

1. Datos Generales de la asignatura

Nombre de la asignatura:	Lenguajes y Autómatas I.
Clave de la asignatura:	SCD - 1015
SATCA¹:	2 - 3 - 5
Carrera:	Ingeniería en Sistemas Computacionales.

2. Presentación

Caracterización de la asignatura

El desarrollo de sistemas basados en computadora y la búsqueda de soluciones para problemas de procesamiento de información son la base tecnológica de la carrera de Ingeniería en Sistemas.

Todo egresado de esta ingeniería debe poseer los conocimientos necesarios para resolver de manera óptima cualquier problema relacionado con procesamiento de información. El conocimiento de las características, fortalezas y debilidades de los lenguajes de programación y su entorno le permitirán proponer las mejores soluciones en problemas de índole profesional y dentro de las realidades de su entorno.

Como parte integral de la asignatura, se debe promover el desarrollo de las habilidades necesarias para que el estudiante implemente sistemas sujetándose en los estándares de desarrollo de software, esto con el fin de incentivar la productividad y competitividad de las empresas donde se desarrollen. Sin duda alguna, los problemas que se abordarán requerirán la colaboración entre grupos interdisciplinarios, por ello el trabajo en grupos es indispensable. Debe quedar claro que los proyectos que serán desarrollados son de diversas áreas y complejidades, y en ocasiones requieren la integración de equipos externos. Esta complejidad debe considerarse una oportunidad para experimentar con el diseño de interfaces hombre-máquina y máquina-máquina.

Como todos sabemos, un mismo problema puede ser resuelto computacionalmente de diversas formas. Una de las condiciones a priori de la asignatura, es el conocimiento de las arquitecturas de computadoras (microprocesadores) y de las restricciones de desempeño que deben considerarse para la ejecución de aplicaciones. Esto aportará los conocimientos que le permitirán al estudiante desarrollar aplicaciones eficientes en el uso de recursos. De manera adicional, es posible que se integren dispositivos externos dentro de las soluciones. En este aspecto, el papel del profesor como guía es fundamental. Es importante diversificar la arquitectura de las soluciones planteadas. Si la inclusión de algún componente de hardware facilita la solución, se recomienda que sea incluido.

Esta área, por sus características conceptuales, se presta para la investigación de campo. Los estudiantes tendrán la posibilidad de buscar proyectos que les permitan aplicar los conocimientos adquiridos durante las sesiones del curso. El desarrollo de este proyecto es una oportunidad excelente para aplicar todos los conceptos, técnicas y herramientas orientadas al modelado. La formalidad con que se traten estos aspectos dotará al estudiante de nuevos conceptos, procedimientos y experiencia.

¹ Sistema de Asignación y Transferencia de Créditos Académicos

En esta asignatura se abordan todos los temas relacionados con teoría de lenguajes formales, algo que permite vislumbrar los procesos inherentes, y a veces, escondidos dentro de todo lenguaje. Las formas de representación formal, procesamiento e implementación de lenguajes de programación se atacan desde un punto de vista de implementación. Los proyectos relacionados y los ejercicios de investigación acercan a los estudiantes al campo de lenguajes formales, base de los procesos de comunicación. Por último se revisan algunos de los puntos eje de la investigación de frontera que aún contienen problemas abiertos, un incentivo para la incorporación de estudiantes a las áreas de investigación.

Las asignaturas directamente vinculadas son estructura de datos por las herramientas para el procesamiento de información que proporciona (árboles binarios, pilas, colas, tablas de Hash), todas aquellas que incluyan lenguajes de programación, porque son las herramientas para el desarrollo de cualquiera de las prácticas dentro de la asignatura y permitirán un enfoque práctico para todos los temas de la misma. La materia de arquitectura de computadoras dota al estudiante de los conocimientos sobre la estructura de registros, modos de direccionamiento, conjunto de operadores, y le da al estudiante una visión sobre cómo mejorar el desempeño de lenguajes.

Esta materia sirve de preámbulo para la asignatura de lenguajes y autómatas II, en la cual se completa el estudio formal de la teoría de lenguajes.

A su vez permitirá el desarrollo de las siguientes competencias específicas:

- 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.

Intención didáctica

Esta asignatura es de vital importancia para toda la carrera, como es una asignatura sobre lenguajes formales, el enfoque debe coincidir con la formalidad de los mismos. Cada tema debe ser acompañado de una serie de ejercicios y prácticas que permitan redondear los temas revisados en clase. Esta asignatura se presta para la participación activa de los estudiantes en la discusión de los temas y ejemplificación de casos. También permite que el estudiante se acerque al análisis de problemas del área industrial, como diseño, manufactura, tratamiento de lenguaje natural, robótica, inteligencia artificial, procesamiento de consultas en base de datos, procesamiento de consultas en Web, análisis y diseño de algoritmos, entre otros.

En este sentido, el profesor debe guiar, comentar, corregir o completar las investigaciones que el estudiante realice. Estas investigaciones deben buscar como objetivo el desarrollo de la creatividad y la integración del estudiante dentro del grupo. La creatividad permitirá vislumbrar las fronteras dentro de este campo.

Como puede apreciarse, las competencias generales que pueden estimularse son, entre otras:

- Capacidad de discernir los aspectos relevantes de investigaciones documentales.
- Comunicación oral y escrita para presentar resultados de investigación documental.
- Análisis y síntesis de problemas de procesamiento de información.
- Integración de grupos de trabajo, a veces multidisciplinarios.

- Solución de problemas a planteamientos específicos.
- Toma de decisiones para determinar la mejor forma de resolver un problema.
- Uso de Estándares de desarrollo para la implementación de soluciones.

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,</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>

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

5. Competencias previas

Conoce, comprende y aplica las estructuras de datos, métodos de ordenamiento y búsqueda para la optimización del rendimiento de soluciones de problemas del contexto.

6. Temario

No.	Temas	Subtemas
1	Introducción a la Teoría de Lenguajes Formales.	1.1 Alfabeto. 1.2 Cadenas. 1.3 Lenguajes, tipos y herramientas. 1.4 Estructura de un traductor 1.5 Fases de un compilador
2	Expresiones Regulares.	2.1. Definición formal de una ER. 2.2. Diseño de ER. 2.3. Aplicaciones en problemas reales.
3	Autómatas Finitos.	3.1 Conceptos: Definición y Clasificación de Autómata Finito (AF). 3.2 Conversión de un Autómata Finito No Determinista (AFND) a Autómata Finito Determinista (AFD). 3.3 Representación de ER usando AFND 3.4 Minimización de estados en un AF 3.5 Aplicaciones (definición de un caso de estudio).
4	Análisis Léxico.	4.1 Funciones del analizador léxico. 4.2 Componentes léxicos, patrones y lexemas. 4.3 Creación de Tabla de tokens.

		4.4 Errores léxicos. 4.5 Generadores de analizadores Léxicos. 4.6 Aplicaciones (Caso de estudio).
5	Análisis Sintáctico.	5.1 Definición y clasificación de gramáticas. 5.2 Gramáticas Libres de Contexto (GLC). 5.3 Árboles de derivación. 5.4 Formas normales de Chomsky. 5.5 Diagramas de sintaxis 5.6 Eliminación de la ambigüedad. 5.7 Tipos de analizadores sintácticos 5.8 Generación de matriz predictiva (cálculo first y follow) 5.9 Manejo de errores 5.10 Generadores de analizadores sintácticos
6	Máquinas de Turing.	6.1 Definición formal MT 6.2 Construcción modular de una MT 6.3 Lenguajes aceptados por la MT.

7. Actividades de aprendizaje de los temas

1. Introducción a la Teoría de Lenguajes Formales.	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <ul style="list-style-type: none"> Identifica los conceptos de lenguajes formales para comprender las fases de un compilador y traductor. <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. 	<ul style="list-style-type: none"> Definir alfabetos y lenguajes en un caso de estudio. Obtener un alfabeto a partir de un lenguaje. Investigar la estructura de diferentes traductores. Estructurar mediante un diagrama, las fases de un compilador.

<ul style="list-style-type: none"> • Liderazgo. • Habilidad para trabajar en forma. Autónoma. • Búsqueda del logro. 	
<p>2. Expresiones Regulares.</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica(s):</p> <ul style="list-style-type: none"> • Crea y reconoce Expresiones Regulares para solucionar problemas del entorno. <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"> • Investigar las expresiones regulares y sus operaciones. • Generar cadenas a partir de una expresión regular. • Obtener una expresión regular a partir de un grupo de cadenas o viceversa.
<p>3. Autómatas Finitos.</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica(s):</p> <ul style="list-style-type: none"> • Crea y reconoce autómatas finitos en un lenguaje de programación para la solución de un problema. <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. 	<ul style="list-style-type: none"> • Determinar la notación formal de un autómata finito. • Conocer la diferencia entre un AFND y AFD. • Construir un AF a partir de un ER. • Construir un AF a partir de la descripción de un caso de estudio (en grupos de trabajo). • Convertir un AFN a AFD.

<ul style="list-style-type: none"> • 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"> • Minimizar estados en un AF.
<p>4. Análisis Léxico.</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica(s):</p> <ul style="list-style-type: none"> • Construye un analizador léxico a partir de un lenguaje de programación. <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"> • Elaborar por equipo, la identificación de lexemas, componentes léxicos y patrones a partir de un lenguaje. • Conocer los elementos de una tabla de tokens. • Definir las reglas de un lenguaje de programación propio. • Identificar patrones válidos, generar autómatas y tabla de tokens del lenguaje propuesto. • Distinguir los Errores léxicos. • Construir un analizador léxico mediante un lenguaje de programación. (Utilizar un generador de analizador léxico como ejemplo).
<p>5. Análisis Sintáctico.</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>

<p>Específica(s):</p> <ul style="list-style-type: none"> • Construye un analizador sintáctico a partir de un lenguaje de programación. <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"> • Identificar la notación formal de una gramática. • Buscar la sintaxis de la construcción de los Lenguajes de Programación por medio de GLC o utilizando notación BNF (Backus-Naur Form). • Investigar las formas normales de Chomsky. • Conocer la notación de los diagramas de sintaxis. • Construir diagramas de sintaxis de un lenguaje. • Construir una GLC a partir de los diagramas de sintaxis. • Eliminar la ambigüedad de una gramática.
<p>6. Máquinas de Turing.</p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica(s):</p> <ul style="list-style-type: none"> • Diseña y construye o simula una Máquina de Turing (MT), para el reconocimiento de cadenas propias de lenguajes. <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. 	<ul style="list-style-type: none"> • Identificar la notación formal de una MT. • Construir una MT a partir de un caso de estudio. • Simular a través de un lenguaje de alto nivel, la representación de una MT

<ul style="list-style-type: none"> • Capacidad de generar nuevas ideas. • Liderazgo. • Habilidad para trabajar en forma Autónoma. • Búsqueda del logro. 	
---	--

8. Práctica(s)

<ul style="list-style-type: none"> • Realizar un cuadro comparativo de los traductores y compiladores que incluya ventajas, desventajas y características. • Clasificar una lista de lenguajes, ambientes de desarrollo y utilerías en herramientas computacionales. • Clasificar los componentes léxicos de un programa fuente. • Obtener un alfabeto a partir de un lenguaje y viceversa. • Relacionar los componentes léxicos con una expresión regular. • Obtener expresiones regulares a partir de casos de estudio. • Realizar un programa que implemente una expresión regular. • Realizar programas que implemente lenguajes simples representados con AFD's. • Realizar ejercicios de construcción de AF a partir de ER o casos de estudio. • Realizar conversiones de AFN a AFD. • Definir las reglas de un lenguaje de programación propio. • Generar el autómata correspondiente al lenguaje definido. • Analizar la funcionalidad de diferentes generadores para análisis léxico de compilador. • Realizar prácticas en algún generador para analizadores léxico. • Construir un analizador léxico (utilizar un generador de analizador como ejemplo). • Construir diagramas de sintaxis para el lenguaje propuesto. • Construir una GLC para el lenguaje propuesto. • Analizar la funcionalidad de diferentes generadores para análisis sintáctico. • Realizar prácticas en algún generador para analizadores sintáctico. • Construir un analizador sintáctico (utilizar un generador de analizador sintáctico o un lenguaje de programación) • Construir MT a partir de casos de estudio. • Simular a través de un lenguaje de alto nivel, la representación de una máquina de Turing.

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 Automatas, 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 Automatas 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. Garbusi Pablo. *Diseño de compiladores*. Obtenido de http://www.fing.edu.uy/inco/cursos/compil/teoricos/01_Introduccion.pdf
12. Ortiz Triviño, Jorge Eduardo. *Lenguajes Regulares*. Obtenido de <http://www.youtube.com/watch?v=2caZNHXsj88>
13. Cubur, Alex. *Expresion Regular a DFA en JFlap*. Obtenido de http://www.youtube.com/watch?v=S6y0Wu_qp6I

<http://www.youtube.com/watch?v=w-KfjJdRas8>