



UNIVERSIDAD DE TALCA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN DESARROLLO DE
VIDEOJUEGOS Y REALIDAD VIRTUAL

Desarrollo de aplicación móvil interactiva de la
Guías de Buenas Prácticas: Cuidados Centrados
en la Persona y Familia, para estudiantes y
docentes de enfermería

ALAN MONSALVE SALAS

Profesor Guía: CLAUDIA DE LA FUENTE BURDILES

Profesor Co-guía: BORIS VERA BARRERA

Memoria para optar al título de
Ingeniero en Desarrollo de Videojuegos y Realidad Virtual

Talca – Chile
Julio, 2026



UNIVERSIDAD DE TALCA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN DESARROLLO DE
VIDEOJUEGOS Y REALIDAD VIRTUAL

Desarrollo de aplicación móvil interactiva de la
Guías de Buenas Prácticas: Cuidados Centrados
en la Persona y Familia, para estudiantes y
docentes de enfermería

ALAN MONSALVE SALAS

Profesor Guía: CLAUDIA DE LA FUENTE BURDILES

Profesor Co-guía: BORIS VERA BARRERA

Memoria para optar al título de
Ingeniero en Desarrollo de Videojuegos y Realidad Virtual

El presente documento fue calificado con nota: _____

Talca – Chile

Julio, 2026

Dedicado a mi Mamá, Papá y Hermanos.

AGRADECIMIENTOS

Agradezco a todas las personas que me apoyaron durante el desarrollo de este proyecto: a mis padres por la confianza, el apoyo y la perseverancia para realizar este camino, y a todos mis amigos, en especial a Sofía Gómez, Gabriel Iturrieta, Alejandro Herrera, Naomi Tobar e Ignacio Silva, por estar siempre con palabras de aliento en el proceso. Gracias también a mis profesores guía y co-guía, siempre atentos y presentes en cada etapa. Especialmente a mi profesora guía Claudia de la Fuente, quien tuvo infinita paciencia y dedicación, nunca perdiendo la esperanza de que lograría terminar mi memoria de título.

TABLA DE CONTENIDOS

	página
Dedicatoria	I
Agradecimientos	II
Tabla de Contenidos	III
Índice de Figuras	VI
Índice de Tablas	VIII
Resumen	IX
1. Introducción	1
1.1. Contexto del problema	3
1.1.1. Contexto de la solución	3
1.2. Alcance	4
1.3. Objetivos	4
1.4. Pregunta de investigación	5
1.4.1. Esquema del Documento	5
2. Estado del Arte	6
2.1. Guías de buenas prácticas	6
2.2. Aplicaciones y guías de Bolsillo	7
2.2.1. CuidaVen	7
2.3. Las TIC en el área de la salud y educación	7
3. Marco Teórico	8
3.1. SDK Android Studio	8
3.1.1. Flutter	8
3.2. Metodologías de Trabajo	9
3.2.1. Modelo de desarrollo	9
3.2.2. Tablero Kanban	10
3.3. Guía clínica: Cuidados Centrados en la Persona y la Familia	10

3.4.	Diseño centrado en el usuario	10
3.4.1.	Principios del Diseño Centrado en el Usuario	11
3.5.	Experiencia de usuario	12
3.6.	Usabilidad	12
4.	Marco Metodológico	13
4.1.	Modelo Incremental	13
4.2.	Herramienta KanbanFlow	13
4.3.	Diseño centrado en el usuario	14
5.	Diseño	15
5.1.	Herramienta de desarrollo del software	15
5.2.	Arquitectura del Software	16
5.3.	Diagrama de flujo del Software	17
5.4.	Boceto de diagrama de navegación	18
5.5.	Planificación	19
5.5.1.	Documento de Diseño	19
5.5.2.	Carta Gantt	19
5.6.	Diseño de pruebas	20
5.6.1.	Encuesta User Experience Questionnaire	20
5.6.2.	Encuesta System Usability Scale	21
6.	Implementación	22
6.1.	Incrementos	22
6.1.1.	Primer Incremento	22
6.1.2.	Segundo Incremento	26
6.1.3.	Tercer Incremento	33
7.	Resultados	36
7.1.	Realización de pruebas de usuarios	36
7.2.	Gráfico UEQ	36
7.3.	Gráfico SUS	37
7.4.	Conclusión de análisis	38
8.	Conclusiones y trabajo a futuro	39

Glosario	41
Bibliografía	42
A: Anexos	45
A.1. Guías de buenas prácticas	45
A.1.1. Recomendaciones y Evidencia de la guía cuidados centrado en la persona y familia	47
A.1.2. Interpretación de la evidencia	50
A.2. Comparativa entre aplicaciones	52
A.3. Atributos de usabilidad	52
A.4. Análisis de los resultados de la encuestas	53
A.4.1. Grabaciones de la aplicación	53
A.5. Carta Gantt	53
A.6. Consentimiento informado y resultados	54
A.6.1. Consentimiento informado	54
A.6.2. Resultados de la Encuesta UEQ	55
A.6.3. Resultados de la Encuesta SUS	64
A.7. Implementación	74
A.7.1. Figuras Primer incremento	74
A.7.2. Figuras segundo incremento	78
A.7.3. Figuras tercer incremento	84

ÍNDICE DE FIGURAS

	página
5.1. Diagrama de capas Arquitectura del software	16
5.2. Diagrama de flujo del software	17
5.3. Boceto de diagrama de navegación Usuario.	18
5.4. Boceto de diagrama de navegación Administrador.	19
6.1. Tablas de SupaBase	23
6.2. Pantalla de Librería	25
6.3. Pantalla de Galería	25
6.4. Pantalla de inicio de sesión	27
6.5. Pantalla de minijuego de preguntas	30
6.6. Pantalla de retroalimentación	30
6.7. Pantalla de minijuego de memoria	32
6.8. Pantalla de minijuego de memoria voltear cartas	32
6.9. Código Login y asignación de rol de usuario	34
7.1. Promedios de puntajes UEQ	37
7.2. Puntaje escala SUS	38
A.1. Carta Gantt	53
A.2. Consentimiento informado presentado a los usuarios.	54
A.3. UEQ Pregunta 1	55
A.4. UEQ Pregunta 2	56
A.5. UEQ Pregunta 3	57
A.6. UEQ Pregunta 4	58
A.7. UEQ Pregunta 5	59
A.8. UEQ Pregunta 6	60
A.9. UEQ Pregunta 7	61
A.10.UEQ Pregunta 8	62
A.11.UEQ Pregunta 9	63
A.12.SUS Pregunta 1	64
A.13.SUS Pregunta 2	65
A.14.SUS Pregunta 3	66

A.15.SUS Pregunta 4	67
A.16.SUS Pregunta 5	68
A.17.SUS Pregunta 6	69
A.18.SUS Pregunta 7	70
A.19.SUS Pregunta 8	71
A.20.SUS Pregunta 9	72
A.21.SUS Pregunta 10	73
A.22.KanbanFlow Primer Incremento.	74
A.23.Dependencias de Flutter	74
A.24.Linea de Inicialización	75
A.25.Script Plantilla de llamado de datos	75
A.26.Scripts Provider de cada Tabla	76
A.27.Funciones Tabla de Librería	76
A.28.Código Librería	77
A.29.Código Controlador Librería	77
A.30.KanbanFlow Segundo Incremento.	78
A.31.Código Binding GetX	78
A.32.Código Controlador	79
A.33.Código FormBuilderTextField	79
A.34.Función plantilla de opciones	80
A.35.Función load	80
A.36.Función next	81
A.37.Función finalizarJuego	81
A.38.Código de Clase CardModel	82
A.39.Función selectCard	82
A.40.Función checkMatch	83
A.41.Función flipAllCards	83
A.42.Código FlipAnimation	84
A.43.KanbanFlow Tercer Incremento.	84
A.44.Código añadir Concepto y Caso clínico	85
A.45.Código eliminar Concepto y Caso clínico	85
A.46.Variables TextInput	86
A.47.Dependencias para comprobar conexión a internet	86
A.48.Función que comprueba el estado de la red	86

ÍNDICE DE TABLAS

	página
3.1. Ventajas modelo incremental	9
A.1. Comparativa entre Atlas y la app Cuidaven	52

RESUMEN

La escasez de recursos digitales basadas en Guías de Buenas Prácticas de la Asociación de Enfermeras Registradas de Ontario (GBP-RNAO) para docentes y estudiantes de enfermería representa un desafío significativo al impartir materias que requieren la interacción con pacientes, ya que obstaculiza un aprendizaje más dinámico y dificulta el acceso inmediato a la información necesaria para docentes y estudiantes. Con el fin de abordar esta problemática, se propuso desarrollar una aplicación interactiva para dispositivos móviles mediante el uso de Flutter, la cual permita acceder de manera más simple a los contenidos de la guía de buenas prácticas de cuidados centrados en la persona y la familia, para estudiantes de enfermería. La aplicación cuenta con un apartado con dos minijuegos, el primero de preguntas y el segundo de memoria, los contenidos de estos están basados en casos clínicos de la guía de buenas prácticas, con el enfoque de generar un aprendizaje más interactivo y ofreciendo retroalimentación al completar los minijuegos. Además, contiene videos con recomendaciones de la guía y otros tipos de contenidos audiovisuales que el docente disponga, con los que se podrán reforzar los contenidos vistos en la guía y sus recomendaciones. Con esto se espera introducir el uso de nuevas tecnologías para la integración de contenidos interactivos sobre los puntos fundamentales de las guías de buenas prácticas de cuidados centrados en la persona y la familia. También generando un impacto positivo con el uso de tecnologías y/o aplicaciones para potenciar los contenidos que se les presentan a los estudiantes. Los resultados obtenidos por las encuestas System Usability Scale (SUS) y User Experience Questionnaire (UEQ) demuestran una aceptación favorable por parte de los usuarios encuestados, resultando en una herramienta adecuada para la consulta de información fiable sobre los contenidos de la guía de buenas prácticas cuidados centrados en la persona y familia.

1. Introducción

Las guías sobre buenas prácticas de la Asociación de Enfermeras Registradas de Ontario representan un estándar de calidad y son una parte fundamental al momento del manejo clínico. Estas guías son un apoyo activo que se utiliza a nivel internacional por parte de especialistas del área de enfermería u otros profesionales de la salud, así como administradores y educadores [1].

Este tipo de guías busca estandarizar la atención clínica de modo que los diversos centros de salud de los distintos niveles de atención, puedan ofrecer una atención de calidad equivalente mediante recomendaciones basadas en evidencia científica. La solidez de cada recomendación depende directamente del nivel de evidencia que la respalda; desde revisiones sistemáticas y ensayos clínicos aleatorizados hasta consensos de expertos, lo cual determina el grado de confianza con que los profesionales de enfermería pueden aplicarla en su labor. [1]. Estas guías ayudan a capacitar a los profesionales de enfermería y permitirles desarrollar competencias fundamentales en el desempeño de su profesión. Entre estas competencias mencionadas por [2] se encuentra: la aplicación del método científico para el análisis y resolución de problemas inherentes a sus tareas profesionales, el desarrollo del pensamiento crítico en la interpretación de la información y la aplicación de conocimientos generados por la disciplina para perfeccionar la calidad de los cuidados de enfermería.

El uso de estas guías es importante, ya que entrega los recursos necesarios basados en evidencia a las y los enfermeros. La guía que se toma en consideración, denominada Cuidados Centrados en la Persona y la Familia según [3], tiene como objetivo promover las prácticas basadas en evidencia en los cuidados de la persona y la familia, junto con permitirles adquirir el conocimiento y habilidades necesarios para ser más competentes. Además, cuentan con sugerencias que les facilitan colaborar

y establecer relaciones con las personas y familias que accedan a los sistemas de salud. La forma en que los contenidos de la guía se exhiben en la universidad para enfermería es de manera convencional, como introducción, los profesores emplean una diapositiva con la información de la guía de buenas prácticas Centrada en el Cuidado de la Persona y la Familia para abordar la parte teórica del proceso. Luego el profesor presenta un caso clínico creado por él para contextualizar la información. En la parte práctica se les permite a los estudiantes ser voluntarios para trabajar con un profesor encargado de alguna guía de buenas prácticas, colaborando con el docente para desarrollar trabajos enfocados en la difusión de las recomendaciones de las guías a través de distintos medios de comunicación, en los cuales se pueden encontrar: YouTube, redes sociales, infografías y material educativo.

En los recursos disponibles en internet, se percibió una escasa digitalización de los recursos existentes, que son demasiado tradicionales, con opción únicamente para encontrar documentos de texto [4], impidiendo un aprendizaje interactivo y un acceso inmediato a los elementos cruciales de la guía de buenas prácticas. Para solucionar este problema, se presenta una propuesta de aplicación interactiva para dispositivos móviles con un enfoque lúdico, se desarrollará en el framework Flutter especializado en el desarrollo de aplicaciones, permitiendo acceso rápido y sencillo a contenidos sobre la guía de buenas prácticas de Cuidados Centrados en la Persona y la Familia; esta aplicación tendrá 2 apartados para el usuario: el primer apartado permite acceder a la información fundamental de la guía (en formato de biblioteca o diccionario), incluyendo una galería con videos informativos presentados por el docente; el segundo apartado es la sección interactiva en donde, se debe seleccionar uno de los dos minijuegos basados en casos clínicos, donde al completarlos, el estudiante recibirá una retroalimentación en pantalla. En el caso del docente este dispondrá de uno adicional en donde se podrá cargar la información para los minijuegos.

Con esto se espera aprovechar las tecnologías para la integración de contenidos interactivos para los estudiantes de enfermería, sobre los puntos fundamentales de la guía de buenas prácticas enfocada en los Cuidados Centrados en la Persona y la Familia. Adicionalmente, se busca entregar un recurso de fácil acceso para los estudiantes que les ayude a reforzar y repasar los contenidos clave.

Esta aplicación interactiva se aplicará en la carrera de enfermería de la Universidad de Talca en un módulo impartido por un profesor denominado Gerontología, que es enseñado a los estudiantes del área de salud, utilizando como base la guía de buenas

prácticas Cuidado Centrado en la Persona y la Familia.

1.1. Contexto del problema

Mediante el análisis e investigación de las herramientas digitales de apoyo disponibles para la enseñanza de la guía de buenas prácticas cuidados centrados en la persona y familia, se identificó una clara escasez de recursos interactivos destinados a los docentes, lo cual limita la implementación de un aprendizaje activo. Al disponer únicamente de materiales basados en texto, se restringen los procesos de enseñanza-aprendizaje. Asimismo, esta situación obstaculiza la adquisición de información esencial debido al tiempo excesivo que conlleva encontrar el contenido clave en las guías, tanto por parte de los estudiantes como de los profesores al aplicar las recomendaciones.

1.1.1. Contexto de la solución

Para abordar esta problemática, se presentó la idea de crear una aplicación interactiva para dispositivos móviles, que facilite el acceso a los contenidos importantes de la guía de buenas prácticas, especialmente en la guía de cuidados centrados en la persona y la familia. Esta aplicación contará con dos apartados: el primero incluirá las recomendaciones fundamentales de la guía, permitiendo un acceso rápido y conteniendo material de apoyo basado en videos informativos sobre cada una; en el segundo apartado, se presenta la parte más interactiva, donde el usuario deberá seleccionar un minijuego de tipo juego de preguntas o memorice. En estos, se expondrán casos clínicos con los cuales el usuario deberá trabajar para completarlos. Al final, se presentará una retroalimentación para el usuario que contendrá tanto las respuestas correctas como las incorrectas; por cada una de estas preguntas se presentará una justificación de la elección escrita por el profesor. Adicionalmente, en la sección del administrador, se hallará un botón adicional que permitirá la incorporación de nuevos contenidos para los dos minijuegos; en este caso sería la incorporación de conceptos y definiciones, nuevos casos clínicos con sus correspondientes respuestas y retroalimentación.

1.2. Alcance

El alcance de este proyecto incluye, como mínimo, la biblioteca con categorías y sus respectivas recomendaciones, junto a los niveles de evidencia de cada una. En cuanto a la funcionalidad interactiva, se planea incluir al menos dos minijuegos basados en casos clínicos. Uno de ellos podría ser un tipo de juego de preguntas donde se presenta un caso clínico compuesto por una pregunta, un escenario y tres posibles respuestas, y consta de una ronda de nueve preguntas. En el minijuego de memorice, el objetivo será emparejar correctamente todas las tarjetas antes de que se acabe el tiempo. Se permitirá al jugador ver las tarjetas gastando una de las fichas de revés, contando con cinco al inicio del minijuego. Al terminar cualquiera de estos juegos, se ofrecerá retroalimentación sobre los aciertos y errores; en el caso del memorice, los pares correctos se marcarán con un color distinto para ayudar al estudiante a asociarlos correctamente.

1.3. Objetivos

En este apartado del documento se presentarán los distintos objetivos que se plantean para el desarrollo de este proyecto.

Objetivo General: Desarrollar una aplicación interactiva en Flutter que facilite el acceso inmediato mediante recursos lúdicos y audiovisuales a los contenidos de la guía de buenas prácticas basada en Cuidados Centrados en la Persona y la Familia para estudiantes y docentes de enfermería.

Objetivo específico 1: Identificar requisitos funcionales y no funcionales para el desarrollo de la aplicación interactiva.

Objetivo específico 2: Diseñar contenidos de casos clínicos basados en requisitos y guía de buenas prácticas basada en Cuidados Centrados en la Persona y la Familia presentada por la Asociación de Enfermeras Registradas de Ontario aplicando diseño centrado en el usuario.

Objetivo específico 3: Implementar una aplicación interactiva funcional según diseño y requerimientos del usuario.

Objetivo específico 4: Evaluar la aplicación interactiva mediante una encuesta de usabilidad y experiencia de usuario, para identificar los puntos

de mejora del software.

Objetivo específico 5: Analizar los resultados de las encuestas con el propósito de verificar la aceptación del software.

1.4. Pregunta de investigación

¿En qué medida el desarrollo de una aplicación interactiva, evaluada bajo criterios de usabilidad y experiencia del usuario, logra superar los umbrales de las encuestas SUS y UEQ, para que se convierta una herramienta aceptable para la consulta de contenidos de la guía de buenas prácticas?

1.4.1. Esquema del Documento

Para facilitar la comprensión del documento, se pondrá a disposición la estructura del documento que se verá a continuación: Estado del arte, es donde se presentarán los materiales y aplicaciones que buscan resolver esta problemática desde distintos enfoques. El capítulo de marco teórico se presentarán las herramientas como el framework y metodologías de trabajo con las cuales se plantea el desarrollo del proyecto. Marco metodológico, se presentará la manera en que se implementarán las metodologías de trabajo, la definición de las tareas y el uso de las herramientas de planificación. Diseño es en donde se presentan las herramientas de desarrollo y su justificación de uso para este proyecto, también encontramos la planificación como el documento de diseño, carta gantt y el tipo de encuestas que se utilizarán para las pruebas de usuarios. Implementación en donde se aborda la manera en que se desarrolla el proyecto con sus diferentes incrementos. Los resultados de cada una de las encuestas que se realizaron durante la pruebas de usuario cada una cuenta con sus correspondientes análisis. Finalmente encontramos las conclusiones y el trabajo a futuro. El último apartado es de los anexos, es donde se presentan los documentos, carta gantt, documento de diseño, carta de aceptación e imágenes.

2. Estado del Arte

En este inciso se presentarán los materiales de apoyo que disponen los docentes para la enseñanza de los contenidos de las guías de buenas prácticas a estudiantes de enfermería.

2.1. Guías de buenas prácticas

Las guías de buenas prácticas son documentos desarrollados sistemáticamente, por la RNAO que se sustentan en la evidencia con el fin de lograr resultados efectivos, eficientes y de alta calidad en los servicios de salud[5]. Estas guías presentan recomendaciones para distintos funcionarios o estudiantes del área de la salud, donde se pueden destacar a los enfermeros y los docentes que se especializan en la formación de nuevos enfermeros; el fin del uso de estas guías es mejorar los resultados y el compromiso con los pacientes y sus redes de apoyo. También estas guías en su versión en español se pueden encontrar en la página de la BPSO España [6]. Esta BPSO es un programa promovido por la RNAO en donde selecciona a instituciones de salud o académicas para implantar y evaluar las guías de buenas prácticas, proporcionando apoyo a las instituciones comprometidas para generar un impacto positivo en los cuidados de los pacientes a través de prácticas clínicas. En el anexo A.1 se presentarán todas las guías clínicas disponibles por la Asociación de Enfermeras de Ontario; cabe destacar que se está usando el nombre traducido por la BPSO y es posible que no todas estén en español.

2.2. Aplicaciones y guías de Bolsillo

Dentro de los recursos disponibles se pueden encontrar una variedad de elementos en donde se presentan las guías de bolsillo, Accesos vasculares para enfermeras, ictus para enfermeras e ictus para TCAE [7]. Junto con esto se incluye la aplicación móvil Cuidaven, enfocada en el manejo de dispositivos venosos [8], esta app tiene el fin de mejorar las competencias de las enfermeras. Por ultimo recursos complementarios como compendios, webinarios y folletos orientados a la educación.[7].

2.2.1. CuidaVen

Es una aplicación diseñada para Android e iOS, reúne recomendaciones basadas en evidencia sobre el manejo de dispositivos venosos. Está dirigida tanto para profesionales y estudiantes sanitarios como a personas adultas, pediátricas y neonatos. Del mismo modo, ofrece videos informativos sobre el cuidado y manejo de dispositivos venosos, además de un espacio para compartir los conocimientos entre los profesionales [8]. Véase la tabla comparativa del Anexo A.1.

2.3. Las TIC en el área de la salud y educación

La implementación de las Tecnologías de la información y comunicación (TIC) constituye instrumentos que fortalecen, asisten y mejoran el modelo sanitario de la actualidad. Según [9], este enfoque mejora la manera en que los especialistas en salud interactúan, participan y cooperan, así como la manera en que pacientes y profesionales se aproximan a las áreas de información de la salud.

Para observar el efecto de las TIC en el área de la educación para la formación de enfermeros, se presenta el siguiente artículo [10], en donde se abordan los resultados del uso de aplicaciones móviles para un proceso de enseñanza-aprendizaje en estudiantes de enfermería de la Ciudad de México, en donde el autor menciona que las TIC deben funcionar como una herramienta de apoyo para los docentes, los cuales necesitarán capacitaciones para hacer uso de estas durante el desarrollo de las clases. Además, este autor de [10] llega a concluir que las TIC tienen la función de innovar en espacios de formación universitaria y que los estudiantes contarán con herramientas de aprendizaje que dispondrán de actualizaciones constantes para estar al día con la información de la actualidad.

3. Marco Teórico

En este apartado se presentan las herramientas y metodologías de trabajo que se emplearán en el desarrollo de este proyecto.

3.1. SDK Android Studio

Android Studio es un entorno de desarrollo enfocado en aplicaciones para dispositivos móviles [11]; este cumple la función de ser un SDK o kit de desarrollo de software, el cual se compone de una variedad de paquetes, junto con depuradores, compiladores y librerías que ayudarán al momento de diseñar una aplicación [12]. Este SDK también ofrece una amplia gama de herramientas, tales como las herramientas de líneas de comando, compilación y la de plataforma SDK de Android [13]. Además, incluye un emulador virtual denominado Android Emulator [14], que simula dispositivos Android en el ordenador con el fin de probar la aplicación desarrollada en diversos dispositivos y niveles API [15]. Los dispositivos simulados operan de manera independiente, permitiendo el uso simultáneo de múltiples dispositivos en caso de ser requeridos por el desarrollador. Además, cuenta con una amplia documentación [16] en español que brinda soporte a los desarrolladores.

3.1.1. Flutter

Flutter es un Framework multiplataforma que tiene como objetivo el desarrollo de aplicaciones para dispositivos móviles, esta es de código abierto y fue lanzado por la empresa de Google en el año 2016 [17]. Las aplicaciones desarrolladas en Flutter se pueden ejecutar en dispositivos Android e iOS [18], este posee una característica principal la cual es que depende de los Widgets OEM del dispositivo sin tener que

generarlos o ejecutarlos de manera externa. Otra de las características de Flutter es la recarga en caliente, esta permite llevar a cabo cambios al código y al software mientras se encuentra en ejecución la maquina virtual, permitiendo arreglar errores o agregar nuevas características al software sin la necesidad de volver a compilar y ejecutar nuevamente [17]. El framework hace uso de el lenguaje de programación Dart para la escritura de código y es una de las principales razones por la que funciona la recarga en caliente de Flutter [19], ademas este es muy similar a los lenguajes de programación orientados a objetos como pueden ser Java, C++ y Swift [20].

3.2. Metodologías de Trabajo

En este apartado se presentarán las metáforas de diseño que serán necesarias para la designación de tareas y gestión del flujo de trabajo del proyecto.

3.2.1. Modelo de desarrollo

El modelo incremental [21] es una metodología ágil para el desarrollo de software; esta cuenta con una característica básica, denominada bosquejo de descripción, y a medida que avanza el tiempo, con cada iteración que se va realizando, se van creando más versiones complejas del software, hasta llegar a una versión final acorde a las necesidades del usuario. Esta metodología aplica secuencias lineales al momento de avanzar en la planificación [22]. A continuación se presenta en la tabla 3.1 las ventajas que dispone el Modelo de desarrollo incremental obtenido de [23]:

Ventajas
El software se genera rápidamente durante el ciclo de vida del software.
Es flexible y menos costoso cambiar los requisitos y el alcance
A lo largo de las etapas del desarrollo se pueden efectuar cambios
Este modelo es menos costoso en comparación con otros
Los errores son fáciles de identificar

Tabla 3.1: Ventajas modelo incremental

3.2.2. Tablero Kanban

Es un tablero de gestión utilizado por la metodología Kanban [24], las columnas de este tablero muestran las tareas que se encuentran en proceso, pendientes y terminadas; para esto, las tareas cumplen la función de una tarjeta visual, la que se desplazará de izquierda a derecha dependiendo del estado en que se encuentre. Esto no solo permite observar el flujo de las tareas, sino que también proporciona una visión general del progreso del equipo que se encuentra trabajando en dicho proyecto [25]. Además, permite mejorar la eficiencia, colaboración y transparencia de los equipos al tener una comprensión clara de sus tareas actuales, lo que resulta en una gestión de proyectos más eficiente y efectiva a la hora de trabajar en un proyecto [26].

3.3. Guía clínica: Cuidados Centrados en la Persona y la Familia

Esta guía [27] es un recurso basado en evidencia que va dirigido a enfermeras y equipos de salud, que utiliza recomendaciones entregadas por la Asociación Profesional de Enfermeras de Ontario con el objetivo de promover una atención colaborativa entre el profesional de salud, el paciente y la familia. Esto incluye formación de competencia y entrega de herramientas, enfatizando la importancia de la adaptación flexible dependiendo de la situación y el individuo; la versión en español de la guía fue presentada por los Centros Comprometidos con la Excelencia en los Cuidados (BPSO) de España. Esta guía de buenas prácticas cuenta con un total de seis apartados: valoración, planificación, implantación, evaluación, formación y, por último, sistema, organización y directrices, los que cuentan con una o más recomendaciones que disponen de un nivel de evidencia y estado.

A en la sección A.1.1 de los anexos se presentarán los tipos y las recomendaciones [27] mencionadas.

3.4. Diseño centrado en el usuario

El diseño centrado en el usuario (DCU) [28] es una filosofía cuyo objetivo es crear y diseñar productos que se adapten a las necesidades y comportamientos de

los usuarios finales. Esta cuenta con unas reglas y diseños claves del DCU [29].

3.4.1. Principios del Diseño Centrado en el Usuario

- Enfoque en el usuario: El usuario debe estar en el centro de todas las decisiones de diseño en el proceso de desarrollo.
- Participación Activa del Usuario: Involucrar a los usuarios en todas las etapas del proceso, partiendo desde el proceso de investigación hasta la evaluación.
- Iteración: El diseño debe ser un proceso iterativo, donde se prueban y mejoran las ideas basadas en la retroalimentación del usuario.
- Usabilidad: Los productos deben ser fáciles de usar y entender para el usuario.
- Accesibilidad: Asegurar que los productos sean accesibles para todos los tipos de usuarios.

Además aparte de esos principios este consta con métodos esenciales en el proceso de diseño centrado en el usuario, los que buscan asegurar un producto final que cumpla con las necesidades del usuario final. Estos métodos claves son:

- Investigación y requisitos de usuario: Método utilizado para entender a los usuarios y sus necesidades, las cuales son fundamentales para definir el producto a desarrollar.
- Prototipo: Técnica para diseñar modelos del sistema que permitan evaluar y refinar el producto antes de la implementación de la versión final.
- Evaluación: Método para obtener la información sobre el diseño y la validación del producto, asegurando que cumpla las necesidades del usuario al cual se encuentra enfocado.

Estos principios y metodologías garantizan que el diseño cumpla las necesidades y expectativas del usuario [30], lo que conlleva a que el producto sea relevante, útil y satisfactorio. Esto a su vez mejora la usabilidad y experiencia general del producto.

3.5. Experiencia de usuario

La experiencia de usuario según la definición del autor [31] es un cambio emergente del concepto de usabilidad, el cual no solo busca mejorar el rendimiento del usuario en la interacción en los aspectos de eficacia, eficiencia y facilidad de aprendizaje, sino que este tiene la intención de resolver un problema estratégico de la utilidad del producto y un problema en las sensaciones de placer y diversión de los usuarios.

3.6. Usabilidad

Es la medida de la facilidad con la que el usuario utiliza un producto, este concepto tiene dos dimensiones, objetiva y subjetiva. La dimensión objetiva se puede medir mediante la observación de cómo el usuario interactúa con el producto, según [30] esta cuenta con cuatro atributos. La dimensión subjetiva es como el usuario se siente al momento de la interacción, si es agradable y sencilla la realización de las tareas, cuyos atributos particulares se detallan en el Anexo A.3.

4. Marco Metodológico

En esta sección se introducirá la manera en la que se utilizarán las metodologías y herramientas para el desarrollo de este proyecto.

4.1. Modelo Incremental

Para el desarrollo de este proyecto, se optó por la utilización de un modelo de desarrollo de software incremental, debido a que este tipo de metodología va construyendo el software final de manera progresiva. En esta metodología, se definirá un conjunto de tareas que se irán agregando al tablero Kanban. A medida que se vaya completando este grupo de tareas y se finalice el primer incremento, se creará un nuevo conjunto para el incremento dos; dependiendo de si se completaron todas las tareas en los plazos dispuestos, se podrán agregar tareas del incremento anterior al nuevo, siempre y cuando estas no se encuentren finalizadas. Este incremento continuará hasta que se dé por finalizado el proyecto.

Una de las razones para hacer uso de esta metodología corresponde a su enfoque iterativo, flexibilidad y ventajas que tiene en comparación con otras metodologías como SCRUM, Waterfall, Lean, entre otras [32].

4.2. Herramienta KanbanFlow

Al momento de implementar la metodología de trabajo, se hace uso de una herramienta en línea llamada KanbanFlow, la que proporciona una plantilla de un tablero Kanban digital, la cual permite gestionar las tareas del desarrollo del software. Este tablero está dividido en diferentes etapas del proceso que se presentan en cada

columna del tablero: Pendiente, Programación, Diseño, Pruebas y Finalizados. Para designar la importancia de cada tarea, se define una métrica de colores, que permite evaluar la dificultad y la relevancia que tiene cada una para la finalización del software. Estos colores se asignan a cada tarea al momento de comenzar a trabajar en ella; esta asignación de colores de la tarea puede variar a medida que avanza el desarrollo del software en el caso de que una tarea se complique de la manera que no se encontraba prevista.

A continuación se presentarán los colores y su contexto al momento de asignarlo en la tarea:

- Azul: Tarea pendiente
- Verde: Tarea sencilla de implementar con una relevancia mínima en el software.
- Amarillo: Tarea de complejidad intermedia, necesaria para crear un entorno óptimo para el uso del software.
- Rojo: Tarea de gran importancia; constituyen el núcleo del software; es fundamental para un funcionamiento completo en sus distintos aspectos.

4.3. Diseño centrado en el usuario

El DCU está involucrado activamente en este proyecto durante la fase de recopilación de los requerimientos funcionales de la aplicación. Además, participa en la obtención de información y contenido relevante para la aplicación, como la guía de buenas prácticas de cuidados centrados en la persona y la familia, videos informativos e ideas o recomendaciones. Otro principio clave de este diseño es la usabilidad, para lo cual se toma en cuenta la retroalimentación de los usuarios con el fin de mejorar aspectos relacionados con la experiencia de usuario. La implementación del DCU se destaca especialmente en el proceso de validación del producto, donde se emplean cuestionarios SUS y UEQ . Estos permiten recopilar datos esenciales para evaluar la calidad del producto. Las características fundamentales de este diseño son la usabilidad y una experiencia de usuario integral, lo que respalda los ajustes en el diseño final para satisfacer las necesidades del usuario. Esto incluye facilitar el acceso a los contenidos, garantizar una buena capacidad de respuesta y proporcionar información confiable a través de la guía de buenas prácticas.

5. Diseño

5.1. Herramienta de desarrollo del software

El desarrollo de esta aplicación se hará mediante el uso del SDK Android Studio complementado con el Framework de Flutter, los cuales están diseñados para la creación de aplicaciones en dispositivos móviles, ofreciendo una amplia documentación con la cual apoyarse y ofreciendo un entorno de trabajo especializado para este proyecto. El lenguaje de programación que se va a emplear para su desarrollo es Dart, el cual forma parte base del Framework mencionado con anterioridad; este posee un amplio conjunto de bibliotecas centrales para ayudar al programador; estas se pueden encontrar en [19]. Otra razón para utilizarlo son las amplias ventajas que tiene para acceder a las distintas características de los dispositivos y también el manejo de bases de datos, ya sea en estado local o en la nube. Algunas de las ventajas que se encuentran presentes en este Framework es la amplia gama de Widget personalizables [20] que ofrece para la creación de interfaces, los cuales se pueden utilizar para adaptar los casos clínicos que se buscan representar en los dos videojuegos interactivos, que corresponden a juego de preguntas y memorización. Por la parte del apartado de Biblioteca permite la creación de un diseño de interfaces organizado para facilitar el acceso a la información, videos y recomendaciones de la guía de buenas prácticas de Cuidados Centrados en la persona y familia. Finalmente un apartado único para los administradores que permite la integración de datos para los minijuegos. Se optó por este diseño, con el fin de no producir una sobrecarga visual para el usuario buscando una experiencia agradable y simple, la separación de los apartados corresponde a que cada una cumpla una función en específico, simplificando las decisiones del usuario al momento de que si quiere acceder a la información o participar en los minijuegos de la aplicación. Para las tonalidades de la interfaz se hizo uso de los colores presentes

de la BPSO, los cuales son una tonalidad Azul marino, turquesa y verde esmeralda, estos son necesarios al hacer uso de contenidos relacionados con ellos.

5.2. Arquitectura del Software

Este apartado expone la arquitectura de software basada en capas, figura 5.1. Comprende la capa de interfaz, representada por pantallas de Flutter, donde el usuario interactúa con el software. La segunda capa es la lógica, en la que se utiliza el gestor de estados GetX y las dependencias del entorno de desarrollo. Finalmente, la capa de datos incluye la base de datos Supabase junto con los scripts de los proveedores para el manejo de datos. El empleo de esta arquitectura por capas aporta una visión clara sobre el funcionamiento del sistema y su estructura, además de proporcionar características como pueden ser escalabilidad, modularidad, mantenimiento entre otras facultades.

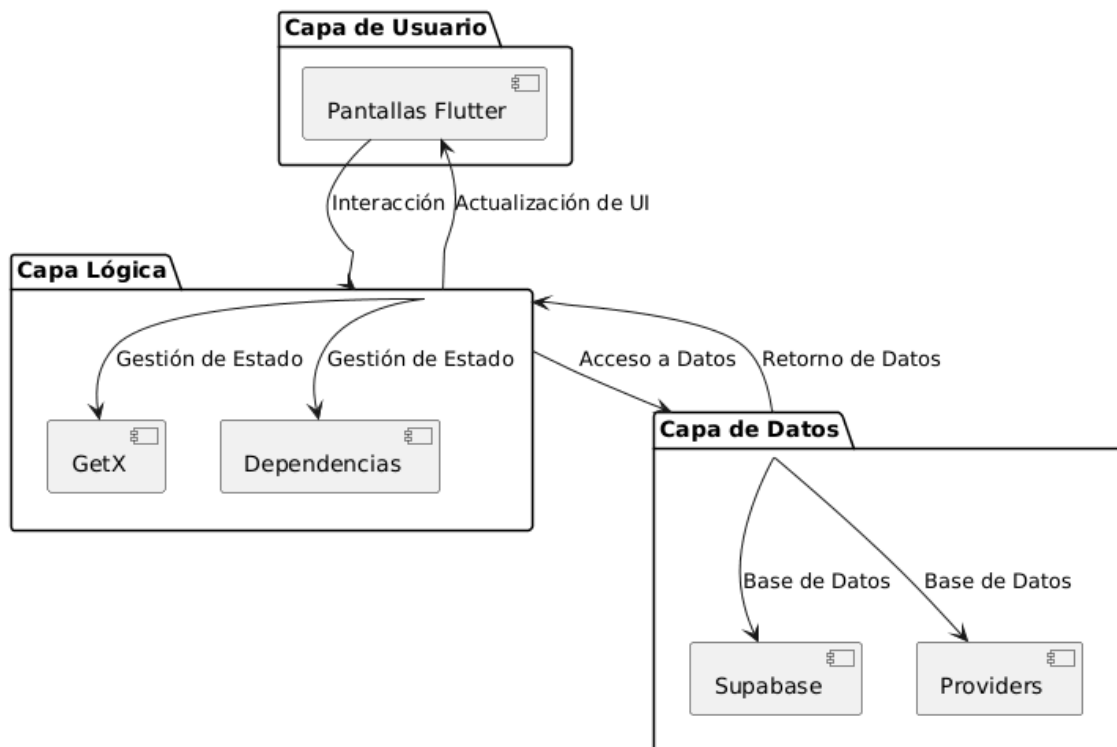


Figura 5.1: Diagrama de capas Arquitectura del software

5.3. Diagrama de flujo del Software

En esta sección del documento se presenta el diagrama de flujo que se puede encontrar en la figura 5.2. El diagrama de flujo, según [33] es una representación gráfica del flujo que tendrá el software, presentando las diferentes fases que tendrán los procesos y su funcionamiento, permitiendo comprenderlos y mejorarlos a medida que avanza el proyecto.

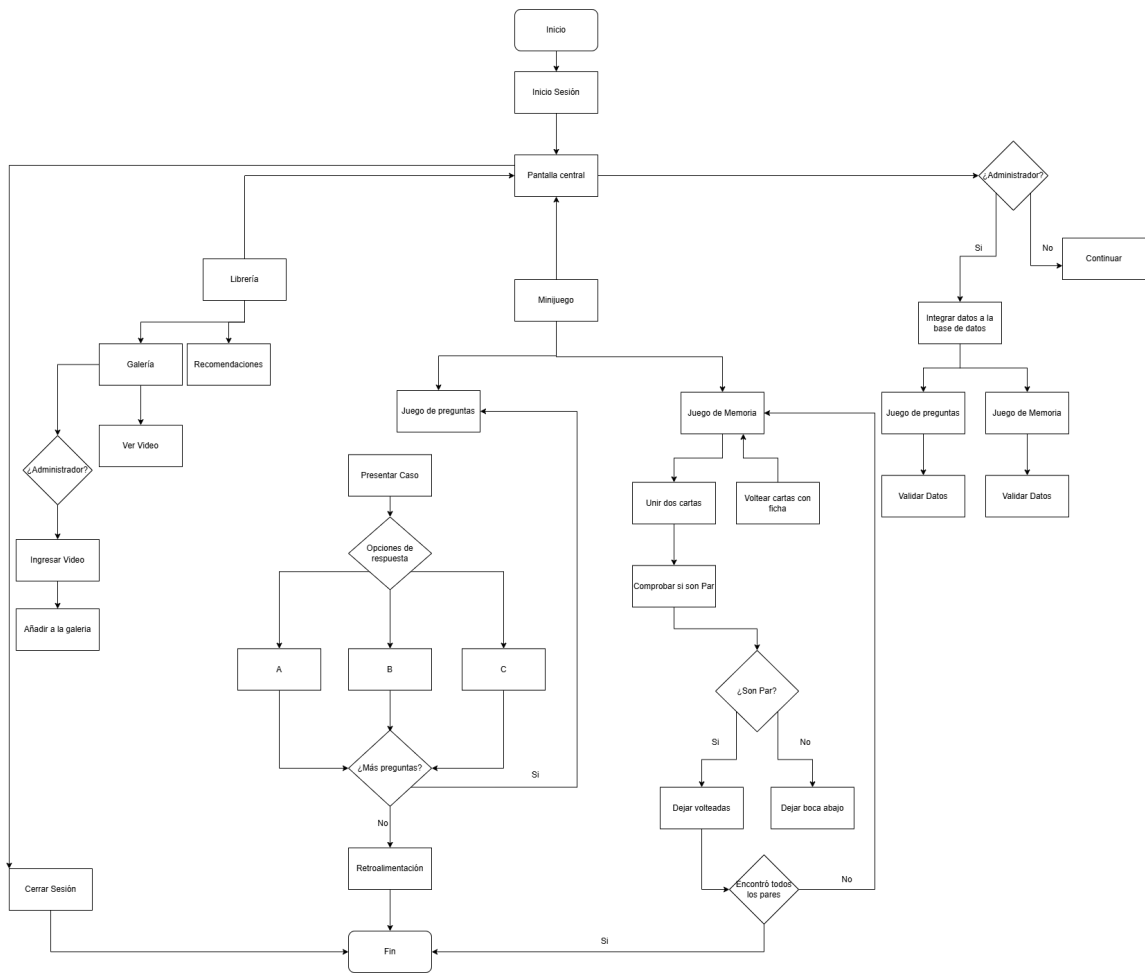


Figura 5.2: Diagrama de flujo del software

5.4. Boceto de diagrama de navegación

En este apartado del documento se presentará un boceto del diagrama de navegación que se puede encontrar en la siguiente figura 5.3 5.4, la cual es una representación gráfica de cómo se verá el software al terminar el proyecto. Es importante señalar que este diseño puede ir variando dependiendo de los ajustes y las mejoras que se obtendrán a medida que avance el desarrollo.

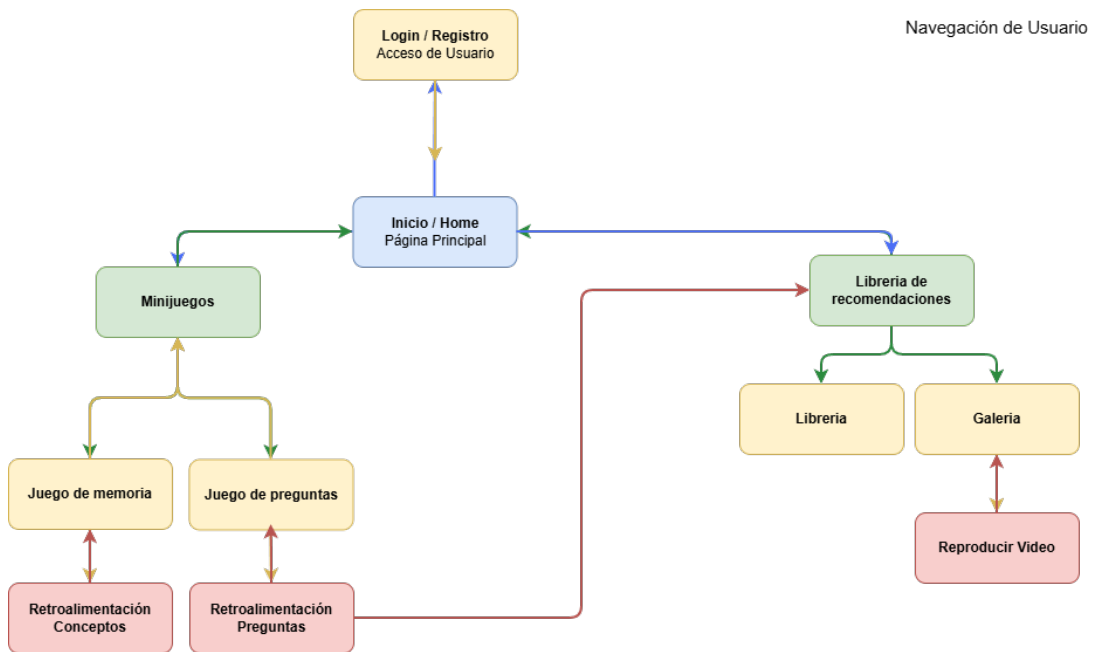


Figura 5.3: Boceto de diagrama de navegación Usuario.

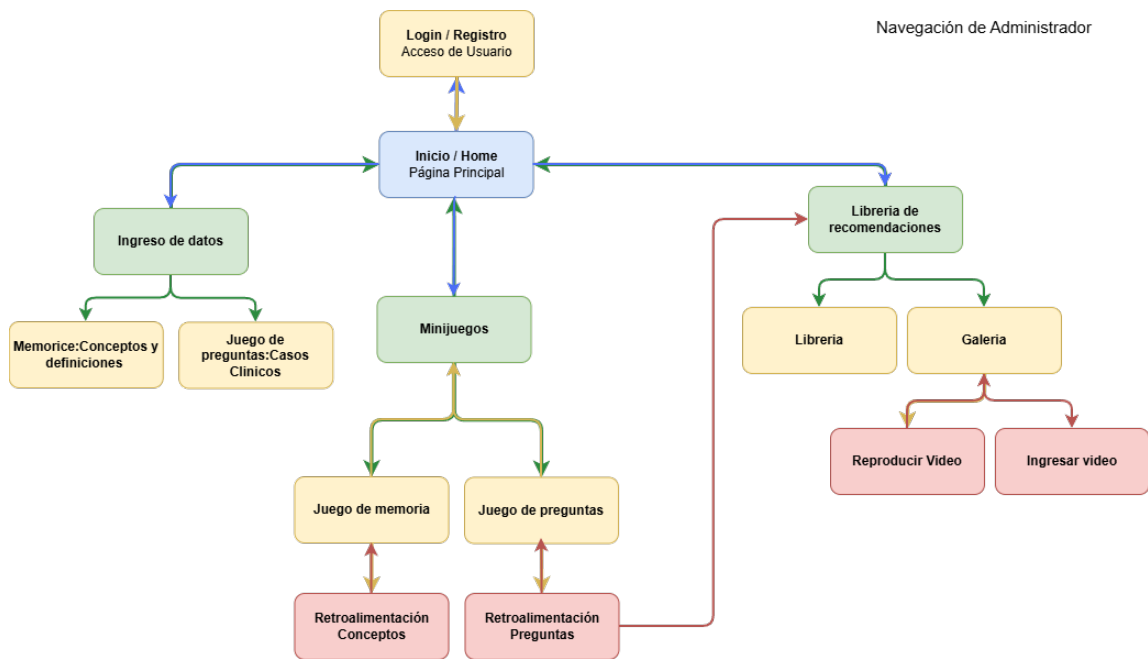


Figura 5.4: Boceto de diagrama de navegación Administrador.

5.5. Planificación

5.5.1. Documento de Diseño

En el documento de diseño se presentarán las distintas secciones claves que son esenciales al momento del desarrollo del software interactivo, como puede ser un resumen del software, público objetivo, alcance del proyecto y tecnologías o herramientas que se harán uso para el desarrollo del software. También se presentará una sección que contendrá los casos de uso y los requerimientos del software [34], los cuales se dividen en dos: funcionales y no funcionales. Documento completo en la sección de anexos A.7.3.

5.5.2. Carta Gantt

Para la creación de la carta Gantt que se muestra en el anexo con la figura A.1, se utilizó la aplicación Gantt Project, la cual es una herramienta de gestión de proyectos que permite la visualización de tareas, hitos y gestión de recursos. Esta herramienta facilita la planificación y visualización del trabajo de los proyectos.

5.6. Diseño de pruebas

Para las pruebas de validación se ha propuesto la interacción de ambos tipos de usuarios del software como se presenta en esta sección del anexo A.4.1, junto con el uso de dos encuestas estandarizadas UEQ y SUS. Cada encuestado tendrá que aceptar el consentimiento informado presentado en el anexo A.2 para hacer uso de la información recopilada, con el objetivo de obtener datos sobre la experiencia del usuario y la usabilidad del software. Para esto se contará con un mínimo de cinco estudiantes y tres profesores de enfermería, que se dispongan a responder las encuestas previstas, estas pruebas se realizarán de manera presencial con el investigador presente, se les dispondrá de un código el cual deberán escanear para instalar la aplicación en su dispositivo, se les presentara una pauta con tareas de interacción y al finalizar se presentara el código de cada una de las encuestas, los encuestados contarán con máximo una hora para realizar estas pruebas.

5.6.1. Encuesta User Experience Questionnaire

Con el fin de obtener los datos de la experiencia del usuario, se planteó la utilización de una encuesta estándar de UEQ adaptada, en donde se tiene pensado evaluar solo cuatro de los seis elementos con los que dispone este tipo de encuesta:

- Atractivo: Impresión general del producto. ¿Les gusta o no a los usuarios?
- Claridad: ¿Es fácil familiarizarse con el producto? ¿Es fácil aprender a usarlo?
- Estímulo: ¿Es emocionante y motivador utilizar el producto?
- Novedad: ¿Es el producto innovador y creativo? ¿Capta el interés de los usuarios?

Por cada uno de los elementos se tiene definido realizar entre dos a cuatro preguntas, relacionadas con los distintos aspectos del software, enfocadas en la experiencia del usuario. Los datos que se obtendrán de esta encuesta se graficarán para poder llevar a cabo un análisis. La elección de hacer uso de estos elementos, en comparación con los seis que presenta la encuesta UEQ, se debe a que los elementos faltantes, como perspicuidad y confianza, abordan preguntas de usabilidad del software y con esta encuesta no se busca obtener esos datos.

5.6.2. Encuesta System Usability Scale

Para la encuesta de usabilidad del sistema se planteó el uso de la encuesta estandarizada de SUS [35]; es una encuesta que hace uso de la escala de Likert, la cual dispone de 5 puntos. En esta escala de Likert se representan, de esta forma, las afirmaciones con las que el encuestado está de acuerdo o desacuerdo; deben ser seleccionadas de manera consciente y minuciosa. Para la aplicación de una encuesta con escala de Likert, es posible encontrar distintas medidas de usabilidad generales presentadas por el autor de [35], que deberían abarcar las siguientes:

- Eficacia: La capacidad del usuario para completar las tareas del sistema y la calidad del resultado de esas tareas.
- Eficiencia: El nivel de recursos consumidos en la realización de tareas.
- Satisfacción: Relaciones subjetivas de los usuarios al utilizar el sistema.

En esta encuesta se plantearán entre cinco y diez preguntas, dentro de las cuales se tomarán en cuenta las medidas mencionadas anteriormente. Esta encuesta se realizará luego de que el usuario utilice el software, permitiendo obtener datos, los cuales se podrán graficar y analizar.

6. Implementación

Para la implementación de esta memoria se debió diseñar y programar un software interactivo para dispositivos móviles, con el propósito de entregar con fácil accesibilidad los contenidos de la guía de buenas prácticas Cuidados Centrados en la Persona y Familia. Para lograr este objetivo se siguieron las siguientes tareas presentadas en un tablero kanban.

6.1. Incrementos

En esta sección se nombraran y se explicaran cada uno de los incrementos que fueron planeados durante el transcurso del desarrollo del software interactivo.

6.1.1. Primer Incremento

Para llevar a cabo el primer incremento, ver anexoA.22, el proceso se dividió en tres partes. En primer lugar, se centró en la creación del proyecto relacionado con el entorno de trabajo. En segundo lugar, se estableció la base de datos y su estructura para asegurar el correcto funcionamiento del software. Por último, se implementó el primer entorno del software, integrando la librería correspondiente.

Creación del Proyecto

Para la creación del proyecto se implementaron los plugins de Dart y Flutter en el programa Android Studio, ya que estos son necesarios para el entorno en el que se desarrollará el software.

- Base de datos: Para el almacenamiento de la información se implementó Supabase como una base de datos Backend, este permite su integración mediante la instalación de una dependencia en el archivo “pubspec.yaml” ver anexo A.23. Luego de esto para inicializar la base de datos es necesario añadir dos líneas un con la función `WidgetsFlutterBinding.ensureInitialized()` y `Supabase.initialize` con su url y llave de acceso como se muestra en el anexo A.24.
- Manejo de datos: Para el manejo de los datos se creó un archivo `Baseprovider.dart` ver anexo A.25, el cual contiene una función para crear al cliente que solicita los datos y otra `SupabaseQueryBuilder table(String name)` que permite hacer el llamado a la información de la tabla mediante el nombre.

Para cada tabla se implementa un archivo `Baseprovider.dart` específico, como se muestra en el anexo A.26. Cada uno de estos scripts maneja dos funciones específicas: `addData` y `delete`, para gestionar la información en la base de datos. No obstante, la tabla de Librería no realiza integración o eliminación de datos; sus funciones buscan información en la base de datos como se muestra en el anexo A.27.

Estructura de la Base de Datos

Para Implementar la estructura de los datos, se crearon 5 tablas las que mantendrán la información de los contenidos de la guía, Casos clínicos, conceptos y definiciones, enlace de los videos y los perfiles registrados en la base de datos.

Library	Cases	VideoUrl	Memorize	profiles
id (Int8)	id (Int8)	id (Int8)	id (Int8)	id (uuid) -- auth.users.id
type_rec (text)	Type (text)	Title (text)	concept (varchar)	email (text)
recommendation (text)	Scenario (text)	Url (varchar)	definition (varchar)	role (text)
evidencelevel (text)	Question (text)	Type (text)		
number (Float4)	A (text)	VideoDescription (text)		
Subcateg (text)	B (text)			
	C (text)			
	Correct (text)			
	Feedback (text)			
	Number (Float4)			

Figura 6.1: Tablas de Supabase

- Tabla de Casos: Esta tabla contiene 9 elementos en cada fila: ID, escenario, pregunta, respuesta A, respuesta B, respuesta C, respuesta correcta, retroalimentación y número de recomendación. Cada fila representa un caso clínico almacenado en la base de datos.
- Tabla de Librería: La tabla de librería se refiere a los contenidos de la guía de buenas prácticas sobre Cuidado Centrado en la Persona y Familia. Cada fila de esta tabla tiene 5 elementos: tipo de recomendación, recomendación, nivel de evidencia, número de recomendación y subcategoría. Cada fila representa una recomendación incluida en la guía.
- Tabla de Memoria: Esta tabla guarda los conceptos y definiciones utilizados en el minijuego de memoria, incluyendo las columnas de ID, Concepto y Definición.
- Tabla de VideoUrl: En esta tabla se almacenan los enlaces de los videos de la galería, con las columnas de ID, Url y Descripción.
- Tabla de Perfiles: Esta tabla contiene la información de registro de los usuarios, integrando las columnas de ID, Correo y Rol.

Estas tablas fueron creadas utilizando el editor de tablas que ofrece Supabase para integrar tablas en la base de datos.

Implementación de la Librería

En la implementación de la librería, ver anexo 6.2, se utilizó el widget Obx en el cuerpo para asegurar la presencia de los contenidos de la librería. Esto se complementó con un `ListView.separated`, utilizado para generar un widget que tiene como hijo un `ExpansionTile`. Dentro de este, hay un `ListTile` que proporciona información sobre la categoría, subcategoría, recomendación, número de recomendación y nivel de evidencia como se muestra en el anexo A.28. Este procedimiento se repite hasta completar el número de recomendaciones según el tamaño de la lista que contiene la información.



Figura 6.2: Pantalla de Librería



Figura 6.3: Pantalla de Galería

El controlador de esta sección gestiona tres funciones para manejar la información , ver anexo A.29, las cuales son:

- **onInit:** Inicializa las funciones y carga los contenidos antes de que aparezcan en la pantalla del usuario.
- **fetchAll:** Recibe información de la base de datos mediante la función `provider.getAll` y la asigna a `items`, que es una lista observable de `Map`, utilizando la función `items.assignAll`.

- `filterByType`: Función que recibe una cadena de texto `type`, utilizada en la función `provider.getByType` del script de la base de datos, para obtener elementos que coincidan con el tipo solicitado.

6.1.2. Segundo Incremento

Este aumento incluye tres componentes: la implementación de `GetX`, el diseño de la pantalla de inicio del software y, finalmente, la incorporación de minijuegos en la sección interactiva, como el juego de preguntas y el de memoria. Estas tareas se detallan en el tablero kanban como se muestra en el anexo A.30.

Implementación de `GetX`

Para implementar `GetX`, se integró mediante una línea de comando `$ flutter pub add get` en la terminal del proyecto, permitiendo importar el "package `get/get.dart`". Esta integración facilita la creación de scripts de clases que extienden de `GetView`. Lo que a su vez permite acceder a variables de tipo observables `Rx`, permitiendo acceder al control de estado de `package`. La estructura que maneja para su funcionamiento es:

- `Page`: Se encarga de presentar el diseño de los widgets visibles e interactivos para el usuario en la interfaz. No se ocupa de la lógica subyacente de las interacciones que ocurren en la pantalla.
- `Binding`: Cumple la función de gestionar la transición entre las diversas pantallas. Además, facilita la inyección del controlador como una dependencia, utilizando un método `Get.Lazyput()` que prepara el controlador antes de que la Página sea renderizada, ver anexo A.31. Este proceso asegura que los datos estén precargados, proporcionando un margen de tiempo necesario para evitar errores de información durante la visualización.
- `Controller`: Gestiona la lógica del comportamiento de `Page`, ver anexo A.32. Este contiene las funciones necesarias para operar las acciones inherentes al script de `Page`, tales como la inicialización de los datos de las variables y las funciones para manejar cada una de las pantallas del software.

Diseño de pantalla inicio de sesión

Para implementar la pantalla de inicio de sesión que se muestra en la figura 6.4, se incorporó el paquete `flutter form builder` junto con `form builder validators` en la sección de dependencias del archivo `pubspec.yaml`. Estos permiten acceder a las siguientes características utilizadas para la creación y validación de los datos del inicio de sesión:



Figura 6.4: Pantalla de inicio de sesión

- **FormBuilder:** Actúa como la plantilla del formulario. Tiene la característica `Key` que acepta una variable de tipo `formkey` declarada en el controlador como `GlobalKey(FormBuilderState)`. Esto permite gestionar el formulario de manera independiente.
- **FormBuilderTextField:** Representa la casilla de texto. Se implementan características como `name`, `decoration`, `keyboardType` y `validator`. `Name` sirve para identificar el `TextField`, mientras que `keyboardType` establece el tipo de contenido, en este caso, `emailAddress`, aceptando solo este tipo de datos. `Validator` se utiliza para verificar los parámetros utilizando funciones de `FormBuilderValidators`, ver anexo A.33.

- FormBuilderValidators: Facilita la validación y el condicionamiento del contenido ingresado en el campo de texto, empleando funciones como FormBuilderValidators.required, FormBuilderValidators.email y FormBuilderValidators.minLength.

En el script LoginController se implementa la lógica principal mediante dos funciones asíncronas: Future(void) login() y Future(void) register(). La primera función guarda el estado actual de la formKey del formulario y verifica que no sea nulo para validar los datos ingresados. Utiliza la función supabase auth.SignInWithPassword, la cual comprueba si el correo electrónico y la contraseña están registrados en la tabla de usuarios de la base de datos. Por otro lado, la segunda función posibilita la creación de un nuevo usuario en la base de datos mediante la función auth.SignUp.

Diseño e implementación de minijuegos

Para el desarrollo del minijuego de preguntas figura 6.5, se diseñó un componente de interfaz que abarca una barra de progreso, la pregunta, el contexto y las opciones para elegir. Todo se organiza dentro de un Obx que verifica que la información del controlador no esté vacía antes de dibujar el widget Column, que contiene el diseño de la pantalla del usuario. Para gestionar la selección de opciones, se implementó la función optionTale, que funciona como una plantilla. Esta función permite modificar la información mediante una lista almacenada en el controlador, tal como se muestra en el siguiente anexo A.34.

El controlador contiene las funciones de onInit, load, updateProgress, skip, next y finalizarJuego, las funciones que manejan la lógica central del minijuego son las siguientes:

- load: Realiza la asignación de los valores a las variables, obtiene la información de la base de datos con provider.getall y la almacena en una RxList de mapas denominada questions, además de reiniciar el estado actual del juego, ver anexo A.35.
- next: Función que valida la selección del usuario, verifica si la respuesta es correcta, almacena en una lista la respuesta del usuario y reinicia los valores para la pregunta siguiente. Si se trata de la última pregunta, llama a la función finalizarJuego, ver anexo A.36.

- finalizarJuego: Modifica el bool completed a verdadero para finalizar el juego, y con la función Get.offname() pasa la información a la pantalla de retroalimentación, ver anexo A.37.

La pantalla de retroalimentación como se muestra en la figura 6.6 se ha implementado con una lista de ExpansionTile que incluye: número de pregunta, pregunta, respuesta del usuario y respuesta correcta con su justificación. En el controlador de retroalimentación se añadieron tres funciones: readyCheck, que verifica las respuestas correctas al comparar la lista de preguntas con la lista de retroalimentación obtenida por referencia; contentNullCheck, que revisa las preguntas que fueron omitidas; y calculateAccurate, que calcula el número total de respuestas correctas.

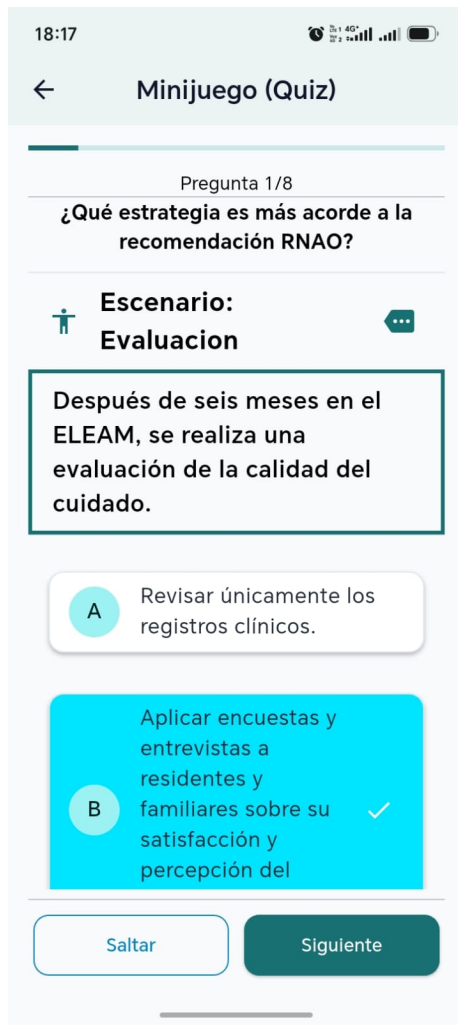


Figura 6.5: Pantalla de minijuego de preguntas

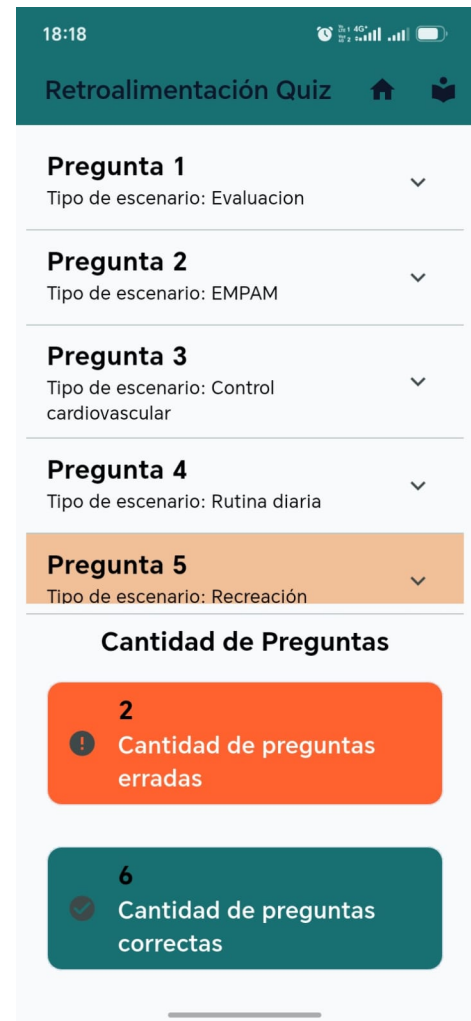


Figura 6.6: Pantalla de retroalimentación

En la implementación del minijuego de memoria, se han añadido un contador de tiempo, una barra de progreso y un `GridView.builder`. En el `gridDelegate`, se configuraron tres columnas con un espaciado de 8 píxeles en el eje X y 12 píxeles en el eje Y, estableciendo un `childAspectRatio` de 0.85 para el contenido de las tarjetas. Las cartas se añadieron mediante una clase personalizada llamada `CardModel`, ver anexo A.38, que gestiona atributos como el color, `isFlipped`, `isMatch`, contenido de la carta, par cartesiano, color del par y tipo de carta. Las variables `isFlipped` e `isMatch` están diseñadas para manejar el estado de la carta. Además de que el script en donde

se dibuja la interfaz maneja tres clases para complementar el GridView que son las siguientes:

- `cardFront`: Recibe por referencia una carta para dibujar la parte frontal utilizando los contenidos que se encuentran en la variable `contendCard`, y ajusta el color de la carta según el estado de `isMatch`.
- `cardBack`: Cumple la función de dibujar la parte posterior de la carta, maneja el color estándar de la carta y añade un ícono.
- `animationFlip`: Añade una animación de rotación mediante un `AnimatedSwitcher`, que agrega la duración y la transición. Esta se condiciona con la variable `isFlipped` de la carta, ver anexo A.42.

Las imágenes incluidas muestran una pantalla del minijuego de memoria figura 6.7 y otra con las cartas volteadas figura 6.8 , proporcionando un vistazo visual a la interfaz.

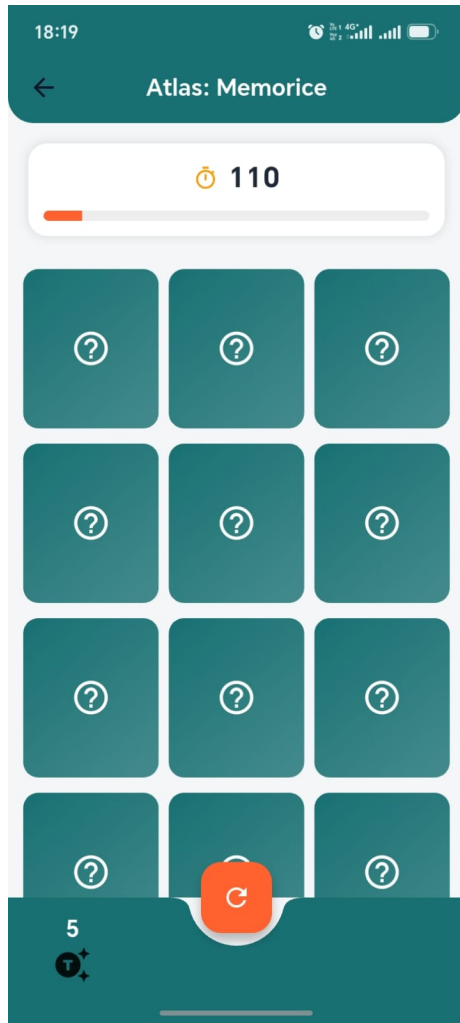


Figura 6.7: Pantalla de minijuego de memoria

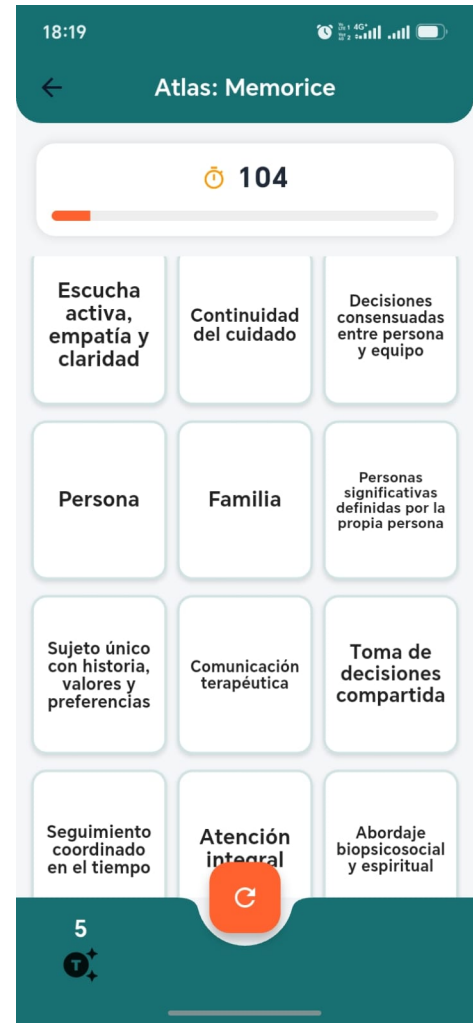


Figura 6.8: Pantalla de minijuego de memoria voltear cartas

El controlador de minijuego de memoria maneja una serie de funciones que ayudan al funcionamiento lógico, como pueden ser iniciar el juego, manejar el tiempo, seleccionar la carta, voltear la carta, comprobar si las cartas son par y finalizar el juego. Las funciones generales del controlador son las siguientes:

- `initializeGame`: La función se encarga de agregar y asignar los contenidos desde la base de datos, organizando las cartas dentro de una lista de objetos del tipo `CardModel`. Además, inicializa otras variables esenciales para el minijuego, como la cantidad de pares encontrados. También restablece variables como

`selectedIndex`, llama a la función que inicia el temporizador y voltea las cartas para dar comienzo al juego.

- `selectedCard`: Retiene el índice de la carta elegida, siempre y cuando esta no forme parte de un par o esté ya volteada. En tal caso, se cambia el estado de `isFlip` a verdadero y se añade el índice al conjunto `selectedIndex`. Al seleccionar una nueva carta, se verifica si los contenidos coinciden con la función `checkMatch`, en cuyo caso el arreglo se vacía. Este proceso realiza una comprobación de si se han ubicado todos los pares de cartas, ver anexo A.39.
- `checkMatch`: Esta función compara los contenidos de `pairCard` de las cartas ubicadas en el arreglo `selectedIndex`. Si coinciden, modifica el valor de `isMatch` y se transforman en pares. De lo contrario, restablece el valor de `isFlipped` a su estado inicial, ver anexo A.40.
- `flipAllCards`: Función tiene como objetivo invertir el estado de todas las cartas utilizando un bucle `For`, alterando el valor de la variable `isFlipped` por un periodo de cinco segundos mediante la función `future.delayed(duration)`. Posteriormente, se implementa otro ciclo para verificar que las cartas que ya están emparejadas mantengan sus valores de `isFlipped` inalterados al volver a estar boca abajo, ver anexo A.41.

Otras funciones que se han implementado son `resetGame` y `showGameDialog`. La segunda función muestra la cantidad de pares obtenidos y permite cerrar la ventana emergente para observar la retroalimentación en el tablero con las cartas, que se representan por los colores de la variable `pairColor`. Las cartas emparejadas comparten el mismo color.

6.1.3. Tercer Incremento

El tercer incremento abarca la implementación del diseño de las funciones del administrador, lo cual incluye el ingreso de datos y videos. Además, se realizarán correcciones de errores en el software, como el ajuste del tamaño de las letras, la corrección de los colores de diseño y la resolución de problemas relacionados con `renderFlex`. Estas tareas se detallan en un tablero Kaban, ver anexo A.43.

Diseño e implementación de funciones del administrador

Para la Implementación de las funciones de administrador, se asignaron rol de usuario y administrador a las cuentas, para esto se implementaron dos funciones Future (AuthService) init(), que almacena variables de manera local dentro de las cuales se encuentra el valor de la sesión actual del usuario, la cual ocupa para buscar el rol del usuario con la función fetchUserRole(string uid) como se muestra en la figura 6.9.

```

20 Future<AuthService> init() async {
21   if (AppConfig.supabaseReady) {
22
23     AppConfig.supabase.auth.onAuthStateChange.listen((data) async {
24       final event = data.event;
25       final session = data.session;
26
27       _session.value = session;
28       _user.value = session?.user;
29
30       if (session != null) {
31         await _fetchUserRole(session.user.id);
32       }
33
34       if (event == AuthChangeEvent.signedIn || event == AuthChangeEvent.userUpdated) {
35         if (Get.currentRoute == AppRoutes.login) {
36           Get.offAllNamed(AppRoutes.home);
37         }
38       }
39       if (event == AuthChangeEvent.signedOut) {
40         _role.value = 'user';
41         Get.offAllNamed(AppRoutes.login);
42       }
43     });
44
45     final currentSession = AppConfig.supabase.auth.currentSession;
46     _session.value = currentSession;
47     _user.value = currentSession?.user;
48
49     if (_user.value != null) await _fetchUserRole(_user.value!.id);
50   } else {
51     _session.value = null;
52     _user.value = null;
53   }
54   return this;
55 }

```

Figura 6.9: Código Login y asignación de rol de usuario

Para Implementar el añadido y eliminación de videos en la galería se agregaron dos funciones al controlador de galería las cuales son:

- insertTile: Esta función recibe cinco parámetros por referencia: nombre, URL, tipo y descripción. Utiliza estos parámetros para invocar a la función provider.insertVideo, añadiendo así el video a la tabla correspondiente en la base de datos.
- deleted: Función que recibe un title por referencia, que se utiliza para llamar la función provider.videoDeleted para remover la fila del title de la base de datos.

En lo que respecta a la interfaz de ingreso de datos, se desarrollaron cuatro funciones destinadas a añadir y eliminar información de la base de datos. Las funciones correspondientes a la adición de datos incluyen una validación para prevenir duplicados, como se observa en la figura A.44. Las funciones específicas son las siguientes:

- `insertConcept`: Procesa los datos introducidos mediante el teclado, almacenándolos en variables `textInputcontroller` como se ve en el anexo A.46, para posteriormente llamar a la función `providerMem.addData` e ingresarlos en la base de datos.
- `insertDataCase`: Recoge los datos de casos clínicos, almacenando cada uno en una variable por referencia, y llama a la función `providerCas.addData` para incorporarlos a la base de datos.
- `deletedConcept`: Recibe un `String` llamado `name` que utiliza en combinación con la función `providerMem.deletedData`, con el fin de encontrar y eliminar la fila correspondiente en la base de datos.
- `deletedCase`: Emplea un valor `Int` que representa el ID del caso clínico, utilizando esta información para ejecutar la función `providerCas.deletedData` y remover la fila correspondiente de la base de datos.

Corrección de errores menores

En este segmento, se abordaron y corrigieron diversos errores menores relacionados con la interfaz de usuario, tales como el ajuste del tamaño de los textos y la mejora en la paleta de colores, asegurando que estos sean coherentes con el diseño de RNAO, entre otros aspectos. Además, se implementó un script para verificar la conexión a internet del dispositivo. Para ello, se integró el paquete `Internet Connection Checker Plus` ver anexo A.47, que permite emplear una variable denominada `internetStatus`, la cual indica el estado de la conexión. Este sistema utiliza un `streamSubscription` para monitorear en tiempo real los cambios en dicho estado, como se ilustra en la Figura correspondiente en el anexo A.48. La función `handleStatusChange` adapta el estado de la pantalla en función del valor de la conexión, iniciando un temporizador. Si esta condición persiste durante treinta segundos, se habilita un botón para cerrar la aplicación.

7. Resultados

esta sección se describirá cómo se llevaron a cabo las pruebas con los usuarios, el promedio de los resultados de las encuestas, y para un análisis detallado de cada pregunta y gráfico, estos están disponibles en los anexos A.4. Para validar las respuestas, se utilizó el cálculo de puntajes proporcionado por las encuestas UEQ y SUS.

7.1. Realización de pruebas de usuarios

Se hicieron pruebas con los usuarios de manera presencial para estudiantes y profesores. A cada uno de ellos se les mostró una serie de diapositivas que explicaban las actividades a seguir durante la prueba de usuario. Estas diapositivas incluían la información de acceso al software y a las encuestas que se evalúan, mediante el uso de un código QR. A los participantes se les dio una hora para completar ambas actividades en conjunto. La primera encuesta que se les pidió que respondieran fue la SUS. En esta encuesta, los participantes dieron su consentimiento informado. Esto significaba que entendían que los datos que se recopilaran se utilizarían para validar el software y que sus respuestas serían anónimas. Las pruebas para estudiantes y profesores se llevaron a cabo en días diferentes. Once estudiantes participaron en la prueba. En cuanto a los profesores, tres de ellos participaron durante las pruebas.

7.2. Gráfico UEQ

En el gráfico del User Experience Questionnaire (UEQ) adaptado, referenciado en la figura 7.1, se evidencian valores favorables en las distintas dimensiones de la usabilidad del producto, en donde destaca el puntaje más alto de la relevancia de

los contenidos, en segundo lugar la confianza de la información y la utilidad de las funciones, lo que confirma la factibilidad de la aplicación, además de mostrar que aún tiene margen de mejora en otras dimensiones como se observa en la imagen.

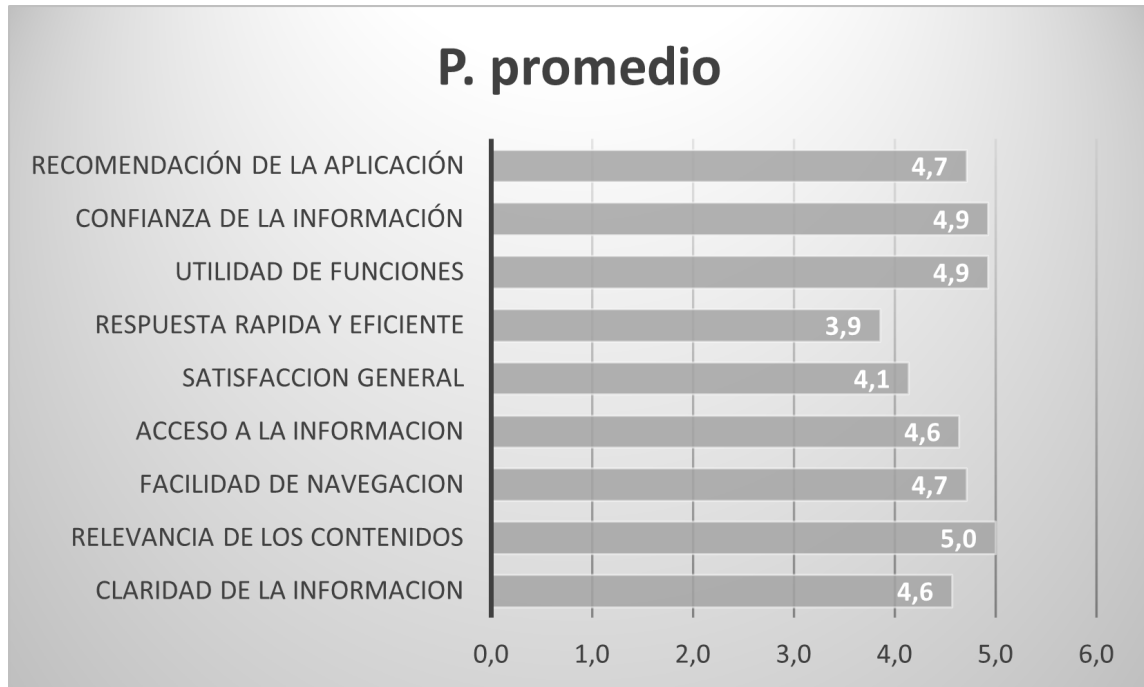


Figura 7.1: Promedios de puntajes UEQ

7.3. Gráfico SUS

En el gráfico del Sistema de Usabilidad de Software (SUS) referenciado en la figura 7.2, se evidencia una significativa variación en las puntuaciones de los participantes evaluados, específicamente en los casos de los encuestados 6 y 13, quienes registraron puntajes de 65 y 52.5, respectivamente. No obstante, el producto alcanzó una calificación media de 83.21 puntos, lo cual lo sitúa en un rango altamente positivo conforme a los estándares del SUS. Cabe resaltar que a pesar de los casos aislados mencionados se observa una gran mayoría sobre los 80 puntos, manifestando una tendencia favorable y consistente para el producto. De particular interés es el caso del encuestado número 8, quien otorgó la máxima puntuación posible, subrayando la eficacia del flujo de trabajo principal del producto.

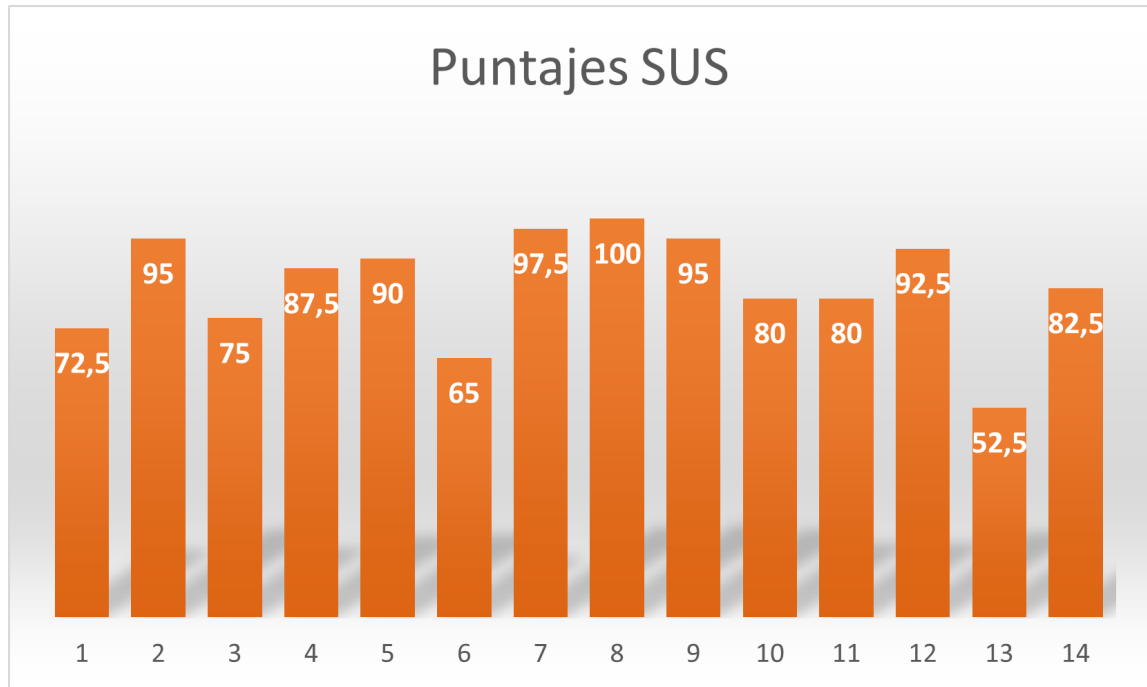


Figura 7.2: Puntaje escala SUS

7.4. Conclusión de análisis

La evaluación de los dos gráficos demuestra que el producto posee una usabilidad robusta y una aceptación favorable por parte de los encuestados. Las altas puntuaciones sugieren que la interfaz, navegación y funcionalidades están bien alineadas con las expectativas y necesidades de los usuarios, sin embargo también se logra percibir un margen de mejora asociados a las puntuaciones mas bajas, con las cuales trabajar para poder entregar una experiencia acorde a lo que el usuario espera.

8. Conclusiones y trabajo a futuro

El presente proyecto tuvo como finalidad el desarrollo de una aplicación móvil para facilitar la consulta de los contenidos de la guía de buenas prácticas Cuidado Centrado en la Persona y Familia, logrando cada uno de los objetivos específicos (OE) presentados en el documento.

- OE 1: Se realizó una toma de requisitos funcionales y no funcionales, a través de reuniones con el cliente y estos fueron redactados en el documento de diseño del software presentado en el Anexo A.7.3.
- OE 2: Para el diseño de los contenidos, se adquirió la información de los contenidos mediante la guía de buenas prácticas Cuidado Centrado en la Persona y Familia, ver anexo A.1.1. Junto con esto el cliente presentó una lista de casos clínicos para complementar la información, esta información se evidencia en el apartado de Implementación con las figuras 6.5 y 6.7.
- OE 3: Para la implementación del software se utilizó un modelo incremental combinado con un tablero Kanban para la asignación de hitos, esto se evidencia en el anexo A.22 A.30 A.43. Además de un video que muestra como se interactúa cada uno de los roles con la aplicación, ver Anexo A.4.1.
- OE 4: La evaluación del software se realizó mediante la aplicación de encuestas de experiencia de usuario y usabilidad. Se realizaron de manera presencial y se realizó un breve introducción a la actividad, esto se evidencia en el apartado 7.1 del documento junto con el consentimiento informado, ver Anexo A.6.1.

- OE 5: Se realizó un análisis y evaluación de los resultados de las encuestas del software, en donde se hizo un análisis exhaustivo en cada una de las preguntas de los cuestionarios como se observa en el anexo A.4. Además esto se evidencia en las figuras 7.1 y 7.2 los que representan promedio de los datos obtenidos por los usuarios.

En respuesta a la pregunta de investigación planteada al inicio del documento, a partir de los resultados obtenidos en las encuestas (OE 5) junto con la carta de aceptación presentada por el cliente en el anexo A.7.3, se puede concluir que el desarrollo de una aplicación interactiva en gran medida es aceptable para la consulta de contenidos. Específicamente, manifestándose principalmente en la usabilidad, aceptación y experiencia de usuario.

En relación con el trabajo a futuro, con base en los resultados de las encuestas realizadas a los usuarios presentadas en la figura 7.1, se identificó la necesidad de resolver estos puntos de mejora de manera prioritaria. En primer lugar, enfocada en la experiencia del usuario, es fundamental mejorar la eficiencia del sistema del software, agilizando la transición entre pantallas y respuestas de las interacciones del usuario con el sistema. En segundo lugar, es necesario expandir su portabilidad para que sea compatible con el sistema operativo iOS, debido a que durante las pruebas de usuarios gran parte de los profesores hacían uso de estos tipos de dispositivos móviles.

Por otro lado, en respecto a las características adicionales que se pueden agregar a futuro para complementar el software a partir de comentarios de los usuarios u observaciones, se pueden implementar de menor a mayor dificultad. En primer lugar, se priorizará la reestructuración de la asignación de roles de la base de datos, añadiendo la característica de asignación de rol de asistente de administrador a estudiantes, para la gestión interna de los contenidos. En segundo lugar, es pertinente la integración de un módulo colaborativo con profesores, para la realización de consultas y puntos de discusiones de los contenidos. Por último, se plantea la ampliación de los contenidos del software, permitiendo la integración de múltiples guías de buenas prácticas junto con la creación de un selector para que el usuario elija el contenido a utilizar.

Glosario

GBP-RNAO Guías de Buenas Prácticas de la Asociación de Enfermeras Registradas de Ontario.

RNAO Registered Nurses' Association of Ontario, en español Asociación de Enfermeras Registradas de Ontario.

BPSO Best Practice Spotlight Organizations, en español Centros Comprometidos con la Excelencia en Cuidados.

TIC Tecnologías de Información y Comunicaciones

DCU Diseño Centrado en el Usuario

UEQ User Experience Questionnaire

SUS System Usability Scale

Bibliografía

- [1] J. Chaparro D. Grispun. A. Belmar, y M. Guell. Implementación de buenas prácticas en enfermería: Programa BPSO como herramienta principal. *Revista Médica Clínica Las Condes*, 29(3):311–321, 2018.
- [2] N. L. Suazo-Díaz W. Bustamante-Muñoz C. Amigo-Rojas I. Pino-Monsalve G. A. Domínguez-Araya, y E. Morales-Herrera. Implementación de tres guías de buenas prácticas clínicas en la escuela de enfermería de la universidad de talca, chile. *Ciencia y enfermería*, 29, 2023.
- [3] Registered Nurses’ Association of Ontario. Cuidado centrado en la persona y la familia, 2015.
- [4] Registered Nurses’ Association of Ontario. Recursos en español, 2023. Accedido: 2025-05-26.
- [5] Hospital de Linares. RNAO - hospital de linares, 2022. Accedido: 2025-06-07.
- [6] BPSO. Programa BPSO internacional, 2025. Accedido: 2025-06-07.
- [7] BPSO. Otros recursos, 2025. Accedido: 2025-06-07.
- [8] Nurse Menorca. App guía terapia intravenosa. Video en YouTube, 2025. Accedido: 2025-06-07.
- [9] M. Isabel y A. Morales. Nuevas tecnologías y nuevos retos para el profesional de enfermería. *Index de Enfermería*, 25:38–41, June 2016.

- [10] J. Roberto y G. Núñez. Aplicaciones de dispositivos móviles como estrategia de aprendizaje en estudiantes universitarios de enfermería. una mirada desde la fenomenología crítica. *RIDE Revista Iberoamericana para la Investigación y el Desarrollo Educativo*, 10(20), 2020.
- [11] R. Thamizharasi. Android mobile application build on android studio. *International Journal of Modern Computer Science (IJMCS)*, 4(1):1–4, 2016.
- [12] J. Piñeiro Gomez. *Entornos de desarrollo*. Ediciones Paraninfo, SA, 2022.
- [13] Google. Herramientas de desarrollo de Android, 2023. Accedido: 12 de agosto de 2025.
- [14] Google. Emulador de Android, 2023. Accedido: 12 de agosto de 2025.
- [15] Android Developers. Emulador de android, 2026. Accedido: 18 de junio de 2026.
- [16] Android Developers. Introducción a android studio, 2026. Accedido: 18 de junio de 2026.
- [17] R. Gala T. Giri y P. Chavhan A. Tashildar, N. Shah. Application development using flutter. *International Research Journal of Modernization in Engineering Technology and Science*, 2(8):1262–1266, 2020.
- [18] Dagne y Lukas. Flutter for cross-platform app and sdk development. 2019.
- [19] Dart Team. Dart overview, 2026. Accedido: 18 de junio de 2026.
- [20] Flutter Team. Learn flutter, 2026. Accedido: 18 de junio de 2026.
- [21] P. Julián y M. E. Gabriel. Metodologías de desarrollo de software. Technical report, Universidad Tecnológica Nacional, 2015.
- [22] J. L. Acosta Espinoza R. A. Díaz Vásquez L. Yacelga, y A. Roberto. Aplicación de la metodología incremental en el desarrollo de sistemas de información. *Revista Universidad y Sociedad*, 13(5):175–182, 2021.
- [23] Guru99. ¿Qué es el modelo incremental en SDLC? ventajas y desventajas, 2024. Accedido: 27 de junio de 2025.

- [24] M. Bermejo. El Kanban. *Barcelona, España: UOC*, 8, 2011.
- [25] Patricio y L. Torres. Una actividad para enseñar el uso de tableros kanban y diagramas de flujo acumulado. In *Actas de las XXI Jornadas sobre la Enseñanza Universitaria de la Informática*, pages 288–295. Universitat Oberta La Salle, 2015.
- [26] Marlon G. Moya C. Anzules, y E. Javier. *Kanban: Una metodología ágil para la gestión eficiente del flujo de trabajo en el desarrollo de software, una revisión sistemática*. Iyayku Innova Editores, 2024.
- [27] BPSO. Guías de buenas prácticas, 2025. Accedido: 2025-06-12.
- [28] Keren y Ramírez-Acosta. Interfaz y experiencia de usuario: parámetros importantes para un diseño efectivo. *Revista tecnología en marcha*, 30:49–54, 2017.
- [29] E. M. Pera y M. G. Domingo. Diseño centrado en el usuario. *Universitat Oberta de Catalunya*, pages 9–12, 2010.
- [30] Y. H. Montero. *Experiencia de usuario: principios y métodos*. El autor, 2015.
- [31] F. J. M. Fernández y Y. H. Montero. La experiencia del usuario. *No solo usabilidad*, (4), 2005.
- [32] A. Bahattab y A. Alshamrani. A comparison between three SDLC models waterfall model, spiral model, and incremental/iterative model. *International Journal of Computer Science Issues (IJCSI)*, 12(1):106, 2015.
- [33] L. Miguel y M. Los diagramas de flujo: su definición, objetivo, ventajas, elaboración, fases, reglas y ejemplos de aplicaciones. *Blog de Gestión de Operaciones*, 22:09–18, 2011.
- [34] M. Arias y Chaves. La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. *InterSedes: Revista de las Sedes Regionales*, 6(10):1–13, 2005.
- [35] Brooke y John. SUS: Una escala de usabilidad rápida y sencilla. *Software Quality Management*, November 1995. Consultado el 31 de julio de 2020.

A. Anexos

En este apartado se presentarán todos los documentos e imágenes que se utilizarán como material de apoyo para el desarrollo de este proyecto.

A.1. Guías de buenas prácticas

- Abordaje proactivo para el manejo vesical e intestinal en adultos
- Acceso vascular.
- Administración de insulina subcutánea en adultos con diabetes tipo 2.
- Apoyo a adultos que esperan o viven con una ostomía.
- Atención y apoyo a las familias.
- Cuidados centrados en la persona y familia.
- Cuidados y mantenimiento de los accesos vasculares para reducir las complicaciones.
- Cuidado de enfermería para la disnea: la sexta constante vital en personas con EPOC.
- Cuidados del asma en adultos: Promover el control de asma.
- Delirio, demencia y depresión en las personas mayores: valoración y cuidados.
- End of life care during the last days and hours.
- Enfoque paliativo de los cuidados en los últimos 12 meses de vida.

- Engaging clients who use substances.
- Establecimiento de la relación terapéutica.
- Facilitating client centred learning.
- Fomento de la seguridad: alternativas al uso de contenciones.
- Fomento de un desarrollo saludable de los adolescentes.
- Implementing Supervised Injection Services.
- Integrar las intervenciones relacionadas con el consumo de tabaco en la práctica diaria.
- Intervención en crisis.
- La disminución de las complicaciones del pie en las personas diabéticas.
- Lactancia materna.
- Maltrato contra la mujer.
- Prevención primaria de la obesidad infantil.
- Prevención de las caídas y disminución de lesiones derivadas de las caídas.
- Preventing and addressing abuse and neglect of older adults: person-centred, collaborative, system-wide approaches.
- Promoción del control del asma infantil.
- Promoting 2SLGBTQI+ health equity.
- Salud oral: apoyo a los adultos que requieren asistencia.
- Strategies to support self-management in chronic conditions: collaboration with clients.
- Transiciones asistenciales.
- Úlceras del pie diabético: prevención, valoración y manejo.

- Valoración de ictus mediante la atención continuada.
- Valoración del riesgo y prevención de las úlceras por presión.
- Valoración e intervenciones para la depresión perinatal.
- Valoración y cuidado de los adultos en riesgo de ideación y comportamiento suicida.
- Valoración y manejo de las lesiones por presión para equipos interprofesionales.
- Valoración y manejo de las úlceras venosas en las piernas.
- Valoración y manejo de úlceras del pie diabético.
- Valoración y manejo del dolor.
- Working with families to promote safe sleep for infants 0-12 months of Age.

A.1.1. Recomendaciones y Evidencia de la guía cuidados centrado en la persona y familia

1. Valoración

- Recomendación 1.1: Establecer una relación terapéutica con la persona utilizando estrategias de comunicación no verbal para construir una relación de confianza y respeto genuina.
- Nivel de evidencia:III
- Recomendación 1.2: Construir relaciones de empoderamiento con la persona para promover su participación activa y significativa como una colaboradora activa en sus cuidados de salud.
- Nivel de evidencia:Ia
- Recomendación 1.3: Escuchar y buscar conocimiento integral de la persona para comprender el significado de la salud para la persona y conocer sus preferencias de cuidado.
- Nivel de evidencia: Ia

- Recomendación 1.4: Documentar la información obtenida del significado y de la experiencia de salud de la persona utilizando sus propias palabras.
- Nivel de evidencia: V

2. Planificación

- Recomendación 2.1: Desarrollar un plan de cuidados en colaboración con la persona que tenga sentido para la persona en el contexto de su vida.
- Nivel de evidencia: Ia
- Recomendación 2.2: Comprometerse con la persona en la creación de un modelo participativo de toma de decisiones, respetando su derecho a elegir las intervenciones que prefiera para su salud, mediante:
 - a) La colaboración con la persona en la identificación de sus prioridades y objetivos en relación con los cuidados de la salud.
 - b) El intercambio de información para promover la comprensión de las opciones disponibles para los cuidados de salud, de modo que la persona pueda tomar una decisión informada.
 - c) El respeto a la persona como experta en sí misma y en su vida.
- Nivel de evidencia: Ia

3. Implantación

- Recomendación 3.1: Personalizar los cuidados y servicios para garantizar que los cuidados no se lleven a cabo desde la perspectiva del profesional de la salud y la institución, colaborando con la persona en:
 - a) Los componentes de los cuidados.
 - b) Funciones y responsabilidades en los cuidados.
 - c) Estrategias de comunicación.
- Nivel de evidencia: Ia
- Recomendación 3.2: Colaborar con la persona para adaptar las estrategias para el autocuidado en base a las características de la persona y sus preferencias para el aprendizaje.

- Nivel de evidencia: V

4. Evaluación

- Recomendación 4.1: Obtener retroalimentación de la persona para determinar su satisfacción con el cuidado y si los cuidados presentados están centrados en la persona y la familia.
- Nivel de evidencia: Ia

5. Formación

- Recomendación 5.1: Formar a los profesionales de la salud, como mínimo. En las siguientes características de los cuidados centrados en la persona y la familia para mejorar los resultados clínicos y la satisfacción de la persona respecto a los cuidados:
 - a) Empoderamiento.
 - b) Comunicación.
 - c) Toma de decisiones compartidas.
- Nivel de evidencia: Ia
- Recomendación 5.2: Las instituciones docentes incorporarán esta guía en los planes de estudios de las enfermeras y, en su caso, en los planes de estudios de otros profesionales de la salud.
- Nivel de evidencia: V

6. Sistema, organización y directrices

- Recomendación 6.1: Crear una cultura institucional que ponga de manifiesto su compromiso con los cuidados centrados en la persona y la familia a través de:
 - a) La demostración de liderazgo y compromiso con este enfoque de cuidados.
 - b) La implicación de la persona en el diseño de los programas y servicios de salud.

- c) La construcción de entornos laborales saludables para todos los profesionales de la salud.
- Nivel de evidencia: V
- Recomendación 6.2: Diseñar un ambiente que mejore de forma demostrable la experiencia de la persona en relación con los cuidados de salud a través de:
 - a) La creación de entornos de curación.
 - b) La flexibilidad y la colaboración para personalizar las rutinas de cuidados.
 - c) La mejora en la accesibilidad a los cuidados y servicios.
 - d) La mejora de la continuidad y la coordinación de los cuidados y los servicios durante las transiciones.
 - e) La provisión de la continuidad de los cuidadores.
- Nivel de evidencia: V
- Recomendación 6.3: Obtener de la persona retroalimentación continua para determinar si su experiencia con los cuidados y servicios de salud se corresponde con cuidados centrados en la persona y la familia, y utilizar esta información para hacer mejoras en todos los niveles del sistema de salud.
- Nivel de evidencia: III
- Recomendación 6.4: Las agencias gubernamentales y organismos reguladores deben monitorizar, medir y utilizar la información de las instituciones en relación con la experiencia de la persona sobre los cuidados de la salud para mejorar el desempeño del sistema de salud.
- Nivel de evidencia: Ia

A.1.2. Interpretación de la evidencia

Los niveles de evidencia se asignan a cada recomendación con el fin de eliminar las explicaciones alternativas de los fenómenos estudiados. Dependiendo del nivel de evidencia, puede aumentar o disminuir la probabilidad de que las relaciones presentadas entre las variables sean verdaderas. Sin embargo, el nivel de evidencia no refleja el valor y la calidad de los estudios que se realizan de manera individual.

Niveles de evidencia

- Ia: Evidencia obtenida de metaanálisis o revisiones sistemáticas de ensayos controlados aleatorizados y/o síntesis de múltiples estudios primarios de investigación cuantitativa.
- Ib: Evidencia obtenida de al menos un ensayo controlado aleatorizado.
- IIa: Evidencia obtenida de al menos un estudio bien diseñado no aleatorizado.
- IIb: Evidencia obtenida de al menos un estudio de otro tipo cuasi-experimental no aleatorizado bien diseñado.
- III: Evidencia obtenida de la síntesis de múltiples estudios, principalmente de investigación cualitativa.
- IV: Evidencia obtenida de estudios observacionales no experimentales, bien diseñados, como estudios analíticos o descriptivos y/o estudios cualitativos.
- V: Evidencia obtenida de los informes elaborados por un comité de expertos o de la opinión o las experiencias clínicas de autoridades respetadas.

A.2. Comparativa entre aplicaciones

Tabla A.1: Comparativa entre Atlas y la app Cuidaven

Criterio	App Cuidaven	Atlas
Guía de buenas prácticas	Accesos Vasculares	Cuidados Centrados en la Persona y Familia.
Público dirigido	Profesionales de enfermería clínicos, estudiantes y ciudadanía.	Docentes y estudiantes de enfermería.
Presentación de contenidos	Estructura estática de consulta clínica.	Navegación dinámica mediante la librería de contenidos.
Material de apoyo	Textos, fichas informativas, glosario de Términos y recursos audiovisuales español e inglés.	Recursos audiovisuales proporcionados por docentes y material interactivo basado en minijuegos.
Interactividad y evaluación	No requiere creación de cuenta, requiere evaluación para adherirse como profesional, sin mecánicas de juego ni autoevaluación de contenidos.	Requiere la creación de una cuenta, permite la asignación de roles y ofrece minijuegos basados en casos clínicos y retroalimentación.

A.3. Atributos de usabilidad

Dimensión Objetiva

- **Facilidad de Aprendizaje:** ¿Cómo de fácil resulta para los usuarios llevar a cabo tareas básicas la primera vez que se enfrentan al diseño?
- **Eficiencia:** Una vez que los usuarios han aprendido el funcionamiento básico del diseño, ¿cuánto tardan en la realización de tareas?
- **Cualidad de ser recordado:** Cuando los usuarios vuelven a usar el diseño después de un periodo sin hacerlo, ¿cuánto tardan en volver a adquirir el conocimiento necesario para usarlo eficientemente?

- Eficacia: Durante la realización de una tarea, ¿cuántos errores comete el usuario?, ¿cómo de graves son las consecuencias de esos errores?, ¿cómo de rápido puede el usuario deshacer las consecuencias de sus propios errores?

Dimensión Subjetiva

- Satisfacción: ¿Como de agradable y sencillo le ha parecido al usuario la realización de las tareas?

A.4. Análisis de los resultados de la encuestas

A.4.1. Grabaciones de la aplicación

- Video del software perspectiva usuario
- Video del software perspectiva administrador

A.5. Carta Gantt

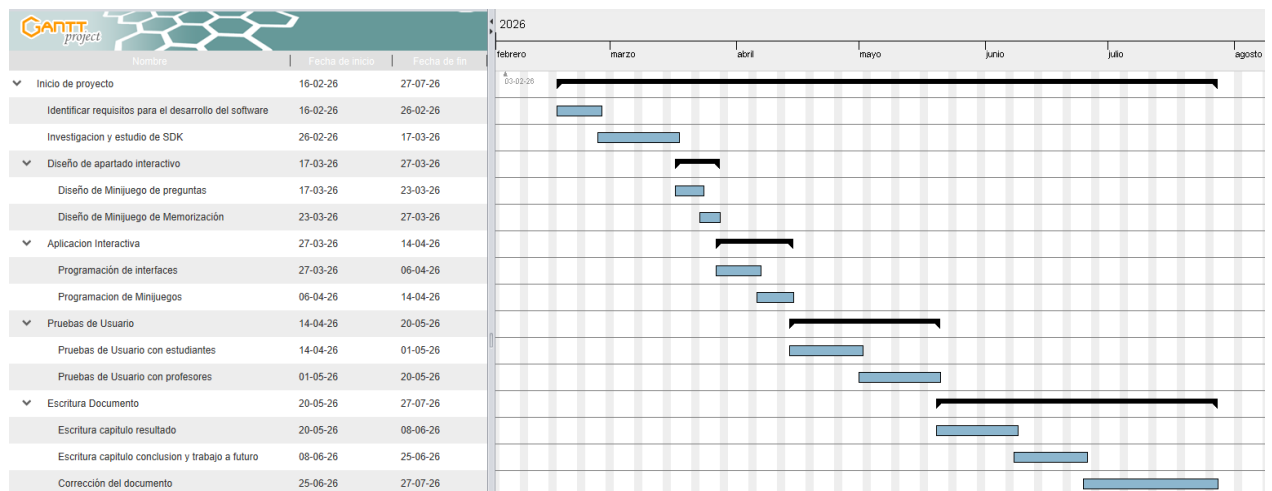


Figura A.1: Carta Gantt

A.6. Consentimiento informado y resultados

A.6.1. Consentimiento informado

Consentimiento informado

Este documento tiene como objetivo informarte sobre tu participación en el proyecto mencionado. Es fundamental que comprendas los objetivos y procedimientos asociados.

Objetivos del estudio

El propósito de este estudio es recopilar información para validar el funcionamiento de la aplicación **Atlas**.

Procedimiento

Si decides participar, se te solicitará que completes dos encuestas:

1. **Encuesta de experiencia de usuario:** Esta encuesta incluirá preguntas sobre tu percepción y sentimientos al interactuar con la aplicación.
2. **Encuesta de funcionalidad:** En esta encuesta se evaluará la funcionalidad de la aplicación.

La duración aproximada de tu participación será de 30 minutos.

Tu participación será completamente anónima. No se recopilará información que pueda identificarte, y todos los datos serán tratados de manera confidencial. La información recopilada se utilizará exclusivamente para el estudio, con el fin de crear gráficos y análisis de datos.

* Indica que la pregunta es obligatoria

¿Das consentimiento para tu participación? *

☐ Aceptar

☐ Rechazar

[Siguiente](#) [Borrar formulario](#)

Figura A.2: Consentimiento informado presentado a los usuarios.

A.6.2. Resultados de la Encuesta UEQ

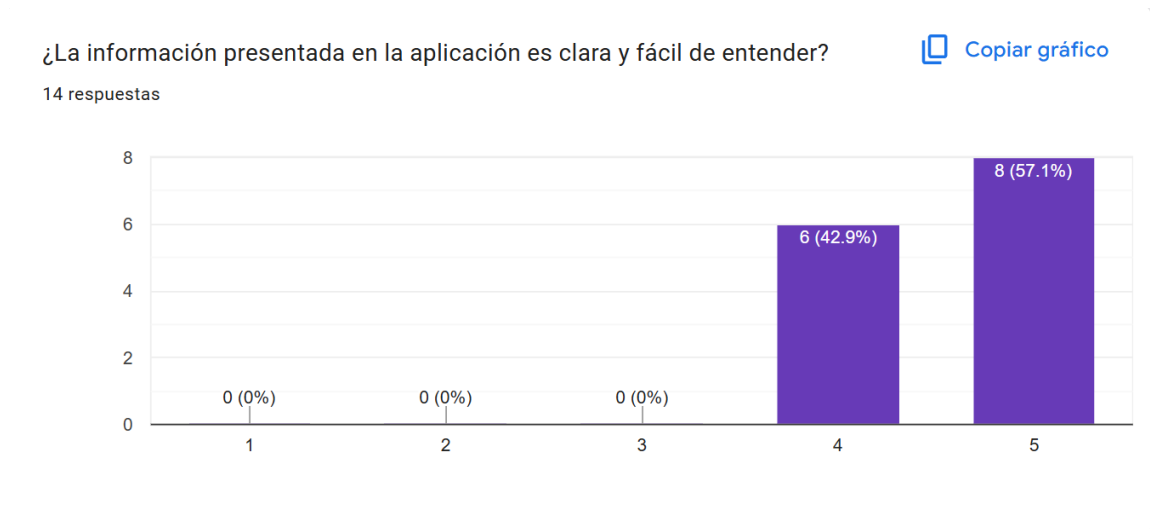


Figura A.3: UEQ Pregunta 1

En el primer gráfico figura A.3 de la encuesta de experiencia de usuario se puede observar que ocho de los usuarios están muy de acuerdo con que la información proporcionada es clara y fácil de entender. Asimismo, seis de los usuarios están de acuerdo con lo mismo pero en menor medida, lo que refleja una recepción positiva de la información en general. A partir de estos resultados, se concluye que la presentación de la información en la aplicación es eficaz y comprensible para el público objetivo.

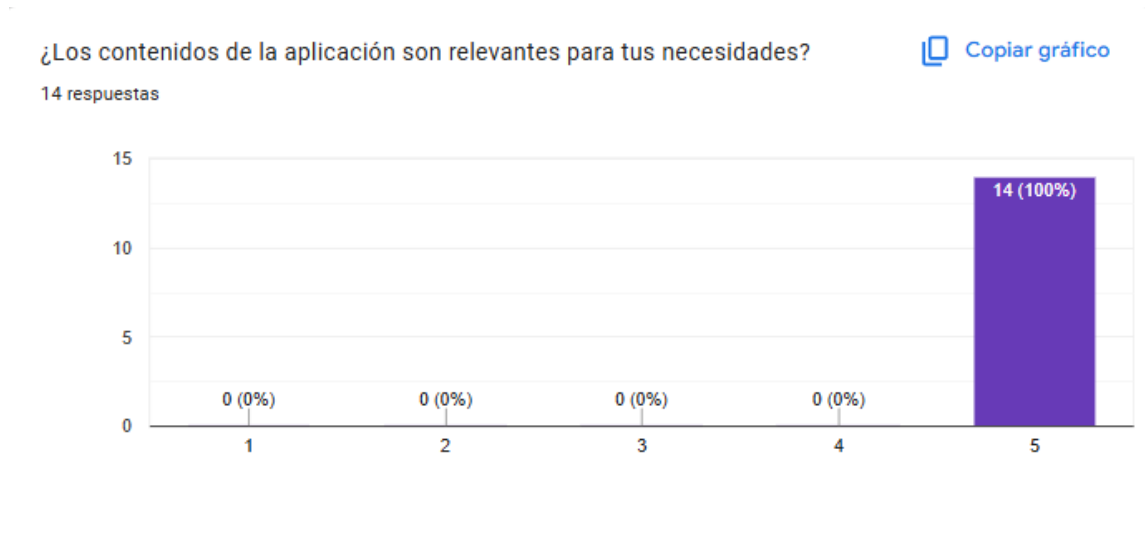


Figura A.4: UEQ Pregunta 2

En el segundo gráfico figura A.4 se observa un consenso total entre los usuarios, quienes coinciden que la aplicación entrega contenido importante y verídico para ayudarlos en sus necesidades de estudio. Con la obtención de estos resultados, se concluye que la información que presenta la aplicación logra cubrir satisfactoriamente las necesidades del público objetivo.

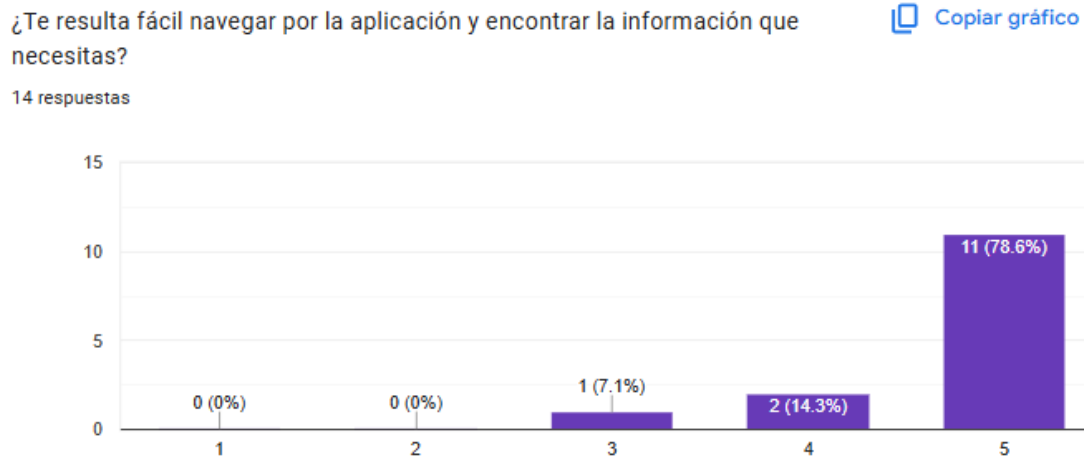


Figura A.5: UEQ Pregunta 3

En el tercer gráfico figura A.5 se puede observar una ligera disparidad en las respuestas, mientras que trece usuarios consideran que la aplicación tiene una navegación sencilla, uno de ellos presentó una leve dificultad en el uso de la aplicación. Debido a esta excepción, se concluye que, si bien la interfaz es funcional para la gran mayoría de usuarios, existe un margen de mejora en la usabilidad y acceso a la información necesaria para el usuario, para garantizar una experiencia fluida para todos los perfiles de usuario del público objetivo.

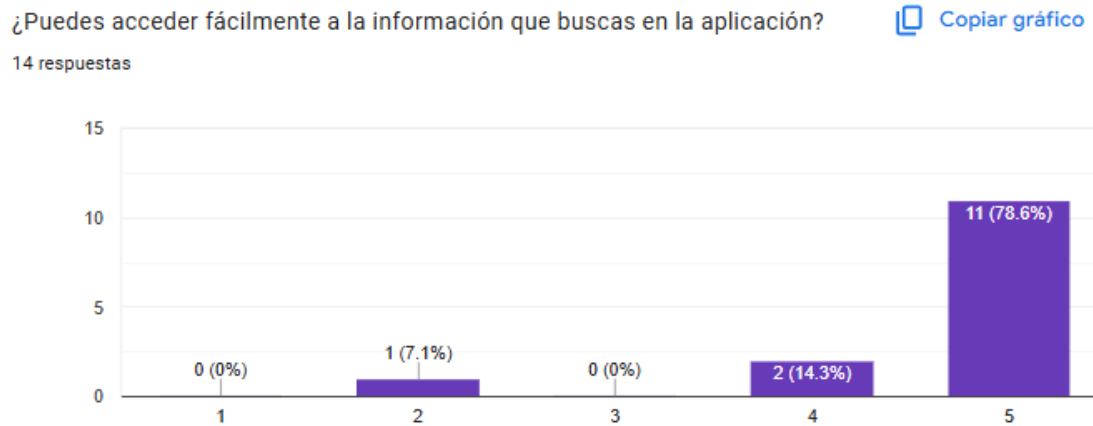


Figura A.6: UEQ Pregunta 4

En el cuarto gráfico figura A.6 se puede observar que trece usuarios tienen una opinión positiva, al estar de acuerdo o muy de acuerdo respecto a la facilidad de acceso a la información. No obstante, se observa que uno de los usuarios percibe este proceso como complejo. En consecuencia, aunque la eficacia del acceso no presenta ninguna dificultad para la mayoría, se identifica una oportunidad de mejora en la manera en que se presenta el contenido y en la disposición de los elementos de búsqueda de la aplicación.

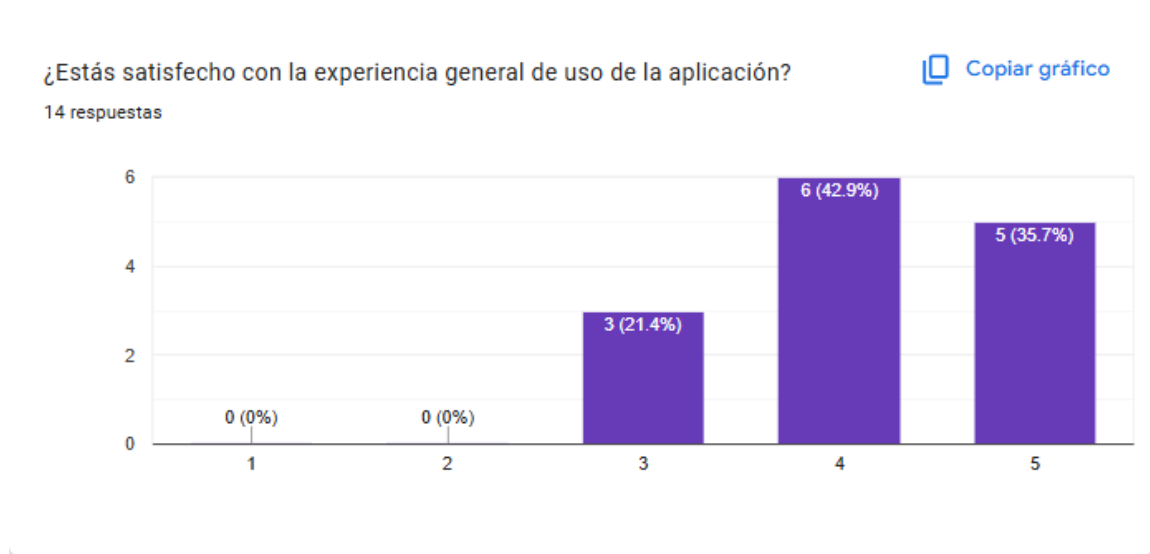


Figura A.7: UEQ Pregunta 5

En el quinto gráfico figura A.7 se presenta una disparidad en las valoraciones sobre la satisfacción general: mientras que once usuarios presentan una postura positiva, tres de ellos mantienen una opinión neutra. Estos resultados demuestran que, aunque la percepción es mayoritariamente favorable, existe un margen de mejora tanto en las funcionalidades como en el diseño para abordar satisfactoriamente todos los perfiles de usuarios.

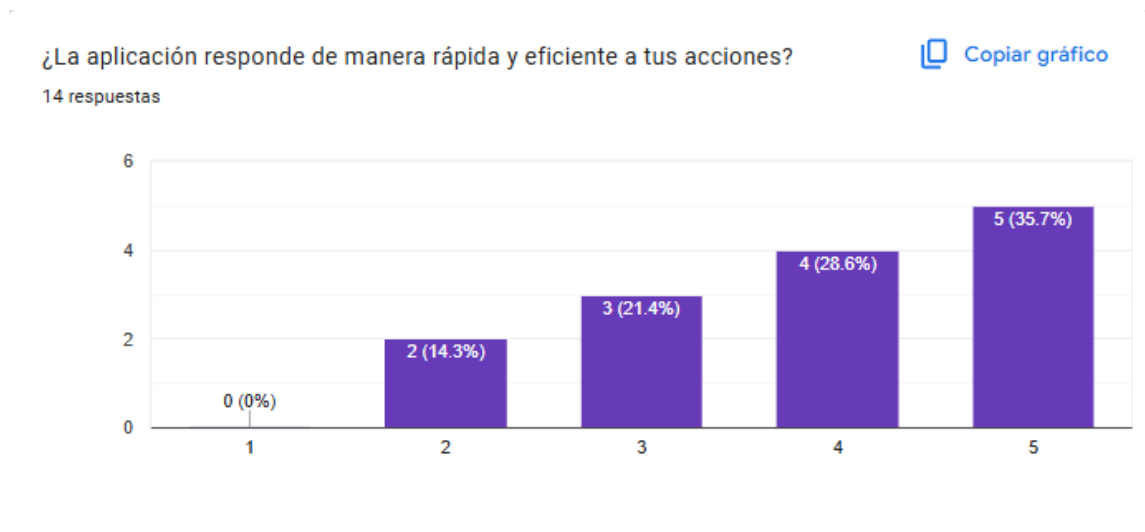


Figura A.8: UEQ Pregunta 6

En el sexto gráfico figura A.8 se observa una disparidad significativa respecto a la velocidad de respuesta de la aplicación a las acciones del usuario. Pese a que nueve de los usuarios valoran positivamente este aspecto, tres mantienen una postura neutral y dos manifiestan insatisfacción sobre este apartado. Estos resultados demuestran la necesidad de mejorar la capacidad de respuesta y fluidez de la interfaz, con el fin de aumentar la valoración positiva de los usuarios.

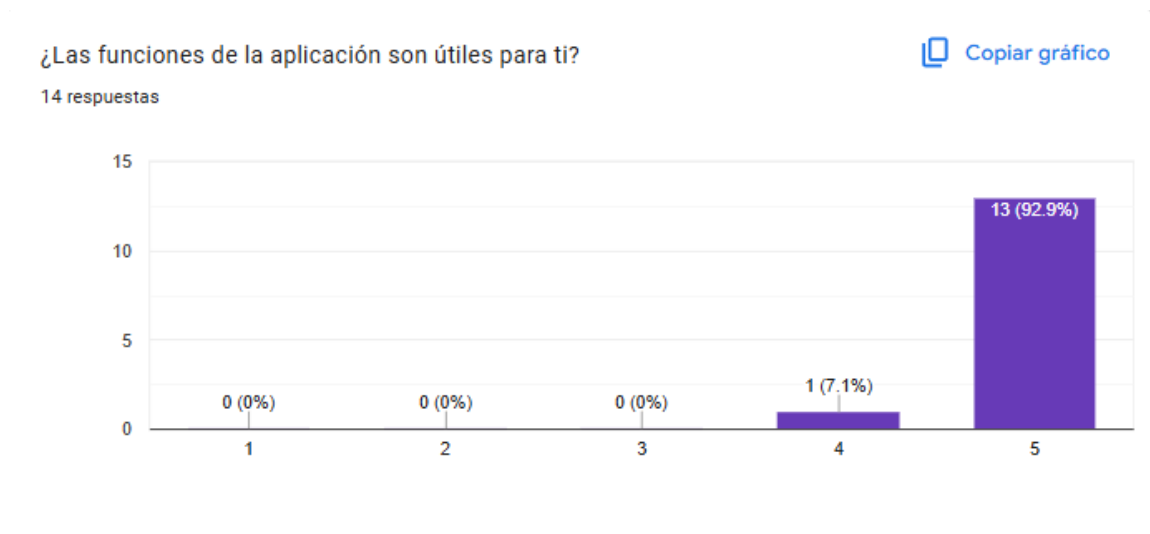


Figura A.9: UEQ Pregunta 7

El séptimo gráfico figura A.9 se observa una valoración mayormente positiva: trece de los encuestados están muy de acuerdo y uno manifiesta estar de acuerdo de que la aplicación presenta funciones útiles. El recibimiento favorable demuestra el propósito de las herramientas implementadas y confirma la utilidad de la aplicación para los usuarios.

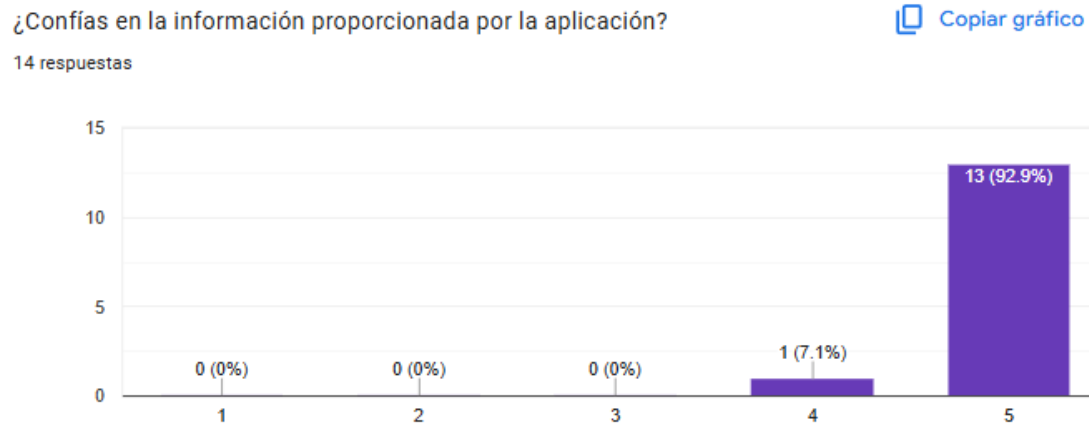


Figura A.10: UEQ Pregunta 8

En el octavo gráfico figura A.10 se observa que la gran mayoría de los usuarios confían en la información entregada por la aplicación: trece están muy de acuerdo y uno manifiesta estar de acuerdo respecto a la seguridad que le brinda los contenidos. Este resultado tan favorable demuestra la fiabilidad de la información que maneja la aplicación ante su público objetivo.

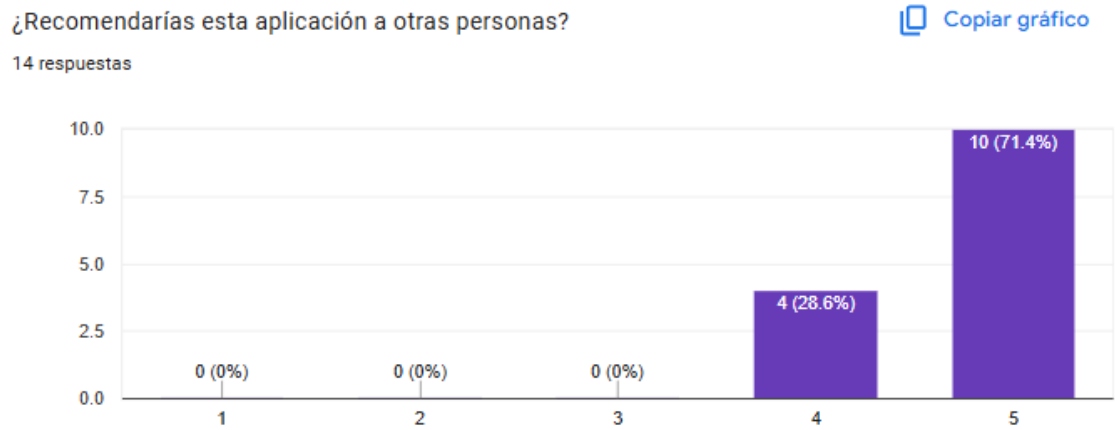


Figura A.11: UEQ Pregunta 9

En el ultimo gráfico figura A.11 se observa una excelente recepción de la aplicación por parte de los encuestados: diez participantes están muy de acuerdo en recomendarla y cuatro manifiestan estar de acuerdo con esta intención. Estos resultados permiten concluir que la aplicación se encuentra positivamente valorada para su uso, logrando un alto nivel de idealización entre el publico objetivo

A.6.3. Resultados de la Encuesta SUS

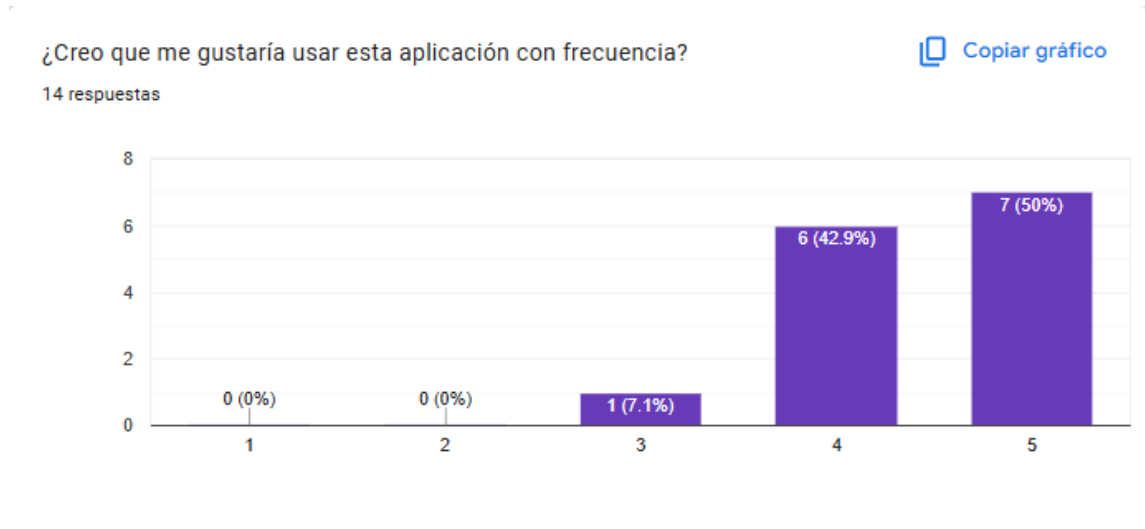


Figura A.12: SUS Pregunta 1

En el primer gráfico figura A.12 se puede observar una valoración positiva por parte de los encuestados respecto al interés de emplear la aplicación con frecuencia; aunque la mayoría muestra una disposición positiva, uno manifiesta una valoración neutral sobre su uso. Los resultados demuestran una aceptación elevada y una disposición favorable a la integración de la aplicación en la rutina de los usuarios.

¿Encontré la aplicación innecesariamente compleja?

 Copiar gráfico

14 respuestas

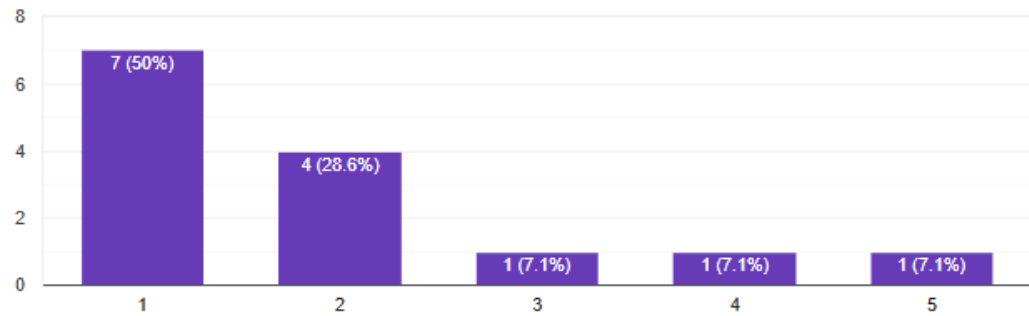


Figura A.13: SUS Pregunta 2

El segundo gráfico figura A.13 se observa una disparidad significativa en las valoraciones respecto a la complejidad de la aplicación. Mientras que once usuarios perciben que el sistema no es complejo, uno manifiesta una postura neutral y dos consideran que la aplicación es innecesariamente compleja. Estos resultados permiten concluir que es necesario mejorar en el aspecto de diseño de la interfaz y lógica de la aplicación, para hacerlo mas intuitivo para los usuarios.

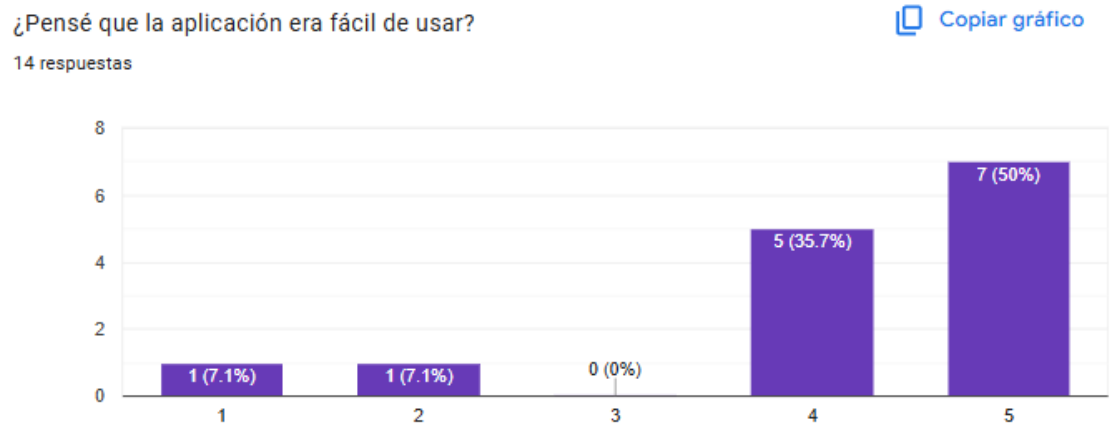


Figura A.14: SUS Pregunta 3

El tercer gráfico figura A.14 se observa una leve disparidad entre las valoraciones de los encuestados, respecto al pensamiento de la facilidad de uso percibida. Mientras en la gran mayoría de encuestados que concuerdan en que la aplicación es fácil de utilizar, dos de ellos manifiestan una opinión contraria. Estos resultados permiten concluir que, si bien la utilización de la aplicación es intuitiva para la mayor parte de la muestra, es necesario revisar el flujo de interacciones para reducir la curva de aprendizaje para los usuarios.

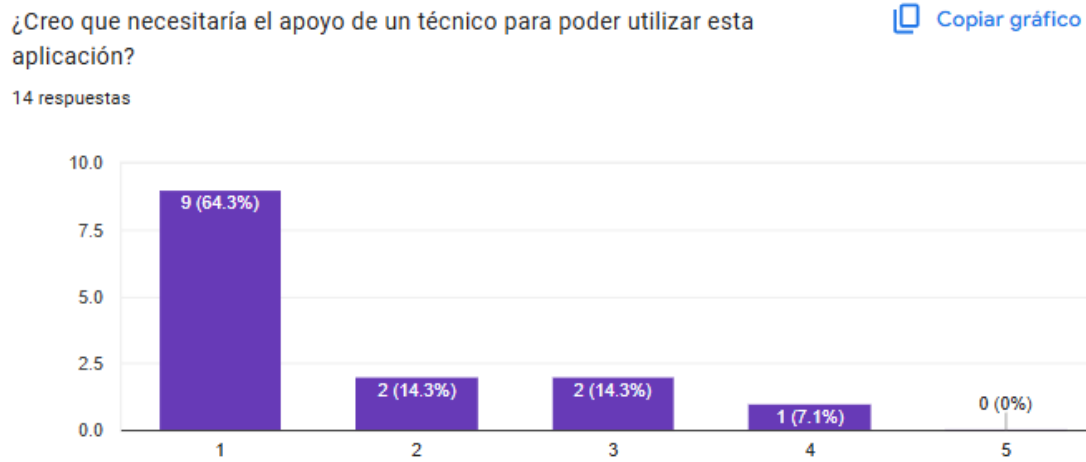


Figura A.15: SUS Pregunta 4

El cuarto gráfico figura A.15 se observa una valoración mixta respecto a si necesitarían el apoyo de un técnico para utilizar la aplicación. Mientras que once participantes manifiestan autonomía, dos manifiestan una postura neutra y uno afirma que requeriría apoyo técnico. Estos resultados evidencian la necesidad de simplificar el diseño de las interfaces y la lógica de las funcionalidades para mejorar una experiencia de uso autodidacta.

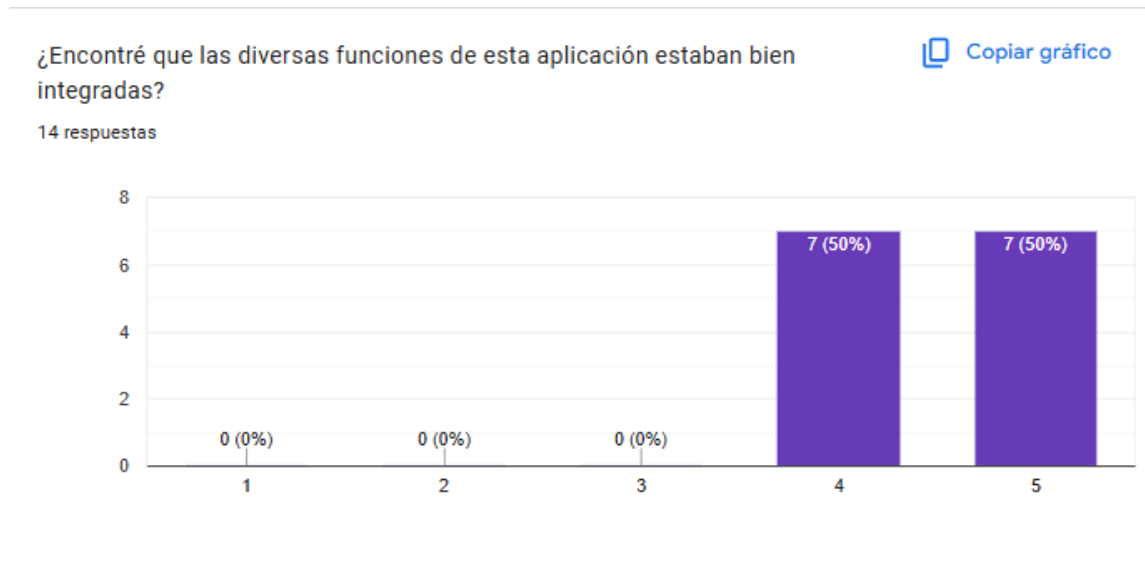


Figura A.16: SUS Pregunta 5

El quinto gráfico figura A.16 se observa una recepción positiva en las valoraciones de los encuestados, quienes concuerdan en que las diversas funciones de la aplicación están bien integradas. Este resultado permite concluir que la estructura de la información y el flujo de navegación son coherentes, permitiendo una experiencia del usuario fluida sin interrupciones entre los distintos módulos de la aplicación.



Figura A.17: SUS Pregunta 6

En el sexto gráfico figura A.17 se observa una disparidad en las valoraciones respecto a la presencia de inconsistencias en la aplicación. Mientras que doce encuestados manifiestan su desacuerdo con esta afirmación, lo que sugiere un sistema robusto, se identifica un encuestado con postura neutra y otro que percibe inconsistencias en la aplicación. Este resultado demuestra que aunque la mayoría del flujo de la aplicación es estable, es necesario realizar una revisión a detalle para encontrar posibles errores de diseño o comportamiento que afectan a la fluidez del software.

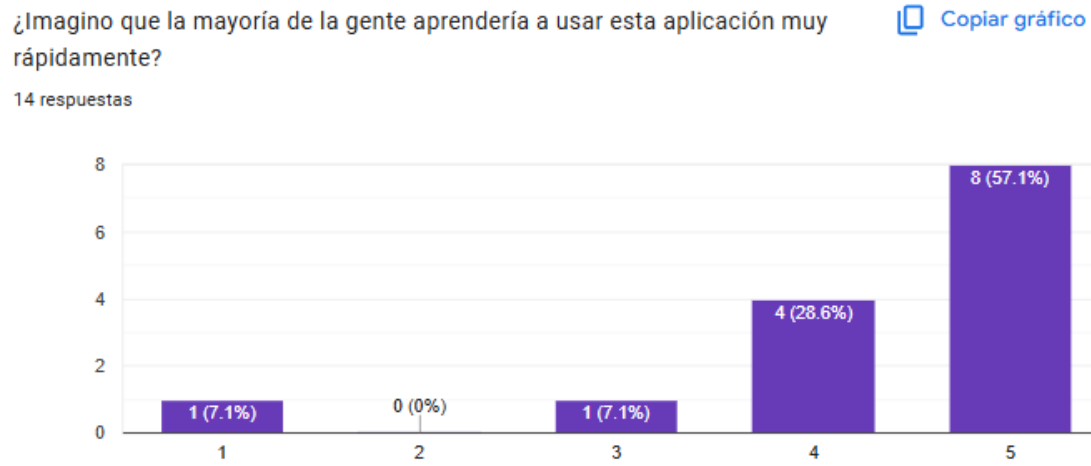


Figura A.18: SUS Pregunta 7

En el séptimo gráfico figura A.18 se observa una tendencia positiva respecto a la facilidad que se aprende a utilizar la aplicación. Mientras que doce encuestados están de acuerdo con que la mayoría de las personas aprenderían a usarla rápidamente, se identifica un encuestado con una postura neutra y otro que no está muy en desacuerdo. Los resultados demuestran que, es altamente intuitiva para la mayoría, es posible mejorar la curva de aprendizaje de la aplicación para obtener una aplicación más amigable para todo tipo de usuarios.

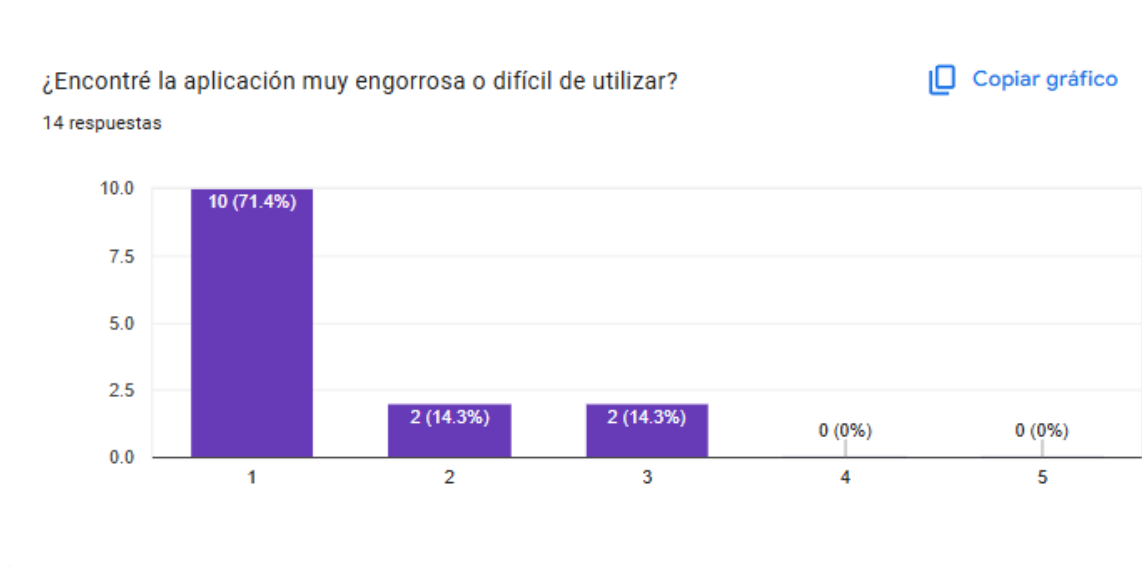


Figura A.19: SUS Pregunta 8

En el octavo gráfico figura A.19 se observa una valoración positiva respecto a la percepción de la fluidez del software, diez encuestados están muy en desacuerdo con que la aplicación sea molesta o difícil de utilizar, dos están en desacuerdo y dos tienen una valoración neutra. Estos resultados permiten concluir que la aplicación no representa una carga cognitiva elevada ni elementos constructivos, presentando un diseño minimalista, con una experiencia de uso accesible y cómoda para los usuarios.

¿Me sentí muy seguro/confiado al usar la aplicación?

 Copiar gráfico

14 respuestas

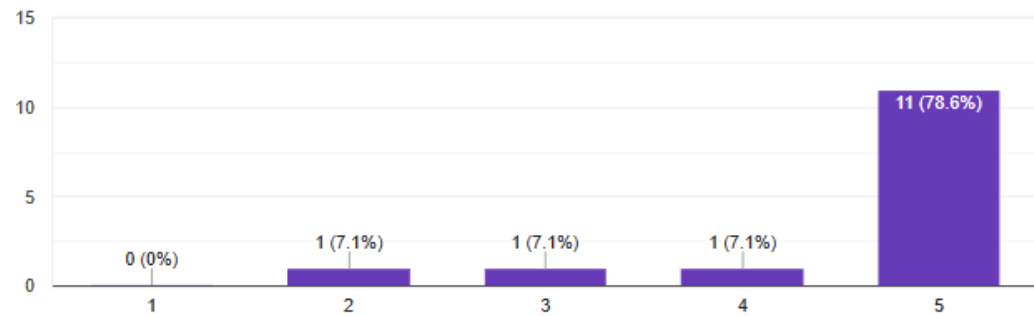


Figura A.20: SUS Pregunta 9

En el noveno gráfico figura A.20 se observa una valoración mayoritariamente positiva con respecto a la seguridad y confianza percibida al usar la aplicación, doce de los encuestados manifiestan estar de acuerdo con esta afirmación, mientras se identifica uno con una postura neutral y otro en desacuerdo. Estos resultados consolidan la fiabilidad de la aplicación, aun que es necesario fortalecer la retroalimentación visual de la aplicación hacia el usuario para ayudar todo tipo de usuario.

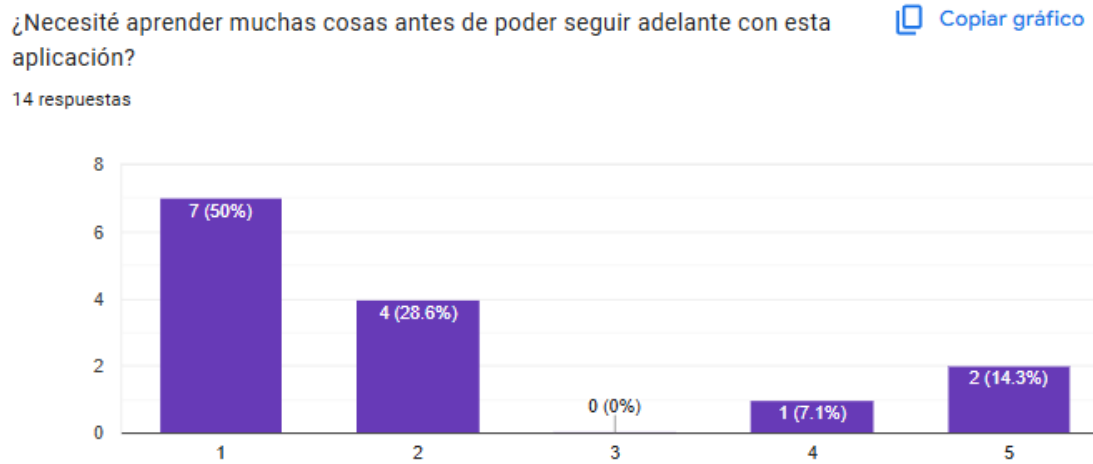


Figura A.21: SUS Pregunta 10

En el último gráfico figura A.21 se observa un leve disparidad con respecto a la necesidad de adquirir conocimientos previos para utilizar la aplicación. Mientras que once encuestados manifiestan su desacuerdo con esta afirmación, mientras que tres señalan que requirieron aprender conceptos previos para su utilización. La obtención de estos resultados refleja que, a pesar de tener una recepción general favorable, es pertinente implementar materiales de apoyo que faciliten la continuidad de uso y reduzca la barrera de entrada para nuevos usuarios con todo tipo de perfiles.

A.7. Implementación

A.7.1. Figuras Primer incremento

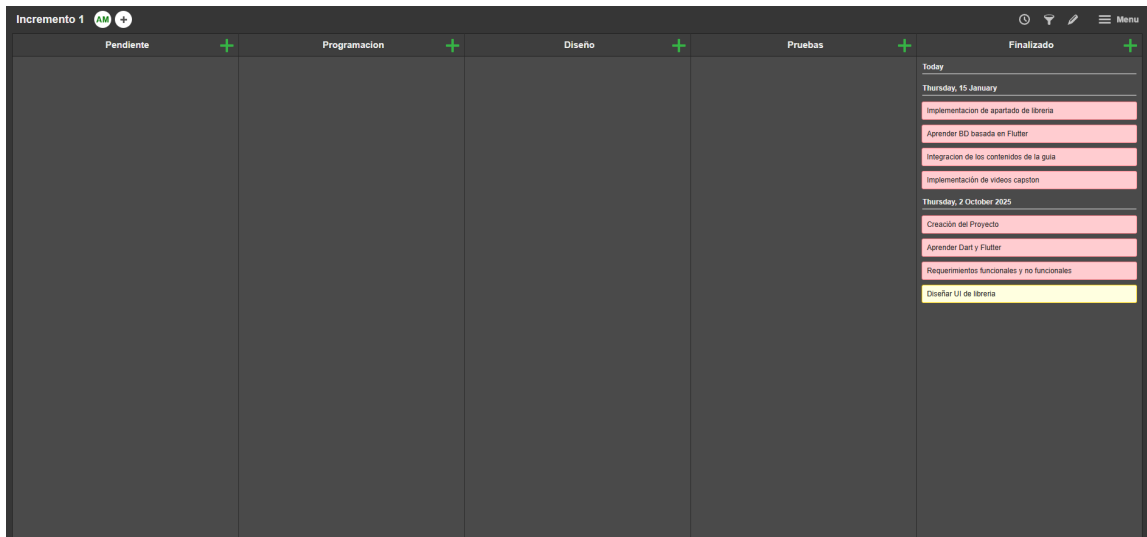


Figura A.22: KanbanFlow Primer Incremento.

```
30 dependencies:
31   connectivity_plus: ^7.0.0
32   internet_connection_checker_plus: ^2.9.1+2
33   curved_navigation_bar: ^1.0.6
34   flutter:
35     sdk: flutter
36   shared_preferences: ^2.5.3
37   get: ^4.7.2
38   dio: ^5.9.0
39   supabase_flutter: ^2.10.3
40   flutter_form_builder: ^10.2.0
41   video_player: ^2.10.0
42   flutter_native_splash: ^2.4.7
43   form_builder_validators: ^11.2.0
```

Figura A.23: Dependencias de Flutter

```
import 'package:supabase_flutter/supabase_flutter.dart';

Future<void> main() async {
  WidgetsFlutterBinding.ensureInitialized();

  await Supabase.initialize(
    url: 'YOUR_SUPABASE_URL',
    publishableKey: 'YOUR_SUPABASE_PUBLISHABLE_KEY',
  );
  runApp(MyApp());
}
```

Figura A.24: Línea de Inicialización

```
1  import 'package:supabase_flutter/supabase_flutter.dart';
2
3  import '../config/supabase_client.dart';
4
5  @
6  abstract class BaseProvider {
7      SupabaseClient get client => AppConfig.supabase;
8
9      SupabaseQueryBuilder table(String name) => client.from(name);
10 }
```

Figura A.25: Script Plantilla de llamado de datos

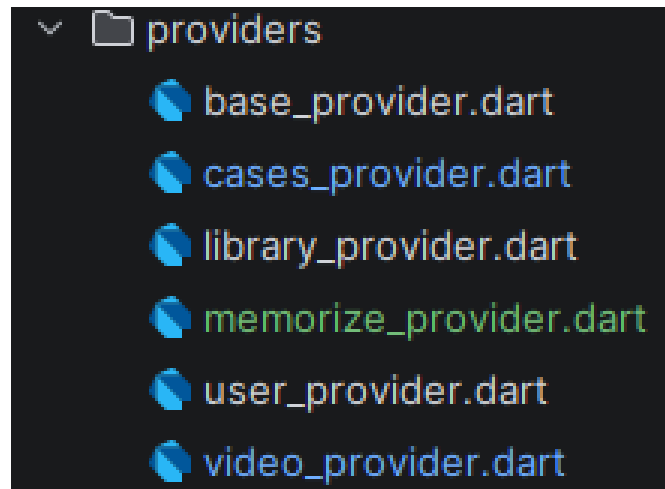


Figura A.26: Scripts Provider de cada Tabla

```
class LibraryProvider extends BaseProvider {
  static const String _table = 'Library';

  Future<List<Map<String, dynamic>>> getAll() async {
    final res = await table(_table).select();
    return (res as List).cast<Map<String, dynamic>>();
  }

  Future<List<Map<String, dynamic>>> getByType(String type) async {
    final res = await table(_table).select().eq('type_rec', type);
    return (res as List).cast<Map<String, dynamic>>();
  }

  Future<List<Map<String, dynamic>>> getBySubcategory(String sub) async {
    final res = await table(_table).select().eq('Subcateg', sub);
    return (res as List).cast<Map<String, dynamic>>();
  }
}
```

Figura A.27: Funciones Tabla de Librería

```

child: Obx(() {
  if (controller.loading.value) {
    return const Center(child: CircularProgressIndicator());
  }
  if (controller.items.isEmpty) {
    return const Center(child: Text('Sin contenidos'));
  }
  return ListView.separated(
    itemCount: controller.items.length,
    separatorBuilder: (_, _) => const Divider(height: 1),
    itemBuilder: (_, i) {
      final item = controller.items[i];
      return ExpansionTile(
        title: Text(
          'Recomendacion ${item['number'] ?? ''}',
          style: AppTextStyle.Title1,
        ),
        subtitle: Text(
          'Categoria: ${item['type_rec'] ?? ''}',
          style: AppTextStyle.body1,
        ),
        children: [
          ListTile(
            title: controller.subCategory(item)
              ? Text(
                  'Tipo: ${item['Subcateg'] ?? ''}',
                  style: AppTextStyle.Subtitle2,
                )
              : null,
          ),
          ListTile
        ],
      );
    },
  );
}

```

Figura A.28: Código Librería

```

@override
void onInit() {
  super.onInit();
  fetchAll();
}

Future<void> fetchAll() async {
  loading.value = true;
  try {
    items.assignAll(await _provider.getAll());
  } finally {
    loading.value = false;
  }
}

Future<void> filterByType(String type) async {
  loading.value = true;
  typeFilter.value = type;
  subFilter.value = '';
  try {
    items.assignAll(await _provider.getByType(type));
  } finally {
    loading.value = false;
  }
}

```

Figura A.29: Código Controlador Librería

A.7.2. Figuras segundo incremento

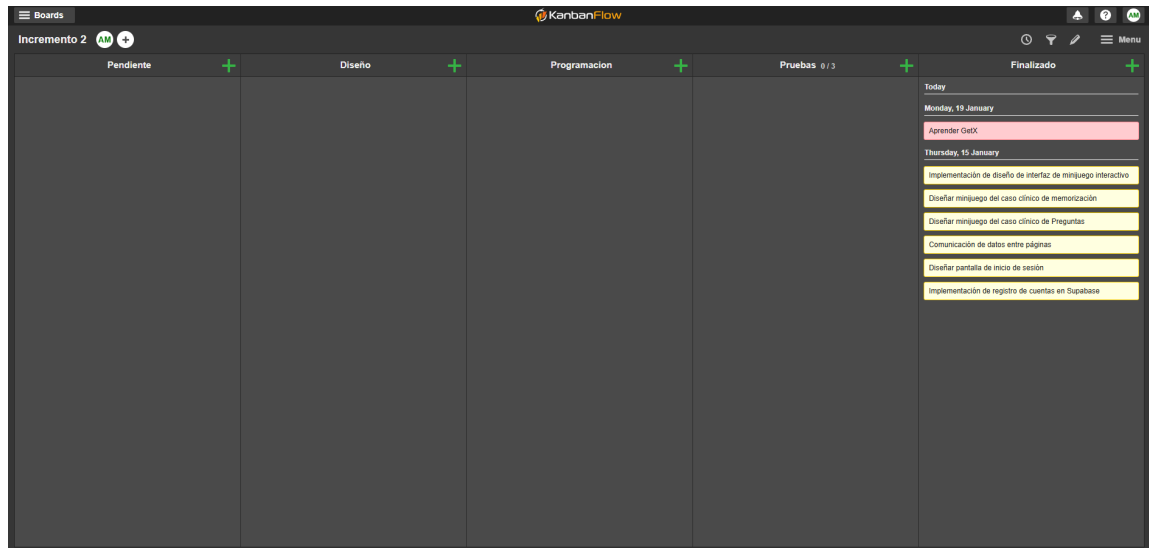


Figura A.30: KanbanFlow Segundo Incremento.

```
import 'package:get/get.dart';
import 'library_controller.dart';

class LibraryBinding extends Bindings {
  @override
  void dependencies() {
    Get.lazyPut<LibraryController>(() => LibraryController());
  }
}
```

Figura A.31: Código Binding GetX

```
import ...

class LibraryController extends GetxController {
  final LibraryProvider _provider = LibraryProvider();

  final RxList<Map<String, dynamic>> items = <Map<String, dynamic>>[].obs;
  final RxString typeFilter = ''.obs;
  final RxString subFilter = ''.obs;
  final RxBool loading = false.obs;
  final RxInt statePage = 0.obs;

  @override
  void onInit() {
    super.onInit();
    fetchAll();
  }

  Future<void> fetchAll() async {
    loading.value = true;
    try {
      items.assignAll(await _provider.getAll());
    } finally {
      loading.value = false;
    }
  }
}
```

Figura A.32: Código Controlador

```
FormBuilderTextField(
  name: 'email',
  decoration: const InputDecoration(labelText: 'Email'),
  keyboardType: TextInputType.emailAddress,
  validator: FormBuilderValidators.compose([
    FormBuilderValidators.required(),
    FormBuilderValidators.email(),
  ]),
), FormBuilderTextField
```

Figura A.33: Código FormBuilderTextField

Figuras Juego de Preguntas

```
113 class _OptionTile extends StatelessWidget {  
114   const _OptionTile({required this.label, required this.text, required this.selected, required this.onTap});  
115   final String label;  
116   final String text;  
117   final bool selected;  
118   final VoidCallback onTap;  
119  
120  
121   @override  
122   Widget build(BuildContext context) {  
123     return Card(  
124       color: selected ? Colors.cyanAccent.shade400 : null,  
125       elevation: 2.0,  
126       child: ListTile(  
127         leading: CircleAvatar(child: Text(label)),  
128         title: Text(text),  
129         onTap: onTap,  
130         trailing: selected ? Icon(Icons.check,color: Colors.white,) : null  
131       ),  
132     );  
133   }  
134 }
```

Figura A.34: Función plantilla de opciones

```
Future<void> load() async {  
  loading.value = true;  
  try {  
    var data = await _provider.getAll();  
    questions.assignAll(data);  
    questions.shuffle(); // Mezclar preguntas al inicio  
    index.value = 0;  
    answerAccurate = 0;  
    answer = [];  
    completed = false;  
    updateProgress();  
    selected.value = null;  
  } finally {  
    loading.value = false;  
  }  
}
```

Figura A.35: Función load

```
63 void next() {
64     if (questions.isEmpty || selected.value == null) return;
65
66     String correctAnswer = questions[index.value]['Correct'];
67
68     if (selected.value == correctAnswer) {
69         answerAccurate++;
70         correctChoose.value = correctAnswer;
71     }
72
73     answer.add(selected.value);
74
75     Timer(const Duration(seconds: 2), () {
76         if (index.value < maxQuestions - 1) {
77             index.value++;
78             selected.value = null;
79             correctChoose.value = null;
80             updateProgress();
81         } else {
82             finalizarJuego();
83         }
84     }); Timer
85 }
```

Figura A.36: Función next

```
88 void finalizarJuego() {
89     completed = true;
90     Get.offNamed('/feedback', arguments: {
91         'Feedback': questions.toList(), // Envia la lista mezclada actual
92         'answer': List.from(answer),    // Copia de las respuestas dadas
93         'maxQuestion': maxQuestions,
94     });
95 }
96
97
```

Figura A.37: Función finalizarJuego

Figuras Juego de Memoria

```
class CardModel{
    final Color color;
    bool isFlipped;
    bool isMatch;
    String contentCard;
    String pairCard;
    final Color pairColor;
    String cardType;

    CardModel({
        required this.color,
        this.isFlipped = false,
        this.isMatch = false,
        required this.contentCard,
        required this.pairCard,
        required this.pairColor,
        required this.cardType
    });
}
```

Figura A.38: Código de Clase CardModel

```
void selectCard(int index) async {
    if (_isProcessing.value || cards[index].isFlipped || cards[index].isMatch || !gameState.value) {
        return;
    }
    _isProcessing.value = true;

    cards[index].isFlipped = true;
    _selectedIndices.add(index);
    update();

    if (_selectedIndices.length == 2) {
        await _checkMatch();
        _selectedIndices.clear();
    }

    update();
    _isProcessing.value = false;

    _winCheck();
}
```

Figura A.39: Función selectCard

```
Future <void> _checkMatch() async {  
  
    int index1 = _selectedIndices[0];  
    int index2 = _selectedIndices[1];  
  
    if (cards[index1].pairCard == cards[index2].contentCard || cards[index2].pairCard == cards[index1].contentCard ) {  
  
        pairFound.value++;  
        progress.value = ((pairFound/(cardsLength ~/ 2))*100).toInt();  
        cards[index1].isMatch = true;  
        cards[index2].isMatch = true;  
  
    } else {  
        await Future.delayed(Duration(seconds: 3));  
  
        cards[index1].isFlipped = false;  
        cards[index2].isFlipped = false;  
        //_failedAttempt();  
  
    }  
}
```

Figura A.40: Función checkMatch

```
Future <void> flipAllCards() async{  
    if(token.value != 0 && gameState.value) {  
        _selectedIndices.clear();  
  
        for (var card in cards) {  
            if(card.isFlipped != true && card.isMatch != true)  
            {  
                card.isFlipped = true;  
            }  
        }  
        await Future.delayed(Duration(seconds: 5));  
        for (var card in cards) {  
            if(card.isMatch != true)  
            {  
                card.isFlipped = false;  
            }  
        }  
        token.value--;  
    }  
}
```

Figura A.41: Función flipAllCards

```

class _FlipAnimation extends StatelessWidget {
  final bool isFlipped;
  final Widget front;
  final Widget back;

  const _FlipAnimation({required this.isFlipped, required this.front, required this.back});

  @override
  Widget build(BuildContext context) {
    return AnimatedSwitcher(
      duration: const Duration(milliseconds: 400),
      transitionBuilder: (child, animation) {
        final rotate = Tween(begin: pi, end: 0.0).animate(animation);
        return AnimatedBuilder(
          animation: rotate,
          builder: (context, child) {
            final isBack = rotate.value > pi / 2;
            return Transform(
              transform: Matrix4.rotationY(rotate.value),
              alignment: Alignment.center,
              child: isBack ? const SizedBox() : child,
            );
          },
          child: child,
        );
      },
      child: isFlipped
        ? SizedBox(key: const ValueKey(1), child: front)
        : SizedBox(key: const ValueKey(2), child: back),
    );
  }
}

```

Figura A.42: Código FlipAnimation

A.7.3. Figuras tercer incremento

A.43

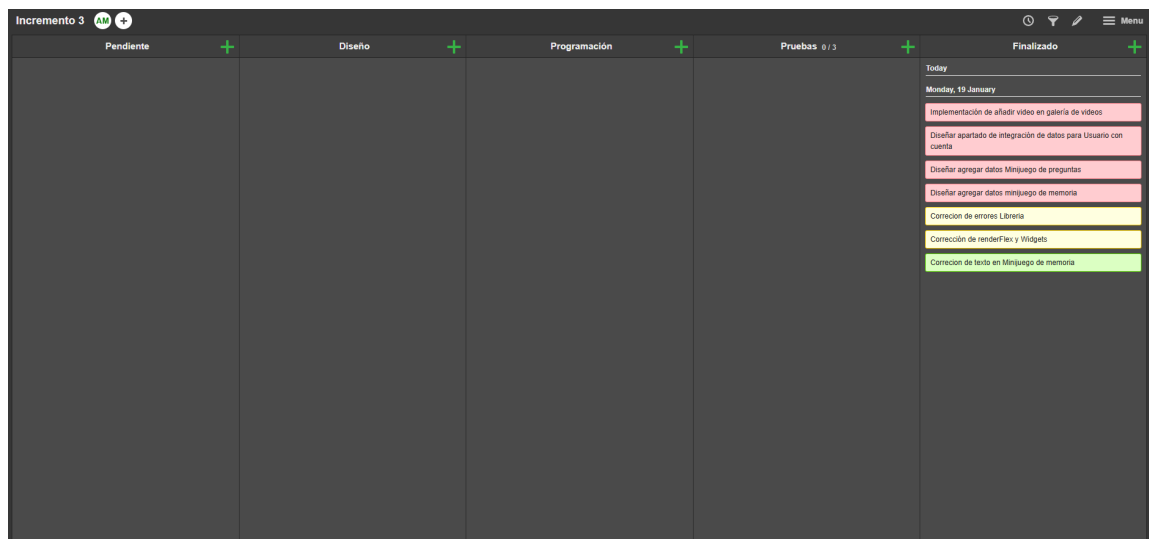


Figura A.43: KanbanFlow Tercer Incremento.

```

39 Future<bool> insertConcept() async {
40     final concept = textInputConcept.text.trim();
41     final definition = textInputDefinition.text.trim();
42
43     if (concept.isEmpty || definition.isEmpty) return false;
44
45     // Validación duplicados
46     if (items.any((i) => i['concept'].toString().toLowerCase() == concept.toLowerCase())) {
47         Get.snackbar("Cuidado", "Este concepto ya existe", backgroundColor: Colors.amber);
48         return false;
49     }
50
51     await _providerMem.addData(concept, definition);
52     await fetchAll();
53     textInputConcept.clear();
54     textInputDefinition.clear();
55     return true;
56 }
57
58 Future<bool> insertDataCase(String t, String sc, String q, String a, String b, String c, String ct, String fb, double n) async {
59     try {
60         await _providerCas.addData(t, sc, q, a, b, c, ct, fb, n);
61         await fetchAll();
62         return true;
63     } catch (e) {
64         return false;
65     }
66 }

```

Figura A.44: Código añadir Concepto y Caso clínico

```

39 Future<bool> insertConcept() async {
40     final concept = textInputConcept.text.trim();
41     final definition = textInputDefinition.text.trim();
42
43     if (concept.isEmpty || definition.isEmpty) return false;
44
45     // Validación duplicados
46     if (items.any((i) => i['concept'].toString().toLowerCase() == concept.toLowerCase())) {
47         Get.snackbar("Cuidado", "Este concepto ya existe", backgroundColor: Colors.amber);
48         return false;
49     }
50
51     await _providerMem.addData(concept, definition);
52     await fetchAll();
53     textInputConcept.clear();
54     textInputDefinition.clear();
55     return true;
56 }
57
58 Future<bool> insertDataCase(String t, String sc, String q, String a, String b, String c, String ct, String fb, double n) async {
59     try {
60         await _providerCas.addData(t, sc, q, a, b, c, ct, fb, n);
61         await fetchAll();
62         return true;
63     } catch (e) {
64         return false;
65     }
66 }

```

Figura A.45: Código eliminar Concepto y Caso clínico

```
// Controladores para Memorice
final textInputConcept = TextEditingController();
final textInputDefinition = TextEditingController();
```

Figura A.46: Variables TextInput

```
dependencies:
  connectivity_plus: ^7.0.0
  internet_connection_checker_plus: ^2.9.1+2
```

Figura A.47: Dependencias para comprobar conexión a internet

```
19 void _handleStatusChange(InternetStatus status){
20
21     if(status == InternetStatus.disconnected){
22         _startTimeOut();
23         if(Get.currentRoute != AppRoutes.notNetwork){
24             Get.toNamed(AppRoutes.notNetwork);
25         }
26     }
27     else
28     {
29         _stopTimeOut();
30         if(Get.currentRoute == AppRoutes.notNetwork){
31             Get.back();
32         }
33     }
34 }
```

Figura A.48: Función que comprueba el estado de la red

Design Document

Aplicación móvil interactiva de la guía de buenas prácticas:
Cuidados Centrados en la persona y la familia

Índice

Resumen del documento	2
Alcance	2
Público objetivo	2
Plataforma	2
Tecnología a usar	3
Diseño de Minijuegos	3
Requerimientos	3
Requerimientos funcionales	3
Descripción de casos de uso	5
Requerimientos no funcionales	5

Resumen del documento

Este documento está diseñado para presentar los distintos aspectos que son necesarios al momento de desarrollar un software. Su propósito es presentar los aspectos y requisitos del proyecto, así como definir el alcance que se tiene planeado para el software e identificar el público objetivo, con el que se espera trabajar al momento de llevar a cabo las pruebas. Esta aplicación tiene el enfoque de presentar las recomendaciones, información de la guía de buenas prácticas basada en cuidados centrados en la persona y la familia, con un fácil acceso y un apartado interactivo con minijuegos de preguntas y memorización basados en casos clínicos, integración de datos mediante una cuenta de administrador para expandir los minijuegos. También la presentación de una retroalimentación escrita por el profesor para los usuarios, tanto para el minijuego de preguntas como para el de memoria.

Alcance

En el alcance de este proyecto se tiene planeado incluir como mínimo los dos tipos de usuario, la biblioteca de recomendaciones con la galería de recomendaciones funcionando, también dos minijuegos; uno de preguntas coloca al usuario a responder un caso clínico, con una pregunta, escenario y el tipo de caso. Para el de memorización se presenta un total de 12 cartas, con 6 pares encontrables; el usuario debe encontrarlos antes de finalizar el tiempo. Al terminar, se presentará el puntaje y las opciones de cerrar la ventana de diálogo o volver a jugar; si cierra la ventana de diálogo, el usuario puede observar la retroalimentación, donde cada uno de los pares tiene su color propio, permitiendo asociarlos de manera más sencilla. Por parte del apartado de integración de datos para los administradores, se espera que puedan agregar vídeos, añadir nuevas definiciones y conceptos para el minijuego de memoria, agregar nuevos casos clínicos al minijuego de preguntas y eliminar datos de la base de datos desde la aplicación. Público objetivo: El desarrollo de la aplicación tiene como su principal usuario a los estudiantes de la carrera de enfermería, más en específico a quienes hacen uso de la guía de buenas prácticas encargada de los cuidados centrados en la persona y la familia, que es entregada por RNAO y la BPSO.

Plataforma

Esta aplicación se encuentra planificada para funcionar en dispositivos móviles con pantalla táctil que dispongan de un sistema operativo Android.

Tecnología a usar

Para la creación de este software se han seleccionado las siguientes herramientas con las cuales se tiene pensado trabajar al momento de desarrollar el proyecto, las cuales son las siguientes:

- Android Studio
- Flutter
- Dart
- Kanban Flow
- Gantt project
- Supabase

Diseño de Minijuegos

1. Minijuego de preguntas: En este minijuego se presentarán nueve preguntas que el usuario debe responder o saltar; cada una de estas está diseñada por el profesor y añade sus respectivas respuestas. Luego de que el usuario responde las preguntas, se presenta una pantalla de retroalimentación, donde se muestran las acertadas e incorrectas; también se le permite al usuario explorar las preguntas y ver su respuesta y la respuesta correcta, con una justificación escrita por el profesor.
2. Minijuego de memorización: En este minijuego se presentará un total de seis pares, los cuales el usuario deberá encontrar. Este dispondrá de un botón que le permitirá observar el contenido de cada una de las cartas; para usarlo es necesario tener los suficientes tokens que se le entregan al usuario al comenzar la partida; este dispondrá de cinco usos. Al finalizar, ya sea que se le acabara el tiempo o haya encontrado todos los pares, se presentará una pantalla emergente felicitándolo y mostrándole la cantidad de pares encontrados, y también dos botones, uno para cerrar la ventana y otro para volver a jugar.

Requerimientos

En este apartado se presentaron las dos categorías de requerimientos que son necesarios al momento de desarrollar una aplicación o software. Los

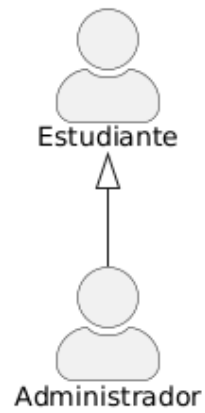
requerimientos funcionales se presentan mediante diagramas de casos de uso en donde se describen las funciones específicas que posee el software de manera interna. Los requerimientos no funcionales especifican cómo debe comportarse el sistema; dentro de los aspectos que se pueden presentar están la seguridad, usabilidad y escalabilidad; estos son cruciales para definir una buena experiencia para el usuario y la eficacia general del software. Estos dos elementos son esenciales al momento de desarrollar un software que cumpla con las necesidades del usuario.

Requerimientos funcionales

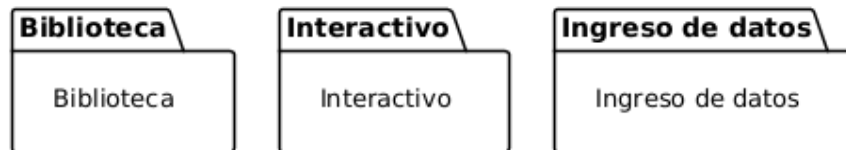
- RF1: La aplicación debe permitir al usuario Estudiante y Administrador ingresar con un correo y contraseña.
- RF2: La aplicación debe permitir al usuario Estudiante y Administrador acceder a los contenidos de la guía de buenas prácticas (Recomendaciones y niveles de evidencia).
- RF3: La aplicación debe permitir al usuario Estudiante y Administrador ver videos de las recomendaciones.
- RF4: La aplicación debe permitir al usuario Estudiante y Administrador jugar un minijuego de preguntas, donde se presentan un listado de casos clínicos con preguntas, escenarios, tipo y sus respuestas, entregados por el cliente.
- RF5: La aplicación debe permitir al usuario Estudiante y Administrador, jugar un minijuego de tipo memorice, con cartas al azar con definiciones y conceptos, tomadas de la guía de buenas prácticas Cuidado Centrado en la Persona y Familia.
- RF6: La aplicación debe permitir al usuario Administrador gestionar los contenidos de los minijuegos, agregar o eliminar contenidos.
- RF7: La aplicación debe permitir al usuario Administrador gestionar los contenidos de la librería, agregar o eliminar contenidos de la galería de videos.
- Escenario principal:



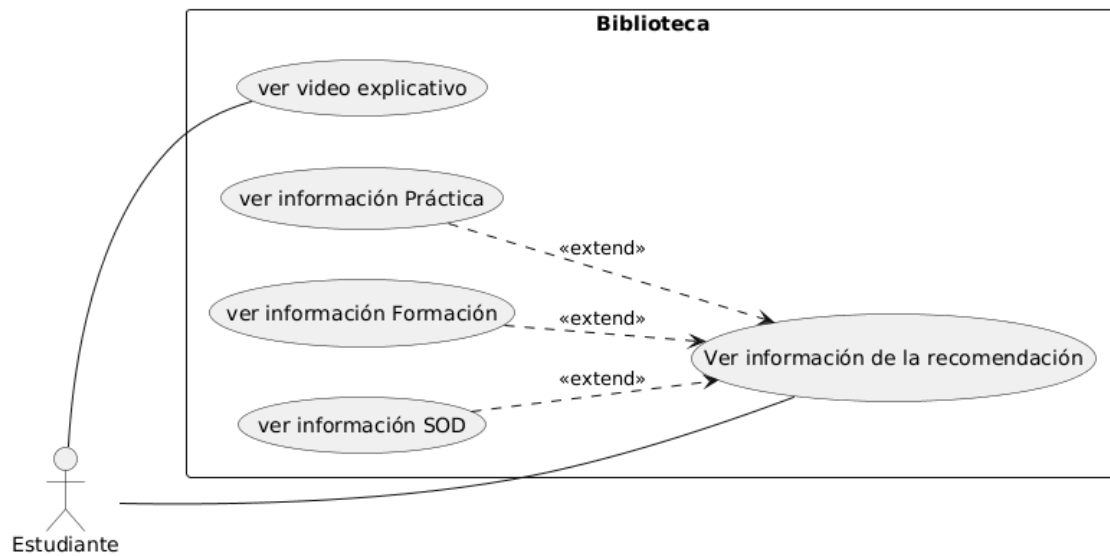
- Actores:

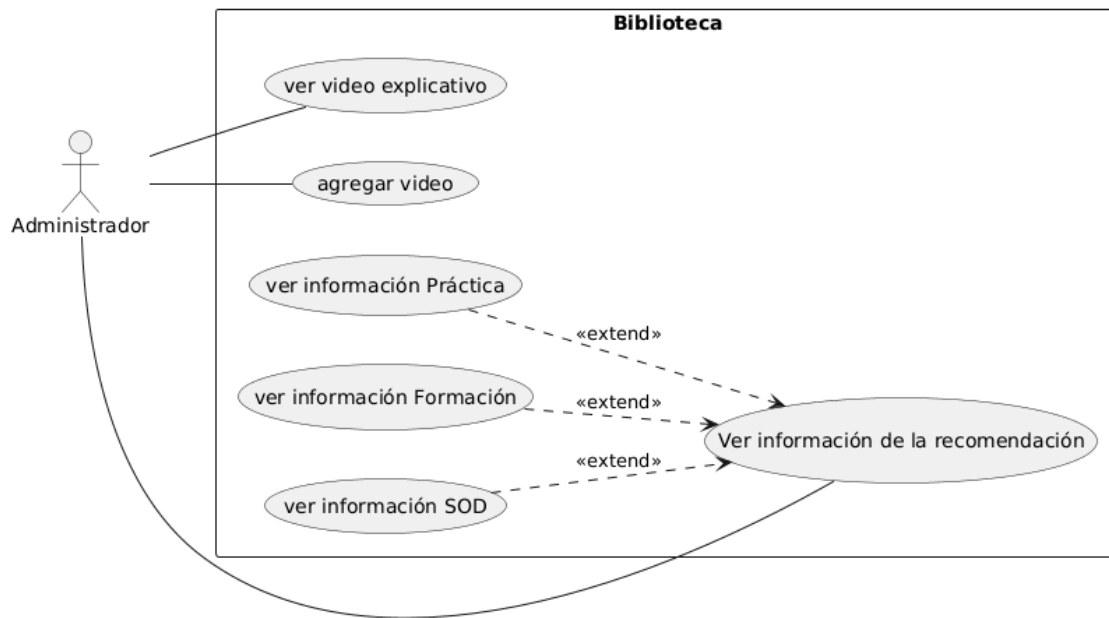


- Escenarios Secundarios

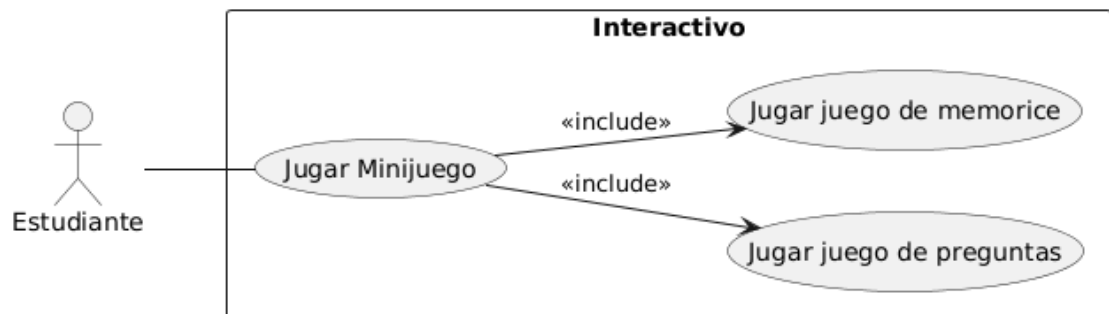


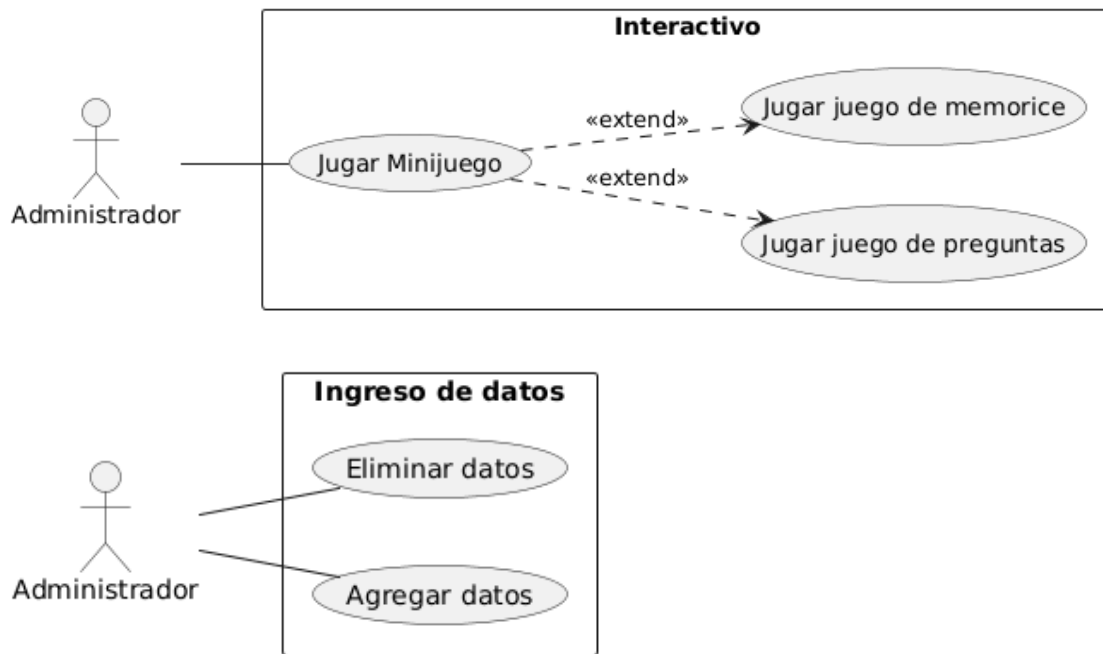
- Escena de Biblioteca





- : Escena interactiva usuario





Descripción de escenarios

1. Escena de biblioteca:
 - Paso 1: Ingresar al apartado de la biblioteca.
 - Paso 2: Seleccionar la información de la recomendación de un tipo de categoría: Práctica, Formación y Sistema, Organización y Directrices.
 - Paso 3: Seleccionar la recomendación de la que se quiere saber la información.
2. Escena interactiva:
 - Paso 1: Ingresar al apartado interactivo.
 - Paso 2: Seleccionar minijuego
 - Paso 3: Jugar el minijuego.

Requerimientos no funcionales

- RNF1: La aplicación debe tener los colores de la RNAO en sus diseños de interfaz.
- RNF2: La aplicación debe ser accesible para usuarios registrados en la base de datos.

- RNF3: La aplicación debe mostrar una retroalimentación al usuario en caso de no tener internet.
- RNF4: Debe ser un dispositivo móvil con sistema operativo android.
- Facilidad de uso: La aplicación será intuitiva para el usuario, facilitando la navegación entre las interfaces y tener un conocimiento previo de lo que va a encontrar en cada sección de la aplicación.
- Tiempo de respuesta: La aplicación tendrá un tiempo de respuesta mínimo en las interfaces y apartados interactivos, sin romper la experiencia del usuario con largos tiempos de espera.
- Robusto: La aplicación funcionará de manera confiable, sin presentar fallos, independientemente del contexto y circunstancia en que se encuentre. Su diseño garantizará un rendimiento constante, otorgándole a los usuarios una experiencia fluida y sin interrupciones.

Carta de Aceptación y Validación de Prototipo de Tesis

Fecha: 28 de abril, 2026

Asunto: Aceptación y conformidad de aplicación "Atlas".

A quien corresponda:

Por medio presente, se hace constar que Boris Vera Barrera, profesor de la Universidad de Talca, ha recibido y evaluado la aplicación denominada "Atlas", desarrollada por Alan Monsalve Salas.

Declaramos nuestra conformidad y aceptación del software entregado. Se certifica que la aplicación cumple con los requisitos técnicos y operativos establecidos originalmente, destacando su desempeño en los pilares de la usabilidad, fiabilidad y utilidad de la información.

Atentamente,
(Firma)



Observaciones

Quiero destacar el trabajo realizado de Alan Monsalve en el diseño de la app para la Guía de Cuidados Centrados en la persona y familia para las/los estudiantes de la carrera de Enfermería de la Universidad de Talca, evidenciando una consistente comprensión del enfoque y su adecuada traducción a una herramienta digital funcional y fácil de usar. Asimismo, demostró una actitud proactiva, amabilidad y disposición al trabajo colaborativo, manteniendo una comunicación respetuosa y abierta a la retroalimentación. Su propuesta refleja compromiso y un valioso aporte al trabajo interdisciplinario entre salud y desarrollo digital.
