

UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO
LICENCIATURA EN INGENIERIA ELECTRONICA



Guía Pedagógica:
Programación Básica

Elaboró: Dra. Ma. de los Ángeles Contreras Flores
Ing. Lilian K. Espinosa de los Monteros Fecha: Junio 2019
Heredia

Fecha de
aprobación

H. Consejo Académico
2 de septiembre 2019

H. Consejo de Gobierno
4 de septiembre 2019



Índice

	Pág.
I. Datos de identificación	3
II. Presentación de la guía pedagógica	4
III. Ubicación de la unidad de aprendizaje en el mapa curricular	5
IV. Objetivos de la formación profesional	5
V. Objetivos de la unidad de aprendizaje	5
VI. Contenidos de la unidad de aprendizaje, y su organización	6
VII. Acervo bibliográfico	20
VIII. Mapa curricular	21



I. Datos de identificación

Espacio educativo donde se imparte

Licenciatura

Unidad de aprendizaje Clave

Carga académica

Horas teóricas Horas prácticas Total de horas Créditos

Período escolar en que se ubica

Seriación

UA Antecedente UA Consecuente

Tipo de Unidad de Aprendizaje

Curso Curso taller

Seminario Taller

Laboratorio Práctica profesional

Otro tipo (especificar)

Modalidad educativa

Escolarizada. Sistema rígido No escolarizada. Sistema virtual

Escolarizada. Sistema flexible No escolarizada. Sistema a distancia

No escolarizada. Sistema abierto Mixta (especificar)

Formación común

Licenciatura de Ingeniería Mecánica (2019)

Licenciatura de Ingeniería en Electrónica (2019)

Formación equivalente

Unidad de Aprendizaje



II. Presentación de la guía pedagógica

Conforme lo indica el **Artículo 87 del** Reglamento de Estudios Profesionales vigente, la guía pedagógica es un documento que complementa al programa de estudios y no tiene carácter normativo. Proporcionará recomendaciones para la conducción del proceso de enseñanza aprendizaje. Su carácter indicativo otorgará autonomía al personal académico para la selección y empleo de los métodos, estrategias y recursos educativos que considere más apropiados para el logro de los objetivos. Con base en la modalidad educativa en que se ofrezca cada plan y/o programa de estudios, las unidades de aprendizaje contarán con una guía pedagógica institucional que será aprobada previamente a su empleo.

La guía pedagógica de la UA de Programación Básica será un referente para el personal académico que desempeña docencia, tutoría o asesoría académicas, o desarrolle materiales y medios para la enseñanza y el aprendizaje. En particular para el docente la guía será un instrumento que le oriente de forma sencilla en el desarrollo de sus actividades de enseñanza, así como de algunas estrategias didácticas que permitirán, que los estudiantes desarrollen las competencias propias de la UA.

El enfoque y los principios pedagógicos que guían el desarrollo de la Guía Pedagógica de la UA Programación Básica, corresponden a la corriente constructivista del aprendizaje y la enseñanza, según la cual el aprendizaje es un proceso constructivo interno que realiza el estudiante a partir de su actividad interna y externa y, por intermediación del profesor –facilitador-, que propicia diversas situaciones de aprendizaje para facilitar la construcción de aprendizajes significativos y contextualizar el conocimiento.

Por tanto, los métodos, estrategias y recursos de enseñanza – aprendizaje está enfocada a cumplir los siguientes principios: El uso de estrategias motivacionales para influir positivamente en la disposición de aprendizaje de los estudiantes; la activación de los conocimientos previos de los estudiantes a fin de vincular lo que ya sabe con lo nuevo que va a aprender; diseñar diversas situaciones y condiciones que posibiliten diferentes tipos de aprendizaje; proponer diversas actividades de aprendizaje que brinden al estudiante diferentes oportunidades de aprendizaje y representación del contenido.

Para facilitar el aprendizaje de los contenidos y lograr los objetivos educativos, se diseñó una metodología de enseñanza centrada en el aprendizaje, para lo cual en cada una de las secuencias didácticas que integran esta guía, se incluyeron diferentes actividades de aprendizaje para que el estudiante tenga oportunidad de integrar, practicar o transferir los conocimientos adquiridos en cada unidad temática.

Asimismo, se seleccionaron los métodos, técnicas estrategias y recursos de enseñanza que se consideraron más adecuados para crear diferentes situaciones de aprendizaje con el apoyo de diferentes estímulos que incidan positivamente en la motivación del estudiante para aprender.



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

Núcleo de formación:	Básico
Área Curricular:	Electrónica Básica
Carácter de la UA:	Obligatoria

IV. Objetivos de la formación profesional.

Objetivos del núcleo de formación:

Núcleo básico: Promoverá en el alumno el aprendizaje de las bases contextuales, teóricas y filosóficas de sus estudios, la adquisición de una cultura universitaria en las ciencias y las humanidades, y el desarrollo de las capacidades intelectuales indispensables para la preparación y ejercicio profesional, o para diversas situaciones de la vida personal y social.

Objetivos del área curricular o disciplinaria:

El alumno dominará los principios de la electrónica básica y tendrá los fundamentos para comprender los principios de electrónica aplicada.

Que el alumno conozca las técnicas de programación básica para la aplicación en su especialidad.

El alumno comprenderá los principios de la Electricidad y el Magnetismo, así como su aplicación a circuitos eléctricos y dispositivos electrónicos.

El alumno adquirirá las bases de programación eficiente y metodológica que le sirvan de apoyo en la elaboración de algoritmos para la solución de problemas de su especialidad.

Incluye conocimientos generales de Electrónica para poseer conceptos, habilidades del manejo de los dispositivos eléctricos y electrónicos, leyes y manejo de ecuaciones en la solución de problemas del área de Electrónica, manejo de equipo para mediciones análisis de circuitos con dispositivos eléctricos y electrónicos y el uso de la computadora para solución de ecuaciones.

V. Objetivos de la unidad de aprendizaje.

Diseñar algoritmos mediante el pseudocódigo y la metodología de programación para implementarlos en lenguajes de programación estructurada.



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

Unidad 1. Metodología de programación
Objetivo: Analizar problemas e integrar su solución diseñando algoritmos y una metodología de programación para aplicaciones en ingeniería.
Contenidos: 1.1. Nociones de programación. 1.1.1. Programación, programa y algoritmo. 1.1.2. Paradigma de programación. 1.1.3. Lenguaje de programación. 1.1.4. Programación estructurada. 1.1.5. Errores comunes durante el proceso de programación. 1.2. Metodología o proceso de programación 1.2.1. Análisis, diseño, codificación, pruebas, documentación, mantenimiento.
Métodos, estrategias y recursos educativos
Métodos: <ul style="list-style-type: none">• Inductivo• Lógico• Simbólico• Expositivo• Demostrativo• Analítico• Encuadre• Lluvia de ideas Estrategias: <ul style="list-style-type: none">• Resumen (Acordeón)• Lectura comentada• Reporte• Quizz Recursos educativos: <ul style="list-style-type: none">• Programa de estudios• Guía de evaluación• Lista de Control Escolar• Plataforma• Diapositivas



- Proyector
- Internet,
- Dispositivos móviles,
- Textos impresos,
- Pizarrón,
- Libro de texto.

Actividades de enseñanza y de aprendizaje

Inicio	Desarrollo	Cierre
<p>Dinámica grupal: El docente se presenta ante el grupo y aplica una dinámica para promover la integración.</p> <p>Encuadre: El docente presenta el objetivo, la secuencia de contenidos, la bibliografía, los criterios de evaluación y los lineamientos del curso.</p> <p>A1. Revisar el programa, comentar dudas, inquietudes y expectativas, para establecer acuerdos con el docente</p> <p>Registro: solicitar al alumno realice el registro en la plataforma del curso y mostrar los pasos que se deben realizar.</p> <p>A2. Revisar la plataforma, comentar dudas, inquietudes y expectativas sobre su uso.</p> <p>Evaluación diagnóstica: aplicar el cuestionario de exploración.</p>	<p>1.1.1 – 1.1.5.</p> <p>Exposición: Dar a conocer el objetivo de la unidad, introducir el tema y explicar las nociones básicas del mismo.</p> <p>A4. Elaborar un resumen de los conceptos teóricos vistos en la Unidad 1.</p> <p>Lectura comentada: Solicitar la lectura del tema, previo a la sesión y discutir la información entregada por el alumno.</p> <p>A5. Elaborar un reporte del tema.</p>	<p>Evaluación de reforzamiento: Aplicar cuestionario en plataforma educativa o software educativo para repasar y verificar conocimientos adquiridos.</p> <p>A6. Resolver el Quizz de manera individual</p> <p>A7. Participar en la demostración y revisión de ejercicios y problemas.</p> <p>A8. Expresar dudas y comentarios.</p>



A3. Resolver el cuestionario de exploración.		
2 Hrs.	5 Hrs.	1 Hrs.
Escenarios y recursos para el aprendizaje (uso del alumno)		
Escenarios		Recursos
Sala de cómputo o digital Aula		Equipo de cómputo Plataforma Material de apoyo de la plataforma Cuaderno de apuntes Libro de texto

Unidad 2. Pseudocódigo para el paradigma estructurado.
Objetivo: Analizar problemas e integrar su solución, aplicando pseudocódigo y una metodología de programación para definir funciones y procedimientos.
<p>Contenidos:</p> <p>2.1. Pseudocódigo básico.</p> <p>2.1.1. Estructura de un programa.</p> <p>2.1.2. Estructuras de datos: tipos de datos, constantes, variables.</p> <p>2.1.3. Primitivas elementales: declaraciones, lectura y escritura de datos, operadores aritméticos, relacionales, lógicos y asignación.</p> <p>2.1.4. Expresiones lógicas.</p> <p>2.1.5. Estructuras de control: secuencia, decisión e iteración.</p> <p>2.1.6. Prueba de escritorio.</p> <p>2.2. Pseudocódigo para arreglos.</p> <p>2.2.1. Arreglos unidimensionales y bidimensionales.</p> <p>2.2.2. Pseudocódigo para modularidad.</p> <p>2.2.3. Funciones y procedimientos: declaración, variables locales y globales, paso de parámetros, llamada a funciones y procedimientos.</p>
Métodos, estrategias y recursos educativos
<p>Métodos:</p> <ul style="list-style-type: none"> • Inductivo • Deductivo • Analógico o comparativo • Lógico • Expositivo



- Demostrativo
- Organizador previo
- Colaborativo
- Diálogos simultáneos
- Preguntas intercaladas
- Encuadre
- Lluvia de ideas

Estrategias:

- Solución de problemas
- Mapas mental
- Ilustraciones
- Prácticas

Recursos educativos:

- Software educativo de pseudocódigo (Pselnt o similar)
- Problemario
- Referencias bibliográficas
- Diapositivas
- Internet
- Video
- Bocinas
- Apuntes
- Plataforma basada en juegos (Kahoot o similar)
- Plataforma de aprendizaje (Moodle, Seduca o similar)
- Material de apoyo de la plataforma
- Libro de texto

Actividades de enseñanza y de aprendizaje

Inicio	Desarrollo	Cierre
<p>Exposición: Dar a conocer el objetivo de la unidad, introducir el tema y proyectar un video sobre la importancia de la programación en el área de ingeniería mecánica.</p> <p>Discusión en grupos: proyectar el video y organizar grupos o equipos de trabajo para integrar la</p>	<p>Organizador previo: presentar síntesis inicial sobre conceptos básicos de programación y pseudocódigo.</p> <p>Del 2.1.1 al 2.1.3</p> <p>Expositiva: presentar la estructura del pseudocódigo y los elementos que lo integran: estructuras de datos, primitivas</p>	<p>Actividad Integradora: Revisar los ejercicios y problemas diseñados, y retroalimentar el aprendizaje.</p> <p>A20. Participar en la demostración y revisión de ejercicios y problemas.</p> <p>A21. Expresar dudas y comentarios.</p>



<p>información de lo expuesto y lo observado en el video.</p> <p>A9. Observar el video y opinar sobre la importancia de la programación en la vida cotidiana y profesional</p>	<p>elementales y estructuras de control.</p> <p>Del 2.1.2 al 2.1.5 Demostrativa: desarrollar ejercicios de programación que integren la estructura general, expresiones y estructuras de control en pseudocódigo.</p> <p>A10. Generar un mapa mental para reforzar los conceptos de la estructura del pseudocódigo y los elementos que lo integran.</p> <p>A11. Analizar ejercicios para identificar tipos de datos, declaraciones, lectura de datos, escritura de datos y operadores.</p> <p>A12. Desarrollar los problemas propuestos por el profesor para utilizar los diferentes tipos de operadores en expresiones aritméticas, lógicas y relacionales válidas.</p> <p>Demostrativa: desarrollar pruebas de escritorio de programas en pseudocódigo que simulen su comportamiento para determinar la validez y, además, detectar errores, omisiones o mejoras. Comprobando los resultados de las pruebas de escritorio con los resultados del pseudocódigo en un software educativo.</p>	<p>A22. Realizar problemario extraclase, con soluciones en pseudocódigo y comprobaciones en PseInt.</p> <p>A23. Primer Evaluación Parcial</p>
---	---	---



	<p>Colaborativo: estructurar en equipo soluciones a problemas propuestos en un software educativo, que integren los elementos de los temas vistos en clase.</p> <p>A13. Participar en la solución de los problemas propuestos por el profesor en software educativo de pseudocódigos como PseInt.</p> <p>A14. Resolver la Primer Evaluación Parcial.</p> <p>A15. Realizar lectura previa del documento: “Arreglos”, disponible en la plataforma en los materiales del curso.</p> <p>A16. Elaborar un resumen de conceptos básicos identificados en la lectura: arreglo, modularidad, función, variable local, variable global y paso de parámetros</p> <p>Diálogos simultáneos y preguntas intercaladas: Integrar grupos de trabajo para conversar sobre los conceptos identificados en la lectura y verificar su comprensión.</p> <p>A17. Participar en el diálogo con sus compañeros y precisar respuestas a las preguntas formuladas por el profesor.</p> <p>Del 2.2.1 al 2.3.3</p>	
--	--	--



	<p>Expositiva: presentar los tipos, elementos y aplicación de: arreglos, módulos y funciones.</p> <p>Exhibir las diferencias, ventajas y desventajas de los módulos y funciones.</p> <p>Desatacar la aplicación de procedimientos y funciones en soluciones de pseudocódigo.</p> <p>Del 2.2.1 al 2.2.3</p> <p>Demostrativa: desarrollar ejercicios de programación sencillos con arreglos unidimensionales y bidimensionales; de manera secuencial y modular.</p> <p>A18. Resolver la actividad de identificación de los conceptos expuestos en clase sobre arreglos y modularidad propuesta en la plataforma de aprendizaje.</p> <p>A19. Diseñar pseudocódigos de problemas que requieran diferentes tipos de arreglos secuencial y modularmente.</p>	
1.0 Hrs.	20.0 Hrs.	3.0 Hrs.
Escenarios y recursos para el aprendizaje (uso del alumno)		
Escenarios	Recursos	
Sala de cómputo o digital Aula	Software educativo de pseudocódigo (Pselnt o similar) Problemario Referencias bibliográficas Diapositivas	



	Internet Video Apuntes Plataforma basada en juegos (Kahoot o similar) Plataforma de aprendizaje (Moodle, Seduca o similar) Material de apoyo de la plataforma Libro de texto
--	--

Unidad 3. Lenguaje de programación para el paradigma estructurado
Objetivo: Programar computadoras con base en pseudocódigo y código de un lenguaje de programación estructurada para aplicaciones en ingeniería.
Contenidos: 3.1. Lenguaje de programación estructurada. 3.1.1. Fases en la creación de un programa. 3.1.2. Estructura de un programa. 3.1.3. Características del compilador o intérprete. 3.2. Código básico. 3.2.1. Estructura de un programa. 3.2.2. Estructuras de datos: tipos de datos, constantes, variables. 3.2.3. Primitivas elementales: declaraciones, lectura y escritura de datos, operadores aritméticos, relacionales, lógicos y asignación. 3.2.4. Expresiones lógicas. 3.2.5. Estructuras de control: secuencia, decisión e iteración. 3.2.6. Prueba de escritorio. 3.2.7. Construcción de un programa para computadora utilizando diseños con pseudocódigo y un lenguaje de programación estructurada. 3.3. Código para arreglos. 3.3.1. Arreglos unidimensionales y bidimensionales. 3.3.2. Construcción de un programa para computadora utilizando diseños con pseudocódigo para arreglos y un lenguaje de programación estructurada. 3.4. Pseudocódigo para modularidad. 3.4.1. Funciones y procedimientos: declaración, variables locales y globales, paso de parámetros, llamada a funciones y procedimientos. 3.4.2. Construcción de un programa para computadora utilizando diseños con pseudocódigo para modularidad y un lenguaje de programación estructurado.
Métodos, estrategias y recursos educativos
Métodos: <ul style="list-style-type: none">• Inductivo• Deductivo• Analógico o comparativo



- Lógico
- Expositivo
- Demostrativo
- Organizador previo
- Colaborativo
- Diálogos simultáneos
- Preguntas intercaladas
- Encuadre
- Lluvia de ideas

Estrategias:

- Solución de ejercicios de programación
- Investigación
- Discusión expositiva
- Tabla comparativa
- Casos
- Resumen
- Diálogos simultáneos
- Preguntas intercaladas
- Identificación y reconocimiento de conceptos
- Diálogo
- Prácticas en lenguaje de programación C

Recursos educativos:

- Problemario
- Serie de ejercicios
- Libro de texto
- Referencias bibliográficas
- Diapositivas

Actividades de enseñanza y de aprendizaje

Inicio	Desarrollo	Cierre
<p>Expositiva: Dar a conocer el objetivo de la unidad, introducir el tema y proyectar un video de aplicaciones desarrolladas en lenguaje C.</p> <p>A24. Lluvia de ideas sobre aplicaciones y problemas</p>	<p>Ilustraciones: presentar conceptos básicos de pseudocódigo y su vinculación a la estructura de un programa en lenguaje C.</p> <p>Del 3.1.1 al 1.1.3</p>	<p>Trabajo colaborativo: Discusión guiada sobre resumen y revisión de conceptos.</p> <p>Actividad Integradora: Revisar los ejercicios y problemas diseñados, y</p>



<p>cotidianos que requieran soluciones programadas.</p>	<p>Expositiva: presentar las fases de creación y estructura de un programa en lenguaje de programación C.</p> <p>Del 3.1.2 al 3.1.3 Demostrativa: desarrollar ejercicios de programación que integren la estructura en lenguaje C.</p> <p>A25. Participar en el Foro de la plataforma educativa Moodle (o similar), para enriquecer el tema con conclusiones propias sobre las diferencias entre compiladores e intérpretes.</p> <p>A26. Identificar las partes de la estructura de un programa en lenguaje de programación C en los problemas propuestos por el profesor.</p> <p>Del 3.2.1 al 3.2.5 Expositiva: presentar la estructura de un programa en lenguaje C y los elementos que lo integran: estructuras de datos, primitivas elementales y estructuras de control.</p> <p>Del 3.2.1 al 3.2.5 Demostrativa: desarrollar ejercicios de programación que integren la estructura general, expresiones y estructuras de control en lenguaje C.</p> <p>A27. Investigar en fuentes electrónicas, los tipos de datos, operadores</p>	<p>retroalimentar el aprendizaje.</p> <p>A44. Participar en la demostración y revisión de ejercicios y problemas.</p> <p>A45. Expresar dudas y comentarios.</p> <p>A46. Realizar problemario extraclase, con soluciones secuenciales y modulares en lenguaje de programación C.</p> <p>A47. Revisión de Segunda Evaluación Parcial.</p>
---	---	---



	<p>aritméticos, lógicos y relacionales en lenguaje C.</p> <p>A28. Discusión expositiva sobre la investigación realizada.</p> <p>A29. Analizar casos propuestos por el profesor y diseñar soluciones funcionales en lenguaje C.</p> <p>A30. Desarrollar una tabla comparativa sobre operadores aritméticos, lógicos y relacionales.</p> <p>Demostrativa: desarrollar pruebas de escritorio de programas en lenguaje de programación estructurado C, que simulen su comportamiento para determinar su validez y correcto funcionamiento, además, detectar errores, omisiones o mejoras. Comprobar los resultados de las pruebas de escritorio con los resultados del programa ejecutado en Dev C++ (o compilador similar).</p> <p>Colaborativo: estructurar en equipo soluciones a problemas propuestos en el compilador de lenguaje C, que integren los elementos de los temas vistos en clase.</p> <p>A31. Participar en la solución de los problemas propuestos por el profesor en Dev C++ (o compilador similar)</p> <p>.</p>	
--	---	--



	<p>A32. Analizar y diseñar solución de casos propuestos por el profesor. Generando soluciones pseudocódigo y lenguaje C.</p> <p>A33. Contrastar los resultados de las soluciones propuestas con pruebas de escritorio y corridas de programas</p> <p>A34. Revisar material de “Arreglo en lenguaje C”, disponible en la plataforma, en los materiales del curso.</p> <p>A35. Elaborar un resumen de conceptos básicos identificados en el material del curso: arreglos, tipos de arreglos y su uso en lenguaje C.</p> <p>Diálogos simultáneos y preguntas intercaladas: Integrar grupos de trabajo para conversar sobre los conceptos identificados en la lectura y verificar su comprensión.</p> <p>A36. Participar en el diálogo con sus compañeros y precisar respuestas a las preguntas formuladas por el profesor.</p> <p>Expositiva: presentar los tipos, elementos y aplicación de: arreglos en lenguaje C. Desatacar la aplicación de procedimientos y funciones en lenguaje C.</p> <p>Demostrativa: desarrollar ejercicios de programación</p>	
--	---	--



	<p>en lenguaje C, con arreglos unidimensionales y bidimensionales.</p> <p>A37. Resolver la actividad de identificación de los conceptos expuestos en clase sobre arreglos propuesta en la plataforma de aprendizaje.</p> <p>A38. Diseñar programas secuenciales en lenguaje C, de problemas que requieran diferentes tipos de arreglos.</p> <p>A39. Identificar y reconocer los conceptos de función, procedimiento, variable local, variable global y paso de parámetros, presentados en el video disponible en los materiales de la plataforma, e integrar un mapa conceptual con los mismos.</p> <p>Diálogos simultáneos y preguntas intercaladas: Integrar grupos de trabajo para conversar sobre los conceptos identificados en el material y verificar su comprensión.</p> <p>A40. Participar en el diálogo con sus compañeros y precisar respuestas a las preguntas formuladas por el profesor.</p> <p>Expositiva: Desatacar la aplicación de procedimientos y funciones en lenguaje C, explicando cada uno de sus elementos.</p> <p>Demostrativa: desarrollar ejercicios de programación</p>	
--	---	--



	<p>en lenguaje C, funciones y procedimientos.</p> <p>A41. Resolver la actividad de identificación de los conceptos expuestos en clase funciones y procedimientos propuesta en la plataforma de aprendizaje.</p> <p>A42. Diseñar programas en lenguaje C, de problemas que requieran funciones o procedimientos.</p> <p>A43. Segundo Examen Parcial</p>	
1 Hrs.	19 Hrs.	4 Hrs.
Escenarios y recursos para el aprendizaje (uso del alumno)		
Escenarios	Recursos	
Sala de cómputo o digital Aula	<p>Compilador de C (Dev C++, Borland C, Code Blocks, etc)</p> <p>Software educativo de pseudocódigo (PseInt o similar)</p> <p>Problemario</p> <p>Referencias bibliográficas</p> <p>Diapositivas</p> <p>Internet</p> <p>Video</p> <p>Apuntes</p> <p>Plataforma basada en juegos (Kahoot o similar)</p> <p>Plataforma de aprendizaje (Moodle, Seduca o similar)</p> <p>Material de apoyo de la plataforma</p> <p>Libro de texto</p>	



VII. Acervo bibliográfico

Básico:

- Cairó, O. (2006). *Fundamentos de Programación. Piensa en C*. México: Pearson-Prentice Hall.
- Cairó, O. (1995). *Metodología de la Programación*. México: Computec.
- Corona, M., & Ancona, M. (2012). *Diseño de Algoritmos y su Codificación en Lenguaje C*. México: McGraw-Hill.
- Joyanes, L. (2008). *Fundamentos de Programación*. México McGraw-Hill.
- López, L. (2004). *Programación estructurada. Un enfoque algorítmico*. España: Alfaomega.

Complementario:

- Ceballos, F. (1991). *Lenguaje C*. Madrid: Ra-Ma Addison.
- Criado, M. (2006). *Programación en lenguajes estructurados*. España: Alfaomega Ra-Ma.
- Gottfried, B. (2005). *Programación en C*. México: McGraw-Hill.
- Joyanes, L. & Zahonero, I. (2000). *Programación en C, metodología, estructura de datos y objetos*. México: McGraw-Hill.



Proyecto curricular de la Licenciatura de Ingeniería en Electrónica
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
								Bioelectrónica ¹	
								Ingeniería de audio	
								Robótica	
								Electrónica de potencia en sistemas sustentables	
								Electrónica de los sistemas de transporte	
								Telefonía	
								Control avanzado	

SIMBOLOGÍA	
Unidad de aprendizaje	HT: Horas Teóricas
	HP: Horas Prácticas
	TH: Total de Horas
	CR: Créditos
→ 24 horas de selección.	
Créditos mínimos 22 y máximos 56 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 sustantivo obligatorio.
	Núcleo Integral obligatorio.
	Núcleo Integral optativo.

PARÁMETROS DEL PLAN DE ESTUDIOS													
Núcleo básico obligatorio: cursar y acreditar 22 UA	56 31 87 143	Total del núcleo básico: acreditar 22 UA para cubrir 143 créditos											
Núcleo sustantivo obligatorio: cursar y acreditar 21 UA	58 47 105 163	Total del núcleo sustantivo acreditar 21 UA para cubrir 163 créditos											
Núcleo Integral obligatorio: cursar y acreditar 13 UA + 2*	28 38 94 130	Núcleo Integral optativo: cursar y acreditar 2 UA	8 21 8 14										
		Total del núcleo Integral acreditar 15 UA + 2* para cubrir 144 créditos											
<table border="1"> <thead> <tr> <th colspan="2">TOTAL DEL PLAN DE ESTUDIOS</th> </tr> </thead> <tbody> <tr> <td>UA obligatorias</td> <td>56 + 2 Actividades académicas</td> </tr> <tr> <td>UA optativas</td> <td>2</td> </tr> <tr> <td>UA a acreditar</td> <td>56 + 2 Actividades académicas</td> </tr> <tr> <td>Créditos</td> <td>490</td> </tr> </tbody> </table>				TOTAL DEL PLAN DE ESTUDIOS		UA obligatorias	56 + 2 Actividades académicas	UA optativas	2	UA a acreditar	56 + 2 Actividades académicas	Créditos	490
TOTAL DEL PLAN DE ESTUDIOS													
UA obligatorias	56 + 2 Actividades académicas												
UA optativas	2												
UA a acreditar	56 + 2 Actividades académicas												
Créditos	490												



Proyecto curricular de la Licenciatura de Ingeniería en Electrónica
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
								Bioelectrónica [†]	
								Ingeniería de audio	
								Robótica	
								Electrónica de potencia en sistemas sustentables	
								Electrónica de los sistemas de transporte	
								Telefonía	
								Control avanzado	

SIMBOLOGÍA	
Unidad de aprendizaje	HT: Horas Teóricas HP: Horas Prácticas TH: Total de Horas CR: Créditos
→ 24 horas de sesión. Créditos mínimos 22 y máximos 56 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 sustantivo obligatorio.
	Núcleo Integral obligatorio.
	Núcleo Integral optativo.

PARÁMETROS DEL PLAN DE ESTUDIOS			
Núcleo básico obligatorio: cursar y acreditar 22 UA	56 31 87 143	Total del núcleo básico: acreditar 22 UA para cubrir 143 créditos	
Núcleo sustantivo obligatorio: cursar y acreditar 21 UA	58 47 105 163	Total del núcleo sustantivo: acreditar 21 UA para cubrir 163 créditos	
Núcleo Integral obligatorio: cursar y acreditar 13 UA + 2*	28 88** 64** 130	Núcleo Integral optativo: cursar y acreditar 2 UA	8 23 8 14
		Total del núcleo Integral: acreditar 15 UA + 2* para cubrir 144 créditos	

TOTAL DEL PLAN DE ESTUDIOS	
UA obligatorias	56 + 2 Actividades académicas
UA optativas	2
UA a acreditar	56 + 2 Actividades académicas
Créditos	490



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



Distribución de unidades de aprendizaje optativas

	PERIODO 1	PERIODO 2	PERIODO 3	PERIODO 4	PERIODO 5	PERIODO 6	PERIODO 7	PERIODO 8	PERIODO 9	PERIODO 10
OPTATIVAS							Ingeniería estados por computadora	Manufactura estados por computadora	Diseño de módulos	
							Desarrollo empresarial	Tendencias tecnológicas para el desarrollo de negocios	Módulo de negocios	
							Sistemas productivos	Gestión Industrial	Sistemas de manufactura	
							Hubs	Materiales compuestos	Materiales avanzados	

SIMBOLOGÍA	
HT: Horas Teóricas	
HP: Horas Prácticas	
TH: Total de Horas	
CR: Créditos	

17 Líneas de sección →

*Actividad académica

** Horas de las actividades académicas

Créditos a cursar por período escolar:
mínimo 21 y máximo 53

	Núcleo básico obligatorio
	Núcleo sustantivo obligatorio
	Núcleo Integral obligatorio
	Núcleo Integral optativo

PARÁMETROS DEL PLAN DE ESTUDIOS			
Núcleo básico obligatorio: cursar y acreditar 20 UA	50 35 85 135	Total del núcleo básico: acreditar 20 UA para cubrir 135 créditos	TOTAL DEL PLAN DE ESTUDIOS
Núcleo sustantivo obligatorio: cursar y acreditar 23 UA	50 46 96 146	Total del núcleo sustantivo: acreditar 23 UA para cubrir 146 créditos	
Núcleo Integral obligatorio: cursar y acreditar 14 UA + 2*	29 28** 87** 134	Total del núcleo Integral: acreditar 17 UA + 2** para cubrir 142 créditos	UA Obligatorias 57 + 2 Actividades académicas
Núcleo Integral optativo: cursar y acreditar 3 UA	6 6 12 18		UA Optativas 3
			UA a acreditar 60 + 2 Actividades académicas
			Créditos 423