

**Programa descriptivo por unidad de competencia**

<b>Programa educativo</b>	<b>Licenciatura en Ingeniería en Desarrollo y Tecnologías de Software</b>	<b>Modalidad</b>		Presencial	
<b>Clave</b>	MA09	<b>H S M</b>		<b>Horas semestrales</b>	<b>Créditos</b>
<b>Unidad de competencia</b>	<b>Teoría matemática de la computación.</b>	<b>Teoría</b>	<b>Práctica</b>	80	8
		3	2		
<b>Ubicación</b>	Quinto semestre.	<b>Unidades CONAIC</b>		58.67	
<b>Prerrequisito</b>	Estructura de datos y Matemáticas discretas.	<b>H S M de cómputo</b>		1	
<b>Perfil docente</b>	Contar con un título profesional o posgrado en áreas relacionadas con informática y computación preferentemente con el grado de doctorado. Demostrar experiencia en docencia en nivel superior mínima de dos años. Dominio de la programación de propósito general y compiladores y deseable que cuente con certificación en lenguajes de programación.				
<b>Presentación</b>	Esta unidad de competencia define modelos y máquinas que pueden ser implementadas en el desarrollo de lenguajes de programación. Las unidades de competencia como conocimiento previo son Estructura de datos por las herramientas para el procesamiento de información que proporciona y Matemáticas discretas que aporta los fundamentos matemáticos de la teoría de conjuntos.				
<b>Propósito</b>	Modela matemáticamente las etapas de léxico y sintaxis de un compilador.				
<b>Competencias genéricas</b>					
Formula propuestas y gestiona proyectos con una visión de sustentabilidad para la solución de problemas. Se mantiene actualizado en los conocimientos y habilidades de manera permanente y los utiliza en su práctica profesional y vida personal.					
<b>Competencias disciplinares</b>					
Aplica habilidades de abstracción y expresión matemática para la solución de problemas. Formula modelos matemáticos para la solución de problemas mediante el desarrollo de aplicaciones de software para diversos entornos. Posee conocimientos formales sobre las bases matemáticas de la computación y los aplica en la solución de problemas.					
<b>Competencias profesionales</b>					
Realiza el modelado de problemas con base en la computación teórica. Realiza la codificación de algoritmos en computación teórica. Selecciona técnicas o algoritmos en computación teórica para la solución de problemas.					

**Mapa de la unidad de competencia**

Unidad de competencia	Subcompetencia	Resultado de aprendizaje
<p><b>Teoría matemática de la computación</b></p>	1. Comprende las bases de la teoría de lenguaje.	1.1. Comprende conceptos sobre lenguajes formales.
	2. Utiliza el lenguaje y las expresiones regulares.	2.1. Aplica los conceptos de lenguaje regular en el desarrollo de ejercicios.
	3. Resuelve problemas de autómatas finitos.	3.1. Aplica los conceptos de autómatas finitos en el desarrollo de ejercicios.
	4. Hace uso de gramáticas.	4.1. Aplica los conceptos de gramáticas en el desarrollo de ejercicios.
	5. Comprende el funcionamiento de las Máquinas de Turing.	5.1. Conoce el concepto de la máquina de Turing.
	6. Comprende la aplicación de la Computabilidad.	6.1. Conoce los conceptos de computabilidad.

**Cuadro descriptivo por subcompetencia**

<b>Subcompetencia</b>	<b>Comprende las bases de la teoría de los lenguajes formales.</b>			<b>Número</b>	<b>1</b>
<b>Propósito de la subcompetencia</b>	Comprende los conceptos básicos relacionados con los lenguajes.			<b>Total de horas</b>	10
<b>Resultado de aprendizaje</b>	1.1. Comprende conceptos sobre lenguajes formales.			<b>Horas asignadas</b>	10
<b>Actividades de evaluación</b>	<b>Evidencias a recopilar</b>	<b>%</b>	<b>Contenido</b>		
1. Resuelve un cuestionario con los temas vistos.	1. Cuestionario resuelto.	10%	1. Teoría de conjuntos orientada a lenguaje formal. 2. Conceptos de lenguaje formal. 3. Tipos de lenguaje. 4. Etapas de un compilador.		

**Cuadro descriptivo por subcompetencia**

<b>Subcompetencia</b>	<b>Utiliza el lenguaje y las expresiones regulares.</b>	<b>Número</b>	<b>2</b>
<b>Propósito de la subcompetencia</b>	Aplica el concepto de lenguaje regular en el desarrollo conceptual de la etapa de léxico de un lenguaje de programación.	<b>Total de horas</b>	10
<b>Resultado de aprendizaje</b>	2.1. Aplica los conceptos de lenguaje regular en el desarrollo de ejercicios.	<b>Horas asignadas</b>	10
<b>Actividades de evaluación</b>	<b>Evidencias a recopilar</b>	<b>%</b>	<b>Contenido</b>
1. Realiza ejercicios de los conceptos vistos.	1. Reporte de ejercicios.	15%	1. Definición formal de lenguaje regular. 2. Definición de expresión regular. 3. Diseño de expresiones regulares.

**Cuadro descriptivo por subcompetencia**

<b>Subcompetencia</b>	<b>Resuelve problemas de autómatas finitos.</b>	<b>Número</b>	<b>3</b>
<b>Propósito de la subcompetencia</b>	Aplica los conceptos de autómatas finitos en el desarrollo conceptual de la etapa de léxico de un lenguaje de programación.	<b>Total de horas</b>	15
<b>Resultado de aprendizaje</b>	3.1. Aplica los conceptos de autómatas finitos en el desarrollo de ejercicios.	<b>Horas asignadas</b>	15
<b>Actividades de evaluación</b>	<b>Evidencias a recopilar</b>	<b>%</b>	<b>Contenido</b>
1. Realiza ejercicios de los conceptos vistos.	1. Reporte de ejercicios.	20%	<ol style="list-style-type: none"> <li>1. Definición formal de autómata finito.</li> <li>2. Tipos de autómatas finitos.</li> <li>3. Conversión de autómata finito no determinista a autómata finito determinista.</li> <li>4. Representación de expresión regular en autómata finito.</li> <li>5. Aplicaciones de los autómatas.</li> </ol>

**Cuadro descriptivo por subcompetencia**

<b>Subcompetencia</b>	<b>Hace uso de gramáticas.</b>	<b>Número</b>	<b>4</b>
<b>Propósito de la subcompetencia</b>	Aplica los conceptos de gramáticas en el desarrollo conceptual de la etapa de sintaxis de un lenguaje de programación.	<b>Total de horas</b>	25
<b>Resultado de aprendizaje</b>	4.1. Aplica los conceptos de gramáticas en el desarrollo de ejercicios.	<b>Horas asignadas</b>	25
<b>Actividades de evaluación</b>	<b>Evidencias a recopilar</b>	<b>%</b>	<b>Contenido</b>
1. Realiza ejercicios de los conceptos vistos.	1. Reporte de ejercicios.	25%	1. Definición formal de gramáticas. 2. Jerarquía de Chomsky. 3. Diseño de gramáticas. 4. Aplicaciones de las gramáticas.

**Cuadro descriptivo por subcompetencia**

<b>Subcompetencia</b>	<b>Comprende el funcionamiento de las Máquinas de Turing.</b>			<b>Número</b>	<b>5</b>
<b>Propósito de la subcompetencia</b>	Conocer los conceptos básicos de Máquina de Turing.			<b>Total de horas</b>	10
<b>Resultado de aprendizaje</b>	5.1. Conoce el concepto de la Máquina de Turing.			<b>Horas asignadas</b>	10
<b>Actividades de evaluación</b>	<b>Evidencias a recopilar</b>	<b>%</b>	<b>Contenido</b>		
1. Desarrolla ejercicios de los temas vistos.	1. Reporte de ejercicios.	15%	1. Definición formal de Máquina de Turing. 2. Diseño de Máquinas de Turing.		

**Cuadro descriptivo por subcompetencia**

<b>Subcompetencia</b>	<b>Comprende la aplicación de la computabilidad.</b>		<b>Número</b>	<b>6</b>
<b>Propósito de la subcompetencia</b>	Conocer la complejidad y acotamiento en teorías de primer orden para la computabilidad.		<b>Total de horas</b>	10
<b>Resultado de aprendizaje</b>	6.1. Conoce los conceptos de computabilidad.		<b>Horas asignadas</b>	10
<b>Actividades de evaluación</b>	<b>Evidencias a recopilar</b>	<b>%</b>	<b>Contenido</b>	
1. Realiza investigación sobre los diferentes conceptos de computabilidad.	1. Informe de investigación que especifique los conceptos de computabilidad.	15%	1. Complejidad y computabilidad. 2. Complejidad de algoritmos. 3. Teorema del acotamiento. 4. Clases de complejidad. 5. Computabilidad polinomio.	

<b>Actitudes y valores</b>	Responsabilidad. Honestidad. Ética.	
<b>Recursos, materiales y equipo didáctico</b>		
<b>Recursos didácticos</b>	<b>Equipo de apoyo didáctico</b>	
Apuntes. Diapositivas. Ejercicios. Guías de práctica.	Proyector de video.	
<b>Fuentes de información</b>		
<b>Bibliografía básica:</b> Hopcroft, J. E. (2006). <i>Introduction to Automata Theory, Languages and Computation</i> , (3rd. Ed.). USA: Adison Wesley. Jiménez Murillo, Jose A. (2008). <i>Matemáticas para computación</i> . México: Alfaomega. Kelley, D. (1995). <i>Teoría de Autómatas y Lenguajes Formales</i> . Madrid: Prentice Hall. Guerra Crespo, H. (2005). <i>Compiladores, el comienzo...</i> México: Tecnología Didáctica.		
<b>Bibliografía complementaria:</b> Aho, Lam, Sethi, Ullman (2008). <i>Compiladores, principios, técnicas y herramientas</i> , (2a. ed.). México: Pearson Adison Wesley. Louden K.C. (2004). <i>Construcción de compiladores: principios y práctica</i> . México: Thomson.		
<b>Recursos digitales:</b> Ninguno.		