

Hygge, búsqueda de negocios y sitios turísticos

Valentina Colmenares Leguízamo

Daniel Lozano Barrero

Mateo Sebastian Ortiz Higuera

Kevin Camilo Rincón Bohorquez

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.

II. DESCRIPCIÓN DEL PROBLEMA A RESOLVER

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.

III. USUARIOS DEL PRODUCTO DE SOFTWARE

Para el usuario común habrán 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 habrán más herramientas para manipulación de la información y los datos que el resto de usuarios crean.

IV. REQUERIMIENTOS FUNCIONALES DEL SOFTWARE

- *Inicio de Sesión de Usuarios/Negocios*
- *Descripción:*
 - El programa pedirá la información de username y contraseña
- *Acciones iniciadoras y comportamiento esperado:*
La entrada de el username y la contraseña y así el programa buscará si son correspondientes

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.

V. DESCRIPCIÓN DE LA INTERFAZ DE USUARIO PRELIMINAR

La interfaz gráfica consiste en ventanas donde se irán desarrollando las diferentes funcionalidades. Como podrían ser:

VI. 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.

VII. PROTOTIPO DE SOFTWARE INICIAL

El enlace al repositorio donde se encuentra el prototipo inicial es <https://github.com/HyggePOO/HYGGGE/tree/impDatos>

VIII. IMPLEMENTACIÓN Y APLICACIÓN DE LAS ESTRUCTURAS DE DATOS

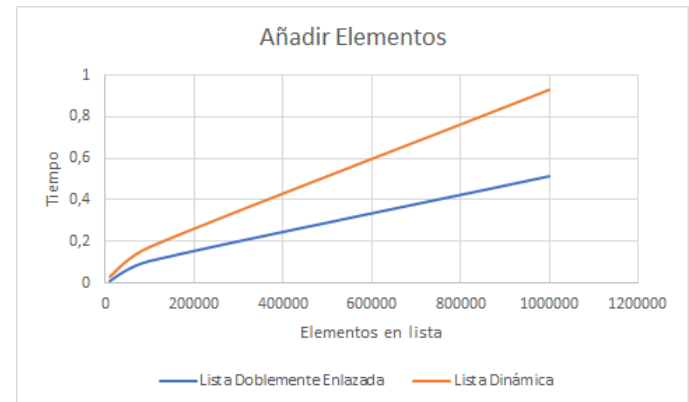
Se han implementado dos estructuras de datos: una lista doblemente enlazada con cola y una lista dinámica, en ambas se implementaron las siguientes funciones: inserción de datos, actualización de datos, eliminación de datos, búsqueda de datos y muestra en consola de todos los datos.

IX. PRUEBAS DEL PROTOTIPO Y ANÁLISIS COMPARATIVO

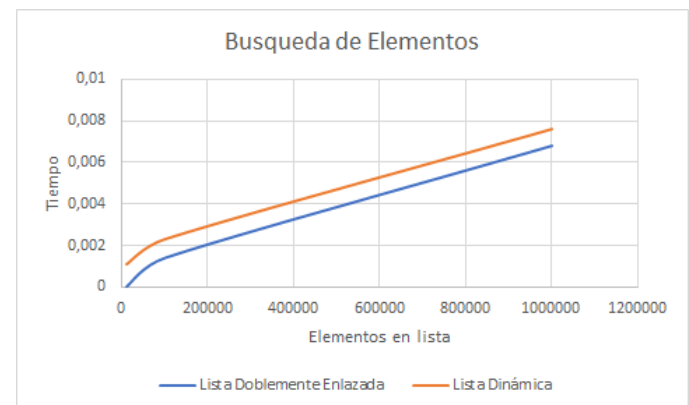
Se han escogido cuatro funciones y se han analizado hasta 1 millón de elementos (ya que a partir de los 10 millones no hay espacio en memoria para añadir todos los datos) y se han obtenido los siguientes datos:

| Añadir Elementos | | | |
|---------------------|----|---------------------------|----------------|
| Numero de Elementos | de | Lista Doblemente Enlazada | Lista Dinámica |
| 10000 | | 0,010906s | 0,029717s |
| 100000 | | 0,106859s | 0,173636s |
| 1000000 | | 0,515988s | 0,932824s |

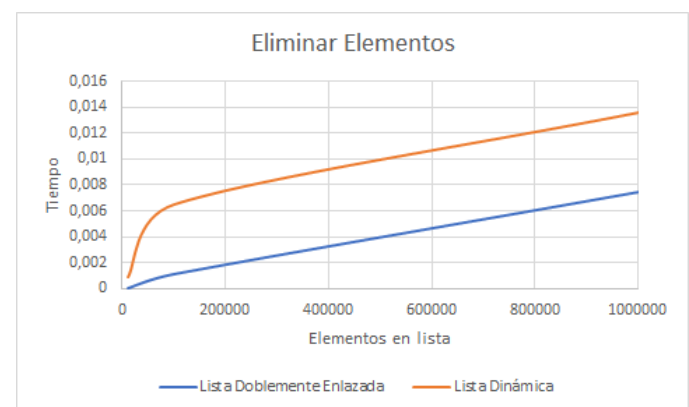
| Actualizar Elementos | | | |
|----------------------|----|---------------------------|----------------|
| Numero de Elementos | de | Lista Doblemente Enlazada | Lista Dinámica |
| 10000 | | 0,000063s | 0,000010s |
| 100000 | | 0,001329s | 0,000007s |
| 1000000 | | 0,003318s | 0,000016s |

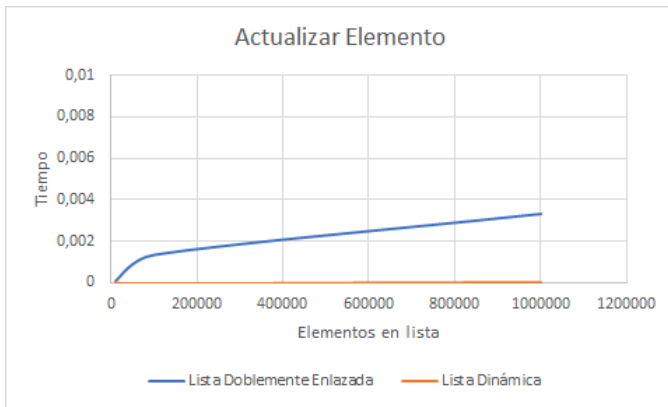


| Búsqueda de Elementos | | | |
|-----------------------|----|---------------------------|----------------|
| Numero de Elementos | de | Lista Doblemente Enlazada | Lista Dinámica |
| 10000 | | 0,000031s | 0,001078s |
| 100000 | | 0,001416s | 0,002303s |
| 1000000 | | 0,006787s | 0,007613s |



| Eliminar Elementos | | | |
|---------------------|----|---------------------------|----------------|
| Numero de Elementos | de | Lista Doblemente Enlazada | Lista Dinámica |
| 10000 | | 0,000042s | 0,000899s |
| 100000 | | 0,001144s | 0,006515s |
| 1000000 | | 0,007464s | 0,013607s |





- Como todos los elementos se añaden al final de la lista la complejidad de esa función es $O(1)$ en ambas implementaciones; en la función de la búsqueda del elemento tenemos una complejidad de $O(n)$ en ambas implementaciones, ya que en ambas debe recorrer la lista para asegurar que el elemento esté en la lista; en la función de eliminar la complejidad es de $O(n)$ en la lista dinámica ya que todos los elementos deben moverse para ocupar el espacio del elemento eliminado (a excepción del último elemento) y de $O(1)$ en la lista doblemente enlazada; y en la función de actualizar el elemento la complejidad es de $O(1)$ en ambas implementaciones.

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

El vídeo demostrativo del programa es https://youtu.be/i_hFd_PdF9E

XI. ROLES Y ACTIVIDADES

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

XII. DIFICULTADES Y LECCIONES APRENDIDAS

No hubo mayores dificultades en la realización de esta primera parte del proyecto.

XIII. 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-el-covid-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-de-record-en-2019-a-caida-sin-precedentes-en-2020-547654> .
- [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.