

**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>	SB02	<b>H S M</b>		<b>Horas semestrales</b>	<b>Créditos</b>
<b>Unidad de competencia</b>	<b>Compiladores</b>	<b>Teoría</b>	<b>Práctica</b>	80	8
<b>Ubicación</b>	Sexto semestre.	<b>Unidades CONAIC</b>		58.67	
<b>Prerrequisito</b>	Teoría matemática de la computación.	<b>H S M de cómputo</b>		2	
<b>Perfil docente</b>	Contar con un título profesional o posgrado en áreas relacionadas con informática y computación preferentemente con el grado de doctorado. Experiencia en docencia en nivel superior mínima de dos años. Dominio de la programación de propósito general y del área de compiladores. Es deseable que cuente con certificación en lenguajes de programación.				
<b>Presentación</b>	Esta unidad de competencia se ubica en el área de software de base, se basa en el desarrollo de software científico y para el desarrollo tecnológico, indispensable para preparar a los egresados y tengan la posibilidad de cursar posgrados de alto nivel. La unidad de competencia ayuda a un estudiante a crear su propio lenguaje de programación. La aportación de la unidad de competencia al perfil del egreso es la de desarrollar, implementar y administrar software de sistemas.				
<b>Propósito</b>	Desarrollar las etapas de análisis léxico y sintáctico, modelar las etapas de análisis semántico, la generación de código, la optimización y la generación del código objeto para desarrollar un traductor, intérprete o compilador.				
<b>Competencias genéricas</b>					
Aplica un pensamiento sistémico y complejo en la construcción de conocimientos y toma de decisiones. 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 técnicas y metodologías para la construcción de software de base y sistemas operativos de las computadoras. Posee conocimientos teóricos para el diseño conceptual del software de base y sistemas operativos de las computadoras.					
<b>Competencias profesionales</b>					
Desarrolla productos de software aplicando modelos y metodologías para el aseguramiento de la calidad en los procesos de desarrollo. Selecciona técnicas o algoritmos en computación teórica para la solución de problemas. Realiza la codificación de algoritmos en computación teórica.					

**Mapa de la unidad de competencia**

Unidad de competencia	Subcompetencia	Resultado de aprendizaje
<b>Compiladores</b>	1. Implementa la etapa de análisis léxico en un lenguaje de programación	1.1. Aplica los conceptos de análisis léxico en el desarrollo de la etapa de léxico de un lenguaje de programación.
	2. Implementa la etapa de análisis sintáctico en un lenguaje de programación	2.1. Aplica los conceptos de análisis sintáctico en el desarrollo de la etapa de sintaxis de un lenguaje de programación.
	3. Implementa la etapa de análisis semántico en un lenguaje de programación	3.1. Aplica los conceptos de análisis semántico en el desarrollo de la etapa de semántica de un lenguaje de programación.
	4. Genera código intermedio.	4.1. Comprende los conceptos de la etapa de generación de código intermedio de un lenguaje de programación.
	5. Optimiza el código intermedio.	5.1. Comprende los conceptos de la etapa de optimización de código intermedio de un lenguaje de programación.
	6. Generación de código.	6.1. Comprende los conceptos de la etapa de generación de código de un lenguaje de programación.

**Cuadro descriptivo por subcompetencia**

<b>Subcompetencia</b>	<b>Implementa la etapa de análisis léxico en un lenguaje de programación.</b>		<b>Número</b>	<b>1</b>
<b>Propósito de la subcompetencia</b>	Aplicar los conceptos que involucran el desarrollo de la etapa de análisis léxico en un lenguaje de programación.		<b>Total de horas</b>	15
<b>Resultado de aprendizaje</b>	1.1. Aplica los conceptos de análisis léxico en el desarrollo de la etapa de léxico de un lenguaje de programación.		<b>Horas asignadas</b>	15
<b>Actividades de evaluación</b>	<b>Evidencias a recopilar</b>	<b>%</b>	<b>Contenido</b>	
1. Implementa la etapa de léxico para un lenguaje de programación.	1. Programa de cómputo incluyendo la evidencia de la subcompetencia 2 y 3 (proyecto integra 3 unidades de aprendizaje).	15%	1. Funciones del analizador léxico. 2. Componentes léxicos, patrones y lexema. 3. Generación de la etapa de léxico a partir de un autómata finito.	

**Cuadro descriptivo por subcompetencia**

<b>Subcompetencia</b>	<b>Implementa la etapa de análisis sintáctico en un lenguaje de programación.</b>		<b>Número</b>	<b>2</b>
<b>Propósito de la subcompetencia</b>	Aplicar los conceptos que involucran el desarrollo de la etapa de sintaxis en un lenguaje de programación.		<b>Total de horas</b>	20
<b>Resultado de aprendizaje</b>	2.1. Aplica los conceptos de análisis sintáctico en el desarrollo de la etapa de sintaxis de un lenguaje de programación.		<b>Horas asignadas</b>	20
<b>Actividades de evaluación</b>	<b>Evidencias a recopilar</b>	<b>%</b>	<b>Contenido</b>	
1. Implementa la etapa de sintaxis para un lenguaje de programación.	1. Programa de cómputo incluyendo la evidencia de la Subcompetencia 1 y 3 (proyecto integra 3 unidades de aprendizaje).	20%	1. Gramáticas independientes del contexto. 2. Métodos sintácticos descendentes - predictivo recursivo o directo. 3. Métodos sintácticos descendentes - predictivo no recursivo. 4. Métodos sintácticos ascendentes - LR.	

**Cuadro descriptivo por subcompetencia**

<b>Subcompetencia</b>	<b>Implementa la etapa de análisis sintáctico en un lenguaje de programación.</b>	<b>Número</b>	<b>3</b>
<b>Propósito de la subcompetencia</b>	Aplicar los conceptos que involucran el desarrollo de la etapa de sintaxis en un lenguaje de programación.	<b>Total de horas</b>	15
<b>Resultado de aprendizaje</b>	3.1. Aplica los conceptos de análisis semántico en el desarrollo de la etapa de semántica de un lenguaje de programación.	<b>Horas asignadas</b>	15
<b>Actividades de evaluación</b>	<b>Evidencias a recopilar</b>	<b>%</b>	<b>Contenido</b>
1. Implementa la etapa de semántica para un lenguaje de programación.	1. Programa de cómputo incluyendo la evidencia de la Subcompetencia 1 y 2 (proyecto integra 3 unidades de aprendizaje).	20%	<ol style="list-style-type: none"> <li>1. Árboles de expresiones.</li> <li>2. Acciones semánticas de un analizador sintáctico.</li> <li>3. Comprobaciones de tipos en expresiones.</li> <li>4. Pila semántica en un analizador sintáctico.</li> <li>5. Esquema de traducción.</li> <li>6. Generación de la tabla de símbolo y de direcciones.</li> </ol>

**Cuadro descriptivo por subcompetencia**

<b>Subcompetencia</b>	<b>Genera código intermedio.</b>	<b>Número</b>	<b>4</b>
<b>Propósito de la subcompetencia</b>	Comprender los conceptos que involucran el desarrollo de la etapa de generación de código intermedio en un lenguaje de programación.	<b>Total de horas</b>	10
<b>Resultado de aprendizaje</b>	4.1. Comprende los conceptos de la etapa de generación de código intermedio de un lenguaje de programación.	<b>Horas asignadas</b>	10
<b>Actividades de evaluación</b>	<b>Evidencias a recopilar</b>	<b>%</b>	<b>Contenido</b>
1. Comprende la etapa de generación de código intermedio para un lenguaje de programación.	1. Informe descriptivo de la etapa de generación de código intermedio de un lenguaje de programación (proyecto integrado a las unidades de aprendizaje 5 y 6).	20%	<ol style="list-style-type: none"> <li>1. Notaciones prefija, infija y posfija.</li> <li>2. Representaciones de código intermedio - notación polaca.</li> <li>3. Representaciones de código intermedio - Código P.</li> <li>4. Representaciones de código intermedio- triplos.</li> <li>5. Representaciones de código intermedio- cuádruplos.</li> <li>6. Esquema de generación - variables y constantes.</li> <li>7. Esquema de generación - expresiones.</li> <li>8. Esquema de generación - instrucciones de asignación.</li> <li>9. Esquema de generación - instrucciones de control.</li> <li>10. Esquema de generación - funciones.</li> <li>11. Esquema de generación - estructuras.</li> </ol>

**Cuadro descriptivo por subcompetencia**

Subcompetencia	Optimiza el código intermedio.	Número	5
Propósito de la subcompetencia	Comprender los conceptos que involucran el desarrollo de la etapa de optimización de código intermedio en un lenguaje de programación.	Total de horas	10
Resultado de aprendizaje	5.1. Comprende los conceptos de la etapa de optimización de código intermedio de un lenguaje de programación.	Horas asignadas	10
Actividades de evaluación	Evidencias a recopilar	%	Contenido
1. Comprende la etapa de optimización de código intermedio para un lenguaje de programación.	1. Informe descriptivo de la etapa de optimización de código intermedio de un lenguaje de programación (proyecto integrado a las unidades de aprendizaje 4 y 6).	10%	<ol style="list-style-type: none"> <li>1. Tipos de optimización - locales.</li> <li>2. Tipos de optimización - ciclos.</li> <li>3. Tipos de optimización - globales.</li> <li>4. Tipos de optimización - de mirilla.</li> <li>5. "Costos - costo de ejecución. (memoria, registros, pilas).</li> <li>6. Costos - criterios para mejorar el código.</li> <li>7. Costos - herramientas para el análisis del flujo de datos.</li> </ol>

**Cuadro descriptivo por subcompetencia**

<b>Subcompetencia</b>	<b>Generación de código.</b>			<b>Número</b>	<b>6</b>
<b>Propósito de la subcompetencia</b>	Comprender los conceptos que involucran el desarrollo de la etapa de generación de código en un lenguaje de programación.			<b>Total de horas</b>	10
<b>Resultado de aprendizaje</b>	6.1. Comprende los conceptos de la etapa de generación de código de un lenguaje de programación.			<b>Horas asignadas</b>	10
<b>Actividades de evaluación</b>	<b>Evidencias a recopilar</b>	<b>%</b>	<b>Contenido</b>		
1. Comprende la etapa de generación de código para un lenguaje de programación.	1. Informe descriptivo de la etapa de generación de código de un lenguaje de programación (proyecto integrado a las unidades de aprendizaje 4 y 5).	15%	1. Arquitectura de computadora. 2. Lenguaje ensamblador. 3. Administración de la memoria. 4. Lenguaje máquina.		



<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> Guerra Crespo, H. (2005). <i>Compiladores, el comienzo...</i> México: Tecnología Didáctica. 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> . Mexico: Thomson. Ruiz Catalan, Jacinto (2010). <i>Compiladores teoría e implementación</i> . México: Alfaomega.		
<b>Bibliografía complementaria:</b> Ninguna.		
<b>Recursos digitales:</b> Ninguna.		