

1.- DATOS DE LA ASIGNATURA

Nombre de la asignatura: Estructura de datos
Carrera: Ingeniería en Sistemas Computacionales
Clave de la asignatura: SCC - 0408
Horas teoría-horas práctica-créditos 4-2-10

2.- HISTORIA DEL PROGRAMA

Lugar y fecha de elaboración o revisión	Participantes	Observaciones (cambios y justificación)
Instituto Tecnológico de Toluca 18 – 22 agosto 2003.	Representantes de la academia de sistemas y computación de los Institutos Tecnológicos.	Reunión nacional de evaluación curricular de la carrera de Ingeniería en Sistemas Computacionales.
Institutos Tecnológicos de: Lázaro Cárdenas, Culiacán, Acapulco. 23 agosto al 7 noviembre del 2003	Academia de sistemas y computación.	Análisis y enriquecimiento de las propuestas de los programas diseñados en la reunión nacional de evaluación
Instituto Tecnológico de León 1 – 5 marzo 2004	Comité de consolidación de la carrera de Ingeniería en Sistemas Computacionales.	Definición de los programas de estudio de la carrera de Ingeniería en Sistemas Computacionales.

3.- UBICACIÓN DE LA ASIGNATURA

a). Relación con otras asignaturas del plan de estudio

Anteriores		Posteriores	
Asignaturas	Temas	Asignaturas	Temas
Fundamentos de programación.			
Programación Orientada a Objetos.			
Matemáticas para la computación.			

b). Aportación de la asignatura al perfil del egresado

Selecciona e implementa las estructuras de datos necesarias para el desarrollo de aplicaciones orientadas a objetos que proporcionen la mejor solución a problemas científicos, tecnológicos y de propósito general.

4.- OBJETIVO(S) GENERAL(ES) DEL CURSO

El estudiante seleccionará estructuras de datos, algoritmos de ordenamiento y búsqueda para optimizar el rendimiento de una aplicación.

5.- TEMARIO

Unidad	Temas	Subtemas
1	Análisis de algoritmos.	1.1 Concepto de Complejidad de algoritmos. 1.2 Aritmética de la notación O. 1.3 Complejidad. 1.3.1 Tiempo de ejecución de un algoritmo. 1.3.2 Complejidad en espacio. 1.4 Selección de un algoritmo.
2	Manejo de memoria.	2.1 Manejo de memoria estática. 2.2 Manejo de memoria dinámica.
3	Estructuras lineales estática y dinámicas.	3.1 Pilas. 3.2 Colas. 3.3 Listas enlazadas. 3.3.1 Simples. 3.3.2 Dobles.
4	Recursividad.	4.1 Definición. 4.2 Procedimientos recursivos. 4.3 Mecánica de recursión. 4.4 Transformación de algoritmos recursivos a iterativos. 4.5 Recursividad en el diseño. 4.6 Complejidad de los algoritmos recursivos.
5	Estructuras no lineales estáticas y dinámicas.	5.1 Concepto de árbol. 5.1.1 Clasificación de árboles. 5.2 Operaciones Básicas sobre árboles binarios. 5.2.1 Creación. 5.2.2 Inserción. 5.2.3 Eliminación. 5.2.4 Recorridos sistemáticos. 5.2.5 Balanceo.

5.- TEMARIO (Continuación)

6	Ordenación interna.	6.1 Algoritmos de Ordenamiento por Intercambio. 6.1.1 Burbuja. 6.1.2 Quicksort. 6.1.3 ShellSort. 6.2 Algoritmos de ordenamiento por Distribución. 6.2.1 Radix.
7	Ordenación externa.	7.1 Algoritmos de ordenación externa. 7.1.1 Intercalación directa. 7.1.2 Mezcla natural.
8	Métodos de búsqueda.	8.1 Algoritmos de ordenación externa. 8.1.1 Secuencial. 8.1.2 Binaria. 8.1.3 Hash. 8.2 Búsqueda externa. 8.2.1 Secuencial. 8.2.2 Binaria. 8.2.3 Hash.

6.- APRENDIZAJES REQUERIDOS

- Analizar un problema y realizar el planteamiento de la solución mediante el uso de las técnicas básicas de análisis y diseño orientado a objetos.
- Desarrollar algoritmos que representen el comportamiento de los objetos involucrados en la solución del problema.
- Implementar el modelo obtenido para la solución de un problema, mediante una herramienta de desarrollo de software.

7.- SUGERENCIAS DIDÁCTICAS

- Exposición de los temas con sesiones de preguntas y respuestas.
- Lecturas recomendadas de libros y direcciones de Internet.
- Uso de correo electrónico para revisión de tareas y ejercicios.
- Uso de un portal de Internet para apoyo didáctico de la materia, el cual cuente por lo menos con un foro, preguntas frecuentes, material de apoyo y correo electrónico.
- Desarrollar el trabajo en equipo.
- Elaborar con los estudiantes una guía de ejercicios para actividades extra clase.
- Uso del laboratorio de cómputo para la elaboración de ejercicios resueltos y propuestos.
- Exposición de temas con apoyo de material didáctico (cañón, proyector de acetatos, rotafolios, pizarrón, entre otros).

8.- SUGERENCIAS DE EVALUACIÓN

- Ponderar tareas.
- Participación en actividades individuales y de equipo.
- Participación y desempeño en el aula y el laboratorio.
- Seguimiento al desempeño en el desarrollo del programa (dominio de los conceptos, capacidad de la aplicación de los conocimientos en problemas reales, transferencia del conocimiento).
- Desarrollo de un proyecto final que integre todas las unidades de aprendizaje.
- Participación en dinámicas grupales.
- Actividades de auto evaluación.
- Exámenes teórico práctico.
- Cumplimiento de los objetivos y desempeño en las prácticas

9.- UNIDADES DE APRENDIZAJE

UNIDAD 1.- Análisis de algoritmos.

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
El estudiante comprenderá el concepto de complejidad de los algoritmos y su aplicación en la selección de los mismos.	1.1 Evaluar la complejidad de algoritmos propuestos. 1.2 Determinar el mejor algoritmo en cuanto tiempo de ejecución. 1.3 Determinar la complejidad de un algoritmo considerando complejidad en el espacio.	1,3,4

UNIDAD 2.- Manejo de memoria.

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
Comprenderá los mecanismos para el manejo de memoria.	2.1 Elaborar un mapa conceptual que represente el almacenamiento de la información estática en la computadora. 2.2 Elaborar un mapa conceptual que represente el almacenamiento de la información dinámica en la computadora. 2.3 Realizar un esquema que represente el mecanismo de liberación de memoria dinámica. 2.4 Realizar un ejercicio que muestre el funcionamiento y diferenciación de cada uno de los mecanismos de manejo de memoria. 2.5 Analizar y discutir en equipos las ventajas y desventajas de cada uno de los mecanismos de manejo de memoria.	1,2

UNIDAD 3.- Estructuras lineales estática y dinámicas.

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
Conocerá las características de las diferentes estructuras de datos lineales y su implementación en un lenguaje de programación	3.1 Utilizar el concepto de clase genérica en la implementación de las estructuras de datos lineales. 3.2 Resolver problemas que hagan uso de las estructuras de datos lineales e implementar las aplicaciones utilizando los diferentes mecanismos de manejo de memoria 3.3 Analizar y discutir en equipos las ventajas y desventajas de la implementación estática y dinámica de las diferentes estructuras de datos lineales	1,2,4,5

UNIDAD 4.- Recursividad

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
Comprenderá el concepto de recursividad y su implementación en un lenguaje de programación.	4.1 Realizar un mapa conceptual que represente la recursividad. 4.2 Analizar en el grupo las definiciones y procedimientos recursivos. 4.3 Esquematizar el uso de una pila en el proceso de recursividad. 4.4 Transformar un algoritmo recursivo a iterativo utilizando pilas. 4.5 Diseñar algoritmos recursivos.	1,2,4,

UNIDAD 5.- Estructuras no lineales estáticas y dinámicas.

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
Conocerá las características de diferentes estructuras de datos no lineales y su implementación en un lenguaje de programación.	5.1 Realizar un esquema que represente la clasificación de los árboles y sus características. 5.2 Utilizar el concepto de clase genérica en la implementación de las estructuras de datos no lineales.	1,2,4,5

	<p>5.3 Resolver por equipos problemas que hagan uso de las estructuras de datos no lineales e implementar las aplicaciones utilizando los diferentes mecanismos de manejo de memoria.</p> <p>5.4 Analizar y discutir en clase las ventajas y desventajas de la implementación estática y dinámica de las diferentes estructuras de datos no lineales.</p>	
--	---	--

UNIDAD 6.- Ordenación interna.

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
Organizará un conjunto de datos y elegirá el método de ordenación interna más conveniente según el problema.	<p>6.1 Realizar un ejercicio que muestre las ventajas y desventajas de los diferentes algoritmos de ordenamiento.</p> <p>6.2 Discutir y determinar cual es el mejor método de ordenación interna, para un problema dado.</p> <p>6.3 Proponer e implementar su propio algoritmo de ordenación.</p> <p>6.4 Discutir en grupo las características de los algoritmos propuestos por los alumnos.</p> <p>6.5 Determinar la complejidad de los algoritmos vistos en clase y los propuestos por los alumnos.</p> <p>6.6 Buscar y analizar algoritmos de ordenamiento interno y realizar un análisis comparativo con respecto a los algoritmos propuestos en el temario.</p>	1 ,2 ,3

UNIDAD 7.- Ordenación externa.

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
Organizará un conjunto de datos almacenados externamente.	7.1 Realizar un ejercicio que muestre las ventajas y desventajas de los diferentes algoritmos de ordenamiento. 7.2 Determinar la complejidad de los algoritmos vistos en clase. 7.3 Proponer e implementar su propio algoritmo de ordenación y determinar su complejidad. 7.4 Discutir en grupo las características de los algoritmos propuestos por los alumnos. 7.5 Buscar y analizar algoritmos de ordenamiento externo y realizar un análisis comparativo con respecto a los algoritmos propuestos en el temario.	1,2,3,8

UNIDAD 8.- Métodos de búsqueda.

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
Comprenderá las características de los diferentes métodos de búsqueda para recuperar datos almacenados y su implementación en un lenguaje de programación.	8.1 Realizar un ejercicio que muestre las ventajas y desventajas de los diferentes métodos de búsqueda para datos almacenados interna y externamente. 8.2 Discutir las ventajas y desventajas de los diferentes mecanismos de búsqueda. 8.3 Analizar y discutir la complejidad de los diferentes algoritmos de búsqueda.	1,2,3,5.

10. FUENTES DE INFORMACIÓN

1. Estructuras de datos con C y C++ (segunda edición).
Yedidyah Lang Sam, Moshe J. Augenstein, Aaron M. Tenenbaum.
Ed. Prentice Hall.
2. Programación en C++ (segunda edición).
Deitel y Deitel.
Ed. Prentice Hall.
3. Estructura de datos y diseño de programas.
Robert L. Kruse.
Ed. Prentice Hall.
4. Estructuras de datos (libro de problemas).
Luis Joyanes Aguilar, Ignacio Z. Martínez, Matilde Fernandez Azuela,
Lucas S. Garcia.
Ed. Mac Graw Hill.
5. Estructura de datos en Java.
Mark Allen Weiss.
Ed. Addison Wesley.
6. Introducción a la Programación Orientada a Objetos con Java.
C. Thomas Wu.
Ed. Pearson Educación.
7. Java.
Paul S. Wong.
Ed. International Thomsom Editores.
8. Programación con Java.
Decker, Hirshfield.
Ed. International Thomsom Editores.
9. Java a través de ejemplos.
Jesús Bobadilla.
Ed. Alfa Omega – RAMA.
10. Aprenda y practique Java.
Kris Jamsa.
Ed. Oxford.

11. PRÁCTICAS

Unidad Práctica

1	1	Con base a aplicaciones (propuestas por el profesor y por el estudiante), determinar la complejidad de los algoritmos utilizados.
2	1	Desarrollar aplicaciones que hagan uso de memoria estática y dinámica.
3	1	Implementar clases genéricas para las estructuras de datos lineales que puedan ser utilizadas en la solución de problemas que requieran de estas estructuras de datos, considerando los diferentes mecanismos de manejo de memoria.
	2	Abordar problemas que hagan uso de las estructuras de datos lineales e implementar su solución
4	1	Dar solución a problemas que requieran del uso de algoritmos recursivos y determinar su complejidad.
	2	A partir de un algoritmo recursivo, generar un algoritmo iterativo utilizando pilas.
5	1	Implementar clases genéricas para las estructuras de datos no lineales que puedan ser utilizadas en la solución de problemas que requieran de estas estructuras de datos, considerando los diferentes mecanismos de manejo de memoria.
	2	Abordar problemas que hagan uso de las estructuras de datos no lineales e implementar su solución.
6	1	Implementar aplicaciones que requieran de algoritmos de ordenamiento interno.
	2	Realizar un análisis de la complejidad de los algoritmos de ordenamiento interno.
7	1	Implementar aplicaciones que requieran de algoritmos de ordenamiento externo.
	2	Realizar un análisis de la complejidad de los algoritmos de ordenamiento externo
8	1	Desarrollar un proyecto final que integre ordenamiento y búsqueda en un conjunto de datos almacenados.

