Programa Educativo:	Ingeniería en Sistemas Computacionales.		
Periodo Escolar:	Enero-Junio 2018	Clave de la Asignatura:	SCD-1015
Nombre de la Asignatura:	Lenguajes y Autómatas I	Clave del grupo:	S6A
Horas teoría-horas práctica-créditos	2-3-5	Número de unidades	6

1. 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.
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:

Evaluación de lenguajes de programación: evaluar un conjunto de lenguajes de programación con base en un problema a resolver y elegir el mejor de ellos para el problema en particular.

Análisis y síntesis para la solución de un problema: dado un problema, proponer el mejor lenguaje que se ajusta a las especificaciones del mismo. Si no hay lenguaje disponible, proponer las características del lenguaje ideal para el problema a resolver.

2. Objetivo(s) general(es) del curso. (Competencias específicas a desarrollar)

Definir, diseñar, construir y programar las fases del analizador léxico y sintáctico de un traductor o compilador.

3. Análisis por unidades

No. de la unidad:	1	**Tema de la unidad:**	Introducción a la Teoría de Lenguajes Formales	**Horas teórico-prácticas**	5

Competencia específica de la unidad	**Desarrollo de competencias genéricas**
Expresar la notación matemática de un lenguaje formal. Identificar las fases de un compilador. Relacionar los componentes léxicos con el alfabeto.	• Capacidad de análisis y síntesis Capacidad de organizar y planificar • Comunicación oral y escrita • Habilidad para buscar y analizar información proveniente de fuentes diversas • Solución de problemas • Toma de decisiones. • Capacidad crítica y autocrítica • Trabajo en equipo • Capacidad de aplicar los conocimientos en la práctica • Habilidades de investigación • Capacidad de aprender • Habilidad para trabajar en forma autónoma

Subtemas:	Actividades de enseñanza	Actividades de aprendizaje	Criterios de evaluación
1.1 Alfabeto. 1.2 Cadenas. 1.3 Lenguajes 1.4 Tipos de lenguajes 1.5 Herramientas computacionales ligadas con lenguajes 1.6 Estructura de un traductor 1.7 Fases de un compilador	Aplicar una evaluación diagnóstica. Discutir con el grupo el objetivo del curso. Fomenta la lectura sobre el tema, el trabajo en equipo y participación en clase. Induce la participación y discusión de los temas de la unidad Exponer sobre conceptos de la unidad.	Asistir al 100% de las clases. Sintetizar la información expuesta. Desarrollo de glosario a partir de archivos indicados por el profesor. Construir su propio conocimiento	**Evidencias cognoscitivas** Glosario 5% **Evidencias procedimentales** **Evidencias actitudinales**

Instrumentos de evaluación (Rubrica, lista de cotejo, etc.)

1.- GLOSARIO INDIVIDUAL **NIVELES DE DESEMPEÑO**			
Suficiente 70	**Bueno 80**	**Notable 90**	**Excelente 100**
Define el 70% de los conceptos correctamente. Incluye bibliografía	Define el 80% de los conceptos correctamente. Incluye bibliografía	Define el 90% de los conceptos correctamente. Incluye bibliografía	Define el 100% de los conceptos correctamente. Incluye bibliografía

Fuentes de información	Apoyos didácticos
1. Hopcroft John E., Introducción a la Teoría de Autómatas, Lenguajes y Computación, 2da ed, Ed. Addison Wesley, 2004. 2. Lemote Karen A. , Fundamentos de compiladores Cómo traducir al lenguaje de computadora, Ed. Compañía Editorial Continental.	Equipo de cómputo. Cañón. Pizarrón

No. de la unidad:	2	Tema de la unidad:	Expresiones regulares	Horas teórico-prácticas	15

Competencia específica de la unidad	Desarrollo de competencias genéricas
Crear y reconocer ER mediante un lenguaje de programación o un analizador léxico.	• Capacidad de análisis y síntesis Capacidad de organizar y planificar • Comunicación oral y escrita • Habilidad para buscar y analizar información proveniente de fuentes diversas • Solución de problemas • Toma de decisiones. • Capacidad crítica y autocrítica

		• Trabajo en equipo • Capacidad de aplicar los conocimientos en la práctica • Habilidades de investigación • Capacidad de aprender • Habilidad para trabajar en forma autónoma	
Subtemas:	**Actividades de enseñanza**	**Actividades de aprendizaje**	**Criterios de evaluación**
2.1. Definición formal de una ER 2.2. Operaciones 2.3. Aplicaciones en problemas reales.	Exponer los conceptos de la unidad. Resolver ejercicios en clase explicando las reglas de ER Diseño de problemario con ejercicios que permitan: Generar cadenas a partir de una expresión regular, Obtener una expresión regular a partir de un grupo de cadenas o viceversa, Obtener una expresión regular a partir de la descripción de un caso de estudio.	Asistir al 100% de las clases. Sintetizar la información expuesta. Construir su propio conocimiento Resolución de ejercicios propuestos por el profesor	**Evidencias cognoscitivas** Problemario 10% Examen 10% **Evidencias procedimentales** **Evidencias actitudinales**

Instrumentos de evaluación (Rubrica, lista de cotejo, etc.)

2.- PROBLEMARIO INDIVIDUAL
NIVELES DE DESEMPEÑO

Suficiente 70	Bueno 80	Notable 90	Excelente 100
Resuelve el 70% de los problemas del problemario correctamente	Resuelve el 80% de los problemas del problemario correctamente	Resuelve el 90% de los problemas del problemario correctamente	Resuelve el 100% de los problemas del problemario correctamente.

3.- EXAMEN INDIVIDUAL
NIVELES DE DESEMPEÑO

Suficiente 70	Bueno 80	Notable 90	Excelente 100
Resuelve el 70% de los problemas del examen correctamente	Resuelve el 80% de los problemas del examen correctamente	Resuelve el 90% de los problemas del examen correctamente	Resuelve el 100% de los problemas del examen correctamente.

Fuentes de información	Apoyos didácticos
1. Hopcroft John E., Introducción a la Teoría de Autómatas, Lenguajes y Computación, 2da ed, Ed. Addison Wesley, 2004. 2. Lemote Karen A. , Fundamentos de compiladores Cómo traducir al lenguaje de computadora, Ed. Compañía Editorial Continental.	Equipo de cómputo. Cañón. Pizarrón

No. de la unidad:	3	Tema de la unidad:	Autómatas finitos.	Horas teórico-prácticas	15

Competencia específica de la unidad	Desarrollo de competencias genéricas
Crear un AF mediante un lenguaje de programación.	• Capacidad de análisis y síntesis Capacidad de organizar y planificar • Comunicación oral y escrita • Habilidad para buscar y analizar información proveniente de fuentes diversas • Solución de problemas • Toma de decisiones. • Capacidad crítica y autocrítica • Trabajo en equipo • Capacidad de aplicar los conocimientos en la práctica • Habilidades de investigación • Capacidad de aprender • Habilidad para trabajar en forma autónoma

Subtemas:	Actividades de enseñanza	Actividades de aprendizaje	Criterios de evaluación
3.1 Definición formal 3.2 Clasificación de AF 3.3 Conversión de un AFND a AFD 3.4 Representación de ER usando AFND 3.5 Minimización de estados en un AF 3.6 Aplicaciones (definición de un caso de estudio)	Exponer los conceptos de la unidad. Resolver ejercicios en clase explicando su solución. Diseño de problemario con ejercicios que permitan: Diseño de autómatas finitos a partir del lenguaje o expresión regular identificando su tipo, convertir NFA a DFA	Asistir al 100% de las clases. Sintetizar la información expuesta. Construir su propio conocimiento. Resolución de ejercicios propuestos por el profesor	**Evidencias cognoscitivas** Problemario 5% Examen 5% **Evidencias procedimentales** **Evidencias actitudinales**

Instrumentos de evaluación (Rubrica, lista de cotejo, etc.)

4.- PROBLEMARIO INDIVIDUAL
NIVELES DE DESEMPEÑO

Suficiente 70	Bueno 80	Notable 90	Excelente 100
Resuelve el 70% de los problemas del problemario correctamente	Resuelve el 80% de los problemas del problemario correctamente	Resuelve el 90% de los problemas del problemario correctamente	Resuelve el 100% de los problemas del problemario correctamente.

5.- EXAMEN INDIVIDUAL
NIVELES DE DESEMPEÑO

Suficiente 70	Bueno 80	Notable 90	Excelente 100
Resuelve el 70% de los problemas del examen correctamente	Resuelve el 80% de los problemas del examen correctamente	Resuelve el 90% de los problemas del examen correctamente	Resuelve el 100% de los problemas del examen correctamente.

Fuentes de información	Apoyos didácticos
1. Hopcroft John E., Introducción a la Teoría de Autómatas, Lenguajes y Computación, 2da ed, Ed. Addison Wesley, 2004. 2. Lemote Karen A. , Fundamentos de compiladores Cómo traducir al lenguaje de computadora, Ed. Compañía Editorial Continental.	Equipo de cómputo. Cañón. Pizarrón

No. de la unidad:	4	**Tema de la unidad:**	Máquinas de Turing.	**Horas teórico-prácticas**	10

Competencia específica de la unidad	**Desarrollo de competencias genéricas**
Diseñar y construir o simular una MT.	• Capacidad de análisis y síntesis Capacidad de organizar y planificar • Comunicación oral y escrita • Habilidad para buscar y analizar información proveniente de fuentes diversas • Solución de problemas • Toma de decisiones. • Capacidad crítica y autocrítica • Trabajo en equipo • Capacidad de aplicar los conocimientos en la práctica • Habilidades de investigación • Capacidad de aprender • Habilidad para trabajar en forma autónoma

Subtemas:	**Actividades de enseñanza**	**Actividades de aprendizaje**	**Criterios de evaluación**

4.1 Definición formal MT 4.2 Construcción modular de una MT 4.3 Lenguajes aceptados por la MT.	Exponer los conceptos de la unidad. Resolver ejercicios en clase explicando su solución. Diseño de problemario con ejercicios que permitan: Diseño de autómatas finitos a partir del lenguaje o expresión regular identificando su tipo, convertir NFA a DFA	Asistir al 100% de las clases. Sintetizar la información expuesta. Construir su propio conocimiento. Resolución de ejercicios propuestos por el profesor	**Evidencias cognoscitivas** Problemario 10% Examen 10% **Evidencias procedimentales** **Evidencias actitudinales**

Instrumentos de evaluación (Rubrica, lista de cotejo, etc.)

6.- PROBLEMARIO INDIVIDUAL
NIVELES DE DESEMPEÑO

Suficiente 70	Bueno 80	Notable 90	Excelente 100
Resuelve el 70% de los problemas del problemario correctamente	Resuelve el 80% de los problemas del problemario correctamente	Resuelve el 90% de los problemas del problemario correctamente	Resuelve el 100% de los problemas del problemario correctamente.

7.- EXAMEN INDIVIDUAL
NIVELES DE DESEMPEÑO

Suficiente 70	Bueno 80	Notable 90	Excelente 100
Resuelve el 70% de los problemas del examen correctamente	Resuelve el 80% de los problemas del examen correctamente	Resuelve el 90% de los problemas del examen correctamente	Resuelve el 100% de los problemas del examen correctamente.

Fuentes de información	Apoyos didácticos
1. Hopcroft John E., Introducción a la Teoría de Autómatas, Lenguajes y Computación, 2da ed, Ed. Addison Wesley, 2004. 2. Lemote Karen A. , Fundamentos de compiladores Cómo traducir al lenguaje de computadora, Ed. Compañía Editorial Continental.	Equipo de cómputo. Cañón. Pizarrón

No. de la unidad:	5	Tema de la unidad:	Análisis léxico.	Horas teórico-prácticas	10

Competencia específica de la unidad	Desarrollo de competencias genéricas
Construir un analizador léxico a partir de un lenguaje de programación o un analizador léxico (p. e. Flex, Lex, JavaCC).	• Capacidad de análisis y síntesis Capacidad de organizar y planificar • Comunicación oral y escrita • Habilidad para buscar y analizar información proveniente de fuentes diversas • Solución de problemas • Toma de decisiones. • Capacidad crítica y autocrítica • Trabajo en equipo • Capacidad de aplicar los conocimientos en la práctica • Habilidades de investigación • Capacidad de aprender • Habilidad para trabajar en forma autónoma

Subtemas:	Actividades de enseñanza	Actividades de aprendizaje	Criterios de evaluación
5.1 Funciones del analizador léxico 5.2 Componentes léxicos, patrones y lexemas 5.3 Creación de Tabla de tokens 5.4 Errores léxicos 5.5 Generadores de analizadores Léxicos 5.6 Aplicaciones (Caso de estudio)	Exponer los conceptos de la unidad. Diseño de proyecto de léxico.	Asistir al 100% de las clases. Sintetizar la información expuesta. Construir su propio conocimiento. Diseño y programación de léxico solicitado por el profesor	**Evidencias cognoscitivas** **Evidencias procedimentales** Diseño y programación de léxico solicitado por el profesor 15% **Evidencias actitudinales**

Instrumentos de evaluación (Rubrica, lista de cotejo, etc.)

8.- DISEÑO DE LÉXICO PARA LENGUAJE DE PROGRAMACIÓN CREADO (EQUIPO)
NIVELES DE DESEMPEÑO

Suficiente 70	Bueno 80	Notable 90	Excelente 100
Software sin errores. Justificación de lenguaje de programación creado Diseño de 5 a 10 token´s de lenguaje de programación creado. Incluir Autómata. Por cada token reglas léxicas (expresión regular)	Software sin errores. Software hecho por integrantes del equipo(no copia) Justificación de lenguaje de programación creado Diseño de 5 a 10 token´s de lenguaje de programación creado. Incluir Autómata. Por cada token reglas léxicas (expresión regular)	Software sin errores. Software hecho por integrantes del equipo(no copia) Justificación de lenguaje de programación creado Diseño de 10 token´s de lenguaje de programación creado. Incluir Autómata. Por cada token reglas léxicas (expresión regular)	Software sin errores. Software hecho por integrantes del equipo(no copia) Justificación de tokens y lenguaje de programación creado Diseño más de 10 token´s de lenguaje de programación creado. Incluir Autómata. Por cada token reglas léxicas (expresión regular) Entrega puntual.

Fuentes de información	Apoyos didácticos
1. Hopcroft John E., Introducción a la Teoría de Autómatas, Lenguajes y Computación, 2da ed, Ed. Addison Wesley, 2004. 2. Lemote Karen A. , Fundamentos de compiladores Cómo traducir al lenguaje de computadora, Ed. Compañía Editorial Continental.	Equipo de cómputo. Cañón. Pizarrón

No. de la unidad:	6	Tema de la unidad:	Análisis Sintáctico	Horas teórico-prácticas	5

Competencia específica de la unidad	Desarrollo de competencias genéricas
Construir un analizador sintáctico a partir de un lenguaje de programación o un analizador sintáctico para el reconocimiento de gramáticas (p.e. YACC).	• Capacidad de análisis y síntesis Capacidad de organizar y planificar • Comunicación oral y escrita • Habilidad para buscar y analizar información proveniente de fuentes diversas • Solución de problemas • Toma de decisiones. • Capacidad crítica y autocrítica • Trabajo en equipo • Capacidad de aplicar los conocimientos en la práctica • Habilidades de investigación • Capacidad de aprender • Habilidad para trabajar en forma autónoma

Subtemas:	Actividades de enseñanza	Actividades de aprendizaje	Criterios de evaluación
6.1 GLC 6.2 Árboles de derivación. 6.3 Formas normales de Chomsky. 6.4 Diagramas de sintaxis 6.5 Eliminación de la ambigüedad. 6.6 Generación de matriz predictiva 6.7 Tipos de analizadores sintácticos 6.8 Manejo de errores 6.9 Generadores de analizadores sintácticos	Exponer los conceptos de la unidad. Diseño de problemario. Resolver en clase ejercicios base. Diseño de proyecto de la unidad.	Asistir al 100% de las clases. Sintetizar la información expuesta. Construir su propio conocimiento. Resolver ejercicios indicados por el profesor. Diseño y programación de proyecto de la unidad.	**Evidencias cognoscitivas** Exámen 5% **Evidencias procedimentales** Problemario 5% Proyecto de software 20% **Evidencias actitudinales**

Instrumentos de evaluación (Rubrica, lista de cotejo, etc.)

9.- PROBLEMARIO INDIVIDUAL **NIVELES DE DESEMPEÑO**			
Suficiente 70	**Bueno 80**	**Notable 90**	**Excelente 100**
Resuelve el 70% de los problemas del problemario correctamente	Resuelve el 80% de los problemas del problemario correctamente	Resuelve el 90% de los problemas del problemario correctamente	Resuelve el 100% de los problemas del problemario correctamente.

10.- EXAMEN INDIVIDUAL **NIVELES DE DESEMPEÑO**			
Suficiente 70	**Bueno 80**	**Notable 90**	**Excelente 100**
Resuelve el 70% de los problemas del examen correctamente	Resuelve el 80% de los problemas del examen correctamente	Resuelve el 90% de los problemas del examen correctamente	Resuelve el 100% de los problemas del examen correctamente.

11.- PROGRAMACIÓN DE SINTAXIS PROPUESTA POR EL PROFESOR (EQUIPO) **NIVELES DE DESEMPEÑO**			
Suficiente 70	**Bueno 80**	**Notable 90**	**Excelente 100**
Software sin errores. Programación de la gramática propuesta por el profesor por un método sintáctico.	Software sin errores. Programación de la gramática propuesta por el profesor por DOS métodos sintácticos.	Software sin errores. Software hecho por integrantes del equipo(no copia) Programación de la gramática propuesta por el profesor por dos métodos sintácticos.	Software sin errores. Software hecho por integrantes del equipo(no copia) Programación de la gramática propuesta por el profesor por dos métodos sintácticos. Entrega puntual.

Fuentes de información	**Apoyos didácticos**
1. Hopcroft John E., Introducción a la Teoría de Autómatas, Lenguajes y Computación, 2da ed, Ed. Addison Wesley, 2004. 2. Lemote Karen A. , Fundamentos de compiladores Cómo traducir al lenguaje de computadora, Ed. Compañía Editorial Continental.	Equipo de cómputo. Cañón. Pizarrón

Registro de avance de la gestión del curso

Periodo Escolar:	ENERO-JUNIO 2018	Clave de la Asignatura:	SCD-1015
Nombre de la Asignatura:	Lenguajes y Autómatas I	Clave del grupo:	S6B, S6A
Horas teoría-horas práctica-créditos	2-3-5	Número de unidades	6

Calendarización de avance y evaluación del curso (semanas):

Semana	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
No. unidad planeada	1	2	2	2	3	3	3	4	4	4	5	6	6	6	6	6
No. unidad real																
Evaluación planeada		☐ -1		☐ -2 O-3			☐ -4 O-5			☐ -6 O-7	O-8					☐ -9 O-10 O-11
Evaluación real																
Índice de aprobación																
Fecha de seguimiento	Primera: 7 al 9 de marzo				Segunda: 19 y 20 de abril				Tercera: 16 al 18 de mayo				Final: 20 al 22 de junio			
Firma del docente																
Firma del Jefe académico																
Observaciones																

Δ = Evaluación diagnóstica. ☐ Evaluación formativa. O = Evaluación sumativa.

FECHA DE ENTREGA: 31 enero 218

M.C. ROSY ILDA BASAVE TORRES
NOMBRE Y FIRMA DEL DOCENTE

DRA. MARIA GUADALUPE MONJARAS VELASCO
Vo. Bo. DEL JEFE DE DEPARTAMENTO