

1. Datos Generales de la asignatura

Nombre de la asignatura:	Fundamentos de Programación
Clave de la asignatura:	AED-1285
SATCA¹:	2-3-5
Carrera:	Ingeniería en Sistemas Computacionales e Ingeniería en Animación Digital y Efectos Visuales

2. Presentación

Caracterización de la asignatura

Esta asignatura aporta al perfil del egresado, la capacidad para desarrollar un pensamiento lógico a través del diseño de algoritmos utilizando herramientas de programación para el desarrollo de aplicaciones computacionales que resuelvan problemas reales.

Está diseñada para el logro de competencias específicas y genéricas dirigidas al aprendizaje de los conceptos básicos de la programación, siendo capaz de aplicar expresiones aritméticas y lógicas en un lenguaje de programación; así como el uso y funcionamiento de las estructuras secuenciales, selectivas y repetitivas, como también la organización de los datos, además de la declaración e implementación de funciones para construir y desarrollar aplicaciones de software que requieran dichas estructuras.

Se relaciona con todas aquellas asignaturas en donde se apliquen metodologías de programación y desarrollo de software de las carreras de Ingeniería en Sistemas Computacionales e Ingeniería en Animación Digital y Efectos Visuales.

Intención didáctica

La asignatura se organiza en cinco temas donde se abordan los conceptos esenciales del diseño algorítmico y de la programación, así como el uso de entornos de desarrollo.

En el primer tema, se estudian los conceptos básicos para introducir al estudiante en la programación con la finalidad de obtener el fundamento conceptual para el análisis, diseño y desarrollo de algoritmos, así como los primeros pasos al mundo de la programación a través de ejercicios dirigidos. En este tema es fundamental propiciar el pensamiento lógico del estudiante.

El segundo tema, tiene la finalidad de propiciar que el estudiante adquiera los conocimientos necesarios para diseñar e implementar soluciones en un lenguaje de programación y resolver problemas del contexto.

El tercer tema, tiene como objetivo que se comprenda, seleccione e implemente la estructura de control adecuada para la resolución de un problema específico.

El cuarto tema, se buscará propiciar la capacidad de aplicar e implementar los arreglos para

¹ Sistema de Asignación y Transferencia de Créditos Académicos

proporcionar un medio eficaz en la organización de datos para la resolución de problemas reales.

El quinto tema tiene como finalidad que el estudiante implemente el diseño de bloques de códigos reutilizables, que optimizan la funcionalidad y reducen la complejidad de los programas desarrollados.

Para cada uno de los temas se sugiere que el docente promueva la práctica a través de herramientas computacionales que propicien la habilidad en el desarrollo de programas.

3.- Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Evento
<p>Instituto Tecnológico de Aguascalientes del 15 al 18 de junio de 2010.</p>	<p>Representantes de los Institutos Tecnológicos de: Centro Interdisciplinario de Investigación y Docencia en Educación Técnica, Acapulco, Aguascalientes, Apizaco, Boca Río, Celaya, Chetumal, Chihuahua, Chilpancingo, Chiná, Cd. Cuauhtémoc, Cd. Juárez, Cd. Madero, Cd. Victoria, Colima, Comitán, Cuautla, Durango, El Llano de Aguascalientes, Huixquilucan, Valle Bravo, Guaymas, Huatabampo, Huejutla, Iguala, La Laguna, La Paz, La Zona Maya, León, Lerma, Linares, Los Mochis, Matamoros, Mazatlán, Mérida, Mexicali, Minatitlán, Nuevo Laredo, Orizaba, Pachuca, Puebla, Querétaro, Reynosa, Roque, Salina Cruz, Saltillo, San Luis Potosí, Tehuacán, Tepic, Tijuana, Tlaxiaco, Toluca, Torreón, Tuxtepec, Valle de Oaxaca, Veracruz, Villahermosa, Zacatecas, Zacatepec, Altiplano de Tlaxcala, Coatzacoalcos, Cuautitlán Izcalli, Fresnillo, Irapuato, La Sierra Norte Puebla, Macuspana, Naranjos, Pátzcuaro, Poza Rica, Progreso, Puerto Vallarta, Tacámbaro,</p>	<p>Elaboración del programa de estudio equivalente en la Reunión Nacional de Implementación Curricular y Fortalecimiento Curricular de las asignaturas comunes por área de conocimiento para los planes de estudio actualizados del SNEST.</p>

	Tamazula Gordiano, Tlaxco, Venustiano Carranza, Zacapoxtla, Zongólica y Oriente del Estado Hidalgo.	
Instituto Tecnológico de Morelia del 10 al 13 de septiembre de 2013.	Representantes de los Institutos Tecnológicos de: Aguascalientes, Apizaco, Boca del Río, Celaya, CRODE Celaya, Cerro Azul, Chihuahua, Cd. Cuauhtémoc, Cd. Hidalgo, Cd. Juárez, Cd. Madero, Cd. Valles, Coacalco, Colima, Iguala, La Laguna, Lerdo, Los Cabos, Matamoros, Mérida, Morelia, Motúl, Múzquiz, Nuevo Laredo, Nuevo León, Oriente del Estado de México, Orizaba, Pachuca, Progreso, Purhepecha, Salvatierra, San Juan del Río, Santiago Papasquiari, Tantoyuca, Tepic, Tlatlauquitpec, Valle de Morelia, Venustiano Carranza, Veracruz, Villahermosa, Zacatecas y Zacatepec.	Reunión Nacional de Seguimiento Curricular de las Asignaturas Equivalentes del SNIT.

4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura
Aplica algoritmos y lenguajes de programación para diseñar e implementar soluciones a problemáticas del entorno.

5. Competencias previas

<ul style="list-style-type: none"> Ninguna

6. Temario

No.	Temas	Subtemas
1.	Diseño Algorítmico	1.1 Conceptos básicos. 1.2 Representación de algoritmos: gráfica y pseudocódigo. 1.3 Diseño de algoritmos. 1.4 Diseño de funciones.
2.	Introducción a la Programación	2.1 Conceptos básicos. 2.2 Características del lenguaje de programación. 2.3 Estructura básica de un programa. 2.4 Elementos del lenguaje: tipos de datos, literales, constantes, variables, identificadores,

		<p>parámetros, operadores y salida de datos.</p> <p>2.5 Traducción de un programa: compilación, enlace, ejecución y errores.</p>
3.	Control de Flujo	<p>3.1 Estructuras secuenciales.</p> <p>3.2 Estructuras selectivas: simple, doble y múltiple.</p> <p>3.3 Estructuras iterativas: repetir mientras, hasta, desde.</p>
4.	Organización de datos	<p>4.1 Arreglos</p> <p>4.2 Unidimensionales: conceptos básicos, operaciones y aplicaciones.</p> <p>4.3 Multidimensionales: conceptos básicos, operaciones y aplicaciones.</p> <p>4.4 Estructuras o registros.</p>
5.	Modularidad	<p>5.1 Declaración y uso de módulos.</p> <p>5.2 Paso de parámetros o argumentos.</p> <p>5.3 Implementación.</p>

7.- Actividades de aprendizaje de los temas

Diseño algorítmico	
Competencias	Actividades de Aprendizaje
<p>Específica(s): Comprende y aplica los conceptos básicos, nomenclatura y herramientas para el diseño de algoritmos orientado a la resolución de problemas.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Habilidades de gestión de información (habilidad para buscar y analizar información proveniente de fuentes diversas). • Capacidad de análisis y síntesis. • Capacidad de comunicación oral y escrita. • Capacidad de aplicar los conocimientos en la práctica • Habilidades en el uso de las tecnologías de la información y de la comunicación. 	<ul style="list-style-type: none"> • Gestionar información sobre los conceptos básicos: algoritmo, programa, programación, paradigmas de programación, tipo de dato, constante, variable, operadores, diagrama de flujo, pseudocódigo y plasmarlos en un glosario. • Gestionar información sobre los tipos de datos en un lenguaje de programación y elaborar una tabla. • Gestionar información sobre la simbología de las herramientas para representar un algoritmo: Diagrama de Flujo y Pseudocódigo. Elaborar un resumen ejemplificando. • Representar mediante diagrama de flujo o pseudocódigo problemas cotidianos a partir de un problemario y elaborar una antología.
Introducción a la programación	
Competencias	Actividades de Aprendizaje
<p>Específica(s): Conoce y aplica un lenguaje de programación para la resolución de problemas.</p>	<ul style="list-style-type: none"> • Identificar las características de los diferentes lenguajes de programación que correspondan al paradigma de estudio y elaborar cuadro comparativo. • Elegir un lenguaje de programación e instalar y



<p>Genéricas:</p> <ul style="list-style-type: none"> • Habilidades de gestión de información (habilidad para buscar y analizar información proveniente de fuentes diversas). • Capacidad de análisis y síntesis. • Capacidad de comunicación oral y escrita. • Capacidad de aplicar los conocimientos en la práctica • Habilidades en el uso de las tecnologías de la información y de la comunicación. 	<p>configurar las herramientas correspondientes para su uso, auxiliándose de algún tutorial, presentar el entorno funcionando en clase.</p> <ul style="list-style-type: none"> • Identificar tipos de datos, literales, constantes, variables, identificadores, operadores y salida de datos propios del lenguaje de programación a través de una investigación y elaborar un reporte. • Aplicar las herramientas de programación, retomando un algoritmo planteado en el tema anterior, para su codificación, compilación, enlace, ejecución e identificación de errores, elaborar un reporte que incluya el código, los resultados de ejecución y exponerlo. • Analizar un programa completo donde identifique los elementos básicos (estructura y sintaxis) del lenguaje, documentando cada instrucción con comentarios. • Aplicar expresiones lógicas y algebraicas, para la solución de problemas, elaborar un reporte que incluya el código, los resultados de ejecución y exponerlo.
Control de flujo	
Competencias	Actividades de Aprendizaje
<p>Específica(s): Conoce y aplica las estructuras condicionales y repetitivas de un lenguaje de programación para resolver problemas reales.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Habilidades de gestión de información (habilidad para buscar y analizar información proveniente de fuentes diversas). • Capacidad de análisis y síntesis. • Capacidad de comunicación oral y escrita. • Capacidad de aplicar los conocimientos en la práctica • Habilidades en el uso de las tecnologías de la información y de la comunicación. 	<ul style="list-style-type: none"> • Investigar sobre la sintaxis, funcionamiento y aplicación de las estructuras de selección y de repetición realizar un cuadro comparativo sobre las estructuras de repetición. • Ejemplificar en programas que resuelvan problemas reales, discutirlo en grupo. • Realizar práctica de ejercicios, donde se utilicen las estructuras secuencial, selectiva e iterativa desarrollando algoritmo y programa, presentarlo y discutirlo en grupo
Organización de datos	
Competencias	Actividades de Aprendizaje
<p>Específica(s): Conoce y aplica estructuras de datos en un</p>	<ul style="list-style-type: none"> • Identificar las estructuras de datos: registros y arreglos unidimensionales, bidimensionales,



<p>lenguaje de programación que permitan la organización de datos en la resolución de problemas reales.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Habilidades de gestión de información (habilidad para buscar y analizar información proveniente de fuentes diversas). • Capacidad de análisis y síntesis. • Capacidad de comunicación oral y escrita. • Capacidad de aplicar los conocimientos en la práctica • Habilidades en el uso de las tecnologías de la información y de la comunicación. 	<p>multidimensionales y plasmarla en un cuadro sinóptico, así como la aplicación de las mismas a través de ejemplos y exponerlos en clase.</p> <ul style="list-style-type: none"> • Aplicar las herramientas para el desarrollo de programas, que implementen operaciones (almacenar y recuperar datos) sobre registros y arreglos (unidimensionales y multidimensionales), documentar y exponer en plenaria.
Modularidad	
Competencias	Actividades de Aprendizaje
<p>Específica(s): Conoce y aplica la modularidad en el desarrollo de programas para la optimización de los mismos y reutilización de código.</p> <p>Genéricas:</p> <ul style="list-style-type: none"> • Habilidades de gestión de información (habilidad para buscar y analizar información proveniente de fuentes diversas). • Capacidad de análisis y síntesis. • Capacidad de comunicación oral y escrita. • Capacidad de aplicar los conocimientos en la práctica • Habilidades en el uso de las tecnologías de la información y de la comunicación. 	<ul style="list-style-type: none"> • Investigar las ventajas, desventajas, estructura y usos de la programación modular y presentar un reporte. • Gestionar información sobre la declaración y el uso de métodos o funciones y presentarla en un resumen. • Identificar la diferencia entre parámetros y argumentos, su estructura e importancia, mediante una investigación bibliográfica, presentando ejemplos de programas en exposición grupal. • Realizar una práctica de ejercicios que involucren la implementación de métodos o funciones con pase de parámetros en la resolución de problemas del contexto, documentar y exponer.

8. Práctica(s)

<ul style="list-style-type: none"> • Elaborar diagrama de flujo, pseudocódigo y programa a problemas planteados que impliquen entrada y salida de datos, que impliquen declaración de variables y uso de expresiones aritméticas. • Solucionar problemas planteados en la clase a través de diagramas de flujo y pseudocódigo y codificarlos para comprobar su funcionamiento. • Resolver ejercicios que involucren diferentes expresiones algebraicas que ilustren la precedencia de los operadores aritméticos, utilizar un programa sugerido por el docente para comprobar los resultados obtenidos.
--

- En un lenguaje de programación, definir y manipular diferentes tipos de datos y comprobar sus características a través de un programa asignado por el docente.
- Elaborar ejercicios que impliquen estructuras secuenciales y selectivas entregando pseudocódigo, diagrama de flujo y programa.
- Elaborar ejercicios que impliquen las estructuras iterativas soportadas por un lenguaje de programación, entregando pseudocódigo, diagrama de flujo y programa.
- Elaborar ejercicios que impliquen organización de datos por medio de la implementación de arreglos entregando diagrama de flujo, pseudocódigo y programa.
- Elaborar ejercicios que impliquen organización de datos por medio de la implementación estructuras o registros entregando diagrama de flujo, pseudocódigo y programa.
- Elaborar ejercicios que impliquen la implementación de funciones entregando diagrama de flujo, pseudocódigo y programa.

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 y 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 (específicas y genéricas de la asignatura)

Para evaluar las actividades de aprendizaje se recomienda solicitar: mapas conceptuales o mentales, reporte de investigación, reportes de prácticas, códigos de programas, estudio de casos, exposiciones en clase, problemarios, portafolio de evidencias, entre otros.

Para verificar el nivel del logro de las competencias del estudiante se recomienda utilizar: listas de cotejo, listas de verificación, matrices de valoración, guías de observación, rúbricas, entre otros.

11. Fuentes de información

1. Albert, R., y Breedlove, T. (2009) C++: An active learning approach. USA: Jones and Bartlett Publishers
2. Brassard G.& et all. (Sin fecha). Fundamentos de Algoritmia. Pearson: Prentice Hall
3. Cairo Battistutti, O., (2005), *Metodología de la Programación, Algoritmos Diagrama de Flujo y Programas*, Ciudad, Estado, Provincia, País: Alfaomega. ISBN 970-15-1100-X
4. Cheng, H. H. (2010). *C for engineers and scientists: An interpretive approach*. USA: McGraw-Hill Higher Education
5. Deitel, P y Deitel, H. (2008) *Como Programar en C++*. México:Pearson Prentice Hall.
6. Deitel, J., y Deitel, M. (2012) C++: How to program. USA: Prentice Hall.
7. Deitel, H. (2008). *Java cómo programar*. (7ª ed.) Prentice Hall México, 2008. ISBN 9789702611905
8. García Molina, J.J., (2005), *Introducción a la programación un Enfoque Algorítmico*, Ciudad, Estado, Provincia, País: Paraninfo.
9. Joyanes, L. (2012) Fundamentos generales de programación. España;McGraw Hill.
10. Joyanes, L. (2008) *Fundamentos de programación: algoritmos, estructura de datos y objetos*. España: McGraw-Hill.
11. Joyanes, L. (2010) *Programación en C, C++, Java y UML*. México: McGraw-Hill. ISBN 978-970-10-6949-3.
12. Joyanes, L., Fernández, M. y Rodríguez L. (2003) *Fundamentos de Programación Libro de Problemas Algoritmos Estructura de Datos y Objetos*. México:Mc. Graw Hill.
13. Kamthane, N. (2011) *Programming in C*. India: Dorling Kindersley.
14. López, L. (2011) *Programación Estructurada y Orientada a Objetos* México:Alfaomega.
15. Marquez, G. (2012) *Introducción a la programación estructurada en C*. España:Pearson
16. McMilan, M. (2011) *Learning C++*. USA: InfiniteSkills.
17. Méndez, A. (2013) *Diseño de algoritmos y su programación en C*. México:Alfaomega.
18. Mothe, M. (2012) *C++ programming: a practical approach*. India: Dorling Kindersley.
19. Urrutia, G. (2012) *Curso de Programación en C para principiantes*. España:@Gorka Urrutia
20. Zavala, R y Llamas, R. (2013) *Fundamentos de programación para principiantes*. España:McGraw Hill.

Electrónicas:

21. Books.google (2003). Arreglos. Recuperado en Febrero de 2014. Disponible en: <http://books.google.com.mx/books?id=tR7k9ga5CjoC&pg=PA246&lpg=PA246&dq=arreglos+en+java&source=bl&ots=w84CRrTvHK&sig=4Lksg7a00XhhX6IpmRSTY7KdFUo&hl=es&sa=X&ei=LVv5UqjNO6bq2gXjrIEo&ved=0CCwQ6AEwATgK#v=onepage&q=arreglos%20en%20java&f=false>