



UNIVERSITAT DE
BARCELONA

Trabajo de Fin de Grado

GRADO DE INGENIERÍA INFORMÁTICA

**Facultad de Matemáticas e Informática
Universidad de Barcelona**

Desarrollo de una Aplicación *Gamificada* para la Gestión Autónoma de la Diabetes en Niños y Adolescentes

Óscar Jaquero Sánchez

Directora: Laura Igual Muñoz
Realizado en: Departamento de
Matemáticas e Informática
Barcelona, 10 de junio de 2024

Resumen

Este Trabajo de Fin de Grado en Ingeniería Informática está centrado en el desarrollo de una aplicación lúdica que utiliza métodos de gamificación para ayudar a niños y adolescentes con Diabetes de Tipo 1 y Tipo 2 a seguir su tratamiento diario de manera autónoma. El objetivo es hacer que el tratamiento sea más sencillo y ameno, transformando el seguimiento diario, que puede llegar a ser un proceso pesado, en una experiencia más atractiva.

El nombre que he decidido escoger para esta aplicación es **Easulin**. El nombre Easulin surge de la combinación de "*Easy*" para simplificar el tratamiento de la diabetes e "*Insulin*" haciendo referencia a la hormona asociada con esta condición médica.

Para el desarrollo de Easulin, se utilizaron diversas tecnologías avanzadas en constante evolución. En el frontend se emplearon: Kotlin, Jetpack Compose y Material Design 3. Para la gestión de *backend* seleccioné: Firebase Authentication y Cloud Firestore. Este proceso incluyó una etapa de autoaprendizaje que fue decisiva para dominar estas herramientas.

A lo largo del proyecto, se lograron todos los objetivos, siguiendo la planificación prevista inicialmente. Aunque hubo algunos ajustes en los tiempos, el resultado final es una aplicación con una interfaz intuitiva y moderna, valorada satisfactoriamente por el grupo de voluntarios que participaron en el testeo. Los voluntarios escogidos para esta prueba, que se encontraban dentro del rango de edad para el que va destinada la aplicación, incluían a un adolescente con Diabetes de Tipo 1, lo que proporcionó valiosos comentarios sobre la usabilidad y el potencial de Easulin.

Easulin tratará de ayudar tanto a niños como a adolescentes a gestionar su enfermedad de manera más efectiva, proporcionando una herramienta que no solo facilita el seguimiento del tratamiento, sino que también hace que jóvenes diagnosticados se sientan motivados y con un interés adicional por su enfermedad .

Resum

Aquest Treball de Fi de Grau en Enginyeria Informàtica està centrat en el desenvolupament d'una aplicació lúdica que utilitza mètodes de ludificació per a ajudar a nens i adolescents amb Diabetis de Tipus 1 i Tipus 2 a seguir el seu tractament diari de manera autònoma. L'objectiu és fer que el tractament sigui més senzill i amè, transformant el seguiment diari, que pot arribar a ser un procés pesat, en una experiència més atractiva.

El nom que he decidit triar per a aquesta aplicació és **Easulin**. El nom Easulin sorgeix de la combinació de "*Easy*" per a simplificar el tractament de la diabetis i "*Insulin*" fent referència a l'hormona associada amb aquesta condició mèdica.

Per al desenvolupament de Easulin, es van utilitzar diverses tecnologies avançades en constant evolució. En el *frontend* es van emprar: Kotlin, Jetpack Compose i Material Design 3. Per a la gestió de *backend* vaig seleccionar: Firebase Authentication i Cloud Firestore. Aquest procés va incloure una etapa d'autoaprenentatge que va ser decisiva per a dominar aquestes eines.

Al llarg del projecte, es van aconseguir tots els objectius, seguint la planificació prevista inicialment. Encara que va haver-hi alguns ajustos en els temps, el resultat final és una aplicació amb una interfície intuïtiva i moderna, valorada satisfactòriament pel grup de voluntaris que van participar en el testatge. Els voluntaris triats per a aquesta prova, que es trobaven dins del rang d'edat per al qual va destinada l'aplicació, incloïen a un adolescent amb Diabetis de Tipus 1, la qual cosa va proporcionar valuosos comentaris sobre la usabilitat i el potencial de Easulin.

Easulin tractarà d'ajudar tant a nens com a adolescents a gestionar la seva malaltia de manera més efectiva, proporcionant una eina que no sols facilita el seguiment del tractament, sinó que també fa que joves diagnosticats se sentin motivats i amb un interès addicional per la seva malaltia.

Abstract

This Final Degree Project in Computer Engineering is focused on the development of a playful application that uses gamification methods to help children and adolescents with Type 1 and Type 2 Diabetes to follow their daily treatment autonomously. The aim is to make the treatment easier and more enjoyable, transforming the daily monitoring, which can be a cumbersome process, into a more engaging experience.

The name I have decided to choose for this application is ***Easulin***. The name Easulin comes from the combination of 'Easy' to simplify the treatment of diabetes and 'Insulin' referring to the hormone associated with this medical condition.

For the development of Easulin, several advanced and constantly evolving technologies were used. For the frontend I used: Kotlin, Jetpack Compose and Material Design 3. For the backend management I selected: Firebase Authentication and Cloud Firestore. This process included a self-learning stage that was decisive to master these tools.

Throughout the project, all the objectives were achieved, following the initially planned schedule. Although there were some adjustments in the timing, the final result is an application with an intuitive and modern interface, satisfactorily assessed by the group of volunteers who participated in the testing. The volunteers chosen for this test, who were within the target age range of the app, included a teenager with Type 1 diabetes, which provided valuable feedback on the usability and potential of Easulin.

Easulin will aim to help both children and adolescents manage their disease more effectively by providing a tool that not only makes it easier to follow their treatment, but also makes young people with a diagnosis feel motivated and take an extra interest in their disease.

Índice

1 Introducción.....	4
1.1 Motivación Personal.....	4
1.2 Objetivos.....	5
2 Planificación y metodología de desarrollo.....	7
2.1 Diagrama de Gantt.....	7
2.1.1 Planificación final.....	10
2.2 Metodología.....	10
3 Análisis de datos previo.....	12
4 Diabetes.....	17
4.1 Diabetes Mellitus Tipo 1 (DM1).....	17
4.2 Diabetes Mellitus Tipo 2 (DM2).....	18
4.3 Tratamiento.....	19
5 Interés de las tecnologías seleccionadas.....	21
5.1 ¿Por qué utilizar Kotlin en vez de Java?.....	21
5.2 ¿Por qué utilizar el kit de herramientas Jetpack Compose?.....	21
5.3 ¿Por qué utilizar Material Design 3?.....	24
6 Aprendizaje del lenguaje y kit herramientas.....	26
7 Métodos de gamificación implementados.....	29
7.1 Elementos.....	30
7.2 Dinámicas.....	31
7.3 Mecánicas.....	32
8 Análisis de Funcionalidades y Requerimientos.....	34
8.1 Necesidades.....	35
8.1.1 Niños y adolescentes con Diabetes.....	35
8.1.2 Padres de niños con Diabetes.....	36

8.1.3 Personal sanitario.....	36
8.2 Requisitos funcionales.....	36
8.3 Historias de Usuario.....	38
8.4 Casos de Uso.....	39
9 Diseño visual.....	40
9.1 Paleta de colores.....	41
9.2 Nombre y Logotipo.....	42
9.3 Mockups.....	44
10 Arquitectura.....	51
10.1 Arquitectura UI, Domain y Data.....	51
10.2 Arquitectura Model, View y ViewModel.....	53
11 Base de datos utilizada – Back End.....	54
11.1 Firebase.....	54
11.1.1 Cloud Firestore.....	54
11.1.2 Firebase Authentication.....	55
11.2 Diagrama Base de Datos NoSQL Cloud Firestore.....	57
11.3 Campos de información Cloud Firestore.....	58
11.4 Base de Datos Local.....	60
12 Implementación Visual – Front End.....	61
12.1 Proceso de implementación.....	61
12.1.1 Estructura de la interfaz.....	61
12.1.2 Navegación entre pantallas.....	62
12.1.3 Componentes de navegación utilizados.....	63
12.1.4 Componentes visuales.....	65
12.1.5 Animaciones y transiciones.....	67
13 Pruebas, Testeo y Validación.....	69
13.1 Pruebas realizadas durante el desarrollo.....	69

13.2 Pruebas con un grupo de usuarios.....	71
13.2.1 Selección grupo de voluntarios.....	71
13.2.2 Metodología.....	71
13.2.3 Tareas para realizar.....	72
13.2.4 Preguntas finales.....	72
13.2.5 Resultados.....	72
14 Conclusiones.....	75
15 Bibliografía.....	76

1 Introducción

En este Trabajo de Fin de Grado en Ingeniería Informática me he centrado en la creación de una aplicación lúdica y con métodos de *gamificación* para ayudar a niños y adolescentes que padecen Diabetes, ya sea de Tipo 1 como de Tipo 2. Se da soporte para desarrollar el hábito y la constancia de seguir su tratamiento a diario de forma autónoma. El objetivo es hacer que el tratamiento sea menos pesado, convirtiéndolo en un proceso más sencillo y ameno para los que se enfrentan a esta enfermedad crónica.

El principal reto que me he propuesto es desarrollar esta aplicación. Para lograrlo, he decidido aprender un lenguaje de programación innovador, no impartido en la carrera, utilizando herramientas punteras. Estas tendrán un gran impacto a futuro por las facilidades y ayudas que nos proporcionan. Gracias a los conocimientos obtenidos de forma autónoma después del proceso de autoaprendizaje, he podido desarrollar Easulin.

Este trabajo incluye un apartado de análisis de datos, que me ayudó a decidir si la diabetes era una patología de peso en la que centrar el desarrollo de esta aplicación.

1.1 Motivación Personal

A lo largo de mi vida, he crecido en un entorno familiar con una vocación y profesión por la salud, lo que me ha dado la oportunidad de conocer y familiarizarme con diversas situaciones que suceden a diario en los hospitales y centros de salud públicos. Debido a estas experiencias, siempre ha surgido en mí un interés por el mundo de la salud.

Como todos sabemos, el seguimiento de las pautas médicas por parte de los niños y adolescentes, frecuentemente, suele ser un proceso bastante desafiante, ya que carecen de la constancia para adaptarse a las indicaciones prescritas por el personal médico. Esta situación da lugar a que los padres se enfrenten a una supervisión estricta, teniendo que estar constantemente alertas y controlando que sigan las indicaciones marcadas por los sanitarios.

Por este motivo, he decidido desarrollar una aplicación que *gamifique* este proceso, utilizando metodologías lúdicas para que niños y adolescentes perciban estas indicaciones como un reto o un desafío y no como una obligación pesada.

A lo largo de la carrera el desarrollo de aplicaciones siempre ha sido un tema que me ha llamado mucho la atención, ya que pese a requerir una gran dedicación y paciencia, la recompensa de ver el resultado final en forma de aplicación y hacerlo funcionar en nuestros dispositivos es muy gratificante.

Más allá de esto, me he propuesto el gran reto de aprender un lenguaje de programación de aplicaciones completamente desconocido para mí, el cual es Kotlin. Junto a este reto, también he decidido aprender cómo utilizar el *kit* de herramientas de Jetpack Compose y Material Design 3 los cuales nos proporciona una gran cantidad de mejoras y facilidades a la hora de desarrollar aplicaciones.

1.2 Objetivos

El objetivo principal de este Trabajo de Fin de Grado es desarrollar una aplicación lúdica utilizando métodos de *gamificación* para ayudar a los niños y adolescentes con Diabetes de Tipo 1 o Tipo 2 a seguir su tratamiento diario de manera autónoma.

Para conseguir esta aplicación, me he propuesto los siguientes objetivos:

- Desarrollar una interfaz de usuario comprensible, adaptada al rango de edades al cual está destinada la aplicación e implementar elementos de navegación simples y fácilmente manejables.
- Desarrollar la parte de *Back End* de la aplicación para que toda la información obtenida provenga de la base de datos. Así, poder mostrar información dinámica y personalizada para cada uno de los usuarios de la aplicación.
- Utilizar una correcta arquitectura de programación basada en la *Clean Architecture*, facilitando la comprensión y mantenibilidad del código mediante la separación de responsabilidades en distintas capas.
- Desarrollar la aplicación con tecnologías nuevas y avanzadas, estudiando un lenguaje de programación alternativo, prometedor y desconocido para mí, como lo es Kotlin y su *kit* de herramientas Jetpack Compose. Con este objetivo, me propongo un reto personal para mejorar mi capacidad de adaptación, aprendizaje y compromiso con las tecnologías cambiantes e innovadoras.

2 Planificación y metodología de desarrollo

2.1 Diagrama de Gantt

Mi Trabajo de Fin de Grado se ha dividido en diversas etapas. A continuación, haciendo uso de un diagrama de Gantt, muestro la planificación inicial de mi proyecto con sus fases y tareas previstas.



Ilustración 1: Diagrama de Gantt con la previsión de fechas y tareas a realizar (Planificación Inicial)

1. Definición de la Idea y Selección de Tecnologías:

Durante el primer semestre, antes de empezar con Trabajo de Fin de Grado, dediqué tiempo a reflexionar, explorar y pensar diferentes ideas que fuesen de mi interés para escoger el enfoque y el tema definitivo de mi TFG. Quise que fuese un trabajo enfocado en el desarrollo de aplicaciones y tecnológicamente puntero, por ello investigué sobre cuales

eran las mejores tecnologías y lenguajes de programación con mayor proyección a futuro en la programación de aplicaciones.

2. Investigación y Aprendizaje:

Esta segunda etapa, consistió en familiarizarme y aprender tanto el lenguaje de programación Kotlin como el *kit* de herramientas Jetpack Compose.

Esta fase tuvo una duración prevista de unas 5 semanas, incluyendo la búsqueda de diversas fuentes de información y de conocimiento, como por ejemplo:

- Foros tecnológicos.
- Sitios webs oficiales y comunitarios con información actualizada.
- Vídeos explicativos y cursos de programación.
- Códigos públicos, desarrollados por la comunidad, con estas tecnologías.
- Manuales, libros, revistas, etcétera.

El objetivo de esta etapa era obtener un nivel de conocimiento notable para empezar a programar la aplicación del proyecto.

Por otro lado, esta etapa también incluyó la investigación de diversos métodos de *gamificación* que pudiesen adaptarse adecuadamente a la aplicación, ayudando a los pacientes enfermos a mantener o adquirir la constancia que aún no poseen.

3. Diseño y Planificación de la aplicación:

Antes de desarrollar la aplicación, realicé esta tercera fase centrada en la definición de los requisitos, funcionalidades y necesidades esenciales de la aplicación. Exponiendo todos los aspectos necesarios a incluir.

Por otro lado, escogí la paleta de colores definitiva de la aplicación, diseñé diversos *mockups* para previsualizar la estética de la aplicación y tener una visión general de la apariencia y estructura de la interfaz de usuario.

Para este proceso, analizar ideas, requisitos y funcionalidades, valorar cada una, diseñar la apariencia de la aplicación y hacer el proceso de ingeniería de software, se prevé una duración de 2 semanas.

4. Desarrollo y programación de la aplicación:

Una vez concluida la fase 3, empecé la programación de la aplicación utilizando todas las tecnologías y lenguajes de programación aprendidos en las etapas anteriores. Desarrollaré las funcionalidades seleccionadas, aplicando las técnicas y métodos de *gamificación*, con el fin de crear una aplicación que haga el tratamiento de la Diabetes mucho más atractivo y motivador para los usuarios.

Preví que esta etapa sería la más larga de todo el trabajo, junto a la del aprendizaje del lenguaje y *kits* de programación, dado que consistió en el desarrollo total de la aplicación. Estimación prevista: 8 semanas.

5. Documentación y presentación

Esta última parte del proceso se incluye en los tiempos estimados de las anteriores fases. A medida que adquiría conocimientos sobre las tecnologías y lenguaje de programación, diseñaba e implementaba la aplicación, redacté la memoria escrita, detallando las evoluciones, decisiones tomadas, imprevistos, análisis, información..., es decir, redactando todo el proceso de creación según avanzaba.

Para revisar la redacción de la memoria el tiempo previsto era de unas 2 semanas. Una vez entregada la memoria, dispondré de otras 2 semanas para preparar mi exposición y presentar mi proyecto final ante el tribunal.

2.1.1 Planificación final

He logrado adaptarme al máximo a la planificación inicial que me propuse al principio del TFG y, al final, he cumplido con los plazos establecidos.

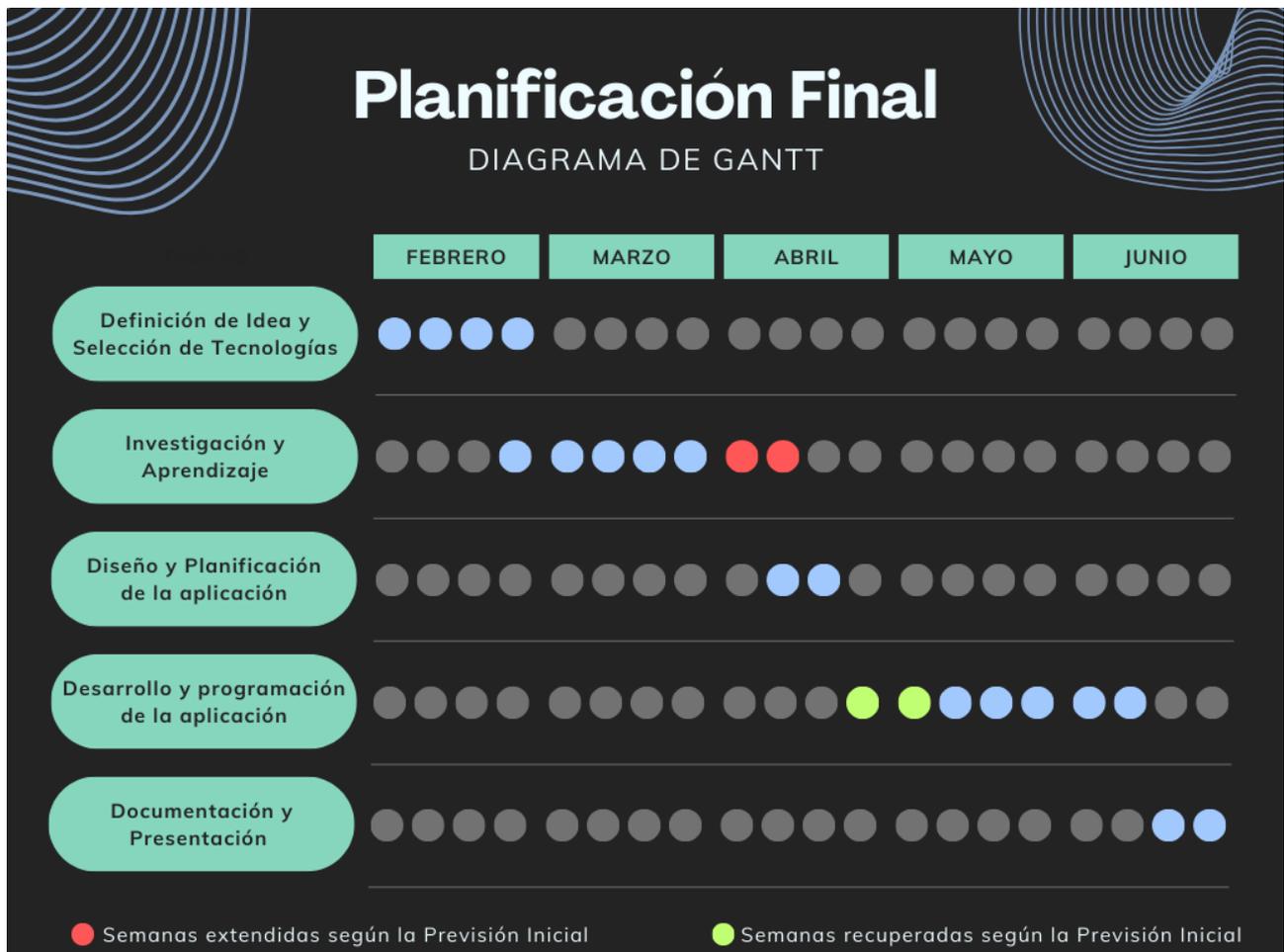


Ilustración 2: Diagrama de Gantt con la planificación final

Como podemos observar en la Ilustración 2, en la fase de aprendizaje, me extendí un par de semanas más de lo previsto, las cuales logré recuperar en la fase de desarrollo del *frontend*, gracias a la clara definición de los diseños visuales (*mockups*) y al dominio adquirido de estas tecnologías. Por lo demás, he seguido al completo la planificación marcada desde un inicio.

2.2 Metodología

Este proyecto se desarrolló utilizando la Metodología Kanban.

Kanban es una palabra que en japonés significa “señal visual”. Esta metodología se centra en mostrar visualmente de forma clara y simple la fase en la que se encuentra el proyecto, observando fácilmente la cantidad de tareas y en qué situación está cada una de ellas.

Esta metodología nos permite que el trabajo, tanto individual como colectivo, avance y fluya constantemente. Mejora la comunicación entre colaboradores del proyecto y ayuda a saber en qué tarea está trabajando cada uno.

En este caso, al ser un proyecto individual, no ha habido esta mejora de la comunicación entre miembros del grupo, pero sí que me ha ayudado a ver las tareas pendientes de realizar, situar en qué fase estaba el proyecto en cada momento y mejorar el flujo personal de trabajo.

Durante las prácticas en empresa realizadas este último curso, he podido familiarizarme con esta metodología, siendo una herramienta de trabajo de gran ayuda para todos los miembros, por ello decidí implementarla en este proyecto.

He utilizado la herramienta de gestión Trello, la cual me ha facilitado crear y utilizar de mi Tablero Kanban.

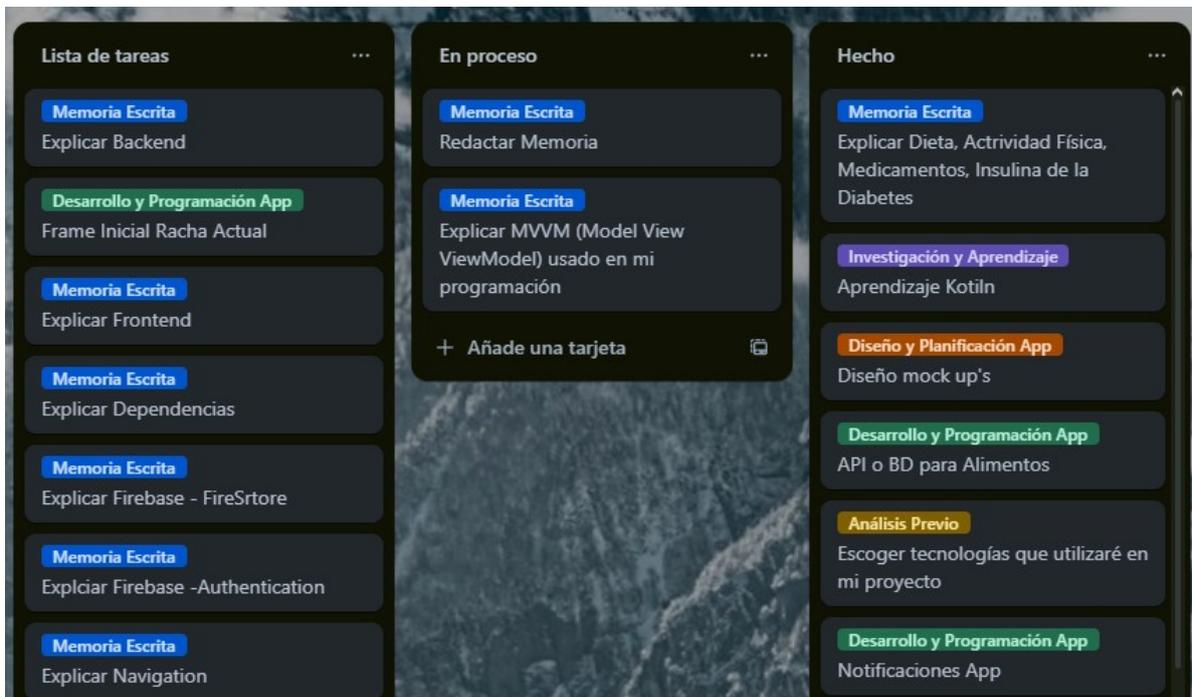


Ilustración 3: Tablero Kanban sobre el cual he ido trabajando a lo largo del Trabajo

3 Análisis de datos previo

Primero, decidí llevar a cabo un proceso de investigación y análisis de datos sobre la diabetes, fundamental para determinar la relevancia de esta enfermedad en la sociedad. Analicé la cantidad de personas que la padecen y observé si existe una cantidad notable de usuarios que potencialmente podrían llegar a utilizar la aplicación.

Para realizar este análisis de datos, accedí a la Base de Datos Clínicos de Atención Primaria (BDCAP) del Portal estadístico de la web del Ministerio de Sanidad del Gobierno de España.^[10]

Una vez familiarizado con esta base de datos, filtré la información realizando una búsqueda según el tipo de enfermedad, para obtener la cantidad de personas diagnosticadas con diabetes.

Para hacer más comprensibles estos datos, realicé un filtrado por años y por casos por cada mil personas. Estos fueron los resultados obtenidos:

	Casos	Casos por mil personas asignadas
2011	1.245.603	66,91
2012	1.817.901	70,44
2013	2.481.349	68,70
2014	2.563.267	70,84
2015	3.362.210	76,00
2016	3.350.915	74,91
2017	3.455.169	76,90
2018	3.658.431	81,21
2019	3.630.020	80,22
2020	3.623.045	79,90
2021	3.802.112	83,07
2022	3.881.529	83,90

Tabla 1: Casos de Diabetes Totales y Casos cada 1000 personas en España desde el 2011 hasta 2022 contemplando todo rango de edades. ^[10]

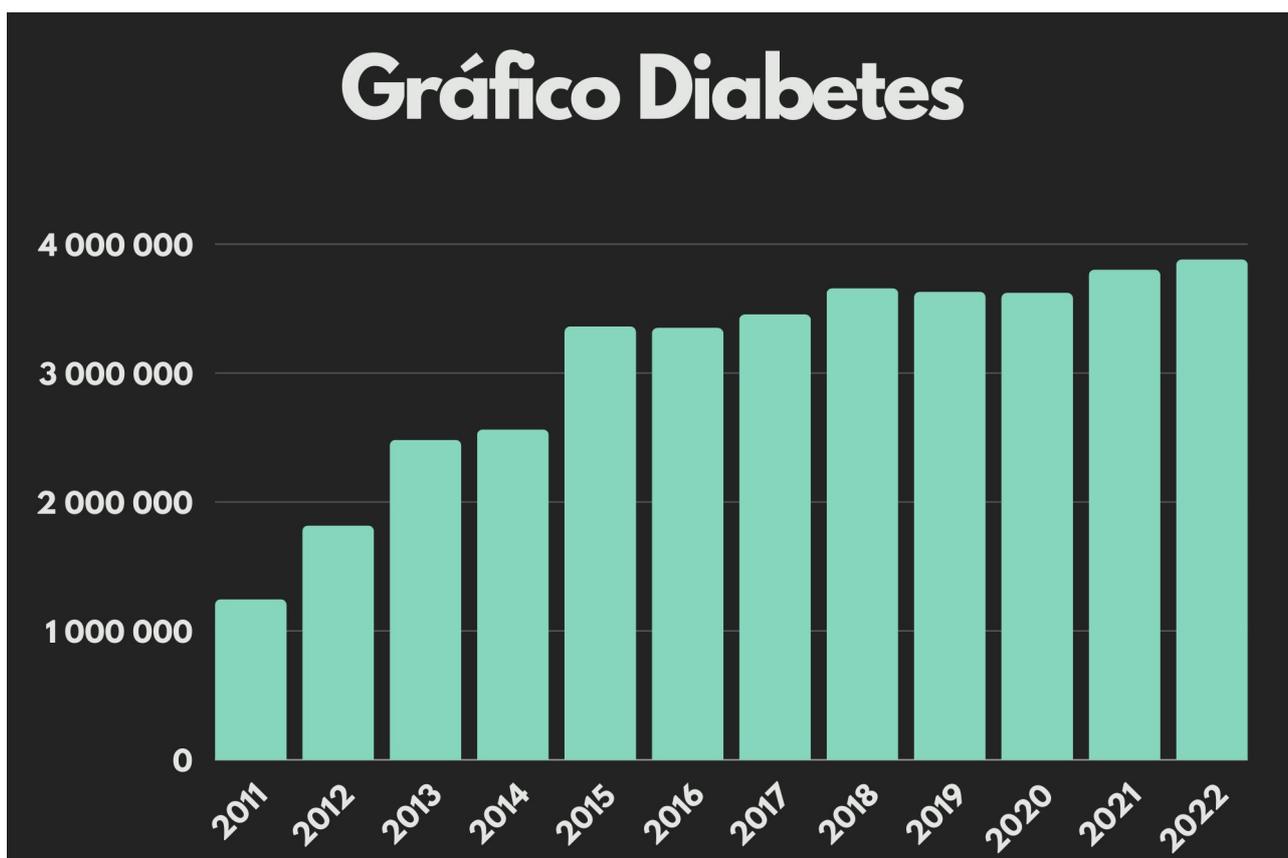


Ilustración 4: Gráfico de la evolución de casos de diabetes en España des del 2011 hasta el 2022, para todas las edades.^[10]

Como podemos observar en la Ilustración 1, la tendencia y evolución de la Diabetes en España es creciente, habiendo un drástico aumento en la cantidad de casos entre los años 2011 y 2015. En tan solo 11 años observamos un aumento de más de 2,5 millones de habitantes que padecen la enfermedad.

Dado que la aplicación que he desarrollado se centra en un público adolescente, que suele presentar menos constancia en el seguimiento de las pautas del tratamiento, he reducido y focalizado los datos a este rango de edades, comprendido entre los 5 y 24 años, sector más propenso a padecer Diabetes de Tipo 1.

Para ello apliqué los siguientes filtros en mi búsqueda:

- Problema de salud:
 - o Diabetes

- Edad:
- 05-09 años
 - 10-14 años
 - 15-19 años
 - 20-24 años

Una vez aplicados observamos los siguientes resultados:

	Casos	Casos por mil personas asignadas
2011	11.079	3,17
2012	18.211	3,65
2013	25.742	3,65
2014	26.884	3,80
2015	39.841	4,49
2016	39.956	4,46
2017	41.697	4,63
2018	41.493	4,59
2019	41.540	4,55
2020	42.270	4,61
2021	48.499	5,26
2022	49.920	5.35

Tabla 2: Número de casos Totales y Casos cada 1000 personas en España desde el 2011 hasta 2022, en un rango de edad entre 5 y 24 años. ^[10]

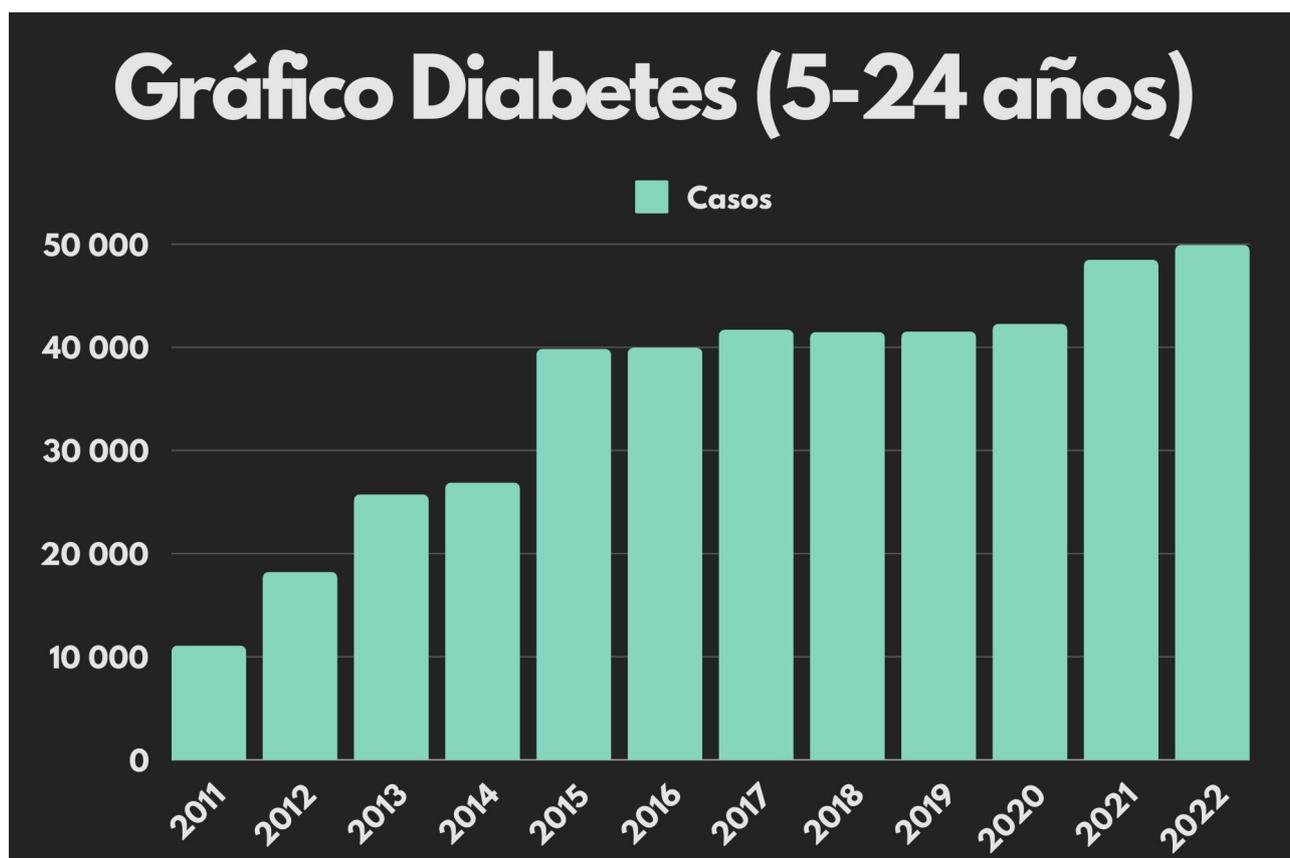


Ilustración 5: Gráfico de la evolución de casos de diabetes en España des del 2011 hasta el 2022, en un rango de edad entre 5 y 24 años.^[10]

En la gráfica y en el recuento de casos, se observa un aumento gradual en el número de casos para el rango de edad de 5 a 24 años durante el período de 2011 a 2022. En este período de 12 años, el número de casos ha experimentado un notable aumento, partiendo de 11,079 casos en 2011 y alcanzando los 49,920 en 2022, lo que representa un crecimiento de aproximadamente el 350% en un periodo de tiempo relativamente corto.

$$\text{Aumento \%} = \frac{\text{Nuevo Valor} - \text{Valor Inicial}}{\text{Valor Inicial}} \times 100$$

$$\text{Aumento \%} = (49.920 - 11.079) / 11.079 * 100 = \mathbf{350.38\%}$$

El ritmo general de crecimiento anual no tiene grandes variaciones, se ha mantenido constante, a excepción del salto entre 2011 y 2015, donde se observa que hubo un aumento de más de 28.000 niños y adolescentes diagnosticados.

Según las declaraciones del Pediatra del Servicio Aragonés de Salud, el Dr. Santiago Conde: ^[6]

“En los estudios recientes, la mayoría de comunidades autónomas encuentran ya una incidencia superior a 20 casos de diabetes tipo 1 en menores de 15 años por cada 100.000 habitantes cada año, y, probablemente, la media nacional sea ya superior a esa cifra. España está a la cabeza de los países del Sur de Europa en cuanto a incidencia de diabetes tipo 1 en edad pediátrica. Es por ello que es muy necesario seguir trabajando para mejorar el manejo de esta patología entre los más jóvenes y apostar por una educación terapéutica que les ayude a entender la diabetes y normalizarla. Una educación que es imprescindible también en sus entornos familiares y de amistad.”

“En la estadística se nota que las fases donde se generan más errores en el software son las fases de estudio, análisis y diseño, que al hacerse con vaguedad e imprecisiones, provoca los problemas ya enunciados.”

Según los resultados obtenidos y analizados de esta base de datos, y considerando el énfasis impulsado por la Sociedad Española de Diabetes (SED) ^[57], los datos muestran una tendencia preocupante con un aumento significativo de casos, llegando a casi 50.000 niños y adolescentes en 2022. Es alarmante que 5.35 niños de cada mil, cifra que va aumentando año tras año, hayan sido diagnosticados con esta enfermedad. Por ello, he tomado la decisión final de centrar la aplicación en la diabetes.

4 Diabetes

La Diabetes Mellitus (DM) es una enfermedad crónica metabólica que se caracteriza por elevados niveles de glucosa en sangre. La diabetes puede ser causada por los siguientes aspectos:

- El páncreas no produce suficiente insulina.
- Las células tienen problemas para identificar y utilizar correctamente la insulina.
- La persona tiene estilos de vida inadecuados.

La **glucosa** es un azúcar que proviene de los alimentos consumidos. Descompuesto el alimento, la glucosa se libera en al torrente sanguíneo. Gracias a la insulina, esta glucosa penetra en las células y actúa como fuente principal de energía del organismo.

La **insulina** es una hormona liberada por el páncreas, esencial para metabolizar la glucosa en sangre. Permite que la glucosa penetre en las células para que se utilice como fuente de energía.

Si el cuerpo es incapaz de consumir toda la glucosa, en vez de penetrar en las células, se acumulará en la sangre, provocando hiperglucemia (cantidad excesiva de glucosa en la sangre). Esta enfermedad requiere un estricto control metabólico para evitar y frenar posibles complicaciones a futuro.

Los tipos de diabetes más frecuentes son:

4.1 Diabetes Mellitus Tipo 1 (DM1)

Es una enfermedad autoinmune en la que el sistema inmunológico destruye las células beta del páncreas, responsables de la producción de insulina. Suele aparecer bruscamente en la infancia, adolescencia y juventud, sin embargo, también puede desarrollarse en la edad adulta. Si no se aplica el tratamiento adecuado de inmediato, se agrava rápidamente.

Existen varios factores, como la predisposición genética, algunos virus y una combinación de factores ambientales, que pueden causar este tipo de diabetes.

Es necesario tratarla con insulina desde el momento en que se diagnostica. De momento, no se ha encontrado la cura para este tipo de diabetes. El tratamiento se centra en controlar la glucosa en sangre usando insulina, seguir una dieta sana y equilibrada, y mantener un estilo de vida saludable para evitar su empeoramiento.

4.2 Diabetes Mellitus Tipo 2 (DM2)

La diabetes de tipo 2 es la más común. Suele manifestarse a partir de los 40 años, sin embargo, debido al aumento de la obesidad infantil en los últimos años, también se está diagnosticando en niños y adolescentes. Este tipo de diabetes se caracteriza porque el cuerpo es resistente a la insulina y la producción por parte del páncreas es insuficiente. Los factores de riesgo más influyentes en la DM2 son los siguientes:

- Factores hereditarios, especialmente familiares de primer grado de parentesco.
- Obesidad y sobrepeso.
- Sedentarismo.
- Hipertensión arterial.
- Alimentación poco saludable.
- Alteraciones de las grasas en la sangre.

Muchos de los que la padecen no son conscientes, ya que pasa desapercibida durante bastante tiempo, lo que dificulta el diagnóstico y el tratamiento precoz. Al inicio no es necesario tratarla con insulina, aunque en algunos casos más severos puede llegar a serlo. La DM2 cada vez es más frecuente, según la Organización Mundial de la Salud (OMS) está considerada como la epidemia del siglo XXI.

Organización	Tipo 1				Tipo 2
	Antes de las comidas	Después de las comidas	Al acostarse	HbA1c	HbA1c
NICE	70-126 mg/dl (4,0-7,0 mmol/l)	90-162 mg/dl (5,0-9,0 mmol/l)	70-126 mg/dl (4,0-7,0 mmol/l)	≤ 6,5 % (≤ 48 mmol/mol)	≤ 6,5 % (48 mmol/mol)
ISPAD	70-130 mg/dl (4,0-7,0 mmol/l)	90-180 mg/dl (5,0-10,0 mmol/l)	80-140 mg/dl (4,4-7,8 mmol/l)	< 7% (< 53 mmol/mol)	< 7,0 % (<47,5 mmol/mol)
ADA	90-130 mg/dl (5,0-7,2 mmol/l)	N/A	90-150 mg/dl (5,0-8,3 mmol/l)	< 7% (< 53 mmol/mol)	< 7 % (53 mmol/mol)

Tabla 3: Niveles de glucosa promedios según tipo de diabetes y organización. ^[46]

4.3 Tratamiento

El tratamiento se centra en tres aspectos esenciales que deben tenerse en cuenta para una correcta evolución:

- **Actividad física:** Evitar el sedentarismo es beneficioso para las personas con diabetes, ya que ayuda a mantener los niveles normales de glucosa en sangre y a tener un peso saludable. Se recomienda realizar como mínimo una hora de ejercicio físico al día, aunque sea de forma moderada.
- **Alimentación:** La alimentación de personas con diabetes debe ser equilibrada y variada. Se deben restringir los azúcares añadidos, priorizar alimentos ricos en fibra, distribuir los carbohidratos y limitar las grasas saturadas. Es importante respetar el número de comidas al día (entre 4 y 6) y sus horarios.
- **Medicación:** Para tratar la hiperglucemia, se utilizan dos tipos de medicación:
 - **Hipoglucemiantes no insulínicos:** Reducen la glucosa en sangre y se usan solo en la DM2 cuando la dieta y el ejercicio no son suficientes. Mayoritariamente se administran vía oral.

- **Insulina:** Administrada subcutáneamente, ya que por vía oral no es activa. Es indispensable para la diabetes de tipo 1 y puede llegar a ser necesaria en personas con diabetes de tipo 2.

Grupo de edad	Momento/tipo de prueba de glucosa en sangre		
	Antes de las comidas	Al acostarse	HbA1c
Bebés y niños en edad preescolar (0-6)	100-180 (mg/dL)	110-200 (mg/dL)	<8,5%
Niños en edad escolar (6-12)	90-180 (mg/dL)	100-180 (mg/dL)	<8%
Adolescentes y adultos jóvenes (13-19)	90-130 (mg/dL)	90-150 (mg/dL)	<7,5%

Tabla 4: Niveles de glucosa promedios según edad (niños y adolescentes). ^[46]

5 Interés de las tecnologías seleccionadas

He decidido aprender desde cero el lenguaje de programación de aplicaciones Kotlin y el *kit* de herramientas Jetpack Compose para desarrollar esta aplicación. Kotlin nos proporciona unas grandes ventajas frente a su gran rival Java. Jetpack Compose nos facilita la programación de interfaces de usuario y tiene un gran enfoque a futuro.

5.1 ¿Por qué utilizar Kotlin en vez de Java?

Kotlin es un lenguaje de programación de aplicaciones desarrollado principalmente por JetBrains.

Kotlin nos proporciona los siguientes beneficios:

- Kotlin se ha convertido en el lenguaje de desarrollo nativo para aplicaciones Android.
- Tiene un soporte de integración completo con el IDE Android Studio, ya que tanto Kotlin como Android Studio han sido desarrollados por JetBrains, por lo que la integración y la configuración del proyecto en Android Studio usando Kotlin es muy simple.
- Kotlin está en constante evolución.
- Nos da mayor sencillez al desarrollar aplicaciones frente a Java.
- Requiere menos código que los lenguajes convencionales, con menos código la probabilidad de *bugs* o errores se ve reducida y por lo tanto tendremos un código mucho más estable.
- Nos permite la permutabilidad con Java, en el mismo código podemos tener ambos lenguajes. Al compilar no generará errores.
- Constante crecimiento de la tendencia de uso en el sector laboral.

5.2 ¿Por qué utilizar el kit de herramientas Jetpack Compose?

Jetpack Compose es el *kit* de herramientas moderno de Android para compilar Interfaces de Usuario (UI) nativas. Simplifica y acelera el desarrollo de la UI en Android.

Jetpack Compose nos brinda las siguientes mejoras:

- **Menos código:** nos permite realizar más tareas con menos código, evitando así la cantidad de errores, dando lugar a un código más simple y fácil de mantener.
- **Intuitivo:** solo nos tenemos que encargar de describir la UI y Compose se encarga de todo el resto, a medida que cambiamos el estado de la aplicación o realizamos algún cambio la UI se actualiza de forma automática.
- **Acelera el desarrollo:** compatible con todo el código previamente creado, para ser añadido cuando y donde queramos. Itera de forma rápida con las vistas y nos genera una previsualización en vivo.
- **Muy potente:** nos permite crear aplicaciones usando las API de la plataforma Android. Es compatible con Material Design, tema oscuro, animaciones, etcétera.

Jetpack Compose es un *kit* de herramientas muy útil y enfocado al futuro, puesto que gracias a él podremos hacer uso y programar en el entorno de trabajo Compose Multiplatform. Este *framework* está en constante evolución y es muy innovador, ya que pretende utilizar un único código para compilar aplicaciones tanto en dispositivos Android como en dispositivos iOS, además de aplicaciones de escritorio y entornos web, todo esto con un mismo código base.

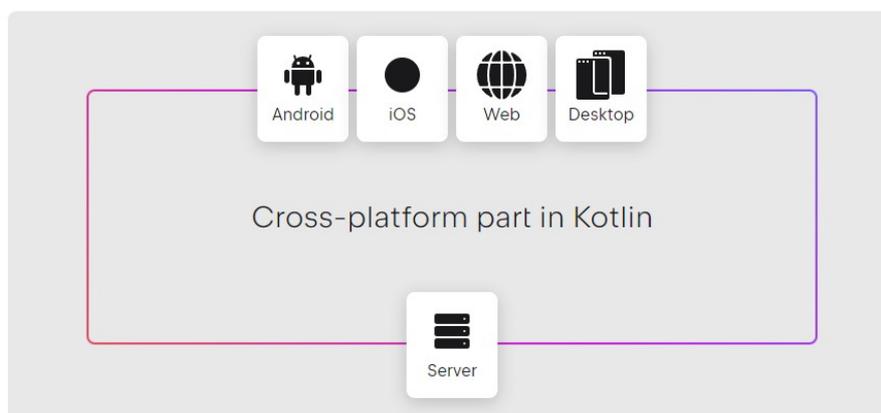


Ilustración 6: Esquema Compose Multiplatform. ^[54]

Jetpack Compose Multiplatform un muy relevante porque pretende afrontar uno de los mayores desafíos en el desarrollo de software, el cual es la fragmentación de plataformas.

La **fragmentación de plataformas** sucede cuando los programadores tienen que crear y mantener diversas versiones de una misma aplicación para todos los diferentes sistemas operativos y dispositivos. Esta problemática aumenta el tiempo, la complejidad y los

costos del desarrollo de software. *Frameworks* en desarrollo, como lo es Jetpack Compose Multiplatform, nos ayudarán a reducir esta fragmentación, permitiendo la reutilización de un único código para múltiples plataformas. El uso de este *framework* nos ahorrará tiempo y recursos, y nos facilitará el mantenimiento y la actualización de las aplicaciones.

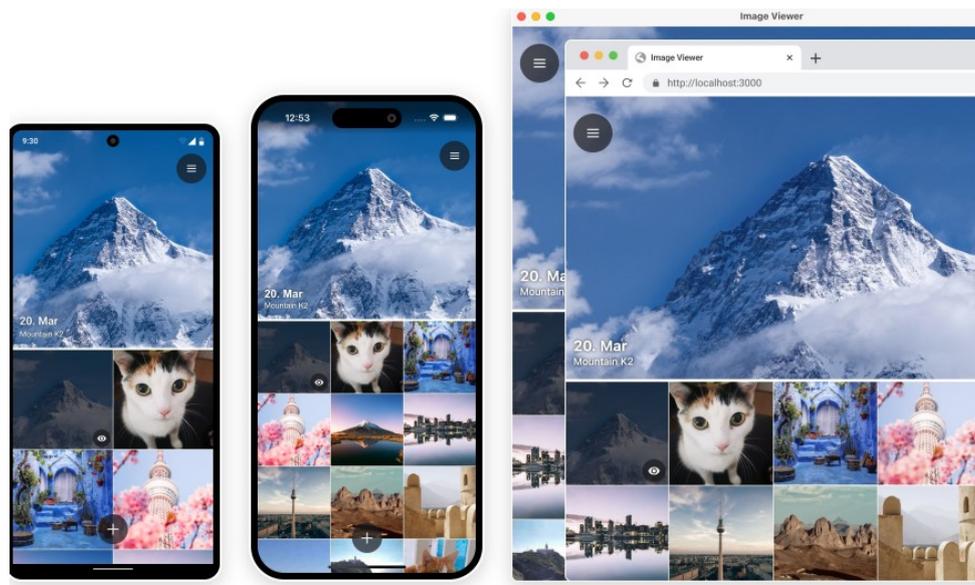


Ilustración 7: Ejemplo aplicación multiplataforma con Compose Multiplatform.^[51]

Para programar utilizando Compose Multiplatform, es imprescindible tener conocimientos sólidos en Jetpack Compose, ya que este *kit* de herramientas es la base sobre la cual se construye Compose Multiplatform.

La decisión de haber aprendido y utilizado Jetpack Compose para desarrollar esta aplicación, considero que ha sido una inversión a futuro muy interesante, no solo por el enfoque moderno y eficiente de diseño de UI, sino también por la oportunidad de poder descubrir, en un futuro, el desarrollo de aplicaciones multiplataforma de forma mucho más sencilla y rápida, al haber aprendido esta tecnología.

Una posible extensión de mi aplicación sería adaptar mi código (Kotlin + Jetpack Compose) al *framework* Compose Multiplatform. Esto no debería ser muy complicado, ya que el código base está programado con Jetpack Compose el cual es la base de

Compose Multiplatform, así pues, podría tener la misma aplicación disponible para múltiples plataformas.

5.3 ¿Por qué utilizar Material Design 3?

Material Design 3 (M3) es la evolución más reciente del sistema de diseño de Google, implementada en Jetpack Compose. Este sistema de diseño nos ofrece temas actualizados, componentes y características de personalización como el color dinámico. M3 es fundamental en la evolución del diseño digital, puesto que incluye nuevas pautas y elementos visuales para mejorar la experiencia del usuario en las plataformas que lo utilizan. Nos brinda mejoras significativas en términos de usabilidad, accesibilidad y estética. Ha sido diseñado para interfaces de usuario con Android 12 o superior.

Material Design 3 nos proporciona tipos de elementos:

Componentes: Los componentes se dividen según el tipo:

- Acciones: Los componentes de acción ayudan a alcanzar un objetivo.
- Comunicación: Los componentes de comunicación proporcionan información útil.
- Contención: Los componentes de contención contienen información y acciones, incluidos otros componentes como botones, menús o fichas.
- Navegación: Los componentes de navegación ayudan a los usuarios a desplazarse por la interfaz de usuario.
- Selección: Los componentes de selección permiten especificar opciones.
- Entradas de texto: Los componentes de entrada de texto permiten introducir y editar texto.

Estilos: Los estilos son los aspectos visuales de una interfaz de usuario que añaden un aspecto distintivo. Gracias a M3 se pueden personalizar cambiando su *Material Theme*. Encontramos diferentes aspectos sobre los estilos que nos proporciona Material 3:

- Color: El color se usa para expresar estilo y comunicar el significado. Con los colores dinámicos, Material 3 añade unas preferencias únicas de color. Añadiendo

el color principal, secundario, terciario y el color de fondo, genera automáticamente una paleta completa de colores. Gracias a esto podremos colorear cada componente y elemento según dónde se encuentre y la importancia que tenga.

- **Elevaciones:** La elevación es la distancia relativa entre dos superficies respecto al eje Z. Todas las superficies y componentes tienen un valor de elevación. Esto nos ayuda a poder resaltar elementos en las interfaces de usuario añadiéndoles más énfasis visual.
- **Movimiento:** Utiliza el movimiento para que la interfaz de usuario sea expresiva y fácil de usar. Las transiciones de Material ayudan a guiar a los usuarios mientras navegan por una aplicación.
- **Forma:** La escala de forma define el estilo de las esquinas de los contenedores, pudiendo utilizar una gama de redondeos que van desde el cuadrado al círculo.
- **Tipografía:** La escala tipográfica de Material incluye estilos contrastados y flexibles para dar soporte a una amplia gama de casos de uso.

Su fecha de lanzamiento fue a mediados de 2021, como podemos observar, se trata de una herramienta muy reciente la cual ha estado en constante evolución y actualización desde su lanzamiento, hasta el punto en que su última versión fue lanzada el 14 de mayo de 2024.

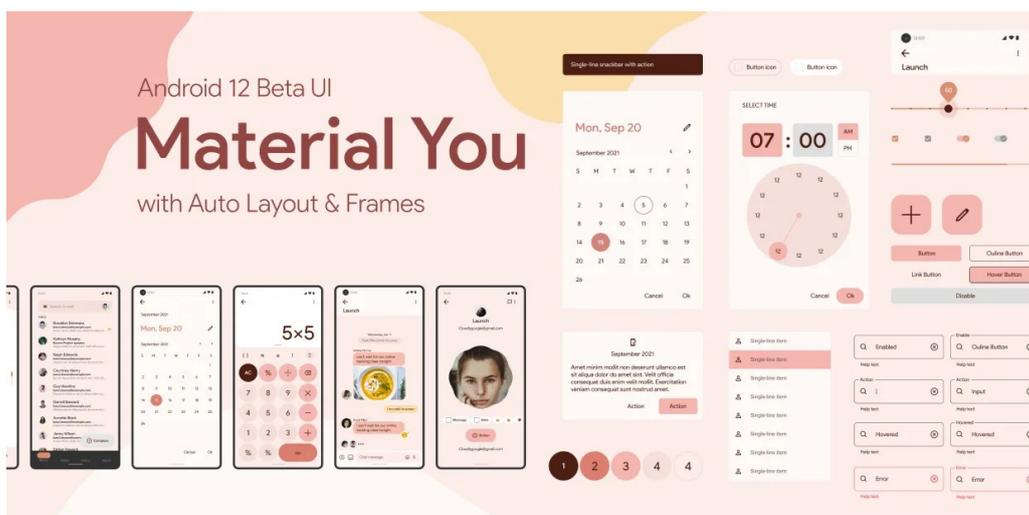


Ilustración 8: Elementos y componentes Material Design 3. [52]

6 Aprendizaje del lenguaje y *kit* herramientas

Dedicué aproximadamente la mitad de la duración total de mi Trabajo de Fin de Grado a aprender el lenguaje Kotlin, el *kit* de herramientas Jetpack Compose y el sistema de diseño Material 3. Este proceso de aprendizaje fue importante, ya que nunca había utilizado estas tecnologías, y porque representan un gran avance en el desarrollo de software con un claro enfoque hacia el futuro.

Al empezar este proyecto, no tenía conocimientos previos de ninguna de estas tres tecnologías, ya que no formaban parte del plan de estudios de la carrera. Investigué y me informé sobre la importancia y el potencial de estas tecnologías y finalmente decidí realizar el aprendizaje de forma autónoma.

El proceso de esta etapa de aprendizaje considero que fue una experiencia bastante intensa que finalmente, terminada la formación, pasó a ser muy gratificante, ya que cuando empecé a desarrollar la aplicación vi que el tiempo dedicado al aprendizaje fue muy productivo, debido a que el progreso avanzaba más rápido de lo esperado.

Primero, tuve que formarme en el lenguaje de programación de aplicaciones Kotlin, tanto para el desarrollo de la lógica como para el desarrollo de interfaces de usuario, utilizando *layouts* y *screens* en XML. Consideré que era esencial aprender todos los aspectos sobre Kotlin para poder programar correctamente mi aplicación, incluso aprender el desarrollo de pantallas en XML, aunque en el futuro no fuese a utilizarlo porque las interfaces de usuario las desarrollaría con el *kit* de herramientas Jetpack Compose. Sin embargo, me fue muy útil porque algunos aspectos no podían ser programados con Jetpack Compose, como por ejemplo, las notificaciones personalizadas, las cuales tuve que programar con *layouts* en XML.

Finalmente, una vez aprendí por completo el lenguaje Kotlin, me dispuse a aprender sobre el desarrollo de interfaces de usuario con el *kit* de herramientas Jetpack Compose para facilitar y agilizar todo el desarrollo de la aplicación.

Para adquirir unos conocimientos sólidos y profundos utilicé una gran variedad de recursos educativos:

- **Webs oficiales:** Primero de todo tenía que aprender los aspectos principales de estas tecnologías y la documentación oficial de Kotlin y Jetpack Compose fueron mi principal fuente de información. Estas webs me proporcionaron los conocimientos básicos y las capacidades de cada tecnología. Gracias a esta documentación oficial pude conocer las mejores prácticas desde las fuentes más confiable.
- **Cursos online:** Después de obtener los conocimientos básicos del lenguaje y *kit* de herramientas, busqué diversos cursos de nivel principiante y avanzado que fuesen fiables y estuviesen bien estructurados, para adquirir un mayor dominio del lenguaje. Para ello realicé varios cursos *online*:
 - **Cursos y lecciones oficiales:** Tanto la plataforma oficial de Kotlin como de Jetpack Compose proporcionan tutoriales y cursos formativos para empezar a formarse.
 - **Curso para principiantes de Kotlin:** Encontré un curso formativo en línea muy útil, con explicaciones claras y sencillas y explicado de forma natural, facilitando la asimilación de los nuevos conocimientos. Curso impartido por el desarrollador senior AristiDevs. Este programador publica contenido en diversas plataformas y redes sociales y su curso de “Android desde cero con Kotlin para principiantes” de una duración aproximada de 15 horas fue de lo más útil a la hora de aprender el lenguaje de programación.
 - **Curso nivel intermedio de Kotlin:** Una vez asimilados todos los aspectos básicos sobre Kotlin, pasé a adquirir conocimientos más sólidos con el curso de nivel Intermedio del mismo desarrollador. Este curso dura aproximadamente 9 horas y se centra en programar una aplicación funcional. Fue muy útil, ya que impulsó mis habilidades prácticas programando una aplicación real. Gracias a esto pude conseguir el manejo y la destreza de programar con Kotlin.
 - **Cursos de nivel básicos e intermedios de Jetpack Compose:** Para este *kit* de herramientas encontré menos recursos, ya que se trata de una tecnología bastante reciente, pero pude utilizar diversos cursos y vídeos para aprender.

La estrategia que considero más útil y que he utilizado para adquirir los conocimientos tanto de Kotlin como de Jetpack Compose, ha sido intentar programar de forma autónoma ejercicios y aplicaciones propuestos, y una vez probado ver las explicaciones de los cursos para corregir aquellos aspectos incorrectos o mejorables. Finalmente, una vez tenía todos los conocimientos, antes de iniciar el proceso de desarrollo de mi propia aplicación, repliqué las interfaces de usuario de algunas aplicaciones reales y famosas del mercado, así obtuve una mayor fluidez.

Aunque algunos de los cursos o fuentes de información estuviesen enfocado desde un punto de vista muy introductorio, los sólidos conocimientos de programación adquiridos durante la carrera facilitaron por completo el proceso de autoaprendizaje de estas tecnologías de programación, ya que la mayor parte de recursos utilizados requieren un nivel elevado de conocimientos sobre la programación y el sector de la informática.

Además, mientras aprendía, también tuve la oportunidad de conocer Material Design 3, la última versión de las especificaciones de diseño de Google. Aprender y aplicar Material 3 fue un aspecto clave para que la interfaz de usuario de mi aplicación no solo fuera funcional, sino también moderna y atractiva, adaptándose a las directrices de diseño más recientes. Aprendí Material 3 con toda la documentación oficial de su web y las guías de diseño de *Developers Android*.

7 Métodos de *gamificación* implementados

La ludificación o *gamificación* se trata de utilizar dinámicas, elementos y mecánicas de los juegos en ámbitos no lúdicos, como la educación, la salud, el trabajo o el marketing, con el objetivo de mejorar el rendimiento, la participación, el compromiso de las personas, la constancia y la motivación.

Además de añadir diversión a las actividades cotidianas, ofrece beneficios significativos en diferentes áreas. En el ámbito sanitario, por ejemplo, la *gamificación* puede ayudar a obtener una mejor adaptación y adherencia a los tratamientos médicos por parte de los pacientes y aumentar el compromiso, haciendo que el proceso sea más interactivo y entretenido.

La ludificación introduce estructuras innovadoras y creativas propias de los juegos para convertir una actividad que a priori es aburrida, en una actividad que motive e incite a las personas a realizarla. Pretende persuadir al usuario para convertir una tarea pesada en un reto atractivo que merezca la pena continuar.

Para realizar el diseño de esta aplicación utilicé el marco MDA (*Mechanics-Dynamics-Aesthetics*) siglas que hacen referencia a los tres elementos básicos de todos los juegos: las mecánicas, las dinámicas y las estéticas o elementos implementados de la siguiente forma en la aplicación.



Ilustración 9: Marco MDA sobre la Gamificación. [53]

7.1 Elementos

- **Gemas:** En la aplicación los usuarios acumulan gemas por registrar los parámetros de la diabetes, como registrar los niveles de glucosa, medicación e insulina administrada, alimentación y actividad física realizada. Las gemas actúan como moneda de cambio para poder comprar escudos de inmunidad. 
- **Racha:** La racha es un contador de los días consecutivos en los que el usuario registra sus parámetros de la enfermedad. Mantener una gran racha motiva a los usuarios a ser constantes con el seguimiento de su tratamiento. Cada día que se realiza un registro esta racha aumenta, fomentando así la rutina y creando un hábito. 
- **Escudos de inmunidad:** Los escudos de inmunidad se pueden comprar con gemas. Su principal función es mantener la racha del usuario en caso de que un día olvide registrar sus aspectos en la aplicación. Esto proporciona flexibilidad y reduce la ansiedad por perder la racha, ayudando así a mantener la motivación y el compromiso a largo plazo. El precio de estos escudos será elevado para que antes de tener la posibilidad de adquirirlos, los usuarios ya hayan desarrollado el hábito. 
 - Según el filósofo y psicólogo William James:

“Para crear o romper un hábito, uno tiene que realizarlo durante 21 días. Pasado ese tiempo, la acción repetida quedará interiorizada, funcionando de manera automática”.
 - Por este motivo, el precio a pagar para adquirir un escudo de inmunidad es de 2000 gemas. Si realizan cada día un registro podrán obtener 50 gemas diarias. $50 \cdot 21 = 1050$ gemas. Si consiguen una racha de 10 días consecutivos obtendrán la Maestría Principiante = 100 gemas y con una racha de 20 días consecutivos obtendrán la Maestría Intermedio = 200 gemas, por lo tanto, en 21 días consecutivos el usuario habrá obtenido 1350 gemas. He añadido este margen de 650 gemas por si los usuarios los primeros días no son constantes,

de esta forma dificultamos la obtención de escudos, evitando que los usuarios se acomoden a la inmunidad.

- **Niveles de maestría:** El nivel de maestría refleja el compromiso del usuario con el tratamiento, estos niveles recompensarán con gemas la máxima racha consecutiva conseguida. A medida que los usuarios alcanzan rachas más largas, suben de nivel, lo que es recompensado con una cantidad de gemas ascendente según el nivel. Los usuarios obtendrán una insignia de maestría que se mostrará en el apartado de perfil. Estos son los niveles de maestría de Easulin:

- **Principiante:** Este nivel lo conseguirán aquellos usuarios que logren mantener una racha de 10 días consecutivos. Será recompensando con 100 gemas, y el usuario obtendrá la siguiente insignia.



- **Intermedio:** Este nivel lo conseguirán aquellos usuarios que logren mantener una racha de 20 días consecutivos. Será recompensando con 200 gemas, y el usuario obtendrá la siguiente insignia.



- **Avanzado:** Este nivel lo conseguirán aquellos usuarios que logren mantener una racha de 50 días consecutivos. Será recompensando con 500 gemas, y el usuario obtendrá la siguiente insignia.



- **Experto:** Este nivel lo conseguirán aquellos usuarios que logren mantener una racha de 100 días consecutivos. Será recompensando con 1000 gemas, y el usuario obtendrá la siguiente insignia.



- **Supremo:** Este nivel lo conseguirán aquellos usuarios que logren mantener una racha de 150 días consecutivos. Será recompensando con 1500 gemas, y el usuario obtendrá la siguiente insignia.



7.2 Dinámicas

- **Progresión:** Los usuarios a través de todos los elementos descritos anteriormente sienten que están progresando, ya sea por tener una racha más larga, o por haber conseguido una determinada insignia. Manteniendo la motivación y haciéndoles completamente partícipes y autónomos del proceso.

- **Competencia:** Los usuarios compiten amistosamente para ver quién puede mantener la racha más larga o conseguir el máximo nivel de maestría. Esta competencia sana hace que los usuarios mantengan la motivación para continuar con sus registros diarios.
- **Recompensa:** La obtención de gemas por realizar un registro diario y la satisfacción de alcanzar niveles de maestría superiores refuerzan el comportamiento positivo e incitan a seguir realizando los registros a diarios.
- **Retroalimentación:** La aplicación proporciona retroalimentación continua sobre el progreso del usuario. En la pantalla de Inicio los usuarios pueden ver toda la actividad del día actual, mostrando qué parámetros aún no han sido registrados de forma clara y atractiva. También en el historial de registros de cada sección se muestran todos parámetros. Por otro lado, gracias a las alertas y recordatorios programados a diario, ayudan al usuario a recodar y evitar perder la racha actual.

7.3 Mecánicas

- **Recolectar:** Los usuarios podrán recolectar y obtener gemas por realizar registros diarios y por alcanzar niveles de maestría superiores. Las gemas podrán acumularse para comprar escudos de inmunidad, lo que añade un punto de estrategia al decidir cuándo y cómo gastar sus gemas, ya que independientemente de la racha actual, si un día no se realiza registro, el escudo será consumido sin tener la opción de decidir si gastarlo o no. Esto se ha hecho para que los usuarios comprendan que lo más importante es mantener la constancia y registrar los parámetros de la diabetes a diario. De esta forma, conseguimos que tenga la misma importancia olvidar un día, independientemente de si la racha actual es de 1 o de 20 días. Si diéramos la opción de decidir cuándo quieren usar la inmunidad, todos elegirían hacerlo cuando tienen una mayor racha.
- **Competir:** Los usuarios competirán entre sí para mantener la racha más larga o alcanzar el nivel de maestría más alto, esto incentiva la adherencia al tratamiento.
- **Explorar:** Al hacer partícipes a los niños y adolescentes de su tratamiento y enfermedad, incentivamos a que exploren sobre consejos nutricionales, hábitos

saludables y recomendaciones. Esto fomenta un aprendizaje continuo y una mejor gestión de su vida con la diabetes.

- **Resolver:** Registrar a diario sus parámetros y mantener una racha elevada ayuda a que los usuarios adquieran y sigan ciertas rutinas, promoviendo un comportamiento proactivo y disciplinado en los niños y adolescentes, que, por lo general, no tienen constancia ni interés en su enfermedad.

8 Análisis de Funcionalidades y Requerimientos

Pasadas las semanas en las que me dediqué exclusivamente al aprendizaje del lenguaje y las herramientas tecnológicas utilizadas para programar mi aplicación, decidí realizar unos pasos previos a la implementación del código, los cuales me ayudaron a saber los requisitos y funcionalidades más importantes que quería implementar, para poder conseguir un diseño inicial de referencia con su arquitectura física.

Antes de realizar la fase de desarrollo, un paso muy importante en la Ingeniería del *Software* es realizar un correcto análisis de requerimientos o requisitos.

Un **análisis de requisitos** busca reconocer las necesidades de un problema o de una situación de partida y cómo el *software* creado las solucionará.

Esta etapa es muy importante realizarla antes del desarrollo de *software*, ya que uno de los principales motivos de proyectos que fracasan, vienen dados por no realizar un previo análisis de requerimientos.

Tal y como nos explica Jesús Zavala Ruiz en su estudio: “¿Por Qué Fracasan los Proyectos de Software?; Un Enfoque Organizacional ” ^[43]

“En la estadística se nota que las fases donde se generan más errores en el software son las fases de estudio, análisis y diseño, que al hacerse con vaguedad e imprecisiones, provoca los problemas ya enunciados.”

Fase	%
Estudio y análisis	56%
Diseño	10%
Código	7%
Otros	27%

Tabla 5: Porcentaje del Origen de los Errores en los proyectos fracasados de Software. ^[43]

8.1 Necesidades

Para detectar las necesidades existen unas preguntas clásicas:

¿Que pretende hacer este software?

- Promover la constancia en el tratamiento en niños y adolescentes con diabetes.

¿Quién utilizará esta aplicación?

- Niños y adolescentes, incluso adultos, con diabetes.

¿Cuándo y dónde se podrá utilizar esta aplicación?

- En cualquier lugar y momento, es una aplicación móvil con gran portabilidad y se adapta a cualquier situación.

¿Qué problemática hay actualmente?

- Niños y adolescentes que padecen cualquier enfermedad que requiera constancia en el tratamiento, en este caso, la diabetes, tienen grandes dificultades para mantener esa perseverancia. Por esta razón, es necesario que los padres y familiares estén atentos y les proporcionen un seguimiento cercano.

¿Cómo solucionar esta problemática?

- Creando una aplicación que pueda ser instalada en cualquier dispositivo, usando metodologías lúdicas y atractivas para que nos ayuden a conseguir esta constancia.

Una vez respondidas estas preguntas, he podido extraer las siguientes necesidades centradas en los diversos agentes a los que ayudaría esta aplicación directa e indirectamente:

8.1.1 Niños y adolescentes con Diabetes

- **Interfaz simple y amigable:** la aplicación debe tener una interfaz atractiva, intuitiva y fácil de usar, adaptada al principal rango de edad de los usuarios de la aplicación.
- **Motivación y diversión:** para conseguir el objetivo principal, adquirir el hábito de seguir el tratamiento, hay que ofrecer diversos elementos de *gamificación* para

hacer del tratamiento una experiencia más divertida y motivadora, ayudando a mantener el compromiso a largo plazo.

8.1.2 *Padres de niños con Diabetes*

- **Fomentar la autonomía:** la necesidad de los padres es intentar que sus hijos desarrollen el hábito y la responsabilidad sobre su tratamiento, así pues, la supervisión constante de los padres se verá reducida.

8.1.3 *Personal sanitario*

- **Registro y seguimiento de pacientes:** permitir a sanitarios tener un historial de tratamiento de los pacientes, incluyendo medicamentos, nivel de glucosa, registros de actividad y alimentación. En resumen, tener un registro completo de los parámetros importantes a considerar en el seguimiento de la diabetes.
- **Facilitar la comunicación:** al generar este interés en niños y adolescentes sobre la diabetes y su seguimiento, obtendremos una mejora en la comunicación entre paciente y personal sanitario, mostrando sus logros, objetivos y registros con entusiasmo.

8.2 *Requisitos funcionales*

Gracias a las necesidades anteriores pude extraer los siguientes requisitos funcionales. He dividido estos requisitos en diferentes secciones, explicados posteriormente en el apartado 11.3 Mockups.

1. Registro de nuevos usuarios

- Usuarios nuevos podrán registrarse en la aplicación realizando un pequeño formulario.

2. Inicio de sesión de usuarios existentes

- Acceso a la cuenta utilizando correo electrónico y contraseña.

3. Apartado de Inicio

- Dar una visión general del progreso del usuario con su racha actual de registros.

- Resumen de registros del día:
 - Mostrar un resumen de los parámetros registrados en la aplicación durante el día actual.

4. Apartado de Registro

- *Registro de Alimentación:*
 - Permite a los usuarios registrar sus comidas realizadas durante el día incluyendo el tipo de alimento y la franja en la que ha realizado esa ingesta.
 - Ofrecer la opción de seleccionar diferentes tipos de alimentos obtenidos de una base de datos.
- *Registro de Niveles de Azúcar:*
 - Permitir que el usuario registre su nivel de glucosa en sangre.
- *Registro de Medicación y/o Inyecciones:*
 - Permite al usuario registrar los medicamentos e inyecciones administradas durante el día, incluyendo la dosis y nombre del medicamento.
- *Registro de Actividad Física:*
 - Permite al usuario registrar la actividad física realizada durante el día agregando el tipo de ejercicio y su duración.

5. Apartado de Recompensas

- Obtención de premios y recompensas diarias para promover la racha de registros.
- Objetivos recompensados por mantener rachas de registros diarios durante largos períodos de tiempo.

6. Apartado de Perfil

- Mostrar información personal del usuario.

8.3 Historias de Usuario

Las historias de usuario son pequeños relatos que nos muestran las necesidades y los requisitos de los usuarios al utilizar una determinada aplicación. Nos ayudan a entender que funcionalidades son importantes para los usuarios. En este apartado podremos ver las historias de usuario para las funcionalidades descritas anteriormente.

Como...	Quiero...	Para...
Usuario Nuevo	Poder registrarme en la aplicación	Acceder a todas las funcionalidades disponibles
Usuario	Iniciar sesión utilizando mi correo electrónico y contraseña	Acceder a mi información personal y registros
Usuario	Ver una vista general de mi progreso en la aplicación	Tener una idea rápida de mi salud y constancia en el tratamiento
Usuario	Revisar un resumen de los registros realizados durante el día	Tener un resumen del día actual
Usuario	Registrar mi alimentación diaria	Monitorear mi dieta y nutrición
Usuario	Registrar mis niveles de glucosa en sangre diarios	Tener un registro preciso que pueda mostrar a sanitarios
Usuario	Registrar la medicación e inyecciones que tomo a diario	Seguir el programa de medicación prescrito
Usuario	Registrar la actividad física que realizo durante el día	Tener un histórico de mi actividad física
Usuario	Recibir premios y recompensas diarias por mantener una racha	Sentirme motivado y mantener la constancia
Usuario	Tener objetivos por mantener rachas de registros	Mantener hábitos saludables a largo plazo

Usuario	Ver mi información personal	Tener un resumen de mi perfil
Usuario	Recibir notificaciones diarias	Acordarme de añadir los registros del día

Tabla 6: Historias de usuario

8.4 Casos de Uso

Un caso de uso es una descripción gráfica de cómo los actores interactúan con una aplicación para lograr un objetivo específico. Es una herramienta clave en el análisis y diseño de sistemas de software, especialmente dentro del marco de desarrollo orientado a objetos. Nos ayudan a asegurar que las funcionalidades estén correctamente implementadas.



Ilustración 10: Casos de uso de las funcionalidades de Easulin

9 Diseño visual

Una vez extraído el listado de funcionalidades de la aplicación, realicé el diseño visual.

El diseño visual de una aplicación tiene como objetivo organizar la navegación, definir los elementos visuales y las diferentes ventanas. Con esto, nos aseguramos de que la información esté bien estructurada y verificamos que la navegación y las funcionalidades de la aplicación fluyan correctamente, garantizando así una mejor Experiencia de Usuario (UX).

En el diseño de aplicaciones también hay que tener en cuenta el aspecto visual y las sensaciones que transmite esta de cara a los usuarios, asegurando que sea atractiva y tenga una identidad propia.

A medida que el mercado de aplicaciones aumenta, las expectativas de los usuarios son cada vez más estrictas. Para que la UX sea óptima y los usuarios no se sientan perdidos, he decidido utilizar aspectos visuales fáciles de comprender y de simple navegabilidad para que tengan la sensación de control total en sus manos.

He diseñado una interfaz de usuario (UI) que, adaptada con la UX, permite a los usuarios sentirse cómodos en todo momento y que, a su vez, ofrezca una experiencia fluida y sin complicaciones. He intentado buscar el equilibrio entre estética y funcionalidad. Los elementos que he escogido para esta interfaz son muy simples de utilizar y comprensibles, como por ejemplo: Progress Bar, Bottom Navigation Bar, Side Navigation Menu... Elementos que facilitan el uso y permiten al usuario saber dónde está en cada momento.

Todos los iconos de la aplicación han sido modificados usando la paleta de colores para conseguir una mejor adaptación visual, generando así una propia identidad de marca.



Ilustración 11: Iconos de la aplicación adaptados a la paleta de colores

9.1 Paleta de colores

La elección de una paleta de colores adecuada a una aplicación es un factor muy importante, ya que marcará la identidad de la aplicación y será la imagen que recibirá el usuario cada vez que la inicie.

En este caso, he escogido una paleta visualmente atractiva y amigable para los usuarios a los que se enfoca mi aplicación.

Modo oscuro: el fondo predeterminado de la aplicación es un gris oscuro, ya que reduce la fatiga visual y combinado con colores llamativos proporciona un contraste adecuado. Además, el modo oscuro actualmente se ha popularizado mucho entre los usuarios, mayoritariamente en usuarios jóvenes, lo cual genera una mayor aceptación.



Ilustración 12: Color de fondo en hexadecimal

Color principal: he escogido un color turquesa brillante, este resaltará sobre el fondo oscuro, además el color turquesa está especialmente relacionado con entornos de la salud pudiendo transmitir una sensación de tranquilidad y bienestar.



Ilustración 13: Color de primario en hexadecimal

Color secundario: he escogido el azul claro que contrasta con el turquesa. Este color está también relacionado con entornos de la salud.



Ilustración 14: Color de secundario en hexadecimal

La elección de colores ha sido un proceso fundamental, ya que quería que la aplicación visualmente fuera atractiva para niños y adolescentes, dando la impresión de que están en una especie de juego, por ello la combinación de fondo oscuro con colores llamativos al frente.

9.2 Nombre y Logotipo

El nombre de la aplicación es "**Easulin**". Este nombre surge de la combinación de dos palabras claves relacionadas con el objetivo principal de la aplicación y con la diabetes.

- La primera parte, "*Easy*", representa el foco de la aplicación: hacer que el proceso de tratamiento de la diabetes sea más sencillo para los niños y adolescentes que la padecen.
- La segunda parte, "*Insulin*" (insulina), representa a la hormona segregada por el páncreas para ayudar a mover la glucosa en la sangre, hormona fundamental en enfermos con diabetes.

Al fusionar estos dos elementos he creado el nombre de mi aplicación "**Easulin**" el cual es simple, corto y atractivo.

Finalmente el logotipo creado contiene tres elementos gráficos:

- En el fondo podemos observar un parche adhesivo sensor para diabéticos.
- En la parte derecha un medidor de glucosa en sangre.
- En la parte izquierda una inyección de insulina.

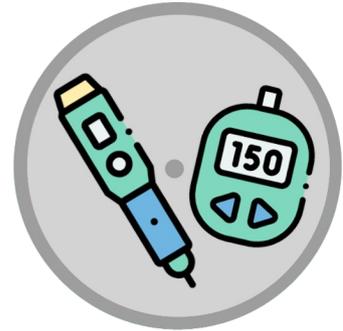
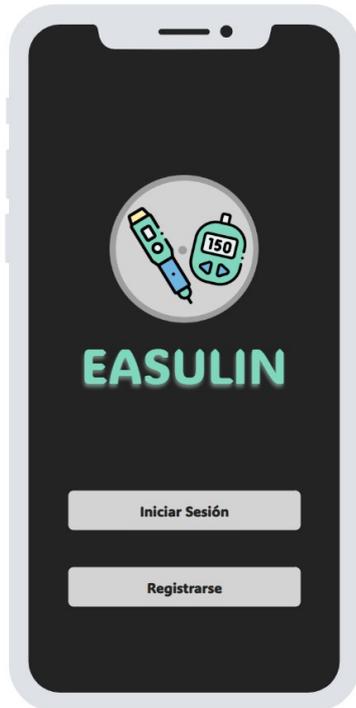


Ilustración 15: Logotipo Aplicación

9.3 Mockups

Para diseñar los mockups de mi aplicación utilicé Moqups, una aplicación web simplificada e intuitiva para crear esquemas, maquetas, diagramas y prototipos.



Mockup 1: Pantalla no Autenticado

Pantalla Inicial

En esta pantalla inicial, podemos observar 2 botones, uno para iniciar sesión, en caso de no tener una cuenta existente, podrán crear una nueva pulsando en el botón Registrarse.



Mockup 2: Pantalla de Inicio Sesión

Pantalla Iniciar Sesión

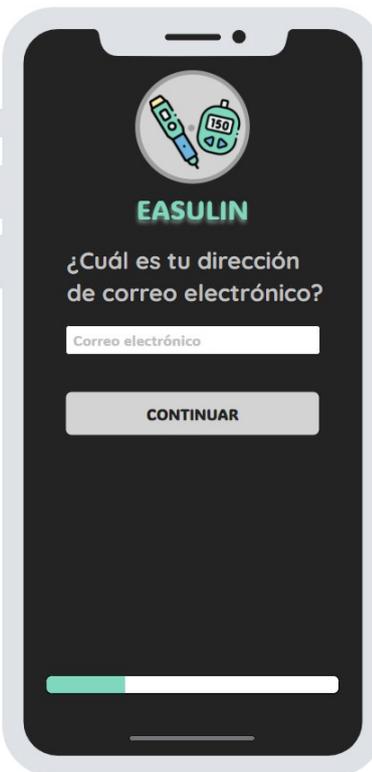
Una vez pulsado el botón de Iniciar Sesión, nos aparecerá el siguiente formulario simple donde el usuario introducirá los datos de su cuenta, correo electrónico y contraseña en los campos de entrada de texto, para acceder al contenido de la aplicación.



Mockup 3: Pantalla Registro 1

Pantalla Formulario Registro 1

En caso de no estar registrado, el método más simple y atractivo es realizar formulario pregunta por pregunta de forma breve y concisa, evitando colapsar a los usuarios con mucha información en una única ventana, por eso cada pregunta aparecerá en una vista distinta. En la parte inferior podemos observar una barra de progreso indicando cuanto nos queda para acabar el formulario. En esta primera vista introduciremos nuestro nombre. Para pasar a la siguiente pregunta del formulario pulsaremos en continuar.



Mockup 4: Pantalla Registro 2

Pantalla Formulario Registro 2

En la segunda pregunta del formulario de registro, el usuario introducirá su correo electrónico, realizaremos una verificación de que no exista ninguna cuenta con este correo introducido, así podremos seguir con el formulario.



Mockup 5: Pantalla Registro 3

Pantalla Formulario Registro 3

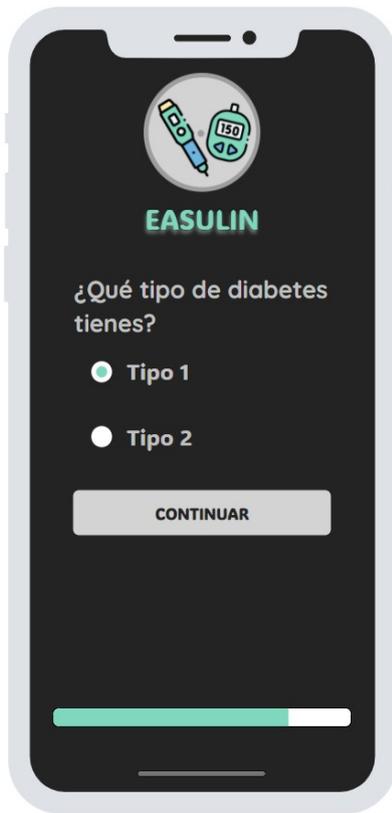
En la tercera pregunta del formulario de registro, una vez verificado que el usuario que quiere registrarse no existe en nuestro sistema, se procederá a introducir la contraseña deseada de su cuenta. Pulsando a continuar podremos seguir con el formulario.



Mockup 6: Pantalla Registro 4

Pantalla Formulario Registro 4

En esta cuarta pregunta del formulario, le preguntaremos la edad al usuario, así en un futuro podremos analizar datos de esta aplicación observando que rango de edad usa más la aplicación, que edad es la que más se implica en registrar los parámetros de la enfermedad, que rango de edad fracasa más al conseguir el hábito, etcétera.



Mockup 7: Pantalla Registro 5

Pantalla Formulario Registro 5

En la quinta pregunta del formulario pretendemos saber qué tipo de diabetes tiene el usuario, aunque la más común en niños y adolescentes sea la diabetes de tipo 1, siempre existen casos con diabetes de tipo 2. Esta pregunta la hacemos para tener ver el porcentaje de usuarios de la aplicación con un tipo u otro, y poder brindarles una información u otra.



Mockup 8: Pantalla Registro 6

Pantalla Formulario Registro 6

Como sexta y última pregunta del formulario, preguntamos a los usuarios si su tratamiento incluye inyecciones de insulina. Con esta respuesta podremos mostrar información adaptada a cada caso. Una vez finalizado el formulario la barra de progreso aparecerá completa y podremos pulsar el botón de finalizar registro.



Mockup 9: Racha Actual

Diseño de racha actual

Una vez iniciada la sesión podremos visualizar nuestra racha actual. Este método de *gamificación* motiva a los usuarios a mantener su racha y a aumentarla día a día, convirtiéndolo en un desafío personal continuo que refuerza el compromiso con el tratamiento. El día que no realicemos ningún registro en nuestra aplicación, perderemos automáticamente la racha actual pasando a 0.

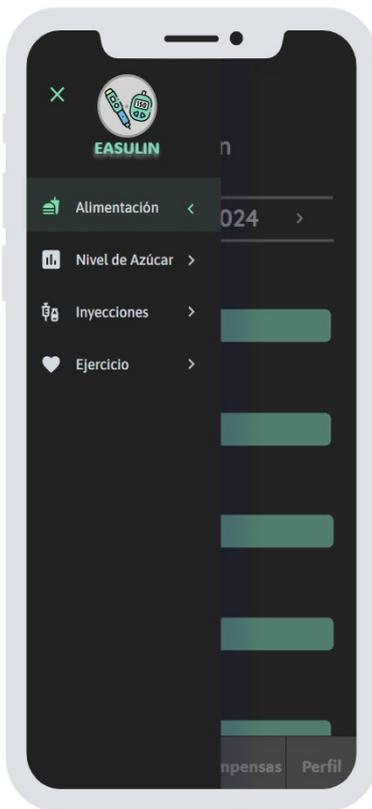


Mockup 10: Pantalla Inicio (Autenticado)

Pantalla de Inicio

En la parte superior de esta pantalla podemos observar la racha actual con el icono de la llama, y las gemas obtenidas por el usuario. Estas gemas nos servirán para comprar inmunidad. La inmunidad sirve para no perder la racha, en caso de olvidar realizar un registro en la aplicación.

Esta pantalla de inicio nos muestra el resumen diario de los últimos parámetros registrados en la aplicación.



Pantalla de Registrar

Como podemos observar en la pantalla de registros tenemos un menú lateral donde podemos acceder a 4 parámetros a registrar en nuestra aplicación.

- Alimentación
- Nivel de Glucosa en sangre
- Inyecciones y Medicamentos
- Ejercicio

Decidí registrar estos 4 aspectos, ya que considero que son los más importantes para tener en cuenta en el seguimiento de un paciente con diabetes.

Mockup 11: Registros SideBar



Pantalla de Registrar - Alimentación

En esta pantalla podremos registrar la alimentación de los usuarios, añadiendo las ingestas de comida diarias. Como podemos observar hay 5 apartados cada uno correspondiente a las comidas diarias. Además de estos 5, he añadido uno más llamado Extras por si el paciente realizar alguna comida entre horas.

Así podremos tener un registro completo de la alimentación del paciente.

Las pantallas de los otros parámetros a registrar serán muy similares a esta pero con su correspondiente información a registrar.

Mockup 12: Registro Alimentación



Mockup 13: Recompensas

Pantalla de Recompensas - Recompensas

En este apartado de recompensas es donde se han aplicado más los métodos de *gamificación* haciendo que la aplicación sea considerada como una especie de juego donde los usuarios tienen retos diarios, objetivos a cumplir...

De esta forma fomentamos que se mantenga la racha, que los usuarios registren sus aspectos diarios, motivando e incentivando así el proceso del tratamiento. En esta pantalla podemos observar la racha actual y la recompensa diaria por realizar algún registro. El premio por registrar información cada día serán 50 gemas.



Mockup 14: Objetivos

Pantalla de Recompensas - Objetivos

En esta pantalla de objetivos, buscamos que los usuarios mantengan su racha, premiando la cantidad de días consecutivos registrados en la aplicación. Dependiendo de la racha, obtendrán una recompensa u otra.

Cada objetivo estará asociado a un nivel, cuanto mayor sea la racha, mayor será el nivel alcanzado. El objetivo de los usuarios será llegar al nivel máximo, lo que implica conseguir una mayor racha de registros diarios. Con una racha más alta, cumpliremos el objetivo principal de la creación de la aplicación: que los niños adquieran constancia y se adapten de forma autónoma al tratamiento de la diabetes, viéndolo como un proceso entretenido y atractivo.

10 Arquitectura

En este apartado explicaré la arquitectura que he implementado en el desarrollo de la aplicación, mostrando los aspectos y organización tanto del código como de la base de datos utilizada.

10.1 *Arquitectura UI, Domain y Data*

La arquitectura implementada en esta aplicación sigue las mejores prácticas recomendadas por la documentación oficial de Android.

En este proyecto podemos observar 3 capas:

- **Capa UI o Capa de presentación:** Esta capa se encarga de mostrar visualmente los datos a los usuarios de la aplicación. Cuando se produce algún cambio en los datos, tanto si viene dado por alguna interacción del usuario como por alguna entrada externa, la interfaz de usuario debe actualizarse mostrando los cambios. En mi caso, he implementado un paquete llamado UI, el cual contiene las siguientes carpetas en su interior:
 - *Components:* Contiene diversos componentes visuales utilizados en la aplicación, como por ejemplo: menú lateral, barra de navegación inferior y barra superior.
 - *Navigation:* Esta carpeta contiene todos los elementos de navegación necesarios de la aplicación para brindar al usuario una mejor experiencia de uso, mantener un historial del flujo de pantallas y evitar la sensación de confusión en los usuarios.
 - *Screens:* Dentro de este paquete, están definidas todas las pantallas, vistas y componentes visuales de la aplicación.
 - *Theme:* He implementado tres archivos dentro de esta carpeta, utilizando Material Design 3, estos nos ayudan a definir la parte visual de la aplicación. Un archivo se encarga de la paleta de colores, otro para la tipografía y el último para la temática de la aplicación.

- **Capa *Domain*:** Capa que contiene la lógica de negocio y las reglas de la aplicación. Ubicada entre la capa UI y la capa de Datos. Es la responsable de encapsular la lógica que utilizarán los ViewModels. Incluye todos los modelos de datos y las interfaces de la aplicación, por ello dentro de esta capa podremos ver los siguientes paquetes en su interior:
 - *Models*: En esta subcarpeta he creado todos los modelos de datos de la aplicación, con sus respectivos atributos y parámetros necesarios, como por ejemplo: User, Alimentos, todos los diferentes tipos de registros, etcétera.
 - *Notifications*: He implementado dos archivos encargados de enviar una notificación recordatoria al usuario a diario, para evitar que pierda su racha. Aquí se implementa toda la lógica relacionada con las notificaciones.
- **Capa *Data*:** Esta capa es la encargada de manejar la obtención de datos. Dentro de esta capa he añadido toda la lógica para manejar la gestión y el acceso a las fuentes de datos, en mi caso, a la base de datos local (*BD_Alimentos.csv*) y a la base de datos en la nube, Cloud Firestore. En esta capa encontramos lo siguiente:
 - *AlimentosRepository*: Clase responsable de cargar los datos desde el archivo CSV de la base de datos local.
 - *Paquete Firebase*: En este paquete he implementado toda la lógica relacionada con la obtención de datos de Firebase.

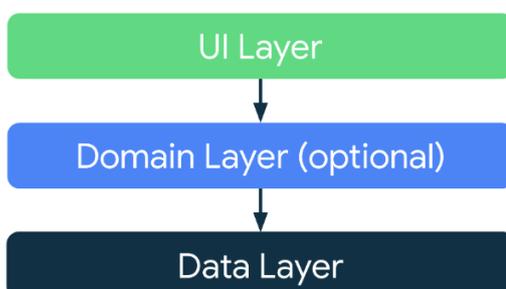


Ilustración 17: Arquitectura UI, Domain, Data recomendada por Android. ^[55]

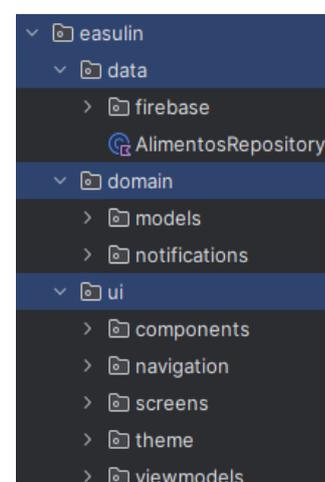


Ilustración 16: Organización código Arquitectura UI, Domain, Data

10.2 Arquitectura Model, View y ViewModel

La arquitectura *Model-View-ViewModel* (MVVM) es un patrón de diseño de software utilizado en el desarrollo de aplicaciones con interfaces de usuario complejas, como por ejemplo las aplicaciones de escritorio y móviles. La arquitectura MVVM divide la aplicación en tres componentes principales:

- **Model (Modelo):** Representa la lógica de negocio y se encarga de manejar los datos de la aplicación. El modelo se encarga de toda lógica de acceso a datos, ya sea de una base de datos local, un servicio web, API, etc. En ningún caso estará en contacto con la interfaz de usuario.
- **View (Vista):** Se trata de toda la interfaz de usuario. Encargada de mostrar y presentar todos los datos e información a los usuarios de la aplicación. También captura las interacciones que puedan realizar. La vista está directamente enlazada al *ViewModel*, por el contrario, nunca estará en contacto directo con el Modelo.
- **ViewModel (Vista-Modelo):** Actúa como un intermediario entre la vista y el modelo. Proporciona propiedades y comandos a los que la vista puede enlazarse. Gestiona la lógica de presentación. Se encarga de todo el flujo de información de datos:
 - Tanto datos de salida: datos que el modelo envía al *ViewModel* para mostrar por pantalla
 - Como datos de entrada: datos obtenidos por interacciones de usuarios para almacenar en la base de datos.
- El *ViewModel* se comunica con el *Model* para obtener y manipular los datos, y luego muestra estos datos en las diferentes *Views*.

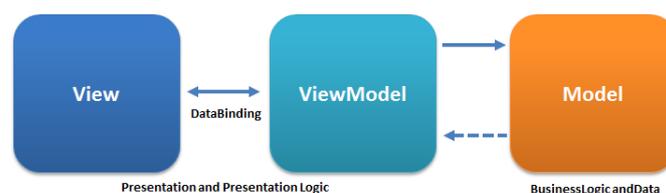


Ilustración 18: Flujo de información en la arquitectura MVVM. ^[56]

11 Base de datos utilizada – *Back End*

Para el desarrollo de mi aplicación, he escogido la plataforma Firebase, ya que nos brinda una gran cantidad de ventajas y beneficios a la hora de desarrollar aplicaciones móviles.

11.1 *Firestore*

Firestore es una plataforma de desarrollo de aplicaciones móviles y web de Google, ofrece una gran variedad de servicios diseñados para facilitar el mantenimiento de grandes aplicaciones. Firestore contiene diversas bibliotecas y SDK para Android, lo que facilita la integración, configuración y consulta de base de datos con Android Studio. Nos ofrece los siguientes beneficios:

- Sincronización de datos en tiempo real.
- Persistencia de datos en el dispositivo.
- Sistema de seguridad integrado con sus respectivas reglas de seguridad.
- Ofrece herramientas de desarrollo y monitoreo.

Para este proyecto he decidido utilizar los siguientes servicios: Cloud Firestore y Firebase Authentication.

11.1.1 *Cloud Firestore*

Cloud Firestore es una base de datos en la nube NoSQL, flexible y escalable para desarrollo móvil y web, construida sobre la infraestructura de Google Cloud.

Los datos se almacenan en documentos y en su interior encontramos los campos. Los campos contienen la información explícita y almacenan los diferentes tipos de datos, desde cadenas y números simples hasta objetos complejos y anidados. Los documentos se organizan en colecciones, estos documentos pueden contener subcolecciones, lo que permite crear estructuras de datos jerárquicas y escalables.

Aspectos clave de Cloud Firestore:

- **Flexibilidad:** Admite estructuras de datos jerárquicas y flexibles adaptándose a las necesidades de la aplicación.
- **Consulta expresiva:** Puedes usar consultas para recuperar documentos individuales o todos los documentos de una colección utilizando filtros. Permitiendo realizar consultas complejas de manera eficiente.
- **Actualización en tiempo real:** Permite la sincronización instantánea de datos entre dispositivos. Esto es esencial para aplicaciones que requieren actualizaciones en tiempo real, como por ejemplo los registros de los parámetros de Easulin.
- **Soporte sin conexión:** Almacena en caché los datos que su aplicación usa activamente, por lo que la aplicación puede escribir, leer, escuchar y consultar datos incluso si el dispositivo está fuera de línea.
- **Escalabilidad:** Diseñado para escalar automáticamente el crecimiento de volumen tanto de usuarios como de información, sin la necesidad de gestionarlo manualmente.

11.1.2 *Firestore Authentication*

Para mostrar una información personalizada a cada usuario, es necesario el proceso de autenticación, así podremos guardar de forma segura sus datos personales en la nube y poder brindar la misma experiencia personalizada en todos los dispositivos del usuario.

Firestore Authentication proporciona servicios *backend*, SDK fáciles de usar y bibliotecas de UI para autenticar a los usuarios en su aplicación. Proporciona métodos variados de autenticación. Simplifica el proceso de registro de usuarios y ofrece los siguientes beneficios:

- **Fácil integración:** Proporciona diversos métodos de autenticación sencillos y rápidos, incluyendo correo electrónico/contraseña, autenticación telefónica y autenticación a través de proveedores externos como Google, Facebook y Twitter.

- **Seguridad:** Sistema robusto para la gestión de usuarios. Incluye almacenamiento seguro de credenciales, verificación de correos electrónicos, recuperación de contraseñas y gestión de sesiones.
- **Compatibilidad:** Compatible con protocolos de autenticación estándar como *OAuth 2.0* y *OpenID Connect*.

De entre todos los métodos de autenticación proporcionados, decidí implementar la autenticación clásica con correo electrónico y contraseña.

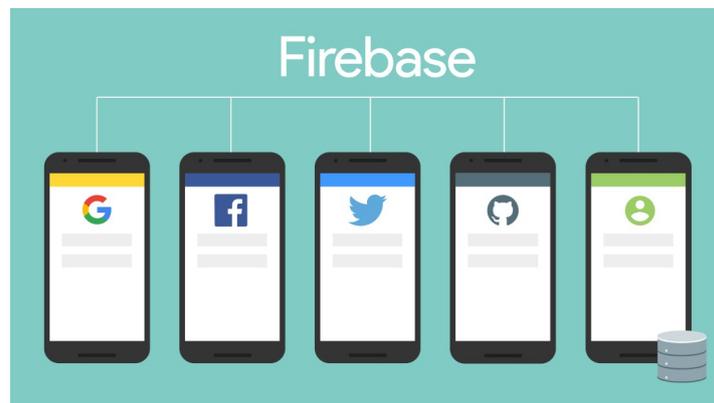
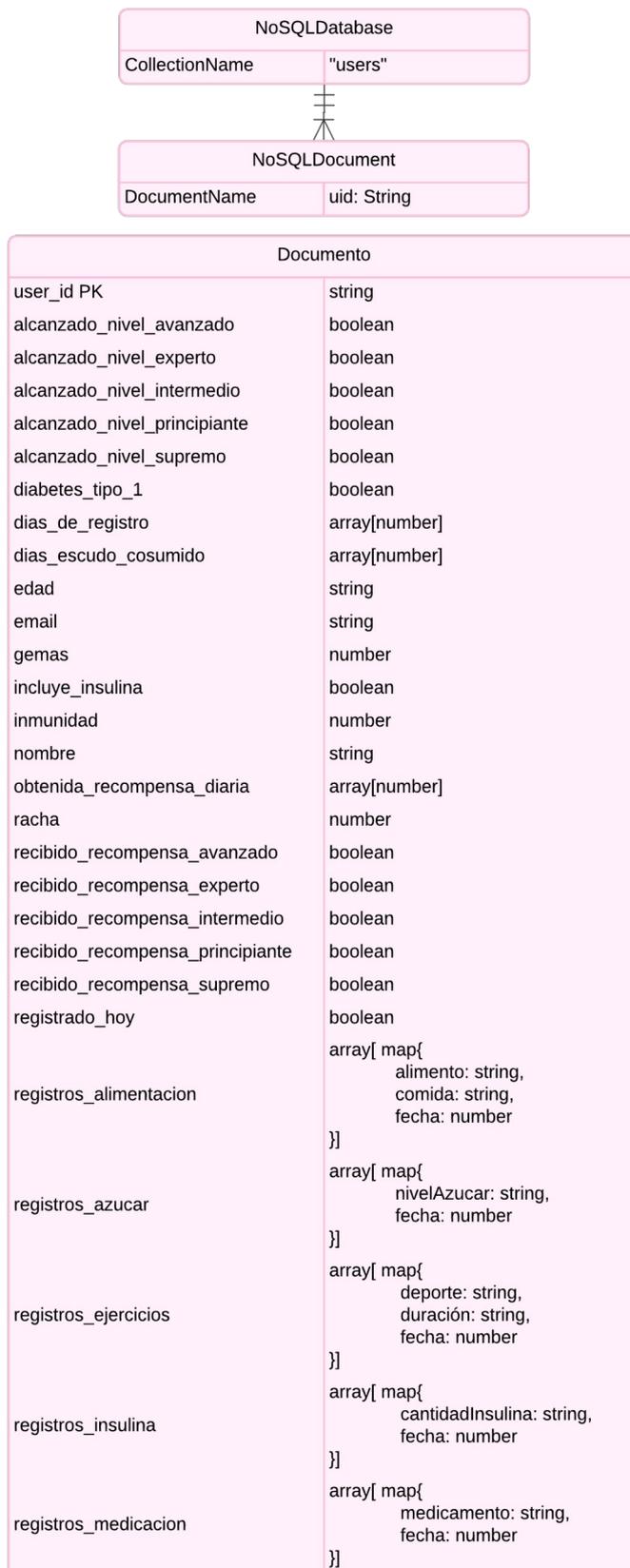


Ilustración 19: Diversos tipos de autenticación proporcionados por Firebase Authentication. ^[27]

11.2 Diagrama Base de Datos NoSQL Cloud Firestore



Como podemos observar, la base de datos creada en Cloud Firestore, al tratarse de una base de datos NoSQL, se almacenan tres tipos de datos: Colecciones, Documentos y Campos.

En el caso de mi aplicación, he creado una única colección llamada users.

Esta colección contendrá tantos documentos como usuarios existentes en la aplicación, el nombre de cada documento vendrá dado por el uid identificador del usuario en cuestión.

Finalmente, cada documento tendrá todos los campos mostrados en la figura. Cada vez que el usuario realice registros o interactúe con la aplicación, se ejecutará la lógica correspondiente, implementada en el desarrollo de la aplicación, para actualizar cada campo.

Ilustración 20: Diagrama Base de Datos NoSQL Cloud Firestore.

11.3 Campos de información Cloud Firestore

Cada uno de estos campos tienen un motivo y una importancia en la aplicación. Esta es la información que se almacena en cada campo:

user_id: Identificador único para cada usuario, generado automáticamente una vez registrado con Firebase Authentication. Este campo sirve para diferenciar y gestionar a los usuarios en la base de datos.

Nombre: El nombre del usuario.

Edad: Campo que contiene la edad del usuario introducida al registrarse.

Email: Correo electrónico del usuario.

Gemas: Cantidad de gemas que posee el usuario. Las gemas se pueden consumir comprando escudos de inmunidad y conseguir alcanzando objetivos y obteniendo recompensas.

Racha: El número de días consecutivos que el usuario ha registrado datos en la aplicación.

Inmunidad: Cantidad de escudos de inmunidad que tiene el usuario, hasta que no realice una compra la cantidad de escudos permanecerá a 0.

incluye_insulina: Valor booleano que indica si el usuario incluye insulina en su tratamiento.

diabetes_tipo1: Indica si el usuario tiene diabetes tipo 1, no se modifica en ningún momento.

registrado_hoy: Indica si en el día actual, el usuario ha registrado información en la aplicación.

dias_de_registro: Lista que contiene los días (en formato *Long*) en los que el usuario ha registrado algún parámetro en la aplicación. Si el usuario realiza un registro para ese día,

solamente se guardará una fecha independientemente de la cantidad de registros realizados ese día. Almacenará las fechas con la hora 00:00h para evitar problemas.

obtenida_recompensa_diaria: *Array* que contiene los días en que el usuario ha obtenido la recompensa diaria. Fechas guardadas igual que con el campo `dias_de_registro`.

dias_escudo_consumido: Lista que contiene los días en los que el usuario ha consumido algún escudo de inmunidad. Guardamos las fechas de la misma forma que con el campo `dias_de_registro`.

alcanzado_nivel_X (donde X indica el nivel de maestría): Valores que indican si el usuario ha alcanzado el nivel de maestría determinado, inicializados a `false`.

recibido_recompensa_X (donde X indica el nivel de maestría): Indica si el usuario ha recibido la recompensa por alcanzar el nivel de maestría determinado, evitando así poder volver a obtener esa recompensa.

registros_X (donde X indica el tipo de registro): Listas que contienen mapas con la información de los registros del usuario. Cada una almacena un mapa distinto según el tipo de registro.

Para obtener la información de los campos de cada usuario realicé la siguiente consulta (similar a la petición *GET* de una base de datos SQL):

1. Guardamos el *userId* (identificador del usuario) que actualmente está autenticado en la aplicación, obteniendo la instancia de la base de datos Cloud Firestore, y la instancia de autenticación de Firebase Authentication.
2. Accedemos a la colección “*users*”.
3. *Dentro de la colección “users” obtenemos el documento con el nombre del userId.*
4. Finalmente, guardamos la información del campo deseado en una variable, utilizando la petición *GET*. Según el tipo de dato que esperemos utilizaremos los diferentes tipos de *get*: *getString*, *getLong*, *getBoolean*, *get...*

11.4 Base de Datos Local

Para poder implementar la funcionalidad de registrar la alimentación consumida durante el día, al inicio mostraba una lista con escasos alimentos, al ver que iba a ser imposible rellenar esa lista manualmente, opté por buscar una base de datos que contuviese una gran cantidad de alimentos e ingredientes y fue aquí donde encontré la Base de Datos Española de Composición de Alimentos (BEDCA). Esta base de datos es una herramienta que proporciona información detallada sobre la composición nutricional y una amplia variedad de alimentos consumidos en España. Consideré que la información nutricional eran datos innecesarios para mi aplicación, ya que el usuario solamente tiene que añadir el alimento consumido, por lo tanto, realicé este proceso de implementación.

1. Descargué la base de datos BEDCA en formato .csv.
2. Este archivo contenía la información nutricional, innecesaria para la aplicación, así que lo modifiqué para que solamente almacenase el nombre de los alimentos.
3. Guardé esta base de datos, en la carpeta de asserts del proyecto, así tendremos un acceso rápido y directo a la información.
4. Para que al seleccionar en el campo de texto nos muestre todo el listado de alimentos, tuve que hacer una lectura completa de la base de datos local y almacenar estos alimentos en una Lista Mutable de Alimentos.

```
val alimentosList = mutableListOf<Alimento>() (Alimento: Data Class  
que contiene un solo parámetro, el nombre del alimento)
```

5. Una vez finalizada la lectura de la base de datos, ya se mostrarán visualmente los todos los alimentos, pudiendo realizar una búsqueda con función de autocompletado.

Al ser una base de datos oficial, aseguro que los registros de alimentación sean precisos y fiables, de esta forma el usuario podrá seleccionar el alimento consumido al tener una gran cantidad de alimentos almacenados. Si no existe el alimento en la base de datos, podrá registrarlo escribiendo manualmente el alimento en el campo de texto.

12 Implementación Visual – *Front End*

Una vez finalizada la fase de diseño, empecé con el desarrollo de la aplicación. Me aseguré de que los diseños y *mockups* fueran lo más realistas posibles para que así el desarrollo de las pantallas y vistas fuera más rápido, evitando pérdidas de tiempo en decidir qué elementos visuales añadir, ya que esto fue un aspecto que tuve en cuenta en la fase de diseño. De esta manera, la interfaz de usuario tenía que ajustarse a estos *mockups*.

Como he comentado en el apartado 6, las tecnologías utilizadas para desarrollar la parte visual de la aplicación será *Jetpack Compose* y *Material Design 3*.

12.1 *Proceso de implementación*

12.1.1 *Estructura de la interfaz*

Implementé las pantallas principales siguiendo los *mockups* creados, incluyendo la pantalla de inicio, pantalla de objetivos y recompensas, pantallas de registro de parámetros, y más.

Podemos observar 17 pantallas en la aplicación, distribuidas de la siguiente forma:

- **LoadingEasulin:** Pantalla de carga al iniciar la aplicación, esta vista muestra una animación durante 3,3 segundos. Seleccioné 3 segundos, ya que quería que fuese una duración reducida y los 0,3 segundos han sido añadidos para finalizar el ciclo de la animación evitando un corte. Esta animación solo se mostrará en dispositivos con API 21 (Android Lollipop) o superior, esto es debido a los elementos utilizados como *Canva* y *ImageBitmap* no están disponibles en versiones anteriores.
- **StartScreen:** Pantalla que muestra las dos opciones de autenticación, “Iniciar Sesión” en caso de estar registrado y “Regístrate” en caso de no tener una cuenta creada.
- **InicioSesionScreen:** Pantalla donde introducir tus credenciales de usuario.
- **SignUpFormX** (donde “X” es el número de pantalla): Para crear un usuario nuevo en la aplicación, he distribuido esta funcionalidad en 6 pantallas, cada una de ellas

contiene una pregunta necesaria para crear la cuenta. Separando el registro en varias pantallas con preguntas individuales evitamos la sobrecarga de información en una única pantalla de registro.

- **Inicio:** Pantalla que aparece al iniciar la aplicación en caso de estar autenticado. Muestra el resumen de registros del día actual.
- **RegistrarX** (donde “X” es el parámetro para registrar): La funcionalidad de registrar parámetros de la enfermedad está dividida en 4 pantallas, cada una de ellas relacionada con su correspondiente parámetro.
- **ProgresoScreen:** Pantalla dividida en 2, por un lado, mostramos la pantalla de **Recompensas** y por otro la pantalla de **Objetivos**.
- **Perfil:** Pantalla que muestra los datos e información del usuario.

12.1.2 *Navegación entre pantallas*

La estructura de navegación de la aplicación fue creada para que fuese intuitiva y fácil de usar. Utilicé Compose Navigation para gestionar la navegación entre diferentes pantallas, teniendo transiciones fluidas y una experiencia de usuario coherente.

Creé tres archivos de navegación:

- **EasulinNavigation:** Define el grafo de navegación de la aplicación. Utilizando NavHost y NavController de Jetpack Compose para gestionar las diferentes pantallas de la aplicación.
- **NavigationUtils:** Contiene las funciones de navegación. Función para obtener la ruta actual y función para comprobar si la ruta actual está dentro de una lista de rutas (pasada por parámetro).
- **NavScreen:** *Enum Class* que contiene las diferentes pantallas (rutas) de Easulin.

En el siguiente diagrama podemos observar la navegación implementada en la aplicación con las diferentes acciones y pantallas origen destino:

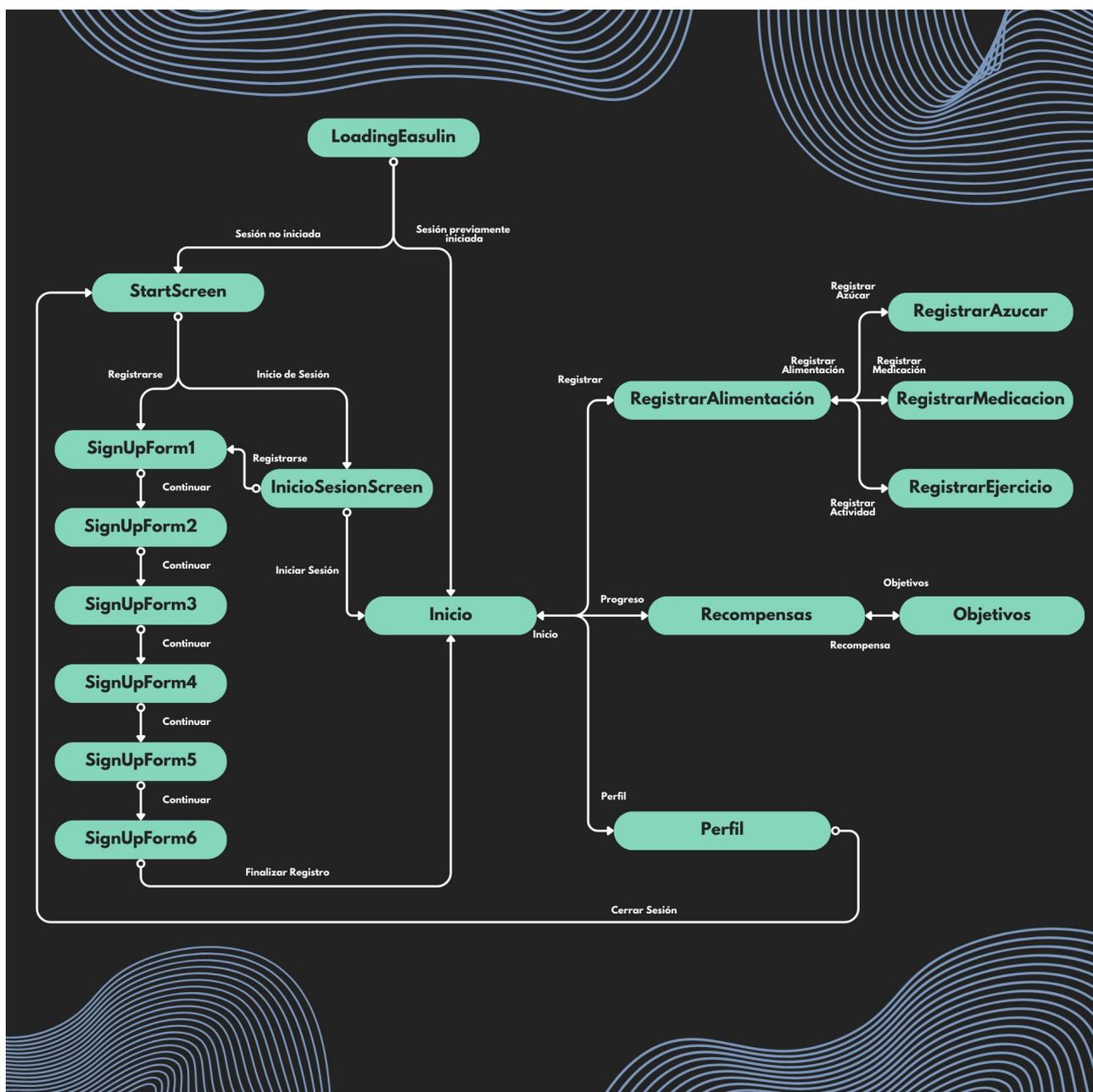


Ilustración 21: Diagrama de navegación Easulin

12.1.3 Componentes de navegación utilizados

He creado tres componentes de navegación para facilitar la experiencia de usuario dentro de la aplicación. Estos componentes ayudan a tener una navegación intuitiva, accediendo fácilmente a las diferentes secciones de la aplicación.

NavegacionInferior: Este elemento es presente en todo el ciclo de la aplicación una vez el usuario ha iniciado sesión. Se trata de una barra de navegación inferior utilizando *BottomAppBar* y *NavigationBar*, componentes de Materia 3. Este componente se muestra en los cuatros apartados principales de la aplicación: Inicio, Registrar, Progreso y Perfil.

- *BottomAppBar*: Define la barra de navegación inferior.
- *NavigationBar*: Contiene los ítems de navegación.
- *NavigationBarItem*: Define cada ítem de navegación en la barra inferior.
- *currentRouteList*: Verifica si la ruta actual está en la lista de rutas asociadas a un ítem. Esto lo he implementado, porque, en el apartado de Registrar, tenemos la posibilidad de encontrarnos en cuatro distintas pantallas. Independientemente de si estamos en una pantalla u otra, la barra de navegación inferior tendrá que ser visible para poder acceder a cualquier otro apartado.

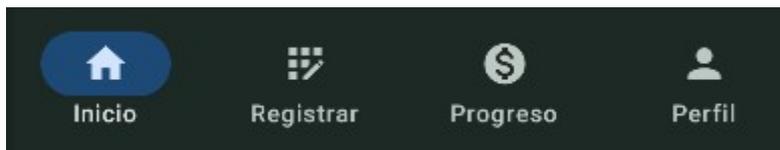


Ilustración 22: Barra de Navegación Inferior

MenuLateral: Este componente es un menú lateral de navegación (*drawer*) que solamente será visible en las pantallas de registrar parámetros. Se trata de un elemento *Composable* el cual contiene por parámetro:

- *NavHostController*: encargado de la navegación entre estas cuatro pantallas de registro.
- *DrawerState*: encargado de saber el estado de este menú lateral (si está abierto o cerrado). Remarca visualmente la pantalla abierta.



Ilustración 23: Menú Lateral: Apartado Registros

- ShowMenu: bandera encargada de mostrar u ocultar el menú lateral.

Utilicé los siguientes componentes de Material 3: ModalNavigationDrawer: muestra el menú lateral y NavigationDrawerItem: define cada ítem del menú.

TopBar: Componente superior mostrado en las pantallas de registro. Ayuda a definir en qué apartado de registro nos encontramos, gracias al texto e icono correspondientes al parámetro a registrar. Utilicé el componente CenterAlignedTopAppBar de M3.



Ilustración 24: Top Bar según el parámetro a registrar

12.1.4 Componentes visuales

Para mejorar la comprensión de la aplicación implementé diversos elementos visuales:

Textos descriptivos: Para clarificar funciones y elementos de la aplicación añadí diversos textos explicativos con un color y tamaño poco llamativos, tratándose de textos clarificativos.

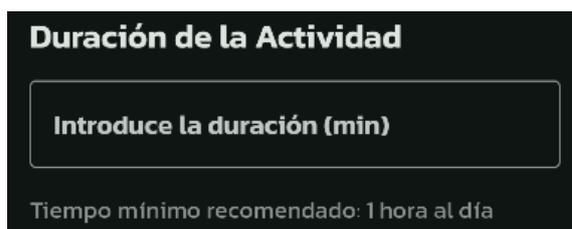


Ilustración 26: Texto explicativo sobre la recomendación de ejercicio diario



Ilustración 25: Textos clarificativos sobre la inmunidad y la recompensa diaria

Barras de progreso: Decidí implementar dos tipos de barras de progreso en la aplicación: una horizontal, que muestra visualmente cuánto falta para completar el formulario de

registro y cuánto falta para alcanzar un objetivo; y otra circular, que indica cuánto tiempo de ejercicio hemos realizado hoy y cuánto tiempo falta para alcanzar la meta diaria. Esto lo conseguí gracias a los componentes *LinearProgressIndicator* y *CircularProgressIndicator* respectivamente de Material Design 3.



Ilustración 28: *LinearProgressIndicator*

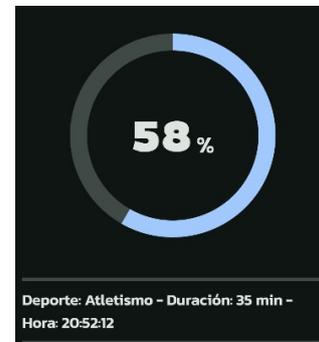


Ilustración 27: *CircularProgressIndicator*

Selector de fecha: Para visualizar los registros realizados en el pasado implementé un selector de fecha, que puede usarse con los iconos de incrementar o disminuir fecha, o pulsando en la fecha resaltada. Una vez pulsado en la fecha, aparecerá un calendario, de esta forma el usuario podrá seleccionar la fecha deseada a lo largo del tiempo y ver los registros de un determinado día. El selector manual fue implementado con elementos Jetpack Compose y la lógica con Kotlin, en cambio el calendario con componentes propios de Android (*DatePickerDialog* y *DatePicker*).



Ilustración 30: *Selector de fecha manual*

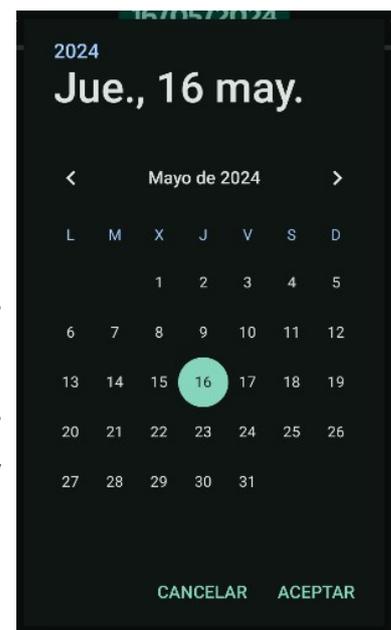


Ilustración 29: *Calendario para seleccionar fecha*

Notificaciones personalizadas: Para recordar al usuario que registre todos los parámetros del día y así evitar perder su progreso, he programado un recordatorio en forma de notificación. Este aparecerá diariamente a las 21:00h, antes de que finalice el día. De esta manera, si durante el transcurso del día no ha recordado registrar los aspectos relacionados con la diabetes, se le notificará para hacerlo. He creado la interfaz para que se adapte según el tema de color del dispositivo (modo claro o modo oscuro).

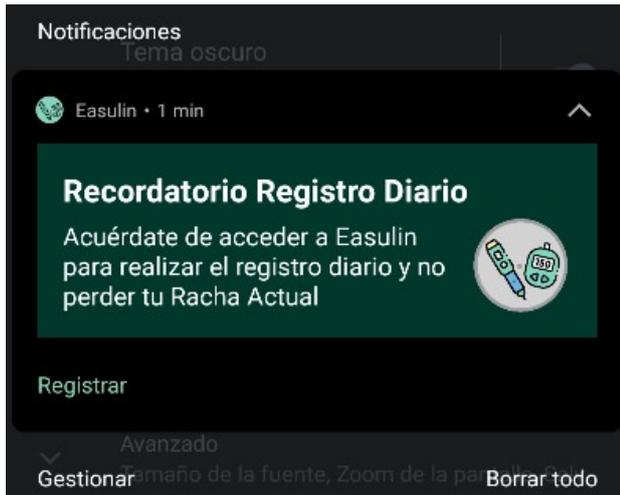


Ilustración 31: Notificación diaria Modo Oscuro



Ilustración 32: Notificación diaria Modo Claro

Historial semanal de la racha actual: Para visualizar la racha actual del usuario he creado una caja visual que nos muestra los registros de la última semana. Este componente creado es dinámico y según si ese día se ha registrado algo o no aparecerá un icono diferente. En caso de no haber registrado nada aparecerá una calavera, indicando la pérdida de la racha. En caso de haber consumido un escudo de inmunidad aparecerá el icono del escudo. En caso de haber realizado el registro mostrará un fuego (el icono correspondiente a la racha en Easulin). Finalmente, en caso de estar en el día actual y no haber registrado nada aún mostrará el icono de interrogante.



Ilustración 33: Racha dinámica semanal

12.1.5 Animaciones y transiciones

He implementado dos tipos de animaciones. Estas animaciones mejoran significativamente la experiencia de usuario (UX). Se han diseñado para proporcionar retroalimentación visual y mejorar la percepción de fluidez de la aplicación.

Pantalla de carga Animada (LoadingEasulin):

Antes de que se inicie la aplicación por completo, se muestra una pantalla de carga animada. La animación fue creada con la clase *Canvas*, clase propia de Kotlin. Se trata de dos arcos con los colores definidos en el esquema de colores de Easulin, girando alrededor del logotipo de la aplicación situado en la parte central de la pantalla.



Ilustración 34: Animación de carga LoadingEasulin

Animación de compra de inmunidad y obtención de gemas exitosa: Creé una animación para indicar al usuario que la compra de un escudo de inmunidad se ha realizado con éxito. Esta misma animación también aparecerá cuando obtengamos alguna recompensa de gemas. Su función es dar *feedback* al usuario sobre que la obtención tanto de gemas como de escudos de inmunidad se ha realizado correctamente. Esta animación está compuesta por dos efectos visuales:

- **Fade Out:** El icono correspondiente (escudo: para la compra de inmunidad y gema: para la obtención de gemas) empieza con una opacidad del 100% y a lo largo de la animación disminuye gradualmente hasta 0%, creando así un efecto de desvanecimiento.
- **Desplazamiento vertical:** Simultáneamente, el icono se desplaza hacia arriba. Partiendo de una posición central hasta la parte superior de la pantalla.

13 Pruebas, Testeo y Validación

Para comprobar la calidad, usabilidad y funcionalidades de Easulin, realicé un proceso de pruebas y testeos. Este proceso se dividió en dos fases.

13.1 *Pruebas realizadas durante el desarrollo*

En esta primera fase, una vez creada la primera versión de la aplicación, realicé repetidas ejecuciones para encontrar posibles errores de las diferentes funcionalidades. Entre estas pruebas, el principal método que utilicé fue el **método heurístico de Prueba y Error**. Este método consiste en, tal y como indica su propio nombre, frente a un determinado error probar una alternativa y verificar si funciona. Si es así, obtenemos la solución al problema, en caso contrario se busca una alternativa diferente, repitiendo este proceso hasta encontrar la solución al fallo. Con este proceso conseguí arreglar aquellos fallos más importantes e incapacitantes.

Por otro lado, una vez arreglados todos aquellos fallos de mayor importancia, intenté encontrar casos extremos donde la aplicación pudiera fallar, y estos fueron algunos ejemplos que solucioné:

- **Ingreso de datos no válidos en campos de texto:** Para evitar que los usuarios pudiesen introducir información inválida en un campo determinado, puse tres restricciones:
 - En campos de texto, indico explícitamente el tipo de datos esperado, texto, correo electrónico, contraseña, número, decimal... De esta forma el usuario no podrá introducir ningún dato inválido y el teclado del dispositivo mostrará por defecto el tipo de dato a introducir, en caso de ser un número mostrará solamente el teclado numérico, y en caso de texto el teclado tradicional.
 - Utilicé expresiones regulares en los *TextFields*, para evitar la introducción de caracteres inválidos o información corrupta. Si la información introducida no es la esperada o no es válida, no permitirá el envío de ese formulario.

- Indicaciones visuales de error en el campo introducido: si el usuario pretende introducir información inválida el campo de texto, se mostrará en rojo un mensaje de error, indicando la información esperada.
- **Registro de usuario a medias:** en el formulario para la creación de un nuevo usuario, una vez introducidos correo electrónico y contraseña deseados, se crea una instancia en Firebase Authentication, en caso de dejar el formulario a medias, esta cuenta se creaba, pero nos mostraba información nula e inválida, ya que no se llegaba a crear el usuario en Cloud Firestore con los campos de información requeridos. Para evitar este error he añadido un *Snackbar* que nos muestra un mensaje de alerta sobre si queremos dejar el formulario a medias, en caso de que el usuario así lo decida, la cuenta se creará también en Cloud Firestore pero con valores por defecto y no nulos.
- **Uso de tu cuenta en otro dispositivo:** cuando se iniciaba sesión en otro dispositivo diferente al habitual, surgían problemas. Esto era porque la zona horaria de algunos dispositivos no coincidía con la de otros, por este motivo, no constaban los registros pasados ni el progreso de racha de los últimos 7 días. Para solucionar este error, decidí imponer la zona horaria Tiempo Universal Coordinado (UTC), de esa forma todos los registros aparecerían y se guardarían correctamente.
- **Consumo de escudo de inmunidad:** encontré diversos errores al consumir los escudos de inmunidad. Uno de los problemas más significativos era el siguiente:
 - Si el usuario poseía varios escudos de inmunidad, cada vez que accedía al inicio de la aplicación se consumía un escudo, pues detectaba que el día anterior no se había registrado nada, consumiendo todos los escudos, hasta restablecer la racha actual a 0. Para solucionar este problema creé un nuevo campo en Cloud Firestore, una lista que guardaría las fechas de los días los cuales se consumieron escudos de inmunidad, de esta forma, si en la lista aparece la fecha actual no consumirá otro escudo, en caso de que el día anterior no se haya registrado nada y en el campo *dias_escudo_consumido* no aparezca la fecha actual, se consumirá solamente un escudo de inmunidad manteniendo así la racha actual.

- **Obtención de recompensas:** Cuando un usuario obtenía la recompensa diaria o la recompensa por alcanzar un nuevo nivel de maestría, podía volver a obtenerla si registraba otro parámetro o alcanzaba de nuevo ese nivel de maestría. Esto suponía un grave problema, ya que los usuarios podían obtener gemas infinitas. Por este motivo, tuve que crear varios campos en Cloud Firestore como lo son: *recibido_recompensa_X* (X indica nivel de maestría) y *dias_escudo_consumido*. Estos campos almacenan las fechas del día en el cual se ha obtenido esa determinada recompensa, negando así la obtención repetida de la recompensa para ese mismo día.

13.2 Pruebas con un grupo de usuarios

La segunda fase de pruebas busca evaluar la usabilidad de la aplicación.

13.2.1 Selección grupo de voluntarios

Se ha seleccionado un grupo diez de niños y adolescentes con edades entre 13 y 16 años. La selección de estos voluntarios se ha realizado en un círculo personal de mis amistades y conocidos. Uno de estos 10 seleccionados, es un niño que está diagnosticado con diabetes de tipo 1 (usuario óptimo para realizar pruebas reales de la aplicación).

13.2.2 Metodología

Primero de todo, realicé una breve explicación sobre la aplicación, para qué usuarios está destinada y los principales conceptos del proyecto. En ningún momento expliqué los pasos a seguir ni que elementos visuales encontrarán dentro de Easulin.

Posteriormente, se les propuso una lista de acciones y tareas a realizar dentro de la aplicación.

Se grabó la pantalla de estos dispositivos para contar la cantidad de pasos realizados en cada acción, observando momentos de duda y pasos fallidos, estas observaciones eran claves para realizar mejoras.

Finalmente, se hicieron preguntas sobre posibles mejoras y observaciones. Para obtener los resultados de este estudio, se realizó la comparativa entre la cantidad de pasos esperados y la cantidad de pasos realizados en cada acción.

13.2.3 *Tareas para realizar*

1. Registro en la aplicación.
2. Cerrar sesión.
3. Iniciar sesión con la cuenta previamente creada.
4. Registrar el alimento “Kit-Kat” en la merienda.
5. Registrar 45 minutos de boxeo.
6. Añadir el registro de 125 mg/dL de nivel de azúcar en sangre.
7. Recibir recompensa diaria.
8. Intentar comprar un escudo de inmunidad y cancelar la compra.
9. Ver que no hay registros de medicación para el día 15/05/2024.

13.2.4 *Preguntas finales*

- ¿Crees que ha sido fácil realizar todas las tareas propuestas?
- ¿Cómo valorarías la estética aplicación?
- ¿Crees que es una aplicación intuitiva, rápida y fácil de manejar?
- ¿Qué aspectos mejorarías de la aplicación?

13.2.5 *Resultados*

Para las tres primeras tareas: Registro, Cerrar sesión e Inicio de sesión, observamos que todos los usuarios las realizaron a la perfección, completándolas con la misma cantidad de acciones que la cantidad ideal. Es cierto que un usuario se equivocó al escribir la contraseña al registrarse, no la revisó con el botón de mostrar contraseña y no pudo iniciar sesión. En este caso, tuvimos que repetir el proceso.

Una vez pedimos a los usuarios que registraran un alimento, noté que tardaban unos segundos en procesar la tarea, ya que no estaban acostumbrados a la aplicación y era su primer contacto con ella. Sin embargo, rápidamente accedieron al apartado de Registro en la barra de navegación inferior y realizaron el registro sin problemas. En esta tarea, observé que algunos usuarios investigaban por su cuenta la aplicación sin ninguna intención de realizar la tarea solicitada, simplemente para ver como era la aplicación. Por ello, la media de acciones para realizar esta tarea está más elevada de lo ideal.

Para la quinta tarea, muchos usuarios se sintieron un poco perdidos al registrar el deporte, ya que no vieron el icono del menú lateral en la pantalla de registro. Cuando lo encontraron, abrieron el desplegable de opciones de registro de parámetros y finalizaron la tarea correctamente.

Al haber usado previamente el desplegable lateral izquierdo, para la sexta tarea todos los usuarios registraron a la perfección el nivel de azúcar en sangre, con la misma cantidad de acciones que la ideal.

En la séptima tarea, algunos usuarios realizaron alguna interacción adicional para encontrar la recompensa, pero en general no fue una tarea con grandes problemas.

Intentar comprar un escudo de inmunidad fue relativamente sencillo, ya que esta funcionalidad se encuentra en la misma pantalla que la recompensa diaria, y todos los usuarios lo consiguieron con la misma cantidad de interacciones que la ideal.

Finalmente, para la novena y última tarea hubo diversidad de enfoques. La mayoría de los usuarios interpretaron fácilmente que al pulsar en la fecha se abriría un calendario seleccionable, mientras que otros pulsaron múltiples veces el botón de disminuir fecha.

Aquellos usuarios que desde un inicio investigaron la aplicación por su cuenta, dejando de lado la tarea solicitada, fueron los que finalmente obtuvieron mejores resultados a posteriori.

Tras realizar las preguntas finales, las valoraciones obtenidas fueron muy positivas. En el apartado de estética todos ellos indicaron una puntuación de 10 sobre 10, sobre todo dos de ellos destacaron muy positivamente la elección de colores y los iconos utilizados. Además, indicaron que la aplicación era fácil de utilizar y muy intuitiva, a pesar de haber teni-

do algunos fallos. A medida que utilizaban Easulin entendían mejor el funcionamiento y la navegación. Algunas mejoras que propusieron fueron las siguientes:

- Incluir música de fondo para hacer más ameno el uso de la aplicación
- Añadir un pequeño tutorial la primera vez que se usa la aplicación, sobretudo en el apartado de Registro, para indicar que pueden escogerse otros parámetros a registrar que no sean solamente la alimentación (pantalla que aparece por defecto al navegar a Registro). Por este motivo se añadió a la *TopBar*, el icono de menú, para indicar que esta pantalla tiene un menú lateral.



Ilustración 35: Resultados de la prueba con un grupo de usuarios

14 Conclusiones

Una vez finalizado este proyecto, he logrado cumplir todos los objetivos planteados inicialmente. Para realizar este trabajo, intenté adaptarme al máximo a la planificación y a los tiempos marcados desde un inicio. Bien es cierto que, a pesar de haber sufrido algunas variaciones en estos tiempos, finalmente logré completar todas las fases del trabajo en los plazos establecidos.

Durante este proceso, he podido aprender una gran cantidad de información, tanto tecnológica como sanitaria. En el ámbito sanitario, me informé mucho más a fondo sobre una enfermedad que afecta a cientos de millones de personas en todo el mundo. En el ámbito más tecnológico, he aprendido desde cero y utilizado estas tres herramientas de programación: Kotlin, Jetpack Compose y Material Design 3. Gracias a esto, pude crear una interfaz intuitiva, moderna y adaptada tanto a las necesidades del usuario como a los diseños iniciales, obteniendo una experiencia de usuario muy bien valorada por el grupo de voluntarios al cual se le realizó el testeo.

Para el desarrollo del *backend*, utilicé la plataforma Firebase, sobre la cual tenía conocimientos básicos, pero centrar una aplicación compleja en esta plataforma me ha ayudado mucho a impulsar mis conocimientos sobre esta tecnología de datos.

Finalmente, conseguí un código bien organizado y estructurado, aplicando buenas prácticas como las arquitecturas *Model-View-ViewModel* y *UI-Domain-Data*.

Este proyecto ha conseguido cumplir todos los objetivos establecidos inicialmente, obteniendo un producto innovador y funcional que combina métodos de *gamificación* para facilitar el manejo de la diabetes en niños y adolescentes, mejorando la adherencia al tratamiento y su calidad de vida.

Estoy muy orgulloso del trabajo realizado y los resultados obtenidos, y confío plenamente en que Easulin tenga un impacto positivo en la vida de aquellos niños y adolescentes que padecen esta enfermedad.

15 Bibliografía

- [1] 3, E. (2024, 3 enero). *¿Qué es la gamificación y cuáles son sus objetivos?* | EDUCACIÓN 3.0. EDUCACIÓN 3.0. <https://www.educaciontrespuntocero.com/noticias/gamificacion-que-es-objetivos/>
- [2] Admin. (2022a, marzo 17). *No empieces la casa por el tejado: ¿por qué es importante el análisis de requisitos antes del desarrollo del software?* Kopen Software | Tryton. <https://www.kopen.es/no-empieces-la-casa-por-el-tejado-por-que-es-importante-el-analisis-de-requisitos-antes-del-desarrollo-del-software/#:~:text=Un%20an%C3%A1lisis%20de%20requisitos%20o,bases%20de%20todo%20el%20proyecto.>
- [3] Admin. (2022b, marzo 17). *No empieces la casa por el tejado: ¿por qué es importante el análisis de requisitos antes del desarrollo del software?* Kopen Software | Tryton. <https://www.kopen.es/no-empieces-la-casa-por-el-tejado-por-que-es-importante-el-analisis-de-requisitos-antes-del-desarrollo-del-software/#:~:text=Un%20an%C3%A1lisis%20de%20requisitos%20o,bases%20de%20todo%20el%20proyecto.>
- [4] *Building an Android app with Jetpack Compose and Firebase.* (s. f.). The Firebase Blog. <https://firebase.blog/posts/2022/04/building-an-app-android-jetpack-compose-firebase>
- [5] C026a9c090eca39c3b398fd. (2022, 5 julio). *¿Qué es un Análisis de Requerimientos?* | *Espacios Business Media.* Espacios Business Media. <https://www.espacios.media/que-es-un-analisis-de-requerimientos/#:~:text=Un%20an%C3%A1lisis%20de%20requerimientos%20es%20un%20estudio%20profundo%20de%20una,necesarios%20para%20su%20buen%20desarrollo.>
- [6] *Cada año se diagnostican en España entre 1.200 y 1.500 nuevos casos de diabetes tipo 1 en menores de 15 años, una de las cifras más altas del sur de Europa.* (2023, 11 mayo). [Comunicado de prensa]. https://www.sanofi.com/assets/countries/spain/docs/press/2023/2023_05_11_NdP-

Cada-a-o-se-diagnostican-en-Espa-a-entre-1.200-y-1.500-nuevos-casos-de-diabetes-tipo-1-en-menores-de-15-a-os.pdf

- [7] *Capa de la IU.* (s. f.). Android Developers. <https://developer.android.com/topic/architecture/ui-layer?hl=es-419>
- [8] Chasse, B. (2021, 30 agosto). *Taking a crack at gamification.* Duolingo Blog. <https://blog.duolingo.com/gamification-design/>
- [9] colaboradores de Wikipedia. (2024, 27 abril). *Ludificación.* Wikipedia, la Enciclopedia Libre. <https://es.wikipedia.org/wiki/Ludificaci%C3%B3n>
- [10] *Consulta interactiva del SNS.* (s. f.). <https://pestadistico.inteligenciadegestion.sanidad.gob.es/publicoSNS/S/base-de-datos-de-clinicos-de-atencion-primaria-bdcap>
- [11] *Creamos una app aplicando Material 3 y Jetpack compose.* (2023, 7 septiembre). [Vídeo]. YouTube. <https://www.youtube.com/watch?v=vHfdoUqimk4&list=PLxFEWPIJUGSo90rENw7YbjpzfKJri-JHd>
- [12] *Cris Web.* (2021, 7 julio). <https://www.crisweb.me/blog/2021-07-07-la-gamificaci%C3%B3n-y-sus-elementos/>
- [13] *CURSO: ANDROID desde CERO con KOTLIN - TUTORIAL ANDROID STUDIO [2023] - PARTE 1.* (2023, 2 marzo). [Vídeo]. YouTube. https://www.youtube.com/watch?v=vJapzH_46a8&list=PL8ie04dqq7_M8nfPA9DPiAy7NsoZQpVAf
- [14] *CURSO: ANDROID desde CERO con KOTLIN - TUTORIAL ANDROID STUDIO [2023] - PARTE 2.* (2023, 2 marzo). [Vídeo]. YouTube. https://www.youtube.com/watch?v=ndqlqh6joGA&list=PL8ie04dqq7_M8nfPA9DPiAy7NsoZQpVAf&index=2
- [15] *CURSO: ANDROID Nivel INTERMEDIO con KOTLIN - TUTORIAL ANDROID STUDIO [2024].* (2023, 14 septiembre). [Vídeo]. YouTube. https://www.youtube.com/watch?v=UaR7GSNACsM&list=PL8ie04dqq7_M8nfPA9DPiAy7NsoZQpVAf&index=3

- [16] Cyberstream. (2024, 4 mayo). *Fecha de lanzamiento de Material Design 3: ¡Descubre cuándo salió al mercado!* Byron Vargas ®. <https://www.byronvargas.com/web/cuando-salio-material-design-3/>
- [17] Davidbritch. (2017, 7 agosto). *The Model-View-ViewModel Pattern - Xamarin.* Microsoft Learn. <https://learn.microsoft.com/en-us/previous-versions/xamarin/xamarin-forms/enterprise-application-patterns/mvvm>
- [18] *Diabetes: MedlinePlus enciclopedia médica.* (s. f.). <https://medlineplus.gov/spanish/ency/article/001214.htm>
- [19] *Diabetes tipo 2 en niños - Síntomas y causas - Mayo Clinic.* (2023, 18 noviembre). <https://www.mayoclinic.org/es/diseases-conditions/type-2-diabetes-in-children/symptoms-causes/syc-20355318#:~:text=Descripci%C3%B3n%20general,graves%20consecuencias%20a%20largo%20plazo.>
- [20] *Diabetes tipo 1 - Síntomas y causas - Mayo Clinic.* (2024, 27 marzo). <https://www.mayoclinic.org/es/diseases-conditions/type-1-diabetes/symptoms-causes/syc-20353011#:~:text=La%20diabetes%20tipo%201%2C%20antes,las%20c%C3%A9lulas%20para%20producir%20energ%C3%ADa.>
- [21] *Diabetes y ejercicio: Cuándo controlar tu nivel de azúcar en sangre.* (2024a, abril 20). Mayo Clinic. <https://www.mayoclinic.org/es/diseases-conditions/diabetes/in-depth/diabetes-and-exercise/art-20045697#:~:text=Descansa%20al%20menos%20un%20d%C3%ADa,tres%20d%C3%ADas%20a%20la%20semana.>
- [22] *Diabetes y ejercicio: Cuándo controlar tu nivel de azúcar en sangre.* (2024b, abril 20). Mayo Clinic. <https://www.mayoclinic.org/es/diseases-conditions/diabetes/in-depth/diabetes-and-exercise/art-20045697#:~:text=Descansa%20al%20menos%20un%20d%C3%ADa,tres%20d%C3%ADas%20a%20la%20semana.>
- [23] *Diabetis.* (s. f.). Canal Salut. <https://canalsalut.gencat.cat/ca/salut-a-z/d/diabetis/>

- [24] *Diguan.es*. (2021, 16 septiembre). diguan.es. <https://www.diguan.es/>
- [25] *Existen dos tipos fundamentales de diabetes: la Diabetes Mellitus tipo 1 (DM1) y la Diabetes Mellitus tipo 2 (DM2)*. (s. f.). <https://www.diabetes.ascensia.es/blog/blog-detail-one/#:~:text=persistente%20o%20cr%C3%B3nica,-,Existen%20dos%20tipos%20fundamentales%20de%20diabetes%3A%20la%20Diabetes%20Mellitus%20tipo,personas%20a%20las%20que%20afecta.>
- [26] FACTORS INFLUENCING QUALITY OF MOBILE APPS: ROLE OF MOBILE APP DEVELOPMENT LIFE CYCLE. (2014, septiembre). <https://airccse.org/journal/ijsea/ijsea.html>.
<https://airccse.org/journal/ijsea/papers/5514ijsea02.pdf>
- [27] *Firebase Authentication*. (s. f.). Firebase.
<https://firebase.google.com/docs/auth?hl=es>
- [28] *Firestore* | *Firebase*. (s. f.). Firebase.
<https://firebase.google.com/docs/firestore?hl=es>
- [29] *Gamificación: el aprendizaje divertido | educativa*. (s. f.). <https://www.educativa.com/blog-articulos/gamificacion-el-aprendizaje-divertido/#:~:text=La%20Gamificaci%C3%B3n%20es%20una%20t%C3%A9cnica,concretas%20entre%20otros%20muchos%20objetivos.>
- [30] *Gamificación en el aula: por qué y cómo aplicarla*. (2024, 4 junio). Santander Open Academy. <https://www.santanderopenacademy.com/es/blog/gamificacion-en-el-aula.html>
- [31] *Gamificación en la enseñanza*. (2022, 18 julio). Biblioteca de Educación | Ministerio de Educación, Formación Profesional y Deportes. <https://www.educacionfpydeportes.gob.es/biblioteca-central/blog/2022/gamificacion-en-la-ensenanza>

- [32] *Guía de arquitectura de apps.* (s. f.). Android Developers. <https://developer.android.com/topic/architecture?hl=es-419#recommended-app-arch>
- [33] *Jetpack Compose, el kit de herramientas para el desarrollo de IU de apps - Android Developers | Jetpack Compose | Android Developers.* (s. f.). Android Developers. <https://developer.android.com/develop/ui/compose?hl=es-419>
- [34] *Kotlin Programming Language.* (s. f.). Kotlin. <https://kotlinlang.org/education/why-teach-kotlin.html>
- [35] Mansur, O. (2023, 20 junio). *Los estudios sobre hábitos detrás de tu racha de Duolingo.* Duolingo Blog. <https://blog.duolingo.com/es/racha-duolingo-habitos/>
- [36] *Material design.* (s. f.). Material Design. <https://m3.material.io/>
- [37] National Library of Medicine. (s. f.). *Glucosa en la sangre.* <https://medlineplus.gov/spanish/bloodglucose.html>
- [38] OrlandDroyd. (2024, 28 enero). “La capa de Presentación” (Parte 3) Clean Architecture + MVVM + Android. *Medium.* <https://medium.com/@orlandroyd/principios-de-clean-architecture-mvvm-enfocado-al-desarrollo-android-nativo-parte-3-la-capa-de-ece7dcf3dab3>
- [39] *¿Qué es la insulina?* (s. f.). Portal de Salud de la Junta de Castilla y León. <https://www.saludcastillayleon.es/AulaPacientes/es/guia-diabetes/respuestas-preguntas-frecuentes/insulina>
- [40] Redacciondiabetes, & Redacciondiabetes. (2023, 11 mayo). Cada año se diagnostican en España entre 1.200 y 1.500 nuevos casos de diabetes tipo 1 en menores de 15 años - Canal Diabetes. *Canal Diabetes - La mejor información para la persona con diabetes.* <https://canaldiabetes.com/diabetes-tipo-1-en-menores-de-15-anos/>
- [41] Rehkopf, D. M. (s. f.). *¿Qué es un tablero kanban?* | Atlassian. Atlassian. <https://www.atlassian.com/es/agile/kanban/boards#:~:text=%22Kanban%22%20es>

- %20una%20palabra%20japonesa,el%20mundo%20trabaje%20en%20sinton
%C3%ADa.
- [42] Reyes, L. (2021, 7 diciembre). Aplicando el patrón de diseño MVVM -
Leomaris Reyes - Medium. *Medium*.
[https://medium.com/@reyes.leomaris/aplicando-el-patr%C3%B3n-de-dise
%C3%B1o-mvvm-d4156e51bbe5](https://medium.com/@reyes.leomaris/aplicando-el-patr%C3%B3n-de-dise%C3%B1o-mvvm-d4156e51bbe5)
- [43] Ruiz, J. Z. (2004a). ¿Por qué fracasan los proyectos de software? Un
enfoque organizacional. *ResearchGate*. <https://doi.org/10.13140/RG.2.1.4741.3206>
- [44] Ruiz, J. Z. (2004b). ¿Por qué fracasan los proyectos de software? Un
enfoque organizacional. *ResearchGate*. <https://doi.org/10.13140/RG.2.1.4741.3206>
- [45] Solutions, V., & Jain, A. (2023, 2 enero). *Qué son los requisitos funcionales:
ejemplos, definición, guía completa*. Visure Solutions.
[https://visuresolutions.com/es/blog/functional-requirements/#:~:text=Un
%20requisito%20funcional%20es%20una,caracter%C3%ADsticas%20que%20el
%20usuario%20detecta.](https://visuresolutions.com/es/blog/functional-requirements/#:~:text=Un%20requisito%20funcional%20es%20una,caracter%C3%ADsticas%20que%20el%20usuario%20detecta.)
- [46] *Tabla de niveles de glucosa en sangre para niños: tabla para diabetes*. (s. f.).
[https://www.makingdiabeteseasier.com/es/gestionando-la-diabetes/para-padres-e-
hijos/tabla-niveles-glucosa-sangre-para-ninos-diabetes](https://www.makingdiabeteseasier.com/es/gestionando-la-diabetes/para-padres-e-hijos/tabla-niveles-glucosa-sangre-para-ninos-diabetes)
- [47] *The causes of project failure*. (1990a, noviembre 1). IEEE Journals &
Magazine | IEEE Xplore. <https://ieeexplore.ieee.org/document/62322>
- [48] *The causes of project failure*. (1990b, noviembre 1). IEEE Journals &
Magazine | IEEE Xplore. <https://ieeexplore.ieee.org/document/62322>
- [49] *Tu primera interfaz en Android JetPack Compose: uso de texto, imágenes,
filas y columnas*. (2022, 30 agosto). [Vídeo]. YouTube.
[https://www.youtube.com/watch?
v=hfVv4tTOZII&list=PLxFEWPIJUGSpIJtxXmOC84vU5kxiVRDpl](https://www.youtube.com/watch?v=hfVv4tTOZII&list=PLxFEWPIJUGSpIJtxXmOC84vU5kxiVRDpl)
- [50] World Health Organization: WHO & World Health Organization: WHO. (2023,
5 abril). *Diabetes*. <https://www.who.int/es/news-room/fact-sheets/detail/diabetes>

- [51] *Curso Kotlin multiplatform - Curso Kotlin para ANDROID*. (2024, 5 junio). Curso Kotlin Para ANDROID. <https://cursokotlin.com/curso-kotlin-multiplatform/>
- [52] *Material You | Figma*. (s. f.). Figma. <https://www.figma.com/community/file/979511084534438755/material-you>
- [53] *Cris Web*. (2021, 7 julio). <https://www.crisweb.me/blog/2021-07-07-la-gamificaci%C3%B3n-y-sus-elementos/>
- [54] Manojkumawat. (2024, 5 abril). Kotlin Multiplatform - Manojkumawat - Medium. *Medium*. <https://medium.com/@manojkumawat134/kotlin-multiplatform-e9d6bb5e4263>
- [55] *Capa de dominio*. (s. f.). Android Developers. <https://developer.android.com/topic/architecture/domain-layer?hl=es-419>
- [56] Wikipedia contributors. (2024, 24 mayo). *Model–view–viewmodel*. Wikipedia. <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel>
- [57] *Sociedad Española de Diabetes | SED*. (s. f.). <https://www.sediabetes.org/>
- [58] *Gemas*. (s. f.). FlatIcon. https://www.flaticon.es/icono-gratis/gemas_3621789?related_id=3621789&origin=pack
- [59] *Racha*. (s. f.). FlatIcon. https://www.flaticon.es/icono-gratis/fuego_599502?term=fuego&page=1&position=4&origin=search&related_id=599502
- [60] *Escudo inmunidad*. (s. f.). FlatIcon. https://www.flaticon.es/icono-gratis/escudo_168951?term=escudo&page=3&position=19&origin=search&related_id=168951
- [61] *Iconos Maestría*. (s. f.). FlatIcon. <https://www.flaticon.es/packs/martial-arts-50>

- [62] *Calavera*. (s. f.). FlatIcon. https://www.flaticon.es/icono-gratis/calavera-y-huesos_1355981?term=calavera&page=1&position=6&origin=search&related_id=1355981
- [63] *Interrogante*. (s. f.). FlatIcon. https://www.flaticon.es/icono-gratis/bloquear_705844?term=interrogante&related_id=705844
- [64] *Elementos logo*. (s. f.). FlatIcon. <https://www.flaticon.es/packs/world-diabetes-day-6>