

Sílabo

Malla 2021

UTEC
Universidad
de Ingeniería
y Tecnología





DEPARTAMENTO

Departamento de Computer Science



CURSO

Algoritmos y Estructuras de Datos



MALLA

2021



MODALIDAD

PRESENCIAL



CREDITOS

4



REGLAS INTEGRIDAD ACADÉMICA

Todo estudiante matriculado en una asignatura de la Universidad de Ingeniería y Tecnología tiene la obligación de conocer y cumplir las reglas de integridad académica, cuya lista a continuación es de carácter enunciativo y no limitativo, ya que el/la docente podrá dar mayores indicaciones:

1. La copia y el plagio son dos infracciones de magnitud muy grave en la Universidad de Ingeniería y Tecnología (UTEC) conforme a lo establecido en el Reglamento de Disciplina de los Estudiantes. Tienen una sanción desde 2 semestres de suspensión hasta la expulsión.
2. Si se identifica la copia o plagio en evaluaciones individuales, el/la docente puede proceder a anular la evaluación.
3. Si la evaluación es personal o grupal-individual, la interacción entre equipos o compañeros se considera copia o plagio, según corresponda. Si la evaluación calificada no indica que es grupal, se presume que es individual.
4. La copia, plagio, el engaño y cualquier forma de colaboración no autorizada no serán tolerados y serán tratados de acuerdo con las políticas y reglamentos de la UTEC, implicando consecuencias académicas y sanciones disciplinarias.
5. Aunque se alienta a los estudiantes a discutir las tareas y trabajar juntos para desarrollar una comprensión más profunda de los temas presentados en este curso, no se permite la presentación del trabajo o las ideas de otros como propios. No se permite el plagio de archivos informáticos, códigos, documentos o dibujos.
6. Si el trabajo de dos o más estudiantes es sospechosamente similar, se puede aplicar una sanción académica a todos los estudiantes, sin importar si es el estudiante que proveyó la información o es quien recibió la ayuda indebida. En ese sentido, se recomienda no proveer el desarrollo de sus evaluaciones a otros compañeros ni por motivos de orientación, dado que ello será considerado participación en copia.
7. El uso de teléfonos celulares, aplicaciones que permitan la comunicación o cualquier otro tipo de medios de interacción entre estudiantes está prohibido durante las evaluaciones o exámenes, salvo que el/la docente indique lo contrario de manera expresa. Es irrelevante la razón del uso del dispositivo.
8. En caso exista algún problema de internet durante la evaluación, comunicarse con el/la docente utilizando el protocolo establecido. No comunicarse con los compañeros dado que eso generará una presunción de copia.
9. Se prohíbe tomar prestadas calculadoras o cualquier tipo de material de otro estudiante durante una evaluación, salvo que el/la docente indique lo contrario.
10. Si el/la docente encuentra indicios de obtención indebida de información, lo que también implica no cumplir con las reglas de la evaluación, tiene la potestad de anular la prueba, advertir al estudiante y citarlo con su Director de Carrera. Si el estudiante no asiste a la citación, podrá ser reportado para proceder con el respectivo procedimiento disciplinario. Una segunda advertencia será reportada para el inicio del procedimiento disciplinario correspondiente.
11. Se recomienda al estudiante estar atento/a a los datos de su evaluación. La consignación de datos que no correspondan a su evaluación será considerado indicio concluyente de copia.



UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

SÍLABO DEL CURSO

1. ASIGNATURA

CS2023 – Algoritmos y Estructuras de Datos

2. DATOS GENERALES

2.1 Ciclo: NIVEL 4

2.2 Créditos: 4

2.3 Condición: Obligatorio para Ciencia de la Computación

2.4 Idioma de dictado: Español

2.5 Requisitos: CS2013 - Programación III, CS1112 - Programación II

3. INTRODUCCIÓN AL CURSO

El fundamento teórico de todas las ramas de la computación descansa sobre los algoritmos y estructuras de datos. Este curso de naturaleza teórica y práctica, aborda el diseño, análisis e implementación de estructuras de datos como listas, colas, pilas, árboles, grafos, y técnicas de optimización. Los estudiantes aprenderán a analizar problemas, diseñar soluciones colaborativas, e implementar estructuras de datos clave para desarrollar competencias esenciales en computación.

4. OBJETIVOS

- Sesión 1: Explicar los conceptos básicos de los algoritmos y estructuras de datos y dar ejemplos de su aplicación en diversos problemas.
- Sesión 2: Introducir el concepto de listas a partir de un problema, y mostrar ejemplos de sus operaciones.
- Sesión 3: Describir los distintos tipos de listas y sus particularidades en las implementaciones. Explicar recorrido utilizando iteradores y el uso de type traits.
- Sesión 4: Explicar el uso de diccionarios y su representación estructural. Identificar los diversos tipos de hashing y su relación con las tablas hash. Mostrar ejemplos de su aplicación en el lenguaje C++ (i.e. unordered maps).
- Sesión 5: Describir el tipo de dato abstracto árbol, y sus subtipos como árboles binarios, heaps y disjoint sets.
- Sesión 6 : Mostrar múltiples problemas donde se pueda aplicar los diversos métodos de los árboles.
- Sesión 7: Mostrar los problemas del árbol binario de búsqueda para introducir los árboles autobalanceados (e.g. AVL, B Tree, B+ Tree). Analizar las diferencias entre los árboles autobalanceados, y sus diferentes usos (e.g. base de datos, diccionarios).



- Sesión 8: Explicar el problema de String Matching y sus diferentes algoritmos de resolución. Describir el funcionamiento de los Tries para diversos problemas de representación e indexación de strings.
- Sesión 9: Explicar la teoría básica de grafos y describir las diversas formas de representarlos. Mostrar las múltiples aplicaciones de búsqueda en grafos.
- Sesión 10: Describir el uso e implementaciones de matrices esparza y su relación con grafos.
- Sesión 11: Introducción a técnicas de optimización para resolución de problemas (i.e. Backtracking, Hill Climbing).

5. COMPETENCIAS Y CRITERIOS DE DESEMPEÑO

Competencias Específicas ABET - COMPUTACION

- Analizar un problema computacional complejo y aplicar principios de computación y otras disciplinas relevantes para identificar soluciones.
- Diseñar, implementar y evaluar una solución computacional para satisfacer un conjunto determinado de requerimientos computacionales en el contexto de la disciplina del programa.

Competencias Generales ABET - COMPUTACION

- Funcionar efectivamente como miembro o líder de un equipo comprometido en actividades apropiadas a la disciplina del programa.

6. RESULTADOS DE APRENDIZAJE

- Analizar problemas e identificar las estructuras de datos y algoritmos apropiados para su solución.
- Implementar estructuras de datos como listas, colas, pilas, heaps, hashing, árboles, conjuntos y grafos
- Evaluar técnicas existentes de estructuras de datos para resolver problemas específicos.
- Desarrollar soluciones computacionales de modo colaborativo usando herramientas adecuadas.

7. TEMAS

1. Listas, Colas y Pilas

- 1.1. Listas
 - 1.1.1. Lista simplemente enlazada
 - 1.1.2. Lista doblemente enlazada
 - 1.1.3. Lista circular
 - 1.1.4. Type traits
 - 1.1.5. Vector (ArrayList)
 - 1.1.6. Iteradores
- 1.2. Colas



1.3. Pilas

2. Tablas Hash

- 2.1. TDA Diccionario
- 2.2. Función Hash y Tabla Hash
- 2.3. Manejo de colisiones

3. Árboles

- 3.1. Árboles binarios
 - 3.1.1. Sets y maps
 - 3.1.2. Iteradores
- 3.2. Heaps binarios
 - 3.2.1. Min heap
 - 3.2.2. Max heap
- 3.3. Disjoint sets

4. Árboles Equilibrados

- 4.1. Definiciones
- 4.2. Árboles AVL
- 4.3. Árboles B
- 4.4. Árboles B+

5. Indexación de strings

- 5.1. String matching
- 5.2. Suffix Tree
- 5.3. Tries

6. Grafos

- 6.1. Definiciones
- 6.2. Implementación de Grafos usando Matrices de Adyacencia
- 6.3. Implementación de Grafos usando Listas de Adyacencia
- 6.4. Inserción, Búsqueda y Eliminación de nodos y aristas
- 6.5. Árbol de mínima expansión
- 6.6. Algoritmos de búsqueda en grafos

7. Matrices Esparzas

- 7.1. Definiciones
- 7.2. Creación de la matriz esparza estática y Dinámicas
- 7.3. Métodos de inserción, búsqueda y eliminación

8. Introducción a Optimización

- 8.1. Backtracking



8.2. Hill Climbing

8. PLAN DE TRABAJO

8.1 Metodología

Este curso presenta por metodología activa el aprendizaje clásico y el aprendizaje basado en problemas; ambos son fundamentales para introducir al estudiante a los conceptos básicos y afianzar la base necesaria para los siguientes cursos de carrera.

Ambos aumentan el interés del estudiante y promueven su compromiso en el aprendizaje.

8.2 Sesiones de teoría

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizan actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos. Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

8.3 Sesiones de práctica (laboratorio o taller)

Para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, se aplica una metodología basada en retos. Los alumnos realizarán actividades que les permitan aplicar los conocimientos adquiridos durante las sesiones de teoría y se les propondrán retos que permitan evaluar su desempeño.

9. SISTEMA DE EVALUACIÓN

El curso consta de los siguientes espacios de evaluación:

Evaluación	Teoría
	<p>TEORÍA 40%</p> <ul style="list-style-type: none">●Examen E1 (20%)●Examen E2 (20%) <p>PRÁCTICA O LABORATORIO 60%</p> <ul style="list-style-type: none">●Evaluación Continua C1(20%)●Evaluación Continua C2 (20%)●Proyecto P1 (10%)●Proyecto P2 (10%) <p>Nota</p> <p>:La ponderación de la evaluación se hará si ambas partes están aprobadas</p> <p>.</p>



	100%

10. REFERENCIAS BIBLIOGRÁFICAS

1. José Fager et al. Estructura de datos. First Edition. Iniciativa Latinoamericana de Libros de Texto Abiertos (LATIN), 2014.
2. R. Sedgewick, Algorithms in C (parts 1-4), 3rd. edition, Addison-Wesley/Longman, 1998.
3. Sedgewick, R., & Wayne, K. (2011). Algorithms. Addison-wesley professional.
4. Thomas H. Cormen et al. Introduction to Algorithms. Third Edition. ISBN: 978-0-262-53305-8. MIT Press, 2009.

