

1. Datos Generales de la asignatura

Nombre de la asignatura:	Programación Básica
Clave de la asignatura:	MTD-1024
SATCA¹:	2 – 3 – 5
Carrera:	Ingeniería Mecatrónica

2. Presentación

Caracterización de la asignatura

Esta asignatura aporta, al perfil del ingeniero mecatrónico, la capacidad para desarrollar un pensamiento lógico, identificar el proceso de creación de un programa y desarrollo de algoritmos para resolver problemas.

La asignatura proporciona al estudiante de mecatrónica una herramienta para resolver problemas de aplicaciones de la vida ordinaria y de aplicaciones de la ingeniería

La Ingeniería Mecatrónica, está conformada por la integración de la Ingeniería Mecánica, Electrónica, Eléctrica y en los Sistemas Computacionales. Esta última está incluida dentro de las otras áreas mediante el uso de software de diseño y simulación, pero debe existir de manera explícita y separada para la creación de software de interface, automatización y control.

Está diseñada para el logro de competencias específicas dirigidas al aprendizaje de los diferentes dominios: manejo de consola y diseño de algoritmos. Comprende los conceptos básicos de la programación y escribe expresiones aritméticas y lógicas en un lenguaje de programación. Así como el uso y funcionamiento de las estructuras secuenciales, selectivas, arreglos unidimensionales y multidimensionales, punteros y archivos en el desarrollo de aplicaciones. Es capaz de aplicarlos al construir y desarrollar aplicaciones de software que requieran dichas estructuras.

La materia de programación básica pretende formar una lógica de pensamiento estructurada, para la comprensión del funcionamiento de los diferentes lenguajes de programación de alto nivel, indispensables para el desarrollo de tecnología informática.

Programación Básica es el soporte directo de la materia de programación avanzada en la que se abordan temas como POO y visual.

Intención didáctica

La asignatura proporciona al alumno los conceptos esenciales del diseño algorítmico, el temario se organiza en siete temas.

En el primer tema se estudian los conceptos básicos para introducir al estudiante en la programación con la finalidad de obtener las bases conceptuales para abordar las siguientes temáticas. También, se desarrolla el análisis y desarrollo de algoritmos, con la finalidad de dar solución a problemas reales utilizando el razonamiento lógico.

¹ Sistema de Asignación y Transferencia de Créditos Académicos

El segundo tema, tiene la finalidad de obtener y aplicar herramientas necesarias para diseñar e implementar soluciones en un lenguaje de programación, utilizando los conceptos adquiridos.

El tercer tema tiene como objetivo que el alumno identifique, comprenda, seleccione e implemente la estructura de control más adecuada a un problema específico, así como el diseño de bloques de códigos reutilizables, dado que es común encontrar en la práctica problemas cuyas operaciones están condicionadas o deban ejecutarse un número repetido de veces.

El cuarto tema tiene la finalidad de implementar arreglos para una gran variedad de propósitos que proporcionan un medio conveniente de agrupar variables relacionadas y organizar datos de una manera que puedan ser fácilmente procesados. Debe extender el concepto a conjuntos de datos relacionados de diferente tipo de dato como lo son las estructuras y las uniones.

El quinto tema, módulos, se trata como un bloque que ayuda al re uso del software, que incluso es adaptable al establecer módulos que pueden ser llamados con diferentes parámetros.

En el tema 6, punteros, se define el concepto, declaración y uso de punteros, llamadas por valor y referencia.

El Tema 7 es Archivos que pueden ser secuenciales o directos, el manejo de operaciones con archivos y la aplicación de estos junto con el resto de las otras estructuras de programación.

3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Evento
Instituto Tecnológico Superior de Irapuato, del 24 al 28 de agosto de 2009.	Representantes de los Institutos Tecnológicos de: Apizaco, Celaya, Ciudad Cuauhtémoc, Cuautla, Durango, Guanajuato, Hermosillo, Huichapan, Irapuato, Jilotepec, Jocotitlán, La Laguna, Oriente del Estado de Hidalgo, Pabellón de Arteaga, Parral, Reynosa, Saltillo, San Luis Potosí, Tlalnepantla, Toluca y Zacapoaxtla.	Reunión Nacional de Diseño e Innovación Curricular para el Desarrollo y Formación de Competencias Profesionales de las Carreras de Ingeniería Eléctrica, Ingeniería Electromecánica, Ingeniería Electrónica e Ingeniería Mecatrónica.
Instituto Tecnológico de Mexicali, del 25 al 29 de enero del 2010.	Representantes de los Institutos Tecnológicos de: Apizaco, Celaya, Ciudad Cuauhtémoc, Cuautla, Durango, Guanajuato, Hermosillo, Huichapan, Irapuato, Jilotepec, Jocotitlán, La Laguna, Mexicali, Oriente del Estado de Hidalgo,	Reunión Nacional de Consolidación de los Programas en Competencias Profesionales de las Carreras de Ingeniería Eléctrica, Ingeniería Electromecánica, Ingeniería Electrónica e Ingeniería Mecatrónica.

	Pabellón de Arteaga, Reynosa, Saltillo, San Luis Potosí, Toluca y Zacapoaxtla.	
Instituto Tecnológico de la Laguna, del 26 al 29 de noviembre de 2012.	Representantes de los Institutos Tecnológicos de: Apizaco, Celaya, Chapala, Cd. Cuauhtémoc, Colima, Culiacán, Huixquilucan, La Laguna, León, Nuevo Laredo, Nuevo León, Oriente del Estado de Hidalgo, Querétaro, Tlalnepantla, Uruapan, Veracruz y Zacapoaxtla.	Reunión Nacional de Seguimiento Curricular de los Programas en Competencias Profesionales de las Carreras de Ingeniería Eléctrica, Ingeniería Electromecánica, Ingeniería Electrónica, Ingeniería Mecánica e Ingeniería Mecatrónica.
Instituto Tecnológico de Toluca, del 10 al 13 de febrero de 2014.	Representantes de los Institutos Tecnológicos de: Aguascalientes, Boca del Río, Celaya, Mérida, Orizaba, Puerto Vallarta y Veracruz.	Reunión de Seguimiento Curricular de los Programas Educativos de Ingenierías, Licenciaturas y Asignaturas Comunes del SNIT.
Tecnológico Nacional de México, del 25 al 26 de agosto de 2014.	Representantes de los Institutos Tecnológicos de: Aguascalientes, Apizaco, Boca del Río, Celaya, Cerro Azul, Cd. Juárez, Cd. Madero, Chihuahua, Coacalco, Coatzacoalcos, Durango, Ecatepec, La Laguna, Lerdo, Matamoros, Mérida, Mexicali, Motúl, Nuevo Laredo, Orizaba, Pachuca, Poza Rica, Progreso, Reynosa, Saltillo, Santiago Papasquiario, Tantoyuca, Tlalnepantla, Toluca, Veracruz, Villahermosa, Zacatecas y Zacatepec. Representantes de Petróleos Mexicanos (PEMEX).	Reunión de trabajo para la actualización de los planes de estudio del sector energético, con la participación de PEMEX.

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura

Plantea metodológicamente la solución de problemas susceptibles de ser computarizados a través del manejo de técnicas estructuradas de diseño y formulación de algoritmos.

5. Competencias previas

- Utiliza las herramientas básicas de la computación para crear archivos de texto y buscar información en Internet.

6. Temario

No.	Temas	Subtemas
1	Conceptos Básicos y Algoritmos	1.1 Clasificación del software de: sistemas y aplicación. 1.2 Definición de: Algoritmo, Lenguaje de Programación, Programa, Programación, Paradigmas de programación, Editores de texto, Compiladores e intérpretes, Ejecutables, Consola de línea de comandos. 1.3 Análisis de problemas. 1.4 Representación de algoritmos: gráfica y pseudocódigo. 1.5 Diseño de algoritmos aplicados a problemas. 1.6 Diseño algorítmico de funciones
2	Introducción a la Programación	2.1 Características del lenguaje de programación 2.2 Estructura básica de un programa. 2.3 Traducción de un programa: compilación, enlace de un programa, errores en tiempo de compilación. 2.4 Ejecución de un programa. 2.5 Elementos del lenguaje: datos, literales y constantes, identificadores, variables, parámetros, operadores, entrada y salida de datos. 2.6 Errores en tiempo de ejecución.
3	Control de flujo	3.1 Estructuras secuenciales. 3.2 Estructuras selectivas: simple, doble y múltiple. 3.3 Estructuras iterativas: repetir mientras, hasta, desde 3.4 Diseño e implementación de funciones
4	Arreglos, Estructuras y Uniones	4.1 Unidimensionales: conceptos básicos, operaciones y aplicaciones. 4.2 Multidimensionales: conceptos básicos, operaciones y aplicaciones. 4.3 Aplicaciones en programas
5	Módulos	5.1 Diferencia entre Funciones y Procedimientos. 5.2 Activación / Invocación de Funciones y Procedimientos 5.3 Uso de Funciones y Procedimientos con Parámetros.

6	Punteros	6.1 Declaración e inicialización 6.2 Operadores de apuntador 6.3 Uso de apuntadores 6.4 Llamadas por valor y referencia
7	Archivos	7.1 Creación, lectura, escritura y cierre de archivos de acceso secuencial 7.2 Creación, lectura, escritura y cierre de archivos de acceso directo

7. Actividades de aprendizaje de los temas

1 Conceptos Básicos y Algoritmos	
Competencias	Actividades de aprendizaje
<p>Específica(s): Domina los conceptos básicos de la programación para analizar problemas y representar su solución mediante algoritmos.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de pensamiento lógico, algorítmico, heurístico, analítico y sintético. • Resolución de problemas. • Búsqueda y manejo de información. • Capacidad crítica y autocrítica 	<ul style="list-style-type: none"> • Investigar la clasificación del software. • Reconocer los conceptos básicos: algoritmo, programa, programación, paradigmas de programación utilizando mapas conceptuales, mentales, cuadros sinópticos, etc. • Conocer el entorno de un lenguaje de programación. • Manejar la consola para la compilación y ejecución de programas. • Explicar los conceptos básicos para la formulación de algoritmos, así como sus ventajas y desventajas. • Generar un catálogo de problemas para su análisis y solución. • Resolver y analizar problemas cotidianos. • Investigar los diferentes métodos para la representación de un algoritmo: diagrama de flujo, N-S (Nassi-Shneiderman), Pseudocódigo, Descripción Narrada.
2 Introducción a la Programación	
Competencias	Actividades de aprendizaje
<p>Específica(s): Conoce las características principales de un lenguaje de programación que le permiten codificar algoritmos, compilar y ejecutar programas.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de pensamiento lógico, algorítmico, heurístico, analítico y sintético. • Resolución de problemas. • Búsqueda y manejo de información. 	<ul style="list-style-type: none"> • Realizar un mapa conceptual sobre los tipos de software y los conceptos básicos de programación. • Buscar y analizar información necesaria para instalar y configurar el compilador del lenguaje de programación a utilizar. • Realizar cambios en expresiones lógicas y algebraicas de un programa modelo y analizar los resultados obtenidos. • Realizar programas completos de menor a mayor grado de dificultad y con base en cada una de las instrucciones que los componen

• Capacidad crítica y autocrítica	enseñar la sintaxis del lenguaje
3 Control de flujo	
Competencias	Actividades de aprendizaje
<p>Específica(s): Construye programas utilizando estructuras condicionales y repetitivas para aumentar su funcionalidad</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de pensamiento lógico, algorítmico, heurístico, analítico y sintético. • Resolución de problemas. • Búsqueda y manejo de información. • Capacidad crítica y autocrítica. • Búsqueda y manejo de información. • Capacidad de aprender. 	<ul style="list-style-type: none"> • Realizar una investigación sobre el funcionamiento y aplicación de las estructuras de selección y de repetición. • Diseñar programas donde se utilicen las estructuras de repetición y selección. • Construir programas que implementen métodos o funciones.
4 Arreglos, Estructuras y Uniones	
Competencias	Actividades de aprendizaje
<p>Específica(s): Construye programas que utilicen arreglos unidimensionales y multidimensionales para solucionar problemas.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de pensamiento lógico, algorítmico, heurístico, analítico y sintético. • Resolución de problemas. • Búsqueda y manejo de información. • Capacidad crítica y autocrítica. • Búsqueda y manejo de información. • Capacidad de aprender. • Capacidad de aplicar los conocimientos en la práctica. 	<ul style="list-style-type: none"> • Diseñar algoritmos que utilicen arreglos unidimensionales y multidimensionales. • Desarrollar programas para implementar las operaciones básicas en arreglos. • Realizar programas que involucren estructuras y uniones.
5 Módulos	
Competencias	Actividades de aprendizaje
<p>Específica(s): Aplica funciones y procedimientos en programas.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. 	<ul style="list-style-type: none"> • Identificar la diferencia entre funciones y procedimientos participando en discusión grupal. • Aplicar la activación / invocación de funciones y procedimientos en ejercicios propuestos.



<ul style="list-style-type: none"> • Capacidad de pensamiento lógico, algorítmico, heurístico, analítico y sintético. • Resolución de problemas. • Búsqueda y manejo de información. • Capacidad crítica y autocrítica. • Búsqueda y manejo de información. • Capacidad de aprender. • Capacidad de aplicar los conocimientos en la práctica. 	<ul style="list-style-type: none"> • Utilizar funciones y procedimientos en programas.
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------

6 Punteros

Competencias	Actividades de aprendizaje
<p>Específica(s): Declara y aplica apuntadores como manera de manipular información en llamadas por valor y referencia.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de pensamiento lógico, algorítmico, heurístico, analítico y sintético. • Resolución de problemas. • Búsqueda y manejo de información. • Capacidad crítica y autocrítica. • Búsqueda y manejo de información. • Capacidad de aprender. • Capacidad de aplicar los conocimientos en la práctica. 	<ul style="list-style-type: none"> • Declarar apuntadores. • Utiliza operadores con apuntadores. • Hacer llamadas a módulos por valor y observa los resultados, saca conclusiones. • Hacer llamadas a módulos por referencia y observa los resultados, saca conclusiones. • Realizar aplicaciones que involucran el uso de apuntadores

7 Archivos

Competencias	Actividades de aprendizaje
<p>Específica(s): Utiliza las librerías de manejos de archivos para crearlos, leerlos, escribirlos, buscar registros y cerrarlos, ya sean secuenciales o directos.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de análisis y síntesis. • Capacidad de pensamiento lógico, algorítmico, heurístico, analítico y sintético. • Resolución de problemas. • Búsqueda y manejo de información. • Capacidad crítica y autocrítica. • Búsqueda y manejo de información. • Capacidad de aprender. • Capacidad de aplicar los conocimientos en la práctica. 	<ul style="list-style-type: none"> • Definir tipos de archivos secuenciales y directos • Crear archivos secuenciales y directos, con diferentes propiedades: W, R, RW, A, etc. • Leer, escribir datos de archivos ya sean tipos de datos simples o estructurados. • Hacer búsqueda de registros en archivos para luego modificar el dato.

8. Práctica(s)

- Investiga los tipos de software que se utilizan en una organización e identificarlos dentro de la clasificación del software.
- Elabora ejercicios que impliquen entrada y salida de datos, así como el uso de operadores, operandos.
- Soluciona problemas con algoritmos a partir de enunciados proporcionados por el profesor.
- Crea, compila y ejecuta programas.
- Declara variables y usa expresiones aritméticas, relacionales, lógicas y de igualdad.
- Resuelve problemas que utilicen entrada y salida de datos.
- Soluciona problemas utilizando sentencias de control.
- Implementa soluciones con arreglos.

9. Proyecto de asignatura

El objetivo del proyecto que plantee el docente que imparta esta asignatura, es demostrar el desarrollo y alcance de la(s) competencia(s) de la asignatura, considerando las siguientes fases:

- **Fundamentación:** marco referencial (teórico, conceptual, contextual, legal) en el cual se fundamenta el proyecto de acuerdo con un diagnóstico realizado, mismo que permite a los estudiantes lograr la comprensión de la realidad o situación objeto de estudio para definir un proceso de intervención o hacer el diseño de un modelo.
- **Planeación:** con base en el diagnóstico en esta fase se realiza el diseño del proyecto por parte de los estudiantes con asesoría del docente; implica planificar un proceso: de intervención empresarial, social o comunitario, el diseño de un modelo, entre otros, según el tipo de proyecto, las actividades a realizar los recursos requeridos y el cronograma de trabajo.
- **Ejecución:** consiste en el desarrollo de la planeación del proyecto realizada por parte de los estudiantes con asesoría del docente, es decir en la intervención (social, empresarial), o construcción del modelo propuesto según el tipo de proyecto, es la fase de mayor duración que implica el desempeño de las competencias genéricas y específicas a desarrollar.
- **Evaluación:** es la fase final que aplica un juicio de valor en el contexto laboral-profesión, social e investigativo, ésta se debe realizar a través del reconocimiento de logros y aspectos a mejorar se estará promoviendo el concepto de “evaluación para la mejora continua”, la metacognición, el desarrollo del pensamiento crítico y reflexivo en los estudiantes.

10. Evaluación por competencias

Son las técnicas, instrumentos y herramientas sugeridas para constatar los desempeños académicos de las actividades de aprendizaje.

- Mapa conceptual
- Problemario
- Examen
- Esquemas
- Representaciones gráficas o esquemáticas
- Mapas mentales
- Ensayos
- Reportes de prácticas
- Resúmenes
- Rúbrica

- Exposiciones orales.
- Lista de cotejo
- Matriz de valoración
- Guía de observación

11. Fuentes de información

1. Luis Joyanes Aguilar, Fundamentos de Programación, Ed. Prentice Hall.
2. Jesús J. García Molina Introducción a la programación un Enfoque Algorítmico, Ed. Paraninfo.
3. Leobardo López Román, Metodología de la Programación Orientada a Objetos, Ed. Alfaomega.
4. Cairo Osvaldo, Metodología de la Programación, Ed. Alfaomega.
5. Deitel y Deitel. Como Programar en C++ quinta Edición. Prentice Hall.
6. Deitel y Deitel. Como Programar en C# quinta Edición. Prentice Hall.
7. Deitel y Deitel. Java como programar. Séptima edición. Prentice Hall.
8. Joyanes Aguilar, Luis Fernández, Azuela Matilde, Rodríguez Baena Luis, Fundamentos de Programación Libro de Problemas Algoritmos Estructura de Datos y Objetos. 2a. edición Ed. Mc. Graw Hill
9. Luis Joyanes Aguilar. Programación en JAVA 2 1ª Edición. Mc Graw Hill.
10. Martín Fowler Kendall Scott. UML Gota a Gota. Addison Wesley.
11. Ramírez Felipe, Introducción a la Programación, Algoritmos y su Implementación En Vb.Net C# Java y C++, 2a. edición, Alfa Omega.
12. Jean-Paul Tremblay, Richar B. Bunt. Introducción a la Ciencia de Las Computadoras. Enfoque Algorítmico. McGraw Hill.
13. Bjarne Strstrup. Lenguaje de Programación C/C++.
14. Cairo Battistutti Osvaldo, Metodología de la Programación, Algoritmos Diagramas de Flujo y Programas, 3a. edición, Alfa Omega.
15. Flores Cueto, Juan José, Método de las 6'D UML – Pseudocódigo – Java Enfoque Algorítmico, Serie Textos Universitarios Facultad de Ingeniería y Arquitectura, ed. Universidad de San Martín de Porres, (<http://books.google.com/>).