



**ESTRUCTURA DE DATOS**  
**Primer Semestre 2020**

**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> 10	<b>Horas de laboratorio a la semana:</b> 10
<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	T-3	209	07:10	08:50			X		X		JESUS GUZMAN
B	T-3	216	17:00	19:00			X				ALVARO OBRAYAN HERNÁNDEZ GARCÍA
B	T-3	410	08:50	10:30						X	ALVARO OBRAYAN HERNÁNDEZ GARCÍA
C	T-3	216	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.

#### IV. COMPETENCIA GENERAL

Que el estudiante desarrolle, independiente del lenguaje de programación, los algoritmos para manipular las estructuras de datos más utilizadas. Decide 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: deductivo
- Técnicas: expositiva y demostrativa
- Instrumentos: guías de trabajo, hojas de trabajo, ejercicios, preguntas, diálogo y observación
- Las clases magistrales se impartirán en 10 períodos semanales
- El laboratorio se impartirá 10 períodos semanales
- Durante el curso, se asignan 2 proyectos de programación, a realizarse de manera individual; así como tareas, ejercicios, prácticas e investigaciones.

#### VI. CALENDARIZACIÓN

Unidad	Enero	Febrero	Marzo	Abril	Mayo
Unidad 1					
Primer parcial					
Lab Práctica 1					
Unidad 2					
Lab Proyecto 1					
Segundo parcial					
Lab Práctica 2					
Unidad 3					
Receso y SS					
Tercer parcial					
Unidad 4					
Lab Proyecto 2					
Examen final					

## VII. CONTENIDO DECLARATIVO

### UNIDAD 1: INTRODUCCIÓN A LAS ESTRUCTURAS DE DATOS (16 PERIODOS)

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

1. Repaso de conceptos (4 periodos)
  - 1.1. Algoritmos, pseudocódigo y código
  - 1.2. Memoria estática y dinámica (stack y heap)
  - 1.3. Tipos primitivos, clases y asociaciones (diagrama de clases)

2. Introducción a las estructuras de datos (4 periodos)
  - 2.1. Arreglos y matrices
  - 2.2. Tipos de datos abstractos
  - 2.3. Recursividad
    - 2.3.1. Definición
    - 2.3.2. Ejemplos complejos
    - 2.3.3. Comparación con lo iterativo

3. Estructuras lineales (8 periodos)
  - 3.1. Listas
    - 3.1.1. Definición y características
    - 3.1.2. Tipos de listas
    - 3.1.3. Operaciones
    - 3.1.4. Implementación TDA (arreglos, punteros y contenedores)
  - 3.2. Pilas
    - 3.2.1. Definición y características
    - 3.2.2. Operaciones
    - 3.2.3. Implementación TDA (arreglos, punteros y contenedores)
  - 3.3. Colas
    - 3.3.1. Definición y características
    - 3.3.2. Operaciones
    - 3.3.3. Implementación TDA (arreglos, punteros y contenedores)
    - 3.3.4. Colas de prioridad
      - 3.3.4.1. Definiciones
      - 3.3.4.2. Ejercicios de aplicación
    - 3.3.5. Ejemplos

**UNIDAD 2: CONCEPTOS AVANZADOS Y ESTRUCTURAS NO LINEALES BÁSICAS  
(18 PERIODOS)**

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

1. Complejidad (2 periodos)
  - 1.1. Algoritmo y eficiencia
  - 1.2. Análisis de casos
  - 1.3. Complejidad de tiempo y espacio
  - 1.4. Notación asintótica
  - 1.5. Notación O grande
2. Búsquedas y ordenamientos (4 periodos)
  - 2.1. Búsquedas
    - 2.1.1. Secuencial
    - 2.1.2. Binaria
  - 2.2. Ordenamientos
    - 2.2.1. Por burbuja
    - 2.2.2. Por selección
    - 2.2.3. Por inserción
    - 2.2.4. Ordenamiento rápido
3. Matrices dispersas (4 periodos)
  - 3.1. Definición
  - 3.2. Matrices dispersas simples
  - 3.3. Matrices dispersas ortogonales
  - 3.4. Matrices dispersas con encabezados
4. Árboles binarios (4 periodos)
  - 4.1. Definiciones
  - 4.2. Árboles binarios de búsqueda
  - 4.3. Operaciones
    - 4.3.1. Inserción
    - 4.3.2. Eliminación
    - 4.3.3. Actualización
  - 4.4. Búsquedas
  - 4.5. Recorridos
5. Árboles balanceados AVL (4 periodos)
  - 5.1. Definición
  - 5.2. Operaciones
    - 5.2.1. Inserción
    - 5.2.2. Eliminación
    - 5.2.3. Actualización
  - 5.3. Búsquedas y recorridos
  - 5.4. Otros árboles
    - 5.4.1. Rojo negro
    - 5.4.2. Tie

### UNIDAD 3: ESTRUCTURAS NO LINEALES COMPLEJAS (10 PERIODOS)

**Competencia específica:** diseña estructuras de datos no lineales para solución de problemas y conoce conceptos de grafos y tablas dispersas.

1. Árboles B (3 periodos)
  - 1.1. Definición
  - 1.2. Operaciones de inserción, eliminación y actualización
  - 1.3. Búsquedas y recorridos
  - 1.4. Árbol B+
2. Otros árboles (3 periodos)
  - 2.1. Árbol de Sintaxis Abstracta (AST)
  - 2.2. Árbol de Merkle y su relación con Blockchain
3. Grafos (2 periodos)
  - 3.1. Definición
  - 3.2. Tipos de implementación
  - 3.3. Recorridos por anchura y profundidad
  - 3.4. Búsquedas por anchura y profundidad
  - 3.5. Algoritmos de rutas cortas
  - 3.6. Algoritmos de árbol recubridor mínimo
4. Tablas dispersas (Hash) (2 periodos)
  - 4.1. Definición
  - 4.2. Colisiones
  - 4.3. Hash cerrado (direccionamiento abierto) y abierto (direccionamiento cerrado)
  - 4.4. Aplicaciones

### UNIDAD 4: TEMAS COMPLEMENTARIOS (2 PERIODOS)

**Competencia específica:** conoce las maneras de codificación de datos, asimismo algoritmos de detección y corrección de errores, compresión, cifrado y cadenas de bloques.

1. Codificación (2 periodos)
  - 1.1. Algoritmos de detección de errores
  - 1.2. Algoritmos de corrección de errores
2. Algoritmos de compresión (2 periodos)
3. Algoritmos de cifrado (2 periodos)

## 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

### UNIDAD 3:

- Construcción de estructuras no lineales complejas para solución de problemas
- Reconocimiento de rutas cortas en un grafo
- Uso de tablas dispersas

### UNIDAD 4:

- Conocimiento de codificación y algoritmos necesarios
- Uso de métodos de cifrado
- Introducción a BlockChain

## 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 los enunciados de los proyectos los redactarán el auxiliar con las siguientes bases:

### **Práctica 1 y Proyecto 1**

Duración: 15 días y 1 Mes respectivamente

Tema principal: implementación de TDA hasta AVL

Lenguaje de programación: C++ (consola)

### **Práctica 2 y Proyecto 2**

Duración: 10 días y 1 Mes respectivamente

Tema principal: implementación de TDA hasta grafos, incluyendo BlockChain para la práctica

Lenguaje de programación: Java (gráfico)

## 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.

Cursos de block chain:

- <https://www.edx.org/course/understanding-blockchain-and-its-implications>
- <https://www.edx.org/course/blockchain-for-business-an-introduction-to-hyperledger-technologies>

Cursos tutoriales de Python y Java:

- <https://www.sololearn.com/Course/CPlusPlus/>
- <https://www.sololearn.com/Course/Java/>