

**UNIVERSIDAD AUTÓNOMA DE CHIAPAS
LICENCIATURA EN SISTEMAS COMPUTACIONALES**

Área de formación: Disciplinaria
Unidad académica: Estructura de Datos
Ubicación: Tercer Semestre
Clave: 655
Horas semana-mes: 4
Horas teoría: 2
Horas práctica: 2
Unidades CONAIC: 42.67
Prerrequisitos: Programación Estructurada
Horas de infraestructura: 2
Créditos: 6

PRESENTACIÓN

En este curso, el alumno aprenderá a establecer estructuras lógicas de datos que le permitan hacer uso más eficiente del espacio de memoria, minimizar el tiempo de acceso y/o lograr formas más efectivas de inserción y eliminación de datos.

OBJETIVO GENERAL

El alumno aplicará métodos de las estructuras de datos, algoritmos de ordenamiento y búsqueda para optimizar el rendimiento de la memoria en las aplicaciones programables en los lenguajes de alto nivel.

UNIDAD I.- MANEJO DE MEMORIA

TIEMPO APROXIMADO: 6 Horas

OBJETIVO DE LA UNIDAD: Identificar los conceptos básicos de manejo de memoria para su implementación en las unidades subsecuentes.

CONTENIDO

- 1.1 Manejo de memoria estática
- 1.2 Manejo de memoria dinámica

UNIDAD II.- ESTRUCTURAS LINEALES ESTÁTICAS Y DINÁMICAS

TIEMPO APROXIMADO: 12 Horas

OBJETIVO DE LA UNIDAD: Aplicar las estructuras de datos básicas para el manejo de la memoria en el desarrollo de programas.

CONTENIDO

- 2.1 Pilas
- 2.2 Colas
- 2.3 Listas enlazadas
 - 2.3.1 Simples
 - 2.3.2 Dobles
 - 2.3.3 N enlaces

UNIDAD III.- RECURSIVIDAD

TIEMPO APROXIMADO: 9 Horas

OBJETIVO DE LA UNIDAD: Aplicar el concepto de recursividad en la elaboración de programas.

CONTENIDO

- 3.1 Definición
- 3.2 Procedimientos recursivos
- 3.3 Mecánica de recursión
- 3.4 Transformación de algoritmos recursivos e iterativos
- 3.5 Recursividad en el diseño
- 3.6 Complejidad de los algoritmos recursivos

UNIDAD IV.- ESTRUCTURAS NO LINEALES ESTÁTICAS Y DINÁMICAS

TIEMPO APROXIMADO: 9 Horas.

OBJETIVO DE LA UNIDAD: Identificar las características de diferentes estructuras de datos no lineales y su implementación en lenguajes de programación.

CONTENIDO

- 4.1 Concepto de árbol
 - 4.1.1 Tipos de árbol
- 4.2 Operaciones básicas sobre árboles binarios
 - 4.2.1 Creación
 - 4.2.2 Inserción
 - 4.2.3 Eliminación
 - 4.2.4 Recorridos sistemáticos
 - 4.2.5 Balanceo

UNIDAD V.- ORDENACIÓN INTERNA

TIEMPO APROXIMADO: 10 Horas

OBJETIVO DE LA UNIDAD: Analizar los diferentes métodos de ordenamiento interno para aplicar el más adecuado en la solución de un problema.

CONTENIDO

- 5.1 Algoritmos de ordenamiento por intercambio
 - 5.1.1 Burbuja
 - 5.1.2 Quicksort
 - 5.1.3 Shell
- 5.2 Algoritmos de ordenamiento por distribución
 - 5.2.1. Radix

UNIDAD VI.- ORDENACIÓN EXTERNA

TIEMPO APROXIMADO: 9 Horas

OBJETIVO DE LA UNIDAD: Analizar los diferentes métodos de ordenamiento externo para aplicar el más adecuado en la solución de un problema.

CONTENIDO

- 6.1 Algoritmos de ordenación externa
- 6.1.1 Intercalación directa
- 6.1.2. Mezcla natural

UNIDAD VII. MÉTODOS DE BÚSQUEDA

TIEMPO APROXIMADO: 9 Horas

OBJETIVO DE LA UNIDAD: Analizar los diferentes métodos de búsqueda para recuperar información e implementarlos en lenguajes de programación.

CONTENIDO

- 7.1 Algoritmos de búsqueda interna
- 7.1.1 Secuencial
- 7.1.2 Binaria
- 7.1.3 Hash
- 7.2 Algoritmos de búsqueda externa
- 7.2.1 Secuencial
- 7.2.2 Binaria
- 7.2.3 Hash

EXPERIENCIAS DE APRENDIZAJE

- Resolución de ejercicios en el desarrollo de los temas de cada unidad
- Aplicación de los conocimientos en la resolución de un problema

CRITERIOS DE EVALUACIÓN

Exámenes escritos	30%
Tareas	30%
Proyecto final	40%
TOTAL	<u>100%</u>

RECURSOS NECESARIOS

Recursos tecnológicos:
Visual C(Microsoft) y DevC++ (GNU).

PRÁCTICAS SUGERIDAS.

- Elaborar un programa de arreglos unidimensionales.
- Elaborar un programa de arreglos bidimensionales.
- Hacer un programa que implemente la asignación dinámica de arreglos.
- Programar los diferentes métodos de ordenación que existen.
- Hacer un programa en lenguaje java que haga uso de la palabra reservada *new*.

- Crear un programa en java que contenga una estructura de libro y que llene al menos 10 libros.
- Crear un programa que muestre un menú de las 4 diferentes tipos de listas: simple, doblemente enlazada, circular, circular doblemente enlazada.
- Desarrollo de un sistema que emplee pilas y colas.

BIBLIOGRAFÍA BÁSICA

CAIRO, O. (2006). Estructura de datos. México: McGraw Hill.

JOYANES, L., et al. (2004). Algoritmos y estructuras de datos una perspectiva en C. México: McGraw Hill.

GUARDATI, S. (2007). Estructura de Datos Orientado a Objetos con C ++. México: Pearson Educación.

LEWIS, J. (2006). Estructuras de datos con Java. Madrid: Pearson Educación.

BIBLIOGRAFÍA COMPLEMENTARIA

NYHOFF, L. (2006). TADs, Estructura de datos y resolución de problemas con C++. Madrid: Pearson Educación.

DALE, N., et al. (2002). Object Oriented Data Structures using Java. USA: Jones and Bartlett.

YEDIDYAH, L., et al.(1996). Data Structures using C and C++. USA: Prentice Hall.

FORD, W., et al. (2002). Data Structures with C++ using STL. USA: Prentice Hall.

SAHNI, S. (2000). Data Structures, algorithms and applications in Java. USA: McGraw Hill.

WIRTH, N. (1987). Algoritmos y estructuras de datos. México: Prentice Hall.