

## 1.- DATOS DE LA ASIGNATURA

Nombre de la asignatura: <b>Estructura de datos</b>
Carrera: <b>Licenciatura en Informática</b>
Clave de la asignatura: <b>IFM - 0412</b>
Horas teoría-horas práctica-créditos <b>3-2-8</b>

## 2.- HISTORIA DEL PROGRAMA

<b>Lugar y fecha de elaboración o revisión</b>	<b>Participantes</b>	<b>Observaciones (cambios y justificación)</b>
Instituto Tecnológico de Puebla del 8 al 12 septiembre 2003.	Representantes de la academia de sistemas y computación de los Institutos Tecnológicos.	Reunión nacional de evaluación curricular de la carrera de Licenciatura en Informática.
Institutos Tecnológicos de: Cancún, Cd. Juárez, Culiacán, Matamoros, Ocotlán, Tijuana. 13 septiembre al 28 de noviembre 2003.	Academias de sistemas y computación.	Análisis y enriquecimiento de las propuestas de los programas diseñados en la reunión nacional de evaluación .
Instituto Tecnológico de Tepic 15 al 19 de marzo 2004.	Comité de consolidación de la carrera de Licenciatura en Informática.	Definición de los programas de estudio de la carrera de Licenciatura en Informática.

### 3.- UBICACIÓN DE LA ASIGNATURA

#### a). Relación con otras asignaturas del plan de estudio

Anteriores		Posteriores	
Asignaturas	Temas	Asignaturas	Temas
Fundamentos de Programación.	Fundamento del paradigma orientado a objetos. Construcción del estado de un objeto. Construcción del comportamiento de un objeto, Modelado de aplicaciones utilizando la POO.	Organización de datos.	Conceptos básicos de archivos. Archivo secuencial. Archivo secuencial indexado. Archivo directo. Métodos de ordenamiento y búsqueda.
Programación I.	Fundamentos del lenguaje. Subprogramas. Punteros, referencias y arreglos. Clases y objetos. Herencia. Polimorfismo.		

#### b). Aportación de la asignatura al perfil del egresado

- Proporciona el conocimiento teórico práctico para la utilización de las estructuras de datos en un programa.
- Desarrolla habilidades y actitudes que le permiten identificar problemas y oportunidades donde se aplique el tratamiento de información para proponer soluciones por medio de modelos y facilitar con ello la toma de decisiones.
- Desarrolla y administra sistemas de información para aumentar la productividad y competitividad de las organizaciones.

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

Aprenderá las principales estructuras de datos desde un punto de vista abstracto y las operaciones que se puedan realizar sobre ellas, aplicando en forma práctica los conceptos adquiridos mediante resolución de problemas.

#### 5.- TEMARIO

Unidad	Temas	Subtemas
1	Tipos de datos.	<ul style="list-style-type: none"><li>1.1 Tipos de datos.<ul style="list-style-type: none"><li>1.1.1 Tipos de datos simples.<ul style="list-style-type: none"><li>1.1.1.1 Definición de bit, byte, carácter y palabra.</li><li>1.1.1.2 Manipulación de bits.</li><li>1.1.1.3 Representación de datos simples.</li></ul></li><li>1.1.2 Tipos de datos abstractos.</li></ul></li><li>1.2 Estructuras de datos.<ul style="list-style-type: none"><li>1.2.1 Definición.</li><li>1.2.2 Clasificación.<ul style="list-style-type: none"><li>1.2.2.1 Lineales y no lineales.</li><li>1.2.2.2 Dinámicas y estáticas.</li></ul></li></ul></li></ul>
2	Estructuras lineales.	<ul style="list-style-type: none"><li>2.1 Arreglos.<ul style="list-style-type: none"><li>2.1.1 Definición.</li><li>2.1.2 Unidimensionales.</li><li>2.1.3 Bidimensionales.</li><li>2.1.4 Multidimensionales.</li><li>2.1.5 Resolución de problemas con arreglos.</li><li>2.1.6 Clases para la implementación de arreglos.</li></ul></li><li>2.2 Pilas.<ul style="list-style-type: none"><li>2.2.1 Definición.</li><li>2.2.2 Operaciones.</li><li>2.2.3 Clases para la implementación de pilas.</li></ul></li></ul>

## 5.- TEMARIO (Continuación)

3	Listas enlazadas.	<ul style="list-style-type: none"><li>2.3 Colas.<ul style="list-style-type: none"><li>2.3.1 Definición.</li><li>2.3.2 Tipos.<ul style="list-style-type: none"><li>2.3.2.1 Colas simples.</li><li>2.3.2.2 Colas circulares.</li><li>2.3.2.3 Colas dobles.</li></ul></li><li>2.3.3 Operaciones.</li><li>2.3.4 Clases para la implementación de colas.</li></ul></li><li>3.1 Listas enlazadas.<ul style="list-style-type: none"><li>3.1.1 Simples.</li><li>3.1.2 Dobles.</li><li>3.1.3 Circulares.</li><li>3.1.4 Multilistas.</li><li>3.1.5 Clases para la implementación de listas.</li></ul></li></ul>
4	Estructuras no lineales.	<ul style="list-style-type: none"><li>4.1 Árboles.<ul style="list-style-type: none"><li>4.1.1 Definición.</li><li>4.1.2 Representación en memoria de árboles.<ul style="list-style-type: none"><li>4.1.2.1 Árboles generales.</li><li>4.1.2.2 Árboles binarios.</li></ul></li><li>4.1.3 Recorridos en un árbol binario.<ul style="list-style-type: none"><li>4.1.3.1 Preorden.</li><li>4.1.3.2 Inorden.</li><li>4.1.3.3 Posorden.</li></ul></li><li>4.1.4 Balanceo de árboles binarios.</li><li>4.1.5 Clases para la implementación de árboles.</li></ul></li><li>4.2 Grafos.<ul style="list-style-type: none"><li>4.2.1 Definición.</li><li>4.2.2 Tipos de grafos.</li><li>4.2.3 Representación de grafos en memoria.</li><li>4.2.4 Clases para la implementación de grafos.</li></ul></li></ul>

## **6.- APRENDIZAJES REQUERIDOS**

- Conocimiento teórico práctico para la solución de problemas utilizando el paradigma de la programación orientada a objetos.
- Manejo de lenguajes de programación orientados a objetos (al menos uno).

## **7.- SUGERENCIAS DIDÁCTICAS**

- Desarrollar en coordinación de los estudiantes ejemplos de lo simple a lo complejo, buscando se asocie el tema de estudio con elementos significativos del entorno.
- Hacer de las horas prácticas un taller donde el profesor coordine el desarrollo de ejemplos, teniendo el cuidado que durante la solución se resalten los puntos de importancia que la teoría define, ya sea como concepto o el uso de la sintaxis en el momento de su aplicación.
- Elaborar en coordinación con los estudiantes un conjunto de problemas actuales asociados al entorno.
- Solicitar el desarrollo de un proyecto del interés de los estudiantes, aplicando los conocimientos adquiridos durante el curso.
- Definir un esquema de documentación de los trabajos ya sea de investigación o desarrollo de aplicaciones, el cual sustente las estrategias metodológicas del caso.
- Seleccionar, en acuerdo de academia, el lenguaje de programación a utilizar de acuerdo a las necesidades del entorno, el contenido del programa y los recursos disponibles.

## **8.- SUGERENCIAS DE EVALUACIÓN**

- Realizar una evaluación diagnóstica al inicio del curso para conocer el desempeño académico del estudiante y programar las actividades del curso.
- Trabajos de investigación.
- Tareas relacionadas con la solución de problemas de programación utilizando las estructuras de datos que forman parte del contenido de la materia.
- Establecer de común acuerdo con los estudiantes, la ponderación de las diferentes actividades del curso.
- En aquellas unidades que requieran la solución de problemas se recomienda que la evaluación se realice mediante el uso de la computadora.
- Seguimiento de los avances y documentación del proyecto propuesto.

## 9.- UNIDADES DE APRENDIZAJE

### UNIDAD 1.- Tipos de datos.

<b>Objetivo Educativo</b>	<b>Actividades de Aprendizaje</b>	<b>Fuentes de Información</b>
<p>El estudiante comprenderá la necesidad de la utilización de las estructuras de datos para la solución de problemas basados en computadora.</p> <p>Comprenderá la representación en memoria de los diferentes tipos de datos.</p>	<p>1.1 Plantear en el aula, problemas donde se justifique la necesidad de utilizar estructuras de datos.</p> <p>1.2 Comprobar a través de ejemplos, que el almacenamiento de los elementos de las estructuras de datos estáticas, se almacenan en forma contigua.</p> <p>1.3 Investigar en diversas fuentes de información las clases predefinidas para el manejo de estructuras de datos en al menos dos lenguajes de programación, elaborar un informe sobre el particular.</p>	<p>1, 2, 5, 6</p>

### UNIDAD 2.- Estructuras lineales.

<b>Objetivo Educativo</b>	<b>Actividades de Aprendizaje</b>	<b>Fuentes de Información</b>
<p>Comprenderá el comportamiento de las estructuras de datos lineales.</p> <p>Comprenderá la importancia de utilizar las estructuras de datos lineales para mejorar el desempeño de las aplicaciones.</p>	<p>2.1 Plantear en el aula, problemas donde se justifique la necesidad de utilizar estructuras de datos lineales.</p> <p>2.2 Desarrollar aplicaciones donde se ejemplifique el comportamiento de las estructuras lineales utilizando un lenguaje orientado a objetos.</p> <p>2.3 Investigar en diversas fuentes de información las clases predefinidas para el manejo de estructuras de datos lineales en al menos dos lenguajes de programación, elaborar un informe sobre el particular.</p>	<p>1, 2, 3, 5, 6</p>

### UNIDAD 3.- Listas enlazadas.

<b>Objetivo Educativo</b>	<b>Actividades de Aprendizaje</b>	<b>Fuentes de Información</b>
Comprenderá el comportamiento de las listas enlazadas .  Comprenderá la importancia de utilizar listas enlazadas para mejorar el desempeño de las aplicaciones.	3.1 Plantear en el aula, problemas donde se justifique la necesidad de utilizar listas enlazadas.  3.2 Desarrollar aplicaciones donde se utilice el comportamiento de las listas enlazadas mediante un lenguaje orientado a objetos.  3.3 Investigar en diversas fuentes de información las clases predefinidas para el manejo de listas enlazadas en al menos dos lenguajes de programación, elaborar un informe sobre el particular.	1, 2, 3, 5, 6

### UNIDAD 4.- Estructuras no lineales.

<b>Objetivo Educativo</b>	<b>Actividades de Aprendizaje</b>	<b>Fuentes de Información</b>
Comprenderá el comportamiento de las estructuras de datos no lineales.  Comprenderá la importancia de utilizar las estructuras de datos no lineales para mejorar el desempeño de las aplicaciones.	4.1 Plantear en el aula, problemas donde se justifique la necesidad de utilizar estructuras de datos no lineales.  4.2 Desarrollar aplicaciones donde se ejemplifique el comportamiento de las estructuras no lineales utilizando un lenguaje orientado a objetos.  4.3 Investigar en diversas fuentes de información las clases predefinidas para el manejo de estructuras de datos no lineales en al menos dos lenguajes de programación, elaborar un informe sobre el particular.	1, 2, 3, 5, 6

## 10. FUENTES DE INFORMACIÓN

1. Cairó.  
Estructuras de datos, 2ª Edición.  
Ed. Mc Graw Hill, 2001.
2. Mark Allen Weiss Mark.  
Estructuras de datos en Java.  
Ed. Pearson.
3. Heileman.  
Estructuras de datos, algoritmos y programación orientada a objetos.  
Ed. Mc Graw Hill, 2001. 1997.
4. Brassard y Bratley.  
Fundamentos de Algoritmia.  
Ed. Prentice-Hall.
5. Schildt.  
C ++, Guía de autoenseñanza.  
Ed. Mc Graw Hill, 2001.
6. Schildt,.  
C#. Manual de referencia.  
Ed. Mc Graw Hill, 2003.

## 11. PRÁCTICAS

### Unidad Práctica

- 1 MANEJO DE ARREGLOS Y CADENAS. Utilizar arreglos de una dimensión y cadenas como arreglos de caracteres. Maneje arreglos declarados estáticamente y arreglos creados durante la ejecución del programa.
- 2 MANEJO DE ARREGLOS BIDIMENSIONALES Y MULTIDIMENSIONALES. Manejar arreglos bidimensionales (matrices) y multidimensionales en la solución de problemas.
- 3 REPRESENTACIÓN DE MATRICES EN VECTORES. Comprender como se representan estructuras bidimensionales, como matrices y arreglos multidimensionales en una estructura lineal(vector).
- 4 REPRESENTACIÓN DE MATRICES POCO DENSAS REGULARES EN UNA ESTRUCTURA LINEAL ÓPTIMA. Representar matrices regulares (triangular, simétrica, antisimétrica, tridiagonal) en una estructura lineal óptima (vector).
- 5 REPRESENTACIÓN DE MATRICES POCO DENSAS NO REGULARES. Comprender la representación de matrices poco densas irregulares (matrices dispersas).
- 6 CLASE DE ARREGLOS, ARREGLOS DESORDENADOS Y ORDENADOS, OPERACIONES EN ARREGLOS. Aplicar los conceptos de herencia y polimorfismo de la programación orientada a objetos (POO) en la definición de una clase base abstracta Arreglo y las definiciones de las clases derivadas para Arreglos Ordenados y Arreglos Desordenados. Definir e implementar en ambas clases las operaciones de inserción, modificación, eliminación, búsqueda y ordenamiento de elementos en los arreglos.
- 7 COLA CIRCULAR. Manejar una estructura de datos Cola, implantada en un arreglo circular (cola estática). Definir las operaciones en la cola y manejarla con una clase genérica a través del uso de una plantilla.
- 8 DIFERENTES TIPOS DE COLAS. Conocer y manejar otro tipo de colas, como son las colas dobles y las colas de prioridad entre otras.

- 9 APLICACIÓN DE COLAS. Hacer uso de las estructuras de tipo cola en la solución de algún problema particular, por ejemplo en la simulación del comportamiento de una cola de espera.
- 10 PILAS. Manejar una estructura de datos, pila, implantada en un arreglo (pila estática). Definir las operaciones en la pila y manejarla con una clase genérica a través del uso de una plantilla.
- 11 ALGORITMOS DE CONVERSIÓN DE EXPRESIONES EN NOTACIÓN INFIXA A NOTACIÓN PREFIJA O POSTFIJA. Implantar y usar los algoritmos de conversión de expresiones infija, postfija y prefija. Comprender las ventajas de las dos últimas notaciones sobre la primera. Implantar luego el algoritmo de evaluación de expresiones.
- 12 MANEJO DE RECURSIVIDAD Y APLICACIÓN DE PILAS PARA LA IMPLANTACIÓN DE RECURSIVIDAD EN FUNCIONES. Entender el concepto de recursividad y comprender como se implanta el manejo de funciones recursivas en un compilador mediante el uso de una pila.
- 13 LISTAS ENLAZADAS. Elaborar programas para ejemplificar el comportamiento de las listas (ordenadas, circulares, doblemente ligadas, anillos, pilas, colas, etc.).
- 14 ÁRBOLES BINARIOS. Representar una estructura de árbol binario a través del uso de apuntadores. Hacer recorridos en el árbol en cualquier orden (inorden, preorden o postorden). Definir una clase genérica de árbol binario a través del uso de una plantilla.
- 15 ÁRBOLES DE BÚSQUEDA BINARIA. Implantar las operaciones de inserción, eliminación, y búsqueda en un árbol de búsqueda binaria. Analizar cuales son el peor caso y el mejor caso para esta clase de árboles, entender de que orden es la búsqueda en un árbol binario, calcular la longitud de camino interno promedio y la longitud de camino externo promedio.
- 16 APLICACIÓN DE ÁRBOLES BINARIOS. Utilizar las estructuras de árbol para la solución de algún problema determinado, por ejemplo en la evaluación de expresiones, en la representación de una estructura sintáctica de un elemento de un lenguaje de programación, etc.