

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

APLICACIONES IOT (INTERNET OF THINGS)

DESARROLLO DE UNA APLICACIÓN MÓVIL QUE PERMITA EL MONITOREO DEL SUEÑO UTILIZANDO APLICACIONES IOT

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
TECNOLOGÍAS DE LA INFORMACIÓN**

LUIS ANGEL SANGUANO CARRERA
luis.sanguano01@epn.edu.ec

DIRECTOR: SORAYA LUCÍA SINCHE MAITA, PhD.
soraya.sinche@epn.edu.ec

DMQ, Agosto 2023

CERTIFICACIONES

Yo, Luis Angel Sanguano Carrera declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

LUIS ANGEL SANGUANO CARRERA

Certifico que el presente trabajo de integración curricular fue desarrollado por Luis Angel Sanguano Carrera, bajo mi supervisión.

SORAYA LUCÍA SINCHE MAITA, PhD.

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el producto resultante del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

LUIS ANGEL SANGUANO CARRERA

SORAYA LUCÍA SINCHE MAITA

DEDICATORIA

Este trabajo lo dedico a toda mi familia: primeramente, a mis padres Rafael y Yolanda, quienes nunca me dejaron solo y siempre estuvieron conmigo en mis altas y bajas, cuidándome, y sobre todo creyendo en mí. A mis abuelitos: Papi Lucho, Mami Chula, Papá Víctor y Mamá Lolita, por estar siempre pendientes de mi bienestar. Mis primos, Pablo y Vanessa por acompañarme en cualquier locura. Mis tíos Ramiro, Hernán y Jorge por ser mi ejemplo para seguir. Finalmente, a mi tía Rosa, por estar siempre ahí como mi segunda mamá.

Este trabajo también está dirigido a todas esas personas fuera de mi círculo familiar que creyeron en mí y me permitieron formar parte de su vida, especialmente a Lizbeth y Melanie, por abrirme espacio en su corazón y apoyarme durante todos estos años.

AGRADECIMIENTO

A mis padres

Por su cuidado, amor y fe en mi persona durante toda esta travesía universitaria.

A mi Mami Chula y Papi Lucho

Por siempre estar conmigo y por ser como mis segundos padres, no existen palabras que expresen lo agradecido que estoy con la vida por tenerlos aquí conmigo.

A la PhD. Soraya Sinche

Por todo su tiempo, paciencia y conocimientos brindados durante todo el desarrollo de este trabajo y en las aulas de clase.

A Lizbeth

Por ser mi fuerza, inspiración y amor constante a lo largo de todo este camino, gracias por estar conmigo cuando nadie más estuvo y demostrarme día a día tu cariño y amor incondicional.

A mis amigos

Aquellos que he tenido el placer de conocer y aquellos que han estado a mi lado desde hace años, a “los cinco de atrás” Jorge, Mariela, Kevin y Karina. Gracias por todos esos momentos de alegría.

A mis profesores

Por ser un motivo de inspiración intelectual, por su voluntad de compartir conocimientos y por ese lado humano que comparten con los estudiantes, en particular agradezco al Ing. Pablo Hidalgo, Ing. Julio Caiza y al Ing. Danny Guamán por creer en mí.

A la Escuela Politécnica Nacional

Por ponerme a prueba semestre a semestre, permitiéndome conocer mis capacidades y potenciándolas día tras día con el conocimiento adquirido en sus aulas durante todo este viaje.

ÍNDICE DE CONTENIDO

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO	V
ÍNDICE DE FIGURAS.....	VII
ÍNDICE DE TABLAS.....	IX
RESUMEN.....	X
ABSTRACT	XI
1 INTRODUCCIÓN.....	1
1.1 Objetivo general	2
1.2 Objetivos específicos.....	2
1.3 Alcance.....	2
1.4 Marco teórico.....	4
1.4.1 El sueño y su importancia.....	4
1.4.2 Herramientas para el monitoreo del sueño.....	6
1.4.3 Herramientas para el desarrollo de aplicaciones móviles.....	9
1.4.4 Aprendizaje de máquina en el monitoreo del sueño.....	11
2 METODOLOGÍA.....	13
2.1 Diseño del prototipo de aplicación móvil.....	14
2.1.1 Análisis de requerimientos	14
2.1.2 Actores y roles del prototipo.....	18
2.1.3 Diseño de la interfaz del prototipo	20
2.1.4 Funcionalidad del prototipo	24
2.1.5 Esquema de la base de datos no relacional del prototipo	28
2.1.6 Arquitectura del prototipo	30
2.2 Implementación del prototipo.....	30
2.2.1 Instalación de herramientas	31
2.2.2 Creación del proyecto para el prototipo en Visual Studio Code.....	32
2.2.3 Arquitectura en capas	33
2.2.4 Integración de Firebase	33
2.2.5 Integración de la API de Google Fit.....	35
2.2.6 Codificación del prototipo.....	36
3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES	43
3.1 Generación del APK	43

3.2	Pruebas funcionales	43
3.2.1	Pruebas funcionales módulos común.....	43
3.2.2	Pruebas funcionales usuario Administrador General.....	45
3.2.3	Pruebas funcionales usuario Estudiante	46
3.2.4	Pruebas funcionales usuario Administrador Máster	47
3.3	Pruebas de validación Estudiantes	48
3.4	Pruebas de validación Administrador General	52
3.5	Pruebas de validación Administrador Máster.....	53
3.6	Resultados de la integración con Firebase	54
3.7	Corrección de errores	55
3.8	Resultados de registros diarios de sueño	55
3.9	Tablero Kanban final.....	58
3.10	Conclusiones	59
3.11	Recomendaciones	60
4	REFERENCIAS BIBLIOGRÁFICAS.....	61
5	ANEXOS	
	Anexo I. Categorización del monitoreo del sueño	
	Anexo II. Arquitectura de la API de Google Fit	
	Anexo III. Diseño del prototipo	
	Anexo IV. Código fuente y APK final del prototipo	
	Anexo V. Guías de usuario del prototipo	
	Anexo VI. Pruebas funcionales	
	Anexo VII. Datos recolectados y resultados obtenidos durante las pruebas de validación	
	Anexo VIII. Encuestas aplicadas durante las pruebas de validación	
	Anexo IX. Tablero Kanban (capítulo resultados, conclusiones y recomendaciones)	

ÍNDICE DE FIGURAS

Figura 1.1. Diagrama de las etapas de sueño basado en [10].....	5
Figura 1.2 Tipos de aprendizaje de máquina basado en [44]	12
Figura 2.1 Tablero Kanban desarrollado con Trello [50].....	13
Figura 2.2 Casos de uso administrador y estudiante registro de cuenta.....	19
Figura 2.3 Casos de uso estudiante registro diario de sueño	20
Figura 2.4 Paleta de colores (a) e ícono del prototipo (b).....	21
Figura 2.5 Diseño de elementos base para la interfaz gráfica	23
Figura 2.6 Diseños de la interfaz gráfica en <i>Figma</i>	24
Figura 2.7 Diagrama de actividades proceso de registro y autenticación	25
Figura 2.8 Diagrama de actividades proceso de visualización de promedios y gráficas del estudiante.....	27
Figura 2.9 Diagrama de actividades proceso de registro de sueño diario.....	27
Figura 2.10 Diagrama de actividades proceso de visualización de promedios y gráficas Administrador	28
Figura 2.11 Esquemas de las bases de datos no relacionales	29
Figura 2.12 Arquitectura del prototipo	30
Figura 2.13 Tablero Kanban etapa de implementación	30
Figura 2.14 Instalación de Visual Studio Code	31
Figura 2.15 Verificación de la instalación de Flutter	31
Figura 2.16 Creación del proyecto de Flutter en Visual Studio Code.....	32
Figura 2.17 Ejecución del ejemplo en el dispositivo virtual	32
Figura 2.18 Estructura de archivos en capas	33
Figura 2.19 Proyecto creado en Firebase (a) e implementación con Flutter (b).....	33
Figura 2.20 Verificación de la integración con Firebase	34
Figura 2.21 Configuración del proveedor de acceso para la autenticación en Firebase ..	34
Figura 2.22 Configuración de colecciones (a) y reglas de seguridad (b) Firebase.....	34
Figura 2.23 Huella digital certificado de <i>debug</i>	35
Figura 2.24 Configuración del SDK en Firebase	35
Figura 2.25 Proyecto DreamWatch en la consola de <i>Google Cloud</i> (a) y permisos de <i>Google Fit</i> para datos de sueño (b).....	36
Figura 2.26 Paquetes empleados en el desarrollo del prototipo	36
Figura 2.27 Arquitectura del módulo de autenticación.....	37
Figura 2.28 Código de ejemplo módulo de autenticación	37
Figura 2.29 Código de autenticación (a) y uso de estado capa de presentación (b).....	38
Figura 2.30 Ejemplo de código para el manejo de roles.....	38
Figura 2.31 Codificación de la base de datos SQLite (a) consulta de bases de datos (b)39	
Figura 2.32 Codificación de la navegación del prototipo	39
Figura 2.33 Codificación de permisos (a) y lectura de datos de Google Fit (b).....	40
Figura 2.34 Codificación interfaz gráfica módulo de registros	41
Figura 2.35 Especificación de colores (a) y tipografía (b)	41
Figura 2.36 Codificación menú lateral	42
Figura 3.1 Construcción del APK del prototipo.....	43
Figura 3.2 Pantallas de inicio (a) selección de rol (b)	44
Figura 3.3 Módulo de cambio de contraseña	44
Figura 3.4 Módulo de inicio de sesión	45
Figura 3.5 Verificación de código de seguridad y formulario de registro (a), aprobación de solicitud de cuentas (b), visualización de Dashboard administrador general (c).....	45
Figura 3.6 Módulos de visualización Administrador General	46

Figura 3.7 Módulo de registro	46
Figura 3.8 Módulo de permisos (a), módulo de registro diario (b).....	47
Figura 3.9 Módulos de visualización de datos usuario Estudiante.....	47
Figura 3.10 Visualización, exportación de registros y datos de usuarios	48
Figura 3.11 Diagrama del proceso de pruebas de validación	49
Figura 3.12 Versiones de Android dispositivos móviles de los participantes	49
Figura 3.13 Medio de envío elegido (a) y género de los participantes (b).....	49
Figura 3.14 Número de usuarios activos por día registrados en <i>Firebase</i>	54
Figura 3.15 Número total de operaciones de lectura, escritura y eliminación en <i>Firebase</i>	54
Figura 3.16 Proceso de análisis de datos de sueño	56
Figura 3.17 Diagrama de dispersión datos originales (a), diagrama de dispersión posterior a la depuración de los datos (b).....	56
Figura 3.18 Análisis por semestre (a), análisis por número de créditos (b)	57
Figura 3.19 Análisis por edad (a), análisis por género (b)	57
Figura 3.20 Tablero Kanban final desarrollado con Trello [50]	58

ÍNDICE DE TABLAS

Tabla 1.1	Aplicaciones de Play Store para el Monitoreo del sueño.....	7
Tabla 1.2	Sensores empleados para el monitoreo del sueño	7
Tabla 1.3	Integración con servicios de terceros.....	8
Tabla 2.1	Resumen de las entrevistas.....	15
Tabla 2.2	Roles y actores del prototipo	18
Tabla 2.3	Módulos comunes Estudiante y Administrador	21
Tabla 2.4	Módulos rol Estudiante	22
Tabla 2.5	Módulos rol Administrador	22
Tabla 2.6	Módulos Administrador Máster	23
Tabla 2.7	Extensiones Visual Studio Code para el desarrollo en Flutter	31
Tabla 2.8	Detalles dispositivo virtual Android Studio.....	32
Tabla 3.1	Resultados de la encuesta a los usuarios del rol Estudiante.....	50
Tabla 3.2	Resultados de la encuesta a los usuarios de la cuenta Administrador General.....	52
Tabla 3.3	Resultados de la encuesta rol Administrador Máster	53
Tabla 3.4	Corrección de errores	55
Tabla 3.5	Porcentaje de asignación de búhos según su calidad de sueño	58

RESUMEN

En la era digital actual, la tecnología y sus aplicaciones en la salud llegan a converger con el propósito de mejorar la calidad de vida de las personas, el presente Trabajo de Integración Curricular se centra en el desarrollo de aplicaciones móviles con enfoque en la integración del Internet de las Cosas (IoT) para tratar un aspecto trascendental en la salud de las personas: el sueño.

En este contexto, se desarrolla un prototipo de aplicación móvil para el sistema operativo Android como una alternativa novedosa para el monitoreo de sueño de las personas. Gracias a esta solución los usuarios pueden adquirir conciencia acerca su hábito de sueño al visualizar las estadísticas y la retroalimentación sobre su calidad de sueño. Adicionalmente, se tiene la capacidad de emplear el prototipo para adquirir datos de sueño de los usuarios, datos que pueden resultar interesantes para la investigación científica.

El presente trabajo se encuentra estructurado en tres capítulos. El primer capítulo detalla aspectos relacionados al sueño, su importancia y las aplicaciones móviles disponibles para el monitoreo del sueño. Adicionalmente, se describe las herramientas que conformaron el entorno de desarrollo para el prototipo. En el segundo capítulo se describe el diseño e implementación del prototipo con el *Framework* de *Flutter* y *Firebase*. El tercer capítulo muestra los resultados de las pruebas funcionales, validación y un análisis de los datos recolectados de los registros diarios de sueño enviados por 28 participantes durante la fase de pruebas del prototipo. Finalmente, se muestran las conclusiones y recomendaciones obtenidas.

PALABRAS CLAVE: monitoreo de sueño, Flutter, Firebase, Aplicaciones IoT, Sueño.

ABSTRACT

In the current digital era, technology and its applications in health converge with the purpose of enhancing the quality of people's lives. This Curricular Integration Work focuses on the development of mobile applications with a focus on the integration of the Internet of Things (IoT) to address a transcendental aspect in the human health: sleep.

In this context, a prototype of a mobile application for the Android operating system is developed as an innovative alternative for monitoring people's sleep. Thanks to this solution, users can develop an awareness of their sleep habits by visualizing statistics and feedback regarding their sleep quality. Additionally, the prototype has the capability to be used to gather sleep data from users, data that can be interest for scientific research.

The present work is structured into three chapters. The first chapter details aspects related to sleep, its significance, and the available mobile applications for sleep monitoring. Furthermore, it describes the tools that formed the development environment for the prototype. In the second chapter, the design and implementation of the prototype using the Flutter Framework and Firebase are described. Third chapter shows the results of the functional tests, validation, and analysis of the data collected from the daily sleep records submitted by 28 participants during the prototype testing phase. Finally, the conclusions and recommendations obtained are presented.

KEYWORDS: Sleep Monitoring, Flutter, Firebase, IoT applications, Sleep

1 INTRODUCCIÓN

En la última década, el número de personas que posee un teléfono inteligente se ha incrementado como consecuencia de un proceso de transformación digital, en el cual los teléfonos inteligentes pasaron de ser un accesorio para comunicación a un dispositivo necesario en el día a día de los usuarios. El acceso a contenido multimedia a través de aplicaciones móviles ha derivado en un aumento en la descarga e instalación de estas, ascendiendo a un 35% en el año 2022 [1]. De igual manera las aplicaciones relacionadas con *Health & fitness* y *Lifestyle* han experimentado un crecimiento del 11% y 15% correspondientemente en el año 2022 [1]. Por estas razones y debido a la capacidad que poseen los dispositivos móviles para obtener información a partir de la recopilación y procesamiento de los datos de los sensores, los convierten en dispositivos a ser considerados el momento de trabajar con aplicaciones del Internet de las Cosas (*IoT*) [2].

La pandemia del COVID-19 trajo consigo graves consecuencias para la humanidad, entre ellas se destacan los efectos sobre la salud de las personas. Estudios publicados por la Organización Mundial de la Salud muestran que existen efectos adversos a largo plazo relacionados a la calidad del sueño en jóvenes adultos de entre 18 a 25 años, como resultado de las medidas de restricción que se aplicaron durante la pandemia del COVID-19 [3]. El no contar con una óptima calidad de sueño puede llegar a producir efectos negativos en la salud humana, como: fatiga, somnolencia, deterioro cognitivo, disminución de la capacidad intelectual, deterioro del rendimiento, entre otros [4]. Para este tipo de problemáticas relacionadas al área de la salud, los sistemas IoT se muestran como una alternativa innovadora para facilitar el monitoreo de las horas de sueño de las personas.

En este contexto, el presente componente tiene por objetivo desarrollar un prototipo de aplicación móvil para teléfonos inteligentes con sistema operativo Android, enfocado al monitoreo de las horas de sueño de una persona. Mediante este prototipo el usuario tendrá acceso a estadísticas relacionadas con su hábito de sueño, inferidas de registros diarios enviados a través del prototipo. Además de información promedio obtenida de los registros globales de los usuarios del prototipo. Con la información recolectada se podrían plantear nuevos estudios sobre la relación entre la calidad de sueño y el rendimiento académico o profesional de una persona.

1.1 Objetivo general

Desarrollar un prototipo de aplicación móvil que permita el monitoreo del sueño, utilizando aplicaciones IoT.

1.2 Objetivos específicos

1. Analizar las herramientas del *Framework* de *Flutter* para el desarrollo de aplicaciones móviles en un teléfono inteligente con sistema operativo Android.
2. Determinar los módulos que permitirán integrar la aplicación *Sleep As Android* y la API de *Google Fit* al prototipo de aplicación móvil para recopilar, almacenar y mostrar la información de sueño al usuario mediante este prototipo.
3. Implementar el prototipo de aplicación móvil para el sistema operativo Android con el *Framework* de *Flutter* en base a los diseños de los módulos previamente desarrollados.
4. Analizar los resultados obtenidos del funcionamiento del prototipo de aplicación móvil.

1.3 Alcance

Se presentan los conceptos básicos relacionados al sueño y el porqué de su importancia, así como el estudio de las herramientas actualmente disponibles para el monitoreo del sueño como *Sleep as Android* y *Google Fit*.

Se revisan las herramientas empleadas para el desarrollo del prototipo como lo son: el *Framework* de *Flutter*, el IDE de *Visual Studio Code*, emulador virtual y *Firebase*. A continuación, se describen los conceptos relacionados al aprendizaje de máquina aplicado en el monitoreo de sueño, en particular el *Deep Learning*.

La gestión del componente se basa en la metodología de *KANBAN* debido a sus ventajas en el desarrollo de software bajo metodologías ágiles. Luego, se definen los requerimientos funcionales y no funcionales para el prototipo empleando historias de usuario en base a entrevistas realizadas al menos a 3 personas. Como parte del diseño se detallan los diferentes módulos del prototipo en base a los requerimientos identificados que forman parte del monitoreo de las horas de sueño.

El prototipo de aplicación móvil con sistema operativo *Android* se implementa con el *Framework* de *Flutter*. El prototipo cuenta con dos roles: estudiante y administrador. Dentro del rol administrador existen dos tipos de cuentas: administrador general y administrador

máster. La cuenta del administrador máster posee privilegios superiores con respecto a la cuenta de administrador general.

En ambos roles los procesos de autenticación y registro emplean un correo electrónico y contraseña para verificar la identidad del estudiante o administrador haciendo uso del servicio de autenticación de *Firestore*. Por otra parte, se cuenta con un módulo que permita el cambio de contraseña mediante el envío de un correo electrónico de cambio de contraseña empleando *Firestore*.

El rol estudiante tiene la capacidad de hacer operaciones CRUD (*create, read, update, delete*) sobre los datos de su perfil personal dentro del prototipo. Adicionalmente, puede permitir o restringir el permiso de acceso del prototipo a los datos de sueño de su cuenta de *Google Fit*. Además, cuenta con un módulo para el envío de registros diarios de sueño a la base de datos del proyecto, empleando un formulario y los datos de sueño obtenidos de *Google Fit*. Estos registros son almacenados en la base de datos general del proyecto en *Firestore* y el estudiante tendrá la capacidad de eliminarlos cuando lo desee. Se tiene un módulo de ayuda que permite enviar un mensaje de *WhatsApp* al administrador máster cuando un estudiante o administrador requiera soporte por problemas relacionados al funcionamiento del prototipo. Finalmente, el estudiante cuenta con varios módulos que le permiten visualizar promedios sobre su hábito de sueño y contrastarlo con los promedios inferidos de la base de datos general del proyecto del resto de estudiantes.

El rol administrador para crear su cuenta primero debe registrarse en el prototipo e ingresar sus datos personales para solicitar la verificación de su cuenta al administrador máster. Una vez verificada su cuenta este rol cuenta con acceso a promedios de los registros de los usuarios mediante módulos que muestran dicha información. Además, puede exportar un archivo .csv con los registros de las horas de sueño disponibles en la base de datos, dicho archivo no cuenta con campos que relacionen los registros con sus usuarios. Finalmente, al igual que el rol estudiante, el administrador contará con un módulo de ayuda.

La cuenta del administrador máster tiene accesos privilegiados para: aprobar solicitudes de cuentas de administración, visualizar y exportar datos de los usuarios de la aplicación, teniendo la capacidad de acceder a sus nombres y apellidos para llevar el control de los usuarios del prototipo.

Para recolectar los datos relacionados a las horas de sueño estimadas de un estudiante se utiliza la aplicación de *Sleep As Android* de *Urbandroid* [5] que hace uso de un modelo de *Deep Learning* en conjunto con los sensores del dispositivo inteligente para estimar las horas de sueño del usuario y enviar estos datos a su cuenta asociada en *Google Fit* [5].

Estos datos son utilizados por el prototipo a través de la *API* de *Google Fit* [6] para obtener los datos de sueño de los usuarios y compartirlos en la nube junto a la información ingresada a través del prototipo empleando formularios.

Finalmente se realizan las pruebas de funcionamiento del prototipo de aplicación móvil implementado con al menos 20 usuarios, que probaron los roles planteados. En base a los resultados de las pruebas de funcionamiento se realizan las correcciones en caso de ser necesario, para finalmente presentar conclusiones, recomendaciones y el APK con el prototipo final.

1.4 Marco teórico

El sueño es una actividad fundamental para los seres humanos y guarda relación con su salud y bienestar. Por esta razón su monitoreo resulta necesario para prevenir problemas a corto o largo plazo derivados de malos hábitos de sueño. Para cubrir esta necesidad, en la actualidad existen herramientas en un abanico de opciones, destacando las aplicaciones móviles que se muestran como soluciones innovadoras y hacen uso de nuevas tecnologías como la inteligencia artificial. Motivando de esta manera el desarrollo de nuevas aplicaciones de salud y bienestar de manera ágil empleando herramientas, por ejemplo, *Flutter* y servicios como *Firebase* o *Google Fit*.

1.4.1 El sueño y su importancia

El sueño es una actividad compleja que se encuentra relacionada con procesos neurobiológicos que incluyen la intervención de múltiples funciones fisiológicas propias de los seres vivos [7]. Con respecto a los seres humanos, esta actividad forma parte importante de su vida, pues este pasará cerca de un tercio [8] de su existencia durmiendo, lo cual corresponde a un porcentaje considerable del tiempo.

Al analizar al sueño como una actividad fisiológica se puede observar que una persona experimenta un patrón de etapas cíclico lo largo de su sueño [8]. Las etapas que una persona normalmente experimenta durante el sueño son: NREM-N1 (*Non-Rapid Eye Movement*) y NREM-N2 que conforman el sueño ligero, la etapa NREM-N3 que forma parte del sueño profundo y finalmente la etapa REM (*Rapid Eye Movement*) [9]. Estas etapas podrían extenderse en otros análisis más complejos de hasta 5 etapas [10].

En la Figura 1.1 se observa un resumen acerca del ciclo del sueño, resaltando aquellas características esenciales para entender este proceso desde un punto de vista fisiológico. El tiempo de duración de cada etapa, como sus principales características permitirán entender cómo funciona este mecanismo natural del ser humano.

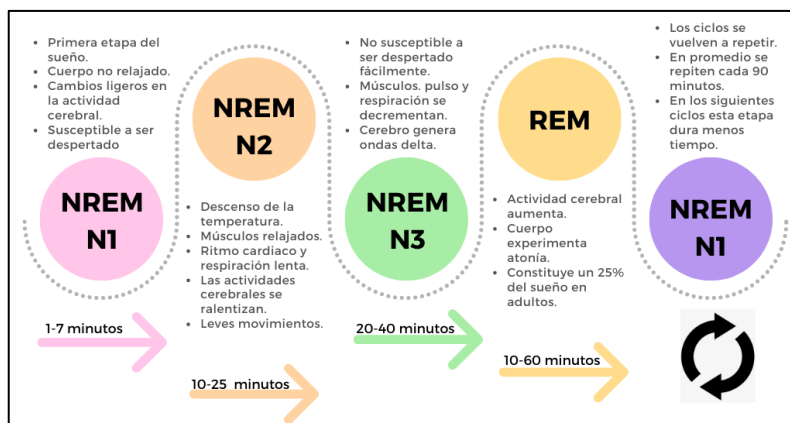


Figura 1.1. Diagrama de las etapas de sueño basado en [10]

Según la *National Sleep Foundation* [11] un joven de entre los 14 a los 17 años necesita cerca de 8 a 10 horas de sueño, mientras que un adulto joven de entre los 18 a 25 y un adulto de entre 26 a 64 años necesitará de 7 a 9 horas. Si bien el sueño es una actividad importante para el bienestar de las personas, en la actualidad no se la toma con la seriedad adecuada; especialmente en los adolescentes. Factores como el estrés o la adicción a la tecnología hacen que el no dormir sea una actividad normal en los jóvenes [12]. Adicionalmente existen factores externos que pueden llegar a influir en la calidad de sueño de los jóvenes y provocar efectos desfavorables a largo plazo. La pandemia del COVID-19 es un claro ejemplo, dado que afectó la calidad de sueño de adultos de entre 18 y 25 años según la Organización Mundial de la Salud [3].

La mala calidad de sueño puede provocar varios problemas de salud, entre los que se destacan: insomnio, enfermedades psiquiátricas y neurológicas [13], disminución de la capacidad intelectual, deterioro del rendimiento, entre otros [4]. Inclusive puede desembocar en episodios trágicos como el suicidio, pues se ha constatado que los adolescentes que duermen menos de seis horas diarias presentan una mayor probabilidad de reportar pensamientos suicidas [14]. Es evidente que no todos los casos relacionados a una mala calidad de sueño desembocarán en un suicidio, no obstante, esto, en conjunto con los efectos negativos en la salud evidencian la importancia de tener una buena calidad de sueño.

Conociendo los riesgos potenciales del mal hábito de no dormir y las horas recomendadas por los expertos, es fundamental contar con herramientas que permitan a las personas conocer las horas que duermen al día [15]. Algunas de las herramientas disponibles para este propósito trabajan en colaboración con los avances tecnológicos para el análisis y comprensión del sueño, desde un punto de vista patológico y psicológico, lo cual brinda a las personas herramientas novedosas para monitorear su sueño [15].

En el ANEXO I.1 basado en la referencia [16] se muestra un diagrama informativo acerca de la categorización del monitoreo del sueño. Esta clasificación define dos categorías fundamentales: tecnologías para laboratorio y tecnologías para el consumidor. Las primeras son más sofisticadas y requieren de equipo especializado, en cambio las de consumidor se presentan como una alternativa novedosa ya que hacen uso de nuevos dispositivos y tecnología como es la inteligencia artificial, así como aplicaciones para teléfonos inteligentes. Conocer la categorización del monitoreo del sueño brinda una visión de los métodos que se están utilizando actualmente para este fin, y de manera particular de aquellos clasificados como tecnologías de consumidor que resultan innovadores y tienen un menor costo que las tecnologías de laboratorio.

1.4.2 Herramientas para el monitoreo del sueño

En la actualidad existen varias herramientas que permiten monitorizar el sueño de una persona. Entre estas, las clasificadas como tecnologías para laboratorio llegan a presentar un mayor nivel de exactitud en el monitoreo del sueño, en particular la Polisomnografía (PSG) que brinda un estudio preciso y profundo acerca de los fenómenos del sueño [16]. Sin embargo, el acceso a esta herramienta llega a ser costoso y requiere ser realizada en un laboratorio, lo cual puede resultar invasivo para el paciente, provocando que se interrumpa su sueño [10].

Dado el alto costo que puede implicar un estudio con alguna de las tecnologías de laboratorio presentadas en el ANEXO I.1, las tecnologías para el consumidor se destacan como una alternativa novedosa. Por ejemplo, las aplicaciones móviles facilitan a los usuarios el monitoreo del sueño utilizando una aplicación instalada en su teléfono inteligente [17] que puede ser descargada gratuitamente o a un costo significativamente menor que un estudio médico.

1.4.2.1 Aplicaciones móviles como herramientas para el monitoreo

Las aplicaciones móviles se encuentran clasificadas como tecnologías para el consumidor acorde al ANEXO I.1. El acceso y descarga de estas aplicaciones se realiza a través de las plataformas digitales de distribución disponibles en los diferentes sistemas operativos de los dispositivos móviles, por ejemplo: *Play Store* para dispositivos *Android* o la *App Store* para iOS. Según *StatCounter* [18], los sistemas operativos para dispositivos móviles más usados en Ecuador son: Android y iOS que ocupan más del 95% de la cuota de mercado nacional. *Android* cuenta con la prevalencia del mercado, pues ocupa un 81.12 % de la cuota del mercado en comparación a iOS que ocupa un 18.36 % ubicándose en segundo lugar [18].

Dado que gran parte de los dispositivos móviles tanto a nivel nacional como global tienen sistema operativo Android [18], existe una amplia variedad de aplicaciones disponibles dentro de *Play Store* en la categoría de *Health & Fitness*, que facilitan a los usuarios el monitorear su sueño únicamente con su dispositivo móvil. La Tabla 1.1 muestra datos acerca de las cinco aplicaciones más descargadas del *Play Store* relacionadas con el monitoreo del sueño.

Tabla 1.1 Aplicaciones de Play Store para el Monitoreo del sueño

Aplicación	<i>Sleep As Android</i> [19]	<i>BetterSleep</i> [20]	<i>Sleep Cycle</i> [21]	<i>RoncoLab</i> [22]	<i>Sleep Monitor</i> [23]
Desarrollador	Urbandroid	Ipnos Software	Sleep Cycle AB	Reviva Softworks Ltd	SM Health Team
Descargas	+10 millones	+10 millones	+10 millones	+5 millones	+5 millones
Puntuación	4.6 367K reviews	4.5 336K reviews	4.4 182K reviews	4.5 42.1K reviews	4.3 45.2K reviews
Acceso	Gratis	\$3,91/mes	Gratis	\$3,99/mes	Gratis

Las aplicaciones presentadas en la Tabla 1.1 hacen uso de los sensores disponibles en los teléfonos inteligentes, para tomar distintas mediciones e inferir el estado de sueño, en el cual se encuentra el usuario con el fin de estimar la duración del sueño del usuario. La Tabla 1.2 muestra un resumen de los distintos sensores que emplean las aplicaciones de la Tabla 1.1 para su funcionamiento:

Tabla 1.2 Sensores empleados para el monitoreo del sueño

Aplicación	<i>Sleep As Android</i> [20]	<i>BetterSleep</i> [21]	<i>Sleep Cycle</i> [22]	<i>RoncoLab</i> [23]	<i>Sleep Monitor</i> [24]
Sensores	Micrófono	✓	✓	✓	✓
	GPS	✓	×	✓	×
	Acelerómetro	✓	×	×	✓
	Sonar (Altavoz – Micrófono)	✓	×	×	×
	Proximidad	✓	×	×	×
	Nivel de Luz	✓	×	×	×

Como se puede observar en la Tabla 1.2, la aplicación que emplea un mayor número de sensores para trabajar es *Sleep As Android*, seguida de *Sleep Cycle* y *Sleep Monitor*. Es

importante que una aplicación móvil que tiene como objetivo monitorear el sueño facilite al usuario integrar servicios que permitan enviar y leer los datos generados durante el monitoreo de sueño. La Tabla 1.3 muestra los servicios de terceros que se encuentran integrados en las aplicaciones de la Tabla 1.1.

Tabla 1.3 Integración con servicios de terceros

Aplicación		<i>Sleep As Android</i>	<i>BetterSleep</i>	<i>Sleep Cycle</i>	<i>RoncoLab</i>	<i>Sleep Monitor</i>
Servicios	Health Connect	✓	×	×	×	✓
	Samsung Connect	✓	×	×	×	×
	Google Fit	✓	×	✓	×	✓

En la Tabla 1.3 la aplicación de *Sleep As Android* se destaca por sus múltiples opciones de integración con servicios de terceros en comparación a las otras aplicaciones. Esta gran variedad de opciones junto al número de sensores que emplea *Sleep As Android* hacen que esta aplicación sobresalga entre las opciones disponibles para *Android*.

1.4.2.2 Sleep as Android

Sleep as Android es una aplicación móvil desarrollada por *Urbandroid*. Esta aplicación se muestra como una solución alternativa a las Tecnologías de Laboratorio como el PSG y la Actigrafía [24], permitiendo a sus usuarios realizar el monitoreo de su sueño.

Esta aplicación tiene la capacidad de realizar el seguimiento de las etapas del ciclo del sueño con los sensores presentados en la Tabla 1.1; además, facilita la integración con los servicios mostrados en la Tabla 1.3 y compatibilidad con dispositivos *wearables* como Mi Band, Amazfit, FitBit, entre otros [25].

La aplicación recopila datos de sensores, como, por ejemplo: el micrófono que es empleado para registrar sonidos que se presenten durante el monitoreo [26]. El acelerómetro es empleado para analizar el movimiento de la persona. Además, se cuenta con una opción de sonar (altavoz-micrófono) que se emplea con el mismo objetivo que el acelerómetro (detectar movimiento), este sensor se encuentra disponible en dispositivos compatibles y la propia aplicación cuenta con un apartado para verificar su compatibilidad [27]. Finalmente, los sensores de luz y proximidad son empleados para detectar la actividad del usuario e inferir su estado de sueño.

El monitoreo del sueño se lo puede realizar de dos maneras: accediendo al apartado de monitoreo de la aplicación para activar y desactivar esta función el momento que el usuario

se disponga a dormir. Al emplear esta opción del monitoreo la aplicación hace uso de todos los sensores descritos anteriormente; además de emplear *Deep Learning* para inferir las distintas etapas del sueño como clasificar los sonidos, etc. La segunda forma es permitir a la aplicación correr en segundo plano analizando el uso del celular y el sensor de proximidad y nivel de luz para inferir si el usuario se encuentra despierto o dormido, claramente el nivel de exactitud en el segundo caso es menor [27].

1.4.3 Herramientas para el desarrollo de aplicaciones móviles

Para desarrollar aplicaciones móviles para teléfonos inteligentes con sistema operativo *Android* es necesario contar el JDK (*Java Development Kit*) [28] y SDK (*Software Development Kit*) [29]. Además, es importante decidir entre el desarrollo con código nativo, por ejemplo: Java o Kotlin y el multiplataforma. Este último cuenta con varios *Frameworks* como *Flutter*, *Xamarin*, *React Native*, entre otros que facilitan el proceso de desarrollo [30].

1.4.3.1 Flutter

Flutter se presenta como un *Framework* completo para el desarrollo de aplicaciones móviles bajo un marco de código abierto desarrollado por *Google* [31], que emplea el lenguaje de programación Dart, el cual posee una compilación *Just-in-Time* que permite ejecutar cambios en el código de forma inmediata [32]. Gracias a estas características las interfaces de usuario en Flutter resultan coherentes, atractivas [33] y funcionan de manera fluida y con un alto rendimiento [31].

El desarrollo de aplicaciones con Flutter presenta una limitación en el acceso a funcionalidades de bajo nivel en los teléfonos inteligentes como; por ejemplo, el acceso hardware [41]. Para solventar esta limitación Flutter cuenta con un repositorio y gestor de paquetes llamado *Puv Deb* [34], en el cual se pueden acceder a paquetes con gran variedad de funcionalidades específicas; por ejemplo, acceso al hardware.

En comparación a otros *Frameworks*, *Flutter* compila a código nativo y por esta razón ofrece un mejor desempeño. Además, no emplea componentes nativos, pues hace uso de sus propios componentes denominados *widgets*, permitiendo mantener la apariencia de la aplicación móvil independientemente de la versión o sistema operativo del dispositivo [32].

Los *widgets* en Flutter son básicamente todo lo que un usuario puede visualizar en la pantalla, pueden ser botones, textos, ventanas, etc. Estos widgets poseen propiedades para personalización y estilo. También, es posible combinar widgets uno dentro de otro para conseguir elementos más complejos y visualmente atractivos [35]. Por defecto Flutter posee un catálogo de 14 categorías de widgets que incluyen a *Cupertino* para estilos iOS y *Material Components* relacionado a *Material Design* de Google.

Finalmente, Flutter posee dos características importantes: *Hot Reload* y *Hot Restart*, el primero permite visualizar los cambios rápidamente en la interfaz de usuario sin la necesidad de detener o reiniciar la aplicación, en cambio *Hot Restart* es un proceso más lento y es empleado cuando se realizan cambios en la configuración de la aplicación [36].

1.4.3.2 Visual Studio Code

Visual Studio Code (VS Code) propiedad de Microsoft es un editor de código fuente disponible para escritorio que posee un amplio ecosistema de extensiones [37]. Estas extensiones permiten a los programadores trabajar con múltiples lenguajes de programación como Dart, C++, Python, etc en un único Entorno de Desarrollo Integrado (IDE). Otro de los atractivos principales de este IDE para los programadores es su interfaz de usuario que resulta simple y sencilla de usar, mostrando varias barras y paneles que permiten acceder a distintas funcionalidades, además de implementar atajos de teclado y otras facilidades para programar como los *Snippets* que se emplean para insertar rápidamente código predefinido [38]. VS Code permite a los programadores ejecutar comandos directamente desde el IDE, lo que facilita tareas de prueba, despliegue y automatización [39].

1.4.3.3 Emulador virtual

El emulador virtual es una herramienta que imita el software y hardware de un dispositivo móvil empleando los recursos de una estación de trabajo. Para entender cómo trabaja un emulador es necesario conocer acerca del *Instruction Set Architecture* (ISA), que es un conjunto de instrucciones que se encuentra escrito en lenguaje de máquina y puede ser entendido por un procesador. Todos los dispositivos desde un celular hasta una computadora emplean ISA, por este motivo un emulador puede imitar el procesador de un teléfono inteligente, esto se logra a través de la traducción binaria para pasar del ISA del dispositivo al del ordenador. El binario que se obtiene de esta traducción puede ser complementado con una API compatible y un sistema operativo [40].

Uno de los emuladores más populares entre los programadores de aplicaciones móviles es *Android Emulator* que se encuentra disponible en *Android Developer Studio*. Este emulador facilita el trabajo con varios niveles de API de Android, consiguiendo de esta manera probar la aplicación móvil en versiones pasadas y actuales de este sistema operativo [41].

1.4.3.4 Firebase

Firebase es un *Backend-as-a-Service* (BaaS) que brinda a los desarrolladores herramientas y servicios bajo la infraestructura de Google [42]. Este BaaS se encuentra

disponible tanto para desarrollo móvil como web. Algunos de los servicios que ofrece *Firebase* son: analíticas, autenticación, *realtime database*, *Firestore Database*, *Cloud Messaging*, *hosting*, *machine learning*, *Test Lab*, entre otros. *Firebase* proporciona una consola para administrar el *backend* de una aplicación móvil, además de incluir facilidades de integración directa de servicios con *Frameworks* como *Flutter*.

1.4.3.5 Google Fit API

Google Fit se encuentra diseñado con un enfoque basado en un ecosistema abierto [43]. Lo cual facilita a los usuarios el poder almacenar, registrar y medir información relacionada con las actividades de entrenamiento y salud en dispositivos inteligentes como celulares y relojes [5]. En cambio, para los desarrolladores de aplicaciones móviles las APIs de *Google Fit* facilitan la implementación de aplicaciones que permitan compartir datos de bienestar y salud a un repositorio centralizado, en el cual los usuarios pueden acceder a sus datos [43]. Esto agiliza el desarrollo de aplicaciones móviles relacionadas con la salud y el bienestar.

El acceso a los datos de salud y bienestar se realiza a través de las APIs que pertenecen a los Servicios de *Google Play*. Estas APIs son compatibles con versiones de Android a partir de la versión 4.1 con un nivel de API 16 [6]. El ANEXO II.1 muestra la arquitectura de *Google Fit* según [6].

1.4.4 Aprendizaje de máquina en el monitoreo del sueño

En la actualidad la inteligencia artificial (IA), particularmente el aprendizaje de máquina (ML) ha experimentado un crecimiento considerable dentro del contexto de la informática y el análisis de datos, permitiendo que las aplicaciones trabajen de forma inteligente [44].

Los métodos de ML se caracterizan por la capacidad de examinar un gran número de datos para obtener información importante [45] y son ampliamente usados para dar solución a problemas complejos en varias áreas de aplicación como; por ejemplo: finanzas, salud, entre otros. Con respecto al área de salud y en particular el cuidado del sueño, estos métodos resultan innovadores, pues optimizan: la fiabilidad, precisión del diagnóstico de enfermedades y trastornos del sueño [46].

Los algoritmos de ML se encuentran divididos principalmente en cuatro tipos: aprendizaje por refuerzo, aprendizaje supervisado, aprendizaje no supervisado y aprendizaje semisupervisado. Estos se diferencian por el tipo de datos (*dataset*) empleado en su entrenamiento, como se muestra en la Figura 1.2:

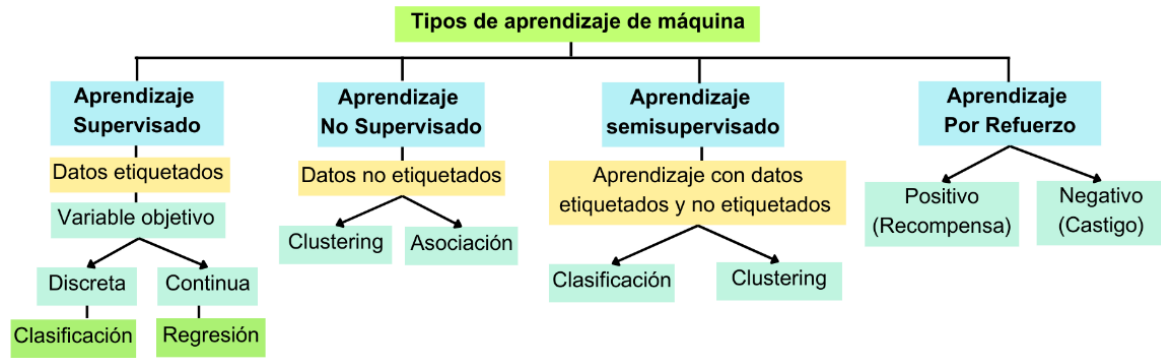


Figura 1.2 Tipos de aprendizaje de máquina basado en [44]

Algunos de los algoritmos de ML más utilizados para el análisis del sueño son: regresión lineal, árbol de decisión, redes neuronales, regresión logística, máquinas de vector soporte, random forest, k-nearest neighbors, Deep Learning, entre otros [47], [17], [10].

1.4.4.1 Deep Learning

Deep Learning forma parte de una variedad más extensa de enfoques de aprendizaje automático que se fundamenta en las redes neuronales artificiales (RNA) [44]. Esta tecnología proporciona una arquitectura en varias capas de procesamiento que se identifican como: capas de entrada, ocultas y salida. *Deep Learning* proporciona un mejor rendimiento sobre el ML tomando en cuenta aplicaciones con una gran cantidad de datos, pero esta característica dependerá del tipo de datos utilizados y la configuración experimental [44].

Los algoritmos de *Deep Learning* más comunes son: Perceptrón Multicapa (MLP), Red Neuronal Convolutiva (CNN) y Red Neuronal Recurrente de Memoria Larga o Corto Plazo (LSTM-RNN) [44]. Los algoritmos pueden aprender patrones ocultos para realizar predicciones de manera efectiva ante problemas complejos [48].

Aplicaciones como *Sleep As Android* [25] o prototipos como *Sleep Hunter* [8] emplean modelos entrenados con *Deep Learning* y datos obtenidos de los sensores disponibles en un teléfono inteligente para inferir entre otras cosas el estado del sueño de una persona.

2 METODOLOGÍA

En la presente sección se detalla el desarrollo e implementación del prototipo de aplicación móvil destinado al monitoreo del sueño.

El presente Trabajo de Integración Curricular (TIC) adopta como referencia la metodología Kanban para cumplir los objetivos planteados, dado que Kanban es un sistema de gestión que permite manejar la cantidad de trabajo adecuada para que un proyecto se pueda desarrollar de manera eficiente [49].

En la Figura 2.1 se muestra la instancia inicial del tablero Kanban planteado para el presente trabajo con todas las actividades descritas en el Plan de Trabajo de Integración Curricular. Este tablero se actualiza a medida que avanza el trabajo.

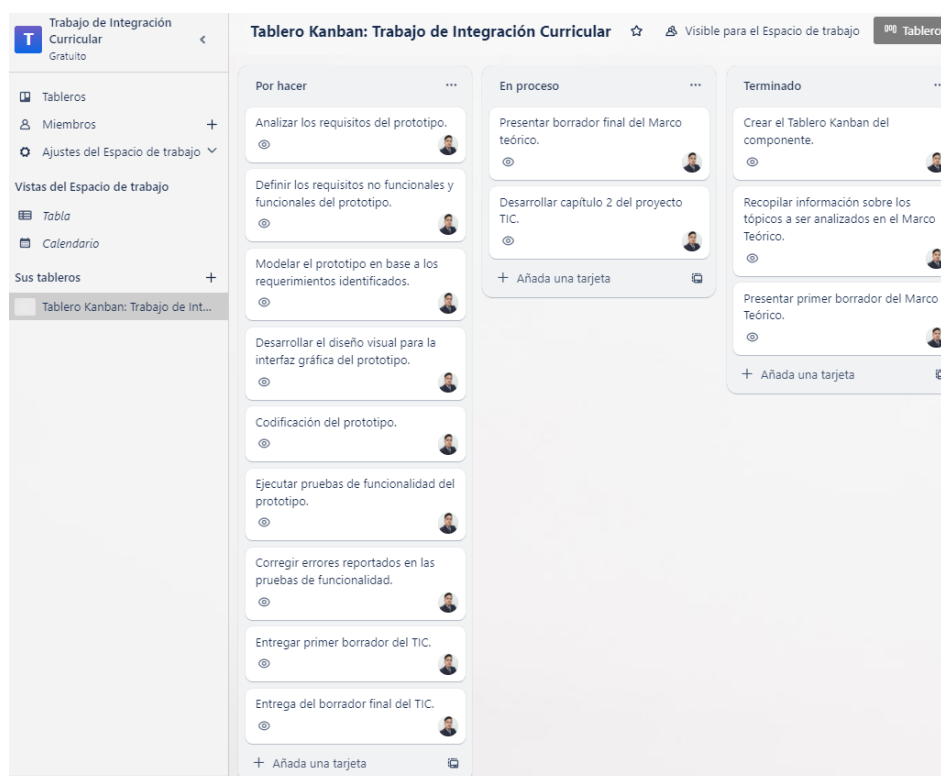


Figura 2.1 Tablero Kanban desarrollado con Trello [50]

El enfoque de este trabajo es de carácter mixto pues se hace uso de elementos cualitativos y cuantitativos en las distintas fases del prototipo. El enfoque cualitativo es empleado en la fase de análisis de requisitos, pues abarca actividades relacionadas al trabajo con los potenciales usuarios del prototipo para comprender sus expectativas y preferencias. Por otro lado, el enfoque cuantitativo es empleado en las fases de implementación y pruebas. En el primer caso por su vínculo con aspectos técnicos y operativos, y el segundo por su relación con la evaluación de la funcionalidad en base a la experiencia de uso de los

usuarios de prueba. En la fase de diseño se adopta un enfoque mixto, pues, se necesita contar con una perspectiva completa del desarrollo del prototipo.

Con respecto a las técnicas de recolección de información empleadas en el desarrollo del prototipo se encuentran: entrevistas y cuestionarios. Las entrevistas forman parte del análisis de requerimientos y fueron desarrolladas de manera presencial y telemática, mientras que los cuestionarios son empleados para obtener los datos de la experiencia de uso del prototipo en la fase de pruebas.

Los repositorios principales de consulta de información utilizados son: *IEEE Xplore* y *Scopus*. La búsqueda de artículos científicos relacionados con el monitoreo del sueño es realizada en *IEEE Xplore* para lo cual se hace uso de las siguientes palabras clave: “*sleep*”, “*monitoring*”, “*mobile*” y “*tracking*”. Para la búsqueda de contenido relacionado al sueño desde un punto de vista médico se emplea la base de datos de *Scopus*, utilizando las palabras clave: “*sleep*”, “*behavior*” y “*health*”.

Para la búsqueda de información referente a *Flutter* y los servicios a implementar en el prototipo se consulta a: *Stack Overflow* y *GitHub*. En el primer caso se hace uso de las *Questions tagged* [51] buscando los artículos relacionados a *Flutter* o cualquier servicio requerido. En cambio, *Github* es empleado para buscar ejemplos de implementaciones en *Flutter* para dispositivos móviles con sistema operativo *Android*.

2.1 Diseño del prototipo de aplicación móvil

En esta sección, inicialmente, se aborda el proceso de análisis de requerimientos para el prototipo, identificando los requerimientos funcionales y no funcionales. Adicionalmente, se muestran los diagramas de casos de uso y de actividades. Se presentan los bosquejos iniciales para la interfaz gráfica. Finalmente se muestra el modelo del sistema y la base de datos no relacional empleada por el prototipo.

2.1.1 Análisis de requerimientos

La toma de requerimientos para el prototipo se realizó empleando la técnica de entrevistas individuales, lo que permitió obtener información detallada y aclarar dudas sobre el prototipo [52]. Los entrevistados fueron: PhD. Soraya Sinche profesora principal Escuela Politécnica Nacional, Dra. Nathaly Chávez médica cirujana de la Dirección de Bienestar Politécnico de la Escuela Politécnica Nacional y el Ing. Rommel Arévalo. En el ANEXO III.1 se muestran las evidencias de las reuniones presenciales y telemáticas.

De estas entrevistas se logró evidenciar la necesidad de contar con un prototipo de aplicación móvil que permita el monitoreo del sueño de un estudiante, el cual permita

visualizar promedios y estadísticas de sus noches. Adicionalmente, se destaca la motivación en el apartado de investigación que conlleva el desarrollo de este prototipo, pues los entrevistados coinciden que a futuro este prototipo facilitará la obtención de datos que permitan realizar nuevos estudios relacionados a la calidad del sueño. En la Tabla 2.1 se resumen las preguntas relevantes abordadas durante las entrevistas:

Tabla 2.1 Resumen de las entrevistas

N°	Pregunta	Sí %	No %
1	¿Considera útil una aplicación móvil que le permita monitorear el tiempo de sueño de una persona?	100	0
2	¿Considera adecuado que el monitoreo del sueño sea realizado con un celular inteligente?	100	0
3	¿Considera usted necesario el proceso de autenticación para una aplicación de esta naturaleza?	100	0
4	¿Considera útil tener los roles de Estudiante y Administrador en la aplicación?	100	0
5	¿Le resultaría útil conocer su calidad de sueño?	100	0
6	¿Cree usted que resulta útil asignar un valor numérico a la calidad del sueño?	100	0
7	¿Estaría usted dispuesto a compartir los resultados relacionados con su calidad de sueño en una base de datos general, siempre y cuando se garantice que estos resultados no permitirán establecer una conexión directa entre usted y su registro?	100	0
8	¿Considera importante tener la capacidad de comparar sus estadísticas de tiempo de sueño con los promedios de otras personas?	100	0
9	¿Considera usted que resulta viable exportar la información reportada por los usuarios con el fin de realizar estudios futuros relacionados con la calidad del sueño de los estudiantes?	100	0
10	¿Cree usted, que resultaría útil que un Administrador tenga acceso a detalles generales sobre el uso del prototipo por parte de los usuarios, como, por ejemplo; el número total de usuarios registrados o promedios generales del tiempo de sueño de los usuarios?	100	0

2.1.1.1 Requerimientos Funcionales (RF)

Los requerimientos funcionales son aquellos que permiten describir la funcionalidad que se espera tener en el prototipo [53]. Se hace uso de las historias de usuario para detallar los requerimientos identificados en las entrevistas realizadas. En total se identificaron diez historias de usuario; a continuación, se muestran las primeras tres historias, el resto se encuentra en el ANEXO III.2.

RF 01	Registrar nuevo usuario
Usuario:	Administrador y Estudiante
Prioridad:	Alta
Descripción:	
Se requiere contar con un módulo de registro diferenciado por rol para el prototipo.	
Criterios de aceptación:	
<ul style="list-style-type: none"> • El prototipo debe contar con un formulario de registro que solicite un correo electrónico y contraseña. • El prototipo debe permitir elegir el rol antes de realizar el registro. • En caso del rol Administrador se debe solicitar un código de seguridad para acceder al formulario de registro, estos códigos de seguridad deben ser manejados por el Administrador Máster del prototipo. • En el caso del rol Estudiante para completar su registro se debe solicitar la verificación de su correo electrónico. • En el caso del rol Administrador para completar su registro se debe solicitar la aprobación de su solicitud para registro al Administrador Máster, el prototipo debe contar con las facilidades para realizar este proceso. 	

RF 02	Autenticar usuario
Usuario:	Administrador y Estudiante
Prioridad:	Alta
Descripción:	
Se requiere contar con la autenticación para el prototipo.	
Criterios de aceptación:	
<ul style="list-style-type: none"> • El prototipo contará con un formulario donde se solicite un correo electrónico y contraseña para ingresar. • En caso de ingresar un correo que no se encuentra registrado se debe mostrar un mensaje de error al usuario detallando lo sucedido. • En caso de introducir una contraseña incorrecta se debe mostrar un mensaje de error al usuario. • El prototipo debe permitir autenticar y desautenticar al usuario, manteniendo el dicho estado dentro del prototipo. • El prototipo debe contar con botones que permitan cerrar la sesión del prototipo. 	

RF 03	Cambiar contraseña
Usuario:	Administrador y Estudiante
Prioridad:	Alta
Descripción:	
Se requiere contar con un módulo de cambio de contraseña.	
Criterios de aceptación:	
<ul style="list-style-type: none"> • El prototipo contará con un módulo que permita el cambio de contraseña de su cuenta personal. • Se debe permitir el acceso al módulo de cambio de contraseña, tanto cuando el usuario esté autenticado como cuando no. 	

2.1.1.2 Requerimientos no funcionales (RNF)

Los requerimientos no funcionales se suelen enfocar al sistema en su totalidad, en lugar de dirigirse a las características o servicios específicos del sistema [53]. Algunos de estos requerimientos se encuentran relacionados con la disponibilidad, seguridad o rendimiento.

Los requerimientos no funcionales se detallan a continuación:

RNF 01	Tamaño máximo del prototipo
Usuario:	Administrador y Estudiante
Prioridad:	Alta
Descripción:	
Se requiere que el prototipo no tenga un tamaño grande en MB.	
Criterios de aceptación:	
<ul style="list-style-type: none">• El tamaño máximo del prototipo no debe superar los 50 megabytes (MB).	

RNF 02	Usabilidad del prototipo
Usuario:	Administrador y Estudiante
Prioridad:	Alta
Descripción:	
Es necesario contar con una interfaz de usuario que sea intuitiva, fácil de usar y amigable, para permitir al usuario interactuar sin problema alguno.	
Criterios de aceptación:	
<ul style="list-style-type: none">• Los formularios del prototipo deben contar con los campos mínimos y necesarios para el funcionamiento.• El acceso a los módulos del prototipo debe ser sencillo e intuitivo.• La fuente y el tamaño de letra deben ser adecuados para que los usuarios puedan visualizar la información fácilmente.• La paleta de colores del prototipo se basa en colores oscuros predominantes y colores claros que sean utilizados de manera sutil para dar contraste.	

RNF 03	Disponibilidad de los registros diarios
Usuario:	Estudiante
Prioridad:	Alta
Descripción:	
Es necesario que los datos relacionados a los registros diarios se encuentren disponibles tanto localmente como en la base de datos del proyecto en línea.	
Criterios de aceptación:	
<ul style="list-style-type: none">• El prototipo debe mostrar los datos y promedios basados en los registros almacenados en el dispositivo para evitar consultas innecesarias a la base de datos del proyecto.• Para los datos y promedios basados en los registros globales de los usuarios del prototipo, se debe contar con acceso a Internet para realizar las consultas correspondientes a la base de datos del proyecto.	

2.1.2 Actores y roles del prototipo

En esta sección se presentan los actores y roles que fueron identificados al realizar el análisis de los requerimientos funcionales y no funcionales del prototipo. Existen dos roles distintos en el prototipo: estudiante y administrador. Dentro del rol de administrador, se encuentra el actor “Administrador Master”, el cual posee una cuenta con privilegios especiales que le permite acceder a funcionalidades específicas dentro del prototipo. En la Tabla 2.2 se muestra la clasificación de los actores en función de sus roles, así como las correspondientes acciones que se pueden realizar.

Tabla 2.2 Roles y actores del prototipo

Rol	Actor	Acciones
Estudiante	No registrado	<ul style="list-style-type: none"> • Visualizar interfaz de bienvenida. • Crear una nueva cuenta.
	Registrado	<ul style="list-style-type: none"> • Personalizar datos personales del perfil. • Realizar registro diario de tiempo y calidad de sueño. • Visualizar promedios y estadísticas de los registros diarios del sueño. • Cambiar contraseña de la cuenta.
Administrador	General	<ul style="list-style-type: none"> • Visualizar promedios y estadísticas de los registros enviados por los usuarios. • Visualizar los registros enviados por los usuarios. • Exportar datos con los registros diarios anonimizados.
	Máster	<ul style="list-style-type: none"> • Exportar datos de usuarios y sus registros diarios. • Aprobar solicitudes de cuentas de administradores. • Manejar códigos de seguridad para cuentas de administradores.

2.1.2.1 Rol Estudiante

El rol estudiante es clave en el diseño, pues será quien realice el monitoreo de su sueño y comparta esta información empleando el prototipo. Se identificaron dos actores dentro de este rol:

- **Estudiante no registrado:** usuario que instala por primera vez la aplicación, su funcionalidad se limita a visualizar interfaces informativas y la opción de crear una nueva cuenta. Para concluir con su registro debe ingresar un correo electrónico y contraseña. Adicionalmente, debe verificar el correo electrónico ingresado, mediante un enlace enviado por el proveedor de *backend*.
- **Estudiante registrado:** usuario que concluyó con su proceso de registro, tiene la capacidad de acceder a todos los módulos que brinde el rol estudiante.

2.1.2.2 Rol Administrador

El rol administrador se encuentra enfocado en el proceso de control y verificación de la participación de los usuarios del prototipo. En base al acceso sobre los datos de los usuarios y a la función de aprobar solicitudes para nuevas cuentas de administradores se identificaron dos actores (cuentas de administración), estos son:

- **Administrador general:** usuario que tiene la capacidad de visualizar promedios y estadísticas generales de los registros enviados por los estudiantes. Adicionalmente, puede tener acceso a dichos registros, tomando en cuenta que no podrá visualizar un campo que asocie un registro a un estudiante en particular.
- **Administrador máster:** única cuenta de administrador con privilegios y accesos especiales, tiene la capacidad de acceder a la información de los usuarios, como a los registros de sueño completos que son almacenados en el proveedor de *backend*. Adicionalmente, tiene la capacidad de aceptar solicitudes de nuevas cuentas de administración.

2.1.2.3 Diagramas de casos de uso para el prototipo de aplicación móvil

Los diagramas de casos de uso proporcionan una representación visual de las interacciones que tiene disponible el usuario con el prototipo [52]. En la Figura 2.2 se muestra el diagrama de casos de uso para el registro de una nueva cuenta de Estudiante o Administrador General. La cuenta de Administrador Máster ya se encuentra registrada y autorizada en el prototipo. Por otro lado, el Servicio de *backend* es empleado para realizar la gestión de la autenticación y el almacenamiento de la base de datos del prototipo.

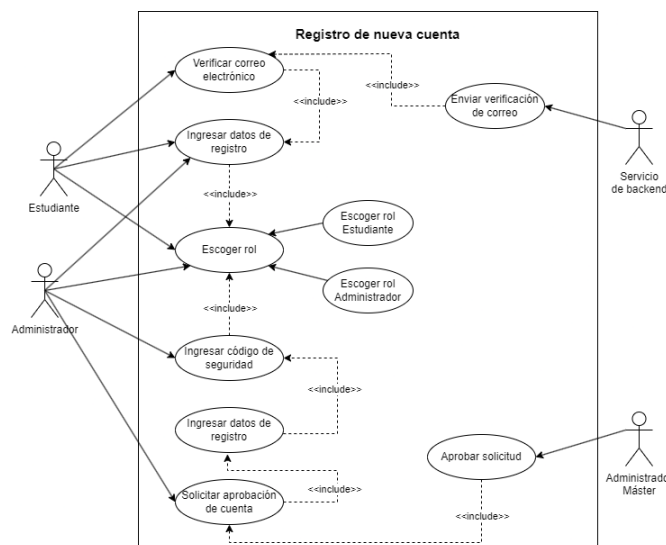


Figura 2.2 Casos de uso administrador y estudiante registro de cuenta

La Figura 2.3 representa los casos de uso para el registro diario del sueño que realiza el usuario Estudiante. El Servicio de backend permite almacenar el registro realizado diariamente.



Figura 2.3 Casos de uso estudiante registro diario de sueño

El diagrama del ANEXO III.3 parte (a) representa los casos de uso para la autenticación del usuario Estudiante y Administrador en el prototipo. El servicio de backend se encarga de verificar las credenciales de acceso. El diagrama del ANEXO III.3 parte (b) representa los casos de uso para la personalización del perfil de usuario para el Estudiante y Administrador, donde, interactúa el Servicio de *backend* para almacenar los datos. El diagrama del ANEXO III.3 parte (c) presenta los casos de uso para visualizar datos y promedios, la información a la que tienen acceso cada actor (Administrador Máster, Administrador General y Estudiante) se encuentra relacionada con los requerimientos identificados en la sección 2.1.1

2.1.3 Diseño de la interfaz del prototipo

En la presente sección se detallan los aspectos relacionados al apartado gráfico del prototipo. En primer lugar, se presenta la selección de la paleta de colores que va a ser utilizada en el prototipo. Los colores escogidos guardan relación con el requerimiento no funcional identificado en la historia de usuario *RNF 02 Usabilidad del Equipo*, acerca de usar colores oscuros predominantes y colores claros que faciliten el contraste entre elementos de la interfaz gráfica. En la Figura 2.4(a) se muestra la paleta de colores generada con la herramienta Colors [54].

La fuente escogida para la interfaz gráfica del prototipo es *Roboto*, desarrollada por Google para el lanzamiento del sistema operativo Android 4.0 y es una tipografía sobria, moderna y accesible [55]. Con respecto al logo seleccionado para representar al prototipo, se adoptó la imagen de un búho (Figura 2.4 (a)), debido a su estrecha relación con la noche, motivo por el cual lo convierte en una elección apropiada para representar al prototipo. Para la búsqueda y descarga del ícono se utilizó la herramienta de *Flaticon* utilizando la palabra clave “owl”. El diseño escogido fue publicado por *Good Ware* [56] y tiene una licencia de uso comercial y personal, siempre y cuando se proporciona la correspondiente atribución. En la Figura 2.4 (b) se muestra el ícono escogido.

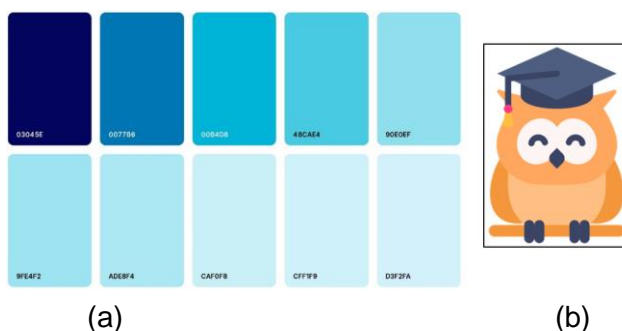


Figura 2.4 Paleta de colores (a) e ícono del prototipo (b)

2.1.3.1 Módulos del prototipo

En este apartado se detallan los módulos que debe tener el prototipo. Las distintas funcionalidades identificadas para el prototipo se encuentran especificadas en diferentes módulos según el rol de usuario. En este sentido, se cuenta con módulos particulares para cada rol y módulos comunes. Adicionalmente, ciertos módulos que se encuentran disponibles únicamente para la cuenta del Administrador Máster. En la Tabla 2.3 se muestran los módulos en común para el rol Estudiante y Administrador:

Tabla 2.3 Módulos comunes Estudiante y Administrador

Módulo	Descripción
Registro	Permite al usuario registrarse en el prototipo utilizando un correo electrónico y contraseña.
Autenticación	Permite al usuario iniciar sesión. Adicionalmente, tiene accesos directos al módulo de registro y cambiar contraseña.
Perfil	Permite al usuario visualizar y modificar la información de su perfil personal.
Cambiar Contraseña	Permite al usuario cambiar la contraseña de su cuenta personal empleando el correo electrónico registrado en el prototipo.
Dashboard	Permite al usuario visualizar distintos gráficos y promedios relacionados con la información del rol que poseen.
Ayuda	Permite al usuario contactar directamente con el Administrador Máster.

En la Tabla 2.4 se detallan los módulos del rol Estudiante:

Tabla 2.4 Módulos rol Estudiante

Módulo	Descripción
Administración de datos	Permite al Estudiante administrar sus datos, puede eliminar los registros que envió, sincronizar sus registros si inicia sesión desde otro dispositivo y eliminar su cuenta personal.
Cómo usar	Permite al Estudiante acceder a información paso a paso de cómo realizar el monitoreo y registro de sueño.
Permisos	Permite al Estudiante conceder y retirar el permiso al prototipo para acceder a los datos de <i>Google Fit</i> .
Registro diario	Permite al Estudiante visualizar su tiempo de sueño y enviar el registro diario, para lo cual se le muestra un cuestionario para calcular su calidad de sueño.
Visualización de registros	Permite al Estudiante visualizar los registros que envió en los últimos 7, 14 y 30 días.
Visualización de promedios semanales	Permite al Estudiante visualizar el promedio de tiempo que durmió según el día de la semana.
Información	Permite al Estudiante conocer el porqué del avatar asignado a su perfil, tomando en cuenta que esto guarda relación con su calidad de sueño.

En la Tabla 2.5 se muestran los módulos que pertenecen al rol de Administrador:

Tabla 2.5 Módulos rol Administrador

Módulo	Descripción
Visualizar y exportar datos	Permite visualizar en una tabla los registros de sueño diarios enviados por los Estudiantes. Permite exportar dicha tabla en un archivo csv. Los registros mostrados no poseen los identificadores de los Estudiantes.
Verificación código de seguridad	Permite al administrador ingresar el código de seguridad que le permite acceder al módulo de registro. El código es proporcionado por el Administrador Máster.
Información semanal	Permite visualizar el promedio de tiempo de sueño por día de la semana en base a todos los registros enviados por los Estudiantes.

En el caso especial de la cuenta de Administrador Máster se tienen módulos disponibles únicamente para esta cuenta con privilegios. En la Tabla 2.6 se detallan estos módulos:

Tabla 2.6 Módulos Administrador Máster

Módulo	Descripción
Administración	Permite al Administrador Máster aceptar solicitudes de cuentas para Administradores Generales.
Visualizar y exportar datos	Permite visualizar y exportar los registros de sueño enviados por los usuarios. Al exportar cada registro a un archivo csv se tendrá el identificador del usuario que lo envió.
Usuarios	Permite visualizar y exportar el listado de los Estudiantes registrados en el prototipo.
Códigos de seguridad	Permite visualizar y editar los códigos de seguridad que serán compartidos a los usuarios que deseen crear una cuenta de Administrador General.

2.1.3.2 Diseño interfaz gráfica

Para el diseño de las interfaces gráficas se empleó la herramienta de *Figma*, herramienta que facilita el maquetado de prototipos [57]. Se tomó en cuenta la paleta de colores especificada en la Figura 2.4, al igual que el tipo de fuente escogida para el prototipo.

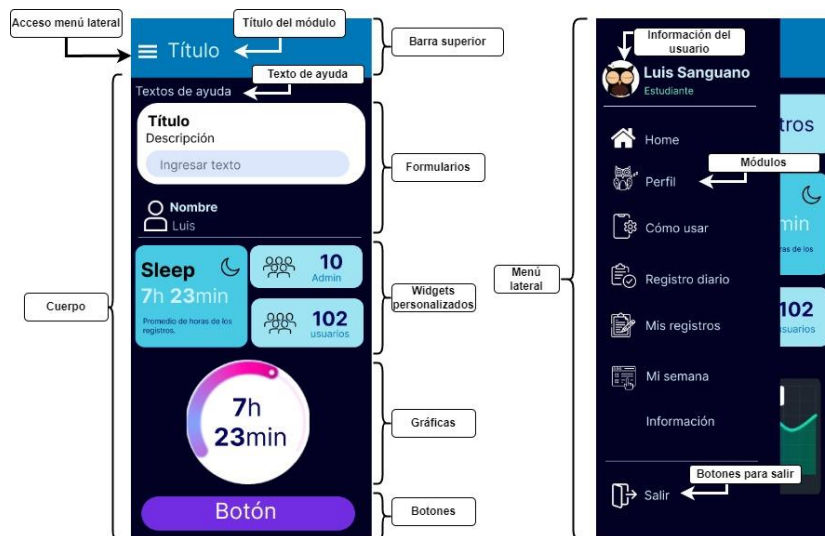


Figura 2.5 Diseño de elementos base para la interfaz gráfica

El diseño de la interfaz gráfica para los módulos se conforma de dos elementos principales: una barra superior y un cuerpo. La barra superior contiene un botón que despliega un menú lateral para acceder a todos los módulos especificados. Adicionalmente, cuenta con un título referente al módulo que se esté visualizando. Por otro lado, el cuerpo puede contener varios elementos: textos de ayuda, gráficas, *widgets* personalizados con información, formularios, botones, entre otros. En la Figura 2.5 se muestra una plantilla básica con los elementos más utilizados en el diseño de los módulos.

El diseño de los demás módulos como su respectiva descripción detallada se encuentra en el ANEXO III.4. En la Figura 2.6 se muestran los mockups diseñados como base del prototipo, estos diseños son referenciales y en la implementación pueden ser modificados para lograr mejores resultados visuales.

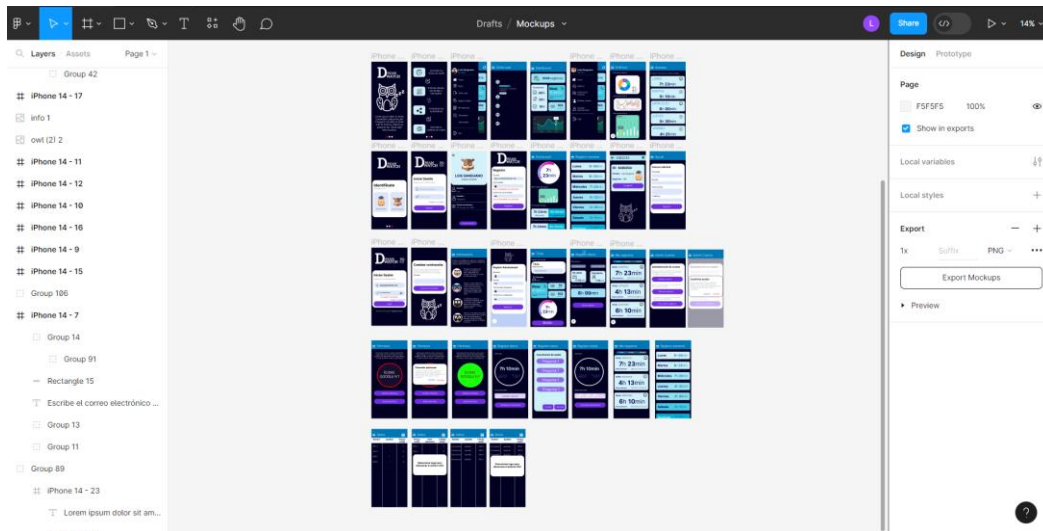


Figura 2.6 Diseños de la interfaz gráfica en *Figma*

2.1.4 Funcionalidad del prototipo

Para detallar la funcionalidad que tiene el prototipo se hace uso de los diagramas UML (*Unified Modeling Language*) de actividades. Este tipo de diagramas permiten definir flujos de procesos que son fáciles de entender. Los diagramas que se presentarán en esta sección fueron elaborados con la herramienta de *Draw.io* [58]. Los diagramas de actividades presentados en las siguientes subsecciones ilustran los procesos más importantes para el prototipo.

2.1.4.1 Proceso de registro y autenticación

Para el proceso de registro y autenticación participan cuatro partes: Usuario, Prototipo, Firebase y Administrador Máster. Es importante recalcar que el apartado “Usuario” hace referencia a los actores “Estudiante” y “Administrador General”. Por otro lado, el “Administrador Máster” ya posee una cuenta creada previo al proceso detallado en esta sección, por lo cual no es encasillada dentro de “Usuario”, pues esta cuenta será la encargada de aprobar solicitudes de nuevas cuentas de administradores.

El proceso comienza cuando el usuario ejecuta el prototipo, al iniciar se le mostrará una pantalla de introducción con información básica como se muestra en la Figura 2.7. Posteriormente, dependiendo de su rol deberá escoger el inicio de sesión adecuado.

Independiente del rol, para iniciar sesión el módulo de autenticación solicita el correo electrónico registrado y la correspondiente contraseña.

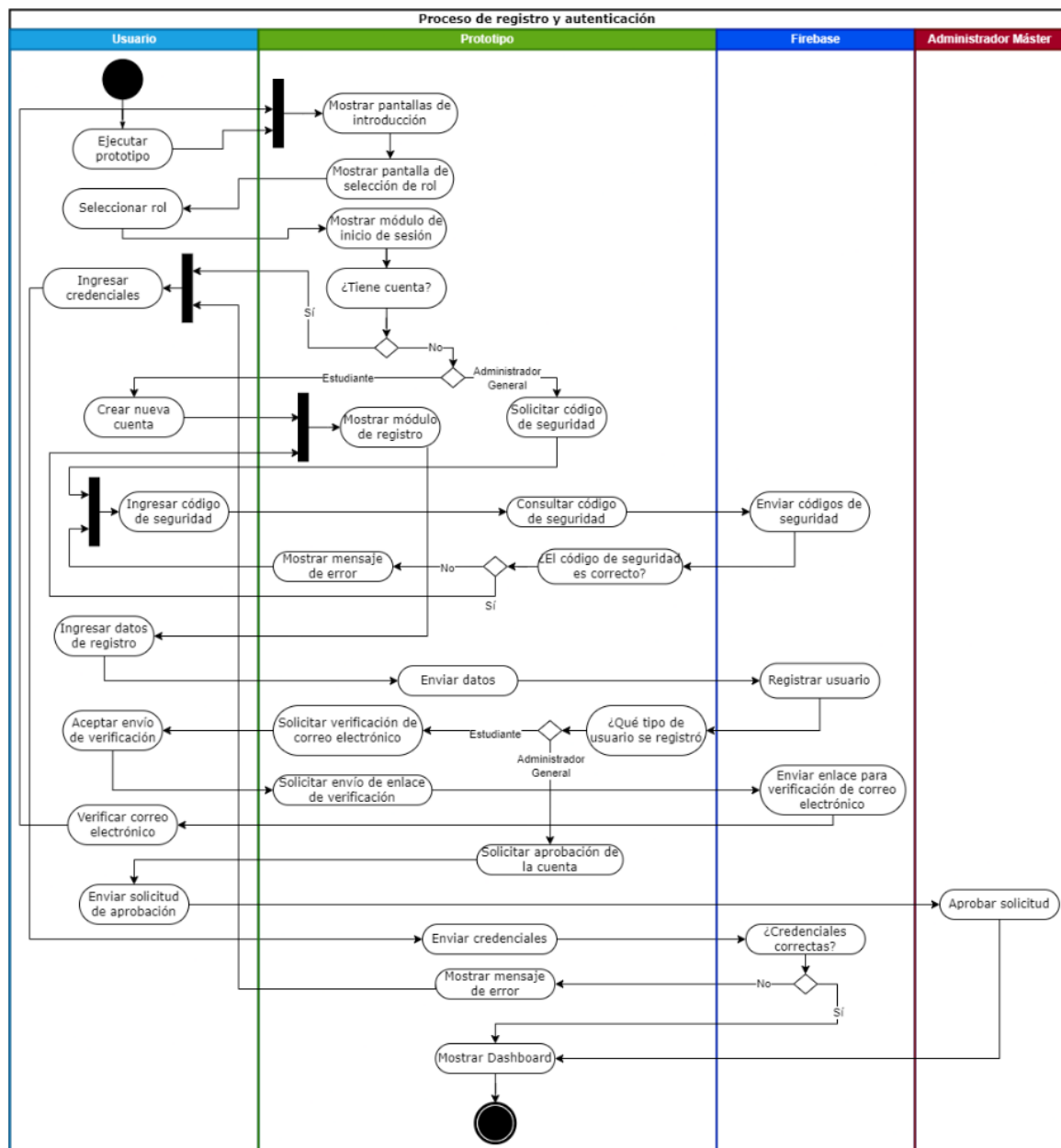


Figura 2.7 Diagrama de actividades proceso de registro y autenticación

El proceso de registro difiere para la cuenta de Estudiante y del Administrador General. En el primer caso, un nuevo Estudiante puede acceder al formulario de registro libremente y completar los campos del correo y contraseña; sin embargo, para terminar con el proceso de registro deberá verificar su cuenta de correo.

En cambio, para que una cuenta de Administrador General inicie el proceso de registro, primeramente, debe ingresar un código de seguridad proporcionado por el Administrador

Máster para acceder al formulario de registro y completar el correo y contraseña. Posteriormente, se debe solicitar la aprobación de la cuenta por parte del Administrador Máster. Finalmente, se mostrará a cada usuario el Dashboard correspondiente a su rol dando fin al proceso.

El proveedor *Firebase* brinda los servicios de autenticación y almacenamiento denominados: *Authentication* y *Cloud Firestore* correspondientemente. Estos servicios facilitan la implementación del *backend* del prototipo permitiendo realizar la verificación de las credenciales y el almacenamiento de las cuentas de los usuarios en la nube. En la Figura 2.7 se detalla a el proceso de registro y autenticación, tomando en cuenta todas las actividades identificadas para cada parte de este proceso.

2.1.4.2 Proceso de registro de sueño diario

En el proceso de registro de sueño diario participan cuatro partes: Estudiante, Prototipo, *Google Fit* y *Firebase*. Adicionalmente, involucra dos módulos: Registro Diario y Permisos. Para que el prototipo pueda acceder a los datos de sueño del Estudiante, primeramente, debe conceder los permisos necesarios para acceder a los datos de su cuenta de *Google Fit*. Además, de haber registrado sus horas de sueño de la noche anterior en su cuenta personal empleando *Sleep As Android* o la propia aplicación de *Google Fit*.

Sleep As Android permite a los Estudiantes estimar su tiempo total de sueño y compartir los resultados con la cuenta de *Google Fit*. En la Figura 2.9 la actividad “Ingresar horas de sueño en *Google Fit*” se contempla el registro de la información de sueño del Estudiante a través de la aplicación de *Sleep As Android* o *Google Fit*. Una vez registrada la información de sueño en *Google Fit*, el Estudiante podrá visualizar dicha información y podrá acceder al cuestionario de sueño. Al concluir este cuestionario se procederá a guardar el registro en local y en *Firebase*. Finalmente, el Estudiante actualizará la pantalla y obtendrá retroalimentación relacionada a su calidad de sueño.

2.1.4.3 Proceso de visualización de promedios y gráficas – Estudiante

El proceso de visualización de promedios y gráficas para el Estudiante comienza el momento que este accede al prototipo. El primer módulo que visualiza es el de Dashboard, en este módulo se presentan varios promedios tanto de los registros de sueño diarios que envió el Estudiante, como de los registros compartidos por los usuarios del prototipo en *Firebase*. Además del Dashboard, el Estudiante podrá acceder a otros módulos que le permitan visualizar promedios semanales e inclusive los registros enviados de los últimos 7, 14 y 30 días.

Previo a presentar estos datos el prototipo deberá procesar las consultas realizadas tanto a la base de datos local como a la de *Firestore*, y en el caso de no contar con información, se mostrarán valores en cero tanto en las gráficas como en los promedios personales del Estudiante. El resto de los detalles se especifica en la Figura 2.8.

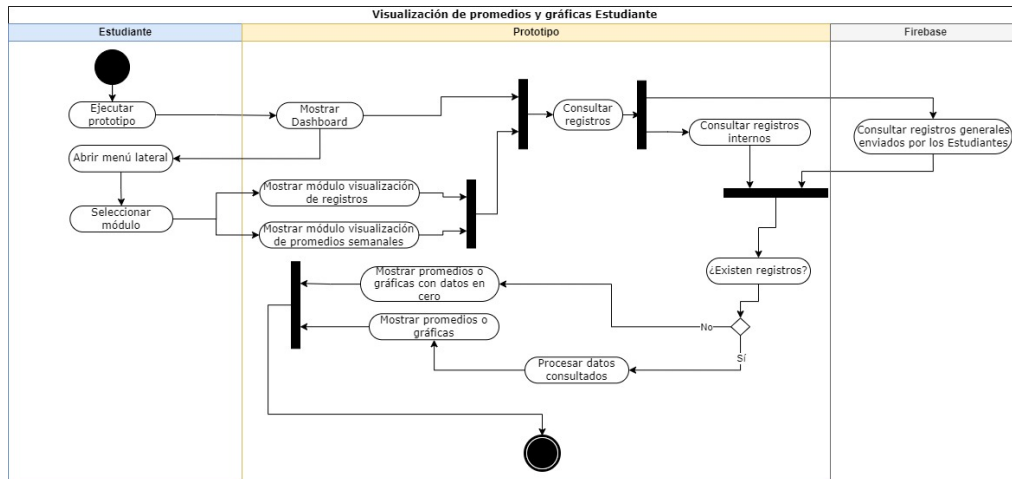


Figura 2.8 Diagrama de actividades proceso de visualización de promedios y gráficas del estudiante

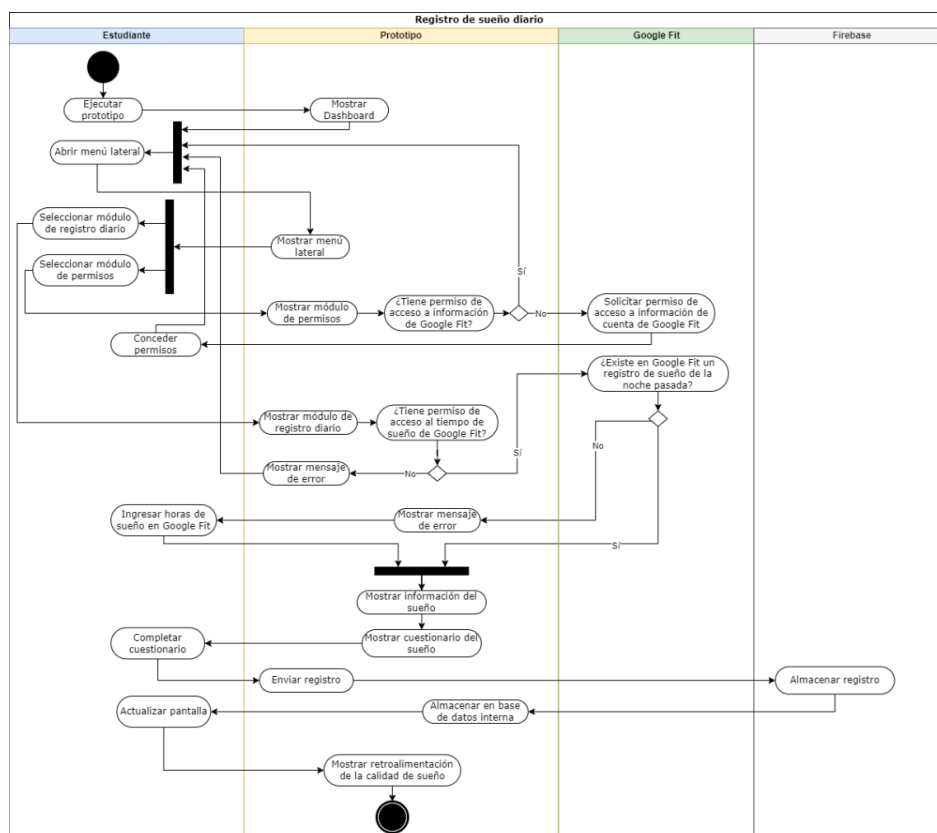


Figura 2.9 Diagrama de actividades proceso de registro de sueño diario

2.1.4.4 Proceso de visualización de promedios y gráficas - Administrador

En este proceso es importante aclarar que tanto el Administrador General como el Administrador Máster comparten el módulo de Dashboard, visualizando los mismos promedios y gráficas. Las consultas a las bases de datos son realizadas únicamente en *Firestore*, pues en este proveedor se almacenan los datos de los usuarios y los registros de sueño diarios de los Estudiantes. Se adiciona una nueva actividad que es la capacidad de exportar datos, en el caso del Administrador Máster puede exportar la información de los Estudiantes registrados, como también los registros de sueño con el identificador del Estudiante que lo envió. En cambio, el Administrador general únicamente puede exportar los registros de sueño sin el identificador del Estudiante. En la **Figura 2.10** se visualiza todo este proceso.

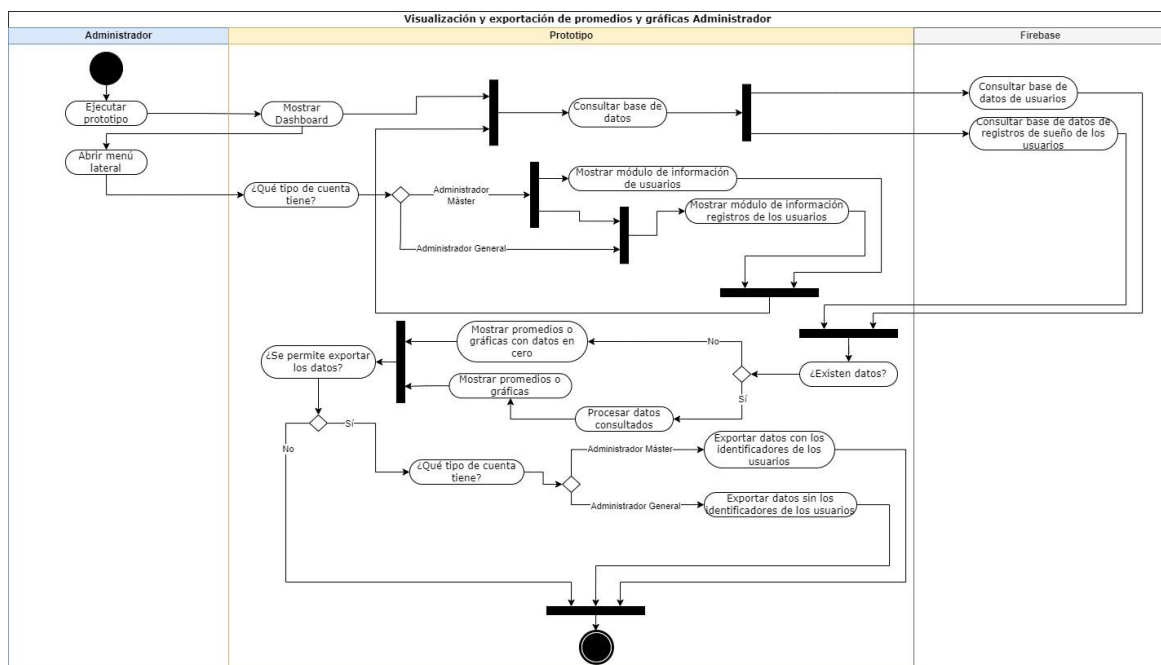


Figura 2.10 Diagrama de actividades proceso de visualización de promedios y gráficas Administrador

2.1.5 Esquema de la base de datos no relacional del prototipo

En la presente sección se muestran los esquemas utilizados para la organización de la base de datos del prototipo. En la Figura 2.11 se presentan los esquemas para los usuarios, registros y códigos de seguridad empleados para la organización de la información. El servicio de *Cloud Firestore* provee de bases de datos no relaciones, las cuales trabajan con documentos y colecciones. Los documentos pueden contener diferentes campos con una gran variedad de valores como: cadenas de caracteres, mapas, etc.

En la Figura 2.11 (a) se presenta el esquema para el Usuario, este contiene un identificador único tomado el momento del registro de su cuenta. El campo de “administrador” permite realizar la gestión de roles dentro del prototipo. El campo “verificación” permite manejar la gestión de la aprobación de las cuentas de Administradores Generales. Por otro lado, el campo “imagen” almacena la ruta de la imagen para el avatar que tiene el Estudiante, como retroalimentación de su calidad de sueño. El resto de los campos son datos personales del usuario.

Para almacenar los registros de sueño diarios de los estudiantes se plantea el esquema de Registros de la Figura 2.11 (b). Los campos de “registroId”, “sexo”, “edad”, “nivelEducacion”. “areaEstudio” guardan los datos del Estudiante que envió el registro. El campo “fechaEnvio” guarda la fecha en la cual fue enviado el registro. Por otro lado, “calidad”, “nivelCansado” y “sentimiento”, son campos que guardarán la información tomada por formularios presentados al Estudiante el momento de realizar el registro diario en el prototipo. El resto de los campos son destinados a la información de sueño que entrega *Google Fit*.

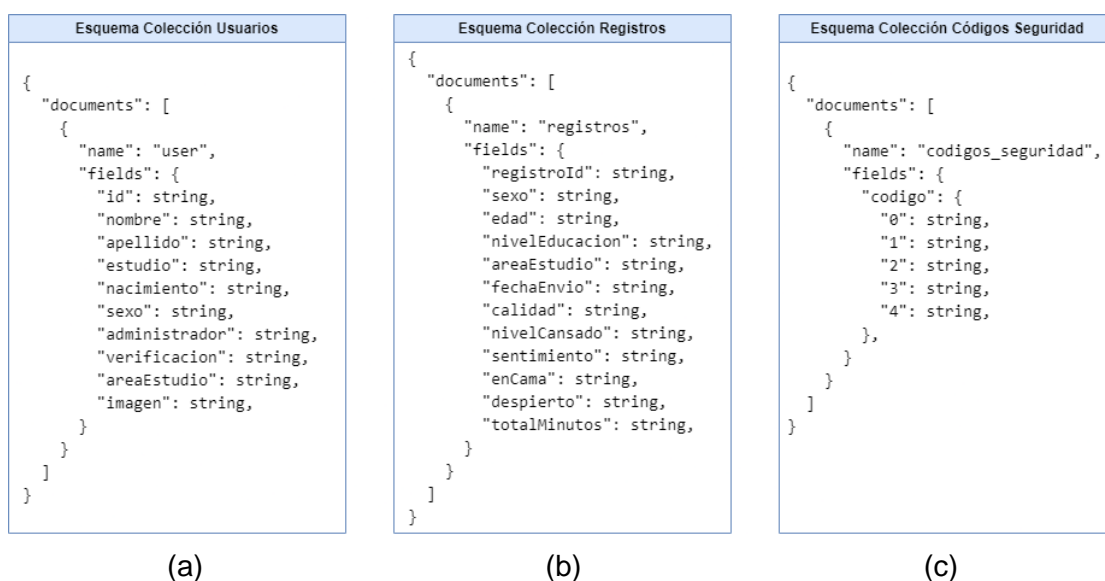


Figura 2.11 Esquemas de las bases de datos no relacionales

Finalmente, la colección de Códigos de Seguridad presentado en la Figura 2.11 (c), permite almacenar los códigos empleados para acceder al registro de una nueva cuenta de Administrador General. Son cinco códigos almacenados en el campo “código” que guarda un mapa de 5 elementos de tipo *string*.

2.1.6 Arquitectura del prototipo

Finalmente, la arquitectura propuesta para realizar el envío, almacenamiento y lectura de los registros diarios de sueño con Firebase, Google Fit y el proveedor externo para estimar el tiempo de sueño (Sleep As Android) se encuentra en la Figura 2.12.

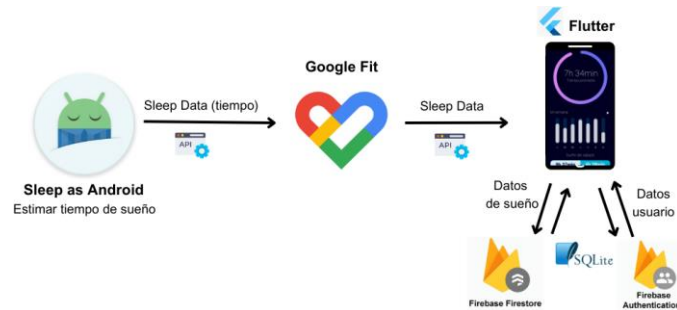


Figura 2.12 Arquitectura del prototipo

2.2 Implementación del prototipo

En la presente sección se desarrolla la implementación del prototipo en base al diseño planteado en la sección 2.1. La implementación es realizada con el *Framework* de *Flutter* y los servicios de *Firebase*. Primeramente, se procede a detallar la arquitectura de tres capas adoptada en el desarrollo de software. Posteriormente, la instalación de las herramientas empleadas para la codificación del prototipo. Luego, se detalla la codificación de la interfaz gráfica del módulo Dashboard del prototipo a manera de ejemplo. Finalmente, la implementación de las bases de datos no relacionales. En la **Figura 2.13** se detalla la instancia actual del Tablero Kanban, las tareas relacionadas con la fase de diseño se encuentran concluidas. Las tareas relacionadas a la implementación se encuentran en fase de proceso.

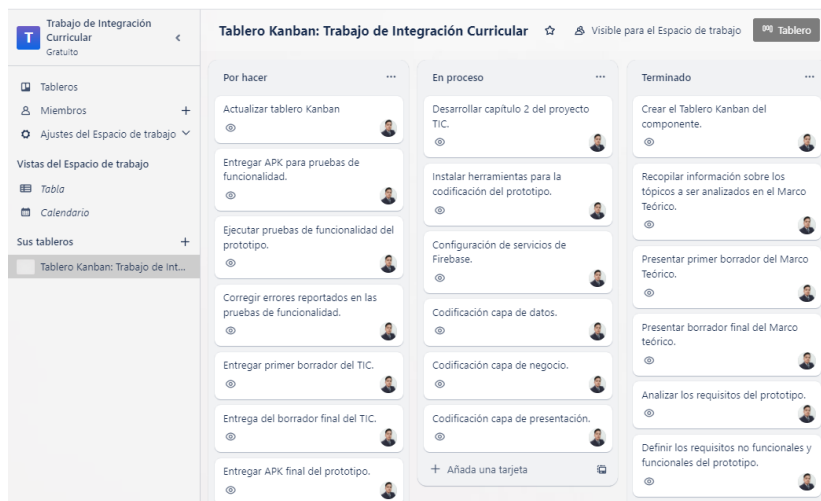


Figura 2.13 Tablero Kanban etapa de implementación

2.2.1 Instalación de herramientas

La implementación del prototipo se realiza en el sistema operativo Windows 11 Home con versión 22H2. El editor de código empleado es Visual Studio Code en su versión 1.79.2, la guía de instalación se encuentra en su página oficial [59].

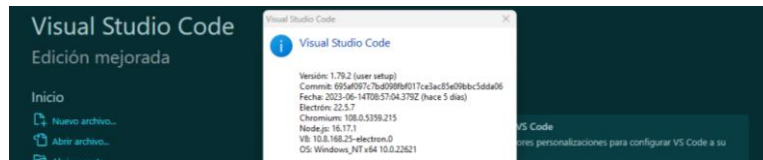


Figura 2.14 Instalación de Visual Studio Code

En Visual Studio Code se instalan varias extensiones para trabajar con el *Framework* de *Flutter*. La Tabla 2.7 detalla las extensiones instaladas.

Tabla 2.7 Extensiones Visual Studio Code para el desarrollo en Flutter

Extensión	Editor	Descripción
Flutter	dartcode.org	Extensión que brinda soporte para el trabajo con el Framework de Flutter en el desarrollo de aplicaciones móviles.
Dart	dartcode.org	Extensión de soporte para el lenguaje de programación Dart. Brinda herramientas para editar, refactorizar y ejecutar aplicaciones móviles con Flutter.

Para el desarrollo del prototipo se hace uso del *Framework* de *Flutter*. La guía de instalación se puede encontrar en la página oficial de *Flutter* [60]. Una vez instalado el SDK de *Flutter*, es necesario actualizar la variable de entorno *Path*, agregando el directorio en donde se instaló *Flutter*. Finalmente, se verifica la instalación correcta de Flutter, para lo cual se usa al comando `flutter --version`, en la Figura 2.15 se muestra el resultado de ejecutar este comando, en este caso el prototipo se desarrolla con la versión 3.7.3 de *Flutter*.

```
C:\Windows\System32>flutter --version
Flutter 3.7.3 • channel stable • https://github.com/flutter/flutter.git
Framework • revision 9944297138 (5 months ago) • 2023-02-08 15:46:04 -0800
Engine • revision 248290d6d5
Tools • Dart 2.19.2 • DevTools 2.20.1
```

Figura 2.15 Verificación de la instalación de Flutter

Para el emulador virtual se hace uso de *Android Studio*. La guía de instalación se detalla en su página oficial [61]. Una vez instalado se procede a crear un dispositivo virtual desde el apartado de *Virtual Device Manager* > *Create Device*. Las configuraciones del dispositivo empleado en el desarrollo del prototipo se detallan en la Tabla 2.8.

Tabla 2.8 Detalles dispositivo virtual Android Studio

Característica	Detalle
Modelo	Pixel 4
API	30
Versión de Android	11
RAM	1536 MB
Almacenamiento interno	6144 MB
Pantalla	1080x2280 440dpi

2.2.2 Creación del proyecto para el prototipo en Visual Studio Code

Una vez instaladas las herramientas se crea el proyecto en Flutter se procede a crear el proyecto para el prototipo denominado “dream_watch”. Los pasos se muestran en la Figura 2.16, paso (a) crear un nuevo proyecto, paso (b) seleccionar el tipo de proyecto y paso (c) colocar el nombre del proyecto

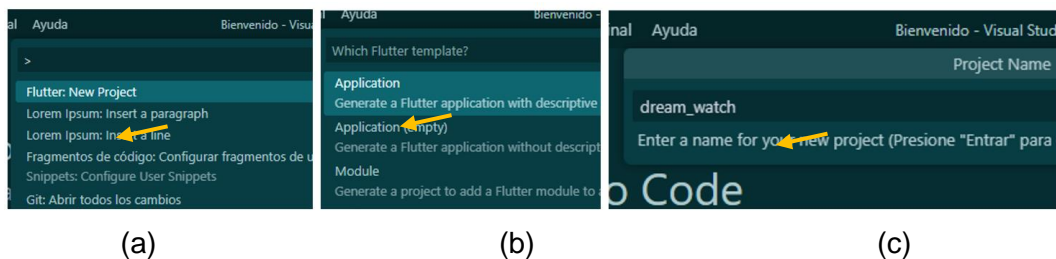


Figura 2.16 Creación del proyecto de Flutter en Visual Studio Code

Para ejecutar el ejemplo que es creado por defecto, se debe seleccionar el dispositivo virtual de Android Studio Figura 2.17(a) y presionar la tecla F5. Empezará a compilar el código y se visualizará el ejemplo como se muestra en la Figura 2.17(b).

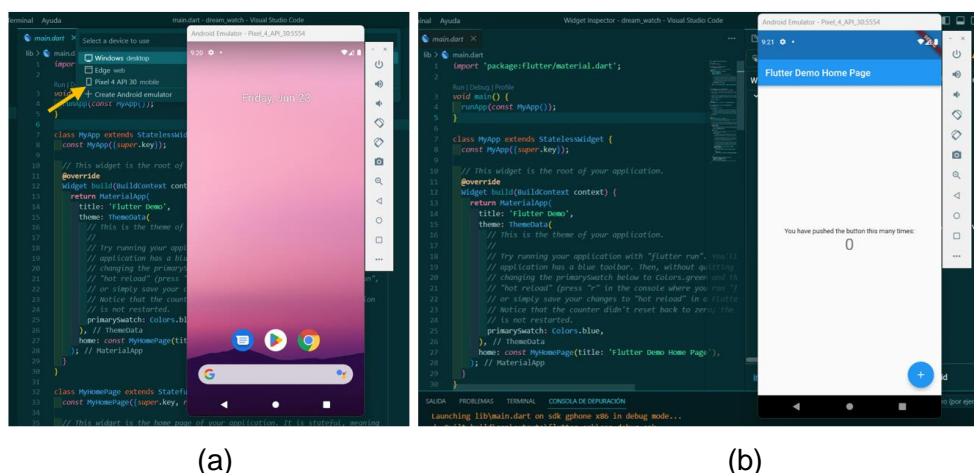


Figura 2.17 Ejecución del ejemplo en el dispositivo virtual

2.2.3 Arquitectura en capas

La codificación del prototipo se basa en una arquitectura de tres capas: capa de datos, capa de negocio y capa de presentación. El proyecto creado en la sección 2.2.2 es adaptado a esta arquitectura. En el directorio *lib* que contiene el código fuente de la aplicación de *Flutter* se crean tres nuevos directorios como se muestra en la Figura 2.18.

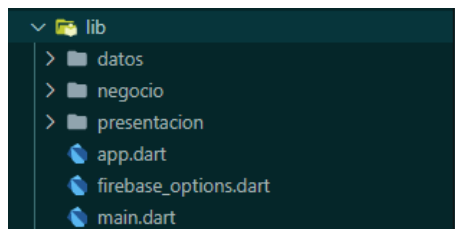


Figura 2.18 Estructura de archivos en capas

En la Figura 2.18 se observan 3 archivos que se encuentran dentro del directorio *lib*. Los archivos *main.dart* como *app.dart* son los puntos de entrada a la aplicación; mientras que el archivo de *firebase_options.dart* maneja la integración con *Firestore*, este archivo es creado automáticamente en la sección 2.2.4

2.2.4 Integración de Firebase

Para integrar *Firestore* al proyecto se debe contar con tres requisitos: instalar el SDK de *Flutter*, instalar *Firestore CLI*, y tener un proyecto de *Flutter* creado. Los dos primeros requisitos se completaron en la sección 2.2.1 y 2.2.2. La instalación del *Firestore CLI* se encuentra en su página oficial [62]. Una vez cumplidos los requisitos se crea un proyecto en la consola de *Firestore* (Figura 2.19 (a)) y se integra directamente con *Flutter* como se muestra en la Figura 2.19 (b).

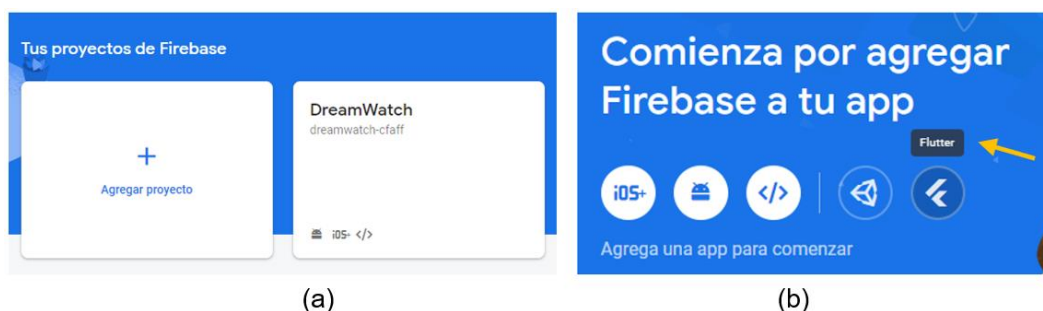


Figura 2.19 Proyecto creado en *Firestore* (a) e implementación con *Flutter* (b)

Al concluir con la guía de *Firestore* se procede a verificar si el proyecto se encuentra integrado correctamente, para lo cual se ejecuta el comando `firebase projects:list`. En la Figura 2.20 se muestra el proyecto *DreamWatch* integrado correctamente.

```
C:\Windows\System32>firebase projects:list
✓ Preparing the list of your Firebase projects
```

Project Display Name	Project ID	Project Number	Resource Location ID
DreamWatch ←	dreamwatch-cfaff	978094618627	us-central

Figura 2.20 Verificación de la integración con Firebase

2.2.4.1 Configuración de servicios en Firebase

En la consola de *Firebase* se deben habilitar los servicios que empleará el prototipo para su funcionamiento. Estos servicios son: *Authentication* y *Cloud Firestore*.

a) Configuración de *Authentication*

Desde la consola de *Firebase* se integra el servicio *Authentication* y posteriormente se habilita el proveedor de acceso. El prototipo emplea para la autenticación el proveedor con correo electrónico y contraseña como se muestra en la Figura 2.21.

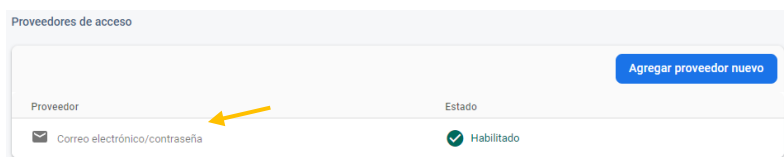


Figura 2.21 Configuración del proveedor de acceso para la autenticación en Firebase

b) Configuración de *Cloud Firestore*

Desde la consola de *Firebase* se habilita *Cloud Firestore*. Después, se inician tres colecciones acordes al diseño en la sección 2.1.5 con los respectivos campos. En la Figura 2.22(a) se muestran las colecciones inicializadas en el proyecto de *Firebase*. Adicionalmente, se configuran las reglas básicas de acceso y seguridad de las colecciones Figura 2.22 (b).



(a)

(b)

Figura 2.22 Configuración de colecciones (a) y reglas de seguridad (b) Firebase

Las reglas de seguridad especificadas para la lectura y escritura de las colecciones creadas se muestran en la Figura 2.22 (b), donde, las colecciones /codigos_seguridad/ y /registros/ permiten la escritura y lectura de todos sus documentos. En cambio, la lectura y escritura de los documentos en la colección /user/ se encuentra limitada para aquellas solicitudes que posean un identificador “myId”, identificador que es único para cada usuario. Adicionalmente, se incluye una regla de lectura para la colección /user/.

2.2.5 Integración de la API de Google Fit

La guía para integrar la API de Google Fit se encuentra detalla en la referencia [63]. Para esta integración es necesario realizar la firma del prototipo para obtener la huella digital SHA-1 del certificado de depuración o lanzamiento. En la Figura 2.23 se muestra el comando empleado y el resultado, por razones de seguridad se oculta el SHA-1 obtenido.

```
C:\Users\Luis\.android>keytool -list -v -keystore debug.keystore -alias androiddebugkey -storepass android -keypass android
Alias name: androiddebugkey
Creation date: 16 oct 2022
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: C=US, O=Android, CN=Android Debug
Issuer: C=US, O=Android, CN=Android Debug
Serial number: 1
Valid from: Sun Oct 16 16:51:35 ECT 2022 until: Tue Oct 08 16:51:35 ECT 2052
Certificate fingerprints:
    SHA1: CC:D6:                                     :3C
    SHA256: 8E:46:                                     4E:5A:09:B5
```

Figura 2.23 Huella digital certificado de *debug*

De acuerdo con la guía oficial [63] es necesario contar con un ID para el cliente OAuth 2.0. Este identificador permite autenticar y autorizar aplicaciones que interactúan con los servicios de Google, empleando el protocolo OAuth 2.0 [64]. Al iniciar un proyecto en Firebase este identificador se crea automáticamente tomando el nombre del prototipo y la huella digital [65] como se muestra en la Figura 2.24.



Figura 2.24 Configuración del SDK en Firebase

Desde el panel de control de *Google Cloud* se administran diferentes servicios como por ejemplo la API de Google Fit [66]. Al ingresar se visualizará el proyecto *DreamWatch* con el ID *dreamwatch-cfaff* (Figura 2.25 (a)), proyecto creado en el apartado 2.2.4 Finalmente, se procede con la configuración de la pantalla de consentimiento para los permisos de *Google Fit*. En este punto se habilitan los permisos relacionados a *Fitness API* para acceder a los datos de sueño como se muestra en la Figura 2.25 (b).

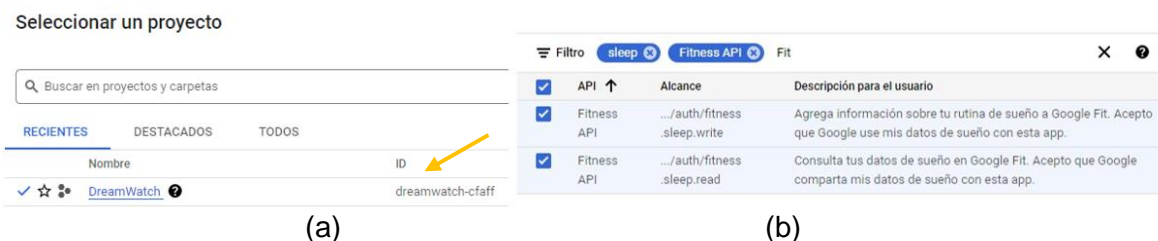


Figura 2.25 Proyecto DreamWatch en la consola de *Google Cloud* (a) y permisos de *Google Fit* para datos de sueño (b)

2.2.6 Codificación del prototipo

En esta sección se detallan aspectos importantes de la codificación del prototipo. El código completo se encuentra en el ANEXO IV.2. En la Figura 2.26 se muestran todos los paquetes utilizados por el prototipo para su funcionamiento, esto paquetes son agregados en el apartado de dependencias en el archivo *pubspec.yaml* del proyecto.

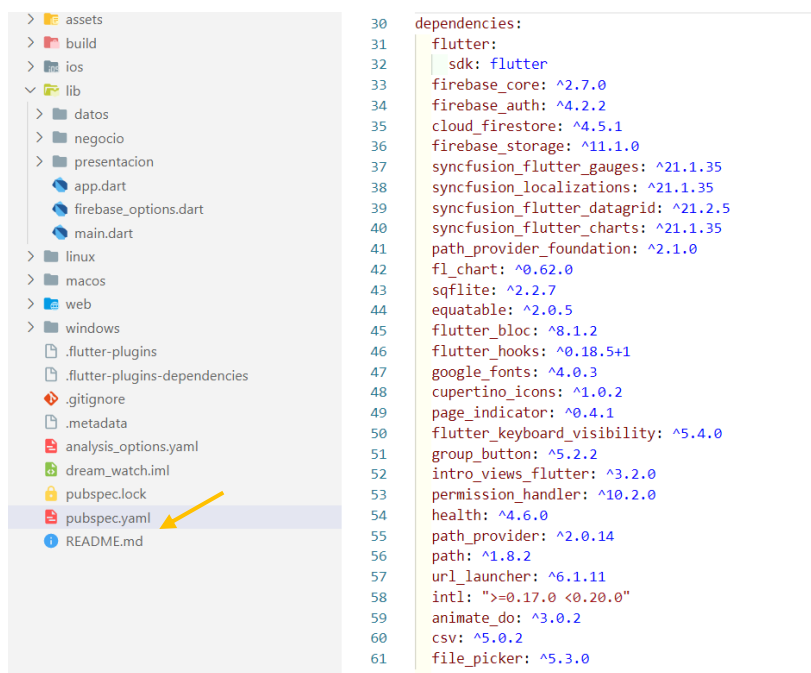


Figura 2.26 Paquetes empleados en el desarrollo del prototipo

2.2.6.1 Arquitectura módulo de autenticación

Para controlar la autenticación dentro del prototipo se toma como base el *Business Logic Components* (BLoC), que es un patrón de diseño que permite gestionar los cambios de estado que se producen en el prototipo [67]. Para implementar esta arquitectura, se emplea el “Cubit”, que es una clase que permite gestionar el estado de la autenticación dentro de la aplicación. De esta manera, la sesión se mantendrá abierta cada vez que el usuario haga uso del prototipo, mostrando las pantallas adecuadas a cada estado, sin la necesidad de volver a ingresar sus credenciales. La capa de negocio se comunica con la de presentación mediante eventos dependiendo del estado que emita el “cubit”. La Figura 2.27 muestra la arquitectura implementada en el prototipo.

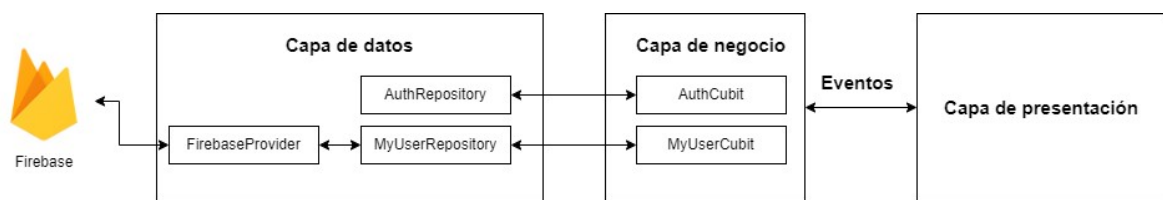


Figura 2.27 Arquitectura del módulo de autenticación

En la Figura 2.28 se muestran algunas de las funciones codificadas para el prototipo, la Figura 2.28 (a) muestra el mapeo del objeto *User* desde y hacia *Firebase*. En Figura 2.28 (b) se muestran algunas de las funciones dentro del *FirebaseProvider*, estas funciones permiten, por ejemplo, conocer el usuario que se encuentra autenticado en el prototipo con *FirebaseAuth.instance.currentUser*. Adicionalmente, consultar los datos del usuario y guardar un nuevo usuario, estas funciones son importantes para los demás módulos, como por ejemplo el de perfil. Todas estas funciones se encuentran en la capa de datos.

```

1 //Funciones para convertir un mapa al objeto para guardar en Firebase
2 Map<String, Object?> toFirebaseMap() {
3   return <String, Object?>{
4     'id': id,
5     'nombre': nombre,
6     'apellido': apellido,
7     'estudio': estudio,
8     'nacimiento': nacimiento,
9     'sexo': sexo,
10    'administrador': administrador,
11    'verificacion': verificacion,
12    'areaEstudio': areaEstudio,
13    'imagen': imagen
14  };
15 }
16 //constructor del objeto que viene de Firebase
17 MyUser.fromFirebaseMap(Map<String, Object?> data)
18   : id = data['id'] as String,
19   nombre = data['nombre'] as String,
20   apellido = data['apellido'] as String,
21   estudio = data['estudio'] as String,
22   nacimiento = data['nacimiento'] as String,
23   sexo = data['sexo'] as String,
24   administrador = data['administrador'] as String,
25   verificacion = data['verificacion'] as String,
26   areaEstudio = data['areaEstudio'] as String,
27   imagen = data['imagen'] as String;
28 }
29
        
```

(a)

```

1 //Verificar que el usuario inicio sesión
2 User get currentUser() {
3   //currentUser permite conocer el estado en Firebase del usuario
4   final user = FirebaseAuth.instance.currentUser;
5   if (user == null) throw Exception("El usuario no ha iniciado sesión");
6   return user;
7 }

1 //Consultar el usuario en Firebase
2 Future<MyUser?> getMyUser() async {
3   //Consulta por documento con el ID del usuario
4   final snapshot = await firestore.doc('user/${currentUser.uid}').get();
5   //Retorna un objeto de tipo usuario
6   if (snapshot.exists) return MyUser.fromFirebaseMap(snapshot.data!);
7   return null;
8 }

1 //Guardar un nuevo usuario
2 Future<void> saveMyUser(MyUser user) async {
3   final ref = firestore.doc('user/${currentUser.uid}');
4   await ref.set(user.toFirebaseMap(), SetOptions(merge: true));
5 }
6
        
```

(b)

Figura 2.28 Código de ejemplo módulo de autenticación

La capa de negocio se comunica por medio de eventos generados por los estados que emite el “*cubit*”, la emisión de estos estados es realizada en la clase *AuthCubit*, en la Figura 2.29 (a) se muestra la función encargada de la autenticación que dependiendo del estado del usuario dentro de la aplicación emite cierto estado. Por otro lado, en la Figura 2.29 (b) se muestra un ejemplo de cómo se hace uso de los estados en la capa de presentación, para acceder al estado y los datos del usuario se hace uso de *BlocBuilder*.



```

1 //Función para manejar la autenticación, la función tiene como parámetro de
2 //entrada el usuario actual en el prototipo
3 Future<void> _signIn(Future<AuthUser?> auxUser) async {
4   try {
5     //emite el estado de SigningIn al iniciar la autenticación
6     emit(AuthSigningIn());
7     final user = await auxUser;
8     //si el usuario es nulo existe un error al iniciar sesion
9     if (user == null) {
10      emit(AuthError("Error desconocido, intenta en otro momento"));
11    } else {
12      //si todo esta bien se emite el usuario
13      emit(AuthSignedIn(user));
14      //emit(AuthSignedOut());
15    }
16  } catch (e) {
17    //en caso de error se emite otro estado
18    emit(AuthError("Error ${e.toString()}"));
19  }
20 }

```



```

1 BlocBuilder<MyUserCubit, MyUserState>(
2   builder: (_, state) {
3     if (state is MyUserReadyState) {
4       return FadeInLeft(
5         child: InfoCard(
6           nombre: '${state.user.nombre} ${state.user.apellido}',
7           rol: 'ESTUDIANTE',
8           imagen: state.user.imagen,
9         ),
10      );
11    }
12    return const Center();
13  },
14 );

```

Figura 2.29 Código de autenticación (a) y uso de estado capa de presentación (b)

2.2.6.2 Manejo de roles

Para el manejo de roles en el prototipo se emplea el campo “administrador” de la colección *user* especificado en la sección 2.1.5 Durante el proceso de registro cada uno de los formularios presentados completará automáticamente este campo acorde a su rol, en el caso de un Administrador (General o Máster) como “*true*” y el Estudiante como “*false*”. Para la validación del rol se consulta el usuario actual de la aplicación con ayuda de *Firebase*, accediendo de esta forma al campo “administrador”. En la Figura 2.30 (a) se muestra la validación principal que permite dirigir al módulo de Dashboard correspondiente a cada usuario, esta validación se encuentra en el archivo *app.dart* dentro del directorio *lib*. En la Figura 2.30 (b) se muestra la validación con un operador ternario que hace uso del campo “administrador” para mostrar el menú lateral (*side menu*) del Administrador (General o Máster) o Estudiante.



```

1
2 //Verificación en Firebase para conocer si el usuario existe
3 verificarUsuarioExistente(usuarioAux).then((resultado) async {
4   //Verificar si es un administrador
5   final admin = await doc.data()?['administrador'];
6   //En caso de ser administrador
7   if (admin.toString() == "true") {
8     _navigatorkey.currentState?.pushNamedAndRemoveUntil(Rutas.dashboardAdmin, (route) => false);
9   } //En caso de no ser estudiante
10  } else if (admin.toString() == "false") {
11    _navigatorkey.currentState?.pushNamedAndRemoveUntil(Rutas.dashboard, (route) => false);
12  }
13 }

```



```

1
2 //El menú lateral se desplegará dependiendo si es o no Administrador
3 drawer: state.user.administrador == "true"
4   //En caso de ser administrador
5   ? const SideMenuAdmin()
6   //En caso de ser estudiante
7   : const SideMenu(),

```

Figura 2.30 Ejemplo de código para el manejo de roles

2.2.6.3 Implementación de la base de datos SQLite

La base de datos SQLite es empleada por el prototipo para almacenar los registros diarios y de los datos del usuario. Esta base de datos permite disminuir el número de consultas a *Firebase*, lo que se traduce en ahorro de recursos. En la Figura 2.31 (a) se muestra la creación de las tablas y en la Figura 2.31 (b) un ejemplo de consulta en la base de datos de registros.



```
1 static Future<void> crearTablas(sql.Database database) async {
2   database.execute("""CREATE TABLE usuarios(
3     id TEXT,
4     nombre TEXT,
5     apellido TEXT,
6     administrador TEXT,
7     estudio TEXT,
8     sexo TEXT,
9     verificacion TEXT,
10    nacimiento TEXT,
11    areaEstudio TEXT,
12    imagen TEXT,
13    createdAt TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
14  )
15  """);
16  await database.execute("""CREATE TABLE registros(
17    id INTEGER PRIMARY KEY AUTOINCREMENT,
18    usuarioID TEXT,
19    areaEstudio TEXT,
20    calidad TEXT,
21    despierto TEXT,
22    edad TEXT,
23    enCama TEXT,
24    fechaEnvio TEXT,
25    nivelCansado TEXT,
26    nivelEducacion TEXT,
27    sentimiento TEXT,
28    sexo TEXT,
29    totalMinutos TEXT,
30    createdAt TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
31  )""");
32 }
```

```
1 //Promedio de sueño en base a todos los registros diarios
2 Future<double> promedioTotal(String id) async {
3   double promedio = 0;
4   int totalRegistros = 0;
5   double totalMinutos = 0;
6   final registros = await SQLiteHelper.getRegistro(id);
7   for (var registro in registros) {
8     totalMinutos += double.parse(registro['totalMinutos']);
9     totalRegistros++;
10  }
11  promedio = totalMinutos / totalRegistros;
12  return promedio;
13 }
```

Figura 2.31 Codificación de la base de datos SQLite (a) consulta de bases de datos (b)

2.2.6.4 Manejo de rutas

Para manejar la navegación dentro del prototipo se hace uso de las rutas para cada pantalla del prototipo. En la Figura 2.32 se muestra un ejemplo de la implementación de las rutas a los módulos de: dashboard, registro administrador y permisos. El resto de las rutas se encuentra en el directorio navegación de la capa de presentación.



```
1 import 'package:flutter/material.dart';
2 import './screens/screens.dart';
3 class Rutas extends StatelessWidget {
4   //Rutas para navegar en la aplicación
5   //ruta para dashboard
6   static const dashboard = '/dashboard';
7   //ruta para el registro
8   static const registroAdmin = '/registroAdmin';
9   //ruta para los permisos
10  static const permisosFit = '/permisosFit';
11  //Creación de la ruta
12  static Route routes(RouteSettings routeSettings) {
13    switch (routeSettings.name) {
14      case dashboard:
15        return _construirRuta(DashBoard.create);
16      case permisosFit:
17        return _construirRuta(PermisosFit.create);
18      case registroAdmin:
19        return _construirRuta(RegistroAdministrador.create);
20      default:
21        throw Exception('Ruta no existe');
22    }
23  }
24  const Rutas({super.key});
25  //Construcción de la ruta
26  static MaterialPageRoute _construirRuta(Function build) =>
27    MaterialPageRoute(builder: (context) => build(context));
28  @override
29  Widget build(BuildContext context) {
30    return Container();
31  }
}
```

Figura 2.32 Codificación de la navegación del prototipo

Para navegar a una ruta específica, se hace uso de la función `Navigator.pushNamed`, la cual toma como parámetros de entrada el `context` del `widget` padre y una de las rutas creadas; por ejemplo, para navegar al Dashboard el código sería: `Navigator.pushNamed(context, Rutas.dashboard)`.

2.2.6.5 Lectura de datos de sueño de Google Fit

Para la lectura de datos de sueño se hace uso del paquete `Health` en su versión 4.6.0. La implementación se realiza en el archivo `google_fit_provider.dart` ubicado en la capa de datos y funciones.dart en la capa de negocio. El primer archivo tiene la implementación de los permisos de acceso a los datos de `Google Fit` como se muestra en la Figura 2.33 (a). En la capa de negocio se hace la lectura de los datos de sueño especificando una fecha de inicio y fin, los registros que se encuentran en `Google Fit` se encuentran almacenados por días Figura 2.33 (b). El resto de las funciones se encuentra en el código fuente del prototipo en el ANEXO IV.2

```
1 Future<AppState> authorize() async {
2   //Autorización de permisos de actividad física
3   await Permission.activityRecognition.request();
4   //Booleano para verificar el permiso
5   bool? hasPermissions =
6     await HealthFactory.hasPermissions(types, permissions: permissions);
7   hasPermissions = false;
8   bool authorized = false;
9   //Verificación de permisos
10  if (!hasPermissions) {
11    authorized =
12      await health.requestAuthorization(types, permissions: permissions);
13  }
14  //Verificación de permisos como autorizado o no autorizado
15  state = (authorized) ? AppState.AUTHORIZED : AppState.AUTH_NOT_GRANTED;
16  //Retorno del estado
17  return state;
18 }
```

(a)

```
1 Future<List<HealthDataPoint>> consultaFit(final inicio, final fin) async {
2   //Consultar los registros del sueño en una fecha de inicio y fin
3   List consulta = [];
4   List<HealthDataPoint> resultados = [];
5   final now = DateTime.now();
6   consulta = await fetchData(now, fin, inicio);
7   for (var elemento in consulta) {
8     HealthDataPoint resultado = elemento;
9     resultados.add(resultado);
10  }
11  //se retorna una lista con los resultados
12  return resultados;
13 }
```

(b)

Figura 2.33 Codificación de permisos (a) y lectura de datos de Google Fit (b)

2.2.6.6 Codificación interfaz gráfica del prototipo

Implementar la interfaz gráfica del prototipo implica el uso de los widgets de `Flutter` para lograr construir los diferentes elementos visuales que fueron diseñados en la sección 2.1.3.2. Cada una de las pantallas tiene una clase que extiende de `StatefulWidget`, este tipo de widget maneja un estado que puede ir cambiando a lo largo de su vida útil. Además, tiene un método estático que permite crear una instancia del propio widget envuelta en un `BlocProvider` que le permite instanciar el `MyUserCubit` y `MyUserRepository` (Figura 2.27) para acceder a los datos del usuario empleando el método `getMyUser()`, esto permite utilizar los datos en diferentes componentes y partes de la pantalla.

La construcción de los elementos visuales para la pantalla se realiza en el `override` para el método `build()`, este método retorna un `BlocBuilder` para manejar el estado en que se encuentra el usuario, es decir, si tiene la sesión abierta o no en el prototipo.

En caso de que el estado del usuario se encuentre como listo (`MyUserReadyState`) se retornará un widget `Scaffold`, este widget facilita la creación de interfaces gráficas, añadiendo elementos base de la pantalla como lo son: el banner superior (`AppBar`), el menú lateral (`drawer`) y el cuerpo de la pantalla (`body`). Estos elementos se encuentran especificados en la Figura 2.34 (a) y la codificación en la Figura 2.34 (b). Esta es la estructura básica de cada pantalla del prototipo. La construcción de los diferentes elementos gráficos se lo realiza partiendo de widgets básicos como `Containers()` y dándoles estilos para asemejar lo diseñado en la sección 2.1.3.2.

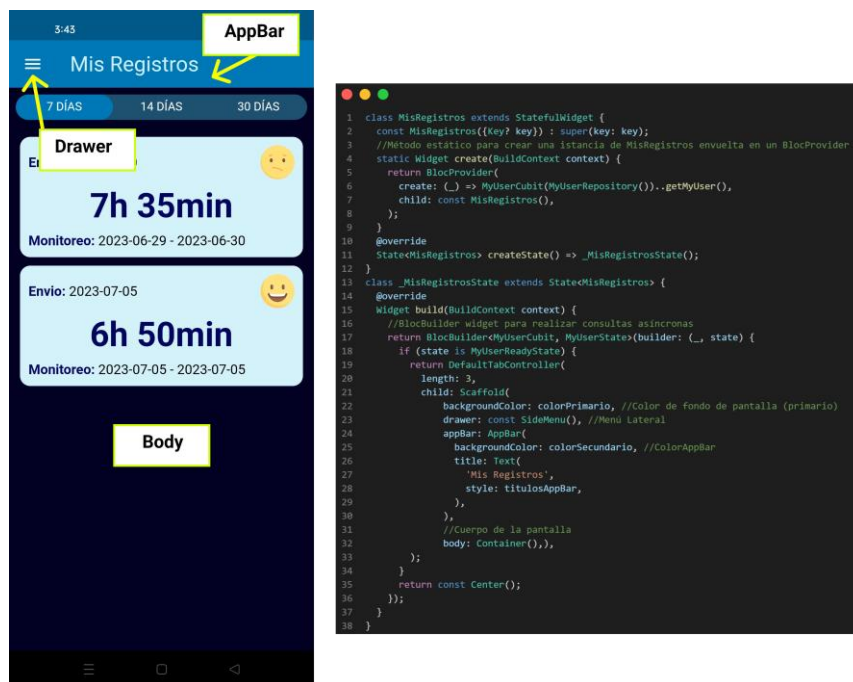


Figura 2.34 Codificación interfaz gráfica módulo de registros

Para declarar los estilos se hizo uso de dos archivos: `colores.dart` y `fontStyle.dart` ubicados en la capa de presentación donde se incorporaron los diferentes colores y tipografía para el prototipo. En la Figura 2.35 se muestra la manera de incorporar los.



(a) (b)
Figura 2.35 Especificación de colores (a) y tipografía (b)

Dentro del `body` en el `Scaffold`, se pueden especificar elementos visuales y su respectiva funcionalidad a través de propiedades; por ejemplo, para implementar las opciones del

menú lateral Figura 2.36 (a) se envolvió cada una de ellas en un *widget FadeInLeft* mostrada en la Figura 2.36 (b), que permite incorporar una animación de entrada. Este *widget* tiene dos propiedades personalizadas: el *delay* para controlar el momento en que se ejecuta la animación y *widget* hijo (*child*). En este caso el hijo es un *InkWell* que tiene las propiedades de *onTap()*, que permite realizar cierta acción al presionar este elemento. En este caso navegar a una de las rutas creadas y la propiedad *child*. En la propiedad *child* se incluye el *widget* personalizado *SideMenuTitle* que se encuentra en la Figura 2.36 (c) donde se especifica el texto y el ícono que tendrá cada opción en el menú.

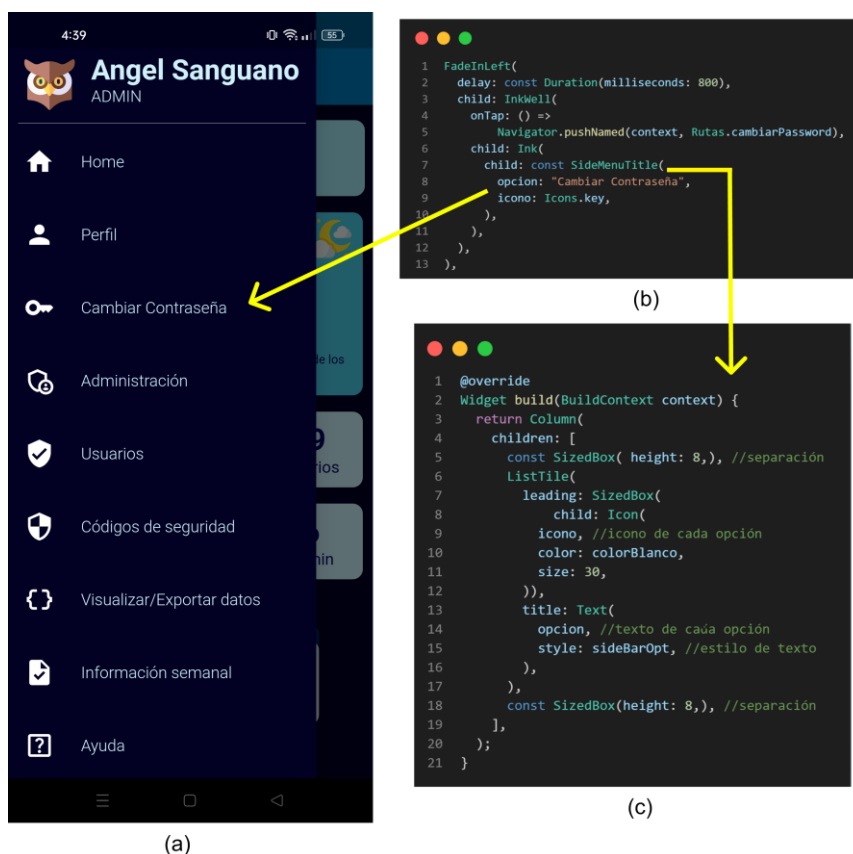


Figura 2.36 Codificación menú lateral

Para la organización de imágenes e íconos que emplea el prototipo se crea el directorio *assets* en la raíz del proyecto. Este nuevo directorio debe ser especificado dentro del archivo *pubspec.yaml* para que se pueda acceder a su contenido desde cualquier lugar del proyecto. Finalmente, el código completo se encuentra en el ANEXO IV.2.

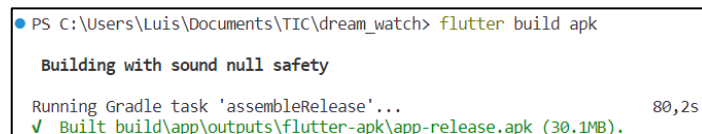
3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

En el presente capítulo se detallan las pruebas funcionales del prototipo. Adicionalmente, los resultados obtenidos en las pruebas de validación del prototipo por un grupo de estudiantes de la Facultad de Ingeniería Eléctrica y Electrónica de la Escuela Politécnica Nacional. Sumado a eso, se detallan los inconvenientes presentados a lo largo del desarrollo del prototipo y las correspondientes soluciones. También, se muestra un análisis preliminar de los datos obtenidos en la fase de pruebas de validación. Finalmente se presentan las conclusiones y recomendaciones.

La instancia actual del Tablero Kanban para esta sección se muestra en el ANEXO IX.1, la columna “por hacer” no tiene actividades pendientes, “en proceso” se encuentran actividades relacionadas a la etapa de pruebas y la redacción del presente capítulo.

3.1 Generación del APK

Una vez finalizada la codificación del prototipo según la sección 2.2.6 y la configuración de los servicios detallada en la sección 2.2.4.1 se procede a obtener el APK del prototipo para su distribución en las pruebas de validación. Se ejecuta el comando *flutter build apk* desde la terminal de Visual Studio Code como se muestra en la Figura 3.1.



```
PS C:\Users\Luis\Documents\TIC\dream_watch> flutter build apk
Building with sound null safety
Running Gradle task 'assembleRelease'... 80,2s
✓ Built build\app\outputs\flutter-apk\app-release.apk (30.1MB).
```

Figura 3.1 Construcción del APK del prototipo

El tamaño final del APK es de 30.1 MB, aproximadamente 20 MB menos de los 50 MB especificados en los requerimientos no funcionales del prototipo en la sección 2.1.1.2.

3.2 Pruebas funcionales

Las pruebas funcionales se realizan empleando un dispositivo físico con Android 12. Las guías de usuario para Estudiante, Administrador General y Administrador Máster se encuentran disponibles en el ANEXO V.1. A continuación, se presentan las pruebas funcionales de los módulos principales del prototipo.

3.2.1 Pruebas funcionales módulos común

En la Figura 3.2 (a) se muestran las pantallas de al prototipo. Al finalizar la introducción se solicita seleccionar el rol al cual se desee ingresar como se muestra en la Figura 3.2 (b).

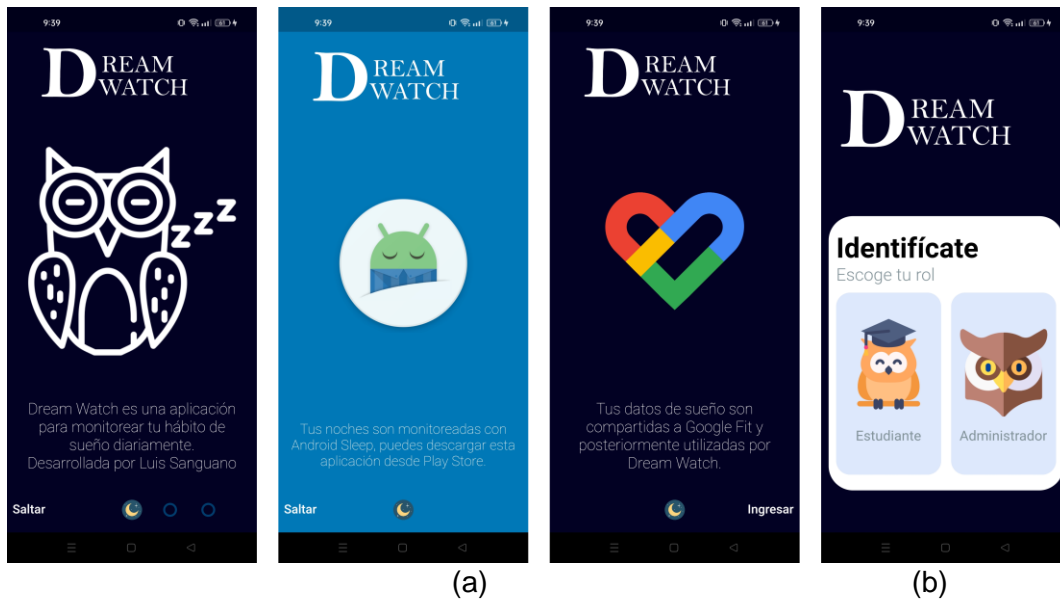


Figura 3.2 Pantallas de inicio (a) selección de rol (b)

3.2.1.1 Módulo de cambio de contraseña

El cambio de contraseñas es gestionado con Firebase, para lo cual se solicita al usuario ingresar su correo electrónico, por medio del cual se envía el enlace proporcionado por Firebase para el cambio como se muestra en la Figura 3.3.

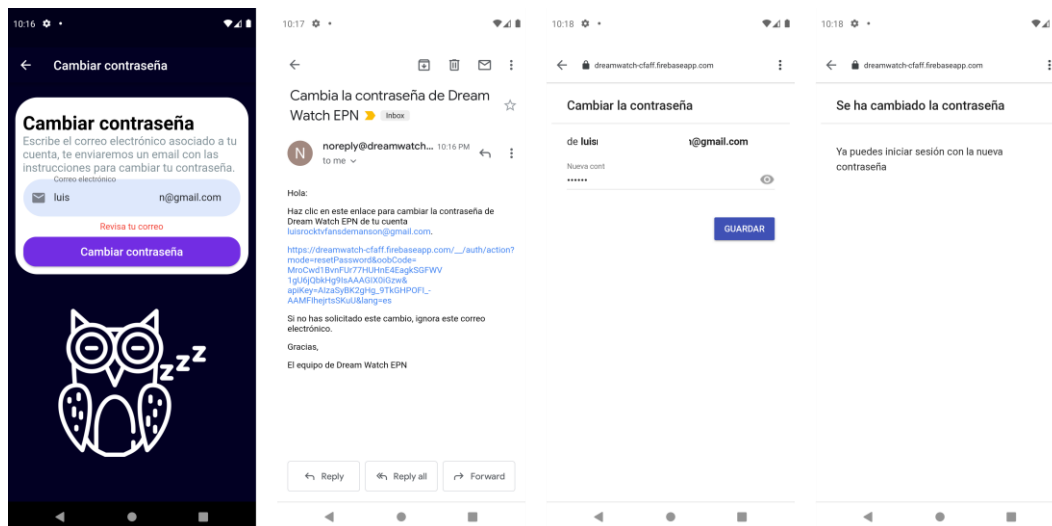
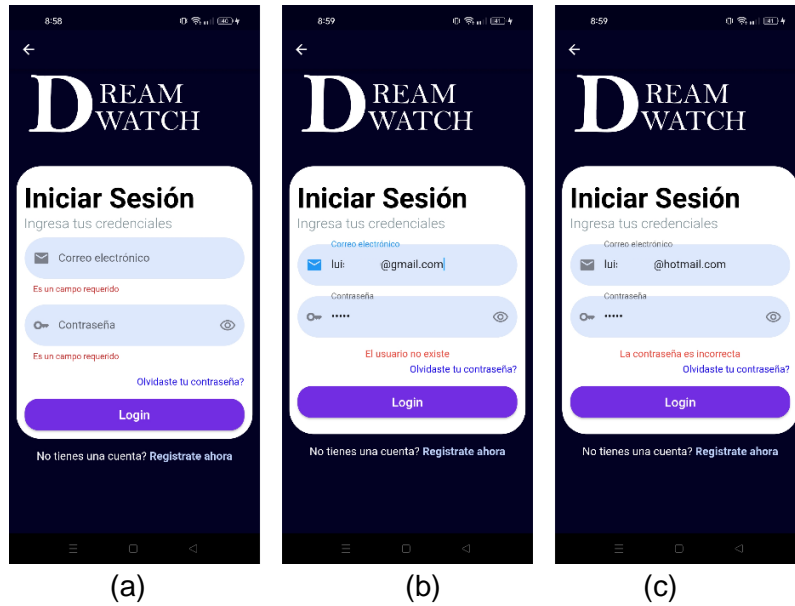


Figura 3.3 Módulo de cambio de contraseña

3.2.1.2 Módulo de inicio de sesión

En la Figura 3.4 se muestran las diferentes validaciones de los campos en el módulo de inicio de sesión. En la Figura 3.4 (a) la validación en caso de que los campos se encuentren vacíos, en la Figura 3.4 (b) en caso de que el correo no se encuentre registrado en Firebase y en la Figura 3.4 (c) cuando la contraseña sea incorrecta.



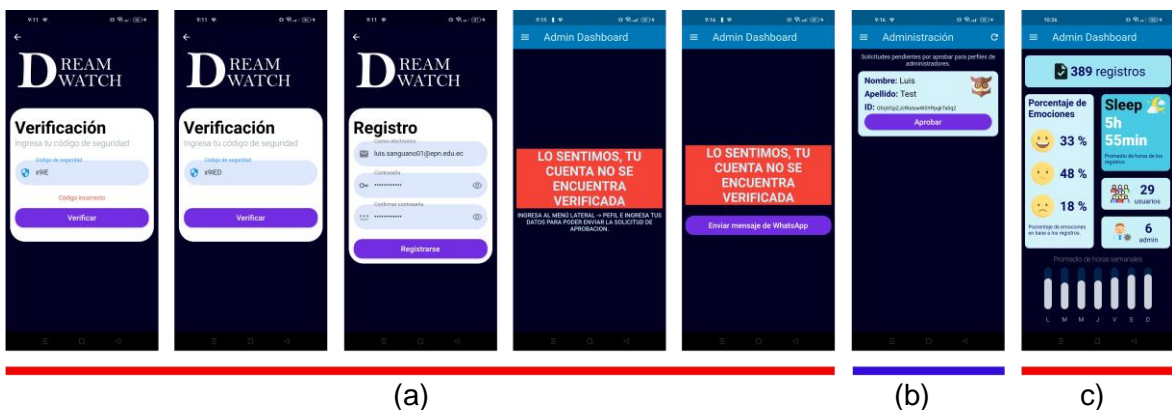
(a) (b) (c)
Figura 3.4 Módulo de inicio de sesión

3.2.2 Pruebas funcionales usuario Administrador General

En la presente sección se muestran los módulos del administrador general.

3.2.2.1 Módulo de registro

Para realizar el registro del administrador general se solicita el código de seguridad para acceder al formulario. Posteriormente se solicita una cuenta de correo electrónico y contraseña (Figura 3.5 (a)), la cuenta debe ser verificada por el Administrador Máster (Figura 3.5 (b)). Una vez verificada la cuenta se pueden visualizar los módulos disponibles para este rol. En rojo se muestran las pantallas del Administrador General y en azul la pantalla del Administrador Máster, empleada para aprobar las solicitudes para nuevas cuentas.



(a) (b) (c)
Figura 3.5 Verificación de código de seguridad y formulario de registro (a), aprobación de solicitud de cuentas (b), visualización de Dashboard administrador general (c)

3.2.2.2 Módulos de visualización de datos

Los datos mostrados en los módulos de visualización de datos son recuperados desde *Firebase*. En la Figura 3.6 se muestran los módulos de Dashboard, datos y Semana.

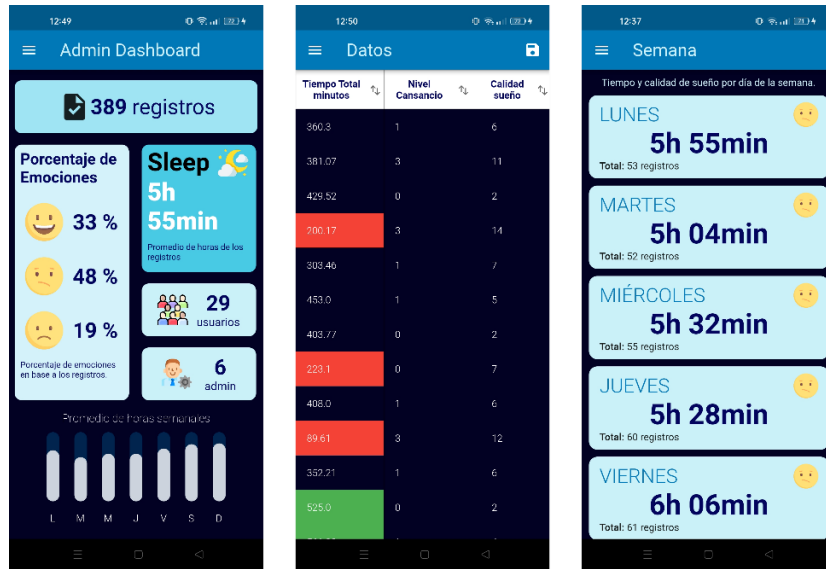


Figura 3.6 Módulos de visualización Administrador General

3.2.3 Pruebas funcionales usuario Estudiante

En esta sección se presentan los módulos disponibles para el usuario Estudiante.

3.2.3.1 Módulo de registro

Para el registro, el usuario ingresa un correo electrónico y contraseña; posteriormente para acceder a todas las funcionalidades debe verificar el correo electrónico ingresado como se muestra en la Figura 3.7.

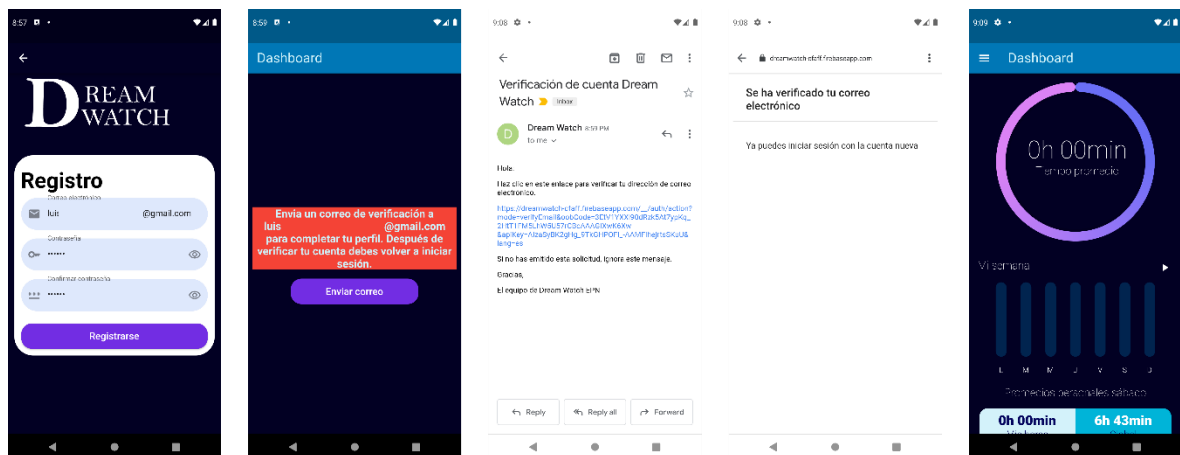


Figura 3.7 Módulo de registro

3.2.3.2 Módulos de permisos y registro diario

El consentimiento de permisos para la actividad física se presentan en la Figura 3.8 (a) se realiza una única vez al instalar la aplicación. Para enviar el registro diario de sueño es necesario completar el cuestionario de sueño (Figura 3.8 (b)).

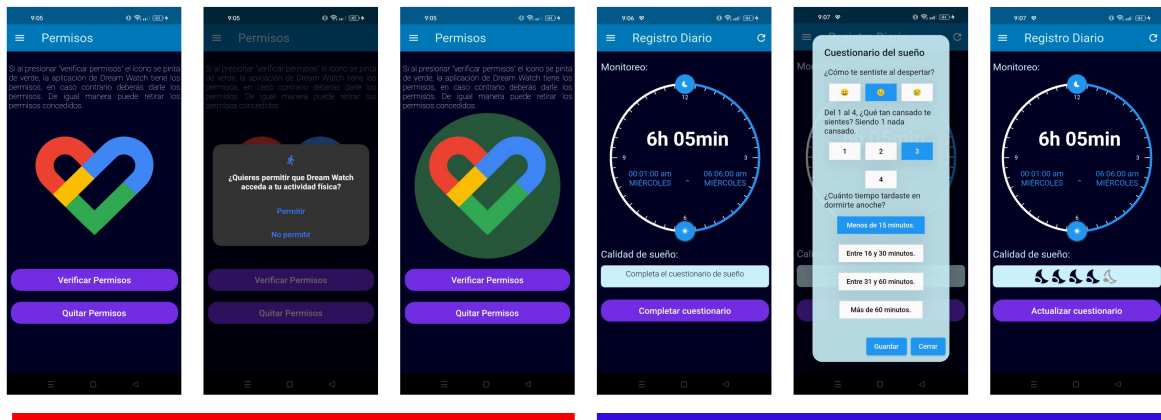


Figura 3.8 Módulo de permisos (a), módulo de registro diario (b)

3.2.3.3 Módulos de visualización de datos

Los datos visualizados en los módulos: Dashboard, Mi Semana y Mis Registros (Figura 3.9) son tomados de la base de datos local como de *Firebase*.

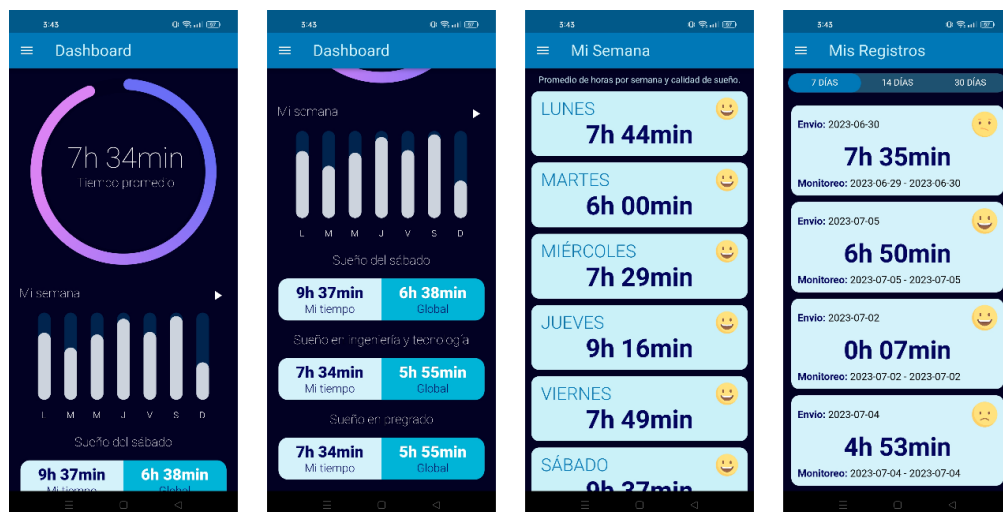


Figura 3.9 Módulos de visualización de datos usuario Estudiante

3.2.4 Pruebas funcionales usuario Administrador Máster

En esta sección se muestran los módulos exclusivos de Administrador Máster. Es importante tomar en cuenta que el módulo de Visualizar/Exportar datos, tanto del Administrador Máster como General se encuentran en la capacidad de acceder y exportar la información, pero el CSV generado en un caso (Administrador Máster) guardará los

registros con los identificadores de los usuarios, y en el caso del General no. En la Figura 3.10 (a) se muestra la visualización de los registros y los usuarios; adicionalmente el proceso de exportación de los datos a un archivo CSV almacenado localmente se presenta en la Figura 3.10 (b). En ambos casos el almacenamiento del archivo sigue el mismo proceso.

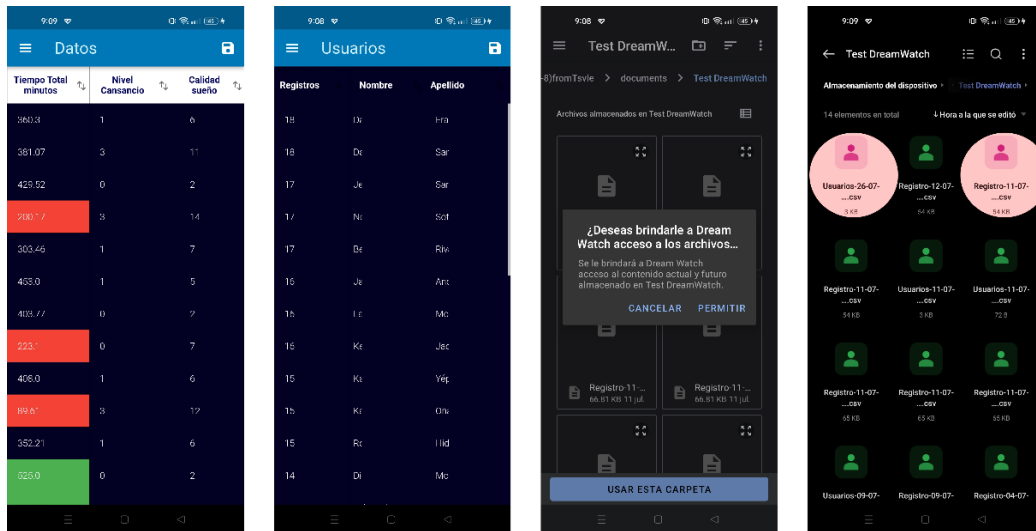


Figura 3.10 Visualización, exportación de registros y datos de usuarios

En el ANEXO VI.1 se muestra el resto de las pruebas funcionales para los módulos implementados en el prototipo.

3.3 Pruebas de validación Estudiantes

Para validar las funcionalidades del rol "Estudiante" se ha puesto a prueba el prototipo con 28 participantes de la Facultad de Ingeniería Eléctrica y Electrónica de la Escuela Politécnica Nacional, en el ANEXO VII.2 se muestran los datos anonimizados de los participantes. La duración de la prueba fue de 18 días, en donde el primer registro de sueño fue enviado el 21 de junio y el último el 9 de julio del 2023.

Para detallar el proceso de pruebas de validación del prototipo se emplea la notación *Business Process Model and Notation* (BPMN), la cual permite construir diagramas para modelos de procesos de negocio sencillos de comprender [68]. En la Figura 3.11 se detalla el proceso, donde interactúan tres actores: Luis Sanguano, PhD. Soraya Sinche y Estudiantes. El proceso inicia con la creación de la guía de usuario para los estudiantes y finaliza a los 18 días de prueba con una encuesta con el objetivo de conocer la experiencia de uso de los participantes.

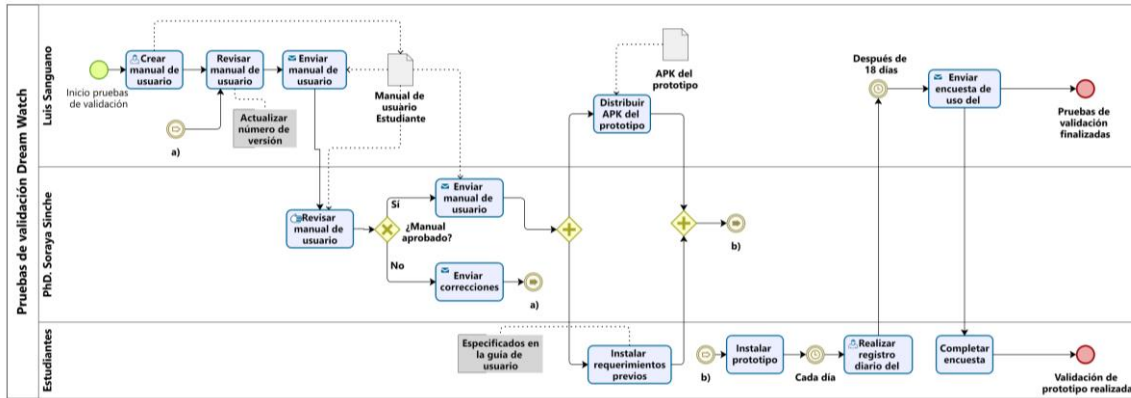


Figura 3.11 Diagrama del proceso de pruebas de validación

Para la distribución del APK del prototipo se empleó un cuestionario de *Google Forms*, en el cual se solicitó la siguiente información: correo electrónico, nombre del participante, versión de Android de su dispositivo móvil y el medio por el cual desearían que se les envíe el APK (*WhatsApp* o correo electrónico). En el ANEXO VIII.1 se encuentra la encuesta aplicada a los participantes y las respectivas respuestas se muestran anonimizadas en la tabla del ANEXO VII.1.

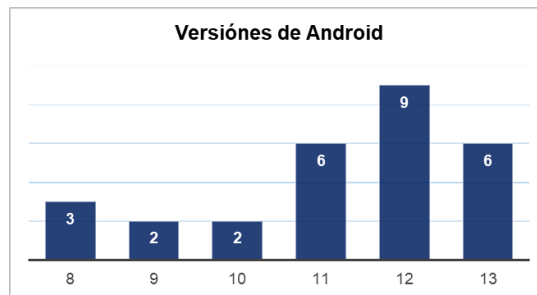


Figura 3.12 Versiones de Android dispositivos móviles de los participantes

La Figura 3.12 muestra que gran parte de los participantes poseen en sus dispositivos móviles una versión de Android superior a la 10. Por otro lado, se observa que un 18% de los participantes (5) aún utiliza una versión inferior a la 10. Con respecto al medio de envío seleccionado por los participantes en la Figura 3.13 (a) se puede evidenciar que *WhatsApp* es la opción más seleccionada entre todos los participantes.

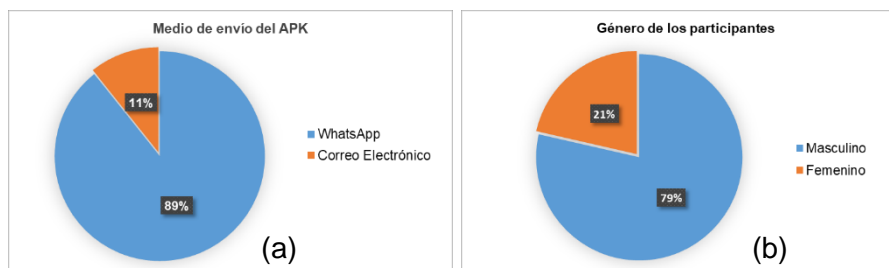


Figura 3.13 Medio de envío elegido (a) y género de los participantes (b)

Para la distribución del APK por *WhatsApp* se utilizó la opción de grupos que ofrece esta red social, el asunto del grupo es “Test prototipo proyecto TIC” creado el 20 de junio del 2023. Para los estudiantes que optaron por el correo electrónico se les envió un enlace de *OneDrive* para la descarga del APK, este enlace se encuentra en el ANEXO IV.1. Con relación al género de los participantes en la Figura 3.13 (b) basada en la tabla del ANEXO VII.2 se evidencia la participación del 21% del género femenino y 79% del masculino. Una vez finalizado el periodo de prueba se hizo uso de una encuesta en *Google Forms* para obtener información acerca de la experiencia de uso de los participantes y validar los requerimientos funcionales y no funcionales del prototipo especificados en la sección. En la Tabla 3.1 se muestran los resultados de la encuesta. El enlace de *Google Forms* de la encuesta se encuentra en el ANEXO VIII.2.

Tabla 3.1 Resultados de la encuesta a los usuarios del rol Estudiante

N°	Pregunta	Resultado %	
		Sí	No
1	¿Pudo instalar el prototipo de Dream Watch en su dispositivo móvil?	100	0
2	¿Pudo ejecutar el prototipo de Dream Watch en su dispositivo móvil?	100	0
3	¿Pudo seleccionar el rol previo a crear su cuenta o iniciar sesión el prototipo de Dream Watch?	100	0
4	¿Pudo crear una cuenta de estudiante con su correo electrónico en el prototipo de Dream Watch?	100	0
5	¿Recibió el correo electrónico de confirmación al momento de crear su cuenta de estudiante en el prototipo de Dream Watch?	100	0
6	¿Pudo iniciar sesión en el prototipo de Dream Watch sin problemas?	100	0
7	¿Pudo visualizar el mensaje de error indicándole que la contraseña es incorrecta?	100	0
8	¿La sesión se mantenía abierta cada vez que volvía a ejecutar el prototipo?	100	0
9	¿Pudo personalizar su cuenta con sus datos personales en el apartado de perfil del prototipo de Dream Watch?	100	0
10	¿Pudo modificar los datos de su cuenta como nombre, área y nivel de estudios, etc?	100	0
11	¿En caso de haber necesitado cambiar su contraseña lo pudo realizar sin problema?	100	0
12	¿Pudo visualizar los promedios y gráficos sobre su tiempo de sueño en el Dashboard del prototipo de Dream Watch?	100	0
13	¿Le resultó interesante el visualizar el promedio de horas de sueño por día de la semana en el Dashboard del prototipo de Dream Watch?	92,86	7,14

N°	Pregunta	Resultado %	
		Sí	No
14	¿Le resultó interesante visualizar la comparación entre sus promedios de sueño y los promedios globales mostrados en el Dashboard del prototipo de Dream Watch?	96,43	3,57
15	¿Pudo visualizar sus registros enviados los últimos 7 y 14 días?	100	0
16	¿Pudo visualizar sin problemas su promedio de sueño y estado emocional por día de la semana en el módulo "mi semana"?	92,86	7,14
17	¿Le resultó útil conocer el promedio de tiempo de sueño por día de la semana y cómo se sentía la mayoría de las veces al despertar?	96,43	3,57
18	¿Pudo sincronizar sus registros diarios de sueño al instalar la última versión del prototipo (Dream Watch v03.apk)?	100	0
19	¿Pudo conceder los permisos de Google Fit al prototipo de Dream Watch?	100	0
20	¿Pudo generar una solicitud en el apartado de "ayuda" del prototipo de Dream Watch?	100	0
21	¿El soporte brindado por el Administrador Máster al enviar una solicitud por WhatsApp le ayudó a solventar sus problemas?	100	0
22	¿Al ingresar al módulo de registro diario, y previo a completar el cuestionario de calidad de sueño, pudo visualizar la información de sueño como fecha y hora en la que se durmió/acostó, total de tiempo que durmió, etc?	100	0
23	¿Pudo enviar sus registros diarios de sueño?	100	0
24	¿Pudo visualizar la retroalimentación presentada en forma de ranking de lunas?	100	0
25	¿Pudo completar el cuestionario de calidad de sueño?	100	0
26	¿Pudo visualizar el avatar de búho asignado a su perfil en el prototipo de Dream Watch?	100	0
27	¿Le parece interesante que la asignación de su avatar de búho sea en base a su calidad de sueño?	100	0
28	¿Considera que la información mostrada en el prototipo de Dream Watch en sus diferentes módulos le pueden ayudar identificar posibles malos hábitos de sueño?	82,14	17,86
29	¿Tuvo algún inconveniente con respecto al tamaño en el espacio de memoria que ocupa el prototipo de Dream Watch en su dispositivo móvil?	3,57	96,43
30	¿La paleta de colores escogida para el prototipo de Dream Watch le pareció adecuada tomando en cuenta la naturaleza del prototipo (monitoreo del sueño)?	100	0
31	¿Considera que los diferentes formularios del prototipo de Dream Watch contienen los campos mínimos y necesarios para el funcionamiento del prototipo?	92,86	7,14
32	¿Pudo acceder a los diferentes módulos de manera intuitiva?	100	0
33	¿Considera que el tamaño y tipo de fuente utilizada en los textos del prototipo de Dream Watch se puede leer sin dificultad?	100	0

3.4 Pruebas de validación Administrador General

Para validar las funcionalidades para la cuenta de Administrador General se puso a prueba el prototipo con 3 participantes ajenos a las pruebas de validación del rol Estudiante. Se emplea una encuesta de *Google Forms* para validar los requerimientos para esta cuenta. En la Tabla 3.2 se muestran los resultados de la encuesta. El enlace de *Google Forms* de la encuesta se encuentra en el ANEXO VIII.3.

Tabla 3.2 Resultados de la encuesta a los usuarios de la cuenta Administrador General

N°	Pregunta	Resultado %	
		Sí	No
1	¿Pudo instalar el prototipo de Dream Watch en su dispositivo móvil?	100	0
2	¿Pudo ejecutar el prototipo de Dream Watch en su dispositivo móvil?	100	0
3	¿Pudo seleccionar el rol previo a crear su cuenta o iniciar sesión el prototipo de Dream Watch?	100	0
4	¿Pudo verificar el código de seguridad proporcionado por el Administrador Máster?	100	0
5	¿Pudo crear una cuenta de Administrador con su correo electrónico en el prototipo de Dream Watch?	100	0
6	¿Pudo enviar la solicitud para la verificación de su cuenta?	100	0
7	¿Su cuenta fue verificada exitosamente?	100	0
8	¿Pudo iniciar sesión en el prototipo de Dream Watch sin problemas?	100	0
9	En caso de ingresar mal su contraseña, ¿pudo visualizar el mensaje de error indicándole que la contraseña es incorrecta?	100	0
10	¿La sesión se mantenía abierta cada vez que volvía a ejecutar el prototipo?	100	0
11	¿Pudo visualizar los promedios de sueño de los usuarios en el Dashboard?	100	0
12	¿Pudo visualizar el número de usuarios y administradores en el Dashboard?	100	0
13	¿Le parece interesante la información que se muestra en el Dashboard?	100	0
14	¿Pudo personalizar su cuenta con sus datos personales en el apartado de perfil del prototipo de Dream Watch?	100	0
15	¿Pudo editar los datos de su cuenta como nombre, apellido y fecha de nacimiento?	100	0
16	¿En caso de haber necesitado cambiar su contraseña lo pudo realizar sin problema?	100	0
17	¿Pudo generar una solicitud en el apartado de "ayuda" del prototipo de Dream Watch?	100	0
18	¿El soporte brindado por el Administrador Máster al enviar una solicitud por WhatsApp le ayudó a solventar sus problemas?	100	0

N°	Pregunta	Resultado %	
		Sí	No
19	¿Pudo visualizar los promedios de sueño de los usuarios por día de la semana en el módulo "Información semanal"?	100	0
20	¿Le parece interesante la información mostrada en el módulo "Información semanal"?	100	0
21	¿Pudo visualizar la tabla con los registros de sueño enviados por los usuarios en el módulo de "visualizar/exportar datos"?	100	0
22	¿Pudo exportar y almacenar el archivo CSV con los registros de sueño de los usuarios en su dispositivo?	100	0
23	¿Tuvo algún inconveniente con respecto al tamaño que ocupa el prototipo de Dream Watch en su dispositivo móvil?	100	0

3.5 Pruebas de validación Administrador Máster

La prueba de validación para el Administrador Máster se realizó únicamente con un participante. Se emplea una encuesta de *Google Forms* para validar los requerimientos para este rol. En la Tabla 3.3 se muestran los resultados de la encuesta. El enlace de *Google Forms* de la encuesta se encuentra en el ANEXO VIII.4.

Tabla 3.3 Resultados de la encuesta rol Administrador Máster

N°	Pregunta	Resultado %	
		Sí	No
1	¿Pudo iniciar sesión en el prototipo de Dream Watch sin problemas?	100	0
2	¿La sesión se mantenía abierta cada vez que volvía a ejecutar el prototipo?	100	0
3	¿Pudo visualizar y copiar los códigos de seguridad?	100	0
4	¿Pudo generar nuevos códigos de seguridad?	100	0
5	¿Pudo aprobar solicitudes para nuevas cuentas de administración?	100	0
6	¿Pudo visualizar la tabla con los datos de los usuarios del prototipo y el número correspondiente de registros enviados?	100	0
7	¿Pudo exportar y almacenar los datos de los usuarios en un archivo CSV en su dispositivo?	100	0
8	¿Pudo visualizar los promedios de sueño de los usuarios en el Dashboard?	100	0
9	¿Le parece interesante la información que se muestra en el Dashboard?	100	0
10	¿Pudo visualizar los promedios de sueño de los usuarios por día de la semana en el módulo "Información semanal"?	100	0
11	¿Le parece interesante conocer el tiempo promedio de sueño y el sentimiento más frecuente por día de la semana de los usuarios?	100	0
12	¿Pudo visualizar la tabla con los registros de sueño enviados por los usuarios en el módulo de "visualizar/exportar datos"?	100	0
13	¿Pudo exportar y almacenar el archivo CSV con los registros de sueño de los usuarios incluyendo sus identificadores en su dispositivo?	100	0

3.6 Resultados de la integración con Firebase

Firestore. En la Figura 3.14 se evidencia un número máximo de 30 usuarios activos por día.



Figura 3.14 Número de usuarios activos por día registrados en *Firebase*

En la Figura 3.15 se evidencia que existieron cerca de 203 mil operaciones de lectura, 1.2 mil operaciones de escritura y 221 operaciones de eliminación.



Figura 3.15 Número total de operaciones de lectura, escritura y eliminación en *Firebase*

3.7 Corrección de errores

Al realizar las pruebas de funcionamiento del prototipo se presentaron algunos inconvenientes que fueron solventados a lo largo del desarrollo del prototipo y su depuración. En Tabla 3.4 se detallan los problemas y sus correspondientes soluciones.

Tabla 3.4 Corrección de errores

Problema	Solución
La cuenta de Administrador Máster no podía acceder a la lista de usuarios del prototipo.	Se modificó la consulta a <i>Firebase</i> , retornando toda la colección que tiene la información de los usuarios, y posteriormente iterando los resultados e instanciando objetos del tipo usuario para representarlos en una tabla.
Errores en los encabezados del archivo CSV exportando en el módulo de Visualizar/Exportar.	El error se solventó colocando correctamente el separador entre cada encabezado, el separador es el signo de puntuación “coma”.
El campo de correo electrónico en el módulo de inicio de sesión y registro se admitían espacios, lo cual provocaba problemas al validar este correo electrónico por parte de <i>Firebase</i> .	Se reemplazan todos los espacios en blanco con una cadena vacía, con esto se eliminan los espacios en blanco que el usuario pueda ingresar por error.
Errores tipográficos.	Los errores tipográficos fueron corregidos en los diferentes archivos de código.
En el módulo Mi Semana del rol Estudiante, el ícono que representa el sentimiento reportado con mayor frecuencia mostraba un sentimiento incorrecto al reportado por el estudiante.	Se verificó el código de este módulo y se corrigió el condicional que controla la asignación del ícono. Adicionalmente, se envió una nueva APK a los participantes, solventando el error.
Al terminar de enviar el registro diario de sueño, la pantalla no se actualizaba automáticamente mostrando el ranking de lunas asignado al registro enviado.	Para solventar este problema se empleó la navegación al mismo <i>Widget</i> , que contiene el módulo de Registro Diario. De esta manera se consigue que al terminar el registro la pantalla se actualice automáticamente.

3.8 Resultados de registros diarios de sueño

El presente análisis preliminar emplea una metodología basada en los pasos fundamentales de la minería de datos [69]. Los datos fueron recolectados durante las pruebas de validación y posteriormente exportados desde el prototipo en un archivo CSV. Para la fase de depuración y análisis se hizo uso de Excel y su función de cálculo para el promedio de los datos importados desde el archivo CSV. En la Figura 3.16 se muestra el proceso adoptado.

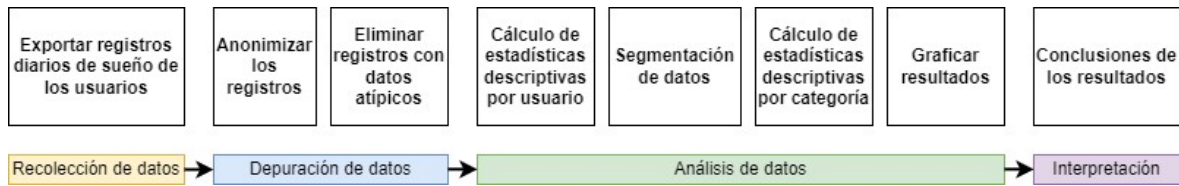


Figura 3.16 Proceso de análisis de datos de sueño

El número de registros al exportar los datos desde el prototipo fue de 388, estos registros fueron anonimizados empleando la siguiente regla: mantener los dos primeros y últimos caracteres del identificador y reemplazar el resto por cuatro asteriscos, por ejemplo: 2A****h1. Para eliminar los registros con datos atípicos se empleó una inspección visual en el diagrama de dispersión del número de registro vs el tiempo total de sueño como se muestra en la Figura 3.17 (a). En la Figura 3.17 (b) se puede evidenciar el conjunto de datos posterior a la eliminación de los datos atípicos.

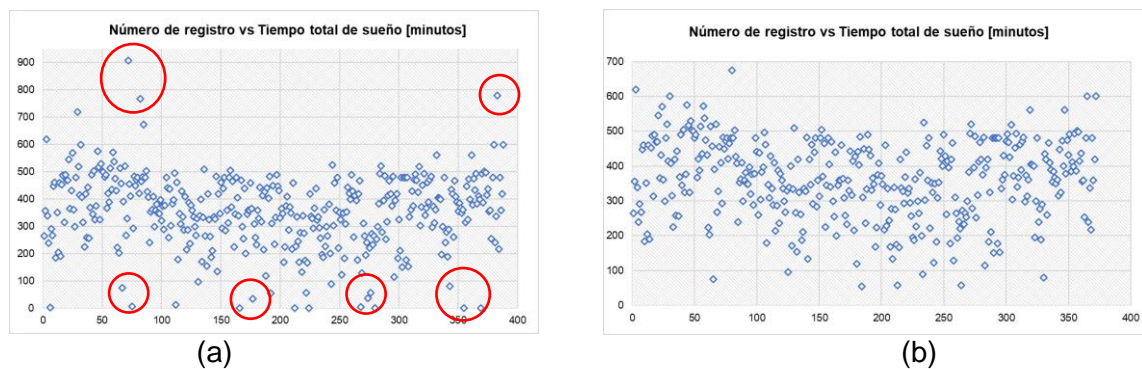


Figura 3.17 Diagrama de dispersión datos originales (a), diagrama de dispersión posterior a la depuración de los datos (b)

Los registros depurados como los registros originales se encuentran en el ANEXO VII.4. Empleando los registros depurados y el identificador anonimizado se procede a realizar el cálculo del promedio de tiempo de sueño y calidad de sueño para cada uno de los participantes, como se muestra en la tabla del ANEXO VII.3. Con los promedios calculados se procede a segmentar la tabla de datos de los participantes bajo las siguientes categorías: edad, semestre, créditos y género. La segmentación de cada categoría con los respectivos registros se encuentra en el ANEXO VII.4. A continuación, se procede a calcular el promedio de tiempo y calidad de sueño para cada categoría.

Finalmente, se grafican los resultados obtenidos en cada categoría para realizar una inspección visual. Las tablas con los promedios por categoría se encuentran en el ANEXO VII.5. Para interpretar los gráficos de la Figura 3.18 y Figura 3.19 derivados de estas tablas se debe tomar en cuenta que acorde al índice de calidad de sueño de Pittsburg [70], donde

el prototipo basa su ponderación para esta variable: un valor numérico alto en este índice representa un mal hábito de sueño, mientras que un valor bajo es lo ideal.

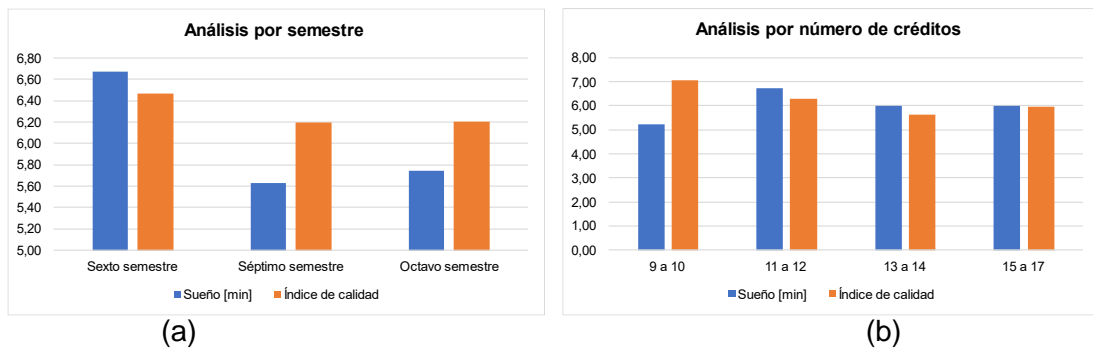


Figura 3.18 Análisis por semestre (a), análisis por número de créditos (b)

En la Figura 3.18 (a) se evidencia que los estudiantes de sexto semestre presentan un mayor número de horas de sueño en comparación a los semestres superiores, pero poseen una mala calidad de sueño. En la Figura 3.18 (b) se evidencia que aquellos estudiantes que toman entre 11 y 12 créditos duermen más tiempo; sin embargo, los que toman de 13 a 14 créditos tienen una mejor calidad de sueño. Adicionalmente se evidencia que los estudiantes que están tomando entre 9 a 10 créditos duermen menos.

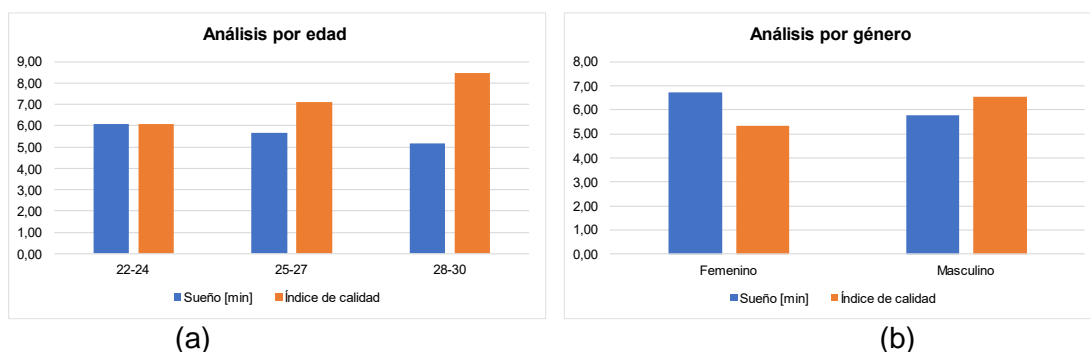






Figura 3.19 Análisis por edad (a), análisis por género (b)

En la Figura 3.19 (a) se evidencia que las horas de sueño disminuyen conforme aumenta la edad. Adicionalmente se evidencia que la calidad de sueño empeora a medida que la edad aumenta. Por otro lado, en la Figura 3.19 (b) se evidencia que el género femenino tiene una mejor calidad de sueño que el masculino. Además, un promedio de horas de sueño superior.

Para finalizar la interpretación de los resultados se debe resaltar aquellos resultados globales obtenidos durante este análisis preliminar. Estos valores se muestran en el ANEXO VII.4: la edad promedio de los participantes es de 23.9 años, las horas de sueño globales son 5.99 horas, la calidad de sueño global fue de 6.28/20 y el avatar más frecuente entre los participantes fue el Búho Sereno como se muestra en la Tabla 3.5.

Tabla 3.5 Porcentaje de asignación de búhos según su calidad de sueño

Avatar	Nombre	Descripción	Porcentaje de usuarios
	Búho estelar	Excelente hábito del sueño.	21,43 %
	Búho sereno	Buen hábito del sueño.	71,43 %
	Búho medianoche	Hábito de sueño insuficiente.	7,14 %
	Búho somnoliento	Mal hábito del sueño	0 %

3.9 Tablero Kanban final

La última instancia del Tablero Kanban para el presente Trabajo de Integración Curricular se muestra en la Figura 3.20, se evidencia que todas las tareas planificadas se encuentran como terminadas y no existen tareas en proceso o por hacer. De esta manera se evidencia el cumplimiento de las actividades planificadas.

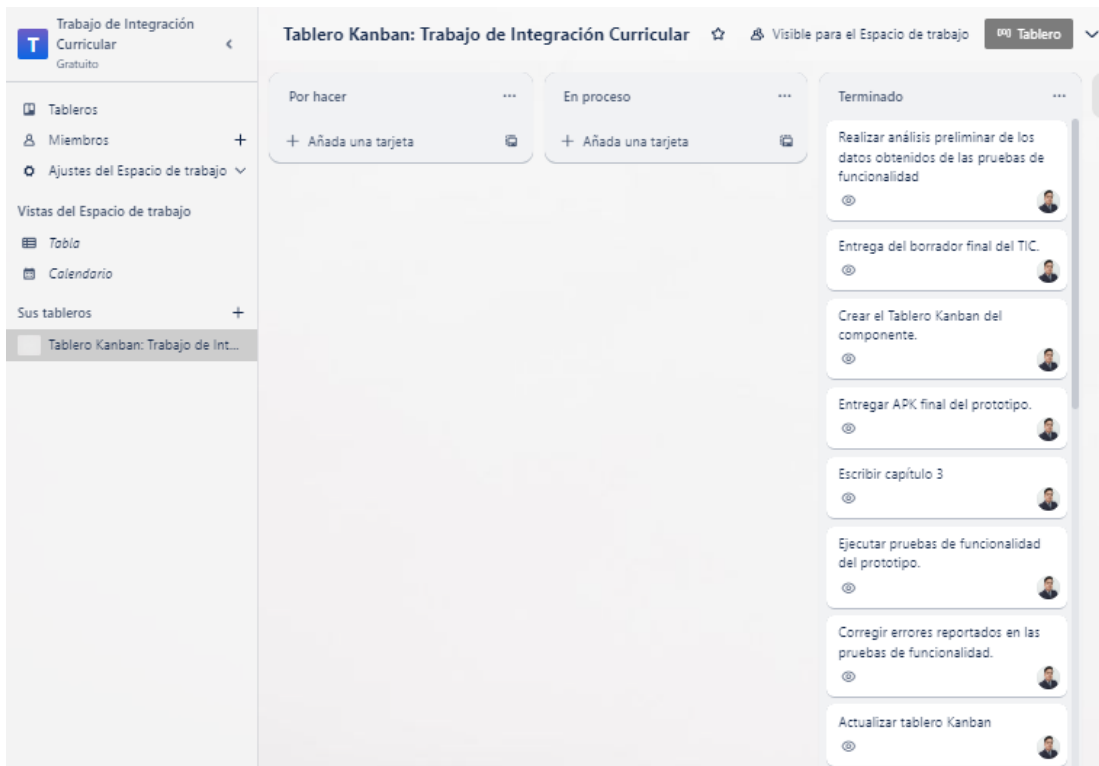


Figura 3.20 Tablero Kanban final desarrollado con Trello [50]

3.10 Conclusiones

En el presente Trabajo de Integración Curricular se ha desarrollado un prototipo de aplicación móvil para el sistema operativo Android orientado al monitoreo de sueño, que puede llegar a convertirse en una alternativa novedosa para el control del hábito de sueño de las personas. Y adicionalmente, facilitar la recolección de datos relacionados a datos de sueño de la población para futuras investigaciones científicas.

Las herramientas disponibles para trabajar con el Framework de Flutter resultaron ser de gran utilidad para el desarrollo del prototipo, pues brindaron un entorno de programación fácil de gestionar y con todas las facilidades para el desarrollo del prototipo, como el emulador virtual o el editor de código fuente con sus diferentes extensiones.

El enfoque basado en la metodología *Kanban* y adoptado durante desarrollo de este componente permitió contar con una buena gestión de tareas y tiempos de entrega acorde a la planificación planteada para la implementación del prototipo *Dream Watch*.

El emplear entrevistas para la toma de requerimientos de software fue fundamental durante el diseño del prototipo, pues la interacción directa con los entrevistados permitió obtener información detallada acerca de los requerimientos funcionales y no funcionales.

Las historias de usuario como los diagramas de actividades y casos de uso facilitaron el desarrollo del diseño del prototipo, logrando identificar y modelar las funcionalidades necesarias para cumplir con los requerimientos identificados en las entrevistas.

Gracias a los paquetes disponibles en el repositorio de *Pub Dev* se consiguió diseñar e implementar los módulos necesarios para lograr integrar la aplicación de *Sleep As Android* y la API de *Google Fit* con el prototipo consiguiendo recopilar, almacenar y mostrar la información de sueño de los usuarios dentro del prototipo.

El prototipo de aplicación móvil se implementó empleando el *Framework* de *Flutter* para teléfonos inteligentes con sistema operativo Android, que cumple con las funcionalidades especificadas en el diseño del prototipo *Dream Watch* desarrollado a lo largo de este componente.

Firebase como BaaS (*Backend as a Service*) permitió agilizar el proceso de implementación del prototipo, brindando los servicios y paquetes necesarios para integrar las diferentes funcionalidades del prototipo como la autenticación, almacenamiento y lectura de los registros de sueño.

Se sometió al prototipo de aplicación móvil a pruebas de funcionamiento y validación, en las cuales se obtuvieron resultados favorables y con un alto porcentaje de aceptación por parte de los participantes. Adicionalmente, durante esta fase de pruebas se corrigieron errores mínimos que no alteraron el funcionamiento del prototipo.

El análisis preliminar de los datos recogidos de los registros diarios de sueño de los participantes evidenció que estos con un promedio de edad de 23,89 años no cumplen con el tiempo recomendado de sueño para su edad (8 a 10 horas de sueño), siendo el promedio de sueño 5.99 horas, lo que puede reflejar un problema de salud.

En las gráficas obtenidas del análisis preliminar se visualizaron relaciones con una leve tendencia como, por ejemplo, a medida que la edad aumenta, el tiempo de sueño disminuye y la calidad empeora. Adicionalmente, se pudo visualizar que al aumentar el nivel educativo o al tomar un mayor número de créditos educativos la calidad de sueño empeora. Finalmente se evidencia que el género femenino posee un tiempo y calidad de sueño superior al masculino.

3.11 Recomendaciones

Se recomienda integrar al prototipo un modelo de lenguaje por Inteligencia Artificial como ChatPT de OpenAI que permita interpretar los datos de sueño que toma el prototipo y brinde una retroalimentación personalizada al usuario en forma de lenguaje natural.

Para mejorar la experiencia de uso dentro del prototipo es de utilidad realizar un análisis de UI (*User Experience*) y UX (*User Interface*) en los diferentes módulos del prototipo.

Se recomienda dar valor a los datos obtenidos de sueño durante la fase de pruebas de validación mediante la analítica de datos, buscando construir un modelo de aprendizaje de máquina que tome los datos de sueño del usuario y permita identificar posibles problemas en los hábitos de sueño de las personas.

Dentro del prototipo, resultaría útil contar con un rol “Especialista” en el cual un participante con conocimientos médicos certificados pueda interactuar con el rol “Estudiante” en caso de presentar problemas en su hábito de sueño. Con este valor añadido el “Especialista” podría acceder a información detallada acerca de cómo duerme su paciente optimizando su diagnóstico.

Con el fin de comprender la relación de la calidad de sueño y el rendimiento académico, se propone una nueva fase de recolección de registros de sueño con el prototipo de Dream Watch aplicada una muestra más representativa de estudiantes universitarios en la Escuela Politécnica Nacional.

4 REFERENCIAS BIBLIOGRÁFICAS

- [1] Adjust GmbH, “Mobile app trends 2022 A global benchmark of app performance,” pp. 11–13, Nov. 17, 2022.
- [2] I. Juan Francisco Fernández Sosa, P. Javier Thomas Asociado, and M. Leonardo Corbalán Adjunto, “Utilización de dispositivos móviles como herramienta de sensado en aplicaciones de IoT,” Buenos Aires, Nov. 2021.
- [3] A. E. Barbouzas, F. Malli, Z. Daniil, and K. Gourgoulianis, “Long-Term Impact of COVID-19 Pandemic in Sleep Quality and Lifestyle in Young Adults,” *Int J Environ Res Public Health*, vol. 19, no. 19, p. 12333, Sep. 2022, doi: 10.3390/ijerph191912333.
- [4] B. Germany, “WHO technical meeting on sleep and health,” Nov. 2004.
- [5] Google, “Cómo comenzar a usar Google Fit.” <https://support.google.com/fit/answer/6075067?hl=es-419&co=GENIE.Platform%3DAndroid> (accessed Apr. 09, 2023).
- [6] Google Developers, “Android APIs .” <https://developers.google.com/fit/android?hl=es-419> (accessed Apr. 09, 2023).
- [7] R. Stickgold and M. Walker, *The Neuroscience of Sleep*, Primera Edición. Massachusetts, California: Acedemic Press 2009, 2009.
- [8] W. Gu, L. Shangguan, Z. Yang, and Y. Liu, “Sleep hunter: Towards fine grained sleep stage tracking with smartphones,” *IEEE Trans Mob Comput*, vol. 15, no. 6, pp. 1514–1527, Jun. 2016, doi: 10.1109/TMC.2015.2462812.
- [9] V. Drago *et al.*, “Cyclic alternating pattern in sleep and its relationship to creativity,” *Sleep Med*, vol. 12, no. 4, pp. 361–366, Apr. 2011, doi: 10.1016/j.sleep.2010.11.009.
- [10] S. Fallmann and L. Chen, “Computational sleep behavior analysis: A survey,” *IEEE Access*, vol. 7. Institute of Electrical and Electronics Engineers Inc., pp. 142421–142440, 2019. doi: 10.1109/ACCESS.2019.2944801.
- [11] E. Suni and S. Abhinav, “How Much Sleep Do We Really Need?,” *SLEEP FOUNDATION*, Aug. 29, 2022. <https://www.sleepfoundation.org/how-sleep-works/how-much-sleep-do-we-really-need> (accessed Feb. 15, 2023).
- [12] M. Villano, “Los adolescentes se enfrentan a una epidemia de falta de sueño. Te explicamos por qué,” *CNN*, Jun. 07, 2022. <https://cnnespanol.cnn.com/2022/06/07/adolescentes-epidemia-falta-sueno-trax/> (accessed Feb. 14, 2023).
- [13] S. Chokroverty, “Overview of sleep & sleep disorders,” 2010.
- [14] N. Thomas, “Dormir poco está asociado con conductas de riesgo en adolescentes, según estudio,” *CNN*, Oct. 01, 2018.
- [15] X. Jiao, X. Wang, X. Wang, and Z. Liu, “Noncontact Sleep Monitoring System under a Mattress,” *IEEE Access*, vol. 9, pp. 111203–111213, 2021, doi: 10.1109/ACCESS.2021.3102281.

- [16] G. Cay *et al.*, “Recent Advancement in Sleep Technologies: A Literature Review on Clinical Standards, Sensors, Apps, and AI Methods,” *IEEE Access*, vol. 10. Institute of Electrical and Electronics Engineers Inc., pp. 104737–104756, 2022. doi: 10.1109/ACCESS.2022.3210518.
- [17] Burçin Camcı, Ali Yavuz Kahveci, Bert Arnrich, and Cem Ersoy, “Sleep Apnea Detection via Smart Phones,” 2017.
- [18] StatCounter, “Mobile Operating System Market Share Worldwide.” <https://gs.statcounter.com/os-market-share/mobile/worldwide> (accessed Jun. 01, 2023).
- [19] Urbandroid, “Sleep as Android user manual.” <https://docs.sleep.urbandroid.org/> (accessed Mar. 19, 2023).
- [20] Ipnos Software, “BetterSleep.” <https://www.bettersleep.com/es/> (accessed Mar. 19, 2023).
- [21] Sleep Cycle AB, “How Sleep Cycle works.” <https://www.sleepcycle.com/how-sleep-cycle-works/> (accessed Mar. 19, 2023).
- [22] Reviva Softworks Ltd, “How SnoreLab Works.” <https://www.snorelab.com/how-snorelab-works> (accessed Mar. 19, 2023).
- [23] SM Health Team, “Sleep Monitor FAQ.” <http://sleepmonitor.emobistudio.com/pc/faq.html> (accessed Mar. 19, 2023).
- [24] Urbandroid, “Sleep tracking theory.” https://docs.sleep.urbandroid.org/sleep/sleep_tracking_theory.html (accessed Mar. 19, 2023).
- [25] Urbandroid, “Sleep as Android: Acerca de esta app.” https://play.google.com/store/apps/details?id=com.urbandroid.sleep&hl=es_419&gl=US (accessed Mar. 19, 2023).
- [26] Urbandroid, “Sleep noise analysis”, Accessed: Mar. 21, 2023. [Online]. Available: https://docs.sleep.urbandroid.org/sleep/sleep_noise_recording.html
- [27] Urbandroid, “Automatic sleep tracking.” https://docs.sleep.urbandroid.org/sleep/automatic_sleep_tracking.html (accessed Mar. 23, 2023).
- [28] Oracle, “¿Cómo puedo empezar a desarrollar programas Java con Java Development Kit (JDK)?” https://www.java.com/es/download/help/develop_es.html (accessed Mar. 24, 2023).
- [29] Oracle, “SDK para Java.”
- [30] Platzi Team, “Cómo hacer una app | Guía completa para diseñar y programar una aplicación móvil.” <https://platzi.com/blog/como-crear-una-app/> (accessed Mar. 24, 2023).
- [31] Documentación AWS, “¿Qué es Flutter?” <https://aws.amazon.com/es/what-is/flutter/> (accessed Mar. 25, 2023).

- [32] inLab FIN UPC, “¿Qué es el lenguaje de programación Dart?” <https://inlab.fib.upc.edu/es/blog/que-es-el-lenguaje-de-programacion-dart> (accessed Mar. 25, 2023).
- [33] Flutter Dev, “FlutterES Comunidad.” <https://esflutter.dev/> (accessed Mar. 25, 2023).
- [34] Google, “pub.dev.” <https://pub.dev/> (accessed Mar. 25, 2023).
- [35] Flutter Dev, “Introducción a los Widgets.” <https://esflutter.dev/docs/development/ui/widgets-intro> (accessed Mar. 25, 2023).
- [36] Geeks For Geeks, “Difference Between Hot Reload and Hot Restart in Flutter,” Feb. 14, 2023. <https://www.geeksforgeeks.org/difference-between-hot-reload-and-hot-restart-in-flutter/> (accessed Mar. 26, 2023).
- [37] Microsoft, “Visual Studio Code.” <https://visualstudio.microsoft.com/es/> (accessed Mar. 26, 2023).
- [38] Microsoft, “Snippets in Visual Studio Code.” <https://code.visualstudio.com/docs/editor/userdefinedsnippets> (accessed Mar. 25, 2023).
- [39] Microsoft, “Terminal Basics.” <https://code.visualstudio.com/docs/terminal/basics> (accessed Mar. 25, 2023).
- [40] BrowserStack, “Emulators and Simulators.” <https://www.browserstack.com/emulators-simulators> (accessed Mar. 27, 2023).
- [41] Google Developers, “Cómo ejecutar apps en Android Emulator.”
- [42] Educative, “What is Firebase?” <https://www.educative.io/answers/what-is-firebase> (accessed Mar. 27, 2023).
- [43] Google Developers, “Descripción general de la plataforma .” <https://developers.google.com/fit/overview?hl=es-419> (accessed Apr. 09, 2023).
- [44] I. H. Sarker, “Machine Learning: Algorithms, Real-World Applications and Research Directions,” *SN Computer Science*, vol. 2, no. 3. Springer, May 01, 2021. doi: 10.1007/s42979-021-00592-x.
- [45] M. Shehab *et al.*, “Machine learning in medical applications: A review of state-of-the-art methods,” *Comput Biol Med*, vol. 145, p. 105458, 2022, doi: <https://doi.org/10.1016/j.combiomed.2022.105458>.
- [46] A. Sathyanarayana, J. Srivastava, and L. Fernandez-Luque, “The Science of Sweet Dreams: Predicting Sleep Efficiency from Wearable Device Data,” *Computer (Long Beach Calif)*, vol. 50, no. 3, pp. 30–38, Mar. 2017, doi: 10.1109/MC.2017.91.
- [47] D. C. Yacchirema, D. Sarabia-Jacome, C. E. Palau, and M. Esteve, “A smart system for sleep monitoring by integrating IoT with big data analytics,” *IEEE Access*, vol. 6, pp. 35988–36001, Jun. 2018, doi: 10.1109/ACCESS.2018.2849822.
- [48] IBM, “Deep learning.” <https://www.ibm.com/mx-es/cloud/deep-learning> (accessed Mar. 27, 2023).
- [49] M. Bermejo, “El Kanban,” Barcelona.

- [50] Atlassian Trello, "Qué es Trello." <https://trello.com/es/tour> (accessed Jun. 09, 2023).
- [51] B. Joel, "Overview of tags," 2023. <https://stackoverflowteams.help/en/articles/5611195-overview-of-tags> (accessed Jun. 08, 2023).
- [52] J. Pradel, M. Jose, and R. Martos, "Introducción a la ingeniería del software."
- [53] F. García and A. García, "Tema 4: Ingeniería de Requisitos," Feb. 20, 2018. https://repositorio.grial.eu/bitstream/grial/1143/1/IS_l%20Tema%204%20-%20Ingenieria%20de%20Requisitos.pdf (accessed Jun. 09, 2023).
- [54] Fabrizio Bianchi, "Coolors The super fast color palettes generator!" <https://coolors.co/> (accessed Jun. 15, 2023).
- [55] Google Fonts, "Roboto Font." <https://fonts.google.com/specimen/Roboto> (accessed Jun. 15, 2023).
- [56] Flaticon, "Owl free icon." https://www.flaticon.com/free-icon/owl_4345004?term=owl&page=1&position=32&origin=search&related_id=4345004 (accessed Jun. 15, 2023).
- [57] Figma, "Figma." <https://www.figma.com/> (accessed Jun. 16, 2023).
- [58] Diagrams.net, "Draw.io." <https://app.diagrams.net/> (accessed Jun. 17, 2023).
- [59] Microsoft, "Visual Studio Code on Windows," Aug. 06, 2023. <https://code.visualstudio.com/docs/setup/windows> (accessed Jun. 22, 2023).
- [60] Flutter, "Windows install." <https://docs.flutter.dev/get-started/install/windows> (accessed Jun. 22, 2023).
- [61] Google Developers, "Download Android Studio Flamingo." <https://developer.android.com/studio> (accessed Jun. 22, 2023).
- [62] Firebase, "Instala Firebase CLI." <https://firebase.google.com/docs/cli?hl=es&authuser=0#install-cli-windows> (accessed Jun. 22, 2023).
- [63] Google, "Google Fit Comienza en Android," Jun. 21, 2023. <https://developers.google.com/fit/android/get-started?hl=es-419> (accessed Jun. 22, 2023).
- [64] Google Play services, "Autenticación de tu cliente ," Jun. 21, 2023. <https://developers.google.com/android/guides/client-auth?hl=es-419> (accessed Jun. 22, 2023).
- [65] Google, "Ya existe un cliente de OAuth2 para este nombre de paquete y SHA-1 aparece en otro proyecto", Accessed: Jun. 22, 2023. [Online]. Available: <https://support.google.com/firebase/answer/6401008?hl=es-419>
- [66] Google, "Consola de Google Cloud." <https://cloud.google.com/cloud-console?hl=es> (accessed Jun. 22, 2023).
- [67] Platzi, "Arquitectura BLoC en Flutter." <https://platzi.com/clases/1603-flutter-avanzado/20220-arquitectura-bloc-en-flutter/#:~:text=%C2%BFQu%C3%A9%20es%20el%20patr%C3%B3n%20BLOC,lu>

gar%20central%20en%20su%20proyecto.&text=Es%20un%20patr%C3%B3n%20que%20nos,negocio%20de%20la%20interfaz%20gr%C3%A1fica. (accessed Jun. 23, 2023).

- [68] Lucidchart, “Símbolos y notación de diagramas BPMN.” <https://www.lucidchart.com/pages/es/simbolos-bpmn#:~:text=Un%20diagrama%20de%20modelo%20y,en%20distintas%20industrias%20y%20organizaciones>. (accessed Jul. 08, 2023).
- [69] Universidad Complutense de Madrid, “¿Qué es el Data Mining?” <https://www.masterdatascienceucm.com/que-es-el-data-mining/> (accessed Jul. 21, 2023).
- [70] M. Shkayeva, “What Is The Pittsburgh Sleep Quality Index (PSQI)?” <https://examine.com/conditions/insomnia/faq/4QvroQ7-what-is-the-pittsburgh-sleep-quality-index-psqi/> (accessed Jul. 21, 2023).

5 ANEXOS

Anexo I. Categorización del monitoreo del sueño

Anexo II. Arquitectura de la API de Google Fit

Anexo III. Diseño del prototipo

Anexo IV. Código fuente y APK final del prototipo

Anexo V. Guías de usuario del prototipo

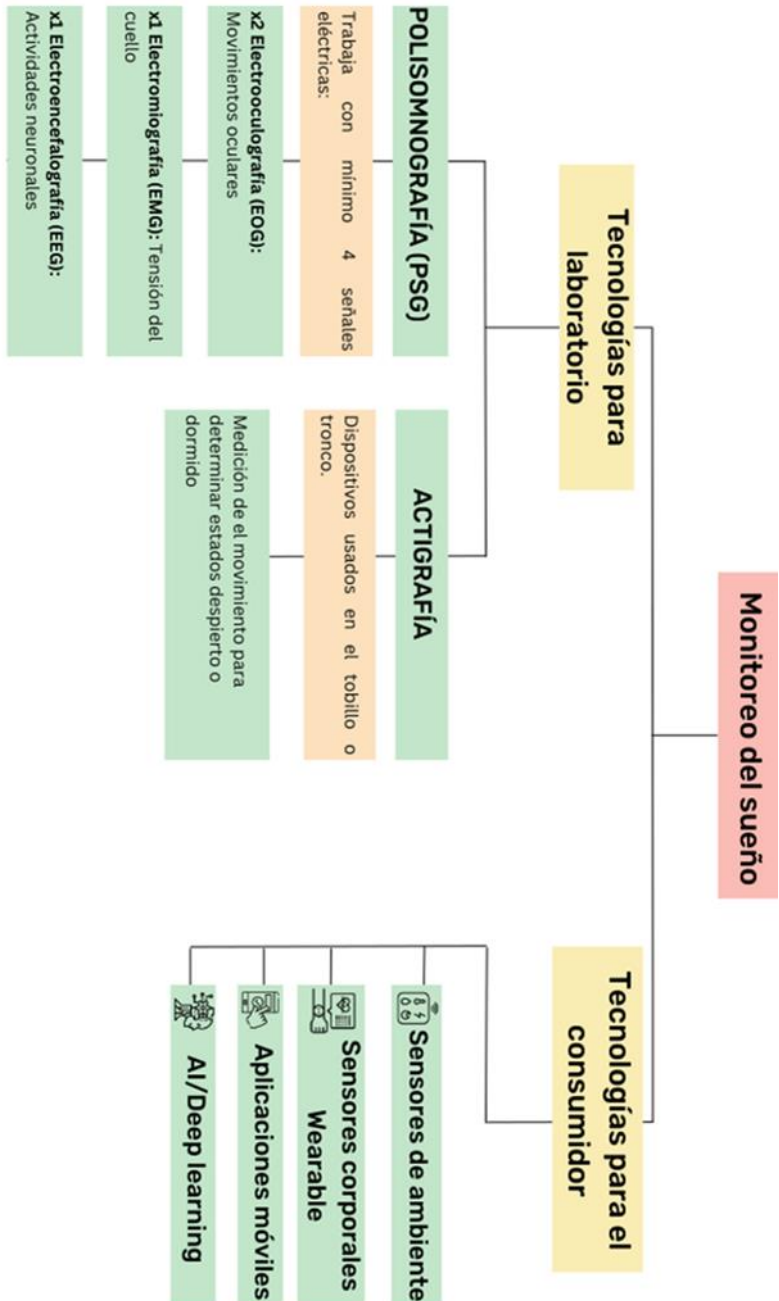
Anexo VI. Pruebas funcionales

Anexo VII. Datos recolectados y resultados obtenidos durante las pruebas de validación

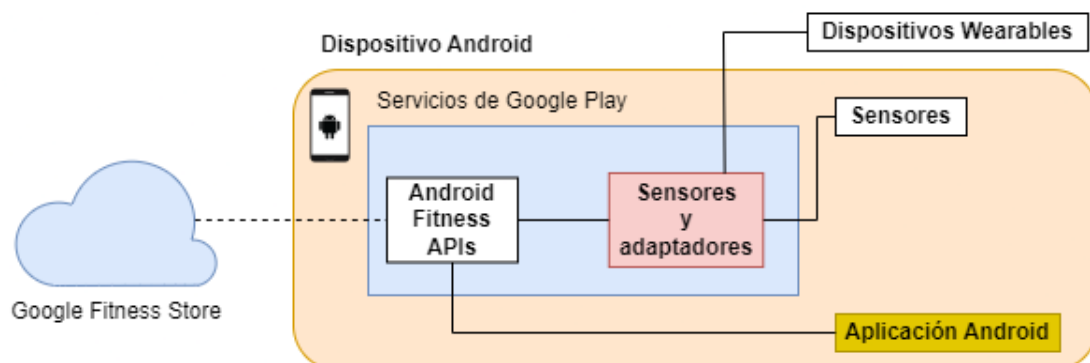
Anexo VIII. Encuestas aplicadas durante las pruebas de validación

Anexo IX. Tablero Kanban (capítulo resultados, conclusiones y recomendaciones)

ANEXO I.1 Diagrama categorización del monitoreo del sueño



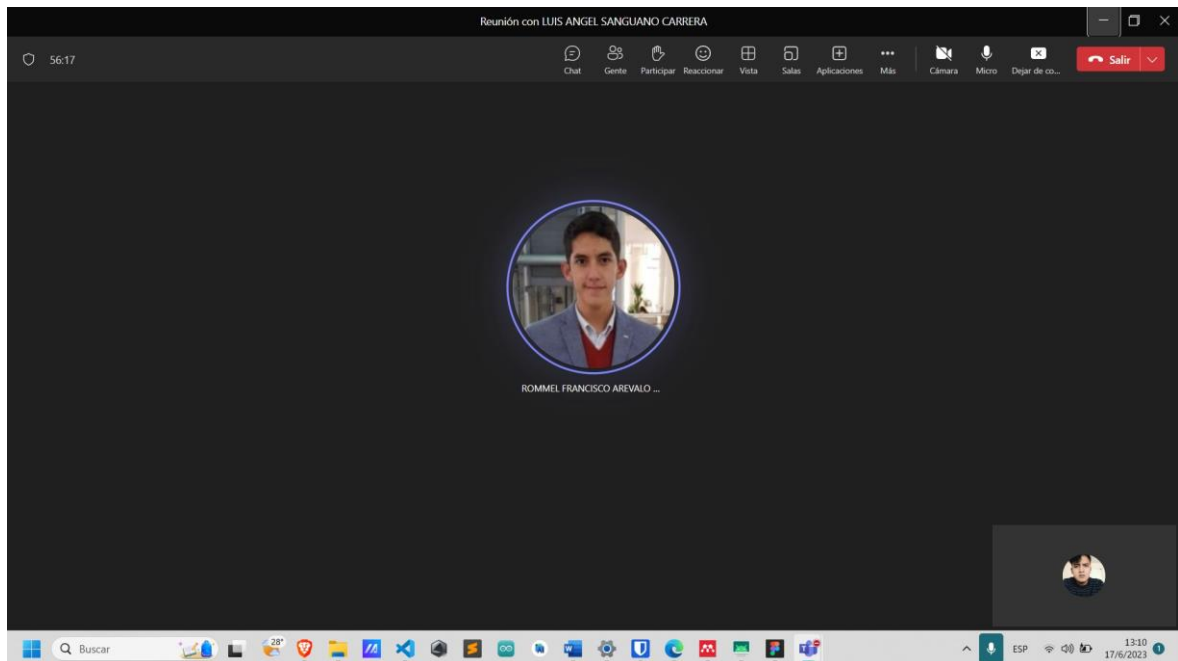
ANEXO II.1 Arquitectura de la API de Google Fit



ANEXO III.1 Entrevistas



(a) Entrevista Dra. Nathaly Chávez



(b) Entrevista Ing. Rommel Francisco Arévalo

ANEXO III.2 Historias de usuario requerimientos funcionales

RF 04	Personalizar perfil
Usuario:	Administrador y Estudiante
Prioridad:	Alta
Descripción:	
Se requiere poder personalizar el perfil de cada cuenta de usuario.	
Criterios de aceptación:	
<ul style="list-style-type: none"> El prototipo debe permitir al usuario realizar el CRUD de sus datos personales en su perfil. 	

RF 05	Realizar registros diarios de sueño
Usuario:	Estudiante
Prioridad:	Alta
Descripción:	
Se requiere almacenar los registros diarios de sueño de los usuarios.	
Criterios de aceptación:	
<ul style="list-style-type: none"> El prototipo se integrará con Google Fit para obtener el tiempo de sueño. El prototipo deberá mostrar información acerca de la fecha y hora en que el usuario se durmió y despertó previo al envío del registro diario. El prototipo deberá informar al usuario acerca del total de horas con minutos que durmió previo al envío del registro diario. El prototipo contará con un formulario corto basado en el cuestionario de Pittsburg para asignar un valor numérico a la calidad de sueño. El prototipo deberá almacenar los registros en la memoria interna del dispositivo como también en una base de datos general para todo el proyecto. El prototipo deberá mostrar un ranking como retroalimentación para el usuario en base al puntaje obtenido en su calidad de sueño una vez terminado el cuestionario. 	

RF 06	Visualizar promedios y datos
Usuario:	Administrador y Estudiante
Prioridad:	Alta
Descripción:	
Se requiere poder mostrar estadísticas y datos en distintos módulos para cada rol.	
Criterios de aceptación:	
<ul style="list-style-type: none"> El prototipo debe contar con gráficas que representen distintas estadísticas para cada rol. Para el Administrador Máster el prototipo debe contar con tablas que permitan representar los datos de los registros y usuarios de la aplicación. El prototipo debe contar con widgets personalizados que muestren promedios relevantes a los usuarios. 	

RF 07	Solicitar ayuda
Usuario:	Administrador y Estudiante
Prioridad:	Alta
Descripción:	
Se requiere poder solicitar ayuda al Administrador Máster.	
Criterios de aceptación:	
<ul style="list-style-type: none"> • El prototipo debe contar con un módulo que permita la comunicación directa con el Administrador Máster. • El módulo debe facilitar al usuario detallar el problema que experimente en el uso del prototipo. 	

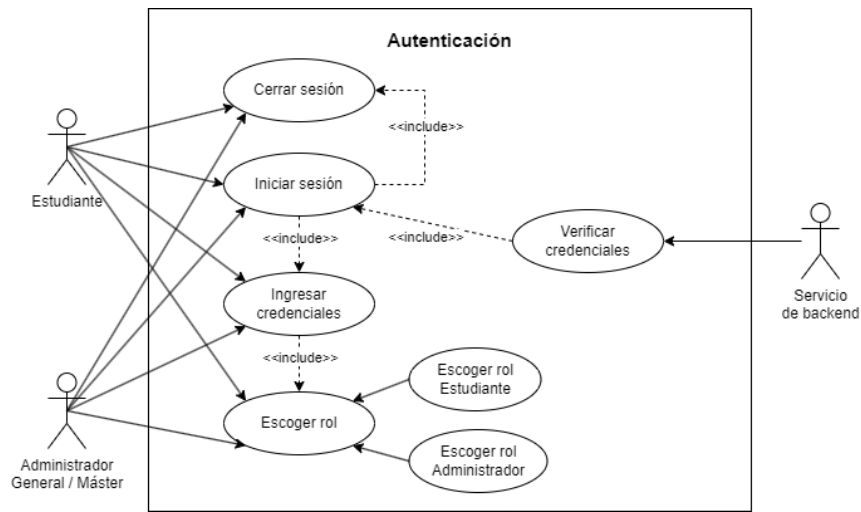
RF 08	Exportar datos
Usuario:	Administrador
Prioridad:	Alta
Descripción:	
Se requiere exportar datos en formato csv.	
Criterios de aceptación:	
<ul style="list-style-type: none"> • El prototipo permitirá exportar los datos de los registros diarios del monitoreo de los usuarios. • El prototipo permitirá exportar los datos de los usuarios con el correspondiente número de registros diarios enviados. • El prototipo debe permitir almacenar internamente el archivo .csv generado al exportar los datos. 	

RF 09	Conceder permisos para Google Fit
Usuario:	Estudiante
Prioridad:	Alta
Descripción:	
Se requiere contar con un módulo en el cual el usuario pueda conceder y retirar los permisos del prototipo para acceder a la información de <i>Google Fit</i> .	
Criterios de aceptación:	
<ul style="list-style-type: none"> • El prototipo contará con un módulo en el cual se le faciliten dos botones para verificar el permiso a Google Fit o retirarlo. 	

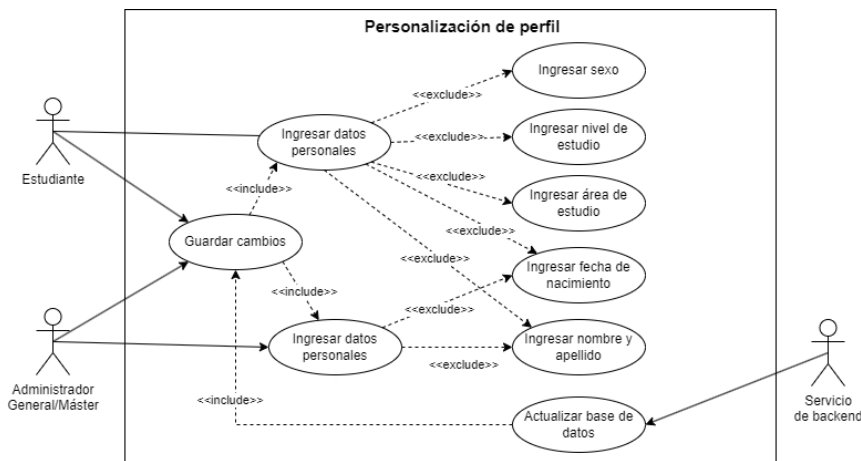
RF 10	Mostrar representación gráfica de la calidad de sueño
Usuario:	Estudiante
Prioridad:	Alta
Descripción:	
Se requiere mostrar al usuario una representación gráfica mediante un avatar que se encuentre relacionado al promedio de su calidad de sueño.	
Criterios de aceptación:	

- Se contarán con un avatar que será asignado en base al promedio de calidad de sueño obtenido de los registros diarios enviados por el usuario.
- La asignación del avatar será realizada de manera automática y se actualizará cada vez que un usuario envíe un registro diario, volviendo a calcular el promedio.
- El avatar asignado al perfil del usuario será mostrado en diferentes lugares del prototipo para que el usuario pueda visualizarlo fácilmente.

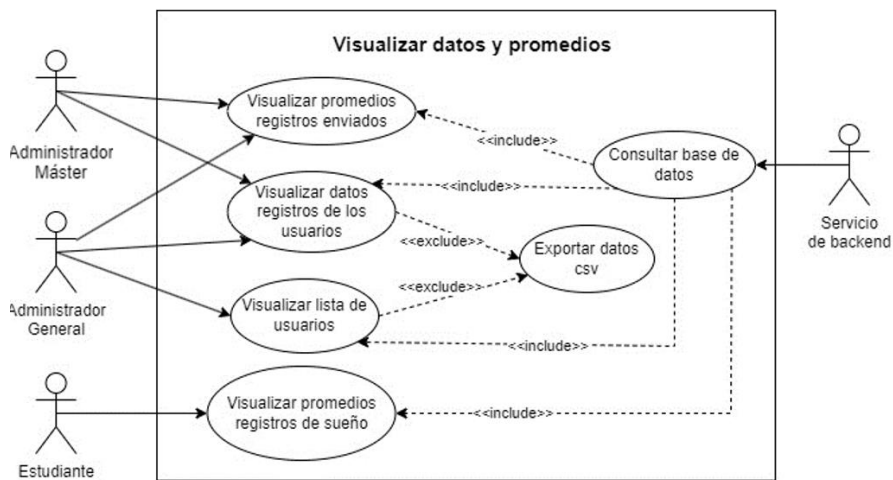
ANEXO III.3 Casos de uso



(a) Caso de uso autenticación estudiante y administrador



(b) Casos de uso personalización de perfil estudiante y administrador



(c) Casos de uso visualizar datos y promedios, administradores y estudiante

ANEXO III.4 Diseño de la interfaz gráfica de los módulos del prototipo

Módulo de registro – cuenta Administrador General y Estudiante

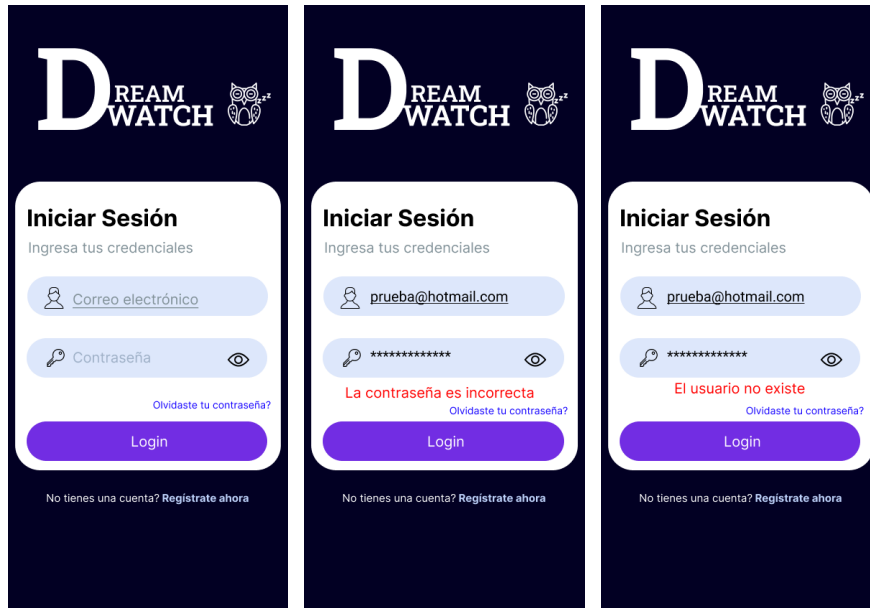
El módulo de registro consta de tres campos que son: correo, contraseña y confirmar contraseña. Para el registro se debe contar con un correo electrónico válido, este será verificado por el servicio de *backend*. El prototipo verificará que los campos se encuentren llenos y en caso de presentarse algún error se mostrará un mensaje en color rojo indicando el problema.

The image displays three sequential screenshots of the registration form in the DREAM WATCH application. Each screenshot features the 'DREAM WATCH' logo at the top and a 'Registro' section with three input fields: 'Correo:', 'Contraseña:', and 'Confirmar contraseña:'. A purple 'Registrar' button is located at the bottom of each form.

- First Screenshot:** The form is empty. The 'Correo:' field contains an envelope icon, 'Contraseña:' contains a lock icon, and 'Confirmar contraseña:' contains a speech bubble icon.
- Second Screenshot:** The 'Correo:' field is filled with 'prueba@hotmail.com'. A red error message 'El correo ya está en uso' is displayed below the 'Confirmar contraseña:' field.
- Third Screenshot:** The 'Correo:' field is filled with 'prueba@hotmail.com'. The 'Contraseña:' field is filled with a password. A red error message 'Las contraseñas no coinciden' is displayed below the 'Confirmar contraseña:' field.

Módulo de autenticación – rol Administrador General y Estudiante

El módulo de autenticación contará con dos campos para ingresar el correo electrónico y la contraseña del usuario. La visualización de la contraseña se puede habilitar. Adicionalmente, se tiene accesos directos a los módulos de cambiar contraseña y de registro. Se muestran los mensajes de error al usuario en color rojo, informando si los campos no se encuentran completos, la cuenta no existe o si la contraseña es incorrecta.



Módulos de perfil y cambio de contraseña – rol Estudiante y Administrador

Este módulo permite personalizar el perfil del usuario con sus datos personales. A pesar de ser un módulo en común los campos de personalización difieren entre los roles. Por ejemplo, el Estudiante puede ingresar: nombre, apellido, fecha de nacimiento, sexo, nivel de estudio y área de estudio parte (a). En cambio, el Administrador únicamente puede modificar su nombre, apellido y fecha de nacimiento parte (b).



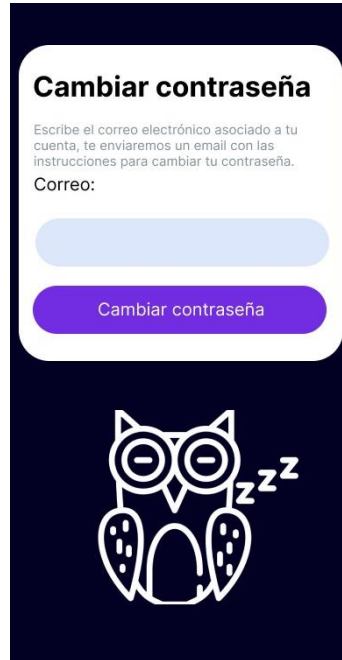
(a)



(b)

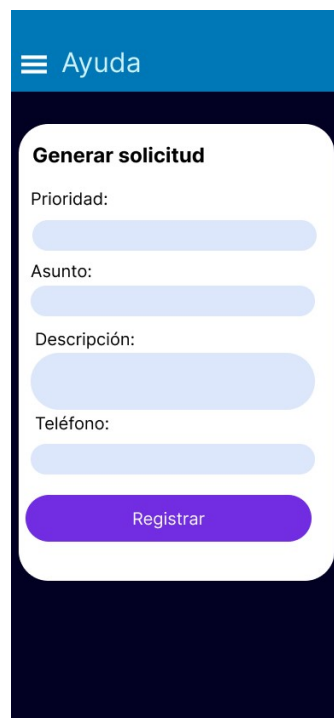
Módulo de cambio de contraseña – rol Estudiante y Administrador

Este módulo tiene un campo que solicita el ingreso de un correo electrónico para enviar el enlace de cambio de contraseña.



Módulo de ayuda – rol Estudiante y Administrador General

El módulo de ayuda permite contactar con el Administrador Máster mediante WhatsApp. Se solicitarán detalles sobre el inconveniente al usuario: prioridad, asunto, descripción y teléfono de contacto.



Módulo Dashboard – rol Estudiante

En este módulo se presentan promedios y gráficos que le permitan al Estudiante tener retroalimentación acerca de sus horas de sueño. En la Figura se muestra el módulo de Dashboard:



Módulo de administración de datos – rol Estudiante

En el módulo de administración de datos se encuentran disponibles las opciones para eliminar cuenta, eliminar registros y sincronizar registros. Al presionar alguna de las opciones se mostrará una ventana emergente para confirmar la acción deseada.



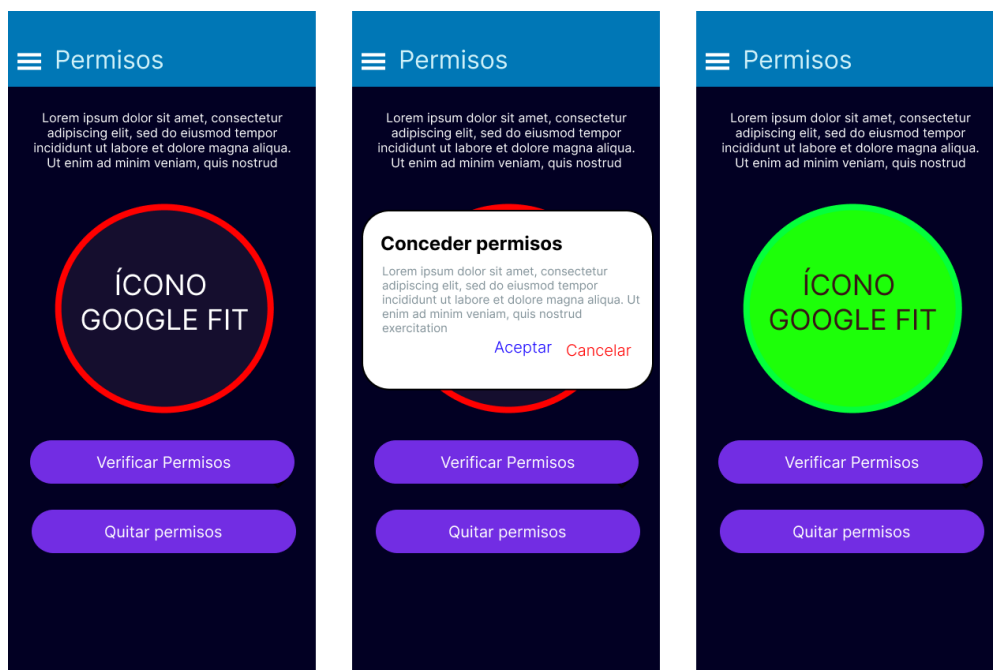
Módulo cómo usar – rol Estudiante

El módulo de cómo usar permite al usuario conocer los pasos a seguir para realizar el monitoreo del sueño, se emplea un elemento llamado *stepper* que le permitirá tener al usuario una guía rápida del prototipo.



Módulo permisos – rol Estudiante

El módulo de permisos permite al prototipo acceder a los datos de *Google Fit*, se presenta el ícono de *Google Fit*, en caso de presionar verificar permisos y vincular la cuenta de Gmail el ícono se pintará de color verde, en caso de quitar los permisos en rojo.



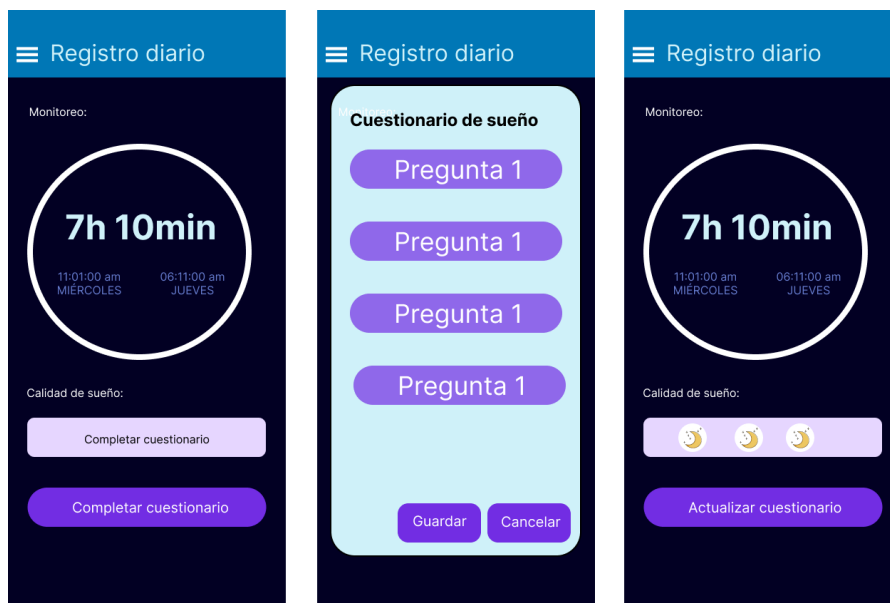
Módulo de información – rol Estudiante

El módulo Información describe el avatar asignado en el perfil según su calidad de sueño, permitiendo de esta forma dar retroalimentación al Estudiante. En la Figura se muestran los diferentes avatares que el Estudiante puede obtener:



Módulo de registro diario – rol Estudiante

Al ingresar al módulo de registro diario se visualiza el tiempo total de sueño del usuario, como detalles de la hora en que se acuesta y despierta. Se solicita completar el cuestionario de sueño, al responder el cuestionario aparece un ranking de lunas según su calidad de sueño y en base a las respuestas que brinde el usuario y el tiempo de sueño.



Módulo registros y registro semanal – rol Estudiante

El módulo de mis registros permite visualizar los registros diarios enviados por el usuario por medio del módulo registro diario, se pueden ver los datos del tiempo de sueño, la fecha de envío, del monitoreo y el sentimiento que registró el usuario al despertar. En cambio, en el módulo de registro semanal se puede ver el promedio de tiempo de sueño por día de la semana.



Módulo Visualizar/Exportar datos – rol Administrador General y Máster

Este módulo permite visualizar los registros enviados por los usuarios, los registros son recuperados de *Firebase*. Al exportar si el usuario es Administrador General exportará los registros en un archivo CSV sin los identificadores de los usuarios, en cambio el Administrador Máster podrá exportar los registros con los identificadores.



Módulo de información semanal – rol Administrador General y Máster

En este módulo los administradores podrán visualizar los promedios de tiempo de sueño por día de la semana de los registros enviados por los participantes del prototipo. Adicionalmente, se puede ver el sentimiento más frecuente registrado por los estudiantes.



Módulo usuarios– rol Administrador Máster

En este módulo el Administrador Máster podrá obtener información acerca de los usuarios con el rol Estudiante, podrá visualizar el número de registros por usuario y demás detalles como el tiempo de sueño. Adicionalmente puede exportar los datos en un archivo CSV que será almacenado en el dispositivo.

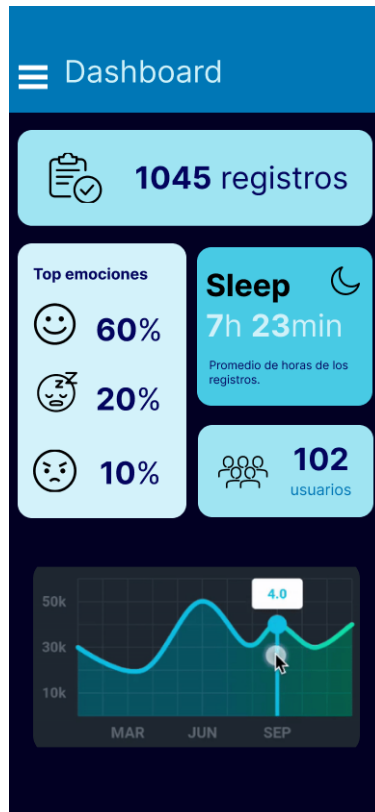
The first screenshot shows a mobile application interface with a blue header labeled 'Datos' and a save icon. Below the header, there is a table with three columns: 'Nombre', 'Apellido', and 'Tiempo sueño'. The table contains four rows of data, each with 'Estudiante' in the 'Nombre' column, 'Apellido' in the 'Apellido' column, and '360.3' in the 'Tiempo sueño' column.

Nombre	Apellido	Tiempo sueño
Estudiante	Apellido	360.3
Estudiante	Apellido	360.3
Estudiante	Apellido	360.3
Estudiante	Apellido	360.3

The second screenshot shows the same mobile application interface, but with a white dialog box overlaid on the table. The dialog box contains the text 'Seleccionar lugar para almacenar el archivo CSV'.

Módulo Dashboard – rol Administrador General y Máster

El módulo Dashboard permite a los administradores visualizar distintos promedios relacionados a los registros diarios enviados por los usuarios con el rol Estudiante, estos datos son recuperados desde *Firebase*. Adicionalmente se podrá visualizar el número de usuarios del prototipo.



ANEXO IV.1 APK final del prototipo

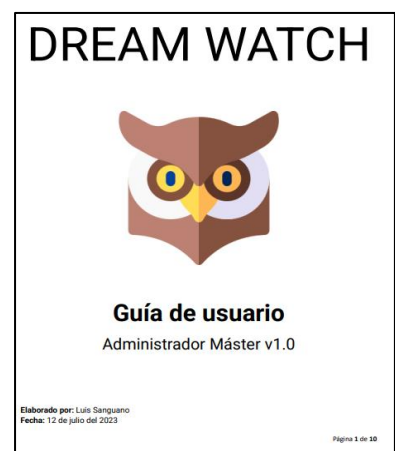
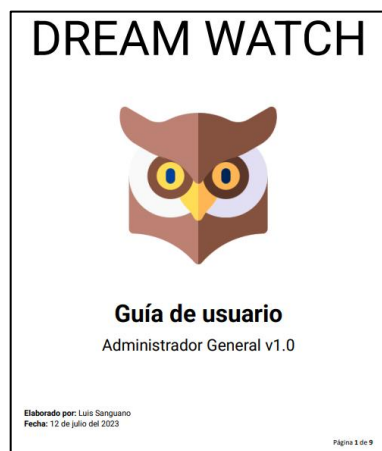
Enlace: [Dream Watch v04.apk](#)

ANEXO IV.2 Código fuente del prototipo

Enlace: [dream_watch](#)

ANEXO V.1 Guías de usuario

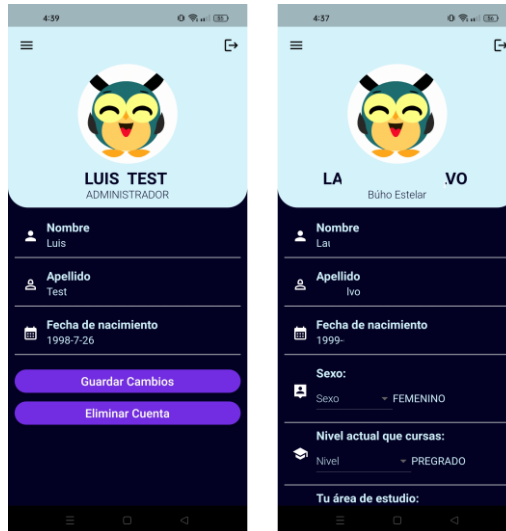
- Guía de Usuario Estudiantes
Enlace: [DREAM WATCH v1.0.pdf](#)
- Guía de usuario Administrador General
Enlace: [Guía de Usuario Administrador General.pdf](#)
- Guía de usuario Administrador Máster
Enlace: [Guía de Usuario Administrador Máster.pdf](#)



ANEXO VI.1 Pruebas funcionales

Módulo de Perfil

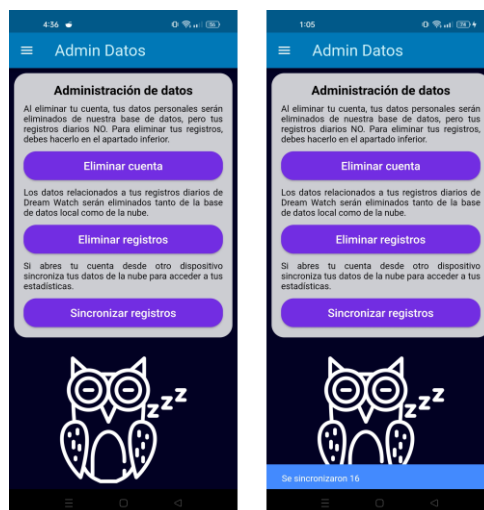
El módulo de perfil se encuentra diferenciado entre los administradores y los estudiantes. Los campos con los datos personales varían entre roles como se muestra en la Figura (a).



(a) Perfil Administrador General y Estudiantes

Módulo de sincronización – rol Estudiante

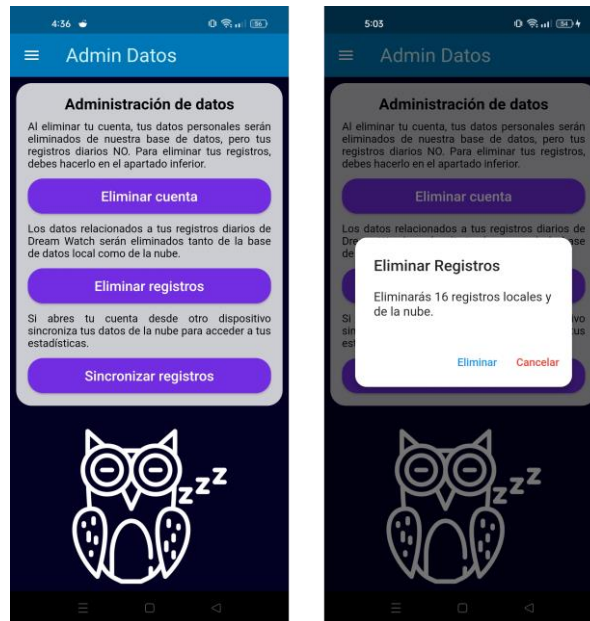
El momento de iniciar sesión en un dispositivo nuevo es necesario sincronizar los datos de Firebase con la base de datos local, al momento de presionar el botón se sincronizan los datos y se muestra el número de registros sincronizados en un mensaje mostrado en la parte inferior de la pantalla.



(b) Sincronizar registros

Módulo de eliminación de registros – rol Estudiante

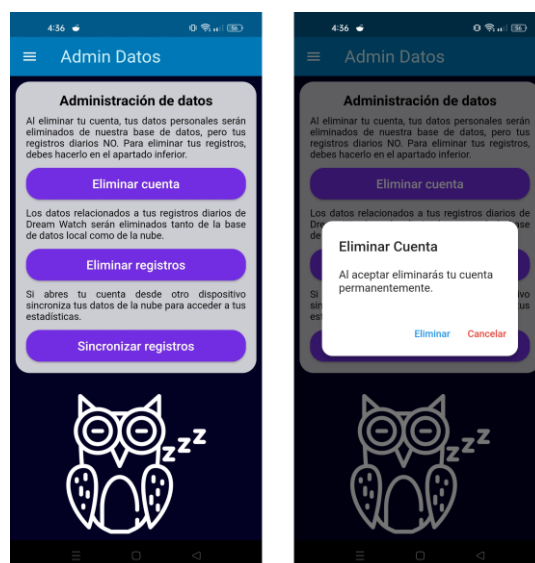
Para eliminar los registros tanto localmente como de Firebase es necesario acceder al módulo de administración de datos y presionar el botón de Eliminar Registros, al presionar se presenta una pantalla emergente indicando el número de registros a eliminar y se espera la confirmación de la eliminación.



(c) Módulo de eliminación de los registros

Módulo de eliminar cuenta – rol Estudiante

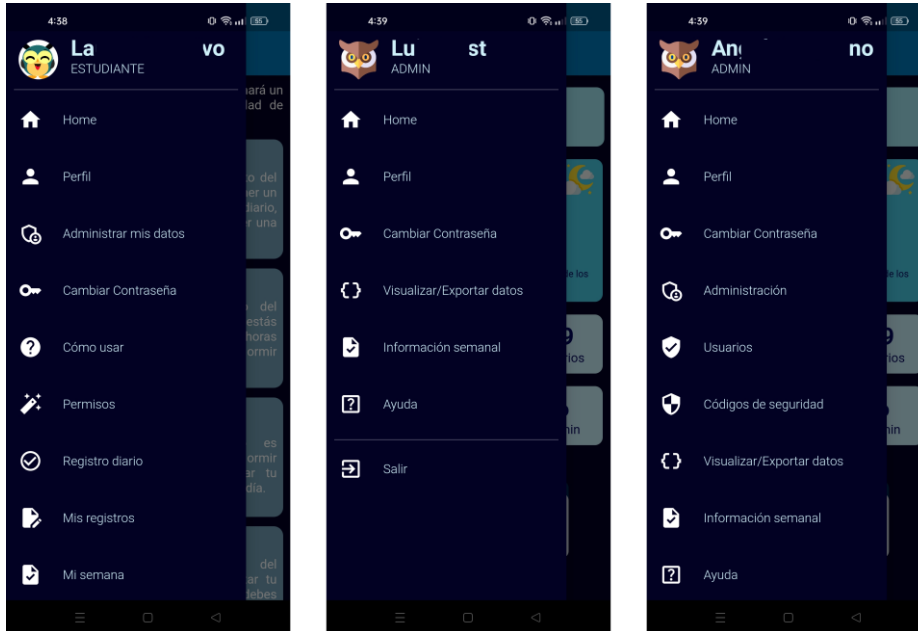
Para eliminar la cuenta se presenta una ventana emergente solicitando la confirmación por parte del usuario.



(d) Módulo de eliminación de la cuenta

Menú lateral – Estudiante, Administrador General y Administrador Máster

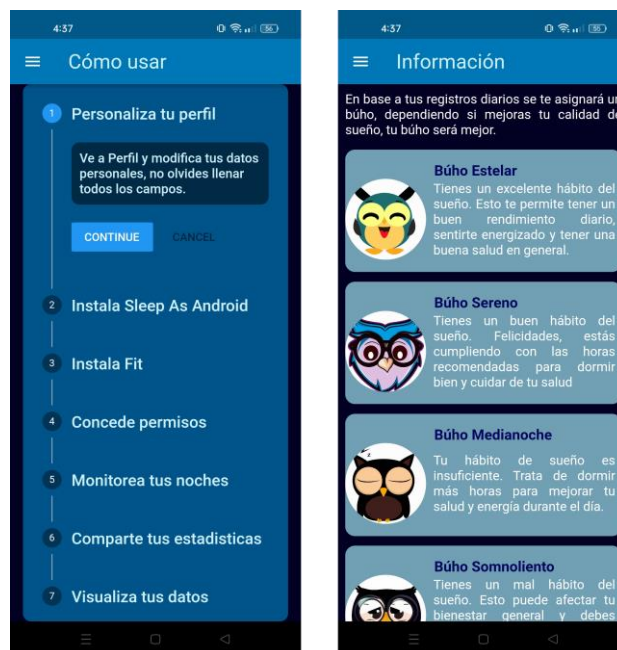
Las opciones disponibles en cada menú lateral corresponde al rol del usuario que inicie sesión en el prototipo.



(e) Menú lateral de los diferentes roles del prototipo

Módulos informativos – rol Estudiante

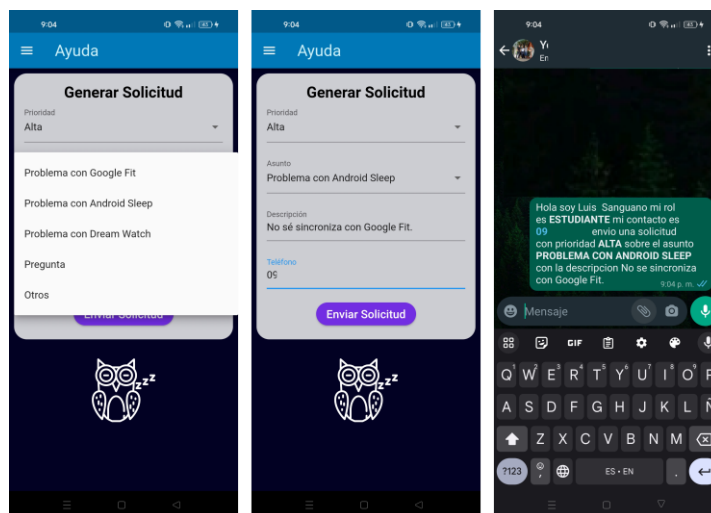
Los módulos informativos “cómo usar” e “información” permiten al usuario conocer cómo realizar el monitoreo del sueño y cómo interpretar el avatar de su perfil.



(f) Módulo de cómo usar e información

Módulo de ayuda - Estudiante, Administrador General

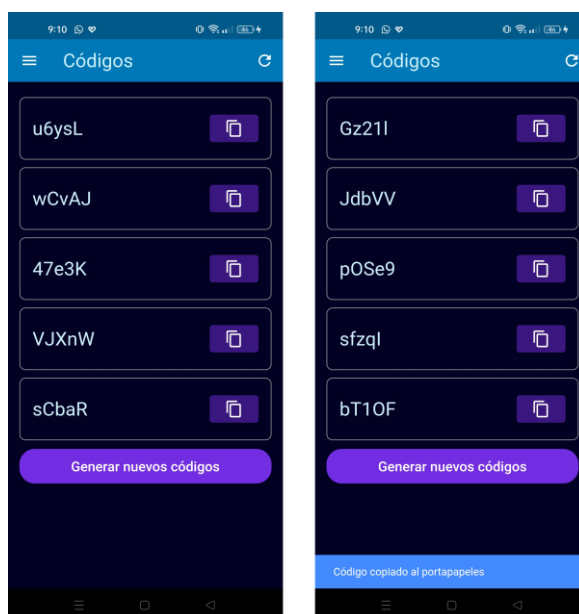
Este rol permite contactar por WhatsApp con el Administrador Máster generando una solicitud detallando el problema.



(g) Módulo de ayuda

Módulo de códigos de seguridad – cuenta Administrador Máster

El Administrador Máster tiene la capacidad de visualizar los códigos de seguridad que emplean los nuevos Administradores Generales para generar su cuenta. Estos códigos se presentan en pantalla y pueden ser copiados empleando el botón con el ícono correspondiente. Adicionalmente, se tiene un botón para generar nuevos códigos, estos se modificarán en *Firestore* automáticamente.



(h) Módulo de códigos de seguridad

ANEXO VII.1 Resultados de encuesta para la distribución del APK del prototipo

N°	¿Qué versión de Android tiene su celular?	¿Desea que el APK (instalador) sea enviado a su WhatsApp?
1	8	Sí
2	9	Sí
3	12	Sí
4	11	Sí
5	13	Sí
6	12	Sí
7	13	Sí
8	12	Sí
9	13	Sí
10	9	Sí
11	12	Sí
12	10	Sí
13	13	Sí
14	11	Sí
15	12	Sí
16	12	Sí
17	11	No
18	10	Sí
19	11	Sí
20	12	Sí
21	11	Sí
22	11	No
23	13	Sí
24	13	Sí
25	12	Sí
26	12	Sí
27	12	No
28	12	Sí

ANEXO VII.2 Datos anonimizados de los participantes de las pruebas de validación

N°	Registros enviados	Edad	Sexo	Área de Estudio	Nivel de Estudio
1	15	24	FEMENINO	INGENIERÍA Y TECNOLOGÍA	PREGRADO
2	15	24	FEMENINO	INGENIERÍA Y TECNOLOGÍA	PREGRADO
3	14	24	MASCULINO	INGENIERÍA Y TECNOLOGÍA	PREGRADO
4	17	24	MASCULINO	INGENIERÍA Y TECNOLOGÍA	PREGRADO
5	10	22	MASCULINO	INGENIERÍA Y TECNOLOGÍA	PREGRADO
6	16	23	MASCULINO	INGENIERÍA Y TECNOLOGÍA	PREGRADO
7	13	24	MASCULINO	INGENIERÍA Y TECNOLOGÍA	PREGRADO
8	18	23	MASCULINO	INGENIERÍA Y TECNOLOGÍA	PREGRADO
9	12	24	MASCULINO	INGENIERÍA Y TECNOLOGÍA	PREGRADO
10	17	22	MASCULINO	INGENIERÍA Y TECNOLOGÍA	PREGRADO
11	17	24	MASCULINO	INGENIERÍA Y TECNOLOGÍA	PREGRADO
12	8	24	MASCULINO	INGENIERÍA Y TECNOLOGÍA	PREGRADO
13	15	30	MASCULINO	INGENIERÍA Y TECNOLOGÍA	PREGRADO
14	13	24	MASCULINO	INGENIERÍA Y TECNOLOGÍA	PREGRADO
15	14	23	MASCULINO	INGENIERÍA Y TECNOLOGÍA	PREGRADO
16	17	23	FEMENINO	INGENIERÍA Y TECNOLOGÍA	PREGRADO
17	14	23	MASCULINO	INGENIERÍA Y TECNOLOGÍA	PREGRADO
18	14	23	MASCULINO	INGENIERÍA Y TECNOLOGÍA	PREGRADO
19	12	22	MASCULINO	INGENIERÍA Y TECNOLOGÍA	PREGRADO
20	16	24	FEMENINO	INGENIERÍA Y TECNOLOGÍA	PREGRADO
21	11	27	MASCULINO	INGENIERÍA Y TECNOLOGÍA	PREGRADO
22	14	22	MASCULINO	INGENIERÍA Y TECNOLOGÍA	PREGRADO
23	10	24	MASCULINO	INGENIERÍA Y TECNOLOGÍA	PREGRADO
24	16	27	MASCULINO	INGENIERÍA Y TECNOLOGÍA	PREGRADO
25	13	27	FEMENINO	INGENIERÍA Y TECNOLOGÍA	PREGRADO
26	14	24	MASCULINO	INGENIERÍA Y TECNOLOGÍA	PREGRADO
27	8	22	FEMENINO	INGENIERÍA Y TECNOLOGÍA	PREGRADO
28	14	23	MASCULINO	INGENIERÍA Y TECNOLOGÍA	PREGRADO

ANEXO VII.3 Resumen de resultados sobre los registros de sueño por participante

N°	Edad	Género	Versión Android	Área de Estudio	Nivel	Semestre	Créditos	Tiempo sueño [horas]	Búho	Calidad
1	22	M	12	INGENIERÍA Y TECNOLOGÍA	PREGRADO	7	15	5,52	Sereno	6,44
2	22	M	11	INGENIERÍA Y TECNOLOGÍA	PREGRADO	6	17	6,37	Medianoche	8,07
3	22	M	9	INGENIERÍA Y TECNOLOGÍA	PREGRADO	8	15	6,59	Estelar	5,45
4	22	M	12	INGENIERÍA Y TECNOLOGÍA	PREGRADO	6	13	6,52	Sereno	6,00
5	22	M	13	INGENIERÍA Y TECNOLOGÍA	PREGRADO	8	15	5,64	Sereno	6,69
6	22	F	8	INGENIERÍA Y TECNOLOGÍA	PREGRADO	6	15	7,37	Estelar	3,25
7	23	M	10	INGENIERÍA Y TECNOLOGÍA	PREGRADO	8	11	5,99	Sereno	7,50
8	23	M	12	INGENIERÍA Y TECNOLOGÍA	PREGRADO	6	15	6,65	Sereno	6,07
9	23	M	13	INGENIERÍA Y TECNOLOGÍA	PREGRADO	8	10	6,55	Sereno	7,69
10	23	M	12	INGENIERÍA Y TECNOLOGÍA	PREGRADO	6	12	7,28	Sereno	7,54
11	23	F	9	INGENIERÍA Y TECNOLOGÍA	PREGRADO	8	12	5,70	Sereno	5,94
12	23	M	12	INGENIERÍA Y TECNOLOGÍA	PREGRADO	8	15	5,02	Sereno	5,44
13	24	M	11	INGENIERÍA Y TECNOLOGÍA	PREGRADO	8	12	7,81	Estelar	4,33
14	24	M	8	INGENIERÍA Y TECNOLOGÍA	PREGRADO	8	10	3,70	Sereno	7,45
15	24	M	12	INGENIERÍA Y TECNOLOGÍA	PREGRADO	8	9	3,39	Medianoche	7,71
16	24	M	8	INGENIERÍA Y TECNOLOGÍA	PREGRADO	7	10	6,06	Sereno	5,20
17	24	F	11	INGENIERÍA Y TECNOLOGÍA	PREGRADO	8	12	7,73	Estelar	3,57
18	24	M	12	INGENIERÍA Y TECNOLOGÍA	PREGRADO	6	14	6,40	Sereno	6,71
19	24	F	11	INGENIERÍA Y TECNOLOGÍA	PREGRADO	6	12	7,45	Sereno	6,64
20	24	M	13	INGENIERÍA Y TECNOLOGÍA	PREGRADO	7	15	5,22	Sereno	6,36
21	24	M	13	INGENIERÍA Y TECNOLOGÍA	PREGRADO	7	15	6,17	Estelar	4,50
22	24	M	11	INGENIERÍA Y TECNOLOGÍA	PREGRADO	8	14	5,09	Estelar	4,13
23	24	M	13	INGENIERÍA Y TECNOLOGÍA	PREGRADO	8	15	5,77	Sereno	6,45
24	24	F	12	INGENIERÍA Y TECNOLOGÍA	PREGRADO	8	15	5,57	Sereno	6,86
25	27	F	13	INGENIERÍA Y TECNOLOGÍA	PREGRADO	8	9	6,48	Sereno	5,83
26	27	M	10	INGENIERÍA Y TECNOLOGÍA	PREGRADO	8	9	5,14	Sereno	8,00
27	27	M	11	INGENIERÍA Y TECNOLOGÍA	PREGRADO	6	10	5,35	Sereno	7,45
28	30	M	12	INGENIERÍA Y TECNOLOGÍA	PREGRADO	7	11	5,18	Sereno	8,47

ANEXO VII.4 Análisis de los datos recolectados de los registros de sueño obtenidos de las pruebas de validación

Enlace: [Análisis de Datos.xlsx](#)



ANEXO VII.5 Promedio de horas de sueño e índice de calidad catalogado por edad

Edad	Sueño [min]	Índice de calidad
22-24	6,06	6,08
25-27	5,66	7,10
28-30	5,18	8,47

ANEXO VII.6 Promedio de horas de sueño e índice de calidad catalogado por semestre

Semestre	Sueño [min]	Índice de calidad
Sexto semestre	6,67	6,47
Séptimo semestre	5,63	6,19
Octavo semestre	5,74	6,20

ANEXO VII.7 Promedio de horas de sueño e índice de calidad catalogado por créditos educativos

Créditos	Sueño [min]	Índice de calidad
9 a 10	5,24	7,05
11 a 12	6,73	6,28
13 a 14	6,00	5,61
15 a 17	5,99	5,96

ANEXO VII.8 Promedio de horas de sueño e índice de calidad catalogado por género

Género	Sueño [min]	Calidad
Femenino	6,72	5,35
Masculino	5,79	6,53

ANEXO VIII.1 Encuesta para la distribución del APK del prototipo

Enlace de la encuesta: <https://forms.gle/B4ryZALV64rHrsEs9>

Preguntas:

1. Nombre y apellido
2. ¿Qué versión de Android tiene su celular?
3. ¿Desea que el APK (instalador sea enviado a su WhatsApp)?

ANEXO VIII.2 Encuesta prueba de validación rol Estudiante

Enlace de la encuesta: <https://forms.gle/hh1DNUvQ73QYnA1N8>

Preguntas:

¿Pudo instalar el prototipo de Dream Watch en su dispositivo móvil?

Sí/No

¿Pudo ejecutar el prototipo de Dream Watch en su dispositivo móvil?

Sí/No

¿Pudo seleccionar el rol previo a crear su cuenta o iniciar sesión el prototipo de Dream Watch?

Sí/No

¿Pudo crear una cuenta de estudiante con su correo electrónico en el prototipo de Dream Watch?

Sí/No

¿Recibió el correo electrónico de confirmación al momento de crear su cuenta de estudiante en el prototipo de Dream Watch?

Sí/No

¿Pudo iniciar sesión en el prototipo de Dream Watch sin problemas?

Sí/No

¿Pudo visualizar el mensaje de error indicándole que la contraseña es incorrecta?

Sí/No

¿La sesión se mantenía abierta cada vez que volvía a ejecutar el prototipo?

Sí/No

¿Pudo personalizar su cuenta con sus datos personales en el apartado de perfil del prototipo de Dream Watch?

Sí/No

¿Pudo modificar los datos de su cuenta como nombre, área y nivel de estudios, etc?

Sí/No

¿En caso de haber necesitado cambiar su contraseña lo pudo realizar sin problema?

Sí/No

¿Pudo visualizar los promedios y gráficos sobre su tiempo de sueño en el Dashboard del prototipo de Dream Watch?

Sí/No

¿Le resultó interesante el visualizar el promedio de horas de sueño por día de la semana en el Dashboard del prototipo de Dream Watch?

Sí/No

¿Le resultó interesante visualizar la comparación entre sus promedios de sueño y los promedios globales mostrados en el Dashboard del prototipo de Dream Watch?

Sí/No

¿Pudo visualizar sus registros enviados los últimos 7 y 14 días?

Sí/No

¿Pudo visualizar sin problemas su promedio de sueño y estado emocional por día de la semana en el módulo "mi semana"?

Sí/No

¿Le resultó útil conocer el promedio de tiempo de sueño por día de la semana y cómo se sentía la mayoría de las veces al despertar?

Sí/No

¿Pudo sincronizar sus registros diarios de sueño al instalar la última versión del prototipo (Dream Watch v03.apk)?

Sí/No

¿Pudo conceder los permisos de Google Fit al prototipo de Dream Watch?

Sí/No

¿Pudo generar una solicitud en el apartado de "ayuda" del prototipo de Dream Watch?

Sí/No

¿El soporte brindado por el Administrador Máster al enviar una solicitud por WhatsApp le ayudó a solventar sus problemas?

Sí/No

¿Al ingresar al módulo de registro diario, y previo a completar el cuestionario de calidad de sueño, pudo visualizar la información de sueño como fecha y hora en la que se durmió/acostó, total de tiempo que durmió, etc?

Sí/No

¿Pudo enviar sus registros diarios de sueño?

Sí/No

¿Pudo visualizar la retroalimentación presentada en forma de ranking de lunas?

Sí/No

¿Pudo completar el cuestionario de calidad de sueño?

Sí/No

¿Pudo visualizar el avatar de búho asignado a su perfil en el prototipo de Dream Watch?

Sí/No

¿Le parece interesante que la asignación de su avatar de búho sea en base a su calidad de sueño?

Sí/No

¿Considera que la información mostrada en el prototipo de Dream Watch en sus diferentes módulos le pueden ayudar identificar posibles malos hábitos de sueño?

Sí/No

En el apartado de perfil bajo su nombre se encontrará el alias de su búho, a continuación, escoja el búho que tiene actualmente su perfil:

Búho estelar/Búho sereno/Búho medianoche/Búho somnoliento

¿Tuvo algún inconveniente con respecto al tamaño que ocupa el prototipo de Dream Watch en su dispositivo móvil?

Sí/No

¿La paleta de colores escogida para el prototipo de Dream Watch le pareció adecuada tomando en cuenta la naturaleza del prototipo (monitoreo del sueño)?

Sí/No

¿Considera que los diferentes formularios del prototipo de Dream Watch contienen los campos mínimos y necesarios para el funcionamiento del prototipo?

Sí/No

¿Pudo acceder a los diferentes módulos de manera sencilla e intuitiva?

Sí/No

¿Considera que el tamaño y tipo de fuente utilizada en los textos del prototipo de Dream Watch se puede leer sin dificultad?

Sí/No

ANEXO VIII.3 Encuesta prueba de validación cuenta Administrador General

Enlace: <https://forms.gle/YmaeUV98SKKmjwSG7>

¿Pudo instalar el prototipo de Dream Watch en su dispositivo móvil?

Sí/No

¿Pudo ejecutar el prototipo de Dream Watch en su dispositivo móvil?

Sí/No

¿Pudo seleccionar el rol previo a crear su cuenta o iniciar sesión el prototipo de Dream Watch?

Sí/No

¿Pudo verificar el código de seguridad proporcionado por el Administrador Máster?

Sí/No

¿Pudo crear una cuenta de Administrador con su correo electrónico en el prototipo de Dream Watch?

Sí/No

¿Pudo enviar la solicitud para la verificación de su cuenta?

Sí/No

¿Su cuenta fue verificada exitosamente?

Sí/No

¿Pudo iniciar sesión en el prototipo de Dream Watch sin problemas?

Sí/No

En caso de ingresar mal su contraseña, ¿pudo visualizar el mensaje de error indicándole que la contraseña es incorrecta?

Sí/No

¿La sesión se mantenía abierta cada vez que volvía a ejecutar el prototipo?

Sí/No

¿Pudo visualizar los promedios de sueño de los usuarios en el Dashboard?

Sí/No

¿Pudo visualizar el número de usuarios y administradores en el Dashboard?

Sí/No

¿Le parece interesante la información que se muestra en el Dashboard?

Sí/No

¿Pudo personalizar su cuenta con sus datos personales en el apartado de perfil del prototipo de Dream Watch?

Sí/No

¿Pudo editar los datos de su cuenta como nombre, apellido y fecha de nacimiento?

Sí/No

¿En caso de haber necesitado cambiar su contraseña lo pudo realizar sin problema?

Sí/No

¿Pudo generar una solicitud en el apartado de "ayuda" del prototipo de Dream Watch?

Sí/No

¿El soporte brindado por el Administrador Máster al enviar una solicitud por WhatsApp le ayudó a solventar sus problemas?

Sí/No

¿Pudo visualizar los promedios de sueño de los usuarios por día de la semana en el módulo "Información semanal"?

Sí/No

¿Le parece interesante la información mostrada en el módulo "Información semanal"?

Sí/No

¿Pudo visualizar la tabla con los registros de sueño enviados por los usuarios en el módulo de "visualizar/exportar datos"?

Sí/No

¿Pudo exportar y almacenar el archivo CSV con los registros de sueño de los usuarios en su dispositivo?

Sí/No

¿Tuvo algún inconveniente con respecto al tamaño que ocupa el prototipo de Dream Watch en su dispositivo móvil?

Sí/No

ANEXO VIII.4 Encuesta prueba de validación cuenta Administrador Máster

Enlace: <https://forms.gle/N8UrfWyYYAvC5BGd8>

¿Pudo iniciar sesión en el prototipo de Dream Watch sin problemas?

Sí/No

¿La sesión se mantenía abierta cada vez que volvía a ejecutar el prototipo?

Sí/No

¿Pudo visualizar y copiar los códigos de seguridad?

Sí/No

¿Pudo generar nuevos códigos de seguridad?

Sí/No

¿Pudo aprobar solicitudes para nuevas cuentas de administración?

Sí/No

¿Pudo visualizar la tabla con los datos de los usuarios del prototipo y el número correspondiente de registros enviados?

Sí/No

¿Pudo exportar y almacenar los datos de los usuarios en un archivo CSV en su dispositivo?

Sí/No

¿Pudo visualizar los promedios de sueño de los usuarios en el Dashboard?

Sí/No

¿Le parece interesante la información que se muestra en el Dashboard?

Sí/No

¿Pudo visualizar los promedios de sueño de los usuarios por día de la semana en el módulo "Información semanal"?

Sí/No

¿Le parece interesante conocer el tiempo promedio de sueño y el sentimiento más frecuente por día de la semana de los usuarios?

Sí/No

¿Pudo visualizar la tabla con los registros de sueño enviados por los usuarios en el módulo de "visualizar/exportar datos"?

Sí/No

¿Pudo exportar y almacenar el archivo CSV con los registros de sueño de los usuarios incluyendo sus identificadores en su dispositivo?

Sí/No

ANEXO IX.1 Instancia Tablero Kanban (capítulo de Resultados, Conclusiones y Recomendaciones)

The image shows a Kanban board interface for a project titled "Trabajo de Integración Curricular". The board is organized into three columns: "Por hacer" (To Do), "En proceso" (In Progress), and "Terminado" (Done). Each task card includes a title, a progress indicator (a circle with a percentage), and an assigned user's profile picture. The "Terminado" column contains the most tasks, indicating that most of the work has been completed.

Trabajo de Integración Curricular
Gratis

Tablero Kanban: Trabajo de Integración Curricular ☆ Visible para el Espacio de trabajo

Por hacer

- + Añada una tarjeta

En proceso

- Actualizar tablero Kanban
- Ejecutar pruebas de funcionalidad del prototipo.
- Corregir errores reportados en las pruebas de funcionalidad.
- Entregar APK final del prototipo.
- Escribir capítulo 3
- Entrega del borrador final del TIC.
- Realizar análisis preliminar de los datos obtenidos de las pruebas de funcionalidad
- + Añada una tarjeta

Terminado

- Crear el Tablero Kanban del componente.
- Recopilar información sobre los tópicos a ser analizados en el Marco Teórico.
- Presentar primer borrador del Marco Teórico.
- Presentar borrador final del Marco teórico.
- Analizar los requisitos del prototipo.
- Modelar el prototipo en base a los requerimientos identificados.
- Instalar herramientas para la codificación del prototipo.
- Definir los requisitos no funcionales y funcionales del prototipo.
- Configuración de servicios de Firebase.
- Codificación capa de negocio.
- Desarrollar el diseño visual para la interfaz gráfica del prototipo.
- Codificación capa de datos.