

STU-AID: ASISTENTE DE MEMORIA BASADO EN UN SISTEMA DE
REPETICIÓN ESPACIADA PARA LA AYUDA DE ESTUDIOS ACADÉMICOS.

JUAN PABLO AVILAN MORENO

UNIVERSIDAD PILOTO DE COLOMBIA

ESCUELA TIC

INGENIERÍA DE SISTEMAS

BOGOTÁ

2023

STU-AID: ASISTENTE DE MEMORIA BASADO EN UN SISTEMA DE
REPETICIÓN ESPACIADA PARA LA AYUDA DE ESTUDIOS ACADÉMICOS.

JUAN PABLO AVILAN MORENO

Proyecto de grado para optar al título de ingeniero de sistemas

Asesor de tesis

Giovanni Fajardo Utria

Ingeniero de sistemas.

UNIVERSIDAD PILOTO DE COLOMBIA

ESCUELA TIC

INGENIERÍA DE SISTEMAS

BOGOTÁ

2023

Nota de aceptación

Firma de presidente del jurado

Firma del jurado

Firma del jurado

Bogotá, 26 de junio de 2023

AGRADECIMIENTOS

Agradezco a mis padres por el apoyo incondicional, el amor y las enseñanzas valiosas que me han servido para formarme como persona.

A toda la comunidad de profesores de la Universidad Piloto de Colombia por la formación, conocimientos y apoyo durante mi vida académica, en especial al ingeniero Giovanni Fajardo Utria por su guía y ayuda durante las asesorías para la realización de este proyecto.

También, agradezco a mis compañeros de la universidad por su compañía en cada uno de los trabajos en grupo, la camarería y el impulso que me dieron por seguir aprendiendo.

CONTENIDO

	pág.
INTRODUCCIÓN	16
1. GENERALIDADES DEL PROBLEMA	17
1.1 PLANTEAMIENTO DEL PROBLEMA	17
1.2 JUSTIFICACIÓN	18
1.3 PREGUNTA DE INVESTIGACIÓN	18
1.4 OBJETIVOS	19
1.4.1 Objetivo general	19
1.4.2 Objetivos específicos	19
1.5 ALCANCE	19
2. MARCO REFERENCIAL	21
2.1 MARCO CONCEPTUAL	21
2.1.1 Sistema de repetición espaciada	21
2.1.2 Curva de olvido	22
2.1.3 Método de Leitner	23
2.2 ANTECEDENTES	24
2.2.1 Anki	24
2.2.2 Memrise	25
2.2.3 Cuadro comparativo de aplicaciones	27
2.3 MARCO ACTUAL	27

2.3.1 Europa	28
2.3.2 Norteamérica	28
2.3.3 Latinoamérica	29
2.3.4 Nivel nacional	30
3. METODOLOGÍA	31
3.1 METODOLOGÍA DE INVESTIGACIÓN	31
3.2 METODOLOGÍA DE DESARROLLO	31
4. ESPECIFICACIÓN DE REQUERIMIENTOS	34
4.1 INVESTIGACIÓN DE LAS TÉCNICAS DE ESTUDIO DE REPETICIÓN ESPACIADA, RECUPERACIÓN ACTIVA Y EL MÉTODO DE FLASHCARDS DE LEITNER	34
4.2 REQUERIMIENTOS NO FUNCIONALES	35
4.2.1 Usabilidad	36
4.2.2 Seguridad	37
4.2.3 Desempeño	38
4.3 IDENTIFICACIÓN DEL PERFIL DE USUARIO	38
4.3.1 Perfil de usuario segmentado por el buyer persona	38
4.4 ESPECIFICACIÓN DE REQUERIMIENTOS	40
4.4.1 Historia de usuario	41
4.4.2 Especificación de historias de usuario	42
5. DISEÑO DE ARQUITECTURA	47
5.1 ESTRATEGIAS DE ARQUITECTURA	47
5.2 DECISIONES DE ARQUITECTURA	48
5.2.1 La Arquitectura limpia (The clean architecture)	49
5.2.2 Decisiones de la aplicación backend	51

5.2.2.1 Express middlewares y la arquitectura pipes and filters	52
5.2.2.2 Postgresql	53
5.2.2.3 Sequelize ORM	54
5.2.2.4 JSON Web Tokens JWT y Passport.js	55
5.2.2.5 Docker	56
5.2.3 Decisiones de la aplicación frontend	57
5.2.3.1 Aplicación móvil construida con Expo y React Native	57
5.2.3.2 Expo	57
5.2.3.3 React Native	57
5.3 MODELO DE DATOS	58
5.3.1 Usuarios	58
5.3.2 Cursos	58
5.3.3 Temas	58
5.3.4 Flashcards	58
6. IMPLEMENTACIÓN	60
6.1 IMPLEMENTACIÓN DE LOS MÓDULOS DEL BACKEND	60
6.1.1 Capa de adaptadores externos.	60
6.1.2 Capa de controladores	61
6.1.2.1 Middlewares de enrutación	62
6.1.2.2 Middleware de autenticación	63
6.1.2.3 Middleware de autorización de usuario	64
6.1.2.4 Middlewares de manejo de errores	65
6.1.3 Capa de casos de uso	66
6.1.3.1 Servicio de usuario	67

6.1.3.2 Servicio de curso	68
6.1.3.3 Servicio de tema	69
6.1.3.4 Servicio de flashcards	70
6.1.3.5 Servicio de sesión de estudio	71
6.1.4 Capa de entidades	72
6.2 IMPLEMENTACIÓN DE LOS MÓDULOS DEL FRONTEND.	76
6.2.1 Componentes estilizados	76
6.2.2 Uso de custom hooks para llamados del API	78
6.2.3 Modo oscuro	79
6.2.4 Autenticación	80
6.2.5 Cursos	82
6.2.6 Temas	86
6.2.7 Flashcards	90
6.2.8 Sesión de estudio	94
7. PRUEBAS.	97
7.1 DISEÑO DEL AMBIENTE DE PRUEBAS.	97
7.2 PRUBAS DE INTEGRACIÓN.	99
8 CONCLUSIONES.	120
9 RECOMENDACIONES.	121
REFERENCIAS	122
ANEXOS	127

LISTA DE TABLAS

	pág.
Tabla 1: Cuadro comparativo de aplicaciones.	27
Tabla 2: Metodología de desarrollo	32
Tabla 4: Atributos de calidad priorizados.	36
Tabla 5: Historias de usuario.	44
Tabla 6: Estrategias de arquitectura por atributo de calidad	48
Tabla 7: Prueba de integración HU-001	99
Tabla 8: Prueba de integración HU-002	100
Tabla 9: Prueba de integración HU-003	101
Tabla 10: Prueba de integración HU-004	102
Tabla 11: Prueba de integración HU-005	103
Tabla 12: Prueba de integración HU-007	104
Tabla 13: Prueba de integración HU-008.	105
Tabla 14 Prueba de integración HU-009	106
Tabla 15: Prueba de integración HU-010.	107
Tabla 16: Prueba de integración HU-011	107
Tabla 17: Prueba de integración HU-012.	108
Tabla 18: Prueba de integración HU-013	109
Tabla 19: Prueba de integración HU-014	110
Tabla 20: Prueba de integración HU-015	111

Tabla 21: Prueba de integración HU-016	112
Tabla 22: Prueba de integración HU-017.	113
Tabla 23: Prueba de integración HU-018.	113
Tabla 24: Prueba de integración HU-019.	114
Tabla 25: Prueba de integración HU-020	115
Tabla 26: Prueba de integración HU-021	116
Tabla 27: Prueba de Integración HU-022	117
Tabla 28: Prueba de integración HU-023	118
Tabla 29: Prueba de integración HU-024.	119

LISTA DE FIGURAS

	pág.
Figura 1: Curva del olvido de Ebbinghaus.	22
Figura 2: Sistema de Leitner	23
Figura 3: Aplicación Anki.	25
Figura 4: Aplicación Memrise.	26
Figura 5: Buyer Persona	39
Figura 6: Tablero backlog de historias de usuario.	43
Figura 7: La arquitectura limpia.	50
Figura 8: Comportamiento de Pipes and filters	53
Figura 9: Modelo de datos ERD.	59
Figura 10: Creación del servidor HTTP	61
Figura 11: Configurando los middlewares	62
Figura 12: Configurando middlewares de enrutación	63
Figura 13: Middleware de autenticación de usuarios	64
Figura 14: Middleware de autorización	65
Figura 15: Middleware de errores HTTP	66
Figura 16: Servicio de usuarios.	67
Figura 17: Servicio de curso.	68
Figura 18: Servicio de tema	69

Figura 19: Servicio de Flashcards	70
Figura 20: Servicio de sesión de estudio.	71
Figura 21: Entidad de usuario	72
Figura 22: Entidad de curso	73
Figura 23: Entidad de tema	74
Figura 24 : Entidad de flashcard	75
Figura 25: Ejemplo de texto estilizado	76
Figura 26: Componente de texto estilizado.	77
Figura 27: Custom hook para una petición GET	78
Figura 28: Modo oscuro y modo claro	79
Figura 29: Vista de inicio de sesión	80
Figura 30: Vista de registro de cuenta	81
Figura 31: Vista de cursos del usuario.	82
Figura 32: Vista de añadir curso.	83
Figura 33: Vista de editar curso.	84
Figura 34: Eliminar curso.	85
Figura 35: Vista de ver temas de estudio.	86
Figura 36: Vista de crear un tema nuevo.	87
Figura 37: Vista de editar tema de estudio.	88
Figura 38: Vista de eliminar tema de estudio.	89
Figura 39: Vista de ver flashcards.	90
Figura 40: Vista de crear flashcards frente-reverso, verdadero-falso y elaborada.	91
Figura 41: Vista de editar flashcards.	92

Figura 42: Vista de eliminar flashcard.	93
Figura 43: Vista de ver temas para repasar.	94
Figura 44: Vista de estudiar con temporizador.	95
Figura 45: Vista de sesión de repaso.	96
Figura 46: Creación del archivo seed.	98

GLOSARIO

ANDROID: Es un sistema operativo para teléfonos móviles.

API REST: Es una interfaz de comunicación entre sistemas de información basado en una arquitectura cliente servidor.

CLIENTE: Es una aplicación que envía y recibe peticiones del servidor.

DOCKER: Es una plataforma que permite la creación y gestión de contenedores que es la virtualización de un sistema operativo para correr aplicaciones.

EXPO: Es un framework para desarrollar aplicaciones con React Native de manera más rápida.

FRAMEWORK: Es un marco de trabajo que proporciona herramientas que facilitan el desarrollo de una aplicación.

HTTP: Es un protocolo que es utilizado en internet para la comunicación entre sistemas de información.

JAVASCRIPT: Es un lenguaje de programación. Fue creado en 1995 y es principalmente usado en aplicaciones web.

LIBRERIA: Es un módulo de código que provee un conjunto de herramientas para solucionar un problema en específico.

NODE JS: Es un ambiente de código abierto que permite correr JavaScript del lado del servidor.

ORM: Es un modelo que permite mapear estructuras de una base de datos para facilitar la comunicación y manipulación de las entidades del modelo de datos.

REACT NATIVE: Es una librería usada para desarrollar aplicaciones nativas para Android, Android TV, IOS y web.

SEQUELIZE: Es un ORM compatible con aplicaciones de JavaScript.

SERVIDOR: Es una aplicación que envía y recibe peticiones del cliente.

RESUMEN

Stu-aid es un aplicativo móvil que implementa un sistema de repetición espaciada utilizando un algoritmo basado en la curva de aprendizaje y la curva de olvido. Funciona por medio de sesiones de repaso de conceptos programadas, que ponen a prueba el conocimiento del usuario y su memoria, a fin de mejorar su retención y disminuir las falsas ilusiones de conocimiento.

Este documento tiene la intención de dar conocer aspectos importantes que se llevaron a cabo para la realización del proyecto y los resultados que se obtuvieron en cada una de las fases del ciclo del software de requerimientos, diseño, implementación y pruebas.

Para la implementación de la solución se utilizó una arquitectura cliente servidor para dar cumplimiento a los requerimientos funcionales y no funcionales. En la parte del cliente, se desarrolló la aplicación para Android utilizando Javascript y la librería React Native. Por la parte del servidor, se desarrolló una aplicación API REST utilizando Node.JS y la librería Express.js.

PALABRAS CLAVE: SISTEMA DE REPETICIÓN ESPACIADA, CURVA DE OLVIDO, MÉTODO DE FLASHCARDS DE LEITNER, ANDROID, REACT NATIVE, NODE JS, CLIENTE SERVIDOR.

INTRODUCCIÓN

La memoria juega un papel fundamental en el aprendizaje. Cuando se es capaz de retener cada concepto, es fácil, utilizar ese conocimiento, aplicarlo a la situación adecuada y cambia por completo la forma en que se ven nuevas oportunidades y amenazas en el entorno¹. No obstante, existe una problemática en la forma que aprendemos y retenemos el conocimiento, es conocida como la curva del olvido propuesta por el psicólogo alemán Herman Ebbinghaus, la cual concluye por investigación que el porcentaje de retención de conocimiento disminuye en función del tiempo que transcurre desde el primer día que se aprende.² Al tener esta condición como seres humanos, surgieron distintos métodos y técnicas de estudio que permiten combatir el olvido de conocimiento, como lo es la repetición espaciada, la práctica espaciada, y el recuerdo activo.

El presente documento tiene la intención de dar a conocer los aspectos importantes para llevar a cabo el proyecto de STU-AID: Asistente de memoria basado en un sistema de repetición espaciada para la ayuda de estudios académicos, que busca diseñar e implementar una aplicación móvil que hace uso de técnicas de estudio para crear un sistema que funcione como asistente de memoria para alumnos, en donde cada uno pueda ingresar al sistema los conceptos que considere importantes y quiere retener a largo plazo en forma de unidades compactas de información, específicamente, flashcards digitales.

¹ BROWN, Peter C. Make it stick. [s.l.]: Harvard University Press, 2014. 329 p. ISBN 9780674419377.

² EBBINGHAUS, Hermann. Memory: A Contribution to Experimental Psychology. [s.l.]: Creative Media Partners, LLC, 2018. ISBN 9780344159756.

1 GENERALIDADES DEL PROBLEMA

1.1 PLANTEAMIENTO DEL PROBLEMA

Según Ebbinghaus³, la curva del olvido plantea como la retención de la información decae a lo largo del tiempo, a menos de que sea repasada por metodologías de estudio.

En la vida académica y profesional es importante tener acceso a la información, ya que, permite ver oportunidades y riesgos, además de maneras de cómo solucionar un problema, la capacidad de tener acceso a esta información y de saber aplicarla de la manera adecuada conlleva al éxito académico y profesional. De acuerdo con el modelo de Ebbinghaus, tan solo 24 horas después de haber aprendido cierta información, se tiene una retención promedio de alrededor del 80% (esto es una estimación, ya que depende de las características de cada persona).

Para evitar olvidar lo que aprendemos, han surgido muchas técnicas y métodos de estudio que abordan esta condición del cerebro. Una de estas técnicas se llama la repetición espaciada, la cual consiste en repasar y poner a prueba lo que aprendemos de manera estratégica. Ya que se trata de un problema en función del tiempo, la repetición espaciada programa las sesiones de repaso en intervalos de tiempo que van incrementando a medida que se repasa y se refuerza un conocimiento. De esta forma, se plantean preguntas y si estas son respondidas de manera acertada, son conocimientos que se programan para repasar en un intervalo cada vez mayor, por lo contrario, si son respondidas de manera errónea, deben ser repasadas con más frecuencia, hasta lograr una mejor retención del conocimiento.

³ EBBINGHAUS, op. cit.

Los resultados de aplicar técnicas de estudio como la repetición espaciada evitan que se olvide la información y que esta pueda estar a la mano, lo cual mejora la productividad, competitividad y competencias en la vida académica y profesional.

1.2 JUSTIFICACIÓN

Implementar un sistema de repetición espaciada es una herramienta que ha sido probada de ayudar a la retención y memoria de conocimientos aprendidos. El problema es que implementar un sistema de forma manual resulta tedioso y poco escalable para las personas, ya que se tendría que llevar un registro manual de repasos y un cálculo manual de la siguiente sesión de repaso para cada pregunta.

Es pertinente entonces, implementar esta funcionalidad en una aplicación móvil, que se encargue de facilitar el hábito de repasar al programar de manera automática las sesiones y frecuencia de tiempo de repetición de cada pregunta.

1.3 PREGUNTA DE INVESTIGACIÓN

¿Cómo se implementa un sistema de repetición espaciada en una aplicación móvil que sea fácil de utilizar y sea segura para los usuarios, que facilite el hábito de repasar al programar de manera automática las sesiones y frecuencia de tiempo de repetición de cada pregunta?

1.4 OBJETIVOS

1.4.1 Objetivo general

Realizar un aplicativo móvil que implemente un sistema de repetición espaciada utilizando un algoritmo basado en la curva de aprendizaje y la curva del olvido que programe sesiones de repaso de conceptos en los que se ponga a prueba el conocimiento del usuario y su memoria, a fin de mejorar su retención de conocimiento y disminuir las falsas ilusiones de conocimiento.

1.4.2 Objetivos específicos

- Identificar los requerimientos del aplicativo móvil teniendo en cuenta el modelo de la curva de olvido y las técnicas de estudio.
- Diseñar un modelo de arquitectura que se adecue a los requerimientos funcionales y no funcionales de tal manera que sean claros los componentes, relaciones, estructuras de control de flujo de datos y patrones de diseño.
- Desarrollar la aplicación siguiendo el modelo de arquitectura y cumpliendo con todos los requerimientos levantados.
- Validar el producto de software usando pruebas de integración al producto para garantizar el correcto funcionamiento de cada requerimiento que se levantó y así asegurar la calidad.

1.5 ALCANCE

El sistema de repetición espaciada que se quiere implementar en la aplicación móvil de Stu-aid contempla una herramienta en la que el usuario pueda:

- Incorporar las materias o cursos que este viendo, o que ya vió.
- Incorporar temas en cada uno de los cursos.
- Incorporar cuestionarios (conjunto de preguntas) que conformaran un tema en particular.
- Visualizar cada pregunta de un tema en forma de flashcard.

- Estudiar y responder conjuntos de flashcards que necesitan ser repasadas según la curva de olvido y el algoritmo de repetición espaciada en sesiones de estudio personalizadas.

Se espera especificar los requerimientos funcionales y no funcionales pertinentes para que la aplicación móvil cumpla con las características de un sistema de repetición espaciada que utiliza el método de flashcards de Leitner y que automatice la generación de sesiones de repaso.

Así mismo, se busca desarrollar un modelo de arquitectura apto para los requerimientos, para que después sea implementado por medio del desarrollo de los distintos módulos.

Al final, se espera que se ponga a prueba la implementación de la herramienta desarrollada, a través de pruebas de integración y la formulación de criterios de aceptación.

2 MARCO REFERENCIAL

2.1 MARCO CONCEPTUAL

2.1.1 Sistema de repetición espaciada

Es una herramienta que implementa la técnica de estudio de repetición espaciada, por lo que, esta consiste en conceptos y temas guardados en un tipo de contenido (video, notas, flashcards, entre otros) que se repasan siguiendo una programación de sesiones de acuerdo con el nivel de retención que se tenga. Un sistema de repetición espaciada debe tener las siguientes características⁴:

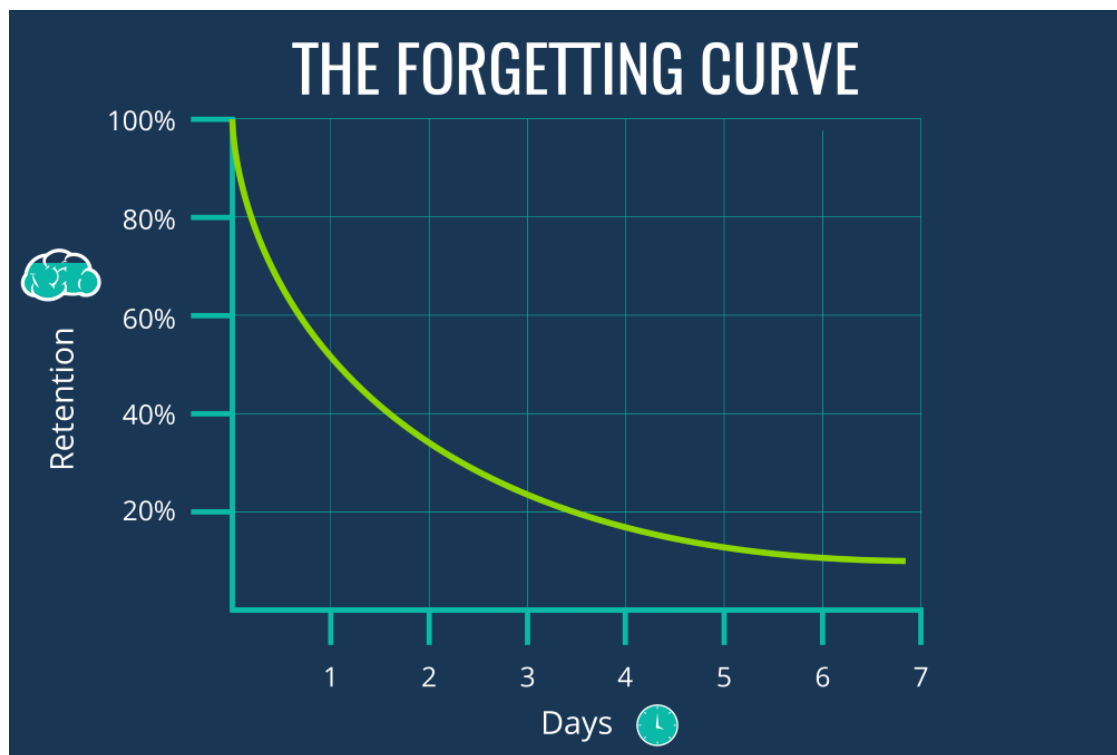
- Reúne el conjunto de conceptos y temas que se quieren memorizar y los liga a una pregunta, o problema.
- Planea el repaso de cada concepto de acuerdo con el nivel de retención del usuario. (Entre mayor capacidad de recordar el contenido, mayor será el tiempo entre repasos).
- Registra la respuesta y retroalimentación del usuario al responder cada pregunta para que, de esta manera, se programe cuando se debe repasar de nuevo.

⁴ BROWN, op. cit.

2.1.2 Curva de olvido

A medida que se aprenden conceptos y temas a lo largo de la vida académica y profesional existen distintos factores como la atención, la memoria y las heurísticas utilizadas para aprender. Entre estos factores, la memoria juega un papel importante para lograr la retención de temas y conceptos que ya se han aprendido. No obstante, existe un problema, y es que a medida que transcurre el tiempo, nuestro cerebro olvida de manera drástica lo que aprendemos. Esta problemática fue definida por el psicólogo Herman Ebbinghaus como la curva del olvido⁵, la cual, es un modelo que expone el nivel de retención de un concepto aprendido, en función de días transcurridos (ver la siguiente imagen):

Figura 1: Curva del olvido de Ebbinghaus.



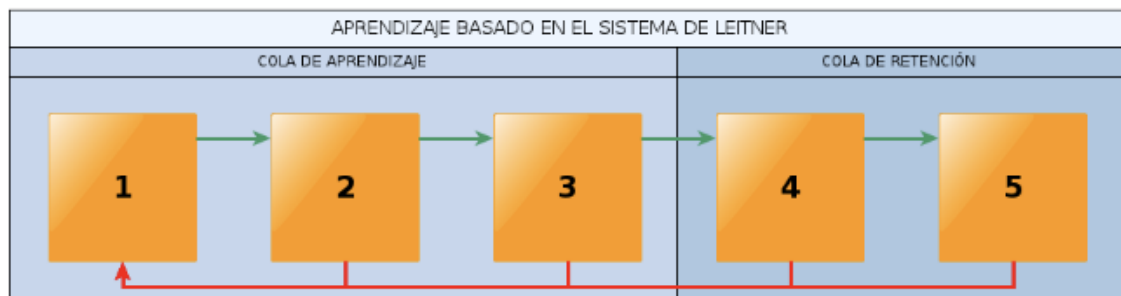
Fuente: <https://elearningindustry.com/forgetting-curve-combat>

⁵ EBBINGHAUS, op. cit.

2.1.3 Método de Leitner

En 1970 Sebastián Leitner, un divulgador y científico alemán que se centró en el estudio de temas médicos y psicológicos, diseñó un método de aprendizaje basado en el uso de tarjetas de preguntas y respuestas que permitían acortar el tiempo de estudio a medio y largo plazo. A continuación, se muestra un esquema del método de Leitner en el cual se observa que en el momento de empezar el estudio todas las tarjetas se colocan en la caja número 1 y, a medida que se van acertando, se colocan en la caja número 2, número 3 y así sucesivamente. En caso de fallar una tarjeta se coloca siempre en la primera caja. El estudio se realiza espaciadamente de forma que el primer día se estudian las tarjetas pertenecientes a la caja número 1; al día siguiente se vuelven a estudiar las tarjetas de la caja número 1 y dos días después se repasan las de la caja número 2. Entre tres y cinco días después de haber iniciado el estudio, se repasan las de la caja número 3 y así sucesivamente⁶.

Figura 2: Sistema de Leitner



Fuente:

<https://upcommons.upc.edu/bitstream/handle/2099.1/16934/77878.pdf?sequence=1&isAllowed=y>

⁶ Gomez, María. Aplicación para memorizar usando repetición espaciada. 2013. [consultado el 11, septiembre, 2022]. Disponible en Internet: <https://search.ebscohost.com/login.aspx?direct=true&db=edsoai&AN=edsoai.on1133019066&lang=es&site=eds-live>.

2.2 ANTECEDENTES

2.2.1 Anki

Según el sitio web⁷, Anki es un programa que hace que recordar cosas sea fácil. Como es mucho más eficiente que los métodos de estudio tradicionales, puede reducir en gran medida el tiempo que se le dedica al estudio, o aumentar en gran medida la cantidad que se aprende.

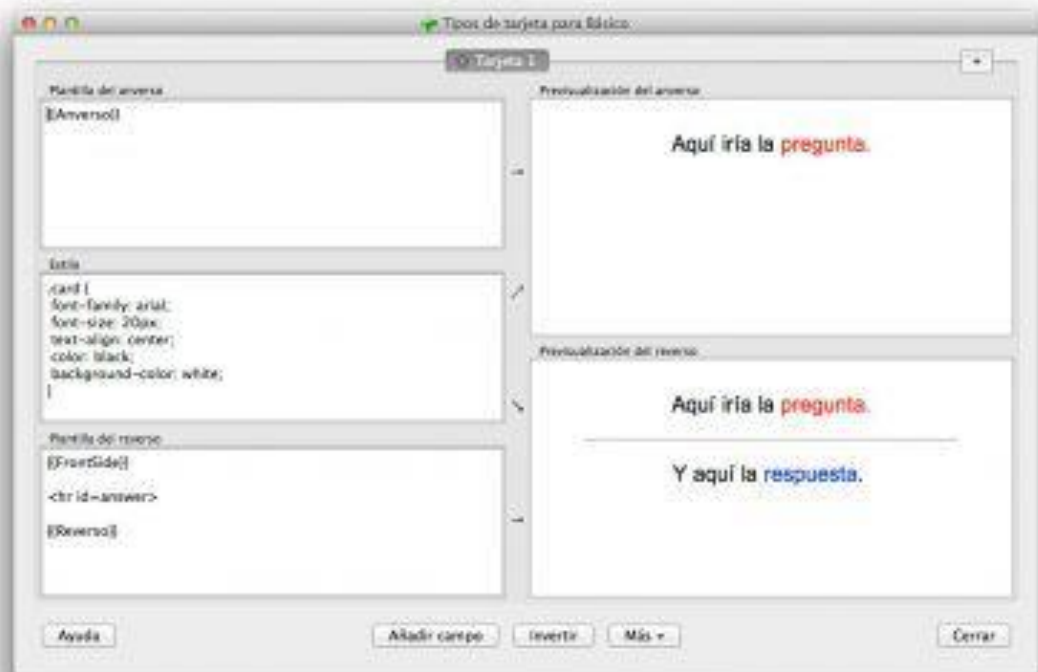
Cualquier persona que necesite recordar cosas en su vida diaria puede beneficiarse de Anki. Dado que es agnóstico en cuanto al contenido y soporta imágenes, audio, vídeos y marcado científico (a través de LaTeX).

Por ejemplo:

- Aprender un idioma
- Estudiar para los exámenes de medicina y derecho
- Memorizar nombres y caras de personas
- Repasar la geografía
- Dominar poemas largos
- Practicar los acordes de la guitarra.

⁷ Anki Web. Aplicativo web para memorizar cosas [en línea] [consultado el 20, septiembre, 2022] Disponible en Internet: <https://apps.ankiweb.net/>

Figura 3: Aplicación Anki.



Fuente: <https://apps.ankiweb.net/>

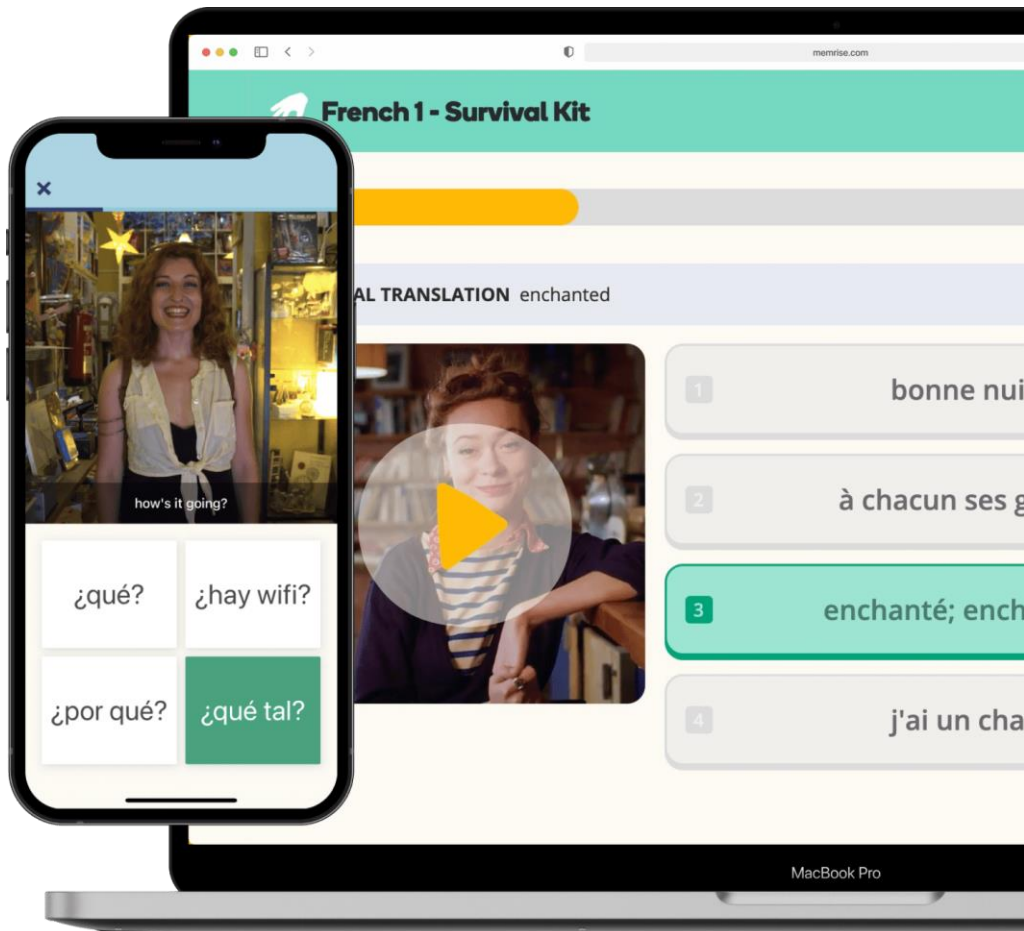
2.2.2 Memrise

Es un Sistema de repetición espaciada enfocado en el aprendizaje de idiomas, esta utiliza aprendizaje inmersivo con videos y audios de nativos reales. Utiliza un algoritmo que se adapta al nivel de cada usuario. El contenido es creado por la aplicación y no por el usuario.

Memrise cuenta con aplicación web y para dispositivos Android y IOS.⁸

⁸ Memrise. Aplicativo web para aprender idiomas [en línea] [consultado el 20, septiembre, 2022] Disponible en Internet: <https://www.memrise.com/>

Figura 4: Aplicación Memrise.



Fuente: <https://www.memrise.com/>

2.2.3 Cuadro comparativo de aplicaciones

Tabla 1: Cuadro comparativo de aplicaciones.

APLICACIÓN	DESCRIPCIÓN	CARACTERÍSTICAS	PRECIO	USUARIO FINAL	CREACIÓN DE CONTENIDO PERSONALIZADO
Anki	Sistema de repetición espaciada que hace fácil el proceso de memorizar conceptos a través de la creación de flashcards personalizadas que acepta múltiples formatos de audio, texto y video.	Sincronización en varios dispositivos. Flexibilidad en la creación de flashcards. Múltiples formatos. Open Source Extensible a través de add-ons	Gratis	Cualquier tipo de usuario que desea memorizar algo.	Permite a través de la creación de flashcards.
Memrise	Aplicación de aprendizaje de idiomas que implementa la técnica de repetición espaciada. Soporta 23 idiomas diferentes. Funciona a través de la enseñanza de frases y vocabulario expuesto en diferentes formatos de pregunta y respuesta en audio, video y texto.	Sincronización en varios dispositivos. Contenido creado y curado por nativos.	Gratis en versión limitada y de Pago para la versión extendida	Usuarios que quieren aprender un idioma soportado por la aplicación.	No permite: El contenido ya está creado
STU-AID	Asistente de memoria que implementa un sistema de repetición espaciada que funciona a través de la creación de flashcards derivadas de las mismas notas que acepta múltiples formatos de audio, texto y video.	Sincronización en varios dispositivos. Flexibilidad en la creación de flashcards en forma de notas. Múltiples formatos. Visualización global del contexto de la pregunta.	Gratis	Estudiantes que toman notas digitales.	Permite

Fuente: Autor

2.3 MARCO ACTUAL

De acuerdo con la investigación realizada se puede ver que existe mucho interés en el tema de técnicas de estudio como la repetición espaciada y el recuerdo activo y el desarrollo de aplicaciones TIC que lo implementen.

2.3.1 Europa

Ejemplo de estas iniciativas son las ya mencionadas aplicaciones en los antecedentes. Memrise fue creada por neurocientíficos y desarrolladores británicos.⁹

Adicionalmente, existen investigaciones relevantes en el tema también, por ejemplo, el algoritmo SuperMemo que fue uno de los primeros algoritmos desarrollados para implementar un sistema de repetición espaciada en una aplicación. Su primera versión fue creada por Piotr Wozniak¹⁰ en Polonia en el año 1985 y su última versión es a hoy en día un algoritmo muy utilizado en aplicaciones como Anki y otras.

Como lo es el caso de la aplicación “I am special, I am in education”¹¹, la cual, fue financiada y apoyada por el gobierno de Turquía. La aplicación está enfocada en la enseñanza de distintos temas a niños superdotados, aplicando distintas técnicas, entre estas, la repetición espaciada.

2.3.2 Norteamérica

En Norteamérica existen numerosas investigaciones sobre los efectos de la practica espaciada. Un artículo llamado “Más allá del tiempo en la tarea: La relación entre el estudio espaciado y la certificación en los MOOCs”¹² realizado en la universidad de Harvard, estudia el impacto que tiene el efecto de espaciar las sesiones de estudio y de repaso en los cursos en línea que tiene la universidad de Harvard en su plataforma virtual llamada HarvardX. El objetivo que se tenía era investigar si existe una relación entre como los estudiantes manejaban el espaciamiento de su práctica y los resultados en rendimiento que obtuvieron.

⁹ Ibíd, Memrise.

¹⁰ PIOTR WOZNIAK - Supermemo [página web]. [Consultado el 3, octubre, 2022]. Disponible en Internet: <https://supermemo.guru/wiki/Piotr_Wozniak>.

¹¹ TAN, Sema. Remote learning through a mobile application in gifted education. En: Gifted Education International [en línea]. 20, diciembre, 2021. vol. 38, no. 1 [consultado el 10, marzo, 2023], p. 95-114. Disponible en Internet: <<https://doi.org/10.1177/02614294211069627>>. ISSN 2047-9077.

¹² MIYAMOTO, Yohsuke R., et al. Beyond Time-on-Task: The Relationship between Spaced Study and Certification in MOOCs. En: SSRN Electronic Journal [en línea]. 2015. [Consultado el 10, octubre, 2022]. Disponible en Internet: <<https://doi.org/10.2139/ssrn.2547799>>. ISSN 1556-5068.

Se demostró que cuando los estudiantes que dedican montos similares de tiempo en múltiples cursos tienden a rendir mejor en sus cursos, en dónde ese tiempo es distribuido entre más sesiones, lo cual, desvela el beneficio de la practica espaciada en el rendimiento de todos los estudiantes independientemente de sus características.

2.3.3 Latinoamérica

Con respecto al tema de aprendizaje adaptativo en aplicaciones de aprendizaje en Latinoamérica existe un artículo llamado Artificial intelligence applications in Latin American higher education: a systematic review¹³. El artículo realiza una revisión sistemática de la literatura sobre el potencial de la aplicación de la Inteligencia Artificial (IA) en el campo de la educación superior en Latinoamérica. Los resultados revelaron que las principales aplicaciones de IA en la educación están en el modelado predictivo, análisis inteligente, tecnología asistida, análisis automático de contenido y análisis de imágenes. Además, se destacó la importancia de comprender el contexto latinoamericano a la hora de estudiar la educación en la región, siendo los principales problemas garantizar el acceso a la educación superior y la retención y graduación de los estudiantes. A pesar de que las aplicaciones de IA están siendo adoptadas gradualmente en diferentes formas por las instituciones de educación superior de muchos países latinoamericanos, como Brasil, Colombia, México, Ecuador, Perú, Chile y Argentina, el ritmo de adopción de la IA en la educación es todavía lenta en comparación con otros campos.

¹³ SALAS-PILCO, Sdenka Zobeida y YANG, Yuqin. Artificial intelligence applications in Latin American higher education: a systematic review. En: International Journal of Educational Technology in Higher Education [en línea]. 18, abril, 2022. vol. 19, no. 1 [consultado el 10, octubre, 2022]. Disponible en Internet: <<https://doi.org/10.1186/s41239-022-00326-w>>. ISSN 2365-9440.

2.3.4 Nivel nacional

En cuanto a aplicaciones de repetición espaciada a nivel nacional existe una aplicación llamada Goconqr¹⁴ la cual implementa la repetición espaciada en un sistema de fichas, notas y mapas mentales. Además, publica contenido de tips de ayuda en los estudios en un blog. La aplicación es web y permite el acceso desde cualquier dispositivo tecnológico.

Referente a investigaciones y trabajos de grado se encuentra “Ambiente e-learning adaptativo en Moodle para estudiantes de grado 11, basado en estilos de aprendizaje” realizado por Javier Orlando Cruz y la Universidad de la Sabana. En dónde se implementó el sistema de práctica espaciada y de selección de contenidos en la plataforma Moodle para los estudiantes de 11 del colegio Simón Bolívar. El desarrollo de este sistema permitía que la selección de contenidos, objetivos de estudio y la experiencia de conocimiento de cada estudiante se analizaran, de tal manera que los contenidos de aprendizaje se adaptaban a cada estudiante. El modelo que se implementó se llama Learning Management System.¹⁵

¹⁴ CREA FICHAS de Memoria con GoConqr [Anónimo]. GoConqr [página web]. [Consultado el 3, octubre, 2022]. Disponible en Internet: <<https://www.goconqr.com/es/fichas/>>.

¹⁵ CRUZ GARZÓN, Javier Orlando. Ambiente e-learning adaptativo en Moodle para estudiantes de grado 11, basado en estilos de aprendizaje. Trabajo de grado. Chía: Universidad de la sabana, 2015. 235 p.

3 METODOLOGÍA

3.1 METODOLOGÍA DE INVESTIGACIÓN

Para el desarrollo del proyecto se utilizará una metodología de investigación cuantitativa, en dónde se hará uso de métricas para medir el cumplimiento de los requerimientos en la implementación a través de las pruebas de integración. La idea es demostrar que se planeó, diseño, implementó y se probó la plataforma de STU-AID de manera exitosa de acuerdo con los requerimientos formulados.

3.2 METODOLOGÍA DE DESARROLLO

Para llevar a cabo el proyecto de software se utilizará una metodología de desarrollo basada en cascada, que contará con las fases del ciclo de vida de software de requerimientos, diseño de arquitectura, implementación y pruebas.

En cada una de las fases se planearon actividades y entregables que son importantes para cumplir con el alcance propuesto.

Tabla 2: Metodología de desarrollo

Objetivo Específico	Metodología de desarrollo
Identificar los requerimientos del aplicativo móvil teniendo en cuenta el modelo de la curva de olvido propuesto por Ebbinghaus y técnicas de estudio como la repetición espaciada y el recuerdo activo para que los usuarios puedan aprender de manera efectiva.	Se realizarán investigaciones semanales haciendo uso de las bases de datos de información y otros recursos que provee la biblioteca de la universidad. Se investigará a nivel nacional y global iniciativas similares e investigaciones sobre el tema de repetición espaciada, curva del olvido y técnicas efectivas de estudio implementadas en aplicaciones TIC.
Diseñar un modelo de arquitectura que se adecue a los requerimientos funcionales y no funcionales de tal manera que sean claros los componentes, relaciones, estructuras de control de flujo de datos y patrones de diseño.	Se realizará un modelo de arquitectura, así como la documentación de las decisiones de arquitectura y el modelo de datos y de clases de tal manera que se tenga en consideración la responsabilidad y comportamiento de cada módulo dentro del sistema STU-AID.

Implementar la solución en código siguiendo el modelo de arquitectura y cumpliendo con todos los requerimientos levantados.	APLICACIÓN MÓVIL. Se realizará una aplicación móvil que le permitirá al usuario interactuar y visualizar el contenido de la aplicación. Se utilizarán las tecnologías Javascript y la librería React Native. API RESTFUL Se realizará un API RESTful que definirá los recursos y la forma de acceder a estos, de tal manera que se implementará la lógica del sistema de repetición espaciada y se interactuará con la base de datos para la persistencia de datos. BASE DE DATOS. Se creará una base de datos que almacenará la información importante de cada usuario y de su sistema de repetición espaciada. Se planea utilizar PostgreSQL como motor de base de datos.
Validar el producto de software usando pruebas de integración al producto para garantizar el correcto funcionamiento de cada requerimiento que se levantó y así asegurar la calidad.	Se implementarán pruebas de integración para validar el correcto funcionamiento. Se planea utilizar para la fase de pruebas un ambiente dedicado en dónde se realicen, seguido de las pruebas, es necesario planificar y realizar los ajustes necesarios a medida que se encuentren errores.

Fuente: Autor

4 ESPECIFICACIÓN DE REQUERIMIENTOS

En este capítulo se explicarán las actividades que se realizaron para llevar a cabo el primer objetivo del proyecto, el cual es:

Identificar los requerimientos de la aplicación móvil teniendo en cuenta el modelo de la curva de olvido propuesto por Ebbinghaus y técnicas de estudio como la repetición espaciada y el recuerdo activo para que los usuarios puedan aprender de manera efectiva.

Para cumplir este objetivo, se realizaron cuatro actividades relacionadas, las cuales se explicarán a detalle a continuación.

4.1 INVESTIGACIÓN DE LAS TÉCNICAS DE ESTUDIO DE REPETICIÓN ESPACIADA, RECUPERACIÓN ACTIVA Y EL MÉTODO DE FLASHCARDS DE LEITNER

Se investigaron las técnicas de estudio de repetición espaciada, recuperación activa y el método de flashcards de Leitner y se conformó un marco teórico con la evidencia de la investigación.

De cada técnica se definió su propósito y funcionamiento de manera detallada, utilizando referencias de estudios y aplicaciones en otras aplicaciones móviles y web relacionadas al estudio y la memoria.

4.2 REQUERIMIENTOS NO FUNCIONALES

Los atributos de calidad se definen como propiedades del servicio que es entregado por el sistema a sus usuarios. Este servicio entregado por el sistema es un comportamiento percibido por sus usuarios. El servicio es iniciado por un evento, el cual indica la necesidad de recibir el servicio.¹⁶

Algunos de los atributos de calidad son:

- Desempeño.
- Fiabilidad.
- Seguridad.
- Usabilidad.
- Procesamiento en tiempo real.

Se realizó una sesión en la cual se concretaron las preocupaciones alrededor de la aplicación y se priorizaron para que estos atributos de calidad se vean reflejados en el sistema.

Se llegó a que la prioridad será la usabilidad y la seguridad, seguido del procesamiento en tiempo real y el desempeño.

¹⁶ BABACCI, Mario, et al. Quality Attributes. En: Carnegie Mellon University Software Engineering Institute [en línea]. 1995. [Consultado el 24, marzo, 2023]. Disponible en Internet: <https://resources.sei.cmu.edu/asset_files/TechnicalReport/1995_005_001_16427.pdf>. p.4

Tabla 3: Atributos de calidad priorizados.

Atributo de calidad	Descripción	Prioridad
Usabilidad	El usuario puede crear, editar y eliminar preguntas tipo flashcards de estudio de manera fácil utilizando atajos.	Alta
	El usuario puede ver y conocer de manera clara sus clases, temas, y preguntas con detalles de manera clara en la interfaz.	
Seguridad	El usuario puede acceder al sistema de forma segura.	Alta
	El acceso a la información del usuario está restringido para todos excepto el mismo usuario.	
Procesamiento en tiempo real	Las acciones y la información del usuario se almacena y se sincroniza a lo largo de todos los dispositivos.	Media
Desempeño	El usuario experimenta tiempos de respuestas rápidos por parte del sistema.	Media

Fuente: Autor

4.2.1 Usabilidad

La usabilidad se define en la norma ISO 9241 como el grado en que un producto puede ser utilizado por usuarios específicos para alcanzar objetivos específicos con eficacia, eficiencia y satisfacción en un contexto de uso específico.¹⁷

¹⁷ ISO. Human-centred design for interactive systems. ISO 9241. [s.l.]: [s.n.], 2010.

4.2.2 Seguridad

La seguridad es una medida de la capacidad del sistema para resistir el uso no autorizado sin dejar de prestar sus servicios a los usuarios legítimos.

La seguridad puede caracterizarse como un sistema que proporciona no repudio, confidencialidad, integridad, garantía, disponibilidad y auditoría.

- El no repudio es la propiedad de que una transacción (acceso o modificación de datos o servicios) no puede ser negada por ninguna de las partes que intervienen en ella.
- La confidencialidad es la propiedad por la que los datos o servicios están protegidos de accesos no autorizados.
- Integridad es la propiedad de que los datos o servicios se entregan según lo previsto.
- La garantía es la propiedad de que las partes de una transacción son quienes dicen ser.
- La disponibilidad es la propiedad de que el sistema estará disponible para su uso legítimo.
- Auditoría es la propiedad de que el sistema rastrea las actividades dentro del mismo a niveles suficientes para reconstruirlas.¹⁸

¹⁸ UNIVERSITY OF NEW BRUNSWICK. Understanding Quality Attributes. Faculty of Computer Science | UNB [página web]. [Consultado el 25, marzo, 2023]. Disponible en Internet: <<https://www.cs.unb.ca/~wdu/cs6075w10/sa2.htm>>.

4.2.3 Desempeño

Según Smith, el desempeño se refiere a la capacidad de respuesta: el tiempo necesario para responder a eventos específicos o el número de eventos procesados en un intervalo de tiempo determinado. El desempeño es el atributo de un sistema informático que caracteriza la puntualidad del servicio prestado por el sistema.¹⁹

4.3 IDENTIFICACIÓN DEL PERFIL DE USUARIO

4.3.1 Perfil de usuario segmentado por el buyer persona

Para identificar el perfil del usuario que utilizaría la aplicación se utilizó el método estratégico que se llama Buyer Persona, el cual busca identificar el cliente potencial de un producto, a través, de la investigación del tipo de persona que más le sacaría provecho a las funcionalidades. Se encontró que el perfil de comprador tiene estas características en general:

¹⁹ SMITH, Connie. Performance Engineering of Software Systems (The Sei Series in Software Engineering). En: Addison-Wesley. 1990. p. 90.

Figura 5: Buyer Persona



Juan
Estudiante

Antecedentes

- Quiere mejorar en sus exámenes.
- Quiere estudiar de manera efectiva.

Datos demográficos.

- Joven de 16 a 25 años.
- Estudiante de colegio o de universidad.

Herramientas que utiliza el usuario para el aprendizaje.

- Celular.
- Computador.
- Tablets.
- Cuadernos.
- Libros.

Redes sociales que utiliza.

- Instagram, TikTok, Youtube, LinkedIn, Pinterest, Facebook.

Fuente: Autor

4.4 ESPECIFICACIÓN DE REQUERIMIENTOS

Una vez realizadas las investigaciones de las técnicas de repetición espaciada, recuperación activa y el método de flashcards de Leitner. Además de especificar los atributos de calidad que se esperan del producto, se realizó el listado de requerimientos funcionales que sirvieron de insumo para la especificación de historias de usuario.

A continuación, se muestra la lista de requerimientos funcionales:

- Registrar cuenta.
- Iniciar sesión.
- Ver mis clases.
- Agregar una clase.
- Eliminar una clase.
- Editar una clase.
- Ver temas de estudio de una clase.
- Crear un tema de estudio.
- Eliminar un tema de estudio.
- Editar un tema de estudio.
- Ver preguntas de un tema de estudio.
- Ver opciones de crear preguntas.
- Crear pregunta elaborada/ construida.
- Crear pregunta dicotómica Verdadero/ Falso.
- Crear pregunta definición concepto (Frente/reverso).
- Eliminar pregunta.
- Editar pregunta.
- Visualizar preguntas en flashcard.

- Empezar sesión de estudio del día.
- Empezar sesión de un tema.
- Empezar sesión de estudio de una materia.
- Empezar sesión por tiempo.
- Responder pregunta.
- Mostrar progreso de sesión.

4.4.1 Historia de usuario

Según Dean Leffingwell²⁰. Una historia de usuario es una sentencia breve que describe algo que el sistema necesita hacer para el usuario. Es una manera sustituta de especificar requerimientos, diferente a la manera tradicional de especificación de casos de uso. Esta es importante porque busca contener el valor que se busca tener para el usuario, lo cual, es sumamente importante en el desarrollo rápido.

Una historia de usuario también define la manera en la que se entrega un valor incremental, en donde:

- Se define una historia de usuario de valor.
- Se implementa y se prueba en una corta iteración.
- Se muestra al usuario.
- Se recibe una retroalimentación.
- Se aprende y se captura la retroalimentación.
- Se repite el ciclo.

²⁰ LEFFINGWELL, Dean y BEHRENS, Pete. A User Story Primer. En: Leffingwell, LLC [en línea]. 2009. p. 1-16. [Consultado el 10, marzo, 2023]. Disponible en Internet: <<https://www.scaledagileframework.com/wp-content/uploads/2021/07/User-Story-Primer.pdf>>.

Leffingwell²¹ también, establece que hay una voz en las historias de usuario en las que se sugiere seguir un tipo de plantilla en la que se define el tipo de usuario, lo que desea o espera del sistema y una justificación del por qué lograr lo que se espera genera valor para él.

4.4.2 Especificación de historias de usuario

Una vez se formularon los requerimientos, se pasó a realizar un tablero de historias de usuario utilizando la aplicación de gestión de proyectos llamada Trello.²²

Se creo un tablero en el cual se crearon las secciones de backlog, diseño (design), para hacer (to-do), en progreso (doing), en pruebas (testing) y terminado (done), y se pusieron tarjetas que representan las historias de usuario.

En la sección de backlog se ponen la lista de requerimientos iniciales que serán los nombres de las historias de usuario.

Después, para indicar que las historias de usuario son especificadas en términos de criterios de aceptación y requerimientos técnicos se ponen en la sección de diseño (design).

Seguido de esto, para indicar que la historia de usuario esta lista para su desarrollo se pasa a la sección de para hacer (to-do).

En la sección en progreso (doing) se ponen las historias que actualmente se están desarrollando, una vez terminado el desarrollo de estas, pasan a la sección de pruebas (testing).

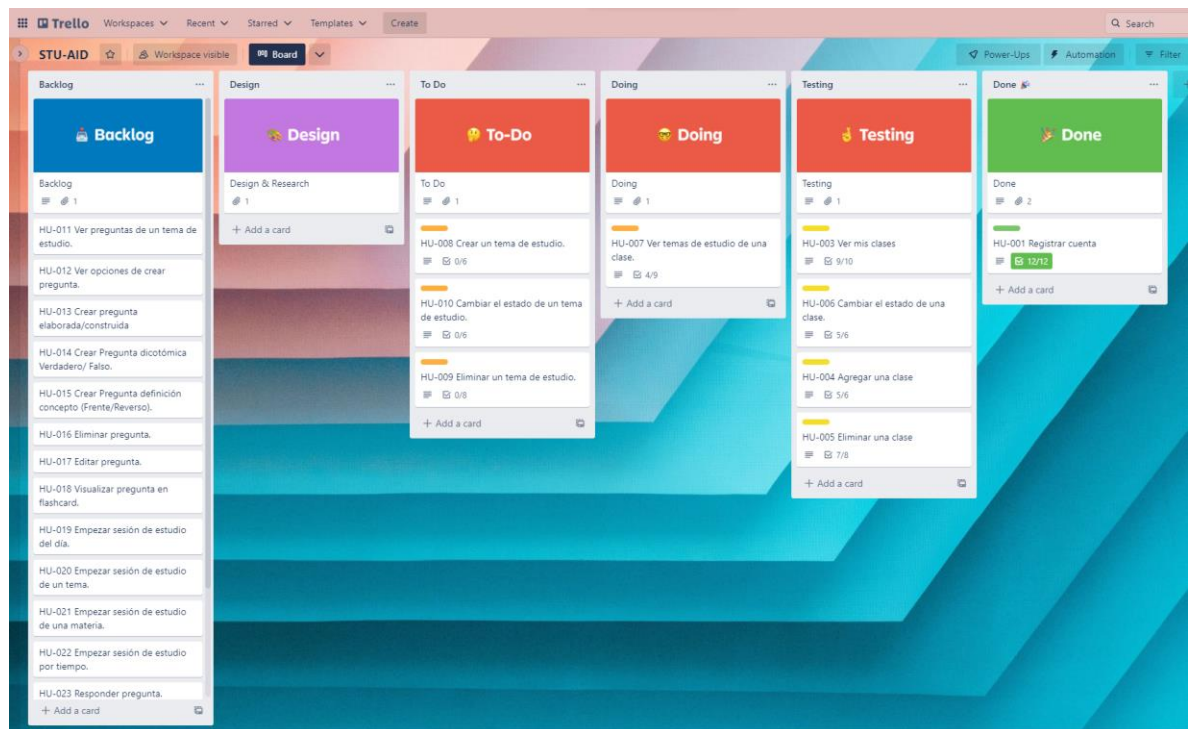
²¹ Ibíd, p. 3.

²² TRELLO | Atlassian. Atlassian [página web]. [Consultado el 10, marzo, 2023]. Disponible en Internet: <<https://www.atlassian.com/software/trello>>.

En la sección de pruebas (testing) se realizan pruebas para verificar que el producto cumple con los criterios de aceptación, con los requerimientos técnicos y con el valor esperado para el usuario que se especificó en la historia de usuario. Una vez terminado el proceso de pruebas, se pasa a la sección de terminado (done).

En la sección de terminado (done) se indica que el producto termino con los procesos anteriores, lo que significa que cuenta con la funcionalidad esperada.

Figura 6: Tablero backlog de historias de usuario.



Fuente: Autor.

Tabla 4: Historias de usuario.

ID	Título	Descripción
HU-001	Registrar cuenta.	Como usuario principal lo que quiero es poder registrar una cuenta para poder empezar a utilizar un servicio con una cuenta propia.
HU-002	Iniciar sesión.	Como usuario de la aplicación lo que quiero es poder iniciar sesión con mi cuenta para poder acceder a las funcionalidades de la aplicación de manera segura y personalizada.
HU-003	Ver mis clases.	Como usuario principal lo que quiero es poder ver la lista de mis clases y sus estados para poder organizar mis temas por clase.
HU-004	Agregar una clase.	Como usuario lo que quiero es poder agregar una clase en mi aplicación para así poder designarle temas de estudio, asignarle un nombre, una descripción y una prioridad.
HU-005	Eliminar una clase.	Como usuario me gustaría ser capaz de borrar una clase para así limpiar mi aplicación con clases que ya no son de mi interés repasar.
HU-006	Cambiar el estado de una clase.	Como usuario quiero cambiar el estado de mi clase para poder ver u ocultar las clases activas de las desactivas además de cambiar la descripción, la prioridad y el nombre de la clase.
HU-007	Ver temas de estudio de una clase.	Como usuario principal lo que quiero es ver los temas de cada curso para así dividir mejor los contenidos de un curso.
HU-008	Crear un tema de estudio.	Como usuario lo que quiero es poder crear temas de estudio que pertenezcan a un curso que este realizando para poder organizar mis flashcards en temas y clases.

HU-009	Eliminar un tema de estudio.	Como usuario principal lo que quiero es poder eliminar temas de estudio que ya no quiero repasar.
HU-010	Editar el estado de un tema de estudio.	Como usuario lo que quiero es poder cambiar el estado de un tema de estudio y poder cambiar su nombre para así tener mayor control sobre los temas que me interesan.
HU-011	Ver preguntas de un tema de estudio.	Como usuario lo que quiero es poder ver las preguntas de un tema de estudio en particular para tener conocimiento de cómo se compone el tema a partir de las preguntas.
HU-012	Ver opciones de crear pregunta.	Como usuario lo que quiero es poder crear una pregunta de tipo verdadero / falso, elaborada y de frente/reverso para poder fragmentar los temas que estoy estudiando.
HU-013	Crear pregunta elaborada/construida.	Como usuario lo que quiero es poder crear preguntas elaboradas / construidas para mis sesiones de repaso.
HU-014	Crear Pregunta dicotómica Verdadero/ Falso.	Como usuario principal de la aplicación lo que quiero es poder crear preguntas de verdadero o falso para mis sesiones de repaso.
HU-015	Crear Pregunta definición concepto (Frente / Reverso).	Como usuario de la aplicación lo que quiero es poder crear preguntas de tipo frente / reverso para mis sesiones de repaso.
HU-016	Eliminar pregunta.	Como usuario principal de la aplicación lo que quiero es poder eliminar una pregunta que ya no quiero que este en las sesiones de repaso.
HU-017	Editar pregunta.	Como usuario lo que quiero es poder editar una pregunta a mi gusto.
HU-018	Visualizar pregunta en flashcard.	Como usuario lo que quiero es poder visualizar las preguntas en forma de flashcards para poder estudiarlas en las sesiones de repaso.

HU-019	Empezar sesión de estudio del día.	Como usuario lo que quiero es poder estudiar las tarjetas que probablemente haya olvidado en cierto porcentaje.
HU-020	Empezar sesión de estudio de un tema.	Como usuario lo que quiero es repasar las preguntas de un tema en específico.
HU-021	Empezar sesión de estudio de una materia.	Como usuario lo que quiero es repasar las tarjetas de un curso en específico sin importar si no necesitan ser repasadas.
HU-022	Empezar sesión de estudio por tiempo.	Como usuario lo que quiero es repasar las flashcards en una sesión de tiempo.
HU-023	Responder pregunta.	Como usuario quiero responder las preguntas en mis sesiones de estudio, de tal forma que pueda poner a prueba mi conocimiento y calificar la dificultad de la pregunta de 1 a 5, donde 1 fue muy difícil y 5 muy fácil.
HU-024	Conocer el progreso de la sesión de estudio.	Cómo usuario quiero conocer el progreso de mi sesión de estudio para saber cuántas flashcards he repasado y cuantas hacen falta.

Fuente: Autor

5 DISEÑO DE ARQUITECTURA

En este capítulo se detallarán las actividades y aspectos importantes que se tomaron en cuenta para cumplir con la fase de diseño de la aplicación, la cual corresponde al segundo objetivo propuesto:

Diseñar un modelo de arquitectura que se adecue a los requerimientos funcionales y no funcionales de tal manera que sean claros los componentes, relaciones, estructuras de control de flujo de datos y patrones de diseño.

Para esta fase es importante tener una comprensión clara del problema, y es por esto por lo que la especificación de requerimientos funcionales y no funcionales fue vital, ya que en este proceso se necesita entender muy bien el problema para que así se busque una solución acorde al problema que se planteó. Esto se realiza a través del modelado de decisiones de arquitectura y diagramas que den a conocer detalles de la aplicación, su estructura, el flujo de funcionamiento y sus patrones de diseño.

5.1 ESTRATEGIAS DE ARQUITECTURA

Para dar cumplimiento a los requerimientos no funcionales, se idearon estrategias que buscan cumplir con cada uno de los atributos de calidad. A continuación, se mostrarán.

Tabla 5: Estrategias de arquitectura por atributo de calidad

Atributo de calidad	Descripción	Estrategia
Usabilidad	El usuario puede crear, editar y eliminar preguntas tipo flashcards de estudio de manera fácil utilizando atajos.	Implementación de un diseño que mejore la experiencia de usuario utilizando animaciones, botones y texto visibles.
	El usuario puede ver y conocer de manera clara sus clases, temas, y preguntas con detalles de manera clara en la interfaz.	
Seguridad	El usuario puede acceder al sistema de forma segura.	Implementación de JSON Web Tokens para la autenticación y autorización.
	El acceso a la información del usuario está restringido para todos excepto el mismo usuario.	Implementación de estrategias de encriptación de password.
Procesamiento en tiempo real	Las acciones y la información del usuario se almacena y se sincroniza a lo largo de todos los dispositivos.	Implementación de middlewares en la API que se encarguen de la persistencia de los datos.
Desempeño	El usuario experimenta tiempos de respuestas rápidos por parte del sistema.	Despliegue de base de datos en un contenedor de Docker.

Fuente: Autor

5.2 DECISIONES DE ARQUITECTURA

Para el desarrollo de la solución se utilizó un estilo de arquitectura de cliente-servidor, el cual es un estilo basado en llamado y retorno. Según Sulyman²³, una un modelo cliente servidor es una arquitectura compuesta por un cliente y un servidor, en donde los clientes siempre envían peticiones, mientras que el servidor responde la petición enviada. Este provee una comunicación de interproceso porque involucra un intercambio de datos de ambos, tanto del cliente, como del servidor, en donde se desarrollan diferentes funciones.

²³ HAROON-SULYMAN, Shakirat. Client-Server Model. En: ResearchGate [en línea]. 2014. p. 67-69. [Consultado el 13, marzo, 2023]. Disponible en Internet: <<https://doi.org/10.9790/0661-16195771>>.

Siguiendo con la explicación de Sulyman, el modelo cliente servidor ofrece los siguientes beneficios²⁴:

- Separa el procesamiento de aplicación a lo largo de múltiples máquinas.
- Permite de manera más fácil compartir recursos desde los clientes.
- Reduce la replicación de los datos al almacenarlos en un servidor común a los clientes, en vez de almacenarlos en cada cliente.

En el proyecto Stu-aid el servidor será una aplicación API REST utilizando el ambiente de javascript llamado NodeJS y la librería Express JS, la cual recibirá las peticiones HTTP de la aplicación cliente que es a su vez una aplicación móvil desarrollada en React Native.

A continuación, se argumentará el uso de estas tecnologías, los estilos de arquitectura y patrones que utilizaremos para el diseño de la solución.

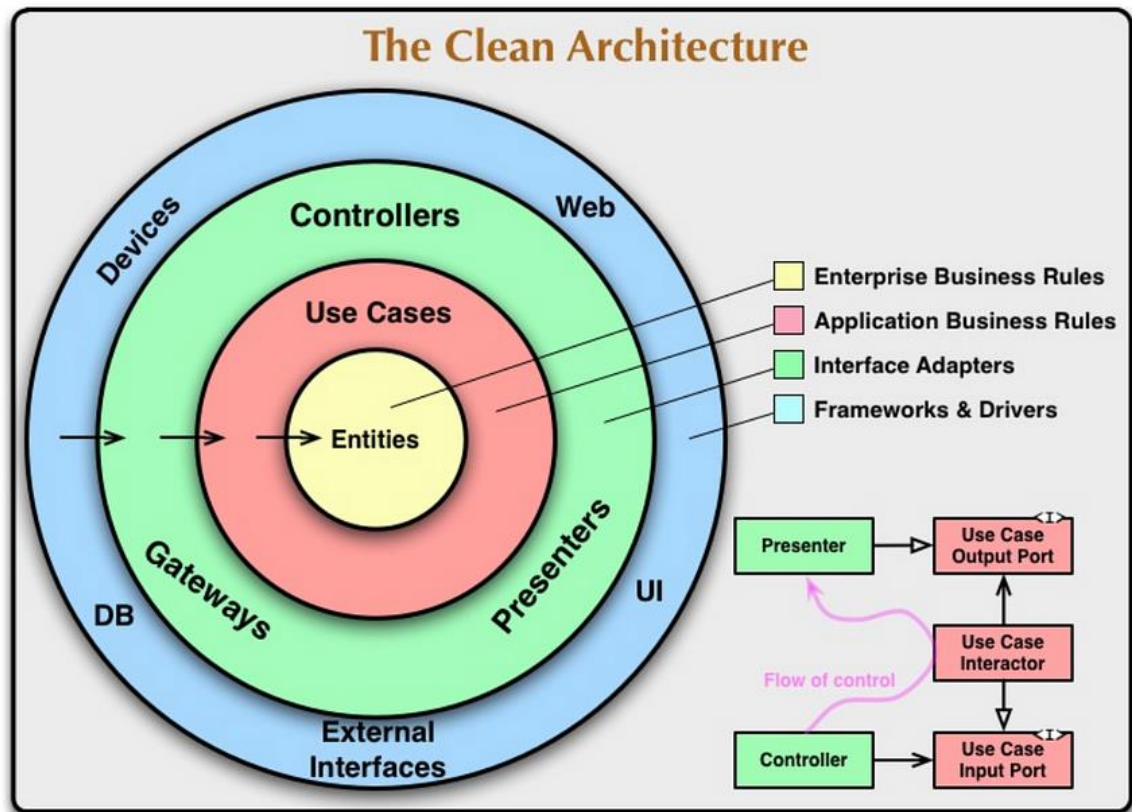
5.2.1 La Arquitectura limpia (The clean architecture)

La Arquitectura limpia es una guía propuesta por Rober Martin en dónde propone un conjunto de directrices para la construcción de software para que sea escalable, modular, independiente, fácil de probar, y mantenible a lo largo del tiempo. Su objetivo, según Martin es minimizar los recursos humanos para construir y mantener un sistema requerido.²⁵

²⁴ Ibíd, p. 68

²⁵ MARTIN, Robert. Clean Architecture: A Craftsman's Guide to Software Structure and Design. [s.l.]: Pearson Education, Limited, 2017. ISBN 9780134494326.

Figura 7: La arquitectura limpia.



Fuente: The Clean Architecture Robert Martin

Esta arquitectura se compone de cuatro capas que representan distintas áreas del software, las dependencias entre capas apuntan hacia adentro del círculo y solamente en esa dirección. Esto significa que una capa no puede saber nada de sus capas externas, pero sí de sus capas internas. Esto es conocido como la regla de la dependencia, la cual permite una organización estratégica que permite una alta cohesión y un bajo acoplamiento entre funciones, clases y variables.²⁶

²⁶ Ibíd, p. 197

Las entidades encapsulan reglas críticas del negocio, pueden ser los objetos de negocio de la aplicación y encapsulan en ellas conceptos y reglas que le importan al negocio y que son poco probables que cambien.²⁷

La capa de casos de uso, contienen reglas específicas de la aplicación. En esta se implementan los casos de uso del sistema, se orquesta el flujo de datos y manipulación de entidades para cumplir con los objetivos de una funcionalidad definida.²⁸

La capa de adaptadores de interfaz es un conjunto de adaptadores que convierten datos al formato más conveniente para la capa de casos de uso y entidades, estos son intermediarios en el flujo de los datos desde la capa de presentación o vistas a la capa de casos de uso, y después entidades, como al revés, es decir, desde las capas de entidades y casos de uso a la presentación.²⁹

La capa de interfaces externas es donde van todos los detalles externos a la aplicación que funcionan en la última, ya que son interfaces externas con las que se tiene que comunicar la aplicación.³⁰

5.2.2 Decisiones de la aplicación backend

Como se mencionaba anteriormente la aplicación de backend será un api REST, construida en Node.js con ayuda de Express JS, se utilizará como base de datos Postgresql, como ORM se utilizará Sequelize, como librería de autenticación y autorización se utilizará JWT, como orquestador de contenedores Docker.

²⁷ Ibíd, p. 197

²⁸ Ibíd, p. 198

²⁹ Ibíd, p. 198

³⁰ Ibíd, p. 199

5.2.2.1 Express middlewares y la arquitectura pipes and filters

ExpressJS es un framework flexible de Node.js para la construcción de APIs de manera rápida y fácil que funciona a través del encadenamiento de llamados a funciones llamadas *middleware functions*.³¹

Las funciones middleware pueden realizar las siguientes tareas:

- Ejecutar código.
- Hacer cambios a los objetos de petición y de respuesta.
- Terminar el ciclo de petición-respuesta (request-response cycle)
- Llamar al siguiente middleware en el stack.

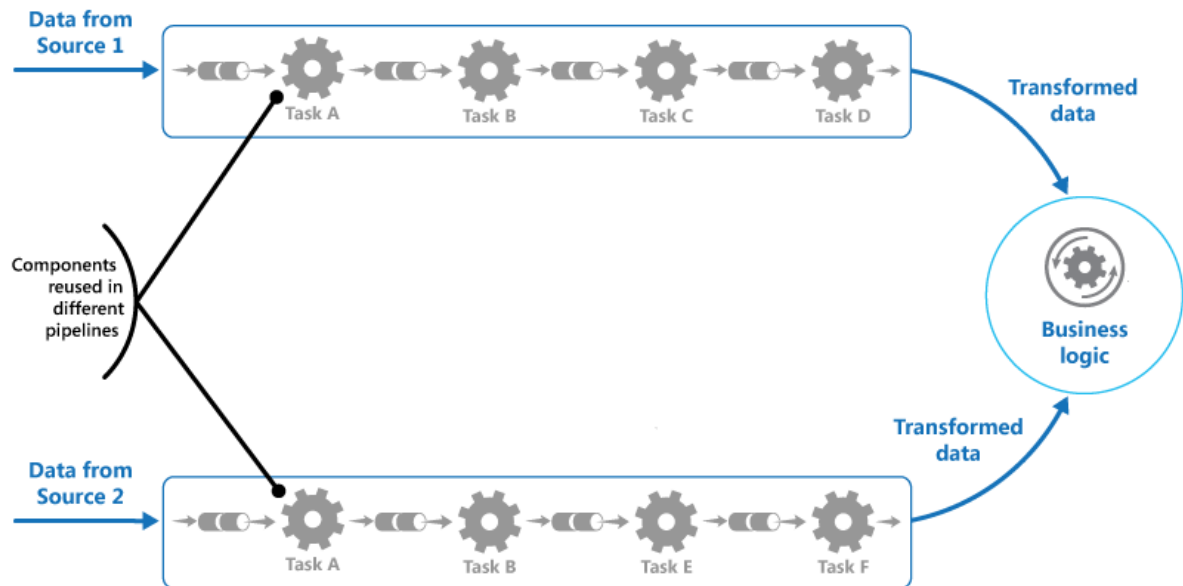
Para aclarar, los middlewares permiten encapsular una funcionalidad que describe un comportamiento, como si fuera un bloque de construcción. Esto permite que se modularice la funcionalidad en estos bloques y que sean reutilizados de manera fácil. En consecuencia, la aplicación constara de distintos procesos y estos procesos se componen de middlewares encadenados entre sí.

Algo similar al patrón de implementación de “pipes and filters”, la cual descompone el procesamiento necesario para cada flujo o proceso en un conjunto de componentes separados (o filtros), cada uno de los cuales realiza una única tarea. Para lograr un formato estándar de los datos que cada componente recibe y envía, los filtros pueden combinarse en la canalización. De este modo se evita la duplicación de código y se facilita la eliminación o sustitución de componentes, o la integración de componentes adicionales, si cambian los requisitos de

³¹ USING EXPRESS middleware [Anónimo]. Express - Node.js web application framework [página web]. [Consultado el 13, marzo, 2023]. Disponible en Internet: <<https://expressjs.com/en/guide/using-middleware.html>>.

procesamiento. Este diagrama muestra una solución implementada con tuberías y filtros.³²

Figura 8: Comportamiento de Pipes and filters



Fuente: https://learn.microsoft.com/en-us/azure/architecture/patterns/_images/pipes-and-filters-solution.png

5.2.2.2 Postgresql

Postgresql es un sistema de base de datos de objetos relacionales de código abierto con alrededor de 35 años de desarrollos activo, con una reputación fuerte en el mercado, con variedad de características, y de buen rendimiento.

Las ventajas que ofrece Postgresql son:

- Es un software de código abierto.

³² PIPES AND Filters pattern - Azure Architecture Center [Anónimo]. Microsoft Learn: Build skills that open doors in your career [página web]. [Consultado el 13, marzo, 2023]. Disponible en Internet: <<https://learn.microsoft.com/en-us/azure/architecture/patterns/pipes-and-filters>>.

- Soporta una gran parte del estándar SQL y ofrece características como consultas complejas, llaves foráneas, disparadores, vistas actualizables, integridad transaccional y control de concurrencia con múltiples versiones.
- Puede ser usado, modificado y distribuido por cualquiera de manera libre, sin importar si es privado, comercial o académico.³³

Se escogió este motor de base de datos sobre otros por su rendimiento porque es de código abierto, es compatible con el ORM Sequelize. También, porque permite persistir e indexar objetos JSON lo cual, es una ventaja sobre otros motores, ya que en una parte del modelo de entidades existen datos no estructurados que se necesitaban almacenar, sin embargo, la otra parte de los datos presentaba una estructura claramente relacional. Sin esta característica de postgres, hubiese sido necesario utilizar una base de datos no relacional aparte de la relacional, lo cual, hubiese aumentado la complejidad de las entidades y de los casos de uso.

5.2.2.3 Sequelize ORM

Según su documentación, Sequelize es una herramienta ORM (Object-relational mapping) para Node.js que es compatible con Postgres, MySQL, MariaDB, SQLite, Oracle Database, Amazon Redshift y Snowflake's Data Cloud.³⁴

Las ventajas que ofrece este ORM son:

- Soporte solido de transacciones.
- Relaciones.
- Eager y Easy loading.
- Replicación.
- Migraciones.

³³ WHAT IS PostgreSQL? [Anónimo]. PostgreSQL Documentation [página web]. [Consultado el 13, marzo, 2023]. Disponible en Internet: <<https://www.postgresql.org/docs/current/intro-what-is.html>>.

³⁴ SEQUELIZE V6 | Sequelize [Anónimo]. Sequelize | Feature-rich ORM for modern TypeScript & JavaScript [página web]. [Consultado el 13, marzo, 2023]. Disponible en Internet: <<https://sequelize.org/docs/v6/>>.

- Manejo de versiones,
- Manejo de seeders.
- Creación de modelos.
- Validaciones y restricciones.

Se escogió esta tecnología para cumplir con los atributos de calidad de escalabilidad y procesamiento en tiempo real. Además, porque facilita la manipulación de tablas en la capa de entidades que conforma el modelo establecido en la arquitectura limpia.

5.2.2.4 JSON Web Tokens JWT y Passport.js

Según su documentación, JWT es un estándar RFC 7519 que establece un método para representar la identificación de una reclamación entre dos partes.

Las reclamaciones de un JWT se codifican como un objeto JSON que se utiliza como carga útil de una estructura JSON Web Signature (JWS) o como texto sin formato de una estructura JSON Web Encryption (JWE), lo que permite firmar digitalmente las reclamaciones o proteger su integridad con un código de autenticación de mensajes.³⁵

Los JSON Web Tokens tienen las siguientes características:

- Son token “stateless” información encriptada de cómo se puede identificar un usuario y que permisos tiene.
- Permite el manejo de sesiones en distintas maquinas distribuidas. Antes se debía tener un manejo complicado de sesiones entre máquinas.
- Permite el manejo de sesiones en dispositivos diferentes a los clientes web y las cookies. Antes la sesión se manejaba con cookies y existen clientes

³⁵ RFC. RFC 7519: JSON Web Token (JWT). RFC Editor [página web]. [Consultado el 13, marzo, 2023]. Disponible en Internet: <<https://www.rfc-editor.org/rfc/rfc7519>>.

diferentes a los navegadores web como lo es el caso de las aplicaciones móviles que no cuentan con el manejo de cookies.

Se escogió esta tecnología para cumplir con los atributos de calidad de seguridad, ya que permite la creación de estrategias de autenticación y autorización para los usuarios de la aplicación móvil de Stu-aid.

Para poder utilizar los JWT en la aplicación de backend se utilizó la librería Passport.js la cual permite la creación, verificación y manejo de JSON Web Tokens de manera fácil.

5.2.2.5 Docker

Según su documentación³⁶, Docker es una tecnología abierta que permite separar las aplicaciones de la infraestructura y empaquetar y ejecutar aplicaciones en contenedores. Docker agiliza el ciclo de vida del desarrollo, permite la entrega rápida de aplicaciones y facilita su despliegue y escalado. Docker utiliza una arquitectura cliente-servidor, en la que el cliente Docker interactúa con el Docker Daemon.

Las ventajas que ofrece esta tecnología son:

- Permite la optimización de recursos entre contenedores.
- Facilita el despliegue de aplicaciones en contenedores que corren en ambientes independientes, pero sobre el mismo hardware.
- Facilita la comunicación entre contenedores.
- Facilitan la gestión dinámica de las cargas de trabajo, ampliando o reduciendo aplicaciones y servicios según las necesidades de la empresa, prácticamente en tiempo real.

³⁶ DOCKER DOCS. Docker overview. Docker Documentation [página web]. [Consultado el 13, marzo, 2023]. Disponible en Internet: <<https://docs.docker.com/get-started/overview/>>.

Se escogió esta tecnología para cumplir con los atributos de calidad de disponibilidad y desempeño al momento del despliegue.

5.2.3 Decisiones de la aplicación frontend

Como se mencionaba anteriormente la aplicación de frontend será una aplicación móvil construida con la librería React Native con la ayuda del framework Expo, se utilizará como librería de routing React Navigation.

5.2.3.1 Aplicación móvil construida con Expo y React Native

5.2.3.2 Expo

Según la documentación oficial³⁷, Expo es un framework de código abierto para aplicaciones que se ejecutan de forma nativa en Android, iOS y la web. Expo reúne características para aplicaciones móviles y web, facilita el desarrollo de muchas características importantes para la construcción y el escalado de una aplicación como actualizaciones en vivo, compartir instantáneamente su aplicación, y soporte web para los proyectos de React Native.

5.2.3.3 React Native

React Native es una librería basada en React y javascript que permite desarrollar aplicaciones web y móviles IOS y Android nativas utilizando una sola base de código. Fue desarrollada por Facebook en el 2015 y desde entonces sigue siendo actualizada y mejorada con nuevas características.³⁸

³⁷ EXPO. What is Expo? Expo Documentation [página web]. [Consultado el 14, marzo, 2023]. Disponible en Internet: <<https://docs.expo.dev/introduction/expo/>>.

³⁸REACT NATIVE DOCS. Documentation. [página web]. [Consultado el 14, marzo, 2023]. Disponible en Internet: < <https://reactnative.dev/> >.

5.3 MODELO DE DATOS

5.3.1 Usuarios

Esta entidad representa a los usuarios de la aplicación. Solo existe un rol de usuario que es el de estudiante. Es cuenta con identificador, nombre completo, correo electrónico, una contraseña y una fecha de creación.

Un usuario puede tener varios cursos, pero un curso solo puede tener un usuario.

5.3.2 Cursos

Esta entidad representa los cursos que puede tener un usuario. Un usuario puede tener varios cursos y cuenta con un identificador, un nombre, un estado y una descripción. Además, un curso puede tener varios temas, pero un tema solo puede corresponder a un tema.

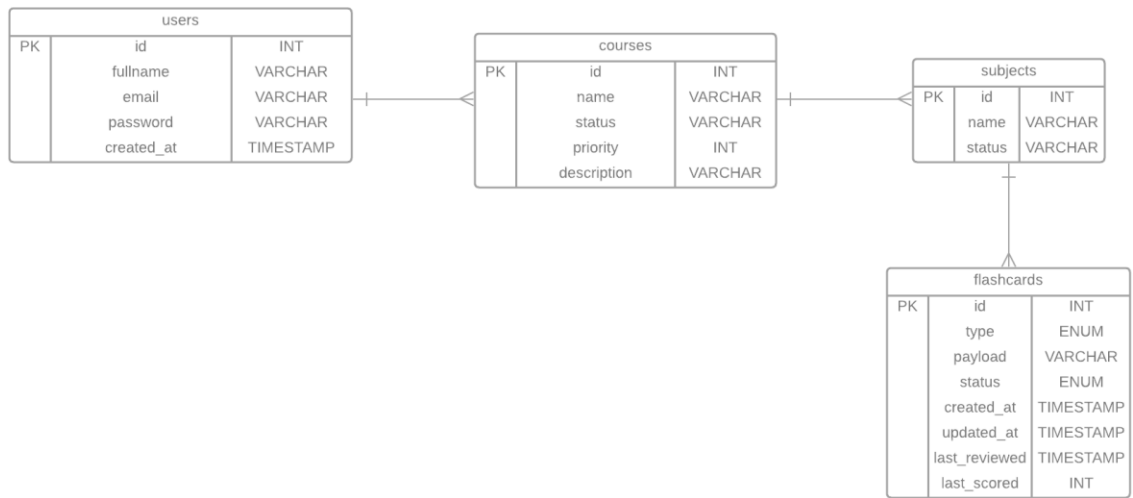
5.3.3 Temas

Esta entidad representa los temas en los que se descompone un curso. Un curso puede tener varios temas y estos cuentan con un identificador, un nombre y un estado. Además de esto, puede tener una o varias flashcards.

5.3.4 Flashcards

Esta entidad representa la unidad básica de estudio, una pieza lógica, la cual es la que se necesita ser repasada en conjunto con otras para poder entender un tema. Una flashcard tiene un identificador, un tipo que puede ser una pregunta verdadero o falso, una pregunta elaborada-construida o una pregunta de frente/reverso. También, tiene un payload que será un objeto JSON con la información de la flashcard, un estado que puede ser “activo” o “inactivo”, una fecha de creación, una fecha de actualización, una fecha de último repaso y una calificación del último repaso de 1 a 5.

Figura 9: Modelo de datos ERD.



Fuente: Autor

6 IMPLEMENTACIÓN

En este capítulo se expondrá las actividades concernientes a la fase de implementación de STU-AID, que es en donde el desarrollo del artefacto de software toma lugar después de haber formulado los requerimientos y el diseño de la solución.

El objetivo que se busca conseguir es desarrollar la aplicación siguiendo el modelo de arquitectura y cumpliendo con todos los requerimientos especificados.

6.1 IMPLEMENTACIÓN DE LOS MÓDULOS DEL BACKEND

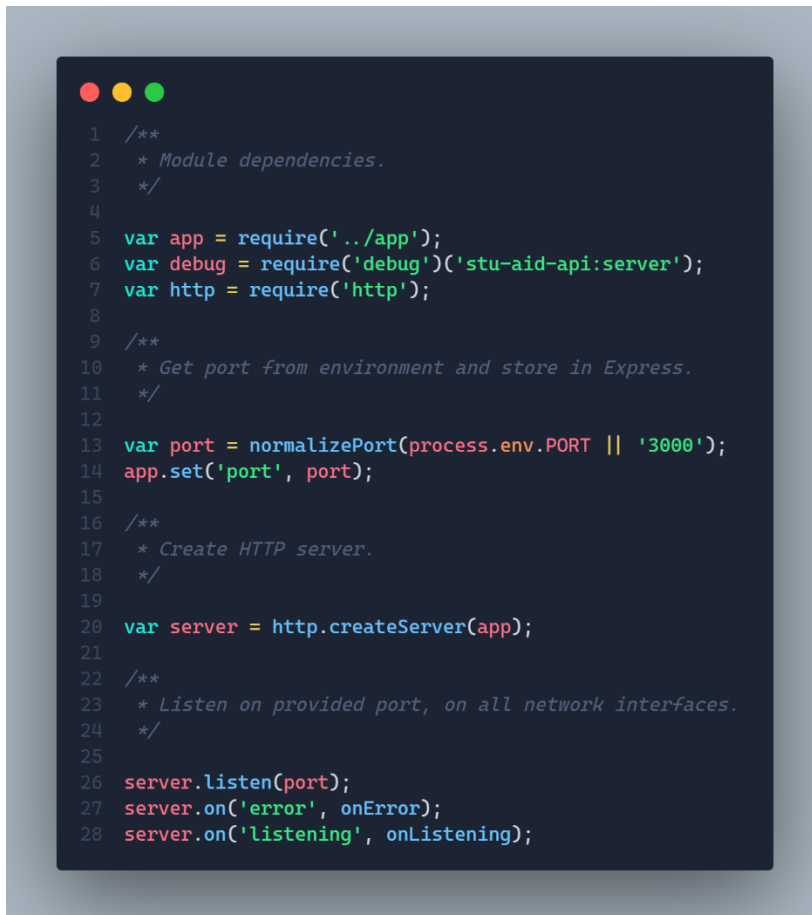
Para la implementación de los módulos del backend se siguió el diseño de The clean architecture o la arquitectura limpia basado en capas, por lo tanto, se hizo uso de las entidades, casos de uso, controladores y adaptadores externos.

6.1.1 Capa de adaptadores externos.

Son representados por los llamados al API REST por parte del cliente que en este caso es la aplicación móvil en React Native que se comunica a través de peticiones HTTP y que son recibidas por la aplicación servidor en NodeJS.

Para levantar la aplicación del servidor en NodeJS se utilizó este código en la raíz del proyecto que es el punto de entrada en la ejecución del programa.

Figura 10: Creación del servidor HTTP

A screenshot of a code editor with a dark background and light-colored text. The code is written in JavaScript and is numbered from 1 to 28. It includes comments in Spanish and uses standard JavaScript syntax for requiring modules, setting environment variables, and creating an HTTP server. The code is as follows:

```
1  /**
2   * Module dependencies.
3   */
4
5  var app = require('../app');
6  var debug = require('debug')('stu-aid-api:server');
7  var http = require('http');
8
9  /**
10   * Get port from environment and store in Express.
11   */
12
13  var port = normalizePort(process.env.PORT || '3000');
14  app.set('port', port);
15
16  /**
17   * Create HTTP server.
18   */
19
20  var server = http.createServer(app);
21
22  /**
23   * Listen on provided port, on all network interfaces.
24   */
25
26  server.listen(port);
27  server.on('error', onError);
28  server.on('listening', onListening);
```

Fuente: Autor

6.1.2 Capa de controladores

En la capa de controladores entran todos los middlewares creados, los cuales sirven como enrutadores de las peticiones, validadores de peticiones, validadores de autenticación, autorización y finalmente de respuesta de petición o manejo de un error. Estos tienen acceso a la capa de servicios que representa la lógica de los casos de uso.

Los middlewares deben ser instanciados al principio de la ejecución utilizando el método “use”

Figura 11: Configurando los middlewares

A screenshot of a code editor with a dark background and light-colored text. The code is written in JavaScript and shows the configuration of Express.js middlewares. It includes imports for 'express', 'cookie-parser', 'morgan', and an error handler. A function 'apiMiddlewares' is defined, which uses 'app.use' to apply these middlewares to an Express application. The code is numbered from 1 to 20.

```
1  const express = require("express");
2  const cookieParser = require("cookie-parser");
3
4  const logger = require("morgan");
5  const {
6    logHttpRequest,
7    boomErrorHandler,
8    errorHandler,
9  } = require("./error.handler");
10
11 function apiMiddlewares(app) {
12   app.use(logger("dev"));
13   app.use(express.urlencoded({ extended: false }));
14   app.use(cookieParser());
15   app.use(logHttpRequest);
16   app.use(boomErrorHandler);
17   app.use(errorHandler);
18 }
19
20 module.exports = apiMiddlewares;
```

Fuente: Autor

A continuación, se explicará detalladamente en que consiste cada tipo de controlador y su uso en la aplicación.

6.1.2.1 Middlewares de enrutación

Estos permiten que la enrutar una petición HTTP al flujo de middlewares que le permitirán llevar al recurso deseado, por ejemplo, si la petición tiene una URL de <http://localhost:3000/auth>, una vez la petición llegue al servidor esta será enviada al middleware que maneja la ruta de *auth*, o autorización.

Figura 12: Configurando middlewares de enrutación



```
1 function routesApi(app) {  
2   app.use("/", indexRouter);  
3   app.use("/users", usersRouter);  
4   app.use("/courses", coursesRouter);  
5   app.use("/auth", authRouter);  
6 }  
7  
8 module.exports = routesApi;
```

Fuente: Autor

6.1.2.2 Middleware de autenticación

Este controlador se utilizó para autenticar el acceso de un usuario a la plataforma. Como se mencionó en el capítulo anterior, se utilizó JWT y la librería Passport que ofrece la funcionalidad de autenticación de forma segura a través de los JSON Web Tokens.

Figura 13: Middleware de autenticación de usuarios



```
1  router.post(  
2    "/login",  
3    passport.authenticate("local", { session: false }),  
4    async (req, res, next) => {  
5      try {  
6        const { user } = req;  
7        delete user.dataValues.password;  
8        const payload = {  
9          sub: user.id,  
10         username: user.email,  
11       };  
12       const token = jwt.sign(payload, config.jwtSecret);  
13       return res.json({ token, user });  
14     } catch (error) {  
15       next(error);  
16     }  
17   }  
18 );
```

Fuente: Autor

6.1.2.3 Middleware de autorización de usuario

Se utilizó para verificar si un usuario autenticado es propietario de un curso y por lo tanto tiene permisos para verlo o modificarlo.

Figura 14: Middleware de autorización



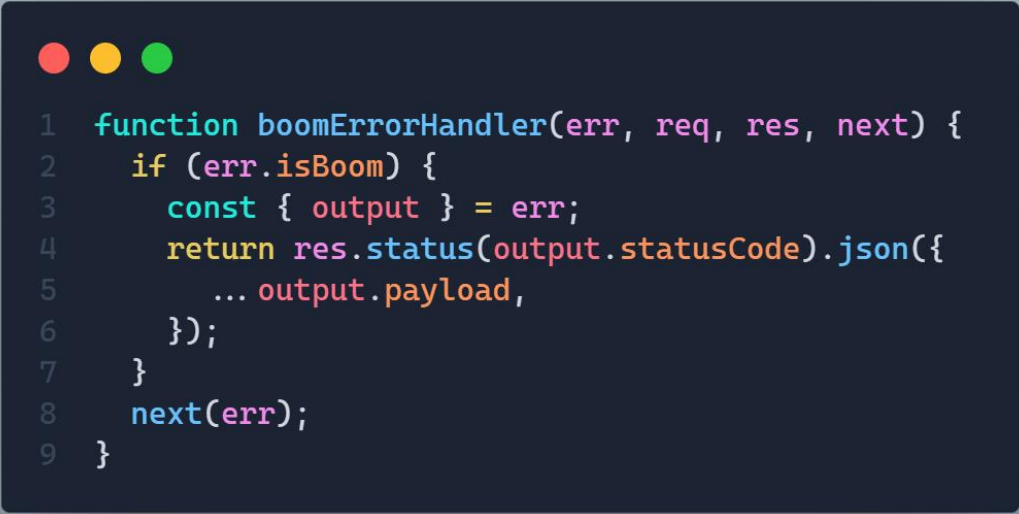
```
1  const boom = require("@hapi/boom");
2
3  function checkOwnerPermission(EntityService, property, attributeName) {
4    return async (req, res, next) => {
5      const userLoggedId = Number(req.user.sub);
6      const entityId = req[property][attributeName];
7      console.log(entityId);
8      try {
9        const isOwner = await EntityService.isOwner(userLoggedId, entityId);
10       console.log(isOwner, "isOwnerMiddleware");
11       if (!isOwner)
12         throw boom.unauthorized("You are not the owner of this resource");
13     } catch (error) {
14       next(error);
15     }
16     next();
17   };
18 }
```

Fuente: Autor

6.1.2.4 Middlewares de manejo de errores

Este middleware se encarga del manejo de errores. Funciona cuando hubo una excepción HTTP y lo que hace es enviar una respuesta con el código del error y el mensaje con los detalles de la excepción. Para este middleware se utilizó una librería llamada boom que permite el manejo de códigos y status de HTTP.

Figura 15: Middleware de errores HTTP



```
1 function boomErrorHandler(err, req, res, next) {  
2   if (err.isBoom) {  
3     const { output } = err;  
4     return res.status(output.statusCode).json({  
5       ... output.payload,  
6     });  
7   }  
8   next(err);  
9 }
```

Fuente: Autor

6.1.3 Capa de casos de uso

Esta capa es la responsable de manejar gran parte de la lógica de negocio, es decir, es la que se encarga de proveer el servicio que es instanciado por el controlador. Esta a su vez, tiene acceso a la capa de entidades. En el caso de la aplicación esta capa está conformada por los servicios. Existe un servicio para cada una de las entidades (Usuario, Curso, Tema, Flashcard). Los servicios tienen acceso a las entidades a través del ORM de Sequelize.

A continuación, se mostrarán cada uno de los servicios que hacen parte de la aplicación.

6.1.3.1 Servicio de usuario

En este servicio se interactúa con la entidad de usuario a través de sequelize.

Figura 16: Servicio de usuarios.

```
1  const sequelize = require("../lib/sequelize");
2  const boom = require("@hapi/boom");
3  const { models } = require("../lib/sequelize");
4
5  class UserService {
6    constructor() {
7      this.db = sequelize;
8    }
9    async findAll() {
10     const users = await models.User.findAll();
11     console.log(users);
12     return users;
13   }
14
15   async findById(id) {
16     const user = await models.User.findByPk(id, {
17       include: ["courses"],
18     });
19     if (!user) throw boom.notFound("User not found by this id");
20     return user;
21   }
22
23   async findByEmail(email) {
24     const user = await models.User.findOne({
25       where: {
26         email,
27       },
28     });
29     return user;
30   }
31
32   async createUser(data) {
33     const userFoundWithSameEmail = await this.findByEmail(data.email);
34     if (userFoundWithSameEmail)
35       throw boom.conflict("There is a user already registered with this email");
36     const newUser = await models.User.create(data);
37     return newUser;
38   }
39 }
40
41 module.exports = UserService;
```

Fuente: Autor

6.1.3.2 Servicio de curso

Este servicio interactúa con la entidad curso.

Figura 17: Servicio de curso.

```
1  const boom = require("@hapi/boom");
2  const { models } = require("../lib/sequelize");
3
4  class CourseService {
5    constructor() {}
6
7    async isOwner(userLoggedId, courseId) {
8      const isCourseOwner = await models.Course.findOne({
9        where: {
10          id: courseId,
11          userId: userLoggedId,
12        },
13      });
14      console.log(isCourseOwner);
15      return Boolean(isCourseOwner);
16    }
17
18    async findAll() {
19      const courses = await models.Course.findAll();
20      return courses;
21    }
22    async findAllByStudentId(studentId) {
23      console.log("FindAllByStudentId");
24      const studentCourses = await models.Course.findAll({
25        where: {
26          userId: studentId,
27        },
28      });
29      return studentCourses;
30    }
31    async findById(id) {
32      const course = await models.Course.findByPk(id, { include: "subjects" });
33      console.log(course, "From service");
34      if (!course) throw boom.notFound("Course not found by this id");
35      return course;
36    }
37    async create(data) {
38      const newCourse = await models.Course.create(data);
39      return newCourse;
40    }
41    async update(id, payload) {
42      const oldCourse = await this.findById(id);
43      const updatedCourse = await oldCourse.update(payload);
44      return updatedCourse;
45    }
46    async delete(id) {
47      const courseFound = await this.findById(id);
48      const deletedCourse = await courseFound.destroy();
49      return { deletedCourse, id };
50    }
51  }
52
53  module.exports = CourseService;
```

Fuente: Autor

6.1.3.3 Servicio de tema

Este servicio interactúa con la entidad de tema.

Figura 18: Servicio de tema

```
1  const { models } = require("../lib/sequelize");
2  const boom = require("@hapi/boom");
3
4  class SubjectService {
5    constructor() {}
6
7    async isOwner(userLoggedId, courseId) {
8      const isOwner = await models.Course.findOne({
9        where: {
10          id: courseId,
11          userId: userLoggedId,
12        },
13      });
14      return Boolean(isOwner);
15    }
16
17    async findById(id) {
18      const subject = await models.Subject.findByIdPk(id, {
19        include: {
20          association: "flashcards",
21        },
22      });
23
24      if (!subject) {
25        throw boom.notFound("No se encontro tema");
26      }
27      return subject;
28    }
29
30    async upsert(payload) {
31      const newSubject = await models.Subject.upsert(payload, {
32        include: {
33          association: "flashcards",
34        },
35      });
36      return newSubject;
37    }
38
39    async delete(id) {
40      const subject = await this.findById(id);
41      const deletedSubject = subject.destroy({ returning: ["id"] });
42      return deletedSubject;
43    }
44  }
45
46  module.exports = SubjectService;
```

Fuente: Autor

6.1.3.4 Servicio de flashcards

Figura 19: Servicio de Flashcards

```
1  const { models } = require("../lib/sequelize");
2  const boom = require("@hapi/boom");
3
4  class FlashcardService {
5    constructor() {}
6    async isOwner(userLoggedId, courseId) {
7      const isOwner = await models.Course.findOne({
8        where: {
9          id: courseId,
10         userId: userLoggedId,
11       },
12     });
13     return Boolean(isOwner);
14   }
15   async findById(id) {
16     const flashcard = await models.Flashcard.findByPk(id);
17     if (!flashcard) {
18       throw boom.notFound("No se encontro flashcard con ese ID");
19     }
20     return flashcard;
21   }
22   async upsert(payload) {
23     const [newFlashcard, isCreated] = await models.Flashcard.upsert(payload);
24     return newFlashcard;
25   }
26
27   async delete(id) {
28     const flashcard = await this.findById(id);
29     const deletedFlashcard = flashcard.destroy({ returning: ["id"] });
30     return deletedFlashcard;
31   }
32 }
33
34 module.exports = FlashcardService;
```

Fuente: Autor

6.1.3.5 Servicio de sesión de estudio

Figura 20: Servicio de sesión de estudio.

```
1 class StudySessionService {
2   constructor() {}
3
4   async subjectsNeedRevision(userId, query) {
5     const options = {
6       include: [
7         {
8           model: models.Course,
9           as: "course",
10          attributes: ["name", "userId"],
11          where: {
12            userId: userId,
13          },
14        },
15        {
16          model: models.Flashcard,
17          as: "flashcards",
18          attributes: ["id"],
19          where: {
20            nextRevision: {
21              [Op.lte]: new Date(),
22            },
23          },
24        },
25      ],
26    };
27    const subjects = await models.Subject.findAll(options);
28    return subjects;
29  }
30
31  async today(userId, query) {
32    const options = {
33      where: {
34        nextRevision: {
35          [Op.lte]: new Date(),
36        },
37        userId: userId,
38      },
39      include: {
40        model: models.Subject,
41        as: "subject",
42        where: {},
43        include: {
44          model: models.Course,
45          as: "course",
46          attributes: ["name", "userId", "id"],
47          where: {
48            userId: userId,
49          },
50        },
51      },
52      order: [["nextRevision"], ["lastScore", "DESC"]],
53    };
54
55    const { offset, limit } = query;
56
57    if (limit && offset) {
58      options.offset = offset;
59      options.limit = limit;
60    }
61
62    const { subject } = query;
63    if (subject) {
64      options.include.where.id = subject;
65    }
66
67    const { course } = query;
68    if (course) {
69      options.include.where.courseId = course;
70    }
71
72    const flashcardsToReview = await models.Flashcard.findAll(options);
73    return flashcardsToReview;
74  }
75 }
76
77 }
```

Fuente: Autor.

6.1.4 Capa de entidades

En esta capa se definen las características de las entidades del sistema que son: usuario, curso, tema y flashcard. Esta definición de entidades se hace por medio de la API del ORM Sequelize. A continuación, se mostrarán cada una de las entidades.

Figura 21: Entidad de usuario

```
1  const { DataTypes, Sequelize } = require("sequelize");
2  const bcrypt = require("bcrypt");
3
4  const USERS_TABLE_NAME = "users";
5  const USERS_SCHEMA = {
6    id: {
7      allowNull: false,
8      autoIncrement: true,
9      primaryKey: true,
10     type: DataTypes.INTEGER,
11   },
12   fullname: {
13     allowNull: false,
14     type: DataTypes.STRING,
15   },
16   email: {
17     allowNull: false,
18     unique: true,
19     type: DataTypes.STRING,
20   },
21   password: {
22     allowNull: false,
23     type: DataTypes.STRING,
24     set(value) {
25       const hash = bcrypt.hashSync(value, 10);
26       this.setDataValue("password", hash);
27     },
28   },
29   createdAt: {
30     field: "created_at",
31     type: DataTypes.DATE,
32   },
33   updatedAt: {
34     field: "updated_at",
35     type: DataTypes.DATE,
36   },
37 };
38
39 module.exports = { USERS_TABLE_NAME, USERS_SCHEMA };
```

Fuente: Autor

Figura 22: Entidad de curso

```
1  const { DataTypes, Sequelize } = require("sequelize");
2  const { USERS_TABLE_NAME } = require("../user.model");
3
4  const COURSES_TABLE_NAME = "courses";
5  const COURSES_SCHEMA = {
6    id: {
7      primaryKey: true,
8      autoIncrement: true,
9      allowNull: false,
10     type: DataTypes.INTEGER,
11   },
12   name: {
13     allowNull: false,
14     type: DataTypes.STRING,
15   },
16   status: {
17     allowNull: true,
18     type: DataTypes.ENUM(["active", "dismissed"]),
19     defaultValue: "active",
20   },
21   priority: {
22     defaultValue: 1,
23     type: DataTypes.INTEGER,
24     validate: {
25       min: 1,
26       max: 5,
27     },
28   },
29   description: {
30     allowNull: true,
31     type: DataTypes.STRING(510),
32   },
33   userId: {
34     field: "user_id",
35     allowNull: false,
36     type: DataTypes.INTEGER,
37     references: {
38       model: USERS_TABLE_NAME,
39       key: "id",
40     },
41     onUpdate: "CASCADE",
42     onDelete: "SET NULL",
43   },
44   createdAt: {
45     field: "created_at",
46     type: DataTypes.DATE,
47   },
48   updatedAt: {
49     field: "updated_at",
50     type: DataTypes.DATE,
51   },
52 };
53
54 module.exports = { COURSES_TABLE_NAME, COURSES_SCHEMA };
```

Fuente: Autor

Figura 23: Entidad de tema

```
1  const { DataTypes, Sequelize } = require("sequelize");
2  const { COURSES_TABLE_NAME } = require("../course.model");
3  const SUBJECTS_TABLE_NAME = "subjects";
4  const SUBJECTS_SCHEMA = {
5    id: {
6      primaryKey: true,
7      allowNull: false,
8      autoIncrement: true,
9      type: DataTypes.INTEGER,
10   },
11   courseId: {
12     field: "course_id",
13     allowNull: false,
14     type: DataTypes.INTEGER,
15     references: {
16       model: COURSES_TABLE_NAME,
17       key: "id",
18     },
19     onUpdate: "CASCADE",
20     onDelete: "CASCADE",
21   },
22   name: {
23     allowNull: false,
24     type: DataTypes.STRING,
25   },
26   status: {
27     allowNull: true,
28     type: DataTypes.ENUM(["active", "unactive"]),
29     defaultValue: "active",
30   },
31   createdAt: {
32     field: "created_at",
33     type: DataTypes.DATE,
34   },
35   updatedAt: {
36     field: "updated_at",
37     type: DataTypes.DATE,
38   },
39 };
40
41 module.exports = { SUBJECTS_SCHEMA, SUBJECTS_TABLE_NAME };
```

Fuente: Autor

Figura 24 : Entidad de flashcard

```
1  const { DataTypes, Sequelize } = require("sequelize");
2  const { SUBJECTS_TABLE_NAME } = require("../subject.model");
3
4  const FLASHCARDS_TABLE_NAME = "flashcards";
5  const FLASHCARDS_SCHEMA = {
6    id: {
7      primaryKey: true,
8      allowNull: false,
9      autoIncrement: true,
10     type: DataTypes.INTEGER,
11   },
12   subjectId: {
13     allowNull: false,
14     type: DataTypes.INTEGER,
15     references: {
16       model: SUBJECTS_TABLE_NAME,
17       key: "id",
18     },
19     onUpdate: "CASCADE",
20     onDelete: "CASCADE",
21   },
22   type: {
23     allowNull: false,
24     type: DataTypes.ENUM(["front-reverse", "true-false", "elaborated"]),
25   },
26   payload: {
27     allowNull: false,
28     type: DataTypes.JSONB,
29   },
30   status: {
31     type: DataTypes.ENUM(["active", "inactive"]),
32     defaultValue: "active",
33   },
34   lastReviewed: {
35     field: "last_reviewed",
36     type: DataTypes.DATE,
37   },
38   lastScore: {
39     field: "last_score",
40     type: DataTypes.DOUBLE,
41     validate: {
42       min: 0,
43       max: 5,
44     },
45   },
46   createdAt: {
47     field: "created_at",
48     type: DataTypes.DATE,
49   },
50   updatedAt: {
51     field: "updated_at",
52     type: DataTypes.DATE,
53   },
54 };
55
56 module.exports = { FLASHCARDS_TABLE_NAME, FLASHCARDS_SCHEMA };
```

Fuente: Autor

6.2 IMPLEMENTACIÓN DE LOS MÓDULOS DEL FRONTEND.

Para la implementación de los módulos del frontend se utilizó la librería React Native que está orientada al patrón de arquitectura flux, y se basa en componentes reutilizables para construir la UI.

6.2.1 Componentes estilizados

Una de las formas en las que se aprovecharon los componentes reutilizables fue en el uso de componentes estilizados que permiten la adaptación de temas de color de manera flexible.

Un ejemplo, es el componente de texto estilizado:

Figura 25: Ejemplo de texto estilizado



Fuente: Autor

Figura 26: Componente de texto estilizado.

```
1  const StyledText = ({ style: propStyleObj = {}, children, ...props }) => {
2    const styles = useThemedStyles(stylesheetCallback);
3    const loadStyles = () => {
4      let allstyles = [styles.default, propStyleObj];
5      allstyles.push(
6        Object.keys(props).map((currentProp) => {
7          if (styles[currentProp]) {
8            return styles[currentProp];
9          }
10         })
11      );
12      return allstyles;
13    };
14    return <Text style={loadStyles()}>{children}</Text>;
15  };
16
17  const stylesheetCallback = (theme) => {
18    return StyleSheet.create({
19      default: {
20        color: theme.themeTokens.textColor,
21        fontSize: theme.typography.size.p,
22      },
23      bold: {
24        fontWeight: "bold",
25      },
26      h1: {
27        fontSize: theme.typography.size.h1,
28      },
29      h2: {
30        fontSize: theme.typography.size.h2,
31      },
32      h3: {
33        fontSize: theme.typography.size.h3,
34      },
35      h4: {
36        fontSize: theme.typography.size.h4,
37      },
38      h5: {
39        fontSize: theme.typography.size.h5,
40      },
41      p: {
42        fontSize: theme.typography.size.p,
43      },
44      underlined: {
45        textDecorationLine: "underline",
46      },
47      wrapText: {
48        flexShrink: 1,
49      },
50      secondaryColor: {
51        color: theme.themeTokens.regularIconColor,
52      },
53      success: {
54        color: theme.themeTokens.successColor,
55      },
56      error: {
57        color: theme.themeTokens.errorColor,
58      },
59      black: {
60        color: theme.themeTokens.colors.black,
61      },
62      red: {
63        color: theme.themeTokens.colors.red,
64      },
65      blue: {
66        color: theme.themeTokens.colors.blue,
67      },
68      gray: {
69        color: theme.themeTokens.colors.gray,
70      },
71      yellow: {
72        color: theme.themeTokens.colors.yellow,
73      },
74      white: {
75        color: theme.themeTokens.colors.white,
76      },
77      green: {
78        color: theme.themeTokens.colors.green,
79      },
80    });
81  };
82
83  export default StyledText;
```

Fuente: Autor.

6.2.2 Uso de custom hooks para llamados del API

Para conectar la aplicación móvil al backend, se utilizaron “custom hooks”, los cuales son funciones que tienen una lógica de negocio específica que puede ser reutilizable en varios componentes.³⁹

Las peticiones HTTP de GET, PUT, DELETE, etcétera son utilizadas por varios componentes a lo largo de la aplicación, es por esto por lo que se hicieron “custom hooks” para reutilizar la lógica cuando se necesite. Un ejemplo es el uso de la petición GET:

Figura 27: Custom hook para una petición GET



```
1 import React, { useCallback, useEffect, useState } from "react";
2 import axios from "axios";
3
4 const useGet = ({ url, headers, params }) => {
5   const [loading, setLoading] = useState(false);
6   const [error, setError] = useState(null);
7   const [data, setData] = useState(null);
8   const [isExecuting, setIsExecuting] = useState(false);
9
10  useEffect(() => {
11    if (isExecuting) {
12      console.log({ data, loading, error });
13    }
14  }, [data, loading, error]);
15
16  const execute = useCallback(async () => {
17    setIsExecuting(true);
18    setLoading(true);
19    try {
20      const response = await axios.get(url, {
21        crossdomain: true,
22        params,
23        headers,
24      });
25      setData(response.data);
26      setError(null);
27    } catch (error) {
28      console.error(error);
29      setData(null);
30      setError(error.response.data.message || error.message);
31    } finally {
32      setLoading(false);
33    }
34    setIsExecuting(false);
35  }, [url, headers, params]);
36
37  return { loading, error, data, execute };
38 };
39
40 export default useGet;
```

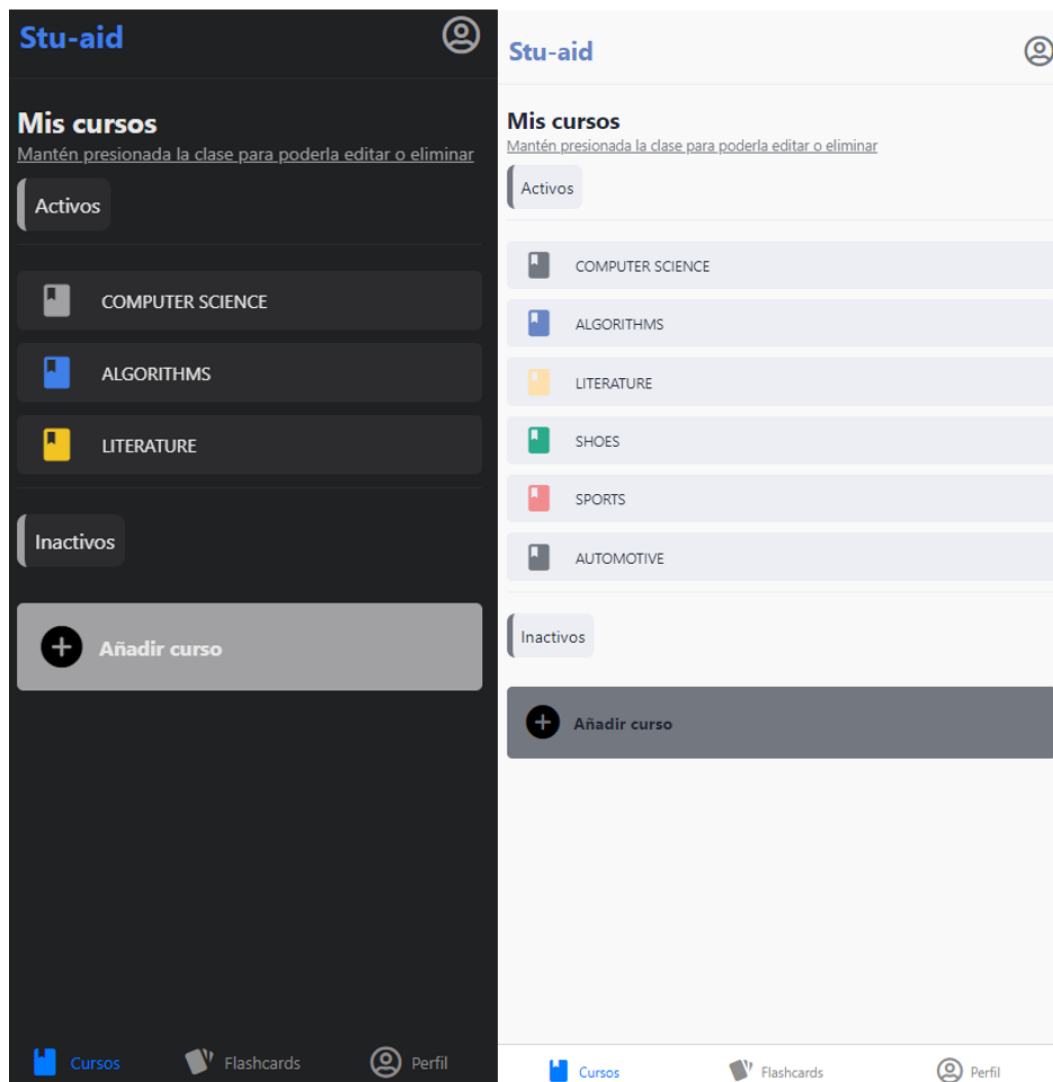
Fuente: Autor

³⁹ REACT DEV. Reusing Logic with Custom Hooks – React. React [página web]. [Consultado el 17, abril, 2023]. Disponible en Internet: <<https://react.dev/learn/reusing-logic-with-custom-hooks>>.

6.2.3 Modo oscuro

Para la optimización de la experiencia de usuario y para el ahorro del consumo de energía de la aplicación, se creó un modo oscuro que se activa si la aplicación se utiliza después de la 6 de la tarde, cuando la luz del ambiente baja. Lo que permite que visualizar el contenido sea cómodo para la vista del usuario.

Figura 28: Modo oscuro y modo claro

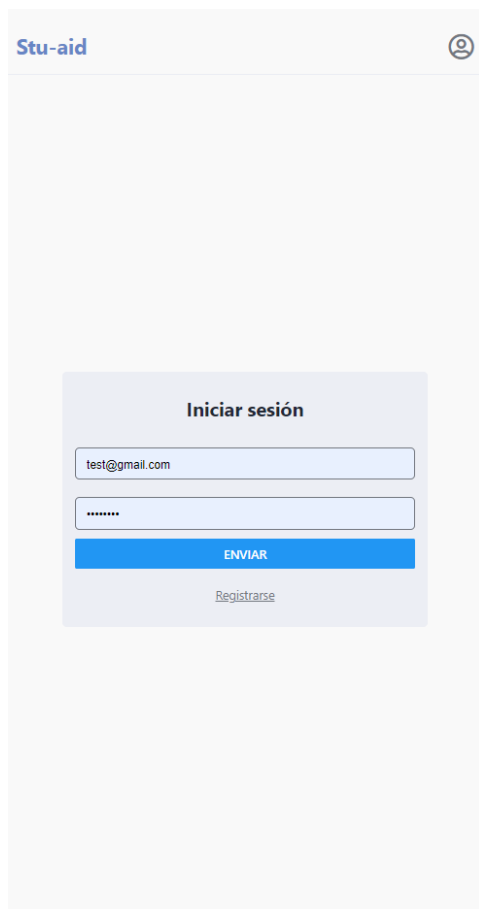


Fuente: Autor.

6.2.4 Autenticación

En este módulo se crearon las vistas de inicio de sesión y registro de cuenta. En estas vistas se utilizaron formularios y la validación de estos se realizó con la librería yup.

Figura 29: Vista de inicio de sesión



Stu-aid

Iniciar sesión

test@gmail.com

ENVIAR

[Registrarse](#)

Fuente: Autor

Figura 30: Vista de registro de cuenta

The image shows a mobile application interface for account registration. At the top, there is a header bar with the text "Stu-aid" on the left and a user profile icon on the right. The main content area is a light gray rectangle. Inside this area, there is a white card with a light blue border. The card has a title "Regístrate para comenzar a usar la app" in bold black text. Below the title are four input fields: "Nombre completo", "test@gmail.com", a password field with six dots, and "Repite la contraseña". Below these fields is a blue button with the text "REGISTRARSE" in white. At the bottom of the card is a link that says "Iniciar sesión" in blue text.

Stu-aid

Regístrate para comenzar a usar la app

Nombre completo

test@gmail.com

.....

Repite la contraseña

REGISTRARSE

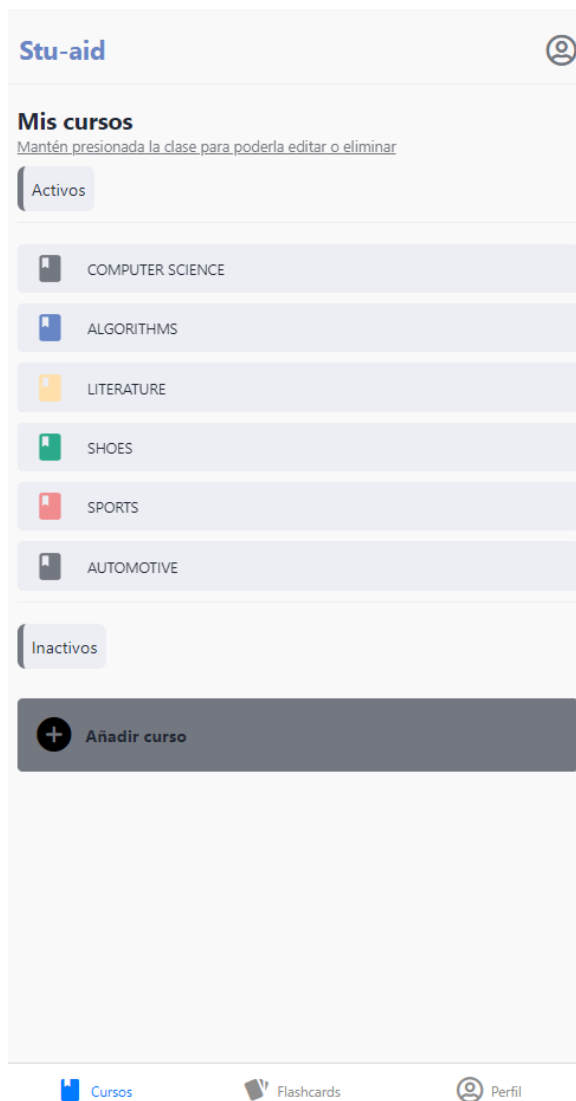
[Iniciar sesión](#)

Fuente: Autor

6.2.5 Cursos

En este módulo se desarrollaron las vistas de ver, añadir, editar y eliminar cursos, los cuales solo pueden ser accedidos una vez el usuario haya iniciado sesión de manera exitosa.

Figura 31: Vista de cursos del usuario.



Fuente: Autor

Figura 32: Vista de añadir curso.

Stu-aid



Añadir curso

Estatus


desactivado

ENVIAR

Fuente: Autor

Figura 33: Vista de editar curso.

Mis cursos
Mantén presionada la clase para poderla editar o eliminar

Activos

- COMPUTER SCIENCE
- ALGORITHMS
- LITERATURE
- SHOES
- SPORTS
- AUTOMOTIVE

Inactivos

Editar curso

Automotive

The Football Is Good For Training And Recreational Purposes

5

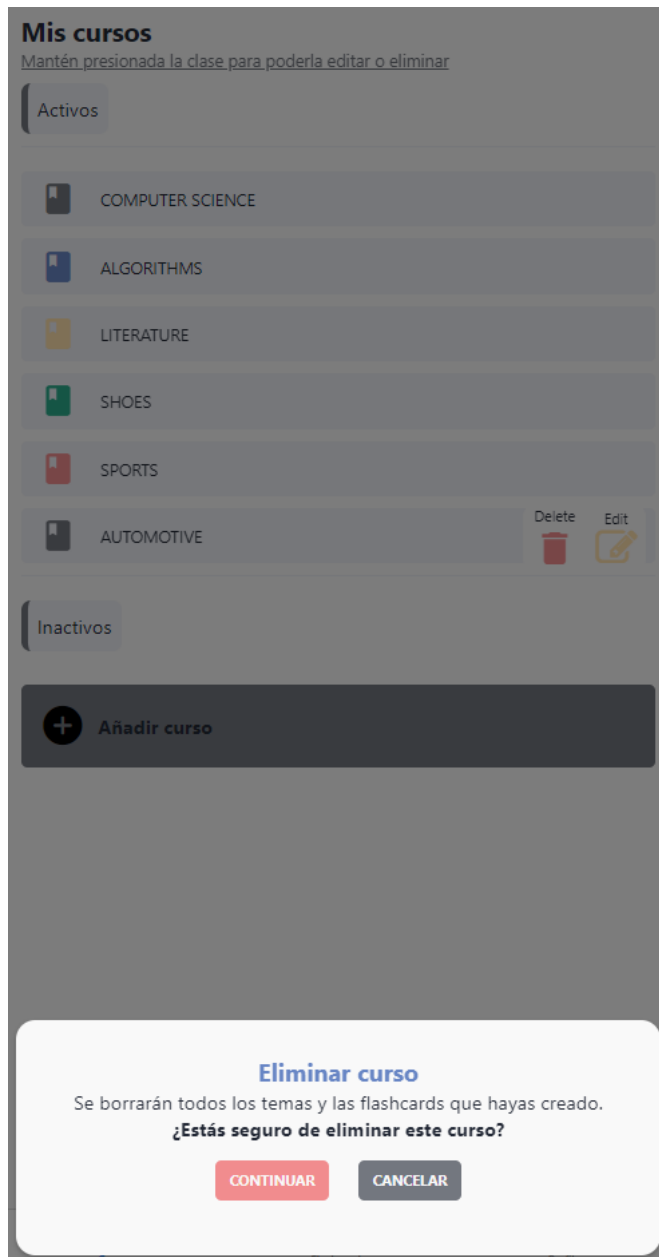
Estatus ☒ activado

ENVIAR

CERRAR

Fuente: Autor

Figura 34: Eliminar curso.

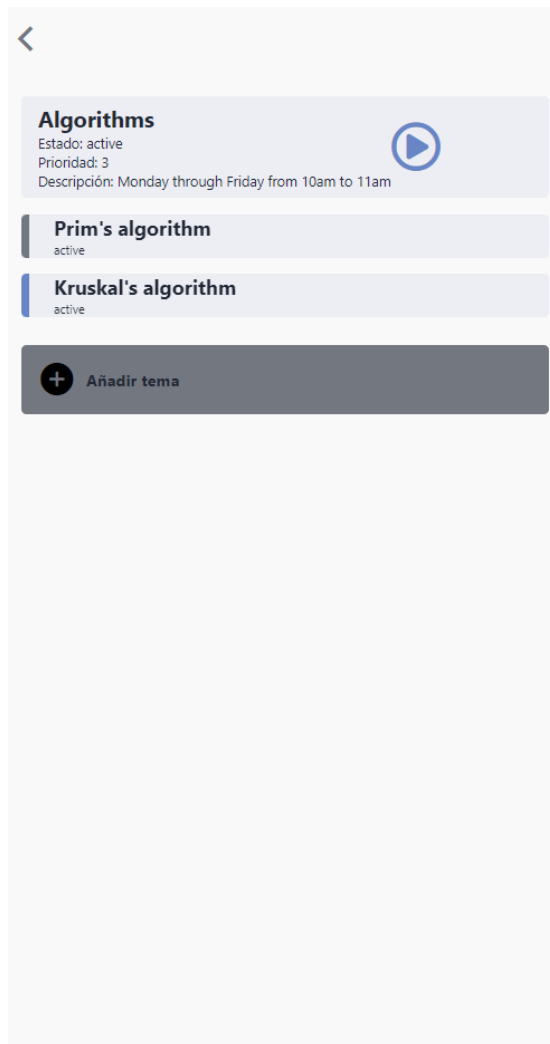


Fuente: Autor

6.2.6 Temas

En este módulo se desarrollaron las vistas de ver, añadir, editar y eliminar temas de estudio, los cuales solo pueden ser accedidos una vez el usuario haya iniciado sesión de manera exitosa y pertenecen a un curso.

Figura 35: Vista de ver temas de estudio.



Fuente: Autor

Figura 36: Vista de crear un tema nuevo.

The screenshot displays a mobile application interface. At the top, there is a back arrow icon. Below it, a section titled "Algorithms" shows a list of items: "Prim's algorithm" and "Kruskal's algorithm", both marked as "active". A play button icon is visible next to the "Algorithms" title. Below the list, there is a button with a plus icon and the text "Añadir tema". At the bottom, a modal titled "Crear tema nuevo" is open. It contains a "Guardar" button, a text input field for "Título", and two read-only fields for "Fecha de creación" and "Fecha de modificación", both showing the date and time "4/17/2023, 12:58:58 PM". There is also a toggle switch for "Estatus" which is currently turned on, labeled "activo".

<

Algorithms
Estado: active
Prioridad: 3
Descripción: Monday through Friday from 10am to 11am

Prim's algorithm
active

Kruskal's algorithm
active

+ Añadir tema

Crear tema nuevo

Guardar

Título

Fecha de creación: 4/17/2023, 12:58:58 PM

Fecha de modificación: 4/17/2023, 12:58:58 PM

Estatus ☒ activo

Fuente: Autor

Figura 37: Vista de editar tema de estudio.



Kruskal's algorithm

Fecha de creación: 4/13/2023, 12:26:44 PM

Fecha de modificación: 4/13/2023, 2:55:02 PM

Estatus

☒ activo

¿Qué es el algoritmo de Prim?

El algoritmo de Prim es un algoritmo de grafos que se utiliza para encontrar el árbol de expansión mínima (MST) de un grafo conexo y no dirigido.

El algoritmo de Prim se utiliza en aplicaciones como la planificación de redes de telecomunicaciones, el enrutamiento de paquetes en redes de computadoras, la ingeniería de circuitos y la biología computacional, entre otras.

Verdadero

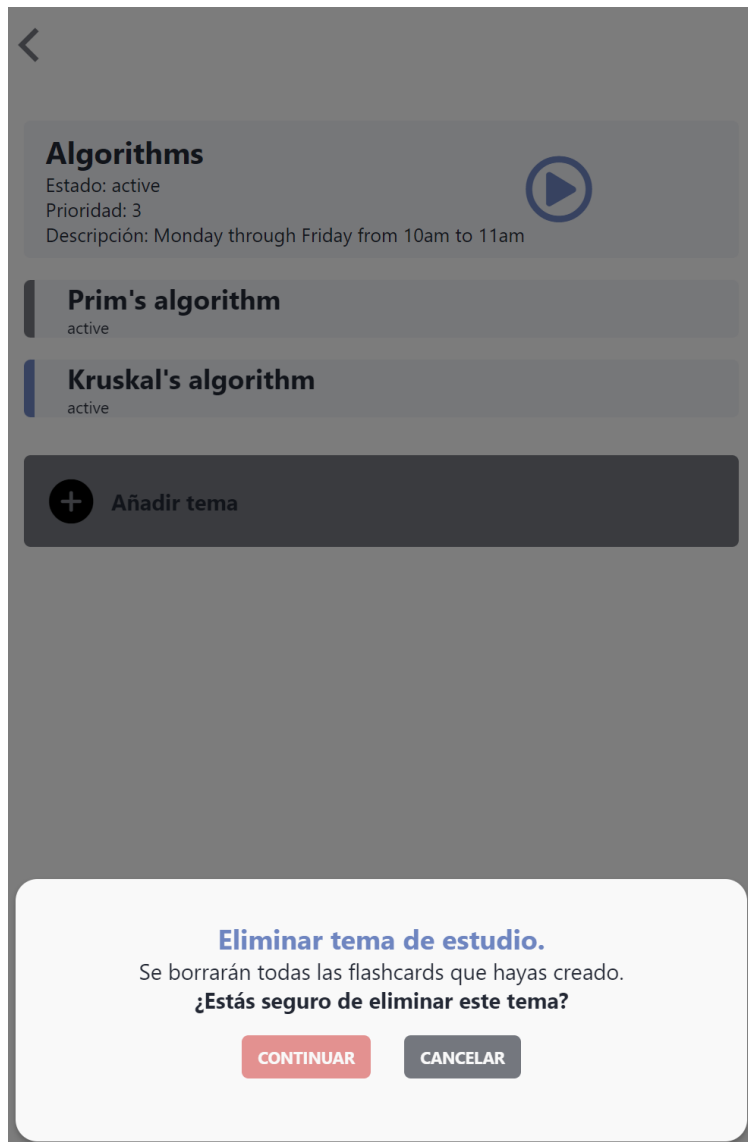
Frente/Reverso

Verdadero/Falso

Elaborada

Fuente: Autor

Figura 38: Vista de eliminar tema de estudio.

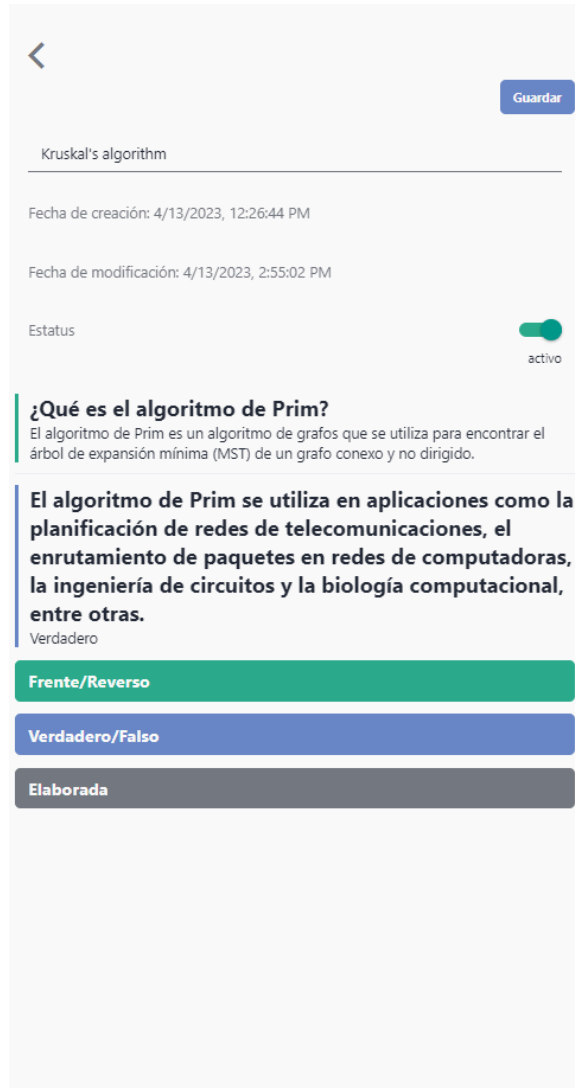


Fuente: Autor.

6.2.7 Flashcards

En este módulo se desarrollaron las vistas de ver, añadir, editar y eliminar, flashcards de tipo frente-reverso, verdadero-falso y elaborada.

Figura 39: Vista de ver flashcards.



<

Guardar

Kruskal's algorithm

Fecha de creación: 4/13/2023, 12:26:44 PM

Fecha de modificación: 4/13/2023, 2:55:02 PM

Estatus ☒ activo

¿Qué es el algoritmo de Prim?
El algoritmo de Prim es un algoritmo de grafos que se utiliza para encontrar el árbol de expansión mínima (MST) de un grafo conexo y no dirigido.

El algoritmo de Prim se utiliza en aplicaciones como la planificación de redes de telecomunicaciones, el enrutamiento de paquetes en redes de computadoras, la ingeniería de circuitos y la biología computacional, entre otras.

Verdadero

Frente/Reverso

Verdadero/Falso

Elaborada

Fuente: Autor

Figura 40: Vista de crear flashcards frente-reverso, verdadero-falso y elaborada.

The image displays three sequential screenshots of a flashcard creation application, all for a card titled "Kruskal's algorithm".

Common Card Content:

- Title:** Kruskal's algorithm
- Creation Date:** Fecha de creación: 4/13/2023, 12:26:44 PM
- Last Modified:** Fecha de modificación: 4/13/2023, 2:55:02 PM
- Status:** activo (indicated by a green toggle switch)
- Question:** ¿Qué es el algoritmo de Prim?
El algoritmo de Prim es un algoritmo de grafos que se utiliza para encontrar el árbol de expansión mínima (MST) de un grafo conexo y no dirigido.
- Answer:** El algoritmo de Prim se utiliza en aplicaciones como la planificación de redes de telecomunicaciones, el enrutamiento de paquetes en redes de computadoras, la ingeniería de circuitos y la biología computacional, entre otras.

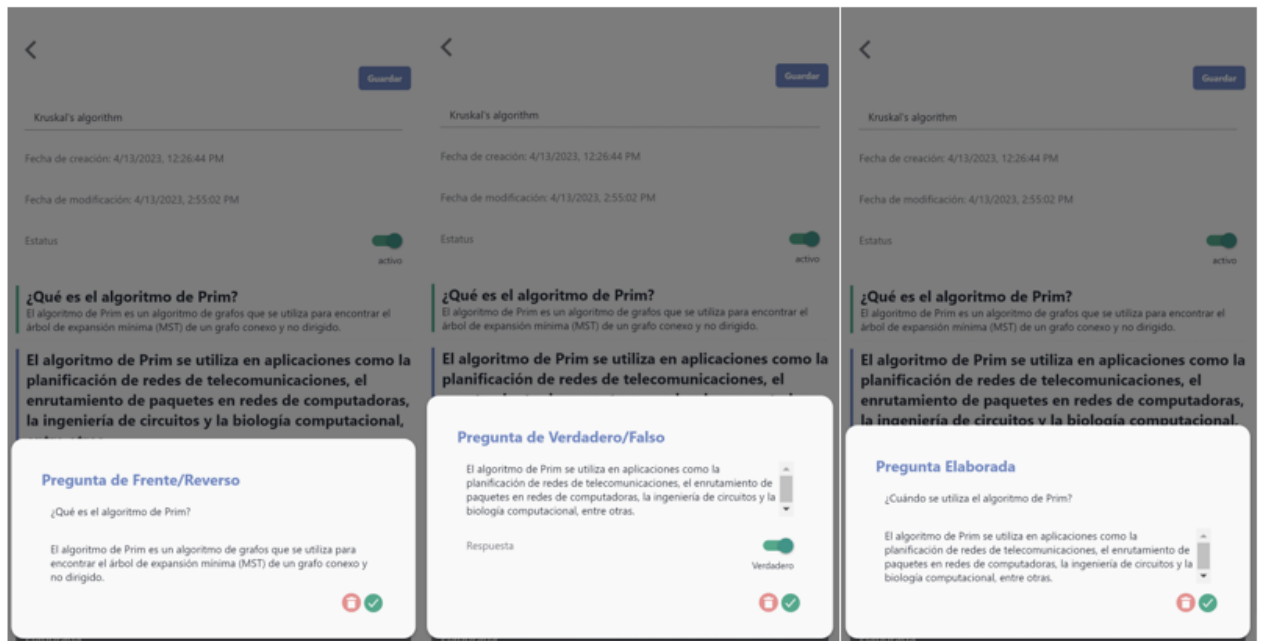
Flashcard Creation Modals:

- Modal 1: Pregunta de Frente/Reverso**
 - Fields: Frente, Reverso
- Modal 2: Pregunta de Verdadero/Falso**
 - Fields: Frente, Respuesta
 - Option: Verdadero (checked)
- Modal 3: Pregunta Elaborada**
 - Fields: Pregunta, Respuesta

Each modal includes a "Guardar" (Save) button and a confirmation icon (a red square with a white 'X' and a green checkmark).

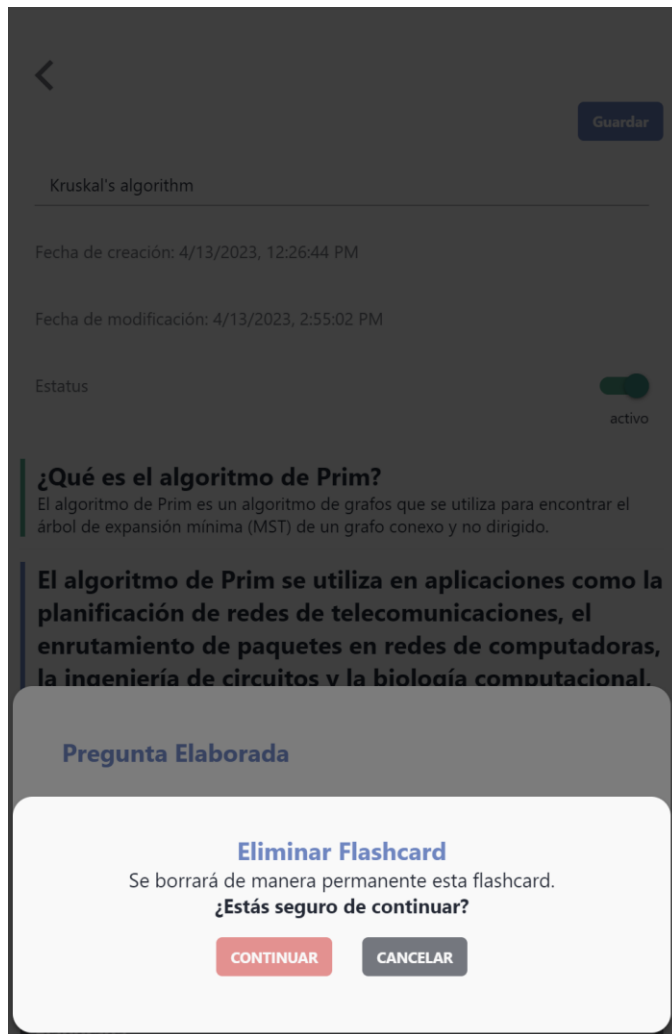
Fuente: Autor.

Figura 41: Vista de editar flashcards.



Fuente: Autor.

Figura 42: Vista de eliminar flashcard.



Fuente: Autor.

6.2.8 Sesión de estudio

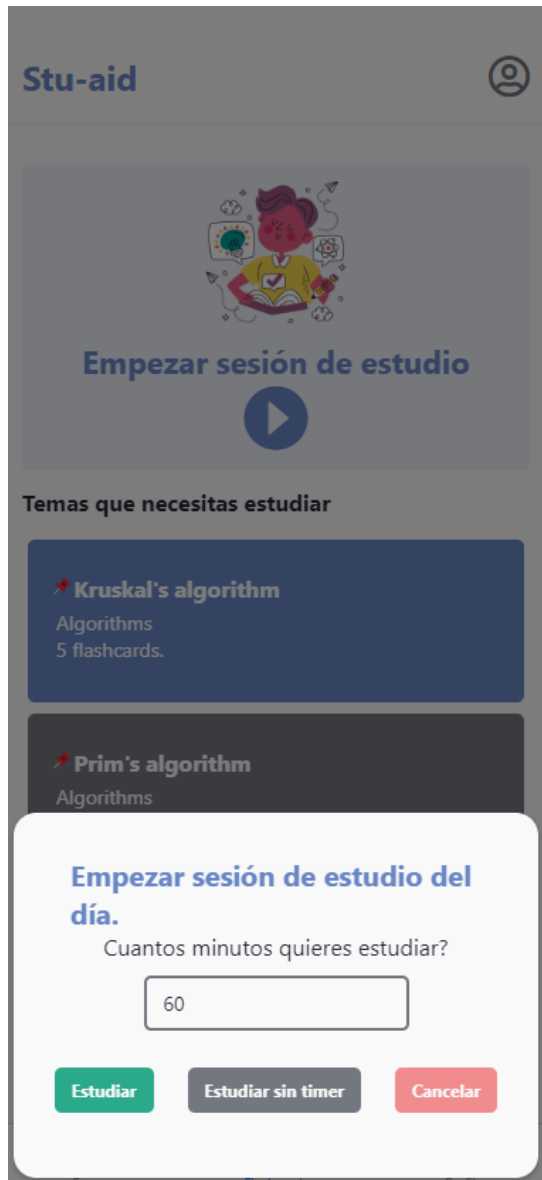
En este módulo se desarrollaron las vistas de sesión de repaso y de ver temas para repasar, estudiar con un temporizador.

Figura 43: Vista de ver temas para repasar.



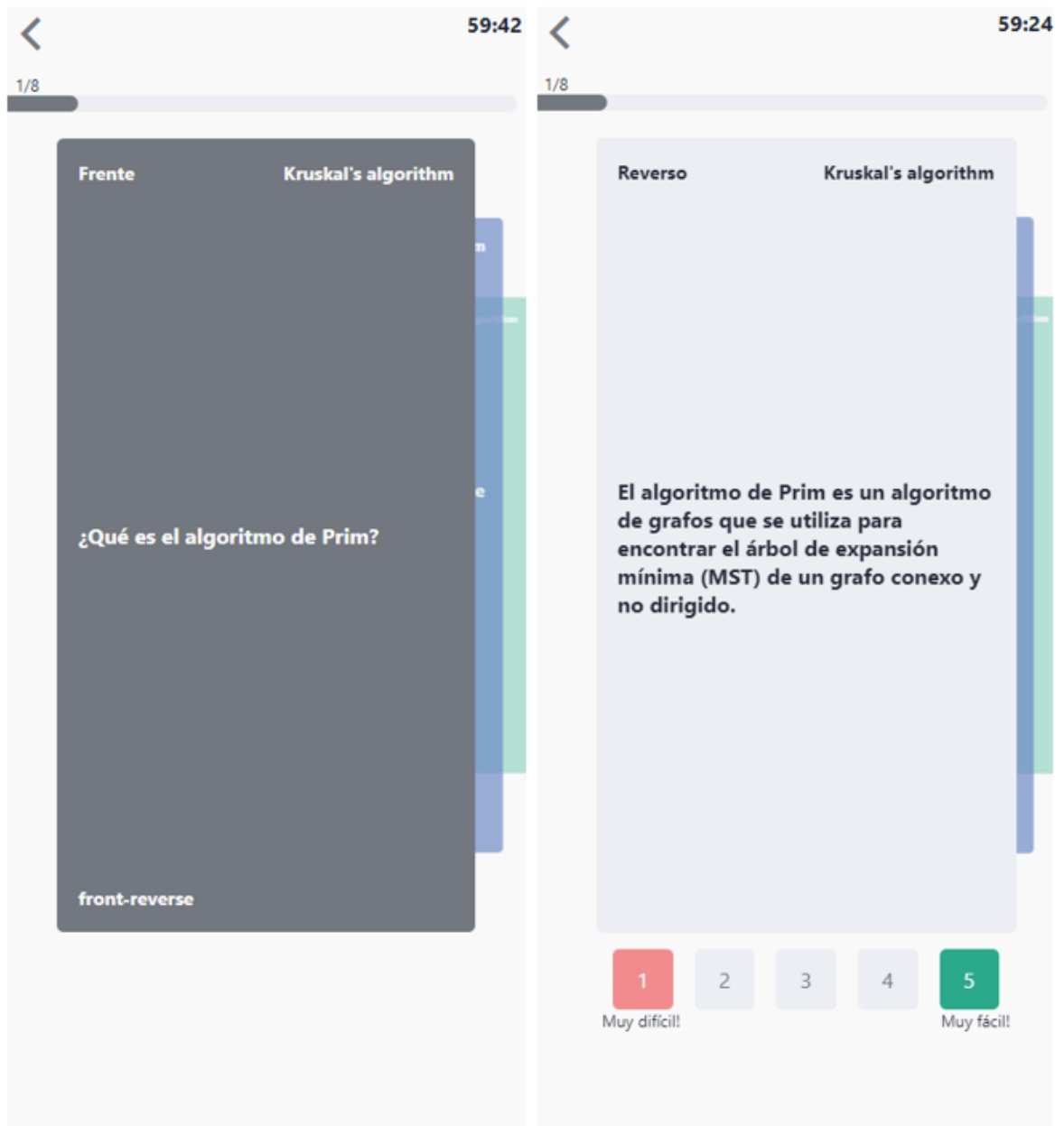
Fuente: Autor.

Figura 44: Vista de estudiar con temporizador.



Fuente: Autor

Figura 45: Vista de sesión de repaso.



Fuente: Autor.

7 PRUEBAS.

En este capítulo se expondrá las actividades concernientes a la fase de pruebas de la aplicación STU-AID, en donde se llevó a cabo la evaluación del artefacto de software para verificar que este cumpla con los requerimientos establecidos.

El objetivo que se busca conseguir es validar el producto de software usando pruebas de integración al producto para garantizar el correcto funcionamiento de cada requerimiento que se levantó y así asegurar la calidad.

7.1 DISEÑO DEL AMBIENTE DE PRUEBAS.

Para el desarrollo de pruebas, se utilizaron datos ficticios de usuarios, cursos, temas y flashcards con la librería Faker.js con el fin de realizar los tests de forma eficaz. Se utilizaron los seeders del ORM Sequelize para poblar la base de datos con los datos de prueba.⁴⁰

⁴⁰ SEQUELIZE. Seeders | Sequelize. Sequelize | Feature-rich ORM for modern TypeScript & JavaScript [página web]. [Consultado el 18, abril, 2023]. Disponible en Internet: <<https://sequelize.org/docs/v6/other-topics/migrations/#creating-the-first-seed>>.

Figura 46: Creación del archivo seed.



```
1  module.exports = {
2    async up(queryInterface, Sequelize) {
3      let users = await models.User.bulkCreate(createUsers(), {
4        returning: ["id"],
5      });
6      console.log("userIds", users);
7
8      let courses = await createCourses(users);
9      console.log("courses", courses);
10     let coursesIds = await models.Course.bulkCreate(courses, {
11       returning: ["id"],
12     });
13     console.log("coursesIds", coursesIds);
14
15     let subjects = await createSubjects(coursesIds);
16     let subjectsIds = await models.Subject.bulkCreate(subjects, {
17       returning: ["id"],
18     });
19
20     console.log("subjectsIds", subjectsIds);
21   },
22
23   async down(queryInterface, Sequelize) {
24     queryInterface.bulkDelete(SUBJECTS_TABLE_NAME, null, {});
25     queryInterface.bulkDelete(COURSES_TABLE_NAME, null, {});
26     queryInterface.bulkDelete(USERS_TABLE_NAME, null, {});
27     console.log("Records borrados");
28   },
29 };;
```

Fuente: Autor

Para el servidor que aloja la API se utilizó nodemon, la cual es una herramienta que ayuda a desarrollar aplicaciones en Node.js reiniciando la aplicación cuando se detecta un cambio, haciendo el proceso de pruebas y correcciones más fácil.⁴¹

Para el servidor que aloja la aplicación móvil de expo y React Native se utilizó la función integrada que permite montar un servidor de pruebas que funciona a través del comando: “expo start”.

⁴¹ NODEMON. nodemon. npm [página web]. [Consultado el 18, abril, 2023]. Disponible en Internet: <<https://www.npmjs.com/package/nodemon>>.

7.2 PRUBAS DE INTEGRACIÓN.

Tabla 6: Prueba de integración HU-001

Prueba de integración		
HU-001 Registrar cuenta		
Descripción		
Como usuario principal lo que quiero es poder registrar una cuenta para poder empezar a utilizar un servicio con una cuenta propia.		
Checklist de criterios de aceptación	Aprobado	Desaprobado
En el evento de registrar cuenta el sistema requiere los campos de (Nombre completo, correo, contraseña, confirmación de contraseña).	X	
Antes de enviar los datos al servidor, se debe confirmar que los campos han sido llenados y de manera válida.	X	
La contraseña debe tener al menos 8 caracteres.	X	
La confirmación de la contraseña y la contraseña deben coincidir.	X	
El email debe ser un email válido	X	
En el evento de recibir la petición de registro por parte del servidor, se deben volver a verificar los campos de la petición. En el caso de ser válidos se procede a registrar el usuario. En el caso contrario se debe devolver un error y el mensaje correspondiente.	X	
En el caso de recibir un error por parte del servidor se debe mostrar el mensaje.	X	
El correo debe ser único, en el caso de que ya exista un usuario con ese correo se debe enviar un error con el mensaje.	X	

Fuente: Autor

Tabla 7: Prueba de integración HU-002

Prueba de integración		
HU-002 Iniciar sesión.		
Descripción		
Como usuario de la aplicación lo que quiero es poder iniciar sesión con mi cuenta para poder acceder a las funcionalidades de la aplicación de manera segura y personalizada.		
Checklist de criterios de aceptación	Aprobado	Desaprobado
En el evento de iniciar sesión con una cuenta, el sistema requiere los campos de (Correo y contraseña).	X	
Antes de enviar los datos al servidor, se debe confirmar que los campos han sido llenados y de manera válida.	X	
La contraseña debe tener al menos 8 caracteres.	X	
El email debe ser un email válido	X	
En el evento de recibir la petición de registro por parte del servidor, se deben volver a verificar los campos de la petición. En el caso de ser válidos se procede a iniciar sesión y redirigir a la vista principal. En el caso contrario se debe devolver un error y el mensaje correspondiente.	X	
En el caso de recibir un error por parte del servidor se debe mostrar el mensaje.	X	

Fuente: Autor.

Tabla 8: Prueba de integración HU-003

Prueba de integración		
HU-003 Ver mis clases		
Descripción		
Como usuario principal lo que quiero es poder ver la lista de mis clases y sus estados para poder organizar mis temas por clase.		
Checklist de criterios de aceptación	Aprobado	Desaprobado
El usuario debe estar debidamente autenticado en la plataforma.	X	
En el caso de que existan clases creadas, se mostraran en la pantalla en forma de lista en donde aparezcan los detalles de la clase que son estado y numero de temas.	X	
En los items de clase se muestra un botón para eliminar la clase.	X	
En la pantalla se muestra un botón para agregar una clase.	X	
En los items de clase se muestra un botón para repasar esa clase.	x	
Se deben mostrar las clases activas e inactivas	x	

Fuente: Autor.

Tabla 9: Prueba de integración HU-004

Prueba de integración		
HU-004 Agregar una clase		
Descripción		
Como usuario lo que quiero es poder agregar una clase en mi aplicación para así poder designarle temas de estudio, asignarle un nombre, una descripción y una prioridad.		
Checklist de criterios de aceptación	Aprobado	Desaprobado
Se debe mostrar un formulario en donde se pida Nombre, Descripción, Prioridad y Estado.	X	
Se deben validar que las entradas del formulario sean válidas antes de enviarlas al sistema.	X	
En el evento de recibir un error por parte del servidor se debe mostrar un aviso de error.	X	
En el evento de recibir un mensaje de éxito se debe notificar un aviso de éxito.	X	

Fuente: Autor.

Tabla 10: Prueba de integración HU-005

Prueba de integración		
HU-005 Eliminar una clase		
Descripción		
Como usuario me gustaría ser capaz de borrar una clase para así limpiar mi aplicación con clases que ya no son de mi interés repasar.		
Checklist de criterios de aceptación	Aprobado	Desaprobado
El usuario debe estar debidamente autenticado en la plataforma.	X	
En el caso de borrar la clase se debe mantener oprimido la clase y se mostrará el botón de editar y borrar clase.	X	
En el caso de que se seleccione borrar la clase se debe notificar al usuario una advertencia de que se borrarán todos los elementos del curso.	X	
En el caso de cancelar la operación se vuelve a la vista de clases	X	
En el caso de aceptar la advertencia se procede a borrar la clase y se vuelve a la vista de clases recargando la clases del estudiante.	X	

Fuente: Autor.

Tabla 11: Prueba de integración HU-007

Prueba de integración		
HU-007 Ver temas de estudio de una clase.		
Descripción		
Como usuario principal lo que quiero es ver los temas de cada curso para así dividir mejor los contenidos de un curso.		
Checklist de criterios de aceptación	Aprobado	Desaprobado
El usuario debe estar debidamente autenticado en la plataforma.	X	
En el caso de que existan temas creados, se mostraran en la pantalla en forma de lista en donde apareceran los temas del curso y su estado.	X	
En el caso de que no existan temas creados se mostrara en la pantalla un aviso de que no existen temas actualmente.	X	
En la pantalla se muestra un botón para agregar una tema.	X	
En los items de clase se muestra un botón para eliminar un tema	X	
En los items de clase se muestra un botón para repasar la clase.	X	
Se deben mostrar los temas activos e inactivos	X	

Fuente: Autor.

Tabla 12: Prueba de integración HU-008.

Prueba de integración		
HU-008 Crear un tema de estudio.		
Descripción		
Como usuario lo que quiero es poder crear temas de estudio que pertenezcan a una curso que este realizando para poder organizar mis flashcards en temas y clases.		
Checklist de criterios de aceptación	Aprobado	Desaprobado
Debe aparecer un botón de agregar tema en la sección de la clase	X	
Se debe mostrar un formulario en donde se pida Nombre y aparezca la fecha de creación y la fecha de actualización y el area de tomar notas.	X	
Se deben validar que las entradas del tema sean válidas antes de guardar	X	
En el evento de recibir un error por parte del servidor se debe mostrar un aviso de error.	X	
En el evento de recibir un mensaje de éxito se debe notificar un aviso de éxito.	X	

Fuente: Autor.

Tabla 13 Prueba de integración HU-009

Prueba de integración		
HU-009 Eliminar un tema de estudio.		
Descripción		
Como usuario principal lo que quiero es poder eliminar temas de estudio que ya no quiero repasar.		
Checklist de criterios de aceptación	Aprobado	Desaprobado
El usuario debe estar debidamente autenticado en la plataforma.	X	
En el caso de borrar un tema se debe mantener oprimido el tema y se mostrará el botón de borrar clase.	X	
En el caso de que se seleccione borrar el tema se debe notificar al usuario una advertencia de que se borrarán todos los flashcards del tema.	X	
En el caso de cancelar la operación se vuelve a la vista de temas	X	
En el caso de aceptar la advertencia se procede a borrar el tema y se vuelve a la vista de temas recargando los temas del estudiante.	X	

Fuente: Autor.

Tabla 14: Prueba de integración HU-010.

Prueba de integración		
HU-010 Editar el estado de un tema de estudio.		
Descripción		
Como usuario lo que quiero es poder cambiar el estado de un tema de estudio y poder cambiar su nombre para así tener mayor control sobre los temas que me interesan.		
Checklist de criterios de aceptación	Aprobado	Desaprobado
Se debe mostrar un formulario con los datos actuales del tema: Nombre, estado y flashcards.	X	
Se deben validar que las entradas del formulario sean válidas antes de enviarlas al sistema.	X	
En el evento de recibir un error por parte del servidor se debe mostrar un aviso de error.	X	
En el evento de recibir un mensaje de éxito se debe notificar un aviso de éxito.	X	

Fuente: Autor.

Tabla 15: Prueba de integración HU-011

Prueba de integración		
HU-011 Ver preguntas de un tema de estudio.		
Descripción		
Como usuario lo que quiero es poder ver las preguntas de un tema de estudio en particular para tener conocimiento de cómo se compone el tema a partir de las preguntas.		
Checklist de criterios de aceptación	Aprobado	Desaprobado
El usuario debe estar debidamente autenticado en la plataforma.	X	
Se muestran las preguntas en campos editables.	X	
Se muestra la información del tema en campos editables	X	
Se muestran las opciones para crear una nueva pregunta	X	
Si se selecciona una pregunta aparece opción de guardar cambios o de eliminar.	X	

Fuente: Autor.

Tabla 16: Prueba de integración HU-012.

Prueba de integración		
HU-012 Ver opciones de crear pregunta.		
Descripción		
Como usuario lo que quiero es poder crear una pregunta de tipo verdadero / falso, elaborada y de frente/reverso para poder fragmentar los temas que estoy estudiando.		
Checklist de criterios de aceptación	Aprobado	Desaprobado
El usuario debe estar debidamente autenticado en la plataforma.	X	
El usuario ve en la pantalla las opciones de crear preguntas de tipo verdadero / falso, elaborada y de frente/reverso.	X	
En el evento de que el usuario selecciona una de las opciones se muestra en la pantalla un formulario con los campos de la pregunta.	X	

Fuente: Autor.

Tabla 17: Prueba de integración HU-013

Prueba de integración		
HU-013 Crear pregunta elaborada/construida		
Descripción		
Como usuario lo que quiero es poder crear preguntas elaboradas/construidas para mis sesiones de repaso.		
Checklist de criterios de aceptación	Aprobado	Desaprobado
El usuario debe estar debidamente autenticado en la plataforma.	X	
El usuario debe estar en la vista del temas	X	
Al seleccionar la opción de crear pregunta elaborada se mostrará un formulario con los campos de pregunta y respuesta.	X	
Aparece un botón de guarda y eliminar pregunta.	X	
El formulario valida que los campos sean llenados antes de guardar los cambios.	X	
En el evento que se guarden los cambios se enviará la petición al servidor.	X	

Fuente: Autor.

Tabla 18: Prueba de integración HU-014

Prueba de integración		
HU-014 Crear Pregunta dicotómica Verdadero/ Falso.		
Descripción		
Como usuario principal de la aplicación lo que quiero es poder crear preguntas de verdadero o falso para mis sesiones de repaso.		
Checklist de criterios de aceptación	Aprobado	Desaprobado
El usuario debe estar debidamente autenticado en la plataforma.	X	
El usuario debe estar en la vista del temas	X	
Al seleccionar la opción de crear pregunta elaborada se mostrará un formulario con los campos de pregunta y respuesta.	X	
Aparece un botón de guarda y eliminar pregunta.	X	
El formulario valida que los campos sean llenados antes de guardar los cambios.	X	
En el evento que se guarden los cambios se enviará la petición al servidor.	X	

Fuente: Autor.

Tabla 19: Prueba de integración HU-015

Prueba de integración		
HU-015 Crear Pregunta definición concepto (Frente/Reverso).		
Descripción		
Como usuario de la aplicación lo que quiero es poder crear preguntas de tipo frente/reverso para mis sesiones de repaso.		
Checklist de criterios de aceptación	Aprobado	Desaprobado
El usuario debe estar debidamente autenticado en la plataforma.	X	
El usuario debe estar en la vista del temas	X	
Al seleccionar la opción de crear pregunta elaborada se mostrará un formulario con los campos de pregunta y respuesta.	X	
Aparece un botón de guarda y eliminar pregunta.	X	
El formulario valida que los campos sean llenados antes de guardar los cambios.	X	
En el evento que se guarden los cambios se enviará la petición al servidor.	X	

Fuente: Autor.

Tabla 20: Prueba de integración HU-016

Prueba de integración		
HU-016 Eliminar pregunta.		
Descripción		
Como usuario principal de la aplicación lo que quiero es poder eliminar una pregunta que ya no quiero que este en las sesiones de repaso.		
Checklist de criterios de aceptación	Aprobado	Desaprobado
El usuario debe estar debidamente autenticado en la plataforma.	X	
El usuario debe estar en la vista del temas	X	
Al seleccionar una preguntata se mostrará un formulario con los campos de pregunta y respuesta.	X	
Aparece un botón de guarda y eliminar pregunta.	X	
En el evento de eliminar la pregunta saldrá un mensaje de confirmación.	X	
En el evento de que se confirme la eliminación, se procede a eliminar la pregunta.	X	
En el caso de cancelar la operación de eliminación se volverá a la vista del tema.	X	

Fuente: Autor.

Tabla 21: Prueba de integración HU-017.

Prueba de integración		
HU-017 Editar pregunta.		
Descripción		
Como usuario lo que quiero es poder editar una pregunta a mi gusto.		
Checklist de criterios de aceptación	Aprobado	Desaprobado
El usuario debe estar debidamente autenticado en la plataforma.	X	
El usuario debe estar en la vista del temas	X	
Al seleccionar la pregunta a editar, se mostrará un formulario con los campos de pregunta y respuesta.	X	
Aparece un botón de guarda y eliminar pregunta.	X	
El formulario valida que los campos sean llenados antes de guardar los cambios.	X	
En el evento que se guarden los cambios se guardaran los cambios se enviará la petición al servidor.	X	

Fuente: Autor.

Tabla 22: Prueba de integración HU-018.

Prueba de integración		
HU-018 Visualizar pregunta en flashcard.		
Descripción		
Como usuario lo que quiero es poder visualizar las preguntas en forma de flashcards para poder estudiarlas en las sesiones de repaso.		
Checklist de criterios de aceptación	Aprobado	Desaprobado
En la tarjeta se muestra la parte frontal al inicio, se muestra el tema de estudio, el tipo de pregunta y el contenido de la pregunta.	X	
En el evento de dar clic en la tarjeta la parte frontal se voltea y se muestra la parte del reverso con el contenido de la respuesta y el tema de estudio.	X	

Fuente: Autor.

Tabla 23: Prueba de integración HU-019.

Prueba de integración		
HU-019 Empezar sesión de estudio del día.		
Descripción		
Como usuario lo que quiero es poder estudiar las tarjetas que probablemente haya olvidado en cierto porcentaje.		
Checklist de criterios de aceptación	Aprobado	Desaprobado
El usuario debe estar debidamente autenticado en la plataforma.	X	
En la pantalla de flashcards existe un botón para empezar la sesión de estudio del día.	X	
En el caso de empezar la sesión del día, se mostraran las flashcards que necesitan ser repasadas en forma de frente y reverso.	X	
A medida que pasan las tarjetas el usuario debe contestar las preguntas dependiendo del tipo de pregunta (Frente/Reverso, Verdadero/Falso o Elaborada).	X	
Al finalizar el recorrido de las tarjetas se muestra un modal avisando que la sesión de estudio finalizó.	X	

Fuente: Autor.

Tabla 24: Prueba de integración HU-020

Prueba de integración		
HU-020 Empezar sesión de estudio de un tema.		
Descripción		
Como usuario lo que quiero es repasar las preguntas de un tema en específico.		
Checklist de criterios de aceptación	Aprobado	Desaprobado
El usuario debe estar debidamente autenticado en la plataforma.	X	
En la pantalla de flashcards se muestran los temas que necesitan ser repasados.	X	
En el caso de seleccionar un tema en específico, se mostrarán las flashcards que necesitan ser repasadas en forma de frente y reverso que pertenecen al tema.	X	
A medida que pasan las tarjetas el usuario debe contestar las preguntas dependiendo del tipo de pregunta (Frente/Reverso, Verdadero/Falso o Elaborada).	X	
Al finalizar el recorrido de las tarjetas se muestra un modal avisando que la sesión de estudio finalizó.	X	

Fuente: Autor.

Tabla 25: Prueba de integración HU-021

Prueba de integración		
HU-021 Empezar sesión de estudio de una materia.		
Descripción		
Como usuario lo que quiero es repasar las tarjetas de un curso en específico sin importar si no necesitan ser repasadas.		
Checklist de criterios de aceptación	Aprobado	Desaprobado
El usuario debe estar debidamente autenticado en la plataforma.	X	
En la pantalla de flashcards se muestran los temas que necesitan ser repasados.	X	
En el caso de seleccionar un tema en específico, se mostrarán las flashcards que necesitan ser repasadas en forma de frente y reverso que pertenecen al tema.	X	
A medida que pasan las tarjetas el usuario debe contestar las preguntas dependiendo del tipo de pregunta (Frente/Reverso, Verdadero/Falso o Elaborada).	X	
Al finalizar el recorrido de las tarjetas se muestra un modal avisando que la sesión de estudio finalizó.	X	

Fuente: Autor.

Tabla 26: Prueba de Integración HU-022

Prueba de integración		
HU-022 Empezar sesión de estudio por tiempo.		
Descripción		
Como usuario lo que quiero es repasar las flashcards en una sesión de tiempo.		
Checklist de criterios de aceptación	Aprobado	Desaprobado
El usuario debe estar debidamente autenticado en la plataforma.	X	
En la pantalla de flashcards se muestran los temas que necesitan ser repasados.	X	
Se mostrará un temporizador en el momento iniciar hasta finalizar la sesión.	X	
En el caso de seleccionar un tema en específico, se mostrarán las flashcards que necesitan ser repasadas en forma de frente y reverso que pertenecen al tema.	X	
A medida que pasan las tarjetas el usuario debe contestar las preguntas dependiendo del tipo de pregunta (Frente/Reverso, Verdadero/Falso o Elaborada).	X	
Al finalizar el recorrido de las tarjetas o al terminar el tiempo se muestra un modal avisando que la sesión de estudio finalizó.	X	

Fuente: Autor.

Tabla 27: Prueba de integración HU-023

Prueba de integración		
HU-023 Responder pregunta.		
Descripción		
Como usuario quiero responder las preguntas en mis sesiones de estudio, de tal forma que pueda poner a prueba mi conocimiento y calificar la dificultad de la pregunta de 1 a 5, donde 1 fue muy difícil y 5 muy fácil.		
Checklist de criterios de aceptación	Aprobado	Desaprobado
Se debe estar en una sesión de estudio para poder responder una pregunta.	X	
Se debe estar debidamente autenticado en la plataforma.	X	
Se debe mostrar el frente de la pregunta primero. Para las flashcards de frente/reverso no se pone campo de respuesta, para las de verdadero/falso se muestran dos botones con la opción de verdadero y falso y para el caso de las elaboradas se muestra un formulario para que el usuario ponga su respuesta.	X	
Una vez el usuario de su respuesta se muestra el reverso con la respuesta correcta y la opción de calificar el nivel de dificultad de 1 a 5.	X	
En el evento de que el usuario seleccione cualquier nivel de dificultad se debe enviar la petición al servidor y este debe calcular y asignar la siguiente fecha de repaso y devolver los datos a la aplicación.	X	
Una vez los datos con la nueva fecha de repaso son recibidos por la aplicación se pasa a la siguiente flashcard.	X	
Para la dificultad 1 se programa la flashcard para el día siguiente, para la dificultad 2 se programa la flashcard para 2 días después, para la dificultad 3 se programa para 7 días después, para la 4 se programa para 15 días después, y para la 5 se programa para 30 días después.	X	

Fuente: Autor.

Tabla 28: Prueba de integración HU-024.

Prueba de integración		
HU-024 Conocer el progreso de la sesión de estudio.		
Descripción		
Cómo usuario quiero conocer el progreso de mi sesión de estudio para saber cuántas flashcards he repasado y cuantas hacen falta.		
Checklist de criterios de aceptación	Aprobado	Desaprobado
Al empezar sesión se observa una línea de progreso.	X	
A medida que se responden las flashcards se ve el cambio en la línea de progreso.	X	
Se observa las flashcards repasadas y las flashcards totales.	X	
Al finalizar el recorrido de las tarjetas se muestra un modal avisando que la sesión de estudio finalizó.	X	

Fuente: Autor.

8 CONCLUSIONES.

Stu-aid es una aplicación móvil para la ayuda de los estudios académicos, pues en los requerimientos se incluyeron técnicas como la repetición espaciada y el método de flashcards de Leitner para combatir el olvido de conceptos a lo largo del tiempo, fenómeno representado en la curva de olvido propuesta por Ebbinghaus.

Se diseñó un modelo de arquitectura que se acomodó para cumplir con los requerimientos funcionales descritos en 24 historias de usuario y no funcionales que fueron usabilidad, seguridad, procesamiento en tiempo real y desempeño.

Se implementaron los módulos de código siguiendo el modelo de arquitectura y los requerimientos levantados, documentando cada uno de estos tanto en la parte del frontend, como del backend.

Se validó el producto de software usando pruebas de integración a la aplicación para garantizar el correcto funcionamiento de cada requerimiento que se levantó. Se realizaron 24 pruebas de integración, las cuales fueron aprobadas en un 100%.

Stu-aid al ser una aplicación digital reduce la tendencia al uso de hojas y fichas que se tendría que utilizar en un sistema de repetición espaciada análogo, lo cual ayuda a la disminución de la huella ecológica.

Se incluyó una función que hace compatible la aplicación de Stu-aid con modo oscuro, la cual permite mejorar el consumo de energía de la aplicación ayudando a optimizar el uso de energía en el celular.

9 RECOMENDACIONES.

Aunque se culminó con el proyecto y los requerimientos establecidos, se puede decir que existe un camino abierto para implementar nuevas funcionalidades como, por ejemplo: el uso de notificaciones que le avisen al usuario de sus sesiones de estudio pendientes, o la ayuda de inteligencia artificial para generar las preguntas de un tema dado, la implementación de entrada de distintos formatos como video, imágenes y texto enriquecido para las flashcards, entre otros.

Todo esto es un trabajo que se propone a futuro si alguna persona está dispuesta a mejorar la aplicación de Stu-aid.

10 REFERENCIAS

Anki Web. Aplicativo web para memorizar cosas [en línea] [Consultado el 20 de abril de 2023.]

BABACCI, Mario, et al. Quality Attributes. En: Carnegie Mellon University Software Engineering Institute [en línea]. 1995. [Consultado el 24, marzo, 2023]. Disponible en Internet: <https://resources.sei.cmu.edu/asset_files/TechnicalReport/1995_005_001_164>.

BROWN, Peter C. Make it stick. [s.l.]: Harvard University Press, 2014. 329 p. ISBN 9780674419377.

CREA FICHAS de Memoria con GoConqr [Anónimo]. GoConqr [página web]. [Consultado el 3, octubre, 2022]. Disponible en Internet: <<https://www.goconqr.com/es/fichas/>>.

CRUZ GARZÓN, Javier Orlando. Ambiente e-learning adaptativo en Moodle para estudiantes de grado 11, basado en estilos de aprendizaje. Trabajo de grado. Chía: Universidad de la sabana, 2015. 235 p.

DOCKER DOCS. Docker overview. Docker Documentation [página web]. [Consultado el 13, marzo, 2023]. Disponible en Internet: <<https://docs.docker.com/get-started/overview/>>.

EBBINGHAUS, Hermann. Memory: A Contribution to Experimental Psychology. [s.l.]: Creative Media Partners, LLC, 2018. ISBN 9780344159756.

EXPO. What is Expo? Expo Documentation [página web]. [Consultado el 14, marzo, 2023]. Disponible en Internet: <<https://docs.expo.dev/introduction/expo/>>.

Gomez, María. Aplicación para memorizar usando repetición espaciada. 2013. [consultado el 11, septiembre, 2022]. Disponible en Internet: <https://search.ebscohost.com/login.aspx?direct=true&db=edsoai&AN=edsoai.on1133019066&lang=es&site=eds-live>.

HAROON-SULYMAN, Shakirat. Client-Server Model. En: ResearchGate [en línea]. 2014. p. 67-69. [Consultado el 13, marzo, 2023]. Disponible en Internet: <<https://doi.org/10.9790/0661-16195771>>.

ISO. Human-centred design for interactive systems. ISO 9241. [s.l.]: [s.n.], 2010.

LEFFINGWELL, Dean y BEHRENS, Pete. A User Story Primer. En: Leffingwell, LLC [en línea]. 2009. p. 1-16. [Consultado el 10, marzo, 2023]. Disponible en Internet: <<https://www.scaledagileframework.com/wp-content/uploads/2021/07/User-Story-Primer.pdf>>.

MARTIN, Robert. Clean Architecture: A Craftsman's Guide to Software Structure and Design. [s.l.]: Pearson Education, Limited, 2017. ISBN 9780134494326.

Memrise. Aplicativo web para aprender idiomas [en línea] [consultado el 20, septiembre, 2022] Disponible en Internet: <https://www.memrise.com/>.

MIYAMOTO, Yohsuke R., et al. Beyond Time-on-Task: The Relationship between Spaced Study and Certification in MOOCs. En: SSRN Electronic Journal [en línea]. 2015. [Consultado el 10, octubre, 2022]. Disponible en Internet: <<https://doi.org/10.2139/ssrn.2547>>.

NODEMON. nodemon. npm [página web]. [Consultado el 18, abril, 2023]. Disponible en Internet: <<https://www.npmjs.com/package/nodemon>>.

PIOTR WOZNIAK - Supermemo [página web]. [Consultado el 3, octubre, 2022]. Disponible en Internet: <https://supermemo.guru/wiki/Piotr_Wozniak>.

PIPES AND Filters pattern - Azure Architecture Center [Anónimo]. Microsoft Learn: Build skills that open doors in your career [página web]. [Consultado el 13, marzo, 2023]. Disponible en Internet: <<https://learn.microsoft.com/en-us/azure/architecture/patt>>.

REACT DEV. Reusing Logic with Custom Hooks – React. React [página web]. [Consultado el 17, abril, 2023]. Disponible en Internet: <<https://react.dev/learn/reusing-logic-with-custom-hooks>>.

REACT NATIVE DOCS. Documentation. [página web]. [Consultado el 14, marzo, 2023]. Disponible en Internet: <<https://reactnative.dev/>>.

RFC. RFC 7519: JSON Web Token (JWT). RFC Editor [página web]. [Consultado el 13, marzo, 2023]. Disponible en Internet: <<https://www.rfc-editor.org/rfc/rfc7519>>.

SALAS-PILCO, Sdenka Zobeida y YANG, Yuqin. Artificial intelligence applications in Latin American higher education: a systematic review. En: International Journal of Educational Technology in Higher Education [en línea]. 18, abril, 2022. vol. 19, no. 1 [c.

SEQUELIZE V6 | Sequelize [Anónimo]. Sequelize | Feature-rich ORM for modern TypeScript & JavaScript [página web]. [Consultado el 13, marzo, 2023]. Disponible en Internet: <<https://sequelize.org/docs/v6/>>.

SEQUELIZE. Seeders | Sequelize. Sequelize | Feature-rich ORM for modern TypeScript & JavaScript [página web]. [Consultado el 18, abril, 2023]. Disponible en Internet: <<https://sequelize.org/docs/v6/other-topics/migrations/#creating-the-first-seed>>.

TAN, Sema. Remote learning through a mobile application in gifted education. En: Gifted Education International [en línea]. 20, diciembre, 2021. vol. 38, no. 1 [consultado el 10, marzo, 2023], p. 95-114. Disponible en Internet: <<https://doi.org/10.1177/02>.

TRELLO | Atlassian. Atlassian [página web]. [Consultado el 10, marzo, 2023]. Disponible en Internet: <<https://www.atlassian.com/software/trello>>.

UNIVERSITY OF NEW BRUNSWICK. Understanding Quality Attributes. Faculty of Computer Science | UNB [página web]. [Consultado el 25, marzo, 2023]. Disponible en Internet: <<https://www.cs.unb.ca/~wdu/cs6075w10/sa2.htm>>.

USING EXPRESS middleware [Anónimo]. Express - Node.js web application framework [página web]. [Consultado el 13, marzo, 2023]. Disponible en Internet: <<https://expressjs.com/en/guide/using-middleware.html>>.

WHAT IS PostgreSQL? [Anónimo]. PostgreSQL Documentation [página web]. [Consultado el 13, marzo, 2023]. Disponible en Internet: <<https://www.postgresql.org/docs/current/intro-what-is.html>>.

11 ANEXOS

ANEXO 1: RECURSOS DEL PROYECTO.

Tipo de recurso	Descripción
Humano	Se tiene al desarrollador Juan Pablo Avilan y al colaborador Giovanni Fajardo.
Financiero	Se tienen destinados \$150.000 COP para los gastos en servicios en la nube (Servidor de aplicaciones, Servidor de base de datos, APIs) requeridos en la etapa de implementación, pruebas y operación.
Tecnológico	Se cuenta con un computador portátil para el desarrollo de documentación, implementación y pruebas.

ANEXO 2: CRONOGRAMA DE ACTIVIDADES DEL PROYECTO.

Objetivo	Actividades
Requerimientos	Requerimientos
Identificar los requerimientos del aplicación móvil teniendo en cuenta el modelo de la curva de olvido propuesto por Ebbinghaus y técnicas de estudio como la repetición espaciada y el recuerdo activo para que los usuarios puedan aprender de manera efectiva.	1. Investigar las técnicas de estudio de repetición espaciada, recuperación activa y el método de flashcards de Leitner para identificar historias de usuario.
	2. Investigar los motivadores de negocio para priorizar los requerimientos no funcionales.
	3. Identificar el perfil del usuario y segmentarlo por medio del buyer persona.
	4. Completar la lista de especificación de requerimientos.
Diseño	Diseño
Diseñar un modelo de arquitectura que se adecue a los requerimientos funcionales y no funcionales de tal manera que sean claros los componentes, relaciones, estructuras de control de flujo de datos y patrones de	1. Documentar las Historias de usuario
	2. Desarrollar vistas de arquitectura (Funcional, Despliegue, Desarrollo, Operacion)
	3. Desarrollar el modelo de clases
	4. Desarrollar el modelo de datos
Implementación	Implementación
Implementar la solución en código siguiendo el modelo de arquitectura y cumpliendo con todos los requerimientos levantados.	1. Desarrollar código de los módulos priorizando las capas de la arquitectura limpia en el backend.
	2. Desarrollar código de los módulos del frontend.
	3. Realizar pruebas unitarias.
	4. Documentar y hacer seguimiento de riesgos y errores de código
Pruebas	Pruebas
Validar el producto de software usando pruebas unitarias y de integración al producto para garantizar el correcto funcionamiento de cada requerimiento que se levantó y así asegurar la calidad.	1. Diseñar un ambiente de pruebas.
	2. Documentar y realizar pruebas de integración.
	3. Planificar y solucionar errores encontrados en el código.
	4. Realizar el manual técnico y el manual de usuario.

Actividades	Semana														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Requerimientos															
1. Investigar las técnicas de estudio de repetición espaciada, recuperación activa y el método de flashcards de Leitner para identificar historias de usuario.															
2. Investigar los motivadores de negocio para priorizar los requerimientos no funcionales.															
3. Identificar el perfil del usuario y segmentarlo por medio del buyer persona.															
4. Completar la lista de especificación de requerimientos.															
Diseño															
1. Documentar las Historias de usuario															
2. Desarrollar vistas de arquitectura (Funcional, Despliegue, Desarrollo, Operacion)															
3. Desarrollar el modelo de clases															
4. Desarrollar el modelo de datos															
Implementación															
1. Desarrollar código de los módulos priorizando las capas de la arquitectura limpia en el backend.															
2. Desarrollar código de los módulos del frontend.															
3. Realizar pruebas unitarias.															
4. Documentar y hacer seguimiento de riesgos y errores de código															
Pruebas															
1. Diseñar un ambiente de pruebas.															
2. Documentar y realizar pruebas de integración.															
3. Planificar y solucionar errores encontrados en el código.															
4. Realizar el manual técnico y el manual de usuario.															