



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Aplicación móvil (app) para
la asignatura de Álgebra
Lineal**

MATERIAL DIDÁCTICO

Que para obtener el título de
Ingeniero en Computación

P R E S E N T A

Cristian Rodolfo Gómez Mares

ASESORA DE MATERIAL DIDÁCTICO

Dra. Rosalba Rodríguez Chávez



Ciudad Universitaria, Cd. Mx., 2025



**PROTESTA UNIVERSITARIA DE INTEGRIDAD Y
HONESTIDAD ACADÉMICA Y PROFESIONAL
(Titulación con trabajo escrito)**



De conformidad con lo dispuesto en los artículos 87, fracción V, del Estatuto General, 68, primer párrafo, del Reglamento General de Estudios Universitarios y 26, fracción I, y 35 del Reglamento General de Exámenes, me comprometo en todo tiempo a honrar a la institución y a cumplir con los principios establecidos en el Código de Ética de la Universidad Nacional Autónoma de México, especialmente con los de integridad y honestidad académica.

De acuerdo con lo anterior, manifiesto que el trabajo escrito titulado APLICACION MOVIL (APP) PARA LA ASIGNATURA DE ALGEBRA LINEAL que presenté para obtener el título de INGENIERO EN COMPUTACIÓN es original, de mi autoría y lo realicé con el rigor metodológico exigido por mi Entidad Académica, citando las fuentes de ideas, textos, imágenes, gráficos u otro tipo de obras empleadas para su desarrollo.

En consecuencia, acepto que la falta de cumplimiento de las disposiciones reglamentarias y normativas de la Universidad, en particular las ya referidas en el Código de Ética, llevará a la nulidad de los actos de carácter académico administrativo del proceso de titulación.

CRISTIAN RODOLFO GOMEZ MARES
Número de cuenta: 301216544

ÍNDICE

Contenido

Introducción	1
Definición del problema.....	2
Contexto de la asignatura de Álgebra Lineal	4
Justificación.....	5
Objetivo.....	6
Marco Teórico	7
Álgebra Lineal en Ingeniería	7
Importancia del Álgebra Lineal en la Formación de Ingenieros	8
Procesos de Enseñanza y Aprendizaje de Álgebra Lineal	8
Tecnologías de la información y la Comunicación y Tecnología móvil en la educación.....	13
Tecnología móvil.....	13
Aprendizaje móvil para Matemáticas en Ingeniería.....	14
Estrategias de enseñanza y aprendizaje con el uso del aprendizaje móvil	15
Rol del docente y rol del estudiante en el uso de la Tecnología móvil	16
Ingeniería de software	18
Metodologías ágiles.....	19
Metodología Scrum.....	21
Implementación de la aplicación móvil de Álgebra Lineal utilizando Scrum	27
Pruebas	57
Resultados	57
Conclusiones	65
Referencias.....	66
Anexo	69
Manual de usuario	69
Código fuente	85

Lista de tablas

Tabla 1 Revisión y aprobación.....	28
Tabla 2 Historia de usuario.....	34
Tabla 3 Sprint Backlog	36
Tabla 4 Sprint Backlog. Tareas asociadas	41
Tabla 5 Sprint Backlog de las tareas.....	47
Tabla 6 Cuadro comparativo entre tecnologías	53

Lista de figuras

Figura 1 Capas de la ingeniería de software (Pressman, 2010)	18
Figura 2 Esquema Scrum, roles, eventos y artefactos.	26
Figura 3 Demostración asociada a la historia de usuario 1 en sprint review.	38
Figura 4 Pantalla principal con la lista de temas disponibles.....	39
Figura 5 Marcado de temas como completados y movimiento automático a una lista de terminados.....	40
Figura 6 Renderizado de contenido en las pestañas de gestión de temas.	43
Figura 7 Botón de navegación y ayuda respectivamente.....	44
Figura 8 Ejercicios marcados como terminados para mostrar en las pestañas respectivas.	45
Figura 9 Selección de ejercicio y pantalla con su respectivo enunciado.	46
Figura 10 Selección de ejercicio y pantalla con su respectivo enunciado y respuesta. ...	49
Figura 11 Opción para ver la solución sin importar si la respuesta es correcta o incorrecta.....	50
Figura 12 Navegación entre pestañas asegurando persistencia.	51
Figura 13 Te ayudado a cumplir el objetivo de aprendizaje de la asignatura.	58
Figura 14 Aspecto y sensación de la aplicación móvil.....	59
Figura 15 Facilidad del aprendizaje de la asignatura.....	59
Figura 16 Satisfacción con la velocidad de carga.....	60
Figura 17 Utilización de la aplicación móvil.	61
Figura 18 Recomendación para realizar otra aplicación para otra asignatura.....	61
Figura 19 Reforzamiento de los temas.....	62
Figura 20 Aprendizaje de la asignatura.....	63
Figura 21 Instalación de la aplicación.....	63
Figura 22 Ubicuidad y portabilidad de la aplicación.	64
Figura 23 Novedad para estudiar.	64
Figura 24 Ícono de la aplicación.	69
Figura 25 Pantalla de carga.....	70
Figura 26 Pantalla principal.	71
Figura 27 Encabezado de la pantalla principal.....	72

Figura 28	Cuerpo de la pantalla principal.....	73
Figura 29	Pie de la pantalla principal con secciones.....	74
Figura 30	Pantalla secundaria con los ejercicios del tema seleccionado.....	75
Figura 31	Encabezado de la pantalla de ejercicios.....	76
Figura 32	Cuerpo de la pantalla de ejercicios.....	77
Figura 33	Pie de la pantalla de ejercicios con las secciones respectivas.....	78
Figura 34	Pantalla de ejercicio con respuesta en incisos.....	79
Figura 35	Encabezado en la pantalla del ejercicio.....	80
Figura 36	Cuerpo en la pantalla del ejercicio.....	81
Figura 37	Pie en la pantalla del ejercicio.....	82
Figura 38	Mensajes cuando se selecciona la respuesta correcta e incorrecta en la pantalla del ejercicio.....	83
Figura 39	Visualizar la solución mediante el cuadro de diálogo de la respuesta incorrecta.....	84

Tabla de ilustraciones

Ilustración 1	Estructura del proyecto.....	85
Ilustración 2	Componentes utilizados.....	86
Ilustración 3	Componente Ask.....	87
Ilustración 4	Componente CardExercise.....	89
Ilustración 5	Componente CardHome.....	91
Ilustración 6	Componente CheckBoxAnswer.....	93
Ilustración 7	Componente ExerciseSolution.....	95
Ilustración 8	Componente Header.....	97
Ilustración 9	Componente NavHeader.....	99
Ilustración 10	Componente NavHeaderCopyRight.....	101
Ilustración 11	Componente NavHeaderQuestion.....	103
Ilustración 12	Componente NavHeaderSolution.....	105
Ilustración 13	Componente NavHeaderTheory.....	107
Ilustración 14	Componente TabBottomMenuAnswer.....	109
Ilustración 15	Componente TabBottomMenuExercise.....	111
Ilustración 16	Componente TabBottomMenuHome.....	113
Ilustración 17	Componente TabBottomMenuTheory.....	115
Ilustración 18	Componente TopicTheory.....	117
Ilustración 19	Componente Txt.....	119
Ilustración 20	Pages.....	120
Ilustración 21	Página CopyRight.....	120
Ilustración 22	PáginaExercise.....	123
Ilustración 23	Página Home.....	128
Ilustración 24	Página Question.....	133
Ilustración 25	Página Solution.....	142
Ilustración 26	Página Theory.....	145
Ilustración 27	Componente contenedor App.....	148

Glosario de términos

Sprint

Periodo de tiempo definido dentro de Scrum (generalmente de 1 a 4 semanas) en el que el equipo de desarrollo trabaja para completar un conjunto de tareas del Backlog. Cada *Sprint* inicia con una planificación (*Sprint Planning*) y termina con una revisión y retrospectiva.

Backlog

Lista priorizada de tareas, funcionalidades o mejoras pendientes en un proyecto. Se divide en:

- **Product Backlog:** Conjunto de requisitos y características del producto a desarrollar.
- **Sprint Backlog:** Selección de elementos del *Product Backlog* que se trabajarán en un *Sprint*.

Daily (Daily Scrum)

Reunión breve (máximo 15 minutos) que se realiza diariamente en Scrum para que el equipo de desarrollo sincronice el trabajo, comparta avances, identifique obstáculos y planifique las siguientes acciones.

Increment

Resultado de un *Sprint* en *Scrum*, que representa una versión mejorada del producto con nuevas funcionalidades completadas y potencialmente utilizables. Cada Increment debe cumplir con la *Definition of Done* (Definición de Hecho).

Artefacto (Artefactos de Scrum)

Elementos clave que proporcionan visibilidad y transparencia en *Scrum*. Los tres principios artefactos son:

1. **Product Backlog:** Lista priorizada de funcionalidades pendientes del producto.
2. **Sprint Backlog:** Tareas seleccionadas para un *Sprint*.

3. **Increment** : Producto entregable al final del *Sprint*.

Tabs

Pestañas de navegación en una interfaz gráfica que permiten cambiar entre diferentes secciones sin recargar la página. Comúnmente utilizadas en aplicaciones web y móviles para mejorar la experiencia del usuario.

Agradecimientos

A mis padres, María Elena y Rodolfo, por ser mi mayor ejemplo de esfuerzo, amor y dedicación. Gracias por su apoyo incondicional en cada etapa de mi vida, por enseñarme el valor del trabajo y la perseverancia, y por estar siempre a mi lado con palabras de aliento y confianza en mis capacidades. Sin su amor y sacrificio, este logro no habría sido posible.

A mis hermanas, Wendy y Jessica, por su amor, apoyo y compañía incondicional. Gracias por ser mi refugio en los momentos difíciles, por alentarme a seguir adelante y por recordarme siempre la importancia de la familia. Su cariño y confianza en mí han sido una fuente inagotable de motivación. Este logro es también para ustedes.

A mi esposa, Nayeli, por su amor, paciencia y apoyo inquebrantable. Gracias por ser mi compañera en este camino, por motivarme en los momentos difíciles y por creer en mí siempre. Tu presencia ha sido mi mayor fortaleza.

A mi hijo, Liam, quien ha sido mi mayor inspiración. Cada día me enseñas el verdadero significado del amor y la perseverancia. Este logro es también para ti, con la esperanza de que un día sigas tus sueños con la misma pasión.

A mis profesores, y en especial a la Dra. Rosalba Rodríguez Chávez, cuya dedicación, paciencia y conocimiento han sido fundamentales en mi formación. Gracias por compartir su sabiduría, por motivarme a dar siempre lo mejor de mí y por guiarme en este camino académico. Sus enseñanzas han dejado una huella imborrable en mi vida y en este trabajo, que es también reflejo de su esfuerzo y compromiso con la educación.

A los integrantes del jurado sinodal por el tiempo, la dedicación y el compromiso con los que revisaron y evaluaron este trabajo.

Su experiencia, observaciones y comentarios fueron fundamentales para enriquecer el contenido y elevar la calidad del presente proyecto.

Reconocimiento

Este informe se realizó gracias al apoyo de la Dirección General de Asuntos del Personal Académico (DGAPA) de la Universidad Nacional Autónoma de México (UNAM) a través del Programa de Apoyo a Proyectos de Innovación y Mejoramiento de la enseñanza (PAPIME) por medio del proyecto PE101424 titulado “Secuencias de didácticas para vincular el contenido temático de Álgebra Lineal con problemas de Ingeniería Aplicada” y el proyecto PE107625 titulado “Evaluación del contenido de las Secuencias didácticas que vinculan el Álgebra Lineal con temas de Ingeniería aplicada con estudiantes de la Facultad de Ingeniería, UNAM”.

Gracias por el apoyo para alojar en la página dicha aplicación móvil.

Introducción

La ciudadanía global requiere que se tengan ingenieros capacitados en conocimientos, habilidades y actitudes para la optimización de la resolución de problemas, así como en la toma de decisiones.

En la Facultad de Ingeniería de la UNAM, se promueven estas capacidades a través del uso de Tecnologías de la Información y la Comunicación (TIC) usando diversos recursos didácticos que apoyan los procesos de enseñanza y aprendizaje de sus diversas asignaturas, y así abatir el rezago estudiantil.

Según la UNESCO (UNESCO, Tecnologías de la información y la comunicación (TICs) en la educación, 2023), el uso de las TIC permite la interactividad entre la enseñanza y el aprendizaje a través de estrategias como pizarras digitales, los smartphones de los estudiantes, empleo de aula invertida, entre otros. El docente que tiene competencias digitales apoya a que los estudiantes tengan habilidades cognitivas superiores, tenga creatividad de forma individual entre otras. Esto permite que los estudiantes estén más preparados para enfrentar su economía nacional y global.

Por otro lado, hay tecnologías que apoyan los procesos de enseñanza y aprendizaje como es la Tecnología móvil, en la Facultad de Ingeniería, UNAM, la mayoría de los estudiantes cuentan con una computadora portátil o un dispositivo móvil que permiten el acceso inmediato a Internet e interactuar con diversos recursos didácticos para el aprendizaje de los contenidos de una asignatura. Además, que les permiten descargar aplicaciones móviles a bajo costo o gratuitas.

Por otro lado, la asignatura de Álgebra Lineal, que se imparte en segundo semestre de la Facultad de Ingeniería, es una asignatura que se concibe por parte de los estudiantes como una asignatura abstracta, que requiere que se comprendan los conceptos y así resolver problemas algebraicos y de ingeniería aplicada. Por lo anterior, en este trabajo se realizaron diversos ejercicios de cada tema con la finalidad de que el alumno los realice y obtenga una realimentación inmediata. Además, que cuenta con el apoyo de los conceptos y teoremas que sustentan sus ejercicios.

Con esto se espera contribuir a que la Facultad de Ingeniería siga siendo líder en el campo educativo en la formación científica y tecnológica de recursos humanos con herramientas didácticas de vanguardia y de alta calidad.

Definición del problema

La Facultad de Ingeniería de la UNAM, requiere que sus estudiantes tengan una formación científica, tecnológica y humanista con la finalidad de llegar a la resolución de problemas de ingeniería, fomentando por un lado el pensamiento crítico y creativo y por otro la toma de decisiones.

En el Plan de trabajo de Facultad de Ingeniería 2023-2027 se señala que una de las problemáticas y necesidades prioritarias es que se requiere retomar y modernizar junto con los profesores son los materiales didácticos digitales que se realizaron durante la pandemia, ya que actualmente la modalidad de trabajo es presencial y no se pueden olvidar los recursos digitales, es por ello que se necesitan revisar las tendencias actuales para que los estudiantes se sientan cercanos a su realidad de ejercicio profesional con herramientas novedosas en ambientes amigables, accesibles y atractivos. (Plan de Desarrollo 2023-2027, 2023)

Es por lo que, en la División de Ciencias Básicas de la Facultad de Ingeniería, se promueve la realización de diversos recursos y materiales digitales con la finalidad de abatir el índice de reprobación por parte del estudiantado, el cual es aproximadamente del 40% en las asignaturas de Matemáticas. Así lo reporta en los indicadores de reprobación de cursos académicos más altos en los programas de Ingenierías de la USTA-Bucaramanga en Colombia también se tiene el 40% de reprobación en la asignatura de Álgebra Lineal. (Zabala Vargas, 2022)

Aunado a que en las diversas asignaturas de Matemáticas se perciben como abstractas y que se requieren del apoyo de diversos recursos y estrategias de aprendizaje, como en el caso de la asignatura de Álgebra Lineal. En la cual, los estudiantes refieren que es una asignatura abstracta, en la que hay que aprenderse diversos conceptos, habilidades algebraicas y teoremas que refuercen lo visto en clase, así como también

ejercicios basados en problemas de aplicación en la vida ingenieril. Algunos de los estudiantes carecen de formación matemática de antecedentes de Álgebra o Cálculo y Geometría Analítica.

Asimismo, algunos de los estudiantes refieren que no perciben la conexión entre la teoría y la práctica de Ingeniería, pues no hay ejemplos de aplicaciones a problemas de ingeniería pues muchas veces el docente se centra en el trabajo de los conceptos y resolución de ejercicios algebraicos.

Por otro lado, existen dentro del aula estudiantes con distintos niveles de aprendizaje, algunos de los estudiantes se sienten muy desafiados al no comprender rápidamente los contenidos de la asignatura dificultando su aprendizaje. Además, el profesorado tiene gran número de estudiantes en sus clases presenciales de la asignatura de Álgebra Lineal por lo que no puede dar realimentación personalizada a cada estudiante. Muchas veces, las metodologías de enseñanza utilizadas en la educación superior son tradicionales y no están adaptadas a las necesidades de los nuevos estudiantes. Esto incluye la falta de utilización de tecnologías interactivas o de casos prácticos que faciliten la comprensión.

Según la Coordinación de Evaluación de la Facultad de Ingeniería, en la encuesta realizada a los profesores en el semestre 2019-2 en una muestra de 3192 profesores, el 64.1% de los profesores consideran que al inicio del curso sus estudiantes no traen buenos conocimientos. (Coordinación de Evaluación Educativa, Facultad de Ingeniería, UNAM, 2019). Aunado a esto, la gran mayoría de la población de estudiantes de la asignatura tiene pocas posibilidades de estudiar en los libros de texto, ya que tienen que trasladarse en transporte público, dificultando muchas veces el tener que transportar varios libros para estudiar.

Actualmente la tecnología móvil es de gran ayuda, ya que facilita que el estudiante pueda estudiar en cualquier lugar y en cualquier hora rompiendo así la barrera espaciotemporal. Además, que los estudiantes pueden estudiar a través del uso de su dispositivo móvil (tableta o celular) y revisar información con diversos recursos didácticos como aplicaciones móviles, documentos, videos, entre otros.

Por lo anterior, es importante evaluar la práctica educativa del uso de la tecnología educativa para la mejora de la enseñanza del álgebra lineal y así impactar el desempeño de los estudiantes en su formación científica, tecnológica y humanista para incorporarse posteriormente a la práctica profesional.

Contexto de la asignatura de Álgebra Lineal

El plan de estudios de las 15 carreras de la Facultad de Ingeniería, UNAM contempla a la asignatura de Álgebra Lineal como básica, debido a que es necesaria para la resolución de problemas algebraicos y de Ingeniería.

Álgebra Lineal se encuentra ubicada en el segundo semestre de las carreras de Ingeniería y es antecedente de asignaturas como Cálculo Vectorial y Ecuaciones Diferenciales, entre otras.

El objetivo de la asignatura es: “El alumno analizará los conceptos básicos del Álgebra Lineal, ejemplificándolos mediante sistemas ya conocidos, haciendo énfasis en el carácter general de los resultados, a efecto de que adquiera elementos que le permitan fundamentar diversos métodos empleados en la resolución de problemas de ingeniería”. (Programa de Álgebra Lineal, 2024)

El temario contiene los temas de: Grupos y campos, Espacios Vectoriales, Transformaciones Lineales, Espacios con producto interno y Operadores en espacios con producto interno.

La asignatura cuenta con 8 créditos para estudiarla y se imparte en modalidad presencial.

Por lo que, desde los primeros semestres en la Facultad de Ingeniería, se promueve la formación científica del Ingeniero, y la comprensión de los conceptos para poder resolver problemas algebraicos y de Ingeniería.

Justificación

La Facultad de Ingeniería de la UNAM tiene como uno de sus propósitos la formación científica, tecnológica y humanista de los Ingenieros del país. Por lo que los estudiantes requieren tener una formación integral.

Las matemáticas, la física y la química son base fundamental en la formación de ingenieros y para la comprensión de los fenómenos del mundo que nos rodea.

Actualmente las Tecnologías de la Información y la Comunicación son herramientas digitales que apoyan en gran medida a que los estudiantes puedan estudiar o reforzar de forma autogestiva o colaborativa.

Es por ello que las asignaturas de Ciencias Básicas de dicha Facultad requieren apoyarse con recursos digitales o físicos a modo que el estudiante pueda reforzar lo que va aprendiendo en sus cursos presenciales, evitando la reprobación y el rezago.

Es de fundamental importancia mencionar que dentro del contexto estudiantil se toma en cuenta la encuesta sociodemográfica de la generación 2020, la cual menciona que los estudiantes de la Facultad de Ingeniería cuentan con teléfono celular (97.8%), computadora en su vivienda el (89.3%) y con internet (93.4%). (Secretaría de Apoyo a la Docencia, 2020). Esto evidencia que la Tecnología móvil es una importante área de oportunidad para apoyar los procesos de enseñanza y aprendizaje de Álgebra Lineal.

Según (Zamora Delgado, 2020), las prácticas de enseñanza dentro y fuera del aula permiten una mejora en la educación.

La educación superior requiere que los estudiantes tengan acceso a recursos en cualquier momento y lugar. Una aplicación móvil permite a los estudiantes revisar contenidos, conceptos y ejercicios de Álgebra Lineal a su conveniencia, facilitando el aprendizaje autónomo. Se proyecta que para el año 2029 más de 6.4 mil millones de personas usarán teléfonos inteligentes, lo que brinda una idea de la importancia de utilizar dichos dispositivos para apoyar la enseñanza y el aprendizaje. (Statista, 2024)

El uso de los dispositivos móviles proporciona un refuerzo en los procesos de enseñanza y aprendizaje, haciendo énfasis en la ubicuidad, facilitando estudiar en cualquier tiempo y espacio, así como hace uso de las características de portabilidad, interactividad, conectividad, entre otras. Además, hace uso de estrategias

metodológicas a partir del aprendizaje móvil, facilitando el aprendizaje autónomo para lograr aprendizajes significativos en los estudiantes.

En la Coordinación de Matemáticas se cuenta con una aplicación móvil de Álgebra donde los estudiantes pueden aprender los conceptos y realizar algunos ejercicios, los estudiantes han mencionado que dicha aplicación les ayuda reforzar sus conceptos y sus habilidades y que les gustaría que más asignaturas de Matemáticas tuvieran algo similar.

Por lo anterior, la implementación de una aplicación móvil centrada en los contenidos de Álgebra Lineal podría no solo mejorar los procesos de enseñanza y aprendizaje, sino que también se adhiere a las tendencias educativas y a las necesidades actuales de los estudiantes de Ingeniería. Además, las aplicaciones móviles son herramientas para el autoconocimiento del progreso del estudiante, permitiendo a los usuarios realizar un seguimiento de su desempeño a través de cuestionarios y ejercicios interactivos. Esto se alinea con la teoría del feedback constructivo, que enfatiza la importancia de recibir realimentación para mejorar el aprendizaje (Hattie & Timperley, 2007). No sólo eso, sino que el uso de esta tecnología permite que se puedan actualizar los contenidos de la asignatura.

Por otro lado, la Facultad contará con diversos recursos didácticos para la enseñanza de la asignatura, favoreciendo que el profesor pueda apoyarse para reforzar los contenidos y las habilidades de la asignatura de Álgebra Lineal. Actualmente las aplicaciones móviles ofrecen portabilidad, interactividad, usabilidad, entre otros. Los estudiantes se pueden beneficiar para estudiar en donde sea a través de su dispositivo móvil y así reforzar la asignatura.

Objetivo

Diseñar e implementar una aplicación móvil para la asignatura de Álgebra Lineal, con teoría y ejercicios interactivos resueltos, que permita a los estudiantes reforzar y complementar los estudios de la asignatura.

Marco Teórico

A continuación, se abundará de forma sucinta la importancia y el proceso de enseñanza y aprendizaje del Álgebra Lineal en Ingeniería, así como el papel de la tecnología móvil en el desarrollo del aprendizaje móvil.

Álgebra Lineal en Ingeniería

El álgebra lineal es una rama fundamental de las matemáticas que se aplica en diversas disciplinas de la ingeniería y la ciencia. En el contexto de la educación superior y la ingeniería, este tema se vuelve crucial, ya que proporciona herramientas esenciales con el estudio de conceptos como espacios vectoriales y transformaciones lineales para resolver problemas algebraicos y que son aplicados a diversas áreas de la ingeniería. Como señala (Velázquez Torres, 2003) “El ingeniero actual debe contemplar la nueva matemática, convencerse de que el Álgebra Lineal, hoy por hoy es una herramienta indispensable en su desarrollo profesional. Un ingeniero en potencia será aquel que logre acompañar la nueva matemática con las tecnologías actuales y la física moderna”. El marco histórico ha cobrado relevancia desde el estudio de las ecuaciones lineales según (Gutiérrez González & Ochoa García, 2014), quienes mencionan que un enfoque del Álgebra Lineal está basado en la notación matricial que se remonta con el término matriz el cual fue acuñado por el matemático James Joseph Sylvester en 1850, quien la definió como un arreglo de términos. Posteriormente Arthur Cayley en 1858 mencionó que los sistemas de ‘m’ ecuaciones con ‘n’ incógnitas lineales se pueden abreviar como una matriz. Además, se definieron las operaciones de adición y multiplicación por un escalar, así como una matriz no singular.

Por otro lado, William Rowan Hamilton en 1843 propuso elementos llamados cuaterniones, al ser complicado su estudio sentaron las bases para explorar el concepto de vector desde el punto de vista matemático, por lo que, junto con Hamilton, Cayley y Hermann Günther Grassmann desarrollaron las ideas de vector y espacio vectorial sentando las bases para el estudio de las propiedades de los vectores y las matrices. En el siglo XX, el álgebra lineal se consolidó todavía más con la formalización de las estructuras matemáticas por parte de matemáticos como David Hilbert y John von

Neumann, quienes aplicaron estos conceptos en diversas áreas de la ciencia y la ingeniería. (Academia Lab, 2024)

Importancia del Álgebra Lineal en la Formación de Ingenieros

El álgebra lineal es fundamental en la formación de ingenieros, su impacto histórico y su adhesión con diversas ramas de la ciencia y la tecnología hacen que sea una asignatura con teoría y práctica que la destacan como una herramienta indispensable en la educación superior.

Los estudiantes de Ingeniería que estudian Álgebra Lineal están preparados con dicha herramienta la cual les permite tener diversas competencias técnicas que los prepara para enfrentar sus asignaturas posteriores como Cálculo Vectorial, Ecuaciones Diferenciales, Probabilidad y Estadística, entre otras. Así como enfrentar los retos del mundo laboral.

Según (Velázquez Torres, 2003) menciona que la teoría matemática de las generalizaciones y nuevos métodos de análisis es importante en varios campos de la investigación y la industria, en geofísica se utiliza para pronósticos del tiempo, en investigación de materiales hay reómetros que estudian las geometrías de los materiales, en robótica se manejan grados de libertad en el diseño de prototipos, en aeroespacial en los sistemas de control de un transbordador, en Ciencias de la tierra, el estudio de la cristalografía, entre otros.

Como se ha mencionado anteriormente, la simulación, el modelado de sistemas físicos complejos, manipular gran cantidad de datos, representación de sistemas dinámicos para el estudio de su comportamiento y estabilidad a través de los espacios vectoriales, diseño asistido por computadora, entre otros hacen uso del Álgebra Lineal.

Procesos de Enseñanza y Aprendizaje de Álgebra Lineal

La enseñanza del álgebra lineal en la asignatura de Álgebra Lineal que se le imparte al estudiantado es un marco teórico con el desarrollo de ejercicios algebraicos, además que se le mencionan algunas aplicaciones de Ingeniería aplicada.

Algunas estrategias que se sugieren para su adecuado estudio, es el utilizar diversas estrategias de aprendizaje como las metacognitivas, donde el estudiante pueda aprenderse los distintos teoremas a través de la realización de mapas mentales o conceptuales.

Se utilizan también estrategias de aprendizaje colaborativo como el aprendizaje basado en problemas para que los estudiantes puedan investigar dónde se pueda aplicar el Álgebra en Ingeniería.

También se hace uso de Nuevas Tecnologías de la Información y la Comunicación para realizar diversas simulaciones. Se utiliza software como Geogebra, Mathematica, Matlab, Python, entre otros.

Por lo anterior, tanto la teoría como la práctica que se enseña en la Facultad de Ingeniería permiten al estudiante resolver diversos problemas algebraicos y complejos, modelarlos y analizarlos, para posteriormente aplicarlos a problemas de ingeniería avanzada.

Como menciona en (Strang, 2003), las personas necesitan más álgebra lineal que cálculo. Al trabajar con líneas curvas y superficies, el primer paso es siempre linealizar. Si se reemplaza la curva por su línea tangente, ajustar la superficie por un plano y el problema se vuelve lineal, entre otros ejemplos.

Sin embargo, es deseable también identificar las características del docente y del estudiante de Matemáticas.

Como mencionan (Loachamín Iza, Vargas Chavarrea, Andrade Villarreal , & Puente Ponce, 2023) quienes citan a (Plaza Gálvez, 2017), señalan que “el docente de matemáticas en programas de ingeniería debe promover determinadas habilidades entre sus alumnos; algunas de crucial importancia son las siguientes: 1. Pensamiento holístico, investigación crítica, análisis y reflexión. 2. Aprendizaje activo y aplicación práctica. 3. Autoconsciencia y empatía y 4. Comunicación y una fuerte escucha”. Así como poseer habilidades de pensamiento crítico y reflexivo. Es deseable que el docente tenga conocimiento de otras disciplinas (pedagógicos, comunicacionales, motivacionales, humanistas, entre otros). Es por lo que el docente debe estar

actualizándose constantemente en diversas áreas del conocimiento. El docente sólo es un facilitador de los aprendizajes.

También, (Villar-Sánchez, Arancibia-Carvajal, Robotham, & González, 2021) quienes citan a (Etuk et al., 2013) que la actitud de los estudiantes hacia el aprendizaje de las matemáticas se ve influido por los conocimientos del profesor con respecto a la asignatura, las habilidades afectivas y de comunicación. También mencionan a (Marchis, 2011) quienes mencionan que las habilidades de afecto son el respeto, el interés del profesor y las creencias sobre las competencias de sus estudiantes tienen más importancia para los estudiantes.

Por otro lado, el estudiante en su formación científica y tecnológica de ingeniero al obtener la teoría la debe vincular con la resolución de problemas algebraicos y de Ingeniería, permitiéndole tomar decisiones, desarrollar su pensamiento analítico, creativo, crítico y reflexivo, además de actuar con humanismo, responsabilidad y ética. En general, se requiere que para el aprendizaje de las matemáticas tenga una actitud de motivación, curiosidad, buenos antecedentes del bachillerato para no tener tropiezos en el estudio, capacidad de investigación para vincular la teoría con la práctica, entre otros.

En general un estudiante que cursa las carreras STEAM (Ciencia, Tecnología, Ingeniería, Arte y Matemáticas) necesita estar motivado para aprender de forma autónoma una vez que se le ha facilitado la teoría en la clase, el estudiante puede aprender de forma autogestiva.

Los perfiles de los egresados de la Facultad de Ingeniería, UNAM, dentro de su perfil de egreso hay varios puntos a destacar como la comprensión de la relación entre la teoría y la práctica, selección de un diseño en particular, utilizar las herramientas y técnicas adecuadas para dar solución a un problema, identificar aplicaciones en su carrera, entre otras. (Facultad de Ingeniería, 2016)

Por otro lado, no hay que dejar de lado la evaluación del aprendizaje de los estudiantes.

Evaluación del aprendizaje

Según (Sánchez Mendiola & Martínez González, 2022) menciona a Estándares para Pruebas Educativas y Psicológicas de AERA-APA-NCME quienes definen la evaluación como: “método sistemático de obtención de información, usado para formular deducciones sobre las características de personas, objetos o programas; proceso sistemático para medir o evaluar las características o el desempeño de individuos, programas u otras entidades con la finalidad de hacer inferencias; en ocasiones se usa como sinónimo de prueba”.

La evaluación en el aprendizaje es de fundamental importancia para medir y mejorar el rendimiento de los estudiantes en su aprendizaje, así como mediar la efectividad de los métodos, estrategias de enseñanza y los recursos educativos, entre otros. A continuación, se presentan tres conceptos clave relacionados con la evaluación educativa.

La evaluación según (Espinoza, 2022) es un proceso sistemático, continuo, duradero y complejo para obtener información y así poder identificar en detalle el alcance del objetivo, es un proceso activo y práctico. Se hace uso de métricas para garantizar datos efectivos y confiables. Además, es la base para la toma de decisiones en cuanto a mejorar las actividades que se están desempeñando en el ámbito educativo. (Espinoza, 2022) cita a (Fernández, 2017) quien menciona que *“la evaluación se asume como una valoración y regulación del proceso de aprendizaje, que posibilita el progreso y alcance de las metas”*. (p. 3).

Según (Sánchez Mendiola & Martínez González, 2022) y (Red Educa, 2023) existen tres tipos de evaluación:

- 1) Evaluación diagnóstica es aquella que se realiza al inicio del proceso educativo, puede ser en un curso u otra actividad académica con el propósito de identificar los conocimientos previos que tienen los estudiantes, dicha información permite que el docente pueda adaptar su curso o estrategias de enseñanza y de aprendizaje.
- 2) La evaluación formativa es aquella que permite monitorear los avances de aprendizaje que van obteniendo los estudiantes durante el curso. Esta evaluación

ayuda a que los estudiantes puedan reforzar los aprendizajes y el docente pueda realimentar lo que no ha quedado claro.

- 3) La evaluación sumativa es aquella que sirve para medir los avances logrados al final del curso, con base en los objetivos propuestos en el programa del curso o actividad por parte de los estudiantes.

La evaluación del aprendizaje de los estudiantes es muy importante en un sistema educativo para medir el rendimiento académico y se aprovecha para mejorar la enseñanza y aprendizaje, contribuyendo al desarrollo de los conocimientos, habilidades y actitudes de los estudiantes.

Hay diversos instrumentos de evaluación, por lo que se describirán de forma sucinta algunos de ellos:

Según (Sánchez Mendiola & Martínez González, 2022) mencionan las siguientes clasificaciones:

- 1) Evaluación escrita se encuentran los ensayos, reportes, exámenes objetivos, respuesta corta, relación de columnas, entre otras.
- 2) Evaluación práctica se encuentran los exámenes orales, exámenes prácticos con casos, resolución de problemas, entre otros.
- 3) Observación por ejemplo: Listas de cotejo, rúbricas, entre otros.
- 4) Portafolios, por ejemplo, registro de procedimiento, portafolio, diario, entre otros.
- 5) Autoevaluación, por ejemplo: reportes del estudiante o de compañeros.

En este material se utilizó el instrumento de evaluación escrita con exámenes de opción múltiple para evaluar los aprendizajes de los contenidos y sus habilidades, además que se le presenta al estudiante una realimentación inmediata de la resolución del ejercicio. Un instrumento de evaluación tiene una pregunta, varias opciones de respuesta dónde una es la correcta. Sus características es la objetividad ya que existe una sola respuesta correcta, entre otras.

Según (Soubirón & Camarano , 2006), las pruebas objetivas, tienen una estructura clara y precisa y los estudiantes dan respuestas precisas, hay menor subjetividad para evaluar las respuestas por parte del evaluador, su calificación es rápida. Los distractores se realizan con base en los errores frecuentes que comete el estudiante.

Tecnologías de la información y la Comunicación y Tecnología móvil en la educación

Actualmente las Tecnologías de la Información y la Comunicación han cobrado relevancia en la educación superior debido a que los estudiantes las utilizan frecuentemente para realizar sus tareas, búsqueda de información, realizar gráficas, socializar con sus pares, entre otras.

Las tecnologías de la información y la comunicación (TIC) según Cruz barragán y Barragán López, (2014) citado en (Álvarez Álvarez & Jiménez Ruiz, 2022) indican que “Las TIC en el aula proporcionan, tanto al educador como al alumno, herramientas útiles y posicionan al estudiante como el protagonista de su propio aprendizaje” (p. 51).

Según la UNESCO (UNESCO, Technology in education: A TOOL ON WHOSE TERMS?, 2023), en la actualidad las TIC y la Tecnología móvil son más utilizadas en educación superior. La tecnología digital ofrece aumentar las oportunidades para practicar y disponibilidad de tiempo, así como personalizar la impartición, así como la búsqueda para motivar la atención de los estudiantes en cuanto a la forma de presentar el contenido, la interactividad, la colaboración, entre otros. Por otro lado, el aprendizaje digital tiene oportunidades en el campo educativo para acceder a la educación, además, de mejorar los resultados del aprendizaje para afrontar los retos del siglo XXI. Promueve la inclusión y la preparación de los usuarios para afrontar los retos del siglo XXI. (UNESCO, Día internacional del aprendizaje digital, 2024)

Tecnología móvil

La Tecnología móvil actualmente incursiona en distintos ámbitos sociales siendo actualmente utilizadas las diversas características que proporcionan los distintos dispositivos móviles como son: Se puede aprender desde cualquier lugar y a cualquier hora (ubicuidad), son portátiles, se puede tener acceso a internet, muchos incluyen la disponibilidad de multimedia. Se puede acceder a numerosas aplicaciones móviles para

apoyar el aprendizaje, entre otras. Se puede aprender de forma individual o colaborativa. (Santos Dimaté, 2023)

Según (Álvarez Álvarez & Jiménez Ruiz, 2022) mencionan que en el mundo hay más de 3200 millones de usuarios con teléfonos móviles, siendo la modalidad de Tecnología de la Información y la Comunicación más utilizada, se indica que en los países en desarrollo 4 de cada 5 personas utilizan el teléfono móvil.

Álvarez et. al. citan al informe regional titulado “la economía móvil en América Latina 2021” de la GSMA, que menciona que la red 4G continuará siendo la base de la industria móvil en la región a corto plazo, y representará casi un 73% de las conexiones totales para finales de 2025 (GSMA, 2020).

Las directrices principales de la Tecnología móvil en el campo educativo son:

Según (Shuler, Winters, & West, 2013) mencionan que los docentes se deben capacitar para incorporar en su práctica docente para fomentar el aprendizaje. Es necesario tener un repositorio de aplicaciones móviles para ser utilizados en las clases, la creación de contenidos pedagógicos es importante, sin embargo, se debe pensar en las limitantes de idioma y para personas con capacidades diferentes. En los distintos recursos para dispositivos móviles se debe facilitar la igualdad de género en el aprendizaje móvil. Promover la equidad para que haya mayores oportunidades en la accesibilidad, se requiere que los usuarios se asuman el gasto y plan de conexión de su dispositivo móvil para su proceso de aprendizaje, los docentes pueden aprovechar el uso de la tecnología móvil para mejorar su práctica educativa. Los dispositivos móviles se utilizan para apoyar el aprendizaje.

[Aprendizaje móvil para Matemáticas en Ingeniería](#)

El aprendizaje móvil apareció en la década de los 90 en Asia y Europa para apoyar a la educación, y se percibe como metodología en la enseñanza y el aprendizaje apoyadas de dispositivos móviles.

(Álvarez Álvarez & Jiménez Ruiz, 2022) mencionan que la inclusión de la tecnología móvil denota nuevos escenarios para la educación. Actualmente se tienen en Latinoamérica los dispositivos móviles al alcance de un gran número de personas. Las

autoras citan a (Ausin, et al, 2016) quienes mencionan que para el uso adecuado de las tecnologías móviles en la educación se debe capacitar al docente, para identificar las necesidades del estudiante, el diseño instrucción pertinente para obtener autoaprendizaje.

Según (Aguas-Díaz, Flores-Flores, Sarmiento-Sarmiento, & Aguirre-Robalino, 2020) mencionan a (Chirino-García & Hernández-Corona, 2020) quienes observan que el aprendizaje móvil se puede ejecutar a través de dispositivos móviles para diseñar aplicaciones concretas a necesidades académicas entre otras, para “formar profesionales proactivos, independientes, líderes en capacidad de asumir retos desde el efectivo manejo emocional”. (p.871).

Según (Santos Dimaté, 2023) los beneficios son:

- Accesibilidad. Estudiar un material desde cualquier lugar y momento.
- Flexibilidad. Los estudiantes pueden avanzar a su propio ritmo y adaptarse a cualquier horario. Incrementando la motivación y el compromiso.
- Interacción. Los estudiantes y los docentes pueden interactuar. Se incrementa la participación. El aprendizaje puede ser interactivo al incluir multimedia, gamificación, entre otras.
- Personalización. Los estudiantes utilizan la tecnología móvil según sus intereses y motivaciones.
- Costo-efectividad. Los dispositivos móviles son cada vez más accesibles en costo, cada vez hay más información al alcance de los ciudadanos.

Estrategias de enseñanza y aprendizaje con el uso del aprendizaje móvil

El aprendizaje móvil o M-Learning es una estrategia que hace uso de dispositivos móviles en los procesos de enseñanza y aprendizaje. Según (Zamora, 2020) quien retoma a Cantú (2016), el aprendizaje móvil mejora los procesos de enseñanza y aprendizaje, cuando el docente se apoya en el ambiente de aprendizaje por dispositivos móviles apoyados en estrategias de aprendizaje que generen y dirijan a los estudiantes en el desarrollo de sus habilidades y capacidades. (p. 87).

Según (Reyes Chávez & Medina Díaz, 2015) el aprendizaje móvil permite el logro de conocimientos y el desarrollo de habilidades digitales (aprendizaje) con el uso de un

dispositivo móvil en diferentes ambientes de aprendizaje. Reyes et.al. (2015) retoman a Espinosa (2010) quien menciona el entorno de aprendizaje aunado con estrategias pedagógicas da una oportunidad al aprendizaje con base en el acceso y la flexibilidad. Las estrategias de aprendizaje que se sugieren son de aprendizaje autónomo y aprendizaje colaborativo.

Según (Mayer, 2009) al promover el aprendizaje activo mejora la comprensión y retención de conceptos complejos, como los que se encuentran en Álgebra Lineal.

Rol del docente y rol del estudiante en el uso de la Tecnología móvil

Según (García Contador & Gutiérrez-Esteban, 2020) mencionan a (Dorfsman, 2012) quien indica que el rol del docente en el uso de Tecnologías de la Información y la Comunicación es más activo. También cita a Cabero (2007), quien indica que el docente deja de ser un transmisor de la información y se convierte en un facilitador del aprendizaje y de la enseñanza, en diseñador de situaciones de aprendizaje, tutor virtual, evaluador continuo y orientador, entre otros. También citan a Viñals y Cuenca (2016) quienes dicen que la forma de aprender ha cambiado por lo que el docente debe adaptarse a las nuevas metodologías de enseñanza y sus funciones son el de “el de organizador, coacher, gestor del aprendizaje, orientador, facilitador, tutor, dinamizador y asesor”.

Por otro lado, también mencionan que en un estudio realizado a una muestra de 151 docentes de distintos niveles educativos que los docentes utilizan redes sociales a diario como WhatsApp, el 81.46%, Facebook el 35.10%, con base en lo anterior, se percibe que utilizan sus dispositivos móviles para utilizar dichas redes sociales. Además, la mayoría de los docentes considera que la integración de las Tecnologías de la Información y la Comunicación puede haber un cambio en la calidad de la enseñanza. Según (UNESCO, Preguntas y respuestas: Por qué es esencial la educación digital para la ciudadanía mundial, 2024) indica que los docentes que incursionan en el ámbito digital tienen características de potenciar la comunicación, la creatividad y la innovación con la finalidad de preparar a los ciudadanos globales hacia su participación en el mundo físico y digital.

No obstante, el estudiante puede desarrollar su aprendizaje mediante el uso de tecnologías digitales de manera autogestiva y activa, buscando información que le permita ampliar los conocimientos o habilidades previamente orientados por el docente. Estudios recientes evidencian que el uso de aplicaciones móviles en la enseñanza del Álgebra Lineal genera mejoras significativas en el rendimiento académico, la comprensión conceptual y la actitud de los estudiantes hacia la asignatura (Pérez & Ramírez, 2022). Estas herramientas permiten representar visualmente conceptos abstractos como espacios vectoriales, transformaciones lineales y operaciones matriciales, favoreciendo un aprendizaje más significativo y autónomo (Lopez & García, 2020).

Entre las aplicaciones más utilizadas se encuentra **GeoGebra**, con más de 10 millones de descargas, que permiten una visualización interactiva en 2D y 3D de vectores y matrices. **Microsoft Math Solver** y **Wolfran Alpha** también figuran entre las más populares, gracias a sus potentes motores de resolución y explicaciones paso a paso de sistemas de ecuaciones lineales, operaciones matriciales y valores propios.

Asimismo, **Mathway**, ampliamente utilizada por estudiantes para resolver problemas paso a paso de álgebra y cálculo, facilita la práctica independiente y la validación de resultados. Por otro lado, **Matrix operation Calculator** y **Linear Algebra** ofrecen funciones específicas para trabajar con matrices, rangos, determinantes y espacios vectoriales, siendo útiles tanto para la práctica como para la enseñanza estructurada. Aunque esta última es más común en su versión web.

Finalmente, plataformas como **khan academy** complementan estas herramientas con contenido didáctico audiovisual que refuerzan los fundamentos teóricos del Álgebra Lineal, aunque su uso se orienta más al refuerzo teórico que al cálculo automatizado. En conjunto, la integración de estas aplicaciones móviles ha permitido una mayor personalización del aprendizaje, fomentando la participación más activa del estudiante, el aprendizaje a ritmo propio y una disposición favorable hacia el estudio de las matemáticas abstractas.

Es de fundamental importancia que para el desarrollo de software se parta del estudio de la ingeniería de software y el estudio de las metodologías ágiles para el desarrollo de aplicaciones móviles.

Ingeniería de software

El origen de la ingeniería del software como palabra es atribuida a la OTAN (Organización del Tratado del Atlántico Norte), en la década de los sesenta se realizaron un par de conferencias relacionadas con la “Crisis del Software” donde buscaban abordar los problemas asociados a fallas en la entrega de software y la mala calidad. La ingeniería de software según (Pressman, 2010) es: “*La aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software; es decir, la aplicación de la ingeniería al software*”.

Como se muestra en la figura tomada de (Pressman, 2010, p.12), las capas de la ingeniería de software se organizan de manera estructurada, siendo la calidad su pilar fundamental. A través de procesos como base para las capas superiores, se logra un desarrollo eficiente, estructurado y oportuno del software.



Figura 1 Capas de la ingeniería de software (Pressman, 2010)

Según (Pressman, 2010), los *métodos* de la ingeniería de software ofrecen el conocimiento técnico necesario para desarrollar software.

Las *herramientas* de la ingeniería de software, tal como señala (Pressman, 2010), brindan soporte automatizado o semiautomatizado para facilitar los métodos y procesos utilizados.

De acuerdo con (Pressman, 2010), un *proceso* se define como un conjunto de actividades, acciones y tareas llevadas a cabo para crear un producto de trabajo.

A continuación, se definen las metodologías ágiles y la selección de Scrum.

Metodologías ágiles

Las metodologías ágiles según (Lasa Gómez, Álvarez García, & de Las Heras del Dedo, 2017) surgieron como una respuesta a los problemas persistentes en el desarrollo de proyectos. La incertidumbre, uno de los mayores retos en este ámbito, se ha intentado abordar mediante la implementación de controles adicionales, planificación detallada, estimaciones precisas y diseños minuciosos. Sin embargo, los proyectos continúan entregándose fuera de tiempo, con costos elevados y sin mejoras significativas en la calidad.

En 1968, el desarrollo de software enfrentó una problemática conocida como la “crisis del software”. Para superarla, se implementaron metodologías, técnicas y herramientas diseñadas para garantizar la calidad de los productos, cumplir con los plazos y respetar los costos imprevistos. Como solución tradicional, se optó por aumentar el control, definir detalladamente los requisitos desde el inicio, emplear técnicas para evaluar la complejidad del trabajo, estimar el esfuerzo requerido y utilizar herramientas que supervisarán el proceso y evaluarán la calidad. No obstante, estas medidas no mejoraron la contribución de las personas al trabajo.

En 1986, Takeuchi y Nonakata describieron un enfoque de trabajo en el que un equipo enfrenta las etapas de un proyecto de manera similar a cómo un equipo de rugby encara una melé (Scrum en inglés): avanzando juntos con determinación. Esta idea se trasladó a la industria de software y, en 1995 se presentaron artículos con ideas semejantes para abordar proyectos Scrum.

Todo lo anterior llevó a la creación del Manifiesto para el Desarrollo Ágil de Software, publicado en 2001. En este documento, un grupo destacado de expertos en desarrollo de software expresó las necesidades y preocupaciones de la industria. El manifiesto se basa en cuatro principios fundamentales:

- Se priorizan los individuos y sus interacciones por encima de los procesos y las herramientas.
- Tiene mayor valor contar con un software funcional que disponer de una documentación extensa.

- La colaboración con el cliente es más importante que centrarse en negociar los términos del contrato.
- Adaptarse a los cambios resulta más relevante que seguir estrictamente un plan.

Estos principios establecen una distinción clara entre las metodologías tradicionales de gestión de proyectos y la manera de operar en el dinámico entorno tecnológico y empresarial actual.

Para el desarrollo del presente proyecto se ha seleccionado la metodología **Scrum**, en virtud de su eficacia comprobada en entornos donde la flexibilidad, la colaboración y la entrega continua de valor son esenciales. Scrum es un marco de trabajo ágil que proporciona una estructura definida a través de roles, eventos y artefactos que permiten gestionar de forma eficaz tanto el trabajo del equipo como la relación con los stakeholders (Schwaber & Sutherland, 2020).

A diferencia de otros enfoques como **Extreme Programming (XP)** o herramientas como el **Business Model Canvas**, Scrum permite una gestión integral del desarrollo de productos, incorporando tanto aspectos técnicos como organizativos. Si bien XP se centra en prácticas de desarrollo como la integración continua, la programación en parejas y las pruebas automatizadas, su alcance se limita principalmente al plano técnico, careciendo de elementos robustos para la gestión del producto y la priorización de valor desde el punto de vista del negocio (Beck, 2004). En este sentido, XP puede ser un excelente complemento dentro de Scrum, pero no reemplaza su estructura de planificación y control.

Por otra parte, el **Business Model Canvas** es una herramienta útil en las fases tempranas del proyecto, particularmente en la formulación del modelo de negocio, pero no constituye una metodología para el desarrollo iterativo e incremental de productos. No aborda la organización del equipo, la gestión de tareas, ni contempla ciclos de revisión y mejora continua (Osterwalder & Pigneur, 2010).

Scrum, en contraste, está diseñado para adaptarse a entornos cambiantes, gracias a su enfoque iterativo basado en **sprints** cortos que permiten entregar versiones funcionales del producto de manera frecuente. Esto favorece la retroalimentación temprana y la incorporación de mejoras a lo largo del ciclo de vida del proyecto. Además, su

escalabilidad ha sido comprobada mediante marcos como **Scrum@Scale** y **SAFe**, lo que la convierte en una metodología adecuada tanto para equipos pequeños como para organizaciones de mayor envergadura (Leffingwell, 2019).

Finalmente, Scrum promueve la **mejora continua** a través de la inspección y adaptación, pilares fundamentales del marco, los cuales se materializan en eventos como la retrospectiva del sprint. Esta práctica permite que los equipos evolucionen constantemente en su forma de trabajo, lo cual es esencial para mantener una alta calidad y eficiencia a lo largo del tiempo.

Por todas estas razones, la elección de la metodología Scrum se considera la más adecuada para este proyecto, ya que proporciona un marco estructurado, flexible y orientado a resultados, facilitando tanto la gestión del trabajo como la alineación con los objetivos del cliente y del equipo.

Metodología Scrum

La metodología Scrum según (Schwaber & Sutherland, 2020) se define como *“un marco de trabajo liviano que ayuda a las personas, equipos y organizaciones a generar valor a través de soluciones adaptativas para problemas complejos”*.

En pocas palabras, Scrum requiere de un Scrum Master para fomentar un entorno donde:

1. Un product Owner organiza las tareas de un problema complejo en un Product Backlog.
2. El Scrum Team transforma una parte seleccionada del trabajo en un increment valioso durante cada Sprint.
3. El Scrum Team junto con los interesados, evalúan los resultados obtenidos y realiza ajustes para el siguiente Sprint.
4. Este proceso se repite.

También (Schwaber & Sutherland, 2020), mencionan que la teoría de Scrum se fundamenta en los principios del empirismo y el pensamiento *Lean*. El empirismo sostiene que el conocimiento se adquiere a través de la experiencia y que las decisiones se toman

basándose en observaciones. Por su parte, el pensamiento *Lean* busca minimizar desperdicios y centrarse únicamente en lo esencial.

Scrum integra cuatro eventos formales diseñados para permitir la inspección y la adaptación, los cuales tienen lugar dentro de un evento principal conocido como sprint. Estos eventos son efectivos por que se fundamentan en los pilares empíricos de Scrum: transparencia, inspección y adaptación.

Transparencia

Es fundamental que tanto el proceso como los resultados sean visibles para quienes participan en su ejecución y quienes reciben el producto. Las decisiones clave deben basarse en la percepción clara del estado de los tres artefactos formales. (Schwaber & Sutherland, 2020)

Inspección

Los artefactos de Scrum, junto con el avance hacia los objetivos establecidos, deben ser evaluados regularmente y con cuidado para identificar posibles problemas o desviaciones no deseadas. Este proceso de evaluación es esencial porque da paso a la adaptación. (Schwaber & Sutherland, 2020)

Sin embargo, inspeccionar sin realizar ajustes se considera una práctica ineficaz. Los eventos Scrum están diseñados específicamente para generar los cambios necesarios.

Adaptación

Cuando algún elemento del proceso se desvía de los límites aceptables o el producto obtenido no cumple con las expectativas, es crucial realizar ajustes tanto en el proceso como en los productos. Estas correcciones deben efectuarse de manera inmediata para evitar que las desviaciones se amplíen. (Schwaber & Sutherland, 2020)

Scrum Team

El elemento central de Scrum es un equipo compacto conocido como el Scrum Team. Este equipo está compuesto por un Scrum Master, un Product Owner y los Developers. En el Scrum Team no existen divisiones internas ni niveles jerárquicos, ya que se trata de un

grupo de profesionales que trabajan juntos con un enfoque compartido en un único propósito: alcanzar el Objetivo del Producto. (Schwaber & Sutherland, 2020)

El Scrum Team, en su conjunto, tiene la responsabilidad de generar un Increment valioso y funcional durante cada Sprint. Dentro de este equipo, Scrum establece tres roles principales: Developers, Product Owner y Scrum Master.

Developers

Los Developers son los miembros del equipo que asumen el compromiso de desarrollar cualquier aspecto de un increment utilizable en cada Sprint. Aunque las habilidades necesarias pueden variar según el contexto, los Developers tienen las siguientes responsabilidades:

- Elaborar un plan para el Sprint, conocido como Sprint Backlog.
- Garantizar la calidad del trabajo cumpliendo con la Definición de Terminado.
- Ajustar diariamente su plan para alcanzar el objetivo del Sprint.
- Asumir responsabilidad compartida como profesionales.

Product Owner

El Product Owner es el encargado de maximizar el valor del producto resultante del trabajo del equipo. Este rol varía según la organización, el equipo y las personas, pero incluye la gestión eficiente del Product Backlog. Sus responsabilidades incluyen:

- Definir y comunicar claramente el objetivo del producto.
- Crear y expresar de forma clara los elementos del Product Backlog.
- Priorizar los elementos del Product Backlog.
- Asegurar que el Product Backlog sea transparente, visible y comprendido por todos.

Scrum Master

El Scrum Master se encarga de implementar Scrum tal como se describe en su guía oficial, promoviendo la comprensión de su teoría y práctica tanto dentro del equipo como

en toda la organización. Su objetivo principal es potenciar la efectividad del Scrum Team ayudando a mejorar sus prácticas dentro del marco de Scrum.

El Scrum Master lidera y apoya al equipo de varias formas, entre ellas:

- Guiar al equipo para que sea autogestionado y multifuncional.
- Asegurar que los Increments cumplan con la Definición de Terminado y sean de alto valor.
- Eliminar los obstáculos que puedan dificultar el progreso del equipo.
- Garantizar que los eventos de Scrum se lleven a cabo de forma productiva, positiva y dentro del tiempo establecido.

Además, el Scrum Master también apoya al Product Owner ayudándolo a:

- Utilizar técnicas efectivas para definir Objetivos del Producto y gestionar el Product Backlog
- Asegurar la claridad y precisión en los elementos del Product Backlog.
- Promover la planificación empírica en entornos complejos.
- Facilitar la colaboración con los interesados según sea necesario.

Finalmente, el Scrum Master contribuye a la organización liderando y guiando la adopción de Scrum. Esto incluye:

- Planificar y asesorar en la implementación de Scrum.
- Ayudar a los empleados e interesados a comprender enfoques empíricos para gestionar trabajos complejos.
- Eliminar barreras entre los interesados y los Scrum Teams, facilitando un entorno colaborativo. (Schwaber & Sutherland, 2020).

Según (Schwaber & Sutherland, 2020), los eventos de Scrum son los siguientes:

1. **Sprint:** Es el evento principal que contiene los demás eventos. Es un periodo fijo, generalmente de una a cuatro semanas, en el cual el Scrum Team trabaja para completar un Increment de valor. Proporciona un marco de inspección y adaptación constante.
2. **Sprint Planning:** Marca el inicio del Sprint. Durante este evento, el equipo planifica el trabajo a realizar en el Sprint, definiendo el Objetivo del Sprint, los elementos del Product Backlog a desarrollar y un plan inicial.

3. **Daily Scrum:** Una reunión diaria de 15 minutos donde los Developers sincronizan actividades y ajustan su plan para alcanzar el Objetivo del Sprint. Promueve la transparencia y permite detectar problemas temprano.
4. **Sprint Review:** Se realiza al finalizar el Sprint para inspeccionar el Increment desarrollado y recibir retroalimentación de los interesados. Esto ayuda a adaptar el Product Backlog según las necesidades emergentes.
5. **Sprint Retrospective:** Es el último evento del Sprint, donde el equipo reflexiona sobre el proceso, identifica mejoras y crea un plan para implementarlas en el próximo Sprint. Este evento fomenta la mejora continua.

Según (Schwaber & Sutherland, 2020), los artefactos de Scrum son los siguientes:

1. **Product Backlog:** Es una lista priorizada y dinámica que contiene todo el trabajo necesario para desarrollar el producto. Es gestionada por el Product Owner y se actualiza continuamente para reflejar las necesidades del producto y la organización.
2. **Sprint Backlog:** Es un subconjunto del Product Backlog que incluye los elementos seleccionados para el Sprint, junto con un plan detallado para entregarlos. Los Developers son responsables de mantenerlo actualizado y de usarlo como guía para alcanzar el Objetivo del Sprint.
3. **Increment:** Es el resultado tangible del trabajo completado durante el sprint. Representa una adición al producto que cumple con la definición de terminado y está listo para ser entregado o inspeccionado.
4. **Definición de Terminado:** Es un estándar comúnmente acordado que asegura que cada Increment cumple con los requisitos de calidad necesarios. Proporciona claridad sobre cuándo un trabajo está realmente completado.

Estos eventos y aparatos están diseñados para implementar los pilares de transparencia, inspección y adaptación, promoviendo la eficacia del marco de trabajo Scrum.

A continuación, se muestra un esquema de cómo funciona Scrum, presentando el ScrumTeam, los Roles, Sprint, eventos y artefactos.

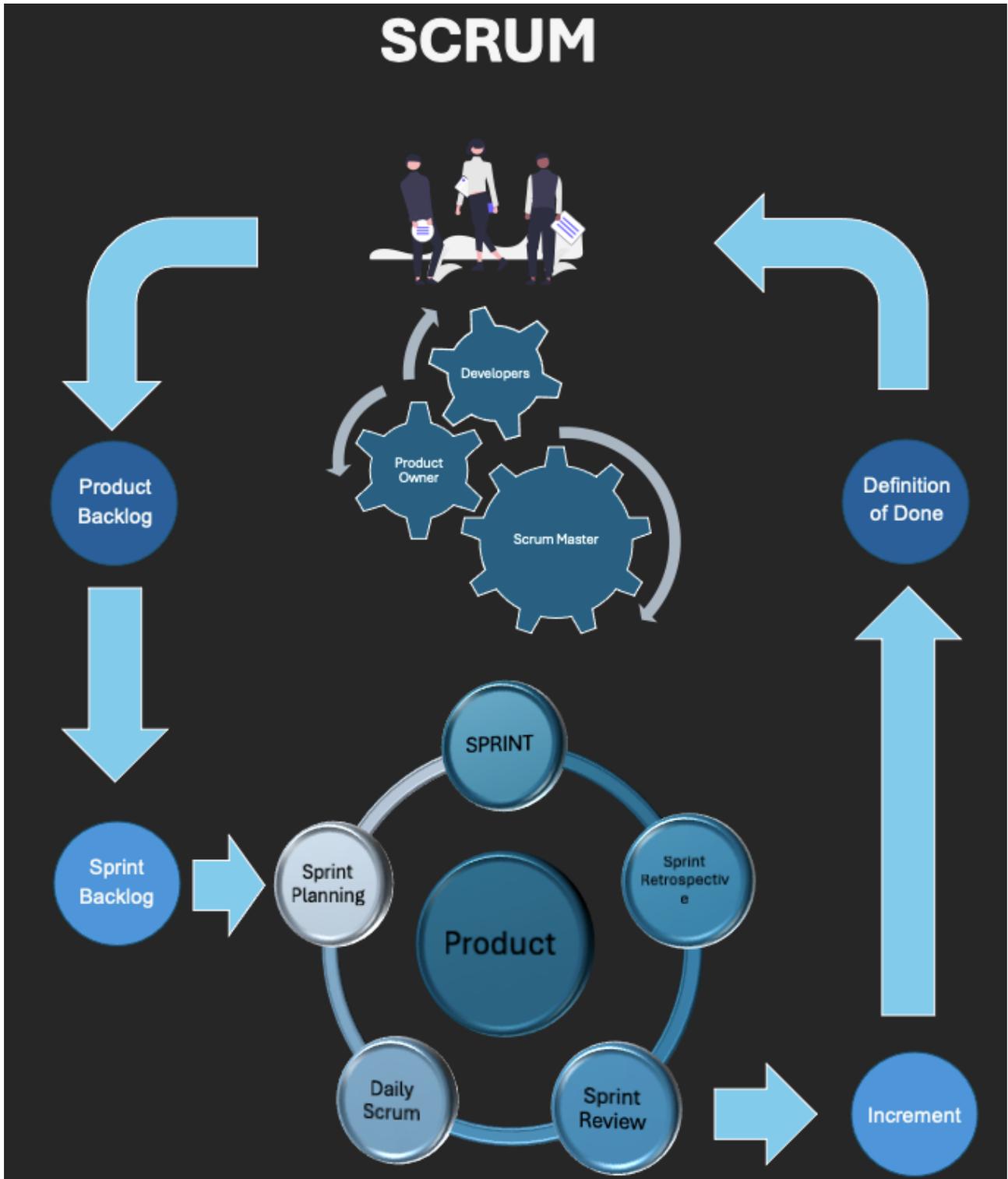


Figura 2 Esquema Scrum, roles, eventos y artefactos.

Implementación de la aplicación móvil de Álgebra Lineal utilizando Scrum

ROLES Y RESPONSABILIDADES DEL EQUIPO

Scrum Master

- **Nombre:** Cristian Rodolfo Gómez Mares.
- **Rol principal:** Facilitar la aplicación del marco Scrum y apoyar al equipo para eliminar impedimentos.
- **Responsabilidades específicas en el proyecto:**
 1. Coordinar las reuniones diarias (Daily)
 2. Asegurar que las retrospectivas identifiquen oportunidades de mejora.
 3. Mitigar riesgos relacionados con bloqueos del equipo.
 4. Mantener comunicación constante con las partes interesadas para abordar problemas organizativos.

Product Owner

- **Nombre:** Cristian Rodolfo Gómez Mares
- **Rol principal:** Maximizar el valor del producto y gestionar las prioridades del *Product Backlog*.
- **Responsabilidades específicas en el proyecto:**
 1. Definir y actualizar la visión del producto.
 2. Priorizar el *Backlog* para alcanzar los objetivos estratégicos.
 3. Revisar y aprobar los incrementos desarrollados al final de cada Sprint.
 4. Mantener reuniones periódicas con los stakeholders para garantizar el alineamiento con sus expectativas.

Equipo de Desarrollo

- **Miembros:**
 - Cristian Rodolfo Gómez Mares, **Rol:** Diseño y Desarrollo.
- **Rol principal:** Crear un incremento funcional del producto al final de cada Sprint.
- **Responsabilidades específicas en el proyecto:**
 1. Implementar las características definidas en las historias de usuario.

2. Participar activamente en la planificación de Sprints para estimar y desglosar tareas.
3. Garantizar la calidad del producto mediante pruebas unitarias e integraciones continuas.
4. Utilizar herramientas como GitHub y Trello, para mantener la transparencia del progreso.

Interacciones y Comunicación

- **Canales de comunicación:**
 1. Reuniones diarias (Daily): Por Zoom
 2. Planificación de Sprints: Cada lunes al inicio del Sprint, vía Zoom.
 3. Revisión del Sprint: Último día del Sprint, presentación vía Zoom.
 4. Retrospectiva: Último día del Sprint, análisis interno del equipo.
- **Herramientas:**
 1. Repositorio de código: GitHub
 2. Gestión del trabajo: Trello

Indicadores de Éxito

- Cumplimiento del *Definition of Done (DoD)* en cada incremento.
- Historias de usuario completadas dentro del Sprint Planificado.
- Satisfacciones basadas en entregas y revisión del producto.

Revisión y Aprobación

Tabla 1 Revisión y aprobación

Nombre	Rol
Cristian Rodolfo Gómez Mares	Scrum Master
Cristian Rodolfo Gómez Mares	Product Owner
Cristian Rodolfo Gómez Mares	Dev Team Member

PRODUCT BACKLOG

Historias de usuario:

1. **Créditos y última actualización:**

- Como usuario, quiero acceder a una pantalla adicional que me permita visualizar los créditos relacionados con la aplicación y ver un mensaje con la última fecha de actualización, para estar informado sobre las actualizaciones del sistema.
- **Criterios de aceptación:**
 - **Visualización de créditos:** El sistema debe proporcionar un botón en la interfaz principal que, al pulsarlo, lleve al usuario a una pantalla donde se muestren los créditos asociados a la aplicación, incluyendo detalles como nombres y roles de los desarrolladores.
 - **Consulta de última actualización:** El sistema debe proporcionar un botón que, al pulsarlo, muestre un mensaje emergente o en pantalla con la fecha más reciente de actualización de la aplicación.

2. Temas de álgebra lineal:

- Como usuario, quiero ingresar a la aplicación móvil y visualizar una lista de temas de álgebra lineal en la pantalla principal, para poder explorar los ejercicios relacionados y gestionar mi progreso.
- **Criterios de aceptación:**
 - **Visualización de temas:**
 - El sistema debe mostrar una lista clara y organizada de los temas de álgebra lineal en la pantalla principal al iniciar la aplicación.
 - El usuario debe poder seleccionar cualquier tema de la lista para acceder a los ejercicios asociados.
 - **Gestión del progreso:**
 - El sistema debe permitir al usuario seleccionar un tema y marcarlo como terminado.
 - Una vez marcado, el tema debe moverse automáticamente a una pestaña específica denominada “Terminados”, donde aparecerá claramente identificado como terminado.

3. Pestañas con los temas en progreso, terminados y todos:

- Como usuario, quiero poder visualizar los temas que están en progreso, los que ya fueron completados y una combinación de ambos en una pestaña denominada “Todos”, para organizar mi aprendizaje de manera eficiente.
- **Criterios de aceptación:**
 - **Visualización por estado:**
 - El sistema debe proporcionar tres pestañas claramente identificadas en la interfaz principal:
 - a. “En Progreso”: Muestra únicamente los temas que están trabajando.
 - b. “Terminados”: Muestra exclusivamente los temas que ya han sido completados.
 - c. “Todos”: Combina y muestra los temas de ambas categorías.
 - **Interacción con las pestañas:**
 - El usuario debe poder cambiar fácilmente entre pestañas mediante un sistema de navegación claro y accesible (Por ejemplo, un menú superior o inferior).
 - El contenido de cada pestaña debe actualizarse dinámicamente al cambiar el estado de un tema (de “En Progreso” a “Terminado”).
 - **Estado visual de los temas:**
 - Dentro de la pestaña “Todos”, los temas deben estar etiquetados visualmente con su estado actual (por ejemplo, un ícono o color distinto para “En Progreso” y “Terminado”).
 - **Persistencia de los estados:**

- Los estados de los temas (en progreso o terminados) deben guardarse de manera persistente, de modo que no se pierdan al cerrar o reiniciar la aplicación.

4. Pantalla con ejercicios asociados al tema:

- Como usuario, quiero visualizar todos los ejercicios relacionados con el tema seleccionado previamente y poder gestionar mi progreso, para completar los ejercicios de manera organizada.
- **Criterios de aceptación:**
 - **Visualización de ejercicios:**
 - El sistema debe mostrar una lista clara y ordenada de los ejercicios relacionados con el tema seleccionado previamente.
 - El usuario debe poder seleccionar cualquier ejercicio de la lista para acceder al enunciado detallado y resolverlo.
 - **Gestión del progreso:**
 - El sistema debe permitir al usuario marcar un ejercicio como terminado.
 - Una vez marcado como terminado, el ejercicio debe moverse automáticamente a una pestaña específica denominada “Terminados”, donde aparecerá claramente identificado con una etiqueta visual (por ejemplo, un ícono o un color distintivo).
 - Los ejercicios seleccionados como completados no deben aparecer en la lista principal de ejercicios.
 - **Persistencia de datos:**
 - El estado de los ejercicios (en progreso o terminados) debe guardarse de forma persistente, de modo que al cerrar o reiniciar la aplicación los datos permanezcan intactos.
 - **Interactividad y navegación:**

- El sistema debe incluir un botón de navegación claro para regresar a la lista de temas y seleccionar otro tema si lo desea el usuario.
- El sistema debe incluir un botón de ayuda para mostrar un mensaje con instrucciones claras de lo que tiene que realizar en esta sección.
- La transición entre la lista de ejercicios y el detalle del ejercicio seleccionado debe ser fluida y sin pérdida de información.

5. Pantalla con enunciado e incisos:

- Como usuario, quiero visualizar de forma clara el enunciado y las posibles respuestas resaltadas como incisos, para poder seleccionar una opción, recibir retroalimentación inmediata y tener la posibilidad de ver la solución correcta si lo deseo.
- **Criterios de aceptación:**
 - **Visualización de enunciado e incisos:**
 - El sistema debe mostrar el enunciado completo de manera clara y legible.
 - Las posibles respuestas deben estar resaltadas como incisos (por ejemplo “a.”, “b.”, “c.” y “d.”) y diseñadas para facilitar la selección de una opción.
 - **Selección de respuestas:**
 - El usuario debe poder seleccionar un inciso como respuesta.
 - Una vez seleccionada una respuesta, el sistema debe bloquear la selección de otras respuestas hasta que el usuario decida continuar o reintentar.
 - **Retroalimentación:**
 - Si la respuesta seleccionada es correcta, el sistema debe mostrar un mensaje indicando que la respuesta es correcta y proporcionar un botón para visualizar la solución completa.

- Si la respuesta seleccionada es incorrecta, el sistema debe mostrar un mensaje indicando que fue errónea y ofrecer las siguientes opciones:
 - a. **Intentar Nuevamente:** Permitir al usuario seleccionar otra opción.
 - b. **Ver solución:** Mostrar la respuesta correcta junto con una explicación detallada.
- **Navegación:**
 - Debe existir un botón para regresar a la lista de ejercicios o temas sin perder el progreso actual.
- **Accesibilidad y diseño:**
 - Las opciones de respuesta deben ser fáciles de interactuar (por ejemplo, botones grandes y accesibles para dispositivos móviles).
 - Los mensajes de retroalimentación deben ser claros, concisos.

6. Pestaña con el ejercicio y la teoría asociada:

- Como usuario, quiero ver la teoría asociada al ejercicio que estoy resolviendo y poder regresar al ejercicio fácilmente, con pestañas ubicadas en la parte inferior de la pantalla, para alternar entre ambos de manera intuitiva.
- **Criterios de aceptación:**
 - **Diseño y disposición:**
 - Las pestañas deben estar ubicadas en la parte inferior de la pantalla del ejercicio.
 - Las pestañas deben estar claramente etiquetadas como “Ejercicio” y “Teoría”.
 - El diseño debe permitir alternar entre las pestañas de manera rápida.
 - **Funcionalidad de la pestaña “Ejercicio”:**

- Debe mostrar el enunciado del ejercicio actual y cualquier progreso realizado hasta el momento.
- Al cambiar de la pestaña “Teoría” a “Ejercicio”, el usuario debe ver el mismo estado del ejercicio que dejó previamente.
- **Funcionalidad de la pestaña “Teoría”:**
 - Debe mostrar la información teórica relevante y relacionada con el ejercicio seleccionado.
 - El contenido debe estar bien estructurado, incluyendo explicaciones, ejemplos y fórmulas (si aplica).
- **Interactividad y navegación:**
 - Las pestañas deben ser fácilmente seleccionables, con un diseño accesible para dispositivos móviles y de escritorio.
 - Cambiar de pestaña no debe causar pérdida de datos o progreso en el ejercicio actual.

Tabla 2 Historia de usuario

ID	Historia de Usuario	Prioridad	Esfuerzo	Estado
1	Créditos y última actualización: Como usuario, quiero una pantalla adicional para visualizar los créditos de la aplicación y la última fecha de actualización.	Alta	5	Terminado
2	Temas de álgebra lineal: Como usuario, quiero visualizar los temas de álgebra lineal en la pantalla principal, seleccionarlos y gestionarlos como completados.	Alta	8	Terminado
3	Pestañas con temas (en progreso, terminados y todos): Como usuario, quiero ver los temas en	Media	5	Terminado

	diferentes categorías (en progreso, terminados y todos) organizados en pestañas.			
4	Pantalla con ejercicios asociados al tema: Como usuario, quiero ver una lista de ejercicios de un tema seleccionado previamente y gestionar mi progreso marcándolos como terminados.	Alta	13	Terminado
5	Pantalla con enunciados e incisos: Como usuario, quiero elegir entre las respuestas resaltadas en incisos, ver si mi respuesta es correcta o incorrecta, y tener opciones para intentarlo nuevamente o ver la solución.	Alta	13	Terminado
6	Pestaña con el ejercicio y la teoría asociada: Como usuario, quiero alternar entre la teoría y el ejercicio mediante pestañas ubicadas en la parte inferior de la pantalla.	Media	5	Terminado

SPRINTS

Sprint 1

- **Proyecto:** Aplicación móvil (app) para la asignatura de Álgebra Lineal.
- **Duración del Sprint:** 3 semanas.
- **Objetivo del Sprint:** Desarrollar la pantalla principal para la visualización y gestión de los temas de álgebra lineal, junto con una pantalla adicional que muestre los créditos y la última fecha de actualización de la aplicación.

Sprint Backlog

Tabla 3 Sprint Backlog

Historia de Usuario	Tareas Asociadas	Responsable	Estimación (Puntos)	Estado
Historia 1	1. Diseñar la UI de la pantalla de créditos.	Desarrollador	1	Completado
	2. Implementar componente de visualización de créditos.	Desarrollador	1	Completado
	3. Crear botón para mostrar la última fecha de actualización.	Desarrollador	1	Completado
	4. Implementar funcionalidad para visualizar la última fecha de actualización de la aplicación.	Desarrollador	1	Completado
	5. Pruebas unitarias e integración de la funcionalidad.	Desarrollador	1	Completado
Historia 2	1. Diseñar la UI de la lista de temas.	Desarrollador	2	Completado

	2. Implementar el componente de visualización de temas.	Desarrollador	2	Completado
	3. Crear funcionalidad para seleccionar temas.	Desarrollador	1	Completado
	4. Implementar funcionalidad para marcar temas como completados.	Desarrollador	1	Completado
	5. Actualizar estado de los temas en el almacenamiento local.	Desarrollador	1	Completado
	6. Pruebas unitarias e integración.	Desarrollador	1	Completado

Resumen del Sprint

- **Número total de tareas:** 11
- **Total de puntos de historia:** 13 puntos
- **Capacidad del equipo:** 18 puntos

Sprint Review

Asistentes:

- Equipo de desarrollo
- Product Owner
- Scrum Master
- Stakeholders (usuarios)

Demostración del Incremento:

A continuación, se presentan las funcionalidades completadas:

Historia 1: Créditos y última actualización

- Se muestra la pantalla que se implementó para visualizar los créditos relacionados con la aplicación:

- Funcionalidad del botón para mostrar la última fecha de actualización.
- Flujo de interacción del usuario y detalles de diseño como se muestra a continuación:

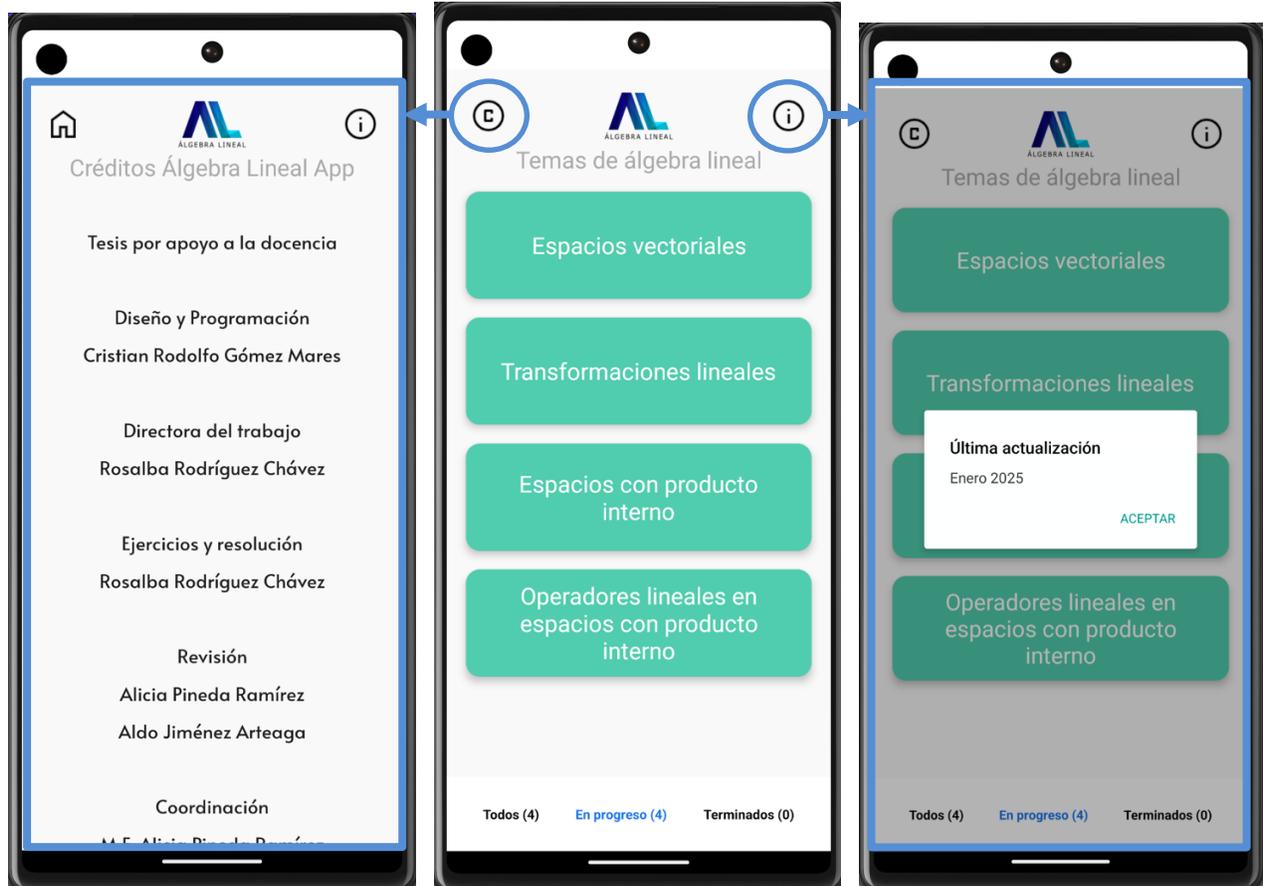


Figura 3 Demostración asociada a la historia de usuario 1 en sprint review.

Historia 2: Temas de álgebra lineal

- Pantalla principal con la lista de temas disponibles.
- Selección por parte del usuario.
- Marcado de temas como completados y movimiento automático a una lista específica como se muestra a continuación:

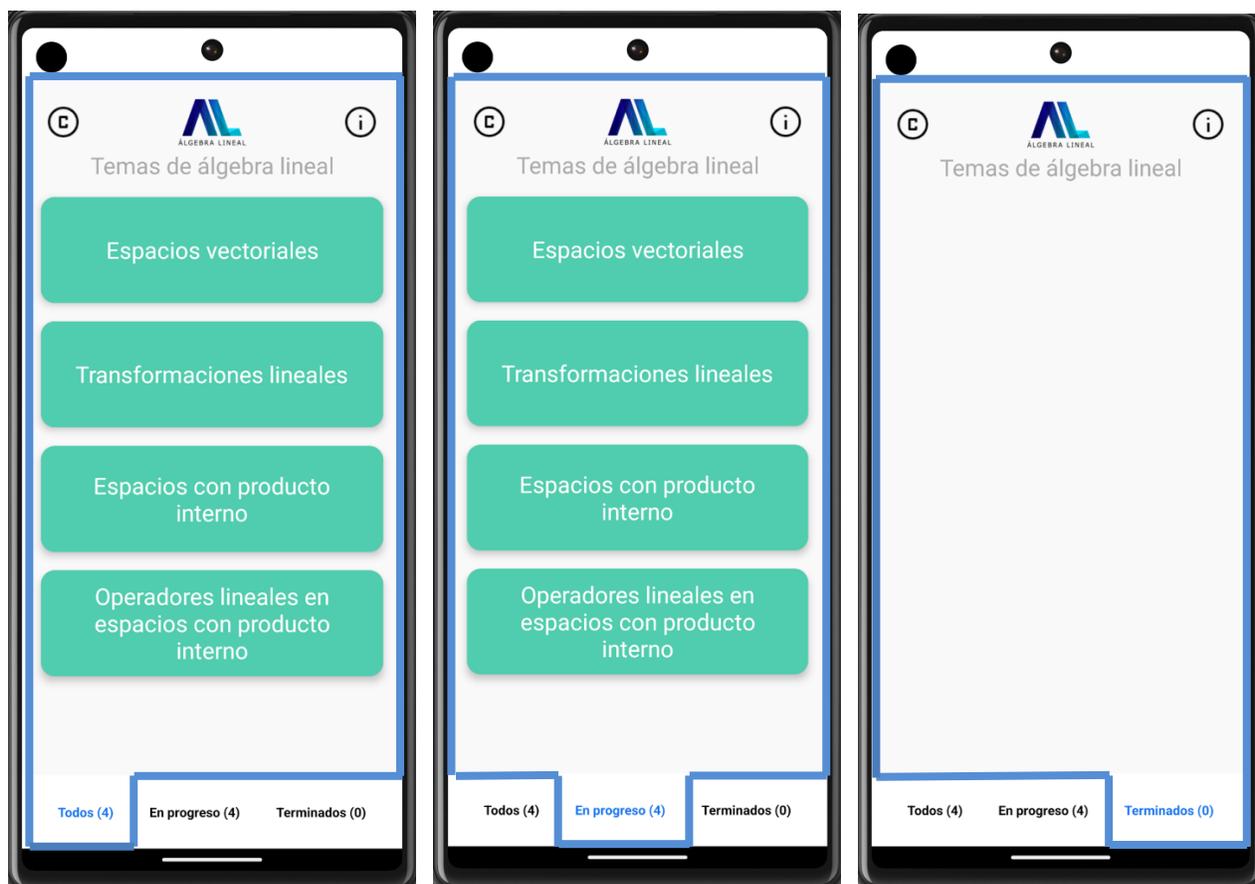


Figura 4 Pantalla principal con la lista de temas disponibles.

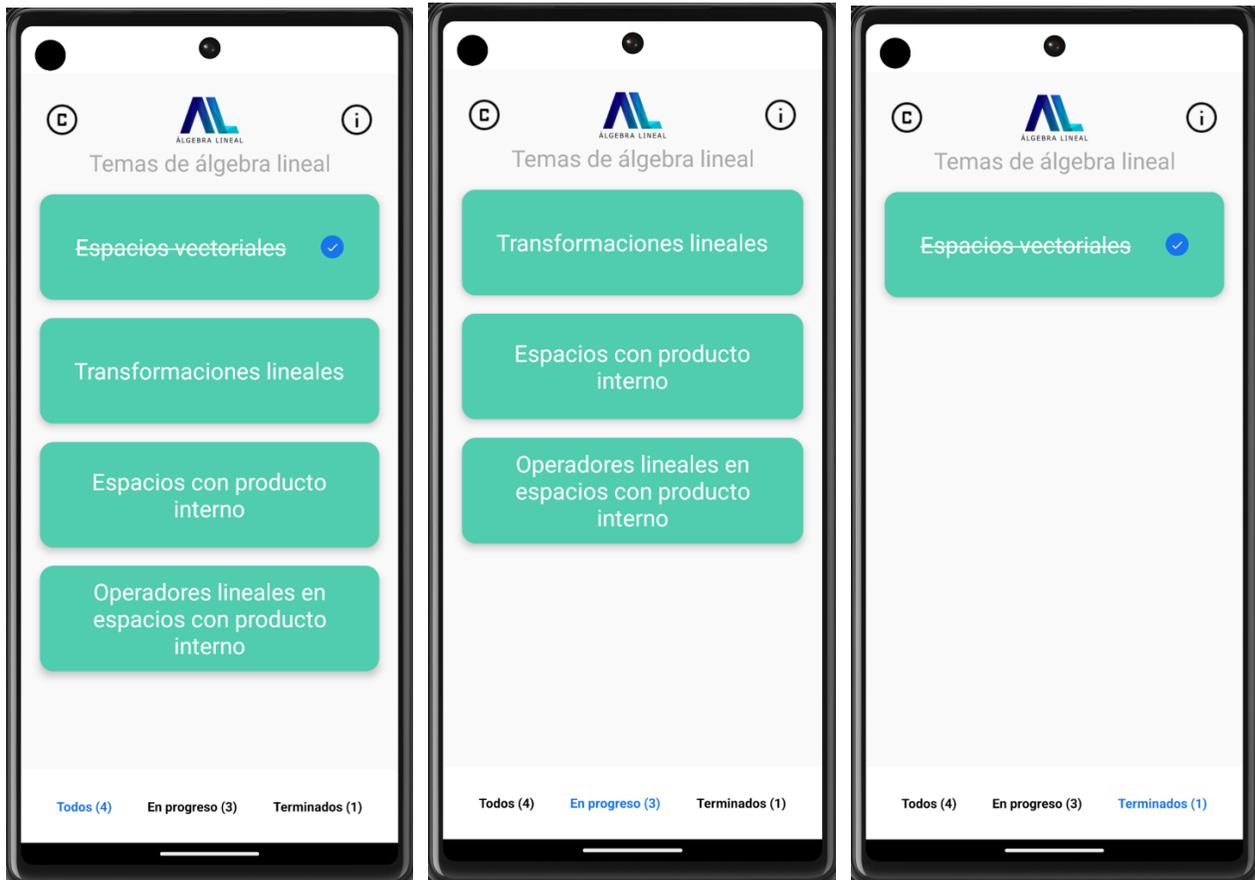


Figura 5 Marcado de temas como completados y movimiento automático a una lista de terminados.

Sprint 2

- **Proyecto:** Aplicación móvil (app) para la asignatura de Álgebra Lineal.
- **Duración del Sprint:** 3 semana.
- **Objetivo del Sprint:** Completar las funcionalidades relacionadas con la gestión de pestañas de (todos, en progreso, terminados) y la pantalla para visualizar los ejercicios asociados a los temas, asegurando una experiencia de usuario fluida.

Sprint Backlog

Tabla 4 Sprint Backlog. Tareas asociadas

Historia de usuario	Tareas asociadas	Responsable	Estimación (Puntos)	Estado
Historia 3	1. Crear diseño UI de las pestañas de gestión.	Desarrollador	1	Completado
	2. Implementar la lógica para cambiar entre pestañas.	Desarrollador	1	Completado
	3. Implementar actualización dinámica del estado y el almacenamiento local.	Desarrollador	1	Completado
	4. Realizar pruebas unitarias y de integración.	Desarrollador	2	Completado
Historia 4	1. Crear diseño UI de la pantalla para los ejercicios.	Desarrollador	1	Completado
	2. Implementar lógica para obtener los ejercicios dependiendo del tema seleccionado.	Desarrollador	3	Completado
	3. Implementar funcionalidad de navegación para regresar a la lista de temas.	Desarrollador	1	Completado
	4. Implementar funcionalidad para el botón de ayuda.	Desarrollador	1	Completado

	5. Implementar gestión de ejercicios para mostrar como terminados.	Desarrollador	3	Completado
	6. Habilitar acceso al detalle del enunciado.	Desarrollador	2	Completado
	7. Realizar pruebas unitarias y de integración.	Desarrollador	2	Completado

Resumen del Sprint

- **Número total de tareas:** 11
- **Total de puntos de historia:** 18 puntos
- **Capacidad del equipo:** 18 puntos

Sprint Review

Asistentes:

- Equipo de desarrollo
- Product Owner
- Scrum Master
- Stakeholders (usuarios)

Demostración del Incremento:

A continuación, se presentan las funcionalidades completadas:

Demostración del Incremento:

Historia 3: Pestañas con temas (en progreso, terminados y todos)

- Mostrar cómo el contenido de las pestañas cambia dinámicamente entre los estados “Todos”, “En progreso” y “Terminados”, como se puede ver a continuación.

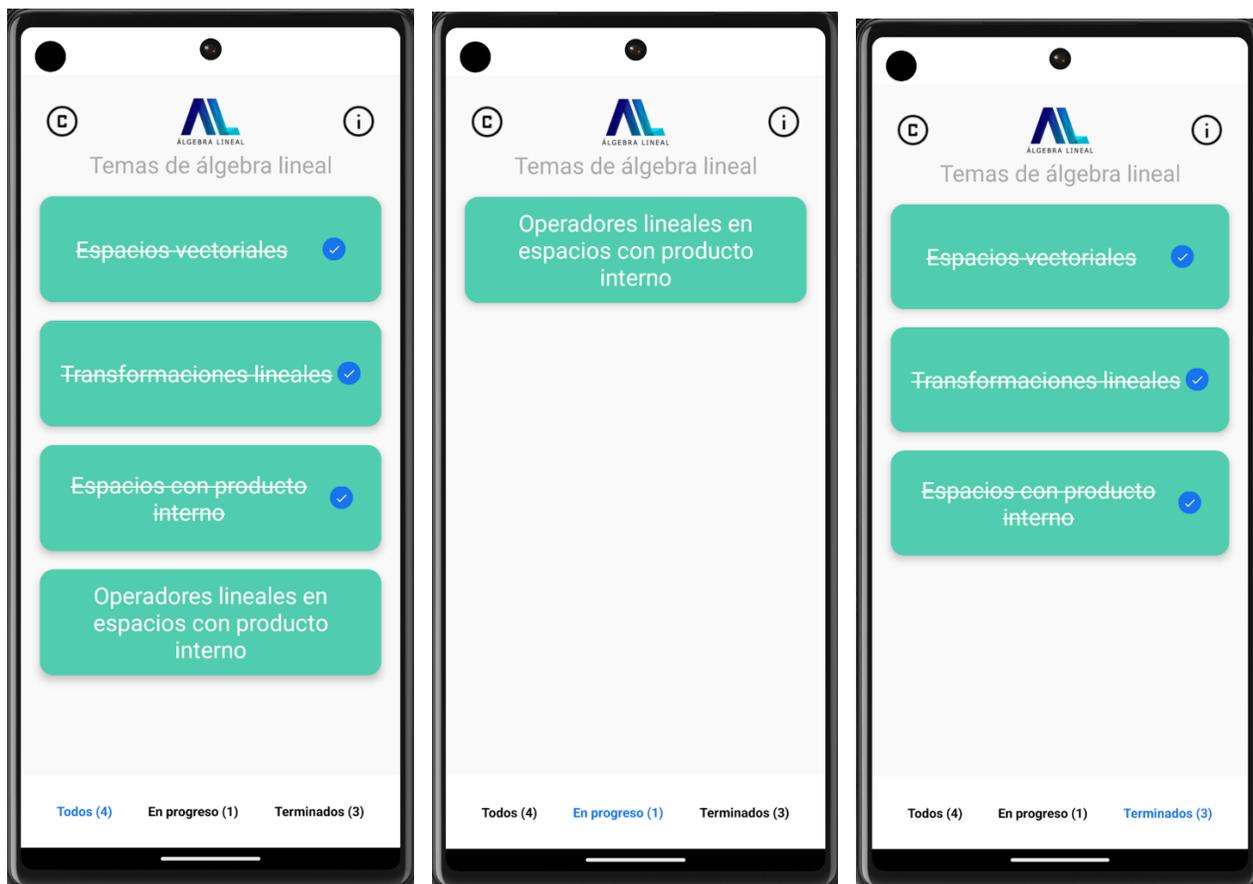


Figura 6 Renderizado de contenido en las pestañas de gestión de temas.

Historia 4: Pantalla con ejercicios asociados al tema

- Navegar a la pantalla de ejercicios asociados al tema seleccionado.
- Funcionalidad de navegación para regresar a la pantalla de temas.
- Mostrar el botón de ayuda de la sección.
- Marcar un ejercicio como terminado y verificar su movimiento a la pestaña de “Terminados”.
- Mostrar el acceso al enunciado de uno de los ejercicios.

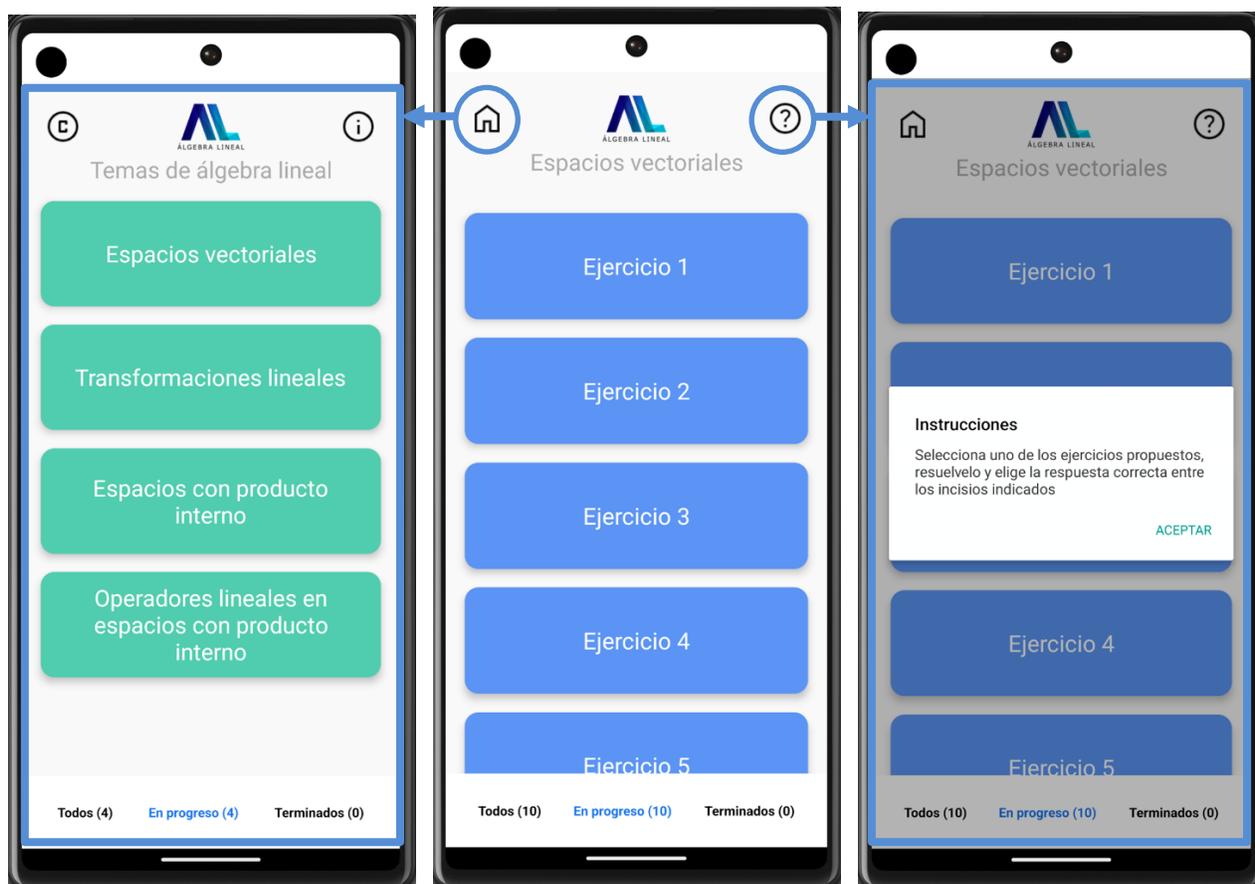


Figura 7 Botón de navegación y ayuda respectivamente.

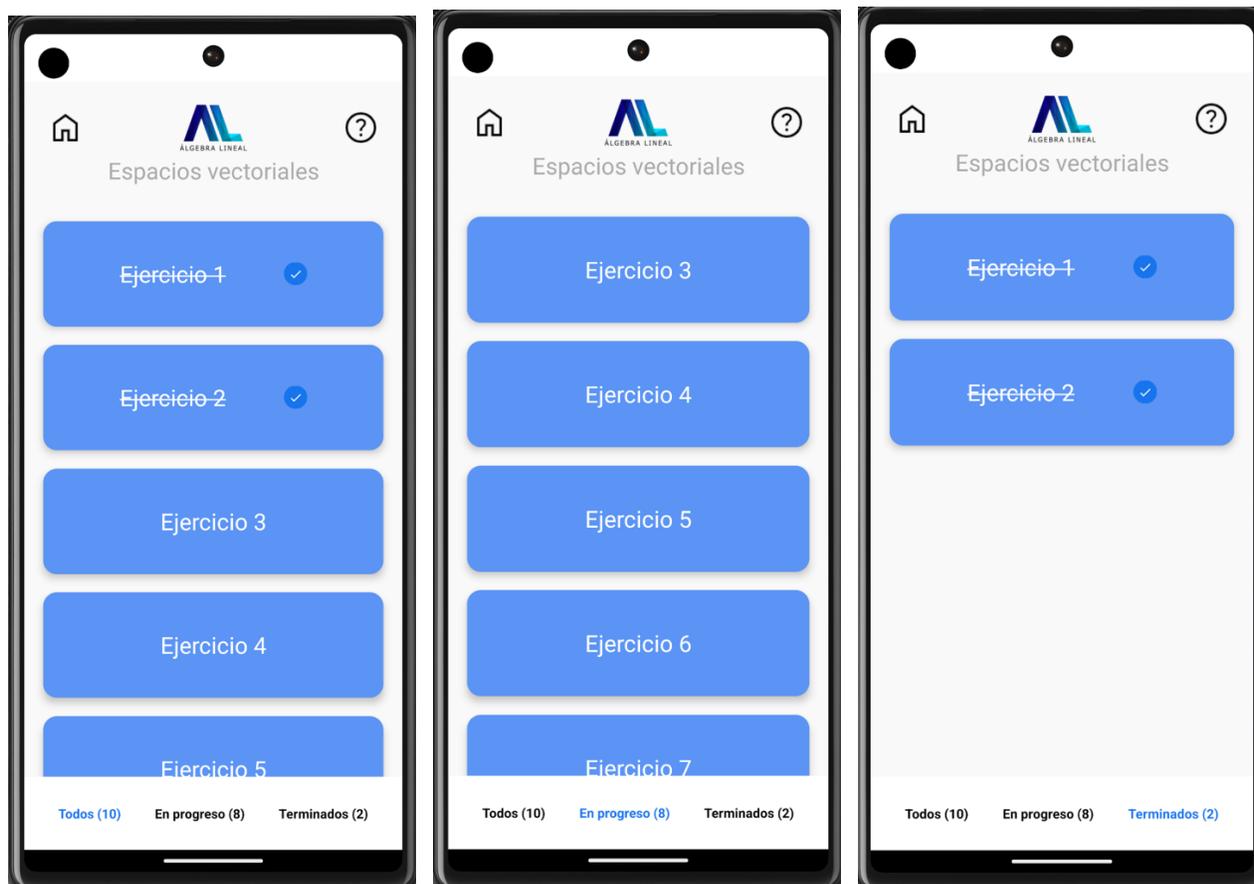


Figura 8 Ejercicios marcados como terminados para mostrar en las pestañas respectivas.

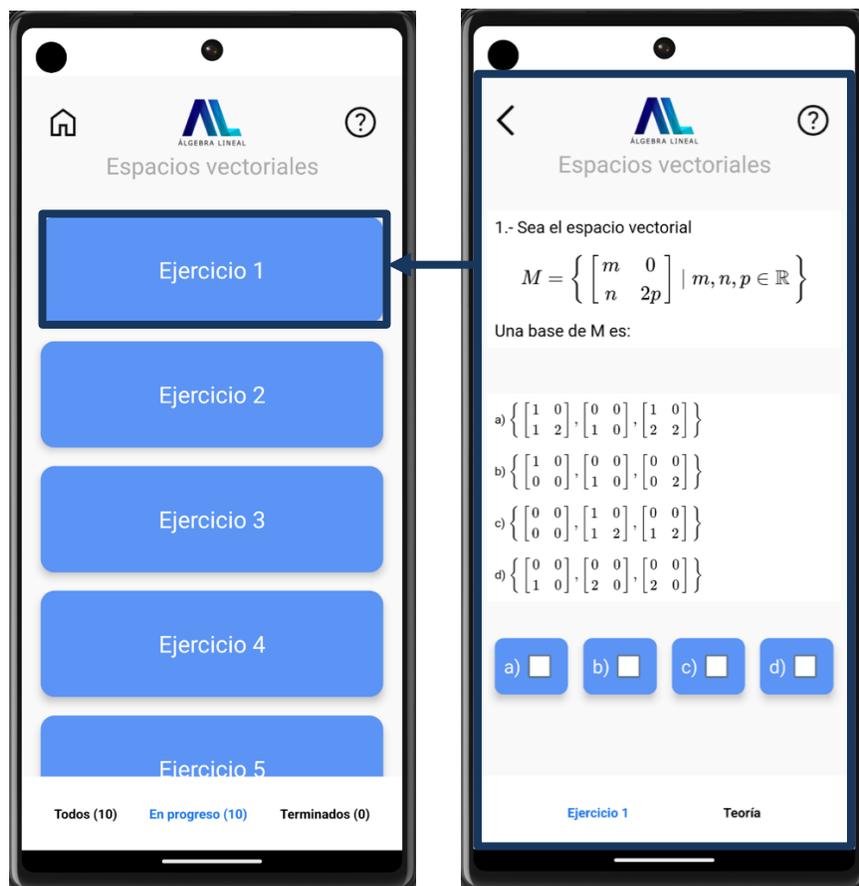


Figura 9 Selección de ejercicio y pantalla con su respectivo enunciado.

Sprint 3

- **Proyecto:** Aplicación móvil (app) para la asignatura de Álgebra Lineal.
- **Duración del Sprint:** 3 semana.
- **Objetivo del Sprint:** Implementar y entregar funcionalidades clave relacionadas con la gestión y visualización de ejercicios y teoría asociada, asegurando una experiencia de usuario fluida y consistente.

Sprint Backlog

Tabla 5 Sprint Backlog de las tareas.

Historia de usuario	Tareas asociadas	Responsable	Estimación (Puntos)	Estado
Historia 5	1. Diseñar la interfaz para mostrar el enunciado y las opciones de respuesta.	Desarrollador	3	Completada
	2. Implementar lógica para validación de respuestas y mostrar mensajes (correcta/incorrecta).	Desarrollador	4	Completada
	3. Agregar opción para visualizar la solución si el usuario lo desea.	Desarrollador	4	Completada
	4. Realizar pruebas unitarias y de integración.	Desarrollador	2	Completada
Historia 6	1. Diseñar tabs en la parte inferior de la pantalla para alternar entre teoría y ejercicio.	Desarrollador	1	Completada

	2. Implementar navegación entre los tabs (teoría y ejercicio).	Desarrollador	1	Completada
	3. Asegurar persistencia de datos al cambiar de un tab a otro.	Desarrollador	1	Completada
	4. Realizar pruebas unitarias para asegurar el correcto funcionamiento.	Desarrollador	2	Completada

Resumen del Sprint

- **Número total de tareas:** 8
- **Total de puntos de historia:** 18 puntos
- **Capacidad del equipo:** 18 puntos

Sprint Review

Asistentes:

- Equipo de desarrollo
- Product Owner
- Scrum Master
- Stakeholders (usuarios)

Demostración del Incremento:

A continuación, se presentan las funcionalidades completadas:

Demostración del Incremento:

Historia 5: Pantalla con enunciados e incisos

- Los usuarios pueden visualizar los enunciados y seleccionar respuestas.
- Mensajes dinámicos que indican si la respuesta es correcta o incorrecta.
- Se implementó la opción de ver la solución o reintentar el ejercicio.

Historia 6: Pestañas con el ejercicio y la teoría asociada

- Las pestañas permiten alternar fácilmente entre la teoría y el ejercicio en curso.
- Diseño intuitivo con tabs ubicados en la parte inferior de la pantalla.

Las imágenes con los resultados se muestran a continuación:

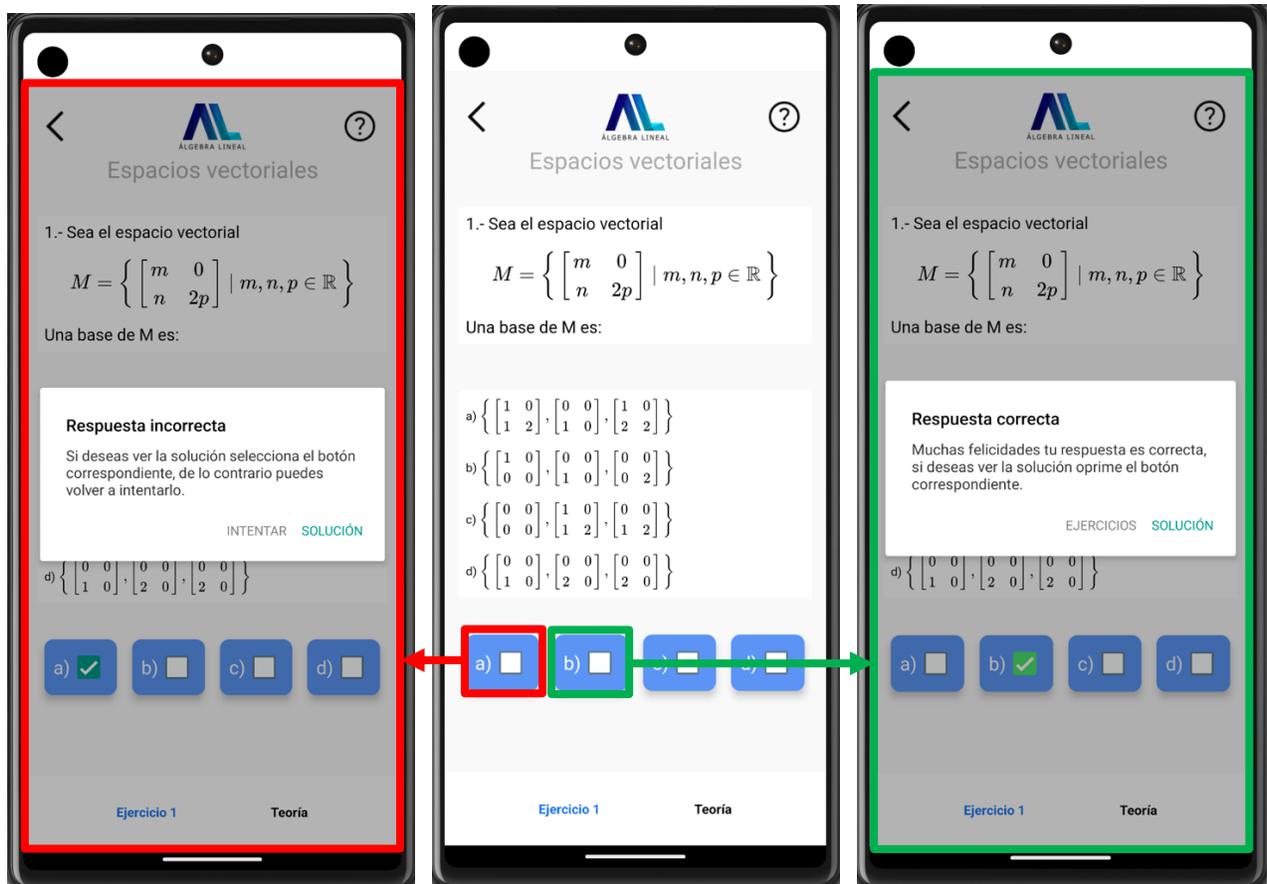


Figura 10 Selección de ejercicio y pantalla con su respectivo enunciado y respuesta.

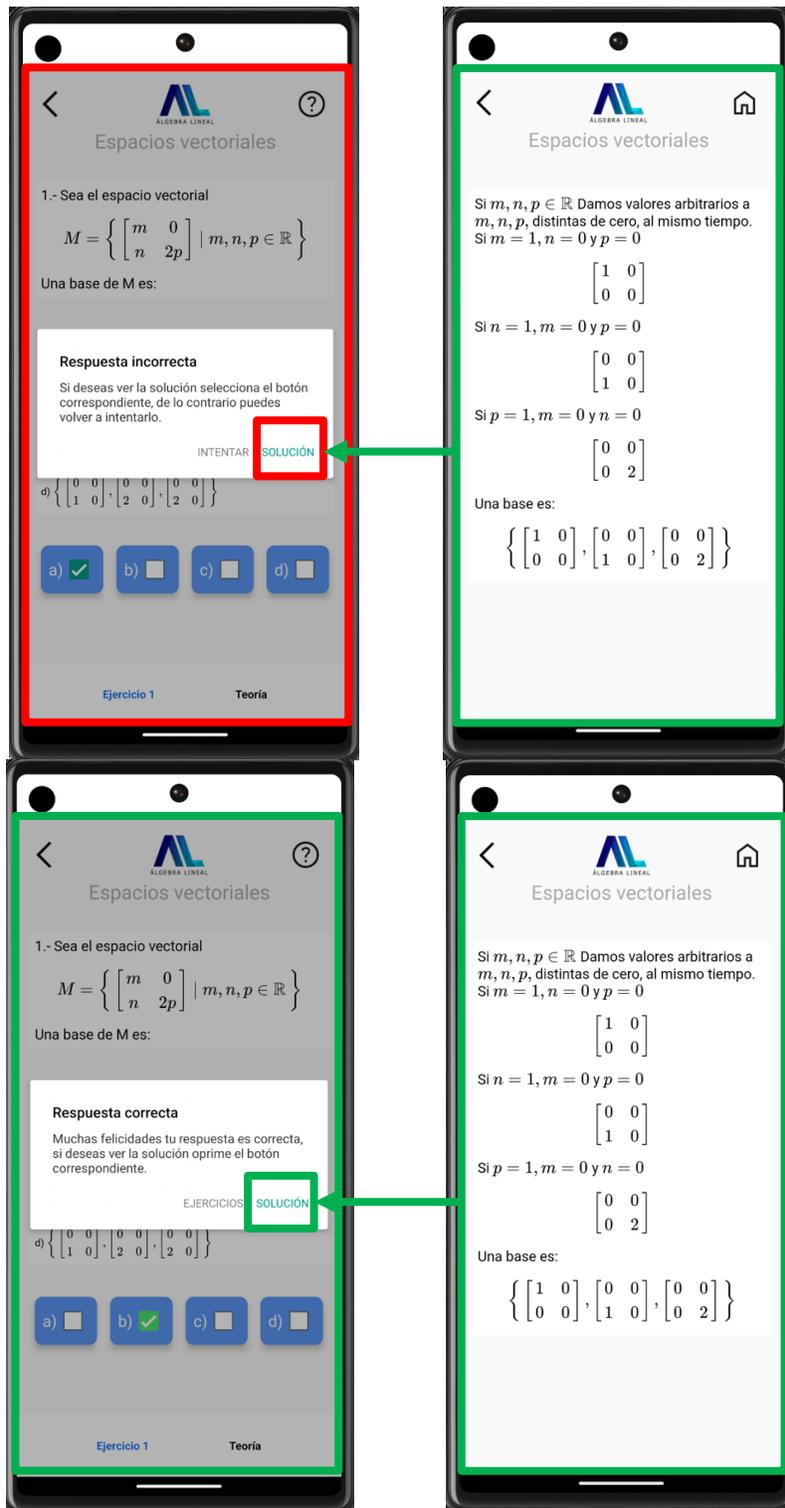


Figura 11 Opción para ver la solución sin importar si la respuesta es correcta o incorrecta.

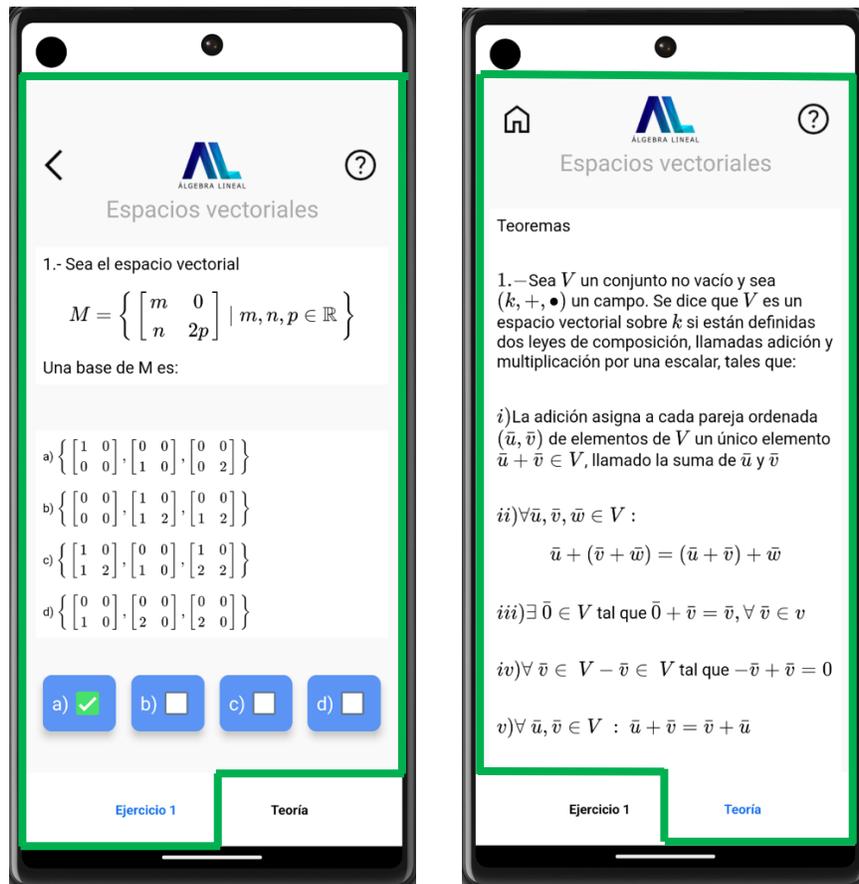


Figura 12 Navegación entre pestañas asegurando persistencia.

Nota: Para calcular la capacidad del equipo en un sprint, necesitamos conocer:

- Duración del Sprint (3 semanas).
- Número de integrantes del equipo (1 desarrollador).
- Horas disponibles por persona (6 horas efectivas por día).
- Días efectivos de trabajo (15 días hábiles en un sprint de 3 semanas).
- Velocidad histórica del equipo (5 horas por punto de historia)

Usamos la fórmula:

Capacidad del equipo = (Miembros) x (Horas efectivas por día) x (Días efectivos)

Para los tres sprint tenemos:

- 1 persona integra el equipo
- 6 horas efectivas por día

- 15 días hábiles

$$1 \times 6 \times 15 = 90 \text{ horas efectivas}$$

Si cada tarea toma aproximadamente 5 horas por punto de historia, la capacidad en puntos sería:

$$\frac{90 \text{ horas}}{5 \frac{\text{horas}}{\text{punto}}} = 18 \text{ puntos de historia}$$

Implementación

Se utilizó como herramienta de desarrollo Expo

El lenguaje fue JavaScript utilizando la biblioteca de react native

Justificación

La elección de utilizar Expo con React Native se basó en criterios clave como velocidad de desarrollo, curva de aprendizaje, madurez del ecosistema, reutilización de código y facilidad de mantenimiento.

Análisis general

1. Expo con React Native:

Expo es un conjunto de herramientas y servicios que mejora la experiencia de desarrollo con React Native, permitiendo lanzar apps móviles rápidamente sin necesidad de configurar entornos nativos complejos.

2. Flutter:

Fraemwork de UI de Google que permite compilar aplicaciones nativas desde un solo código base en Dart. Tiene rendimiento y una comunidad creciente.

3. Kotlin (Android Nativo):

Lenguaje oficial para desarrollo Android, altamente eficiente para aplicaciones específicas de Android, con control total sobre la plataforma nativa.

Tabla 6 Cuadro comparativo entre tecnologías

Característica	Expo + React Native	Flutter	Kotlin (Android Nativo)
Lenguaje principal	JavaScript / TypeScript	Dart	Kotlin
Curva de aprendizaje	Baja – ideal para equipos JS/TS	Media – requiere aprender Dart	Alta – require conocimiento profundo de Android SDK
Velocidad de desarrollo	Alta – configuración mínima con Expo	Media – require configuración manual	Baja – mucho código y configuración manual
Reutilización de código	Alta (IOS, Android, Web con React)	Alta (IOS y Android, Web limitado)	Nula – solo Android
Acceso a APIs nativas	Limitado con Expo (sinEAS) – ampliable con EAS	Completo y flexible	Completo y directo
Rendimiento	Bueno, aunque ligeramente inferior a nativo	Excelente – UI compilada a código nativo	Óptimo – acceso total a hardware
Ecosistema y librerías	Muy amplio – gran comunidad React	Amplio, pero en crecimiento	Muy maduro para Android
Actualizaciones y mantenimiento	Muy fácil con OTA y Expo Go	Requiere más control manual	Requiere subir actualizaciones a la Play Store
Testing y debugging	Herramientas modernas con hot reload	Muy buenas herramientas integradas	Herramientas oficiales robustas

Razones clave de la elección de Expo y React Native

Velocidad de desarrollo: Gracias a Expo, no es necesario configurar entornos nativos para comenzar. Esto aceleró la etapa inicial.

Curva de aprendizaje baja: Mi experiencia en JavaScript/TypeScript y React, redujo significativamente los tiempos de capacitación.

Despliegues más rápidos: Expo ofrece actualizaciones OTA (Over-the -Air), permitiendo corregir errores o actualizar partes de la app sin necesidad de pasar por las tiendas.

Multi-plataforma real: Posibilidad de escalar la aplicación a web en el futuro con React Native Web.

Ecosistema Maduro: Amplia disponibilidad de librerías, soluciones de UI y comunidad activa para solución de problemas rápidamente.

Patrón de arquitectura Clean Architecture

Este proyecto fue realizado con el patrón de arquitectura Clean Architecture porque permite una separación clara de responsabilidades entre las capas de la aplicación. Esta estructura es especialmente útil en proyectos móviles destinados a crecer y mantenerse en el tiempo.

Facilita las pruebas, el mantenimiento, la escalabilidad, y permite que los detalles de implementación (como UI o acceso a datos) cambien sin afectar la lógica del negocio.

Estructura del proyecto:

```
src/  
├── components/  
│   ├── Ask/  
│   │   ├── Ask.jsx  
│   │   └── Ask.style.js  
│   └── CardExercise/
```

```
| | └─ CardExercise.jsx
| | └─ CardExercise.style.js
| └─ CardHome/
| | └─ CardHome.jsx
| | └─ CardHome.style.js
| └─ CheckBoxAnswer/
| | └─ CheckBoxAnswer.jsx
| | └─ CheckBoxAnswer.style.js
| └─ ExerciseSolution/
| | └─ ExerciseSolution.jsx
| | └─ ExerciseSolution.style.js
| └─ Header/
| | └─ Header.jsx
| | └─ Header.style.js
| └─ NavHeader/
| | └─ NavHeader.jsx
| | └─ NavHeader.style.js
| └─ .../
└─ data/
  └─ exercises.json
  └─ options.json
  └─ solutions.json
  └─ theory.json
  └─ topics.json
└─ pages/
  └─ CopyRight/
    └─ CopyRight.jsx
    └─ CopyRight.style.js
  └─ Exercise/
    └─ Exercise.jsx
```

```

| | └─ Exercise.style.js
| └─ Home/
| | └─ Home.jsx
| | └─ Home.style.js
| └─ Question/
| | └─ Question.jsx
| | └─ Question.style.js
| └─ Solution/
| | └─ Solution.jsx
| | └─ Solution.style.js
| └─ Theory/
| | └─ Theory.jsx
| | └─ Theory.style.js
└─ App.js

```

Requerimientos de instalación

Dado que la aplicación fue desarrollada con React Native y Expo, es importante considerar los requisitos mínimos para Android.

- 1) **Sistema operativo:** Android 5.0 (Lollipop) o superior.
- 2) **Arquitectura:** ARM64 o x86_64 (para compatibilidad con emuladores y dispositivos modernos).
- 3) **RAM:** Mínimo 2 GB, recomendado 4 GB o más.
- 4) **Almacenamiento:** Al menos 50 MB a 100 MB de espacio libre (varía según la aplicación).
- 5) **Procesador:** Snapdragon 450 o superior (o equivalente en otras marcas).

Se menciona en el anexo la implementación de las pantallas y el código fuente de la aplicación de Álgebra Lineal.

A continuación, se describen las pruebas realizadas.

Pruebas

Pruebas unitarias, se realizó la verificación de cada uno de los módulos de los fragmentos de código, verificando que se asegurarán los datos y conservarán su integridad. También se vio cómo manejar los errores y su corrección. Siguiendo la estructura AAA (Arrange, Act, Assert) para organizar, accionar y comprobar el código.

Pruebas de integración, en estas pruebas se observó la interacción entre los módulos para ver su correcto funcionamiento de acuerdo con el diseño para la corrección de errores.

Pruebas de validación, durante esta prueba se valida el funcionamiento específico de cada sprint con los requerimientos determinados. Se hizo la presentación de los sprints entregables por parte del equipo Scrum. Se hizo la evaluación de los entregables y se notificaron los detalles y se hizo la realimentación.

El implementar este tipo de pruebas permitió detectar errores con prontitud y facilitar la integración de los sprint y la entrega del producto final que fue la aplicación móvil.

Resultados

Se realizó una investigación de tipo descriptiva con un diseño cuasi-experimental, con una muestra de 16 estudiantes de un grupo de 50 estudiantes de la asignatura de Álgebra Lineal que estudian las distintas carreras de Ingeniería a nivel superior y los cuales utilizaron la aplicación durante y fuera de la clase. Se aplicó un instrumento de evaluación como cuestionario de 15 ítems para medir el aprendizaje móvil (m-learning), se utilizó la encuesta como técnica de recopilación de información el cual fue realizado a través de online. Se realizó estadística descriptiva.

A continuación, se presentan los resultados de la técnica de evaluación de interrogatorio, la cual fue una encuesta que tiene elementos cuantitativos y cualitativos, el instrumento de evaluación utilizado fue un cuestionario. Se muestran las preguntas y algunos resultados se describen los gráficos o se dan las opiniones.

1.- La pregunta fue: Si les ayudó a repasar los conceptos vistos en clase.

Los estudiantes opinaron que les apoya a su aprendizaje y estudio, a repasar los conceptos vistos en clase, aprender más sobre la asignatura, es necesaria para la asignatura y para ejercitar lo aprendido.

El análisis de las respuestas de los estudiantes de Ingeniería sobre su uso de una aplicación móvil en el contexto del aprendizaje de Cálculo y Álgebra Lineal revela varios puntos clave y tendencias en sus experiencias y motivaciones como: Recomendaciones de los docentes, apoyo en el aprendizaje, práctica y ejercicios, mejora las habilidades, entre otras.

2.- ¿Te ha ayudado la aplicación a cumplir tu objetivo de aprendizaje de la asignatura?

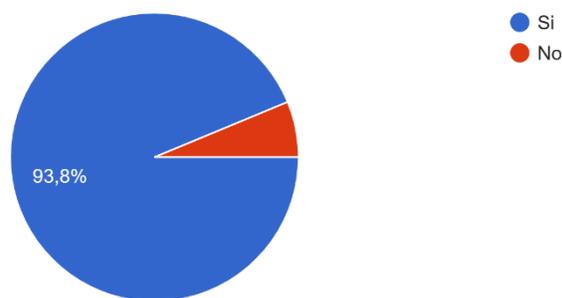


Figura 13 Te ayudado a cumplir el objetivo de aprendizaje de la asignatura.

Figura 13. Te ayudado a cumplir el objetivo de aprendizaje de la asignatura.

El 93.8% opina que si le ha ayudado.

3.- ¿Cómo calificaría el aspecto y la sensación de la aplicación y la sensación de la aplicación?

El 43.8% la observa regular, el 31.3 % bueno y el 25% excelente, por lo que se está mejorando la visualización de la aplicación.

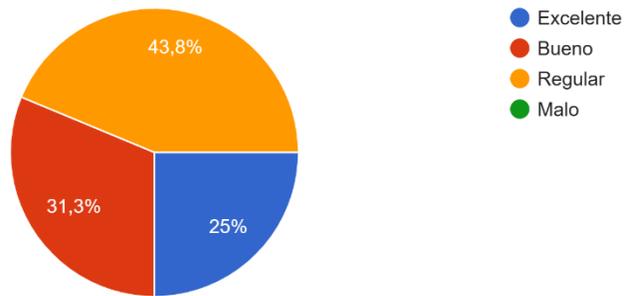


Figura 14 Aspecto y sensación de la aplicación móvil.

4.- ¿La aplicación te facilitó el aprendizaje de la asignatura?

El 100% opinó que sí le facilitó el aprendizaje del Álgebra Lineal.

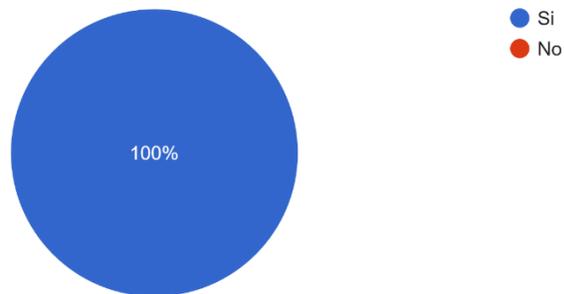


Figura 15 Facilidad del aprendizaje de la asignatura.

5.- ¿Qué grado de satisfacción tiene con la velocidad de carga de la aplicación?

Los estudiantes mostraron que:

El 25% estuvo satisfecho.

El 18.8% fue buena la velocidad.

El 31.3% menciona que es regular.

El 25% menciona que es mala.

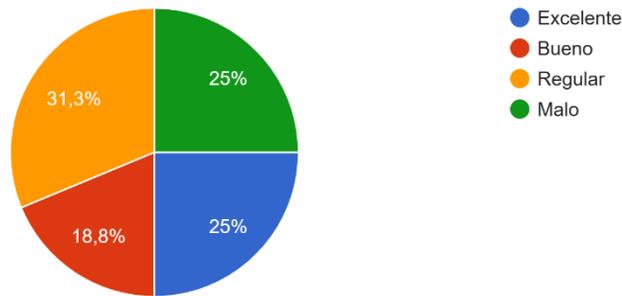


Figura 16 Satisfacción con la velocidad de carga.

6.- ¿Cuáles son las tres características de la aplicación que más te gustan y encuentras más útiles?

A partir del análisis de las respuestas de los estudiantes, se pueden identificar las siguientes tres características de la aplicación que más les gustan y encuentran útiles:

1. Accesibilidad y Portabilidad:

- Muchos estudiantes destacan la facilidad de acceso a la aplicación desde dispositivos móviles, lo que les permite consultar información y realizar ejercicios en cualquier momento y lugar. Esta portabilidad es altamente valorada, ya que facilita el estudio en diferentes contextos.

2. Teoría y Explicaciones Detalladas:

- La aplicación ofrece una sólida base teórica, con explicaciones de conceptos y teoremas que son claras y comprensibles. Los estudiantes aprecian que la teoría esté bien estructurada y presentada, lo que les ayuda a comprender mejor los temas antes de enfrentarse a los ejercicios.

3. Ejercicios y Soluciones Detalladas:

- Los estudiantes valoran la inclusión de ejercicios prácticos y la posibilidad de ver soluciones detalladas. Esta característica les permite practicar lo aprendido y comparar sus respuestas con el desarrollo que ofrece la aplicación, lo que contribuye a un aprendizaje más efectivo y a la consolidación de conceptos.

7.- ¿Con qué frecuencia utilizas la aplicación móvil?

Los estudiantes opinaron
50% la utiliza en clase
50% la utiliza una vez a la semana.

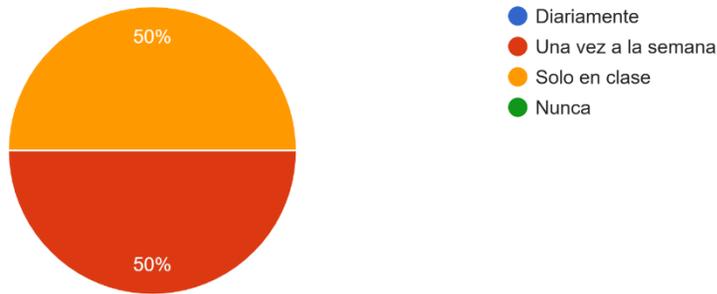


Figura 17 Utilización de la aplicación móvil.

8.- ¿Recomendarías que se realizará otra aplicación de este tipo para otra asignatura?
El 100% opina que sí la necesita.

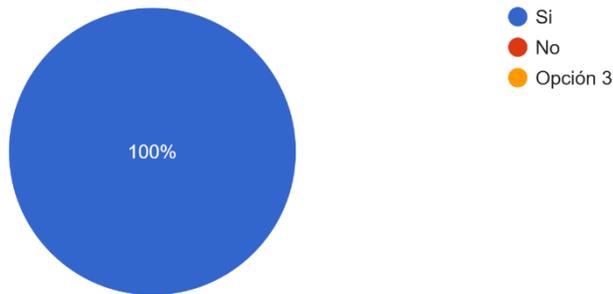


Figura 18 Recomendación para realizar otra aplicación para otra asignatura.

9.- ¿Qué le falta a la aplicación?

Videos, podcast, interactivos
disponibilidad en iOS
Optimización para celulares que tienen pocos recursos en memoria

10.- ¿En la comprensión de conceptos como te ayuda la aplicación?

El análisis de las respuestas de los estudiantes sobre la relación entre la comprensión de conceptos en Álgebra Lineal y su aplicación revela varios patrones y temas recurrentes. Aquí se desglosan los aspectos más destacados: Conexión entre la teoría y la práctica, realimentación, claridad en la explicación, refuerzo al aprendizaje, uso de definiciones, ayuda a identificar errores y a memorizar.

11.- ¿Te ha apoyado la aplicación para reforzar los temas?

El 100% de los estudiantes le ayudó a reforzar los temas.

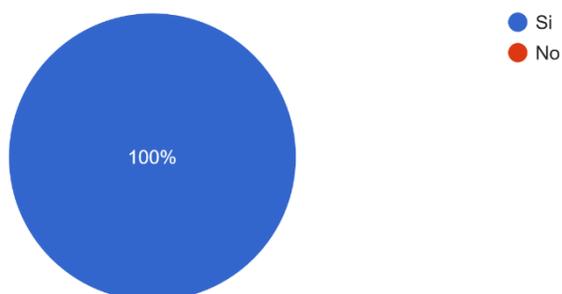


Figura 19 Reforzamiento de los temas.

12.- ¿Se ve favorecido el aprendizaje de la asignatura?

El 100% de los estudiantes opina que si le ha favorecido la aplicación móvil a su aprendizaje.

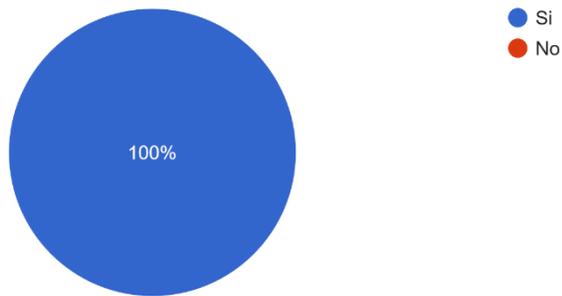


Figura 20 Aprendizaje de la asignatura.

13.- Consideras que puedes tener la aplicación instalada para estudiar más adelante otras asignaturas.

El 93.8% de los estudiantes considera que la aplicación les servirá para estudiar otras asignaturas.

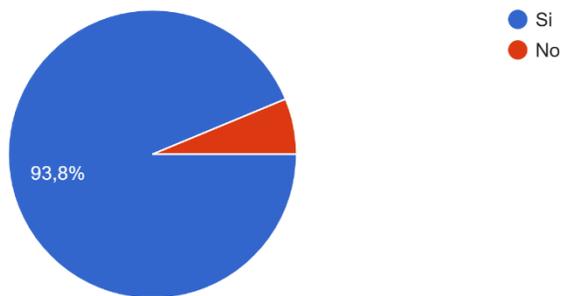


Figura 21 Instalación de la aplicación.

14.- ¿Consideras que tener la aplicación en tu celular, la puedes consultar a cualquier hora y en cualquier lugar?

El 100% considera que tiene ubicuidad y portabilidad la aplicación.

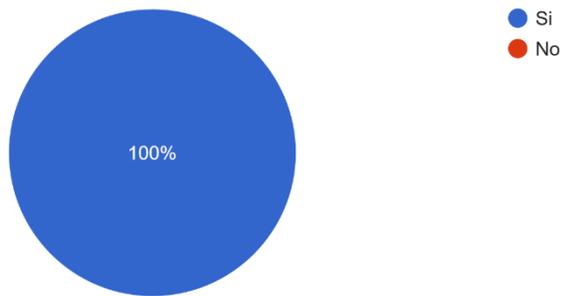


Figura 22 Ubicuidad y portabilidad de la aplicación.

15.- ¿Te causa alguna novedad el estudiar en los dispositivos móviles?

Para el 75% de los estudiantes es novedoso estudiar en este tipo de dispositivos.

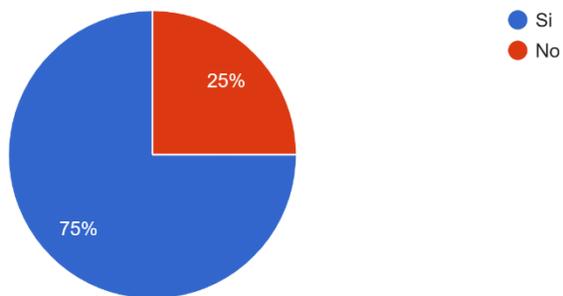


Figura 23 Novedad para estudiar.

Conclusiones

- 1. La alfabetización digital** ha ganado terreno en múltiples instituciones educativas, facilitando la integración significativa de las Tecnologías de la Información y la Comunicación (TIC) en los procesos formativos. En este contexto, el **aprendizaje móvil (m-learning)** emerge como una herramienta clave para reducir la brecha digital, gracias a su portabilidad, accesibilidad, ubicuidad y la disponibilidad de aplicaciones gratuitas y de libre acceso.
- 2.** La aplicación móvil diseñada e implementada permitió **realizar ejercicios interactivos con retroalimentación inmediata**, los cuales fueron bien recibidos por los estudiantes. Este tipo de actividades promueve el compromiso, la autonomía y la mejora en el rendimiento académico.
- 3. Tanto docentes como estudiantes se benefician del uso de esta tecnología:** los docentes cuentan con nuevas herramientas didácticas para facilitar el proceso de enseñanza, mientras que los estudiantes acceden a contenidos digitales complementarios que fortalecen lo aprendido en clase, favoreciendo la autoevaluación y el aprendizaje continuo.
- 4.** Los profesores han comenzado a **diseñar estrategias pedagógicas específicas** para integrar la aplicación móvil en sus prácticas, alineando sus actividades con los objetivos de aprendizaje de la asignatura, en este caso, Álgebra Lineal.
- 5.** En la Facultad de Ingeniería existen ya recursos como libros de texto y plataformas educativas. No obstante, la **incorporación de esta aplicación móvil complementa el ecosistema educativo existente**, promoviendo tanto el aprendizaje autónomo como el colaborativo. Además, brinda un acceso más ágil y flexible a contenidos y ejercicios que refuerzan conceptos clave.
- 6.** Esta implementación demuestra que **la tecnología móvil no solo mejora la accesibilidad al conocimiento, sino que también transforma el rol del estudiante**, quien puede ahora gestionar su propio aprendizaje de forma más activa, y el docente, quien se convierte en facilitador y guía del proceso educativo.

Referencias

- Lasa Gómez, C., Álvarez García, A., & de Las Heras del Dedo, R. (2017). *Metodos Ágiles Scrum, Kanban, Lean*. Anaya.
- Leffingwell, D. (2019). *SAFE 5.0 Distilled: Achieving Business Agility with the Scaled Agile Framework*. Addison-Wesley Professional.
- Loachamín Iza, H. D., Vargas Chavarrea, Á. P., Andrade Villarreal, J. V., & Puente Ponce, P. F. (2023). Enseñanza, aprendizaje y enfoque de la matemática en la ingeniería. *AlfaPublicacione*, 5(3.2), 6-20. doi:<https://doi.org/10.33262/ap.v5i3.2.400>
- Coordinación de Evaluación Educativa, Facultad de Ingeniería, UNAM. (2019). *Cuestionario para profesores: Informe de resultados del semestre 2019-2 Facultad de Ingeniería*. Obtenido de https://www.ingenieria.unam.mx/evaluacioneducativa/pdf/FI_Encuesta%20profesores%202019-2.pdf
- Lopez, M., & García, A. (2020). Impacto de las aplicaciones móviles en la enseñanza de matemáticas universitarias. *Revista de Educación Matemática*, 34(2), 55-68.
- Álvarez Álvarez, E., & Jiménez Ruiz, L. K. (2022). Aprendizaje móvil mediado por apps: Impacto para la innovación en ambientes educativos en América Latina. *Revista de Investigación en Ciencias de la Educación, Horizontes*, 6(26), 2265–2278. doi:<https://doi.org/10.33996/revistahorizontes.v6i26.490>
- Academia Lab. (2024). *Espacio vectorial*. Obtenido de <https://academia-lab.com/enciclopedia/espacio-vectorial/>
- Aguas-Díaz, C. J., Flores-Flores, J. L., Sarmiento-Sarmiento, I. K., & Aguirre-Robalino, D. F. (2020). Aprendizaje móvil (m-learning) como método educativo en Educación Superior. *Revista Arbitrada Interdisciplinaria Koinonía*, 5(1), 867-879. doi:<https://doi.org/10.35381/r.k.v5i1.887>
- Beck, K. (2004). *Extreme Programming Explained: Embrace Change (2nd ed.)*. Addison-Wesley Professional.
- Espinoza, F. E. (2022). La evaluación de los aprendizajes. *Conrado*, 18(85). doi:http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1990-86442022000200120&lng=es&tlng=es.
- Facultad de Ingeniería. (2016). *Plan de estudios de la carrera de Ingeniería en Computación*. Obtenido de https://www.ingenieria.unam.mx/programas_academicos/licenciatura/computacion.php
- García Contador, Y., & Gutiérrez-Esteban, P. (2020). El rol docente en la sociedad digital. *Digital Education Review*, 38. Obtenido de <https://revistes.ub.edu/index.php/der/article/view/27102>
- GeoGebra. (2025). *GeoGebra 3D Calculator [Aplicación móvil]*. Obtenido de GeoGebra GmbH: <https://www.geogebra.org>
- Gómez, C. L. (2017). *Métodos Ágiles Scrum, Kanban, Lean*.
- Gutiérrez González, E., & Ochoa García, S. (2014). *Álgebra Lineal y sus aplicaciones*. México: Grupo Editorial Patria.
- Hattie, J., & Timperley, H. (2007). The Power of Feedback. *Review of Educational Research*(77), 81-112. Obtenido de https://www.researchgate.net/publication/258182775_The_Power_of_Feedback

- Hinojo Lucena, F. J., Aznar Díaz, I., & Romero Rodríguez, J. M. (2018). Dispositivos móviles para el aprendizaje: análisis de la investigación doctoral sobre mobile learning en España. *Texto Livre, Belo Horizonte-MG, 11(3)*. doi:<https://doi.org/10.17851/1983-3652.11.3.154-175>
- Mayer, R. (2009). *Multimedia Learning*. Cambridge University Press. Obtenido de <https://drive.google.com/file/d/1UQKDYfqoNjHuPZp62t-yckVDa-VMRXYC/view>
- Osterwalder, A., & Pigneur, Y. (2010). *Business Model Generation: A Handbook for Visionaries, Game Changers, and Changers*. Wiley.
- Plan de Desarrollo 2023-2027*. (2023). Obtenido de https://www.ingenieria.unam.mx/planeacion/eg/PDD2023-2027/PDD2023-2027_re.pdf
- Pérez, D., & Ramírez, L. (2022). Evaluación del rendimiento académico en Álgebra Lineal mediante el uso de aplicaciones móviles.
- Pressman, R. S. (2010). *Ingeniería del software: Una enfoque práctico: (7ª ed.)*. McGraw-Hill.
- Programa de Álgebra Lineal*. (2024). Obtenido de <https://dcb.ingenieria.unam.mx/index.php/coordinaciones/maticas/algebra-lineal/>
- Red Educa. (2023). *Evaluación*. Obtenido de <https://www.rededuca.net/contexto-educativo/e/evaluacion>
- Reyes Chávez, R., & Medina Díaz, J. H. (2015). *Aprendizaje móvil. Enfoque pedagógico con el uso de TIC*. (P. UAT, Ed.) Obtenido de <https://centrodeinvestigacioneducativauatx.org/publicacion/pdf2015/B222.pdf>
- Schwaber, K., & Sutherland, J. (2020). *La guía de Scrum: La guía definitiva de Scrum: Las reglas del juego (Traducción al español)*. Obtenido de Scrum.org: <https://scrumguides.org>
- Sánchez Mendiola, M., & Martínez González, A. (2022). *Evaluación y aprendizaje en educación universitaria: estrategias e instrumentos* (1a. ed.). (I. E. Universidad Nacional Autónoma de México. Coordinación de Universidad Abierta, Ed.) México. Obtenido de <https://cuaed.unam.mx/publicaciones/libro-evaluacion/pdf/ELibro-Evaluacion-y-Aprendizaje-en-Educacion-Universitaria-ISBN-9786073060714.pdf>
- Santos Dimaté, V. (2023). ¿Cómo el mobile learning está revolucionando la educación? *Revista educación virtual*. Obtenido de <https://revistaeducacionvirtual.com/archives/3869#:~:text=E1%20aprendizaje%20m%C3%B3vil%20o%20mobile%20learning>
- Secretaría de Apoyo a la Docencia. (2020). *Perfil de ingreso de los alumnos de la generación 2020*. Obtenido de https://www.ingenieria.unam.mx/evaluacioneducativa/pdf/Reporte_Sociodemográfico_2020G.pdf
- Shuler, C., Winters, N., & West, M. (2013). *El Futuro del aprendizaje móvil: implicaciones para la planificación y la formulación de políticas*. Obtenido de https://unesdoc.unesco.org/ark:/48223/pf0000219637_spa.locale=es
- Soubirón, E., & Camarano, S. (2006). *Pruebas objetivas*. Obtenido de <https://ice.ua.es/es/documentos/recursos/materiales/ev-pruebas-objetivas.pdf>
- Statista. (2024). *Number of smartphone users worldwide from 2014 to 2029*. Obtenido de <https://www.statista.com/forecasts/1143723/smartphone-users-in-the-world>
- Strang, G. (2003). *Linear Algebra and Its Applications* (Fourth Edition ed.). Brooks/Cole. Obtenido de <https://dn720003.ca.archive.org/0/items/linear-algebra-by-strang-4th-edition/linear%20algebra%20by%20strang%204%20th%20edition.pdf>

- UNESCO. (2023). *Tecnologías de la información y la comunicación (TICs) en la educación*. Obtenido de <https://learningportal.iiep.unesco.org/es/fichas-praticas/mejorar-el-aprendizaje/tecnologias-de-la-informacion-y-la-comunicacion-tics-en-la>
- UNESCO. (2023). *Technology in education: A TOOL ON WHOSE TERMS?* Obtenido de <https://gem-report-2023.unesco.org/es/tecnologia-en-la-educacion/>
- UNESCO. (2024). *Día internacional del aprendizaje digital*. Obtenido de <https://www.unesco.org/es/days/digital-learning>
- UNESCO. (2024). *Preguntas y respuestas: Por qué es esencial la educación digital para la ciudadanía mundial*. Obtenido de <https://www.unesco.org/es/articulos/preguntas-y-respuestas-por-que-es-esencial-la-educacion-digital-para-la-ciudadania-mundial?hub=103615>
- Velázquez Torres, J. (2003). Álgebra Lineal: La nueva matemática. En F. d. Ingeniería (Ed.), *La enseñanza de las matemáticas para ingenieros*, (págs. 165-168). México. Obtenido de <http://dcb.fi-c.unam.mx/Eventos/ForoMatematicas2/memorias/veintinueve.pdf>
- Villar-Sánchez, P., Arancibia-Carvajal, S., Robotham, H., & González, F. (2021). Factores que inciden en la actitud hacia el aprendizaje de las matemáticas en primer año de Ingeniería. *Revista Complutense de Educación*, 33(2), 337-349. doi:<https://dx.doi.org/10.5209/rced.74356>
- Zabala Vargas, S. A. (2022). *Tesis Doctoral. Estrategia de enseñanza con metodología de aprendizaje basado en el juego, para el mejoramiento del desempeño académico y la motivación de los estudiantes en cursos de matemáticas de primer año de ingeniería*. Obtenido de https://dspace.uib.es/xmlui/bitstream/handle/11201/159807/Sergio%20Andr%C3%A9s%20Zabala%20Vargas_TESIS%20DEF.pdf?sequence=1
- Zamora Delgado, R. I. (2020). Las ventajas de la utilización de dispositivos móviles en el proceso de aprendizaje en la educación básica. 5(1), 82-91. Obtenido de <https://www.redalyc.org/articulo.oa?id=673171024008>

Anexo

Manual de usuario

Inicio de la aplicación

1. Ubica el ícono de la aplicación

En la pantalla principal del dispositivo móvil, busca el ícono de la aplicación. Este estará identificado con el logotipo y el nombre de la aplicación, como se muestra a continuación:

Ícono de aplicación

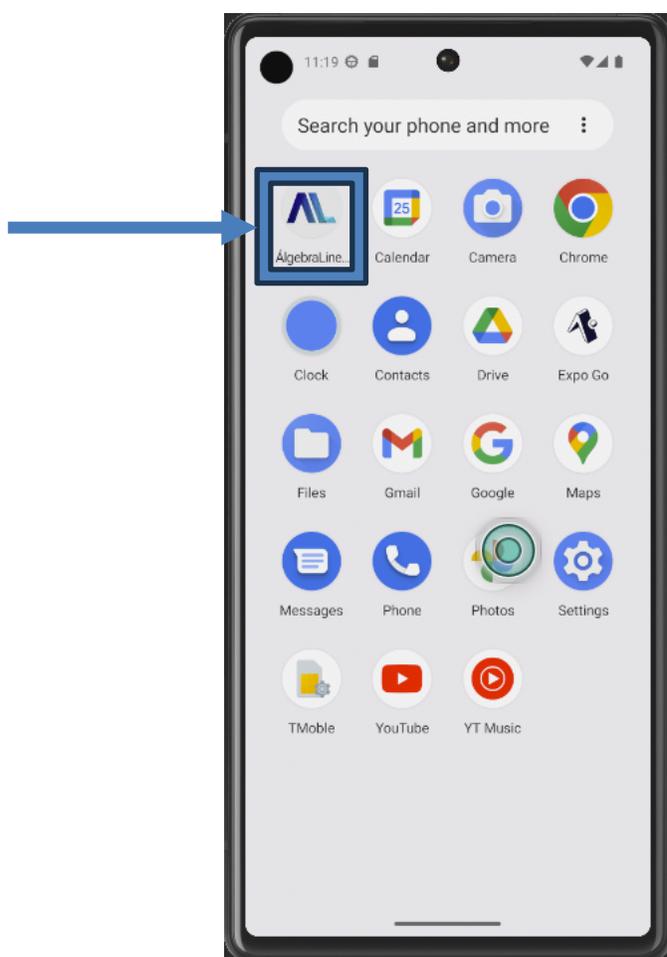


Figura 24 Ícono de la aplicación.

2. Toca el ícono para abrir

Pulsa una vez sobre el ícono con el dedo. Esto iniciará la aplicación y cargará la pantalla principal.

3. **Pantalla de carga**

La aplicación tiene una pantalla de carga , espere unos segundos mientras se prepara como se muestra a continuación:



Figura 25 Pantalla de carga.

1. **Acceso a la pantalla principal**

Una vez completada la carga inicial, se mostrará la pantalla principal con las opciones disponibles como se muestra a continuación:



Figura 26 Pantalla principal.

Partes de la pantalla principal

a) Encabezado

El encabezado está ubicado en la parte superior de la pantalla. Contiene herramientas esenciales para la navegación y la identificación de la aplicación.

Elementos destacados:

- **Logotipo:** Representa la aplicación y, debajo del logotipo se encuentra el título de la sección actual.
- **Botón para ver créditos:** Accede a la pantalla con los créditos asociados a la aplicación.
- **Botón informativo:** Muestra la fecha de la última actualización.

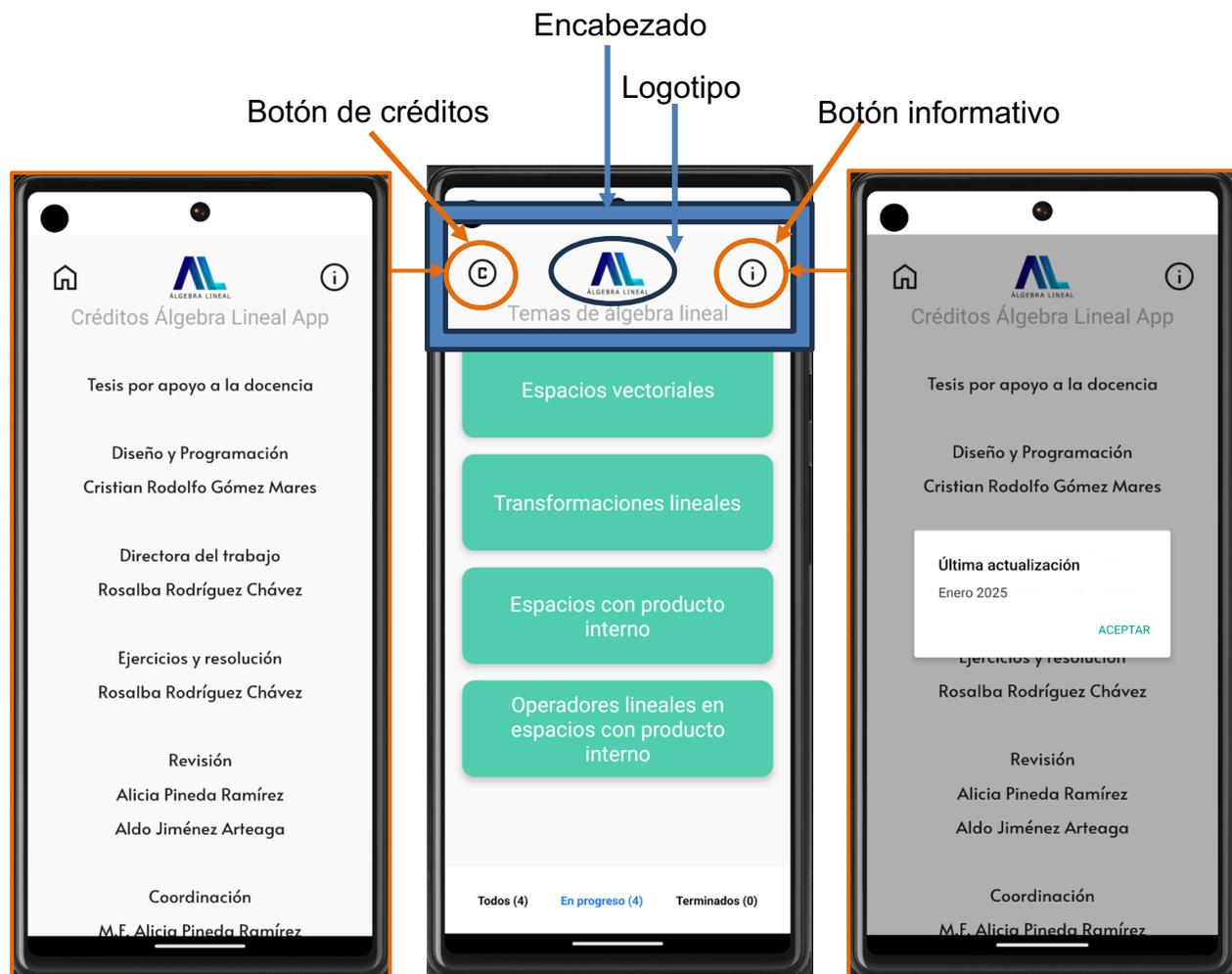


Figura 27 Encabezado de la pantalla principal.

b) Cuerpo

El cuerpo ocupa la mayor parte de la pantalla y muestra el contenido principal.

Elementos destacados:

- **Botones con los temas de álgebra lineal:** Al pulsar una vez el botón con el tema de interés se muestra una nueva pantalla con los ejercicios asociados a dicho tema.

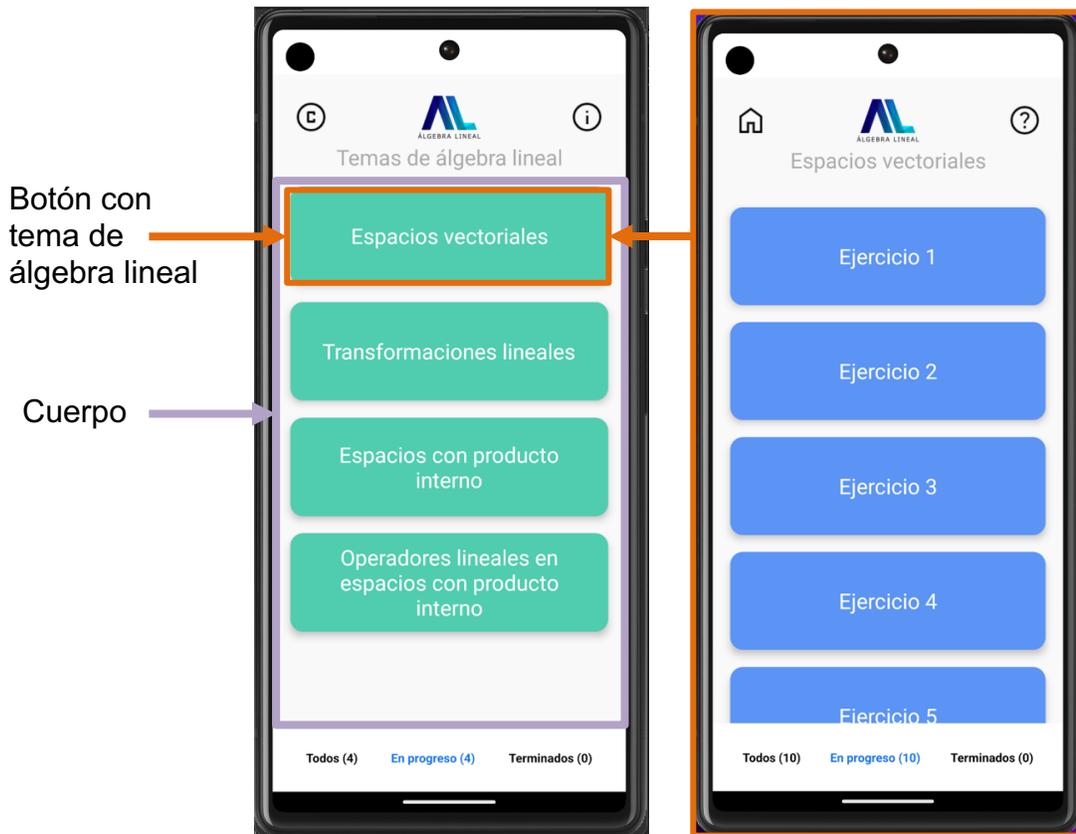


Figura 28 Cuerpo de la pantalla principal.

c) Pie

El pie está en la parte inferior de la pantalla y contiene accesos rápidos a secciones con los temas terminados y en progreso.

Incluye:

- Indicadores de estado, como el progreso de los temas.

“Por ejemplo, si el botón de alguno de los temas es pulsado por más de tres segundos este pasará a un estado terminado y se visualizará en la sección de *Terminados* como se muestra a continuación”.

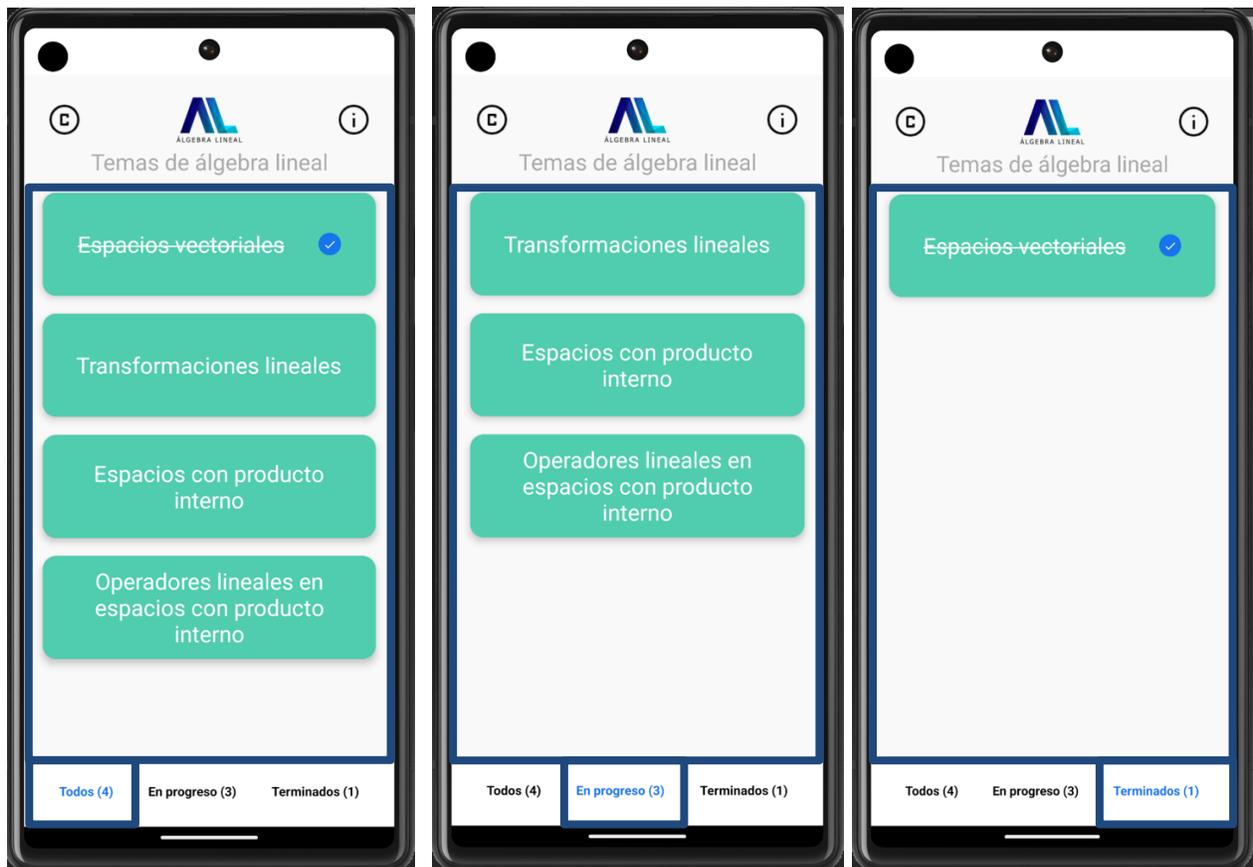


Figura 29 Pie de la pantalla principal con secciones.

2. Acceso a pantalla de ejercicios

En la pantalla principal, localiza el tema de álgebra lineal de tu interés, como se muestra a continuación:

- Estos botones están ubicados en el cuerpo de la pantalla.
- Identifica el tema de tu preferencia como se ve en la siguiente imagen:

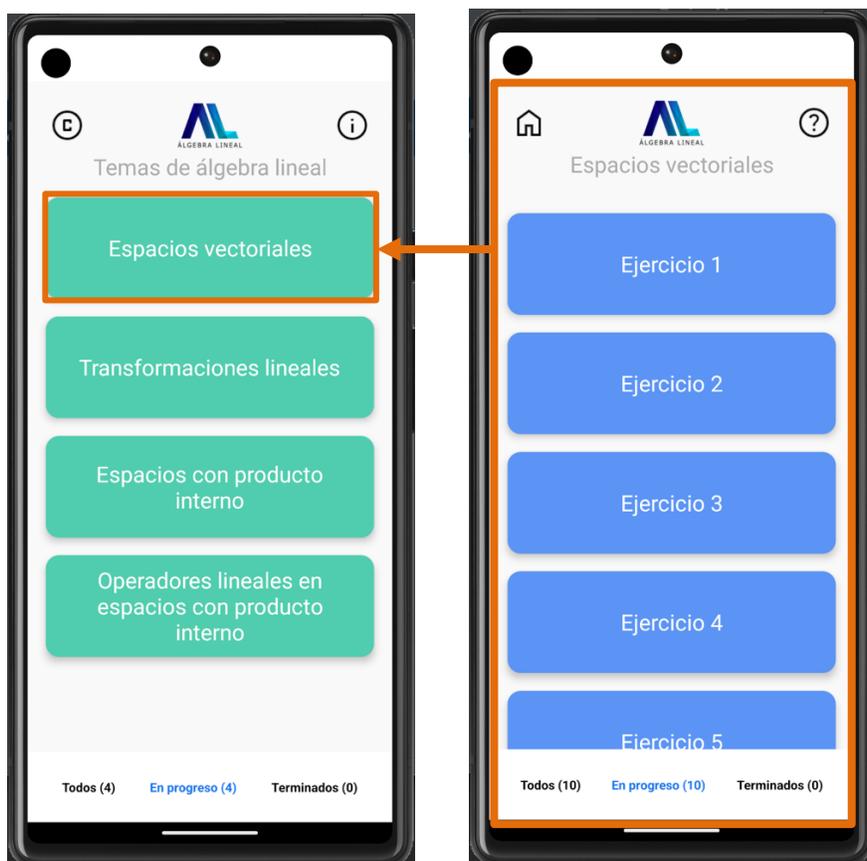


Figura 30 Pantalla secundaria con los ejercicios del tema seleccionado.

La pantalla con los ejercicios del tema seleccionado se divide en las siguientes partes:

- **Encabezado:** Incluye el título del tema seleccionado, logotipo de la aplicación, un botón para regresar a la pantalla principal y un botón de ayuda que muestra las instrucciones de la sección.
- **Cuerpo:** Aquí encontraras los ejercicios asociados al tema seleccionado.
- **Pie:** Contiene accesos rápidos a secciones con los ejercicios terminados y en progreso.

A continuación, se muestran las imágenes asociadas a cada una de las partes contenidas en la pantalla de ejercicios:

a) Encabezado

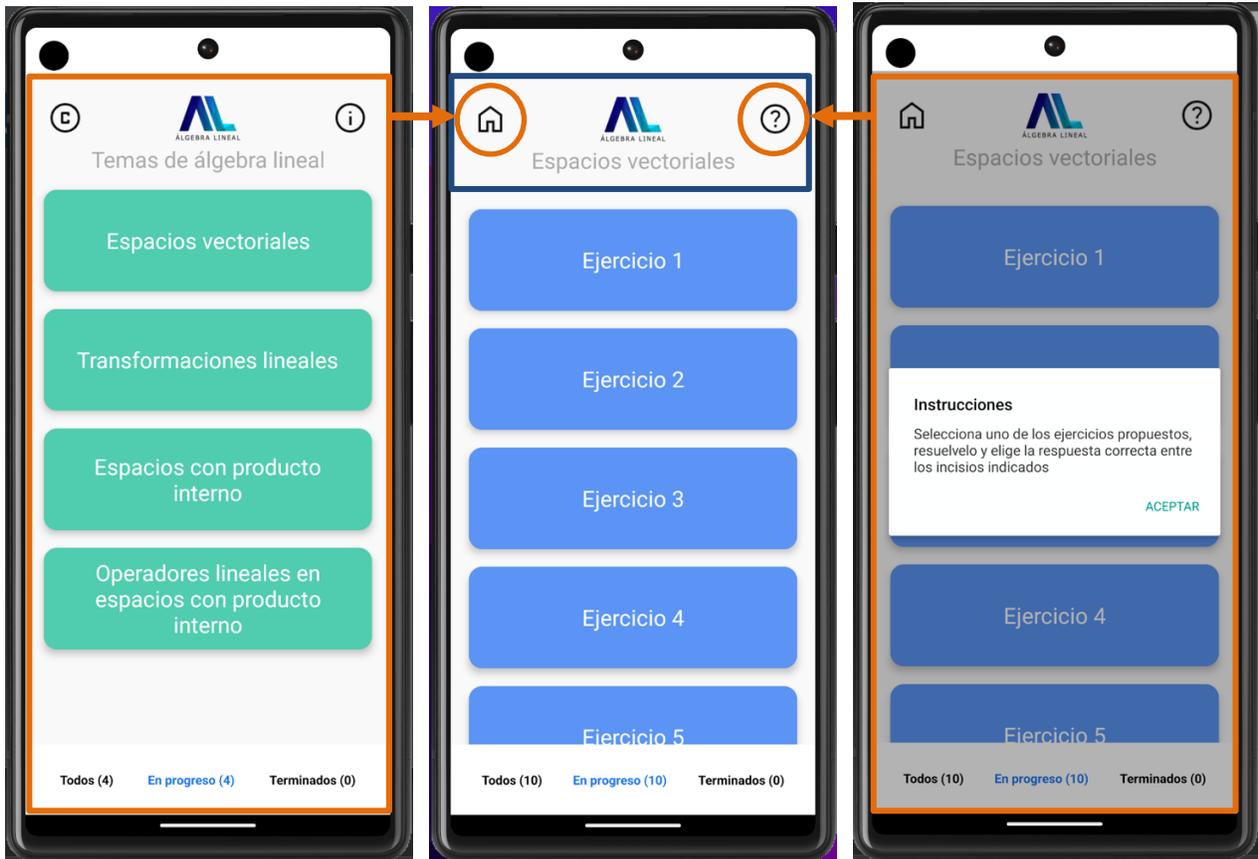


Figura 31 Encabezado de la pantalla de ejercicios.

b) Cuerpo

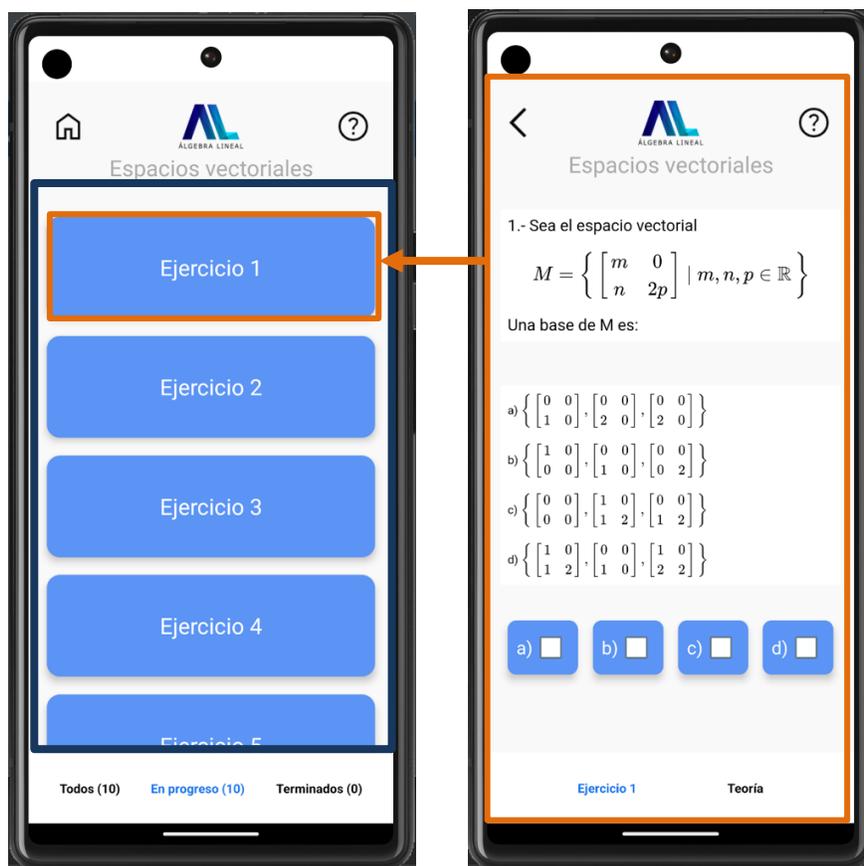


Figura 32 Cuerpo de la pantalla de ejercicios.

c) Pie

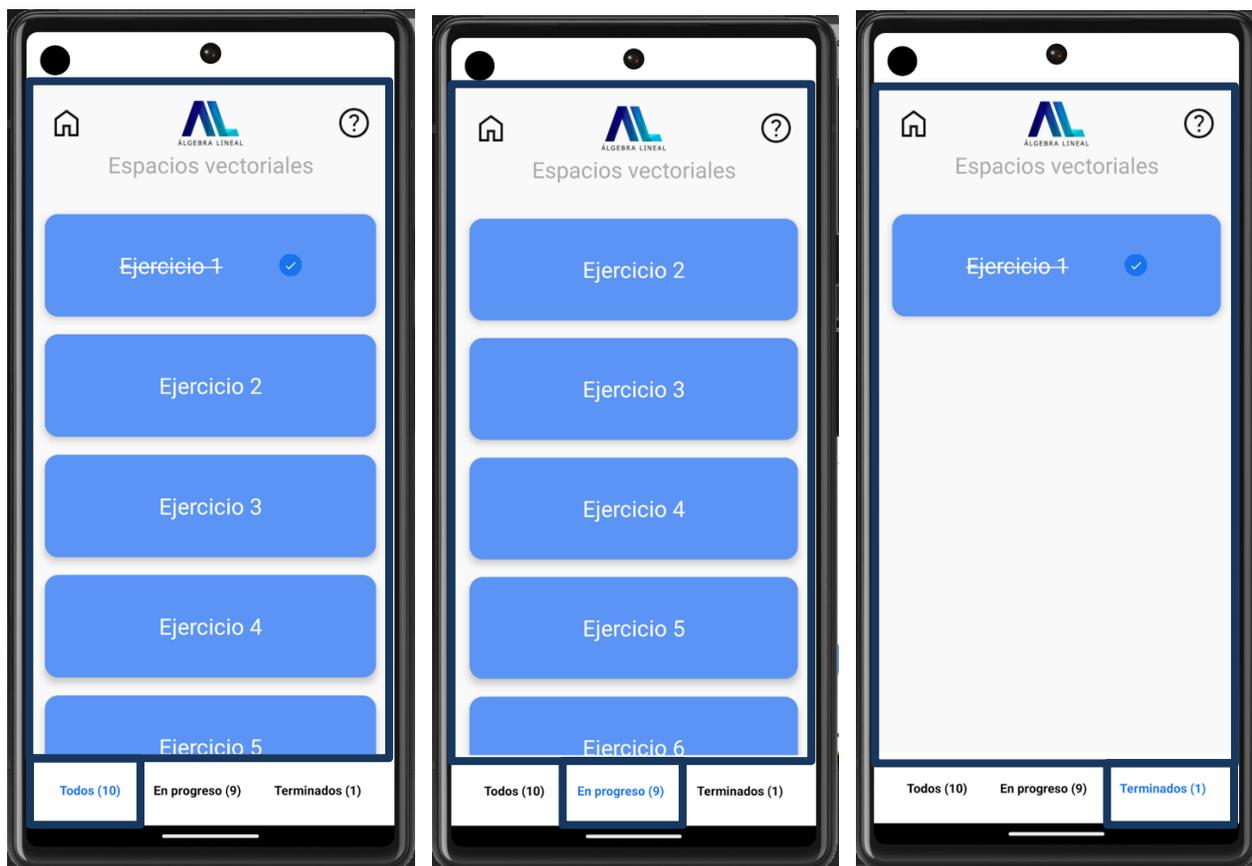


Figura 33 Pie de la pantalla de ejercicios con las secciones respectivas.

3. Acceso a pantalla de pregunta y respuestas

En la pantalla donde se muestran los ejercicios, selecciona el número de ejercicio que quieres resolver, como se muestra a continuación:

- Los botones con los ejercicios numerados se encuentran en el cuerpo de la pantalla.
- Selecciona el ejercicio de tu preferencia.

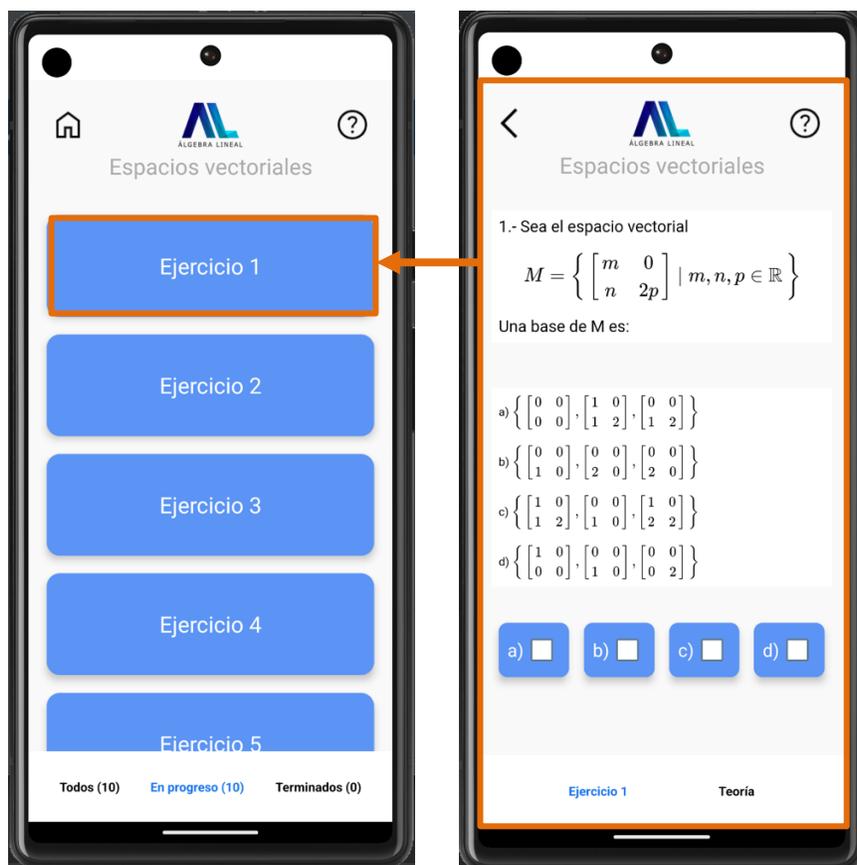


Figura 34 Pantalla de ejercicio con respuesta en incisos.

La pantalla que contiene el ejercicio y la respuesta correcta en uno de los incisos se divide en las siguientes partes:

- **Encabezado:** Incluye el título del tema seleccionado, logotipo de la aplicación, un botón para regresar a la pantalla anterior y un botón de ayuda que muestra las instrucciones de la sección.
- **Cuerpo:** Aquí encontraras el ejercicio, las posibles respuestas y los botones con los incisos para seleccionar la respuesta.
- **Pie:** Contiene acceso rápido a la teoría asociada al ejercicio y el ejercicio.

A continuación, se muestran las imágenes asociadas a cada una de las partes contenidas en la pantalla del ejercicio seleccionado.

a) Encabezado

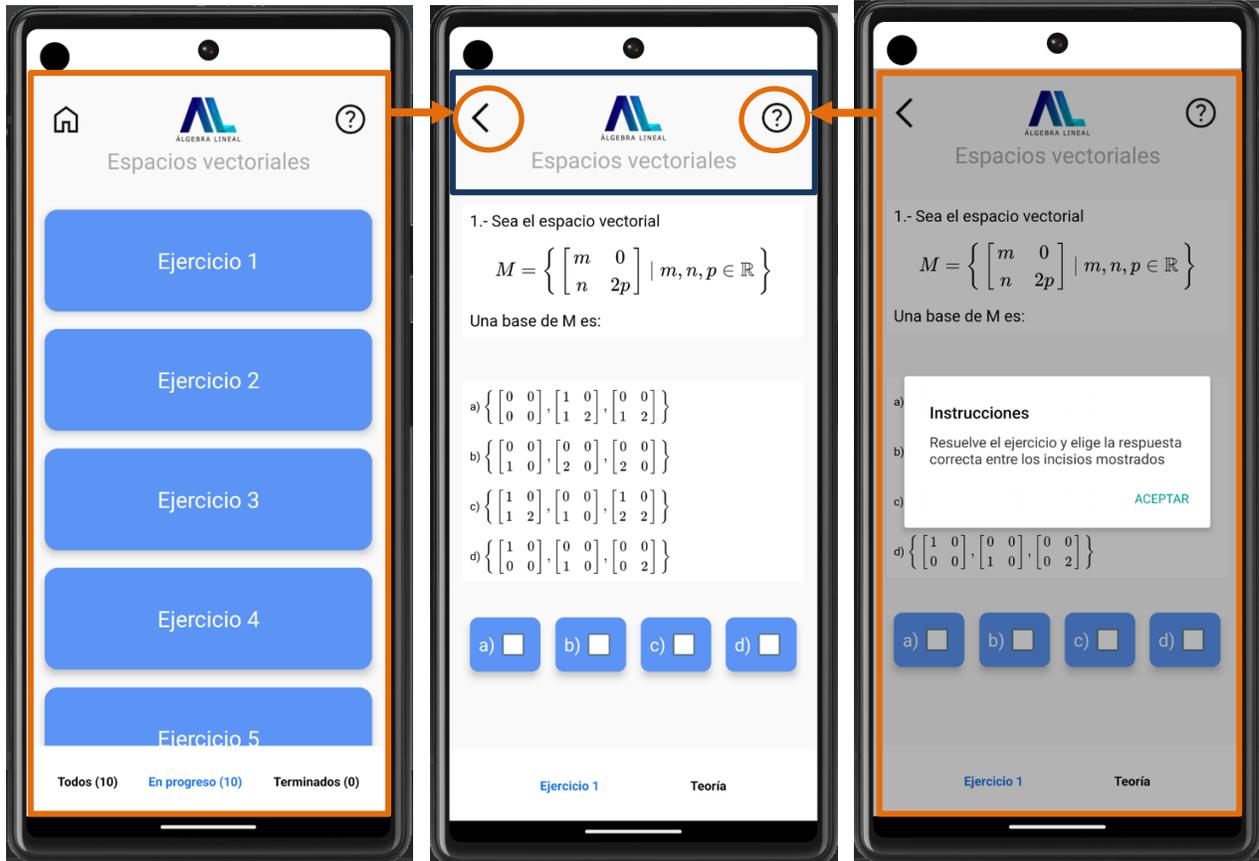


Figura 35 Encabezado en la pantalla del ejercicio.

b) **Cuerpo**

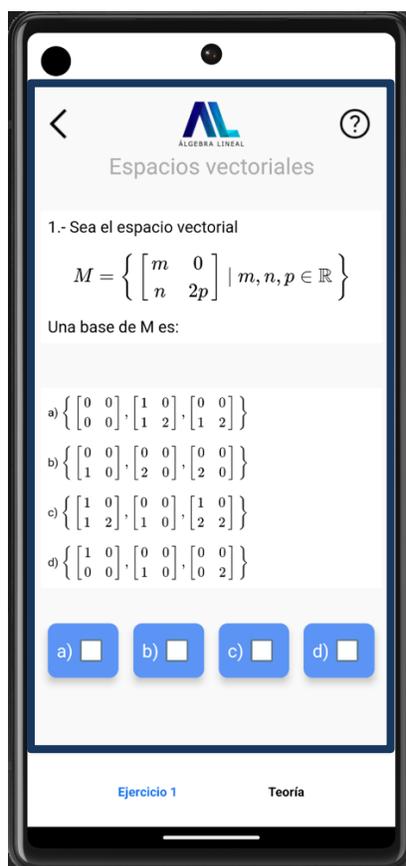


Figura 36 Cuerpo en la pantalla del ejercicio.

c) Pie



Figura 37 Pie en la pantalla del ejercicio.

4. Selección de respuesta

Para seleccionar la respuesta que consideres correcta, pulsa en la casilla del inciso correspondiente. Una vez realizado, aparecerá una pantalla con un mensaje indicando si la respuesta es correcta o incorrecta, como se muestra a continuación:

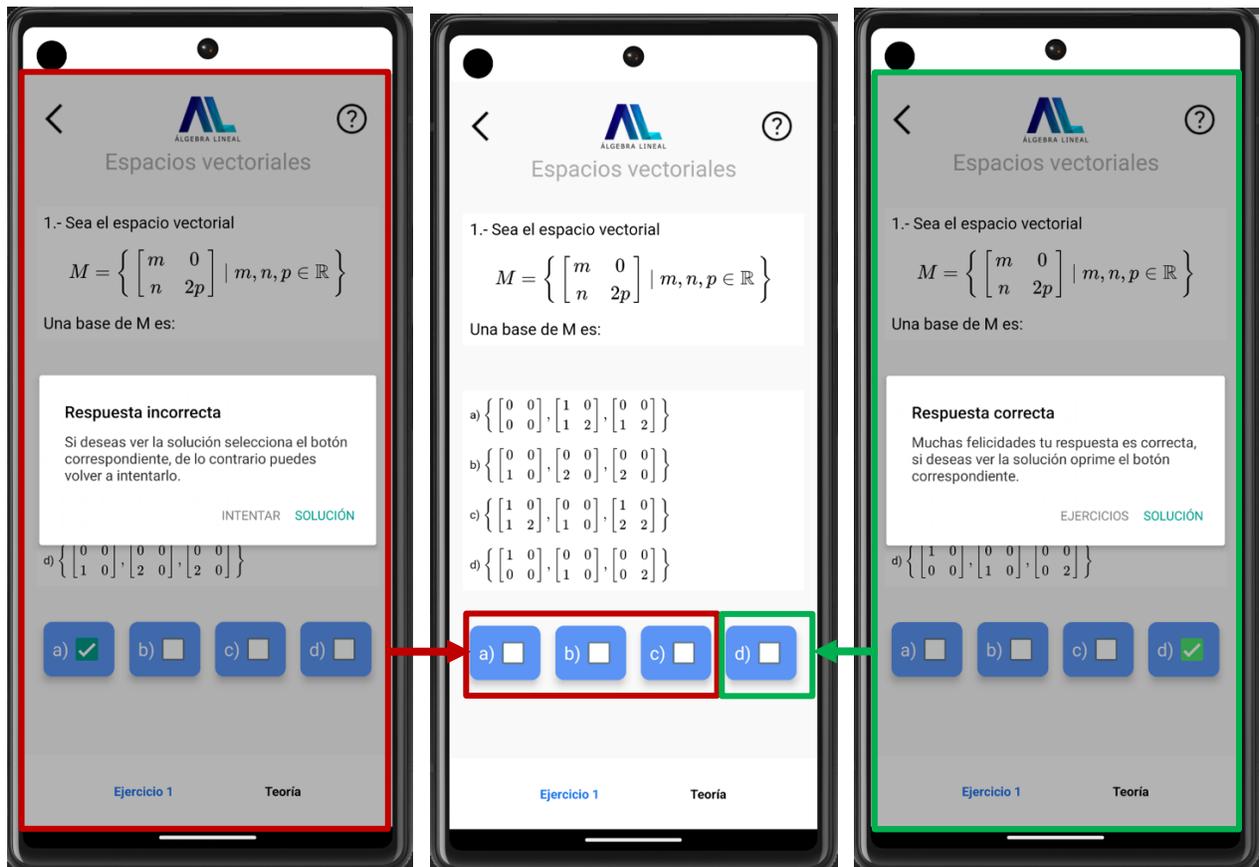


Figura 38 Mensajes cuando se selecciona la respuesta correcta e incorrecta en la pantalla del ejercicio.

Si la respuesta seleccionada es incorrecta, el mensaje de diálogo ofrecerá la opción de ver la solución o intentar nuevamente. Del mismo modo, si la respuesta es correcta, podrás elegir entre ver la solución o regresar a la pantalla de ejercicios para seleccionar uno nuevo, como se muestra a continuación:

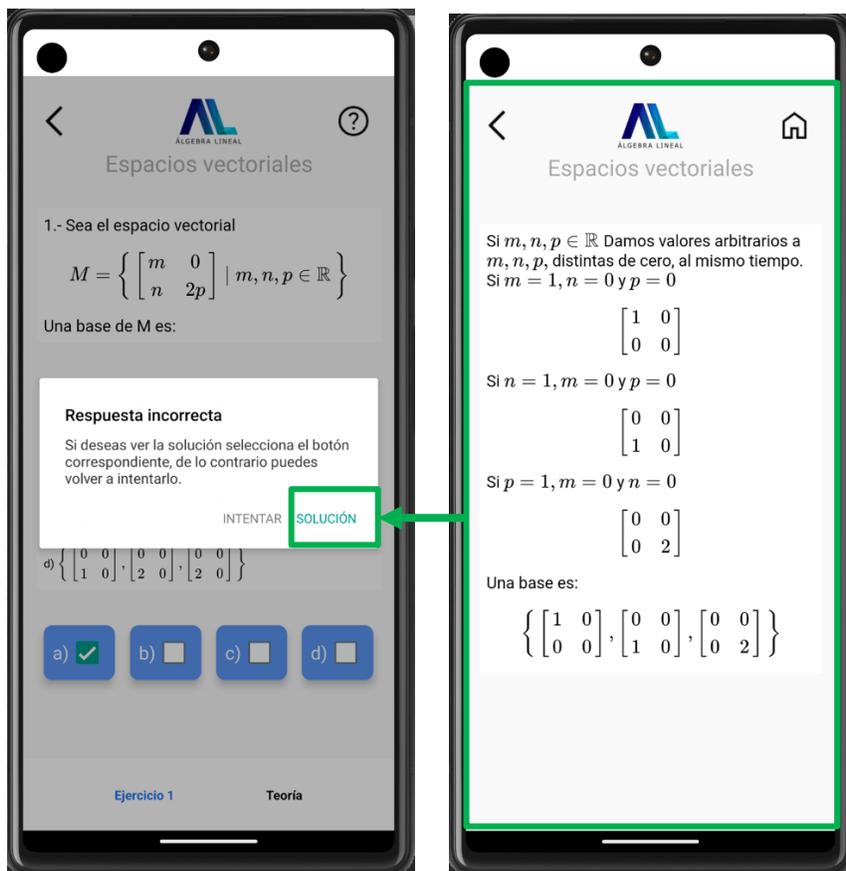


Figura 39 Visualizar la solución mediante el cuadro de diálogo de la respuesta incorrecta.

Código fuente

Estructura del proyecto:

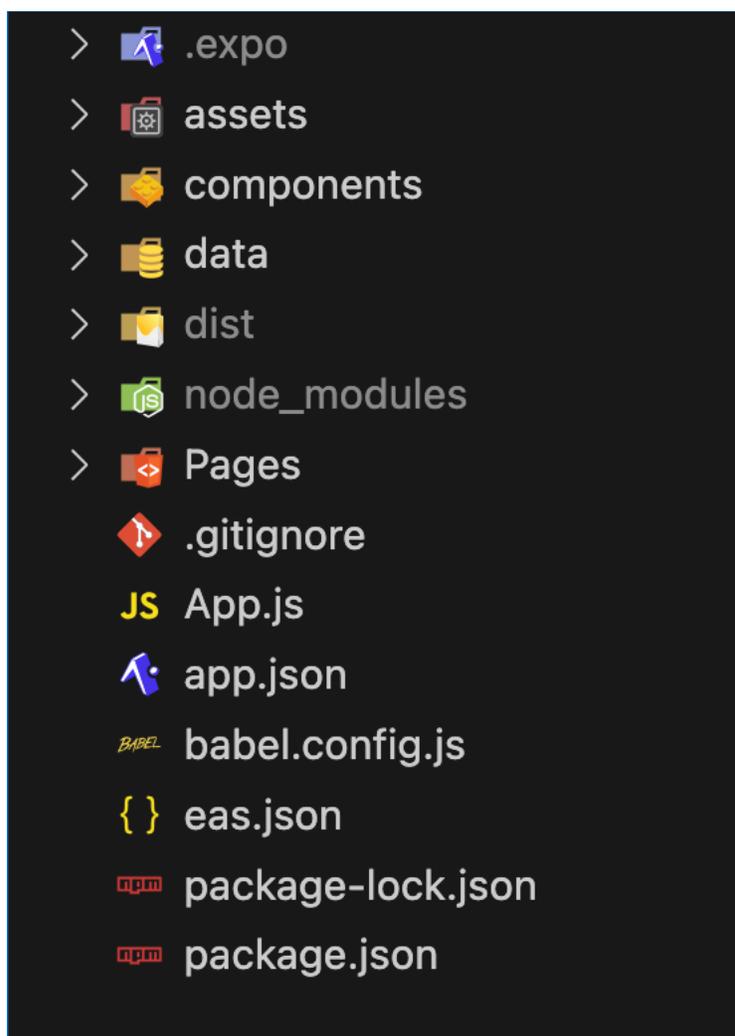


Ilustración 1 Estructura del proyecto.

Componentes del proyecto:

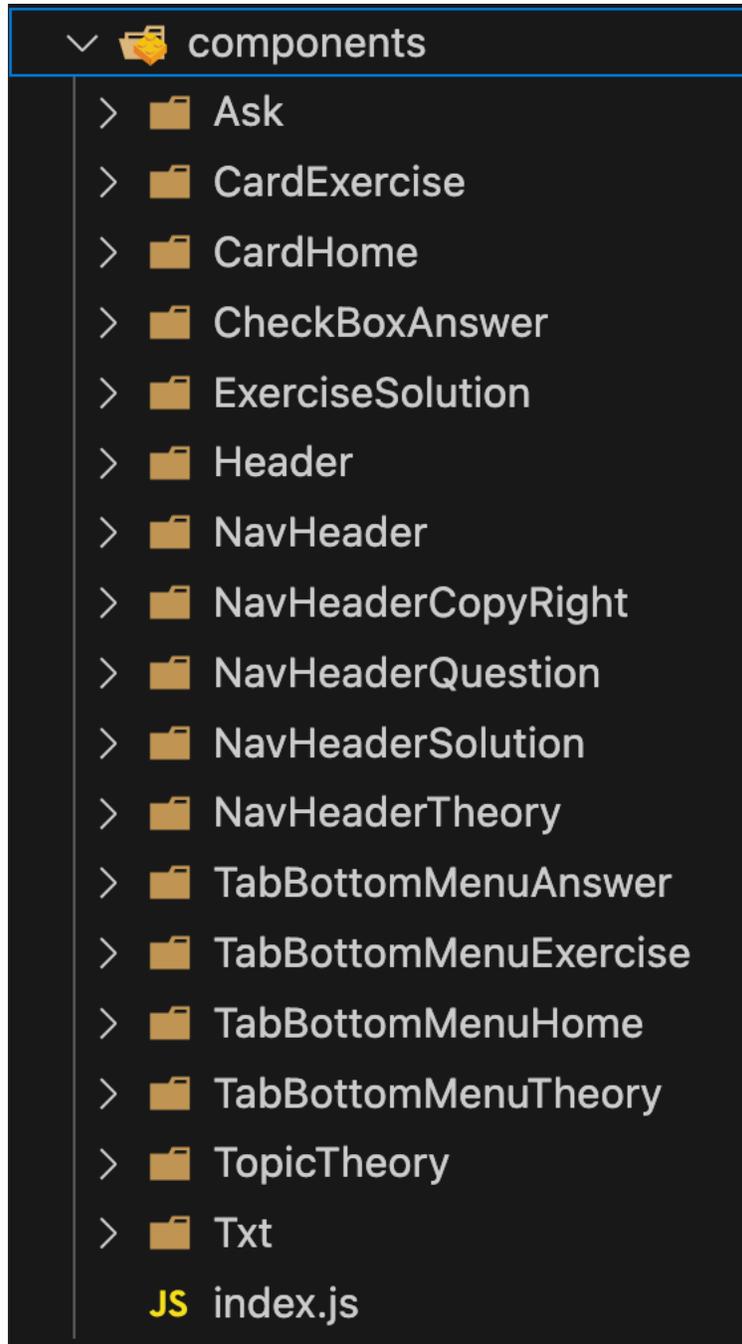


Ilustración 2 Componentes utilizados.

Componente para mostrar el enunciado de los ejercicios:

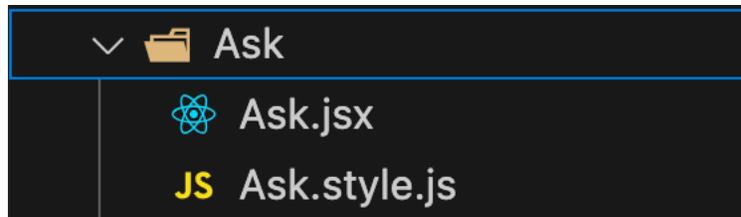


Ilustración 3 Componente Ask.

```
import { View } from "react-native";
import { s } from "./Ask.style";
import MathJax from "react-native-mathjax";

const mmlOptions = {
  messageStyle: "none",
  extensions: ["tex2jax.js"],
  jax: ["input/TeX", "output/HTML-CSS"],
  tex2jax: {
    inlineMath: [
      ["$", "$"],
      ["\\(", "\\)"],
    ],
    displayMath: [
      ["$$", "$$"],
      ["\\[", "\\]"],
    ],
    processEscapes: true,
  },
  TeX: {
    extensions: [
      "AMSmath.js",
      "AMSsymbols.js",
      "noErrors.js",
      "noUndefined.js",
    ],
  },
};
```

```

    ],
  },
};

const Ask = ({ exerciseNum, statement }) => {
  return(
    <View style={s.container} >
      <MathJax
        mathJaxOptions={mmlOptions}
        html={`<font size=4>${exerciseNum}.-
${statement}</font>`}
      />
    </View>
  );
};

export default Ask;

import { StyleSheet } from "react-native";

export const s = StyleSheet.create({
  container: {
    paddingBottom:50,
  },
  statement: {
    fontSize: 25,
  }
});

```

Componente para seleccionar el ejercicio:

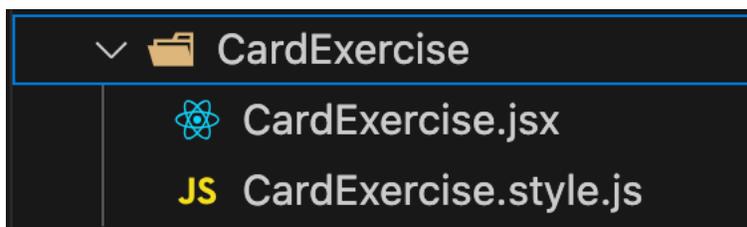


Ilustración 4 Componente CardExercise.

```
import { Image, Text, TouchableOpacity } from "react-native";
import { s } from "./CardExercise.style";
import checkImg from "../../assets/check.png";
import { useNavigation } from "@react-navigation/native";
```

```
const CardExercise = ({ exercise, onLongPress }) => {
  const nav = useNavigation();

  return(
    <TouchableOpacity style={s.card} onPress={() =>
nav.navigate("Question", {...exercise}) } onLongPress={() =>
onLongPress(exercise)} >
      <Text style={[s.exercise, exercise.resolved &&
{textDecorationLine: "line-through"}]}>Ejercicio
{exercise.exercise}</Text>
      {exercise.resolved && <Image style={s.img}
source={checkImg} />}
    </TouchableOpacity>
  );
};
```

```
export default CardExercise;
```

```
import { StyleSheet } from "react-native";
```

```

export const s = StyleSheet.create({
  card: {
    backgroundColor: "#6495ED",
    width: "90%",
    height: 115,
    shadowColor: "#000",
    shadowOffset: {
      width: 0,
      height: 2,
    },
    shadowOpacity: 0.25,
    shadowRadius: 3.84,
    elevation: 5,
    borderRadius: 15,
    flexDirection: "row",
    alignItems: "center",
    justifyContent: "space-around",
    paddingHorizontal: 20,
  },
  exercise: {
    color: "white",
    fontSize: 25,
    textAlign: "center",
  },
  imgExercise: {
    width: 65,
    height: 65,
    borderRadius: 10,
  },
  img: {
    width: 25,
    height: 25,
  }
}

```

```
})
```

Componente para seleccionar el tema:

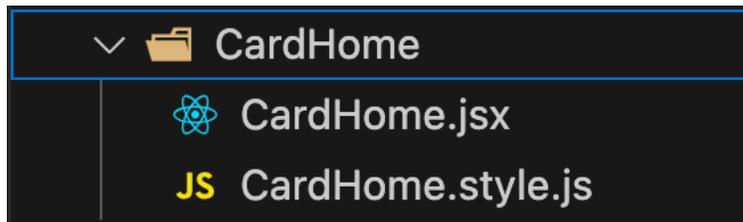


Ilustración 5 Componente CardHome.

```
import { TouchableOpacity, Text, Image } from "react-native";
import { s } from "../CardHome.style";
import { useNavigation } from "@react-navigation/native";
import checkImg from "../../assets/check.png";

/**
 *
 * @topic prop del componente que contiene un objeto con la
información del tema ej: {id:1, title: "Grupos y Campos",
isCompleted: false}
 * @onLongPress prop del componente para cambiar el estado
boolean del tema ej: {isCompleted: true}
 *
 */

const CardHome = ({ topic, onLongPress }) => {
  const nav = useNavigation();

  return(
    <TouchableOpacity onPress={() => nav.navigate("Exercise",
{...topic})} onLongPress={() => onLongPress(topic)}
style={s.card} >
```

```

        <Text style={[s.topic, topic.isCompleted &&
{textDecorationLine: "line-through"}]} >{topic.title}</Text>
        {topic.isCompleted && <Image style={s.img}
source={checkImg} />}
    </TouchableOpacity>
  );
};

```

```
export default CardHome;
```

```
import { StyleSheet } from "react-native";
```

```

export const s = StyleSheet.create({
  card: {
    backgroundColor: "#72CAAF",
    width: "90%",
    height: 115,
    shadowColor: "#000",
    shadowOffset: {
      width: 0,
      height: 2,
    },
    shadowOpacity: 0.25,
    shadowRadius: 3.84,
    elevation: 5,
    borderRadius: 15,
    flexDirection: "row",
    alignItems: "center",
    justifyContent: "space-around",
    paddingHorizontal: 20
  },
  topic: {
    color: "white",
    fontSize: 25,
    textAlign: "center"
  }
});

```

```

    },
    img: {
      height: 25,
      width: 25,
    }
  });

```

Componente para la selección de respuesta:

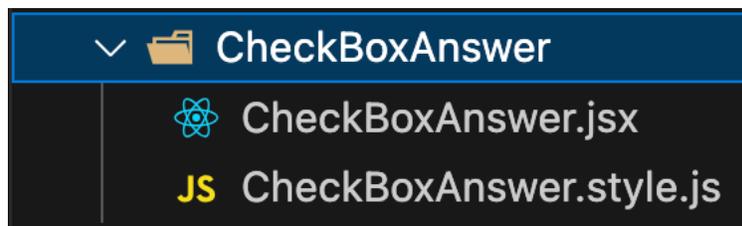


Ilustración 6 Componente CheckBoxAnswer.

```

import { Text, TouchableOpacity } from "react-native"
import { s } from "../CheckBoxAnswer.style"
import CheckBox from "expo-checkbox";

const CheckBoxAnswer = ({answer, onPress}) => {
  //Deestructuramos answer
  const {option, isChecked, correctAnswer } = answer;
  return(
    <TouchableOpacity style={s.card} onPress={()=>
onPress(answer)}>
      <Text style={s.txtAnswer}>{option}</Text>
      <CheckBox
        style={s.checkbox}
        value={isChecked}
        onChange={() => onPress(answer)}
        color={isChecked && correctAnswer ? "#77DD77" :
undefined}
      />

```

```

        </TouchableOpacity>
    );
};

export default CheckBoxAnswer;

import { StyleSheet } from "react-native";

export const s = StyleSheet.create({
  card: {
    flexDirection: "row",
    alignItems: "center",
    backgroundColor: "#6495ED",
    height: "auto",
    shadowColor: "#000",
    shadowOffset: {
      width: 0,
      height: 2,
    },
    shadowOpacity: 0.25,
    shadowRadius: 3.84,
    elevation: 5,
    borderRadius: 10,
    padding: 10,
  },
  checkbox: {
    backgroundColor: "white",
    width: 25,
    height: 25,
    margin: 8,
  },
  txtAnswer: {
    color: "white",
    fontSize: 20,
  }
});

```

```
});
```

Componente que muestra la solución de los ejercicios:

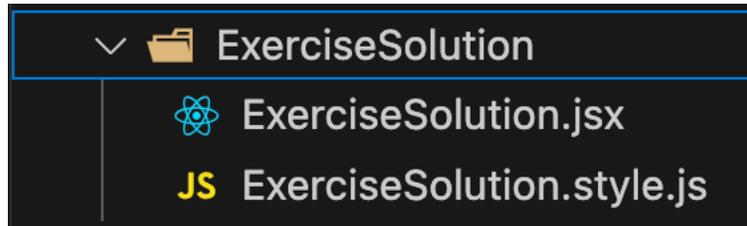


Ilustración 7 Componente ExerciseSolution

```
import { View } from "react-native";
import { s } from "./ExerciseSolution.style";
import MathJax from "react-native-mathjax";
```

```
const mmlOptions = {
  messageStyle: "none",
  extensions: ["tex2jax.js"],
  jax: ["input/TeX", "output/HTML-CSS"],
  tex2jax: {
    inlineMath: [
      ["$", "$"],
      ["\\(", "\\)"],
    ],
    displayMath: [
      ["$$", "$$"],
      ["\\[", "\\]"],
    ],
    processEscapes: true,
  },
  TeX: {
    extensions: [
      "AMSmath.js",
      "AMSsymbols.js",
    ],
  },
};
```

```

        "noErrors.js",
        "noUndefined.js",
    ],
  },
};

const ExerciseSolution = ({ solution }) => {
  return(
    <View style={ s.container }>
      <MathJax
        mathJaxOptions={mmlOptions}
        html={`<font size=4> ${ solution } </font>`}
      />
    </View>
  );
};

export default ExerciseSolution;

import { StyleSheet } from "react-native";

export const s = StyleSheet.create({
  container: {
    paddingBottom:50,
  },
  statement: {
    fontSize: 25,
    textAlign: "justify",
  }
});

```

Header de la página principal:

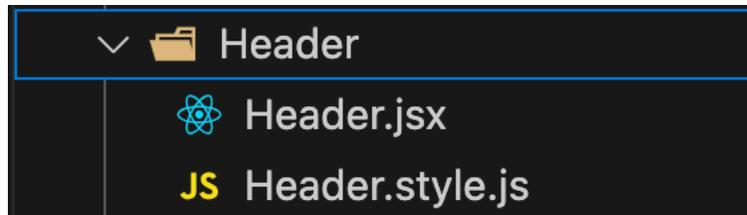


Ilustración 8 Componente Header.

```
import {Text, Image, View, TouchableOpacity} from "react-native";
import { s } from "./Header.style";
import imgLogo from "../../assets/logo.png";
import copyRight from "../../assets/copyright.png";
import infoImg from "../../assets/info.png";
import { useNavigation } from "@react-navigation/native";

const Header = ({ onPress }) => {
  const nav = useNavigation();

  return(
    <>
      <View style={s.container}>
        <TouchableOpacity onPress={() =>
nav.navigate("CopyRight")}>
          <Image style={s.copyRight_img}
source={copyRight}/>
        </TouchableOpacity>
        <View>
          <Image style={s.img} source={imgLogo}
resizeMode="contain" />
        </View>
        <TouchableOpacity onPress={onPress}>
          <Image style={s.info_img} source={infoImg} />
        </TouchableOpacity>
    </>
  )
}
```

```

        </View>
        <View>
            <Text style={s.subtitle} >Temas de álgebra
lineal</Text>
        </View>
    </>
    );
};

export default Header;

import { StyleSheet } from "react-native";

export const s = StyleSheet.create({
    container:{
        flexDirection: "row",
        alignItems: "center",
        justifyContent: "space-between",
        paddingHorizontal: 15
    },
    img:{
        width: 170,
        alignSelf: "center"
    },
    subtitle: {
        marginTop: -20,
        fontSize:25,
        color: "#ABABAB",
        textAlign: "center",
    },
    copyright_img : {
        width: 40,
        height: 40
    },
},

```

```

    info_img: {
      width: 40,
      height: 40
    }
  });

```

NavHeader de la página de ejercicios:

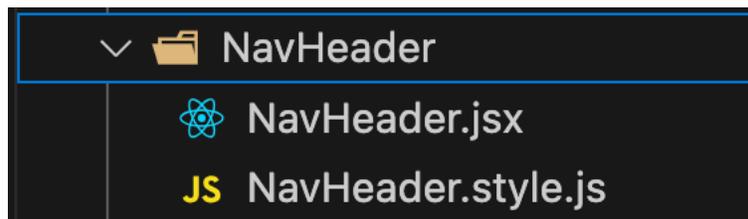


Ilustración 9 Componente NavHeader.

```

import {Text, Image, View, TouchableOpacity} from "react-native";
import { s } from "./NavHeader.style";
import imgLogo from "../../assets/logo.png";
import { useNavigation } from "@react-navigation/native";
import imgHome from "../../assets/home.png";
import imgHelp from "../../assets/help.png";

const NavHeader = ({title, onPress}) => {
  const nav = useNavigation();

  return(
    <>
      <View style={s.container}>
        <TouchableOpacity onPress={() => nav.goBack()} >
          <Image style={s.home_img} source={imgHome} />
        </TouchableOpacity>
        <View>
          <Image style={s.img} source={imgLogo}
resizeMode="contain" />

```

```

        </View>
        <TouchableOpacity onPress={onPress} >
            <Image style={s.help_img} source={imgHelp} />
        </TouchableOpacity>
    </View>
    <View>
        <Text style={s.subtitle} >{title}</Text>
    </View>
</>
);
};

```

```
export default NavHeader;
```

```
import { StyleSheet } from "react-native";
```

```

export const s = StyleSheet.create({
  container: {
    flexDirection: "row",
    alignItems: "center",
    justifyContent: "space-between",
    paddingHorizontal: 15
  },
  img: {
    width: 170,
    alignSelf: "center"
  },
  subtitle: {
    marginTop: -20,
    fontSize: 25,
    color: "#ABABAB",
    textAlign: "center",
    paddingHorizontal: 15
  },
},

```

```

    home_img: {
      width:40,
      height: 40
    },
    help_img: {
      width:40,
      height: 40
    },
  });

```

Header de navegación en la página de créditos:

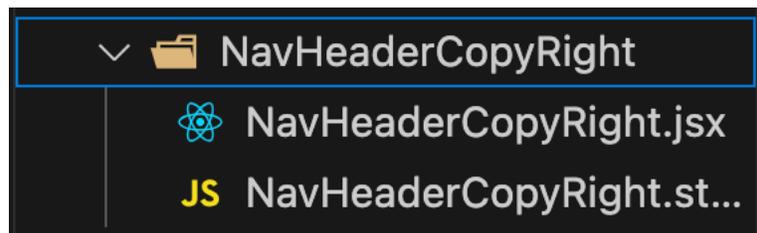


Ilustración 10 Componente NavHeaderCopyRight.

```

import { s } from "./NavHeaderCopyRight.style";
import imgLogo from "../../assets/logo.png";
import infoImg from "../../assets/info.png";
import { TouchableOpacity, View, Image, Text } from "react-native";
import imgHome from "../../assets/home.png";
import { useNavigation } from "@react-navigation/native";

const NavHeaderCopyRight = ({ onPress }) => {
  const nav = useNavigation();

  return(
    <>
      <View style={s.container} >
        <TouchableOpacity onPress={() => nav.goBack()} >
          <Image style={s.home_img} source={imgHome} />

```

```

        </TouchableOpacity>
        <View>
            <Image style={s.img} source={imgLogo}
resizeMode="contain" />
        </View>
        <TouchableOpacity onPress={onPress} >
            <Image style={s.info_img} source={infoImg} />
        </TouchableOpacity>
    </View>
    <View>
        <Text style={s.subtitle} >Créditos Álgebra Lineal
App</Text>
    </View>
</>
);
};

```

```
export default NavHeaderCopyRight;
```

```
import { StyleSheet } from "react-native";
```

```

export const s = StyleSheet.create({
  container:{
    flexDirection: "row",
    alignItems: "center",
    justifyContent: "space-between",
    paddingHorizontal: 15
  },
  img:{
    width: 170,
    alignSelf: "center"
  },
  subtitle: {
    marginTop: -20,
    fontSize:25,

```

```

        color: "#ABABAB",
        textAlign: "center",
    },
    home_img: {
        width:40,
        height: 40
    },
    info_img: {
        width: 40,
        height: 40
    }
});

```

Header de navegación para la página de preguntas:

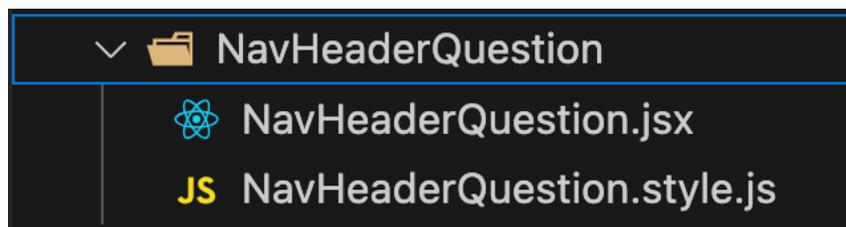


Ilustración 11 Componente NavHeaderQuestion

```

import { Image, Text, TouchableOpacity, View } from "react-native";
import { s } from "./NavHeaderQuestion.style";
import { useNavigation } from "@react-navigation/native";
import imgLogo from "../../assets/logo.png";
import imgBack from "../../assets/arrow_back.png";
import imgHelp from "../../assets/help.png";

const NavHeaderQuestion = ({title, onPress}) => {
  const nav = useNavigation();

  return(
    <>

```

```

    <View style={s.container} >
      <TouchableOpacity onPress={() => nav.goBack()} >
        <Image style={s.back_img} source={imgBack} />
      </TouchableOpacity>
      <View>
        <Image style={s.img} source={imgLogo}
resizeMode="contain" />
      </View>
      <TouchableOpacity onPress={onPress} >
        <Image style={s.help_img} source={imgHelp} />
      </TouchableOpacity>
    </View>
    <View>
      <Text style={s.subtitle} >{title}</Text>
    </View>
  </>
);
};

```

```
export default NavHeaderQuestion;
```

```
import { StyleSheet } from "react-native";
```

```

export const s = StyleSheet.create({
  container: {
    flexDirection: "row",
    alignItems: "center",
    justifyContent: "space-between",
    paddingHorizontal: 15
  },
  img: {
    width: 170,
    alignSelf: "center",
  },
  subtitle: {

```

```

        marginTop: -20,
        fontSize: 25,
        color: "#ABABAB",
        textAlign: "center",
        paddingHorizontal: 15
    },
    back_img: {
        width: 40,
        height: 40,
    },
    help_img: {
        width: 40,
        height: 40,
    }
});

```

Header de navegación en la página de solución:

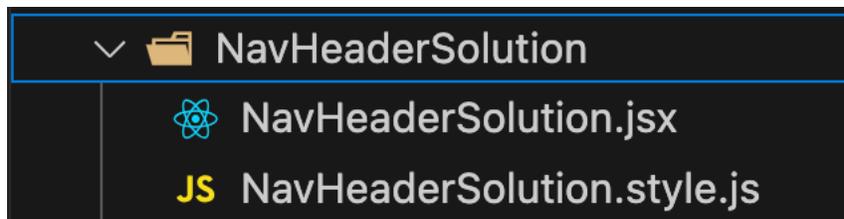


Ilustración 12 Componente NavHeaderSolution.

```

import { Image, Text, TouchableOpacity, View } from "react-native";
import { s } from "./NavHeaderSolution.style";
import imgLogo from "../../assets/logo.png";
import imgHome from "../../assets/home.png";
import imgBack from "../../assets/arrow_back.png";
import { useNavigation } from "@react-navigation/native";

const NavHeaderSolution = ({title}) => {
    const nav = useNavigation();

```

```

return(
  <>
    <View style={s.container}>
      <TouchableOpacity onPress={ () => nav.goBack() }>
        <Image style={s.back_img} source={imgBack} />
      </TouchableOpacity>
      <View>
        <Image style={s.img} source={imgLogo}
resizeMode="contain" />
      </View>
      <TouchableOpacity onPress={ () =>
nav.navigate("Home") } >
        <Image style={s.home_img} source={imgHome} />
      </TouchableOpacity>
    </View>
    <View>
      <Text style={s.subtitle} >{title}</Text>
    </View>
  </>
)
}

```

```
export default NavHeaderSolution;
```

```
import { StyleSheet } from "react-native";
```

```
export const s = StyleSheet.create({
  container: {
    flexDirection: "row",
    alignItems: "center",
    justifyContent: "space-between",
    paddingHorizontal: 15
  },

```

```

img: {
  width: 170,
  alignSelf: "center",
},
subtitle: {
  marginTop: -20,
  fontSize: 25,
  color: "#ABABAB",
  textAlign: "center",
  paddingHorizontal: 15
},
back_img: {
  width: 40,
  height: 40,
},
home_img: {
  width: 40,
  height: 40,
}
});

```

Header para navegación en la página de teoría:

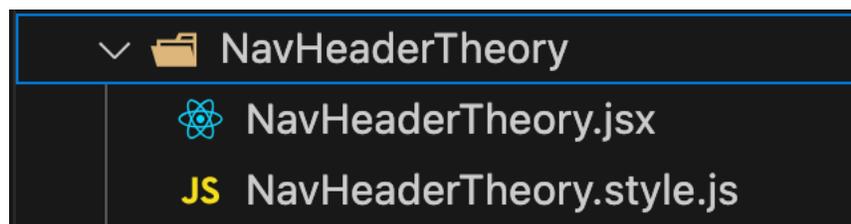


Ilustración 13 Componente NavHeaderTheory

```

import { s } from "./NavHeaderTheory.style";
import imgLogo from "../../assets/logo.png";
import imgHome from "../../assets/home.png";
import imgHelp from "../../assets/help.png";

```

```

import { useNavigation } from "@react-navigation/native";
import { Image, Text, TouchableOpacity, View } from "react-native";

const NavHeaderTheory = ({title, onPress}) => {
  const nav = useNavigation();

  return(
    <>
      <View style={s.container} >
        <TouchableOpacity onPress={() =>
nav.navigate("Home")} >
          <Image style={s.home_img} source={imgHome} />
        </TouchableOpacity>
        <View>
          <Image style={s.img} source={imgLogo}
resizeMode="contain" />
        </View>
        <TouchableOpacity onPress={onPress} >
          <Image style={s.help_img} source={imgHelp} />
        </TouchableOpacity>
      </View>
      <View>
        <Text style={s.subtitle} >{title}</Text>
      </View>
    </>
  );
};

export default NavHeaderTheory;

import { StyleSheet } from "react-native";

export const s = StyleSheet.create({

```

```

container: {
  flexDirection: "row",
  alignItems: "center",
  justifyContent: "space-between",
  paddingHorizontal: 15
},
img: {
  width: 170,
  alignSelf: "center",
},
subtitle: {
  marginTop: -20,
  fontSize: 25,
  color: "#ABABAB",
  textAlign: "center",
  paddingHorizontal: 15
},
home_img: {
  width: 40,
  height: 40,
},
help_img: {
  width: 40,
  height: 40,
}
});

```

Pestañas al pie de la página de respuestas:

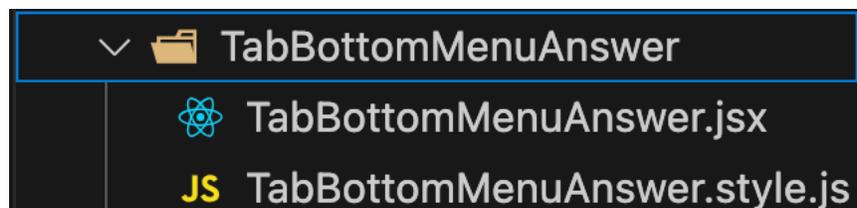


Ilustración 14 Componente TabBottomMenuAnswer

```

import { Text, TouchableOpacity, View } from "react-native";
import { s } from "./TabBottomMenuAnswer.style";
import { useNavigation } from "@react-navigation/native";

const TabBottomMenuAnswer = ({selectedTabName, onPress,
exerciseNum, exerciseId}) => {

    const nav = useNavigation();

    // Función getTextStyle que resalta en azul el tab
seleccionado
    const getTabName = (tabName) => {
        return {
            fontWeight: "bold",
            color: selectedTabName === tabName ? "#2F76E5" :
"black"
        }
    }

    return(
        <View style={s.root}>
            <TouchableOpacity onPress={() => onPress("Exercise")}
>
                <Text style={getTabName("Exercise")}>Ejercicio
{exerciseNum}</Text>
            </TouchableOpacity>
            <TouchableOpacity onPress={() =>
nav.navigate("Theory",{exerciseNum, exerciseId})} >
                <Text style={getTabName("Theory")} >Teoría</Text>
            </TouchableOpacity>
        </View>
    );

```

```
import { StyleSheet } from "react-native";

export const s = StyleSheet.create({
  root: {
    flex: 1,
    flexDirection: "row",
    justifyContent: "space-around",
    alignItems: "center",
  }
});
```

Pestañas al pie en la página de los ejercicios:

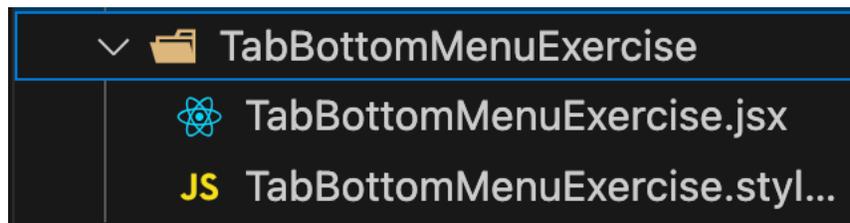


Ilustración 15 Componente TabBottomMenuExercise

```
import { Text, TouchableOpacity, View } from "react-native";
import { s } from "../TabBottomMenuExercise.style";

const TabBottomMenuExercise = ({ selectedTabName, onPress,
exerciseList }) => {

  const countByStatus = exerciseList.reduce((acc, exercise) =>
  {
    exercise.resolved ? acc.done++ : acc.inProgres++
    return acc
  }, {
    all: exerciseList.length,
    inProgres: 0,
    done: 0
```

```

});

// Función getTextStyle que resalta en azul el tab
seleccionado
const getTextStyle = (tabName) => {
  return {
    fontWeight: "bold",
    color: selectedTabName === tabName ? "#2F76E5" :
"black"
  };
};

return(
  <View style={s.root} >
    <TouchableOpacity onPress={() => onPress("all")}>
      <Text style={getTextStyle("all")} >Todos
({countByStatus.all})</Text>
    </TouchableOpacity>
    <TouchableOpacity onPress={() =>
onPress("inProgress")}>
      <Text style={getTextStyle("inProgress")}>En
progreso ({countByStatus.inProgress})</Text>
    </TouchableOpacity>
    <TouchableOpacity onPress={() => onPress("done")}>
      <Text style={getTextStyle("done")}>Terminados
({countByStatus.done})</Text>
    </TouchableOpacity>
  </View>
);
};

export default TabBottomMenuExercise;

import { StyleSheet } from "react-native";

```

```

export const s = StyleSheet.create({
  root: {
    flex: 1,
    flexDirection: "row",
    justifyContent: "space-around",
    alignItems: "center",
  }
});

```

Pestañas al pie de la página principal:

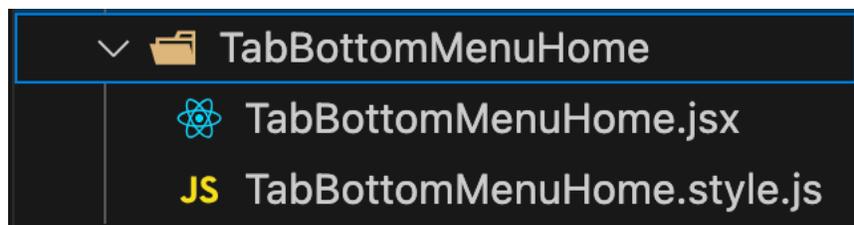


Ilustración 16 Componente TabBottomMenuHome

```

import { View, Text, TouchableOpacity } from "react-native";
import { s } from "../TabBottomMenuHome.style";

/**
 *
 * @selectedTabName prop del componente para seleccionar el tab y
filtrar los temas a mostrar
 * @onPress prop del componente para seleccionar el tab "all",
"inProgress" y "done"
 * @topicList prop del componente que pasa un arreglo de objetos
que contienen la información de los temas a mostrar
 */

const TabBottomMenuHome = ({ selectedTabName, onPress, topicList
}) => {

```

```

// Función countByStatus para mostrar la cuenta en las
opciones del footer en la aplicación
const countByStatus = topicList.reduce((acc, topic) => {
  topic.isCompleted ? acc.done++ : acc.inProgress++
  return acc
},{
  all: topicList.length,
  inProgress: 0,
  done: 0,
})

```

// Función getTextStyle que resalta en azul el tab
seleccionado

```

const getTextStyle = ( tabName ) => {
  return{
    fontWeight: "bold",
    color: selectedTabName === tabName ? "#2F76E5" :
"black"
  }
}

return(
  <View style={s.root}>
    <TouchableOpacity onPress={() => onPress("all")} >
      <Text style={getTextStyle("all")} >Todos
({countByStatus.all})</Text>
    </TouchableOpacity>
    <TouchableOpacity onPress={() =>
onPress("inProgress")} >
      <Text style={getTextStyle("inProgress")} >En
progreso ({countByStatus.inProgress})</Text>
    </TouchableOpacity>
    <TouchableOpacity onPress={() => onPress("done")} >

```

```

        <Text style={getTextStyle("done")} >Terminados
({countByStatus.done})</Text>
      </TouchableOpacity>
    </View>
  );
};

```

```
export default TabBottomMenuHome;
```

```
import { StyleSheet } from "react-native";
```

```
export const s = StyleSheet.create({
  root: {
    flex: 1,
    flexDirection: "row",
    justifyContent: "space-around",
    alignItems: "center",
  }
});
```

Pestaña al pie de la página del ejercicio:

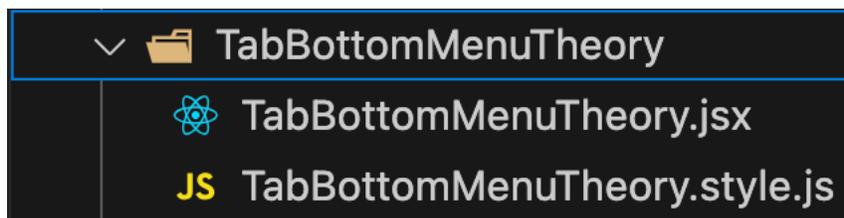


Ilustración 17 Componente TabBottomMenuTheory

```
import { Text, TouchableOpacity, View } from "react-native";
import { s } from "../TabBottomMenuTheory.style";
import { useNavigation } from "@react-navigation/native";
```

```
const TabBottomMenuTheory = ({selectedTabName, onPress,
exerciseNum}) => {
```

```

const nav = useNavigation();

// Función getTextStyle que resalta en azul el tab
seleccionado
const getTabName = (tabName) => {
  return {
    fontWeight: "bold",
    color: selectedTabName === tabName ? "#2F76E5" :
"black"
  }
}

return(
  <>
    <View style={s.root} >
      <TouchableOpacity onPress={() => nav.goBack()} >
        <Text style={getTabName("Exercise")} >
>Ejercicio {exerciseNum}</Text>
      </TouchableOpacity>
      <TouchableOpacity onPress={() =>
onPress("Theory")} >
        <Text style={getTabName("Theory")} >
>Teoría</Text>
      </TouchableOpacity>
    </View>
  </>
);
};

export default TabBottomMenuTheory;

import { StyleSheet } from "react-native";

export const s = StyleSheet.create({

```

```

root: {
  flex: 1,
  flexDirection: "row",
  justifyContent: "space-around",
  alignItems: "center",
}
});

```

Componente para visualizar la teoría asociada:

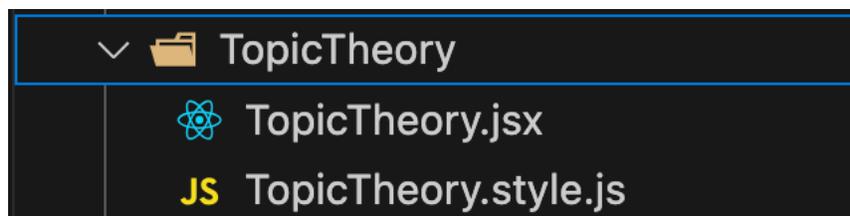


Ilustración 18 Componente TopicTheory.

```

import { View } from "react-native";
import { s } from "../TopicTheory.style";
import MathJax from "react-native-mathjax";

const mmlOptions = {
  messageStyle: "none",
  extensions: ["tex2jax.js"],
  jax: ["input/TeX", "output/HTML-CSS"],
  tex2jax: {
    inlineMath: [
      ["$", "$"],
      ["\\(", "\\)"],
    ],
    displayMath: [
      ["$$", "$$"],
      ["\\[", "\\]"],
    ],
    processEscapes: true,
  }
};

```

```

    },
    TeX: {
      extensions: [
        "AMSmath.js",
        "AMSsymbols.js",
        "noErrors.js",
        "noUndefined.js",
      ],
    },
  };

const TopicTheory = ({ theory }) => {
  return(
    <View style={s.container}>
      <MathJax
        mathJaxOptions={mmlOptions}
        html={`<font size=4> ${ theory } </font>`}
      />
    </View>
  );
};

export default TopicTheory;

import { StyleSheet } from "react-native";

export const s = StyleSheet.create({

});

```

Componente para darle un formato específico al texto que se usa en la aplicación:

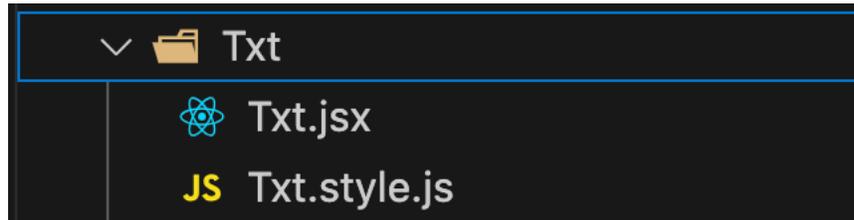


Ilustración 19 Componente Txt

```
import { Text } from "react-native";
import { s } from "./Txt.style";

const Txt = ({ children }) => {
  return(
    <Text style={s.txt} >{ children }</Text>
  );
};

export default Txt;

import { StyleSheet } from "react-native";

export const s = StyleSheet.create({
  txt: {
    fontSize: 20,
    color: "#1c1c1c",
    fontFamily: "Alata-Regular",
    textAlign: "center",
    paddingBottom: 10
  }
});
```

Páginas del proyecto:

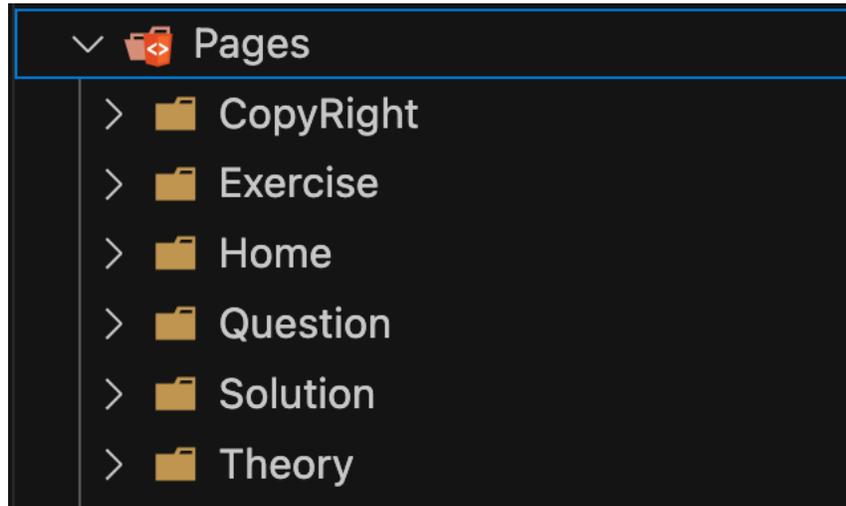


Ilustración 20 Pages.

Página de créditos y copy right:

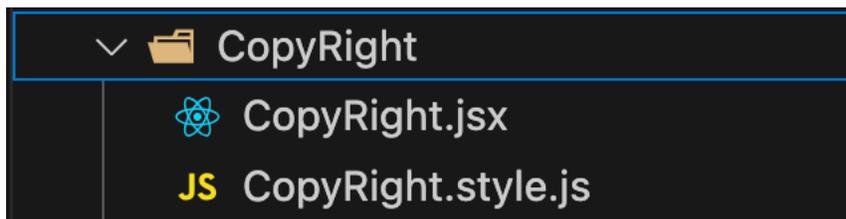


Ilustración 21 Página CopyRight.

```
import { ScrollView, View } from "react-native";
import { s } from "../CopyRight.style";
import { SafeAreaView, SafeAreaView } from "react-native-safe-area-context";
import { NavHeaderCopyRight, Txt } from "../../components";

const CopyRight = ({onPressInfo}) => {
```

```

const renderCredits = () => (
  <>
    <Txt>Tesis por apoyo a la docencia</Txt>
    <Txt/>
    <Txt>Diseño y Programación</Txt>
    <Txt>Cristian Rodolfo Gómez Mares</Txt>
    <Txt/>
    <Txt>Directora del trabajo</Txt>
    <Txt>Rosalba Rodríguez Chávez</Txt>
    <Txt/>
    <Txt>Ejercicios y resolución</Txt>
    <Txt>Rosalba Rodríguez Chávez</Txt>
    <Txt/>
    <Txt>Revisión</Txt>
    <Txt>Alicia Pineda Ramírez</Txt>
    <Txt>Aldo Jiménez Arteaga</Txt>
    <Txt/>
    <Txt>Coordinación</Txt>
    <Txt>M.F. Alicia Pineda Ramírez</Txt>
    <Txt>Mtra. María del Rocío Ávila Núñez</Txt>
    <Txt/>
    <Txt>© {new Date().getFullYear()} Álgebra Lineal
App.</Txt>
    <Txt>Esta aplicación móvil puede ser utilizada con
fines no lucrativos y siempre que se cite la fuente.</Txt>
  </>
);

return(
  <>
    <SafeAreaView>
      <SafeAreaView style={s.root}>
        <View style={s.nav_HeaderCopyright} >
          <NavHeaderCopyright

```

```

                onPress={onPressInfo}
            />
        </View>
        <View style={s.body}>
<ScrollView>{renderCredits()}</ScrollView>
        </View>
        </SafeAreaView>
        </SafeAreaProvider>
    </>
    );
};

export default Copyright;

import { StyleSheet } from "react-native";

export const s = StyleSheet.create({
    root: {
        flex: 1,
        backgroundColor: "#F9F9F9",
        textAlign: "center"
    },
    nav_HeaderCopyright: {
        flex: 1,
        marginBottom: 25,
        paddingHorizontal: 10
    },
    body: {
        flex: 5,
        paddingHorizontal: 15
    },
});

```

Página donde se muestra el listado de los ejercicios:

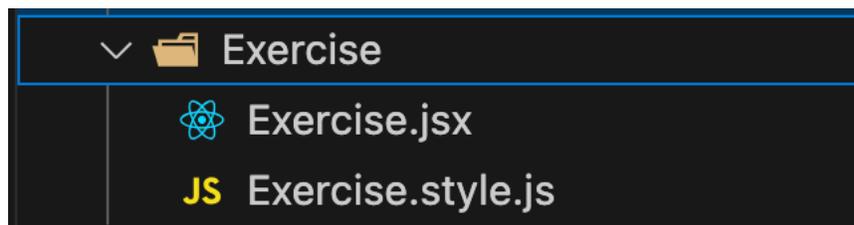


Ilustración 22 PáginaExercise.

```
import { SafeAreaView, SafeAreaContext } from "react-native-safe-area-context";
import { s } from "./Exercise.style";
import { ScrollView, View } from "react-native";
import { useRoute } from "@react-navigation/native";
import { useState, useEffect } from "react";
import exerciseData from "../../data/exercises.json";
import AsyncStorage from "@react-native-async-storage/async-storage";
import { CardExercise, NavHeader, TabBottomMenuExercise } from "../../components";

let isFirstRender = true; //Bandera para controlar la carga de los datos
let isLoadingUpdate = false; //Bandera para controlar cuando se actualizan los datos y guardarlos

const Exercise = ({onPressHelp}) => {
  // Datos del tema seleccionado que se pasan mediante params
  const { params } = useRoute();
```

```

    // Hook para inicializar los ejercicios obtenidos del
    exercisesData
    const [ exerciseList, setExerciseList ] = useState([]);

    // Hook para filtrar la lista de ejercicios
    const [ filterExerciseList, setFilterExerciseList ] =
    useState([]);

    // Inicializamos siempre el tab en "all" mediante un hook
    useState
    const [ selectedTabName, setSelectedTabName ] =
    useState("InProgress");

    // Cargar los datos guardados cuando se inicia la aplicación
    useEffect(() => {
        loadExerciseList();
        isFirstRender = true;
    }, []);

    useEffect(() => {
        if (!isLoadingUpdate) {
            !isFirstRender ? saveExerciseList() : isFirstRender =
false;
        } else {
            isLoadingUpdate = false;
        };
    }, [exerciseList]);

    // Función para filtrar los ejercicios asociados al tema, se
    filtra mediante el id del tema(topic) seleccionado que se pasa
    mediante params
    const listFilteredExercise = (exercisesList) => {
        if(exercisesList) {

```

```

        return exercisesList.filter((exercise) =>
exercise._topic[0].id === params.id);
    }
    return exerciseData.filter((exercise) =>
exercise._topic[0].id === params.id);
};

```

//Función que se encarga de pasar los datos de string a un objeto de JSON y así poder cargarlos de inicio

```

const loadExerciseList = async () => {
    //console.log("LOAD");
    try {
        const exerciseListString = await
AsyncStorage.getItem("@exerciseList");
        const parsedExerciseList =
JSON.parse(exerciseListString);
        const listFiltered =
listFilteredExercise(parsedExerciseList);
        isLoadingUpdate = true;
        parsedExerciseList === null ||
parsedExerciseList.length === 0 ? setExerciseList(exerciseData):
setExerciseList(parsedExerciseList);
        setFilterExerciseList(listFiltered);
        //removeData();
    } catch (error) {
        alert(error);
    }
};
};

```

//Función que guarda los cambios que el usuario va realizando en la aplicación ej: cuando se actualiza el boolean resolved

```

const saveExerciseList = async () => {
    try {

```

```

        await AsyncStorage.setItem("@exerciseList",
JSON.stringify(exerciseList));
    } catch (error) {
        alert(error);
    };
};
};

```

// función que filtra el contenido de cards para los ejercicios dependiendo el tab seleccionado

```

const getFilteredList = () => {
    switch (selectedTabName) {
        case "all":
            return exerciseList.filter((exercise) =>
exercise._topic[0].id === params.id);
        case "InProgress":
            return exerciseList.filter((exercise) =>
exercise.resolved === false && exercise._topic[0].id ===
params.id);
        case "done":
            return exerciseList.filter((exercise) =>
exercise.resolved === true && exercise._topic[0].id ===
params.id);
    }
}
}

```

//Función que renderiza las cards dependiendo el tab seleccionado

```

const renderExerciseList = () => {
    return getFilteredList().map((exercise) =>
        <View key={exercise.id} style={s.cardItem} >
            <CardExercise onLongPress={updateExercise}
exercise={exercise} />
        </View>
    )
}

```

```

    }

    //Función que actualiza la bandera resolved después de
    presionar la card por más de tres segundos
    const updateExercise = (exercise) => {
      const updatedExercise = {
        ...exercise,
        resolved: !exercise.resolved
      };
      const updateExerciseList = [...exerciseList];
      const indexToUpdate = updateExerciseList.findIndex((t) =>
t.id === updatedExercise.id);
      updateExerciseList[indexToUpdate] = updatedExercise;
      const listFiltered =
listFilteredExercise(updateExerciseList);
      setExerciseList(updateExerciseList);
      setFilterExerciseList(listFiltered);
    };

    return(
      <>
        <SafeAreaView>
          <SafeAreaView style={s.root} >
            <View style={s.nav_header} >
              <NavHeader
                title={params.title}
                onPress={onPressHelp}
              />
            </View>
            <View style={s.body} >

<ScrollView>{renderExerciseList()}</ScrollView>
          </View>
        </SafeAreaView>
      </>
    );
  };
}

```

```

    </SafeAreaView>
    <View style={s.footer} >
      <TabBottomMenuExercise
        selectedTabName={selectedTabName}
        onPress={setSelectedTabName}
        exerciseList={filterExerciseList}
      />
    </View>
  </>
);
};

export default Exercise;

```

Página inicial o home:

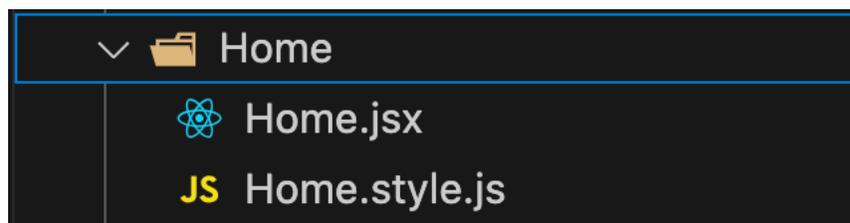


Ilustración 23 Página Home.

```

import { ScrollView, View } from "react-native";
import { SafeAreaView, SafeAreaView } from "react-native-safe-area-context";
import { s } from "./Home.style";
import { CardHome, Header, TabBottomMenuHome } from
"../../components";
import { useEffect, useState } from "react";
import AsyncStorage from "@react-native-async-storage/async-storage";
import TopicData from "../../data/topics.json";
import { useNavigation } from "@react-navigation/native";

```

```

let isFirstRender = true; //Bandera para controlar la carga de
los datos
let isLoadingUpdate = false; //Bandera para controlar cuando se
actualizan los datos y guardarlo

const Home = ({ onPressInfo }) => {
  const nav = useNavigation();

  //hook para inicializar la data de los temas
  const [ topicList, setTopicList ] = useState([]);

  // Inicializamos siempre el tab en "all" mediante un hook
useState
  const [ selectedTabName, setSelectedTabName ] =
useState("InProgress");

  //Cargar los datos guardados cuando se inicia la aplicación
useEffect(() => {
  loadTopicList();
}, []);

  //Ejecuta la función saveTopicList para guardar cambios en la
data cada que se actualiza topicList
useEffect(() => {
  if(!isLoadingUpdate){
    !isFirstRender ? saveTopicList() : isFirstRender =
false;
  }else{
    isLoadingUpdate = false;
  }
}, [topicList]);

```

//Función que se encarga de pasar los datos de string a un objeto de JSON y así poder cargarlos de inicio

```
const loadTopicList = async () => {
  try {
    const topicListString = await
AsyncStorage.getItem("@topicList");
    const parsedTopicList = JSON.parse(topicListString);
    isLoadingUpdate = true;
    if(parsedTopicList === null) {
      setTopicList(TopicData);
    } else {
      parsedTopicList.length === 0 ?
setTopicList(TopicData) : setTopicList(parsedTopicList);
    }
  } catch (error) {
    alert(error)
  };
};
```

//Función que guarda los cambios que el usuario va realizando en la aplicación ej: cuando se actualiza el boolean isComplete

```
const saveTopicList = async () => {
  try {
    await AsyncStorage.setItem("@topicList",
JSON.stringify(topicList));
  } catch (error) {
    alert(error);
  };
};
```

// Función que filtra el contenido de cards dependiendo el tab seleccionado

```
const getFilteredList = () => {
  switch(selectedTabName){
```

```

        case "all":
            return topicList;
        case "InProgress":
            return topicList.filter((topic) =>
topic.isCompleted === false);
        case "done":
            return topicList.filter((topic) =>
topic.isCompleted === true);
    };
};

```

//Función que renderiza las cards dependiendo el tab
seleccionado

```

const renderTopicList = () => {
    return getFilteredList().map((topic) =>
        <View key={topic.id} style={s.cardItem} >
            <CardHome onLongPress={updateTopic} topic={topic}
/>
        </View>
    );
};

```

//Función que actualiza la bandera isCompleted después de
presionar la card por más de tres segundos

```

const updateTopic = (topic) => {
    const updatedTopic = {
        ...topic,
        isCompleted: !topic.isCompleted
    };
    const updateTopicList = [ ...topicList ];
    const indexToUpdate = updateTopicList.findIndex( (t) =>
t.id === updatedTopic.id );
    updateTopicList[indexToUpdate] = updatedTopic;
    setTopicList(updateTopicList);
};

```

```

};

return(
  <>
    <SafeAreaView>
      <SafeAreaView style={s.home} >
        <View style={s.header}>
          <Header
            onPress={onPressInfo}
          />
        </View>
        <View style={s.body} >
<ScrollView>{renderTopicList()}</ScrollView>
          </View>
        </SafeAreaView>
      </SafeAreaView>
      <View style={s.footer} >
        <TabBottomMenuHome
          selectedTabName={selectedTabName}
          onPress={setSelectedTabName}
          topicList={topicList}
        />
      </View>
    </>
  );
};

```

```
export default Home;
```

```
import { StyleSheet } from "react-native";
```

```
export const s = StyleSheet.create({
  home:{
```

```

        flex: 1,
        backgroundColor: "#F9F9F9"
    },
    cardItem: {
        marginBottom: 15,
        alignItems: "center",
        paddingTop: 5,
    },
    header: {
        flex: 1,
        paddingHorizontal: 10,
    },
    body: {
        flex: 5
    },
    footer: {
        height: 80,
        backgroundColor: "white",
    }
});

```

Página donde se muestra el ejercicio con las posibles respuestas:

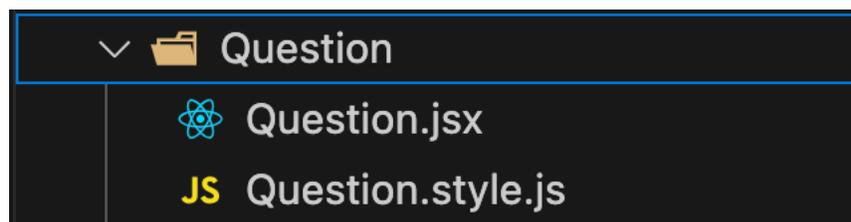


Ilustración 24 Página Question.

```

import { SafeAreaView, SafeAreaContext } from "react-native-safe-area-context";
import { s } from "./Question.style";
import { ScrollView, View } from "react-native";

```

```

import { Ask, CheckBoxAnswer, NavHeaderQuestion,
TabBottomMenuAnswer } from "../../components";
import optionData from "../../data/options.json";
import { useNavigation, useRoute } from "@react-
navigation/native";
import { useEffect, useState, React } from "react";
import Dialog from "react-native-dialog";

// Arreglo donde se agregaran las respuestas de forma aleatoria
del ejercicio seleccionado
let options = [
  {option: "a"},isChecked: false, viewAnswer: false},
  {option: "b"},isChecked: false, viewAnswer: false},
  {option: "c"},isChecked: false, viewAnswer: false},
  {option: "d"},isChecked: false, viewAnswer: false}
]

import MathJax from "react-native-mathjax";

const mmlOptions = {
  messageStyle: "none",
  extensions: ["tex2jax.js"],
  jax: ["input/TeX", "output/HTML-CSS"],
  tex2jax: {
    inlineMath: [
      ["$", "$"],
      ["\\(", "\\)"],
    ],
    displayMath: [
      ["$$", "$$"],
      ["\\[", "\\]"],
    ],
    processEscapes: true,
  },
},

```

```

TeX: {
  extensions: [
    "AMSmath.js",
    "AMSSymbols.js",
    "noErrors.js",
    "noUndefined.js",
  ],
},
};

const Question = ({onPressHelp}) => {
  const nav = useNavigation();

  // Datos del ejercicio seleccionado que se pasan mediante nav
  en param
  const { params } = useRoute();

  // Hook para inicializar la lista de respuestas opcionales
  const [answerList, setAnswerList] = useState([]);

  // Hook para inicializar el tab siempre en Exercise
  const [selectedTabName, setSelectedTabName] =
  useState("Exercise");

  // Hook para ocultar y mostrar el dialogo de respuesta
  erronea
  const [isWrongAnswerDialogDisplay,
  setIsWrongAnswerDialogDisplay] = useState(false);

  // Hook para ocultar y mostrar el dialogo de respuesta
  correcta
  const [isCorrectAnswerDialogDisplay,
  setIsCorrectAnswerDialogDisplay] = useState(false);

```

```

// Hook para cargar la lista de respuestas asociadas al
Ejercicio seleccionado
useEffect(() => {
    loadAnswerList();
}, [])

// Función que filtra las respuestas mediante el Id de
ejercicio seleccionado
const listFilteredAnswers = (exerciseId) => {
    if(exerciseId !== 0) {
        return optionData.filter( (option) =>
option._exercise[0].id === exerciseId )
    };
    return [];
};

// Ordenamos las respuestas de forma aleatoria en un arreglo
const orderRandomOptions = (arrOptions) => {
    if (arrOptions) {
        return arrOptions.sort(() => Math.random() - 0.5);
    }
    return [];
};

// Función que se encarga de cargar la lista de respuestas
opcionales
const loadAnswerList = () => {
    //console.log("LOAD ANSWERS")
    try {
        const listFilterAnswers =
listFilteredAnswers(params.id);
        const orderedOptions =
orderRandomOptions(listFilterAnswers);
        const listOrdered = options.map((option, id) => {

```

```

        option.statement = orderedOptions[id].choice;
        option.correctAnswer =
orderedOptions[id].correctAnswer;
        return option;
    });
    setAnswerList(listOrdered || []);
} catch (error) {
    alert(error);
};
};

```

// Función que renderiza la lista de respuestas dependiendo del ejercicio seleccionado, funciona para android

```

const renderAnswerList = () => {
    return answerList.map((answer, index) =>
        <MathJax
            key={index}
            mathJaxOptions={mmlOptions}
            html={`<font size=2>${answer.option}
${answer.statement}</font>`}
        />
    )
}

```

//Función para renderizar las opciones y poder seleccionar mediante un CheckBox

```

const renderOptionAnswerList = () => {
    return answerList.map((answer, index) =>
        <CheckBoxAnswer
            key={index}
            answer = {answer}
            onPress = {updateAnswer}
        />
    )
}

```

```

    }

    // Funcion que actualiza el checkbox de la respuesta
    seleccionada por el usuario
    const updateAnswer = (answer) => {
      const updatedAnswer = {
        ...answer,
        isChecked: !answer.isChecked
      }
      const updateAnswerList = [...answerList];
      const indexToUpdate = updateAnswerList.findIndex((answer)
=> answer.option === updatedAnswer.option );
      const updatedAnswerList = updateAnswerList.map((element,
index) => {
        if (indexToUpdate !== index) {
          element.isChecked ? element.isChecked = false :
element.isChecked
        }
        return element
      })
      updatedAnswerList[indexToUpdate] = updatedAnswer;
      setAnswerList(updatedAnswerList);
      updatedAnswer.correctAnswer ?
setIsCorrectAnswerDialogDisplay(true) :
setIsWrongAnswerDialogDisplay(true);
    }

    //Diálogo de respuesta incorrecta seleccionada
    const renderWrongAnswerDialog = () => {
      return(
        <Dialog.Container visible={isWrongAnswerDialogDisplay}
onBackdropPress={() => setIsWrongAnswerDialogDisplay(false)} >
          <Dialog.Title >Respuesta incorrecta</Dialog.Title>

```

```

        <Dialog.Description>Si deseas ver la solución
selecciona el botón correspondiente, de lo contrario puedes
volver a intentarlo.</Dialog.Description>
        <Dialog.Button label="Intentar" color="grey"
onPress={() => setIsWrongAnswerDialogDisplay(false) } />
        <Dialog.Button
            label="Solución"
            onPress={() => {
                nav.navigate("Solution",{...params});
                setIsWrongAnswerDialogDisplay(false);
            }}
        />
    </Dialog.Container>
);
};

```

```

//Diálogo de respuesta correcta seleccionada
const renderCorrectAnswerDialog = () => {
    return(
        <Dialog.Container
visible={isCorrectAnswerDialogDisplay} onBackPressed={() =>
nav.goBack()} >
            <Dialog.Title >Respuesta correcta</Dialog.Title>
            <Dialog.Description>Muchas felicidades tu
respuesta es correcta, si deseas ver la solución oprime el botón
correspondiente.</Dialog.Description>
            <Dialog.Button label="Ejercicios" color="grey"
onPress={() => nav.goBack() } />
            <Dialog.Button
                label="Solución"
                onPress={() => {
                    nav.navigate("Solution",{...params});
                    setIsCorrectAnswerDialogDisplay(false);
                }}
            />
        </Dialog.Container>
    );
};

```

```

        />
    </Dialog.Container>
    );
};

return(
    <>
    <SafeAreaView>
        <SafeAreaView style={s.root} >
            <View style={s.nav_header} >
                <NavHeaderQuestion
                    title={params._topic[0].title}
                    onPress={onPressHelp}
                />
            </View>
            <View style={s.body} >
                <ScrollView >
                    <Ask exerciseNum={params.exercise}
statement={params.statement} />
                    {renderAnswerList()}
                    <View style={s.check_option}>
                        {renderOptionAnswerList()}
                    </View>
                </ScrollView>
            </View>
        </SafeAreaView>
    </SafeAreaView>
    <View style={s.footer} >
        <TabBottomMenuAnswer
            selectedTabName={selectedTabName}
            onPress={setSelectedTabName}
            exerciseNum={params.exercise}
            exerciseId = {params.id}
        />
    </View>
    </View>
);
};

```

```

        </View>
        {renderCorrectAnswerDialog()}
        {renderWrongAnswerDialog()}
    </>
    );
};

```

```
export default Question;
```

```
import { StyleSheet } from "react-native";
```

```

export const s = StyleSheet.create({
  root: {
    flex: 1,
    backgroundColor: "#F9F9F9"
  },
  nav_header: {
    flex: 1,
    paddingBottom: 25,
    paddingHorizontal: 10
  },
  cardItem: {
    paddingBottom: 15
  },
  body: {
    flex: 5,
    paddingHorizontal: 15
  },
  footer: {
    height: 80,
    backgroundColor: "white",
  },
  check_option: {
    flexDirection: "row",
    justifyContent: "space-around",

```

```
paddingVertical: 40,  
  }  
});
```

Página donde visualizamos la solución del ejercicio seleccionado:

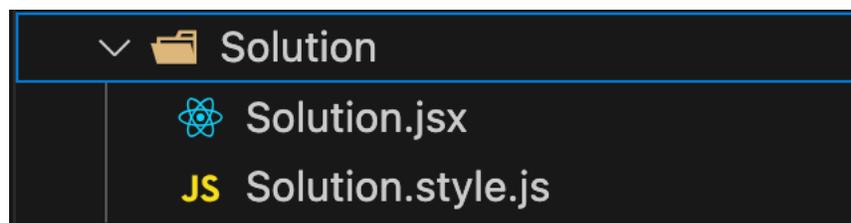


Ilustración 25 Página Solution.

```
import { ScrollView, Text, View } from "react-native";  
import { s } from "./Solution.style";  
import { SafeAreaProvider, SafeAreaView } from "react-native-  
safe-area-context";  
import { ExerciseSolution, NavHeaderSolution } from  
"../../components";  
import { useRoute } from "@react-navigation/native";  
import solutionData from "../../data/solutions.json";  
import { useEffect, useState } from "react";  
  
const Solution = () => {  
  const { params } = useRoute();  
  
  const [ selectedSolution, setSelectedSolution ] =  
  useState("");
```

```

const [ topicTitle, setTopicTitle ] = useState("");

//Hook que carga la solución asociada al ejercicio
seleccionado
useEffect(()=> {
    loadSelectedSolution();
}, [])

// Función que filtra la solución mediante el id del
ejercicio seleccionado
const filteredSolution = (exerciseId) => {
    if(exerciseId !== 0) {
        return solutionData.filter((solution) =>
solution._exercise[0].id == exerciseId);
    }
    return[];
};

// Función que carga la solución del ejercicio seleccionado
const loadSelectedSolution = () => {
    try {
        const filterSolution = filteredSolution(params.id)
        setSelectedSolution(filterSolution[0].explanation ||
""");

setTopicTitle(filterSolution[0]._exercise[0]._topic[0].title);
    } catch (error) {
        alert(error);
    }
}

return(
    <>
        <SafeAreaProvider>

```

```

        <SafeAreaView style={s.root} >
          <View style={s.nav_header} >
            <NavHeaderSolution title={topicTitle} />
          </View>
          <View style={s.body} >
            <ScrollView >
              <ExerciseSolution
solution={selectedSolution} />
            </ScrollView>
          </View>
        </SafeAreaView>
      </SafeAreaProvider>
    </>
  )
};

export default Solution;

import { StyleSheet } from "react-native";

export const s = StyleSheet.create({
  root: {
    flex: 1,
    backgroundColor: "#F9F9F9"
  },
  nav_header: {
    flex: 1,
    paddingBottom: 25,
    paddingHorizontal: 10
  },
  body: {
    flex: 5,
    paddingHorizontal: 15
  },
});

```

Página para ver la teoría asociada al ejercicio:

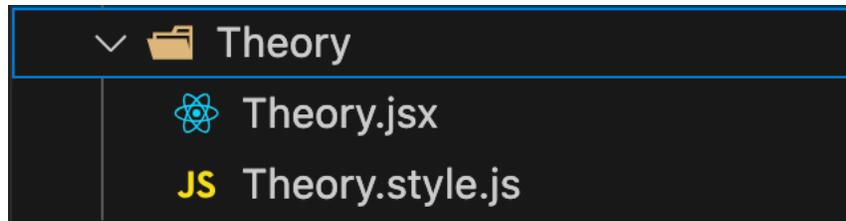


Ilustración 26 Página Theory.

```
import { SafeAreaView, SafeAreaView } from "react-native-safe-area-context";
import { s } from "./Theory.style";
import { useRoute } from "@react-navigation/native";
import { ScrollView, View } from "react-native";
import { NavHeaderTheory, TabBottomMenuTheory, TopicTheory } from
"../../components";
import { useEffect, useState } from "react";
import theoryData from "../../data/theory.json";

const Theory = ({onPressHelp}) => {
  const { params } = useRoute();

  // Hook para inicializar el tab en Theory
  const [ selectedTabName, setSelectedTabName ] =
useState("Theory");

  const [ theorySelected, setTheorySelected ] = useState("");

  const [ topicTitle, setTopicTitle ] = useState("");

  //Hook que carga la teoría asociada al ejercicio seleccionado
  useEffect(() => {
    loadSelectedTheory();
  }, [])
```

```

// Función que filtra la teoría mediante el id del ejercicio
seleccionado
const filteredTheory = (exerciseId) => {
  if(exerciseId !== 0){
    return theoryData.filter((theory) =>
theory._exercise[0].id == exerciseId)
  }
  return []
}

// Función que carga la teoría del ejercicio seleccionado
const loadSelectedTheory = () => {
  try {
    const filterTheory =
filteredTheory(params.exerciseId);
    setTheorySelected(filterTheory[0].explanation || "");
setTopicTitle(filterTheory[0]._exercise[0]._topic[0].title ||
"");
  } catch (error) {
    alert(error);
  }
}

return(
  <>
    <SafeAreaView>
      <SafeAreaView style={s.root} >
        <View style={s.nav_header} >
          <NavHeaderTheory
            title={topicTitle}
            onPress={onPressHelp}
          />

```

```

        </View>
        <View style={s.body}>
          <ScrollView>
            <TopicTheory theory={theorySelected}
          />
          </ScrollView>
        </View>
      </SafeAreaView>
    </SafeAreaProvider>
    <View style={s.footer} >
      <TabBottomMenuTheory
        selectedTabName={selectedTabName}
        onPress={setSelectedTabName}
        exerciseNum={params.exerciseNum}
      />
    </View>
  </>
);
};

```

```
export default Theory;
```

```
import { StyleSheet } from "react-native";
```

```

export const s = StyleSheet.create({
  root: {
    flex: 1,
    backgroundColor: "#F9F9F9"
  },
  nav_header: {
    flex: 1,
    paddingBottom: 25,
    paddingHorizontal: 10
  },
  body: {

```

```

        flex: 5,
        paddingHorizontal: 15
    },
    footer: {
        height: 80,
        backgroundColor: "white",
    }
});

```

Componente App donde se encuentra el ruteo de navegación entre páginas:



Ilustración 27 Componente contenedor App.

```

import { CopyRight, Exercise, Home, Question, Solution, Theory }
from "./Pages";
import { NavigationContainer } from "@react-navigation/native";
import { createNativeStackNavigator } from "@react-
navigation/native-stack";
import Dialog from "react-native-dialog";
import { useState } from "react";
import { useFonts } from "expo-font";
const Stack = createNativeStackNavigator();

export default function App() {

    const [isHelpDialogDisplay, setIsHelpDialogDisplay] =
useState(false);
    const [isHelpDialogQuestionDisplay,
setIsHelpDialogQuestionDisplay] = useState(false);
    const [isHelpDialogTeoryDisplay, setIsHelpDialogTeoryDisplay] =
useState(false);
    const [isInfoDialogDisplay, setIsInfoDialogDisplay] =
useState(false);

```

```

const [isFontLoaded] = useFonts({
  "Alata-Regular": require("./assets/fonts/Alata-Regular.ttf"),
})

//Diálogo de ayuda para selección de ejercicios
const renderHelpDialog = () => {
  return(
    <Dialog.Container visible={isHelpDialogDisplay}
onBackdropPress={() => setIsHelpDialogDisplay(false)} >
      <Dialog.Title >Instrucciones</Dialog.Title>
      <Dialog.Description>Selecciona uno de los ejercicios
propuestos, resuelvelo y elige la respuesta correcta entre los
incisios indicados</Dialog.Description>
      <Dialog.Button label="Aceptar" onPress={() =>
setIsHelpDialogDisplay(false)}/>
    </Dialog.Container>
  );
};

//Diálogo de ayuda para selección de respuesta
const renderHelpDialogQuestion = () => {
  return(
    <Dialog.Container visible={isHelpDialogQuestionDisplay}
onBackdropPress={() => setIsHelpDialogQuestionDisplay(false)} >
      <Dialog.Title >Instrucciones</Dialog.Title>
      <Dialog.Description>Resuelve el ejercicio y elige la
respuesta correcta entre los incisos
mostrados</Dialog.Description>
      <Dialog.Button label="Aceptar" onPress={() =>
setIsHelpDialogQuestionDisplay(false)}/>
    </Dialog.Container>
  );
};

```

```

//Diálogo de ayuda en la sección de teoría
const renderHelpDialogTheory = () => {
  return(
    <Dialog.Container visible={isHelpDialogTheoryDisplay}
onBackdropPress={() => setIsHelpDialogTheoryDisplay(false)} >
      <Dialog.Title >Instrucciones</Dialog.Title>
      <Dialog.Description>En esta sección encontraras la teoría
asociada al tema seleccionado</Dialog.Description>
      <Dialog.Button label="Aceptar" onPress={() =>
setIsHelpDialogTheoryDisplay(false)} />
    </Dialog.Container>
  );
};

```

```

//Diálogo que muestra la última actualización
const renderInfoDialog = () => {
  return(
    <Dialog.Container visible={isInfoDialogDisplay}
onBackdropPress={() => setIsInfoDialogDisplay(false)} >
      <Dialog.Title >Última actualización</Dialog.Title>
      <Dialog.Description>Enero 2025</Dialog.Description>
      <Dialog.Button label="Aceptar" onPress={() =>
setIsInfoDialogDisplay(false)} />
    </Dialog.Container>
  );
};

```

```

return (
  <>
    <NavigationContainer>
      <Stack.Navigator
        screenOptions={{headerShown: false,}}
        initialRouteName="Home"

```

```

    >
      <Stack.Screen name="Home">
        {() => <Home onPressInfo={() =>
setIsInfoDialogDisplay(true)}/>}
      </Stack.Screen>
      <Stack.Screen name="Exercise" >
        {() => <Exercise onPressHelp={() =>
setIsHelpDialogDisplay(true)} />}
      </Stack.Screen>
      <Stack.Screen name="Question" >
        {() => <Question onPressHelp={() =>
setIsHelpDialogQuestionDisplay(true)} />}
      </Stack.Screen>
      <Stack.Screen name="Solution" component={Solution} />
      <Stack.Screen name="Theory">
        {() => <Theory onPressHelp={() =>
setIsHelpDialogTheoryDisplay(true)} />}
      </Stack.Screen>
      <Stack.Screen name="CopyRight">
        {() => <CopyRight onPressInfo={() =>
setIsInfoDialogDisplay(true)} />}
      </Stack.Screen>
    </Stack.Navigator>
  </NavigationContainer>
  {renderHelpDialog()}
  {renderHelpDialogQuestion()}
  {renderHelpDialogTheory()}
  {renderInfoDialog()}
</>
);
}

```