

## 1. Datos Generales de la asignatura

<b>Nombre de la asignatura:</b>	<b>Programación Estructurada</b>
<b>Clave de la asignatura:</b>	<b>ETD-1024</b>
<b>SATCA<sup>1</sup>:</b>	<b>2-3-5</b>
<b>Carrera:</b>	<b>Ingeniería Electrónica</b>

## 2. Presentación

<p><b>Caracterización de la asignatura</b></p> <p>Esta asignatura aporta al perfil profesional, la capacidad para desarrollar aplicaciones en un lenguaje de programación de alto nivel, para la solución de problemas relacionados con las diferentes disciplinas en el área.</p> <p>Las competencias adquiridas en la asignatura permitirán al estudiante desarrollar aplicaciones de computo en el modo de consola, las cuales podrán facilitar la solución de problemas de ingeniería, tales como los fenómenos físicos cubiertos en la asignatura de mecánica clásica, circuitos de corriente directa estudiados en electricidad y magnetismos, estadística descriptiva definida en la materia de probabilidad y estadística.</p> <p>El desarrollo de programas estructurados aporta las bases de programación para la solución de problemas de cálculo en ingeniería y para el desarrollo de programas de aplicación con interfaces graficas de usuario. Es un recurso valioso que permite la implementación de los algoritmos de la materia de análisis numérico.</p>
<p><b>Intención didáctica</b></p> <p>La asignatura la componen seis temas, el primero aborda la importancia de la programación de computadoras para la resolución de problemas de ingeniería. Se sugiere que el docente presente al estudiante algunos programas de aplicación utilizados en diferentes áreas. Para el diseño de algoritmos se pueden abordar problemas del área de las ciencias básicas de la ingeniería, por ejemplo: Física, Matemáticas y Probabilidad y Estadística.</p> <p>El segundo y tercer tema se destina al estudio de los elementos propios del lenguaje de programación. La academia correspondiente deberá seleccionar el lenguaje de programación más adecuado a su entorno laboral, así como la plataforma de desarrollo a utilizar. Es de suma importancia que los temas sean cubiertos con la mayor profundidad que sea posible. Se recomienda un enfoque progresivo, en el cual se vayan desarrollando aplicaciones simples que demuestren el uso adecuado de cada uno de los elementos del lenguaje y a medida que se avanza en el programa, construir aplicaciones más complejas y funcionales, integrando los conocimientos previamente adquiridos.</p> <p>El cuarto tema aborda la metodología de funciones de código basada en la programación modular que permita una mayor flexibilidad tanto en su reutilización como en su mantenimiento. Se plantea el uso de bibliotecas de funciones que permitan la entrada y salida de datos, su almacenamiento permanente en archivos y el tratamiento de datos de tipo cadena. Se recomienda desarrollar programas que integren múltiples alternativas de operación mediante el uso de menús, algoritmos recursivos, implementación de métodos numéricos por medio de funciones.</p>

<sup>1</sup> Sistema de Asignación y Transferencia de Créditos Académicos

El quinto tema son arreglos unidimensionales y multidimensionales, este tema es importante conocerlo y saber cómo se desplazarse a través del arreglo, para escribir, leer, o modificar. La profundidad con que este tema dependerá de las necesidades en los módulos de especialidad.

El sexto tema es de suma importancia y relevancia para la carrera de ingeniería electrónica, ya que realiza comunicaciones de la PC con el mundo exterior. Este tema se ha utilizado el término “puertos de comunicación”, dividiendo en dos categorías, los puertos paralelos y los puertos seriales. En donde cada academia decidirá que puertos para sus necesidades son requeridos. Algunos protocolos propuestos son: RS-232, USB, Bluetooth, WiFi, I2.

El docente debe contar con el perfil propio de la carrera en Ingeniería Electrónica, con los conocimientos necesarios en cuanto a la programación en un lenguaje estructurado de alto nivel. De igual manera debe fomentar en el estudiante el interés por el desarrollo de programas para la búsqueda de soluciones a problemas existentes en el área (simulación, instrumentación, análisis, etc.).

### 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: Aguascalientes, Apizaco, Cajeme, Celaya, Chapala, Chihuahua, Ciudad Guzmán, Ciudad Juárez, Cosamaloapan, Cautla, Culiacán, Durango, Ecatepec, Ensenada, Hermosillo, Irapuato, La Laguna, Lázaro Cárdenas, Lerdo, Lerma, Los Mochis, Matamoros, Mérida, Mexicali, Minatitlán, Nuevo Laredo, Orizaba, Piedras Negras, Reynosa, Salina Cruz, Saltillo, Sur De Guanajuato, Tantoyuca, Tijuana, Toluca, Tuxtepec, Veracruz y Xalapa.	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: Aguascalientes, Apizaco, Cajeme, Celaya, Chapala, Chihuahua, Ciudad Guzmán, Ciudad Juárez, Cosamaloapan, Cautla, Durango, Ecatepec, Ensenada, Hermosillo, Irapuato, La Laguna, Lázaro Cárdenas, Lerdo, Lerma, Los Mochis, Matamoros, Mérida, Mexicali, Minatitlán, Nuevo Laredo,	Reunión Nacional de Consolidación de los Programas en Competencias Profesionales de las Carreras de Ingeniería Eléctrica, Ingeniería Electromecánica, Ingeniería Electrónica e Ingeniería Mecatrónica.

	Orizaba, Piedras Negras, Reynosa, Salina Cruz, Saltillo, Sur De Guanajuato, Tantoyuca, Toluca, Tuxtepec, Veracruz y Xalapa.	
Instituto Tecnológico de la Laguna, del 26 al 29 de noviembre de 2012.	Representantes de los Institutos Tecnológicos de: Orizaba, Querétaro, Celaya, Aguascalientes, Alvarado, Cuautitlán Izcalli, La Laguna y Lerdo.	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

<b>Competencia(s) específica(s) de la asignatura</b>
Diseña algoritmos y desarrolla programas de aplicación, utilizando un lenguaje de programación estructurado de alto nivel, para su aplicación en la solución de problemas propios del área electrónica.

## 5. Competencias previas

- Uso de computadora personal
- Capacidad de abstracción, análisis y síntesis
- Habilidades en el uso de las tecnologías de la información y de la comunicación.
- Capacidad crítica y autocrítica.
- Uso del multímetro para medición de voltaje, corriente.
- Empleo de diodos emisores de luz
- Uso del osciloscopio.

## 6. Temario

No.	Temas	Subtemas
1	Fundamentos de programación	1.1. Importancia de la programación de computadoras. 1.2. Clasificación de los tipos de lenguajes de programación. 1.3. Diseño de algoritmos. 1.4. Diagramas de flujo. 1.4.1. Uso de programas de simulación para diagramas de flujo. 1.5. Importancia del compilador en los lenguajes de alto nivel.
2	Elementos del lenguaje de programación	2.1. Introducción al entorno de programación. 2.2. Estructura básica de un programa. 2.2.1. Comentarios. 2.2.2. Identificadores. 2.2.3. Palabras reservadas. 2.2.4. Clases. 2.2.5. Tipos de datos. 2.2.5.1. Simples. 2.2.5.2. Compuestos. 2.2.6. Variables y Constantes. 2.2.7. Atributos. 2.2.8. Operadores. 2.2.8.1. Aritméticos. 2.2.8.2. Lógicos. 2.2.8.3. Condicionales. 2.2.8.4. De desplazamiento. 2.2.9. Manejo de cadena de caracteres.
3	Estructuras de control	3.1. Estructuras de selección. 3.1.1. IF 3.1.2. IF/ELSE 3.1.3. IF/ELSEIF 3.2. Estructuras de repetición. 3.2.1. WHILE

		<p>3.2.2. DO WHILE 3.2.3. FOR 3.3. Estructura de múltiple selección. 3.3.1. SWITCH/CASE 3.4. Formulación y aplicación de algoritmos utilizando estructuras de control.</p>
4	Funciones	<p>4.1. Estructura de la función. 4.2. Llamado o invocación de una función. 4.3. Uso de funciones con parámetros. 4.3.1. De entrada. 4.3.2. De salida. 4.4. Funciones externas. 4.4.1. Del usuario. 4.4.2. De bibliotecas.</p>
5	Arreglos	<p>5.1. Definición e importancia de los arreglos en la programación. 5.2. Declaración de arreglos. 5.2.1. Vectores. 5.2.2. Matrices. 5.3. Lectura escritura de arreglos. 5.3.1. Vectores. 5.3.2. Matrices. 5.4. Operaciones con arreglos.</p>
6	Uso de puertos de comunicación	6.1. Proyecto.

## 7. Actividades de aprendizaje de los temas

1. Fundamentos de programación	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <ul style="list-style-type: none"> <li>• Conoce la importancia y la clasificación de los lenguajes de programación, dando realce a la programación de alto nivel.</li> <li>• Desarrolla algoritmos y diagramas de flujo para agilizar el pensamiento lógico en la solución de problemas.</li> </ul> <p>Genéricas:</p> <ul style="list-style-type: none"> <li>• Capacidad cognitiva para comprender y manipular ideas, pensamientos e información.</li> <li>• Capacidad para aprender.</li> <li>• Capacidad crítica y autocrítica.</li> <li>• Trabajo en equipo</li> <li>• Capacidad de investigación</li> <li>• Capacidad de comunicación oral y escrita</li> </ul>	<ul style="list-style-type: none"> <li>• Investigar en diferentes fuentes de información los lenguajes de programación y su clasificación.</li> <li>• Exponer frente a grupo los resultados de la investigación previamente realizada.</li> <li>• Desarrollar algoritmos a partir de problemas perfectamente delimitados usando pseudocódigo y diagramas de flujo.</li> <li>• Simular el algoritmo mediante diagramas de flujo. Algunos software para esta actividad son: <ul style="list-style-type: none"> <li>○ Raptor</li> <li>○ DFD</li> <li>○ Visustin</li> </ul> </li> </ul>

2. Elementos del lenguaje de programación	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <ul style="list-style-type: none"> <li>• Conoce el entorno de programación para utilizar adecuadamente las herramientas de diseño para el desarrollo de programas</li> <li>• Implementa programas a partir del diseño de algoritmos aplicaciones básicas en la solución de problemas reales.</li> </ul> <p>Genéricas:</p> <ul style="list-style-type: none"> <li>• Capacidad creativa</li> <li>• Capacidad cognitiva para comprender y manipular ideas, pensamientos e información.</li> <li>• Capacidad para aprender.</li> <li>• Capacidad crítica y autocrítica.</li> <li>• Trabajo en equipo.</li> <li>• Capacidad de investigación.</li> <li>• Capacidad de comunicación oral y escrita</li> <li>• Capacidad para tomar decisiones</li> </ul>	<ul style="list-style-type: none"> <li>• Familiarizarse con el entorno de programación por medio del diseño, edición y compilación de programas sencillos</li> <li>• Investigar cuales son las palabras reservadas del lenguaje específico a utilizar.</li> <li>• Desarrollar programas que utilicen variables, constantes y los diferentes tipos de datos</li> <li>• Desarrollar programas que utilicen los operadores aritméticos, lógicos y relacionales.</li> <li>• Desarrollar programas que utilicen la mayoría de los puntos vistos en la estructura básica de un programa con la finalidad de dar solución a problemas cotidianos.</li> </ul>
3. Estructuras de control	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <p>Implementa programas a partir del diseño de algoritmos aplicaciones en la solución de problemas reales.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> <li>• Capacidad creativa</li> <li>• Capacidad cognitiva para comprender y manipular ideas, pensamientos e información.</li> <li>• Capacidad para aprender.</li> <li>• Capacidad crítica y autocrítica.</li> <li>• Trabajo en equipo.</li> <li>• Capacidad para tomar decisiones</li> <li>• Capacidad de investigación.</li> <li>• Capacidad de comunicación oral y escrita</li> </ul>	<ul style="list-style-type: none"> <li>• Desarrollar programas que utilicen estructuras de selección y repetición.</li> <li>• Diseñar algoritmos más complejos donde se vincule los conocimientos vistos en la unidad anterior.</li> </ul>
4. Funciones	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <p>Implementar programas que interactúen entre funciones, permitiendo observar las ventajas y desventajas de este tipo de programación.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> <li>• Capacidad creativa</li> </ul>	<ul style="list-style-type: none"> <li>• Investigar los elementos que conforman una función o método o subfunción, estos términos dependiendo el lenguaje a utilizar.</li> <li>• Retomar los programas previamente elaborados y crear funciones a partir de ellos.</li> </ul>

<ul style="list-style-type: none"> <li>• Capacidad cognitiva para comprender y manipular ideas, pensamientos e información.</li> <li>• Capacidad para aprender.</li> <li>• Capacidad crítica y autocrítica.</li> <li>• Trabajo en equipo.</li> <li>• Capacidad para tomar decisiones</li> <li>• Capacidad de investigación.</li> <li>• Capacidad de comunicación oral y escrita</li> </ul>	<ul style="list-style-type: none"> <li>• Los programas ya diseñados con funciones, deberán ser invocados en un programa principal, observando las ventajas y desventajas de este tipo de programación.</li> <li>• Desarrollar programas que utilicen bibliotecas de funciones de entrada y salida, de manejo de archivos y manipulación de cadenas.</li> </ul>
5. Arreglos	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <ul style="list-style-type: none"> <li>• Implementar programas que contengan arreglos unidimensionales y multidimensionales.</li> <li>• Comprende la lectura y escritura en arreglos observando el recorrido de filas y columnas</li> <li>• Analiza y desarrolla algoritmos con arreglos para realizar operaciones entre ellos.</li> </ul> <p>Genéricas:</p> <ul style="list-style-type: none"> <li>• Capacidad creativa</li> <li>• Capacidad cognitiva para comprender y manipular ideas, pensamientos e información.</li> <li>• Capacidad para aprender.</li> <li>• Capacidad crítica y autocrítica.</li> <li>• Trabajo en equipo.</li> <li>• Capacidad para tomar decisiones</li> <li>• Capacidad de investigación.</li> <li>• Capacidad de comunicación oral y escrita</li> </ul>	<ul style="list-style-type: none"> <li>• Define la importancia del uso de arreglos en la programación estructurada y su relación con la electrónica en semestres posteriores.</li> <li>• Desarrolla programas que utilicen arreglos unidimensionales y multidimensionales con ejemplos didácticos para comprensión de la lectura y escritura, y modificación de datos.</li> <li>• Desarrolla programas con arreglos que permitan operaciones entre ellos.</li> </ul>
6. Uso de puertos de comunicación	
Competencias	Actividades de aprendizaje
<p>Específica(s):</p> <p>Desarrollar programas que incluyan manejo de puertos para permitir la interacción con sistemas electrónicos externos a la computadora.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> <li>• Capacidad creativa</li> <li>• Capacidad cognitiva para comprender y manipular ideas, pensamientos e información.</li> <li>• Capacidad para aprender.</li> <li>• Capacidad crítica y autocrítica.</li> <li>• Trabajo en equipo.</li> <li>• Capacidad para tomar decisiones</li> </ul>	<ul style="list-style-type: none"> <li>• Investigar y exponer frente a grupo los diferentes tipos de puertos de comunicación, dividiéndolos en dos grupos: seriales y paralelos.</li> <li>• Comprender la interacción de estos puertos con los sistemas electrónicos</li> <li>• Desarrollar programas que permitan interactuar con sistemas electrónicos externos a la computadora (motores, leds, actuadores, sensores, etc.).</li> </ul>

- |   |  |
|---|--|
| <ul style="list-style-type: none"> <li>• Capacidad de investigación.</li> <li>• Capacidad de comunicación oral y escrita</li> </ul> |  |
|---|--|

## 8. Práctica(s)

- Manejo del entorno de programación
- Desarrollar un programa que comprenda la estructura básica del lenguaje.
- Desarrollar un programa que muestre el valor almacenado en una constante y en variables de los tipos de datos básicos
- Desarrollar un programa que haga uso de tipos de datos abstractos
- Desarrollar un programa que calcule el área, perímetro y diámetro de un círculo a partir de la declaración de una constante ( $\pi$ ) y la asignación de valor del radio. Realizarlo con diferentes valores
- Repetir la práctica anterior solicitando el ingreso de los valores del radio por medio del teclado.
- Desarrollar un programa que realice la conversión entre sistemas numéricos
- Desarrollar un programa que a partir del ingreso de valores binarios de 8 bits, realice las operaciones lógicas (and, or, xor, xnor)
- Desarrollar un programa que utilice una función que calcule el factorial de un número
- Desarrollar un programa que a partir de un rango de años obtenga los que son bisesto
- Desarrollar un programa que genere números aleatorios y se almacenen en un archivo
- Desarrollar un programa que genere números de la serie de fibbonacci y se almacenen en un archivo
- Desarrollar un programa que muestre un menú para la realización de operaciones con matrices
- Desarrollar un programa que realice una animación de led's.

## 9. Proyecto de asignatura

El objetivo del proyecto que planteé 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

Instrumentos y herramientas sugeridas para evaluar las actividades de aprendizaje:

- Reportes investigación que cubran los requisitos establecidos por el profesor.
- Reportes de práctica vistas en clase.
- Evaluación de las exposiciones que cubran una lista de cotejo establecida por el profesor.
- Desarrollo de programas que cubran los contenidos plasmados en el programa de estudios.
- Desarrollo de programas derivados de problemas planteados por el docente.

## 11. Fuentes de información

1. Sharp, J. (2010). *Microsoft Visual C# 2010 Step by Step*. Microsoft Press.
2. Bell, D. (2010). *C# para estudiantes*. Addison Wesley.
3. Carrasco, R., Patiño, I., & SANTOS, M. (2006). *Fundamentos De Programación - 2ª ed.* . Ra-Ma, Alfaomega.
4. Ceballos, F. (2010). *El Lenguaje De Programación C# - 2ª ed.* . Alfaomega, Ra-Ma .
5. Ceballos, F. J. (2008). *Microsoft C# - Curso De Programación*. Alfaomega, Ra-Ma .
6. Ceballos, F. J. (2008). *Microsoft C# - Lenguaje Y Aplicaciones – 2ª ed.* Alfaomega, Ra-Ma
7. Ceballos, F. J. (2010). *Enciclopedia De Microsoft Visual C# - 3ª ed.* Alfaomega, Ra-Ma .
8. Deitel, D., & Deitel, H. (n.d.). *C# 2010 for Programmers*. 2010: Prentice Hall.
9. Deitel, H., Deitel, P., Listfield, J., Nieto, T., Yaeger, C., & Zlatkina, M. (2001). *C# How to Program*. Prentice Hall (Pearson).
10. Ferguson, J., Patterson, B., Beres, J., Boutquin, P., & Gupta, M. (2003). *La biblia de C #*. Anaya Multimedia.
11. Liberty, J. (2010). *Programming C #*. O'Reilly.