Plataforma Colaborativa ECO-LOMBIA

Felipe Gutiérrez, Daniel Perez, Alejandro Camargo

No. grupo del curso: 3 y No. de Equipo de Trabajo: 4

I. [1] INTRODUCCIÓN

El proyecto ECO-LOMBIA se basa en una plataforma colaborativa que alimenta una base de datos sobre la gran riqueza biológica que tiene nuestro país. Destinada para el público en general, pero en especial para los amantes de la biodiversidad en Colombia y los interesados a nivel internacional sobre el tema. En las siguientes secciones se hacen las especificaciones sobre el proyecto. El énfasis son las estructuras de datos que pueden ser utilizadas para manejar en memoria los datos y dar resultado a los requerimientos de la aplicación.

I. DESCRIPCIÓN DEL PROBLEMA

Colombia es el segundo país más biodiverso del mundo, cuenta con 56.343 especies [1], sin embargo, el desconocimiento de la mayoría de la población es motor de su deterioro como lo enuncia Liliana Vélez en un Histórico del Colombiano[2] el . Esto tiene diversos factores, pero el principal es el poco acceso a la información. El objetivo de ECO-LOMBIA es facilitar ese acceso a la mayoría de la población colombiana, en especial a quienes cohabitan con esta biodiversidad e ignoran su importancia biológica. Ecolombia por otro lado busca informar sobre los diversos parques ecológicos que existen en la nación, para fomentar su visita, mejora y preservación. por lo anterior el objetivo de este proyecto es desarrollar una aplicación de escritorio que permita a los colaboradores añadir contenido y a los usuarios consultar el contenido de especies y parques.

II. USUARIOS DEL PRODUCTO DE SOFTWARE

Al ser una red colaborativa, sus roles no son tan específicos. Pero los principales tipos de usuarios son:

A. Investigadores colaboradores: Son las personas que tienen acceso a las bases de datos, ellos agregan y verifican la información que estará en la plataforma. Son los encargados de agregar especies de fauna y flora y Ecoparques. Son profesionales pertenecientes a instituciones públicas o privadas que se relacionen con la biodiversidad Colombiana.

B. Visitantes del sitio: Personas que tienen acceso a observar la información, más no modificarla, son la población objetivo, por quienes se hace esta plataforma.

III. REQUERIMIENTOS FUNCIONALES DEL SOFTWARE

A. Inicio de sesión de Usuarios del sistema

Descripción: Procedimiento en el cual un usuario no inicia sesion en la aplicación para acceder a su perfil y realizar operaciones de aporte o edición de sus aportes.

Acciones iniciadoras y comportamiento esperado:

Para iniciar sesión el usuario puede acceder a la ventana de iniciar sesión e ingresar su número de identificación y contraseña para acceder a su perfil mediante el clic una vez finalizado el ingreso de datos. el software debe buscar por el número de identificación al usuario que se desea consultar para obtener los datos y compararlos con los ingresados para lograr el procedimiento.

- *Cambio de contraseña de Usuarios del sistema*

- *Descripción:* Procedimiento en el cual un usuario cambia su contraseña con una actual ya sea temporal o la suya propia .

- *Acciones iniciadoras y comportamiento esperado*:

Una vez en su perfil el usuario puede cambiar su contraseña si lo desea mediante un clic en un enunciado resatado y una ventana emergente se abre para que con su contraseña actual podrá ingresar una nueva.

B. Actualización:

En el caso de actualizar la contraseña , una vez el usuario haya ejecutado el cambio debe garantizar mediante un cuadro de diálogo que se realizó de manera correcta y que internamente haya modificado este atributo del objeto usuario.

Si los datos ingresados en el registro no son validos, como por ejemplo que el numero de cedula no sea un entero se le indicará al al usuario mediante un cuadro de diálogo que un dato de los suministrados no es correcto para este lo cambie y pueda completar el procedimiento. Al igual al iniciar sesión puede que el número de identificación ingresado sea mal digitado y no concuerde con ningún usuario o digite incorrectamente la contraseña para lo cual el software mediante un cuadro de diálogo le indicará que el usuario con ese ID no está registrado o que la contraseña es incorrecta respectivamente indicando que intente de nuevo llenar los

[1]

campos y solicitar el acceso haciendo clic en el botón correspondiente.

Paginación de resultados de búsqueda:

- *Descripción:* Mostrar un número determinado de resultados en el canvas de la aplicación, organizando los datos por página

- *Acciones iniciadoras:* Al realizar una búsqueda, aparecerán los primeros n datos, n es el número de resultados por página y p, el número de datos por página, y al dar "siguiente" aparezcan los resultados del n+1 al resultado 2n, y al dar anterior, se muestran los datos del 1 al n.
  También se puede determinar la variable n, Siendo n=10 el valor por defecto, se puede cambiar a 15, a 20, o al valor que desee el usuario para mayor comodidad.
- Almacenamiento: Se necesita una estructura que almacene y ordene los resultados que están dentro de las bases de datos de la aplicación, para que sean impresos según lo indique el usuario.

Banner de Noticias

- *Descripción:* Muestra en la ventana de inicio noticias referentes al desarrollo ecológico dentro del territorio nacional y regional.

- *Acciones Iniciadoras:* Directamente cuando se abre la aplicación aparece el seccionario de noticias, tiene predeterminadas 5 objetos, a través de la interacción directa con los botones podemos avanzar y retroceder de noticia a noticia.
- Almacenamiento: Se requiere de una estructura que archive temporalmente las noticias, de tal manera que salgan de circulación cuando pierdan vigencia y existan archivos más recientes.

A. Operaciones de datos tipo Flora, Fauna o Ecoparque

Descripción: Es uno de las funcionalidades principales del software, permite al al ser un usuario registrado, contribuir al crecimiento de las bases de datos agregando, actualizando o eliminando elementos ya sea de flora, fauna o ecoparque.

Acciones iniciadoras y comportamiento esperado:
Para añadir un nuevo elemento, el usuario debe registrarse, y posteriormente seleccionar el tipo de elemento que quiere añadir, en ese momento el programa toma los datos insertados y los pone al final de el arreglo dinámico que contiene todos los objetos del elemento insertado.

Almacenamiento: Al momento de cerrar el programa, todos los elementos son insertados a un documento .txt de salida, donde se guardan los cambios hechos durante la ejecución del programa.

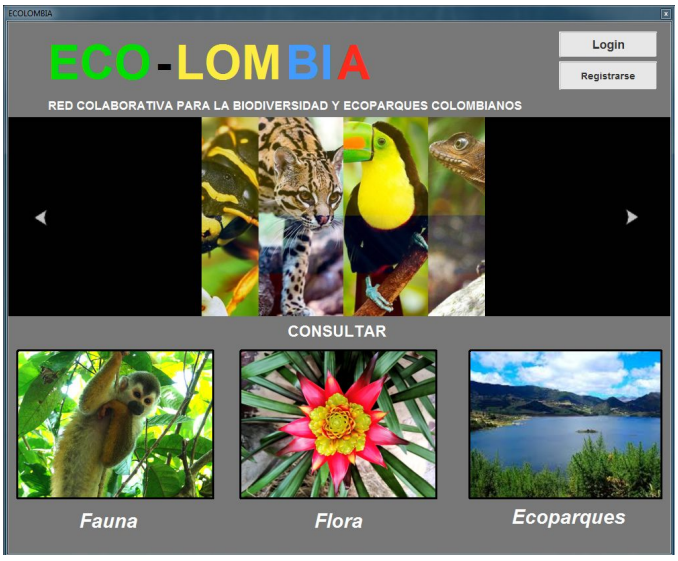
B. Búsqueda de datos

Descripción: Permite la búsqueda y visualización de datos dependiendo de un criterio específico, teniendo en cuenta las características de los diferentes objetos, ya sea flora, fauna o ecoparque.

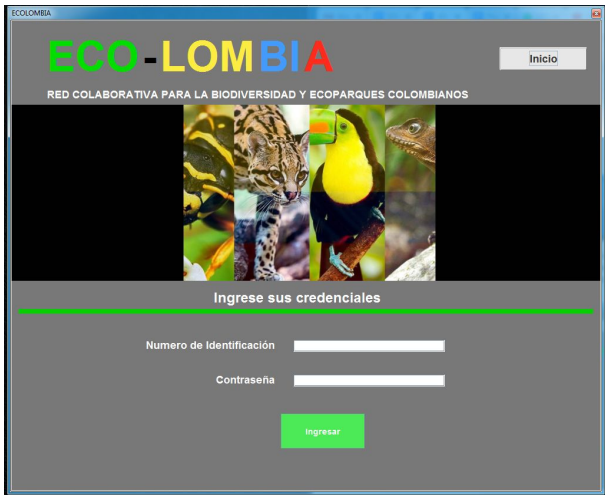
Acciones iniciadoras y comportamiento esperado:
Para la búsqueda de uno o varios elementos, basta con que el usuario se remita al inicio del programa y seleccione que desea buscar, si un elemento de flora, uno de fauna o uno de ecoparque, posteriormente debe seleccionar un criterio de búsqueda, y pasar un parámetro acorde al criterio elegido anteriormente, el programa debe buscar en todos los objetos del tipo de elemento seleccionado, y los que coincidan con el criterio de búsqueda los insertará a un arreglo dinámico y lo retornara, para su posterior paginación

IV. Descripción de la interfaz de Usuario

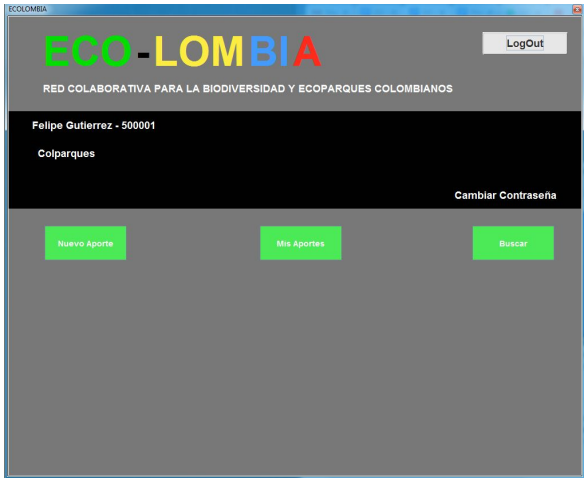
La aplicación de escritorio está pensada masivamente para usuarios no registrados de tal manera que su principal objetivo es buscar algún ser vivo o ecoparque de la base de datos, de tal manera en el inicio la apariencia es la siguiente.



un inicio donde puede iniciar sesión un usuario registrado o solicitar el registro en la aplicación mediante dos botones. la zona con fondo negro hace referencia a la función de visualización de noticias, y en las imágenes de abajo se accede a la búsqueda.



Anteriormente se visualiza las interfaces para el inicio de sesión y en caso de olvidar la contraseña. podrán ser visualizadas como anexo en caso de que desee verlo con más detalle.



En la anterior figura se puede observar la interfaz de perfil de cada usuario donde aparecen un par de datos del usuario, la posibilidad de cambiar su contraseña, hacer o modificar sus aportes y buscar contenido.

La anterior imagen muestra la visualización del perfil de una persona registrada, aquí permite visualizar y editar los aportes realizados por un usuario,y la opción para generar un aporte nuevo.

Siguiendo con la descripción de la interfaz en la figura anterior se puede visualizar las interfaces de filtrado y parámetros de búsqueda para la información de el animal, la planta o el ecoparque.

Finalmente la idea es mostrar los datos mediante tablas con una determinada cantidad de elementos para la visualización y un paginador que permite avanzar hacia más resultados.

V. Entornos de desarrollo y de operación

El proyecto se desarrolla en el IDE Netbeans, es una aplicación de escritorio y está configurada para operar en cualquier SO con java 8 o superior. El equipo debe tener requerimientos de memoria Ram dependiendo del número de datos que desee manejar.

VI. DESCRIPCIÓN DEL PROTOTIPO DE SOFTWARE

Link de gitHub: https://github.com/lfgutierrezg/Ecolombia

	O(n)		
10000	92	5	88
100000	355	20	393
1000000	6543	115	3122
10000000	42423	710	28820

VII. PRUEBAS DEL PROTOTIPO

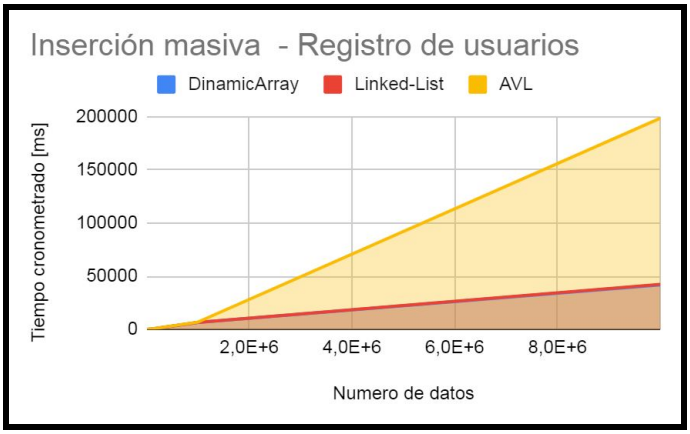
- Registro de Usuarios

Para las pruebas de esta funcionalidad se realizó la implementación mediante arreglo dinámico y AVL. Los resultados se muestran a continuación en tablas y se realizará un análisis comparativo en la siguiente sección.
Para calibrar el tiempo se utilizó un dato long para medir los milisegundos antes de iniciar el método y posterior a terminar el método, la diferencia entre estos valores genera el tiempo en milisegundos que tomo la operación. Se trabajó con un máximo de 10 millones de datos con un gasto de Memoria ram cercano a las 6Gb . Los datos se obtuvieron de un generador de datos online de 10000 datos y posteriormente fueron duplicados hasta el millón de datos en excel donde se le agregó un id que identifica el número de dato que es es decir hay id con valores hasta de 999999, y posteriormente en Access se realizó la inserción y multiplicación de los datos hasta alcanzar los 10 Millones también con su respectivo Id.
El análisis para usuarios se realizó en un equipo con Windows 7, un procesador Intel Core I7, 8 Gb de RAM en el IDE Netbeans 11.0.

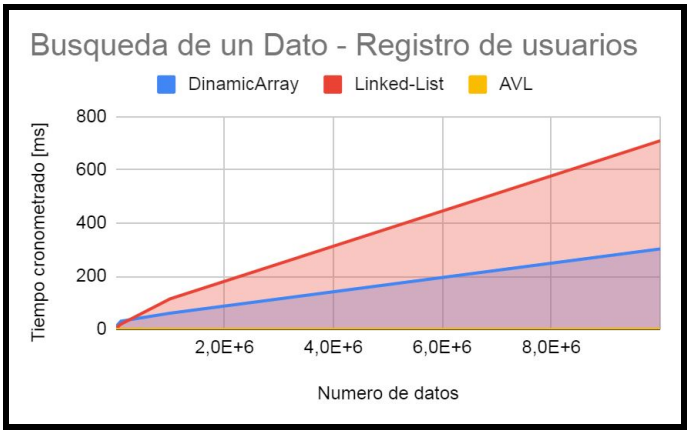
- Pruebas de Flora Fauna y Ecoparques

Se trabajó con un máximo de 1 millón, los datos se obtuvieron de un generador de datos y de wikipedia, y se copiaron hasta obtener el número de datos esperados, las pruebas se realizaron en un equipo con windows 10, un procesador intel inside, una memoria ram de 4Gb y el IDE Netbeans 11.0

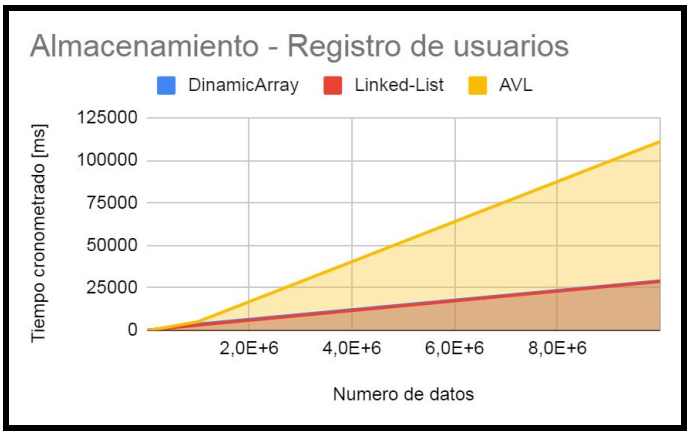
Registro de usuarios			
AVL			
Número de datos	Tiempo cronometrado [ms]		
	insertado masivo O(nlog(n))	Buscar un dato O(log(n))	Almacenar O(n)
10000	108	0	98
100000	352	0	320
1000000	6865	0	5049
10000000	198660	0	110990



Registro de usuarios			
DinamicArray			
Número de datos	Tiempo cronometrado [ms]		
	insertado masivo O(n)	Buscar un dato O(n)	Almacenar O(n)
10000	96	11	80
100000	344	31	418
1000000	6626	61	3672
10000000	41907	303	29093

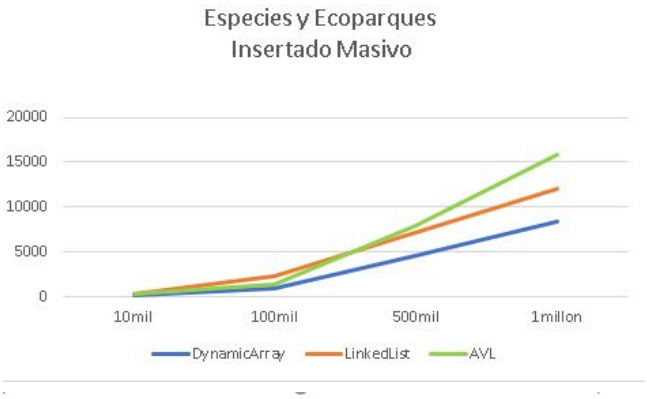


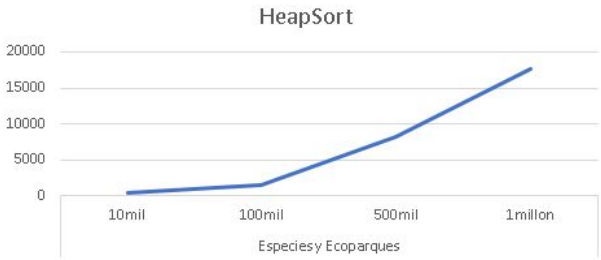
Registro de usuarios			
Linked-List			
Número de datos	Tiempo cronometrado [ms]		
	insertado masivo	Buscar un dato O(n)	Almacenar O(n)

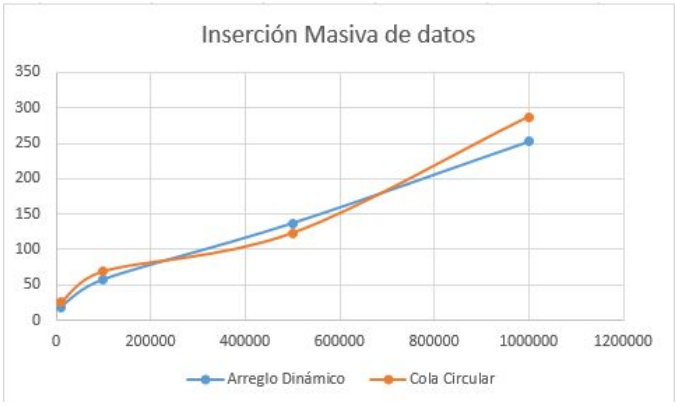


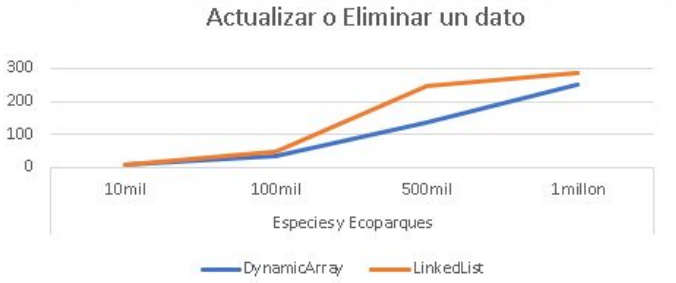
Linked List Especies y Ecoparques					
	Tiempo en ms				
Numero de datos	Insertado masivo O(n)	Insertado de un dato O(1)	Buscar un dato O(n)	Actualizar un dato O(n)	Eliminar un dato O(n)
10,000	283	0	13	7	7
100,000	2258	0	70	40	47
500,000	7106	0	155	261	246
1,000,000	12069	0	202	203	285

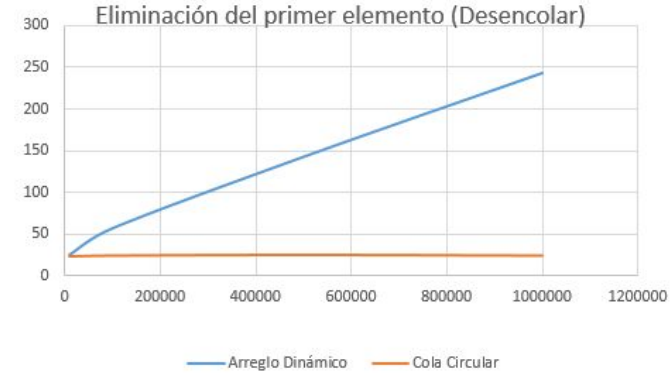
AVL Especies y Ecoparques			
	Tiempo en ms		
Numero de datos	Insertado masivo O(nlogn)	Insertado de un dato O(logn)	Buscar un dato O(logn)
10,000	304	0	0
100,000	1012	0	0
500,000	5877	0	0
1,000,000	12756	0	0









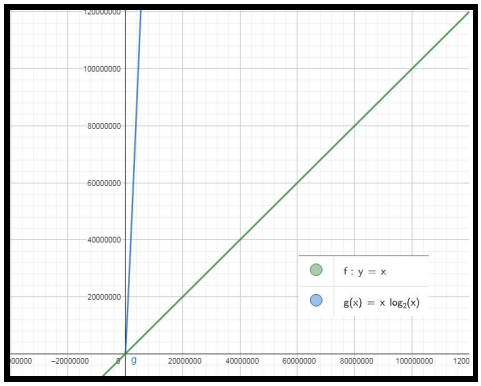


Dynamic Array Especies y Ecoparques					
Numero de datos	Tiempo en ms				
	Insertado masivo O(n)	Insertado de un dato O(1)	Buscar un dato O(n)	Actualizar un dato O(n)	Eliminar un dato O(n)
10,000	157	0	17	6	7
100,000	905	0	66	34	35
500,000	4511	0	134	125	134
1,000,000	8400	0	251	450	250

Análisis Comparativo

- Registro de Usuarios

La diferencia del tiempo de inserción masiva de datos tanto en el arreglo dinámico como en la lista enlazada es mínima lo que se puede explicar mediante el análisis de amortización en el cual se concluye que al comparar los métodos de inserción de ambas estructuras el orden daba lineal por lo tanto era igual de complejo insertar datos de manera masiva en una o en la otra. Es decir aunque en notación big O la inserción de un dato en la lista enlazada es de O(1) y en un arreglo dinámico es de O(n) para un conjunto de datos n la complejidad es la misma si para el arreglo dinámico se va ahorrando para cuando llegue el momento del resize. En el caso del AVL la complejidad es de O(Log(n)) para la inserción de un dato dado que debe ser organizado en la estructura, y si se toma el hecho de hacer una inserción masiva o de n usuarios la complejidad entonces resulta ser O(nlog(n)). En la siguiente imagen se puede constatar cómo la complejidad del AVL tiene una pendiente más grande que la complejidad de las estructuras lineales.



En el método de almacenamiento al momento de terminar la ejecución de la estructura se ve de manera muy definida la complejidad lineal para n datos y la similitud entre las estructuras Dynamic-Array y Linked-List .Para el AVL se puede observar un comportamiento lineal que tiene una pendiente más grande dado que la operación se realiza de manera recursiva y con un recorrido en Pre-order .

No se realiza gráfica comparativa para la inserción de un dato dado que el orden es constante(O(1)) para la inserción en las estructuras lineales y de orden logarítmico para el AVL, sin embargo, se mantiene un valor inferior a 1ms en las tres estructuras.

Para la actualización de datos y la búsqueda de datos los tiempos son similares en cada estructura lineal dado que se requiere primero buscar el dato para poder actualizarlo sin embargo la pendiente de crecimiento es más grande en las listas enlazadas dado que se debe definir en cada paso del ciclo un elemento nodo que me permita buscar si el objeto (key) de ese nodo es el buscado es decir se requieren más instrucciones en el ciclo para poder hacer el método. Para el AVL el tiempo en todos los casos de datos probados es inferior a 1 ms lo que se traduce en como es de esperar que alcance un valor y tiende asintóticamente a este al aumentar el número de datos de prueba con su complejidad de O(log(n)).

Se enfatiza que además de las tablas comparativas, se deben presentar gráficas que ilustran el análisis de complejidad realizado.

- Especies y Ecoparques

En nuestra implementación, para manejar estos datos se usa principalmente el arreglo dinámico, cuyas complejidades de búsqueda, actualización y eliminación son de complejidad lineal O(n), y la inserción al final, en caso de que no se tenga que ejecutar el reSize es constante O(1), usamos también un árbol binario de búsqueda autobalanceado AVL que se arma en tiempo de ejecución, donde buscar un dato tiene una complejidad logarítmica O(logn), también para la búsqueda, donde se tienen en cuenta las búsquedas como criterio de prioridad, usamos un HeapSort, que se arma también en tiempo de ejecución y tiene una complejidad de O(nlogn)

- Paginación

En la paginación de datos se siguió utilizando un arreglo dinámico, facilita la inserción de datos y el acceso a través del índice. Este método tiene una complejidad O(n), debido a que se añaden n objetos al arreglo. La impresión de los datos en pantalla depende de un loop tipo for, el cual maneja dos variables de control, el número de resultados por página y el número de páginas. Se crearon subarreglos, estos del tamaño de el número de resultados por página. De este subarreglo se va a imprimir en pantalla. Existe una clase de Paginador para cada tipo de dato (Flora, Fauna, Ecoparques).

- Sección de noticias

En esta entrega se agregó una sección de noticias al prototipo del proyecto, para su implementación se observaron dos alternativas, una es el uso de dos pilas lineales, una para avanzar de página y otra para retroceder. La otra alternativa es una Cola Circular. El objetivo es almacenamiento temporal, de tal manera que al agregar una nueva noticia, la más antigua salga de circulación.

Por esta razón se optó por la cola circular, se aprovecha el comportamiento FIFO de la estructura, de tal manera que las primera inserciones son las primeras en desencolarse. La mayor ventaja y utilidad de esta estructura está en la inserción de datos, pues cuando se copa la capacidad del arreglo, la cola elimina el primer elemento y cambia de posición todos los elementos del grupo, esto tendría una complejidad de O(n), pero como el tamaño de la cola circular es siempre definido, la eliminación de los primeros datos (desencolar) tiene una complejidad O(n).

VIII. DIFICULTADES Y LECCIONES APRENDIDAS

- Para poder implementar la interfaz de manera correcta fue necesario crear métodos estáticos de las funcionalidades que ya teníamos, tuvimos que repetirlos un par de veces porque no estábamos familiarizados con ese concepto.

- La interfaz es un desarrollo que resta tiempo de análisis e implementación de las estructuras que aunque sirven los conocimiento adquiridos desenfocan la atención en las estructuras y su profundo análisis.

- El avance de la cuarentena afecta los ánimos de continuar dedicándose a las responsabilidades académicas de la misma manera

- Los árboles son una herramienta que no se puede imaginar su eficiencia de búsqueda hasta que se implementa y compara con los tiempos de ejecución en este caso de estructuras lineales.

- Se generaron ciertos errores en la realización de algunas estructuras, errores de dimensionamiento y de Nulidad de apuntadores fueron los más comunes, pero se lograron corregir para la presentación de esta entrega .

IX. BIBLIOGRAFÍA

[1] "Biodiversidad en Colombia", *SiB Colombia*, 2012. [Online]. Available: https://sibcolombia.net/el-sib-colombia/. [Accessed: 15- Jun- 2020].

[2] L. Vélez, "Desconocimiento sobre biodiversidad en Colombia: motor de su deterioro", *www.elcolombiano.com*, 2013. [Online]. Available: https://www.elcolombiano.com/historico/desconocimiento_sobre_biodiversidad_en_colombia_motor_de_su_deterioro-EYEC_265414. [Accessed: 15- Jun- 2020].