

# Studdy Buddy

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

## I. INTRODUCCIÓN

Studdy 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 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 primera 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 lineales (LinkedList, Array, Queue, Stack) 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 primera 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 que cuenten con las siguientes características de forma unificada:

1. Gran cantidad de ejercicios para las cuatro habilidades básicas del idioma.
2. Dinámicas variadas respecto al formato de los ejercicios.
3. Personalización de la rutina de estudio que se adapte a los conocimientos previos del usuario.
4. Almacenamiento y estrategias didácticas para el aprendizaje de vocabulario nuevo
5. Experiencia gamificada enfocada en la competición con compañeros de estudio (otros usuarios).

Es por ello que Studdy 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

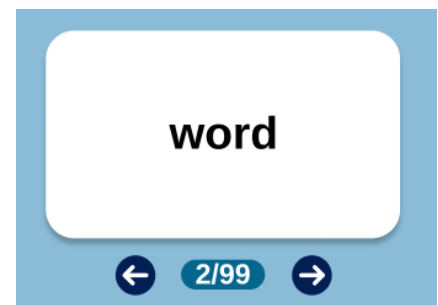
Esta aplicación está inicialmente pensada únicamente para estudiantes, que cumplen las siguientes características:

- 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 las lecciones, ejercicios de práctica, seguimiento de progreso y otros recursos de aprendizaje.

- Nivel de Acceso: Dependiendo del nivel, los estudiantes tienen acceso a lecciones y recursos correspondientes a su nivel de experiencia.
- 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

- **Consulta de flashcards de vocabulario**



### ○ Descripción:

Las flashcards funcionan como una tarjeta pequeña que tiene dos contenidos, uno por cada lado, por el frente estará la palabra en inglés que esté estudiando el usuario y en el reverso su respectiva traducción en español. Esta funcionalidad se enfoca en operaciones de almacenamiento, consulta total y búsqueda parcial de datos.

### ○ Acciones iniciadoras y comportamiento esperado:

1. El usuario visualiza una flashcard que contiene una palabra en inglés que el usuario decidió almacenar.
2. Si el usuario da click sobre la flashcard el programa mostrará la respectiva traducción de la misma en español.
3. Si el usuario vuelve a dar click sobre la flashcard con la traducción en español, esta retornará a su estado original en inglés.
4. El usuario tiene la posibilidad de ir revisando uno a uno los términos y traducciones de cada palabra almacenada como vocabulario nuevo. Esto a través de flechas ubicadas bajo las flashcards que

permitirán desplazarse hacia la izquierda o derecha para recorrer todas las flashcards creadas, el programa recorrerá la estructura de datos correspondiente e irá devolviendo todas las palabras almacenadas.

5. El usuario podrá visualizar la cantidad de flashcards totales y el número de la actual en el siguiente formato: 1/100. El programa irá actualizando los valores respecto al desplazamiento del usuario o el ingreso de nuevas palabras.
- *Requerimientos funcionales*
    - El sistema devolverá al usuario el término en inglés y al darle click, devolverá su traducción en español, al recibir nuevamente un click el sistema retornará nuevamente el término en inglés. El sistema se encargará de devolver la traducción o el término cuantas veces se de click.
    - Para lo anterior, el sistema deberá hacer una búsqueda parcial para acceder a la traducción de cada palabra rápidamente y retornarla al usuario.
    - El sistema permitirá ir actualizando la flashcard una a una de manera eficaz.
    - Al iniciar la aplicación por primera vez el usuario contará con una flashcard por defecto.
    - La cantidad mínima de flashcards que podrá tener el usuario es una flashcard.
    - El sistema devolverá la cantidad de flashcards totales y el número de la actual.
    - Los símbolos de flecha permitirán el desplazamiento entre flashcards.
  - **Práctica de traducción de palabras**
    - *Descripción:*  
Se enfoca en el aprendizaje de vocabulario. En esta actividad, se presentará una tarjeta (flashcard) con una palabra en inglés y en el reverso una pista/uso de la palabra en español, se solicitará al usuario que ingrese su traducción al español. El sistema debe evaluar la respuesta del usuario y proporcionar retroalimentación, indicando si la traducción es correcta o no. Se

almacenarán las palabras con las cuales el usuario falló para poder ser estudiadas de nuevo. Esta funcionalidad se enfoca en operaciones de almacenamiento, actualización, consulta total y búsqueda parcial de datos.

- *Acciones iniciadoras y comportamiento esperado:*
  1. *El usuario ingresa al apartado de traducción y el programa le da la bienvenida. El usuario debe darle a iniciar para empezar con los ejercicios de traducción.*
  2. *Al entrar el sistema devuelve una flashcard, un campo para completar, un botón para enviar la traducción y otro para revisar los errores. Desde esta pantalla el usuario puede realizar lo siguiente:*
    - *Ver la palabra en inglés.*
    - *El usuario da click sobre la palabra en inglés y así el sistema devuelve el significado/uso de la palabra en español.*
    - *El usuario puede ingresar la traducción que considere correcta y el programa debe verificar si acierta o no dando retroalimentación. El sistema devuelve emotes de la mascota de la aplicación que relacionan si el ejercicio fue correcto o incorrecto.*
    - *El usuario puede ver la lista de palabras en las que se equivocó. El sistema le permite además de ver la lista de palabras, acceder al detalle de cada una en el cual se encuentra el término en inglés, su traducción, su significado/uso y un ejemplo.*
    - *El usuario puede salir en el momento en el que desee y el sistema lo devolverá a la pantalla principal del apartado de Flashcards.*
- *Requerimientos funcionales:*  
Se abordarán los requerimientos funcionales a partir de las 4 secciones numeradas en la imagen siguiente:



### 1. Requerimientos funcionales de Flashcard:

- El sistema retornará al usuario el término en inglés y al darle click, buscará y devolverá su significado/uso en español, al recibir nuevamente un click el sistema retornará nuevamente el término en inglés. El sistema se encargará de consultar y devolver la traducción o el término cuántas veces se de click.
- El sistema no cambiará de flashcard hasta que se ingrese una traducción en el campo correspondiente y se verifique.
- Después de la retroalimentación, el sistema debe cambiar de palabra cada vez que se dé al botón de “Siguiente”.

### 2. Requerimientos funcionales de “Completar”:

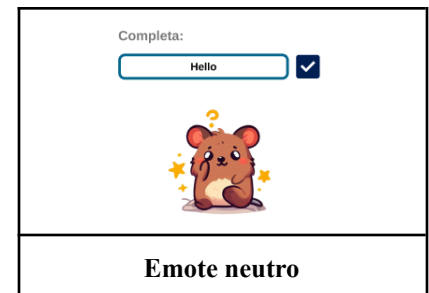
- El campo de traducción no aceptará valores numéricos, solo aceptará caracteres alfabéticos y especiales.
- Si el usuario desea continuar a la siguiente flashcard, debe ingresar mínimo un carácter en el campo y darle al icono de verificar. De lo contrario, se quedará en la misma pantalla.

- La interfaz debe responder a la retroalimentación, antes de esta debe mantener su estado neutro, después según el estado de incorrecto o correcto debe cambiar el color del campo. Así:

Completa: <input type="text" value="Hello"/> ✓	Completa: <input type="text" value="Hola"/> ✓	Completa: <input type="text" value="Buen día"/> ✓
<b>Estado neutro</b>	<b>Estado correcto</b>	<b>Estado incorrecto</b>

### 3. Requerimientos funcionales de “Emote”

- El software en estado inicial, es decir, antes de realizar una verificación debe mostrar en la interfaz el siguiente emote, el cual se asocia al comportamiento del campo de traducción en estado neutro:



- El software debe poder emitir la retroalimentación al dar click sobre el icono de verificar y mostrar en la interfaz lo siguiente:
  - Si la entrada del usuario en el campo corresponde con la traducción de la palabra almacenada, el sistema deberá devolver el emote de “correcto”.
  - En caso contrario, el sistema deberá devolver el emote de “incorrecto” y esa palabra deberá ser almacenada en la lista de “errores”.

Completa: Hola  	Completa: Buen día  
Emote correcto	Emote incorrecto

#### 4. Requerimientos funcionales “Ver errores”

- El sistema permitirá al usuario visualizar la lista de palabras en las que ha cometido algún error.
- El sistema actualizará la lista de errores conforme el desarrollo de la práctica.
- El sistema consultará y retornará al usuario la información de la palabra si el usuario da clic sobre esta en la lista.
- Si el usuario no comete errores, la lista estará vacía.
- La interfaz gráfica debe responder de la siguiente manera:

		
Estado neutro	Lista de errores	Información de la palabra

#### • Desafío de parejas

##### ○ Descripción:

El desafío de parejas se enfoca en el proceso de aprendizaje de idiomas. Presenta cinco filas de dos columnas de flashcards en la pantalla, que contienen palabras en inglés y sus respectivas traducciones en desorden. El estudiante debe

seleccionar la palabra en inglés junto con su respectiva traducción al español y completar todas las parejas.

##### ○ Acciones iniciadoras y comportamiento esperado:

1. El usuario ingresa al apartado de parejas y el programa le da la bienvenida. El usuario debe darle a iniciar para empezar con el desafío de parejas.
2. El programa debe mostrar dos columnas con flashcards en español e inglés en desorden y el usuario debe emparejarlas. El programa debe responder en el caso de que la pareja sea correcta e incorrecta.
3. Si el usuario empareja correctamente, las casillas del término y la traducción desaparecerán. De otra forma, permanecerán hasta que se emparejen correctamente.
4. Una vez que el usuario complete las 5 parejas el sistema retornará un mensaje de felicitación y la opción de regresar al menú inicial o volver a jugar.

##### ○ Requerimientos funcionales:

- El sistema muestra una pantalla de bienvenida con la opción de regresar al menú principal o de iniciar el juego.
- El sistema consulta la lista de flashcards y elige 5 al azar para generar las parejas.
- El sistema muestra la matriz 5x2 de flashcards a partir de los seleccionados términos y traducciones.
- El sistema comprueba si la selección de la pareja corresponde a una combinación término-traducción existente en la estructura de datos que almacena el vocabulario del usuario.
- Después de la comprobación, el sistema es capaz de responder así:
  - Si la combinación es correcta, muestra iconos de estrella y elimina las casillas de la pantalla.
  - Si la combinación es incorrecta, muestra iconos de “x” y mantiene las casillas en la pantalla.

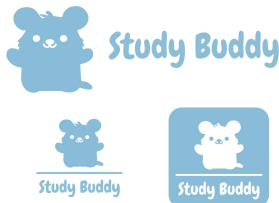
<div> <div>red</div> <div>cafe</div> <div>blue</div> <div>rojo</div> </div>	<div> <div>cafe</div> <div>blue</div> </div>
Pareja correcta	Pareja incorrecta

- El sistema verifica la cantidad de casillas restantes, cuando esta es igual a 0 acaba el juego.
- El sistema retorna al usuario una pantalla de felicitación con la opción de regresar al menú principal o jugar de nuevo.

## V. DESCRIPCIÓN DE LA INTERFAZ DE USUARIO PRELIMINAR

La creación del prototipo de Study Buddy se enfocó en proporcionar a los usuarios una experiencia de aprendizaje dinámica y visualmente atractiva. Para ello, se definió un conjunto de recursos como una gama de colores, fuentes definidas, logo y mascota propios del branding de la aplicación. Los siguientes son los recursos utilizados:

- **Gama de colores:**
  - Azul oscuro: #021f54
  - Azul rey: #05668d
  - Azul claro: #8abcd7
  - Amarillo: #fefb64
  - Verde: #6EBA6C
  - Rojo: #FF0000
  - Gris: #7A7A7A
- **Fuentes:**
  - Chewy
  - Arimo
- **Logo**



- **Mascota:** Se definió a un quokka como mascota de la app, esto en pro de la gamificación de la experiencia de aprendizaje.



El prototipo fue realizado utilizando el programa Figma y está diseñado para ser intuitivo y fácil de usar. El prototipo se puede visualizar aquí.

Por otro lado, la interfaz de usuario de nuestra aplicación se dividirá en 5 categorías que se pueden identificar en el menú inferior de la aplicación. De izquierda a derecha son: ejercicios de práctica(Home), flashcards, competición, estadísticas y perfil de usuario.



En esta entrega el enfoque fue el apartado de flashcards, dentro de esta categoría se encuentran las funcionalidades consulta de flashcards de vocabulario, práctica de traducción de palabras y desafío de parejas descritas en el numeral anterior.

## 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.

## VII. PROTOTIPO DE SOFTWARE INICIAL

Para esta entrega se inició el desarrollo de la aplicación mediante un repositorio alojado en Github que 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 implementación de la actividad de traducción, se hace uso de arreglos como estructura de datos clave. Estos arreglos son utilizados para permitir el acceso eficiente a las palabras individuales y para almacenar errores durante el proceso de traducción. Aunque los arreglos en sí no establecen una correspondencia explícita entre las palabras en los idiomas de origen y destino, desempeñan un papel fundamental al facilitar el acceso secuencial a las palabras y la capacidad de registrar posibles errores.

Para la actividad de parejas, se ha seleccionado la lista doblemente enlazada. Esto debido a la necesidad de realizar eliminaciones de elementos que pueden estar ubicados en diversas partes de la estructura. Durante la actividad, puede ser necesario eliminar elementos tanto al inicio, al final como en cualquier posición intermedia de la lista, y la lista doblemente enlazada se destaca por su capacidad para manejar estas operaciones de manera eficiente.

La lista doblemente enlazada permite un acceso rápido tanto al elemento anterior como al siguiente de cada nodo en la lista, lo que resulta esencial para realizar las eliminaciones de manera efectiva sin necesidad de recorrer toda la estructura desde el principio cada vez que se necesita eliminar un elemento en cualquier posición. Esta característica agiliza considerablemente el proceso de emparejamiento y eliminación de elementos en la actividad de parejas, contribuyendo a una implementación más eficiente y efectiva de la tarea.

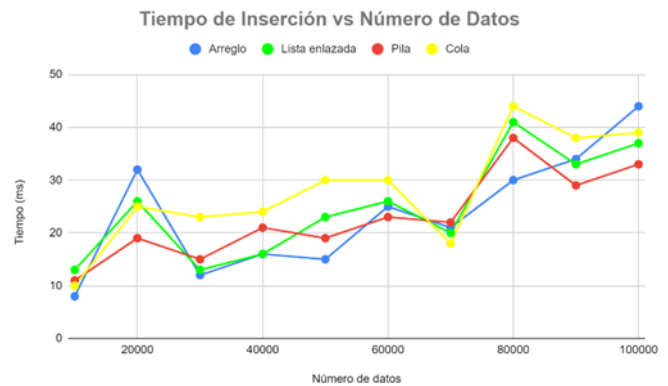
## IX. PRUEBAS DEL PROTOTIPO Y ANÁLISIS COMPARATIVO

Para probar la complejidad de las distintas estructuras se sometieron a pruebas con un número elevado de datos para 3 operaciones distintas para 4 estructuras, Array, LinkedList con cola, Stack y Queue. Estas comprobaciones se hicieron para 2 tipos de elementos, la clase Palabra, cuyos atributos son String 'español', String 'inglés', String 'definición' y List<String>'ejemplos'; y el tipo de dato primitivo String.

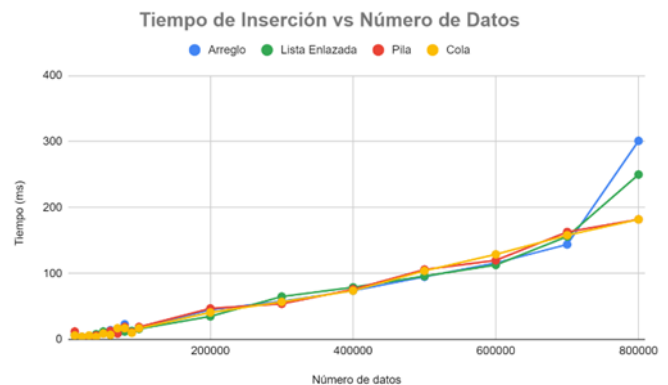
### Análisis de inserción:

Primero se hizo el análisis de inserción de datos para las 4 estructuras, arrojando los siguientes resultados.

Para la clase Palabra:



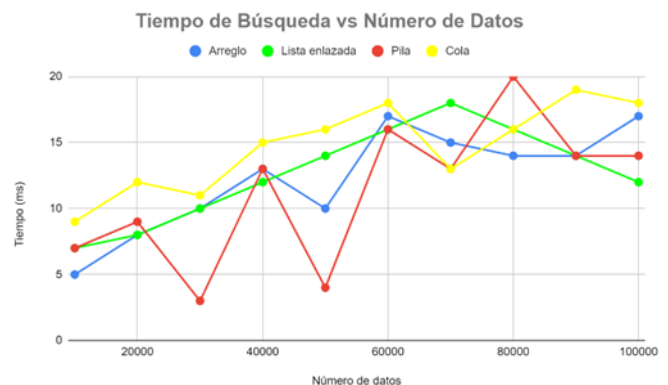
Para el tipo String:



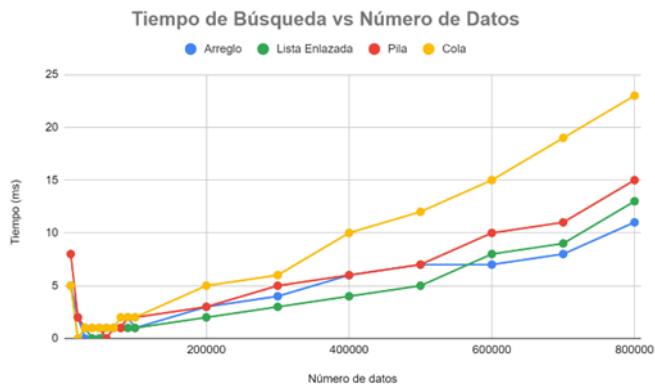
### Análisis de búsqueda:

Luego se realizó la operación de búsqueda, que consistía en buscar un dato que se encontraba alrededor de la mitad del conjunto, además de buscar uno que no se encontraba en lo absoluto. Con lo cual se obtuvieron los siguientes resultados:

Para la clase Palabra:



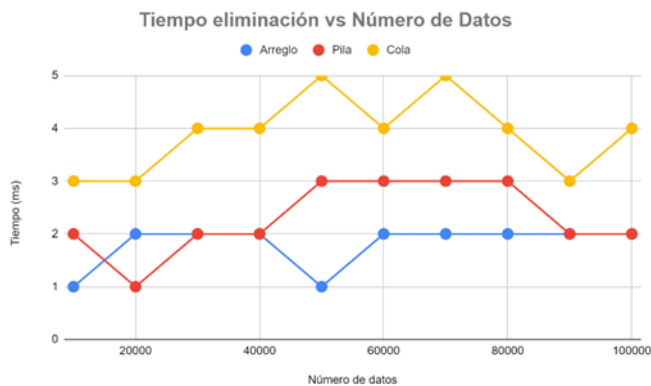
Para el tipo String:



### Análisis de eliminación:

Por último, se realizó la operación eliminación, donde se presentó un aumento significativo cuando se operaba sobre la estructura LinkedList, ya que para realizar la eliminación se requirió una complejidad  $O(n)$ .

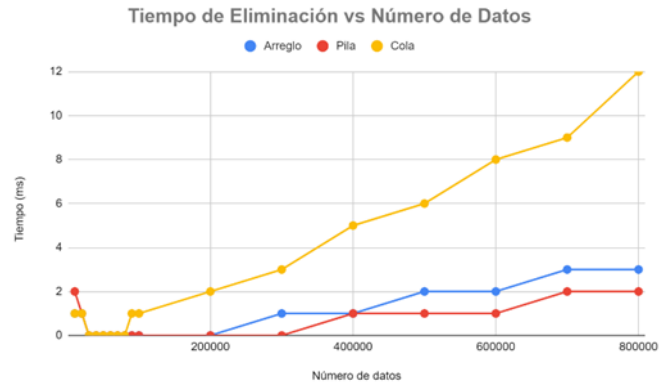
Para la clase Palabra:



Debido al cambio brusco de escala, se requirió realizar un gráfico adicional para la estructura LinkedList:



Para el tipo String:



La observación de todos los datos recopilados se puede encontrar en el siguiente enlace [https://docs.google.com/spreadsheets/d/15\\_19VtxrANv36vJ31flpz2V-8FJXNDZHtqy96wMk2U/edit?usp=sharing](https://docs.google.com/spreadsheets/d/15_19VtxrANv36vJ31flpz2V-8FJXNDZHtqy96wMk2U/edit?usp=sharing) y en el repositorio, dentro de la carpeta correspondiente.

## X. ROLES Y ACTIVIDADES

Para el desarrollo adecuado del proyecto, se decidió distribuir los roles de trabajo del grupo de la siguiente manera:

Integrante	Rol(es)	Actividades Realizadas
Samuel Vargas	Líder	<ul style="list-style-type: none"> <li>Desarrollar el funcionamiento de la interfaz utilizando Flutter.</li> <li>Implementar las estructuras</li> </ul>
	Experto	
	Técnico	



		de datos en Dart <ul style="list-style-type: none"> <li>Desarrollar el apartado de 'Traducción' para el aprendizaje de una palabra específica</li> </ul>
Cristian Montañez	Técnico	<ul style="list-style-type: none"> <li>Desarrollar el apartado de 'Parejas' para practicar la traducción de diversas palabras.</li> </ul>
	Investigador	
	Animador	
Daniel Rodríguez	Investigador	<ul style="list-style-type: none"> <li>Desarrollar el apartado de 'Flashcards' para repaso de vocabulario aprendido</li> </ul>
	Observador	
	Técnico	
Ana González	Secretaria	<ul style="list-style-type: none"> <li>Diseñar el prototipo de la interfaz según las necesidades de las funcionalidades buscadas.</li> </ul>
	Observadora	
	Coordinadora	

A pesar de las dificultades, se aprendió mucho sobre el mundo del desarrollo de aplicaciones móviles, la importancia de reconocer las limitaciones técnicas de los equipos para realizar un análisis comparativo de estructuras de datos y en general el valor de una buena planificación para cumplir con las metas establecidas de cada entrega.

## XI. DIFICULTADES Y LECCIONES APRENDIDAS

Una de las principales dificultades que se enfrentó fue el aprendizaje de un nuevo lenguaje de programación. Flutter se basa en Dart, un lenguaje de programación relativamente nuevo que tiene algunas características únicas, como la existencia de null-safety, que obliga un control más estricto del valor null, aumentando la robustez del código. Esto requirió que se tomara el tiempo para aprender el nuevo lenguaje y sus características.

Otra dificultad que fue afrontada fue la generación de datos coherentes con el proyecto. Para poder probar la aplicación, era necesario generar datos que fueran realistas y que se ajustaran a las necesidades del proyecto. Sin embargo, esto fue bastante limitado tanto por el tiempo como por el almacenamiento de los dispositivos.