.

6. Temario

No.	Temas	Subtemas
1	Análisis semántico.	1.1 Árboles de expresiones. 1.2 Acciones semánticas de un analizador sintáctico. 1.3 Comprobaciones de tipos en expresiones. 1.4 Pila semántica en un analizador sintáctico. 1.5 Esquema de traducción. 1.6 Generación de la tabla de símbolo y tabla de direcciones. 1.7 Manejo de errores semánticos.
2	Generación de código intermedio.	2.1 Notaciones. 2.1.1 Prefija.

		<ul style="list-style-type: none"> 2.1.2 Infija. 2.2.3 Postfija. 2.2 Representaciones de código. Intermedio. <ul style="list-style-type: none"> 2.2.1 Notación Polaca. 2.2.2 Código P. 2.2.3 Triplos. 2.2.4 Cuádruplos. 2.3 Esquema de generación. <ul style="list-style-type: none"> 2.3.1 Variables y constantes. 2.3.2 Expresiones. 2.3.3 Instrucción de asignación. 2.3.4 Instrucciones de control. 2.3.5 Funciones. 2.3.6 Estructuras.
3	Optimización.	<ul style="list-style-type: none"> 3.1 Tipos de optimización. <ul style="list-style-type: none"> 3.1.1 Locales. 3.1.2 Ciclos. 3.1.3 Globales. 3.1.4 De mirilla. 3.2 Costos. <ul style="list-style-type: none"> 3.2.1 Costo de ejecución. (memoria, registros, pilas). 3.2.2 Criterios para mejorar el código. 3.2.3 Herramientas para el análisis del flujo de datos.
	Generación de código objeto.	<ul style="list-style-type: none"> 4.1 Registros. 4.2 Lenguaje ensamblador. 4.3 Lenguaje máquina. 4.4 Administración de memoria.

7. Actividades de aprendizaje de los temas

1. Análisis semántico.	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <ul style="list-style-type: none"> • Diseña mediante el uso de reglas semánticas dirigidas por sintaxis, un analizador semántico para un compilador. <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Solución de problemas. • Toma de decisiones. • Trabajo en equipo. • Capacidad de aplicar los conocimientos. • Habilidades de investigación. • Capacidad de generar nuevas ideas. • Liderazgo. • Habilidad para trabajar en forma Autónoma. • Búsqueda del logro. 	<ul style="list-style-type: none"> • Detectar errores semánticos. • Diseñar y seleccionar información sobre la construcción de un analizador semántico. • Reconocer el manejo de tipos en las expresiones y el uso de operadores. • Establecer las reglas para la conversión de tipos (casting) en expresiones. • Agregar acciones semánticas a la estructura de la gramática. • Manipular la tabla de conversión de símbolos y de errores y direcciones. • Integrar equipos de trabajo para la construcción de un analizador semántico.
2. Generación de código intermedio.	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <ul style="list-style-type: none"> • Diseña las reglas para traducir el código fuente a un código intermedio. <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Habilidad para buscar y analizar información proveniente de fuentes diversas. 	<ul style="list-style-type: none"> • Aplicar los tipos de notación para la conversión de expresiones: Infija, prefija y posfija. • Representar expresiones mediante el código intermedio. • Reconocer el manejo de tipos en las expresiones y el uso de operadores. • Desarrollar las acciones que representen la estructura de un lenguaje de programación de alto nivel en un código intermedio.

<ul style="list-style-type: none"> • Solución de problemas. • Toma de decisiones. • Trabajo en equipo. • Capacidad de aplicar los conocimientos. • Habilidades de investigación. • Capacidad de generar nuevas ideas. • Liderazgo. • Habilidad para trabajar en forma. Autónoma. • Búsqueda del logro. 	<ul style="list-style-type: none"> • Aplicar las acciones construidas a la gramática del lenguaje prototipo. • Integrar equipos de trabajo para la generación de un código intermedio.
<p>3. Optimización.</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica(s):</p> <ul style="list-style-type: none"> • Conoce e identifica los diferentes tipos de optimización que permita eficientar el código intermedio. <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de organizar y planificar. • Habilidad para buscar y analizar información proveniente de fuentes diversas. • Solución de problemas. • Toma de decisiones. • Trabajo en equipo. • Capacidad de aplicar los conocimientos. • Habilidades de investigación. • Capacidad de generar nuevas ideas. • Liderazgo. • Habilidad para trabajar en forma. Autónoma. • Búsqueda del logro. 	<ul style="list-style-type: none"> • Aplicar las técnicas para la optimización del código intermedio generado. • Tener nociones algebraicas para estimar el número de veces que se realiza una instrucción dentro de un ciclo o ciclos anidados. • Conocer que recursos se consumen en invocación a funciones y expresiones simples. • Estudiar nuevas técnicas para la optimización de código, sobre todo para aquellos lenguajes que requieren de una máquina virtual para su ejecución sobre multiplataformas. • Escribir un ensayo que establezca las tendencias y técnicas empleadas para este propósito. • Conocer los criterios de tiempo de ejecución o extensión de código generado. • Integrar equipos, para analizar códigos intermedios existentes y proponer algunas mejoras.
<p>4. Generación de código objeto.</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>

<p>Específica(s):</p> <ul style="list-style-type: none">• Utiliza un lenguaje de bajo nivel para traducir el código construido a lenguaje máquina para su ejecución. <p>Genéricas:</p> <ul style="list-style-type: none">• Capacidad de análisis y síntesis.• Capacidad de organizar y planificar.• Habilidad para buscar y analizar información proveniente de fuentes diversas.• Solución de problemas.• Toma de decisiones.• Trabajo en equipo.• Capacidad de aplicar los conocimientos.• Habilidades de investigación.• Capacidad de generar nuevas ideas.• Liderazgo.• Habilidad para trabajar en forma. Autónoma.• Búsqueda del logro.	<ul style="list-style-type: none">• Conocer la arquitectura de los microprocesadores intel y compatibles.• Conocer la estructura y funcionamiento del lenguaje ensamblador.• Conocer las características principales del lenguaje maquina a fin de llevar un código intermedio y este pueda ser reconocido por el hardware.• Conocer las técnicas de administración de memoria para el almacenamiento de un programa en momento de ejecución.• Experimentar con simuladores de arquitectura de microprocesadores.
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

8. Práctica(s)

- Diseñar y construir el generador de código semántico para el lenguaje del caso de estudio.
- Realizar arboles de expresiones en casos de estudio.
- Realizar conversiones de tipos en expresiones.
- Construir la tabla de símbolos y de direcciones para la gramática propuesta.
- Detectar errores de semántica en expresiones dadas.
- Modificar la GLC agregando las acciones semánticas correspondientes.
- Convertir expresiones mediante el uso de notaciones prefijas, infijas y postfijas.
- Definir e implementar la notación que más se ajuste a las estructuras de evaluación de expresiones de lenguaje.
- Proponer una estructura de código intermedio en base a las características propias de cada lenguaje.
- Desarrollar esquemas de generación de código intermedio.
- Definir y construir el generador de código intermedio para su caso de estudio.
- Agregar acciones de representación intermedia al lenguaje de programación propuesto.
- Saber cuántos recursos y cuánto tiempo consume cada instrucción de código intermedio.
- Evaluar el código intermedio generado para los programas escritos en el lenguaje de su caso de estudio y si aplica realizar la optimización correspondiente.
- Poder establecer una equivalencia entre las instrucciones del lenguaje intermedio y las instrucciones en ensamblador.
- Diseñar y construir el generador de código máquina u objeto para el lenguaje del caso de estudio.

9. Proyecto de asignatura

El objetivo del proyecto que planteé el docente que imparta esta asignatura, es demostrar el desarrollo y alcance de la(s) competencia(s) de la asignatura, considerando las siguientes fases:

- **Fundamentación:** marco referencial (teórico, conceptual, contextual, legal) en el cual se fundamenta el proyecto de acuerdo con un diagnóstico realizado, mismo que permite a los estudiantes lograr la comprensión de la realidad o situación objeto de estudio para definir un proceso de intervención o hacer el diseño de un modelo.
- **Planeación:** con base en el diagnóstico en esta fase se realiza el diseño del proyecto por parte de los estudiantes con asesoría del docente; implica planificar un proceso: de intervención empresarial, social o comunitario, el diseño de un modelo, entre otros, según el tipo de proyecto, las actividades a realizar los recursos requeridos y el cronograma de trabajo.
- **Ejecución:** consiste en el desarrollo de la planeación del proyecto realizada por parte de los estudiantes con asesoría del docente, es decir en la intervención (social, empresarial), o construcción del modelo propuesto según el tipo de proyecto, es la fase de mayor duración que implica el desempeño de las competencias genéricas y específicas a desarrollar.
- **Evaluación:** es la fase final que aplica un juicio de valor en el contexto laboral-profesión, social e investigativo, ésta se debe realizar a través del reconocimiento de logros y aspectos a mejorar se estará promoviendo el concepto de “evaluación para la mejora continua”, la metacognición, el desarrollo del pensamiento crítico y reflexivo en los estudiantes.

10. Evaluación por competencias

Para evaluar las actividades de aprendizaje se recomienda solicitar: mapas conceptuales, reportes de prácticas, estudios de casos, exposiciones en clase, ensayos, problemarios, reportes de visitas, portafolio de evidencias y cuestionarios, cuadro sinóptico.

Para verificar el nivel del logro de las competencias del estudiante se recomienda utilizar: listas de cotejo, listas de verificación, matrices de valoración, guías de observación, coevaluación y autoevaluación.

11. Fuentes de información

Impresas:

1. Aho Alfred V., U. J. (2007). Compiladores. Principios, técnicas y herramientas (2da. ed.). México: Pearson Educación.
2. Alfonseca Moreno, M. (2006). Compiladores e intérpretes: teoría y práctica (1ra ed.). España: Pearson/Prentice Hall.
3. Carrión Viramontes, J. E. (2008). Teoría de la computación. México: Limusa.
4. Hopcroft John E., M. R. (2002). Introducción a la Teoría de Autómatas, Lenguajes y Computación (2da. ed.). Madrid: Addison-Wesley.
5. Isasi Pedro, M. P. (1997). Lenguajes, gramáticas y autómatas. Un enfoque Práctico. Addison-Wesley.
6. Kelley, D. (1995). Teoría de Autómatas y Lenguajes Formales, (1ra. ed.). Madrid: Prentice Hall.
7. Lemone, K. A. (1996). Fundamentos de compiladores: cómo traducir al lenguaje de computadora. México D.F.: Compañía Editorial Continental.
8. Martin, J. (2004). Lenguajes formales y teoría de la computación. México: McGraw-Hill / Interamericana de México.
9. Ruíz, J. (2009). Compiladores-Teoría e implementación. México: Alfaomega.
10. Grune, Dick. (2007). Diseño de compiladores modernos. McGraw-Hill.

Electrónicas:

11. Sacristán Donoso, Juan Marcos. Desarrollo de compiladores. Obtenido de <http://megazar.tripod.com/compil.pdf>
12. COFETEL (Comisión Federal de Telecomunicaciones). (2014). Industria. Obtenido de <http://www.cft.gob.mx:8080/portal/industria-2/industria-intermedia-nv/>
13. Corning Incorporated. (2014). Corning Telecommunications. Obtenido de http://www.corning.com/products_services/telecommunications/index.aspx

14. Corning Incorporated. (2014). CorningIncorporated. Obtenido de <http://www.youtube.com/user/CorningIncorporated>
 15. IEEE. (2014). IEEE Standards Association. Obtenido de <http://www.youtube.com/user/IEEESA>
 16. IEEE. (2014). Technology Standards & Resources. Obtenido de <http://standards.ieee.org/findstds/index.html>
 17. Panduit Corp. (2014). Panduit videos. Obtenido de <http://www.youtube.com/user/PanduitVideos>
 18. Panduit Corp. (2014). Panduit. Obtenido de http://www.panduit.com/wcs/Satellite?pagename=PG_Wrapper&friendlyurl=/es/home
 19. TED. (2014). TED Topics Internet. Obtenido de <http://www.ted.cnom/topics/Internet>
 20. The Siemon Company. (2014). Siemon Company Videos. Obtenido de <http://www.youtube.com/user/SiemonNetworkCabling>
 21. The Siemon Company. (2014). Siemon Network Cabling Solutions. Obtenido de <http://www.siemon.com/la/>
- <http://megazar.tripod.com/compil.pdf>
<http://www.youtube.com/watch?v=YyAhap8HJmc>
<http://www.youtube.com/watch?v=tV-lgtT2IjY>