



**ESTRUCTURAS DE DATOS**  
**Primer Semestre 2023**

**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		EDGAR RENE ORNELIS HOIL
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
  - 1.3. Recursividad y programación dinámica
  - 1.4. Matrices dispersas
    - 1.4.1. Estáticas
    - 1.4.2. 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

### UNIDAD 3: OTRAS ESTRUCTURAS Y ALGORITMOS (8 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. Tablas Hash
    - 3.1.1. Funciones Hash
    - 3.1.2. Hash cerrado, direccionamiento abierto
    - 3.1.3. Hash abierto, direccionamiento cerrado
    - 3.1.4. Resolución de colisiones
    - 3.1.5. Rehashing
  - 3.2. Grafos
    - 3.2.1. Definición
    - 3.2.2. Representación por matrices y listas de adyacencia
    - 3.2.3. Recorridos por anchura y profundidad
    - 3.2.4. Algoritmos de árboles de recubrimiento mínimo
    - 3.2.5. Algoritmos de rutas cortas y búsquedas
  - 3.3. Algoritmos de codificación, detección y corrección de errores
    - 3.3.1. ASCII y UTF8
    - 3.3.2. Bit de paridad, Código de Hamming y Checksum de Internet
  - 3.4. Algoritmos de compresión
    - 3.4.1. Algoritmo de Huffman y Algoritmo LZW

**UNIDAD 4: SEGURIDAD Y CALIDAD EN EL SOFTWARE  
(8 PERIODOS)**

**Competencia específica:** analiza aspectos de seguridad en el software y plantea el uso de patrones de seguridad; y conoce los algoritmos más importantes de cifrado para aplicarlos en la seguridad del software.

4. Seguridad el software y patrones
  - 4.1. Seguridad y factor humano
  - 4.2. Políticas para la ingeniería de la Seguridad
    - 4.2.1. Políticas de Seguridad
    - 4.2.2. Criterios de evaluación
    - 4.2.3. Árboles de representación
    - 4.2.4. Métodos formales
    - 4.2.5. Aproximaciones Semiformales
  - 4.3. Patrones de Seguridad
    - 4.3.1. Plantilla para patrones de seguridad.
  - 4.4. Blockchain
    - 4.4.1. Historia y funcionamiento
    - 4.4.2. Tipos de blockchain
    - 4.4.3. Seguridad en operaciones
  - 4.5. Criptografía
    - 4.5.1. Definición e historia
    - 4.5.2. Cifrado simétrico y asimétrico
    - 4.5.3. Cifrado por bloques y por flujo
    - 4.5.4. Estándares en la actualidad
  - 4.6. Calidad del software
    - 4.6.1. Definición
    - 4.6.2. Medidas
    - 4.6.3. Tipos de pruebas

**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

#### **UNIDAD 4:**

- Aplicación de la seguridad en el software
- Utilización de patrones de seguridad
- 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: 3 semanas

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

20 puntos

#### Fase 2

Duración: 4 semanas

Tema principal: Matrices dispersas, árboles AVL y B.

30 puntos

#### Fase 3

Duración: 4 semanas

Tema principal: Grafos, criptografía y compresión.

40 puntos

#### Tareas

10 puntos

Lenguajes de programación: Python y/o Java y/o C++ y/o 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
- Entiende los conceptos de compresión, cifrado y blockchain

## 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. Cualquier actividad que esté involucrada en plagio tendrá un valor de 0 puntos y será notificada a la Escuela de Ciencias y Sistemas.

3 parciales	45 puntos
Tareas, prácticas, otras actividades	06 puntos
Laboratorio	24 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 tutoriales de C++, Python, Go y Java:

- <https://www.sololearn.com/learning/1073>
- <https://www.sololearn.com/learning/1051>
- <https://tour.golang.org/welcome/1>