



**Programa de estudios por competencias**  
**Traductores de lenguajes II**

**1. Identificación del curso**

<b>Programa educativo: Ing. En Computación</b>		<b>Unidad de aprendizaje: Traductores de lenguajes II</b>			<b>Departamento de adscripción: Estudios Organizacionales</b>			
<b>Academia:</b> Sistemas digitales y de información		<b>Programa elaborado por:</b> María Obdulia González Fernández			<b>Modificado por:</b>		<b>Fecha elaboración/modificación:</b> Diciembre	
<b>Clave de la asignatura:</b>	<b>Horas teóricas:</b>	<b>Horas prácticas:</b>		<b>Total de Horas:</b>	<b>Créditos:</b>	<b>Tipo de materia</b>	<b>Área de formación:</b>	<b>Modalidad:</b>
	51	17		68	8	Curso		<b>Presencial</b>
<b>Conocimientos previos:</b>				<b>Unidad de aprendizaje precedente:</b>			<b>Unidad de aprendizaje subsecuente:</b>	
Interpreta algoritmos computacionales y notaciones matemáticas (autómatas de estado finito y autómatas de pila), manejar la programación para la solución de aplicaciones, aplicar las estructuras de datos en la solución de problemas, manipular las operaciones básicas de los archivos.				Traductores de lenguajes I Teoría de la Computación			Ninguno	

**2. Presentación**

La presente unidad tiene la finalidad de desarrollar las capacidades necesarias para comprender las bases teóricas para la construcción e implementación de un compilador. Así como Comprender los algoritmos que se utilizan para resolver cada una de las fases del compilador. Al mismo tiempo el dominar completamente el proceso para convertir un lenguaje de alto nivel a un lenguaje de bajo nivel.



### 3. Competencia general (Unidad de competencia)

Analiza los conceptos básicos para la construcción de un compilador permitiendo diseñar el proceso que interpretación de un lenguaje de alto nivel a un lenguaje de bajo nivel.

### 4. Elementos de competencia

A. Comprende el funcionamiento del análisis léxico, el tipo de gramáticas que debe reconocer y la forma en la que se puede implementar.		
Requisitos		
Cognitivos: (Contenidos).	Procedimentales:	Actitudinales:
Conceptualiza los conceptos de un compilador y procesadores de lenguajes. Conoce y comprende conceptos como: <ul style="list-style-type: none"><li>• Estructura del compilador.</li><li>• Funciones del analizador léxico</li><li>• Componentes léxicos, patrones y lexemas</li><li>• Determinación de los componentes léxicos mediante autómatas finitos.</li><li>• Tabla de tokens</li><li>• Errores léxicos</li><li>• Generadores de analizadores Léxicos</li></ul>	Aplica procedimientos de : <ul style="list-style-type: none"><li>• Comprensión</li><li>• Análisis</li><li>• Abstracción</li><li>• Síntesis</li><li>• Clasificación</li></ul> Que le permitan identificar y explicar la forma en la que se construye un analizador léxico. Además de la representación de forma lógica el funcionamiento de un autómata finito, mediante un grafo.	<ul style="list-style-type: none"><li>• Acepta y respeta las opiniones de los demás.</li><li>• Colabora con sus compañeros con la finalidad de mejorar el trabajo en equipo.</li><li>• Muestra interés al aprendizaje continuo y autogestivo.</li><li>• Valora la retroalimentación grupal.</li></ul>
Estrategias didácticas:	Recursos requeridos	Tiempo estimado:
Estrategias para indagar sobre conocimientos previos.	-Bibliografía básica -Antología	4 semanas



Estrategias que promuevan la comprensión mediante la organización de información. Exposición por parte del profesor. Investigación y participación de los alumnos	-Cañón, laptop, plataforma virtual. -Herramientas de Web 2.0	
<b>Criterios de desempeño:</b>	<b>Evidencias:</b>	<b>Producto esperado:</b>
-ejercicios resueltos a partir de operaciones de lenguajes, representación matemática de lenguajes y expresiones regulares. -Reportes de investigación de diversas fuentes de información.	- Cuadernillo de ejercicios -Reportes de investigación. -Exámenes	Reportes de lecturas, ejercicios y autómata del analizador léxico.

<b>B. Comprende el funcionamiento del analizador sintáctico, sus gramáticas y la forma de implementarlo en un lenguaje de programación.</b>		
<b>Requisitos</b>		
<b>Cognitivos: (Contenidos).</b>	<b>Procedimentales:</b>	<b>Actitudinales:</b>
Gramáticas libres de contexto Árboles de derivación. Análisis sintáctico descendente Análisis sintáctico ascendente Diagramas de sintaxis Eliminación de la ambigüedad. Manejo de errores	Aplica procedimientos de : Comprensión, análisis, abstracción, síntesis, Clasificación. Que le permiten utilizar los métodos adecuadamente para la construcción de un autómata, lenguaje regular y reconoce la gramática que genera, así como las expresiones regulares que representan. Selecciona la mejor metodología para la construcción de gramáticas adecuadas para la generación del analizador sintáctico, así como el tipo de análisis más adecuado.	<ul style="list-style-type: none"> <li>• Acepta y respeta las opiniones de sus compañeros.</li> <li>• Colabora con sus compañeros con la finalidad de mejorar el trabajo en equipo.</li> <li>• Muestra interés al aprendizaje continuo y autogestivo.</li> <li>• Valora la retroalimentación grupal.</li> </ul>



	Determina los errores que debe manejar en analizador sintáctico.	
<b>Estrategias didácticas:</b>	<b>Recursos requeridos</b>	<b>Tiempo estimado:</b>
Estrategias para indagar sobre conocimientos previos. Estrategias que promuevan la comprensión mediante la organización de información. Exposición por parte del profesor. Investigación y participación de los alumnos	--Bibliografía básica -Antología -Cañón, laptop, plataforma virtual. -Herramientas de Web 2.0	6 semanas
<b>Criterios de desempeño:</b>	<b>Evidencias:</b>	<b>Producto esperado:</b>
-ejercicios. -Reportes de investigación de diversas fuentes de información.	- Cuadernillo de ejercicios -Reportes de investigación. -Exámenes	Cuadernillo de ejercicios resueltos. Representación animada de un problema real mediante un autómata finito.

<b>C. Comprende la diferencia entre los tipos de analizadores semánticos para diseñar la verificación de tipos utilizando los árboles de expresiones y la tabla de símbolos de las expresiones dirigidas por la sintaxis de un analizador semántico.</b>		
<b>Requisitos</b>		
<b>Cognitivos: (Contenidos).</b>	<b>Cognitivos: (Contenidos).</b>	<b>Cognitivos: (Contenidos).</b>
Arboles de expresiones y acciones semánticas de un analizador sintáctico. Tipos de analizadores semánticos Expresiones de tipos. Manejo de errores semánticos.	Aplica procedimientos de : Comprensión, análisis, abstracción, síntesis, Clasificación. Que le permitirán diferenciar los tipos de analizadores semánticos, la verificación de tipos utilizando los árboles de expresiones.	<ul style="list-style-type: none"> <li>• Acepta y respeta las opiniones de sus compañeros.</li> <li>• Colabora con sus compañeros con la finalidad de mejorar el trabajo en equipo.</li> <li>• Muestra interés al aprendizaje continuo y autogestivo.</li> <li>• Valora la retroalimentación grupal.</li> </ul>
<b>Estrategias didácticas:</b>	<b>Recursos requeridos</b>	<b>Tiempo estimado:</b>



Estrategias para indagar sobre conocimientos previos. Estrategias que promuevan la comprensión mediante la organización de información. Exposición por parte del profesor. Investigación y participación de los alumnos	-Bibliografía básica -Antología -Cañón, laptop, plataforma virtual. -Herramientas de Web 2.0	3 semanas
<b>Criterios de desempeño:</b>	<b>Evidencias:</b>	<b>Producto esperado:</b>
-Ejercicios -Reportes de investigación de diversas fuentes de información.	- Cuadernillo de ejercicios -Reportes de investigación. -Exámenes	Cuadernillo de ejercicios resueltos.

<b>D.</b> Comprender la implementación programas en código objeto, para después aprender a traducir código de un lenguaje de alto nivel a código intermedio		
<b>Requisitos</b>		
<b>Cognitivos: (Contenidos).</b>	<b>Cognitivos: (Contenidos).</b>	<b>Cognitivos: (Contenidos).</b>
Introducción Notaciones Representaciones de código intermedio 1. Código P 2. Triplos 3. Cuádruplos Esquemas e generación Traducción de expresiones. Código ensamblador.	Aplica procedimientos de : Comprensión, análisis, abstracción, síntesis, Clasificación. Que le permitan comprender los conceptos necesarios para convertir una expresión aritmética en notación prefija y postfija. Comprender los conceptos para convertir el programa objeto a un código intermedio.	<ul style="list-style-type: none"> <li>• Acepta y respeta las opiniones de sus compañeros.</li> <li>• Colabora con sus compañeros con la finalidad de mejorar el trabajo en equipo.</li> <li>• Muestra interés al aprendizaje continuo y autogestivo.</li> <li>• Valora la retroalimentación grupal.</li> </ul>
<b>Estrategias didácticas:</b>	<b>Recursos requeridos</b>	<b>Tiempo estimado:</b>
Estrategias para indagar sobre conocimientos	--Bibliografía básica	3 semanas



previos. Estrategias que promuevan la comprensión mediante la organización de información. Exposición por parte del profesor. Investigación y participación de los alumnos	-Antología -Cañón, laptop, plataforma virtual. -Herramientas de Web 2.0	
<b>Criterios de desempeño:</b>	<b>Evidencias:</b>	<b>Producto esperado:</b>
-ejercicios . -Reportes de investigación de diversas fuentes de información.	- Cuadernillo de ejercicios -Reportes de investigación. -Exámenes	Cuadernillo de ejercicios resueltos.

E. Conoce e identifica los diferentes tipos de optimización al código generado por el compilador.		
<b>Requisitos</b>		
<b>Cognitivos: (Contenidos).</b>	<b>Cognitivos: (Contenidos).</b>	<b>Cognitivos: (Contenidos).</b>
Tipos de optimizaciones: 1. Optimización de mirilla. 2. A partir de expresiones locales 3. A partir de bucles 4. Optimización Global Generación de código óptimo para expresiones.	Aplica procedimientos de : Comprensión, análisis, abstracción, síntesis, Clasificación. Que le permita analizar los diversos tipos de optimización. Aplicar las técnicas para transformar el código intermedio en código optimización.	<ul style="list-style-type: none"> <li>• Acepta y respeta las opiniones de sus compañeros.</li> <li>• Colabora con sus compañeros con la finalidad de mejorar el trabajo en equipo.</li> <li>• Muestra interés al aprendizaje continuo y autogestivo.</li> <li>• Valora la retroalimentación grupal.</li> </ul>
<b>Estrategias didácticas:</b>	<b>Recursos requeridos</b>	<b>Tiempo estimado:</b>
Estrategias para indagar sobre conocimientos previos. Estrategias que promuevan la comprensión	--Bibliografía básica -Antología -Cañón, laptop, plataforma virtual.	3 semanas



mediante la organización de información. Exposición por parte del profesor. Investigación y participación de los alumnos	-Herramientas de Web 2.0	
<b>Criterios de desempeño:</b>	<b>Evidencias:</b>	<b>Producto esperado:</b>
-Ejercicios -Reportes de investigación de diversas fuentes de información.	-Cuadernillo de ejercicios -Reportes de investigación. -Exámenes	Cuadernillo de ejercicios resueltos.

**b- Evaluación y acreditación**

Área de conocimiento: a) Reportes de lecturas (20%) b) Examen departamental (20%) c) Exámenes parciales (20%) Área de habilidades y destrezas: a) actividades de aprendizaje (cuaderno de ejercicio) 10% b) Actividades integradoras 20% c) Portafolio 5% Área de actitud: d) Participación 5% (Podrá agregar cuantos elementos requiera en cada uno de los apartados de evaluación)
--

**c- Bibliografía**

Ruiz Catalán, Jacinto. (2010). Compiladores: teoría e implementación. Primera edición. México: Alfaomega. Alfonseca Moreno, Manuel(2006) Compiladores e intérpretes : teoría y práctica.Madrid. Pearson Cantú Treviño Thelma(2015) Teoría de autómatas un enfoque práctico. México. Pearson
---



**d- Perfil docente**

El docente de esta materia deberá ser un profesionalista con formación en las áreas de la computación, comunicaciones o informática; capaz de motivar a la investigación y creación de conocimiento, con habilidades para transmitir sus conocimientos y enseñar de forma interactiva propiciando en los alumnos el auto-aprendizaje.

Vo.Bo Dr. Juan Jorge Rodríguez Bautista

Jefe del departamento

Vo.Bo Mtra. María Obdulia González

Fernández

Presidente de Academia