UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

LICENCIATURA DE INGENIERÍA EN COMPUTACIÓN

PROGRAMA DE ESTUDIOS

PROGRAMACIÓN II

Elaboró:

Dra. Mireya Salgado Gallegos	Facultad de Ingeniería
M. en C.A. Eduardo Trujillo Flores	Facultad de Ingeniería
M. en I. Sara Vera Noguez	Facultad de Ingeniería
M. en I. David Martínez Martínez	CU UAEM Teotihuacán
Dra. Ivonne Rodríguez Pérez	CU UAEM Valle de México

Fecha de aprobación:

H. Consejo Académico	H. Consejo de Gobierno
21 de marzo de 2019	21 de marzo de 2019

Facultad de Ingeniería

FACULTAD DE INGENIERIA

DIRECCIÓN DE ESTUDIOS PROFESIONALES

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 Teotihuacan** **Centro Universitario UAEM Zumpango**
Estudios profesionales	**Licenciatura de Ingeniería en Computación, 2019**
Unidad de aprendizaje	**Programación II**
Clave	

Carga académica				
	3	**1**	**4**	**7**
	Horas teóricas	Horas prácticas	Total de horas	Créditos

Carácter	**Obligatorio**	Tipo	**Curso**	Periodo escolar	**Segundo**
Área curricular	**Ciencias de la Ingeniería**			Núcleo de formación	**Sustantivo**

Seriación	**Programación I**	**Ninguna**
	UA Antecedente	UA Consecuente

Formación común

No presenta **X**

FACULTAD DE INGENIERIA

DIRECCIÓN DE ESTUDIOS PROFESIONALES

II. Presentación del programa de estudios.

En el ámbito profesional relacionado con el desarrollo de software, el Ingeniero en Computación debe ser capaz de proveer soluciones óptimas sustentadas en un adecuado análisis y diseño. Por lo tanto, una vez adquiridas las habilidades básicas de programación bajo el paradigma orientado a objetos en la unidad de aprendizaje Programación I, el alumno debe adentrarse en cuestiones de estructuras de datos y algorítmica con el fin de comenzar a analizar y diseñar programas de la mejor manera posible. En unidades de aprendizaje consecuentes, el discente continuará su formación hasta llegar a obtener mejores soluciones.

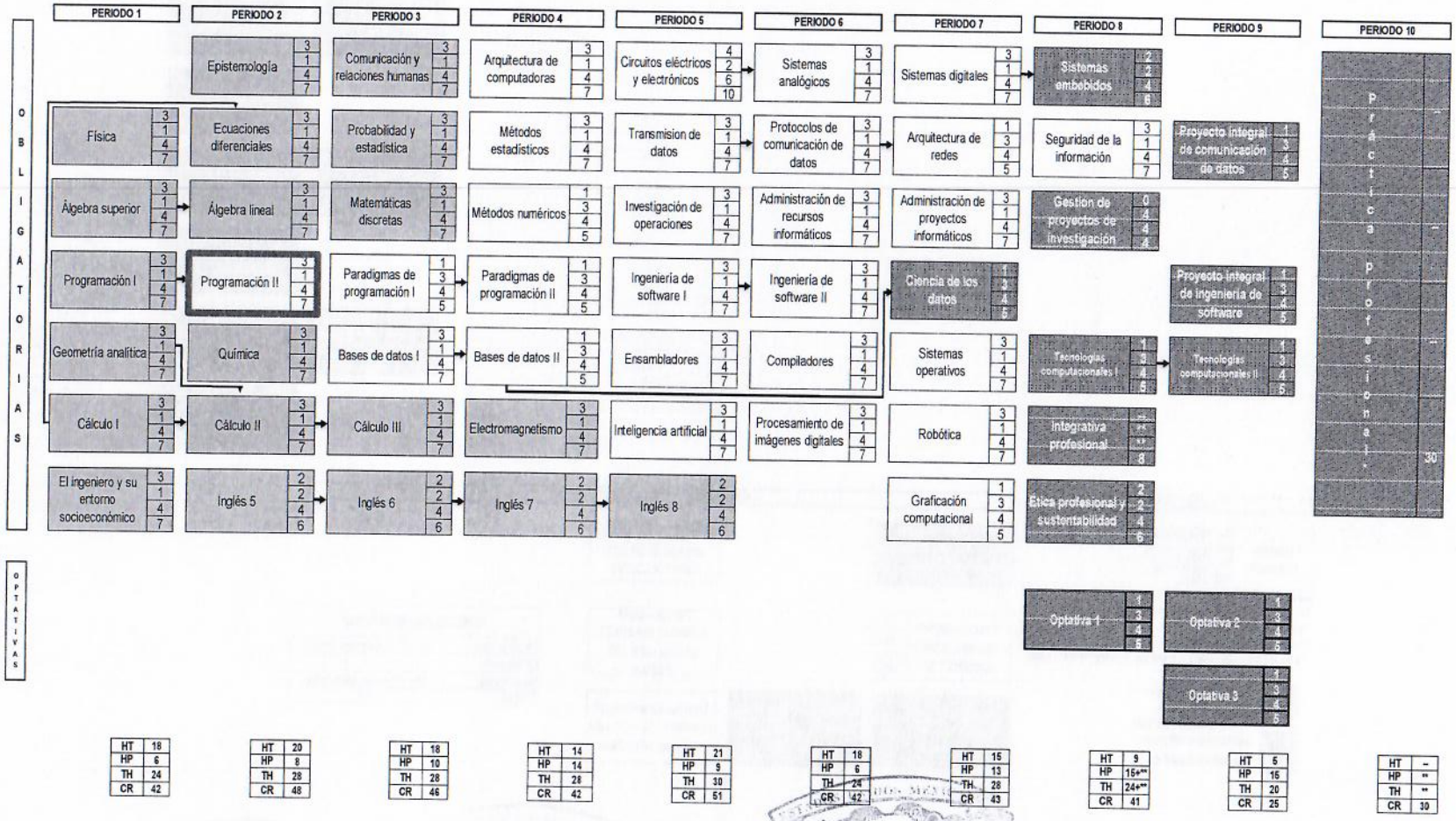
Esta formación permitirá al futuro ingeniero enfrentarse a retos de programación de alta complejidad con la certeza de poder no sólo dar una solución a un problema dado, sino de dar la solución óptima y ser capaz de evaluar que tan buena es la solución dada.

En la unidad de aprendizaje Programación II, el alumno conocerá y aplicará estructuras de datos clásicas como listas, pilas, colas, árboles binarios y grafos, comúnmente utilizadas para la solución de procesamiento de datos en problemas complejos. Asimismo, será capaz de organizar datos e información y lograr su persistencia en medios de almacenamiento secundario para su conservación, recuperación y procesamiento en cualquier momento. Con el propósito de que los datos puedan recuperarse, procesarse y presentarse en función de las necesidades planteadas, es importante conocer y manejar métodos de ordenamiento y búsqueda. Finalmente, conocerá técnicas de diseño como algoritmos voraces, algoritmos divide y vencerás, programación dinámica, algoritmos vuelta atrás, algoritmos ramifica y poda.

En este nivel del plan de estudios, el docente acompaña al discente en la apropiación de los conocimientos descritos para que los aplique en la solución de problemas de baja complejidad y obtenga programas lo más óptimo posible.

III. Ubicación de la unidad de aprendizaje en el mapa curricular

MAPA CURRICULAR DE LA LICENCIATURA DE INGENIERÍA EN COMPUTACIÓN, 2019



FACULTAD DE INGENIERÍA

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
							Análisis y diseño de redes (1, 3, 4, 5)	Gestión de redes (1, 3, 4, 5)	
								Computing in industry (1, 3, 4, 5)	
							Visión artificial (3, 4, 5)	Interacción hombre-máquina (1, 3, 4, 5)	
								Tecnologías emergentes (1, 3, 4, 5)	
							Reconocimiento de patrones (1, 3, 4, 5)	Tópicos de tecnologías de datos (1, 3, 4, 5)	
								Sistemas interactivos (1, 3, 4, 5)	

SIMBOLOGÍA

Unidad de aprendizaje	
	HT: Horas Teóricas
	HP: Horas Prácticas
	TH: Total de Horas
	CR: Créditos

→ 18 líneas de seriación.
Créditos mínimos 21 y máximos 51 por periodo 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

PARÁMETROS DEL PLAN DE ESTUDIOS

Núcleo básico obligatorio: cursar y acreditar 20 UA	
	56
	24
	80
	136

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

Núcleo sustantivo obligatorio: cursar y acreditar 27 UA	
	70
	40
	110
	180

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

Núcleo integral obligatorio: cursar y acreditar 8 UA + 2*	
	9
	28**
	38**
	79

Núcleo integral optativo: cursar y acreditar 3 UA	
	3
	9
	12
	15

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

FACULTAD DE INGENIERIA

DIRECCIÓN DE ESTUDIOS PROFESIONALES

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 conocimiento 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:

Analizar los fundamentos de las ciencias de la ingeniería mediante el estudio de las teorías de las ciencias de la computación, la ingeniería de software y programación, hardware y los sistemas electrónicos, las comunicaciones, los sistemas, señales y control que permita el desarrollo de tecnología.

V. Objetivos de la unidad de aprendizaje.

Construir programas orientados a objetos utilizando estructuras de datos lineales y no lineales, algoritmos de búsqueda y ordenamiento, persistencia de objetos y lenguaje de programación orientado a objetos con interfaces textuales de usuario, para la resolución de problemas simples mediante la arquitectura modelo-vista-controlador.

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

Unidad temática 1. Conceptos suplementarios de programación orientada a objetos
Objetivo: Revisar conceptos complementarios de programación orientada a objetos involucrados en el análisis y diseño de programas.
Temas: 1.1 Herencia simple y Modificadores de acceso protegidos 1.2 Polimorfismo de sobrecarga y sobre-escritura 1.3 Abstracción: clases abstractas y métodos virtuales 1.4 Iteradores y Colecciones de objetos

Unidad temática 2. Ordenamiento y búsqueda
Objetivo: Implementar programas de computadora para la solución de problemas que involucren el procesamiento de datos mediante recursividad, búsqueda y ordenamiento de datos.
Temas: 2.1 Recursividad directa 2.2 Algoritmos de ordenamiento 2.3 Algoritmos de búsqueda

Unidad temática 3. Estructuras de datos lineales y no lineales
Objetivo: Construir programas de computadora que incluyan estructuras de datos lineales mediante listas, pilas, colas y árboles, utilizando colecciones de objetos, para la solución de problemas que requieran el procesamiento de datos.
Temas: 3.1 Listas 3.2 Pilas 3.3 Colas 3.4 Árboles binarios (de búsqueda y balanceados) 3.5 Árboles multicamino (B, B+, B*)

Unidad temática 4. Manejo de archivos
Objetivo: Implementar programas de computadora utilizando persistencia de objetos, archivos secuenciales de texto y directos, para la solución de problemas que impliquen el procesamiento de datos.
Temas: 4.1 Archivos de texto 4.2 Archivos aleatorios 4.3 Persistencia de objetos

VII. Acervo bibliográfico.

Básico:

Brassard, G.; Bratley, P. (1997). "Fundamentos de Algoritmia", Ed. Prentice Hall.

Guardati Buemo Silvia, (2007). "Estructura De Datos Orientada A Objetos Algoritmos Con C++]", Pearson Educación.

Guardati Buemo Silvia. (2016). Estructuras de datos básicas programación orientada a objetos con java. Alfaomega Grupo Editor.

López Goytia, José Luis y Ángel Gutiérrez González. (2014). Programación orientada a objetos con C++ y Java. Un acercamiento disciplinario. Grupo Editorial Patria.

Oliet N. M. (2008). "Estructuras De Datos Y Métodos Algorítmicos", Prentice Hall /Pearson.

Complementario:

Base, S.; Van Gelder, A. (2002). "Algoritmos Computacionales: Introducción al análisis y diseño" Ed. Addison Wesley.

Bovet, D. P.; Crescenci, P. (2006). "Introduction to the theory of complexity" Ed. Creative Commons.

Cairó, Osvaldo y Guardati, Silvia. (2006). Estructuras de datos (3a. Edición). McGraw-Hill.

Criado, Ma. Asunción. (2006). Programación en lenguajes estructurados. AlfaOmega Ra-Ma.

Drozdeck, Adam. (2007). Estructuras de datos y algoritmos en Java (2a Edición). Thomson.

Joyanes, Luis. (2008). Fundamentos de programación (4a Edición). McGraw-Hill.

Joyanes, Luis. M. Fernández, L: Sánchez, I. Zahonero. (2005). Estructuras de datos en C. McGraw-Hill. Schaum.

Koffman, Elliot y Wolfgang, Paul. (2008). Estructura de datos con C++. Objetos, abstracciones y diseño. McGraw-Hill.

Lee R. Teng. S. Chang R. y Tsai Y. (2007). Introducción al diseño de algoritmos. McGraw- Hill.

Levitin Anany (2002). Introduction to Design and Analysis of Algoritms. Addison Wesley.

Loomis, Mary. (1991). Estructura de datos y Organización de archivos (2a. Edición). Prentice Hall.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein. (2009). "Introduction to Algorithms", Third Edition. The MIT Press. Cambridge, Massachusetts London, England.