

Shipping control: programa para la gestión de puertos

Maria Alejandra Perez Petro, Jhon Jairo Silva Zabala, Sergio Nicolas Siabatto Cleves, Esteban David Umbarila Pachón

No. de equipo de trabajo: 7

I. INTRODUCCIÓN

Este documento describe de manera amplia y explícita el proceso de ideación, diseño y ejecución del proyecto de programación correspondiente a la asignatura de Estructuras de datos 2021-2 de la Universidad Nacional de Colombia. El lector de este informe, mediante las descripciones dadas a continuación y a través de las diferentes estrategias multimedia utilizadas, será capaz de comprender cuáles fueron las metodologías de ideación y desarrollo que utilizó el grupo para concretar la idea de proyecto y llevar a cabo su posterior ejecución.

En busca de cumplir con el proyecto, primero se realiza una descripción amplia de la problemática que se abordará y la solución que podemos aportar, dentro de esto se presentan los conceptos básicos, interacciones y en general estado de arte de la problemática en la actualidad. Posteriormente se realizará un resumen acerca de las funcionalidades básicas del proyecto, además de explicar de forma detallada cómo se llevará a cabo la ejecución, presentando las implementaciones de software, estructuras de datos usadas, y análisis de la eficiencia de los algoritmos implementados.

El proyecto que va a presentar es el software “Shipping Control”, cuyo propósito es proporcionar una alternativa que pueda ayudar a solventar algunos de los problemas planteados a continuación, siendo específico en aspectos tales como de qué manera funciona, las implementaciones que posee en cuanto a estructuras de datos y la organización del equipo de desarrollo.

II. DESCRIPCIÓN DEL PROBLEMA A RESOLVER

Los puertos marítimos son importantes estructuras dentro de la cadena de suministros y comercialización que conectan al transporte marítimo y terrestre con los componentes de distribución de carga como lo son la entrada de productos, mercancías y pasajeros a un país, así como también son la puerta de salida para todas las exportaciones hacia los mercados internacionales.

El comercio internacional, puede decirse, es movido gracias a estas estructuras, pero además es la base de los comercios locales ya que son puntos de convergencia entre el interior y

los sistemas de transporte costeros, lo que se define como el área de influencia interior del puerto. Esta función puede ser directa a través del acceso por carreteras, indirecta en la medida que la carga llegue a una terminal interina (por ejemplo, una estación de tren), o por medio de la consolidación del tráfico en un puerto regional y embarcados por cabotaje. Los puertos marítimos son puntos de distribución hacia el interior y el sistema de transporte costero, lo que se define como el área de influencia exterior del puerto.

Las labores que se realizan dentro de un puerto marítimo debido a la confluencia de diversos factores propios y externos resultan ser de alta complejidad, dentro de un puerto de contenedores encontramos trabajos básicos como embarque, desembarque y transferencia de cargo hacia y desde un buque o almacenamiento y depósito de mercancías en tierra y estiba hacia y desde barcos, estas pueden ser dos operaciones básicas sin embargo al mirarlás más explícitamente nos damos cuenta que son una gran cantidad de gestiones que se deben realizar para que se cumplan estas labores, se debe contar con el personal capacitado para realizarlas y además de cumplir con los tiempos adecuados que permitan un funcionamiento normal de la totalidad del puerto, como vemos esta complejidad es absurdamente grande si consideramos todos los procesos que se realizan dentro del puerto. Además de esto, hoy en día, debido al amplio consumismo y facilidad de ordenamiento de bienes que ha conseguido la humanidad, la vasta red de puertos, buques portacontenedores y empresas de transporte por carretera que mueven dichas mercancías es demasiado laberíntica y congestionada. La probabilidad de que una embarcación llegue a tiempo es aproximadamente del 40%, a diferencia del año pasado cuando esta era del 80% [1].

Actualmente hay un elevado número de buques portacontenedores que transportan mercancías, principalmente en Estados Unidos y China, inactivos en puertos mientras esperan descargar su cargamento, asimismo, hay escasez de contenedores en muchos puertos mientras que se acumulan contenedores vacíos en bastantes otros [2]. Estas dificultades en la cadena de suministro han provocado importantes retrasos y costos logísticos adicionales que demuestran una crisis de logística en los puertos la cual podría solucionarse mediante un enfoque en el papel de la inmensidad de los datos mediante una interfaz para la gestión y el control de los puertos.

III. USUARIOS DEL PRODUCTO DE SOFTWARE

Debido a que es una implementación general para la gestión de varios aspectos de los puertos tales como el número de bodegas disponibles o de contenedores descargados y en espera, el único rol con el que se contará es el de administrador, teniendo el acceso para agregar, buscar, modificar y eliminar cualquier dato pertinente a aspectos como los mencionados anteriormente, por lo que quién maneje este software deberá ser alguien que tenga, como mínimo, un alto rango dentro de la actividad portuaria.

IV. REQUERIMIENTOS FUNCIONALES DEL SOFTWARE

El software que se estará desarrollando tiene diversos requerimientos funcionales que permiten un flujo normal del programa, estas funcionalidades son descritas a continuación:

- Descripción de los datos a ser ingresados en el sistema:
 - Inicio de sesión: Los usuarios podrán iniciar sesión con sus credenciales registradas anteriormente, si los datos ingresados no existen, se le pedirá al usuario que realice la creación de estos.
 - Registro de usuarios: Debido a que el programa está enfocado en usuarios de alto rango dentro de la organización portuaria, se podrán crear distintos perfiles con las mismas funcionalidades pero vinculadas todas a un único puerto por copia del software, para esto el trabajador debe crear un nombre de usuario y una contraseña que serán guardados en un archivo de extensión csv.
 - Registro de Puerto: La primera vez que se ejecute el software, este le pedirá al usuario registrar los datos del puerto que no podrán ser cambiados según la lógica de estos, como su nombre, el número de bodegas que posee y la capacidad de almacenamiento de estas.
 - Registro de Embarcaciones: Mediante el software se podrá realizar el registro de las diferentes embarcaciones donde se especificarán datos tales como su número IMO, la bandera de su país, su tipo y su estado de disponibilidad, entre otros, estos serán guardados en un archivo de extensión csv.
 - Consulta de embarcaciones: El software permitirá realizar búsquedas dentro de las embarcaciones registradas anteriormente filtrándose por alguno de los ítems mencionados anteriormente.
 - Actualización de embarcaciones: El software permitirá realizar updates a las embarcaciones registradas anteriormente, de igual manera se considera en este punto la eliminación de

registros según el flujo del programa o las acciones del usuario.

- Registro de entrada de embarcaciones: El software permitirá llevar un registro de las embarcaciones que llegan a los puertos, esto por medio del uso de una cola.
 - Registro de salida de embarcaciones: El software permitirá llevar un registro de las embarcaciones que dejan los puertos, esto por medio del uso de una cola.
 - Registro de carga de embarcaciones: El software permitirá registrar los barcos cargados, estos, al desencolar (salida de embarcaciones), se apilan y se cargarán.
 - Consulta de bodegas: El software permitirá visualizar las bodegas que fueron estipuladas al momento de registrar el puerto, permitiendo modificar su zona y sus contenedores actuales.
- Descripción de los flujos de trabajo realizados por el sistema.
 - Ejecución inicial del programa: Al iniciar por primera vez el programa, se procede a realizar el inicio de sesión de usuario, si no posee estas podrá darle a la opción “¿no tienes cuenta? **Crear**” y sus credenciales serán guardadas en almacenamiento local mediante archivos .csv.
 - Interfaces de selección: Mediante el ingreso a pestañas y uso de botones el usuario se encargará de decidir cuáles de las acciones mostradas desea realizar, el flujo del programa dependerá entonces de las selecciones propias hechas por cada usuario.
 - Registro con estructuras de datos: De acuerdo a las selecciones de los usuarios se podrían realizar tareas de registro de diferentes datos que consideramos necesarios para el funcionamiento normal del software.
 - Almacenamiento local: En busca de evitar pérdida de datos, cada vez que se realicen creaciones, modificaciones, o eliminaciones de se realizarán estos registros en archivos .csv al finalizar la sesión, de esta manera cuando se vuelva a ejecutar el programa se cargará nuevamente esta información para que quede guardada en tiempo de ejecución.
 - Cierre de sesión: Sólo se permitirá la salida del software mediante la pestaña del cierre de sesión puesto que de ahí ejecutará el guardado de los datos modificados por el usuario.
 - Descripción de los reportes del sistema y otras salidas.
 - Muestra de registros: se considera que el usuario sea capaz de visualizar mediante una interfaz gráfica que sea capaz de abstraer las diferentes estructuras utilizadas en modelos relacionales tales como tablas, los registros que ya se han

realizado, además de la posibilidad de interactuar dinámicamente con estos.

- Definición de quiénes pueden ingresar datos en el sistema:
 - Todo usuario que esté registrado en el archivo de extensión .csv tendrá acceso para ingresar, buscar, modificar y eliminar los datos correspondientes a los diferentes apartados disponibles.

V. DESCRIPCIÓN DE LA INTERFAZ DE USUARIO PRELIMINAR

Al iniciar el programa se le mostrará al usuario la ventana de inicio de sesión (figura 1), en caso de no estar registrado puede hacer click en el botón crear.

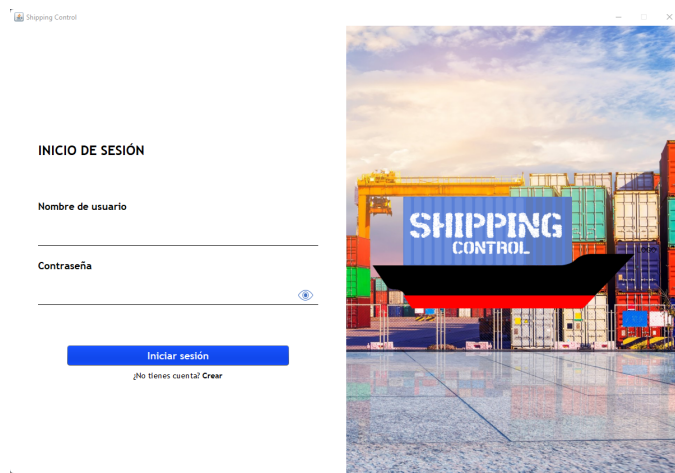


Figura 1. Ventana de inicio de sesión.

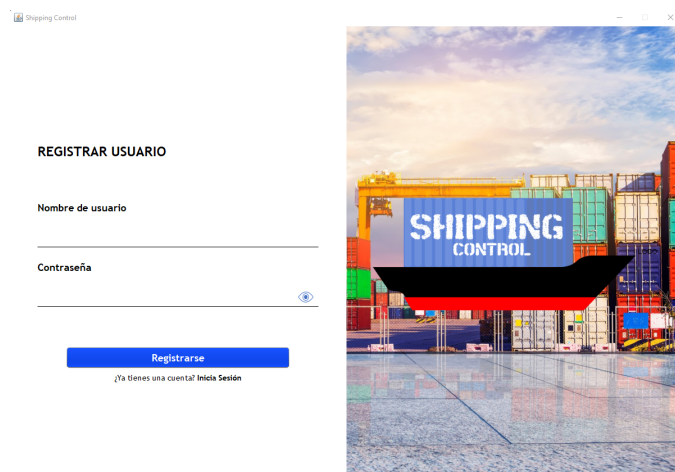


Figura 2. Ventana registro de usuario.

Una vez iniciada la sesión se le mostrará la ventana del perfil del puerto (figura 3) con el nombre, capacidad de embarcaciones y número de bodegas. El usuario podrá usar el menú de la parte superior para consultar las embarcaciones y las bodegas.



Figura 3. Ventana perfil del puerto

En la pestaña de embarcaciones se mostrará una tabla con la información de cada embarcación. El usuario podrá registrar, editar, descargar una embarcación y actualizar la tabla..



Figura 4. Ventana para consultar las embarcaciones.

Figura 5. Ventana de registro/Edición de la información de la embarcación.

En la ventana de bodegas se mostrará una tabla con la información de cada bodega. El usuario podrá transportar contenedores de una bodega fuera del puerto.

ID	Capacidad máxima contenedores	Contenedores actuales	Zona
0	1000	0	A
1	1000	0	E
2	1000	0	B
3	1000	0	D
4	1000	0	D
5	1000	0	D
6	1000	0	D
7	1000	0	A
8	1000	0	E
9	1000	0	A
10	1000	0	E
11	1000	0	C
12	1000	0	A
13	1000	0	B
14	1000	0	C
15	1000	0	B

Figura 6. Ventana para consultar las bodegas.

VI. ENTORNOS DE DESARROLLO Y DE OPERACIÓN

Este software se desarrollará en el lenguaje de programación Java 15 y funcionará gracias al compilador de este mismo mediante entornos de desarrollo tales como NetBeans. Ya que este programa necesitará Java 15 para ejecutarse, sus requisitos de hardware y sistema operativo son los mismos que el de esta versión del lenguaje. En cuanto al software y hardware de los desarrolladores y sobre el cuál se realizó la prueba mostrada en el video del apartado X se trata de un equipo con Windows 10 y las siguientes características:

- Procesador: Intel i5 10300 h a 2.5GHz
- Gráfica: Nvidia GTX 1650 4GB VRAM
- Memoria RAM: 8GB

- Disco duro: SSD 500GB

VII. PROTOTIPO DE SOFTWARE

La versión funcional del software Shipping Control se mantendrá actualizada por todos los miembros del equipo utilizando la herramienta de trabajo remota GitHub, donde se encontrará el repositorio con la estructura sugerida y que podrá ser accedido por el siguiente link:

<https://github.com/jhsilvaz13/ShippingControl>

Con base en los esquemas que se pueden ver mediante los mockups fue implementada la interfaz gráfica con las ventanas que se verán a continuación:

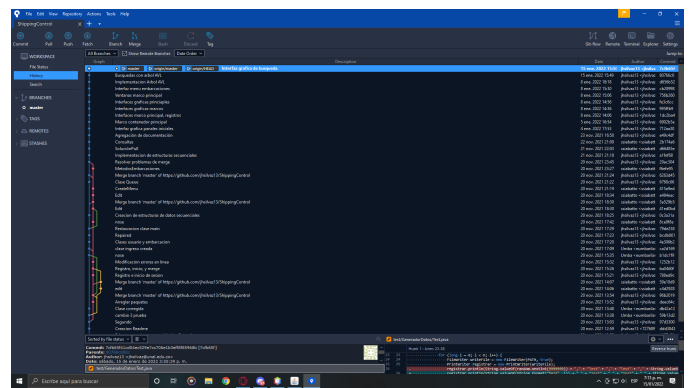


Figura 7. Versión gráfica del desarrollo del repositorio.

VIII. IMPLEMENTACIÓN Y APLICACIÓN DE LAS ESTRUCTURAS DE DATOS

Para la entrega de este proyecto se añadieron las estructuras de datos que consideramos adecuadas teniendo en cuenta el avance del curso, es decir, se implementaron únicamente estructuras de datos secuenciales para las acciones de creación, edición, impresión y eliminación tanto de embarcaciones como de usuarios y bodegas. La siguiente lista describe algunas de las estructura que fueron implementadas e instanciadas en el proyecto:

- Arreglo $O(1)$: El uso de arreglos no dinámicos es implementado principalmente para el almacenamiento parcial de datos, es decir, se encargan de recolectar datos ingresados manualmente o cargados por los archivos csv y son llevados a clases más complejas donde se utilizan para la creación de objetos e impresión de registros, debido a que con estos solo están realizando operaciones de inserción al final del arreglo y búsquedas por índices ambas operaciones son de complejidad $O(1)$.
- Listas simplemente enlazada $O(n)$: Esta estructura tiene mayor importancia como clase, y no instancia, debido a que es utilizada para implementar demás estructuras secuenciales, sin embargo, sí existen algunas lista enlazadas que se encargan de facilitar

labores de recorridos completos mediante otras estructuras.

- Pila $O(1)$: En busca de tener un registro de historial de las embarcaciones salientes se decide utilizar esta estructura para poder buscar mediante la operación pop, cuales han sido las embarcaciones más recientes que han abandonado el puerto, información que puede ser utilizada de manera provechosa en un software de mayor complejidad.
- Cola $O(1)$: Esta estructura permite simular de manera eficaz una fila de embarcaciones, puesto que en un puerto marítimo las embarcaciones que llegan primero deben ser las que salgan primero. Esta es la estructura de mayor relevancia debido a que constantemente se revisan las inserciones y eliminaciones que se realizan sobre esta, es decir se están realizando las operaciones básicas de una cola, Enqueue y Dequeue, ambas operación de complejidad $O(1)$ si implementamos la cola mediante lista enlazada.
- ArrayList $O(n)$: Se hace uso de arreglos dinámicos para almacenar información resultante de cualquier tipo de búsqueda, es decir, simula un modelo relacional de búsqueda, donde las filas son representadas por un arreglo estáticos, y la colección completa corresponde al arreglo dinámicos.
- Árboles $O(\log n)$: Específicamente el árbol AVL fue el utilizado para la búsqueda de datos por índices, los cuales se caracterizan por no repetirse, logrando así una complejidad de $O(\log n)$.
- HashMap $O(1)$: El objetivo de uso de esta estructura es para realizar búsquedas que tengan identificadores repetidos, ya que al filtrar por campos donde se puedan repetir valores, se implementó en este diccionario un método insert en el cual se permiten llaves repetidas y así organizar de manera eficiente en listas enlazadas aquellos objetos con atributos iguales.

IX. PRUEBAS DEL PROTOTIPO Y ANÁLISIS COMPARATIVO

Cada nueva embarcación que llega al puerto se guarda en una cola implementada mediante una lista enlazada, de modo que, la funcionalidad “Registrar entrada de embarcación” es $O(1)$, puesto que, la operación Enqueue de la cola se implementó agregando las embarcaciones al final de la lista enlazada. Por otra parte, cada salida de una embarcación del puerto se guarda en una pila implementada mediante una lista enlazada bajo el nombre de “Registrar salida de embarcación” siendo $O(1)$, ya que, la operación Push de la pila se implementó agregando las embarcaciones al inicio de la lista enlazada. Finalmente, las embarcaciones actuales dentro del puerto quedan guardadas en la cola implementada mediante lista enlazada (debido a que se desencola cada embarcación que sale del puerto), de modo que la funcionalidad “mostrar embarcaciones actuales” es $O(n)$ porque para poder imprimir

las embarcaciones actuales por orden de llegada, es necesario recorrer la lista completa, empezando desde el primer nodo. Asimismo, para poder escribir el archivo csv con los embarcaciones actuales, es necesario también recorrer la cola, dejando a esta funcionalidad como $O(n)$.

La tabla 1 y figura X muestra los resultados obtenidos al medir los tiempos que toma realizar a tres funcionalidades consideradas como las más tardadas para el software (registrar embarcación, mostrar embarcaciones actuales y escribir archivo csv) para los siguientes tamaños de datos de prueba (n):

- 10 mil datos.
- 100 mil datos.
- 1 millón de datos.
- 10 millones de datos.
- 100 millones de datos.

Datos	Registro embarcación	Mostrar embarcaciones actuales	Escribir archivo csv
10 000	1	24592	2953
100 000	1	265831	45527
1 000 000	1	2631321	139273
10 000 000	1	9975722	1415138
100 000 000	1	95517793	94342207

Tabla 1. Tiempo obtenido para cada funcionalidad considerada (los valores resaltados en amarillo son aproximados).

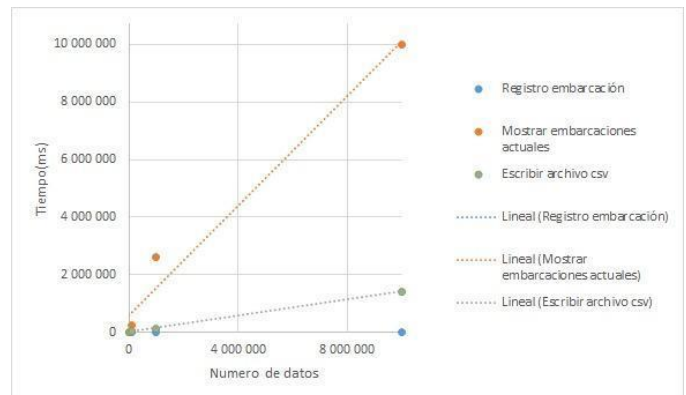


Figura 8. Tiempo vs número de datos para cada funcionalidad considerada (lineal).

Como se observa en la figura 8, las funcionalidades evaluadas siguen los comportamientos esperados. La funcionalidad mostrar embarcaciones y escribir archivo csv son $O(n)$, pero esta primera es la que posee el mayor costo computacional pues es la recta de mayor pendiente. Por otra parte, la funcionalidad registrar embarcación es de complejidad $O(1)$, significando que su costo computacional no varía según la cantidad de datos que esté manejando.

Igualmente vemos en las figuras 10 y 11, las gráficas correspondientes a los costos computacionales para las operaciones de inserción y búsqueda en un árbol AVL, efectivamente los datos obtenidos en las medidas del tiempo en ejecución se ajustan a las gráficas esperadas, ya que el costo total para las inserciones es de $O(n \log_2(n))$ pues primero recorre por completo una lista enlazada y posterior a esto realiza las inserciones en el árbol. El tiempo de búsqueda en el árbol AVL corresponde a la gráfica de $O(\log_2(n))$ ya que se realiza una búsqueda recursiva de acuerdo al valor que se está comparando.

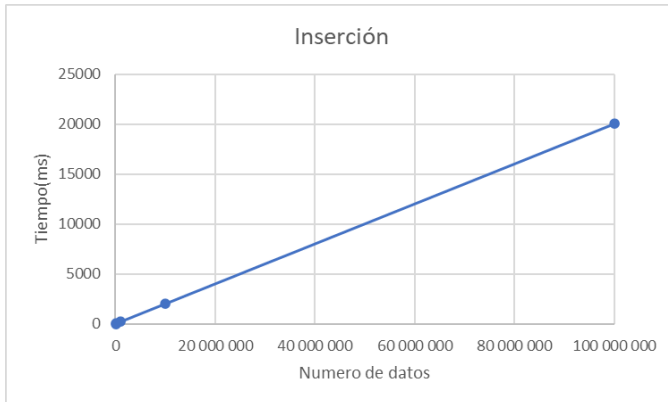


Figura 9. Tiempo vs número de datos para la funcionalidad Inserción.

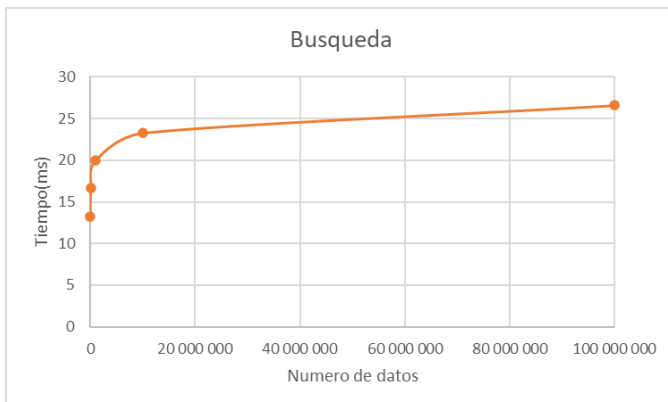


Figura 10. Tiempo vs número de datos para la funcionalidad de búsqueda.

Numero de datos	tiempo (ms)	
	Inserción	Busqueda
10 000	30	13
100 000	43	17
1 000 000	229	20
10 000 000	2025	23
100 000 000	20025	27

Tabla 2. Tiempo obtenido para funcionalidad inserción y búsqueda en árbol AVL (los valores resaltados en amarillo son aproximados).

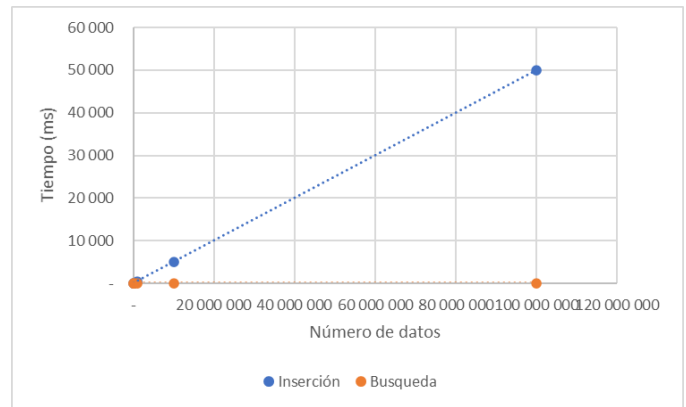


Figura 11. Tiempo vs número de datos para la funcionalidad de búsqueda e inserción con HashMap.

numero de datos	tiempo(ms)	
	Inserción	Busqueda
10 000	10	100
100 000	45	100
1 000 000	490	100
10 000 000	5 001	100
100 000 000	50 001	100

Tabla 3. Tiempo obtenido para funcionalidad inserción y búsqueda en árbol HashMap (los valores resaltados en amarillo son aproximados).

En las tablas 2 y 3 se puede apreciar que el tiempo de inserción se comporta como $O(n)$ cuando debería ser $O(1)$, esto se debe a que debe recorrer todo el archivo, por lo tanto entre más datos deba recorrer, la función `System.currentTimeMillis` (la cual se utilizó para medir el tiempo de ejecución) retorna valores mayores.

X. INFORMACIÓN DE ACCESO AL VIDEO DEMOSTRATIVO DEL PROTOTIPO DE SOFTWARE

El video correspondiente a la presentación del prototipo funcional (el cuál podrá ser revisado a nivel de código mediante el link presentado en el apartado VII) se encuentra disponible en:

[Estructura de Datos Proyecto Entrega 3 Equipo 07](#)

XI. ROLES Y ACTIVIDADES

Para el desarrollo de esta entrega, los roles y actividades realizadas por cada uno de los integrantes se encontrarán en la siguiente tabla:

INTEGRANTE	ROL(ES)	ACTIVIDADES FUNDAMENTALES
------------	---------	---------------------------

Maria Alejandra Perez Prieto.	Observadora	Realizó aportes visuales a la interfaz gráfica.
	Investigadora	Averiguó funcionamientos de JFrame
Jhon Jairo Silva Zabala	Líder	Creó el plan de trabajo pertinente a la entrega.
	Experto	Resolvía inquietudes en cuanto a código
Sergio Nicolas Siabatto Cleves	Técnico	Se encargó de problemas con Git Bash.
	Secretario	Llevó registro de lo realizado en reuniones.
Esteban David Umbarila Pachón	Coordinador	Agendó las reuniones pertinentes a la entrega.
	Animador	Animó al equipo a no cancelar.

XII. DIFICULTADES Y LECCIONES APRENDIDAS

El trabajo cooperativo siempre representa un reto mayor para cualquier proyecto que se desee desarrollar, es por eso que se debe hacer uso de diferentes estrategias que faciliten la comunicación, la asignación de deberes y el desarrollo de actividades, dentro de las dificultades que se presentaron en estas dos entregas consideramos que los problemas que más complicaron la realización del proyecto fueron:

- **Organización de tiempos:** A pesar de que todos los integrantes del grupo mostraron interés para la realización del proyectos en los espacios que concretamos, consideramos que pudimos haber organizado de mejor manera los tiempos de cada uno para concretar reuniones periódicas que permitieran avanzar sobre el proyecto, y así evitar acumular diversas tareas para un espacio temporal más corto.
- **Utilización de git y GitHub:** Aunque Git y GitHub resultan ser dos herramientas muy útiles para el control de versiones del proyecto y el trabajo conjunto de diversas personas en el mismo proyecto, concluimos que nos faltó mayor preparación para comenzar a usarlo plenamente en el proyecto.
- **Errores de implementación de JFrame:** El salto de comandos en consola a una interfaz gráfica trajo consigo ciertos problemas tanto visuales como de funcionalidad que tuvimos que sortear en orden para mostrar un prototipo agradable y práctico de manejar para el/los usuario/s.
- **Limitación de SQL:** El conocimiento de algunos miembros del equipo en semestres pasados manejando SQL mediante XAMPP y PhpMyAdmin parecía excelente para aplicar en este software pero debido a las limitaciones de cargo de datos en la nube (en el caso de PhpMyAdmin siendo 40mb),

imposibilitaba el manejo de grandes cantidades de datos como las que se piden en las pruebas del apartado IX así que se decidió seguir trabajando con archivos csv.

Las lecciones aprendidas en estas entregas es que podemos organizar de mejor manera los tiempos de cada uno buscando espacios semanales donde podamos revisar el avance del proyecto, igualmente como compromiso de grupo se destaca buscar mayor información sobre el uso de las funcionalidades de JFrame y versiones del proyecto que a futuro para lograr plasmar una versión más fiel a la ideada desde un principio.

XIII. REFERENCIAS BIBLIOGRÁFICAS

- [1] "The shipping crisis is getting worse. Here's what that means for holiday shopping - CNN." [Online]. Available: <https://edition.cnn.com/2021/08/23/business/global-supply-chains-christmas-shipping/index.html>
- [2] "Why even giant ships can't solve the shipping crisis," BBC News, Sep. 13, 2021. [Online]. Available: <https://www.bbc.com/news/business-58479148>