



UNIVERSIDAD AUTÓNOMA DE SINALOA
SECRETARÍA ACADÉMICA UNIVERSITARIA
 Coordinación General de Evaluación, Innovación y Calidad Educativa
UNIDAD ACADÉMICA: FACULTAD DE INFORMÁTICA CULIACÁN

PROGRAMA DE ESTUDIOS

1. DATOS DE IDENTIFICACIÓN			
UNIDAD DE APRENDIZAJE	Análisis y Diseño Orientado a Objetos (Fundamentos de Programación)		
Clave:	4209		
Horas y créditos:	Teóricas: 20	Prácticas: 60	Estudio Independiente: 15
	Total de horas: 80		Créditos: 8
Tipo de unidad de aprendizaje:	Teórico:	Teórico-práctico:X	Práctico
Competencia (s) del perfil de egreso que desarrolla o a las que aporta.	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.		
Cursos consecuentes relacionados:	Algoritmia	Cursos subsecuentes relacionados:	Lenguajes de Programación
Responsables de elaborar y/o actualizar el programa:	MC. Gerardo Gálvez Gámez MC. Gerardo Beltrán Gutiérrez DR. Edy López Cervantes MC José Guadalupe Quintero Armenta		
Fecha de elaboración:	Junio 2011	Actualización: Febrero 2020	
2. PROPÓSITO			
Aplica algoritmos y lenguajes de programación para diseñar e implementar soluciones a problemáticas del entorno.			
3. SABERES			

Teóricos:	Conoce y comprende los conceptos básicos programación, así como de las herramientas y mecanismos fundamentales para la conceptualización, en un área de estudio.
Prácticos:	Diseñará e implementar soluciones de un requerimiento (problema) en algún lenguaje de programación.
Actitudinales :	Se muestra capaz de mantener una actitud positiva, de trabajar en equipo y de respetar las ideas contrarias.

4. CONTENIDOS

BLOQUE I: Introducción a la Programación (10 horas)	Aprendizajes Esperados
<ol style="list-style-type: none"> 1. Conceptos básicos 2. Características del lenguaje de programación 3. Estructura básica de un programa. 4. Elementos del lenguaje: tipos de datos, literales, constantes, variables, identificadores, parámetros, operadores y salida de datos. 5. Traducción de un programa: compilación, enlace, ejecución y errores. 6. Estructuras secuenciales. 	<ul style="list-style-type: none"> ● Elegir un lenguaje de programación e instalar y configurar las herramientas correspondientes para su uso, auxiliándose de algún tutorial, presentar el entorno funcionando en clase. ● 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 la asignatura de Algorimia, 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.

	<ul style="list-style-type: none"> ● Aplicar expresiones lógicas y algebraicas, para la solución de problemas, elaborar un reporte que incluya el código, los resultados de ejecución y exponerlo. ● Realizar práctica de ejercicios, donde se utilicen las estructuras secuencial utilizando la plataforma OmegaUP
<p>BLOQUE II: Estructuras de control de flujo (20 horas)</p>	
<ol style="list-style-type: none"> 1. Estructuras selectivas: <ol style="list-style-type: none"> a. simple (if - else) b. doble (if - else if - else) c. múltiple (switch) 2. Estructuras iterativas <ol style="list-style-type: none"> a. Ciclos basados en condición (while, do - while) b. Ciclos basados en contador (for) 	<ul style="list-style-type: none"> ● Conocer 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 selectiva e iterativa desarrollando algoritmo y programa utilizando la plataforma OmegaUP
<p>BLOQUE III: Organización de datos (20 horas)</p>	

<p>1. Arreglos unidimensionales</p> <ol style="list-style-type: none"> a. Conceptos básicos, b. Operaciones c. Aplicaciones. <p>2. Arreglos Multidimensionales</p> <ol style="list-style-type: none"> a. Conceptos básicos b. Operaciones c. aplicaciones <p>3. Estructuras o registros.</p>	<ul style="list-style-type: none"> ● Identificar las estructuras de datos: registros y arreglos unidimensionales, bidimensionales, multidimensionales, así como la aplicación de las mismas a través de ejemplos y exponerlos en clase. ● 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.
<p>BLOQUE IV: Modularidad (15 horas)</p>	
<p>1. Declaración y uso de módulos.</p> <p>2. Paso de parámetros y argumentos</p>	<ul style="list-style-type: none"> ● Identificar las ventajas, desventajas, estructura y usos de la programación modular. ● Identificar la diferencia entre parámetros y argumentos, su estructura e importancia, mediante una investigación bibliográfica, presentando ejemplos de programas en exposición grupal. ● Realizar ejercicios en la plataforma OmegaUp que involucren la implementación de métodos o funciones con pase de parámetros en la resolución de problemas, documentar y exponer
<p>BLOQUE V: Tratamiento de archivos (15 horas)</p>	
<p>1. Clasificación de los archivos</p> <p>2. Modos de apertura</p> <p>3. Operaciones de acceso secuencial</p> <ol style="list-style-type: none"> a. Archivos de texto 	<p>Conocer y aplicar herramientas de acceso a archivos en sus distintas modalidades</p>

b. Archivos binarios 4. Operaciones de acceso aleatorio	
--	--

5. ACTIVIDADES PARA DESARROLLAR LAS COMPETENCIAS

Actividades del maestro.

- Actividades de inicio: técnica expositiva, conferencia, la pregunta.
- Actividad de desarrollo: panel, simposio, investigación bibliográfica, estudio supervisado, diálogo, mesa redonda con moderador y/o relator,

Actividad de evaluación: informe de investigación documental, ensayo, mapa conceptual, cuadro sinóptico, cuadro comparativo, portafolio de evidencias, rúbrica.

Actividades del estudiante.

- Actividades de inicio: Diario, fichas de trabajo, memoria, lluvia de ideas.
- Actividades de desarrollo: concordar y discordar,

Actividades finales: Informe de investigación documental o de campo, ensayo, mapa conceptual, cuadro sinóptico, cuadro comparativo, portafolio de evidencias, rúbrica.

6. EVALUACIÓN DE LAS COMPETENCIAS

6.1. Evidencias de aprendizaje	6.2. Criterios de desempeño	6.3. Calificación y acreditación
<ul style="list-style-type: none"> • Escala de rango • Rúbrica • La pregunta • Debate • Ensayo • Mapa conceptual • Texto paralelo 	<ul style="list-style-type: none"> • Obtenga el 80% de asistencia. • Participa en clases. • Realiza prácticas en laboratorio. • Realiza proyectos • Aprueba exámenes parciales (calificación mínima de 6.0) • Practique una lectura activa, y comprenda los textos a revisar. • Investigue para abundar en fuentes. • Participar en las discusiones y debates. 	<ul style="list-style-type: none"> • 10% de asistencia • 60% Calificación aprobatoria en exámenes parciales • 30% Revisión de productos requeridos: Proyectos, Estudio de casos, técnica expositiva.

7. FUENTES DE INFORMACIÓN

Bibliografía:

- Kernighan, Brian W.; Ritchie, Dennis M. (1991). El lenguaje de programación C (2ª edición). Prentice Hall Hispanoamericana. ISBN 968-880-205-0
- Bjarne Stroustrup, The C++ Programming Language, Addison-Wesley Pub Co; Tercera edición (15 de febrero de 2000); ISBN 0-201-70073-5
- Herbert Schildt, C. Manual de Referencia, McGraw-Hill Interamericana de México, Cuarta edición, 2001
- Kruse, Robert Leroy; Tondon, Clovis L.; Leung, Bruce P: "Data structures and program design in C", Prentice-Hall, 1997.
- Cormen, Thomas H.; Leiserson Charles E.; Rivest Ronald L; Stein Clifford: "Introduction to Algorithms", The MIT Press, 2009.
- Sedgewick, Robert: "Algorithms in C++", Addison-Wesley, 1992.
- Tenenbaum, Aaron M.; Langsam, Yedidyah; Augenstein, Moshe J.: "Estructuras de datos en C", Prentice-Hall Hispanoamericana, 1993.

8. PERFIL DEL PROFESOR:

Licenciatura en informática, Ingeniería en Computación.