

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 DEL SEGUNDO AVANCE DEL PROYECTO DE PROGRAMACIÓN

1. Datos generales de la actividad

Tipo de actividad:	Avance de proyecto.		
Fecha de entrega:	26 de octubre del 2025, 11:59pm.	Valor porcentual:	10%.
Formato de entrega:	Archivo comprimido.	Puntaje total:	100.
Individual: No.	Grupal: Sí.		

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 grupal. Cualquier intento de plagio será sancionado de acuerdo con el reglamento académico vigente.
3. Al completar la actividad, un representante del grupo de trabajo debe subir un archivo comprimido al buzón del entorno de Moodle designado para ese fin. El archivo comprimido debe contener los archivos de código fuente **.cpp** y **.h** necesarios para la compilación y posterior ejecución del programa solicitado.

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:

El proyecto de programación consiste en el desarrollo de una aplicación para la gestión de un sistema de ventas de productos en línea. Para satisfacer dicha finalidad, el sistema por desarrollar debe implementar y utilizar estructuras de datos dinámicas para el control del inventario de productos y de los pedidos realizados. En los diferentes avances del proyecto, se realizarán añadiduras y cambios graduales al código, con la finalidad de lograr un funcionamiento cada vez más eficiente del sistema elaborado.

Instrucciones:

Para el segundo avance, los grupos deben tomar como punto de partida el programa presentado en el primer avance del proyecto y sumarle funcionalidades de ordenamiento y búsqueda a la **ListaProductos** que cada **Cliente** que haya en el sistema de ventas de

productos en línea tiene. El ordenamiento de **ListaProductos** debe implementarse mediante los algoritmos de selección y mezcla, y debe ser posible ordenar la lista tanto ascendente como descendientemente, tomando como criterio de ordenamiento cualquiera de los atributos correspondientes al ADT **Producto**. Los métodos asociados con el ordenamiento de la lista deben incluir la impresión en pantalla de la cantidad total de comparaciones que su ejecución requirió. La búsqueda debe implementarse de forma lineal y binaria para cualquiera de esos mismos atributos; en el caso de la búsqueda binaria, el programa debe asegurarse primero de que la lista esté ordenada de acuerdo con ese atributo en particular. El programa debe gestionar la conversión de la lista enlazada simple **ListaProductos** a una estructura adecuada para las tareas de ordenamiento y búsqueda, lo cual debe implementarse en un método independiente que forme parte de las operaciones del **Cliente**.

El ADT **Cliente** debe agregarse al programa a partir de este segundo avance, así como su correspondiente estructura de colección **ColaClientes**, la cual debe ser una cola de prioridad que funcione con base en una calificación numérica ascendente de **prioridad: 1** para clientes ordinarios, **2** para clientes regulares, **3** para adultos mayores, personas discapacitadas y mujeres embarazadas. Además de su **nombre**, **cedula** y **prioridad**, cada **Cliente** tiene como atributo su lista personal de **Productos**, la cual representa su **carritoCompras**.

El programa debe incluir una rutina **menu()** que coordine la operación de las partes solicitadas, así como una rutina **main()** dentro de la cual se invoque dicho menú. El programa debe ser capaz de gestionar la inserción y la eliminación de **Clientes** de la cola, así como del ordenamiento de los **Productos** de la lista de un **Cliente** en particular, la búsqueda de un cierto **Producto** en la lista de un **Cliente** particular, y la impresión en pantalla de dicha lista. La inserción de un **Cliente** requiere siempre el llenado de su **ListaProductos**, basándose en lo que esté disponible en la **ListaProductos** que representa el **catalogo** de la tienda; a dicho **catalogo** también se puede agregar nuevos **Productos** desde el **menu()** del programa. La eliminación de un **Cliente** implica la generación de un reporte de su **ListaProductos**, con el monto correspondiente a cada uno de los **Productos** que incluye y al total de éstos. Es importante no olvidar la liberación de la memoria dinámica reservada por la estructura de datos durante su utilización, por lo que la opción del menú que represente la salida del mismo debe encargarse de dicha necesidad.

5. Rúbrica

Esta actividad de aprendizaje será evaluada mediante la siguiente rúbrica:

1. Implementación adecuada de los ADTs **Cliente** y **ColaClientes**: **30 puntos**.

2. Implementación adecuada de los métodos de ordenamiento y búsqueda en el ADT

ListaProductos: 50 puntos.

3. Desarrollo adecuado de la UI requerida: **20 puntos.**

Total: 100 puntos.

Criterio	Deficiente (1 punto)	Regular (2 puntos)	Bueno (3 puntos)	Excelente (4 puntos)
Requerimientos y especificaciones	Indaga sobre los requerimientos y especificaciones del problema de manera inadecuada	Indaga sobre los requerimientos y especificaciones del problema de manera incipiente	Indaga sobre los requerimientos y especificaciones del problema de manera regular	Indaga sobre los requerimientos y especificaciones del problema de manera adecuada
Diagrama de lenguaje unificado básico	Confecciona el diagrama de lenguaje de modelado (UML) de manera inadecuada e incluye las clases y parámetros erróneamente	Confecciona el diagrama de lenguaje de modelado (UML) de manera elemental e incluye las clases y parámetros con varios errores	Confecciona el diagrama de lenguaje de modelado (UML) de manera regular y las clases parámetros tienen algún error	Confecciona el diagrama de lenguaje de modelado (UML) de manera adecuada e incluye las clases y parámetros sin errores

Estructuras de datos	Realiza una justificación de las estructuras de datos que utiliza en la resolución del problema de manera escueta y sin profundidad	Realiza una justificación de las estructuras de datos que utiliza en la resolución del problema de manera básica y con poca profundidad	Realiza una justificación de las estructuras de datos que utiliza en la resolución del problema de manera aceptable y con profundidad	Realiza una justificación de las estructuras de datos que utiliza en la resolución del problema de manera apropiada y profunda
Diseño de estructuras de datos	Diseña las estructuras de datos que implementa en el proyecto de manera inadecuada y con múltiples errores	Diseña las estructuras de datos que implementa en el proyecto de manera básica y con varios errores	Diseña las estructuras de datos que implementa en el proyecto de manera inadecuada y con algún error	Diseña las estructuras de datos que implementa en el proyecto de manera adecuada y sin errores
Interfaz gráfica	Realiza la codificación de la interfaz gráfica del programa de manera inadecuada, con múltiples errores y sin relación con los requerimientos del problema	Realiza la codificación de la interfaz gráfica del programa de manera básica, con varios errores y con poca relación con los requerimientos del problema	Realiza la codificación de la interfaz gráfica del programa de manera regular, con algún error y lo relaciona con los requerimientos del problema	Realiza la codificación de la interfaz gráfica del programa de manera adecuada, sin errores y lo relaciona con los requerimientos del problema
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
Tipos de datos abstractos (TAD)	Realiza los datos de tipo abstractos necesarios para la solución del	Realiza los datos de tipo abstractos necesarios para la solución del	Realiza los datos de tipo abstractos necesarios para la solución del	Realiza los datos de tipo abstractos necesarios para la solución del

	problema de forma escueta	problema de forma incipiente	problema de forma suficiente	problema de forma adecuada
Documentación interna	Entrega la documentación interna de la programación de manera inapropiada	Entrega la documentación interna de la programación de manera básica	Entrega la documentación interna de la programación de manera aceptable	Entrega la documentación interna de la programación de manera apropiada
Documentación de usuario	Entrega la documentación de usuario (en la web) de la programación de manera inapropiada	Entrega la documentación de usuario (en la web) de la programación de manera básica	Entrega la documentación de usuario (en la web) de la programación de manera aceptable	Entrega la documentación de usuario (en la web) de la programación de manera apropiada
Informe técnico	Elabora el informe técnico sin coherencia y sin orden	Elabora el informe técnico de manera coherente, pero está incompleto	Elabora el informe técnico de manera coherente, está completo, pero falta profundidad	Elabora el informe técnico de manera coherente, clara, amplia y está completo
Retroalimentación del docente	Incorpora la retroalimentación brindada por el docente de forma incoherente y escasa	Incorpora la retroalimentación brindada por el docente de manera básica	Incorpora la retroalimentación brindada por el docente de manera incompleta	Incorpora la retroalimentación brindada por el docente de manera clara y completa