

Universidad CENFOTEC



Escuela de Software

Código del curso: BISOFT-11.

Nombre del curso: Estructuras de Datos 1.

Sección: SCV1.

Periodo: C3-2025.

Docente facilitador: Romario Salas Cerdas.

CONSIGNA DE LA PRIMERA PRÁCTICA

1. Datos generales de la actividad

Tipo de actividad:	Práctica.		
Fecha de entrega:	28 de setiembre del 2025, 11:59pm.	Valor porcentual:	10%.
Formato de entrega:	Archivo comprimido.	Puntaje total:	100.
Individual: Sí.	Grupal: No.		

2. Instrucciones generales

1. Lea cuidadosamente las instrucciones de la actividad. En caso de tener alguna duda, puede consultar con su docente.
2. Esta actividad se desarrolla de manera individual. Cualquier intento de plagio será sancionado de acuerdo con el reglamento académico vigente.
3. Al completar la actividad, el estudiante debe subir un archivo comprimido con los archivos de código fuente y la documentación solicitada a la plataforma Moodle en el tiempo y espacio indicado por el docente.

3. Objetivos o competencias del curso que se evaluarán en la actividad de aprendizaje

Objetivo general del curso	Construir aplicaciones de software pequeñas y medianas, usando las estructuras de datos lineales, las estructuras de datos jerárquicas y los algoritmos de ordenamiento y búsqueda, para implementar aplicaciones de software más eficientes en tiempo y en recursos de computadora.
Objetivos específicos o	<ul style="list-style-type: none"> • Comprender el funcionamiento y la algorítmica de

<p>resultados de aprendizaje que se evalúan</p>	<p>las estructuras de datos lineales y jerárquicas y de los algoritmos de ordenamiento y búsqueda, mediante el estudio de ejemplos y prácticas en clase, a fin de que el estudiante pueda conocer y distinguir las diferentes estructuras de datos.</p> <ul style="list-style-type: none"> • Usar las estructuras de datos lineales y jerárquicas, y los algoritmos de ordenamiento y búsqueda, mediante el uso de Tipos Abstractos de Datos y de lenguajes de programación, para resolver un problema con una aplicación de software pequeña o mediana. • Analizar la eficiencia en tiempo y en espacio, de las estructuras de datos y de los algoritmos, usando la notación O, a fin de aplicar este criterio y escoger las estructuras y los algoritmos de menor complejidad. • Diseñar estructuras de datos lineales y jerárquicas por medio de Tipos Abstractos de Datos, para construir aplicaciones de software fáciles de entender, corregir y cambiar. • Elaborar aplicaciones de software eficientes, implementando estructuras de datos lineales y jerárquicas, y algoritmos de búsqueda y ordenamiento, que garanticen el menor uso de recursos y de tiempo.
--	---

4. Descripción de la actividad

Instrucciones:

El objetivo de esta práctica es que el estudiante se ejercite en la definición de ADTs, así como en su especificación informal y formal. Para realizar lo anterior, el estudiante debe seleccionar uno de los ADTs geométricos listados debajo, así como uno de los ADTs convencionales mostrados seguidamente, y realizar la documentación de su definición y especificación completa de acuerdo con los principios estudiados en clase.

ADTs geométricos:

- **rectangulo** (debe considerarse la posibilidad de que el **rectangulo** se encuentre rotado respecto a los ejes del plano cartesiano)
- **triangulo** (debe contemplarse la posibilidad de que el **triangulo** sea de cualquiera de los tres tipos básicos: equilátero, isósceles o escaleno; además de la posibilidad de que los datos ingresados no sean válidos para la construcción de un **triangulo**)

ADTs convencionales:

- **correo_electronico**
- **fecha_hora**

La especificación de los dos ADTs seleccionados deben incluir como mínimo las operaciones para la construcción y modificación de las instancias de los mismos, así como al menos cuatro operaciones más para cada uno de ellos. La elección de las operaciones queda a criterio del estudiante, pero debe ser adecuada para el ADT en cuestión (i.e., deben ser operaciones que razonablemente se pueda esperar que una instancia de dicho concepto lleve a cabo).

Además de la especificación anteriormente descrita, la cual se debe entregar en formato .pdf, el estudiante debe desarrollar la implementación en C++ de ambos ADTs, presentando como mínimo una pareja de archivos de código fuente para cada ADT (i.e., el archivo de encabezado .h en el que se desarrolla la implementación estructural del ADT y el archivo de código fuente .cpp en el que se implementan su interfaz operativa pública). Además de los dos archivos .h y los dos archivos .cpp requeridos para la implementación de los dos ADTs, el estudiante debe presentar un archivo .cpp adicional que incluya una rutina **main()**, dentro de la cual debe instanciar uno o dos objetos de la clase definida por cada ADT implementado, según sea la necesidad en cada caso, utilizándolos para probar las operaciones implementadas en cada uno de ellos.

5. Rúbrica

Esta actividad de aprendizaje será evaluada mediante la siguiente rúbrica:

1. Documentación completa y adecuada de la especificación informal y formal de ambos ADTs: **40pts.**
 2. Implementación completa y correcta de ambos ADTs: **40 puntos.**
 3. Evidencia del funcionamiento de la implementación de ambos ADTs: **20 puntos.**
- Total: **100 puntos.**

Criterio	Deficiente (1 punto)	Regular (2 puntos)	Bueno (3 puntos)	Excelente (4 puntos)
Requerimientos de la práctica	Analiza los requerimientos presentados en la práctica de manera inadecuada	Analiza los requerimientos presentados en la práctica de manera básica	Analiza los requerimientos presentados en la práctica de manera aceptable	Analiza los requerimientos presentados en la práctica de manera adecuada
Estructuras de datos	Selecciona las estructuras de datos que se utilizan en el desarrollo de la práctica de manera inadecuada y con múltiples errores	Selecciona las estructuras de datos que se utilizan en el desarrollo de la práctica de manera básica y con varios errores	Selecciona las estructuras de datos que se utilizan en el desarrollo de la práctica de manera regular y con algún error	Selecciona las estructuras de datos que se utilizan en el desarrollo de la práctica de manera adecuada y sin errores
Codificación de las funcionalidades	Desarrolla la codificación de las funcionalidad que permiten la solución del problema de manera inadecuada y con múltiples errores	Desarrolla la codificación de las funcionalidad que permiten la solución del problema de manera elemental y con varios errores	Desarrolla la codificación de las funcionalidad que permiten la solución del problema de manera aceptable y con algún error	Desarrolla la codificación de las funcionalidad que permiten la solución del problema de manera adecuada y sin errores
Análisis de eficiencia	Emplea casos de prueba para la comprobación de la funcionalidad y eficiencia de los algoritmos de manera inadecuada y con múltiples errores	Emplea casos de prueba para la comprobación de la funcionalidad y eficiencia de los algoritmos de manera elemental y con varios errores	Emplea casos de prueba para la comprobación de la funcionalidad y eficiencia de los algoritmos de manera aceptables y con algún error	Emplea casos de prueba para la comprobación de la funcionalidad y eficiencia de los algoritmos de manera adecuada y sin errores
Entrega del informe de la solución de la práctica	Realiza la entrega de las secciones de la práctica de forma inadecuada			Realiza la entrega de la práctica con todas las secciones de forma adecuada