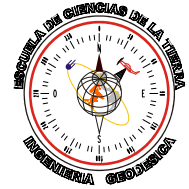




Universidad Autónoma de Sinaloa

Escuela de Ciencias de la Tierra

LICENCIATURA EN INGENIERÍA GEOMÁTICA



PROGRAMA DE ESTUDIOS

1. DATOS DE IDENTIFICACIÓN			
UNIDAD DE APRENDIZAJE	PROGRAMACIÓN E INGENIERÍA DE SOFTWARE I		
Clave:	(pendiente)		
Semestre:	Vsemestre		
Eje Curricular:	<input type="checkbox"/> Tronco Común <input checked="" type="checkbox"/> Profesionalizante		
Área:	<input type="checkbox"/> Física-Matemática <input type="checkbox"/> Cs. Sociales y Humanidades <input type="checkbox"/> Idiomas <input type="checkbox"/> Básico Profesional <input checked="" type="checkbox"/> Profesional		
Horas y créditos:	Teóricas: 40	Prácticas: 40	Estudio Independiente: 16
	Total de horas: 96		Créditos: 6
Tipo de curso:	Teórico	Teórico-práctico (X)	Práctico
Competencias del perfil de egreso a la que aporta	Capacidad de aplicación de la formación científica, organización y coordinación de proyectos, que le permitan hacer uso óptimo de las complejas tecnologías de computo modernas, así como desarrollar nuevos modelos, procedimientos y sistemas para la obtención, manejo y presentación de datos geográficos requeridos para la solución de problemas en el aprovechamiento del suelo, de los recursos naturales y la infraestructura del país.		
Unidades de aprendizaje relacionadas	Programación I, Programación e Ingeniería de software II, Programación e Ingeniería de software III, Tratamiento de la Información.		
Responsables de elaborar y/o actualizar el programa:	Dr. Miguel Contreras Montoya, M.C. Gerardo Gálvez Gámez, Lic. Rogelio Prieto Alvarado, M.C. José Guadalupe Quintero Armenta, MC. Edy López Cervantes, MC. Rodrigo García Carlos, Dr. Inés Fernando Vega López, LI. Diego Alonso Gastélum Chavira, MC. Gerardo Beltrán Gutiérrez, Lic. Cosme Adrián López Inda, M.C. Rodrigo García Carlos, Dr. Juan Martín Aguilar Villegas.		
Fecha de:	Elaboración: Febrero del 2012		Actualización: Febrero del 2012
2. PROPÓSITO			
Al finalizar el curso, el alumno será capaz de analizar, diseñar, probar y construir programas aplicando los nuevos paradigmas de ingeniería de software.			
3. SABERES			
Teóricos:	<ul style="list-style-type: none"> • Conocer de manera general la historia y el desarrollo de la computadora, así como de las herramientas existentes para el análisis y diseño de programas. • Obtener los conocimientos básicos sobre algoritmos, así como, los tipos de datos existentes y expresiones algorítmicas. • Comprender las diferentes estructuras algorítmicas. • Obtener las herramientas básicas de la metodología de programación en un lenguaje de programación de alto nivel. 		

	<ul style="list-style-type: none"> • Comprender el concepto de programación modular empleado en un lenguaje de programación de alto nivel. • Entender los tipos de datos estructurados.
Prácticos:	<ul style="list-style-type: none"> • Aplicar los conocimientos básicos sobre algoritmos, así como, sobre los tipos de datos existentes y su aplicación en las diferentes sentencias y expresiones algorítmicas. • Construir programas en pseudocódigo aplicando las estructuras algorítmicas. • Aplicar las herramientas básicas de la metodología de programación en un lenguaje de programación de alto nivel. • Crear programas en pseudocódigo aplicando el concepto de programación modular para luego codificarlos en un lenguaje de programación de alto nivel. • Desarrollar programas aplicando los tipos de datos estructurados.
Actitudinales:	<ul style="list-style-type: none"> • Valorar el papel de la ciencia y la tecnología en la programación dirigida a objetos. • Disposición al trabajo colectivo. • Cultivar la disciplina de la lectura científica. • Desarrollar la ética profesional. • Reflexividad ante las diferentes propuestas técnicas y metodológicas. • Atención a la actualización profesional

4. CONTENIDO TEMÁTICO

1. INTRODUCCIÓN A LA PROGRAMACIÓN E INGENIERÍA DE SOFTWARE.

- 1.1. Breve historia de la computadora
- 1.2. Conceptos básicos
- 1.3. Concepto de lenguaje
- 1.4. Clasificación de los lenguajes
- 1.5. Compiladores e intérpretes
- 1.6. Paradigmas de la Programación
- 1.7. Principales herramientas para el análisis y diseño.

2. BASES PARA LA CREACIÓN DE UN PROGRAMA.

- 2.1. Solución de problemas aplicando la computadora.
- 2.2. Algoritmos.
- 2.3. Tipos de datos.
- 2.4. Constantes y variables.
- 2.5. Sentencias y expresiones.
- 2.6. Pseudocódigo.

3. ESTRUCTURAS ALGORÍTMICAS.

- 3.1. Estructuras Secuenciales.
- 3.2. Estructuras Selectivas.
- 3.3. Estructuras Repetitivas.

4. CODIFICACIÓN DE PSEUDOCODIGOS

- 4.1. Tipos de datos.
- 4.2. Constantes y Variables.
- 4.3. Conversión entre tipos de datos.
- 4.4. Expresiones y Operadores.
- 4.5. Operaciones de entrada/salida.
- 4.6. Sentencias de control.
- 4.7. Estructuras secuenciales.
- 4.8. Estructuras selectivas.
- 4.9. Estructuras repetitivas.

5. PROGRAMACIÓN MODULAR.

- 5.1. Concepto de módulo.
- 5.2. Técnicas de programación modular.
- 5.3. Funciones.
- 5.4. Procedimientos.
- 5.5. Codificación en lenguaje de programación

6. TIPOS DE DATOS ESTRUCTURADOS

- 6.1. Conceptos básicos.
- 6.2. Tipos Estructurados.
 - 6.2.1. Arreglos.
 - 6.2.2. Cadenas de Caracteres.
 - 6.2.3. Estructuras y Registros.
 - 6.2.4. Archivos.
 - 6.2.5. Codificación en Lenguaje de programación

5. ACCIONES ESTRATÉGICAS PARA EL APRENDIZAJE

- Exposición de introducción al tema, así como de los antecedentes y vigencia del mismo en cada unidad.
- Lecturas de investigación para complementar la comprensión del tema.
- Análisis y debate grupal sobre el tema.
- Transferencia de contenidos temáticos mediante los medios electrónicos.
- Planteamiento y solución de problemas concretos.
- Trabajos de investigación y redacción de resúmenes.
- Solución de problemas extra clase.
- Trabajo colectivo de exposición.

6. EVALUACIÓN DEL APRENDIZAJE

6.1. Evidencias de aprendizaje	6.2. Criterios de desempeño	6.3. Calificación y acreditación
1. Resolución de ejercicios en clase en forma individual y grupal. 2. Resolución de ejercicios de tarea.	<ul style="list-style-type: none">• Comprensión y profundización de conceptos teóricos.• Capacidad de trabajo colectivo intelectual y	<ul style="list-style-type: none">• Asistencia y participación 20%

<p>3. Debate en torno a las dudas de los estudiantes.</p> <p>4. Resolución de problemas utilizando herramientas de cómputo.</p> <p>5. Examen.</p>	<p>práctico.</p> <ul style="list-style-type: none"> • Capacidad de exposición y dominio temático. • Capacidad de análisis, de redacción y síntesis de la investigación bibliográfica. • Planeación y desarrollo metodológico en la solución de problemas. • Capacidad de responder de manera precisa, clara y completa los reactivos de exámenes en forma oral y escrita. 	<ul style="list-style-type: none"> • Tareas 30% • Examen 50%
---------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------

7. FUENTES DE INFORMACIÓN

Fuentes de Información Básica:

- LÓPEZ ROMÁN, LEOBARDO (2006) *Metodología de la Programación Orientada a Objetos*. Edit Alfaomega, México
- D.F. Primera Edición.
- CAIRO BATTISTUTTI, OSVALDO. (2006) *Metodología de la Programación, Algoritmos, Diagramas de Flujo y Programas*.
- Edit Alfaomega, México D.F Tercera edición.
- LÓPEZ ROMÁN, LEOBARDO. (2003) *Programación Estructurada, un enfoque algorítmico*. Edit Alfaomega, México D.F.
- Segunda Edición.
- SCHILDT HERBERT (2003), *C#. Manual de referencia*. Mc Graw Hill, Primera Edición.
- CEBALLOS, Fco. Javier (2008), *C# Lenguajes y Aplicaciones*, Edit Alfaomega, México D.F , Segunda Edición.

Fuentes de Información Complementaria

- JOYANES AGUILAR, LUIS. (1996) *Fundamentos de Programación, Algoritmos y Estructura de datos*. Edit. McGraw-Hill. Madrid, España. Segunda edición.
- JOYANES AGUILAR, LUIS. (1997) *Metodología de la Programación, Diagramas de flujo algoritmos y programación estructurada*. Edit. McGraw-Hill. México, D.F. Primera edición.
- AYALA SAN MARTIN, GERARDO. (1987) *Computación I, Introducción a la Computación*. Edit. Porrúa,
- Universidad Nacional Autónoma de México. México, D.F. Primera edición.
- LEVINE GUTIERREZ, GUILLERMO. (1994) *Introducción a la Computación y a la Programación Estructurada*. Edit.
- McGraw-Hill. Primera edición.
- SANDERS, DONALD H. (1994) *Informática, Presente y Futuro*. Edit. McGraw-Hill. Tercera Edición.
- PRESSMAN, ROGER S. (1998) *Ingeniería del Software, un enfoque práctico*. Edit. Mc
- Graw-Hill. Madrid, España. Cuarta edición.

8. PERFIL DEL PROFESOR

El profesor debe de contar con posgrado en el área de las ciencias de la información con orientación en

Programación o en alguna disciplina de la informática. Debe de contar con experiencia docente y en trabajos de investigación o aplicación de los métodos de programación para la generación de información geográfica.