

Study Buddy

Ana González, Cristian Montañez, Daniel Rodriguez, Samuel Vargas.
No. de Equipo de Trabajo: 5

I. INTRODUCCIÓN

Study Buddy se presenta en este documento como un proyecto de una aplicación móvil cuyo propósito es proporcionar a los usuarios una variedad de ejercicios y experiencias interactivas de aprendizaje para el vocabulario del idioma inglés. Basados en un público estudiantil, se listan las diferentes funcionalidades y sus respectivos requerimientos funcionales enfocados en una experiencia didáctica y educativa. Asimismo, se presenta el prototipo de la interfaz de usuario de la aplicación realizado en Figma junto a la segunda versión del software la cual fue realizada en Dart y Flutter y registrada en un repositorio de Github. Se finaliza con la implementación de las estructuras de datos (Árboles, Priority Heaps y Disjoint Sets) con Dart, el respectivo análisis comparativo con n diferentes tamaños de datos de prueba, una descripción de los roles y actividades de cada integrante del equipo y las lecciones aprendidas durante el desarrollo de la segunda entrega.

II. DESCRIPCIÓN DEL PROBLEMA A RESOLVER

Aprender inglés puede ser una tarea desafiante, pero también muy gratificante. Sin embargo, es difícil encontrar recursos de aprendizaje totalmente enfocados en la práctica de vocabulario que cuenten con dinámicas variadas respecto al formato de los ejercicios, almacenamiento y estrategias didácticas para el aprendizaje de vocabulario nuevo y una experiencia gamificada enfocada en la competición con compañeros de estudio (otros usuarios). Es por ello que Study Buddy está pensada para unificar las características mencionadas y con ayuda de la tecnología crear una aplicación con una experiencia innovadora, didáctica y diferente respecto a la práctica del inglés.

III. USUARIOS DEL PRODUCTO DE SOFTWARE

A. ESTUDIANTES:

- **Frecuencia de Uso:** Varía según el nivel y los objetivos personales, desde diario hasta ocasionalmente.
- **Características:** Incluye a personas que están aprendiendo inglés en diferentes niveles, desde principiantes hasta avanzados.
- **Funcionalidades utilizadas:** Acceso a un gran repertorio de vocabulario, ejercicios de práctica, seguimiento de progreso y otros recursos de aprendizaje.

- **Nivel de Experiencia:** Puede variar desde principiantes con conocimientos mínimos de inglés hasta estudiantes avanzados con un mayor dominio del idioma.

IV. REQUERIMIENTOS FUNCIONALES DEL SOFTWARE

A. Registro/Inicio de Sesión

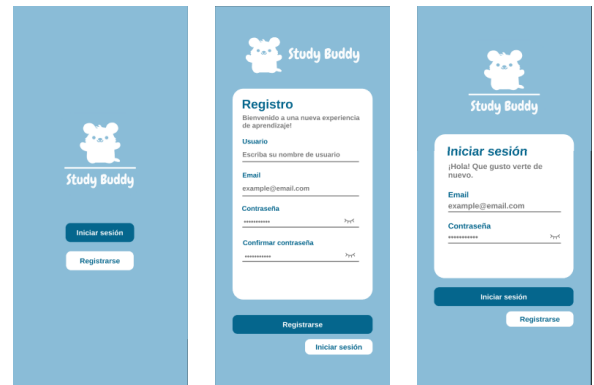


Fig 1. Pantallas de registro e inicio de sesión

- **Descripción:** Esta funcionalidad permitirá a los usuarios crear un perfil en la aplicación, asimismo ingresar con sus respectivas credenciales a la app. Esto con el fin de que los usuarios puedan almacenar su progreso e información de manera individual.
- **Acciones iniciadoras y comportamiento esperado**
 - El usuario podrá elegir entre iniciar sesión o registrarse.
 - Si el usuario elige iniciar sesión, deberá ingresar en los campos correspondientes su correo y contraseña. También tendrá la opción de ir a registrarse.
 - Si el usuario da click a iniciar sesión y sus credenciales son correctas, el programa

deberá enviarlo a la pantalla del menú principal.

- Si las credenciales son incorrectas, el usuario no podrá ingresar hasta que las ingrese correctamente.
- Si el usuario elige registrarse, deberá ingresar en los campos correspondientes su correo, nombre de usuario, contraseña y una confirmación de la misma. El usuario también tendrá la opción de ir a iniciar sesión.
- Si el usuario ingresa los datos de manera correcta, lo manda a iniciar sesión.
- Si los datos tienen algún error, el usuario no podrá registrarse hasta ingresarlos correctamente.

- **Requerimientos funcionales**

- El sistema controlará el acceso y lo permitirá solamente a usuarios autorizados.
- El campo email verificará que la entrada del usuario corresponda con el formato de un email.
- El campo contraseña verificará que esté tenga mínimo 6 caracteres.
- El campo confirmar contraseña verificará que la contraseña y la confirmación sean iguales.
- El sistema en la parte de registro reconocerá los siguientes errores:
 - Si el email ingresado está en uso, saldrá un mensaje de error.
 - Si la dirección de email no es válida, generará un mensaje de error.
 - Si el email y la contraseña no están habilitados, saldrá un mensaje de error.
 - Si la contraseña no es lo suficientemente fuerte, se generará un mensaje de error.
- El sistema en la parte de login reconocerá los siguientes errores:
 - Si la dirección de email no es válida, generará un mensaje de error.
 - Si el correo del usuario ha sido deshabilitado, se generará un mensaje de error.
 - Si no existe un usuario correspondiente al email

ingresado, saldrá un mensaje de error.

- Si la contraseña es inválida respecto al email ingresado, se generará un mensaje de error.
- Si el usuario se registra de manera exitosa, el sistema mostrará un mensaje de registro exitoso y mandará al usuario a iniciar sesión.

B. Clasificación de tiempos entre los usuarios globales

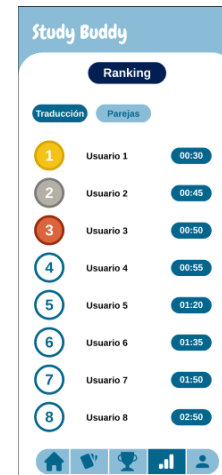


Fig. 3. Ranking entre los tiempos de los usuarios

- **Descripción:** Dependiendo del desempeño de los usuarios en los juegos de traducción y parejas, se creará un ranking entre los usuarios globales.
- **Acciones iniciadoras y comportamiento esperado**
 - El usuario podrá visualizar el ranking y su respectiva posición en el juego de traducción.
 - El usuario podrá visualizar el ranking y su respectiva posición en el juego de parejas.
- **Requerimientos funcionales**
 - En la funcionalidad de parejas y traducción se agregará la medición de tiempo.
 - El sistema deberá actualizar el ranking respecto al menor tiempo utilizado en cada juego.
 - La interfaz deberá resaltar los 3 usuarios con menores tiempos.
 - El sistema diferenciará el ranking según el juego.
 - El sistema deberá tener en cuenta los tiempos de todos los usuarios.

C. Adición de flashcards por parte del usuario

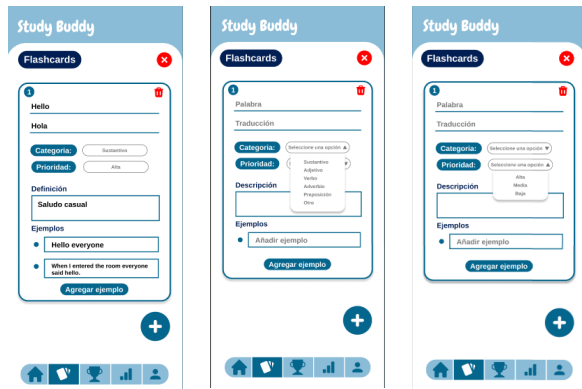


Fig. 4. Adición de flashcards por el usuario

- **Descripción:** La lista de flashcards del usuario podrá contar con palabras nuevas ingresadas por el mismo usuario.
- **Acciones iniciadoras y comportamiento esperado**
 - El usuario podrá agregar palabras ingresando el término, su traducción y respectiva categoría.
 - El usuario podrá eliminar flashcards que haya ingresado con anterioridad.
- **Requerimientos funcionales**
 - La interfaz le permite elegir la categoría al usuario.
 - El sistema agrega la información ingresada a la lista de flashcards del usuario.
 - El sistema permite la eliminación de flashcards.

D. Perfil del usuario

- **Descripción:** El usuario tendrá la opción de cambiar su nombre de usuario y cerrar sesión.
- **Acciones iniciadoras y comportamiento esperado**
 - El usuario podrá cambiar su nombre de usuario.
 - El usuario podrá cerrar sesión dando click en el respectivo botón.
- **Requerimientos funcionales**
 - El sistema responderá al cambio de usuario actualizando la información en la base de datos.
 - El sistema responderá al cierre de sesión enviando al usuario a la pantalla de Registro/Inicio de sesión.



Fig.5 Perfil del usuario

V. DESCRIPCIÓN DE LA INTERFAZ DE USUARIO PRELIMINAR

El diseño de la interfaz de usuario se trabajó en Figma, de forma que se agregaron las diferentes pantallas necesarias para la implementación de las cuatro funcionalidades. El diseño se puede encontrar en este [link](#). Para el diseño de esta entrega además de tener en cuenta la importancia de un diseño intuitivo, se tuvo en cuenta el manejo de errores y su respectiva respuesta desde la interfaz.

VI. ENTORNOS DE DESARROLLO Y DE OPERACIÓN

El desarrollo del proyecto se llevará a cabo utilizando el SDK de Flutter junto con el lenguaje Dart, con Visual Studio Code como el entorno de desarrollo preferido. La aplicación tendrá como objetivo principal funcionar en dispositivos móviles con sistema operativo Android, aunque gracias a la versatilidad de Flutter, se considerará la posibilidad de generar versiones adicionales para aplicaciones web y iOS en el futuro, lo que permitirá una mayor flexibilidad y accesibilidad a través de diferentes plataformas. Por otro lado, en esta entrega se utilizó Firebase para alojar nuestra base de datos en la nube y almacenar y sincronizar datos entre nuestros usuarios en tiempo real.

VII. PROTOTIPO DE SOFTWARE INICIAL

En esta entrega se continuó el desarrollo de la aplicación mediante el repositorio alojado en Github pues permite el trabajo colaborativo de manera más sencilla, comparado con otros operadores de control de versiones.

Enlace para poder acceder al repositorio:

<https://github.com/Samvargasca/StudyBuddy>

VIII. IMPLEMENTACIÓN Y APLICACIÓN DE LAS ESTRUCTURAS DE DATOS

En la funcionalidad de "Clasificación de tiempos entre los usuarios globales", se ha elegido utilizar un Binary Search Tree (BST) para gestionar los menores tiempos de los usuarios en los minijuegos Traducir y Parejas. Esta decisión se basa en su eficiencia en búsquedas logarítmicas, capacidad para mantener un orden automático, realizar inserciones y eliminaciones eficientes, adaptarse a cambios dinámicos en el ranking y utilizar espacio de memoria de manera eficiente.

En la funcionalidad "Adición de flashcards por parte del usuario", se utiliza un Max Heap para gestionar las flashcards. Cada palabra en las flashcards se almacena con una prioridad basada en su nivel de importancia o dificultad, permitiendo a los usuarios centrarse en aprender las palabras más relevantes primero y optimizando su proceso de estudio.

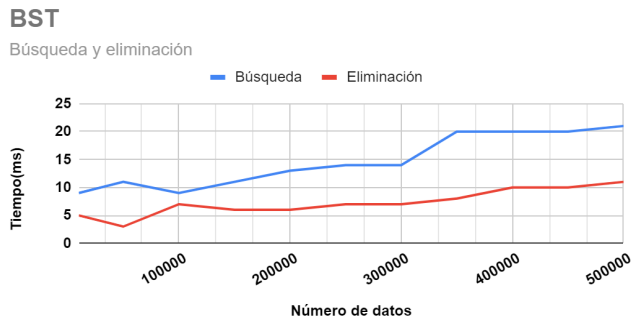
Por otro lado, para las funcionalidades de Registro/Inicio de Sesión y Perfil del usuario se hizo uso de las herramientas que ofrece Firebase por lo que no se utilizó alguna implementación de estructura de datos particular.

IX. PRUEBAS DEL PROTOTIPO Y ANÁLISIS COMPARATIVO

Para probar la complejidad de las distintas estructuras se sometieron a pruebas con un número de datos desde 10000 hasta 500000 datos para tres operaciones distintas en cuatro diferentes implementaciones de estructuras: BST, AVL, MaxHeap y Disjoint Sets.. Estas comprobaciones se hicieron para el tipo de dato primitivo entero.

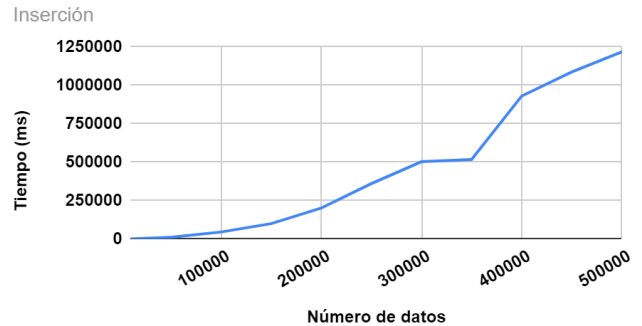
A. BST

Para esta estructura de datos los resultados en búsqueda y eliminación son los siguientes:



Debido al cambio brusco de escala en términos de tiempo para la inserción se tuvo que hacer la siguiente gráfica aparte.

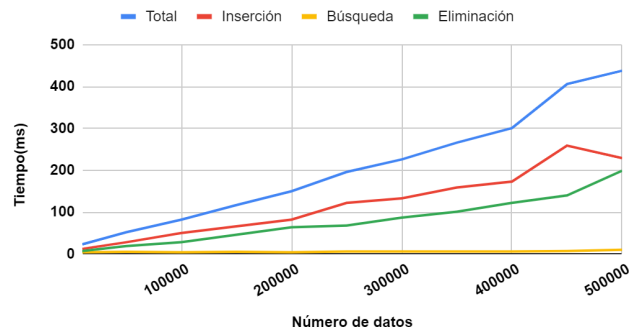
BST



B. AVL

Para esta estructura las mediciones se presentan en el gráfico siguiente:

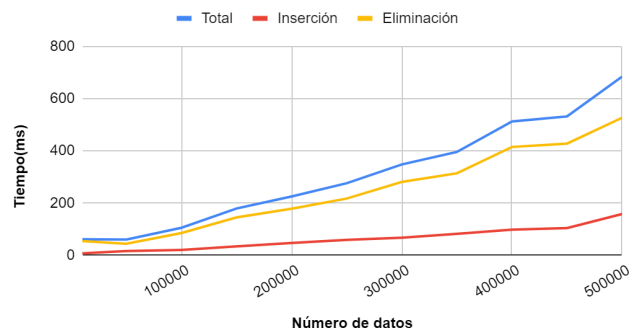
AVL



C. MaxHeap

En este caso dadas las características de esta estructura solo se midió el tiempo de inserción y eliminación generando los siguientes resultados.

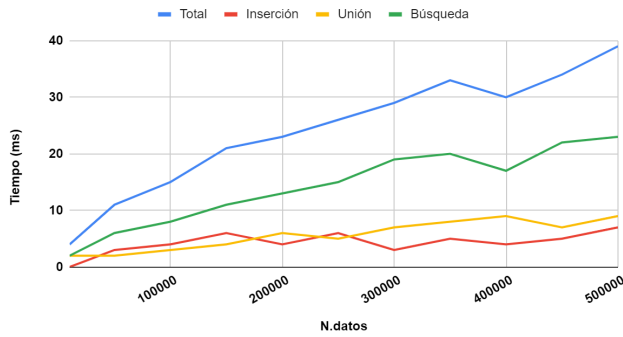
MaxHeap



D. Disjoint Sets

Para esta estructura se midió el tiempo de la "Unión" en vez de la "Eliminación", el gráfico correspondiente se encuentra a continuación:

Disjoint Sets

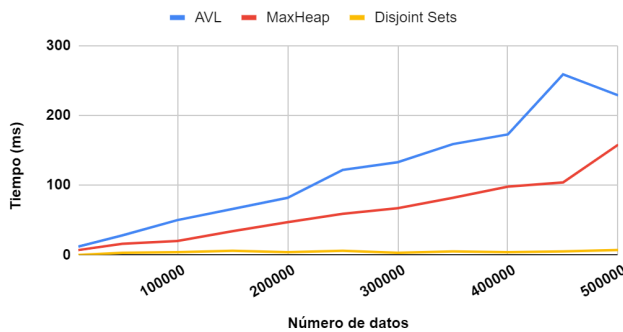


Ahora se adjuntan las comparaciones realizadas entre el desempeño de las estructuras.

A. Inserción

Debido al gran cambio de escala en los tiempos del BST, convirtiendo a esta estructura en la de mayor tiempo de inserción $O(n)$, no se agregó a la gráfica. La comparación entre AVL, MaxHeap y Disjoint Sets es la siguiente:

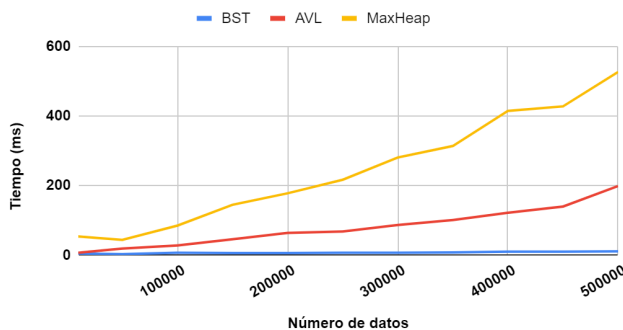
Inserción



B. Eliminación

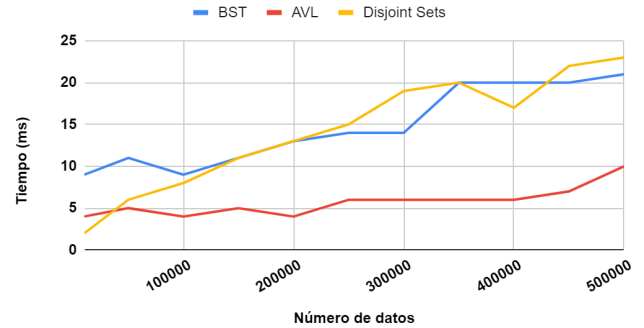
En este caso no se incluyó Disjoint Sets en la comparación.

Eliminación



C. Búsqueda

Búsqueda



Dado que la altura del AVL se reduce por los balanceos propios de la estructura vemos que tiene el menor tiempo en búsqueda $O(\log n)$ en comparación con BST y Disjoint Sets.

X. ROLES Y ACTIVIDADES

Integrante	Rol(es)	Actividades realizadas
Samuel Vargas	Líder	Desarrollo de las funcionalidades de clasificación de tiempos y adición de flashcards.
	Experto	
	Técnico	
Cristian Montañez	Técnico	Desarrollo y perfeccionamiento de la funcionalidad de "Parejas", con nuevos atributos como la medición del tiempo. Además, trabajó en el código relacionado con el diseño de la interfaz.
	Investigador	
	Animador	
Daniel Rodríguez	Investigador	Diseño de la base de datos NoSQL. Trabajo en material de apoyo para la sustentación.
	Observador	
	Técnico	
Ana González	Secretaria	Ejecución de las pruebas del prototipo y análisis comparativo de la estructura de datos. Trabajo en el documento de la entrega.
	Observadora	
	Coordinadora	

XI. DIFICULTADES Y LECCIONES APRENDIDAS

Uno de los mayores obstáculos fue comprender Firebase, la plataforma de desarrollo de aplicaciones móviles en la nube. Su configuración, la gestión de la autenticación de usuarios y la comprensión de su estructura de base de datos nos llevaron tiempo y esfuerzo. Además, aprender a trabajar con una base de datos en un servidor y garantizar la seguridad de los datos resultó ser un desafío adicional, ya que requería una comprensión profunda de las reglas de acceso y la organización de la información. La conexión eficiente de Dart y Flutter con la base de datos Firebase también nos

supuso un reto técnico, que implicó la gestión de datos en tiempo real y la sincronización adecuada. En última instancia, estas dificultades nos enseñaron la importancia de una planificación detallada, la paciencia en la resolución de problemas y la colaboración efectiva en un equipo de

desarrollo. Además, reforzaron la necesidad de mantenernos actualizados con las últimas tecnologías y herramientas en constante evolución.