
Desarrollo de una aplicación para la detección
de conductas de riesgo en adolescentes
Development of an Application for the
Detection of Risk Behaviors in Adolescents



Trabajo de Fin de Grado
Curso 2023–2024

Autores

Pilar Gutiérrez Merino

Li Jie Chen Chen

Directores

Gonzalo Méndez Pozo

Pablo Gervás Gómez-Navarro

Grado en Ingeniería Informática

Facultad de Informática

Universidad Complutense de Madrid

Desarrollo de una aplicación para la
detección de conductas de riesgo en
adolescentes

Development of an Application for the
Detection of Risk Behaviors in Adolescents

Trabajo de Fin de Grado en Ingeniería Informática

Autores

Pilar Gutiérrez Merino

Li Jie Chen Chen

Directores

Gonzalo Méndez Pozo

Pablo Gervás Gómez-Navarro

Convocatoria: *Septiembre 2024*

Grado en Ingeniería Informática

Facultad de Informática

Universidad Complutense de Madrid

13 de septiembre de 2024

Dedicatoria

A todas aquellas personas que enfrentan desafíos en su camino hacia la salud mental. A aquellos que buscan la comprensión, la aceptación y la curación y que se esfuerzan cada día por encontrar la luz en la oscuridad, por superar los obstáculos y por mantener la esperanza viva. Esperamos que este trabajo pueda ser de alguna ayuda en su camino hacia la recuperación y el bienestar.

Agradecimientos

Queremos expresar nuestro más sincero agradecimiento a todas aquellas personas que han apoyado y guiado nuestro camino hacia la realización de este trabajo.

En primer lugar, queremos agradecer a nuestro profesor, Gonzalo, por su dedicación, sabiduría y apoyo constante. Su guía y orientación han sido fundamentales para el éxito de este proyecto.

También queremos agradecer a nuestras familias, que han sido una fuente de amor, apoyo y motivación a lo largo de estos años universitarios, y en especial en este trabajo de fin de curso. Su paciencia y comprensión han sido esenciales para superar los desafíos y alcanzar nuestros objetivos.

Finalmente, queremos agradecer a Dios por la oportunidad de haber estudiado este grado, que nos encanta y por la bendición de contar con personas tan especiales en nuestras vidas. Su amor y providencia han sido una constante fuente de inspiración y fortaleza.

A todos ellos, nuestra más sincera gratitud y aprecio.

Resumen

Desarrollo de una aplicación para la detección de conductas de riesgo en adolescentes

La salud mental de los jóvenes es un tema de gran importancia en la sociedad actualmente, especialmente por el incremento de conductas autolesivas y suicidas. Es fundamental encontrar formas efectivas de prevenir y tratar estas conductas.

En este contexto, la tecnología móvil nos ha ofrecido una oportunidad para desarrollar una posible herramienta de apoyo a la solución de este problema.

Esta herramienta será una aplicación móvil dirigida a estos jóvenes, a sus tutores legales y a los psicólogos encargados de ellos. En esta memoria se explica cómo ha sido el proceso de diseño de esta herramienta, y se va descubriendo poco a poco la idea de ir registrando mediciones instantáneas de las emociones de los jóvenes en cualquier momento del día para, finalmente, ser analizado y gestionar una solución personalizada valorando la gravedad de cada joven, ajustando el tipo de preguntas y la frecuencia de los tests.

Palabras clave

Salud mental, auto-lesiones, suicidio, psicología, app, prevención, cuestionario, medición instantánea

Abstract

Development of an Application for the Detection of Risk Behaviors in Adolescents

Mental health of young people is a highly significant issue in today's society, especially due to the increase in self-harming and suicidal behaviors. It is crucial to find effective ways to prevent and address these behaviors.

In this context, mobile technology has provided us with an opportunity to develop a potential tool to support the solution to this problem.

This tool will be a mobile application aimed at young people, their legal guardians, and the psychologists responsible for their care. This document explains the process of designing this tool and gradually reveals the idea of recording real-time measurements of young people's emotions at any time of day. Ultimately, this data will be analyzed to manage a personalized solution by assessing the severity of each individual's condition, adjusting the type of questions and the frequency of the tests.

Keywords

Mental, health, self-harm, suicide, psychology, app, prevention, questionnaire

Índice

1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	2
1.3. Plan de trabajo	2
1.4. Estructura de la Memoria	3
2. Estado de la Cuestión	5
2.1. Introducción	5
2.2. Aplicaciones relacionadas	5
3. Descripción del Trabajo	13
3.1. Introducción	13
3.2. Arquitectura de la aplicación	13
3.2.1. Descripción de la arquitectura	14
3.2.2. Diagrama de Arquitectura	15
3.2.3. Explicación de los Diagramas	16
3.2.4. Conclusión	16
3.3. Diseño centrado en el usuario de la aplicación	16
3.3.1. Investigación del usuario	18
3.3.2. Definición de requisitos	22
3.3.3. Diseño de la interfaz centrado en la usabilidad	28
3.3.4. Pruebas con usuarios	38
3.3.5. Iteración y mejora continua	38
3.4. Desarrollo de la aplicación móvil en React Native	39
3.4.1. Librerías y Herramientas Utilizadas en React Native	39

3.4.2.	Diseño de las pantallas	41
3.4.3.	Diagrama de Navegación de la Aplicación	49
3.5.	Desarrollo de la Base de Datos en MySQL	50
3.5.1.	Relaciones entre Tablas de la Base de Datos	51
3.5.2.	Detalles de la estructura de la base de datos	52
3.6.	Desarrollo del Servidor en Flask	57
3.6.1.	Detalles del Desarrollo del Servidor	58
3.6.2.	Ejecución del Servidor	68
3.7.	Implementación y Funcionamiento de Notificaciones Push	69
3.7.1.	Configuración de Firebase Cloud Messaging (FCM)	69
3.7.2.	Integración en la Aplicación React Native	69
3.7.3.	Configuración del Servidor Flask	70
3.7.4.	Funcionamiento General y Ventajas de la Implementación	70
3.8.	Funcionamiento de la aplicación móvil	71
3.8.1.	Interfaz Profesional	72
3.8.2.	Interfaz Adolescente	72
3.8.3.	Interfaz Padre	73
4.	Conclusiones y Trabajo Futuro	75
4.1.	Conclusiones	75
4.2.	Trabajo Futuro	76
4.3.	Conclusión General	77
	Introduction	79
	Conclusions and Future Work	83
	Contribuciones Personales	87
	Bibliografía	93

Índice de figuras

2.1. Pacifica	6
2.2. TalkSpace	6
2.3. MoodTune	7
2.4. DailyBean	8
2.5. Petit Bambou	8
2.6. Fabulous	9
2.7. Fabulous 2	9
2.8. DaylioDiario	10
2.9. DaylioDiario2	10
3.1. Diagrama de Arquitectura Sivaría	15
3.2. Entrada y salida del sistema experto	18
3.3. Jaime	23
3.4. Javier	24
3.5. Sol	26
3.6. Pantalla Iniciar Sesión	29
3.7. Pantalla Adolescente	30
3.8. Pantalla Padre	31
3.9. Pantalla Padre	31
3.10. Pantalla Profesional	32
3.11. Pantalla Profesional	32
3.12. Pantalla Cuestionario	33
3.13. Pantalla Cuestionario	33
3.14. Pantalla Enviar respuestas voluntarias	34
3.15. Pantalla Ver progresos	35

3.16. Pantalla Citas Adolescente	36
3.17. Pantalla Citas Profesional 1	37
3.18. Pantalla Citas Profesional 2	37
3.19. Interfaz Iniciar Sesión	42
3.20. Interfaz Adolescente	43
3.21. Interfaz Padre	44
3.22. Interfaz Profesional	45
3.23. Interfaz Cuestionario Uniselección	46
3.24. Interfaz Cuestionario Multiselección	46
3.25. Interfaz Cuestionario BarStatus	46
3.26. Interfaz Respuestas Voluntarias	47
3.27. Interfaz Ver progreso	48
3.28. Interfaz Final	49
3.29. Diagrama de Navegación	50
3.30. Diagrama de Relaciones entre las Tablas de la Base de Datos	52
3.31. Diagrama de secuencia Verificar Contraseña	59
3.32. Diagrama de flujo Verificar Contraseña	60
3.33. Diagrama de secuencia Token	61
3.34. Diagrama de flujo Token	61
3.35. Diagrama de secuencia guardar respuestas	62
3.36. Diagrama de flujo guardar respuestas	62
3.37. Diagrama de secuencia enviar preguntas	63
3.38. Diagrama de flujo enviar preguntas	64
3.39. Diagrama de secuencia Datos gráfica	65
3.40. Diagrama de flujo Datos gráfica	65
3.41. Diagrama de secuencia Guardar Texto	66
3.42. Diagrama de flujo Guardar Texto	66
3.43. Diagrama de secuencia enviar notificaciones	67
3.44. Diagrama de flujo enviar notificaciones	68
3.45. Funcionamiento aplicación Profesional	72
3.46. Funcionamiento aplicación Adolescente	73
3.47. Funcionamiento aplicación Padre	74

Índice de tablas

2.1. Comparativa de aplicaciones de salud mental	11
3.1. Variables del auto-informe	19
3.2. Variables de las familias	20
3.3. Variables de las profesionales	21
3.4. Tabla Usuario	53
3.5. Tabla de RelacionesHijos	53
3.6. Tabla RelacionProfHijo	54
3.7. Tabla RelacionTutorHijo	54
3.8. Tabla Variable	55
3.9. Tabla Pregunta	55
3.10. Tabla Test	56
3.11. Tabla Item	56

Introducción

1.1. Motivación

En la actualidad, el suicidio y las conductas autolesivas se han convertido en uno de los problemas de salud y social de primer orden, afectando a individuos de todas las edades, géneros y contextos socio-económicos (Baños-Chaparro, 2024), (Organization et al., 2021). Estas formas de sufrimiento psicológico y emocional no solo tienen un impacto devastador en las vidas de quienes las experimentan, sino que también generan repercusiones profundas en sus familias y en la sociedad en general.

Según datos de la (Organización Mundial de la Salud (OMS), 2014), en 2014, se estimó que cada año más de 800.000 personas pierden la vida por suicidio, lo que equivale a una persona cada 40 segundos. En 2017, se vio que la cifra de estas muertes en jóvenes al día, ascendió a más de 3000 muertes diarias, sumando un total de 1,2 millones de muertes de jóvenes cada año (Organización Mundial de la Salud (OMS), 2017), y convirtiéndose en la segunda causa más frecuente de muerte no accidental en la población de entre los 15 y 29 años.

Esta cifra es alarmante y cuando la conocimos, nos llevó a reflexionar sobre las medidas que podemos implementar para prevenir estas tragedias ya que pensamos que la salud mental debe ser una prioridad en todos los ámbitos, para asegurar que cada individuo tenga el apoyo y los recursos necesarios para mantener su bienestar psicológico.

Para prevenir el suicidio y las conductas autolesivas es necesario garantizar que las personas que están luchando con pensamientos suicidas tengan acceso a la atención adecuada y en el tiempo oportuno, aquí es donde encaja la idea del proyecto SIVARIA, que propone exactamente la implementación de la solución. El proyecto SIVARIA, propuesto por un equipo de psicólogos de la UNED, se basa en un sistema experto que, evaluando las respuestas a unas preguntas, seleccionadas por él mismo, mide la gravedad de la situación del adolescente. Nuestra motivación a hacer esta aplicación es generar esta aplicación para la recolecta de estas respuestas y así ayudar a esta herramienta que creemos que será una gran medida preventiva.

1.2. Objetivos

El objetivo principal de este trabajo de fin de grado es desarrollar una aplicación móvil que ayude a valorar y gestionar el riesgo de conductas autolesivas, tanto suicidas como no suicidas, en adolescentes de entre 12 y 21 años. Esta herramienta estará basada en un Sistema Experto diseñado por psicólogos de la UNED, quienes proporcionarán la base científica para su funcionamiento.

La aplicación permitirá realizar mediciones instantáneas de cómo se ha sentido el usuario en un determinado tiempo. Estas mediciones serán llevadas a cabo gracias a unos tests creados específicamente para el usuario, con una serie de preguntas de respuestas de selección. Estos tests, serán enviados a una hora específica del día, una o varias veces a la semana, con el importante fin de captar sus emociones de manera inmediata y facilitar la intervención si es necesario. Dichas preguntas deberán poder ser contestadas fácilmente para que se cumpla la característica principal de esta medición, la instantaneidad.

Además, será necesario crear un sistema que permita almacenar y gestionar de forma segura toda la información recogida, para que los psicólogos puedan analizar los datos y alimentar el Sistema Experto con las mediciones tomadas de manera adecuada. Todo esto se realizará garantizando la confidencialidad de los usuarios.

Para llevar a cabo este trabajo de fin de grado, se requerirá el uso de diversas herramientas tecnológicas para el desarrollo de la aplicación, la creación de un servidor que soporte el funcionamiento de la misma, y la creación y gestión de una base de datos.

1.3. Plan de trabajo

Para el plan de trabajo seguimos una pauta marcada desde principio de curso en el que dividimos el proyecto en 3 hitos intentando seguirlos estrictamente.

En el primer hito, tuvimos como objetivo principal escribir la motivación del proyecto, cómo nos íbamos a organizar y estudiamos la situación actual relacionada con el tema del trabajo para saber desde dónde partíamos y qué es lo que ya habían realizado otras personas. Después de todo esto tendríamos que desarrollar un prototipo tecnológico. En él, debíamos abordar los riesgos tecnológicos que pensásemos que podíamos tener durante el proyecto, determinando las tecnologías a usar haciendo un estudio previo de las posibles opciones con sus respectivos pros y contras. Los entregables para este hito consistían en el código del prototipo básico y una memoria parcial que abarcará la introducción, objetivos y el estado de la cuestión.

En el segundo hito, nos centramos en avanzar significativamente en el desarrollo del prototipo tecnológico, apuntando a alcanzar un 85 % de completitud en el código. Las tareas incluyeron la continuación de la codificación y la redacción de los capítulos centrales de la memoria. Los entregables para este hito fueron el código del prototipo (85 % completado) y una memoria parcial que abordó los capítulos centrales del proyecto.

En el tercer y último hito, se buscó finalizar completamente el desarrollo del prototipo tecnológico. Las tareas finales incluyeron la conclusión de la codificación y la redacción de los resultados y conclusiones en la memoria. Los entregables para este hito constaron del código del prototipo (100% completado) y la memoria final, que contenía ya los resultados y conclusiones obtenidas a lo largo del proyecto.

Además de los entregables correspondientes a cada hito, en cada uno de ellos se programaron reuniones regulares con el director del TFG para evaluar el progreso y realizar ajustes según fuese necesario.

También fue imprescindible tener reuniones con el equipo de psicólogos de la UNED para saber si nuestro prototipo cubría sus necesidades y de nuevo, realizar los ajustes que fuesen necesarios para garantizar que cumplíamos con los requisitos impuestos.

1.4. Estructura de la Memoria

1. Introducción

En este apartado de la memoria se presenta la introducción del proyecto, comenzando por la motivación que lo impulsa, destacando la importancia de abordar el suicidio y las conductas autolesivas en adolescentes. A continuación, se describen los objetivos del trabajo, para el desarrollo de la aplicación móvil descrita. Además, se detalla el plan de trabajo, dividido en tres hitos clave, que guía el desarrollo del proyecto y las reuniones con el equipo. Finalmente, se explica cómo se organiza la estructura de la memoria, ofreciendo un esquema claro de las secciones que componen el documento.

2. Estado de la Cuestión

En esta sección, se presenta un análisis de aplicaciones o proyectos que comparten similitudes con el nuestro. Se examinan las características más destacadas de estas soluciones existentes, identificando tanto sus puntos fuertes como las áreas donde se podría mejorar. Este análisis comparativo sirve como base para entender cómo nuestro proyecto se posiciona en el contexto actual de este campo.

3. Descripción del Trabajo

En este apartado, se detalla el proceso de diseño y desarrollo de nuestra aplicación. Se describen las decisiones de diseño, las tecnologías empleadas, y los pasos seguidos desde la concepción de la idea hasta la implementación final. Esta sección ayudará a comprender el enfoque que hemos elegido y los desafíos superados durante el desarrollo.

4. Conclusiones y Trabajo Futuro

Esta sección presenta una valoración final del proyecto, basada en los resultados obtenidos para medir el éxito de la aplicación en función de los objetivos planteados inicialmente.

Se discuten posibles mejoras y futuras líneas de trabajo que podrían derivarse del proyecto actual para expandir o perfeccionar la aplicación, buscando soluciones a nuestras limitaciones y dando opciones a nuevas funcionalidades que han surgido durante el proyecto.

5. Bibliografía

Al final de la memoria, se incluye una lista de todas las fuentes bibliográficas consultadas y citadas a lo largo del documento. Esta sección asegura la correcta atribución de ideas y trabajos previos, que sirve además para proporcionar referencias adicionales para profundizar en los temas tratados.

Capítulo 2

Estado de la Cuestión

2.1. Introducción

En este capítulo, recopilamos y documentamos toda la información relevante obtenida de los documentos estudiados que están relacionados con el tema de nuestro trabajo de fin de grado. Esta sección es fundamental para establecer una base sólida de conocimiento y comprensión del contexto en el que se encuentra nuestra aplicación.

Nos dedicamos a revisar una variedad de fuentes, incluyendo artículos, informes técnicos y otros documentos pertinentes que abordan aspectos específicos de este tema. Cada documento fue analizado para identificar los datos que nos ayudaron a comprender el problema propuesto y así facilitarnos el desarrollo de una buena aplicación.

2.2. Aplicaciones relacionadas

Para encontrar aplicaciones que fuesen parecidas a la que debemos realizar, buscamos en diversas páginas de internet recogiendo muchos nombres de aplicaciones y dispositivos físicos que existen para intentar llevar un seguimiento de la salud mental de los usuarios o proporcionarles ayuda de expertos en el campo mediante chats, rutinas o incluso a veces mediante inteligencia artificial.

Después de leer un poco sobre todas ellas, decidimos que sería buena idea probar alguna de ellas nosotros mismos. En ese momento, nos dimos cuenta de que muchas de ellas solo estaban disponibles en las tiendas de otros países así que probamos las que pudimos descargar en la Play Store española y en la App Store española.

Entre toda la información que recabamos, adjuntamos la que nos ha ayudado a obtener ideas para hacer nuestra aplicación.

- **PACIFICA:** El fin de esta app es controlar la ansiedad, con principios de terapia cognitivo-conductual¹, relajación y bienestar. También permite hacer un seguimiento diario de estado de ánimo y de salud, un historial de estado de ánimo, un análisis de los pensamientos, ejercicios de respiración profunda y de relajación muscular. Además propone actividades que fomentan la tranquilidad mediante sonidos como las olas del mar, música clásica, etc. (Moberg et al., 2019)

Vimos además, un vídeo de algunas pantallas que muestran las opciones proporcionadas por la app que nos inspiró a cómo plantear las pantallas de las que constaría nuestra aplicación.

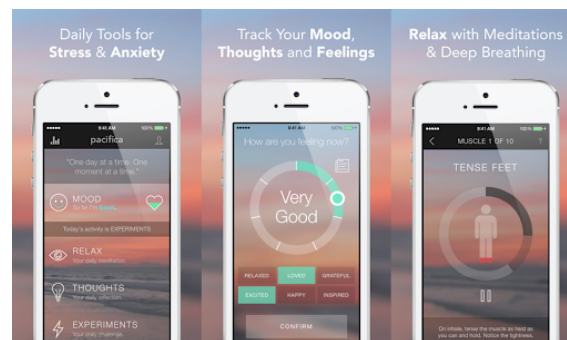


Figura 2.1: Pacifica

- **TALKSPACE:** Esta app está basada en la página web creada en una tesis doctoral de la universidad de Washington (Palaniappan, 2015). Su fin es proporcionar ayuda para controlar la ansiedad. Para ello conecta de forma anónima a los usuarios con terapeutas licenciados para hablar de lo que les preocupa. El programa cuenta con varias opciones de planes de pago, desde la terapia de mensajes ilimitados hasta sesiones de vídeo en directo. Esta aplicación nos ha dado idea de cómo se podría ampliar el TFG, haciendo chats con el profesional que lleva el caso del adolescente o incluso otro terapeuta especializado.

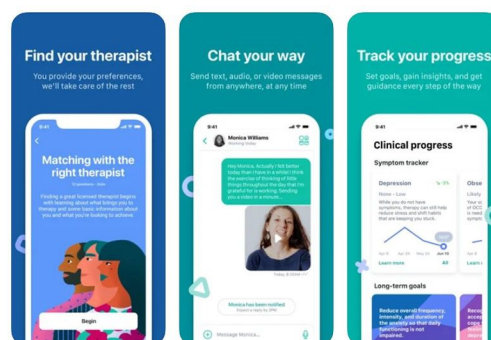


Figura 2.2: TalkSpace

¹La terapia cognitiva-conductual es un conjunto de modalidades de psicoterapia que se dirigen a producir cambios en la percepción o pensamientos que los pacientes tienen acerca de las situaciones de sus vidas, vinculando pensamiento conducta (Wikipedia, 2024b) (Wikipedia, 2024a)

- **MOOD24/7:** Es una plataforma electrónica de monitoreo del estado de ánimo. Aunque esta plataforma está aún en estudio de validación su objetivo es el mismo que el de la aplicación que debemos desarrollar. Este objetivo es rastrear de forma precisa y eficiente el estado de ánimo a través del tiempo. La manera en la que lo hace mood24/7 es a través de textos SMS o correos electrónicos diarios automatizados. En nuestro caso será a través de las mediciones instantáneas realizadas por los tests respondidos por los usuarios en la app. (Kumar et al., 2020)
- **MOODTUNE:** Esta plataforma móvil es también parecida a la nuestra, su objetivo es evaluar, rastrear y prevenir problemas de salud mental mediante pruebas y encuestas de evaluación estandarizadas, seguimientos diarios de estados de ánimo, herramientas de comunicación, una herramienta personalizable de noticias y segmentación para el seguimiento específico. (Moodtune, 2017)



Figura 2.3: MoodTune

Además decidimos descargar las aplicaciones que estuviesen disponibles en España en la PlayStore y en la appStore, para fijarnos bien en la interfaz y funcionalidad de algunas de ellas, interactuando personalmente con ellas:

- **DAILYBEAN:** Es una aplicación de diario y seguimiento personal que permite a los usuarios registrar sus actividades diarias y estados de ánimo de manera muy visual utilizando iconos y colores para representar estados de ánimo y actividades diarias.

Cuenta con herramientas de análisis para identificar patrones y tendencias a través de estadísticas y gráficos. Esta funcionalidad también será implementada en nuestra aplicación.

Los usuarios pueden añadir también notas y fotos a las entradas diarias para que se asemeje realmente a un diario tradicional. Además, la aplicación envía recordatorios y notificaciones para recordar al usuario que debe realizar el registro de ese día.

Nos gustó mucho como herramienta para monitorear la salud mental y emocional porque registra las rutinas diarias, y cómo consigue destacar hábitos positivos. (Lauren Teeter, 2024)

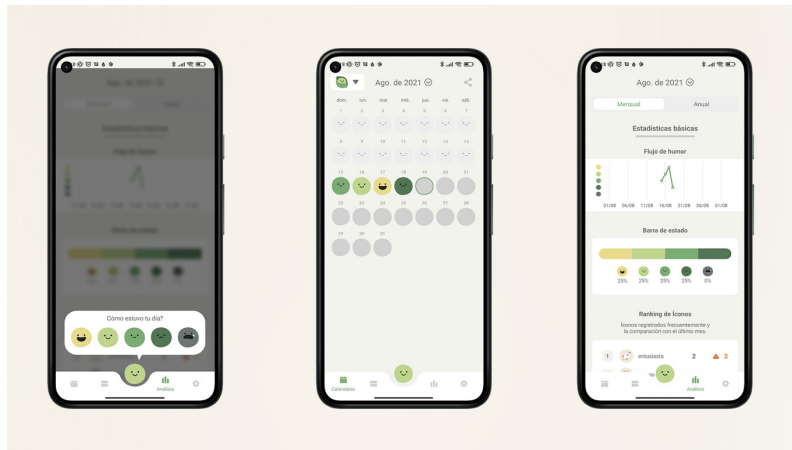


Figura 2.4: DailyBean

- **PETIT BAMBOU:** Es una aplicación de meditación y bienestar que ofrece una variedad de ejercicios y guías para ayudar a los usuarios a reducir el estrés, mejorar la concentración y fomentar el bienestar general. Para ello ofrece sesiones guiadas de meditación mindfulness y técnicas de relajación adaptadas a diferentes niveles de experiencia. Además, como muchas de las apps mencionadas anteriormente, ofrece seguimiento del progreso, estadísticas personalizadas, y recordatorios para fomentar la práctica diaria. (Plenament, 2022)

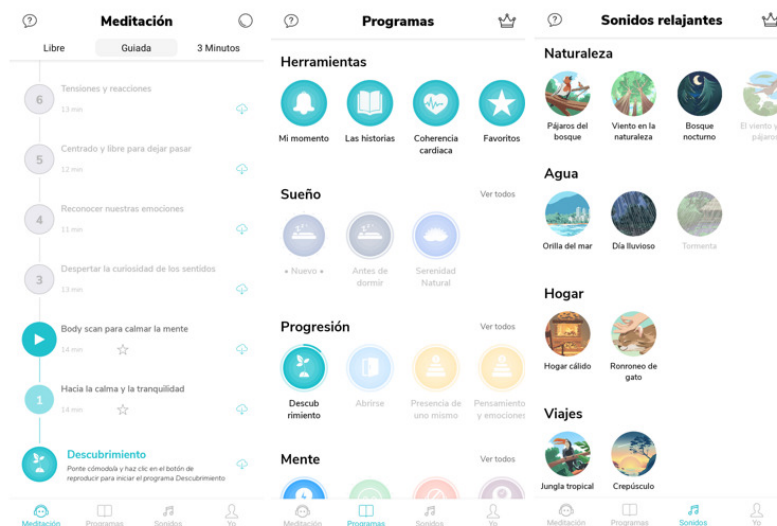


Figura 2.5: Petit Bambou

- **FABULOUS:** Como el resto, es una aplicación de bienestar y formación de hábitos. Tiene como objetivo dar apoyo a los usuarios para alcanzar sus objetivos personales y mejorar su calidad de vida.

Fabulous proporciona programas y desafíos guiados que intentan centrarse en el establecimiento de hábitos positivos, la gestión del estrés, la mejora del sue-

ño, la productividad y el bienestar general. Como casi todas las anteriores apps, utiliza recordatorios, mensajes motivacionales y herramientas de seguimiento del progreso para mantener a los usuarios comprometidos y enfocados en sus objetivos. (Fabulous, 2024)

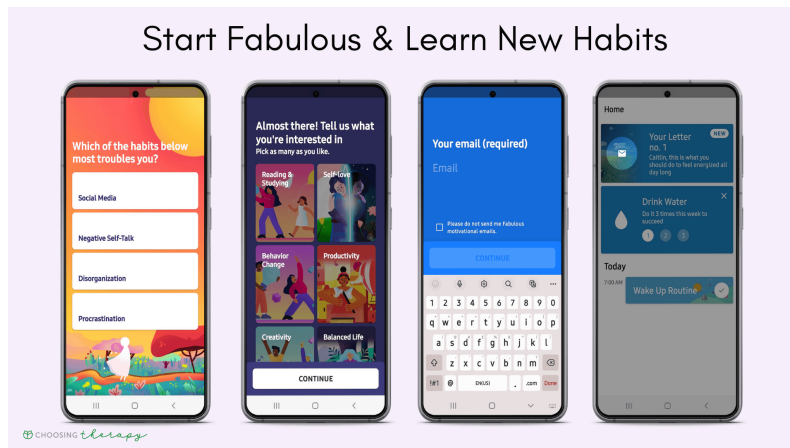


Figura 2.6: Fabulous



Figura 2.7: Fabulous 2

- **DAYLIO DIARIO:** Es una aplicación de seguimiento personal diseñada para registrar y analizar estados de ánimo y actividades diarias sin necesidad de escribir largos textos. Esta característica se asemeja al propósito de los psicólogos, los cuestionarios deben ser pocas respuestas de selección para que los usuarios no la dejen de usar por la no disponibilidad de tiempo.

La interfaz de usuario es extremadamente intuitiva y visual, utilizando iconos y colores para representar diferentes estados de ánimo y actividades, lo que permite un registro rápido y eficiente.

También proporciona estadísticas y gráficos, para visualizar las tendencias y patrones en el estado de ánimo y las actividades a lo largo del tiempo pudiendo así ajustar las rutinas diarias dependiendo de esto. La aplicación permite agregar notas a cada entrada, proporcionando contexto adicional y detalles esto

nos hizo pensar que debíamos añadir un cuadro de texto por si los usuarios quisieran especificar más su situación. (Daylio, 2024)

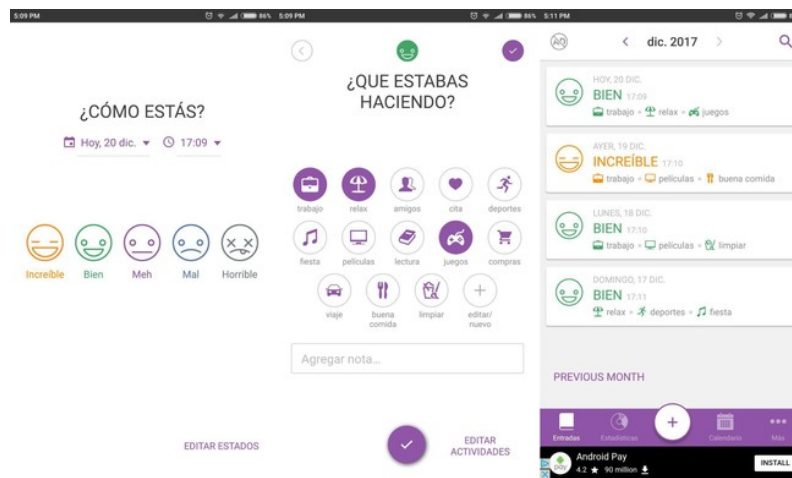


Figura 2.8: DaylioDiario



Figura 2.9: DaylioDiario2

A continuación se muestra una tabla comparativa de las distintas aplicaciones con sus objetivos y características principales

Tabla 2.1: Comparativa de aplicaciones de salud mental

Aplicación	Objetivo principal	Funcionalidades destacadas	Tecnología de monitoreo	Otras características
PACIFICA	Controlar la ansiedad	Terapia cognitivo-conductual, ejercicios de respiración y relajación	Historial de estado de ánimo y análisis de pensamiento	Actividades con sonidos relajantes como técnica de relajación
TALKSPACE	Terapia con terapeutas licenciados	Comunicación anónima con terapeutas	Mensajes y sesiones de video	Integración de chats con profesionales, soporte para terapias online
MOOD24/7	Monitoreo del estado de ánimo	Rastreo de estado de ánimo a través de SMS o correos electrónicos diarios	Mensajes SMS, correos electrónicos automáticos	Seguimiento diario sencillo y eficaz, adaptable a mediciones instantáneas En estudio de validación
MOODTUNE	Evaluar, rastrear y prevenir problemas mentales	Seguimiento diario de estados de ánimo	Pruebas y encuestas estandarizadas	Evaluación de estado de ánimo, pruebas estandarizadas
DAILYBEAN	Registro de actividades diarias y estados de ánimo	Registro visual de estados de ánimo mediante iconos y colores, estadísticas y gráficos	Iconos y colores para representar estados de ánimo, recordatorios	Herramientas de análisis con gráficos y estadísticas
PETIT BAMBOU	Meditación y bienestar	Sesiones guiadas de meditación y mindfulness	Seguimiento del progreso personalizado	Estadísticas, recordatorios para la práctica diaria
FABULOUS	Bienestar y formación de hábitos positivos	Programas guiados para la formación de hábitos	Herramientas de seguimiento del progreso	Mensajes motivacionales

Aplicación	Objetivo principal	Funcionalidades destacadas	Tecnología de monitoreo	Otras características
DAYLIO DIARIO	Registro de estado de ánimo sin escribir	Registro de estado de ánimo y actividades mediante iconos, estadísticas y gráficos	Registro rápido, visual y eficiente del estado de ánimo	Estadísticas y gráficos para visualizar tendencias

Tras analizar las aplicaciones mencionadas, hemos identificado una serie de características que serían de interés para nuestra aplicación, estas nos han servido de gran ayuda para mejorar la funcionalidad de nuestra aplicación, y proporcionar una experiencia más atractiva y efectiva para los usuarios.:

1. **Monitoreo diario de estado de ánimo:** Al igual que en MOOD24/7 y DAILYBEAN, es imprescindible implementar un sistema que permita a los usuarios registrar su estado de ánimo de manera rápida y sencilla para cumplir con el requisito impuesto, las mediciones instantáneas. Esto será llevado a cabo con los tests, con preguntas de respuesta de selección.
2. **Uso de iconos y colores:** Aplicaciones como DAILYBEAN y DAYLIO destacan por sus interfaces visuales intuitivas, donde los usuarios pueden registrar su estado de ánimo utilizando iconos y colores. Queremos tomar este enfoque visual porque es atractivo y facilita la interacción, especialmente para adolescentes que buscan rapidez y simplicidad.
3. **Recordatorios y notificaciones:** Muchas de las aplicaciones analizadas (DAILYBEAN, FABULOUS, entre otras) utilizan recordatorios para fomentar la constancia en el uso de la aplicación. Implementaremos nosotros también notificaciones automáticas para recordar a los usuarios que completen sus tests diarios para garantizar la participación activa y el monitoreo constante.
4. **Análisis de patrones y estadísticas:** Herramientas como las de DAILYBEAN y DAYLIO permiten a los usuarios visualizar patrones y tendencias a lo largo del tiempo mediante gráficos y estadísticas. Esta funcionalidad será útil para que tanto los usuarios como los psicólogos puedan identificar posibles tendencias en el estado de ánimo de los adolescentes.
5. **Soporte para comunicación con profesionales:** A diferencia de TALKSPACE La comunicación en nuestra app solo será posible en caso de emergencia, el terapeuta podrá llamar al adolescente o a su tutor en caso de la detección de alto riesgo de auto-lesión o suicidio. Pero inspirados en la misma aplicación, un futuro desarrollo para nuestra aplicación podría incluir la posibilidad de que los adolescentes se comuniquen directamente con psicólogos o terapeutas a través de un chat, lo que mejoraría la intervención temprana y el seguimiento de casos.

Descripción del Trabajo

3.1. Introducción

En este apartado, se especificará el trabajo realizado, que abarca el desarrollo e implementación de un servidor, la creación y gestión de una base de datos y la creación y el diseño de la aplicación Sivaria. Este trabajo es una oportunidad para aplicar los conceptos teóricos aprendidos durante nuestra formación académica en un entorno práctico y real. A través de este proceso, hemos enfrentado desafíos significativos que han ampliado nuestra comprensión de diversos aspectos de la ingeniería de software, desde la arquitectura de sistemas hasta la seguridad de datos. Nuestro objetivo es ofrecer una visión detallada de las etapas, decisiones y soluciones encontradas durante el desarrollo de este proyecto.

3.2. Arquitectura de la aplicación

La arquitectura de la aplicación es un aspecto crucial para comprender cómo interactúan los componentes y cuál es su funcionamiento general. A continuación, se presentan dos diagramas que describen las dos posibles arquitecturas para nuestra aplicación: una utilizando nuestro servidor y base de datos, y otra conectando la aplicación a REDCap. REDCap (Research Electronic Data Capture) es una plataforma segura de software diseñada para la recolección y gestión de datos en investigaciones clínicas y estudios científicos. Se utiliza principalmente para crear bases de datos y formularios de captura de datos en línea, permitiendo a los investigadores recoger, organizar y exportar información de manera eficiente. (Harris et al., 2019) El equipo técnico que trabaja con los psicólogos de la UNED en este proyecto, nos comentaron en una reunión que deberíamos conectarnos desde nuestro servidor a REDCap, para enviarles todas las respuestas de los formularios y que ellos, con esa información se encargarían de alimentar el Sistema Experto y una vez alimentado, se encargarían de mandarnos los datos necesarios, para eso les proporcionamos las rutas con las que nos enviarían los datos al servidor. Asegurando así una buena alineación entre

nuestra implementación y la suya.

3.2.1. Descripción de la arquitectura

Nuestra aplicación puede ser organizada de dos maneras principales, cada una adaptada a diferentes necesidades y flujos de trabajo:

1. Arquitectura con servidor y Base de Datos Propios:

- **Frontend (React Native):** La interfaz de usuario, desarrollada en React Native, se ejecuta en dispositivos móviles con sistemas operativos iOS y Android. Esta interfaz proporciona una experiencia de usuario fluida y consistente. La aplicación ofrece una serie de funcionalidades que incluyen la visualización de datos o la realización de cuestionarios.
- **Backend (Servidor Flask):** El backend, desarrollado utilizando el framework Flask en Python, expone una API RESTful que facilita la comunicación con el frontend mediante solicitudes HTTP/HTTPS. Este backend es responsable de gestionar la lógica de nuestro código, procesar las solicitudes del frontend, y manejar la autenticación y autorización de usuarios. Además, se encarga de realizar las operaciones de procesamiento de datos, como el manejo de las respuestas de los cuestionarios o el inicio de sesión .
- **Base de Datos (MySQL):** Los datos de la aplicación, que incluyen información de usuarios o registros de las respuestas de los cuestionarios, se almacenan en una base de datos MySQL relacional. Esta base de datos permite realizar consultas complejas y mantener la integridad de los datos. El backend realiza operaciones de lectura y escritura en la base de datos, asegurando que la información esté actualizada y disponible para el frontend.

2. Arquitectura con Integración a REDCap:

- **Frontend (React Native):** La interfaz de usuario sigue siendo la misma, permitiendo a los usuarios interactuar con la aplicación desde dispositivos móviles. La experiencia de usuario se mantiene consistente, proporcionando acceso a todas las funcionalidades necesarias para los usuarios.
- **Backend (API de REDCap):** En esta arquitectura, el backend se conecta directamente a REDCap a través de su API para gestionar datos. La API de REDCap proporciona endpoints específicos para la recuperación, envío y actualización de datos, lo que permite al backend interactuar de manera eficiente con la plataforma. La comunicación entre el frontend y REDCap se realiza mediante solicitudes HTTP/HTTPS, facilitando la integración y asegurando que los datos sean gestionados de manera centralizada en REDCap.

- **Base de Datos (REDCap):** En lugar de utilizar una base de datos MySQL propia, los datos se almacenan y gestionan directamente en REDCap. REDCap actúa como el repositorio central de datos, donde los psicólogos tienen toda la información necesaria para sus actividades. Esta integración permite una sincronización directa con los sistemas y flujos de trabajo existentes de los psicólogos, simplificando la gestión de datos y asegurando que toda la información relevante esté centralizada en una sola plataforma.

3.2.2. Diagrama de Arquitectura

A continuación se muestran un diagrama de la arquitectura de la aplicación: el de abajo con la infraestructura de servidor y base de datos propios, y el de arriba con la integración a REDCap.

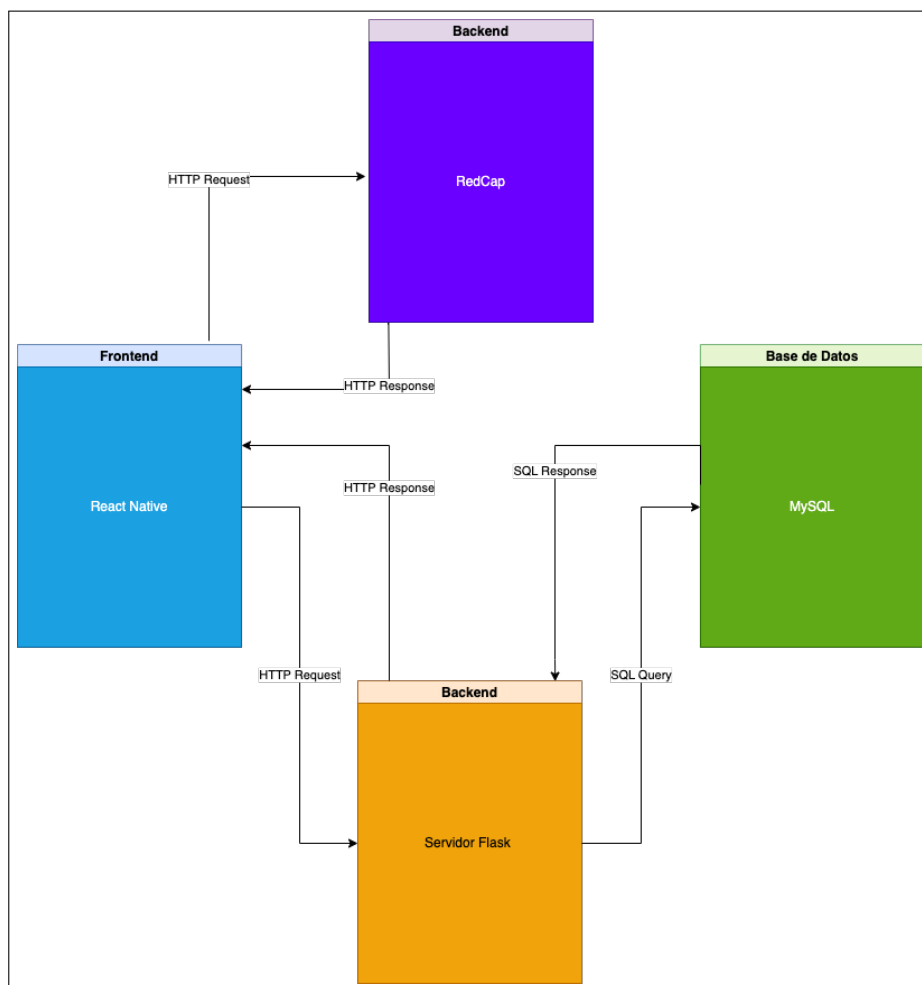


Figura 3.1: Diagrama de Arquitectura Sivaría

3.2.3. Explicación de los Diagramas

En el primer diagrama (con servidor y base de datos propios), el **frontend** se comunica con el **backend** a través de solicitudes HTTP/HTTPS. El **backend** (servidor Flask) procesa estas solicitudes, interactúa con la **base de datos MySQL** para leer o escribir información, y envía respuestas al frontend.

En el segundo diagrama (con integración a REDCap), el **frontend** sigue enviando solicitudes al **backend**. Sin embargo, el **backend** en esta arquitectura se conecta directamente a **REDCap** a través de su API para gestionar la información, en lugar de usar una base de datos MySQL propia. Los datos se almacenan y actualizan en **REDCap**, y el backend se encarga de enviar y recibir información entre el frontend y REDCap.

3.2.4. Conclusión

Ambas arquitecturas tienen sus ventajas. La arquitectura con servidor y base de datos propios proporciona un control completo sobre el entorno de la aplicación, mientras que la integración con REDCap puede simplificar la gestión de datos y aprovechar la infraestructura existente de REDCap.

3.3. Diseño centrado en el usuario de la aplicación

El desarrollo centrado en el usuario (DCU) es fundamental para la aplicación que estamos construyendo, dado que se trata de una herramienta que busca monitorear y gestionar la salud mental de adolescentes. Al ser un público sensible y de características específicas, como la vulnerabilidad emocional y la necesidad de un acceso rápido y efectivo a la ayuda, es vital que la experiencia del usuario sea intuitiva, accesible y, sobre todo, adecuada a sus necesidades. Ya que si es difícil de usar o desconectada de sus necesidades, es probable que la abandonen rápidamente.

Este tipo de desarrollo se centra en los siguientes pasos:

1. Investigación del usuario:

En la investigación del usuario se analizan las características de los usuarios, es decir, se identifica a las personas que usarán el producto, y su contexto de uso, es decir, para qué lo usarán y las condiciones en las que lo usarán.

En nuestro caso, este paso ha sido realizado en la búsqueda del contexto en el que se encuentra este proyecto en la actualidad. Además, en este punto, fue crucial tener varias reuniones con los psicólogos para que nos contasen en qué contexto se usaría la aplicación, y que además nos acercasen un poco más al pensamiento, sentimientos y rutinas de los adolescentes.

Así pudimos definir claramente las necesidades y expectativas del usuario, de modo que la aplicación responda a ellas eficazmente.

2. Definición de requisitos:

Tras estudiar a los usuarios, determinamos una lista de requisitos, y se plantearon los perfiles de usuario o personas, representaciones ficticias y detalladas de los tipos de usuarios que usarán la aplicación.

A partir de estas personas se presentaron soluciones a los requisitos e investigamos cuáles eran cruciales para la aplicación y cuáles eran técnicamente factibles.

3. Diseño de la interfaz centrado en la usabilidad:

En esta etapa, se transforma la aplicación de una idea conceptual a un diseño real. Basándonos en lo aprendido y en las ideas generadas, se diseñaron las interfaces y los flujos de la aplicación para que fuesen intuitivas y fáciles de usar. Esto incluye la creación de wireframes¹, y mapas de navegación.

El objetivo principal es diseñar una experiencia de usuario que sea sencilla, rápida y clara. Estos requisitos son fundamentales para que los usuarios puedan, por ejemplo, completar cuestionarios de forma eficaz y sin frustraciones, ya que el tiempo y la simplicidad son factores clave en la experiencia de uso.

4. Pruebas con usuarios:

En este paso del DCU, usuarios reales prueban los prototipos para que podamos observar cómo interactúan con el diseño para identificar problemas en la interacción, lo que permite validar que el diseño es adecuado y si fuese necesario realizar ajustes antes de pasar a la etapa de desarrollo. En nuestro caso solo le enseñamos el prototipo a nuestro tutor y a los expertos y nos dieron su opinión de la interacción ya que no contábamos con otro tipo de usuario.

5. Iteración y mejora continua:

En esta fase se analizan los resultados de las pruebas con los usuarios y se ajusta el diseño en función de los comentarios y observaciones. Este proceso se repite tantas veces como sea necesario para mejorar la aplicación. Se hacen cambios y se vuelve a probar.

6. Lanzamiento controlado y monitoreo:

Una vez que la aplicación está lista, se hace un lanzamiento controlado (beta) a un grupo pequeño de usuarios para verificar cómo se comporta en un entorno real. Durante este periodo, se monitorean los datos de uso y la retroalimentación de los usuarios para detectar posibles problemas que no se identificaron en las pruebas anteriores. Este paso estaba previsto como parte del TFG pero desgraciadamente, no se pudo llegar a realizar.

¹En diseño web, un wireframe es una representación esquemática de una página web sin elementos gráficos que muestran contenido y comportamiento de las páginas (Wikipedia, 2022)

3.3.1. Investigación del usuario

La investigación sobre el contexto en el que se posiciona la app SIVARIA no solo terminó en nuestra fase de investigación, si no que tuvimos que ponernos al día con cómo de avanzado iba el proyecto. Antes de reunirnos con ellos, nos proporcionaron varios documentos de investigación que ya habían realizado para crear la idea del proyecto SIVARIA en el que nos explicaron en qué consistía el proyecto y todas los términos técnicos que debíamos conocer, para ello, nos proporcionaron documentos en los que lo explicaban. Después de hacer la lectura de los mismos, nos reunimos varias veces con ellos para resolver varias dudas que nos surgieron, sobretodo sobre el “Sistema Experto”, que en un principio no supimos si teníamos que implementar o no. Cuando supimos que ese sistema ya estaba en funcionamiento con pruebas realizadas a mano, indagamos sobre cómo debíamos usarlo y cómo él nos daría la información precisa que debíamos mostrar, como por ejemplo, las preguntas. Después de todo este proceso, podemos destacar la siguiente información recolectada de dichos documentos y reuniones. En primer lugar, que el proyecto SIVARIA se enmarcaba de la convocatoria de proyectos I+D+i 2020 del Ministerio de Ciencia e Innovación del Gobierno de España y que en él participaban muchas otras universidades y servicios de salud. También nos presentaron el objetivo del proyecto, crear una herramienta que mediante un Sistema Experto, fuese capaz de valorar de manera protocolizada y personal el riesgo de conductas autolesivas (suicidas y no suicidas) entre adolescentes. Y que a su vez, permitirá una monitorización, seguimiento y devolución de prácticas protectoras y amortiguadoras del riesgo detectado. Lo que nos introdujeron como Ecological Momentary Assessment. Para poder hacer dicha valoración el sistema experto debía ser alimentado por distintas fuentes informantes, con las que más tarde crearíamos nuestros tipos de persona. Después de ser alimentado y haber procesado la información, el sistema proporcionaría como salida el resultado del riesgo de las conductas autolesivas como falsos o verdaderos positivos y falsos o verdaderos negativos tal y como mostramos en la imagen

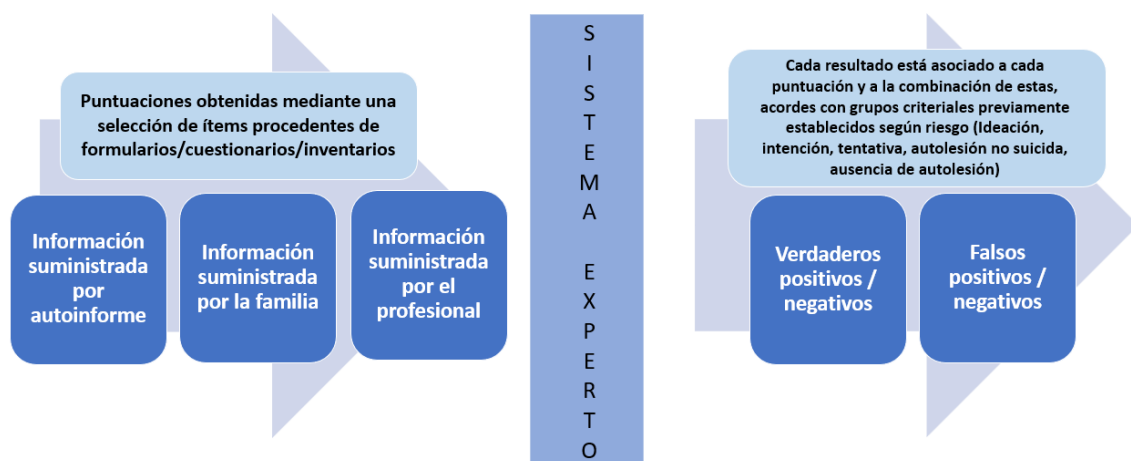


Figura 3.2: Entrada y salida del sistema experto

Cada una de las fuentes informantes proporcionaría la información de diferen-

tes variables, a continuación se especifica qué variables corresponde a cada fuente informante.

Tabla 3.1: Variables del auto-informe

VARIABLES PROPORCIONADAS POR EL AUTOINFORME	
VARIABLES DE IDENTIFICACIÓN	Curso
	Código postal
	Siglas
VARIABLES SOCIODEMOGRÁFICAS	Edad
	Sexo asignado
	Transgénero
	Situación laboral materna/paterna
	Nivel profesional madre/padre
	Nivel promedio del rendimiento académico
	Nivel de autopercepción masculinidad/feminidad
VARIABLES DE SALUD	Nivel de heteropercepción masculinidad/feminidad
	Peso
	Altura
	Tratamiento psiquiátrico previo
VARIABLES DE CONSTRUCTO DEL COMPORTAMIENTO SUICIDA	Presencia de enfermedad crónica
	Percepción de discriminación
	Fuente de discriminación
	Motivo de discriminación
	Nivel de satisfacción vital
	Nivel de desesperanza
PRESENCIA DE BULLYING	Victimización (traumas vividos/presenciados)
	Bullying perpetrador
	Bullying víctima
PRESENCIA DE CYBERBULLYING	Bullying perpetrador-víctima
	Cyberbullying perpetrador
	Cyberbullying víctima
ADICCIÓN/ABUSO	Cyberbullying perpetrador-víctima
	Alcohol
	Sustancias
TECNOLOGÍAS Y RRSS	Internet
	Contacto info autolesión
	Búsqueda de info autolesión
	Compartir en RRSS pensamientos autolesivos
	Motivo compartir en RRSS autolesión
	Vía de contacto para RRSS
	Contagio
Petición de ayuda en internet	

VARIABLES PROPORCIONADAS POR EL AUTOINFORME	
	Tipo de ayuda en internet
	Conocidos que comparten autolesión en internet
	Denuncia de autolesión en internet
Volitivo-motivacionales para el suicidio	Atrapamiento interno
	Atrapamiento externo
	Derrota/fracaso
	Sentido de pertenencia frustrada
	Percepción de ser una carga
	Autoeficacia para el suicidio
	Razones para vivir
Regulación y resistencia	Resiliencia
	Regulación positiva
	Regulación negativa
Autolesivos suicidas	Comunicación
	Deseo
	Ideación
	Planificación
	Intención
	Motivos del comportamiento suicida
	Finalidad
	Antecedentes familiares
Contagio	
Autolesivos no suicidas	Presencia de autolesión
	Duración
	Dolor percibido
	Consumo de sustancias
	Interferencia social
	Razones de autolesión

Tabla 3.2: Variables de las familias

VARIABLES PROPORCIONADAS POR LAS FAMILIAS	
Psicopatología	Problemas interiorizados
	Problemas exteriorizados/función ejecutivas
	Problemas contexto
	Problemas recursos psicológicos
Perfil de riesgo psicosocial	Madre/padre adolescente
	Padres divorciados
	Familia monoparental
	Familia reconstruida
	Psicopatología padre/madre
	Adicción padre/madre
	Antecedentes penales padre/madre
Relaciones conflictivas hijos-padre/madre	

VARIABLES PROPORCIONADAS POR LAS FAMILIAS	
	Relaciones conflictivas padre-madre
Control parental	
Aceptación-rechazo parental	

Tabla 3.3: Variables de las profesionales

VARIABLES PROPORCIONADAS POR LOS PROFESIONALES
Motivo consulta
Tratamiento farmacológico
Situación económica familiar
Estudios padre/madre
Profesión padre/madre
Ingreso mensual
Perfil de riesgo psicosocial
Insuficiente supervisión parental
Presencia de maltrato
Duelo
Enfermedad crónica
Impresión diagnóstica

Cada una de las anteriores tablas, podría tener nuevas variables en cualquier momento según nos señalaron los psicólogos, además de cada variable, nos dijeron que habría diferentes items, que aún no estaban definidos, pero nos proporcionaron un ejemplo. En el tipo de variables de los comportamientos autolesivos suicidas existe la variable IDEACIÓN, y de esta variable saldrían los siguientes items:

1. Edad inicio
2. Frecuencia
3. Duración
4. Intensidad
5. Método
6. Razones
7. Consumo
8. Contagio
9. Expectativa futura de ideación
10. Grado de control
11. Grado de impedimento

Habiendo entendido todos estos conceptos, nos indicaron que las preguntas que se realizarían en los tests correspondería cada una, únicamente a un ítem de una variable determinada.

Una vez alimentado al Sistema experto con las respuestas a dichas preguntas, y dependiendo de los resultados obtenidos por la evaluación, se podrían dar los siguientes escenarios:

1. Alerta a Servicios de emergencias o atención en crisis
2. Alerta a familiares o responsables legales
3. Establecimiento de Plan de seguridad y medidas de protección
4. Provisión de material de auto-ayuda y guías para familiares y profesionales
5. Recomendación de intervención profesional
6. Recomendación de intervención en procesos clave (por ejemplo: desesperanza, razones para vivir, acceso a métodos autolesivos, etc..) que guíen el diseño de una intervención
7. Recomendación de monitorización y seguimiento de la dinámica autolesiva

3.3.2. Definición de requisitos

Después de haber recabado toda la información anterior, hemos decidido, consensuadamente con los psicólogos de la UNED, que la aplicación que vamos a desarrollar va a ser utilizada únicamente por tres tipos de personas. Estos tipos serán: adolescentes (15-21 años), los padres de dichos adolescentes y los profesionales de la salud en este campo. Para dicho diseño, vamos a realizar breves historias de usuario de cada tipo de persona, tal y como aprendimos en la asignatura de desarrollo de sistemas interactivos, siendo estos Jaime (adolescente), Javier (padre) y Sol (profesional).

En este apartado, gracias a la descripción que nos dieron los psicólogos en las reuniones, exploraremos en detalle los diversos perfiles de usuarios que podrían beneficiarse del uso de nuestra aplicación describiendo las funcionalidades específicas que cada uno podrá utilizar dentro de la aplicación, así como sus preferencias y necesidades particulares al interactuar con esta herramienta. Todas las imágenes mostradas son de personas que no existen en la realidad. Estas imágenes han sido generadas por la inteligencia artificial “This Person Does Not Exist” (TPDNE), una aplicación web que utiliza redes neuronales generativas, específicamente una técnica llamada Generative Adversarial Networks (GANs), para crear retratos realistas de rostros humanos ficticios. Cada vez que se recarga la página, la IA genera una nueva imagen de una persona imaginaria que parece auténtica, aunque no existe en realidad. Seleccionamos una imagen que se ajustara al perfil de persona que teníamos en mente.(de Zeeuw et al., 2023) (Xataka, 2017).

3.3.2.1. Historia de usuario

1. Jaime Iglesias Calatayud



Figura 3.3: Jaime

Jaime es un joven de 18 años que actualmente estudia derecho en la Universidad de Cataluña. Desde una edad temprana, Jaime fue presionado por su padre, un abogado de renombre, para seguir sus pasos y estudiar su misma carrera. Inicialmente, Jaime se sintió emocionado con la idea, imaginando un futuro lleno de desafíos interesantes y un trabajo apasionante junto a su padre. Sin embargo, después de los exámenes del primer cuatrimestre, su entusiasmo se desvaneció rápidamente. Suspendió cuatro de las cinco asignaturas, lo que fue un golpe devastador para su autoestima.

La carga académica comenzó a abrumarlo, generándole un estrés cada vez mayor. Jaime se sentía incompetente y, comparándose con sus compañeros, se consideraba inferior. Esta sensación de fracaso constante lo llevó a cuestionarse el propósito de su vida y, en su desesperación, comenzó a tener conductas autolesivas. Durante este período, Jaime hubiera agradecido enormemente contar con apoyo psicológico, pero su situación familiar y social complicó las cosas. Apenas veía a sus padres, y comenzó a alejarse de sus antiguos amigos porque sentía que cualquier tiempo dedicado a ellos era tiempo que no empleaba para estudiar.

Jaime pasa la mayor parte de sus días en la biblioteca, estudiando incansa-

blemente. Criado en las afueras de Extremadura, se mudó a Cataluña para estudiar, pero se enfrenta una barrera adicional: no domina el catalán. Esta barrera lingüística ha dificultado su capacidad para hacer amigos y establecer una red de apoyo dentro de la universidad. A menudo se siente solo y aislado, y a pesar de su deseo de que alguien le pregunte cómo se siente o cómo le van los estudios, nadie parece notarlo.

En cuanto a sus hábitos de comunicación, Jaime no es muy aficionado a utilizar el teléfono. Solo lo utiliza cuando recibe llamadas o notificaciones de WhatsApp. Su estilo de comunicación es breve y conciso, prefiriendo utilizar emoticonos para responder mensajes, ya que se considera un hombre de pocas palabras. Esta forma de comunicarse refleja su introversión y su dificultad para expresar sus sentimientos de manera verbal.

2. Javier Iglesias Galdo



Figura 3.4: Javier

Javier es un padre de familia de 58 años que ha trabajado como abogado durante más de 25 años. Su dedicación al trabajo ha sido inquebrantable, dedicando innumerables horas a estudiar cada caso al detalle para mantener su reputación impecable. Esta dedicación, sin embargo, ha tenido un costo significativo en su vida personal, especialmente en la relación con su hijo, Jaime. Javier se ha dado cuenta de que, a pesar de su éxito profesional, ha pasado poco tiempo con su familia y no conoce bien a su hijo.

Hace unos meses, Javier se enteró de que Jaime estaba lidiando con un estrés severo que lo llevó a auto-lesionarse. Este descubrimiento sacudió profundamente a Javier, haciéndolo reflexionar sobre sus prioridades. Decidió reducir su carga de trabajo, delegando más tareas a los pupilos que ha formado a lo largo de los años, con el objetivo de pasar más tiempo con su hijo y estar más presente en su vida. Javier lamenta no haber estado más atento y no haber demostrado a Jaime cuánto lo quiere. Ahora, está decidido a enmendar esta situación.

Para ello, Javier busca una manera de monitorear de manera precisa la salud mental de su hijo, especialmente en situaciones críticas. Quiere ser alertado cuando Jaime esté emocionalmente inestable, para poder intervenir y brindarle el apoyo que necesita. Sin embargo, Javier se siente inseguro cuando se trata de temas de psicología. Su formación y experiencia son en leyes, no en el manejo de problemas emocionales o mentales.

Afortunadamente, Javier cuenta con el apoyo de Sol, la psicóloga de Jaime. Sol le ha ayudado a identificar y clasificar los problemas emocionales de Jaime, preguntándole semanalmente sobre posibles situaciones problemáticas. Esto ha sido invaluable para Javier, quien agradece poder entender mejor los desafíos que enfrenta su hijo. Sol también mantiene a Javier informado sobre la evolución de Jaime, dentro de los límites de la confidencialidad profesional, lo cual le da una mayor sensación de control y tranquilidad.

En cuanto a sus preferencias, Javier es un hombre práctico y directo. No le gusta perder tiempo escribiendo largos textos para expresarse; prefiere hacer una llamada rápida para comunicar sus pensamientos. Las encuestas extensas y los formularios que requieren respuestas detalladas le resultan tediosos y frustrantes ya que está saturado de dichos formularios y largos textos que usa en el día a día debido a su profesión. En cambio, Javier prefiere interfaces que le permitan seleccionar opciones o utilizar barras de estado para indicar su opinión o el estado emocional de su hijo. Aunque a veces escribe breves descripciones de situaciones específicas, generalmente lo hace con la sensación de no estar respondiendo adecuadamente debido a su falta de conocimientos en psicología.

Javier es un perfeccionista y se siente incómodo cuando debe responder preguntas en áreas donde no se siente competente. Por eso, valora enormemente cualquier herramienta o entorno que simplifique este proceso y le permita expresar sus preocupaciones de manera rápida y eficiente. En ocasiones, escribe brevemente sobre la situación de Jaime, pero siempre con la incertidumbre de si está identificando correctamente los problemas.

3. Sol Nikolaou García



Figura 3.5: Sol

Sol es una psicóloga especializada en trastornos depresivos y conductas autolesivas en adolescentes. Su compromiso con sus pacientes es profundo, ya que ha experimentado estas conductas de cerca en su propia familia. Ha tratado a muchos jóvenes con comportamientos autolesivos y considera que una aplicación para hacer seguimiento de estos casos sería de gran ayuda para controlarlos y asistirlos adecuadamente en caso de recaída.

Cada vez que Sol atiende a un paciente, le gusta guardar un informe del progreso. Este informe incluye las mismas preguntas con respuestas de opción múltiple, lo que le permite generar gráficos visuales del progreso del paciente. Esta sistematización facilita un balance claro sobre si el estado del paciente ha mejorado o empeorado.

Sol tiene una fuerte empatía y desea ayudar a todos los adolescentes posibles, asegurándose de que ninguno esté sufriendo. Maneja un gran número de pacientes, lo que implica memorizar una cantidad significativa de información, expedientes y progresos. La aplicación que imagina almacenaría toda la información de cada paciente de manera separada y le enviaría alertas mediante mensajes o llamadas telefónicas si detecta algún problema, permitiéndole intervenir rápidamente.

Además, Sol ha notado que la recogida de datos en la consulta no siempre es efectiva, ya que los pacientes a menudo informan que las situaciones críticas

ocurren durante la semana, cuando ella no está presente y no puede enterarse. Con esta aplicación, Sol podría recibir actualizaciones continuas y precisas sobre el estado de sus pacientes, mejorando significativamente la calidad de su atención y apoyo.

3.3.2.2. Escenarios de contexto

1. Jaime Iglesias Calatayud

Jaime responde un cuestionario

Jaime se encuentra en clase de Derecho Romano, de pronto le suena una notificación en el móvil. Está perdonado ya que el profesor sabe qué significa. Es hora de responder un cuestionario de cómo se siente, tiene una autorización de la psicóloga que afirma que es necesario por su salud mental.

Jaime, se dispone a responder el cuestionario muy animado, hoy ha pasado un buen día y está contento. Como no tiene mucho tiempo, ya que está en clase, agradece que las preguntas sean muy cortas y simples, y que se respondan con una simple escala. Terminado el cuestionario, guarda el teléfono y sigue intentando aprender las leyes que seguían los romanos en el siglo VIII a.C.

Jaime se encuentra mal y decide notificarlo

Hoy Jaime no ha tenido un buen día, ha estado estudiando durante toda la tarde y siente que no ha aprendido nada. Además, ha discutido con su padre, otra vez el trabajo. Su padre se esfuerza por mejorar la relación pero a veces no se entienden. Jaime estaba mejorando, pero hoy, después de la pelea, ha tenido pensamientos autolesivos de nuevo. Sabe que esto no es bueno para él y para su progreso. Por lo que decide anotarlo cuanto antes en la aplicación, para que su psicóloga esté al tanto y no empeorar.

2. Javier Iglesias Galdo

Javier responde un cuestionario

Javier se encuentra en una reunión del bufete y le ha sonado una notificación de la aplicación que considera más importante. La del seguimiento de la salud mental de su hijo. Decide disculparse y responder el cuestionario, le hacen preguntas sobre cómo se ha sentido Jaime en la semana. Como Javier y su hijo han tenido problemas esta semana y sabe que a Jaime le afectan mucho, decide, además de la encuesta, escribir lo sucedido.

Javier avisa de una emergencia

Javier ha estado notando algunos indicios de que Jaime está planeando suicidarse, ha visto que sus últimas búsquedas de Google han sido relacionadas con el tema. Además he recibido una llamada del único amigo que tiene en la universidad, le ha visto muy decaído y raro, además dice creer haber visto marcas de lesiones en su muñeca, está muy preocupado. Esta preocupación ha sido contagiada a Javier y ha decidido avisar a Sol de esta emergencia a través de la aplicación que comparten, mediante un mensaje de texto.

3. Sol Nikolaou García

Sol mira los progresos de Jaime

Sol tiene consulta con Jaime dentro de poco, por lo tanto decide revisar los progresos que ha tenido en la semana. Abre la aplicación y observa los gráficos que esta le ofrece. Jaime ha estado progresando pero sabe hacia dónde orientar la consulta del próximo día para mejorar aún más la situación de Jaime.

3.3.3. Diseño de la interfaz centrado en la usabilidad

En este apartado, presentamos todos los diseños correspondientes al prototipo de nuestra aplicación móvil. Cada diseño ha sido creado en Figma, con el objetivo de optimizar la funcionalidad, mejorar la usabilidad, y asegurar que la aplicación cumpla con las expectativas y necesidades de nuestros usuarios.

Estos prototipos que mostramos a continuación nos han servido para tener presente en todo momento hacia donde va dirigida nuestra aplicación y recibir el feedback de los psicólogos desde el primer momento para ir corrigiendo los errores hasta llegar a la implementación final. Esperando que dicha implementación sea intuitiva y eficaz.

1. Pantalla Iniciar Sesión

La pantalla de inicio de sesión está diseñada para ofrecer una experiencia simple y directa. En la parte superior, se mostrará el logo de nuestra aplicación, lo que permite al usuario familiarizarse inmediatamente con nuestra identidad visual. Justo debajo del logo, y después del icono de “Iniciar Sesión”, se encuentra un campo de texto donde los usuarios pueden ingresar su nombre de usuario o correo electrónico. Este campo incluye un texto de marcador de posición que dice “Usuario”, lo que facilita la identificación del propósito de este campo.

Debajo del campo de usuario, se encuentra el campo de contraseña, que está protegido y oculta los caracteres ingresados mediante puntos o asteriscos, garantizando así la privacidad del usuario. A la derecha de este campo, un icono de ojo permite alternar entre mostrar u ocultar la contraseña, por si los usuarios quisieran ver si está bien escrita.

Debajo de los campos de entrada, se encuentra un botón negro con la etiqueta “Iniciar Sesión”, diseñado para destacar sobre el fondo y atraer la atención del usuario. Al hacer clic en este botón, la aplicación valida las credenciales ingresadas, permitiendo el acceso si son correctas y volviendo a solicitarlas si fueron incorrectas. Justo encima de este botón, hay un enlace pequeño con el texto “¿Olvidaste tu contraseña?” que ofrecerá a los usuarios la opción de recuperar su contraseña en caso de que la hayan olvidado. Esta función deberá ser aprobada por los solicitantes ya que ni fue solicitada en un principio.

A continuación, se presenta el diseño prototípico de esta pantalla.



Figura 3.6: Pantalla Iniciar Sesión

2. Pantalla principal del Adolescente

La pantalla de inicio para adolescentes en nuestra aplicación ha sido cuidadosamente diseñada para ser atractiva, amigable y fácil de usar, teniendo en cuenta las necesidades y preferencias de los usuarios jóvenes. En la parte superior de la pantalla, se muestra un mensaje de bienvenida personalizado que busca establecer una conexión inmediata con el adolescente. Este mensaje puede incluir un saludo cálido o una frase motivadora, fomentando un ambiente de confianza y apoyo desde el primer momento.

Justo debajo del mensaje de bienvenida, se encuentran tres botones grandes y coloridos, organizados de manera que cada uno sea fácilmente accesible y distinguible. Estos botones no solo son visualmente atractivos, sino que también están diseñados para guiar al usuario hacia las funciones clave de la aplicación, asegurando que puedan navegar sin complicaciones.

El primer botón, etiquetado como “Responder cuestionario”, lleva al usuario a una pantalla donde puede realizar un test o cuestionario.

El segundo botón, “Enviar respuestas voluntarias”, permite al adolescente acceder a una pantalla donde puede escribir y enviar un mensaje de texto.

El tercer botón, “Ver próximas citas”, dirige al usuario a una pantalla donde puede visualizar y gestionar las citas programadas con su profesional.

Cada botón y funcionalidad en esta pantalla ha sido diseñado teniendo en cuenta la estética y el comportamiento de los adolescentes. Los colores, las fuentes y los iconos utilizados son juveniles y modernos, asegurando que la aplicación no solo sea funcional, sino también atractiva para este tipo de usuario. Además, la disposición de los elementos sigue un diseño limpio y minimalista, según la solicitud de simplicidad de los psicólogos, evitando el desorden y manteniendo la atención del usuario en lo que realmente importa.

A través de este diseño, buscamos no solo cumplir con las necesidades prácticas

del usuario, sino también crear un entorno digital donde los adolescentes se sientan motivados y seguros.

A continuación, se presenta el diseño prototípico de esta pantalla.

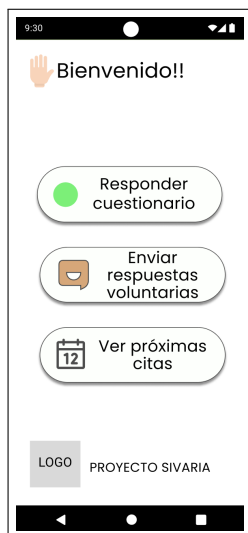


Figura 3.7: Pantalla Adolescente

3. Pantalla principal del Padre

La pantalla principal para padres en nuestra aplicación está diseñada para ofrecer una experiencia organizada y personalizada, permitiendo a los padres gestionar y supervisar las actividades de cada uno de sus hijos de manera efectiva. Al ingresar a esta pantalla, los padres son recibidos con un mensaje de saludo en la parte superior.

Debajo del saludo, se despliega un listado que muestra a todos los hijos registrados bajo la cuenta del padre, por si tuviese varios en el proyecto. Cada hijo aparece con un botón que lleva directamente su nombre, lo que facilita la identificación rápida y directa de cada perfil. Estos botones personalizados aseguran que los padres puedan acceder rápidamente a la gestión individual de cada hijo, con solo un clic en el nombre correspondiente.

Al seleccionar uno de estos botones con el nombre del hijo, el padre es dirigido a una pantalla dedicada exclusivamente a ese hijo, donde puede acceder a varias funciones diseñadas para monitorear su bienestar y progreso.

La primera función disponible es “Responder cuestionario”, que permite al padre completar un cuestionario para evaluar el estado o las necesidades de su hijo. Esta función es similar a la disponible en la pantalla inicial del adolescente.

La segunda funcionalidad es “Enviar respuestas voluntarias”, que también se asemeja a la opción disponible en la pantalla del adolescente. Aquí, el padre puede enviar comentarios o responder a preguntas específicas relacionadas con

su hijo, ofreciendo su perspectiva y contribuyendo a un entendimiento más completo del estado emocional del adolescente.

La tercera funcionalidad, “Ver Progresos”, permite al padre revisar el progreso y las respuestas anteriores de su hijo.

Cada una de estas funciones está representada por un botón claro y accesible, diseñado para que los padres puedan realizar las tareas necesarias con facilidad.

En conjunto, esta pantalla para padres facilita la gestión de las actividades y el seguimiento del progreso de cada hijo haciendo que los padres apoyen y se involucren activamente en este proceso con sus hijos, den ejemplo y apoyo.

A continuación, se presenta el diseño prototípico de ambas pantallas.

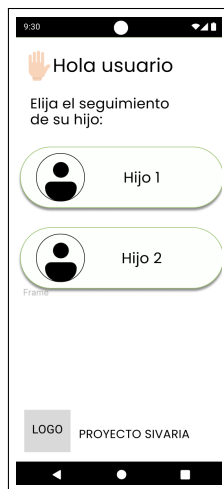


Figura 3.8: Pantalla Padre

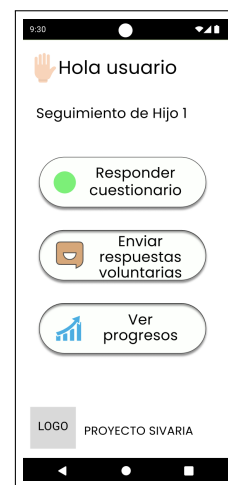


Figura 3.9: Pantalla Padre

4. Pantalla principal del Profesional

La pantalla principal para profesionales en nuestra aplicación está diseñada para que los profesionales puedan gestionar y supervisar las actividades de sus pacientes adolescentes de manera efectiva. Al acceder a esta pantalla, el profesional es recibido con un mensaje de saludo en la parte superior.

Inmediatamente debajo del saludo, se presenta un listado que muestra a todos los pacientes adolescentes asignados al profesional. Cada paciente está representado por un botón que lleva directamente su nombre, lo que facilita la identificación rápida y directa de cada uno. Estos botones personalizados aseguran que el profesional pueda acceder rápidamente a la gestión individual de cada paciente, con solo un clic en el nombre correspondiente.

Al seleccionar uno de estos botones con el nombre del paciente, el profesional es dirigido a una pantalla dedicada exclusivamente a ese paciente, donde puede acceder a varias funciones clave diseñadas para monitorear y apoyar su progreso.

La primera función disponible es “Ver próximas citas”, que permite al profesional revisar y modificar las citas programadas con ese paciente.

La segunda funcionalidad, “Ver Progresos”, ofrece al profesional la posibilidad de revisar el historial y el avance del paciente a lo largo del tiempo. Esta función es similar a la disponible para los padres.

La tercera funcionalidad es “Avisar a padres”, una herramienta crucial que permite al profesional notificar rápidamente al padre o tutor del paciente si surge alguna situación preocupante.

Cada una de estas funciones está representada por un botón claro y accesible, diseñado para que el profesional pueda navegar intuitivamente y realizar las tareas necesarias con eficacia facilitando el trabajo del profesional en un entorno organizado y centrado en el paciente.

A continuación, se presenta el diseño prototípico de ambas pantallas.

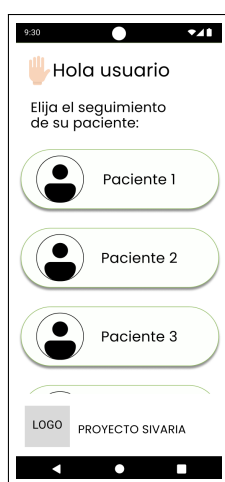


Figura 3.10: Pantalla Profesional

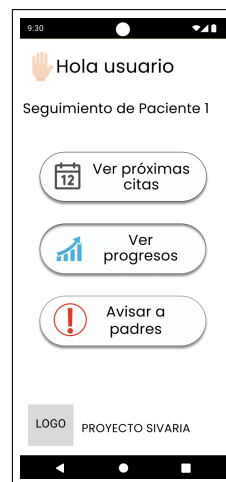


Figura 3.11: Pantalla Profesional

5. Pantalla del Cuestionario

La pantalla de cuestionarios en nuestra aplicación está diseñada para ser clara y fácil de usar, permitiendo a los usuarios concentrarse en responder las preguntas de manera eficiente. En la parte superior de la pantalla, se muestra la pregunta actual, presentada en un formato de texto grande y legible para asegurar que el usuario pueda entenderla con claridad. El diseño de esta sección está pensado para que la pregunta sea el foco principal.

Debajo de la pregunta, se presenta una serie de opciones de respuesta, cada una claramente etiquetada y espaciosamente dispuestas de manera que cada opción sea fácilmente seleccionable con un simple toque. Estas opciones pueden estar en forma de botones o cuadros de selección. El diseño asegura que solo una opción sea seleccionada a la vez, evitando confusiones y asegurando la precisión en las respuestas.

En la parte inferior de la pantalla, se encuentran dos botones que permiten al usuario navegar a través de las preguntas del cuestionario. El botón “Siguiente” se sitúa a la derecha, permitiendo avanzar a la siguiente pregunta una vez

que el usuario ha seleccionado su respuesta actual. A la izquierda, el botón “Anterior” permite retroceder a la pregunta anterior, en caso de que el usuario desee revisar o cambiar una respuesta previa. Estos botones serán grandes y tendrán su descripción facilitando la navegación a lo largo del cuestionario de manera fluida y sin complicaciones.

Una vez que el usuario ha respondido todas las preguntas del cuestionario, se le presenta una pantalla final que resume las opciones elegidas. En esta pantalla, se ofrecen dos opciones principales.

La primera opción es “Modificar Respuestas”, que permite al usuario revisar todas sus respuestas y hacer cambios si es necesario antes de finalizar el proceso. Esta opción es fundamental para asegurar que el usuario tenga la oportunidad de corregir cualquier error o reconsiderar sus respuestas.

La segunda opción es “Enviar Respuestas”, que permite al usuario enviar sus respuestas al servidor una vez que está satisfecho con ellas. Este botón está claramente destacado para asegurar que el usuario comprenda que esta acción finalizará el cuestionario y enviará las respuestas para su procesamiento o revisión. Al hacer clic en este botón, las respuestas se guardan y el usuario recibe una confirmación de que el cuestionario ha sido completado con éxito.

El diseño de esta pantalla de cuestionarios está pensada para que cada paso sea claro y directo. Desde la presentación de las preguntas hasta la navegación y la revisión final. Lo que garantiza que los datos recogidos sean fiables y útiles para el posterior análisis.

A continuación, se presenta el diseño prototípico de ambas pantallas.

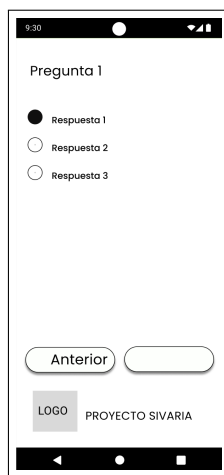


Figura 3.12: Pantalla Cuestionario

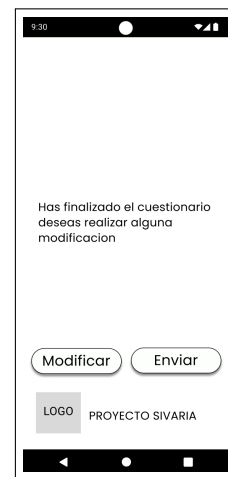


Figura 3.13: Pantalla Cuestionario

6. Pantalla Enviar respuestas voluntarias

La pantalla de “Enviar Respuestas Voluntarias” en nuestra aplicación está diseñada para que el usuario pueda dar información adicional en cualquier momento del día. Nos pareció buena idea ya que es una manera de dejar que se expreso siempre que lo necesite, y no sentirse abandonado. Al ingresar a esta

pantalla, el usuario se encuentra con un cuadro de texto amplio y centrado, donde puede escribir libremente el mensaje que desea enviar. Este cuadro de texto ocupa la mayor parte de la pantalla, ofreciendo un espacio cómodo y adecuado para redactar mensajes, ya sean cortos o extensos. La fuente utilizada resalta con el fondo, asegurando que el usuario pueda ver fácilmente lo que está escribiendo.

El cuadro de texto está diseñado para ser usado sin distracciones, lo que permite al usuario concentrarse en su mensaje. Además, el cuadro es lo suficientemente grande como para manejar entradas de texto largas, por si al usuario le cuesta expresarse o simplemente quiere usar muchas palabras. Como en cualquier cuadro de texto, el contenido se podrá borrar si fuese necesario.

Justo debajo del cuadro de texto, se encuentran dos botones, claramente etiquetados para facilitar su uso. El primer botón, “Enviar”, está colocado a la derecha y está diseñado para enviar el mensaje al servidor una vez que el usuario ha terminado de escribir. Al pulsarlo, el usuario recibe una confirmación de que su respuesta ha sido enviada correctamente. Para que el usuario tenga la certeza de que su mensaje ha sido recibido.

A la izquierda del botón “Enviar” se encuentra el botón “Cancelar”. Este botón permite al usuario optar por no enviar el mensaje, ya sea porque ha cambiado de opinión o porque no desea continuar con la redacción. Al pulsar “Cancelar”, se cierra la pantalla, devolviendo al usuario a la pantalla anterior sin guardar el contenido escrito.

Al diseñar esta pantalla, pensamos mucho en la libertad del usuario en todo momento, haciendo que pueda no enviar el mensaje en cualquier momento, así como poder hacerlo sin distracciones.

A continuación, se presenta el diseño prototípico de la pantalla.

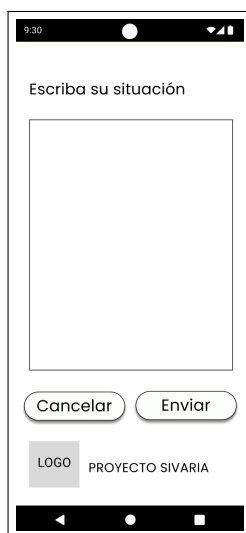


Figura 3.14: Pantalla Enviar respuestas voluntarias

7. Pantalla Ver progresos

La pantalla de “Ver Progresos” en nuestra aplicación está diseñada para ofrecer una visión clara y comprensible de la evolución de las respuestas del adolescente a lo largo del tiempo. Al acceder a esta pantalla, el usuario se encuentra con una gráfica de barras que ocupa la mayor parte del espacio, presentando de manera visual y directa los cambios en las respuestas del adolescente.

La gráfica de barras está cuidadosamente diseñada para ser fácil de interpretar. Cada barra representa un punto en el tiempo o una sesión específica, mostrando cómo han variado las respuestas en función de diferentes parámetros, como el estado emocional, el comportamiento o cualquier otro criterio evaluado a través de los cuestionarios. Las barras están codificadas por colores o acompañadas de etiquetas para diferenciar claramente entre los distintos aspectos evaluados, permitiendo una rápida identificación de tendencias y patrones.

El diseño de esta pantalla está centrado en la claridad y la facilidad de uso, asegurando que cualquier usuario, ya sea un padre, un profesional o el propio adolescente, pueda entender la información presentada sin dificultades. La gráfica es interactiva, permitiendo que el usuario toque una barra específica para obtener más detalles o ver las respuestas correspondientes a ese punto en el tiempo, proporcionando una experiencia más enriquecedora y detallada.

En conjunto, la pantalla de “Ver Progresos” ofrece una herramienta poderosa para visualizar y comprender el desarrollo del adolescente a lo largo del tiempo. Su diseño claro y accesible asegura que los datos se presenten de manera que sean fáciles de entender y de utilizar para tomar decisiones informadas sobre el bienestar del adolescente.

A continuación, se presenta el diseño prototípico de la pantalla.

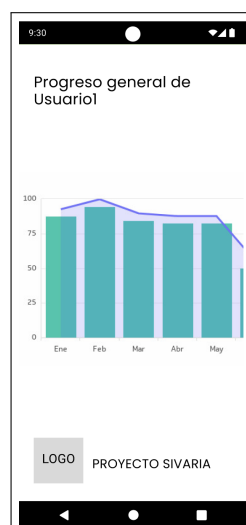


Figura 3.15: Pantalla Ver progresos

8. Pantalla Ver próximas citas

La pantalla de “Ver Próximas Citas” en nuestra aplicación está diseñada para facilitar la gestión y visualización de las citas programadas entre el profesional y el adolescente. Esta pantalla se presenta de manera ligeramente diferente para cada tipo de usuario, ajustándose a sus necesidades específicas.

Pantalla para Adolescentes

Para los adolescentes, la pantalla de “Ver Próximas Citas” es más simple que la del profesional, que será explicada a continuación, para enfocarse exclusivamente en la visualización de las citas programadas. Al acceder a esta pantalla, el adolescente ve una lista de sus próximas citas, organizada cronológicamente. Cada cita muestra la fecha y la hora, proporcionando al adolescente una visión clara de sus próximos compromisos.

A diferencia de la pantalla del profesional, el adolescente no tiene opciones para modificar la lista de citas. No hay íconos de papelera ni botones de “Agendar Cita”.

A continuación, se presenta el diseño prototípico de la pantalla.



Figura 3.16: Pantalla Citas Adolescente

Pantalla para Profesionales

Al ingresar a la pantalla, el profesional ve una lista clara y organizada de las próximas citas programadas con cada paciente. Cada cita está representada por una fila que muestra la fecha y la hora específicas, junto con el nombre del adolescente correspondiente. A la derecha de cada cita, se encuentra un ícono de papelera, que permite al profesional eliminar la cita si es necesario. Este ícono facilita la rápida y eficiente gestión de las citas.

En la parte inferior de la lista de citas, se encuentra un botón destacado con la etiqueta “Agendar Cita”. Este botón, diseñado para ser fácilmente identificable, permite al profesional añadir nuevas citas al calendario. Al pulsar “Agendar Cita”, el profesional es dirigido a una nueva pantalla donde se muestra un calendario estilo mes, con una vista completa de todos los días del mes actual. Aquí, el profesional puede seleccionar el día deseado para la nueva cita simplemente tocando la fecha correspondiente.

Debajo del calendario, hay un campo donde el profesional puede ingresar la hora exacta de la cita. Esta opción de entrada de hora permite una selección precisa de la hora que mejor se adapte a las necesidades del paciente y el profesional. Una vez seleccionados el día y la hora, un botón en la parte inferior de la pantalla, etiquetado como “Confirmar”, permite guardar la nueva cita. Al pulsar este botón, la cita se guarda y se añade automáticamente a la lista de próximas citas, actualizando la pantalla principal para reflejar este cambio.

A continuación, se presenta el diseño prototípico de ambas pantallas.



Figura 3.17: Pantalla Citas Profesional 1

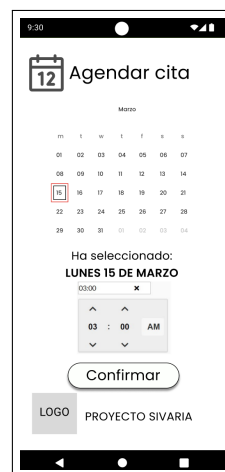


Figura 3.18: Pantalla Citas Profesional 2

El diseño de la pantalla para ambos tipos de usuarios es muy visual, teniendo las fechas ordenadas cronológicamente. Además es fácil de usar ya que sus elementos visuales están dispuestos de manera ordenada, y las acciones disponibles (como eliminar o añadir citas) están muy bien diferenciadas. La tipografía es clara y legible, asegurando que la información clave esté siempre visible y fácil de entender.

En conjunto, la pantalla de “Ver Próximas Citas” proporciona una herramienta eficaz tanto para los profesionales como para los adolescentes, adaptándose a sus necesidades específicas. El diseño asegura que la gestión de las citas sea una tarea simple y directa, apoyando la organización y el seguimiento de los compromisos de manera eficiente.

3.3.4. Pruebas con usuarios

Tras completar el prototipado inicial de nuestra aplicación, organizamos una reunión con los psicólogos para revisar el diseño y las funcionalidades propuestas. Esta sesión fue fundamental para evaluar cada característica incluida en el prototipo y determinar su relevancia y utilidad en el contexto real de uso. Durante la reunión, se llevó a cabo un análisis detallado del prototipo y se discutieron las diferentes funcionalidades implementadas.

Los psicólogos expresaron su deseo de simplificar la aplicación, con el objetivo de enfocarse en las funciones más esenciales y evitar cualquier complejidad innecesaria. En particular, se identificaron varias áreas donde las funcionalidades propuestas podrían ser optimizadas para mejorar la experiencia del usuario y hacer el sistema más eficiente.

3.3.5. Iteración y mejora continua

Como resultado del feedback recibido, se decidieron eliminar varias funcionalidades específicas que se consideraban innecesarias para los objetivos principales de la aplicación. Primero, se eliminó la funcionalidad de “Ver Próximas Citas”, que permitía a los profesionales programar nuevas citas directamente a través de la aplicación. Se concluyó que esta función no aportaba un valor significativo, por lo que se optó por reducir la complejidad del sistema y concentrar a los usuarios en la realización de los cuestionarios.

Asimismo, se descartó la funcionalidad de “Avisar a Padres”, que ofrecía a los profesionales la posibilidad de enviar notificaciones a los padres en situaciones preocupantes. Se decidió mantener la comunicación con los padres a través de métodos más directos y personalizados, eliminando la dependencia de mensajes automatizados o integrados en la plataforma.

También se eliminaron las pantallas de lista de hijos y pacientes. Inicialmente destinadas a mostrar una lista de los hijos y pacientes de un usuario, estas pantallas se consideraron innecesarias debido a la nueva concepción del sistema, que asume que habrá un usuario por cada paciente.

Una vez que se acordaron y detallaron estos cambios, actualizamos el prototipo para reflejar las nuevas especificaciones. El diseño revisado fue presentado nuevamente al cliente, quien aprobó las modificaciones finales. Con la confirmación de que el prototipo cumplía con las expectativas y necesidades solicitadas, procedimos a la siguiente fase del proyecto: el desarrollo de la aplicación. Este enfoque revisado asegura que el desarrollo se centre en las funcionalidades esenciales acordadas, optimizando la aplicación para cumplir con los objetivos específicos del cliente y mejorar la experiencia del usuario.

3.4. Desarrollo de la aplicación móvil en React Native

Para el desarrollo de la aplicación, como mencionamos anteriormente, hemos optado por utilizar React Native. Este entorno nos brinda la capacidad de crear tanto aplicaciones móviles como web, lo que nos ofrece la flexibilidad de desarrollar para múltiples plataformas, optimizando tanto el tiempo como los recursos, sin sacrificar la calidad o el rendimiento en cada entorno. En esta sección, detallaremos las diversas funciones que hemos tenido que implementar para que la aplicación pueda ejecutar todas las funcionalidades ideadas en el prototipo inicial, el cual fue desarrollado al inicio del curso con la colaboración de psicólogos de la UNED.

Durante el proceso de desarrollo, nos enfrentamos a diversos desafíos que requerían soluciones específicas y eficientes. Entre estos desafíos se encontraban el uso de un lenguaje de programación completamente nuevo para nosotros, la gestión de datos en tiempo real y los avisos mediante notificaciones. Para abordar estos desafíos, nos centramos en aprender JavaScript, e intentar aprovechar al máximo las capacidades que ofrece React-Native, sobre todo usando las bibliotecas que proporciona su comunidad, que ofrecen todo tipo de funcionalidad que podíamos necesitar.

Además, colaboramos estrechamente con psicólogos de la UNED para asegurarnos de que las funcionalidades implementadas en la aplicación cumplieran con los requisitos. Su experiencia y retroalimentación fueron fundamentales para garantizar que la aplicación ofreciera la funcionalidad idónea para los usuarios.

En resumen, el proceso de desarrollo de la aplicación fue un esfuerzo colaborativo que combinó nuestro conocimiento técnico con la experiencia profesional de los psicólogos. El resultado final es una aplicación que integra funcionalidades avanzadas con un enfoque centrado en el usuario.

3.4.1. Librerías y Herramientas Utilizadas en React Native

En el desarrollo de la aplicación, se han integrado diversas librerías y herramientas para aprovechar sus funcionalidades específicas y facilitar la implementación de características clave. (React Native Community, 2024) A continuación, se detalla el uso y propósito de cada una:

1. @react-native-firebase/app

- **Descripción:** La librería `@react-native-firebase/app` forma parte del conjunto de paquetes de Firebase para React Native. Esta librería facilita la configuración inicial y la comunicación con Firebase. Firebase es una plataforma de desarrollo de aplicaciones de Google que ofrece una variedad de herramientas para crear, mejorar y gestionar aplicaciones móviles y web. En nuestro caso usaremos su servicio de mensajería, Firebase Cloud Messaging (FCM), que permite enviar mensajes

y notificaciones de manera gratuita a dispositivos móviles y aplicaciones web. (Khawas y Shah, 2018)

- **Uso en el Proyecto:** En nuestro proyecto, se utilizó para configurar y conectar la aplicación con Firebase, lo cual es crucial para la gestión de tokens de usuarios y envío de notificaciones mediante Firebase Cloud Messaging ideal para nuestro proyecto porque envía notificaciones push, directamente a los usuarios en sus dispositivos, incluso cuando la aplicación no está abierta y funciona tanto en Android, iOS como en navegadores web, fundamental para no restringir el uso a ciertos usuarios.

2. `@react-native-firebase/messaging`

- **Descripción:** `@react-native-firebase/messaging` es una librería diseñada para integrar Firebase Cloud Messaging (FCM) en aplicaciones React Native. También, permite a las aplicaciones recibir y gestionar alertas y mensajes desde el servidor. Ofrece una integración sencilla con las APIs de FCM y proporciona herramientas para la gestión de tokens de dispositivo, el manejo de mensajes en segundo plano y la personalización de notificaciones.
- **Uso en el Proyecto:** En el proyecto, ‘`@react-native-firebase/messaging`’ se utiliza para implementar la funcionalidad de notificaciones push. Esto permite que la aplicación envíe y reciba mensajes importantes, como recordatorios para completar cuestionarios o alertas sobre nuevas actualizaciones. La librería asegura que los usuarios reciban notificaciones relevantes, incluso cuando la aplicación está en segundo plano, mejorando así la comunicación y el compromiso con la aplicación.

3. `@react-navigation/native` y `@react-navigation/stack`

- **Descripción:** `@react-navigation/native` es una librería esencial para la navegación en aplicaciones React Native, proporcionando un contenedor de navegación y hooks para gestionar la navegación entre pantallas. `@react-navigation/stack` ofrece una implementación de navegación basada en pilas, lo que permite una transición de pantallas tipo pila (stack), ideal para la mayoría de las aplicaciones móviles.
- **Uso en el Proyecto:** Se emplearon estas librerías para implementar la navegación dentro de la aplicación, permitiendo a los usuarios moverse entre diferentes pantallas de manera intuitiva. `@react-navigation/stack` facilita la navegación mediante pilas, lo que simplifica la gestión de las pantallas abiertas y las transiciones entre ellas.

4. `react-native-paper`

- **Descripción:** `react-native-paper` es una librería que ofrece una serie de componentes de interfaz de usuario (UI) que siguen las directrices de Material Design de Google. Estos componentes incluyen botones, tarjetas, diálogos, y más, y están diseñados para proporcionar una apariencia coherente y moderna en aplicaciones React Native.

- **Uso en el Proyecto:** Esta librería se utilizó para implementar componentes UI estilizados y funcionales, como botones y radio buttons. La utilización de **react-native-paper** garantiza una interfaz de usuario atractiva y consistente con los principios de diseño modernos, mejorando la experiencia visual de la aplicación.

5. @react-native-community/slider

- **Descripción:** **@react-native-community/slider** proporciona un componente de deslizador (slider) que permite a los usuarios ajustar valores mediante una interfaz de deslizamiento. Es altamente personalizable y se integra bien con React Native para ofrecer una experiencia de usuario interactiva y precisa.
- **Uso en el Proyecto:** Se utilizó para implementar controles deslizantes en la aplicación.

6. react-native-chart-kit

- **Descripción:** **react-native-chart-kit** es una librería que facilita la creación de gráficos y visualizaciones de datos en aplicaciones React Native. Ofrece una variedad de tipos de gráficos, como gráficos de líneas, barras y circulares, permitiendo la representación visual clara de datos numéricos.
- **Uso en el Proyecto:** Se empleó para generar gráficos lineales, permitiendo a los usuarios visualizar datos de manera gráfica y comprensible. Esto es útil para mostrar métricas, tendencias y otros datos importantes que componen la pantalla “ver progresos”.

3.4.2. Diseño de las pantallas

El diseño de pantallas de nuestra aplicación ha sido cuidadosamente desarrollado para ofrecer una buena y eficiente experiencia de usuario. A lo largo del proceso de desarrollo, se ha puesto un énfasis especial en la funcionalidad, la simplicidad y la accesibilidad, asegurando que cada pantalla cumpla con su propósito de manera clara y efectiva.

En las siguientes secciones, se detallan las diferentes pantallas que componen la interfaz de usuario de la aplicación, mostrando cómo cada una ha sido concebida para facilitar la interacción del usuario con las distintas funcionalidades que ofrece la plataforma. Desde el acceso inicial hasta las tareas específicas que se pueden realizar, cada pantalla ha sido diseñada para guiar al usuario de manera fluida a través de la aplicación, siempre teniendo en cuenta sus necesidades y expectativas.

A través de esta presentación, se destacarán las principales características visuales y funcionales de cada pantalla, reflejando cómo han sido optimizadas en la versión final del producto para brindar una experiencia de usuario superior.

3.4.2.1. Interfaz de Iniciar Sesión

En la versión final de la aplicación, la pantalla de inicio de sesión ha sido optimizada para ofrecer una experiencia aún más directa y centrada en la funcionalidad principal: permitir a los usuarios acceder rápidamente a su cuenta. Se ha mantenido el diseño limpio y eficiente, con el logo de la aplicación visible en la parte superior, este logo ha sido diseñado provisionalmente por nosotros con ayuda de una inteligencia artificial llamada (FreeLogoDesign, 2022) , ya que el que será el logo oficial del proyecto y app aún no está creado por los precursores. Seguido de este se encuentran los campos de entrada para el nombre de usuario y la contraseña. Estos elementos han sido diseñados para ser claros y accesibles, asegurando que los usuarios puedan ingresar sus credenciales sin complicaciones.

Uno de los ajustes clave en esta fase fue la decisión tomada por los psicólogos de eliminar la opción de recuperación de contraseña, quienes nos dijeron que esta funcionalidad sería llevada a cabo a través de su página web, que ya está en funcionamiento y con la que también se han creado y crearán los usuarios del proyecto y sus respectivas claves.

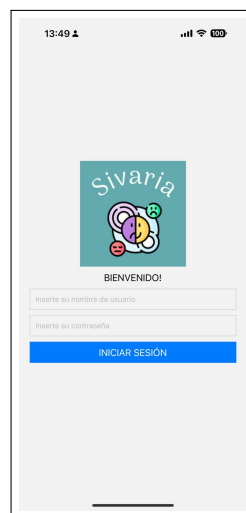


Figura 3.19: Interfaz Iniciar Sesión

3.4.2.2. Interfaz del Adolescente

En la versión final de la aplicación, la pantalla principal para los adolescentes se ha diseñado para ofrecer una experiencia de usuario optimizada y directa. Al ingresar a la aplicación, los adolescentes encuentran un saludo personalizado en la parte superior de la pantalla, lo que proporciona una recepción cálida y personalizada desde el inicio.

La pantalla está organizada en torno a tres botones principales que permiten a los adolescentes acceder a las funciones más relevantes. El primer botón les da acceso a un cuestionario para evaluar su bienestar. Este cuestionario ha sido implementado para facilitar la recolección de información sobre su estado de ánimo y otras métricas

importantes, con una interfaz clara que hace que completar las preguntas sea un proceso fluido y sencillo.

El segundo botón proporciona acceso a la funcionalidad de mensajería, permitiendo a los adolescentes enviar mensajes de texto. Este componente está diseñado para que la comunicación con profesionales o el equipo de apoyo sea rápida y efectiva, eliminando cualquier barrera en el proceso de interacción y ofreciendo un canal de comunicación adicional.

Una adición importante en esta versión es el botón de “Cerrar Sesión”, ubicado en la parte inferior de la pantalla. Este botón ha sido incorporado para garantizar que los usuarios puedan finalizar su sesión de manera segura. Su colocación y diseño buscan ofrecer una solución práctica para asegurar que las cuentas permanezcan protegidas después de cada uso, sin que este elemento interfiera con el acceso a las funcionalidades principales.

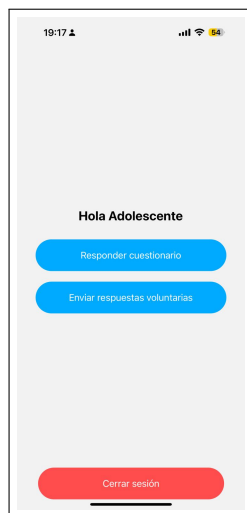


Figura 3.20: Interfaz Adolescente

3.4.2.3. Interfaz del Padre

En la versión final de la aplicación, la pantalla principal del padre ha sido optimizada para ofrecer una experiencia de usuario más directa y simplificada.

Al iniciar sesión, el padre es recibido con un saludo personalizado en la parte superior de la pantalla, estableciendo una bienvenida amigable y adaptada a cada usuario. En lugar de presentar una lista de hijos, la pantalla ahora se centra en proporcionar acceso inmediato a las funcionalidades clave mediante una interfaz consolidada.

Desde esta pantalla, el padre puede acceder a un cuestionario diseñado para evaluar el estado del hijo. Esta funcionalidad permite recopilar información detallada sobre el bienestar del niño a través de una serie de preguntas estructuradas, con una interfaz clara que facilita la navegación y la respuesta.

Además, el padre tiene la opción de enviar respuestas voluntarias, proporcionan-

do un canal de comunicación adicional para compartir comentarios o información relevante sobre el hijo. Esta funcionalidad está diseñada para asegurar una interacción fluida y efectiva con los profesionales.

También se incluye la opción de ver el progreso de las respuestas del hijo, permitiendo al padre revisar y analizar los datos obtenidos de los cuestionarios anteriores. La visualización del progreso está diseñada para ofrecer una comprensión clara del desarrollo del niño, ayudando al padre a seguir de cerca su evolución.

Para asegurar una gestión segura de la cuenta, al igual que en la interfaz del adolescente, se ha añadido un botón de “Cerrar Sesión” en la parte inferior de la pantalla. Este elemento proporciona una forma sencilla y segura de finalizar la sesión al concluir el uso de la aplicación, garantizando que la cuenta del usuario permanezca protegida.

En resumen, la pantalla principal del padre se ha rediseñado para eliminar la lista de hijos y concentrar todas las funciones esenciales en una única interfaz. Este enfoque simplifica la navegación y mejora la eficiencia, asegurando que el padre pueda gestionar y supervisar el bienestar del hijo de manera efectiva en un entorno de aplicación optimizado.

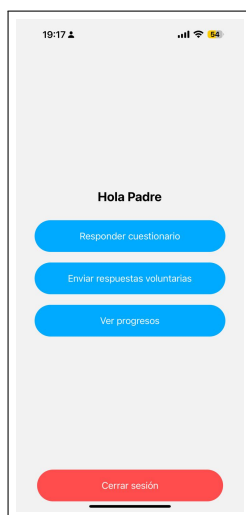


Figura 3.21: Interfaz Padre

3.4.2.4. Interfaz del Profesional

En la versión final de la aplicación, la pantalla principal del profesional ha sido optimizada para ofrecer una experiencia de usuario más directa y simplificada, con una gran similitud con la pantalla del padre y del adolescente.

Esta pantalla solo incluye la opción de ver el progreso de las respuestas del paciente, permitiendo al profesional revisar y analizar los datos obtenidos de los cuestionarios que ha realizado el paciente. La visualización del progreso está diseñada para ofrecer una comprensión clara del desarrollo del adolescente, ayudando al profesional analizar con detalles su evolución.

Para asegurar una gestión segura de la cuenta, al igual que en la interfaz del adolescente y la del padre , se ha añadido un botón de “Cerrar Sesión” en la parte inferior de la pantalla. Esta funcionalidad nos aporta una facilidad a la hora de finalizar la sesión al concluir el uso de la aplicación, garantizando que la cuenta del usuario permanezca protegida.

En resumen, la pantalla principal del profesional se ha cambiado para eliminar la lista de pacientes y concentrar todas las funciones esenciales en una única interfaz. Con estos cambios simplificamos la navegación y mejoramos la eficiencia, asegurando que el profesional vea la evolución de su paciente de manera efectiva en un entorno de aplicación optimizado.

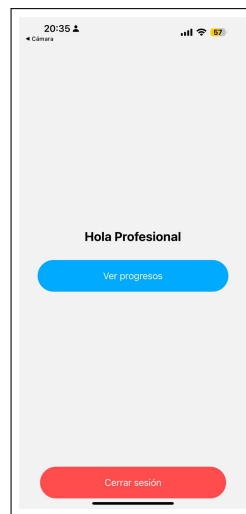


Figura 3.22: Interfaz Profesional

3.4.2.5. Interfaz del cuestionario

La pantalla del cuestionario está diseñada para proporcionar una experiencia de usuario clara y adaptada a diferentes tipos de respuestas, facilitando la recopilación detallada de información sobre el estado del adolescente.

Al acceder a esta pantalla, el usuario se encuentra con una pregunta en la parte superior, formulada de manera precisa para dirigir la atención hacia un aspecto específico del bienestar o estado del adolescente.

Debajo de la pregunta, el diseño presenta varias opciones de respuesta, cada una adaptada a diferentes formatos de entrada.

Entre las opciones disponibles, se incluyen:

- **Uniselección:** En este caso, el usuario solo puede elegir una opción de entre varias. Es útil para preguntas donde solo una respuesta es válida o se desea forzar una elección clara y definida. Un ejemplo típico sería preguntar sobre la frecuencia de las conductas autolesivas en el último mes, con opciones como “nunca”, “una vez”, “varias veces”, o “casi todos los días”.

- **Multiselección:** Este tipo de pregunta permite al usuario seleccionar múltiples opciones de una lista. Es ideal para situaciones en las que se necesita recopilar más de una respuesta posible o cuando las opciones no se excluyen mutuamente. Por ejemplo, se puede preguntar al adolescente sobre los motivos que lo llevan a auto-lesionarse, permitiendo seleccionar múltiples opciones como alivio del dolor emocional, escape de la realidad, sentimiento de control, o expresión de sentimientos difíciles.
- **Barstatus:** Este tipo de pregunta utiliza una barra de estado o de progreso para medir una respuesta en una escala continua. Es particularmente útil para evaluar niveles de intensidad emocional, gravedad de los síntomas, o cualquier otra variable que se pueda cuantificar en un rango. Por ejemplo, se puede pedir al adolescente que indique el nivel de angustia emocional que siente antes y después de auto-lesionarse, en una escala del 1 al 10, arrastrando un control deslizante para reflejar con precisión sus sentimientos.

A continuación se presentan las imágenes de las tres pantallas:

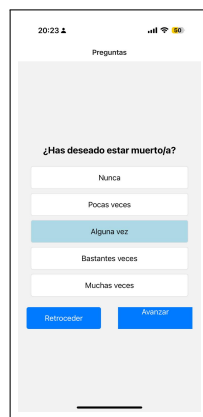


Figura 3.23: Interfaz Cuestionario Uniselección

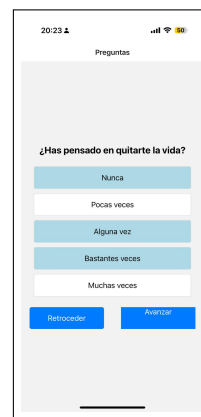


Figura 3.24: Interfaz Cuestionario Multiselección

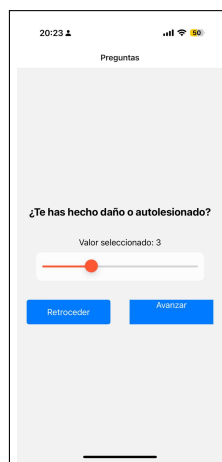


Figura 3.25: Interfaz Cuestionario BarStatus

En la parte inferior de la pantalla, se encuentran dos botones clave. El botón “Avanzar” permite al usuario pasar a la siguiente pregunta del cuestionario, mientras que el botón “Retroceder” brinda la opción de regresar a la pregunta anterior para realizar revisiones o ajustes.

En conjunto, la pantalla del cuestionario ha sido diseñada para abordar diferentes tipos de respuestas. La inclusión de opciones de uniselección, multiselección y *bars-tatus* permite una recopilación detallada y precisa de la información, optimizando la experiencia de usuario durante el proceso de respuesta.

3.4.2.6. Interfaz Enviar respuestas voluntarias

La pantalla para enviar respuestas voluntarias ha mantenido su diseño esencial tal y como se definió en el prototipado inicial, asegurando consistencia y familiaridad en la experiencia del usuario.

Al acceder a esta pantalla, el usuario encuentra un cuadro de texto en la parte superior, diseñado para ingresar libremente sus comentarios adicionales. Este cuadro de texto es el principal medio para que el usuario proporcione información adicional relevante que no quiera compartir sin poder hacerlo en un cuestionario porque no haya uno disponible.

Justo debajo del cuadro de texto, se presentan dos botones funcionales. El primer botón, “Enviar”, permite al usuario enviar el contenido del cuadro de texto al servidor.

El segundo botón, “Cancelar”, ofrece al usuario la opción de abortar el proceso de envío si decide no proporcionar una respuesta adicional o si cambia de opinión. Al seleccionar este botón, el usuario regresa a la pantalla anterior sin enviar la información ingresada, preservando la integridad de su interacción con la aplicación.

Este diseño asegura que la funcionalidad de enviar respuestas voluntarias sea simple y efectiva, proporcionando un canal claro y directo para la comunicación de información adicional sin complicaciones innecesarias.

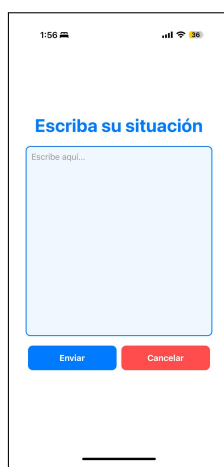


Figura 3.26: Interfaz Respuestas Voluntarias

3.4.2.7. Interfaz Ver progreso

La pantalla de “Ver Progreso” se ha desarrollado para proporcionar a los usuarios una visión clara y comprensible del avance del adolescente en función de las respuestas dadas en los cuestionarios durante los meses.

En el núcleo de la pantalla se encuentra una gráfica de barras que visualiza el progreso del adolescente. Esta gráfica ofrece una representación gráfica de las respuestas acumuladas, permitiendo una fácil interpretación de las tendencias y cambios a lo largo del tiempo. Las barras en la gráfica reflejan los diferentes aspectos evaluados, con cada barra representando un área específica del cuestionario y mostrando la evolución en cada una.

Una adición significativa al diseño final es la inclusión de un botón “Volver” situado justo debajo de la gráfica. Este botón proporciona a los usuarios una manera rápida y sencilla de salir de la vista de progreso y regresar a la pantalla anterior o al menú principal. La incorporación de esta funcionalidad mejora la navegación, permitiendo un acceso más fluido y conveniente a otras partes de la aplicación sin necesidad de realizar múltiples acciones para volver atrás.

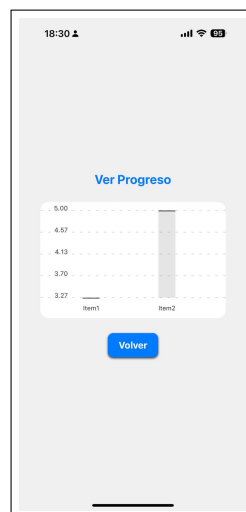


Figura 3.27: Interfaz Ver progreso

3.4.2.8. Interfaz de la pantalla final

Una vez que el usuario ha completado todas las preguntas del cuestionario, se le dirige a una pantalla de resumen final diseñada para revisar y confirmar sus respuestas antes de enviarlas. En esta pantalla, se presenta un listado completo de todas las preguntas respondidas, mostrando cada pregunta junto con la respuesta seleccionada por el usuario. Este listado se organiza en un formato de desplazamiento vertical (ScrollView), lo que permite al usuario revisar cómodamente todas sus respuestas en un solo lugar, sin necesidad de navegar por múltiples pantallas.

El ScrollView facilita la navegación a través de las respuestas, asegurando que incluso en cuestionarios largos, el usuario pueda revisar todas sus selecciones de

manera ordenada y sin perder de vista ninguna pregunta. Esta funcionalidad es especialmente útil para garantizar que el usuario pueda verificar que todas sus respuestas sean correctas antes de finalizar el proceso.

En la parte inferior de la pantalla, se encuentran dos botones clave que permiten al usuario tomar la acción final sobre el cuestionario. El botón “Modificar Respuestas” brinda al usuario la opción de regresar al cuestionario para realizar cambios o correcciones en sus respuestas si considera que alguna respuesta necesita ser ajustada. Este botón proporciona una capa adicional de flexibilidad, permitiendo que el usuario se sienta seguro de que las respuestas enviadas son las más precisas posibles.

El segundo botón, “Enviar Respuestas”, se ubica junto al anterior y permite al usuario enviar todas las respuestas recopiladas al servidor para su procesamiento y análisis. Al presionar este botón, el usuario completa el cuestionario, asegurando que la información proporcionada sea capturada de manera efectiva para su uso posterior.

Esta pantalla de resumen ha sido diseñada para proporcionar una experiencia de usuario intuitiva y eficiente, asegurando que todas las respuestas se revisen y confirmen antes del envío final, minimizando errores y aumentando la precisión de los datos recopilados.

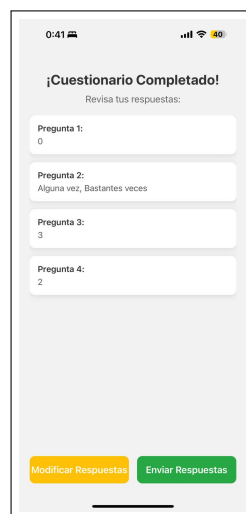


Figura 3.28: Interfaz Final

3.4.3. Diagrama de Navegación de la Aplicación

Una vez explicados los diferentes formatos y componentes de cada una de las pantallas de la aplicación, es importante visualizar cómo se estructuran y conectan entre sí. A continuación, se presenta un diagrama de navegación que refleja el flujo de la aplicación según los distintos roles de usuario (Adolescente, Padre y Profesional).

Este diagrama permite entender cómo, a partir de la pantalla de inicio de sesión, el usuario es dirigido a diferentes funcionalidades en función de su perfil. Además, en cada una de las pantallas principales se incluye la opción de cerrar sesión para

facilitar la salida del sistema. El diagrama también muestra las relaciones entre las pantallas principales y secundarias, como los cuestionarios, el envío de respuestas voluntarias, y la visualización del progreso en caso de los padres y profesionales, tal y como se muestra en la siguiente imagen.

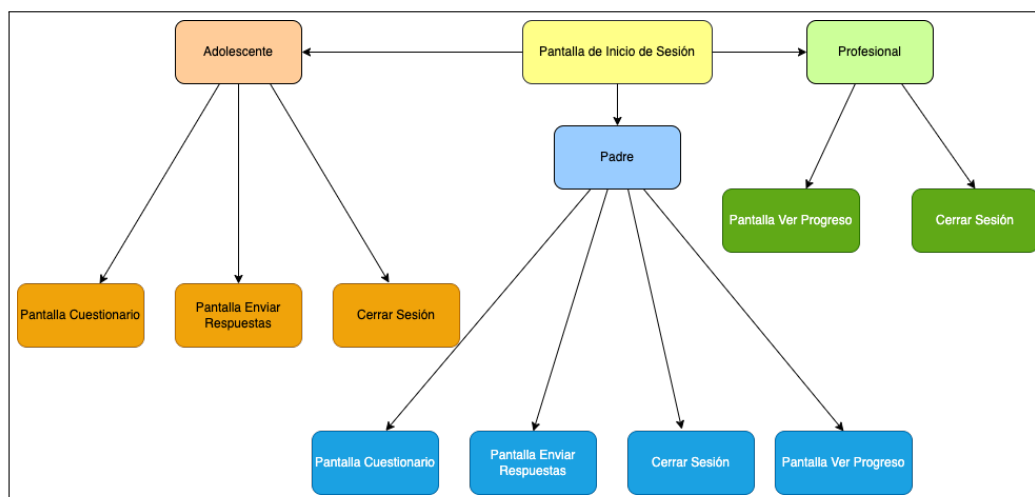


Figura 3.29: Diagrama de Navegación

3.5. Desarrollo de la Base de Datos en MySQL

Como bien hemos mencionado este trabajo de fin de grado ha sido encargado por los psicólogos de la UNED. Esto nos ha llevado a tener que afrontar retos adicionales con respecto a saber exactamente cómo debían estar estructurados los datos, dónde, etc. En principio, los psicólogos nos comentaron que ellos tendrían su propia base de datos. Sin embargo, para avanzar a nuestro propio ritmo, ampliar el alcance del proyecto y aprender a gestionar el almacenamiento de datos, decidimos que la creación de la base de datos debía formar parte integral de nuestro trabajo.

Hemos decidido utilizar **MySQL** como el sistema de gestión de bases de datos (SGBD). Esta elección se basa en la idoneidad de MySQL para los requisitos específicos del proyecto, ya que se ajusta mejor a nuestras necesidades de organizar y conservar datos estructurados. A diferencia de otras alternativas como MongoDB, que almacena los datos de manera menos organizada, o Redis, que no permite guardar datos de forma permanente, MySQL nos ofrece una solución robusta y eficiente para el almacenamiento seguro y ordenado de la información.

MySQL es un sistema de bases de datos relacional que sigue el modelo de tablas y relaciones entre datos, ofreciendo una estructura clara y bien definida. A diferencia de **MongoDB**, que es una base de datos NoSQL basada en documentos, MySQL permite gestionar datos en un esquema estructurado con soporte completo para transacciones ACID. Esto garantiza la atomicidad, consistencia, aislamiento y durabilidad de las operaciones, aspectos fundamentales para la integridad de los datos en este proyecto. La capacidad de MySQL para manejar consultas complejas y

relaciones entre tablas se ajusta perfectamente a las necesidades del proyecto, donde la integridad y la precisión de los datos son cruciales.

En comparación con **Redis**, que es una base de datos en memoria conocida por su alta velocidad y almacenamiento temporal, MySQL ofrece una solución más adecuada para la persistencia a largo plazo de datos estructurados. Redis es excelente para casos de uso específicos que requieren acceso ultrarrápido a datos en memoria, pero su enfoque en el almacenamiento en caché y la falta de un esquema estructurado lo hacen menos adecuado para el almacenamiento y la gestión de datos permanentes en este proyecto.

En resumen, hemos decidido usar MySQL debido a:

- **Modelo Relacional:** La estructura de tablas y relaciones de MySQL es ideal para gestionar datos con una estructura fija y realizar consultas complejas, asegurando la integridad y consistencia de los datos.
- **Soporte para Transacciones ACID:** MySQL garantiza la durabilidad y fiabilidad de las operaciones de base de datos, lo que es esencial para la consistencia de los datos a lo largo del proyecto.
- **Familiaridad y Herramientas:** La experiencia previa con MySQL y la disponibilidad de herramientas como MySQL Workbench facilitan el diseño, la gestión y la optimización de la base de datos.

Estas características hacen de MySQL la opción ideal para este proyecto, donde la prioridad es una gestión de datos estructurada y confiable.

3.5.1. Relaciones entre Tablas de la Base de Datos

A continuación, se muestra un diagrama que ilustra las relaciones entre las tablas de la base de datos. Este diagrama proporciona una visión general de cómo las diferentes tablas están conectadas y cómo se relacionan entre sí a través de las claves foráneas. Las conexiones y dependencias entre las tablas son esenciales para comprender la estructura de la base de datos y cómo se organizan los datos en el sistema. Este diagrama ha sido realizado utilizando la aplicación phpMyAdmin, una herramienta de administración de bases de datos basada en la web, que permite gestionar fácilmente bases de datos MySQL.

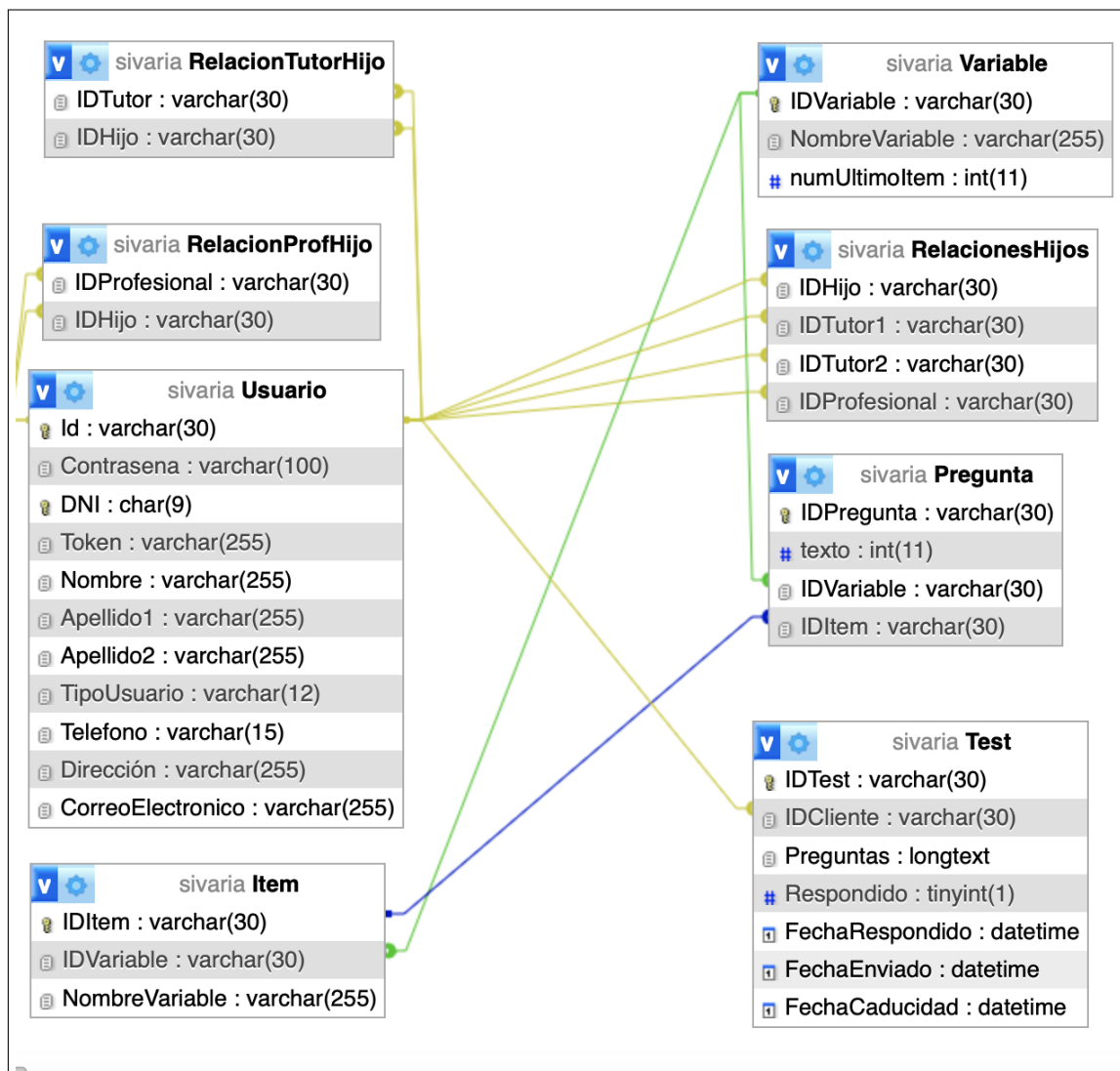


Figura 3.30: Diagrama de Relaciones entre las Tablas de la Base de Datos

3.5.2. Detalles de la estructura de la base de datos

Siguiendo el diagrama de relaciones y según los requisitos recogidos a lo largo de las reuniones con los psicólogos, detallaremos la estructura de las tablas de la base de datos creada, cabe destacar de dicha reunión que el identificador de cada usuario nos lo iban a proporcionar ellos por lo tanto al ser clave única y suponiendo que cada clave que ellos nos den no están repetidos:

3.5.2.1. Tablas

1. **Tabla Usuario:** Esta tabla es la tabla central de la base de datos. Almacena la información personal de todos los usuarios del proyecto Sivaria, siendo estos padres, jóvenes y profesionales, como ya se mencionó. Cada usuario tendrá su identificador, único, que se usará como clave primaria.

Además tendrá los siguientes campos:

Tabla 3.4: Tabla Usuario

Campo	Tipo de Dato	Descripción
Id	varchar(30)	Identificador único de cada usuario.
Contraseña	varchar(100)	Contraseña del usuario, almacenada de manera segura.
DNI	char(9)	Documento Nacional de Identidad del usuario, que es único y por lo tanto, también está marcado como 'unique', evitando duplicados.
Token	varchar(255)	Token de notificaciones, proporcionado al abrir la app.
Nombre	varchar(255)	Nombre del usuario.
Primer apellido	varchar(255)	Primer apellido del usuario.
Segundo apellido	varchar(255)	Segundo apellido del usuario (puede ser nulo).
Tipo de usuario	varchar(12)	Define el rol del usuario dentro del sistema (padre, joven, profesional).
Teléfono	varchar(15)	Número de teléfono del usuario.
Dirección	varchar(255)	Dirección del usuario (puede ser nulo).
Correo electrónico	varchar(255)	Correo electrónico del usuario (puede ser nulo).
Sensaciones	TEXT	Texto en el que el usuario expresa sus sensaciones.

Como índices tendrá el id, ya que es clave primaria y un índice único, DNI, que nunca habrá dos iguales.

2. **Tabla RelacionesHijos** Esta tabla se utiliza para gestionar las relaciones entre los hijos (jóvenes), sus tutores y los profesionales que los atienden. Tiene los siguientes campos:

Tabla 3.5: Tabla de RelacionesHijos

Campo	Tipo de Dato	Descripción
IDHijo	varchar(30)	Identificador único del usuario del que describimos las relaciones.
IDTutor1	varchar(30)	Identificador único para el primer tutor legal del joven.
IDTutor2	varchar(30)	Identificador único para el segundo tutor legal del joven. Puede ser nulo ya que no sabemos la situación familiar del joven.

Campo	Tipo de Dato	Descripción
IDProfesional	varchar(30)	Identificador único del profesional que atiende al joven.

Todos los Identificadores son referencias a la tabla Usuario. Como índice tendrá el identificador del joven, que será la clave única, ya que queremos definir las relaciones que tiene con sus responsables.

3. **Tabla RelacionProfHijo** Esta tabla almacena las relaciones entre profesionales y jóvenes (pacientes). Tiene los siguientes campos:

Tabla 3.6: Tabla RelacionProfHijo

Campo	Tipo de Dato	Descripción
IDProfesional	varchar(30)	Identificador único del profesional que trabaja con el joven.
IDHijo	varchar(30)	Identificador único del joven.

Todos los Identificadores son referencias a la tabla Usuario. Como clave principal tendremos ambos campos ya que un profesional puede estar repetido por atender a varios jóvenes. Esta tabla está realizada para facilitar la obtención de todos los jóvenes que atiende un único profesional. Se podría prescindir de ella si se optimiza la búsqueda de un mismo profesional en la tabla RelacionesHijos

4. **Tabla RelacionTutorHijo** Esta tabla almacena las relaciones entre los tutores y los jóvenes que están a su cargo. Tiene los siguientes campos:

Tabla 3.7: Tabla RelacionTutorHijo

Campo	Tipo de Dato	Descripción
IDTutor	varchar(30)	Identificador único para el primer tutor legal del uno de los jóvenes que hay en la base de datos.
IDHijo	varchar(30)	Identificador único del joven que tiene al tutor con IDTutor.

Todos los Identificadores son referencias a la tabla Usuario. Al igual que en la tabla anterior, como clave principal tendremos ambos campos ya que un tutor puede estar repetido por tener a su cargo a varios jóvenes. Esta tabla está realizada para facilitar la obtención de todos los jóvenes que están a cargo de un único tutor. Se podría prescindir de ella si se optimiza la búsqueda de un mismo tutor en la tabla RelacionesHijos por un cursor de búsqueda, por ejemplo.

5. **Tabla Variable:** Esta tabla almacena las posibles variables que los psicólogos contemplan para el estudio de las conductas de los jóvenes. Cada una de ellas tendrá un número indeterminado de items que podrán recuperarse a través de

la anterior tabla. Se guardará un conteo de cuantos items tiene, de manera informativa, a petición de los psicólogos.

Además tendrá los siguientes campos:

Tabla 3.8: Tabla Variable

Campo	Tipo de Dato	Descripción
IDVariable	varchar(30)	Identificador único de la variable.
NombreVariable	varchar(255)	Nombre descriptivo de la variable.
numUltimoItem	int	Número del último ítem asociado a la variable, que se usará para generar nuevos ítems secuenciales.

Como clave primaria tendrá el IDVariable.

6. **Tabla Pregunta:** Esta tabla almacena las preguntas que forman parte de los cuestionarios o tests que responderán los usuarios. Cada pregunta está asociada a una variable y un ítem específico, para que el sistema experto de los psicólogos pueda hacer la valoración pertinente teniendo en cuenta este único ítem.

Tendrá los siguientes campos:

Tabla 3.9: Tabla Pregunta

Campo	Tipo de Dato	Descripción
IDPregunta	varchar(30)	Identificador único para cada pregunta.
texto	int	Texto de la pregunta.
IDVariable	varchar(30)	Identificador de la variable a la que está asociada la pregunta (relacionada con la tabla Variable).
IDItem	varchar(30)	Identificador del ítem específico dentro de la variable al que la pregunta pertenece.

Cada pregunta será única y no podrá estar relacionada con dos variables o ítems distintos según la especificación de los psicólogos.

7. **Tabla Test:** Esta tabla almacena la información de los tests o cuestionarios asignados a los usuarios. Es el conjunto de preguntas que se enviará a los usuarios cuando el sistema experto considere oportuno. Cada test está asociado con un cliente (usuario) para el que se diseñó.

Tendrá los siguientes campos:

Tabla 3.10: Tabla Test

Campo	Tipo de Dato	Descripción
IDTest	varchar(30)	Identificador único del test.
IDCliente	varchar(30)	Identificador del usuario al que va dirigido el test.
Preguntas	longtext	Lista de preguntas incluidas en el test, almacenada en formato JSON. Cada una estará compuesta de otro JSON, en el que tendrá el id de la pregunta, del cual se podrán sacar la variable y el ítem, y la respuesta asignada.
Respondido	tinyint(1)	Indicador de si el test ha sido respondido.
FechaRespondido	datetime	Fecha en la que el test fue respondido.
FechaEnviado	datetime	Fecha en la que el test fue enviado al usuario.
FechaCaducidad	datetime	Fecha límite para responder el test ya que los psicólogos comentaron que no tenía sentido responder un cuestionario que no era relevante en la situación actual.

Cada test será único, no pudiéndose asignar el mismo test a varios usuarios distintos, requisito impuesto para poder recopilar y analizar mejor los datos obtenidos. Por ello el IDTest será la clave primaria. y no podrá estar relacionada con dos variables o ítems distintos según la especificación de los psicólogos. Como el IDCliente es una clave foránea, referenciado en la tabla Usuario, se usará como índice, para optimizar las consultas sobre qué tests están asignados a ese usuario.

8. **Tabla Item:** Esta tabla almacena información sobre cada ítem, incluyendo la IdVariable a la que está asociado y el nombre de la variable correspondiente. Se utilizará esta tabla para acceder de manera precisa al valor de la variable que corresponde a cada ítem.

Tendrá los siguientes campos:

Tabla 3.11: Tabla Item

Campo	Tipo de Dato	Descripción
IDItem	varchar(30)	Identificador del ítem específico dentro de la variable.

Campo	Tipo de Dato	Descripción
IDVariable	varchar(30)	Identificador de la variable a la que está asociada la pregunta (relacionada con la tabla Variable).
NombreVariable	varchar(255)	Nombre descriptivo de la variable.

Cada ítem en la tabla será único; no se permite que varios ítems apunten a la misma variable, ya que cada ítem proporciona información detallada y específica, que es crucial para evaluar al usuario que ha completado el cuestionario. Por esta razón, IDItem se definirá como clave primaria y IDVariable será una clave foránea que hace referencia a la tabla Variable.

3.6. Desarrollo del Servidor en Flask

En este proyecto, utilizamos **Flask** como framework para el desarrollo de la aplicación web. Esta decisión se basa en la simplicidad y facilidad de manejo que ofrece Flask en comparación con otros frameworks o servidores más robustos, como Django o Node.js.

Flask es un microframework para Python que sigue una filosofía minimalista: proporciona solo lo esencial para construir una aplicación web, dejando al desarrollador la libertad de añadir las extensiones o bibliotecas que necesite. A diferencia de **Django**, que es un framework más completo y viene con muchas funcionalidades preconfiguradas (como un sistema de autenticación, administración y ORM incorporado), Flask permite una mayor flexibilidad y control sobre cada componente de la aplicación. Esta simplicidad hace que Flask sea mucho más ligero y fácil de aprender, ideal para proyectos donde se necesita rapidez en el desarrollo sin comprometer la estructura del proyecto.

Por otro lado, si comparamos Flask con **Node.js** (una plataforma para ejecutar JavaScript del lado del servidor), la diferencia no solo radica en los lenguajes que utilizan (Python en Flask y JavaScript en Node.js). Flask es un microframework, lo que significa que es ligero y proporciona solo las herramientas básicas para construir aplicaciones web, mientras que Node.js es una plataforma más amplia, que soporta muchos frameworks y librerías adicionales. Además, Flask trabaja de manera sincrónica, procesando las solicitudes de una en una, mientras que Node.js utiliza un modelo asíncrono, lo que le permite manejar muchas conexiones al mismo tiempo de manera eficiente. Node.js también es más adecuado para aplicaciones en tiempo real, mientras que Flask es ideal para proyectos más pequeños o medianos.

En resumen, hemos optado por Flask debido a su:

- **Simplicidad:** Fácil de aprender y usar, lo que me ha permitido concentrarme en las funcionalidades específicas de la aplicación.
- **Flexibilidad:** Me ha permitido integrar solo las herramientas y bibliotecas que realmente necesito, sin la sobrecarga de características innecesarias.

- **Rapidez de desarrollo:** La ligereza de Flask facilita un ciclo de desarrollo ágil, sin tener que lidiar con la complejidad inherente de frameworks más grandes.

Estas características hacen de Flask la opción más adecuada para nuestro proyecto, donde priorizamos un desarrollo rápido, manejable y enfocado en las necesidades específicas, sin la complejidad que añaden otros frameworks.

3.6.1. Detalles del Desarrollo del Servidor

A continuación se detallan las partes fundamentales del desarrollo del servidor utilizando Flask:

3.6.1.1. Inicialización y Configuración del Servidor

En el script se importan varias bibliotecas y módulos para configurar una aplicación web con **Flask**. Se utiliza Flask para gestionar la aplicación web y **request** para manejar las solicitudes HTTP. También se emplea **jsonify** para facilitar la creación de respuestas en formato JSON. Se incorpora **mysql.connector** para interactuar con una base de datos MySQL, y **CORS** de **flask-cors** para permitir el intercambio de recursos entre diferentes dominios. Para la integración con Firebase, se emplean **firebase-admin** junto con sus componentes **credentials** y **messaging**. Finalmente, se usa **requests** para realizar solicitudes HTTP a otras APIs.

3.6.1.2. Configuración de Flask y CORS

Flask se configura inicialmente creando una instancia de la aplicación mediante **Flask(__name__)**. Además, se habilita **CORS (Cross-Origin Resource Sharing)**, lo que permite que el servidor acepte solicitudes desde diferentes dominios. Esto es especialmente útil cuando el frontend de la aplicación está alojado en un dominio diferente al del backend.

3.6.1.3. Configuración de la Base de Datos

Para interactuar con la base de datos MySQL, se configura un diccionario **db_config** que contiene las credenciales de acceso y el nombre de la base de datos. Esta configuración permite abrir conexiones a la base de datos en cualquier punto del código donde sea necesario realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar).

3.6.1.4. Integración con Firebase para Notificaciones Push

Una característica importante del servidor es su capacidad para enviar notificaciones push a dispositivos móviles utilizando Firebase Cloud Messaging (FCM). Para ello, se inicializa el SDK de Firebase con un archivo de credenciales que permite la autenticación y el acceso a los servicios de Firebase.

3.6.1.5. Gestión de las Rutas del Servidor

El servidor expone varias rutas a través de las cuales el cliente (por ejemplo, una aplicación móvil) puede interactuar con la base de datos y otras funcionalidades. A continuación, se describe cada una de las rutas principales, se aportara un diagrama de flujo y un diagrama de secuencia de cada una de ellas:

1. Verify password

Esta ruta tiene como objetivo verificar la autenticidad de un usuario en función de su nombre de usuario y contraseña. Recibe estos datos a través de una solicitud POST en formato JSON, recuperándolas con `request.get-json()`. Estos datos deben incluir `username` (nombre de usuario) y `password` (contraseña), que serán usados para verificar la identidad del usuario, es decir, que es un paciente que forma parte del programa Sivaria. Si alguno de estos campos falta, la función retorna una respuesta JSON con un mensaje de error. Para comprobar que los datos obtenidos son válidos, se intenta establecer una conexión a nuestra base de datos utilizando la configuración especificada en `db-config` y se crea un cursor para ejecutar la consulta que recupera la `password` del usuario con el `id` obtenido y además el tipo de usuario.

Obtenida ya la contraseña, se verifican las credenciales, si las contraseñas coinciden, se retorna una respuesta JSON con un mensaje de éxito, en caso contrario se retorna una respuesta JSON indicando “Contraseña inválida”.

Independientemente de si la operación fue exitosa o fallida, el bloque finally se asegura de que la conexión a la base de datos se cierre correctamente, liberando recursos.

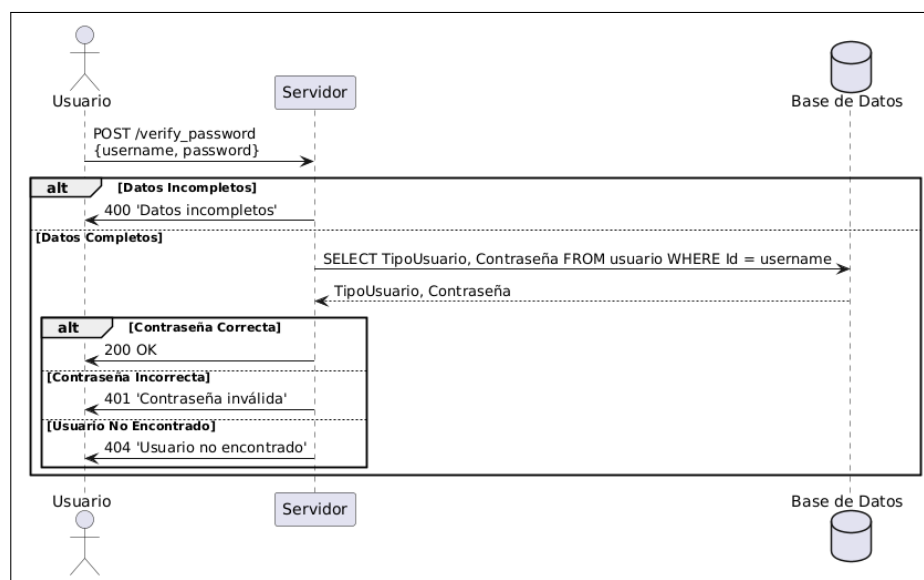


Figura 3.31: Diagrama de secuencia Verificar Contraseña

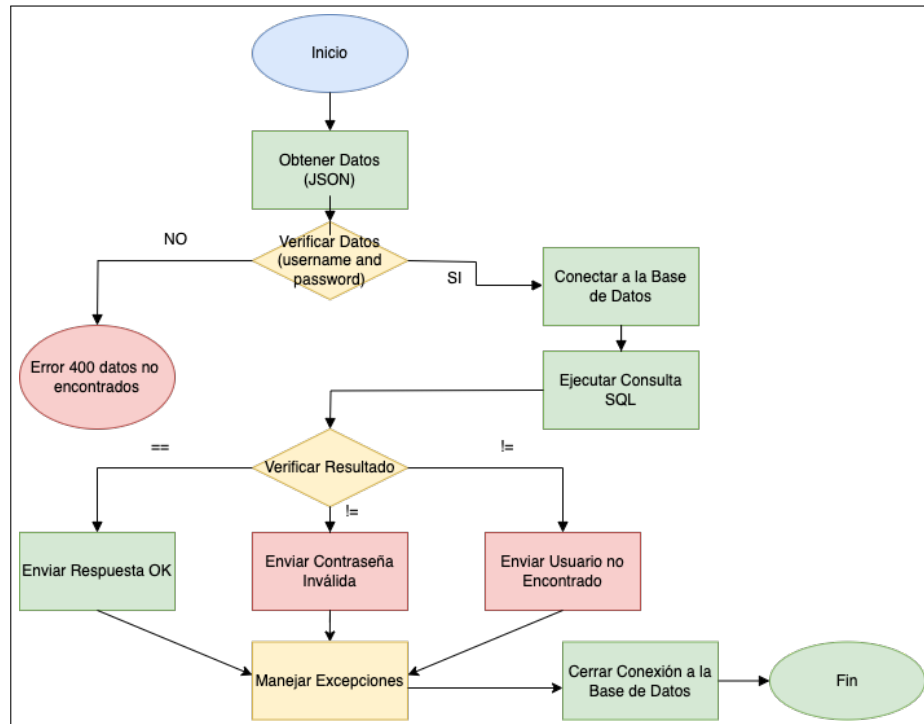


Figura 3.32: Diagrama de flujo Verificar Contraseña

2. Token

Esta ruta se encarga de recibir datos en formato JSON desde una solicitud POST dirigidas al endpoint `/token`. Verifica que contienen un token y un nombre de usuario (`username`), y luego procede a actualizar un registro específico en nuestra base de datos. El flujo del código incluye varios pasos, el primero, la validación de la información recibida, para ello se usa la función `handle-quiz` que comienza por recibir los datos de la solicitud en formato JSON a través de `request.get-json()`, como en la ruta especificada anteriormente. Estos datos deben incluir un token y un `username`, si dichos datos no estuviesen presentes para actualizar la base de datos correctamente, se devuelve un error.

Para realizar dicha actualización en la base de datos, se intenta establecer una conexión con la base de datos usando la configuración ya descrita y creando un cursor para ejecutar las consultas SQL pertinentes. Para confirmar la transacción se realiza `connection.commit()`, para asegurar que los cambios sean permanentes en la base de datos. Si la actualización es exitosa, se imprime un mensaje en la consola indicando la actualización exitosa. Si por el contrario ocurre algún error durante el proceso (por ejemplo, problemas con la conexión a la base de datos o con la ejecución de la consulta), se captura la excepción y se devuelve una respuesta JSON con el mensaje de error. Independientemente de si la operación fue exitosa o fallida, el bloque `finally` se asegura de que la conexión a la base de datos se cierre correctamente, liberando recursos.

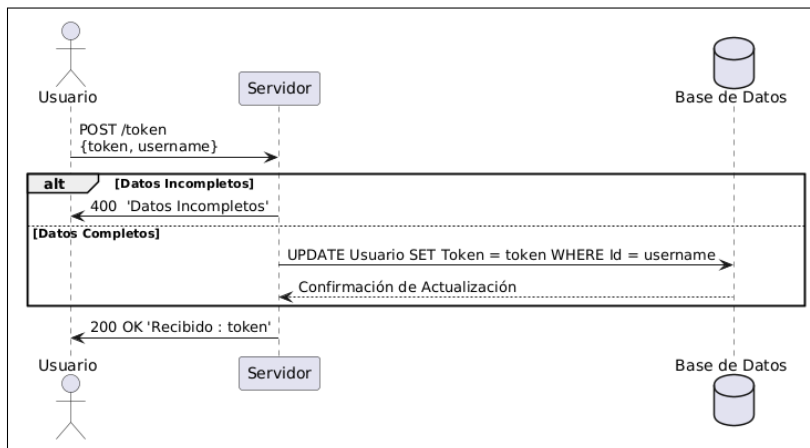


Figura 3.33: Diagrama de secuencia Token

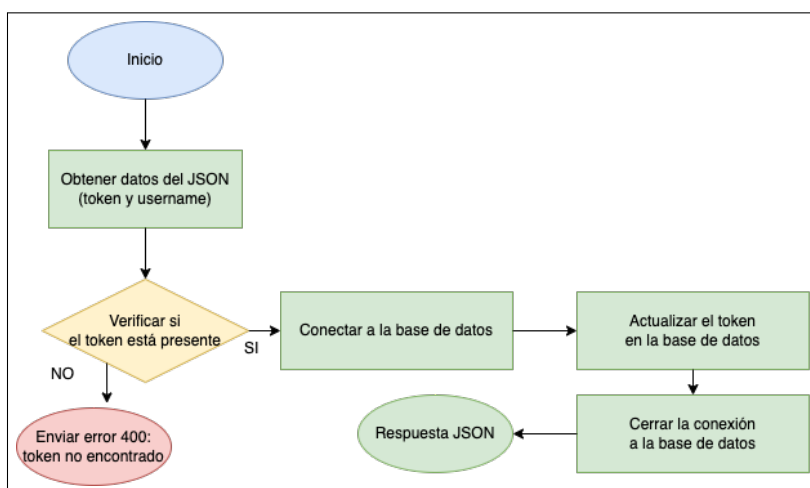


Figura 3.34: Diagrama de flujo Token

3. Guardar respuestas

Esta ruta se encarga de guardar las respuestas que el usuario ha realizado en la aplicación. Recibe estos datos a través de una solicitud POST en formato JSON, obteniéndolos con `request.get-json()`. Estos datos deben incluir `answers` (las respuestas obtenidas de todas las preguntas formuladas y enviadas en los test al usuario, junto a su `testID` y su `questionId`), que serán usados para guardar las respuestas que se han obtenido de dicho envío. Si falta este campo, la función retorna una respuesta JSON con un mensaje de error.

Para la validación de los datos, se establecerá una conexión a nuestra base de datos utilizando la configuración especificada en `db-config` y se crea un cursor para ejecutar la consulta que recupera las preguntas que se le han realizado al usuario, si no se recupera se devuelve un mensaje de error. Una vez recuperadas las preguntas guardamos todo lo obtenido en una variable. Esta variable será modificada dentro de un `for` en el que recorreremos cada pregunta añadiremos la respuesta del usuario.

Una vez esta variable contenga las preguntas y respuestas añadidas en el for, actualizaremos la tabla Test con la nueva información. Independientemente de si la operación fue exitosa o fallida, el bloque finally se asegura de que la conexión a la base de datos se cierre correctamente, liberando recursos.

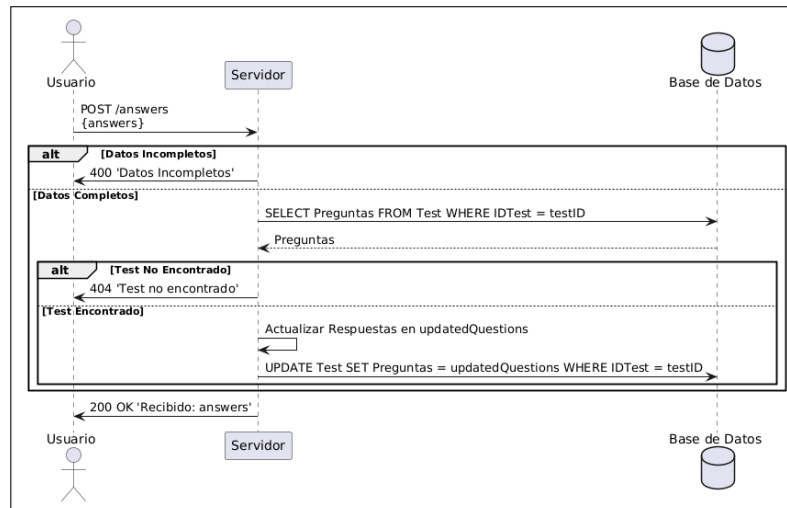


Figura 3.35: Diagrama de secuencia guardar respuestas

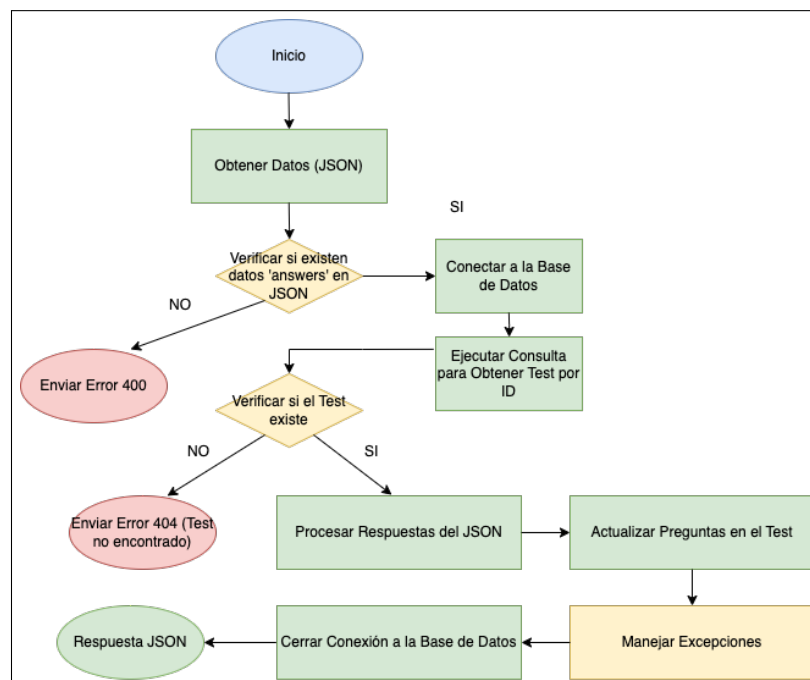


Figura 3.36: Diagrama de flujo guardar respuestas

4. Enviar preguntas

Esta ruta se encarga de enviar las preguntas que el usuario tiene que responder. Para ello necesita recuperar una serie de datos a través de una solicitud POST en formato JSON, obteniéndolas con `request.get-json()`. El dato que incluye es

el nombre del usuario que ha iniciado la sesión (username), necesitaremos este dato para recuperar el test que se le ha asignado a dicho usuario.

Para la validación de los datos, se establecerá una conexión a nuestra base de datos utilizando la configuración especificada en db-config y se crea un cursor para ejecutar la consulta que recupera IDTest que se le ha asignado a este usuario, correspondiente al día actual y que no haya sido respondido, si no se recupera se devuelve un mensaje de error. Una vez recuperada esta variable consultaremos las preguntas que se le ha asignado a este Identificador.

Tendremos que editar estas preguntas en un formato que nos permita enviarlo a la aplicación de dicho usuario junto a su IDTest. Independientemente de si la operación fue exitosa o fallida, el bloque finally se asegura de que la conexión a la base de datos se cierre correctamente, liberando recursos.

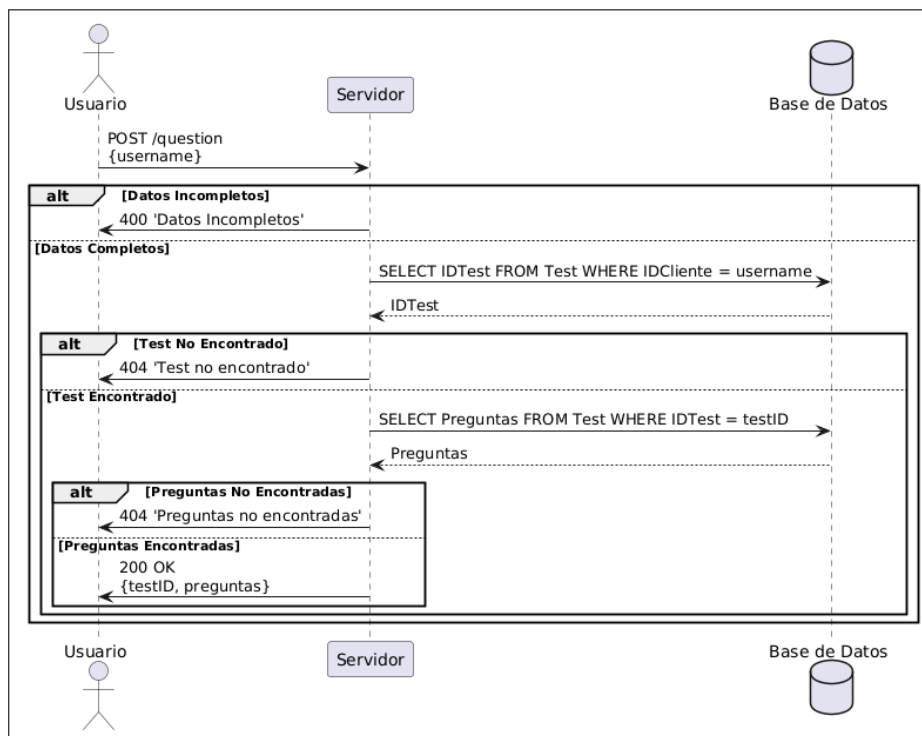


Figura 3.37: Diagrama de secuencia enviar preguntas

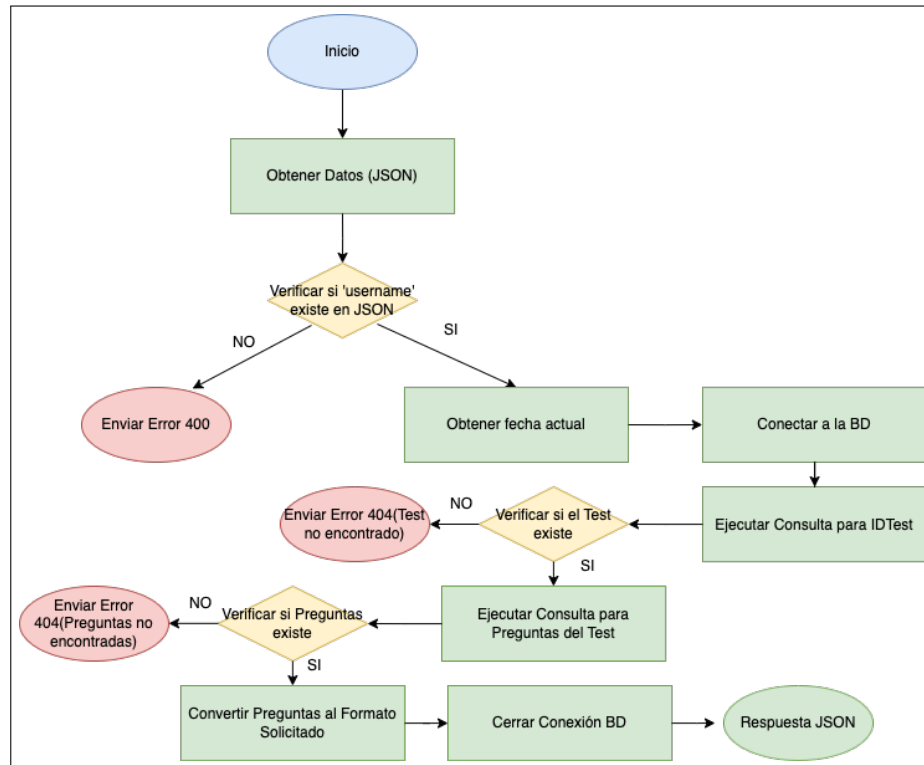


Figura 3.38: Diagrama de flujo enviar preguntas

5. Datos gráfica

Esta ruta se encarga de enviar los datos que necesita el usuario para ver el progreso en una gráfica de su paciente o hijo. Para ello necesita recuperar una serie de datos a través de una solicitud POST en formato JSON, obteniéndolos con `request.get-json()`. El dato que recuperado es el usuario que ha iniciado la sesión (`username`). Lo necesitaremos para recuperar los datos necesarios para que la visualización del diagrama de ese usuario.

Para la validación de los datos, se establecerá una conexión a nuestra base de datos utilizando la configuración especificada en `db-config` y se crea un cursor para ejecutar la consulta para recuperar las repuestas y los ítems a los que ha respondido ese usuario. Una vez recuperadas estas variables calcularemos la media de las respuestas, que pasaremos a un formato float, por cada ítem a la que se ha respondido.

Tendremos que adaptar estos datos en un formato que nos permita enviarlo a la interfaz del usuario y que esta sea capaz de generar la gráfica. Independientemente de si la operación fue exitosa o fallida, el bloque finally se asegura de que la conexión a la base de datos se cierre correctamente, liberando recursos.

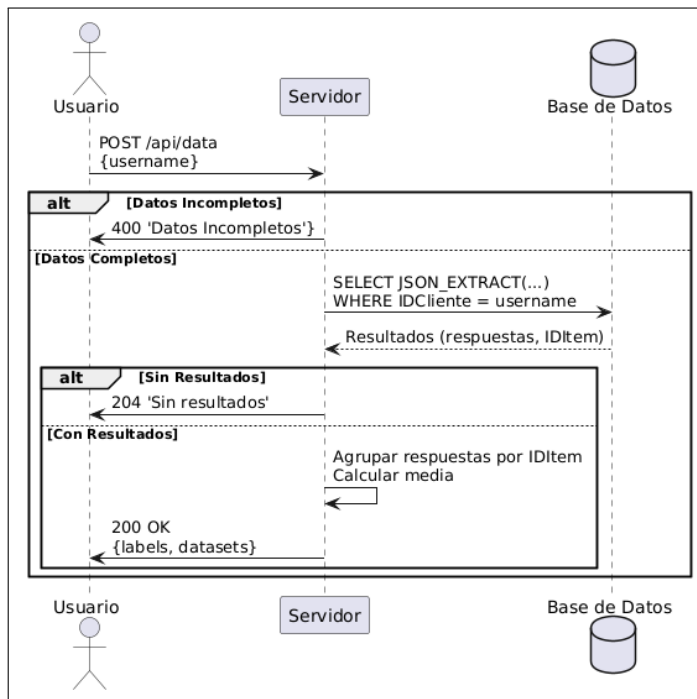


Figura 3.39: Diagrama de secuencia Datos gráfica

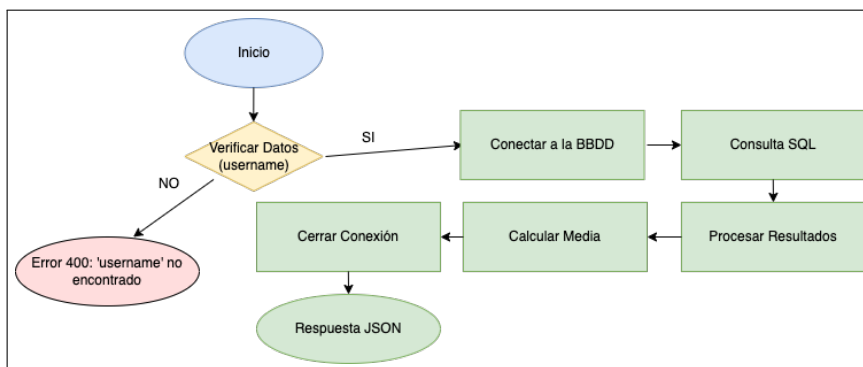


Figura 3.40: Diagrama de flujo Datos gráfica

6. Guardar Texto

Esta ruta se encarga de recibir datos en formato JSON desde una solicitud POST dirigidas al endpoint /guardar-texto. Verifica que contienen un nombre de usuario (username) y un texto (texto), y luego procede a actualizar la columna “sensaciones” de un registro específico en nuestra base de datos. El flujo del código incluye varios pasos. El primero es la validación de la información recibida, usando la función guardar-texto que comienza por recibir los datos de la solicitud en formato JSON a través de request.get-json(), como se especifica en la ruta mencionada. Estos datos deben incluir un username y un texto, y si dichos datos no estuviesen presentes para actualizar la base de datos correctamente, se devuelve un error indicando “Faltan datos”.

Para realizar dicha actualización en la base de datos, se intenta establecer una conexión con la base de datos usando la configuración ya descrita y creando

un cursor para ejecutar las consultas SQL pertinentes. La consulta actualiza la columna “sensaciones” en la tabla usuario para el usuario identificado por el Id. Para confirmar la transacción se utiliza `connection.commit()` para asegurar que los cambios sean permanentes en la base de datos. Si la actualización es exitosa, el código verifica que efectivamente se haya modificado algún registro; en caso de no haber modificaciones (es decir, si el usuario no es encontrado), se devuelve un error.

Si ocurre algún error durante el proceso, ya sea por problemas con la conexión a la base de datos o con la ejecución de la consulta, se captura la excepción y se devuelve una respuesta JSON con el mensaje de error. Independientemente de si la operación fue exitosa o fallida, el bloque `finally` se asegura de que la conexión a la base de datos se cierre correctamente, liberando recursos.

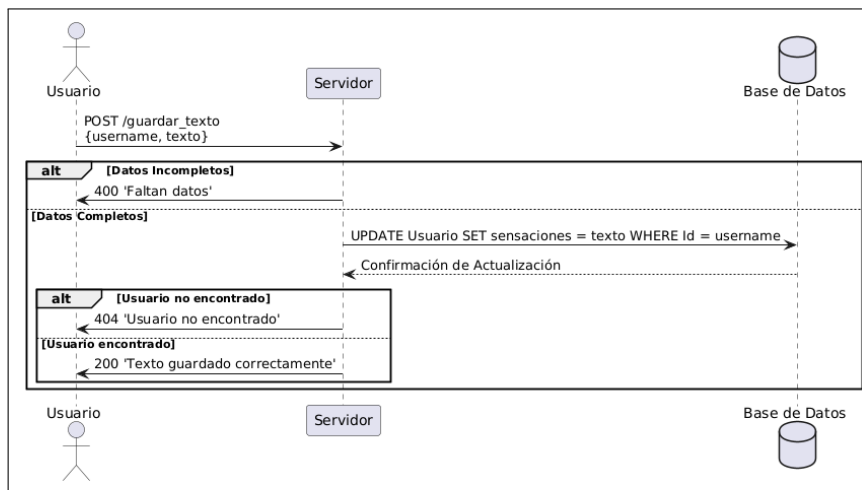


Figura 3.41: Diagrama de secuencia Guardar Texto

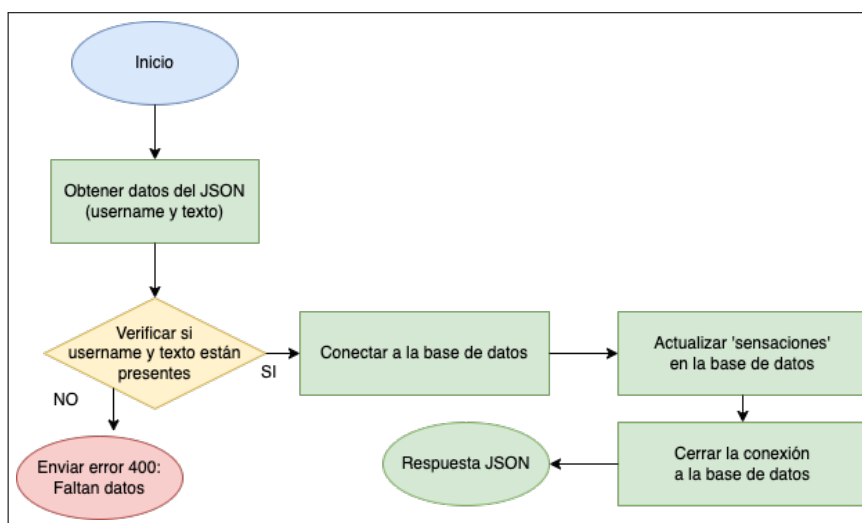


Figura 3.42: Diagrama de flujo Guardar Texto

7. Enviar Notificaciones

Esta ruta se encarga de enviar una notificación a un usuario específico. Para ello, recibe los datos necesarios a través de una solicitud POST en formato JSON, los cuales son procesados utilizando `request.get-json()`. La solicitud debe incluir un `IdUser` en el cuerpo JSON; si este dato no se encuentra presente, se devuelve un error indicando la ausencia de esta información esencial. Después, se intenta establecer una conexión con la base de datos utilizando la configuración predefinida `db-config` y se crea un cursor para ejecutar las consultas SQL necesarias. La consulta selecciona el token asociado con el usuario especificado por `IdUser`.

Si el usuario existe y se obtiene un token, se construye un cuerpo de notificación que incluye el token, un sonido predeterminado, un título, un mensaje en el cuerpo, y datos adicionales. Esta notificación es enviada a través de una solicitud POST al servicio de notificaciones Firebase, utilizando la URL del servicio correspondiente y encabezados que indican que el contenido es JSON.

Si la solicitud POST se completa con éxito, se devuelve una respuesta JSON que confirma el envío de la notificación y el token utilizado. En caso de que ocurra un error al intentar enviar la notificación, se captura y devuelve una respuesta JSON que incluye el mensaje de error. Además, si se produce una excepción durante la operación, se devuelve un error en formato JSON. Finalmente, independientemente del resultado, el bloque `finally` asegura la liberación de todos los recursos, cerrando la conexión a la base de datos y el cursor utilizados.

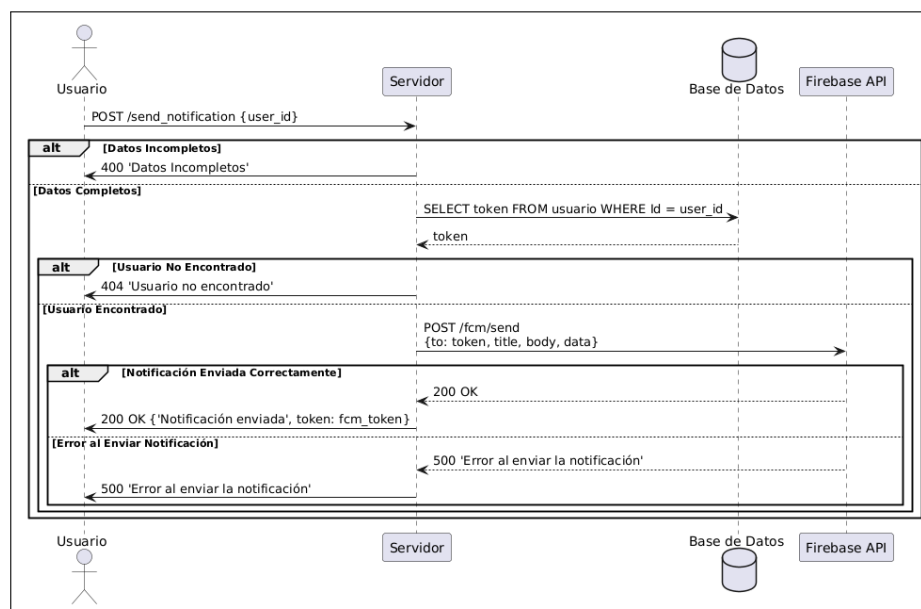


Figura 3.43: Diagrama de secuencia enviar notificaciones

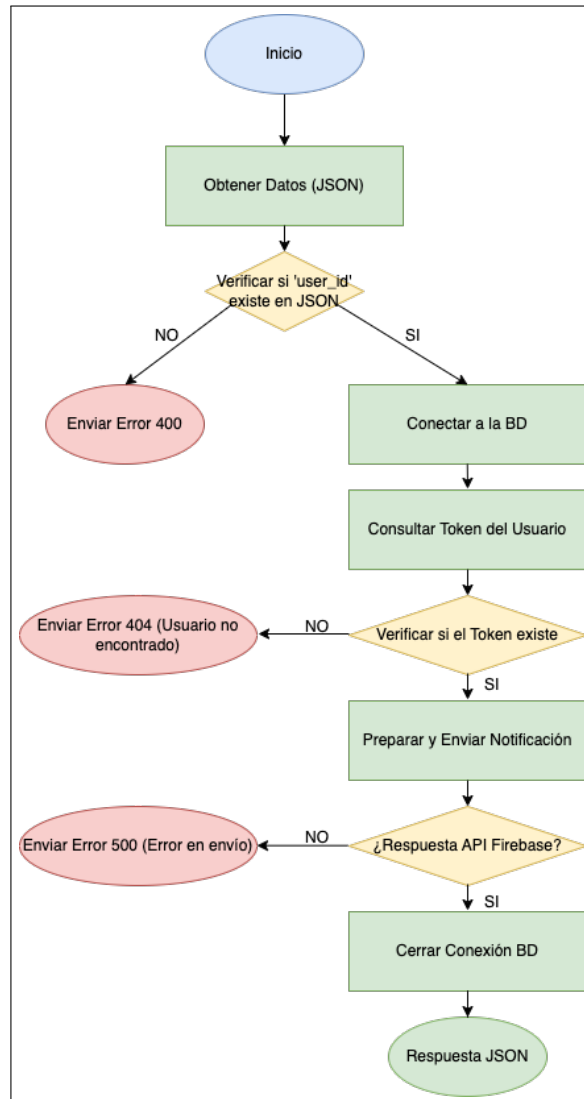


Figura 3.44: Diagrama de flujo enviar notificaciones

3.6.2. Ejecución del Servidor

Finalmente, el servidor Flask se ejecuta en modo de depuración, lo que permite la monitorización de errores y la recarga automática en tiempo real durante el desarrollo. Está configurado para escuchar en todas las interfaces de red (0.0.0.0) y en el puerto 443, que es el único puerto libre asignado, permitiendo que el servidor sea accesible de manera segura desde clientes remotos. Con respecto a este punto hemos tenido varios inconvenientes que solucionábamos y aumentaban su tamaño, en particular nos referimos a la conectividad de la UCM, que cuando empezamos no estaba restringida, más tarde solo fueron accesibles unos determinados puertos, esta información nos fue revelada gracias al equipo de técnicos del laboratorio. Después de hacer funcionar el servidor en el puerto 443, tuvimos que descargar una VPN en nuestros ordenadores. Arreglado esto, después de unos meses, nos fue imposible acceder a través de los puertos abiertos, por lo tanto decidimos que para seguir adelante debíamos migrar el servidor, y con él, la base de datos, a nuestros portátiles.

Volviendo a la funcionalidad del servidor, se ha añadido además, la configuración de `ssl-context=('cert.pem', 'key.pem')` para habilitar HTTPS, proporcionando una capa adicional de seguridad mediante la encriptación de las comunicaciones entre el servidor y los clientes.

3.7. Implementación y Funcionamiento de Notificaciones Push

En este apartado, detallaremos cómo hemos implementado las notificaciones push en nuestra aplicación utilizando **Firebase Cloud Messaging (FCM)**, Flask y React Native. La integración de estas tecnologías permite enviar notificaciones a los usuarios de la aplicación de manera efectiva y en tiempo real. A continuación, explicaremos el proceso que se ha seguido en detalle, cómo funciona cada componente y por qué elegimos esta configuración.

3.7.1. Configuración de Firebase Cloud Messaging (FCM)

Firebase Cloud Messaging (FCM) es un servicio que nos ayuda a enviar mensajes y notificaciones a las aplicaciones móviles y web. Esto significa que podemos hacer que nuestra aplicación envíe mensajes a los usuarios, como alertas o recordatorios, de manera eficiente. Para poner en marcha esta funcionalidad, seguimos algunos pasos clave.

1. **Creación del Proyecto en Firebase Console:** Primero, creamos un nuevo proyecto en la consola de Firebase, que es una herramienta en línea proporcionada por Google para gestionar servicios para aplicaciones. Dentro del proyecto, activamos la opción de Firebase Cloud Messaging. Esto nos permite enviar mensajes a dispositivos específicos, como teléfonos móviles, utilizando identificadores únicos llamados “tokens”.
2. **Obtención de Credenciales:**

Desde la consola de Firebase, obtuvimos una clave especial llamada “clave del servidor”. Esta clave es esencial porque nos permite autenticar las solicitudes de envío de notificaciones desde nuestro servidor. También descargamos un archivo de configuración. Dependiendo del tipo de dispositivo (Android o iOS), este archivo se llama ‘google-services.json’ o ‘GoogleService-Info.plist’. Este archivo contiene la información necesaria para que nuestra aplicación pueda comunicarse correctamente con Firebase y recibir las notificaciones que enviamos.

3.7.2. Integración en la Aplicación React Native

Como mencionamos antes, React Native es el marco que usamos para construir nuestra aplicación móvil. Para poder gestionar las notificaciones dentro de nuestra

aplicación, seguimos algunos pasos específicos que te explicamos a continuación:

1. **Instalación de Librerías:** Primero, necesitamos una herramienta especial que nos ayude a manejar las notificaciones en nuestra aplicación. En este caso, utilizamos una librería llamada `@react-native-firebase/messaging`. Esta librería facilita la conexión entre nuestra aplicación y un servicio llamado Firebase Cloud Messaging (FCM), que es el encargado de enviar los mensajes y notificaciones. Gracias a esta librería, podemos integrar fácilmente el servicio de notificaciones en nuestra aplicación.
2. **Configuración de Permisos y Tokens:** Para que nuestra aplicación pueda enviar notificaciones a los usuarios, primero necesitamos pedirles permiso. Esto significa que mostramos un mensaje a los usuarios pidiendo su aprobación para enviarles notificaciones. Una vez que los usuarios aceptan, nuestra aplicación necesita obtener un identificador único para cada dispositivo. Este identificador, llamado **token**, nos permite enviar notificaciones específicamente a cada dispositivo. Implementamos la lógica necesaria para guardar estos tokens y usarlos cuando queremos enviar notificaciones.

3.7.3. Configuración del Servidor Flask

Flask es una herramienta en Python que utilizamos para manejar la parte del servidor de nuestra aplicación. En esta sección, te explicamos cómo configuramos Flask para enviar notificaciones a los usuarios. A continuación, detallamos los pasos que seguimos:

1. **Configuración del Endpoint de Envío de Notificaciones:**

Creamos una “ruta” en el servidor Flask. Esta ruta es como una dirección a la que se envían las solicitudes para enviar notificaciones. Cuando se recibe una solicitud en esta ruta, toma los detalles necesarios, como el identificador del dispositivo (token) y el contenido del mensaje que queremos enviar.

2. **Envío de Notificaciones:**

Utilizamos una clave especial, llamada **clave del servidor**, que obtuvimos de Firebase. Esta clave es como una contraseña que nos permite autenticar nuestras solicitudes y asegurarnos de que solo nosotros podamos enviar notificaciones. Flask usa esta clave para enviar una solicitud a la API de Firebase Cloud Messaging (FCM). La solicitud se envía a una dirección web especial **<https://fcm.googleapis.com/fcm/send>** con la información que incluye el token del dispositivo, el título y el texto del mensaje.

3.7.4. Funcionamiento General y Ventajas de la Implementación

En nuestra aplicación, el sistema de notificaciones funciona de la siguiente manera:

Primero, cuando la aplicación React Native se inicia, solicita y recibe un token de dispositivo. Este token es un identificador único que permite a nuestro sistema enviar notificaciones específicamente a ese dispositivo.

Cuando es necesario enviar una notificación, como una actualización o un recordatorio, el servidor Flask recibe una solicitud para hacerlo. Flask entonces envía una petición a Firebase Cloud Messaging (FCM) para que entregue la notificación.

FCM se encarga de enviar la notificación al dispositivo correspondiente utilizando el token que se proporcionó. Así, el mensaje llega directamente al usuario de la aplicación.

Ventajas de Esta Implementación

- **Escalabilidad:** Firebase Cloud Messaging es muy escalable. Esto significa que podemos enviar notificaciones a muchos dispositivos a la vez sin necesidad de infraestructura adicional, lo cual es ideal para aplicaciones con una gran cantidad de usuarios.
- **Eficiencia:** La combinación de React Native y Flask ofrece una solución eficiente para manejar notificaciones en tiempo real. Cada tecnología aporta sus fortalezas para asegurar que las notificaciones se envíen y reciban de manera oportuna.
- **Flexibilidad:** Con esta configuración, podemos enviar notificaciones a usuarios individuales o a grupos de usuarios. Esto facilita la personalización de la comunicación y permite adaptar los mensajes a diferentes necesidades.

En resumen, con esta configuración hemos implementado un sistema de notificaciones push robusto y eficiente que mejora la experiencia del usuario y permite una comunicación efectiva con los usuarios de la aplicación. Esta solución no solo asegura que las notificaciones lleguen a su destino, sino que también se adapta a la escala y las necesidades de nuestra aplicación.

3.8. Funcionamiento de la aplicación móvil

En esta aplicación, creada para tres tipos de usuarios diferentes, nuestro objetivo es ofrecer una experiencia personalizada para tres roles principales: adolescente, padre y profesional. Cada uno de estos usuarios utilizará la aplicación de manera distinta, según sus necesidades específicas. El objetivo es ofrecer una plataforma donde los adolescentes puedan completar cuestionarios y enviar sus respuestas voluntarias, los padres pueda realizar las mismas funcionalidades y visualizar el progreso de sus hijos, y los profesionales únicamente visualizar el progreso de sus pacientes.

A lo largo de esta demostración, presentaremos cómo cada uno de estos roles interactúa con la aplicación, mostrando las características específicas diseñadas para cada perfil.

3.8.1. Interfaz Profesional

En la pantalla inicial del profesional, el usuario deberá iniciar sesión para acceder a la aplicación. Una vez autenticado, será redirigido a la pantalla principal del profesional, donde tendrá acceso a dos funcionalidades principales: ver progresos y cerrar sesión.

Al seleccionar la opción de “Ver Progresos”, el profesional será llevado a una interfaz que muestra un gráfico detallado con el progreso de los adolescentes en los cuestionarios. Este gráfico proporciona una visión general del rendimiento y el avance de los adolescentes bajo su supervisión, permitiendo al profesional monitorear el progreso de manera eficiente.

El profesional también encontrará un botón de “Cerrar sesión” en la pantalla principal, que le permitirá finalizar su sesión y regresar a la pantalla de inicio de sesión.

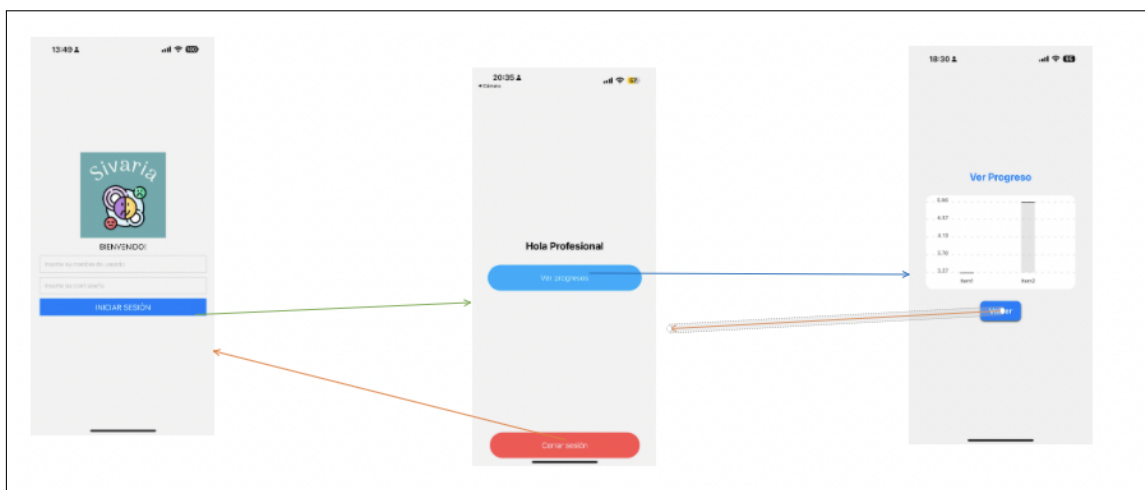


Figura 3.45: Funcionamiento aplicación Profesional

3.8.2. Interfaz Adolescente

En la pantalla destinada al adolescente, el usuario primero deberá iniciar sesión. Una vez autenticado, será redirigido a la pantalla principal del adolescente, donde tendrá acceso a dos funcionalidades clave: realizar un cuestionario y enviar respuestas voluntarias. Además, contará con un botón de “Cerrar sesión”, que lo devolverá a la pantalla de inicio de sesión.

Al seleccionar la opción de “Realizar cuestionario”, el usuario accederá a la pantalla correspondiente, donde podrá responder las preguntas del cuestionario. Estas preguntas pueden ser de tres tipos: selección única, selección múltiple y barra de estado. Durante la realización del cuestionario, se dispondrá de un botón de “Retroceder” para volver a la pregunta anterior, y un botón de “Avanzar” para pasar a la siguiente. Si no hay más preguntas, se accederá a la pantalla final del cuestionario.

En esta pantalla final, el usuario tendrá la opción de modificar las respuestas seleccionando el botón de “Modificar Respuestas”, o bien enviar las respuestas presionando el botón de “Enviar Respuestas”.

Por otro lado, al seleccionar la opción de “Enviar Respuestas Voluntarias”, el usuario será dirigido a una pantalla donde podrá redactar su situación personal o comentario. En esta pantalla, si el usuario decide no enviar nada, puede presionar el botón de “Cancelar” para salir sin enviar información, o utilizar el botón de “Enviar” si desea enviar sus respuestas voluntarias.

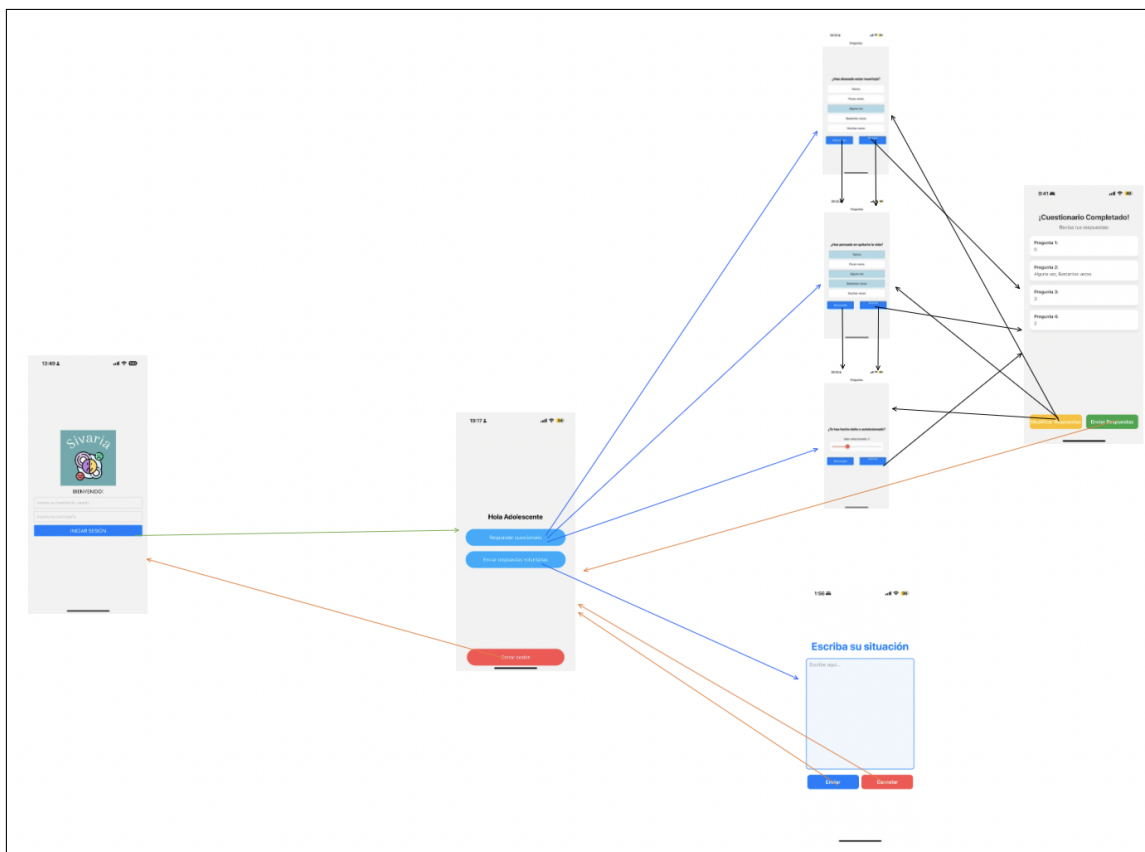


Figura 3.46: Funcionamiento aplicación Adolescente

3.8.3. Interfaz Padre

En la pantalla principal del padre, el usuario deberá iniciar sesión primero. Una vez autenticado, será llevado a la pantalla principal del padre, desde donde podrá acceder a tres funciones principales: ver progresos, realizar un cuestionario y enviar respuestas voluntarias. También encontrará un botón de “Cerrar sesión”, que le permitirá regresar a la pantalla de inicio de sesión.

Al elegir la opción de “Ver Progresos”, el padre será dirigido a una interfaz que presenta un gráfico detallado con el progreso de su hijo en los cuestionarios. Esta gráfica ofrece una visión clara del rendimiento y el avance del hijo a lo largo del

tiempo. En esta misma interfaz, el padre encontrará un botón de “Volver”, que le permitirá volver a la pantalla principal.

Si el padre selecciona “Realizar cuestionario”, será llevado a la pantalla del cuestionario, donde podrá responder preguntas de tres tipos: selección única, selección múltiple y barra de estado. Durante el cuestionario, el usuario tendrá la opción de usar el botón de “Retroceder” para regresar a la pregunta anterior y el botón de “Avanzar” para continuar a la siguiente. Si no hay más preguntas, el padre será dirigido a la pantalla final del cuestionario.

En la pantalla final, podrá modificar las respuestas haciendo clic en el botón de “Modificar Respuestas”, o enviar las respuestas presionando el botón de “Enviar Respuestas”.

Finalmente, si selecciona “Enviar Respuestas Voluntarias”, el padre será llevado a una pantalla donde podrá escribir un comentario o situación personal. En esta pantalla, puede optar por cancelar y salir sin enviar información, o enviar sus respuestas voluntarias si así lo desea.

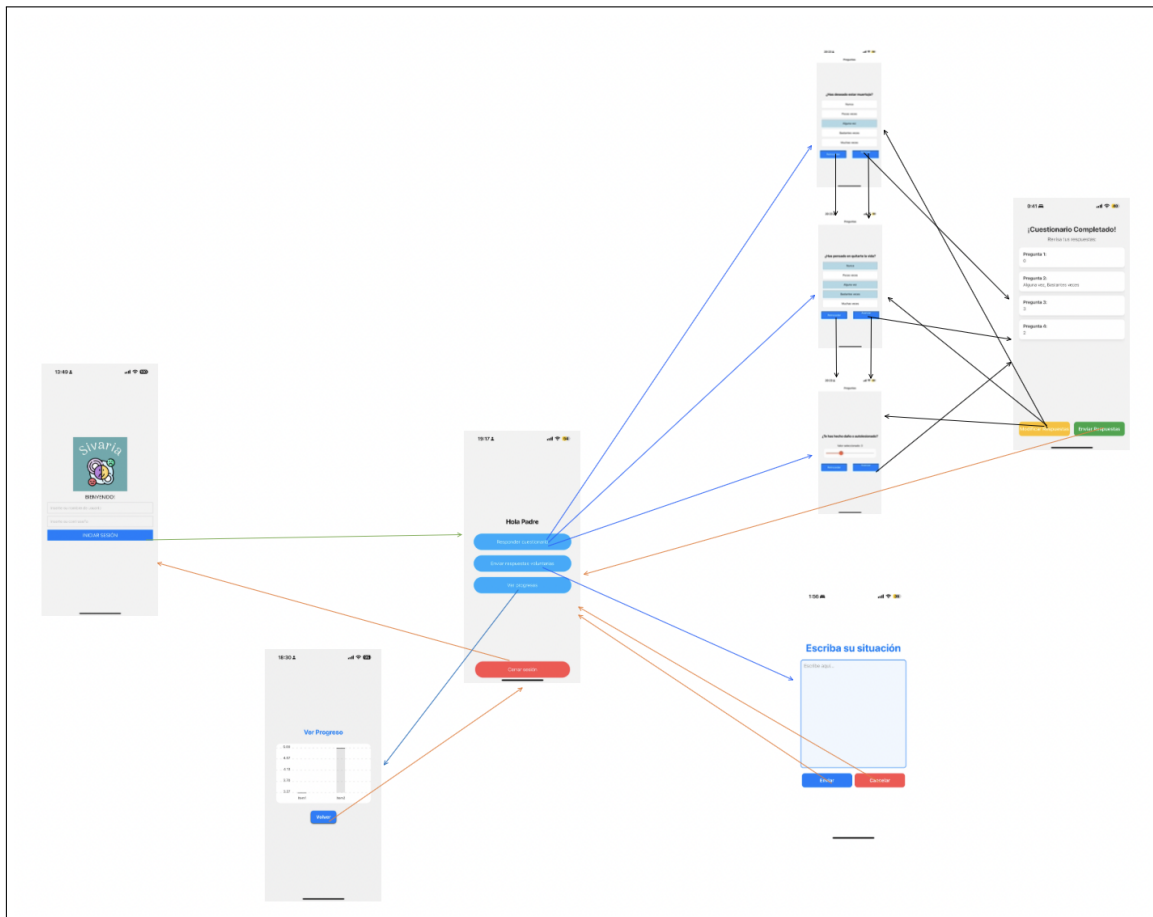


Figura 3.47: Funcionamiento aplicación Padre

Conclusiones y Trabajo Futuro

El desarrollo de nuestra aplicación ha seguido un proceso muy estructurado desde el diseño inicial hasta la implementación final, logrando una plataforma funcional y efectiva que responde a las necesidades de adolescentes, padres y profesionales. A continuación, se presentan las conclusiones alcanzadas y las áreas potenciales para futuros desarrollos.

4.1. Conclusiones

- **Uso de tecnologías para el desarrollo:** El uso de múltiples lenguajes de programación como JavaScript, Python y SQL, ha puesto a prueba nuestra capacidad de aprender a usar nuevas tecnologías e integrarlas según la petición de alguien externo, aprendiendo a investigar soluciones cuando se presentaban problemas, como por ejemplo, a la implementación de las notificaciones, la investigación en foros y comunidades encontrando la biblioteca de Firebase y aprendiendo a crear un proyecto y usarlo.
- **Eficiencia en el desarrollo:** Pudimos realizar la transición del prototipo a la versión final de la aplicación de manera eficiente, consiguiendo la implementación efectiva de las funcionalidades esenciales. El diseño de la pantalla de inicio de sesión, la interfaz del cuestionario, y las pantallas de visualización de progreso y envío de respuestas voluntarias se ajustaron según las especificaciones iniciales modificadas durante el proceso, asegurando una experiencia de usuario según las expectativas de los solicitantes.
- **Optimización de funcionalidades:** La decisión de simplificar la aplicación, basada en la retroalimentación de los psicólogos, fue reflejo de nuestra adaptación a cambios según su opinión. Al eliminar la funcionalidad de agendar citas y la opción de avisar a los padres, pudimos enfocar los esfuerzos en las características que realmente aportan valor a los usuarios, como la pantalla para realizar los test y la gestión de respuestas voluntarias, y en especial, al envío de notificaciones, que era crucial para el proyecto.

- **Reflexión sobre la experiencia de trabajo con los psicólogos:** Trabajar en estrecha colaboración con los psicólogos de la UNED resultó ser un aspecto fundamental para el entendimiento del proyecto. Después de estar trabajando codo a codo con ellos, nos quedamos con un sabor agrisado ya que, cuando más motivados estábamos, al principio del proyecto, era cuando menos definido estaba el proyecto y tuvimos que pedir que asentasen algunos términos en algunas de las reuniones para poder seguir trabajando, como por ejemplo qué tipos de preguntas habría o cómo funcionarían los ítems de las variables, entre otras cosas. Por el otro lado, su retroalimentación en cuanto al diseño permitió ajustar la interfaz, asegurando que fueran correctas y útiles para los adolescentes. Esta experiencia ha sido muy enriquecedora ya que es un acercamiento al mundo laboral donde dependeremos de nuestros clientes.

4.2. Trabajo Futuro

- **Validación con usuarios:** Como una tarea pendiente, es imprescindible con pruebas de usuarios reales para validar lo ya implementado para ver si efectivamente la implementación es correcta, funciona como se esperaba y es aceptada por el público al que se dirige, los adolescentes. De hecho, considerábamos lanzar la app para algunos usuarios ya pertenecientes al proyecto para probarla, pero por algunos inconvenientes, no pudimos.
- **Optimización continua de la interfaz y usabilidad:** A pesar de los avances realizados, siempre hay margen para mejorar la interfaz de usuario y la usabilidad. En futuros desarrollos se podrían llevar a cabo estudios de usabilidad más exhaustivos y aceptar comentarios de los usuarios que usan la aplicación para identificar posibles áreas de mejora. La integración de características adicionales, basadas en los comentarios de los usuarios y las tendencias emergentes, podría hacer que la aplicación sea aún más agradable.
- **Expansión de funcionalidades:** Considerar la adición de nuevas funcionalidades que podrían enriquecer la experiencia del usuario es un paso importante. Por ejemplo, se podrían explorar herramientas avanzadas para el monitoreo del bienestar emocional como mediciones del estrés con dispositivos físicos, integración con servicios de telemedicina o características personalizadas para un seguimiento más detallado de la salud.
- **Implementación de tecnologías emergentes:** La incorporación de tecnologías emergentes, como la inteligencia artificial y el aprendizaje automático, podría mejorar significativamente las capacidades de la aplicación. Por ejemplo, la implementación de chatbots para asistencia instantánea podrían mejorar la interacción del usuario.
- **Exploración de nuevas plataformas y dispositivos móviles:** A medida que avanzamos, la ampliación de la compatibilidad de la aplicación a diferentes dispositivos móviles será un enfoque clave. Adaptar la aplicación para asegurar

su funcionalidad óptima en una variedad de dispositivos móviles y tamaños de pantalla puede aumentar la la conveniencia para los usuarios. La optimización para plataformas móviles garantizará que la aplicación se perpetúe a lo largo de los años.

- **Mejoras en la seguridad y privacidad de los datos:** Con el objetivo de proteger la información sensible de los usuarios, se plantea como futura mejora la implementación de cifrado de datos en el servidor. Este cifrado garantizaría que los datos almacenados estén protegidos frente a accesos no autorizados. Además, se priorizará la transmisión segura de los datos entre la aplicación y el servidor mediante protocolos de comunicación encriptados, como HTTPS, asegurando que la información enviada por los usuarios esté siempre protegida.

4.3. Conclusión General

El proyecto ha sido un costoso éxito en la implementación de una aplicación funcional y útil que satisface las necesidades básicas de los usuarios. La experiencia adquirida durante el desarrollo y el feedback recibido por los psicólogos en las reuniones y por nuestro tutor, ofrecen una base sólida para futuras mejoras. Con un enfoque continuo en la optimización, expansión y adaptación a nuevas tecnologías, la aplicación tiene el potencial de evolucionar y ofrecer un valor aún mayor a sus usuarios en los próximos años. La capacidad para adaptarse a las necesidades cambiantes y explorar nuevas oportunidades garantizará que la aplicación siga siendo relevante y efectiva en el contexto de la gestión del bienestar.

Introduction

Motivation

Currently, suicide and self-harming behaviors have become critical health and social issues, affecting individuals of all ages, genders, and socio-economic backgrounds (Baños-Chaparro, 2024), (Organization et al., 2021). These forms of psychological and emotional suffering not only have a devastating impact on the lives of those who experience them but also have profound repercussions on their families and society as a whole.

According to (Organización Mundial de la Salud (OMS), 2014), in 2014, it was estimated that over 800,000 people lose their lives to suicide each year, which translates to one person every 40 seconds. By 2017, the number of daily deaths among young people had risen to over 3,000 per day, totaling 1.2 million young deaths annually (Organización Mundial de la Salud (OMS), 2017), making it the second leading cause of accidental death among individuals aged 15 to 29.

This alarming statistic prompted us to reflect on the measures we could implement to prevent these tragedies, as we believe mental health should be a priority across all areas to ensure that each individual has the support and resources needed to maintain their psychological well-being.

To prevent suicide and self-harming behaviors, it is necessary to ensure that people struggling with suicidal thoughts have access to appropriate and timely care. This is where the idea of the SIVARIA project fits, which proposes the implementation of a solution. The SIVARIA project, proposed by a team of psychologists from UNED, is based on an expert system that, by evaluating responses to selected questions, assesses the severity of the adolescent's situation. Our motivation for developing this application is to collect these responses and thus support this tool, which we believe will be a significant preventive measure.

Objectives

The main objective of this thesis is to develop a mobile application that helps assess and manage the risk of self-harming behaviors, both suicidal and non-suicidal, in adolescents aged 12 to 21. This tool will be based on an Expert System designed by psychologists from UNED, who will provide the scientific foundation for its functionality.

The application will allow for real-time measurements of how the user has felt over a given period. These measurements will be conducted through tests specifically created for the user, with a series of multiple-choice questions. These tests will be sent at a specific time of day, once or several times a week, with the crucial aim of capturing their emotions immediately and facilitating intervention if necessary. The questions must be easy to answer to maintain the key characteristic of these measurements: immediacy.

Additionally, it will be necessary to create a system to securely store and manage all collected information so that psychologists can analyze the data and properly feed the Expert System with the measurements taken. All of this will be done while ensuring user confidentiality.

To carry out this thesis, various technological tools will be required for the development of the application, the creation of a server to support its functionality, and the creation and management of a database.

Work Plan

For the work plan, we followed a guideline set at the beginning of the course, dividing the project into three milestones and attempting to adhere to them strictly.

In the first milestone, our primary goal was to write the project's motivation, organize ourselves, and study the current situation related to the project topic to understand our starting point and what had already been done by others. Following this, we needed to develop a technological prototype, addressing any potential technological risks we anticipated, determining the technologies to use by conducting a preliminary study of possible options with their respective pros and cons. The deliverables for this milestone included the code for the basic prototype and a partial report covering the introduction, objectives, and state of the art.

In the second milestone, we focused on significantly advancing the development of the technological prototype, aiming to achieve 85% completion of the code. Tasks included continuing coding and writing the central chapters of the report. The deliverables for this milestone were the prototype code (85% completed) and a partial report covering the central chapters of the project.

In the third and final milestone, the goal was to complete the development of the technological prototype. Final tasks included concluding coding and writing the results and conclusions in the report. The deliverables for this milestone included

the prototype code (100% completed) and the final report, which contained the results and conclusions obtained throughout the project.

In addition to the deliverables for each milestone, regular meetings were scheduled with the thesis advisor to evaluate progress and make adjustments as necessary.

It was also essential to have meetings with the UNED psychology team to ensure that our prototype met their needs and to make any necessary adjustments to ensure we met the requirements.

Structure of the Report

1. Introduction

This section of the report presents the project introduction, starting with the motivation behind it, highlighting the importance of addressing suicide and self-harming behaviors in adolescents. It then describes the objectives of the work, focusing on the development of the described mobile application. Additionally, it details the work plan, divided into three key milestones, which guides the project's development and meetings with the team. Finally, it explains the structure of the report, providing a clear outline of the sections that make up the document.

2. State of the Art

This section presents an analysis of applications or projects similar to ours. It examines the most notable features of these existing solutions, identifying both their strengths and areas for improvement. This comparative analysis serves as a basis for understanding how our project positions itself within the current context of this field.

3. Description of the Work

This section details the design and development process of our application. It describes design decisions, technologies used, and the steps taken from the conception of the idea to the final implementation. This section helps understand the approach we have chosen and the challenges overcome during development.

4. Conclusions and Future Work

This section provides a final evaluation of the project based on the results obtained to measure the success of the application according to the initially set objectives.

It discusses potential improvements and future lines of work that could derive from the current project to expand or refine the application, addressing limitations and exploring new functionalities that have emerged during the project.

5. Bibliography

At the end of the report, a list of all consulted and cited bibliographic sources is included. This section ensures proper attribution of ideas and prior work, and provides additional references for further exploration of the topics covered.

Conclusions and Future Work

The development of our application has followed a very structured process from initial design to final implementation, resulting in a functional and effective platform that meets the needs of adolescents, parents, and professionals. The following presents the conclusions reached and potential areas for future development.

Conclusions

- **Use of Development Technologies:** The use of multiple programming languages such as JavaScript, Python, and SQL tested our ability to learn and integrate new technologies as requested by external parties. We learned to research solutions when problems arose, such as implementing notifications, by exploring forums and communities, finding the Firebase library, and learning how to create and use a project.
- **Development Efficiency:** We successfully transitioned from the prototype to the final version of the application efficiently, achieving effective implementation of essential functionalities. The design of the login screen, questionnaire interface, progress visualization screens, and voluntary response submission screens were adjusted according to the modified initial specifications, ensuring a user experience that met the expectations of the stakeholders.
- **Functionality Optimization:** The decision to simplify the application, based on feedback from psychologists, reflected our adaptation to changes based on their input. By removing the appointment scheduling feature and the option to notify parents, we were able to focus efforts on features that truly add value for users, such as the test-taking screen and the management of voluntary responses, especially the critical notification feature for the project.
- **Reflection on Working with Psychologists:** Working closely with the UNED psychologists proved to be a crucial aspect of understanding the project. After working closely with them, we experienced a mix of emotions; while we were highly motivated at the start of the project, it was at this time that the project was least defined, and we had to request clarification on some terms

in meetings to proceed, such as the types of questions or how the variables would function, among other things. On the other hand, their feedback on the design allowed us to adjust the interface, ensuring it was correct and useful for adolescents. This experience has been very enriching as it provides a glimpse into the professional world where we will depend on our clients.

Future Work

- **User Validation:** As an outstanding task, it is essential to conduct real user testing to validate the implemented features to see if they function as expected and are accepted by the target audience, adolescents. We considered launching the app for some project-related users for testing, but due to some constraints, we were unable to proceed.
- **Continuous Optimization of Interface and Usability:** Despite the progress made, there is always room for improving user interface and usability. Future developments could involve conducting more comprehensive usability studies and accepting feedback from users to identify potential areas for improvement. Integrating additional features based on user feedback and emerging trends could make the application even more user-friendly.
- **Expansion of Features:** Considering the addition of new features that could enhance user experience is an important step. For example, exploring advanced tools for emotional well-being monitoring, such as stress measurements with physical devices, telemedicine integration, or personalized features for more detailed health tracking, could be beneficial.
- **Implementation of Emerging Technologies:** Incorporating emerging technologies, such as artificial intelligence and machine learning, could significantly enhance the application's capabilities. For instance, implementing chatbots for instant assistance could improve user interaction.
- **Exploration of New Platforms and Mobile Devices:** As we move forward, expanding the application's compatibility with different mobile devices will be a key focus. Adapting the application to ensure optimal functionality across a variety of mobile devices and screen sizes can increase convenience for users. Optimization for mobile platforms will ensure the application's longevity over the years.
- **Improvements in data security and privacy:** To protect users' sensitive information, a future enhancement involves implementing data encryption on the server. This encryption will ensure that stored data is safeguarded against unauthorized access. Additionally, secure data transmission between the app and the server will be prioritized using encrypted communication protocols, such as HTTPS, ensuring that user-submitted information is always protected.

General Conclusion

The project has been a valuable success in implementing a functional and useful application that meets the basic needs of users. The experience gained during development and the feedback received from psychologists in meetings and from our advisor provide a solid foundation for future improvements. With a continued focus on optimization, expansion, and adaptation to new technologies, the application has the potential to evolve and offer even greater value to its users in the coming years. The ability to adapt to changing needs and explore new opportunities will ensure that the application remains relevant and effective in the context of well-being management.

Contribuciones Personales

Pese a la colaboración de ambos participantes a lo largo del proyecto, entre ellos, las contribuciones de ambos están bien delimitadas, debido a que cada uno se ha centrado en una parte de la implementación.

Li Jie Chen Chen

Desarrollo de funciones del servidor

Mi contribución principal al servidor se ha centrado en el desarrollo de cuatro funciones que sirven para el manejo de datos, la autenticación de usuarios, envío de preguntas al usuario y almacenar texto voluntarios.

- **Función de recuperación de datos del usuario:** Desarrollé una API que permite recuperar los datos de las respuestas de los cuestionarios completados por un usuario específico. Esta función se encarga de procesar los datos almacenados en la base de datos, calcular las medias de las respuestas para cada ítem, y devolver estos datos en un formato JSON que puede ser utilizado por la aplicación para generar gráficos. Esta API es esencial para la visualización del progreso y el análisis de los resultados de los cuestionarios en la aplicación.
- **Función de enviar preguntas al usuario :** He implementado una API que facilita el envío de preguntas que los usuarios deben responder. Esta función envía al usuario un conjunto de preguntas en formato JSON, lo cual es fundamental para que el usuario pueda completar los cuestionarios asignados en la aplicación. Esta API es crucial para el correcto funcionamiento de la aplicación, asegurando que los usuarios reciban y respondan a las preguntas que se les asignan de manera efectiva.
- **Función de verificación de contraseña:** Implementé una API destinada a verificar las credenciales de los usuarios durante el proceso de inicio de sesión. Esta función recibe el nombre de usuario y la contraseña proporcionados por el usuario, y valida si coinciden con los registros almacenados en la base de datos. La correcta implementación de esta función garantiza la seguridad y

autenticidad de las sesiones de usuario, permitiendo un acceso seguro a la aplicación.

- **Función de almacenamiento de texto enviado por el usuario:** Añadí una API adicional que permite a los usuarios enviar texto desde la aplicación para ser almacenado en el servidor. Esta función asegura que cualquier entrada de texto del usuario, como comentarios o respuestas abiertas, se guarde de manera segura en la base de datos para su posterior análisis.

Colaboración en el desarrollo y en la investigación

Además de las tareas específicas del servidor, he trabajado estrechamente con mi compañera para asegurar una integración efectiva de nuestras respectivas áreas del proyecto. Nuestra colaboración ha incluido:

- **Integración de componentes:** He colaborado en la integración de las funciones del servidor con la aplicación en React, asegurando que los datos se intercambien correctamente y que la interfaz de usuario se actualice de manera coherente con la información recibida del servidor.
- **Prototipo en Figma:** Nos dividimos mi compañera y yo la mitad de las pantallas del prototipo de la aplicación para desarrollarlo en Figma. El prototipo nos sirvió como una herramienta clave para la validación temprana del diseño y la usabilidad de la aplicación. También me encargué de grabar un vídeo para mostrar cómo funciona el flujo de navegación entre las pantallas.
- **Investigación y reuniones con el cliente:** Participé en la investigación de aplicaciones similares para entender mejor las necesidades del usuario y las tendencias del mercado. Además, asistí a reuniones con el cliente para recibir retroalimentación y ajustar el prototipo en función de sus necesidades y expectativas.

Desarrollo de la aplicación en React

En el desarrollo de la aplicación, he liderado la implementación de la mayoría de las funcionalidades del prototipo final, con la ayuda inestimable de mi compañera, asegurando que cada componente y característica se desarrollara y funcionara según lo previsto y lo que hemos acordado con los psicólogos. Este proceso incluyó:

- **Implementación de funcionalidades del prototipo:** Me encargué de traducir las funcionalidades definidas en el prototipo final a una aplicación funcional en React. Esto implicó la creación de componentes de interfaz, la integración con el backend, y la implementación de la lógica necesaria para asegurar que la aplicación operara de manera coherente con los requisitos del prototipo. Además, realicé una investigación exhaustiva sobre las funcionalidades de React Native para desarrollar los cuestionarios, la pantalla de inicio de sesión,

las gráficas y el envío de datos al servidor, garantizando así una implementación eficiente y acorde con las mejores prácticas.

- **Colaboración con la compañera de proyecto:** A lo largo del proceso, conté con el apoyo de mi compañera para ajustar y refinar ciertos aspectos del desarrollo. Aunque he tomado la iniciativa en la mayor parte de las funcionalidades de la implementación, su colaboración fue valiosa para resolver problemas específicos y asegurar que todas las funcionalidades estuvieran alineadas con las expectativas del proyecto y así cumplir con el objetivo marcado inicialmente .

Diagrama de navegación y diagramas de flujo y secuencia

Aparte de las funciones descritas anteriormente, también diseñé varios diagramas para facilitar la comprensión de la arquitectura y los procesos del proyecto:

- **Diagrama de navegación de la aplicación:** Creé un diagrama de navegación que detalla las diferentes pantallas de la aplicación y cómo los usuarios pueden moverse entre ellas. Esto resultó clave para estructurar el flujo de la interfaz de usuario, asegurando una experiencia coherente y lógica.
- **Diagramas de flujo y secuencia del servidor:** Desarrollé diagramas de flujo y secuencia para las rutas del servidor. Estos diagramas describen de manera detallada cómo se procesan las peticiones a través de las diferentes API del servidor, facilitando una mejor comprensión de la lógica interna y ayudando en el proceso de desarrollo.

Diagrama de arquitectura del proyecto

Finalmente, creé un diagrama de la arquitectura del proyecto que muestra la relación entre el frontend desarrollado en React, el backend (Flask) y la base de datos (MySQL), proporcionando una visión global del sistema. Este diagrama es esencial para entender cómo los diferentes componentes interactúan entre sí y contribuye a la planificación futura de las mejoras y optimizaciones del proyecto.

Pilar Gutiérrez Merino

Investigación inicial y análisis

- **Motivación del proyecto y contexto:** Me encargué de investigar y definir la motivación subyacente al proyecto, enfocándome en el contexto actual relacionado con los problemas de salud mental, en particular, el suicidio y las conductas autolesivas. Esta investigación fue fundamental para comprender la gravedad del problema y el impacto que nuestra aplicación podría tener en la

prevención de estas conductas. Además, esta fase inicial me permitió familiarizarme con el perfil de los usuarios potenciales, ayudándome a identificar sus necesidades y cómo la aplicación podría satisfacerlas de manera efectiva.

- **Estudio de aplicaciones similares:** Junto con mi compañero Li, llevé a cabo una investigación exhaustiva de aplicaciones y plataformas existentes que tratan temas similares al nuestro. Analizamos cada aplicación en detalle, evaluando sus fortalezas y debilidades para obtener una visión clara del panorama competitivo. Además, algunas de las mejores prácticas observadas en otras aplicaciones fueron adaptadas e implementadas en nuestro proyecto para mejorar su funcionalidad y usabilidad.

Definición de usuarios y casos de uso

- **Tipos de usuarios y casos de uso:** Desarrollé una definición detallada de los diferentes tipos de usuarios que interactuarían con nuestra aplicación, incluyendo adolescentes, padres y profesionales de la salud mental. Para cada tipo de usuario, elaboré casos de uso específicos que describen cómo interactuarían con la aplicación en diversos escenarios. Esta información fue crucial durante las etapas de diseño y desarrollo, ya que aseguraba que todas las funcionalidades necesarias fueran incluidas y adaptadas a las necesidades de los distintos perfiles de usuarios. Al tener en cuenta las posibles interacciones y flujos de trabajo, garantizamos que la aplicación ofreciera una buena experiencia para todos los usuarios.

Investigación y estructura de la base de datos

- **Investigación de tecnologías de base de datos:** Realicé una investigación extensa sobre las distintas tecnologías de bases de datos que podríamos utilizar en nuestro proyecto. Este análisis incluyó la evaluación de bases de datos relacionales como MySQL, así como bases de datos no relacionales como MongoDB y Redis. Analicé las ventajas y desventajas de cada una, considerando factores como la escalabilidad, la facilidad de integración, la seguridad, y el rendimiento bajo diferentes volúmenes de datos. Esta investigación fue fundamental para tomar una decisión informada sobre la tecnología más adecuada para cumplir con los requisitos del proyecto.
- **Diseño de la base de datos:** Tras seleccionar la tecnología más adecuada, diseñé la estructura completa de la base de datos. Este proceso incluyó la creación de esquemas detallados que describían la relación entre las distintas tablas, asegurando que todas las relaciones fueran coherentes y funcionales. Trabajé estrechamente con los psicólogos del proyecto para consensuar cómo se almacenarían y organizarían los datos, asegurando que la base de datos no solo cumpliera con los requisitos funcionales del proyecto, sino que también respetara las consideraciones clínicas necesarias. Una vez finalizado el diseño,

procedí a crear la base de datos e introducir algunos registros iniciales para realizar pruebas y validar su correcto funcionamiento.

Coordinación y liderazgo en reuniones

- **Liderazgo en reuniones con psicólogos:** Tomé un rol de liderazgo en las reuniones con los psicólogos colaboradores del proyecto. Durante estas reuniones, me encargué de recopilar y documentar todos los requisitos y expectativas necesarias para el desarrollo de la aplicación. Además, gestioné la comunicación para asegurar que los cambios solicitados durante el proceso de prototipado fueran implementados de manera efectiva. Esta fase fue crucial para alinear las funcionalidades de la aplicación con las expectativas de los expertos en salud mental.

Desarrollo del prototipo

- **Prototipo en Figma:** Realicé el diseño de la mitad de las pantallas del prototipo de la aplicación utilizando la herramienta de diseño Figma. Mi compañero Li se encargó de diseñar las otras pantallas. El prototipo sirvió como una herramienta esencial para la validación temprana del diseño y la usabilidad de la aplicación, permitiéndonos realizar ajustes antes de proceder con el desarrollo completo. También facilitó la comunicación con los psicólogos, ya que les permitió visualizar cómo se verían y funcionarían las diferentes partes de la aplicación.

Desarrollo de APIs del Servidor

- **Implementación de APIs:** Desarrollé varias APIs esenciales para el funcionamiento del servidor, con un enfoque especial en la API de notificaciones. Esta tarea fue particularmente desafiante debido a la necesidad de asegurar que las notificaciones se enviaran de manera fiable y oportuna. El proceso de desarrollo incluyó análisis exhaustivos y pruebas rigurosas para garantizar que la API cumpliera su funcionalidad usando la aplicación en cualquier lugar. Además, implementé la API de Almacenamiento de Respuestas, la cual es responsable de guardar las respuestas de los cuestionarios en la base de datos. Esta API fue crítica para la gestión eficiente de la información recopilada por la aplicación, asegurando que los datos se almacenaran en el formato adecuado para su posterior análisis.

Desarrollo de notificaciones en Firebase

Un aspecto destacado de mi contribución al proyecto fue el desarrollo e integración de notificaciones push utilizando Firebase Cloud Messaging (FCM). Este trabajo incluyó:

- **Configuración de Firebase:** Configuré Firebase en el servidor para habilitar el envío de notificaciones push a la aplicación React. Esto incluyó la creación de un proyecto en Firebase y la integración de las credenciales necesarias en el servidor.
- **Desarrollo de la lógica de notificación:** Implementé la lógica en el servidor para enviar notificaciones en respuesta a eventos específicos, asegurando que las notificaciones fueran enviadas de manera eficiente y oportuna.
- **Integración con la aplicación React:** Realicé los ajustes necesarios en la aplicación React para recibir y manejar las notificaciones push. Esto incluyó la implementación del código necesario para recibir las notificaciones y actualizar la interfaz de usuario en consecuencia.
- **Pruebas y ajustes:** Realicé pruebas exhaustivas para garantizar que las notificaciones se enviaran y recibieran correctamente, y ajusté la configuración según fuera necesario para mejorar la fiabilidad y el rendimiento del sistema de notificaciones.

Bibliografía

- BAÑOS-CHAPARRO, J. ¿ por qué el suicidio es un problema social? *MediSur*, vol. 22(1), páginas 205–209, 2024.
- DAYLIO. Daylio. 2024.
- FABULOUS. Fabulous: Bienestar y rutinas. 2024.
- FREELOGODESIGN. Freelogodesign. 2022. 2024-09-11.
- HARRIS, P. A., TAYLOR, R., MINOR, B. L., ELLIOTT, V., FERNANDEZ, M., OÑEAL, L., MCLEOD, L., DELACQUA, G., DELACQUA, F., KIRBY, J. ET AL. The redcap consortium: building an international community of software platform partners. *Journal of biomedical informatics*, vol. 95, página 103208, 2019.
- KHAWAS, C. y SHAH, P. Application of firebase in android app development-a study. *International Journal of Computer Applications*, vol. 179(46), páginas 49–53, 2018.
- KUMAR, A., WANG, M., RIEHM, A., YU, E., SMITH, T. y KAPLIN, A. An automated mobile mood tracking technology (mood 24/7): Validation study. *JMIR Ment Health*, vol. 7(5), página e16237, 2020. ISSN 2368-7959.
- LAUREN TEETER, C. M. L., MS. The 10 best anxiety apps, reviewed by a licensed therapist. *Fortune Recommends Health*, 2024.
- MOBERG, C., NILES, A. y BEERMANN, D. Guided self-help works: Randomized waitlist controlled trial of pacifica, a mobile app integrating cognitive behavioral therapy and mindfulness for stress, anxiety, and depression. *J Med Internet Res*, vol. 21(6), página e12556, 2019. ISSN 1438-8871.
- MOODTUNE. Moodtune. 2017.
- ORGANIZACIÓN MUNDIAL DE LA SALUD (OMS). First who report on suicide prevention. 2014.
- ORGANIZACIÓN MUNDIAL DE LA SALUD (OMS). More than 1.2 million adolescents die every year, nearly all preventable. 2017.

- ORGANIZATION, W. H. ET AL. Suicide worldwide in 2019: global health estimates. 2021.
- PALANIAPPAN, K. *Talk Space: Developing the Online Salon*. Tesis Doctoral, University of Washington, 2015.
- PLENAMENT, V. Análisis de las principales apps de meditación. 2022.
- REACT NATIVE COMMUNITY. React native directory. 2024.
- WIKIPEDIA. Modelo de estructura alámbrica — wikipedia, la enciclopedia libre. 2022. [Internet; descargado 21-enero-2022].
- WIKIPEDIA. Terapia cognitiva — wikipedia, la enciclopedia libre. 2024a. [Internet; descargado 14-abril-2024].
- WIKIPEDIA. Terapias cognitivo-conductuales — wikipedia, la enciclopedia libre. 2024b. [Internet; descargado 26-enero-2024].
- XATAKA. This person does not exist. 2017.
- DE ZEEUW, D., GEIL, A. ET AL. ‘this person does not exist’: From real generalisation to algorithmic abstraction in photographic portraiture. 2023.