

SlipChat

Diego Alejandro Carvajal, Mateo Herrera, Juan Santamaria

No. de Equipo Trabajo: {13}

I. INTRODUCCIÓN

Este documento pretende describir las principales funcionalidades de la aplicación SlipChat, dando lugar a una apertura conceptual de sus características y prototipos.

DESCRIPCIÓN DEL PROBLEMA A RESOLVER

Es esperable que en una red social los datos crezcan gran velocidad, por lo cual es de importancia abordar conocimientos en el manejo de los datos; cómo son manipulados, desde el guardado, actualización y borrado.

II. USUARIOS DEL PRODUCTO DE SOFTWARE

Se espera que el software sea utilizado por aficionados a los juegos de plataformas, indie o de recorrido simple. Personas que ingresen al sitio web buscando un juego que no les tome mucho tiempo; y conversaciones amenas con personas con las que se sientan familiarizados. El sitio web se centrará en los encuentros casuales, en una interacción tú a tú que no se centre en una continua relación con una misma persona; sino que sea adecuado para conocer nuevas personas. Aquellos usuarios tendrán una cuenta privada y un perfil de usuario, en donde podrán modificar información personal; también tendrán acceso al juego interactivo y los chats.

III. REQUERIMIENTOS FUNCIONALES DEL SOFTWARE

Los requerimientos funcionales de un sistema son aquellos que describen cualquier actividad que este deba realizar, en otras palabras, el comportamiento o función particular de un sistema o software cuando se cumplen ciertas condiciones.

Entre los posibles requerimientos funcionales de un sistema, se incluyen:

SISTEMA

1. *Reporte de errores por usuarios*: Esta función permitirá a los usuarios reportar errores dentro del aplicativo.

Un usuario descubre un error; lo reporta; lo recibe y lo procesa.

2. *Reporte de errores por el sistema*: Permitirá el guardado de errores por parte del sistema.

Los errores son capturados; se llena un formato con la descripción del error.

3. *Revisión de credenciales*: Esta función permitirá que el sistema verifique si la contraseña y el usuario coinciden.

El usuario quiere ingresar; activa el sistema de revisión de credenciales para poder hacerlo.

CHAT

4. *Almacenamiento del chat*: El sistema permitirá almacenar un formato de tipo chat.

El sistema activa el algoritmo de guardado al cerrar un chat.

5. *Almacenamiento de chat temporal- local (opcional)*

6. *Envío de mensajes*: El sistema permitirá el envío de mensajes entre pares.

El usuario escribe un mensaje; lo envía.

7. *Recepción de mensajes*: El sistema dispondrá de una funcionalidad para la recepción de mensajes por parte de los usuarios.

Un usuario envía mensajes; su receptor los recibe con éxito.

8. *Búsqueda de un chat*: Se dispondrá de un sistema para la búsqueda de chats.

El usuario dispondrá de un buscador a la mano; se activa al ingresar palabras sobre el espacio de texto.

9. *Búsqueda de un mensaje*: Función que busca mensajes con palabras clave.

Al activarse y llenarse la opción de búsqueda; el sistema encontrará mensajes que coincidan.

10. *Decisión conservar chats*: El usuario podrá decidir entre guardar o no un chat nuevo.

El usuario quiere salir del chat; se muestran opciones; elige; el sistema borra o persiste los datos.

USUARIOS

11. *Almacenamiento de perfil*: El sistema guardará el perfil de usuario en formato.

Al modificar o al registrar la información de un usuario; el sistema permite el guardado.

12. *Visualización perfil usuario*: Acceso al perfil de usuario.

Click sobre su perfil; el sistema despliega una ventana con la información.

13. *Visualización perfil otro usuario*: Acceso a la información de perfiles de terceros.

Click sobre el usuario; se muestra información del usuario al cual se le dio click.

14. *Creación de usuario*: Se permite la creación de nuevos usuarios.

Llenado de formulario; el sistema recibe los datos y los procesa.

15. *Acceso al chat*: Acceso al apartado de chats.

Un usuario que tiene cuenta; el sistema le permitirá ingresar a sus chats.

JUEGO

16. *Acceso al servidor del juego*: Permite el ingreso de usuarios de forma ordenada.

Usuario intenta ingresar; se posiciona en cola; el sistema permite el ingreso FIFO.

17. *Control de usuarios en el servidor*: Regular la cantidad de usuarios en el servidor, para evitar retrasos en el servidor.

Servidor activo; permite el ingreso de nuevos usuarios dependiendo del espacio en el servicio.

18. *Visualización de información parcial del usuario*: Se permite visualizar datos parciales de terceros

Click sobre el usuario; se muestra su información básica

19. *Estados del Usuario (última fecha de cogedura)*: Permite al usuario modificar su estado: Jugar/Chatear/Ausente.

El sistema actualiza los datos de estado según se cambien en la web.

20. *Apertura de un nuevo chat*: La aplicación permitirá la apertura de nuevas conversaciones.

Dos usuarios se encuentran; se abre un nuevo chat.

IV. DESCRIPCIÓN DE LA INTERFAZ DE USUARIO PRELIMINAR

Página de inicio

- Título de la aplicación
- Inicio de sesión
- Creación de usuario (enlace a formulario registro)
- Descripción de la aplicación
- Imagen descriptiva del servicio

Página de juego

- Encabezado
 - Logo
 - Menú
 - Mis conversaciones
 - Mi perfil
 - Ajustes
 - Acerca de
 - Donar
 - Nombre de usuario
 - Imagen de usuario
- Espacio para el juego (canvas)

Página mis conversaciones

- Encabezado
- Chats activos (a la izquierda)
- Chat seleccionado (a la derecha)

V. ENTORNOS DE DESARROLLO Y DE OPERACIÓN

Se desarrollará bajo los fundamentos de construcción web, esto se refiere a html, CSS, JavaScript y además el uso de otras herramientas: Node.js, Angular, Phaser.

Operará sobre un servidor web y se ejecutará en un navegador cliente compatible con ES6.

VI. PROTOTIPO DE SOFTWARE INICIAL

El proyecto estará apoyado sobre el control de proyectos que ofrece github. El repositorio está en:

<https://github.com/Dicarvajalb/SlipChat>

Figura 1. Estructura del repositorio requerida durante el proyecto.

Para esta entrega se implementaron listas dinámicas y arreglos por medio de lista enlazada, los archivos de estas dos estructuras y sus ejecutables estarán en el repositorio, rama Entrega1 – root/dlist.

VII. PRUEBAS DEL PROTOTIPO

Para las pruebas del prototipo fueron implementadas dos estructuras: Lista doblemente encadenada y Arreglo dinámico. A las que se sometieron a prueba sus principales funcionalidades con 10 mil, 100 mil, 500 mil, 1 millón y 10 millones de datos, generados por un algoritmo de creación e inserción secuencial.

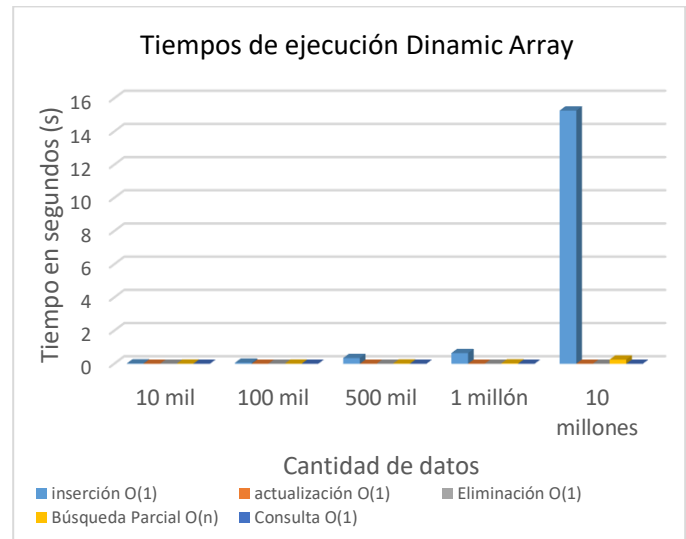
Las funcionalidades probadas fueron:

- Inserción.
- Actualización.
- Eliminación.
- Búsqueda parcial.
- Consulta.

Los registros de dichas pruebas son mostrados a continuación.

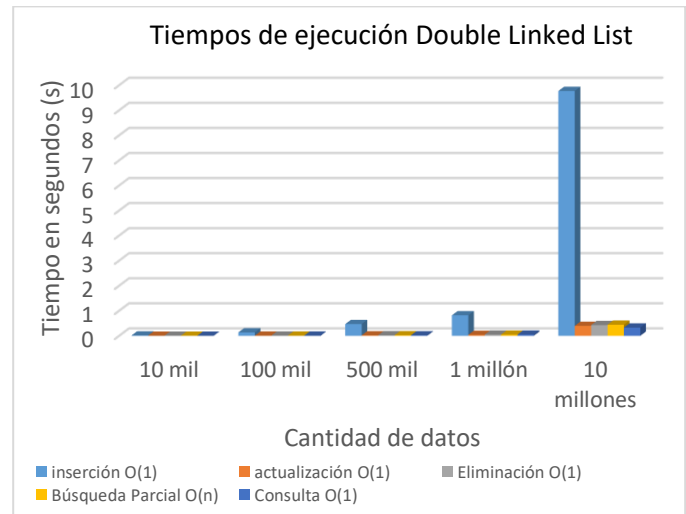
Arreglo dinámico

Funcionalidad	Inserción	actualización	Eliminación	Búsqueda	Consulta
Cantidad	O(1)	O(1)	O(1)	Parcial O(n)	O(1)
10 mil	0,0162	0,0002	0,0002	0,0029	0,0002
100 mil	0,0596	0,0002	0,0003	0,0067	0,0003
500 mil	0,3437	0,0002	0,0002	0,0120	0,0002
1 millón	0,6431	0,0002	0,0003	0,0223	0,0003
10 millones	15,2584	0,0003	0,0003	0,2496	0,0003



Lista doblemente encadenada

Funcionalidad	inserción	actualización	Eliminación	Búsqueda	Consulta
Cantidad	O(1)	O(1)	O(1)	Parcial O(n)	O(n)
10 mil	0,0131	0,0011	0,0028	0,0037	0,0011
100 mil	0,1366	0,0066	0,0062	0,0065	0,0052
500 mil	0,4683	0,0170	0,0209	0,0218	0,0191
1 millón	0,8178	0,0367	0,0466	0,0439	0,0458
10 millones	9,7687	0,3980	0,4278	0,4435	0,3277



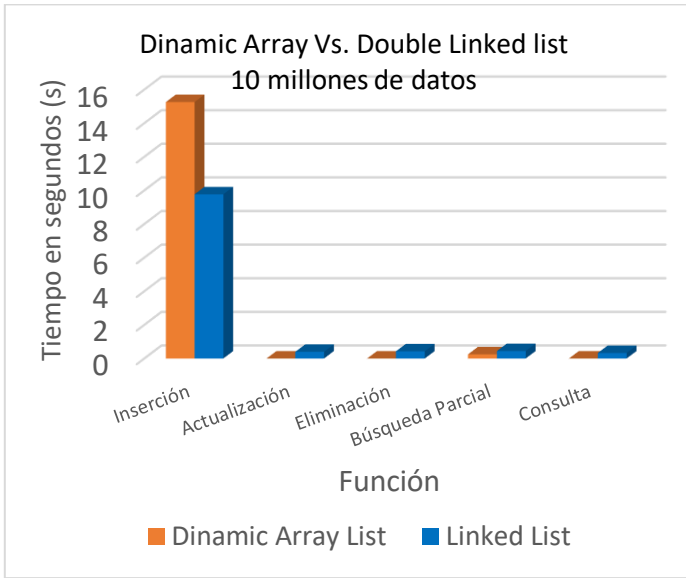
I. ANÁLISIS COMPARATIVO

A continuación, se presenta la tabla comparativa entre los tiempos de ejecución de las principales funciones para 10 millones de datos, junto con la gráfica respectiva.

Función Estructura	Inserción	Actualización	Eliminación	Búsqueda Parcial	Consulta
Arreglo dinámico	15,2584	0,0003	0,0003	0,2496	0,0003
Lista doblemente encadenada	9,7687	0,3980	0,4278	0,4435	0,3277

III. DIFICULTADES Y LECCIONES APRENDIDAS

Se ha emprendido en métodos de estudio autónomo y se han seguido caminos erróneos, se leyeron distintos volúmenes de libros, pero se concluyó que un libro no es mejor opción que un curso, se ha usado estrategias para buscar materiales en internet. Es de importancia tener una estrategia que reparta el tiempo de cada semana para la realización del proyecto. También surgieron problemas con el manejo de la lectura de la información, que nos han enseñado a ser más cuidadosos. Se emprenderá en la realización de un calendario que comprenda las funciones de cada integrante y sus tiempos de entrega.



De los datos comparativos anteriores es observable que los tiempos de ejecución para las funciones probadas siguen el orden en notación O y a su vez, a excepción de inserción, son notablemente menores para el arreglo dinámico. Esto es causado en gran parte por la facilidad que proporciona un arreglo para acceder a una posición de memoria.

Para la función de inserción la lista doblemente encadenada tiene un tiempo de ejecución menor, debido a que está creciendo con cada nodo insertado a la estructura y no está limitada por una cantidad máxima de nodos. A diferencia del arreglo, que para proveer una propiedad dinámica es necesario aumentar su capacidad.

Cabe anotar que para aumentar la eficiencia de la función de eliminación para la estructura de arreglo dinámico se realizó una implementación en la que el dato eliminado es remplazado por el último del arreglo.

II. ROLES Y ACTIVIDADES

El documento de excel que contiene esta tabla estará en el repositorio de git, en la carpeta root/docs.