



## ESTRUCTURA DE DATOS Segundo 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	40	7:10	8:50			X			X	ALVARO OBRAYAN HERNÁNDEZ GARCÍA
B	MEET	21	7:10	8:50			X		X		JESÚS ALBERTO GUZMÁN POLANCO
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.



## 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
  - 1.1. Estructuras de datos básicas
    - 1.1.1. Tipos de datos
    - 1.1.2. Estructuras de datos lineales
2. Uso de memoria y análisis de algoritmos
  - 2.1. Administración de memoria
    - 2.1.1. Memoria estática y dinámica
    - 2.1.2. Áreas de memoria Stack y Heap
    - 2.1.3. Row-major y Col-major
  - 2.2. Análisis de algoritmos
    - 2.2.1. Algoritmos y eficiencia
    - 2.2.2. Análisis de casos
    - 2.2.3. Notación O grande
    - 2.2.4. Desarrollo de pruebas
  - 2.3. Recursividad y programación dinámica
    - 2.3.1. Concepto básicos para estructuras dinámicas
    - 2.3.2. Aplicaciones de recursividad
    - 2.3.3. Principales algoritmos recursivos
  - 2.4. Matrices dispersas
    - 2.4.1. Estáticas
    - 2.4.2. 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.

3. Estructuras no lineales
  - 3.1. Árboles
    - 3.1.1. Definición
    - 3.1.2. Árboles binarios de búsqueda
    - 3.1.3. Operaciones

- 3.1.4. Recorridos
- 3.2. AVL
  - 3.2.1. Definición
  - 3.2.2. Operaciones
  - 3.2.3. Rotaciones
- 3.3. Árboles B
  - 3.3.1. Definición
  - 3.3.2. Operaciones
- 3.4. Variantes de árboles B
  - 3.4.1. Definición
  - 3.4.2. Operaciones
- 3.5. Otros árboles
  - 3.5.1. Rojo negro
  - 3.5.2. Tie
  - 3.5.3. Árboles AST
  - 3.5.4. Árboles de Merkle (Hash Tree)

### **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, tablas hash, compresión, cifrado y cadenas de bloques.

- 4. Otras estructuras y algoritmos de codificación
  - 4.1. Grafos
    - 4.1.1. Definición
    - 4.1.2. Representación por matrices y listas de adyacencia
    - 4.1.3. Recorridos por anchura y profundidad
    - 4.1.4. Algoritmos de árboles de recubrimiento mínimo
    - 4.1.5. Algoritmos de rutas cortas y búsquedas
  - 4.2. Tablas Hash
    - 4.2.1. Función Hash
    - 4.2.2. Resolución de colisiones
    - 4.2.3. Redimensionamiento
  - 4.3. Compresión
    - 4.3.1. Introducción
    - 4.3.2. Compresión sin pérdida y con pérdida
      - 4.3.2.1. Algoritmo de Huffman
      - 4.3.2.2. Algoritmo LZW

- 4.4. Criptografía y seguridad
  - 4.4.1. Historia y conceptos
  - 4.4.2. Cifrado por bloques y por flujo
  - 4.4.3. Cifrado simétrico
    - 4.4.3.1. DES y AES
  - 4.4.4. Cifrado asimétrico
    - 4.4.4.1. RSA
  - 4.4.5. Funciones hash criptográficas
  - 4.4.6. Checksum
- 4.5. Blockchain
  - 4.5.1. Historia y funcionamiento
  - 4.5.2. Descentralización
  - 4.5.3. Prueba de trabajo
  - 4.5.4. Algoritmos de consenso
  - 4.5.5. Smart Contracts
  - 4.5.6. Tipos de blockchain

## 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 hash
- Conocimiento de compresión
- 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 actividades:

### 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 C++

## 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 hash 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	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 escrito y cursos en línea.

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

## XIV. RECURSOS DIDÁCTICOS

- Presentaciones
- Pizarrón electrónico
- Recursos 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>