

1.- DATOS DE LA ASIGNATURA

Nombre de la asignatura:	Estructuras de Datos
Carrera:	Ingeniería Informática e Ingeniería en Sistemas Computacionales.
Clave de la asignatura:	AED-1026
SATCA ¹	2 - 3 - 5

2.- PRESENTACIÓN

Caracterización de la asignatura.

La importancia de la materia radica en que aporta al perfil del egresado el conocimiento, la correcta selección y aplicación de las estructuras de datos en la solución de problemas, así como el determinar la eficiencia de algoritmos que permitan la selección de los mismos con el fin de desarrollar soluciones eficientes.

Puesto que para llevar esta materia es indispensable conocer la programación orientada a objetos, esta se encuentra ubicada para ser cursada después de las materias de Fundamentos de Programación y de Programación Orientada a Objetos, siendo esta materia pilar fundamental en el análisis, diseño y desarrollo de sistemas de información.

Intención didáctica.

Esta materia está organizada en siete unidades. En ella, se distinguen claramente dos apartados: primero, la implementación de las estructuras de datos lineales y no lineales a través del manejo de memoria estática y dinámica; segundo, el análisis de los métodos de ordenamiento de datos internos para considerar su eficiencia en la aplicación de soluciones computacionales.

Se inicia el curso con el tratamiento de los tipos de datos abstractos. Para estudiar cada tipo de dato abstracto, es necesario aplicar la modularidad, analizando la forma en que se gestiona la memoria para almacenarlos.

La segunda unidad aborda la definición, mecanismos y características de la recursividad, aplicando éstos a la creación de procedimientos, así como el análisis de las ventajas y desventajas de estas soluciones recursivas. Los estudiantes identifican dichas características de la recursividad y ejemplifican el caso de las Torres de Hanoi, Serie de Fibonacci y Factorial entre otros para comprender mejor el mecanismo recursivo.

La tercera unidad trata sobre las estructuras lineales: listas, pilas y colas. La representación de pilas y colas puede darse a través de vectores (memoria estática) o apuntadores y/o referencias (memoria dinámica). Se analizan también otras variantes como el caso de colas circulares, colas de prioridad, listas simples y doblemente enlazadas. Los estudiantes desarrollan aplicaciones para resolver problemas que requieran de estos tipos de

¹ Sistema de asignación y transferencia de créditos académicos

estructuras.

La cuarta unidad trata de las estructuras no lineales conocidas como árboles y grafos que permiten dar solución a problemas más complejos a través de la recursividad y la utilización de memoria dinámica. Se analizan los recorridos típicos de árboles binarios, búsquedas entre otros, así como el algoritmo del viajero para operaciones con grafos.

En la quinta unidad, los estudiantes identifican la metodología de cada algoritmo de ordenamiento interno (memoria principal) y externos (memoria secundaria) midiendo su comportamiento en condiciones similares.

Con la intención de que el estudiante conozca otras estrategias para almacenar y recuperar los datos así como fortalecer la seguridad de la información que se administra, se estudia la sexta unidad encargada precisamente de los métodos de recuperación de información.

Para concluir el curso se realiza un estudio sobre el análisis de la complejidad y eficiencia de los algoritmos, lo cual permitirá determinar cuáles son los algoritmos más eficientes para solucionar un problema. Al finalizar la materia se habrá adquirido las bases para evaluar e implementar soluciones por medio de estructuras.

Los contenidos se abordarán de manera secuencial como lo marca el programa, buscando la aplicación del conocimiento en un proyecto integrador que incorpore de manera progresiva los temas revisados en la materia; con un enfoque basado en actividades que promuevan en el estudiante el desarrollo de sus habilidades para trabajar en equipo y aplicar el conocimiento a la práctica.

El profesor además de ser un motivador permanente en el proceso educativo deberá ser promotor y director de la enseñanza a través de la transmisión de su conocimiento, así como la aplicación de sus habilidades y destrezas utilizando las herramientas tradicionales y digitales a su alcance para cautivar a sus estudiantes e interesarlos en el tema.

3.- COMPETENCIAS A DESARROLLAR

<p>Competencias específicas:</p> <p>Identificar, seleccionar y aplicar eficientemente tipos de datos abstractos, métodos de ordenamiento y búsqueda para la optimización del rendimiento de soluciones de problemas del mundo real.</p>	<p>Competencias genéricas:</p> <p>Competencias instrumentales</p> <ul style="list-style-type: none">• Capacidad de análisis y síntesis• Capacidad de organizar y planificar• Conocimientos básicos de la carrera• Comunicación oral y escrita• Habilidades básicas de manejo de la computadora• Habilidad para buscar y analizar información proveniente de fuentes diversas• Solución de problemas• Toma de decisiones. <p>Competencias interpersonales</p> <ul style="list-style-type: none">• Capacidad crítica y autocrítica• Trabajo en equipo• Capacidad de comunicación interdisciplinaria• Apreciación de la diversidad y multiculturalidad.• Compromiso ético <p>Competencias sistémicas</p> <ul style="list-style-type: none">• Capacidad de aplicar los conocimientos• Habilidades de investigación• Capacidad de aprender• Capacidad de generar nuevas ideas (creatividad)• Liderazgo• Habilidad para trabajar en forma autónoma• Búsqueda del logro
--	---

4.- HISTORIA DEL PROGRAMA

Lugar y fecha de elaboración o revisión	Participantes	Observaciones (cambios y justificación)
Instituto Tecnológico de 5 al 10 de Octubre de 2009. Saltillo, Coah.	Representantes de los Institutos Tecnológicos participantes en el diseño de la carrera de Ingeniería en Informática.	Reunión Nacional de Diseño e Innovación Curricular de la carrera de Ingeniería en Informática
Institutos Tecnológicos de: Colima, Lerdo, Tijuana y Zacapoaxtla.	Representantes de las Academias de Sistemas y Computación.	Análisis, enriquecimiento y elaboración del programa de estudio propuesto en la Reunión Nacional de Diseño Curricular de la Carrera de: Ingeniería en Informática y Ingeniería en Sistemas Computacionales.
Institutos Tecnológicos de: Tijuana, Mexicali, Toluca.		
Institutos Tecnológicos Superiores de: Coahuila de Zaragoza, Occidente del Estado de Hidalgo, Lerdo y Tepexi de Rodríguez. 14 de Octubre de 2009 al 19 de Febrero de 2010.		
Instituto Tecnológico de Poza Rica, Veracruz. 22 al 26 de Febrero 2010.	Representantes de los Institutos Tecnológicos: El Llano, Ciudad Juárez, Colima, Comitancillo, Los Mochis, Oaxaca y Tijuana.	Reunión Nacional de Consolidación de la Carrera de Ingeniería en Informática.
Instituto Tecnológico de Aguascalientes, del 15 al 18 de Junio de 2010.	Representantes de los Institutos Tecnológicos de: Aguascalientes, Apizaco, Cd. Madero, Colima, La Paz, Toluca y Villahermosa.	Reunión Nacional de Fortalecimiento Curricular de las asignaturas comunes por áreas de conocimiento para los planes de estudios actualizados del SNEST.

5.- OBJETIVO(S) GENERAL(ES) DEL CURSO (competencias específicas a desarrollar en el curso)

Identificar, seleccionar y aplicar eficientemente tipos de datos abstractos, métodos de ordenamiento y búsqueda para la optimización del rendimiento de soluciones de problemas del mundo real.

6.- COMPETENCIAS PREVIAS

- Utilizar técnicas de modelado para la solución de problemas.
- Aplicar la sintaxis de un lenguaje orientado a objetos.
- Aplicar un lenguaje orientado a objetos para la solución de problemas.

7.- TEMARIO

Unidad	Temas	Subtemas
1	Introducción a las estructuras de datos.	1.1 Tipos de datos abstractos (TDA). 1.2 Modularidad. 1.3 Uso de TDA. 1.4 Manejo de memoria estática. 1.5 Manejo de memoria dinámica
2	Recursividad	2.1 Definición 2.2 Procedimientos recursivos 2.3 Ejemplos de casos recursivos
3	Estructuras lineales	Listas. Operaciones básicas con listas. Tipos de listas. Listas simplemente enlazadas. Listas doblemente enlazadas. Listas circulares. Aplicaciones. Pilas. Representación en memoria estática y dinámica. Operaciones básicas con pilas. Aplicaciones. Notación infija y postfija. Recursividad con ayuda de pilas. Colas. Representación en memoria estática y dinámica. Operaciones básicas con colas. Tipos de colas: Cola simple, Cola circular y Colas dobles. Aplicaciones: Colas de prioridad.
4	Estructuras no lineales	4.1 Árboles. 4.1.1 Concepto de árbol. 4.1.2 Clasificación de árboles.

		4.1.3 Operaciones básicas sobre árboles binarios. 4.1.4 Aplicaciones. 4.1.5 Árboles balanceados (AVL). 4.2 Grafos. 4.2.1 Terminología de grafos. 4.2.2 Operaciones básicas sobre grafos.
5	Métodos de ordenamiento	5.1 Algoritmos de Ordenamiento Internos 5.1.1 Burbuja. 5.1.2 Quicksort. 5.1.3 ShellSort. 5.1.4 Radix 5.2 Algoritmos de ordenamiento Externos 5.2.1 Intercalación 5.2.2 Mezcla Directa 5.2.3 Mezcla Natural
6	Métodos de búsqueda	6.1 Búsqueda secuencial 6.2 Búsqueda binaria 6.3 Búsqueda por funciones de HASH
7	Análisis de los algoritmos	7.1 Complejidad en el tiempo. 7.2 Complejidad en el espacio. 7.3 Eficiencia de los algoritmos.

8.- SUGERENCIAS DIDÁCTICAS (desarrollo de competencias genéricas)

El profesor debe:

Ser conocedor de la disciplina que está bajo su responsabilidad, conocer su origen y desarrollo histórico para considerar este conocimiento al abordar los temas. Desarrollar la capacidad para coordinar y trabajar en equipo; orientar el trabajo del estudiante y potenciar en él la autonomía, el trabajo cooperativo y la toma de decisiones. Mostrar flexibilidad en el seguimiento del proceso formativo y propiciar la interacción entre los estudiantes. Tomar en cuenta el conocimiento de los estudiantes como punto de partida y como obstáculo para la construcción de nuevos conocimientos.

- Propiciar actividades de metacognición. Ante la ejecución de una actividad, señalar o identificar el tipo de proceso intelectual que se realizó: una identificación de patrones, un análisis, una síntesis, la creación de un heurístico, etc. Al principio lo hará el profesor, luego será el alumno quien lo identifique, como: reconocer el mecanismo natural de implementación de listas; identificar las variantes para implementar pilas y colas.
- Propiciar actividades de búsqueda, selección y análisis de información en distintas fuentes, como: buscar y contrastar definiciones de las estructuras de datos lineales y no lineales identificando puntos de coincidencia entre unas y otras definiciones e identificar cada estructura en situaciones concretas.
- Fomentar actividades grupales que propicien la comunicación, el intercambio argumentado de ideas, la reflexión, la integración y la colaboración de y entre los estudiantes, como: presentar en equipo o plenaria los resultados de las soluciones a los problemas propuestos.
- Relacionar los contenidos de esta asignatura con las demás del plan de estudios a las que ésta da soporte para desarrollar una visión interdisciplinaria en el estudiante, como: identificar las estructuras de datos, los casos en que se aplica las estructuras lineales y en cuales se aplican las no lineales; conocer cómo el sistema operativo requiere de las estructuras de datos en el manejo de los diferentes procesos; distinguir cuando aplicar los métodos de ordenamiento y búsqueda en el manejo de los datos.
- Propiciar el desarrollo de capacidades intelectuales relacionadas con la lectura, la escritura y la expresión oral, como: redactar reportes sobre resultados obtenidos, así como la exposición de las conclusiones, ante el grupo, obtenidas durante el desarrollo de la actividad.
- Propiciar el desarrollo de actividades intelectuales de inducción-deducción y análisis-síntesis, que encaminen hacia la investigación.
- Desarrollar actividades de aprendizaje que propicien la aplicación de los conceptos, modelos y metodologías que se van aprendiendo en el desarrollo de la asignatura.
- Proponer problemas que permitan al estudiante la integración de contenidos de la asignatura y entre distintas asignaturas, para su análisis y solución.
- Relacionar los contenidos de la asignatura con el uso responsable de los recursos del medio ambiente.
- Cuando los temas lo requieran, utilizar Tecnologías de Información y Comunicación para una mejor comprensión del estudiante.
- Propiciar el uso de las nuevas tecnologías en el desarrollo de la asignatura (lenguajes de programación orientados a objetos, procesador de texto, hoja de cálculo, Internet, etc.).

- Promover actividades de educación holística, como promover valores personales y sociales a través de actividades de crecimiento personal, asistencia social y el cuidado del medio ambiente.
- Definición y diseño de rúbricas.

9.- SUGERENCIAS DE EVALUACIÓN

La evaluación será permanente, enfocada en los conocimientos, su aplicación y la actitud mostrada por el estudiante. Se considerará el desempeño en cada una de las actividades de aprendizaje, haciendo especial énfasis en:

- Reportes escritos de los resultados u observaciones obtenidas durante las actividades realizadas en cada unidad académica, así como de las conclusiones obtenidas de dichas observaciones.
- Información recabada durante las consultas e investigaciones solicitadas, plasmadas en documentos escritos.
- Descripción de otras experiencias concretas que se obtendrán al participar en discusiones, exposiciones o cualquier otro medio didáctico-profesional que trate sobre la materia y que deberán realizarse durante el curso académico.
- Exámenes teórico-prácticos para comprobar la efectividad del estudiante en la comprensión de aspectos teóricos y su aplicación a la solución de casos prácticos.
- Presentación y exposición de cada actividad de aprendizaje. Algunas se evaluarán por equipo.
- Integración de rúbricas en el Portafolio de evidencias.

10.- UNIDADES DE APRENDIZAJE

Unidad 1: Introducción a las estructuras de datos

Competencia específica a desarrollar	Actividades de Aprendizaje
<ul style="list-style-type: none">• Representar y aplicar los tipos de datos abstractos por medio de un lenguaje de programación.	<ul style="list-style-type: none">• Práctica de ejercicios. Elaborar un programa en un lenguaje de programación que incorpore en un vector de n elementos sus operaciones básicas: insertar, eliminar, ordenar, buscar, cantidad de memoria estática utilizada y tiempo de ordenamiento.• Práctica de ejercicios. Elaborar un programa en un lenguaje de programación que incorpore en una lista de n elementos de forma ordenada. Considere además las funciones de eliminar, buscar y cantidad de memoria dinámica utilizada.

Unidad 2: Recursividad

Competencia específica a desarrollar	Actividades de Aprendizaje
<ul style="list-style-type: none">• Comprender y aplicar la recursividad como herramienta de programación para el manejo de las estructuras de datos.	<ul style="list-style-type: none">• Consultar en las fuentes impresas el concepto de recursividad.• Ejemplificar un caso recursivo de la vida cotidiana como calcular el Factorial de un número entero positivo.• Práctica de ejercicios. Identificar problemas resueltos de manera iterativa y encontrar su

	<p>solución recursiva mediante codificación en algún lenguaje de programación.</p> <ul style="list-style-type: none"> • Participar en plenaria para determinar las ventajas y desventajas del uso de la recursividad.
--	--

Unidad 3: Estructuras lineales

Competencia específica a desarrollar	Actividades de Aprendizaje
<ul style="list-style-type: none"> • Conocer, identificar y aplicar las estructuras lineales en la solución de problemas del mundo real. 	<ul style="list-style-type: none"> • Consultar las características de las estructuras lineales en fuentes bibliográficas y comentarlas en plenaria. • Prácticas de ejercicios. Utilizando un lenguaje de programación implemente las operaciones básicas (insertar, eliminar y buscar) en listas simples y doblemente enlazadas. • Práctica de ejercicios. Utilizando un lenguaje de programación implemente las operaciones básicas (insertar y eliminar) en una pila en sus modalidades estática y dinámica. • Práctica de ejercicios. Utilizando un lenguaje de programación implemente las operaciones básicas (insertar y eliminar) en una cola en sus modalidades estática y dinámica. Incorpore además sus variantes de cola circular y bicola.

Unidad 4: Estructuras no lineales

Competencia específica a desarrollar	Actividades de Aprendizaje
<ul style="list-style-type: none"> • Conocer, identificar y aplicar las estructuras no lineales en la solución de problemas del mundo real. 	<ul style="list-style-type: none"> • Consultar en las fuentes bibliográficas la terminología sobre árboles. Comentar información en plenaria. • Práctica de ejercicios. Utilizando un lenguaje de programación implemente las operaciones básicas (insertar, eliminar, buscar) en un árbol binario de búsqueda, así como los recorridos en preorden, inorden y postorden.

Unidad 5: Métodos de ordenamiento

Competencia específica a desarrollar	Actividades de Aprendizaje
<ul style="list-style-type: none">• Aplicar el método de ordenamiento pertinente en la solución de un problema real.	<ul style="list-style-type: none">• Consultar en las fuentes bibliográficas los diversos algoritmos de ordenamiento y analizarlos en plenaria.• Plantear en el aula, problemas donde se justifique la necesidad de utilizar métodos de ordenamiento.• Práctica de ejercicios. Implementar en un lenguaje de programación los métodos de ordenamiento para un conjunto de n datos generados aleatoriamente.• Generar cuadro comparativo con los tiempos obtenidos en cada método de ordenamiento.

Unidad 6: Métodos de búsqueda

Competencia específica a desarrollar	Actividades de Aprendizaje
<ul style="list-style-type: none">• Aplicar el método de búsqueda pertinente en la solución de un problema real.	<ul style="list-style-type: none">• Consultar en las fuentes bibliográficas los diversos algoritmos de búsqueda y analizarlos en plenaria.• Práctica de ejercicios. Implementar en un lenguaje de programación los métodos de búsqueda para un conjunto de n datos generados aleatoriamente.• Generar cuadro comparativo con los tiempos obtenidos en cada método de búsqueda.

Unidad 7: Análisis de algoritmos

Competencia específica a desarrollar	Actividades de Aprendizaje
<ul style="list-style-type: none">• Comprender la complejidad de los algoritmos e identificar la eficiencia de los mismos.	<ul style="list-style-type: none">• Generar cuadro comparativo con los tiempos obtenidos en cada método de ordenamiento y búsqueda.• Comentario. Emitir su propia conclusión al observar la complejidad de los algoritmos estudiados y contrastarla con lo que dicen al respecto, los autores citados en las fuentes bibliográficas.

11.- FUENTES DE INFORMACIÓN

Impresas:

1. *Cairo, Osvaldo; Guardati, Silvia*. Estructura de Datos, Tercera Edición. Mc Graw Hill, México, 2006.
2. *Joyanes Aguilar, Luis*. Fundamentos de Programación. Algoritmos y Estructuras de Datos. Tercera Edición 2003. McGraw – Hill.
3. *Guardati, Silvia*. Estructura de Datos Orientada a Objetos Algoritmos con C++, Primera Edición. Prentice Hall, México, 2007.
4. *Mark Allen Weiss*. Estructura de datos en Java. Ed. Addison Wesley.
5. *C. Thomas Wu*. Introducción a la Programación Orientada a Objetos con Java. Ed. Pearson Educación.
6. *Decker, Hirshfield*. Programación con Java. Ed. International Thomson Editores.
7. *Roman Martínez, Elda Quriga*. Estructura de Datos Referencia practica con orientación a objetos. Ed. Thomson, Mexico, 2004.

Electrónicas:

- Abdiel Cáceres González. Mayo 2005. Estructuras de datos en C++. [Publicación en línea]. Disponible desde Internet en: <http://computacion.cs.cinvestav.mx/~acaceres/courses/estDatosCPP/index.html>. [Con acceso el 30-01-2010].
- SMETE. Sin fecha. Data Structure. [Publicación en línea]. Disponible desde Internet en: <http://courses.cs.vt.edu/~csonline/DataStructures/Lessons/index.html?downloadURL=true&lold=17BEFD1C-BF52-417E-98D2-9582AF0C7662> [Con acceso el 30-01-2010].

12.- PRÁCTICAS PROPUESTAS

1. Introducción a las estructuras de datos.
 - a. Elaborar un cuadro comparativo que muestre la representación de la memoria estática y dinámica.
 - b. Desarrollar una aplicación donde se utilicen TDA para comprobar el comportamiento estático y dinámico.
2. Recursividad.
 - a. Elaborar práctica de ejercicios utilizando un lenguaje de programación que resuelva mediante la técnica de recursividad, para generar la Serie de Fibonacci.
 - b. Elaborar práctica de ejercicios utilizando un lenguaje de programación que resuelva mediante la técnica de recursividad, el número de movimientos de anillos en la Torre de Hanoi.
3. Estructuras lineales.
 - a. Elaborar práctica de ejercicios utilizando un lenguaje de programación que resuelva los siguientes ejercicios de estructuras lineales:
 - i. Desarrollar programa que simule una lista de espera para la asignación de mesas en un restaurante.
 - ii. Desarrollar programa para simular las llamadas a funciones utilizando una pila.
 - iii. Un evaluador de expresiones posfijas funciona sobre expresiones aritméticas que tienen esta forma: op1 op2 operador.... Usando dos

- pilas, una para los operandos y una para los operadores. Diseña e implementa una clase Calculadora que convierta expresiones infijas a expresiones posfijo y después utiliza las pilas para evaluar las expresiones.
- iv. Desarrollar programa que simule la salida de aviones en un aeropuerto utilizando colas de prioridad.
4. Estructuras no lineales.
 - i. Generar una aplicación utilizando los conceptos de árboles AVL.
 - ii. Implementar el algoritmo “El viajero” mediante (camino mínimo) grafos.
 - iii. Construir un grafo ponderado que modele una sección de su Estado de origen. Utilice el algoritmo de Dijkstra para determinar el camino más corto, desde el vértice inicial hasta el último vértice.
 5. Métodos de ordenamiento.
 - a. Elaborar práctica de ejercicios utilizando un lenguaje de programación que resuelva los siguientes ejercicios de métodos de ordenamiento:
 - i. Aplicar los métodos de ordenamiento a un conjunto de n datos y determinar su complejidad.
 - b. Elaborar un cuadro comparativo de los registros del tiempo empleado por cada método de ordenamiento para datos en orden aleatorio, ordenados y orden inverso.
 6. Métodos de búsqueda.
 - a. Aplicar los métodos de búsqueda a un conjunto de n datos y determinar su eficiencia.
 - b. Elaborar su propio método de búsqueda
 7. Realizar dos cuadros comparativos que reflejen la complejidad en el tiempo y en el espacio de los métodos de:
 - a. Ordenamiento.
 - b. Búsqueda.