

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 TERCERA PRÁCTICA

1. Datos generales de la actividad

| | | | |
|----------------------------|------------------------------------|--------------------------|------|
| Tipo de actividad: | Práctica. | | |
| Fecha de entrega: | 30 de noviembre 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 al buzón del entorno de Moodle designado para ese fin. El archivo comprimido debe contener el archivo de código fuente **.cpp** necesario para la compilación y posterior ejecución del programa solicitado, además de un archivo **.pdf** con el desarrollo de la documentación solicitada.

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

| | |
|---|--|
| Objetivo general o competencia 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. |
|---|--|

| | |
|--|--|
| <p>Objetivos específicos que se evalúan</p> | <ul style="list-style-type: none"> • Comprender el funcionamiento y la algorítmica de 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. • 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

Contexto:

La tercera práctica consiste en el desarrollo de una investigación en torno a la aplicación de algoritmos de ordenamiento en el software de sistema, acompañada por la correspondiente implementación de los algoritmos de ordenamiento en C++. El estudiante debe elegir un algoritmo de la siguiente lista, y con base en su elección, desarrollar lo solicitado en el apartado de instrucciones:

- Ordenamiento radix.
- Ordenamiento rápido.
- Ordenamiento shell.

Instrucciones:

Según la elección realizada, el estudiante debe desarrollar y documentar una investigación breve (dos páginas en Arial 12 sin contar las imágenes, interlineado de 1.5) con los siguientes contenidos:

- Explicación general del funcionamiento del algoritmo.
- Mención de algunos escenarios o casos de uso en los que el algoritmo destaca.
- Comentario sobre las ventajas y desventajas del algoritmo, haciendo referencia a otros algoritmos clásicos de ordenamiento para efectos de comparación y contraste.
- Explicación breve de la complejidad temporal del algoritmo.
- Representación gráfica de la operación del algoritmo.

Seguidamente, el estudiante debe desarrollar un programa en C++ que permita al usuario, mediante un **menu()** de consola amigable y efectivo, llenar una estructura de datos (puede ser un arreglo simple) y ordenarla mediante la ejecución sobre ella del algoritmo seleccionado. La rutina que representa al algoritmo de ordenamiento en sí debe incluir comentarios que expliquen adecuadamente los diversos pasos que llevan al resultado buscado.

5. Rúbrica

Esta actividad de aprendizaje será evaluada mediante la siguiente rúbrica:

1. Implementación adecuada del algoritmo de ordenamiento: **50 puntos**.
2. Desarrollo adecuado de la UI requerida: **15 puntos**.
3. Desarrollo adecuado de la documentación solicitada: **35 puntos**.

Total: **100 puntos**.

| Criterio | Deficiente (1 punto) | Regular (2 puntos) | Bueno (3 puntos) | Excelente (4 puntos) |
|-----------------------------------|--|--|---|--|
| Requerimientos y especificaciones | Describe los requerimientos y especificaciones del estudio de caso de forma insuficiente y sin profundidad | Describe los requerimientos y especificaciones del estudio de caso de forma elemental y con poca profundidad | Describe los requerimientos y especificaciones del estudio de caso de forma aceptable y con profundidad | Describe los requerimientos y especificaciones del estudio de caso de forma adecuada y con profundidad |

| | | | | |
|--|--|--|--|---|
| Estructuras de datos | Analiza las estructuras de datos situación planteada de manera inadecuada y con limitaciones | Analiza las estructuras de datos situación planteada de manera básica y con varias limitaciones | Analiza las estructuras de datos situación planteada de manera regular y con alguna limitación | Analiza las estructuras de datos situación planteada de manera adecuada y sin limitaciones |
| Funcionamiento de estructuras de datos | Propone el funcionamiento de las estructuras de datos de manera apropiada | Propone el funcionamiento de las estructuras de datos de manera elemental | Propone el funcionamiento de las estructuras de datos de manera regular | Propone el funcionamiento de las estructuras de datos de manera apropiada |
| Estrategia de solución | Desarrolla una estrategia de solución al caso propuesto de manera inapropiada y no lo fundamenta con los tipos abstractos de datos | Desarrolla una estrategia de solución al caso propuesto de manera básica y hay poca fundamentación de los tipos abstractos de datos | Desarrolla una estrategia de solución al caso propuesto de manera aceptable y lo fundamenta con los tipos abstractos de datos | Desarrolla una estrategia de solución al caso propuesto de manera apropiada y lo fundamenta con los tipos abstractos de datos |
| Informe de la solución | Elabora un informe de la solución del estudio de caso con las pautas y líneas a seguir de manera escasa, con múltiples errores y sin respetar el mínimo de páginas | Elabora un informe de la solución del estudio de caso con las pautas y líneas a seguir de manera elemental, con varios errores y sin respetar el mínimo de páginas | Elabora un informe de la solución del estudio de caso con las pautas y líneas a seguir de manera regular, con algún error y respeta el mínimo de páginas | Elabora un informe de la solución del estudio de caso con las pautas y líneas a seguir de manera adecuada, sin errores y respeta el mínimo de páginas |
| Entrega de la solución | Realiza la entrega del estudio de caso de manera inadecuada | | | Realiza la entrega del estudio de caso de manera adecuada |