

1. Datos Generales de la asignatura

Nombre de la asignatura:	Algoritmos y Programación
Clave de la asignatura:	MEA-1001
SATCA¹:	0-4-4
Carrera:	Ingeniería Mecánica

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.

3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Evento
Instituto Tecnológico de Estudios Superiores de Ecatepec del 9 al 13 de noviembre de 2009.	Representantes de los Institutos Tecnológicos de: Aguascalientes, Superior de Alvarado, Boca del Río, Campeche, Celaya, Ciudad Guzmán, Ciudad Juárez, Superior de Ciudad Serdán,	Reunión Nacional de Diseño e Innovación Curricular para el Desarrollo y Formación de Competencias Profesionales de las Carreras de Ingeniería en Materiales, Ingeniería Mecánica e Ingeniería Industrial.

¹ Sistema de Asignación y Transferencia de Créditos Académicos

	Ciudad Victoria, Superior de Coatzacoalcos, 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.	
Instituto Tecnológico de Zacatecas del 12 al 16 de abril de 2010.	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 Coatzacoalcos, 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.	Reunión Nacional de Consolidación de los Programas en Competencias Profesionales de las Carreras de Ingeniería en Materiales, Ingeniería Mecánica e Ingeniería Industrial.
Instituto Tecnológico de la Laguna, del 26 al 29 de noviembre de 2012.	Representantes de los Institutos Tecnológicos de: Aguascalientes, Alvarado, Boca del Río, Cajeme, Cd. Serdán, Cd. Victoria, Chihuahua, Culiacán, La Laguna, Pachuca, Querétaro, Tláhuac II y Veracruz.	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,	Reunión de trabajo para la actualización de los planes de estudio del sector energético, con la participación de PEMEX.

	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).	
--	--	--

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura
Comprende y aplica los conceptos de la programación estructurada en la construcción de programas utilizando un lenguaje de programación para la solución de problemas.

5. Competencias previas

<ul style="list-style-type: none"> ■ 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.
--

6. Temario

No.	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.

7. Actividades de aprendizaje de los temas

I.Introducción a la computación	
Competencias	Actividades de aprendizaje
<p>Específica(s): Aplica los conceptos del entorno de programación para estructurar un programa de solución a un problema con el uso de la computadora.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de aplicar conocimientos en la práctica. • Conocimiento sobre el área de estudio y la profesión • Capacidad de investigación • Capacidad de identificar, plantear y resolver problemas. 	<ul style="list-style-type: none"> • 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. • Analizar y discutir en el aula la investigación realizada en el punto anterior, donde se resalten las diferentes etapas. • Realizar un mapa conceptual sobre los tipos de software y los conceptos básicos de programación. • Emplear software para diseño y validación de diagramas de flujo. • Emplear software para generar código a partir de diagramas de flujo. • Trabajo en equipo para la solución de casos prácticos.

II. Operadores, variables, constantes, tipos de datos y expresiones	
Competencias	Actividades de aprendizaje
<p>Específica(s): Identifica y aplica los operadores, variables, constantes, tipos de datos y expresiones del lenguaje de programación, para desarrollar programas de aplicación.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de aplicar conocimientos en la práctica. • Conocimiento sobre el área de estudio y la profesión • Capacidad de investigación • Capacidad de identificar, plantear y resolver problemas. 	<ul style="list-style-type: none"> • Investigar y describir la sintaxis del lenguaje a utilizar. • Configurar y manipular el entorno de desarrollo del lenguaje, resolviendo problemas que hagan énfasis a cada uno de los elementos de éste. • Desarrollar un programa de aplicación en el cual globalice los puntos comprendidos en la unidad. • Realizar ejercicios de codificación de expresiones aritméticas y lógicas en un lenguaje de programación. • Buscar la información necesaria para Instalar y configurar el compilador del lenguaje de programación a utilizar • Compilar y ejecutar un programa modelo. • Realizar cambios en expresiones lógicas y algebraicas de un programa modelo y analizar los resultados obtenidos.
III. Control de flujo de programa	
Competencias	Actividades de aprendizaje
<p>Específica(s): Aplica los diferentes tipos de controles que intervienen en el flujo de un programa.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de aplicar conocimientos en la práctica. • Conocimiento sobre el área de estudio y la profesión • Capacidad de investigación • Capacidad de identificar, plantear y resolver problemas. 	<ul style="list-style-type: none"> • Realizar una síntesis sobre el funcionamiento y aplicación de las estructuras secuenciales y selectivas. • Compilar y ejecutar un programa modelo. • Implementar el diseño de programas que requieran estructuras secuenciales y selectivas para probarlas en una aplicación.
IV. Funciones básicas	
Competencias	Actividades de aprendizaje
<p>Específica(s): Diseña funciones definidas por el usuario para resolver problemas específicos.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de aplicar conocimientos en la práctica. • Conocimiento sobre el área de estudio y la profesión 	<ul style="list-style-type: none"> • Compilar y ejecutar un programa modelo • Realizar ejemplos que requieran funciones estándar. • Realizar ejemplos utilizando funciones definidas por el usuario. • Desarrollar programas que ejemplifiquen el uso de funciones definidas por el

<ul style="list-style-type: none"> • Capacidad de investigación • Capacidad de identificar, plantear y resolver problemas. 	<p>programador.</p> <ul style="list-style-type: none"> • 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. • Elaborar un programa que permita ejemplificar algunas de las funciones descritas en el punto anterior.
V.Arreglos	
Competencias	Actividades de aprendizaje
<p>Específica(s): Aplica el concepto y tratamiento de los arreglos estructurales.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de aplicar conocimientos en la práctica. • Conocimiento sobre el área de estudio y la profesión • Capacidad de investigación • Capacidad de identificar, plantear y resolver problemas. 	<ul style="list-style-type: none"> • Compilar y ejecutar un programa modelo. • Desarrollar los programas de manipulación de los arreglos para realizar operaciones básicas. • Realizar ejemplos utilizando funciones y arreglos.
VI.Apuntadores	
Competencias	Actividades de aprendizaje
<p>Específica(s): Utiliza la herramienta de apuntadores para hacer más eficientes y flexibles los programas.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de aplicar conocimientos en la práctica. • Conocimiento sobre el área de estudio y la profesión • Capacidad de investigación • Capacidad de identificar, plantear y resolver problemas. 	<ul style="list-style-type: none"> • Realizar una síntesis sobre el funcionamiento y aplicación de los apuntadores. • Implementar el diseño de programas que requieran apuntadores para probarlos en una aplicación.
VII.Cadenas	
Competencias	Actividades de aprendizaje
<p>Específica(s): Aplica el concepto de cadena y sus diferentes funciones predefinidas</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de aplicar conocimientos en la práctica. • Conocimiento sobre el área de estudio y la profesión • Capacidad de investigación 	<ul style="list-style-type: none"> • Realizar una síntesis sobre el funcionamiento y aplicación de las funciones cadena. • Desarrollar los programas de manipulación de las funciones de cadena para realizar operaciones básicas • Implementar el diseño de programas que requieran funciones de cadena para

<ul style="list-style-type: none"> • Capacidad de identificar, plantear y resolver problemas. 	probarlas en una aplicación.
VIII.Estructuras	
Competencias	Actividades de aprendizaje
<p>Específica(s): Conoce y Aplica el concepto y tratamiento de las estructuras de programas.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Capacidad de aplicar conocimientos en la práctica. • Conocimiento sobre el área de estudio y la profesión • Capacidad de investigación • Capacidad de identificar, plantear y resolver problemas. 	<ul style="list-style-type: none"> • Explicar los conceptos básicos de las estructuras para diferentes tipos de datos, así como el enfoque usado para su manejo computacional. • Elaborar algoritmos y programas para la solución de problemas que involucren estructuras.

8. Práctica(s)

<ul style="list-style-type: none"> ■ 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. ■ Elaborar ejercicios que impliquen el uso de operadores, operandos y expresiones aritméticas. ■ Implementar aplicaciones que utilicen funciones con comportamientos que impliquen el uso de estructuras selectivas y expresiones lógicas. ■ Implementar aplicaciones que utilicen funciones que impliquen el uso de estructuras repetitivas. ■ El alumno desarrollará arreglos incluyendo todas las operaciones básicas que operan sobre un arreglo, tales como crear, insertar, eliminar, recorrer, buscar y modificar. ■ Implementar aplicaciones que utilicen apuntadores. ■ Implementar aplicaciones que utilicen funciones de cadena para manipulación de texto. ■ Implementar aplicaciones que utilicen estructuras de datos para almacenar la información en bases de datos.

9. Proyecto de asignatura

<p>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:</p> <ul style="list-style-type: none"> • 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:

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:

Instrumentos:

- Reportes escritos de las observaciones hechas durante las actividades como reportes de visitas de campo, así como de las conclusiones obtenidas de dichas observaciones.
- Información obtenida durante las investigaciones solicitadas plasmada en documentos escritos.
- Exámenes escritos para comprobar el manejo de aspectos teóricos y declarativos para cada tema.
- Solución de ejercicios asignados por cada tema del programa.
- Reporte escrito de prácticas realizadas en laboratorio.
- Prototipos físicos. Desarrollo de prototipos.
- Prototipos virtuales. Elaboración de prototipos para simulación.

Herramientas:

- Rubrica
- Listas de cotejo
- Matrices de valoración
- Guías de observación

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. Peñalosa 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.