



**ESTRUCTURA DE DATOS**  
**Primer semestre de 2025**

**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 OBRAYAN 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, haciendo uso de la metodología Orientada a objetos.

#### IV. COMPETENCIA GENERAL

Al finalizar el curso, se espera que el estudiante sea capaz de:

- desarrollar, independiente del lenguaje de programación, los algoritmos para manipular las estructuras de datos más utilizadas.
- Decidir las estructuras de datos más convenientes a utilizar, dado un problema específico, basado en el conocimiento de dichas estructuras y las necesidades de la solución.

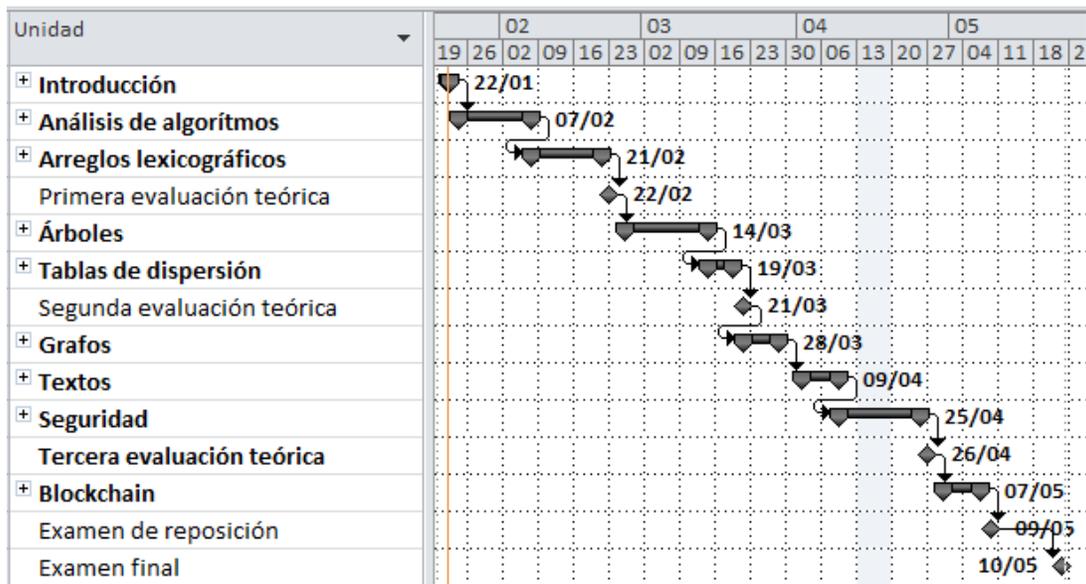
#### V. METODOLOGÍA

- Método: Constructivista.
- Técnicas: autodidáctica y práctica.
- Instrumentos: guías de contenido y prácticas en clase.
- Las clases magistrales se impartirán en 4 períodos semanales.
- El laboratorio se impartirá 2 períodos semanales.
- Durante el curso, se asignan 1 proyectos de programación, a realizarse de manera individual; así como tareas, ejercicios, prácticas e investigaciones.

Se utilizará la combinación de autoestudio y clases virtuales así:

- Las guías de estudio serán publicadas en la plataforma oficial de la Facultad (<https://uedi.ingenieria.usac.edu.gt/campus/my/>).
- El estudiante deberá revisar las guías e investigar en otras fuentes sobre los temas del curso.
- Se tendrán clases virtuales, según el horario semanal especificado, en las salas virtuales asignadas en el portal de la Facultad y solo podrán ingresar los que están asignados oficialmente.
- Las clases virtuales serán para:
  - Resolver dudas de los temas estudiados en la guía publicados en la plataforma
  - Solucionar los problemas propuestos.
  - Realizar evaluaciones prácticas.
- Se tendrá comunicación y asesoría permanente a través de las herramientas que provee la plataforma: chat, correo electrónico y videoconferencias.
- Se establece el compromiso de responder las dudas en un máximo de 1 día hábil (no se incluye fines de semana y asuetos)

## VI. CALENDARIZACIÓN



## VII. CONTENIDO DECLARATIVO

### UNIDAD 1: INTRODUCCIÓN A LA MEMORIA Y ANÁLISIS DE ALGORITMOS

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

1. Análisis de algoritmos
  - 1.1. Algoritmos y eficiencia
  - 1.2. Análisis de casos
  - 1.3. Notación O grande
  - 1.4. 1.4. Desarrollo de pruebas
2. Arreglos
  - 2.1. Arreglos lexicográficos
  - 2.2. Row-major y Col-major
  - 2.3. Matrices dispersas
    - 2.3.1. Estáticas
    - 2.3.2. Dinámicas

## UNIDAD 2: CONCEPTOS AVANZADOS Y ESTRUCTURAS NO LINEALES BÁSICAS

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

3. Árboles
  - 3.1. Definición
  - 3.2. Árboles binarios de búsqueda
  - 3.3. Recorridos
  - 3.4. AVL y HB[k]
  - 3.5. Árboles B
  - 3.6. Variantes de árboles B
  - 3.7. Otros árboles
    - 3.7.1. Rojo negro
    - 3.7.2. Tie
    - 3.7.3. Árboles AST
    - 3.7.4. Árboles de Merkle (Hash Tree)
4. Tablas Hash
  - 4.1. Función Hash
  - 4.2. Resolución de colisiones
  - 4.3. Redimensionamiento

## UNIDAD 3: OTRAS ESTRUCTURAS Y ALGORITMOS

**Competencia específica:** diseña otras estructuras de datos no lineales para solución de problemas y conoce conceptos de grafos, tablas hash, compresión, cifrado y cadenas de bloques.

5. Grafos
  - 5.1. Definición
  - 5.2. Representación por matrices y listas de adyacencia
  - 5.3. Algoritmos de árboles de recubrimiento mínimo
  - 5.4. Algoritmos de rutas cortas y búsquedas
6. Textos
  - 6.1. Representaciones
  - 6.2. Algoritmos de codificación, detección y corrección de errores
    - 6.2.1. ASCII y UTF8
    - 6.2.2. Bit de paridad, Código de Hamming y Checksum de Internet
  - 6.3. Búsqueda de patrones
  - 6.4. Algoritmos de compresión
    - 6.4.1. Algoritmo de Huffman
    - 6.4.2. Algoritmo LZW

## UNIDAD 4: SEGURIDAD Y CALIDAD EN EL SOFTWARE

**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.

1. Seguridad el software y patrones
  - 1.1. Seguridad y factor humano
  - 1.2. Criptografía
    - 1.2.1. Definición e historia
    - 1.2.2. Cifrado simétrico y asimétrico
    - 1.2.3. Cifrado por bloques y por flujo
    - 1.2.4. Estándares en la actualidad
  - 1.3. Políticas para la ingeniería de la Seguridad
    - 1.3.1. Políticas de Seguridad
    - 1.3.2. Criterios de evaluación
    - 1.3.3. Árboles de representación
    - 1.3.4. Métodos formales
    - 1.3.5. Aproximaciones Semiformales
  - 1.4. Patrones de Seguridad
    - 1.4.1. Plantilla para patrones de seguridad.
  - 1.5. Blockchain
    - 1.5.1. Historia y funcionamiento
    - 1.5.2. Tipos de blockchain
    - 1.5.3. Seguridad en operaciones
    - 1.5.4. Descentralización
    - 1.5.5. Prueba de trabajo
    - 1.5.6. Algoritmos de consenso
    - 1.5.7. Smart Contracts
  - 1.6. Calidad del software
    - 1.6.1. Definición
    - 1.6.2. Medidas
    - 1.6.3. Tipos de pruebas

### VIII. 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 hash para indexar

- Entiende los conceptos de compresión, cifrado y blockchain

## IX. EVALUACIÓN

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.

Evaluación de 3 unidades                      45 puntos

- Evaluación de primera unidad
  - Evaluación práctica 1.1      4
  - Evaluación práctica 2.1      4
  - Evaluación práctica 2.2      4
  - Evaluación teórica 1          3
  - Total                              15
- Evaluación de segunda unidad
  - Evaluación práctica 3.1      4
  - Evaluación práctica 3.2      4
  - Evaluación práctica 3.3      4
  - Evaluación teórica 1          3
  - Total                              15
- Evaluación de tercera unidad
  - Evaluación práctica 4.1      4
  - Evaluación práctica 5.1      4
  - Evaluación práctica 6.1      4
  - Evaluación teórica 3          3
  - Total                              15

Prácticas de clase                              06 puntos

Laboratorio                                      24 puntos

----- Zona                                  75 puntos

Examen final                                    25 puntos

Nota final                                        100 puntos

## X. 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 escrito y cursos en línea.

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

## XI. RECURSOS DIDÁCTICOS

- Presentaciones
- Pizarrón electrónico
- Recursos multimedia
- Libros de texto
- Documentos en Internet

## XII. BIBLIOGRAFÍA Y RECURSOS

### Libros de texto:

- Deitel, Harvey M, et.al. (2009) Cómo programar en C++, Pearson, México.
- 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:

- <https://www.sololearn.com/es/learn/courses/c-plus-plus-introduction>
- <https://www.sololearn.com/es/learn/courses/c-plus-plus-intermediate>
- Cplusplus.com
- 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>

