

Aplicación Solidaria

Diego Alvarez, Andres Orduz, Jhoan Franco, Carlos Reyes

I. INTRODUCCIÓN

En Colombia se pierden 9.76 millones de toneladas de alimentos al año, generando un impacto tanto económico como social, pero a la vez, 2.7 millones de colombianos sufren de hambre crónica y las medidas del gobierno no han sido suficientes para solucionar esta problemática. Tomando en cuenta lo anterior, se ve la necesidad de implementar una solución, que reduzca tanto la pérdida de alimentos y a la vez, se ayude a las personas con necesidades económicas.[1]

II. DESCRIPCIÓN DEL PROBLEMA A RESOLVER

En nuestro país, la pérdida de alimentos obedece a distintos factores, como lo son plagas, cambio climático, carencias de tecnología y tecnología, deficiencias en infraestructura y capacidad, malos canales de distribución y cadenas de mercado, unos malos hábitos de consumo y la falta de coordinación entre los sectores público y privado.[2]

Como vemos son varios los factores, y encontrarles una solución a todos en un corto plazo es casi imposible, por lo cual, nosotros nos vamos a enfocar en 3 etapas de la distribución y consumo de los alimentos: La primera etapa es la de distribución, donde se pierde aproximadamente el 20.3% de los alimentos, la segunda etapa es la de almacenamiento donde se pierde el 19.8% y la última etapa que nos enfocaremos es la de los hogares, donde se pierde el 15.6% de los alimentos. [3]

Para solucionar este problema, se plantea crear una aplicación que permita a las empresas y donadores ocasionales publicar los productos que tienen disponibles para donar, de igual manera, los beneficiarios pueden acceder a dicha aplicación para consultar qué productos están disponibles, conectando así a personas con necesidades económicas con productos que ofrecen otras personas para evitar que se echen a perder.

III. USUARIOS DEL PRODUCTO DE SOFTWARE

La aplicación contará con 3 usuarios:

- **Beneficiario:** Con este rol el usuario podrá ver toda la lista de productos disponibles y toda la información relacionada con estos.
- **Donador Ocasional:** Este rol puede publicar los productos que tiene para donar, donde podrá registrar sus datos personales y la información sobre el producto, que puede ser alimentos, como algún útil escolar que ya no utiliza.

- **Empresa:** Este rol se asocia con una tienda de barrio, un almacén de cadena o una bodega de Corabastos, la cual, puede registrar toda la lista de productos que tienen disponible para la donación.

IV. REQUERIMIENTOS FUNCIONALES DEL SOFTWARE

- Creación de usuarios

Descripción: Cada usuario que desee usar la aplicación primero debe registrarse, para esto debe proporcionar cierta información. El usuario deberá recordar el correo y la contraseña con la que se registra ya que estos datos se usarán para loguearse.

Acciones iniciadoras y comportamiento esperado:

Al dar click en el botón de registrarse el usuario será redirigido a una sección en donde se le solicitará diligenciar los datos correspondientes a número de documento de identidad, nombres, apellidos, correo, contraseña y rol.

Requerimientos funcionales:

- ❖ RF - 1 Registro de Usuario: El sistema debe permitir a los usuarios registrarse proporcionando la siguiente información: número de documento de identidad, nombres, apellidos, correo electrónico, contraseña.
- ❖ RF - 2 Selección de rol: Durante el proceso de registro, el sistema debe permitir a los usuarios seleccionar un rol entre las opciones disponibles: beneficiario, donador ocasional o empresa.
- ❖ RF - 3 Almacenamiento de datos de usuario: El sistema debe almacenar de manera segura los datos del usuario registrado, incluyendo su número de documento de identidad, nombres, apellidos, correo electrónico, contraseña y rol.

- Creación de productos

Descripción: Los usuarios con el rol de ‘donador ocasional’ y ‘empresa’ pueden crear productos, esto con el fin de publicarlos y que los usuarios con el rol de ‘beneficiario’ tengan la opción de visualizar todos los productos disponibles para ellos.

Acciones iniciadoras y comportamiento esperado:

Al dar click en el botón de registrar producto, el usuario será redirigido a una sección en donde se le solicitará diligenciar los datos correspondientes a tipo de producto, nombre del producto, cantidad del producto, fecha de vencimiento, email del donador. Posteriormente, el usuario será redirigido a la vista de todos los productos disponibles, esto para que se verifique que el producto fue publicado.

Requerimientos funcionales:

- ❖ RF - 1 Registro de producto: El sistema debe permitir a los usuarios con el rol de 'donador ocasional' y 'empresa' crear productos proporcionando la siguiente información: tipo de producto, nombre del producto, cantidad del producto, fecha de vencimiento, email del donador.
- ❖ RF - 2 Almacenamiento de datos: El sistema debe almacenar de manera segura los datos del producto registrado.

● Consulta total de productos

Descripción: Todos los usuarios sin importar su rol pueden ver la lista de productos disponibles, aunque es una funcionalidad que tiene mayor funcionalidad en el rol 'Beneficiario', ya que permite mostrar la totalidad de productos.

Acciones iniciadoras y comportamiento esperado:

Al ingresar al apartado de "Productos disponibles" se mostrarán todos los productos que estén disponibles en ese momento de forma automática, por lo que no será necesario activar esta funcionalidad de forma externa.

Requerimientos funcionales:

- ❖ RF -1 Acceso de datos: El sistema debe tener la capacidad de acceder a la base de datos para poder recuperar la información necesaria.
- ❖ RF - 2 Ordenamiento de datos: Debe ser posible ordenar los resultados de la consulta en función de uno o varios campos, ya sea de forma ascendente o descendente.

● Búsqueda parcial de productos

Descripción: Permite al usuario hacer una búsqueda más específica de aquellos productos que sean de su interés, por medio de esta función permitirá filtrar de manera más rápida los productos, ya sea por su nombre, tipo o cantidad disponible.

Acciones iniciadoras y comportamiento esperado:

Por medio de una barra de búsqueda el usuario podrá escribir aquel o aquellos productos que sean de su interés y al dar click a buscar en el apartado de productos disponibles aparecerá únicamente aquellos productos relacionados con la búsqueda del usuario.

Requerimientos funcionales:

- ❖ RF -1 Búsqueda por criterios: Los usuarios deben poder especificar criterios de búsqueda que limiten los resultados de la consulta.

- ❖ RF -2 Visualización de datos: Los resultados de la consulta deben mostrarse de manera legible y comprensible para los usuarios, con opciones de visualización.
- ❖ RF - 3 Validación de datos: Al hacer uso de la función se debe verificar que lo que se está solicitando exista en la base de datos.

● Actualización de productos

Descripción: Únicamente aquellos usuarios con rol de donador, ya sea ocasional o de empresa, tienen acceso a esta funcionalidad, ya que permite realizar cambios en la información de los productos que han sido publicados.

Acciones iniciadoras y comportamiento esperado:

En el apartado del donador contará con un botón en cada producto, el cual al darle click dirigirá al usuario a una ventana en la cual podrá editar la información que desee del producto.

Requerimientos funcionales:

- ❖ RF -1 Autenticación: El usuario donador debe estar logueado, ya que se debe validar su rol y los permisos con los que cuenta.
- ❖ RF -2 Confirmación de actualización: El sistema proporciona una confirmación al usuario después de realizar una actualización de datos
- ❖ RF -2 Notificación: Se debe notificar al usuario donador que la actualización de su producto ha sido realizada correctamente.

● Actualización de usuario

Descripción: Por medio de la funcionalidad el usuario tendrá la opción de actualizar la información que necesite de su cuenta, ya sea su email, contraseña, dirección, etc.

Acciones iniciadoras y comportamiento esperado:

Para iniciar esta acción será necesario que el usuario se dirija a su perfil y haga clic en el botón de actualizar información, una vez hecho esto se dirigirá a una ventana en la cual le permitirá modificar los campos que necesite y posteriormente guardar estos cambios.

Requerimientos funcionales:

- ❖ RF -1 Validación de datos: Se deben implementar validaciones de datos para garantizar que los datos ingresados sean precisos y cumplan con ciertas restricciones, como rangos de fechas o formatos específicos.
- ❖ RF -2 Seguridad: Deben implementarse medidas de seguridad para proteger los datos y garantizar que la información sensible no sea modificada por usuarios no autorizados.

- Eliminación de productos

Descripción: Por medio de esta funcionalidad permitirá a los usuarios que sean donadores eliminar aquellos productos que ya no estén disponibles o en el caso de la comida que hayan vencido.

Acciones iniciadoras y comportamiento esperado:

En el apartado de los productos del donador se encontrará un botón que al darle clic le permitirá elegir aquellos productos que quiere eliminar y cuando los haya seleccionado deberá confirmar la eliminación de los productos para que dejen de aparecer en el apartado de beneficiario.

Requerimientos funcionales:

- ❖ RF -1 Autenticación: Se debe validar que el usuario que está eliminando un producto sea un beneficiario.
- ❖ RF -2 Confirmación de eliminación: Antes de realizar una eliminación, el sistema debe solicitar una confirmación al usuario para asegurarse de que la acción se lleva a cabo intencionalmente y para evitar eliminaciones accidentales.
- ❖ RF -3: Eliminación permanente o marcado: Se debe decidir si la eliminación de datos es permanente o si se marcan los registros como eliminados pero se mantienen en el sistema de manera que puedan recuperarse en el futuro.

- Ordenamiento de productos:

Descripción: Permite mantener un orden dado a los datos almacenados como productos, este orden dado puede ser delimitado por el nombre de manera alfabética o por la fecha de vencimiento, entre otras formas.

Acciones iniciadoras y comportamiento esperado:

Se espera que el ordenamiento se haga de manera automática para que así al mostrarse a los usuarios sea más fácil de entender y mantener un orden de los productos que necesitan.

Requerimientos funcionales:

- ❖ RF -1 Visualización de datos ordenados: Los datos deben mostrarse de manera legible y comprensible para los usuarios
- ❖ RF -2 Selección de criterios: El usuario debe tener la posibilidad de cambiar el ordenamiento de los productos a como le parezca más conveniente.

- Almacenamiento de productos:

Descripción: Por medio del almacenamiento de datos permite a los usuarios que son donantes guardar la

información de sus productos y que estos persistan en el programa, para así posteriormente poder ser consultados, modificados, etc.

Acciones iniciadoras y comportamiento esperado:

Cada vez que un usuario donador ingrese un nuevo producto este será almacenado en la base de datos y por lo tanto persistirá.

Requerimientos funcionales:

- ❖ RF -1 validación: Se deben aplicar validaciones de datos para garantizar que los datos ingresados sean precisos y cumplan con las restricciones y formatos especificados.
- ❖ RF -2 autenticación: El usuario que desee almacenar la información de un producto debe haber sido validado como donador.

V. DESCRIPCIÓN DE LA INTERFAZ DE USUARIO PRELIMINAR

Para la interfaz de usuario de la aplicación serán creadas varias ventanas, al momento iniciar la aplicación lo primero que verá el usuario será la ventana de logueo o registro en la cual deberá ingresar su correo, contraseña y tipo de usuario en caso de haberse registrado anteriormente, o en caso contrario, deberá dirigirse a la pestaña de registro, una vez logueado según su rol será dirigido a una pestaña diferente, en caso de ser un donador ya sea ocasional o una empresa, será dirigido a la ventana de “donar un nuevo producto” en el cual podrán ingresar los nuevos productos que tengan disponibles para ser donados junto con sus características (tipo de producto, cantidad, etc.), a su vez los donantes podrán ver los productos que han ingresado a la aplicación y editar información de estos según se necesite; por otro lado, si el rol del usuario logueado es beneficiario, será dirigido a la ventana de “productos disponibles” en la cual podrán ver los productos en forma de lista, a su vez pueden ver todas las características de cada producto y su disponibilidad. Por último, los usuarios que sean beneficiarios, pueden apartar y obtener aquellos productos que se encuentren disponibles en la aplicación.

Aplicación solidaria

Correo:

Contraseña:

Tipo de usuario:

☐ Donador ocasional ☐ Empresa ☐ Beneficiario

Login **Sign Up**

Si no te has unido haz clic aquí:

Menú inicial de logueo y registro

Productos disponibles

Producto	Tipo de producto	Fecha de vencimiento	Email donador	Cantidad disponible

Ventana en la que los beneficiarios podrán ver los productos disponibles

Donar un nuevo producto

Nombre del producto:

Tipo de producto:

Fecha de vencimiento:

Cantidad:

Guardar **Borrar todo**

Ventana en la que los donadores podrán ingresar un nuevo producto

VI. ENTORNOS DE DESARROLLO Y DE OPERACIÓN

En esta sección se debe describir el entorno en el que se desarrollará el software, así como el entorno en el que funcionará cuando se ponga en operación, para esto último se debe especificar el sistema operativo y el hardware sobre el que operará.

Para la creación de este aplicativo se hace uso del lenguaje de programación Java y el entorno de desarrollo NetBeans, en cuanto a la base de datos se hace manejo de ficheros por medio de archivos .txt y .json que permiten mantener la información almacenada.

Al poner en marcha la aplicación es necesario que el dispositivo del usuario cuente mínimo con 2Gb de memoria

RAM y un procesador de doble núcleo para que el programa pueda funcionar de manera óptima.

VII. PROTOTIPO DE SOFTWARE INICIAL

El primer prototipo de software del proyecto funciona por medio de la consola, en la cual se pueden realizar inserciones, búsquedas, entre otros.

Se usa el software de Github para llevar a cabo un control de versiones y avances, por medio de este repositorio se logra acceder remotamente al proyecto por lo que facilita el trabajo en grupo, a su vez, permite mantener una organización tanto en el entorno de desarrollo como en el trabajo como equipo. Se hace uso de las ramas para evitar sobreescrituras de código o equivocaciones en las versiones realizadas.

En cuanto a la versión gráfica de desarrollo del repositorio se usaron herramientas como github desktop y gitkraken, las cuales permiten una mejor interacción con el repositorio y un mejor entendimiento de las versiones realizadas.

El repositorio del proyecto se encuentra en el siguiente link: <https://github.com/ACorduz/ProyectoED.git>

VIII. IMPLEMENTACIÓN Y APLICACIÓN DE LAS ESTRUCTURAS DE DATOS

Para la implementación de las estructuras de datos en este proyecto se hizo uso de Colas, listas enlazadas, pilas y listas doblemente enlazadas.

La implementación de la cola se hace para llevar un orden de ingreso de los beneficiarios, dando prioridad a aquellos beneficiarios que ingresaron primero a la aplicación y así puedan obtener primero los productos disponibles, permitiendo que las donaciones sean recibidas con un orden establecido en los beneficiarios.

Se instancian dos listas simplemente enlazada para almacenar a los beneficiarios y a los donadores, se realizan de forma separada para mantener el orden de los roles creados en el proyecto. La lista enlazada es usada ya que permite un almacenamiento dinámico de datos, permitiendo aumentar y disminuir su tamaño según se requiera, por lo que nos permite almacenar grandes cantidades de beneficiarios y donadores, y a su vez mantenerlos organizados.

La implementación principal de la lista doblemente enlazada es para el almacenamiento de los productos que van a ser donados, por medio de esta estructura de datos nos permite hacer eliminación e inserción de un elemento de forma más eficaz; a su vez nos permite recorrer los productos almacenados en dos direcciones por lo que la búsqueda de un elemento es más rápida.

La pila es usada en la clase donador que se extiende de usuario, su principal función es almacenar las donaciones que ha hecho un donador, al ser implementada esta funcionalidad con una pila se está haciendo un historial de donaciones hechas por lo que un usuario puede acceder a esta pila y saber las últimas donaciones que ha hecho.

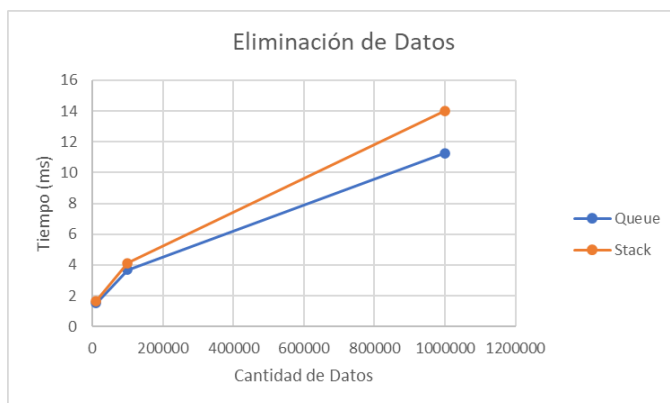
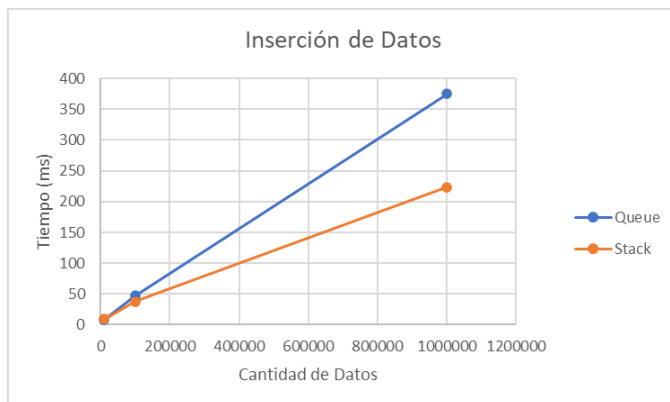
Estas 4 estructuras de datos implementadas en el proyecto son creadas desde cero con sus respectivos atributos y métodos, y no se hace uso de las estructuras por defecto, cada estructura cuenta con las operaciones funcionales de lectura, actualización, actualización y eliminación. .

IX. PRUEBAS DEL PROTOTIPO Y ANÁLISIS COMPARATIVO

• Registro Usuarios

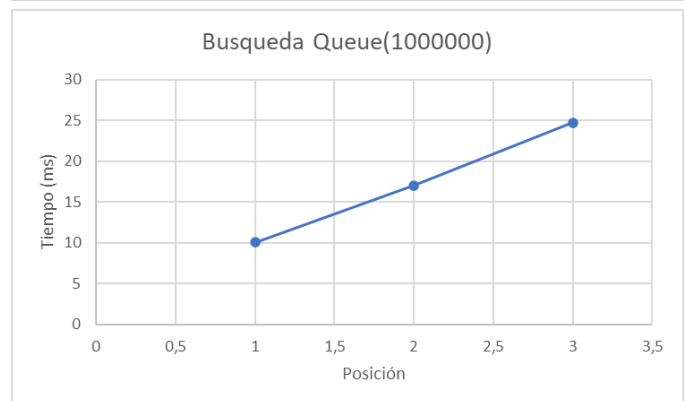
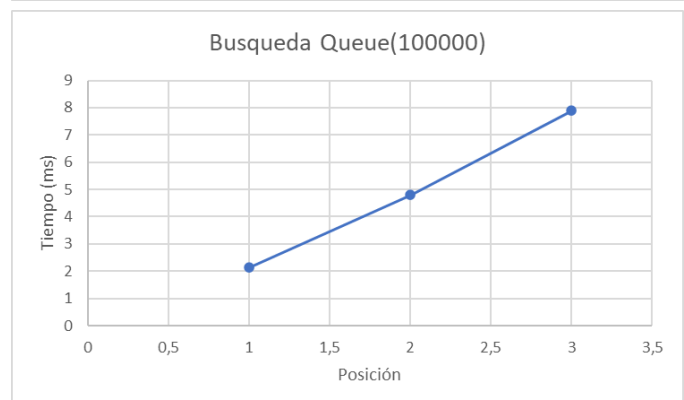
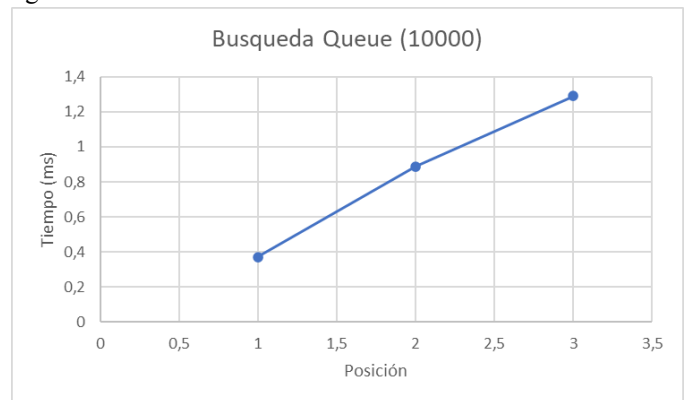
Las pruebas para esta funcionalidad se realizaron en un equipo con un procesador Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz y 8GB de memoria RAM.

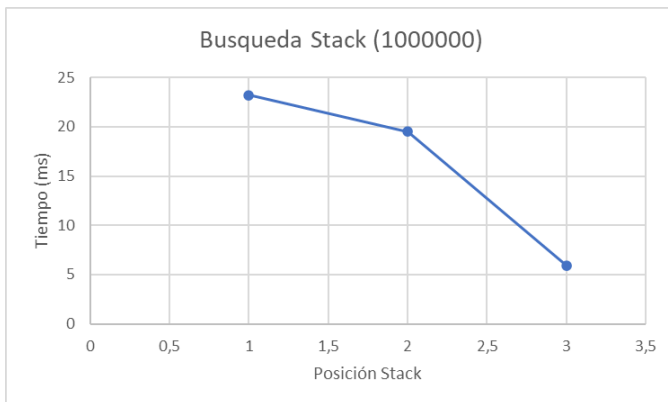
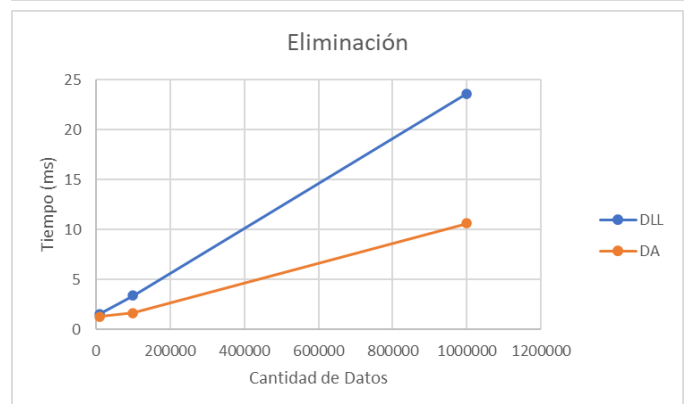
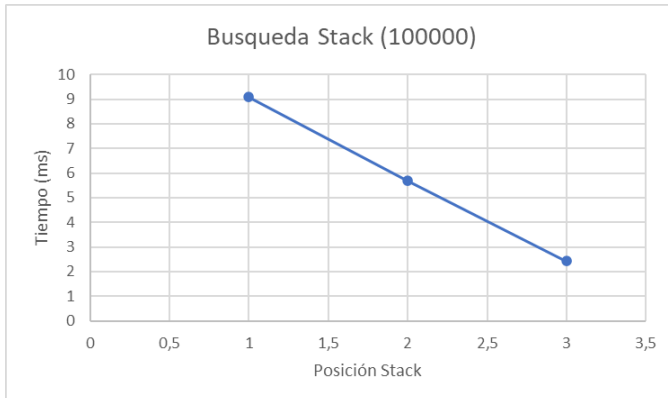
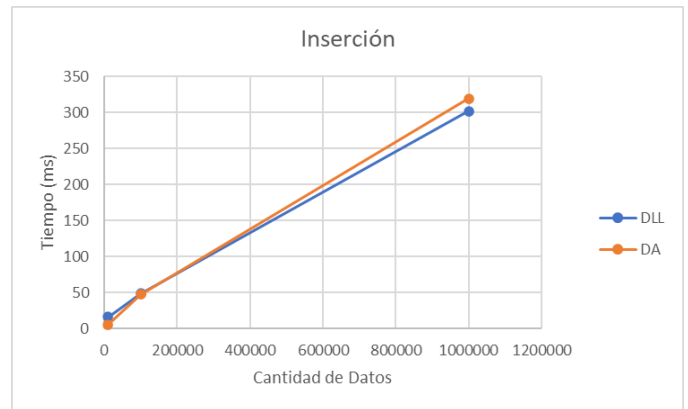
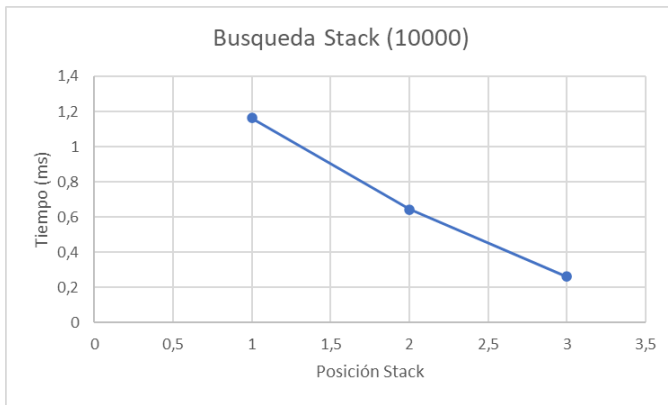
Para la creación de usuarios se obtuvieron los siguientes resultados:



Para ambas estructuras de datos se obtiene una complejidad de $O(n)$ para un conjunto de datos n . La diferencia está en que la que la pila es más rápida en la inserción masiva de datos, pero la cola es más eficiente en la eliminación.

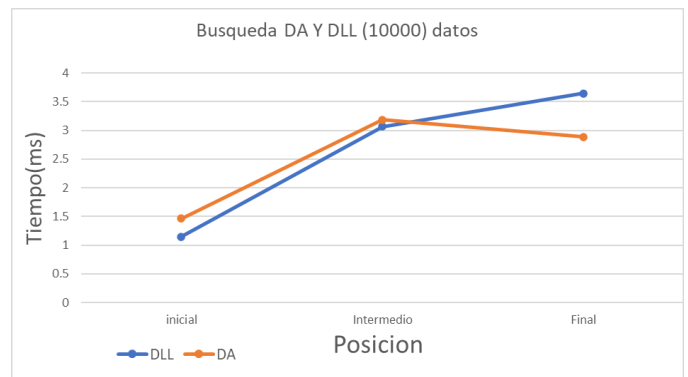
Respecto a la operación funcional de búsqueda se tienen los siguientes resultados:





En ambos casos se observa un comportamiento con complejidad $O(n)$, la principal diferencia radica en la operación funcional de la eliminación, ya que la eliminación es más rápida en el arreglo dinámico a comparación de la Lista Doblemente Enlazada.

Referente a la operación funcional de búsqueda se obtuvieron los siguientes resultados:

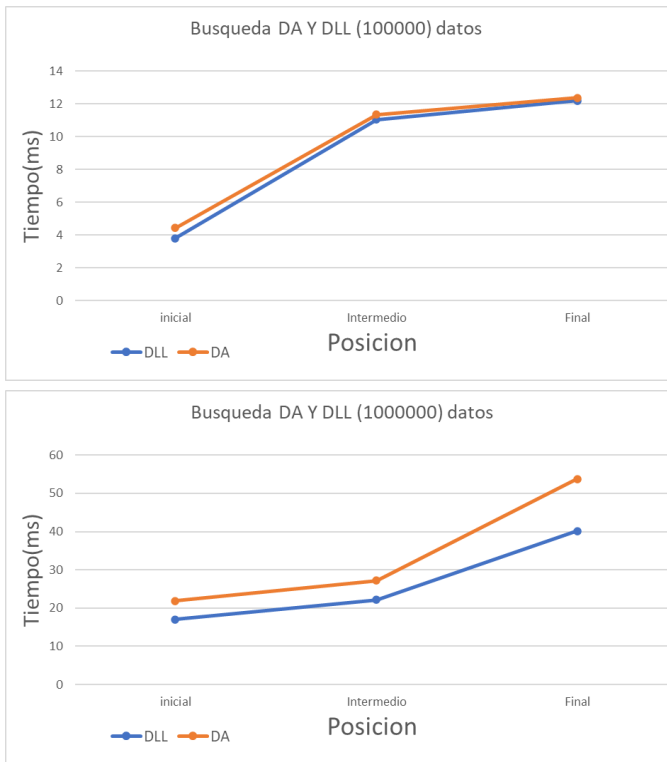


En ambos casos se obtiene una complejidad $O(n)$ que depende de la cantidad n del conjunto de datos, pero la diferencia está en si se sabe la ubicación de los datos que se van a buscar más frecuentemente, es decir, si la mayoría de los datos que se va a buscar son insertados al principio, se debe usar la estructura de datos Queue ya que los datos se encontrarán al principio de la estructura de datos, en caso contrario, se deberá usar pila debido a que al principio de la pila se encontrarán los datos insertados al final.

● Registro Producto

Las pruebas para esta funcionalidad se realizaron en un equipo con un procesador Intel(R) Core(TM) i5 de 11 generación con 8GB de memoria RAM.

Para las operaciones funcionales de inserción y eliminación se obtuvieron los siguientes resultados:



Para todos los casos de búsqueda se asemeja un comportamiento $O(n)$, debido a que para encontrar el elemento se debe recorrer toda la estructura de datos lineal, por lo que los menores tiempos de búsqueda se encuentran en los elementos del principio, y los peores tiempos se encuentran al final de la estructura de datos, tanto para Lista doblemente enlazada como para arreglo dinámico.

X. ROLES Y ACTIVIDADES

ROL	Actividades fundamentales
Líder(esa)	Consultar a los otros miembros del equipo, atento que la información sea constante para todos. Aportar con la organización y plan de trabajo.
Coordinador(a)	Mantener el contacto entre todos, Programar y agendar reuniones; ser facilitador para el acceso a los recursos.
Experto(a)	Líder técnico que propende por coordinar funciones y actividades operativas.
Investigador(a)	Consultar otras fuentes. Propender por resolver inquietudes comunes para todo el equipo.
Observador(a)	Siempre está atento en el desarrollo del proyecto y aporta en el momento

	apropiado cuando se requiera apoyo adicional por parte del equipo.
Animador(a)	Energía positiva, motivador en el grupo.
Secretario(a)	Se convierte en un facilitador de la comunicación en el grupo. Documentación (actas) de los acuerdos/compromisos realizados en las reuniones del equipo.
Técnico(a)	Aporta técnicamente en el desarrollo del proyecto.

Puede utilizar la siguiente tabla para definir los integrantes del grupo, los roles asignados y el listado de actividades durante el desarrollo de la entrega:

Los roles se repartieron equitativamente entre los integrantes, cada uno llevando a cabo sus funcionalidades pero a su vez dispuestos a realizar cualquier rol que sea necesario en el momento.

- **Jhoan Franco: Experto - Técnico**
 - Definir el rol y los trabajos destinados para cada uno de los participantes del grupo de trabajo.
 - Desarrollo y apoyo en la programación del proyecto.
- **Diego Alvarez: Líder - Animador**
 - Mantener el orden y comunicación entre los integrantes
 - Impulsar e incentivar al grupo de trabajo para llevar a cabo un mejor trabajo.
- **Camilo Orduz: Coordinador - Observador**
 - Organizador y facilitador de reuniones entre el grupo, a su vez, proponer fechas de entrega y plazos de trabajo.
 - Atento ante cualquier problema o necesidad en la que se requiera apoyo.
- **Carlos Reyes: Investigador - Secretario**
 - Indagar sobre inquietudes que presenten los miembros del equipo.
 - Redacción de acuerdos y documentos entre el grupo.

XI. DIFICULTADES Y LECCIONES APRENDIDAS

Una de las principales dificultades que se presentó al momento de realizar el proyecto fue la selección de la estructura de datos adecuada para cada caso en específico, ya que se deben analizar variables externas como el manejo de memoria, el manejo del tiempo o incluso la complejidad de cada una de los algoritmos implementados; por otro lado, también se encuentran las dificultades que se presentan en la escalabilidad de los datos, ya que, si bien algún algoritmo puede funcionar bien para una baja cantidad de datos, puede que al aumentar la cantidad de datos que funcionan en el algoritmo deje de funcionar de la misma manera en la que lo hacía con pocos datos.

En cuanto a las lecciones aprendidas se pueden recalcar varias entre ellas están: el trabajo en grupo, ya que se debe aprender a trabajar con compañeros de forma organizada, responsable y respetuosa; implementación de estructuras, permite aplicar lo aprendido en clase de una manera más práctica y a una escala más grande, a su vez conocer más a fondo la construcción y uso de cada una de las estructuras implementadas; persistencia, si bien las estructuras nos permiten almacenar información de manera organizada, la persistencia de datos también es importante, ya que, nos permite conservar información del programa para un uso posterior o incluso recuperar datos que han sido suprimidos durante la ejecución del programa.

En conclusión, este proyecto ha permitido poner en práctica tanto nuestro conocimiento como nuestras habilidades blandas y reunir ambas en un solo trabajo para aprender y mejorar de nuestros aciertos y desaciertos..

XII. REFERENCIAS BIBLIOGRÁFICAS

- [1] Instituto de Estudios Urbanos – IEU (2021, mayo 24). Mientras 2.7 millones de colombianos sufren hambre, 10 millones de toneladas de alimentos se desperdician anualmente [Online]. Disponible: <http://ieu.unal.edu.co/medios/noticias-del-ieu/item/mientras-2-7-millones-de-colombianos-sufren-hambre-10-millones-de-toneladas-de-alimentos-se-desperdician-anualmente>
- [2] Departamento Nacional de Planeación (2016, abril) . PÉRDIDA Y DESPERDICIO DE ALIMENTOS EN COLOMBIA [Online]. Disponible : https://sinergia.dnp.gov.co/Documentos%20de%20Interes/Perdida_y_Desperdicio_de_Alimentos_en_colombia.pdf
- [3] Departamento Nacional de Planeación (2016, abril) . PÉRDIDA Y DESPERDICIO DE ALIMENTOS EN COLOMBIA [Online]. Disponible : https://sinergia.dnp.gov.co/Documentos%20de%20Interes/Perdida_y_Desperdicio_de_Alimentos_en_colombia.pdf