

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO
LICENCIATURA DE INGENIERÍA EN COMPUTACIÓN



PROGRAMA DE ESTUDIOS
Compiladores

Elaboró:	Dra. Nely Plata Cesar	Facultad de Ingeniería
	Ing. Héctor Torres Aguilar	Facultad de Ingeniería
	Dr. José Raymundo Marcial Romero	Facultad de Ingeniería
	M.C.C. Enrique José Tinajero Pérez	C.U. UAEM Ecatepec
	M. en C. José Jair Vázquez Palma	C.U. UAEM Texcoco

Fecha de
aprobación:

H. Consejo Académico

07 de junio de 2021

H. Consejo de Gobierno

09 de junio de 2021

Facultad de Ingeniería

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO
FACULTAD DE INGENIERÍA
DIRECCIÓN ACADÉMICA

09 JUN 2021

CONSEJOS ACADÉMICO Y DE GOBIERNO
DICTAMEN: APROBADO



Índice

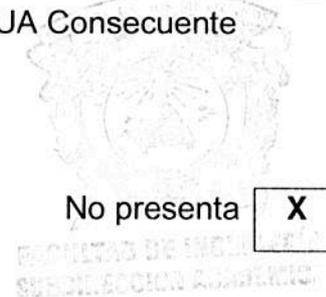
	Pág.
I. Datos de identificación.	3
II. Presentación del programa de estudios.	4
III. Ubicación de la unidad de aprendizaje en el mapa curricular.	5
IV. Objetivos de la formación profesional.	7
V. Objetivos de la unidad de aprendizaje.	8
VI. Contenidos de la unidad de aprendizaje y su organización.	9
VII. Acervo bibliográfico.	11





I. Datos de identificación.

Espacio académico donde se imparte	Facultad de Ingeniería Centro Universitario UAEM Atlacomulco Centro Universitario UAEM Ecatepec Centro Universitario UAEM Texcoco Centro Universitario UAEM Valle de Chalco Centro Universitario UAEM Valle de México Centro Universitario UAEM Valle de Teotihuacán Centro Universitario UAEM Zumpango Unidad Académica Profesional Tianguistenco				
Estudios profesionales	Licenciatura de Ingeniería en Computación, 2019				
Unidad de aprendizaje	Compiladores	Clave	LINC23		
Carga académica	3	1	4	7	
	Horas teóricas	Horas prácticas	Total de horas	Créditos	
Carácter	Obligatoria	Tipo	Curso	Periodo escolar	Sexto
Área curricular	Ingeniería Aplicada y Diseño de Ingeniería		Núcleo de formación	Sustantivo	
Seriación	Ninguna		Ninguna		
	UA Antecedente		UA Consecuente		
Formación común	No presenta <input checked="" type="checkbox"/>				





II. Presentación del programa de estudios.

El uso de lenguajes de programación es una tarea cotidiana de cualquier Ingeniero en Computación. Esta actividad debe ser complementada con la enseñanza de los diferentes paradigmas de lenguajes de programación para tener un amplio criterio de estos, así como de la elección adecuada para realizar una aplicación.

El propósito de esta unidad de aprendizaje es presentar las fases que tienen lugar en el desarrollo de un compilador, esto es con la finalidad de que el alumno pueda identificar y ser capaz de realizar algunas de estas fases para el desarrollo de un compilador. Así mismo, en este curso, el profesor hace una presentación de las herramientas que se utilizan para la generación de un compilador.

La unidad de aprendizaje compiladores está estructurada por cinco unidades temáticas en las cuales se describe el proceso de traducción de un lenguaje de programación escrito en un lenguaje de alto nivel a código intermedio. En la unidad temática uno se describen las características más importantes del proceso de compilación, sus fases y los elementos principales para su construcción. En la unidad dos se describe la fase del análisis léxico, en la cual se identifican los tokens. En la unidad tres se describe el análisis sintáctico el cual se basa en una gramática o serie de reglas que permiten crear sentencias coherentes para el lenguaje de programación. En la unidad cuatro se describe el análisis semántico en el que se analiza la estructura de las frases y su coherencia entre sí para formar un árbol semántico. En la unidad cinco se describe la generación de código intermedio el cual permite traducir sentencias de lenguaje de alto nivel a lenguaje entendible por la máquina que, en algunos casos, suele ser la máquina virtual.

De esta forma, el Ingeniero en Computación debe conocer que, de entre todas las herramientas que hace uso para la programación de lenguajes están los compiladores.

La unidad de aprendizaje pertenece al núcleo de formación sustantivo, por lo que pretende desarrollar en el alumno el dominio teórico en cuanto respecta al diseño de compiladores, las fases por las que pasa el código fuente antes de ser convertido en código intermedio.

La unidad de aprendizaje de compiladores contribuye al perfil de egreso del Ingeniero en Computación permitiendo la creación de nuevos prototipos de compiladores.





Proyecto curricular de la Licenciatura de Ingeniería en Computación
Reestructuración, 2019
Secretaría de Docencia • Dirección de Estudios Profesionales



DISTRIBUCIÓN DE LAS UNIDADES DE APRENDIZAJE OPTATIVAS																	
PERIODO 1	PERIODO 2	PERIODO 3	PERIODO 4	PERIODO 5	PERIODO 6	PERIODO 7	PERIODO 8	PERIODO 9	PERIODO 10								
							<table border="1"> <tr><td>1</td></tr> <tr><td>3</td></tr> <tr><td>4</td></tr> <tr><td>5</td></tr> </table> Análisis y diseño de redes	1	3	4	5	<table border="1"> <tr><td>1</td></tr> <tr><td>3</td></tr> <tr><td>4</td></tr> <tr><td>5</td></tr> </table> Gestión de redes	1	3	4	5	
1																	
3																	
4																	
5																	
1																	
3																	
4																	
5																	
								<table border="1"> <tr><td>1</td></tr> <tr><td>3</td></tr> <tr><td>4</td></tr> <tr><td>5</td></tr> </table> Computing in industry ¹	1	3	4	5					
1																	
3																	
4																	
5																	
							<table border="1"> <tr><td>1</td></tr> <tr><td>3</td></tr> <tr><td>5</td></tr> </table> Visión artificial	1	3	5	<table border="1"> <tr><td>1</td></tr> <tr><td>3</td></tr> <tr><td>4</td></tr> <tr><td>5</td></tr> </table> Interacción hombre-máquina	1	3	4	5		
1																	
3																	
5																	
1																	
3																	
4																	
5																	
								<table border="1"> <tr><td>1</td></tr> <tr><td>3</td></tr> <tr><td>4</td></tr> <tr><td>5</td></tr> </table> Tecnologías emergentes	1	3	4	5					
1																	
3																	
4																	
5																	
							<table border="1"> <tr><td>1</td></tr> <tr><td>3</td></tr> <tr><td>5</td></tr> </table> Reconocimiento de patrones	1	3	5	<table border="1"> <tr><td>1</td></tr> <tr><td>3</td></tr> <tr><td>4</td></tr> <tr><td>5</td></tr> </table> Tópicos de tecnologías de datos	1	3	4	5		
1																	
3																	
5																	
1																	
3																	
4																	
5																	
								<table border="1"> <tr><td>1</td></tr> <tr><td>3</td></tr> <tr><td>4</td></tr> <tr><td>5</td></tr> </table> Sistemas interactivos	1	3	4	5					
1																	
3																	
4																	
5																	

PARAMETROS DEL PLAN DE ESTUDIOS

Unidad de aprendizaje	HT: Horas Teóricas	HP: Horas Prácticas	TH: Total de Horas	CR: Créditos
Núcleo básico obligatorio cursar y acreditar 20 UA	56	24	80	136
Núcleo sustantivo obligatorio cursar y acreditar 27 UA	70	40	110	180
Núcleo integral obligatorio cursar y acreditar 4 UA + 2*	8	28**	36**	37
Núcleo integral optativo cursar y acreditar 3 UA	3	9	12	15

→ 18 horas de práctica.
Créditos mínimos 21 y máximos 51 por período escolar.
* Actividad académica.
** Las horas de la actividad académica.
|| UA optativa que debe impartirse, cursarse y acreditarse en el idioma inglés.

Núcleo básico obligatorio
Núcleo básico optativo
Núcleo sustantivo obligatorio
Núcleo integral obligatorio
Núcleo integral optativo

Total del núcleo básico sustantivo: acreditar 20 UA para cubrir 136 créditos

Total del núcleo sustantivo: acreditar 27 UA para cubrir 180 créditos

Total del núcleo integral: acreditar 11 UA + 2* para cubrir de 94 créditos

TOTAL DEL PLAN DE ESTUDIOS	
UA obligatorias	55 + 2 Actividades académicas
UA optativas	3
UA a acreditar	58 + Actividades académicas
Créditos	410



IV. Objetivos de la formación profesional.

Objetivos del programa educativo:

Son objetivos de la Licenciatura de Ingeniería en Computación formar profesionistas que sean capaces de proveer soluciones computacionales innovadoras y sustentables a los problemas, requerimientos y necesidades específicas de la sociedad con responsabilidad ética y mediante la aplicación de metodologías y normas adecuadas en el desarrollo, implantación, optimización, administración y mantenimiento de sistemas de cómputo, que impliquen el uso o la integración de hardware, software y comunicación en diferentes plataformas y dispositivos y desarrollar los aprendizajes y competencias para:

Generales

- Ejercer el diálogo y el respeto como principios de la convivencia con sus semejantes, y de apertura al mundo.
- Reconocer la diversidad cultural y disfrutar de sus bienes y valores.
- Convivir con las reglas de comportamiento socialmente aceptables, y contribuir en su evolución.
- Adquirir los valores de cooperación y solidaridad.
- Cuidar su salud y desarrollar armoniosamente su cuerpo; ejercer responsablemente y de manera creativa el tiempo libre.
- Ampliar su universo cultural para mejorar la comprensión del mundo y del entorno en que vive, para cuidar de la naturaleza y potenciar sus expectativas.
- Participar activamente en su desarrollo académico para acrecentar su capacidad de aprendizaje y evolucionar como profesional con autonomía.
- Asumir los principios y valores universitarios, y actuar en consecuencia.
- Emplear habilidades lingüístico-comunicativas del idioma inglés.
- Evaluar el progreso, integración e incertidumbre de las ciencias, ante la creciente complejidad de las profesiones.

Particulares

- Crear proyectos de sistemas computacionales a través de la identificación de necesidades, metodologías ad hoc, teorías de la computación, empleo de sistemas de programación, sistemas electrónicos, comunicaciones y de sistemas, señales y control, para mejorar la cobertura y calidad de los servicios de cómputo de la sociedad y en sectores prioritarios como la educación, salud y seguridad social.





- Evaluar redes de cómputo a través del análisis, el diseño y la administración de la interconexión de dispositivos en redes de computadoras de área local y abierta, considerando estándares y modelos internacionales, para garantizar el rendimiento óptimo en la transmisión de datos.
- Crear nuevas tecnologías computacionales, empleando tecnologías emergentes tales como la inteligencia artificial, la visión computacional, el reconocimiento de patrones, la graficación por computadora, los sistemas embebidos y la ciencias de los datos; para resolver problemas específicos de la sociedad y en sectores prioritarios como la educación, salud y seguridad social.

Objetivos del núcleo de formación:

Desarrollará en el alumno el dominio teórico, metodológico y axiológico del campo de conocimientos donde se inserta la profesión.

Comprenderá unidades de aprendizaje sobre los conocimientos, habilidades y actitudes necesarias para dominar los procesos, métodos y técnicas de trabajo; los principios disciplinares y metodológicos subyacentes; y la elaboración o preparación del trabajo que permita la presentación de la evaluación profesional.

Objetivos del área curricular o disciplinaria:

Evaluar sistemas computacionales empleando paradigmas como la teoría de la computación, la teoría de juegos, la teoría de las bases de datos, las comunicaciones de datos, la ingeniería de software, tecnologías de vanguardia y arquitecturas de hardware, para optimizar la cobertura y calidad de los servicios de cómputo en las organizaciones de los sectores prioritarios como la educación, salud y seguridad social, así como la creación de aplicaciones específicas.

V. Objetivos de la unidad de aprendizaje.

Diseñar un compilador mediante la definición de una gramática y la aplicación de las fases de análisis lexicográfico y sintáctico para el análisis de código fuente y la generación de código objeto para traducir código de alto nivel a código de bajo nivel que sea ejecutable por un equipo de cómputo o sistema embebido.





VI. Contenidos de la unidad de aprendizaje, y su organización.

Unidad temática 1. Fases de un compilador

Objetivo: Examinar las fases de la metodología de compilación, con base en estructuras discretas, para la creación de lenguajes de programación.

Temas:

- 1.1 El proceso de compilación.
- 1.2 Fases principales de un compilador: Análisis léxico, análisis sintáctico, análisis semántico, generación de código intermedio y optimización de código.
- 1.3 Programas y herramientas relacionados con un compilador.

Unidad temática 2. Análisis léxico

Objetivo: Construir un analizador léxico, utilizando técnicas de desarrollo de autómatas finitos y expresiones regulares, para la separación de tokens de un programa.

Temas:

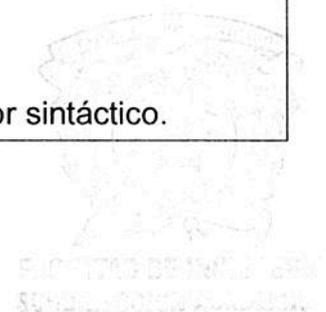
- 2.1 Proceso de análisis léxico.
- 2.2 Funciones del analizador léxico.
- 2.3 Expresiones regulares.
- 2.4 Autómatas finitos (deterministas y no deterministas).
- 2.5 Diseño de un generador de análisis léxico.
- 2.6 Herramientas para generar automáticamente un analizador léxico.

Unidad temática 3. Análisis sintáctico

Objetivo: Construir un analizador sintáctico, utilizando gramáticas libres de contexto, para verificar la estructura de un programa.

Temas:

- 3.1 Proceso de análisis sintáctico.
- 3.2 Gramáticas independientes del contexto.
- 3.3 Análisis sintáctico descendente.
- 3.4 Análisis sintáctico ascendente.
- 3.5 Gramáticas ambiguas.
- 3.6 Herramientas para generar automáticamente un analizador sintáctico.





Unidad temática 4. Análisis semántico

Objetivo: Crear un analizador semántico, utilizando la tabla de símbolos y el árbol sintáctico, para construir un compilador básico.

Temas:

- 4.1 Análisis semántico.
- 4.2 La tabla de símbolos.
- 4.3 Árbol sintáctico y árbol sintáctico abstracto.
- 4.4 Tipos de datos y verificación de tipos.
- 4.5 Creación de un analizador semántico para un lenguaje básico.

Unidad temática 5. Generación de código intermedio.

Objetivo: Construir la representación intermedia de un compilador básico, utilizando la estructura de un lenguaje ensamblador, para la generación de código intermedio de un programa.

Temas:

- 5.1 Técnicas básicas de generación de código.
- 5.2 Código intermedio.
- 5.3 Generación de código.





VII. Acervo bibliográfico

Básico

Aho, A.V., SethiR., Ullman, D.J., (2008), *Compiladores Principios Técnicas y herramientas*, 2da. Edición, Addison Wesley.

Appel, A.W., Palsberg, J., (2002), *Modern Compiler implementation in Java*, 2da. Edición, Cambridge University Press.

Grune, D., Reeuwijk, K. V., Bal, H., Jacobs C. J. H., (2012), *Modern Compiler Design*, 2da. Edición, Springer.

Louden, K.C., (1997), *Construcción de Compiladores principios y práctica*, Internacional Thomson Editores.

Pittman, T., Peters, J., (1992), *The Art of Compiler Design; Theory and Practice*, New Jersey USA: Prentice Hall.

Wilhelm, R., Seidl, H., (2013), *Compiler Design: Syntactic and Semantic Analysis*, Springer-Verlag Berlin Heidelberg.

Complementario:

Bennett, J. P., (1996), *Introduction to Compiling Techniques. A first Course using Ansi C, LEX and YACC*, Mc. Graw-Hill. Book Company Europe.

Levine, J. R, Mason, T., Brown., D., (1992), *Lex and Yacc*, 2da. Edición, U.S.A: O'Reilly.

Mak, R., (2011), *Writing Compilers and Interpreters: A Software Engineering Approach*, 3ra. Edición, John Wiley & Sons.

