



**ESTRUCTURA DE DATOS**  
**Primer Semestre 2021**

**I. INFORMACIÓN GENERAL**

<b>Código:</b> 772	<b>Créditos:</b> 5
<b>Escuela:</b> Ciencias y Sistemas	<b>Área:</b> Desarrollo de Software
<b>El curso tiene laboratorio:</b> Si	<b>Categoría:</b> Obligatorio
<b>Horas magistrales a la semana:</b> 4	<b>Horas de laboratorio a la semana:</b> 2
<b>Prerrequisitos:</b> 771 - Introducción a la Programación y Computación 2 796 – Lenguajes formales y de programación 962 - Matemática de Computo 2	<b>Postrequisitos:</b> 722 - Teoría de Sistemas 1 773 - Manejo e Implementación de Archivos 781 - Organización de Lenguajes y Compiladores 2

**II. DISTRIBUCIÓN DE SECCIONES**

Sección	Edificio	Salón	De:	A:	Lu	Ma	Mi	Ju	Vi	Sa	Catedrático
A	MEET	21	7:10	8:50			X		X		JESÚS ALBERTO GUZMÁN POLANCO
B	MEET	36	7:10	8:50			X			X	ALVARO OBAYAN HERNÁNDEZ GARCÍA
C	MEET	12	7:10	10:30					X		LUIS FERNANDO ESPINO BARRIOS

**III. DESCRIPCIÓN DEL CURSO**

Este curso continúa el estudio formal sobre varias técnicas de representación de los datos en la memoria de una computadora y de los algoritmos que los manipulan. Se enfatiza en las características de una buena programación: modularidad, ocultamiento de información, reutilización de código y estilo de programación. Todas estas características se estudian en función de los conceptos de Tipo de Dato Abstracto y Objeto, haciéndose uso de la metodología Orientada a objetos.



## VII. CONTENIDO DECLARATIVO

### UNIDAD 1: INTRODUCCIÓN A LA MEMORIA Y ANÁLISIS DE ALGORITMOS (16 PERIODOS)

**Competencia específica:** conoce los conceptos generales de las estructuras de datos y algoritmos de manejo de datos.

1. Introducción a la memoria y análisis de algoritmos
  - 1.1. Administración de memoria
    - 1.1.1. Administración de datos en la memoria (Stack y Heap).
    - 1.1.2. Memoria estática y dinámica.
    - 1.1.3. Tipos de datos primitivos, punteros y definición de tipos abstractos. Row-major y Col-major.
    - 1.1.4. Aplicación de stack y heap en la programación imperativa y la orientada a objetos.
  - 1.2. Análisis de algoritmos
    - 1.2.1. Algoritmos y eficiencia
    - 1.2.2. Análisis de casos
    - 1.2.3. Notación O grande
    - 1.2.4. Búsquedas
      - 1.2.4.1. Secuencial
      - 1.2.4.2. Binaria
    - 1.2.5. Ordenamientos
      - 1.2.5.1. Burbuja
      - 1.2.5.2. Selección
      - 1.2.5.3. Inserción
      - 1.2.5.4. Quick Sort
    - 1.2.6. Recursividad
    - 1.2.7. Programación dinámica
    - 1.2.8. Matrices dispersas
      - 1.2.8.1. Estáticas
      - 1.2.8.2. Dinámicas

### UNIDAD 2: CONCEPTOS AVANZADOS Y ESTRUCTURAS NO LINEALES BÁSICAS (16 PERIODOS)

**Competencia específica:** clasifica los algoritmos según su complejidad algorítmica y conoce las estructuras no lineales de tipo arbóreo.

2. Estructuras no lineales
  - 2.1. Árboles

- 2.1.1. Definición
- 2.1.2. Árboles binarios de búsqueda
- 2.1.3. Operaciones
- 2.1.4. Recorridos
- 2.2. AVL
  - 2.2.1. Definición
  - 2.2.2. Operaciones
  - 2.2.3. Rotaciones
- 2.3. Árboles B
  - 2.3.1. Definición
  - 2.3.2. Operaciones
- 2.4. Variantes de árboles B
  - 2.4.1. Definición
  - 2.4.2. Operaciones
- 2.5. Otros árboles
  - 2.5.1. Rojo negro
  - 2.5.2. Tie
  - 2.5.3. Árboles AST
  - 2.5.4. Árboles de Merkle (BlockChain)

### UNIDAD 3: OTRAS ESTRUCTURAS Y ALGORITMOS (12 PERIODOS)

**Competencia específica:** diseña otras estructuras de datos no lineales para solución de problemas y conoce conceptos de grafos y tablas dispersas; y conoce las maneras de codificación de datos, asimismo algoritmos de detección y corrección de errores, compresión, cifrado y cadenas de bloques.

- 3. Otras estructuras y algoritmos de codificación
  - 3.1. Grafos
    - 3.1.1. Definición
    - 3.1.2. Representación por matrices y listas de adyacencia
    - 3.1.3. Recorridos por anchura y profundidad
    - 3.1.4. Algoritmos de árboles de recubrimiento mínimo
    - 3.1.5. Algoritmos de rutas cortas y búsquedas
  - 3.2. Tablas Hash
    - 3.2.1. Hash cerrado, direccionamiento abierto
    - 3.2.2. Hash abierto, direccionamiento cerrado
  - 3.3. Algoritmos de detección y corrección de errores
    - 3.3.1. Bit de paridad
    - 3.3.2. Código de Hamming
    - 3.3.3. Checksum de Internet
  - 3.4. Algoritmos de compresión
    - 3.4.1. Algoritmo de Huffman

- 3.4.2. Algoritmo LZW
- 3.5. Algoritmos de cifrado
  - 3.5.1. Definición e historia
  - 3.5.2. Cifrado simétrico y asimétrico
  - 3.5.3. Cifrado por bloques y por flujo
  - 3.5.4. Estándares en la actualidad
- 3.6. Blockchain
  - 3.6.1. Historia y funcionamiento
  - 3.6.2. Descentralización
  - 3.6.3. Prueba de trabajo
  - 3.6.4. Algoritmos de consenso
  - 3.6.5. SmartContracts
  - 3.6.6. Tipos de blockchain
- 3.7. Seguridad en el software (investigación)
  - 3.7.1. Seguridad y factor humano
  - 3.7.2. Políticas de seguridad
  - 3.7.3. Patrones de seguridad

## VIII. CONTENIDO PROCEDIMENTAL

### UNIDAD 1:

- Descripción de conceptos
- Análisis diferentes estructuras lineales
- Clasificación de memoria

### UNIDAD 2:

- Clasificación de algoritmos según su complejidad
- Elaboración de estructuras no lineales
- Construcción de estructuras no lineales complejas para solución de problemas
- Introducción a BlockChain

### UNIDAD 3:

- Reconocimiento de rutas cortas en un grafo
- Uso de tablas dispersas
- Conocimiento de codificación y algoritmos necesarios
- Uso de métodos de cifrado

## IX. CONTENIDO ACTITUDINAL

Para todas las unidades: se interesa por el contenido, participa activamente y promueve el aporte del grupo.

## X. ACTIVIDADES DE LABORATORIO

El programa de laboratorio y el enunciado del proyecto lo redactarán los auxiliares con las siguientes bases:

### Fase 1

Duración: 2 semanas

Tema principal sugerido: Conversión de matrices estáticas a listas dinámicas.

10 puntos

### Fase 2

Duración: 4 semanas

Tema principal: Matrices dispersas, árboles AVL y B.

25 puntos

### Fase 3

Duración: 2 semanas

Tema principal: Grafos, criptografía y compresión.

15 puntos

### Fase 4

Duración: 4 semanas

Tema principal: Árboles de Merkle, Hash y Blockchain.

40 puntos

Tareas: 10 puntos

Lenguajes de programación: Python y Go

## XI. INDICADORES DE LOGRO

- Describe conceptos de estructuras de datos
- Analiza adecuadamente los diferentes algoritmos
- Clasifica correctamente los algoritmos según complejidad
- Implementa diferentes TDA
- Encuentra rutas cortas de un grafo
- Utiliza tablas dispersas para indexar

## XII. PRODUCTOS DE APRENDIZAJE

Según el Reglamento General de Evaluación y Promoción del Estudiante de la Universidad de San Carlos de Guatemala, la zona tiene valor de 75 puntos, la nota mínima de promoción es de 61 puntos y la zona mínima para optar a examen final es de 36 puntos.

3 parciales	45 puntos
Tareas, prácticas, otras actividades	05 puntos
Laboratorio	25 puntos
	-----
Zona	75 puntos
Examen final	25 puntos
	-----
Nota final	100 puntos

## XIII. CRITERIOS DE EVALUACIÓN

**Conceptual declarativo:** demuestra conceptos teóricos y matemáticos de las estructuras de datos mediante preguntas directas en clase.

**Procedimental:** aplica los conocimientos adquiridos de los TDA mediante hojas de trabajo, ejercicios, trabajo escritos y cursos en línea.

**Actitudinal:** participa activamente en la clase evaluado mediante la observación.

## XIV. RECURSOS DIDÁCTICOS

- Pizarrón
- Marcadores y almohadilla
- Cañonera y presentaciones multimedia
- Libros de texto
- Documentos en Internet

## XV. BIBLIOGRAFÍA Y RECURSOS

### Libros de texto:

- Joyanes Aguilar, Luis (2007). Estructuras de datos en C++, McGraw-Hill, España.
- Mark Allen Weiss (2013). Estructuras de datos en Java, Pearson. España.
- Joyanes, L., Zahonero, I. (2008) Estructura de datos en Java. McGraw Hill.
- Espino, Luis (2020). Estructuras de datos en C++, Guatemala.
- Robert Sedgewick, Kevin Wayne (2011). Algorithms. Pearson Education. Estados Unidos.

### Cursos en línea:

- mycodeschool (YouTube) <https://bit.ly/330Mx3f>
- Data Structures in C++ (Udemy) <https://bit.ly/2BDzTMa>
- Geeksforgeeks (Website) <https://bit.ly/2Df82Cq>
- Data Structures and Algorithms by Google (Udacity) <https://bit.ly/2DjQKUA>
- Data Structures and Algorithms (Coursera) <https://bit.ly/3gbFf09>

### Cursos tutoriales de Python y Java:

- <https://www.sololearn.com/learning/1073>
- <https://tour.golang.org/welcome/1>