



UNIVERSIDAD AUTÓNOMA DE SINALOA

FACULTAD DE INGENIERIA MOCHIS LICENCIATURA EN INGENIERIA DE SOFTWARE



PROGRAMA DE ESTUDIO

1. DATOS DE IDENTIFICACIÓN			
UNIDAD DE APRENDIZAJE O MÓDULO:	TEORÍA DE LA COMPUTACIÓN		
Clave:			
Ubicación:	Semestre V	Área: Profesionalizante	
Horas y créditos:	Teóricas: 50	Prácticas: 30	Estudio Independiente: 80
	Total de horas: 160		Créditos: 10
Competencia(s) del perfil de egreso al que aporta:	<p>CG1. Desarrolla su potencial intelectual para generar el conocimiento necesario en la resolución de problemas y retos, tanto de su vida individual y como parte de una comunidad, con sentido de pertinencia, identidad y empatía.</p> <p>CG7. Cultiva el compañerismo, el trabajo en equipo y la coordinación de esfuerzos bajo la aspiración de mejorar las tareas académicas, los entornos laborales y la convivencia social en beneficio para la consecución de metas que impactan en las formas de entablar y mantener relaciones humanas positivas.</p> <p>CE2. Desarrolla habilidades de abstracción y la expresión de formalismos, además de proporcionar conocimientos específicos fundamentales para la informática y la computación.</p> <p>CE3. Aplica los conceptos básicos de conjuntos, lógica matemática, relaciones, grafos y árboles para resolver problemas afines al área computacional e identifica las estructuras básicas de las matemáticas discretas, cómo aplicarlas en el manejo y tratamiento de las TICS.</p>		
Unidades de aprendizaje relacionadas:	Software de sistemas		
Responsable(s) de elaborar el programa:	Alan David Ramírez Noriega		Fecha: 13 de mayo de. 2025
Responsable(s) de actualizar el programa:			Fecha:
2. PROPÓSITO			
La materia tiene como objetivo fundamental proporcionar a los estudiantes de Ingeniería de Software las bases teóricas que sustentan la computación moderna. A través del estudio de modelos formales como autómatas, lenguajes formales, máquinas de Turing y teoría de la complejidad computacional, los alumnos comprenderán los límites y capacidades de los algoritmos y sistemas computacionales. Esto les permitirá diseñar soluciones eficientes,			



PROGRAMA DE ESTUDIO

evaluar la viabilidad de problemas y fundamentar el desarrollo de software robusto y escalable, integrando conocimientos matemáticos y lógicos esenciales para su formación profesional.

3. SABERES

Teóricos:	Conoce temas relacionados con autómatas finitos, autómatas de pilas y máquinas de Turing; así mismo, los conceptos de resolubilidad y complejidad computacional.
Prácticos:	Diseña y programa autómatas finitos, autómatas de pilas y máquinas de Turing para su posterior implementación en la solución de problemas de clase. Expone temas relacionados con resolubilidad y complejidad computacional.
Actitudinales:	Proactivo, trabaja en equipo, tiene disposición para el desarrollo de prácticas, asiste a reuniones virtuales de manera puntual y es constante, pone interés por la clase virtual, entre otros.

4. CONTENIDOS

Unidad 1. Lenguajes Regulares y Autómatas Finitos

- 1.1 Análisis léxico
- 1.2 Autómatas finitos deterministas
- 1.3 Límites de los autómatas finitos deterministas
- 1.4 Autómatas finitos no deterministas
- 1.5 Gramáticas regulares
- 1.6 Expresiones regulares

Unidad 2. Lenguajes Independientes del Contexto y Autómatas de Pila

- 2.1 Autómatas de pila.
- 2.2 Gramáticas independientes del contexto.
- 2.3 Límites de los autómatas de pila
- 2.4 Analizadores sintácticos LL.
- 2.5 Analizadores sintácticos LR.

Unidad 3. Lenguajes Estructurados por Frases y Máquinas de Turing

- 3.1 Máquinas de Turing (MT).
- 3.2 Construcción modular de MT.
- 3.3 MT como aceptadores de lenguajes.
- 3.4 Lenguajes aceptados por MT.

Unidad 4. Resolubilidad y Complejidad Computacional

- 4.1 Resolubilidad y Complejidad Computacional



5. ACTIVIDADES PARA DESARROLLAR LAS COMPETENCIAS

Actividades del docente:

- Explicaciones del tema en clase.
- Uso de cañón.
- Uso de videos educativos.
- Prácticas.
- Trabajo extra clase.
- Actividades de programación

Actividades del estudiante:

- ❖ Trabajo en equipo.
- ❖ Investigación. Asistencia.
- ❖ Participación.
- ❖ Cumplimiento de prácticas y tareas.

6. EVALUACIÓN DE LAS COMPETENCIAS

6.1. Criterios de desempeño

1. El alumno describe que es la teoría de la computación
2. El alumno identifica las fases de un compilador
3. El alumno elabora un analizador léxico
4. El alumno desarrolla y emplea expresiones regulares
5. El alumno diseña e implementa AFND y AFD
6. El alumno diseña e implementa un analizador sintáctico.
7. El alumno diseña y simula máquinas de Turing.
8. El alumno describe los conceptos de resolubilidad y complejidad computacional

6.2 Portafolio de evidencias

Práctica 1. Implementar una rutina para reconocer la tabla de transición

Práctica 2. Implementa un programa que valide los elementos para un lenguaje de programación.

Práctica 3. Elaborar una conversión de un AFND a un AFD.

Práctica 4. Elaborar un programa donde valides a través de expresiones regulares algunas cadenas.

Práctica 5. Elaborar un programa donde me valide el uso de los paréntesis para una expresión aritmética.

Práctica 6. Empleando gramáticas libres de contexto, genera tu propia sintaxis de un lenguaje de programación

Práctica 7. Empleando la gramática libre de contexto generada en la práctica 6, elabora un programa que valide sintácticamente el código de un programa

Práctica 8. Elaborar un reporte sobre las máquinas de Turing.

Práctica 9. Elaborar un reporte y exposición sobre resolubilidad y complejidad computacional.

6.3. Calificación y acreditación:



UNIVERSIDAD AUTÓNOMA DE SINALOA

FACULTAD DE INGENIERIA MOCHIS LICENCIATURA EN INGENIERIA DE SOFTWARE



PROGRAMA DE ESTUDIO

Parcial: Examen 50% Prácticas y tareas 50% Otras consideraciones <ul style="list-style-type: none">• 8.0 al menos exentan ordinario• 80% de asistencia al menos para hacer examen parcial• Prácticas y reportes entregados después de la fecha no se reciben• 3 días hábiles para justificar faltas• Tolerancia de 10 Minutos	Final: Parcial 1 25% Parcial 2 25% Parcial 3 25% Ordinario 25%			
7. RECURSOS DIDÁCTICOS				
Proyector, libreta, lápices, plumones, plumas.				
8. FUENTES DE INFORMACIÓN				
<i>Bibliografía básica</i>				
Autor(es)	Título	Editorial	Año	URL o biblioteca digital donde está disponible
J. Glenn Bookshear	Teoría de la computación: Lenguajes formales, autómatas y complejidad.	Addison–Wesley	1993	
Araceli Sanchis de Miguel, Agapito Ledezma Espino, José A. Iglesias Martínez, Beatriz García Jiménez, Juan Manuel Alonso Weber	Teoría de Autómatas y Lenguajes Formales	Departamento de Informática Universidad Carlos III de Madrid	2012	http://163.117.136.247/ingenieria-informatica/teoria-de-automatas-y-lenguajes-formales
<i>Bibliografía complementaria</i>				



UNIVERSIDAD AUTÓNOMA DE SINALOA

FACULTAD DE INGENIERIA MOCHIS
LICENCIATURA EN INGENIERIA DE SOFTWARE



PROGRAMA DE ESTUDIO

Autor(es)	Título	Editorial	Año	URL o biblioteca digital donde está disponible
Dean Kelley	Teoría de autómatas y lenguajes formales.	Prentice–Hall	1995	
9. PERFIL DEL DOCENTE				
Carrera Profesional con perfil en computación, preferentemente con posgrado				