Hygge, búsqueda de negocios y sitios turísticos

Valentina Colmenares Leguízamo, Daniel Lozano Barrero,

Mateo Sebastian Ortiz Higuera, Kevin Camilo Rincón Bohorquez

No. grupo del curso 2 y No. de Equipo de Trabajo: 6

I. INTRODUCCIÓN

El turismo es uno de los sectores económicos más importantes del país y además uno de los más afectados por la pandemia, ya que en el año 2020 el PIB turístico tuvo una bajada del 34%, lo cual ha afectado a todos los tipos de turismo y a las personas que viven de ello. En este documento buscamos explicar el problema mencionado anteriormente y la solución que buscamos implementar.o.

I. DESCRIPCIÓN DEL PROBLEMA

El sector turístico ha sido afectado gravemente por la pandemia del covid-19, dejando muchas personas sin empleo y afectando la economía del país. Según el DANE, en las áreas metropolitanas el porcentaje de población total que realizó viajes fue en el 2020 del 3.9% reduciendo considerablemente frente al 10% que lo hizo en el 2019, lo que supone una disminución de un 70% de ingresos; aproximadamente 34.600 personas ligadas al turismo perdieron su empleo en el 2020 y se prevé que este sector no volverá a ser el mismo del 2019 hasta el 2024.

Por lo cual, con este proyecto pretendemos ayudar a impulsar la reactivación económica en este sector incentivando y difundiendo diversas actividades y tradiciones que ocurren a lo largo del territorio nacional.

II. USUARIOS DEL PRODUCTO DE SOFTWARE

Para el usuario común habrá dos tipos de perfiles, la gente que quiera utilizar el programa para buscar sitios turísticos tendrá un perfil de usuario y quienes quieran subir su negocio tendrán por separado otro perfil de negocio, en donde podrán introducir toda la información sobre el sitio que publicitan. Por otro lado, habrá otro tipo de usuario para administradores donde habrá más herramientas para manipulación de la información y los datos que el resto de usuarios crean.

III. REQUERIMIENTOS FUNCIONALES DEL SOFTWARE

- Los requerimientos funcionales de un sistema son aquellos que *Inicio de Sesión de Usuarios/Negocios*
- *Descripción:*
- El programa pedirá la información de username y contraseña

Acciones iniciadoras y comportamiento esperado: *Requerimientos funcionales*:

Si el programa reconoce que el usuario está registrado y la contraseña corresponde a la del usuario en cuestión entonces mantendrá esa cuenta activa en esa sesión. Si no es así mandará un mensaje diciendo que el usuario no está registrado o que la contraseña no es la correcta.

Para ello el programa hará una consulta en la lista de usuarios/negocios para buscar si existe el dato (usuario/negocio) y si así es entonces accederá a sus datos de username y contraseña para compararlos.

- *Registro de Usuarios/Negocios*
- *Descripción:*
- El usuario creará su usuario/negocio
- *Acciones iniciadoras y comportamiento esperado*:

Requerimientos funcionales:

El programa hará una búsqueda en la implementación de datos para verificar que ese usuario no esté registrado ya, si no lo está entonces el programa añadirá este nuevo dato al final de la implementación.

- *Edición de perfil de Usuarios/Negocios*
- *Descripción:*
- El usuario podrá cambiar la información de su perfil/negocio
- *Acciones iniciadoras y comportamiento esperado*:

Requerimientos funcionales:

Si hay un usuario/negocio con sesión activa podrá acceder a su perfil y desde allí podrá editar cierta información como puede ser el nombre de usuario, contraseña, biografía, edad, etc.

El programa hará una consulta en la lista de usuarios/negocios para buscar al usuario logeado y desde allí se pueden modificar la información del dato.

- *Eliminar Usuario/Negocio*
- *Descripción:*
- El usuario podrá eliminar su usuario/negocio
- *Acciones iniciadoras y comportamiento esperado*:

Requerimientos funcionales:

Una acción que tanto un administrador como el usuario en concreto podría hacer. El programa hará una consulta en la lista de usuarios/negocios para identificar la posición del

dato(usuario/negocio) y así eliminarlo, si se está desde el usuario en concreto entonces la sesión activa se reiniciará.

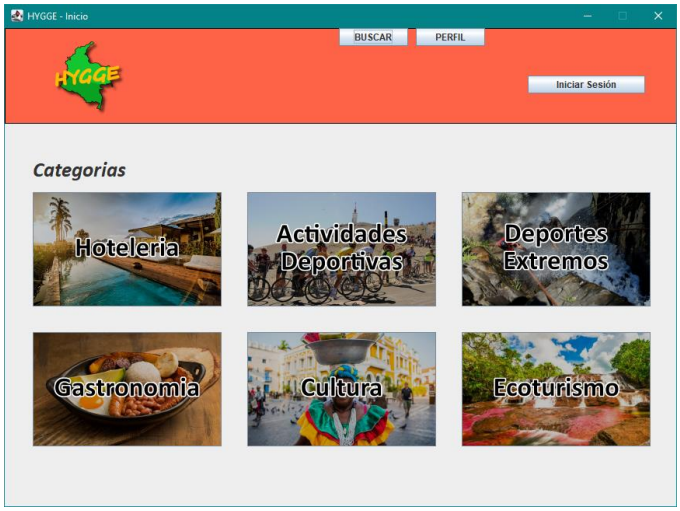
- *Buscar Negocio*
- *Descripción:*
- El usuario podrá acceder a la lista de negocios
- *Acciones iniciadoras y comportamiento esperado*:

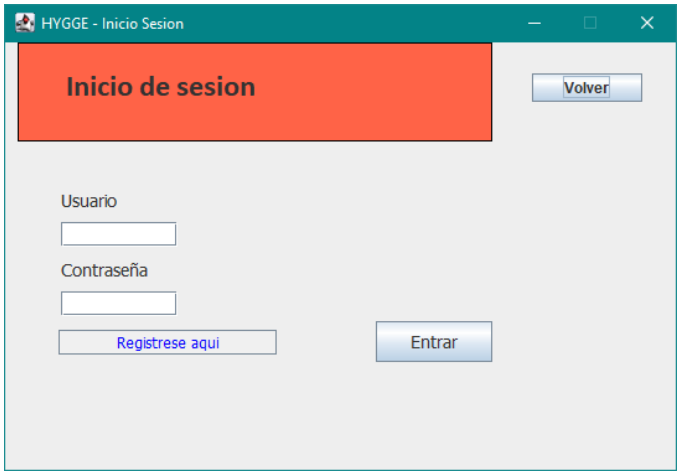
Requerimientos funcionales:

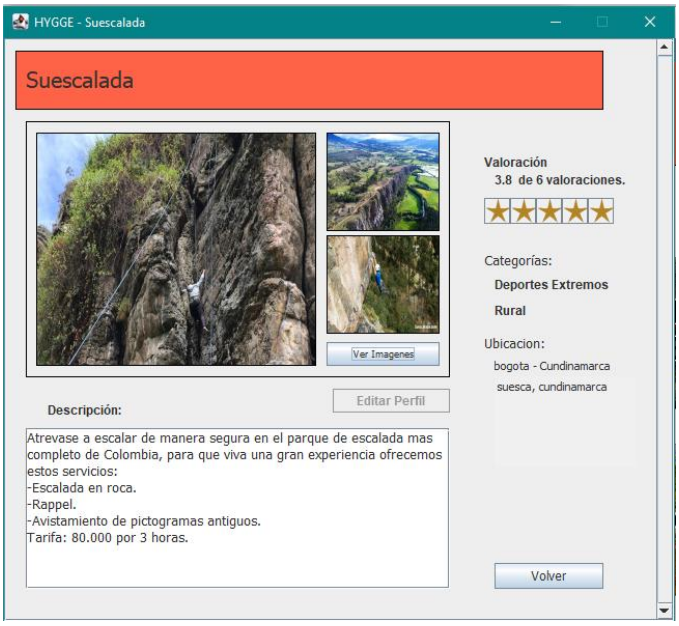
El usuario podrá acceder a una ventana en donde podrá buscar cualquier negocio registrado en el programa. El programa accederá a la lista e imprimirá en pantalla cada uno de los datos con la información más importante como nombre, categoría y ciudad.

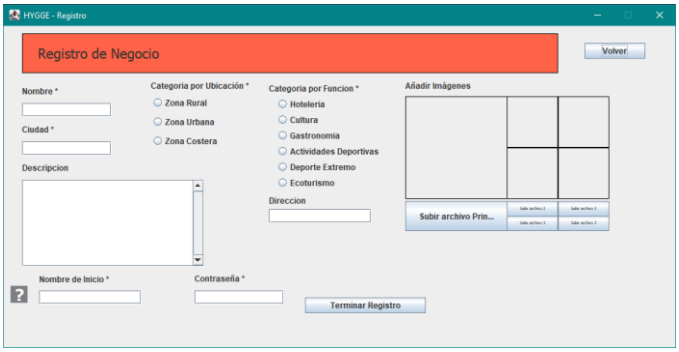
IV. AVANCE EN LA IMPLEMENTACIÓN DE LA INTERFAZ DE USUARIO

Las ventanas que se han desarrollado como parte gráfica del programa son:









V. ENTORNOS DE DESARROLLO Y DE OPERACIÓN

El programa se está desarrollando en Java utilizando entornos de desarrollo como Eclipse, Visual Code Studio, Replit y otras herramientas como GitHub.

VI. DESCRIPCIÓN DEL PROTOTIPO DE SOFTWARE

Para esta entrega hemos desarrollado la interfaz gráfica del programa compuesta por varias ventanas. Para cargar los usuarios y negocios se ha utilizado listas doblemente enlazadas, para la tabla de búsqueda de negocios se ordena con una cola de prioridad y se carga con un arraylist, y para guardar contraseñas hemos utilizado hash tables (para inicio de sesión).

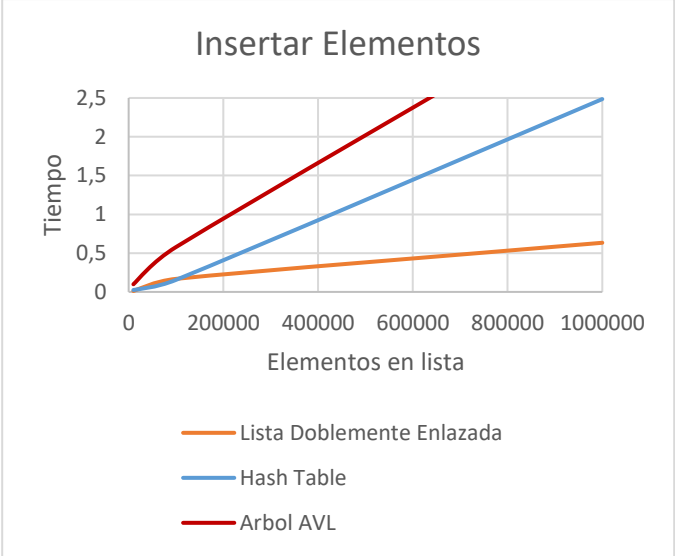
VII. PRUEBAS DEL PROTOTIPO DE SOFTWARE

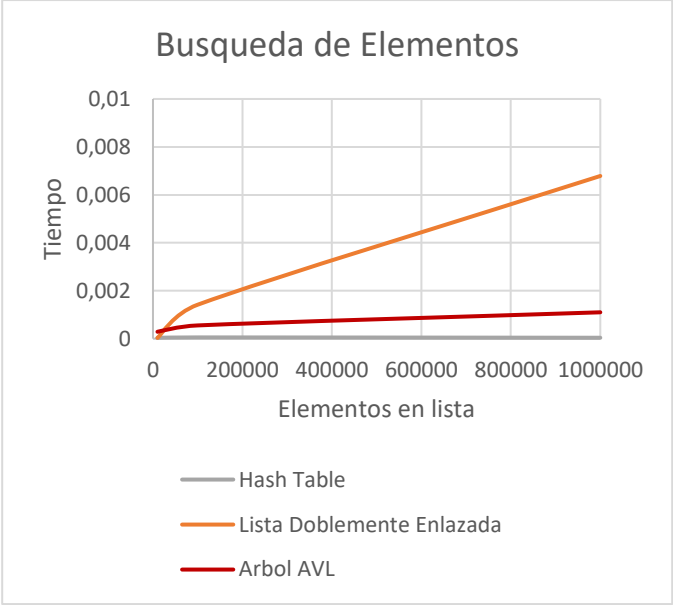
Se han escogido listas doblemente enlazadas, hash tables y arboles avl (para eliminar array dinámico) para comparar los tiempos de las funciones de insertar, buscar y eliminar datos (el de modificar datos entra dentro de la función de buscar). Y, también se han escogido una cola de prioridad, arrays ordenados y desordenados para comparar los tiempos de ejecuciones de las funciones de insertar y extraer el máximo elemento. Por medio de las siguientes tablas y gráficas comparativas:

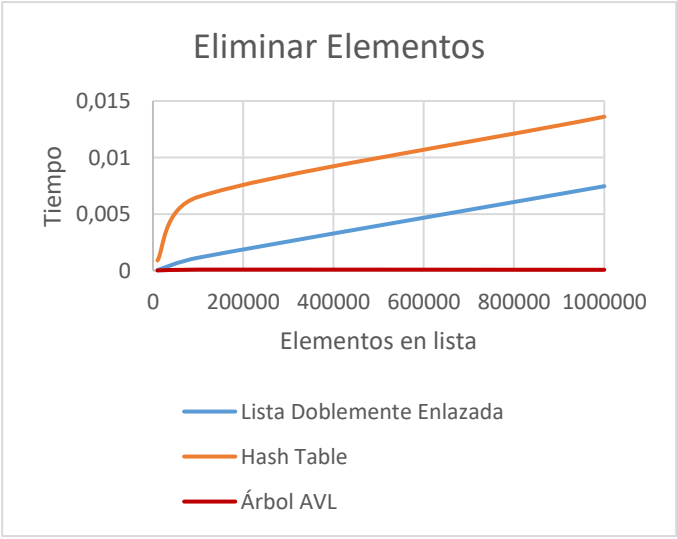
Inserción			
Número de Elementos	Lista Doblemente Enlazada	Hash Table	Árbol AVL
10000	0,011653	0,024624	0,098654
100000	0,16798	0,155371	0,578218
1000000	0,633375	2,485415	3,799458
Análisis (Notación O)	O(1)	O(1)	O(log n)

Búsqueda			
Número de Elementos	Lista Doblemente Enlazada	Hash Table	Árbol AVL
10000	0,000031	0,000036	0,000287
100000	0,001416	0,000052	0,000552
1000000	0,006787	0,000038	0,001099
Análisis (Notación O)	O(n)	O(1)	O(log n)

Eliminar Elementos			
Número de Elementos	Lista Doblemente Enlazada	Hash Table	Árbol AVL
10000	0,000042	0,000899	0,000019
100000	0,001144	0,006515	0,000093
1000000	0,007464	0,013607	0,000075
Análisis (Notación O)	O(n)	O(n)	O(log n)

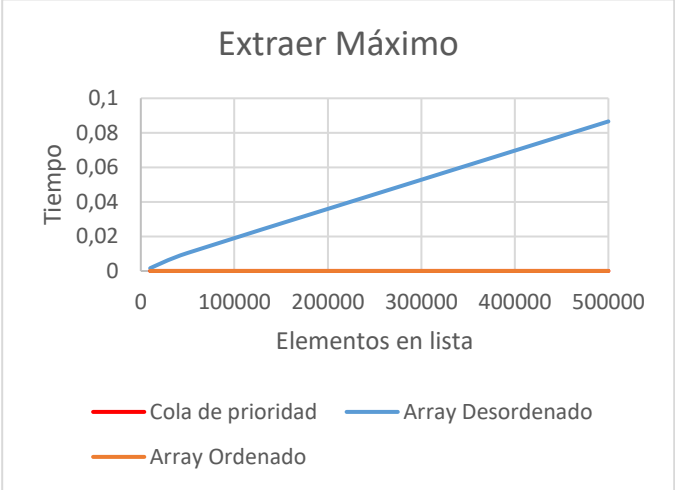






Insertar Elemento			
Número de Elementos	Cola de prioridad	Array Desordenado	Array Ordenado
10000	0,1875506	0,063349125	0,7069551
50000	0,4982200	0,3774783	22,86433167
500000	3,365321	3,7984035	
Análisis (Notación O)	O(1)	O(1)	O(n)

Extraer Máximo			
Número de Elementos	Cola de Prioridad	Array Desordenado	Array Ordenado
10000	0,0000205	0,001597125	0,0000045
50000	0,000027	0,0104285	0,00001475
500000	0,000028	0,0866036	0,000009
Análisis (Notación O)	O(1)	O(n)	O(1)



VIII. ACCESO AL REPOSITORIO DE SOFTWARE

El enlace al repositorio donde se encuentra el prototipo inicial es https://github.com/HyggePOO/HYGGE .

IX. ACCESO AL VIDEO DEMOSTRATIVO DE SOFTWARE

En enlace a la demostración del segundo prototipo es https://youtu.be/UNBxtB_w_9U .

X. ROLES Y ACTIVIDADES

Integrantes	Roles	Actividades Realizadas
Valentina Colmenares Leguízamo	Coordinadora	Desarrollo del código / Diapositivas
Daniel Lozano Barrero	Observador	Desarrollo del código/ Diapositivas
Mateo Sebastian Ortiz Higuera	Líder	Desarrollo del código / Plantilla de la entrega/ Diapositivas
Kevin Camilo Rincon Bohorquez	Experto	Desarrollo del código/ Diapositivas

XI. DIFICULTADES Y LECCIONES APRENDIDAS

Por los análisis hechos se puede observar que los árboles AVL tienen, en general, mejores tiempos de ejecución que las listas doblemente enlazadas, por lo que idealmente se planteaba utilizar esta estructura de datos para almacenar los usuarios y negocios, pero ante la imposibilidad de tener índices en un árbol avl, se presentaban errores a la hora de guardar los archivos donde se almacena la información de las cuentas. Luego por esto y por ser la mejor opción de estructura lineal, se ha preferido utilizar las listas doblemente enlazadas para esto.

En general, con este proyecto aprendimos las utilidades de las estructuras de datos a la hora de querer optimizar una tarea. Aprendimos a implementarlas utilizando datos de tipo String, Integers y Objects (particularmente los objetos del proyecto: Usuario y Negocio).

XII. REFERENCIAS BIBLIOGRÁFICAS

[1] Colprensa (27,03,2021) “El turismo interno en Colombia cayó un 6.1% durante 2020”. El Universal Disponible en:. https://www.eluniversal.com.co/colombia/turismo-interno-en-colombia-cayo-61-durante-2020-AD4401705 .

[2] Andrés Camacho y Omar Vanegas. “La crisis del turismo por el Covid-19”. Universidad Externado de Colombia. Disponible en: https://www.uexternado.edu.co/economia/la-crisis-del-turismo-por-elcovid-19/ .

[3] Portafolio (18,12,2020) “Turismo: de récord en 2019 a caída sin precedentes en 2020”. Portafolio. Disponible en: https://www.portafolio.co/economia/turismo-Campos

[4] 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.