

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO
LICENCIATURA DE INGENIERÍA EN COMPUTACIÓN



PROGRAMA DE ESTUDIOS
Ingeniería de software II

Elaboró:	Ing. Francisco Ignacio Chávez Castañeda	Facultad de Ingeniería
	M. en I. Sara Vera Noguez	Facultad de Ingeniería
	Ing. Élfego Gutiérrez Ocampo	Facultad de Ingeniería
	Ing. Gregorio García Estrada	C.U. UAEM Atlacomulco
	Dra. En T.I.E. Jaqueline Sánchez Espinoza	C.U. UAEM Teotihuacán

**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.	12





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

Ingeniería de software II

Clave

LINC27

Carga académica

3

1

4

7

Horas
teóricas

Horas
prácticas

Total de
horas

Créditos

Carácter

Obligatoria

Tipo

Curso

Periodo
escolar

Sexto

Área
curricular

Ciencias de la Ingeniería

Núcleo de
formación

Sustantivo

Seriación

Ingeniería de software I

Ninguna

UA Antecedente

UA Consecuente

Formación común

No presenta

X



II. Presentación del programa de estudios.

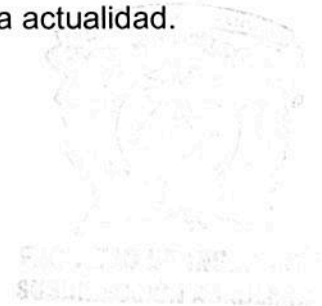
La Ingeniería de Software impulsa el estudio de modelos de desarrollo de software, así como de diversas herramientas para abordar cada fase o flujo de trabajo del proceso de desarrollo, que garantice el buen término del proyecto, en miras a la obtención de un producto de calidad.

En el curso precedente, Ingeniería de Software I, el discente ha obtenido un panorama general del proceso de desarrollo. Como parte importante de este panorama podemos citar los distintos modelos de proceso útiles para el desarrollo de software, y el estudio de herramientas adecuadas para las primeras etapas de este proceso. Logrando los conocimientos y habilidades tanto para la recopilación de requerimientos como para el análisis de sistemas, pero una vez obtenidas esas habilidades resulta indispensable que se capacite al discente en las últimas etapas del proceso, especialmente en el diseño.

Para completar los conocimientos y habilidades, asociadas al proceso de desarrollo de software, en la unidad de aprendizaje de Ingeniería de Software II, se estudian las diversas arquitecturas de software, así como patrones de diseño, y tecnologías para la implementación y pruebas, que le permitan el diseño y posteriormente la construcción de un sistema que además de cumplir los requerimientos funcionales, permita cumplir con requisitos no funcionales como escalabilidad, disponibilidad, extensibilidad y mantenibilidad entre otros, logrando así un software de calidad. Evidenciando dicha claridad a través de la documentación adecuada y profesional tanto de los proyectos, como de los procesos y productos en unidades de medición (métricas).

El hecho de que el discente sea capaz de hacer software de calidad es lo que permite afirmar que será un profesionista capaz de proveer soluciones computacionales innovadoras y sustentables a los problemas, requerimientos y necesidades específicas de la sociedad, contribuyendo así a lograr el perfil del egresado.

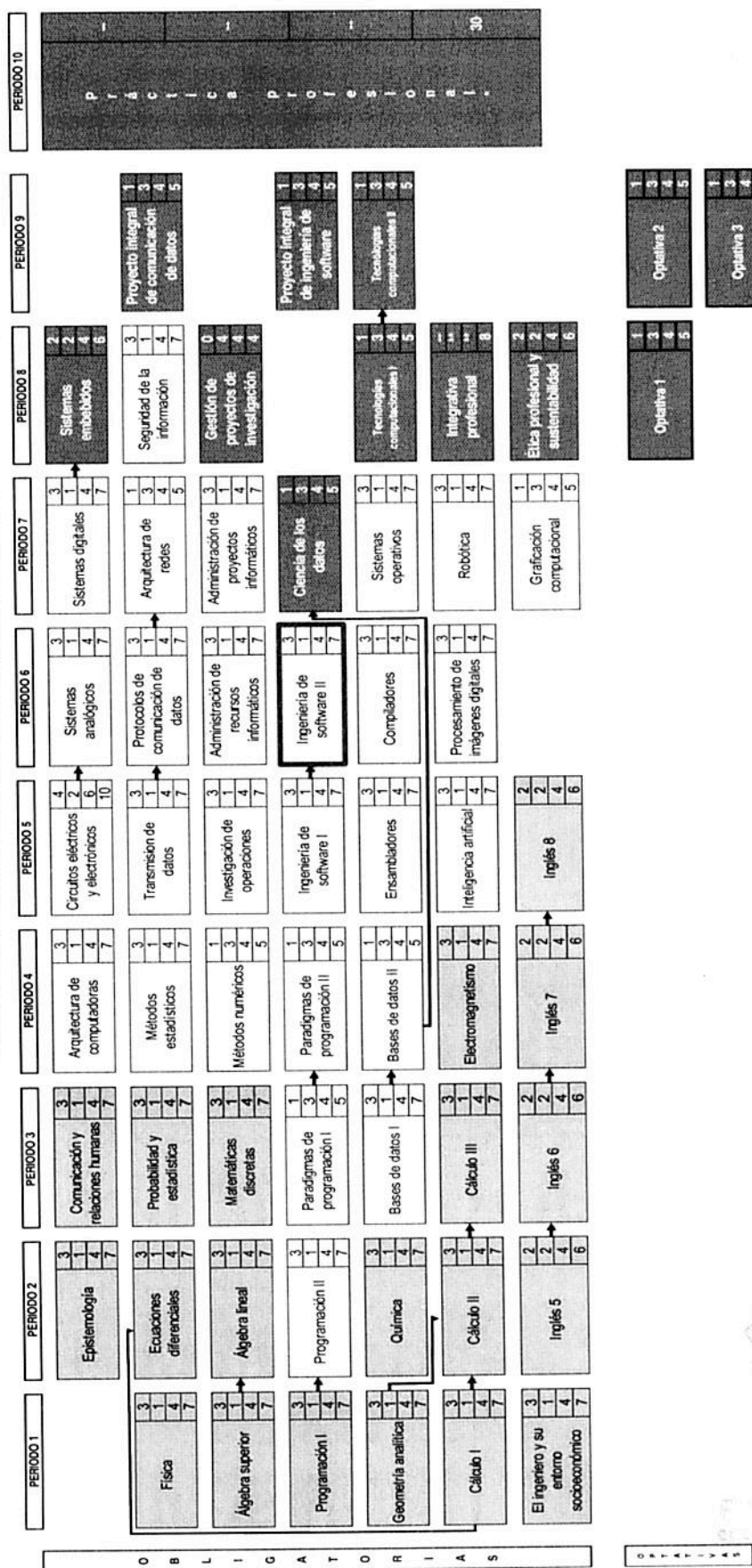
El aporte de esta unidad de aprendizaje al futuro profesionista es involucrarlo en los procesos fundamentales del desarrollo de software, familiarizándolo con la teoría y la práctica del ciclo de vida, a fin de que la comprensión de las técnicas de desarrollo sean un instrumento científico eficaz para el análisis de la problemática inherente al desarrollo de software con calidad, a tiempo y dentro de presupuesto, y, al mismo tiempo contrastar de manera realista estos principios básicos con las realidades de la gestión de software en las empresas, examinándolos en el contexto de las prácticas del desarrollo de software que aún prevalecen en la actualidad.





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

MAPA CURRICULAR DE LA LICENCIATURA DE INGENIERÍA EN COMPUTACIÓN, 2019





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 ciencia 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, metrológico 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 disciplinarios 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.

Crear sistemas de información utilizando patrones de diseño y con la adopción de una plataforma tecnológica adecuada para la automatización óptima de procesos en organizaciones y en sectores prioritarios como la educación, salud y seguridad social.





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

Unidad temática 1. Introducción al diseño de sistemas

Objetivo: Seleccionar patrones de diseño, para definir arquitecturas que atiendan necesidades específicas de un sistema de información, comparando las características de los diferentes patrones.

Temas:

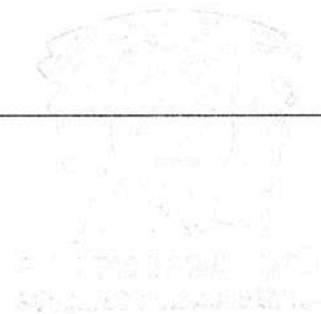
- 1.1 El diseño de sistemas dentro del proceso de desarrollo
 - 1.1.1 Importancia del diseño de sistemas
- 1.2 Patrones de diseño
 - 1.2.1 Definición
 - 1.2.2 Clasificación

Unidad temática 2. Métricas de Software

Objetivo: Examinar las métricas del software, analizando el tamaño, el esfuerzo, el tiempo, el costo y la calidad del software, a través de casos de estudio, con la finalidad de evaluar el diseño de un sistema de software.

Temas:

- 2.1 Teoría de la medición del software
 - 2.1.1 Medición en el software
 - 2.1.2 Clasificación de métricas
- 2.2 Métricas del producto
 - 2.2.1 Líneas de código y complejidad ciclométrica
 - 2.2.2 Métricas de Halstead
- 2.3 Análisis de puntos de función
 - 2.3.1 Modelo IFPUG
 - 2.3.2 Modelo COSMIC
- 2.4 Estimación del software
 - 2.4.1 Modelos de estimación
 - 2.4.2 Algoritmos de estimación
- 2.5 Calidad del software
 - 2.5.1 Modelo de McCall
 - 2.5.2 Modelo ISO 9126





Unidad temática 3. Arquitectura de software

Objetivo: Diferenciar las arquitecturas de software, contrastando sus características mediante los criterios de evaluación, para su aplicación en el diseño de sistemas de software.

Temas:

- 3.1 Arquitectura del software y su importancia
- 3.2 Tipos de arquitectura
 - 3.2.1 Arquitectura monolítica
 - 3.2.2 Arquitectura distribuida
- 3.3 Patrones de arquitectura
 - 3.3.1 Criterios de evaluación de la arquitectura
 - 3.3.2 Ejemplos de arquitecturas y su aplicación
- 3.4 Arquitectura cliente – servidor

Unidad temática 4. Arquitecturas Monolíticas

Objetivo: Diseñar un sistema con arquitectura monolítica, aplicando patrones de diseño ad-hoc a requerimientos específicos, para el desarrollo de un sistema que resuelva una problemática particular.

Temas:

- 4.1 Arquitectura de capas
- 4.2 Aplicación de patrones de diseño en las capas
- 4.3 Herramientas de prueba en las diferentes capas
 - 4.3.1 Pruebas unitarias
 - 4.3.2 Pruebas unitarias para la capa de persistencia
 - 4.3.3 Pruebas de carga
- 4.4 Implementación de un sistema con arquitectura de capas





Unidad temática 5. Arquitecturas distribuidas

Objetivo: Diseñar un sistema con arquitectura distribuida, aplicando patrones de diseño ad-hoc a requerimientos específicos, para el desarrollo de un sistema que resuelva una problemática particular.

Temas:

- 5.1 Microservicios
- 5.2 Servicios REST
- 5.3 Orquestación de servicios
- 5.4 Implementación de un sistema con arquitectura de microservicios

Unidad temática 6. Liberación del producto

Objetivo: Evaluar el proyecto de desarrollo de software, utilizando métricas de software así como los flujos de trabajo propuesto en la etapa de transición, para determinar si la calidad del producto es adecuada para su liberación.

Temas:

- 6.1 Plan de liberación
- 6.2 Tipos de pruebas
 - 6.2.1 Pruebas sobre el producto
 - 6.2.2 Pruebas con usuarios
- 6.3 Capacitación de usuarios
- 6.4 Integración de la documentación del proyecto
- 6.5 Autopsia del proyecto





VII. Acervo bibliográfico

Básico

- Dathan, B., Ramnath, S. (2015). *Object-Oriented Analysis, Design and Implementation*, India: Springer.
- Fenton, N. E., Lawrence P. S. (1997). *Software Metrics: A Rigorous & Practical Approach*, Estados Unidos: An International Thomson Publishing.
- Gamma, E., Helm, R., Johnson, R., Vlissides, J. (2003). *Patrones de Diseño*, Madrid, España: Addison Wesley.
- Jacobson, I., Booch, G., Rumbaugh, J. (2000). *El proceso unificado de desarrollo de software*, Madrid, España: Addison Wesley.
- Noback, M. (2019). *Object Design Style Guide*, NY, Estados Unidos, Manning Publications.
- Richards, M. (2015). *Software Architecture Patterns*, Estados Unidos: O'Reilly Media, Inc.
- Wazlawick, R.S. (2013). *Object-Oriented Analysis and Design for Information Systems*, Edición Internacional, Morgan Kaufmann.

Complementario:

- Basham, B., Sierra, K., Bates, B (2008). *Head First Servlets and JSP*, Estados Unidos: O'Reilly Media, Inc.
- Boninov, V. (2016). *RESTful Web API Design with Node.js*, Birmingham, UK. Packt Publishing.
- Carnel, J., Sánchez, I. H. (2020). *Spring Microservices in Action*, Estados Unidos: Manning Publications.
- Larman, C. (2002). *Applying UML and Patterns: An Introduction to Object-oriented Analysis and Design*, Estados Unidos: Prentice Hall.
- Newman, S. (2015). *Building Microservices*, Estados Unidos: O'Reilly Media, Inc.
- O'Regan, G. (2017). *Concise Guide to Software Engineering*, Edición Internacional, Springer.
- Scott, J.A. (2019). *A Practical Guide to Microservices and Containers*, Estados Unidos: MAPR.

