

Economerk

**Pinzón García Miguel Angel, Olaya Arias Pablo
Esteban, Rojas Luna Juan Manuel, Quintero Castillo
Carlos Fernando, Becerra Pulido Julian Santiago**

No. de Equipo Trabajo: {6}

I. INTRODUCCIÓN

Desde factores como la pandemia, la guerra en ucrania, la crisis de los contenedores, la reactivación económica y las fluctuaciones en los mercados de valores, ha desencadenado una inflación y un encarecimiento del costo de vida en todo el mundo, generando una crisis, especialmente en el alza de los precios de la canasta familiar.

Una manera de contener la crisis económica de los hogares colombianos en el corto plazo en la adquisición de productos de la canasta familiar, es la búsqueda de ofertas que brinde al consumidor final el producto con el menor precio.

Por eso, desde una aplicación, se busca crear una herramienta de selección para los usuarios, donde tendrán al alcance las diferentes variedades en precios de una amplia gama de productos, destacando los almacenes que presentan la mejor oferta.

Gracias al desarrollo de software y las estructuras de datos, se facilitará la toma de decisiones en la búsqueda de los productos con bajo costo. Adicionalmente, los usuarios tendrán acceso a las diferentes ofertas que brindarán los almacenes de manera directa.

II. DESCRIPCIÓN DEL PROBLEMA A RESOLVER

Se implementará una aplicación que reciba la información de los productos que el usuario desea cotizar, luego, ejecutará la búsqueda de los almacenes que ofertan el producto, almacenando estos datos, y luego organizando de menor a mayor precio, para que el cliente pueda ver los almacenes con las mejores ofertas en los productos solicitados, dando una respuesta completa y detallada con el mejor precio, todo desde la comodidad de un celular.

En la primera fase de desarrollo del proyecto se realizará el análisis de comparación en la implementación de las diferentes estructuras de datos valoradas en la entrega, únicamente en la inserción de los productos a ofertar por los diferentes almacenes.

En la fase dos del proyecto se ejecutarán las pruebas de inserción y búsqueda del catálogo de los productos de los

almacenes que ofertarán en la plataforma para poder determinar qué estructura realizará estos procesos de la manera más eficiente, teniendo en cuenta que la inserción se efectuará con la lectura de un documento .txt. Adicionalmente se iniciará la interfaz gráfica de las funcionalidades del usuario, que se analizaron en la primera fase del proyecto.

III. USUARIOS DEL PRODUCTO DE SOFTWARE

La aplicación está dirigida para todas las personas que busquen reducir sus costos en la compra de productos con el mejor precio, especialmente las familias que se encuentren en crisis económica.

IV. TIENDAS DEL PRODUCTO DE SOFTWARE

La aplicación está dirigida para todas las tiendas que deseen ofertar su catálogo de productos de la canasta familiar, brindando los precios actualizados de sus productos, lo que permitirá a los usuarios comparar y escoger las tiendas con la mejor opción de sus productos a comprar.

V. REQUERIMIENTOS FUNCIONALES DEL SOFTWARE

- *Ingreso del nombre del producto a comparar*

El usuario se encontrará con una interfaz en la cual se le pedirá que ingrese el nombre del producto o productos de los cuales quiere saber en qué tienda de las que están disponibles se encuentra el producto a un precio más económico.

Acciones iniciadas y comportamiento esperado:

El usuario después de ingresar a la aplicación llegará al home donde habrá un espacio en el cual dirá "Ingrese el nombre del producto que desea buscar", donde el usuario podrá añadir el producto, adicionalmente, en la parte inferior están los botones "Agregar" y "Eliminar" para poder adicionar o eliminar nuevos producto de la lista. Se espera que salga una lista en la que se mostrará el o los productos de las tiendas ordenados de menor a mayor precio.

- *Comparar productos entre tienda*

La aplicación buscará el producto que el usuario digitó entre las diferentes tiendas disponibles.

Acciones iniciadas y comportamiento.

Cuando el usuario da clic en el botón "buscar", la aplicación iniciará la búsqueda tienda por tienda, de cada uno de los productos agregados por el usuario; el

orden de búsqueda se ejecutará en el mismo orden de ingreso de los productos.

- **Guardar productos**

En este proceso los usuarios pueden agregar productos al catálogo de su tienda.

Acciones iniciadas y comportamiento.

Los usuarios dispondrán de una ventana en la cual puedan ver la lista de productos de sus respectivas tiendas. En esta ventana encontrarán un botón que diga “Agregar producto”.

- **Eliminar productos**

El usuario puede eliminar productos de la lista que desea buscar.

Acciones iniciadas y comportamiento.

Antes de realizar la búsqueda, el usuario tiene la opción para eliminar productos de la lista de productos a comparar, únicamente debe ingresar el nombre del producto en la ventana de texto y dar clic en la opción “Eliminar”.

- **Buscar**

Desde el botón “buscar” iniciará la búsqueda y comparación de los productos ingresados por el usuario.

Acciones iniciadas y comportamiento.

Desde el momento en que el usuario da clic en el botón “buscar”, la aplicación buscará las tiendas que ofertan cada uno de los productos ingresados por el usuario, la cual se realizará en cola, iniciando con el primer producto ingresado. Se fijará el precio de referencia con la primera tienda que contenga el producto a validar; y luego se actualizará el precio a comparar cuando se encuentre una tienda que contenga el producto y un precio inferior al de referencia.

- **Mostrar lista de productos**

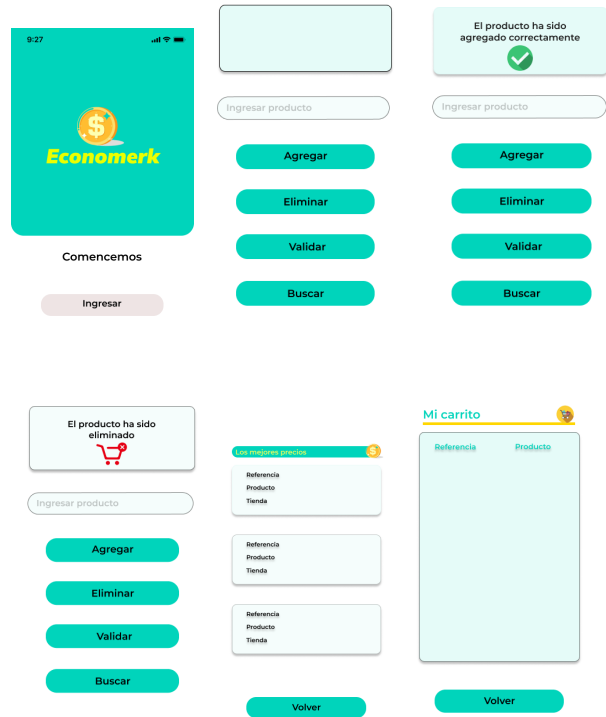
La aplicación muestra los productos que el usuario buscó ordenado de menor a mayor precio con su respectiva tienda.

Acciones iniciadas y comportamiento.

El usuario previamente habrá digitado el nombre del producto que deseaba buscar y posteriormente pulsar el botón “Buscar”. Luego espera que salga una lista en la que se mostrará el o los productos de las tiendas ordenados de menor a mayor precio.

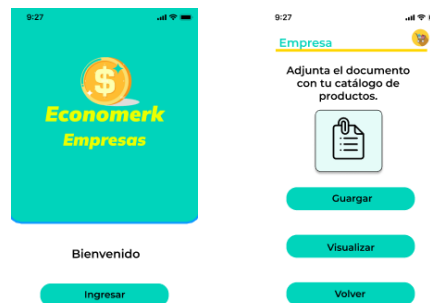
VI. DESCRIPCIÓN DEL PROTOTIPO DE SOFTWARE

Para realizar la parte gráfica de la aplicación decidimos optar por figma:



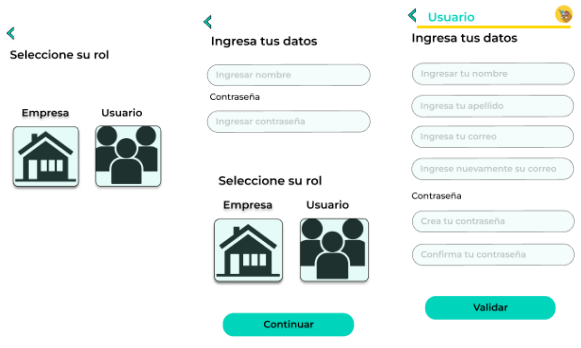
Para la fase 1 se desarrollaron los mockups para la interfaz de usuario, donde encontrará las opciones de navegación de nuestra plataforma.

Para la fase 2, nos concentramos en el diseño de la interfaz de los almacenes, donde podrás recorrer todas las opciones para el ingreso y actualización de su catálogo de productos.



!''#

Adicionalmente, se diseñó la sección de ingreso y registro de los usuarios y almacenes.



Las imágenes de forma interactiva están en el siguiente link:

<https://www.figma.com/file/O8hch7GB0gnbtZIEVB0HN0/Economerk?type=design&node-id=0%3A1&t=cqRRzgzPnJhe17tu-1>

Para la primera fase se realiza el diseño de la aplicación enfocada al usuario final, es decir, en las personas que ingresarán a consultar los precios ofertados de los diferentes productos en el catálogo de las tiendas.

Los clientes visualizarán en el inicio de la aplicación el nombre y logo de la programa y en la parte inferior un botón de “inicio” para que puedan ingresar al entorno donde se solicitarán los productos a comparar.

Dentro de la ventana principal, se encontrarán las diferentes funcionalidades para que el usuario pueda ingresar y eliminar los productos al carrito que luego se comparan.

La opción “Carrito” brindará al usuario la visual de los productos que han sido ingresados por el cliente.

Teniendo los productos en el carrito, el usuario podrá dar clic en la opción “Buscar” para que el programa realice la comparación de cada uno de los productos en las tiendas registradas en la plataforma, para que pueda encontrar que almacén tiene el precio más bajo en cada producto, proceso que será registrado en la ventana con el nombre “Los mejores Precios”; cada producto tendrá la referencia, el nombre, la tienda que lo oferta con el precio más competitivo y el valor.

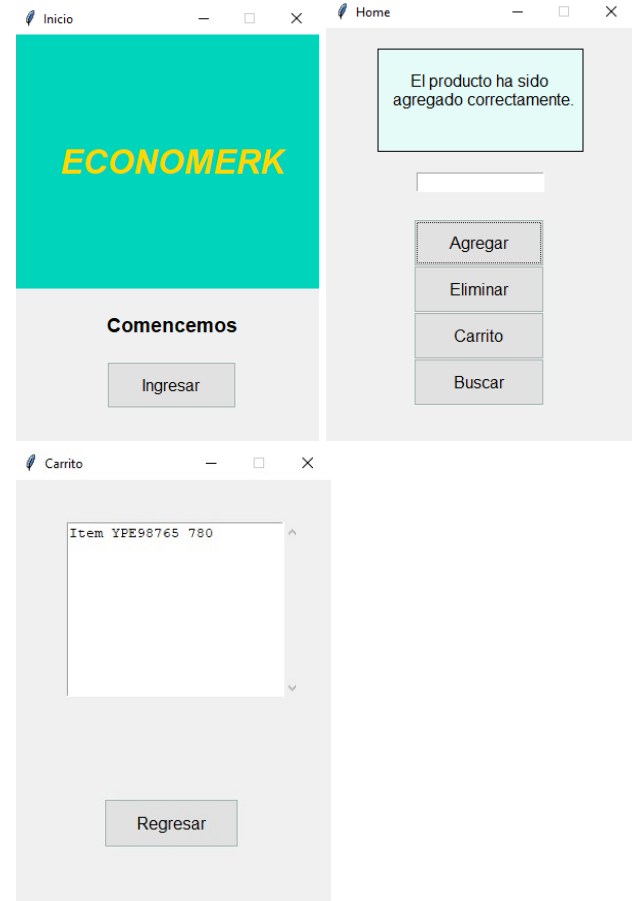
Por último, en la parte inferior se encontrará la opción “Volver” el cual habilitará la ventana principal, para que vuelva a ingresar nuevos productos.

VII. AVANCE EN LA IMPLEMENTACIÓN DE LA INTERFAZ DE USUARIO

Como avance en la implementación de la interfaz para la segunda entrega, realizamos los mockups de la primera entrega que está destinada a los usuarios que desean usar la aplicación para encontrar los mejores productos al mejor precio en las tiendas disponibles. Se adicionaron otros

mockups para empresas que quieran mostrar su productos en la aplicación, pero esto se realizará para la entrega final.

Las interfaces de la aplicación quedaron de la siguiente forma:



VIII. ENTORNOS DE DESARROLLO Y DE OPERACIÓN

El entorno de desarrollo por el cual nos decantamos para realizar la aplicación fue Python, así mismo también usaremos Replit para trabajar de manera colaborativa.

Para esta primera entrega decidimos que la aplicación funcionará en el sistema operativo de windows y el hardware en el que se usará será en computador.

En la fase inicial del programa las pruebas para la puesta en marcha del programa no presentará interfaz gráfica, aunque se tienen los diseños preliminares del programa, por funcionalidad la interfaz se implementará en la segunda entrega.

IX. ACCESO AL PROTOTIPO DE SOFTWARE

El prototipo de software inicial estará disponible en github y el pertinente enlace esta a continuación :

<https://github.com/N1ghtlyC0de/Proyecto-EEDD>

Además también se puede observar el código en replit:

<https://replit.com/@Carlos-Fernan26/Proyecto-EEDD#main.py>

Finalmente, incluimos el video demostrativo del prototipo:

<https://www.loom.com/share/971743e868ad4f6a8111f86d29b24de8>

Se escogen estos entornos, por facilidad de trabajo ya que permiten la edición de manera conjunta y la puesta en marcha de las actualizaciones que tendrá el programa en su etapa de desarrollo.

La aplicación en su primera entrega estará destinada al usuario final, el desarrollo de la de la plataforma para los almacenes se implementará para la segunda entrega.

El software se implementa principalmente para evaluar las estructuras ideales de inserción, eliminación, búsqueda y comparación de los datos que el cliente desea procesar.

X. IMPLEMENTACIÓN Y APLICACIÓN DE LAS ESTRUCTURAS DE DATOS

Para esta primera entrega hicimos pruebas usando las estructuras de datos tal como la Lista simple enlazada sin cola, Lista simplemente enlazada con cola, Lista doblemente enlazada sin cola, lista doblemente enlazada con cola y por último arreglos dinámicos.

El uso que le dimos a cada una de estos tipo de estructuras en el programa fue el de insertar datos que les pasamos por un .txt y así mismo realizar la búsqueda de datos en estos para poder realizar la pruebas de eficiencia con una gran cantidad de datos y probar cuál estructura es la que mejor nos funciona para realizar estas operaciones en nuestro proyecto.

Para la fase 2 el análisis se efectuó en búsqueda e inserción de catálogos de los productos, para determinar qué estructura tendrá el mayor rendimiento, para ello la valoración se generó desarrollando estas funcionalidades con la estructura de datos de árboles AVL y arreglos dinámicos.

Por último la fase 3 evaluará la interconexión de las estructuras de manera modular.

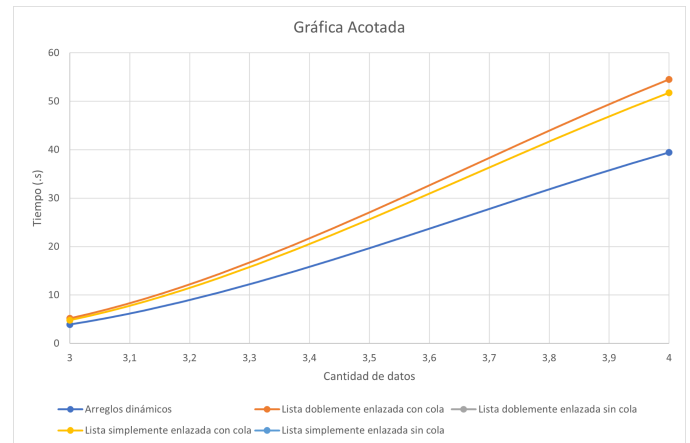
XI. PRUEBAS DEL PROTOTIPO Y ANÁLISIS COMPARATIVO

Para esta fase las pruebas desarrolladas consisten en la búsqueda, comparación y visualización de resultados. Realizados conjuntamente y analizando sus tiempos de ejecución

Tipo de estructura	Cantidad de datos	Tiempo 1	Tiempo 2	Tiempo 3	Tiempo 4	Mediana
Lista simplemente enlazada sin cola	10	2,78 0027 39	2,779 57296 4	2,846 49062 2	2,8214 00881	2,80 6873 0
	100	329, 8367 474	324,4 88231 7	316,2 47731 2	329,10 32906	324, 9190 002
	1000	TIME EXCEPTION :C				
	10000					
Lista simplemente enlazada con cola	10	0,03 0014 0	0,030 9660	0,029 9678	0,0301 716	0,03 0279 9
	100	0,50 7571 2	0,398 0372	0,412 7643	0,4080 906	0,43 1615 8
	1000	6,15 2656 6	4,281 4648	4,540 6497	4,2942 171	4,81 7247 0
	10000	58,5 3602 36	53,20 06586	47,72 74270	47,457 7844	51,7 3047 34
Lista doblemente enlazada sin cola	10	2,82 6107 74	2,660 37511 8	2,551 19562 1	2,5335 26421	2,64 2801 2
	100	312, 8090 746	328,4 72086	293,8 25830 5	293,54 09198	307, 1619 777
	!!"#	TIME EXCEPTION :C				
	10000					
Lista doblemente enlazada con	10	0,03 6867 1417 2	0,035 59803 963	0,034 88373 756	0,0318 965911 9	0,03 4811 4
	100	0,69 9032	0,536 04650	0,463 16742	0,4833 059311	0,54 5388

cola		7835	5	9		2
	1000	5,12 9103 422	5,079 95295 5	5,237 92457 6	5,2668 65492	5,17 8461 6
	1000 0	59,7 0358 419	53,43 58260 6	52,55 47800 1	52,342 48877	54,5 0916 98
Arreg Los dinám icos	10	0,05 2334 0702 1	0,046 77033 424	0,046 32949 829	0,0479 307174 7	0,04 8341 2
	100	0,51 2592 7925	0,517 02475 55	0,446 80953 03	0,5345 63303	0,50 2747 6
	1000	3,80 6548 834	4,036 89670 6	3,950 71601 9	3,8130 42402	3,90 1801 0
	1000 0	38,9 9545 145	38,36 62750 7	41,82 86798	38,458 7388	39,4 1228 63

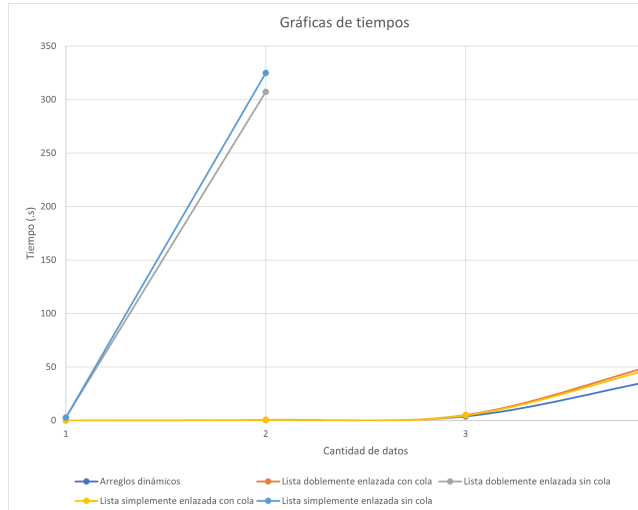
*Tiempo dado en segundos
 *Datos en miles
 *Se realizó una búsqueda de 10 datos



En ella apreciamos los comportamientos de los tipos de estructuras. Tanto en “Listas simplemente enlazadas con cola” como en “Listas doblemente enlazadas con cola” se aprecian comportamientos presuntamente lineal-logarítmico $O(n \log(n))$. Mientras que en los Arreglos Dinámicos, se evidencia un comportamiento presumiblemente lineal $O(n)$. Lo que apoya la idea de ser la estructura de datos más idónea²

En la segunda entrega se hizo una comparativa entre arreglos dinámicos, los cuales en tiempos de prueba de la primera entrega obtuvimos los mejores resultados, y árboles AVL evaluando sus tiempo de búsqueda así como el tiempo que tardaban en insertar un dato. Evaluando sus tiempo obtuvimos los siguientes resultados:

Donde se evidencia que el tipo de estructura más eficiente y por tanto más idónea, son los arreglos dinámicos, esto en la tarea con mayor complejidad



Gráfica 1.

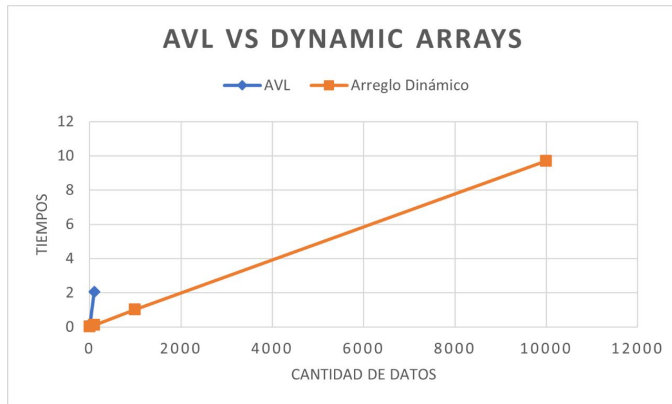
En tanto a la gráfica 1, se observa una amplia diferencia entre los tiempos de un tipo de estructura y de otra, marcando comportamientos exponenciales ($O(2^n)$) en dos de ellas “Listas simplemente enlazadas sin cola” y “Listas doblemente enlazadas sin cola”. Para analizar las otras estructuras, acotamos la gráfica a la región de interés.

Tiempos de inserción AVL				
Número de elementos en miles	Tiempo 1	Tiempo 2	Tiempo 3	Promedio
10	0,1998474598	0,1673591137	0,191519022	0,1862418652
100	2,023566961	2,076426744	2,076426744	2,058806817
1000	ERROR EN CREACIÓN DE NODOS			
10000				
Tiempos de búsqueda en segundos				

² Los valores en los ejes X de las gráficas corresponden al número de datos siendo 1, 2, 3 y 4 los equivalentes a 10, 100, 1000 y 10000 (datos en miles) respectivamente.

Tiempos de inserción Arreglo dinámico				
Número de elementos en miles	Tiempo 1	Tiempo 2	Tiempo 3	Promedio
10	0,009598255157	0,01113939285	0,01163053513	0,01078939438
100	0,09792780876	0,09299206734	0,1125338078	0,101151228
1000	0,9669082165	0,9589958191	1,122968197	1,016290744
10000	9,469784737	9,904289007	9,716901541	9,696991762
Tiempos de búsqueda en segundos				

Para poder tener una mejor noción y compara los resultado se realizó la una gráfica comparando los resultados de las estructuras de datos:



En la gráfica podemos observar que en cuanto a la inserción de entre los arreglos dinámicos y los árboles AVL es lo árboles parece tener un comportamiento exponencial debido a su alta pendiente y además que no completo la inserción de elementos a partir del millón de datos: Por otra parte en en los arreglos dinámico la inserción de elementos parece tener un comportamiento lineal $O(n)$ y además pudimos completar todas la pruebas.

Cabe destacar que en la búsqueda de un elemento en los árboles AVL tiene un comportamiento aparentemente constante, la cual resulta se una mejoría en cuanto a los tiempos de búsqueda.

XII. ROLES Y ACTIVIDADES

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

		REALIZADAS
Olaya Arias Pablo Esteban	Técnico	Realización del software. Montar el proyecto en github. Backend.
Pinzón Garcia Miguel Angel	Técnico Animador	Documento técnico. Mockups. Frontend.
Quintero Castillo Carlos Fernando	Técnico	Realización del software. Video. Backend.
Rojas Luna Juan Manuel	Líder	Documento técnico. Mockups. Frontend
Becerra Pulido Julian Santiago	Técnico.	Documento Interfaz Gráfica. Frontend

XIII. DIFICULTADES Y LECCIONES APRENDIDAS

Con respecto a la planeación de la aplicación, se tuvieron leves dificultades en el proceso de abstracción para que esta pudiera ser solucionada mediante estructuras de datos. Después de realizar un consenso entre los integrantes, decidimos hacer uso de 'números seriales' para representar los productos que cada tienda tenía disponibles dando a cada producto y almacén una referencia dentro de la plataforma, con lo que logramos exitosamente darle un modelo a la entrada del programa para su consecuente procesamiento y ejecución. Luego, nos enfrentamos al problema sobre cómo procesar las entradas usando las estructuras de datos, cómo lo son los arreglos dinámicos, las listas enlazadas, las colas y las pilas. Entre todas las estructuras, tuvimos una mayor parte de tiempo dedicada a la implementación de los arreglos dinámicos y las listas enlazadas, esto principalmente dado a que fueron las primeras estructuras desarrolladas para el proyecto, por lo que estábamos comprendiendo cómo otorgarle a cada elemento de la entrada un formato que satisficiera la estructura a usar, ya sea mediante índices o punteros. Una vez desarrollados, no tuvimos dificultades realizando la implementación para las estructuras restantes.

Terminada la fase de desarrollo, continuamos con la fase de pruebas con diversas cantidades de datos, donde probamos y determinamos cuales estructuras de datos eran más eficientes

y descubrimos que una de ellas no daban una salida dado a una limitación de poder de procesamiento, siendo esta la lista doblemente enlazada sin cola. Dado a que este problema consiste en una limitación técnica, no logramos hallar el tiempo de ejecución para esta estructura.

Con este avance, logramos adquirir experiencia en el desarrollo de aplicaciones destinadas a solucionar problemas de la vida cotidiana y juzgar la eficiencia de un programa de software, usando la relación cantidad/tiempo para precisar su tiempo de ejecución. En cuanto a la segunda entrega del proyecto aprendimos a aplicar una nueva de estructura de datos la cuales son los árboles AVL.

XIV. REFERENCIAS BIBLIOGRÁFICAS

- [1] Streib, J. T., & Soma, T. (2018). *Guide to Data Structures: A Concise Introduction Using Java*. Springer International Publishing.
- [2] Hernández, Z.J. y otros: *Fundamentos de Estructuras de Datos. Soluciones en Ada, Java y C++*, Thomson, 2005.
- [3] Shaffer, Clifford A.: *Data Structures and Algorithm Analysis in C++*, Third Edition, Dover Publications, 2013. (En línea.)
- [4] Campos Laclaustra, J.: *Apuntes de Estructuras de Datos y Algoritmos*, segunda edición, 2018. (En línea.)
- [5] Martí Oliet, N., Ortega Mallén, Y., Verdejo López, J.A.: *Estructuras de datos y métodos algorítmicos: 213 ejercicios resueltos*. 2ª Edición, Ed. Garceta, 2013.
- [6] Joyanes, L., Zahonero, I., Fernández, M. y Sánchez, L.: *Estructura de datos. Libro de problemas*, McGraw Hill, 1999.
- [7] J. Doe, "Introduction to Data Structures," Coursera, 1 enero 2020. [En línea]. Disponible en: <https://www.coursera.org/learn/data-structures>