



1. Identificación del curso

Compiladores							
Programa educativo				Departamento de adscripción			
Ingeniería en Computación				Departamento de Ingenierías			
Área de formación				Tipo de Unidad de Aprendizaje			
Básico particular obligatoria							
Carga horaria					Créditos		Clave
Teoría	40	Práctica	40	Total	80	8	IL359
Modalidad de Enseñanza - Aprendizaje				Prerrequisito			
Presencial				Teoría de la computación			
Academia				Profesor responsable			
Ciencias Computacionales							
Elaboró / Modificó				Fecha de elaboración / modificación			
María Obdulia González Fernández/ Héctor González Sánchez				15 de septiembre de 2025			

2. Competencias que abonan al perfil de egreso

Transversal	Disciplinaria	Profesional
Posee habilidades de trabajo en equipo que le permiten desarrollarse como líder de proyectos en su campo profesional o integrarse a un grupo ya establecido. Reconoce sus responsabilidades éticas y profesionales para actuar con rigor en su desarrollo como ingeniero	Posee capacidad de razonamiento crítico, lógico y matemático para resolver problemas dentro de su área de estudio a través de modelos abstractos que reflejan situaciones reales. Posee saberes, conceptos, principios y teorías relacionadas con las ciencias computacionales y a sus disciplinas afines	Se formará con ética y responsabilidad, en búsqueda de la calidad y la innovación tecnológica en las organizaciones. Podrá trabajar en equipo, con liderazgo y una visión emprendedora para aportar soluciones de ingeniería en los contextos global, económico, sustentable y social

3. Saberes previos

Fundamentos de teoría de la computación y estructuras de datos

4. Presentación de la unidad de aprendizaje

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.

5. Objetivo de aprendizaje

El alumno construirá un compilador que traduzca un lenguaje de programación a otro realizando el análisis léxico, sintáctico, semántico y generación de código.

6. Competencia general de la unidad de aprendizaje

CE.CC.200 Capacidad para conocer los fundamentos teóricos de los lenguajes de programación y las técnicas de procesamiento léxico, sintáctico y semántico asociadas, y saber aplicarlas para la creación, diseño y procesamiento de lenguajes. (BOE/SFIA CE.CC.200).



7. Habilidades, valores y actitudes

Liderazgo, respeto a las opiniones de los demás, colaboración y trabajo en equipo. Muestra interés en el aprendizaje continuo. Valora la retroalimentación grupal

8. Elementos de competencias

Bloque. No. I: Analizador léxico		
Sub-competencia	Comprende el funcionamiento del análisis léxico, el tipo de gramáticas que debe reconocer y la forma en la que se puede implementar	
Cognitivos (Contenido)		
<ul style="list-style-type: none">• Estructura del compilador• Funciones del analizador léxico• Componentes léxicos, patrones y lexemas• Determinación de los componentes léxicos mediante autómatas finitos.• Tabla de tokens• Errores léxicos• Generadores de analizadores Léxicos		
Procedimentales		
Aplica los conocimientos para identificar y explicar la forma en la que se construye un analizador léxico		
Estrategias didácticas		
Exposición por parte del profesor y de los alumnos		
Estrategias para motivar el trabajo en equipo y estrategias de aprendizaje activo		
Criterios de desempeño	Producto esperado	Sesiones estimadas
Orden, limpieza, puntualidad en la entrega Presentar trabajos de investigación con información clara y concisa	Ejercicios resueltos a partir de operaciones de lenguajes, representación matemática de lenguajes y expresiones regulares. -Reportes de investigación de diversas fuentes de información. -Proyecto final de bloque: Analizador léxico capaz de identificar y clasificar los componentes léxicos de un lenguaje de programación mediante el uso de autómatas finitos deterministas.	16
Área de conocimiento	6.3 Sistemas de software	

Bloque No. II: Analizador sintáctico

Sub-competencia	Comprende el funcionamiento del analizador sintáctico, sus gramáticas y la forma de implementarlo en un lenguaje de programación.
Cognitivos (Contenido)	
<ul style="list-style-type: none">• Gramáticas libres de contexto• Árboles de derivación.• Análisis sintáctico descendente	



CENTRO UNIVERSITARIO DE LOS ALTOS

División de Ciencias Agropecuarias e Ingenierías

Departamento de Ciencias Pecuarias y Agrícolas

Departamento de Ingenierías

- Análisis sintáctico ascendente
- Diagramas de sintaxis
- Eliminación de la ambigüedad.
- Manejo de errores

Procedimentales

Utiliza algoritmos 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.

Estrategias didácticas

Exposición por parte del profesor y de los alumnos

Estrategias para motivar el trabajo en equipo y estrategias de aprendizaje activo

Criterios de desempeño	Producto esperado	Sesiones estimadas
Orden, limpieza, puntualidad en la entrega. Presentar trabajos de investigación con información clara y concisa. Programación de ejercicios.	Ejercicios. Reportes de investigación de diversas fuentes de información. Construcción de un analizador sintáctico que, valide la estructura gramatical de un lenguaje de programación a partir de una gramática libre de contexto.	22
Área de conocimiento	6.3 Sistemas de software	

Bloque No. III: Analizador Semántico

Sub-competencia	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.
-----------------	--

Cognitivos (Contenido)

- Árboles de expresiones y acciones semánticas de un analizador sintáctico.
- Tipos de analizadores semánticos
- Expresiones de tipos.
- Manejo de errores semánticos.

Procedimentales

Aplica los principios de los analizadores semánticos para clasificar los analizadores semánticos, y la verificación de tipos utilizando los árboles de expresiones.

Estrategias didácticas

Exposición por parte del profesor

Resolución de problemas/ejercicios

Trabajo colaborativo

Criterios de desempeño	Producto esperado	Sesiones estimadas
Orden, limpieza, puntualidad en la entrega Presentar trabajos de investigación con información clara y concisa	Ejercicios Reportes de investigación Producto final de bloque : Reporte de la simulación del funcionamiento de un analizador semántico que	18



CENTRO UNIVERSITARIO DE LOS ALTOS

División de Ciencias Agropecuarias e Ingenierías

Departamento de Ciencias Pecuarias y Agrícolas

Departamento de Ingenierías

	verifique la coherencia lógica y los tipos de datos de un lenguaje de programación previamente analizado en las fases léxica y sintáctica.	
Área de conocimiento	7.3.3 Desarrollo e implantación.	

Bloque No. IV: Generación y optimización de código

Sub-competencia	Simula la generación de código intermedio, optimizado y traducido a lenguaje máquina o de bajo nivel.	
Cognitivos (Contenido)		
<ul style="list-style-type: none">• Representación intermedia (cuádruplas, árboles de sintaxis abstracta)• Optimización de código• Generación de código ensamblador o de máquina• Tipos de analizadores semánticos• Expresiones de tipos.• Manejo de errores semánticos.		
Procedimentales		
<ul style="list-style-type: none">• Implementa la traducción de expresiones a código intermedio.• Aplica técnicas básicas de optimización.		
Estrategias didácticas		
Exposición por parte del profesor Resolución de problemas/ejercicios Trabajo colaborativo		
Criterios de desempeño	Producto esperado	Sesiones estimadas
Orden, limpieza, puntualidad en la entrega Presentar trabajos de investigación con información clara y concisa	Ejercicios Reportes de investigación	24
Área de conocimiento		

Nota: 1 sesión = 1 hora;

9. Recursos requeridos

Videoproyector, computadora, paquetería de Ofimática, programas.

10. Evaluación y acreditación de la unidad de aprendizaje

- Actividades de Investigación 25%
- Actividades prácticas 30%
- Proyecto 20%
- Participación 5%



CENTRO UNIVERSITARIO DE LOS ALTOS

División de Ciencias Agropecuarias e Ingenierías
Departamento de Ciencias Pecuarias y Agrícolas
Departamento de Ingenierías

11. Referencias (APA)

Básica

Mak, R. (2021). Writing compilers and interpreters: A software engineering approach (5th ed.). Mercury Learning and Information
Sandler, N. (2024). Writing a C compiler: Build a real programming language from scratch. No Starch Press.
Thain, D. (2020). Introduction to compilers and language design (2nd ed.). Independently published.

Complementaria

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

Sitios web

Appel, A. W. (s.f.). Modern compiler implementation in C. Princeton University. Recuperado el 1 de noviembre de 2025] de <http://www.cs.princeton.edu/~appel/modern>


12. Campo de aplicación profesional


Diseño e implementación de nuevos lenguajes de propósito específico o general.
Creación de herramientas que traducen y optimicen código fuente para diferentes plataformas o arquitecturas.
Implementación de técnicas para mejorar la eficiencia del código generado y el uso de recursos del sistema.

13. Perfil docente

El docente de esta materia deberá ser un profesionalista con formación en las áreas de las ciencias computacionales con enfoque en programación; 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 autoaprendizaje.




Dr. Cesar Eduardo Aceves Aldrete
Jefe de departamento de Ingenierías


Mtro. Héctor González Sánchez
Presidente de la academia