



DATOS DEL ESTUDIANTE

Apellidos y Nombres: Valentin Garcia Johan Aamir ID: 001599856
Dirección Zonal/CFP: Lima / Callao
Carrera: Desarrollo de Software Semestre: II
Curso/ Mód. Formativo: CPA / CLA programming essentials (CISCO)
Tema de Trabajo Final: Optimización del Manejo de Datos utilizando Estructuras Dinámicas en C y C++

1. INFORMACIÓN

- **Identifica la problemática del caso práctico propuesto.**

La situación o conflicto que se nos presenta es el siguiente:

En la actualidad, muchas empresas necesitan manejar grandes volúmenes de datos de manera eficiente. Desde la gestión de clientes y productos hasta el procesamiento de transacciones en tiempo real, el uso de estructuras de datos adecuadas es fundamental para optimizar el rendimiento del software empresarial. La empresa "LogiTrans S.A.", dedicada al transporte y distribución de mercancías a nivel nacional, ha experimentado un aumento en la demanda de sus servicios. Sin embargo, su actual sistema de gestión de envíos y rutas presenta problemas de rendimiento debido a la forma en que almacena y procesa los datos. Actualmente, los registros de paquetes, clientes, rutas y horarios están almacenados en archivos de texto plano, lo que genera demoras al buscar información específica y afecta la toma de decisiones en tiempo real. Además, el sistema no optimiza el uso de memoria, lo que ocasiona un consumo excesivo de recursos en los servidores.

Resumen de la situación:

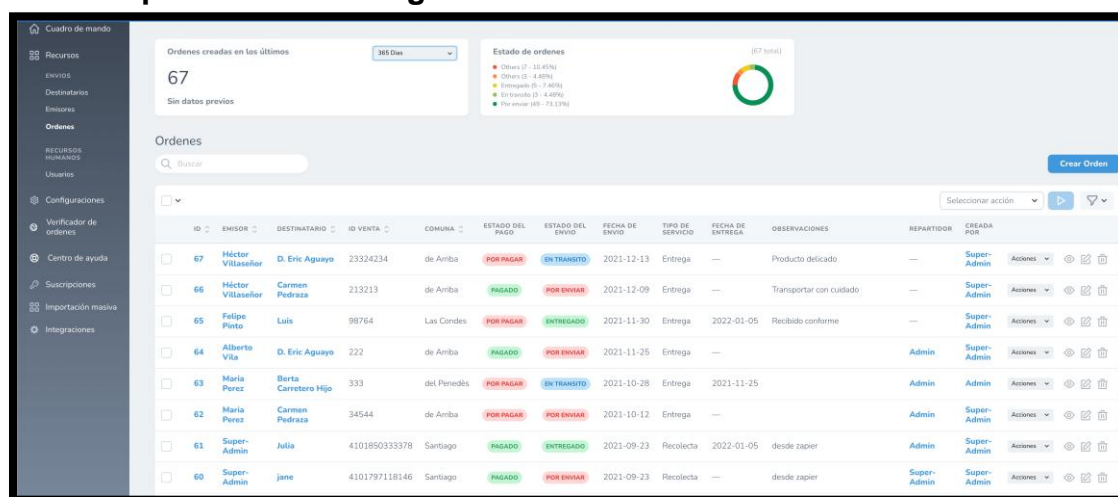
Debido a la alta demanda de pedidos en la empresa LogiTrans S.A, los programas de estos administran o manejan no llega a ser eficiente para atender ese gran número de pedidos.

Identifica propuesta de solución y evidencias.

La solución que nosotros plantearemos, debemos tener en cuenta las siguientes características o requisitos:

- ✓ Colección de Estructuras: Implementar arrays y contenedores dinámicos (vector, queue, stack, priority_queue) para gestionar clientes, envíos y rutas.
- ✓ Arreglos Asociativos: Utilizar map, set y multimap para organizar datos, optimizando la búsqueda y clasificación de paquetes.
- ✓ Procesamiento de Datos: Aplicar algoritmos (for_each, min_element, max_element, mismatch, equal) para analizar y comparar registros de clientes y envíos.
- ✓ Operaciones entre Conjuntos: Implementar algoritmos de fusión para combinar rutas y optimizar entregas, encontrando valores mínimos y máximos en las colecciones.
- ✓ Formatos de Salida: Generar reportes claros con put y manipuladores (setw, setprecision) para mejorar la visualización de datos.
- ✓ Fundamentos de Programación: Aplicar estructuras de control (if, switch, for), manejo de variables, conversión de datos y bucles para automatizar procesos.
- ✓ Estructuras y Funciones: Implementar arrays, estructuras y funciones para modularizar el código, mejorar la reutilización y optimizar la compilación en archivos .h y .cpp.
- ✓ Presentar en formato de informe técnico de los procesos seguidos para desarrollar este trabajo.

La imagen que nos dan como guía o referencia del resultado final de la solución planteada es la siguiente:



ID	EMISOR	DESTINATARIO	ID VENTA	COMUNA	ESTADO DEL PAGO	ESTADO DEL ENVIO	FECHA DE ENVIO	TIPO DE SERVICIO	FECHA DE ENTREGA	OBSERVACIONES	REPARTIDOR	CREADA POR
67	Héctor Villaseñor	D. Eric Aguayo	23324234	de Amba	POR PAGAR	EN TRANSITO	2021-12-13	Entrega	—	Producto delicado	Super-Admin	Acciones
66	Héctor Villaseñor	Carmen Pedraza	213213	de Amba	PAGADO	POR ENVIAR	2021-12-09	Entrega	—	Transportar con cuidado	Super-Admin	Acciones
65	Felipe Pinto	Luis	98764	Las Cordes	POR PAGAR	ENTREGADO	2021-11-30	Entrega	2022-01-05	Recibido conforme	Super-Admin	Acciones
64	Alberto Vila	D. Eric Aguayo	222	de Amba	PAGADO	POR ENVIAR	2021-11-25	Entrega	—	—	Admin	Super-Admin
63	Maria Perez	Berta Carretero Hijo	333	del Penedes	POR PAGAR	EN TRANSITO	2021-10-28	Entrega	2021-11-25	—	Admin	Admin
62	Maria Perez	Carmen Pedraza	34544	de Amba	POR PAGAR	POR ENVIAR	2021-10-12	Entrega	—	—	Admin	Super-Admin
61	Super-Admin	Julia	4101850333378	Santiago	PAGADO	ENTREGADO	2021-09-23	Recolecta	2022-01-05	desde zapier	Admin	Super-Admin
60	Super-Admin	jane	4101797118146	Santiago	PAGADO	POR ENVIAR	2021-09-23	Recolecta	—	desde zapier	Super-Admin	Super-Admin

▪ **Respuestas a preguntas guía**

Durante el análisis y estudio del caso práctico, debes obtener las respuestas a las interrogantes:

Pregunta 01:	¿Cómo pueden las estructuras dinámicas mejorar la eficiencia en la gestión de rutas y envíos?
Las estructuras dinámicas de datos mejoran la eficiencia de gestión de rutas y envíos porque al contrario de los arreglos (que son estáticos), estas permiten trabajar con un límite indefinido que favorece con trabajar con varias cantidades de información y poder buscarla y eliminarla de mejor forma.	
Pregunta 02:	¿Qué ventajas ofrece el uso de <code>priority_queue</code> para la clasificación de entregas urgentes?
Las ventajas que ofrece el uso de <code>priority_queue</code> en este caso son que permite mantener automáticamente los pedidos ordenados según su nivel de urgencia. Esto asegura que al aplicar métodos como <code>top()</code> , <code>push()</code> o <code>pop()</code> , siempre se procese primero el pedido más prioritario, sin necesidad de reordenar manualmente la estructura.	
Pregunta 03:	¿Cómo optimizar la búsqueda de clientes y paquetes utilizando <code>map</code> y <code>set</code>?
<p>Para optimizar la búsqueda de clientes y paquetes, <code>std::set</code> y <code>std::map</code> ofrecen ventajas clave:</p> <ul style="list-style-type: none"> • Set es útil cuando necesitas almacenar elementos únicos (como nombres de clientes o IDs) y buscar rápidamente si un elemento ya existe. La estructura interna garantiza búsquedas eficientes con $O(\log n)$ utilizando <code>find()</code>. • Map es ideal cuando deseas asociar una clave (como un ID) con un valor (como el nombre del cliente o los detalles del paquete). La búsqueda por llave también es eficiente, con $O(\log n)$, permitiendo obtener la información asociada rápidamente. 	
Pregunta 04:	¿Qué técnicas se pueden aplicar para fusionar datos de diferentes listas de envíos?
Para fusionar datos de diferentes listas de envíos, una técnica efectiva es el uso de estructuras como <code>struct</code> o <code>class</code> . Ambas permiten agrupar múltiples variables de distintos tipos (como nombre del cliente, ID de envío, dirección, prioridad, etc.) en una sola unidad. Esto facilita almacenar y manejar cada encomienda como un solo elemento dentro de una lista o arreglo, permitiendo combinar fácilmente información proveniente de distintas fuentes o formatos.	
Pregunta 05:	¿Cómo mejorar la presentación de reportes logísticos con manipuladores de salida en C++?
C++ ofrece manipuladores de salida que permiten dar formato específico a los datos mostrados. Estos manipuladores, como <code>left</code> , <code>right</code> , <code>showpoint</code> o <code>noshowpoint</code> , facilitan la alineación del texto y el control sobre cómo se presentan los valores numéricos. Gracias a ellos, es posible generar reportes más legibles, ordenados y profesionales.	

2. PLANIFICACIÓN DEL TRABAJO

▪ **Cronograma de actividades:**

N°	ACTIVIDADES	CRONOGRAMA					
		Planificación	Algoritmo inicial	Bases de la solución	Aplicación del algoritmo	Aplicación del algoritmo	Informe del trabajo
01	Leer el enunciado y comprender el problema.	28/04/25					
02	Planificar la solución y la propuesta teniendo en cuenta las pautas.		27/04/25				
03	Crear el algoritmo de la solución.			28/04/25			
04	Escribir el código fuente del programa I.				29/04/25		
05	Escribir el código fuente del programa II.					30/04/25	
06	Llenar el informe del trabajo final y comprimir el programa final.						01/05/25

▪ **Lista de recursos necesarios:**

1. MÁQUINAS Y EQUIPOS	
Descripción	Cantidad
Laptop	1
Celular	1
Mouse	1
Router (internet)	1

2. HERRAMIENTAS E INSTRUMENTOS	
Descripción	Cantidad
Visual Studio Community 2022	1
WinRAR	1
Bloc de notas	1
GitHub	1

3. MATERIALES E INSUMOS	
Descripción	Cantidad
Dibujos y gráficos	2
Ideas escritas	5

3. DECIDIR PROPUESTA

- Describe la propuesta determinada para la solución del caso práctico

PROPUESTA DE SOLUCIÓN
<p>Descripción de la propuesta de solución brindada:</p> <ol style="list-style-type: none"> Objetivo general: <p>Crear un programa en C++ que me permita poder crear nuevas encomiendas, trabajar con ellas a tiempo real (uso de estructuras de datos dinámicos), poder incluir funciones (uso de Programación Orientada a Objetos) para modificar y eliminar.</p> Características de la solución: <ul style="list-style-type: none"> Vector dinámico: Facilita la inserción de datos a tiempo real sin tener un tamaño límite fijo. Sistema Do-While: Crea un menú sencillo en el cual podemos escoger de forma sencilla y directa las opciones que ofrece el programa. Clases (POO): Permite que los arreglos admitan más de una sola variable o información. Métodos (POO): Facilita la reutilización de código ayudando a ejecutar funciones específicas de forma sencilla, llamando el método. Tecnologías por utilizar: <ul style="list-style-type: none"> Entorno de desarrollo integrado de preferencia: Visual Studio Community 2022. Programa de compresión: WinRAR. Respaldo de versiones/ código fuente: GitHub. Estructura del programa: <ul style="list-style-type: none"> <vector>: Permite la declaración de un vector dinámico para guardar información. <string>: Permite el uso de tipos de datos de cadena de texto. <iostream>: Permite la entrada y salida de datos (cin y cout). <algorithm>: (Opcional) Permite invertir el vector mostrando primero el último índice y por último el primer índice. Class Cliente: Crea una estructura que permite usar más de un solo tipo de dato, permitiendo tener más información de los clientes (nombre, DNI, teléfono). Class Encomienda: Crea una estructura que permite inicializar varias variables de forma sencilla y reutilizable.

- **Class Empresa:** Cumple la función de administrador, es decir, realizar las acciones con los datos. Esta incluirá todos los métodos necesarios para poder trabajar con el vector, además de la propia declaración del vector.
- **Do-While:** Cumple la función como menú de acciones, permitiendo diversas opciones con funciones dedicadas.

5. Funcionamiento del programa:

Menú: Al iniciar el programa iniciaremos en el menú do-while, con una variable “puntero” lograremos navegar por este menú e ingresar a cualquier opción disponible.

Opción 1: Se llamará a las tres clases, la clase empresa tomará el método para poder ingresar datos, para esto, llamará a la clase encomienda para permitir llevar las variables, de la misma forma, esta clase llamará a la clase cliente para pedir su respectiva información. Al terminar de ingresar los datos, se guardará la información completa (con diversos tipos de datos) en un solo elemento en el vector.

Opción 2: Pedirá al usuario escribir un código, de ahí, ingresará al método de búsqueda de la clase empresa y recorrerá todos los elementos del vector hasta encontrar una coincidencia, al encontrar el valor, mostrará la información guardada en ese índice (posición).

Opción 3: Simplemente accede al método de mostrar encomiendas de la clase empresa y un bucle for-each recorrerá todos los elementos del vector y los mostrará uno a uno.

Opción 4: Pedirá al usuario escribir un código, de ahí, ingresará al método de búsqueda de la clase empresa y recorrerá todos los elementos del vector hasta encontrar una coincidencia, al encontrar el valor, pedirá ingresar un nuevo estado de la encomienda, al ingresar, la información se modifica en la encomienda seleccionada.

Opción 5: Siguiendo la misma lógica, pedirá ingresar un código, buscará en el vector una coincidencia, y al ubicarlo, lanzará un mensaje de confirmación para eliminar el elemento seleccionado.

Opción 6: Al seleccionar esta opción en el menú, el do-while deja de ser true y sale de su bucle, esto hace que se termine el proceso.

Todo el código fuente del programa/ propuesta de solución, está en el GitHub dedicado a este proyecto:

[aamirXD/C-C-Proyecto-final: Muestra del código fuente, y trabajo elaborado para el proyecto final del curso.](#)

4. EJECUTAR

- Resolver el caso práctico, utilizando como referencia el problema propuesto y las preguntas guía proporcionadas para orientar el desarrollo.
- Fundamentar sus propuestas en los conocimientos adquiridos a lo largo del curso, aplicando lo aprendido en las tareas y operaciones descritas en los contenidos curriculares.

INSTRUCCIONES: Ser lo más explícito posible. Los gráficos ayudan a transmitir mejor las ideas. Tomar en cuenta los aspectos de calidad, medio ambiente y SHI.

OPERACIONES / PASOS / SUBPASOS	NORMAS TÉCNICAS - ESTANDARES / SEGURIDAD / MEDIO AMBIENTE
Leer conceptos sobre set, map, estructuras de datos dinámicos.	Mantener una postura adecuada al sentarse. Tomar descansos de 5 minutos.
Leer conceptos de arrays, vectores y matrices, estructuras de datos estáticos.	Mantener una postura adecuada al sentarse. Tomar descansos de 5 minutos.
Plantear el algoritmo y planificar el proceso de la solución.	Mantener una postura adecuada al sentarse. Tomar descansos de 5 minutos.
Configurar y preparar el IDE para poder comenzar a escribir el código fuente (verificar que todo funciona correctamente).	Mantener una postura adecuada al sentarse. Tomar descansos de 5 minutos.
Realizar el código fuente usando estructuras de datos estáticos y	Mantener una postura adecuada al sentarse. Tomar descansos de 5 minutos.
Realizar pruebas del funcionamiento y solución de errores (bugs).	Mantener una postura adecuada al sentarse. Tomar descansos de 5 minutos.
Compilar el programa final y comprimirlo.	Mantener una postura adecuada al sentarse. Tomar descansos de 5 minutos.

IMÁGENES DEL PROGRAMA EN EJECUCIÓN/ RESULTADOS

(Adicionar las páginas que sean necesarias)

```
-----  
      empresa LogiTrans S.A  
-----  
Menu de opciones:  
>1. Crear nueva encomienda  
>2. Buscar encomienda por codigo  
>3. Mostrar encomiendas  
>4. Cambiar estado de la encomienda  
>5. Eliminar encomienda por codigo  
>6. Salir del programa  
-----  
|
```

Ilustración 1 Menú principal del programa

```
1  
-----  
Ingrese codigo de la encomienda:  
01  
Ingrese descripcion de su encomienda:  
coca cola de litro  
Ingrese peso de su encomienda (kg):  
1.2  
Ingrese nombre del emisor:  
aamir  
Ingrese DNI del emisor:  
1234567  
Ingrese telefono del emisor:  
123456789  
Ingrese nombre del destinatario:  
juan  
Ingrese DNI del destinatario:  
1234567  
Ingrese telefono del destinatario:  
123456789  
-----  
Datos guardados correctamente!  
-----
```

Ilustración 2 Opción 1: Creando una encomienda

```
2
-----
Ingrese el codigo de la encomienda:
01
-----
>Codigo de la encomienda: 01
>Estado de la encomienda: por enviar
>descripcion      : coca cola de litro
>peso de la encomienda : 1.2
>Nombre del emisor   : aamir
>DNI del emisor      : 1234567
>Tel del emisor      : 123456789
>Nombre del destinatario: juan
>DNI del destinatario : 1234567
>Tel del destinatario : 123456789
-----
```

Ilustración 3 Opción 2: Buscar una encomienda por código

```
3
-----
>Codigo de la encomienda: 01
>Estado de la encomienda: por enviar
>descripcion      : coca cola de litro
>peso de la encomienda : 1.2
>Nombre del emisor   : aamir
>DNI del emisor      : 1234567
>Tel del emisor      : 123456789
>Nombre del destinatario: juan
>DNI del destinatario : 1234567
>Tel del destinatario : 123456789
-----
>Codigo de la encomienda: 02
>Estado de la encomienda: por enviar
>descripcion      : paquete de galletas
>peso de la encomienda : 0.86
>Nombre del emisor   : miguel
>DNI del emisor      : 1234567
>Tel del emisor      : 123456789
>Nombre del destinatario: lopez
>DNI del destinatario : 1234567
>Tel del destinatario : 123456789
-----
>Codigo de la encomienda: 03
>Estado de la encomienda: por enviar
>descripcion      : laptop lenovo
>peso de la encomienda : 0.89
>Nombre del emisor   : silvia
>DNI del emisor      : 1234567
>Tel del emisor      : 123456789
>Nombre del destinatario: pancho
>DNI del destinatario : 1234567
>Tel del destinatario : 123456789
-----
```

Ilustración 4 Opción 3: Mostrar todas las encomiendas creadas

```

4
-----
Ingrese el codigo de la encomienda:
01
Ingrese nuevo estado (en transito, por enviar, entregado):
entregado
-----
Estado actualizado correctamente!
-----

```

Ilustración 5 Opción 4: Cambiar el estado de una encomienda específica

```

5
-----
Ingrese el codigo de la encomienda:
03
-----
Estas seguro de eliminar la encomienda? S/N
S
-----
Encomienda eliminada correctamente!
-----

```

Ilustración 6 Opción 5: Eliminar encomienda específica

```

6
-----
Saliendo del programa. . .
-----
Presione una tecla para continuar . . . |

```

Ilustración 7 Opción 6: Saliendo del programa

 <p>SENATI</p>	Optimización del Manejo de Datos en una Empresa de Logística utilizando Estructuras Dinámicas en C y C++	
	VALENTIN GARCIA JOHAN AAMIR	COMPLETA

5. CONTROLAR

- Verificar el cumplimiento de los procesos desarrollados en la propuesta de solución del caso práctico.

EVIDENCIAS	CUMPLE	NO CUMPLE
• ¿Se identificó claramente la problemática del caso práctico?	<input checked="" type="checkbox"/>	<input type="checkbox"/>
• ¿Se desarrolló las condiciones de los requerimientos solicitados?	<input checked="" type="checkbox"/>	<input type="checkbox"/>
• ¿Se formularon respuestas claras y fundamentadas a todas las preguntas guía?	<input checked="" type="checkbox"/>	<input type="checkbox"/>
• ¿Se elaboró un cronograma claro de actividades a ejecutar?	<input checked="" type="checkbox"/>	<input type="checkbox"/>
• ¿Se identificaron y listaron los recursos (máquinas, equipos, herramientas, materiales) necesarios para ejecutar la propuesta?	<input checked="" type="checkbox"/>	<input type="checkbox"/>
• ¿Se ejecutó la propuesta de acuerdo con la planificación y cronograma establecidos?	<input checked="" type="checkbox"/>	<input type="checkbox"/>
• ¿Se describieron todas las operaciones y pasos seguidos para garantizar la correcta ejecución?	<input checked="" type="checkbox"/>	<input type="checkbox"/>
• ¿Se consideran las normativas técnicas, de seguridad y medio ambiente en la propuesta de solución?	<input checked="" type="checkbox"/>	<input type="checkbox"/>
• ¿La propuesta es pertinente con los requerimientos solicitados?	<input checked="" type="checkbox"/>	<input type="checkbox"/>
• ¿Se evaluó la viabilidad de la propuesta para un contexto real?	<input type="checkbox"/>	<input checked="" type="checkbox"/>

6. VALORAR

- Califica el impacto que representa la propuesta de solución ante la situación planteada en el caso práctico.

CRITERIO DE EVALUACIÓN	DESCRIPCIÓN DEL CRITERIO	PUNTUACIÓN MÁXIMA	PUNTAJE CALIFICADO POR EL ESTUDIANTE
Identificación del problema	Claridad en la identificación del problema planteado.	3	
Relevancia de la propuesta de solución	La propuesta responde adecuadamente al problema planteado y es relevante para el contexto del caso práctico.	8	
Viabilidad técnica	La solución es técnicamente factible, tomando en cuenta los recursos y conocimientos disponibles.	6	
Cumplimiento de Normas	La solución cumple con todas las normas técnicas de seguridad, higiene y medio ambiente.	3	
PUNTAJE TOTAL		20	

