

## 1.- DATOS DE LA ASIGNATURA

Nombre de la asignatura :	<b>Algoritmos y Programación</b>
Carrera :	<b>Ingeniería Mecánica</b>
Clave de la asignatura :	<b>MEA-1001</b>
SATCA <sup>1</sup>	<b>0-4-4</b>

## 2.- PRESENTACIÓN

### **Caracterización de la asignatura.**

Esta asignatura aporta al perfil del Ingeniero Mecánico los conocimientos, habilidades, metodología, así como capacidades de análisis y síntesis, para plantear la solución de problemas susceptibles de ser computarizados, a través de diagramas de flujo, pseudocódigo y algoritmos utilizando las estructuras de un lenguaje de programación estructurado.

### **Intención didáctica.**

La materia se conforma de 8 unidades las cuales permiten que el alumno adquiera los conocimientos básicos relacionados la solución de problemas utilizando una computadora como herramienta para implementar soluciones prácticas. Se sugiere un ambiente práctico, donde el profesor propone el planteamiento de un problema y el estudiante deberá resolverlo mediante diversas técnicas, por mencionar algunas: diagramas de flujo, pseudocódigo y herramientas de modelado; con el monitoreo del profesor.

En la primera unidad se tratan temas básicos de la historia de los lenguajes de programación así como una introducción a la abstracción de datos. En la segunda unidad da una introducción a la sintaxis de lenguaje de programación a utilizar. En la tercera unidad se presentan los controles de flujo que nos ayudan a diseñar programas básicos. En la cuarta unidad se presentan las funciones predeterminadas y las funciones definidas por el usuario. En la quinta unidad se presentan los arreglos como una forma básica de almacenamiento y manipulación de información. En la sexta unidad se presentan los apuntadores para gestionar y hacer eficiente el acceso a los datos. En la séptima unidad se presentan las funciones de cadena para el almacenamiento y manipulación de texto. En la octava unidad se presentan las estructuras como formas primitivas para el desarrollo de bases de datos.

---

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

### 3.- COMPETENCIAS A DESARROLLAR

<p><b>Competencias específicas:</b></p> <ul style="list-style-type: none"><li>▪ Identificar conceptos básicos, propiedades y características de un sistema computacional</li><li>▪ Valorar los elementos que conforma un equipo computacional en software, hardware y firmware.</li><li>▪ Realizar una clasificación de los sistemas operativos, valorando sus diferencias</li><li>▪ Aplicar un algoritmo para actividades cotidianas</li><li>▪ Expresar un algoritmo en un diagrama de flujo.</li><li>▪ Aplicar la codificación de los algoritmos con un lenguaje de programación (C o C++)</li><li>▪ Identificar las actividades en donde se utiliza la programación.</li><li>▪ Identificar el ambiente de programación a utilizar</li><li>▪ Investigar en Internet sobre las tecnologías de la información.</li><li>▪ Fomentar actividades grupales que propicien la comunicación, el intercambio argumentado de ideas y la sugerencia del profesor</li><li>▪ Proponer problemas que permitan al estudiante la integración de contenidos de la asignatura y entre distintas asignaturas para su análisis y solución.</li></ul>	<p><b>Competencias genéricas:</b></p> <p><u>Competencias instrumentales</u></p> <ul style="list-style-type: none"><li>• Pensamiento lógico, algorítmico, heurístico, analítico y sintético</li><li>• Capacidad de análisis y síntesis</li><li>• Capacidad de organización y planificación</li><li>• Conocimientos básicos de matemáticas</li><li>• Comunicación oral y escrita</li><li>• Conocimientos de informática</li><li>• Capacidad para procesar e interpretar datos</li><li>• Solución de problemas</li><li>• Potenciar las habilidades para el uso de lenguajes de programación</li><li>• Toma de decisiones</li></ul> <p><u>Competencias interpersonales</u></p> <ul style="list-style-type: none"><li>• Razonamiento crítico</li><li>• Trabajo en equipo</li></ul> <p><u>Competencias sistémicas</u></p> <ul style="list-style-type: none"><li>• Capacidad de aplicar los conocimientos en la práctica</li><li>• Habilidades de investigación</li><li>• Capacidad de aprender</li><li>• Capacidad de generar nuevas ideas (creatividad)</li><li>• Habilidad para trabajar en forma autónoma</li><li>• Liderazgo</li></ul>
---	--

#### 4.- HISTORIA DEL PROGRAMA

Lugar y fecha de elaboración o revisión	Participantes	Evento
<p>Instituto Tecnológico de Estudios Superiores de Ecatepec del 9 al 13 de noviembre de 2009.</p>	<p>Representantes de los Institutos Tecnológicos de:  Aguascalientes, Superior de Alvarado, Boca del Río, Campeche, Celaya, Ciudad Guzmán, Ciudad Juárez, Superior de Ciudad Serdán, Ciudad Victoria, Superior de Coahuila de Zaragoza, Culiacán, Durango, Estudios Superiores de Ecatepec, Hermosillo, La Laguna, Mérida, Superior de Monclova, Orizaba, Pachuca, Saltillo, San Luis Potosí, Superior de Tepexi de Rodríguez y Tuxtla Gutiérrez.</p>	<p>Reunión Nacional de Diseño e Innovación Curricular para el Desarrollo y Formación de Competencias Profesionales de la Carrera de Ingeniería Mecánica.</p>
<p>Desarrollo de Programas en Competencias Profesionales por los Institutos Tecnológicos del 16 de noviembre de 2009 al 26 de mayo de 2010.</p>	<p>Academias de Ingeniería Mecánica de los Institutos Tecnológicos de:  Ciudad Juárez y Mérida.</p>	<p>Elaboración del programa de estudio propuesto en la Reunión Nacional de Diseño Curricular de la Carrera de Ingeniería Mecánica.</p>
<p>Instituto Tecnológico de Zacatecas del 12 al 16 de abril de 2010.</p>	<p>Representantes de los Institutos Tecnológicos de:  Aguascalientes, Superior de Alvarado, Boca del Río, Campeche, Celaya, Ciudad Guzmán, Ciudad Juárez, Superior de Ciudad Serdán, Ciudad Victoria, Superior de Coahuila de Zaragoza, Culiacán, Durango Estudios Superiores de Ecatepec, Hermosillo, La Laguna, La Piedad, Mérida, Superior de Monclova, Orizaba, Pachuca, Saltillo, San Luis Potosí, Superior de Tepexi de Rodríguez y Tuxtla Gutiérrez.</p>	<p>Reunión Nacional de Consolidación de los Programas en Competencias Profesionales de la Carrera de Ingeniería Mecánica.</p>

## 5.- OBJETIVO GENERAL DEL CURSO

Comprender y aplicar los conceptos de la programación estructurada en la construcción de programas utilizando un lenguaje de programación para la solución de problemas.

## 6.- COMPETENCIAS PREVIAS

- Conocimientos básicos sobre la arquitectura de la computadora y de los sistemas numéricos.
- Utilizar equipo de cómputo y software básico
- Aplicar los conocimientos básicos de matemáticas (álgebra, trigonometría y geometría analítica)
- Analizar, sintetizar y abstraer.
- Aplicar los conocimientos en la práctica.

## 7.- TEMARIO

Unidad	Temas	Subtemas
1.	Introducción a la computación	1.1. Historia de los Lenguajes de programación. 1.2. Procesos de edición, compilación, enlazado 1.3. Entorno del lenguaje "C" y software de aplicación de última generación 1.4. Diagramas de flujo 1.5. Pseudocódigos 1.6. Estructura general de un programa 1.7. Conceptos básicos de identificador, localidad de memoria y Palabras reservadas.
2.	Operadores, variables, constantes, tipos de datos y expresiones	2.1. Elementos de léxico y sintaxis 2.2. Variables y constantes 2.3. Tipos de datos fundamentales 2.4. Operadores y precedencia 2.5. Construcción de expresiones 2.6. Sentencias y bloques de sentencias de programa. 2.7. Funciones básicas de entrada y salida.
3.	Control de flujo de programa	3.1. Control secuencial 3.2. Control selectivo 3.3. Control repetitivo.
4.	Funciones básicas	4.1. Funciones de biblioteca 4.2. Definición de funciones 4.3. Declaración e invocación de funciones.
5.	Arreglos	5.1. Inicialización y generación de arreglos 5.2. Arreglos multidimensionales 5.3. Arreglos de caracteres 5.4. Utilización de arreglos.
6.	Apuntadores	6.1. Representación de un apuntador 6.2. Precedencia de operadores de

		apuntadores 6.3. Llamadas por referencia 6.4. Utilización de apuntadores.
7.	Cadenas	7.1. Fundamentos de cadenas y caracteres 7.2. Funciones de manejo de cadenas y caracteres.
8.	Estructuras	8.1. Definición de estructuras 8.2. Inicialización de estructuras 8.3. Utilización de estructuras con funciones.

## **8.- SUGERENCIAS DIDÁCTICAS**

El docente debe:

- Propiciar el trabajo en equipo.
- Desarrollar ejemplos de lo simple a lo complejo, buscando que el estudiante, asocie el tema con elementos significativos de su entorno.
- Propiciar actividades de búsqueda, selección y análisis de información en distintas fuentes.
- Propiciar el uso de las nuevas tecnologías en el desarrollo de los contenidos de la asignatura.
- Fomentar actividades grupales que propicien la comunicación, el intercambio argumentado de ideas, la reflexión, la integración y la colaboración de y entre los estudiantes.
- Propiciar, en el estudiante, el desarrollo de actividades intelectuales de inducción-deducción y análisis-síntesis, las cuales lo encaminan hacia la investigación, la aplicación de conocimientos y la solución de problemas.
- Desarrollar actividades de aprendizaje que propicien la aplicación de los conceptos, modelos y metodologías que se van aprendiendo en el desarrollo de la asignatura.
- Propiciar el uso adecuado de conceptos, y de terminología científico tecnológico.
- Proponer problemas que permitan al estudiante la integración de contenidos de la asignatura y entre distintas asignaturas, para su análisis y solución.
- Observar y analizar fenómenos y problemáticas propias del campo ocupacional.
- Relacionar los contenidos de esta asignatura con las demás del plan de estudios para desarrollar una visión interdisciplinaria en el estudiante.

## 9.- SUGERENCIAS DE EVALUACIÓN

La evaluación debe ser continua y formativa por lo que se debe considerar el desempeño en cada una de las actividades de aprendizaje, haciendo especial énfasis en:

- La solución de problemas específicos empleando el uso de la computadora como herramienta.
- Reportes escritos de las soluciones planteadas durante las actividades, así como de las conclusiones obtenidas de dichas soluciones.
- Investigaciones documentales, después de lo cual se reunirán para realizar una lluvia de ideas o bien mesas redondas, donde los estudiantes interactúan con la información investigada de cada equipo.
- Elaborar resúmenes de lo consultado bibliográficamente, para desarrollar el tema elegido por los estudiantes y avanzar en su trabajo de investigación.
- Elaborar mapas conceptuales, por equipo, de los temas explicados en el aula, con el fin de reforzar el aprendizaje adquirido.
- Usar tecnología de información (internet, revistas electrónicas, libros, artículos, etc.) para efectuar una recopilación de temas afines a los contenidos temáticos de su interés.
- Exámenes escritos y prácticos por unidades de aprendizaje.
- Explicar por equipo frente al grupo los sistemas computacionales.

## 10.- UNIDADES DE APRENDIZAJE

### Unidad 1: Introducción a la computación

<i>Competencia específica a desarrollar</i>	<i>Actividades de Aprendizaje</i>
Aplicar los conceptos del entorno de programación para estructurar un programa de solución a un problema con el uso de la computadora.	<ul style="list-style-type: none"> <li>• Desarrollar una investigación en distintas fuentes y elaborar un informe donde identifique las principales etapas de la evolución que ha tenido el lenguaje a utilizar.</li> <li>• Analizar y discutir en el aula la investigación realizada en el punto anterior, donde se resalten las diferentes etapas.</li> <li>• Realizar un mapa conceptual sobre los tipos de software y los conceptos básicos de programación.</li> <li>• Emplear software para diseño y validación de diagramas de flujo.</li> <li>• Emplear software para generar código a partir de diagramas de flujo.</li> <li>• Trabajo en equipo para la solución de casos prácticos.</li> </ul>

### Unidad 2: Operadores, variables, constantes, tipos de datos y expresiones

<i>Competencia específica a desarrollar</i>	<i>Actividades de Aprendizaje</i>
Identificar y aplicar los operadores, variables, constantes, tipos de datos y expresiones del lenguaje de programación, para desarrollar programas de aplicación.	<ul style="list-style-type: none"> <li>• Investigar y describir la sintaxis del lenguaje a utilizar.</li> <li>• Configurar y manipular el entorno de desarrollo del lenguaje, resolviendo</li> </ul>

	<p>problemas que hagan énfasis a cada uno de los elementos de éste.</p> <ul style="list-style-type: none"> <li>• Desarrollar un programa de aplicación en el cual globalice los puntos comprendidos en la unidad.</li> <li>• Realizar ejercicios de codificación de expresiones aritméticas y lógicas en un lenguaje de programación.</li> <li>• Buscar la información necesaria para instalar y configurar el compilador del lenguaje de programación a utilizar</li> <li>• Compilar y ejecutar un programa modelo.</li> <li>• Realizar cambios en expresiones lógicas y algebraicas de un programa modelo y analizar los resultados obtenidos.</li> </ul>
--	---

### Unidad 3: Control de flujo de programa

<i>Competencia específica a desarrollar</i>	<i>Actividades de Aprendizaje</i>
Aplicar los diferentes tipos de controles que intervienen en el flujo de un programa.	<ul style="list-style-type: none"> <li>• Realizar una síntesis sobre el funcionamiento y aplicación de las estructuras secuenciales y selectivas.</li> <li>• Compilar y ejecutar un programa modelo.</li> <li>• Implementar el diseño de programas que requieran estructuras secuenciales y selectivas para probarlas en una aplicación.</li> </ul>

### Unidad 4: Funciones básicas

<i>Competencia específica a desarrollar</i>	<i>Actividades de Aprendizaje</i>
Diseñar funciones definidas por el usuario para resolver problemas específicos.	<ul style="list-style-type: none"> <li>• Compilar y ejecutar un programa modelo</li> <li>• Realizar ejemplos que requieran funciones estándar.</li> <li>• Realizar ejemplos utilizando funciones definidas por el usuario.</li> <li>• Desarrollar programas que ejemplifiquen el uso de funciones definidas por el programador.</li> <li>• Elaborar una guía rápida, en la que se describan las funciones de bibliotecas o librerías que proporciona el lenguaje de programación, propuestas por el profesor.</li> <li>• Elaborar un programa que permita ejemplificar algunas de las funciones descritas en el punto anterior.</li> </ul>

### Unidad 5: Arreglos

<i>Competencia específica a desarrollar</i>	<i>Actividades de Aprendizaje</i>
---	-----------------------------------



<p>Aplicar el concepto y tratamiento de los arreglos estructurales.</p>	<ul style="list-style-type: none"> <li>• Compilar y ejecutar un programa modelo.</li> <li>• Desarrollar los programas de manipulación de los arreglos para realizar operaciones básicas.</li> <li>• Realizar ejemplos utilizando funciones y arreglos.</li> </ul>
---	---

### Unidad 6: Apuntadores

<i>Competencia específica a desarrollar</i>	<i>Actividades de Aprendizaje</i>
<p>Utilizar la herramienta de apuntadores para hacer más eficientes y flexibles los programas.</p>	<ul style="list-style-type: none"> <li>• Realizar una síntesis sobre el funcionamiento y aplicación de los apuntadores.</li> <li>• Implementar el diseño de programas que requieran apuntadores para probarlos en una aplicación.</li> </ul>

### Unidad 7: Cadenas

<i>Competencia específica a desarrollar</i>	<i>Actividades de Aprendizaje</i>
<p>Aplicar el concepto de cadena y sus diferentes funciones predefinidas.</p>	<ul style="list-style-type: none"> <li>• Realizar una síntesis sobre el funcionamiento y aplicación de las funciones cadena.</li> <li>• Desarrollar los programas de manipulación de las funciones de cadena para realizar operaciones básicas</li> <li>• Implementar el diseño de programas que requieran funciones de cadena para probarlas en una aplicación.</li> </ul>

### Unidad 8: Estructuras

<i>Competencia específica a desarrollar</i>	<i>Actividades de Aprendizaje</i>
<p>Aprender el concepto y tratamiento de las estructuras de programas.</p>	<ul style="list-style-type: none"> <li>• Explicar los conceptos básicos de las estructuras para diferentes tipos de datos, así como el enfoque usado para su manejo computacional.</li> <li>• Elaborar algoritmos y programas para la solución de problemas que involucren estructuras.</li> </ul>

## 11.- FUENTES DE INFORMACIÓN

1. Joyanes Aguilar L. (2001). "Fundamentos de programación, algoritmos y estructuras de datos", McGraw-Hill.
2. Joyanes Aguilar L. (2002). "Problemas de Metodología de la Programación", McGraw-Hill.
3. Byron G. (1998). "Programación en C". Ed. McGraw-Hill.
4. Kernighan B. y Ritchie D.M. (1994). "El lenguaje de Programación C" Ed. Prentice Hall.
5. Deitel H.M. y Deitel P.J. (1994). "Cómo Programar en C/C++". Ed. Prentice Hall.
6. Porter A. (1994). "Programación en C++ para Windows". Ed. McGraw-Hill.
7. Savitch W. (2000). "Resolución de problemas con C++". Ed. Prentice Hall.
8. Peñaloza Romero E. (2004). "Fundamentos de programación". Ed. Alfaomega.
9. Ceballos F.J. (1998). "Curso de Programación C++", Addison Wesley.
10. Farrell J. (2001). "Introducción a la programación", Thomson Internacional.

## 12.- PRÁCTICAS PROPUESTAS

### Unidad 1

- Utilizando diagrama de flujo, diagrama N-S, diagrama estructurado y pseudocódigo, elaborar algoritmos.
- Se pueden utilizar problemas presentados por el facilitador o utilizar problemas presentados por el alumno.

### Unidad 2

- Elaborar ejercicios que impliquen el uso de operadores, operandos y expresiones aritméticas.

### Unidad 3

- Implementar aplicaciones que utilicen funciones con comportamientos que impliquen el uso de estructuras selectivas y expresiones lógicas.

### Unidad 4

- Implementar aplicaciones que utilicen funciones que impliquen el uso de estructuras repetitivas.

### Unidad 5

- El alumno desarrollará arreglos incluyendo todas las operaciones básicas que operan sobre un arreglo, tales como crear, insertar, eliminar, recorrer, buscar y modificar.

### Unidad 6

- Implementar aplicaciones que utilicen apuntadores.

### Unidad 7

- Implementar aplicaciones que utilicen funciones de cadena para manipulación de texto.

### Unidad 8

- Implementar aplicaciones que utilicen estructuras de datos para almacenar la información en bases de datos.