



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	Estructura de Datos		
Clave:	4419		
Horas y créditos:	Teóricas: 40	Prácticas: 40	Estudio Independiente:
	Total de horas: 80		Créditos: 5
Tipo de unidad de aprendizaje:	Teórico:	Teórico-práctico: X	Práctico
Competencia (s) del perfil de egreso que desarrolla o a las que aporta.	Conoce y aprende a aplicar las estructuras de datos como técnicas de diseño necesarias para formular y expresar algoritmos computacionales, estructurando en forma eficiente la representación elegida para la información, logrando así, la construcción de programas como soluciones a problemas del mundo real en forma correcta y metodológica, empleando un lenguaje de programación orientado a objetos como C# y java.		
Cursos consecuentes relacionados:	Algoritmia, Análisis y Diseño Orientado a Objetos, Lenguajes de Programación	Cursos subsecuentes relacionados:	Inteligencia Artificial, Programación distribuida y paralela, Graficación
Responsables de elaborar y/o actualizar el programa:	MC. Gerardo Gálvez Gámez MC. Gerardo Beltran Gutierrez		
Fecha de elaboración:	Junio 2011	Actualización: Junio 2018	
2. PROPÓSITO			
Conocerá las diferentes estructuras de datos, tales como arreglos, colas, pilas, árboles y grafos, programándolas en un lenguaje de alto nivel orientado a objetos como C# y Java, con la finalidad de dar solución a problemas computacionales que las requieran.			
3. SABERES			
Teóricos:	Identifica y comprende las diferentes estructuras de datos a utilizar en programación.		
Prácticos:	Analiza, construye e implementa estructuras de datos para dar solución a problemas computacionales en un lenguaje de programación orientado a objetos como c# y Java.		
Actitudinales:	Se muestra capaz de tener actitud positiva, trabaja en equipo, participa en clase y mantiene una relación de respeto y apoyo con sus compañeros.		

4. CONTENIDOS	
BLOQUE I: ESTRUCTURAS FUNDAMENTALES DE DATOS.	Aprendizajes Esperados
<p>1.1. Tipos de datos.</p> <p>1.1.1. Tipos de datos simples.</p> <p>1.1.2. Tipos de datos abstractos (TDA).</p> <p>1.1.3. Representación de datos.</p> <p>2.1. Estructuras de datos.</p> <p>2.1.1. Definición.</p> <p>2.1.2. Clasificación.</p> <p>2.1.3. Lineales y no lineales.</p> <p>2.1.4. Dinámicas y estáticas.</p> <p>3.1. Implementación de TDA en Lenguaje de programación.</p>	<ul style="list-style-type: none"> • Identifica y comprende los diferentes tipos de datos. • Aprende sobre la medición de la eficiencia de una estructura de datos y Diseño de un TDA. • Comprende la definición y clasificación de las diferentes estructuras de datos, enlazadas que están en la base de muchas aplicaciones.
BLOQUE II: ESTRUCTURA DE DATOS ARREGLOS	
<p>2.1. Arreglos</p> <p>2.1.1. Definición.</p> <p>2.1.2. Unidimensionales.</p> <p>2.1.3. Bidimensionales.</p> <p>2.1.4. Multidimensionales.</p> <p>2.1.5. Operaciones.</p> <p>2.1.6. Clases para la implementación de arreglos.</p> <p>2.1.7. Resolución de problemas con arreglos.</p> <p>2. 1. Recursividad.</p> <p>2.1.1. Concepto de Recursividad.</p> <p>2.1.2. Utilización de Recursividad.</p> <p>2.1.3. Resolución de problemas con Recursividad.</p> <p>2.4. Métodos de ordenamiento.</p> <p>2.1.4. intercambio directo</p> <p>2.1.5. inserción directa</p> <p>2.1.6. inserción binaria</p> <p>2.1.7. selección</p> <p>2.1.8. Shell</p> <p>2.1.9. quicksort</p> <p>2.5. Métodos de Búsqueda.</p> <p>2.5.1. Secuencial o lineal</p> <p>2.5.2. Binaria</p> <p>2.6. Ordenamiento y Búsqueda en Archivos</p>	<ul style="list-style-type: none"> • Comprende e identifica las estructuras de datos arreglos. • Razona y domina los conceptos de arreglos. • Implementa clases para resolver problemas con arreglos. • Razona, comprende y compara los diferentes algoritmos de ordenación y búsqueda. • Aplicará el análisis de algoritmos a los diferentes métodos de búsqueda y ordenamiento para la selección del método todo óptimo para el procesamiento de un conjunto de datos

<p>BLOQUE III: ESTRUCTURAS DE DATOS LINEALES</p>	
<p>2.3. Pilas.</p> <p>2.3.1. Definición.</p> <p>2.3.2. Características.</p> <p>2.3.3. Representación.</p> <p>2.3.4. Utilización</p> <p>2.3.5. Operaciones.</p> <p>2.3.6. Tipos de Errores.</p> <p>2.3.7. Estados.</p> <p>2.3.8. Clases para la implementación, utilizando arreglo.</p> <p>2.3.9. Resolución de problemas con Pilas.</p> <p>2.4. Colas.</p> <p>2.4.1. Definición.</p> <p>2.4.2. Características.</p> <p>2.4.3. Representación.</p> <p>2.4.4. Utilización.</p> <p>2.4.5. Tipos</p> <p>2.4.5.1. Simples.</p> <p>2.4.5.2. Circulares.</p> <p>2.4.5.3. Colas dobles.</p> <p>2.4.6. Operaciones.</p> <p>2.4.7. Estados.</p> <p>2.4.8. Clases para la implementación de colas.</p> <p>2.4.9. Resolución de problemas con colas</p> <p>2.5. Listas enlazadas.</p> <p>2.5.1. Descripción general.</p> <p>2.5.2. Representación gráfica y estructura del Nodo.</p> <p>2.5.3. Tipos.</p> <p>2.5.3.1. Simples.</p> <p>2.5.3.2. Dobles.</p> <p>2.5.3.3. Circulares.</p> <p>2.5.3.4. Multilistas.</p> <p>2.5.4. Operaciones.</p> <p>2.5.5. Clases para la implementación de listas.</p> <p>2.5.6. Resolución de problemas con listas</p>	<ul style="list-style-type: none"> • Aplica las principales estructuras de datos lineales, así como su representación, manejo y aplicación, en un Lenguaje de programación, para dar soluciones a problemas computacionales.
<p>BLOQUE IV: Estructuras de Datos no Lineales</p>	

<p>4.1. Árboles.</p> <p>4.1.1. Definición.</p> <p>4.1.2. Representación en memoria de árboles.</p> <p>4.1.3. Árboles generales.</p> <p>4.1.4. Árboles binarios y operaciones.</p> <p>4.1.5. Recorridos en un árbol binario.</p> <p>4.1.5.1. Preorden.</p> <p>4.1.5.2. Inorden.</p> <p>4.1.5.3. Posorden.</p> <p>4.1.6. Balanceo de árboles binarios.</p> <p>4.1.7. Clases para la implementación de árboles.</p> <p>4.1.8. Resolución de problemas con árboles</p> <p>4.2. Grafos</p> <p>4.2.1. Definición.</p> <p>4.2.2. Representación de grafos.</p> <p>4.2.3. Grafos dirigidos y no dirigidos.</p> <p>4.2.4. Aplicaciones.</p> <p>4.2.5. Clases para la implementación de grafos.</p> <p>4.2.6. Resolución de problemas con grafos.</p>	<ul style="list-style-type: none"> • Aplica las principales estructuras de datos no lineales, así como su representación, manejo y aplicación, en un Lenguaje de programación para dar soluciones a problemas computacionales.
---	---

5. ACTIVIDADES PARA DESARROLLAR LAS COMPETENCIAS

Actividades del maestro.

- Actividades de inicio: la pregunta.
- Actividad de desarrollo: técnica expositiva, estudio de casos, diálogo
- Actividad de evaluación: informe de investigación, ensayo, mapa conceptual, cuadro sinóptico, cuadro comparativo, portafolio de evidencias, rúbrica.

Actividades del estudiante.

- Actividades de inicio: Fichas de trabajo, memoria, lluvia de ideas.
- Actividades de desarrollo: investigación de tópicos, documentación y debate de resultados
- 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

1 6.1. Evidencias de aprendizaje	2 6.2. Criterios de desempeño	3 6.3. Calificación y acreditación
<p>Actividades de inicio:</p> <ul style="list-style-type: none"> • La pregunta <p>Actividades de desarrollo:</p> <ul style="list-style-type: none"> • Técnica expositiva • Ensayo <p>Actividades de evaluación:</p> <ul style="list-style-type: none"> • Proyecto • Estudio de casos 	<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

4 Bibliografía:

- Cairo, Osvaldo. Estructura de Datos. Tercera edición. Ed. McGraw Hill; 2006.
- Joyanes Aguilar, Luis. Estructura de Datos en Java. Primera edición. Ed. McGraw Hill. 2007.
- Guardati Buemo, Silvia. Estructura de Datos orientada a objetos: Algoritmos con C++. Primera edición. Ed. Pearson. 2007.
- Lewis, John. Estructura de Datos con JAVA: Diseño de estructuras y algoritmos. Primera edición. Ed. Pearson. 2007.
- Hernandez, Roberto; Lázaro, Juan Carlos; Dormido Raquel; Ros, Salvador: "Estructuras de Datos y Algoritmos". Prentice Hall, 2001.
- Cormen, Thomas H.: "Introduction to Algorithms". MIT Press / Mc Graw-Hill, 2001. 2ª Edición.
- R. HERNÁNDEZ, J. C. LÁZARO, R. DORMIDO yS. ROS: Estructuras de Datos y Algoritmos, Prentice Hall, Madrid, 2000.
- Cairó., Estructuras de datos, 2ª Edición., Ed. Mc Graw Hill, 2001.
- Mark Allen Weiss Mark., Estructuras de datos en Java., Ed. Pearson.2002.
- Heileman., Estructuras de datos, algoritmos y programación orientada a objetos., Ed. Mc Graw Hill, 2001.
- Mark Allen Weiss Mark., Estructuras de datos en Java., Ed. Pearson.2002.
- Schildt., C ++, Guía de autoenseñanza., Ed. Mc Graw Hill, 2001.
- Schildt, C#. Manual de referencia., Ed. Mc Graw Hill, 2003.

8. PERFIL DEL PROFESOR:

Licenciatura en informática, Ingeniería en Computación.