1. Datos Generales de la asignatura

Nombre de la asignatura:	Lenguajes y autómatas
Clave de la asignatura:	CDD-2415
SATCA[1]:	2-3-5
Carrera:	Ingeniería en ciencia de datos

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 ciencia de datos.

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 a la asignatura de lenguajes y autómatas 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.

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.

Lugar y fecha de elaboración o revisión	Participantes	Observaciones
Instituto Tecnológico Superior de Alvarado del 21 al 23 agosto de 2023.	Representante del Instituto Tecnológico Superior de Alvarado.	Propuesta inicial.
Tecnológico Nacional de México 30 octubre 2023	Representante del Instituto Tecnológico de: Querétaro y del Instituto Tecnológico Superior de Alvarado.	Presentación de la propuesta de la carrera de Ingeniería en Ciencia de Datos.
Instituto Tecnológico de Querétaro Campus Norte del 19 al 22 de marzo 2024.	Representantes de los Institutos Tecnológicos de: Morelia, Puebla, Querétaro, Tehuacán. Instituto Tecnológico Superior de Alvarado. CENIDET. Representante de Ciencias Básica de los Institutos de: Celaya, Morelia y CIIDET.	Diseño y/o desarrollo curricular de la carrera de Ingeniería en Ciencia de Datos.
Tecnológico Nacional de México del 22 al 24 de abril del 2024	Representante del Instituto Tecnológico de Querétaro e Instituto Tecnológico Superior de Alvarado.	Contraste y ajuste de las asignaturas de Ingeniería en Ciencia de Datos con respecto a las de Ing. en Inteligencia Artificial, Ing. en Desarrollo WEB e Ing. en Ciberseguridad
Tecnológico Nacional de México del 27 al 31 de mayo del 2024.	Representantes de los Institutos Tecnológicos de: Morelia, Querétaro. Instituto Tecnológico Superior de Alvarado. CENIDET.	Consolidación curricular de la carrera de Ingeniería Ciencia de Datos

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura
Diseña y programa las fases del analizador léxico, sintáctico y semántico de un traductor o compilador para la construcción de un compilador.

5. Competencias previas

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	Lenguajes regulares.	2.1. Definición formal de una ER. 2.2. Diseño de ER. 2.3. Definición y clasificación de autómata finito (AF). 2.4. Conversión de un autómata finito no determinista (AFND) a autómata finito determinista (AFD). 2.5. Representación de ER usando AFND 2.6. Minimización de estados en un AF.
3	Lenguajes libres de contexto.	3.1. Gramáticas libres de contexto. 3.2. Árboles de derivación. 3.3. Formas normales de Chomsky. 3.4. Eliminación de la ambigüedad. 3.5. Autómatas Push-Down.
4	Análisis léxico.	4.1. Funciones del analizador léxico. 4.2. Componentes léxicos, patrones y lexemas. 4.3. Creación de tabla de tokens y errores léxicos. 4.4. Generadores de analizadores Léxicos. 4.5. Caso práctico de análisis léxico.
5	Análisis sintáctico.	5.1. Diagramas de sintaxis. 5.2. Tipos de analizadores sintácticos. 5.3. Generación de matriz predictiva (cálculo first y follow). 5.4. Manejo de errores sintácticos. 5.5. Generadores de analizadores sintácticos. 5.6. Caso Práctico de análisis sintáctico.
6	Análisis semántico.	6.1. Árboles de expresiones. 6.2. Acciones semánticas y pila semántica de un analizador sintáctico. 6.3. Comprobaciones de tipos en expresiones. 6.4. Esquema de traducción. 6.5. Generación de la tabla de símbolo y tabla de direcciones y manejo de errores semánticos. 6.6. Caso práctico de análisis semántico.

7. Actividades de aprendizaje de los temas

1. Introducción a la teoría de lenguajes formales.	
Competencias	Actividades de aprendizaje
Específica(s): Identifica los conceptos de lenguajes formales para comprender las fases de un compilador y traductor. *Genérica(s):* ● 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.	● 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.
2. Lenguajes regulares	
Competencias	Actividades de aprendizaje
Específica(s): Crea y reconoce lenguajes regulares (expresiones regulares y/o autómatas finitos) en un lenguaje de programación para la solución de un problema. *Genéricas:* ● 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.	● 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. ● 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. Minimizar estados en un AF.

• Liderazgo. • Habilidad para trabajar en forma. • Autónoma. • Búsqueda del logro.	
3. Lenguajes libres de contexto	
Competencias	Actividades de aprendizaje
Específica(s): • Crea y reconoce la teoría de lenguajes de contexto libre y su representación en un lenguaje de programación. *Genérica(s):* • 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.	• Identificar los diferentes tipos de lenguajes de acuerdo a la clasificación de Chomsky. • Realizar ejercicios que permitan desarrollar la habilidad para representar lenguajes libres de contexto. • Utilizar un lenguaje de alto nivel para representar lenguajes libres de contexto, solamente como casos tipo. • 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 (BackusNaur Form). • Investigar las formas normales de Chomsky Eliminar la ambigüedad de una gramática.
4. Análisis léxico	
Competencias	Actividades de aprendizaje
Específica(s): Construye un analizador léxico a partir de un lenguaje de programación. *Genérica(s):* • 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.	• 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).

• Capacidad de generar nuevas ideas. • Liderazgo. • Habilidad para trabajar en forma. • Autónoma. • Búsqueda del logro.	
5. Análisis sintáctico	
Competencias	Actividades de aprendizaje
Específica(s): Construye un analizador sintáctico a partir de un lenguaje de programación. *Genérica(s):* • 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.	• 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. • Realizar prácticas en algún generador para analizadores sintácticos. Construir un analizador sintáctico (utilizar un generador de analizador sintáctico o un lenguaje de programación).
6. Análisis semántico	
Competencias	Actividades de aprendizaje
Específica(s): Diseña mediante el uso de reglas semánticas dirigidas por sintaxis, un analizador semántico para un compilador. *Genérica(s)*: • 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.	• 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.

• Habilidades de investigación. • Capacidad de generar nuevas ideas. • Liderazgo. • Habilidad para trabajar en forma. • Autónoma. • Búsqueda del logro.	

8. Práctica(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 sintácticos.
- Construir un analizador sintáctico (utilizar un generador de analizador sintáctico o un lenguaje de programación)
- Diseñar y construir el generador de código semántico para el lenguaje del caso de estudio
- Detectar errores de semántica en expresiones dadas
- Modificar la GLC agregando las acciones semánticas correspondientes

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

La evaluación debe ser continua y formativa por lo que se debe considerar el desempeño en cada una de las actividades de aprendizaje, haciendo especial énfasis en:

- Reportes escritos de las prácticas realizadas durante clase y las actividades inherentes, así como de las conclusiones obtenidas.
- Análisis de la información obtenida durante las investigaciones solicitadas plasmada en documentos escritos.
- Descripción de otras experiencias concretas que podrían realizarse adicionalmente.
- Exámenes escritos para comprobar el manejo de aspectos teóricos y declarativos.

11. Fuentes de información

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. AddisonWesley. Kelley, D. (1995). Teoría de Autómatas y Lenguajes Formales, (1ra. ed.). Madrid: Prentice Hall.
6. Lemone, K. A. (1996). Fundamentos de compiladores: cómo traducir al lenguaje de computadora. México D.F.: Compañía Editorial Continental.
7. Martin, J. (2004). Lenguajes formales y teoría de la computación. México: McGraw-Hill / Interamericana de México.
8. Ruíz, J. (2009). Compiladores-Teoría e implementación. México: Alfaomega.
9. Grune, Dick. (2007). Diseño de compiladores modernos. McGraw-Hill.

Electrónicas:

Garbusi Pablo. Diseño de compiladores. Obtenido de http://www.fing.edu.uy/inco/cursos/compil/teoricos/01_Introduccion.pdf

Ortiz Triviño, Jorge Eduardo. Lenguajes Regulares. Obtenido de http://www.youtube.com/watch?v=2caZNHXsj88 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