



UNIVERSIDAD DE MÁLAGA



Graduado en Ingeniería del Software

# Aplicación Móvil para el Control y Ayuda al Paciente de la Diabetes

## Mobile Application for the Control and Assistance for Patients with Diabetes

Realizado por  
Ángel Baeza Expósito

Tutorizado por  
Gabriel Jesús Luque Polo  
Christian Cintrano López

Departamento  
Lenguajes y Ciencias de la Computación  
UNIVERSIDAD DE MÁLAGA

MÁLAGA, junio de 2021



UNIVERSIDAD  
DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA  
GRADUADO EN INGENIERÍA DEL SOFTWARE

## **Aplicación Móvil para el Control y Ayuda al Paciente de la Diabetes**

### **Mobile Application for the Control and Assistance for Patients with Diabetes**

Realizado por  
**Ángel Baeza Expósito**

Tutorizado por  
**Gabriel Jesús Luque Polo**  
**Christian Cintrano López**

Departamento  
**Lenguajes y Ciencias de la Computación**

UNIVERSIDAD DE MÁLAGA  
MÁLAGA, JUNIO DE 2021

Fecha defensa: (POR DEFINIR)



## **Dedicatoria**

Tras todo el esfuerzo realizado para llegar aquí y finalizar así esta etapa de mi vida, sólo puedo dar gracias a las personas que me han acompañado y ayudado a estar aquí.

A mis padres por permitirme vivir esta etapa de mi vida.

A mi hermano, por aguantarme tanto tiempo y alegrarse de mis éxitos.

A Mar, que ha sido el motivo principal de que haya aguantado tanto tiempo.

A Raúl, Emilio, Francisco y Villa, con los que he compartido momentos increíbles y que han comprendido que la distancia sólo ha sido un trámite.

A mis compañeros de facultad, Carlos, Paco, Jesús y Joslu,  
el otro pilar que me encontré al venir a Málaga.

Y, por último, a los profesores que se han preocupado de que este momento llegase.

Especialmente a mi tutor, Gabriel, por confiar en mí para hacer este proyecto bajo su tutela.

Y en lo personal, el que más me ayudó al comenzar todo esto, Agustín.



# Resumen

Comidas con altas calorías, refrescos con azúcar y un estilo de vida cada vez más sedentario son causas cada vez más comunes de problemas relacionados con la alimentación que pueden ser desde simples dolores de estómago hasta el tema tratado con este Trabajo de Fin de Grado (TFG) como puede ser la diabetes. La facilidad hoy en día de acceder a comida rápida y los continuos anuncios ha provocado que una enfermedad hasta ahora presente en personas mayores invierta de manera cada vez más pronunciada su rango de edad, siendo cada vez más jóvenes los que se ven afectados por ella.

Para la mayoría de las personas que la padecen la diabetes no es difícil de gestionar, pero sí muy engorroso. El médico debe conocer en la medida de lo posible los niveles de glucosa que el paciente experimenta previo a cada comida. Además, hay un cambio en cómo estas personas perciben las comidas, teniendo que considerar qué es lo que están verdaderamente ingiriendo para llevar sus niveles e inyecciones de insulina de manera correcta. Con todos estos aspectos se desarrolla este TFG cuyo objetivo principal es el desarrollo de una aplicación móvil que permita facilitar la gestión de todos estos datos ayudando así al usuario en términos de alimentación, información y control en su vida diaria.

**Palabras clave:** Diabetes, Android, Flutter, iOS, Dart, Móvil, Open-FoodFacts

# Abstract

High-calorie meals, sugary soft drinks and an increasingly sedentary lifestyle are increasingly common causes of food-related problems that can range from simple stomach pains to the topic treated with this Final Degree Project (FDP) such as diabetes. The ease of access to fast food today and the continuous announcements have caused a disease so far present in older people to invest in an increasingly pronounced way their age range, with more and more young people being affected by it.

For most people with diabetes it is not difficult to manage, but it is very cumbersome. The doctor should know as much as possible the glucose levels that the patient experiences prior to each meal. In addition, there is a change in how these people perceive meals, having to consider what they are actually ingesting to get their insulin levels and injections right. With all these aspects is developed this FDP whose main objective is the development of a mobile application that allows to facilitate the management of all this data helping the user in terms of food, information and control in their daily life.

**Keywords:** Diabetes, Android, Flutter, iOS, Dart, Mobile, OpenFood-Facts

# Índice

<b>1. Introducción</b>	<b>27</b>
1.1. Motivación . . . . .	27
1.2. Objetivos . . . . .	28
1.3. Estructura del Documento . . . . .	29
1.4. Sumario de Entregables . . . . .	31
<b>2. Tecnologías y Herramientas</b>	<b>33</b>
2.1. Tecnologías . . . . .	33
2.1.1. Dart . . . . .	33
2.1.2. Flutter . . . . .	34
2.1.3. SQLite . . . . .	34
2.2. Herramientas de Desarrollo . . . . .	35
2.2.1. Visual Studio Code . . . . .	35
2.2.2. Android Studio . . . . .	35
2.2.3. VirtualBox . . . . .	36
2.2.4. MoonModeler . . . . .	36
2.2.5. Postman . . . . .	36
2.2.6. DB Browser . . . . .	37
2.3. Herramientas de Gestión . . . . .	37
2.3.1. Git / Github . . . . .	37
2.3.2. Microsoft Office: Teams . . . . .	37
2.4. Herramientas de Documentación . . . . .	38
2.4.1. Microsoft Office: Word . . . . .	38
2.4.2. LaTeX: MiKTeX con Texmaker . . . . .	38
2.4.3. MagicDraw . . . . .	38
2.5. Herramientas Adicionales . . . . .	39
2.5.1. Microsoft Forms . . . . .	39



<b>3. Metodología</b>	<b>41</b>
3.1. Enfoque . . . . .	41
3.2. Metodología . . . . .	41
3.3. Iteraciones Realizadas . . . . .	42
3.3.1. Tareas Previas . . . . .	42
3.3.2. Iteraciones Posteriores . . . . .	43
3.4. Estudio de Trabajos Similares . . . . .	44
<b>4. Análisis y Modelado</b>	<b>45</b>
4.1. Estudio de Mercado . . . . .	45
4.1.1. Información sobre la Diabetes . . . . .	45
4.1.2. Decisión de Aplicación Móvil . . . . .	49
4.2. Requisitos Definidos . . . . .	51
4.3. Diagrama de Navegación . . . . .	51
4.4. Diagramas de Clases . . . . .	53
4.4.1. Modelos de Objetos . . . . .	53
4.4.2. Servicios de Base de Datos . . . . .	55
4.4.3. Servicios para Librerías . . . . .	57
4.4.4. Útiles . . . . .	58
4.5. Constantes . . . . .	60
4.6. Interfaz Gráfica . . . . .	61
4.6.1. Línea de Diseño . . . . .	61
4.6.1.1. Icono de la aplicación . . . . .	62
4.6.1.2. Colores Principales y Adaptación de Modos . . . . .	62
4.6.1.3. Explicación del Nombre: Mesuri . . . . .	63
4.7. Estructura del Proyecto . . . . .	64
4.7.1. Directorio <i>assets</i> . . . . .	65
4.7.2. Directorio <i>lib</i> . . . . .	65
<b>5. Desarrollo e Implementación</b>	<b>69</b>
5.1. Base de Datos . . . . .	69
5.1.1. Modelo . . . . .	70

5.1.2.	Explicación de las Tablas . . . . .	71
5.1.2.1.	Users (Usuarios) . . . . .	72
5.1.2.2.	Measures (Mediciones) . . . . .	72
5.1.2.3.	Medicines (Medicinas) . . . . .	73
5.1.2.4.	Measures_Medicines . . . . .	73
5.1.2.5.	Suggestions (Sugerencias) . . . . .	74
5.1.2.6.	Medicines_Suggestions . . . . .	75
5.1.2.7.	Exercises (Ejercicios) . . . . .	75
5.1.2.8.	Symptoms (Síntomas) . . . . .	76
5.1.2.9.	Notifications (Notificaciones) . . . . .	76
5.2.	Librerías Adicionales Utilizadas . . . . .	77
5.2.1.	Open Food Facts . . . . .	77
5.2.2.	SQFLite . . . . .	78
5.2.3.	Local Notifications . . . . .	78
5.2.4.	Flutter Native TimeZone . . . . .	79
5.2.5.	Flutter Localizations . . . . .	79
5.2.6.	Flutter Charts . . . . .	79
5.3.	Iteraciones Realizadas . . . . .	80
5.3.1.	Iteración 0 . . . . .	81
5.3.2.	Iteración 1 . . . . .	82
5.3.2.1.	Elaboración e Implementación de la Base de Datos . . . . .	82
5.3.2.2.	Gestión de Usuarios . . . . .	83
5.3.2.3.	Mediciones y Sugerencias . . . . .	85
5.3.2.4.	Configuración Inicial de Aplicación . . . . .	85
5.3.3.	Iteración 2 . . . . .	86
5.3.3.1.	Gestión de Mediciones . . . . .	86
5.3.3.2.	Detalles de Medición: Sugerencias y Medicinas . . . . .	87
5.3.4.	Iteración 3 . . . . .	87
5.3.4.1.	Creación de Gráficos . . . . .	87
5.3.4.2.	Cambio de Usuario . . . . .	88
5.3.4.3.	Apartado de Nutrición . . . . .	89

5.3.4.4.	Sistema de Notificaciones . . . . .	90
5.3.5.	Iteración 4 . . . . .	90
5.4.	Informe de Pruebas . . . . .	90
<b>6.</b>	<b>Conclusiones y Líneas Futuras</b>	<b>93</b>
6.1.	Conclusiones . . . . .	93
6.2.	Líneas Futuras . . . . .	94
<b>Apéndice A.</b>	<b>Manual de Usuario</b>	<b>101</b>
A.1.	Introducción y Conceptos . . . . .	102
A.2.	¿Cómo acceder a Mesuri? . . . . .	102
A.3.	Funcionalidades . . . . .	103
A.3.1.	Inicio y Primera Vista . . . . .	103
A.3.2.	Perfil de Usuario: Medicinas y Sugerencias . . . . .	104
A.3.2.1.	Edición de Perfil . . . . .	105
A.3.2.2.	Listado de Medicinas . . . . .	107
A.3.2.3.	Listado de Sugerencias . . . . .	109
A.3.3.	Sugerencias / Recomendaciones . . . . .	109
A.3.3.1.	Detalles de una Sugerencia . . . . .	109
A.3.3.2.	Adición / Edición . . . . .	110
A.3.4.	Mediciones . . . . .	113
A.3.4.1.	Detalles de Medición . . . . .	113
A.3.4.2.	Adición / Edición de Medición . . . . .	115
A.3.4.3.	Borrado de Mediciones . . . . .	117
A.3.5.	Gráficos y Estadísticas . . . . .	117
A.3.6.	Otros . . . . .	119
A.3.7.	Usuarios . . . . .	120
A.3.7.1.	Adición de Usuario . . . . .	120
A.3.7.2.	Intercambio de Usuario . . . . .	120
A.3.7.3.	Eliminación de un Usuario . . . . .	121
A.3.8.	Notificaciones . . . . .	122
A.3.8.1.	Adición de una Notificación . . . . .	123

A.3.8.2.	Borrado de una Notificación . . . . .	123
A.3.9.	Nutrición . . . . .	124
A.3.9.1.	Búsqueda de Alimentos . . . . .	125
A.3.9.2.	Consulta de Información Nutricional de un Alimento . . . . .	126
A.4.	Detalles Adicionales . . . . .	127
A.4.1.	Soporte para Diferentes Idiomas . . . . .	127
A.4.2.	Soporte para Modo Oscuro . . . . .	128
A.4.3.	Adaptación al Tamaño de Texto del Dispositivo . . . . .	129
A.5.	Consideraciones Finales . . . . .	129
 <b>Apéndice B. Manual de Instalación</b>		<b>131</b>
B.1.	Aspectos Importantes . . . . .	132
B.2.	Recursos . . . . .	132
B.3.	Instalación . . . . .	133
B.3.1.	Ejecución del Código Fuente . . . . .	133
B.3.1.1.	Configuración de la Base de Datos . . . . .	133
B.3.1.2.	Ejecución en Emulador . . . . .	134
B.3.1.3.	Ejecución en Dispositivo Físico . . . . .	134
B.3.1.4.	Ejecución de la Aplicación . . . . .	135
B.3.2.	Instalación de una APK . . . . .	135
 <b>Apéndice C. Documento General de Requisitos</b>		<b>137</b>
C.1.	Introducción . . . . .	139
C.1.1.	Historial de Versiones . . . . .	139
C.1.2.	Información del Proyecto . . . . .	139
C.1.3.	Aprobaciones . . . . .	139
C.1.4.	Objetivo del Documento . . . . .	140
C.1.5.	Alcance del Producto . . . . .	140
C.1.6.	Definiciones, Acrónimos y Abreviaturas . . . . .	140
C.1.7.	Resumen . . . . .	141
C.2.	Directivas del Proyecto . . . . .	143
C.2.1.	Oportunidades de Negocio . . . . .	143

C.2.2.	Descripción del Problema . . . . .	143
C.2.3.	Descripción del Producto . . . . .	143
C.3.	Descripción de Participantes y Usuarios . . . . .	145
C.3.1.	Resumen de los Participantes . . . . .	145
C.3.2.	Resumen y Entorno de los Usuarios . . . . .	145
C.3.3.	Perfiles de los Participantes . . . . .	146
C.3.3.1.	UMA . . . . .	146
C.3.3.2.	AEPD . . . . .	146
C.3.4.	Perfiles de Usuario . . . . .	147
C.3.4.1.	Usuario Temprano (menor a 16 años) . . . . .	147
C.3.4.2.	Usuario Estándar (16 a 55 años) . . . . .	147
C.3.4.3.	Usuario de Edad Avanzada (mayor a 55 años) . . . . .	148
C.3.5.	Alternativas y Competencia . . . . .	148
C.3.5.1.	Contour Diabetes App . . . . .	148
C.3.5.2.	Diabetes - Diario de Glucosa [42] . . . . .	150
C.4.	Visión General del Producto . . . . .	151
C.4.1.	Entorno de Despliegue . . . . .	151
C.4.1.1.	Entorno para la Implementación del Sistema Actual . . . . .	151
C.4.1.2.	Aplicaciones Colaboradoras . . . . .	151
C.4.1.3.	Paquetes Comerciales . . . . .	151
C.4.2.	Suposiciones y Dependencias . . . . .	152
C.4.2.1.	Factores Externos que tienen un Efecto en el Producto . . . . .	152
C.4.2.2.	Suposiciones que asume el Equipo en torno al Proyecto . . . . .	152
C.4.3.	Precio y Coste . . . . .	152
C.5.	Requisitos Funcionales . . . . .	154
C.5.1.	Datos de Usuario (USR) . . . . .	154
C.5.2.	Mediciones (MSR) . . . . .	155
C.5.3.	Información (INFO) . . . . .	156
C.5.4.	Notificaciones (NTF) . . . . .	156
C.5.5.	Alimentación y Nutrición (FOOD) . . . . .	156
C.6.	Casos de Uso (CU) . . . . .	157

C.6.1.	Actores . . . . .	157
C.6.2.	Casos de Uso . . . . .	157
C.6.2.1.	Datos de Usuario (USR) . . . . .	157
C.6.2.2.	Mediciones (MSR) . . . . .	169
C.6.2.3.	Información (INFO) . . . . .	175
C.6.2.4.	Notificaciones (NTF) . . . . .	176
C.6.2.5.	Alimentación / Nutrición (FOOD) . . . . .	178
C.7.	Diagramas de Secuencia . . . . .	180
C.7.1.	Datos de Usuario (USR) . . . . .	180
C.7.2.	Mediciones (MSR) . . . . .	194
C.7.3.	Información (INFO) . . . . .	201
C.7.4.	Notificaciones (NTF) . . . . .	202
C.7.5.	Alimentación / Nutrición (FOOD) . . . . .	205
C.8.	Requisitos No Funcionales . . . . .	206
C.8.1.	Requisitos de Aspecto . . . . .	206
C.8.1.1.	Interfaz . . . . .	206
C.8.2.	Requisitos de Facilidad de Uso y Aprendizaje . . . . .	206
C.8.2.1.	Facilidad de Uso . . . . .	206
C.8.2.2.	Facilidad de Aprendizaje . . . . .	207
C.8.3.	Requisitos de Funcionamiento . . . . .	207
C.8.3.1.	Requisitos de Velocidad . . . . .	207
C.8.3.2.	Requisitos de Seguridad Crítica . . . . .	207
C.8.3.3.	Requisitos de Precisión . . . . .	207
C.8.3.4.	Requisitos de Fiabilidad y Disponibilidad . . . . .	207
C.8.3.5.	Requisitos de Capacidad . . . . .	208
C.8.3.6.	Requisitos de Escalabilidad . . . . .	208
C.8.4.	Requisitos Operacionales . . . . .	208
C.8.4.1.	Entorno Físico . . . . .	208
C.8.4.2.	Entorno Tecnológico . . . . .	208
C.8.4.3.	Aplicaciones Recomendadas . . . . .	208
C.8.4.4.	Soporte . . . . .	208

C.8.5.	Requisitos de Mantenimiento y Portabilidad . . . . .	208
C.8.5.1.	Requisitos de Mantenimiento . . . . .	208
C.8.5.2.	Requisitos de Portabilidad . . . . .	208
C.8.6.	Requisitos de Seguridad . . . . .	209
C.8.7.	Requisitos Culturales y Políticos . . . . .	209
C.8.8.	Requisitos Legales . . . . .	209
C.9.	Requisitos de Documentación . . . . .	210
C.9.1.	Manual de Usuario . . . . .	210
C.9.2.	Ayuda en Línea . . . . .	210
C.9.3.	Guía de Instalación y Configuración . . . . .	210
C.10.	Atributos de las Características Funcionales . . . . .	211
C.10.1.	Esfuerzo . . . . .	211
C.10.2.	Riesgo . . . . .	211
C.10.3.	Estabilidad . . . . .	211
C.10.4.	Asignada A . . . . .	211

**Apéndice D. Documento General de Pruebas 213**

D.1.	Historial de Versiones . . . . .	214
D.2.	Información del Proyecto . . . . .	214
D.3.	Aprobaciones . . . . .	214
D.4.	Resumen Ejecutivo . . . . .	215
D.5.	Alcance de las Pruebas . . . . .	215
D.5.1.	Elementos de las Pruebas . . . . .	215
D.5.2.	Funcionalidades a Probar . . . . .	215
D.5.3.	Funcionalidades a no Probar . . . . .	218
D.5.4.	Enfoque de Pruebas (Estrategia) . . . . .	218
D.6.	Criterios de Aceptación y Rechazo . . . . .	219
D.6.1.	Criterios de Aceptación . . . . .	219
D.6.2.	Criterios de Rechazo . . . . .	219
D.6.3.	Criterios de Reanudación . . . . .	219
D.7.	Recursos . . . . .	220

D.7.1.	Requerimientos de Entornos - Hardware . . . . .	220
D.7.2.	Requerimientos de Entornos - Software . . . . .	220
D.7.3.	Personal . . . . .	220
D.8.	Pruebas Realizadas . . . . .	221
D.8.1.	PR-01: Acceso a la Aplicación . . . . .	221
D.8.2.	PR-02: Gestión de Usuarios . . . . .	221
D.8.3.	PR-03: Gestión de Medicinas . . . . .	222
D.8.4.	PR-04: Gestión de Sugerencias . . . . .	222
D.8.5.	PR-05: Gestión de Mediciones . . . . .	223
D.8.6.	PR-06: Construcción y Visualización de Gráficos . . . . .	223
D.8.7.	PR-07: Gestión de Notificaciones . . . . .	224
D.8.8.	PR-08: Búsqueda de Información Nutricional . . . . .	224
D.8.9.	PR-09: Revisión de Documentos . . . . .	225





# Índice de figuras

1.	Avance de la Diabetes por Grupos de Edad. 1993-2017 [1]	28
2.	Logo de Dart	33
3.	Logo de Flutter	34
4.	Logo de SQLite	34
5.	Logo de VSCode	35
6.	Logo de AndroidStudio	35
7.	Logo de VirtualBox	36
8.	Logo de MoonModeler	36
9.	Logo de Postman	36
10.	Logo DB Browser	37
11.	Git / GitHub Logo	37
12.	Microsoft Teams Logo	37
13.	Microsoft Word Logo	38
14.	MagicDraw Logo	38
15.	Evolución de la Diabetes(%). 1980 - 2014.	46
16.	Evolución de la Diabetes en España(%). 1993 - 2017. [1]	46
17.	Diabetes en el Futuro(%). 2019 - 2045. [25]	47
18.	Enfermedades Mortales. 2000-2019 [26]	47
19.	Encuesta [28]: ¿Conoces a Personas con Diabetes?	48
20.	Encuesta [28]: ¿Ha cambiado su alimentación al padecer Diabetes?	48
21.	Encuesta [28]: ¿Has visto a esta persona tomar mediciones de glucosa?	48
22.	Evolución en uso de Móviles y PC globalmente. 2009 - 2016 [30]	49
23.	Dispositivos de Acceso a Internet (%). 2020 [32]	50
24.	Venta de Disps. Móviles según su Sistema Operativo (%). 2010-2019 [33]	50
25.	Diagrama de Navegación de Mesuri	52
26.	Diagrama de clases de los modelos usados en la aplicación	54
27.	Diagrama de los servicios que gestionan Formularios y Base de Datos	56
28.	Modelo de los servicios para establecer la comunicación con la base de datos	57

29.	Clases que gestionan librerías incorporadas en la aplicación . . . . .	58
30.	Modelo de las clases que ayudan a la gestión de información y transformación de datos . . . . .	59
31.	Clases para gestionar el sistema de navegación de la aplicación . . . . .	60
32.	Colores usados en la aplicación declarados como constantes . . . . .	61
33.	Géneros declarados como constantes enteras para su procesamiento . . . . .	61
34.	Icono de Mesuri . . . . .	62
35.	Símbolo Mundial de la Diabetes . . . . .	62
36.	Comparación entre Modo Claro y Oscuro . . . . .	63
37.	Comparación entre tamaño de letra S y XL . . . . .	63
38.	Estructura General Completa del Proyecto . . . . .	64
39.	Estructura del Proyecto. Directorio ' <i>assets</i> ' . . . . .	65
40.	Estructura del Proyecto. Directorio ' <i>lib</i> ' . . . . .	65
41.	Estructura del Proyecto. Directorio ' <i>config</i> ' . . . . .	66
42.	Estructura del Proyecto. Directorio ' <i>src</i> ' . . . . .	66
43.	Estructura del Proyecto. Directorio ' <i>models</i> ' . . . . .	66
44.	Estructura del Proyecto. Directorio ' <i>screens</i> ' . . . . .	67
45.	Estructura del Proyecto. Directorio ' <i>utils</i> ' . . . . .	67
46.	Modelo de Base de Datos. Última versión. . . . .	71
47.	Logo de <i>OpenFoodFacts</i> . . . . .	78
48.	Diagrama de Gantt del Trabajo Realizado . . . . .	80
49.	Icono y Pantalla de Carga de Mesuri . . . . .	103
50.	Barra de Navegación Principal . . . . .	104
51.	Acciones desde el Perfil del Usuario en Mesuri . . . . .	104
52.	Edición de Usuario. Datos Obligatorios y Opcionales . . . . .	105
53.	Edición de Usuario. Acciones en Datos Obligatorios . . . . .	106
54.	Edición de Usuario. Ventana de Selección de Icono . . . . .	106
55.	Edición de Usuario. Selección de Diabetes . . . . .	106
56.	Edición de Usuario. Selección de Fecha . . . . .	106
57.	Edición de Usuario. Acciones en Datos Opcionales . . . . .	107
58.	Listado de Medicinas . . . . .	108

59.	Gestión de Medicinas. Acciones . . . . .	108
60.	Gestión de Medicinas. Edición . . . . .	108
61.	Gestión de Medicinas. Borrado . . . . .	108
62.	Listado de Sugerencias . . . . .	109
63.	Detalles de Sugerencia. Acciones . . . . .	110
64.	Detalles de Sugerencia. Borrado . . . . .	110
65.	Gestión de Sugerencias. Formulario Principal . . . . .	111
66.	Sugerencia. Gestión de Medicinas . . . . .	111
67.	Sugerencia. Gestión de Medicinas. Acciones . . . . .	111
68.	Sugerencia. Gestión de Medicinas. Edición . . . . .	112
69.	Sugerencia. Gestión de Medicinas. Borrado . . . . .	112
70.	Registro de Mediciones. Listado Vacío . . . . .	113
71.	Registro de Mediciones. Listado con Elementos . . . . .	113
72.	Mediciones. Vista en Detalle . . . . .	114
73.	Medición en Detalle. Vista de las Sugerencias . . . . .	114
74.	Medición en Detalle. Gestión de Medicinas Tomadas . . . . .	114
75.	Formulario de Medición. Datos Obligatorios . . . . .	115
76.	Formulario de Medición. Listado de Síntomas . . . . .	115
77.	Formulario de Medición. Datos Opcionales . . . . .	116
78.	Mediciones. Borrado de una Medición . . . . .	117
79.	Estadísticas. Diagrama de Sectores de Tiempo . . . . .	118
80.	Estadísticas. Diagrama Comparativo de los Últimos Meses . . . . .	118
81.	Estadísticas. Falta de Mediciones . . . . .	118
82.	Otros. Listado de Opciones . . . . .	119
83.	Usuarios. Listado . . . . .	120
84.	Usuarios. Intercambio de Usuario Principal . . . . .	121
85.	Usuarios. Eliminación de un Usuario . . . . .	122
86.	Notificaciones. Listado . . . . .	122
87.	Notificaciones. Adición de una Notificación . . . . .	123
88.	Notificaciones. Borrado de una Notificación . . . . .	124
89.	Nutrición. Ventana Principal . . . . .	124

90.	Nutrición. Búsqueda de Alimentos . . . . .	125
91.	Nutrición. Búsqueda de Alimentos en Varias Páginas . . . . .	126
92.	Nutrición. Búsqueda sin Resultados . . . . .	126
93.	Nutrición. Información Detallada de un Alimento . . . . .	126
94.	Multilenguaje. Comparación del Perfil . . . . .	127
95.	Multilenguaje. Comparación de Estadísticas . . . . .	127
96.	Modo Oscuro. Comparación del Perfil . . . . .	128
97.	Modo Oscuro. Comparación de Estadísticas . . . . .	128
98.	Tamaño de Texto. Comparación del Perfil . . . . .	129
99.	Tamaño de Texto. Comparación de Nutrición . . . . .	129
100.	Instalación. Configuración de la Base de Datos . . . . .	134
101.	Proceso de Instalación de una APK. Pasos a seguir en el Dispositivo . . . . .	136
102.	ContourApp. Principal . . . . .	149
103.	CountourApp. Añadir Medición . . . . .	149
104.	DiabetesApp. Principal . . . . .	150
105.	DiabetesApp. Listado de Registros . . . . .	150
106.	CU-USR: Diagrama de Casos de Uso . . . . .	157
107.	CU-MSR: Diagrama de Casos de Uso . . . . .	169
108.	CU-INFO: Diagrama de Casos de Uso . . . . .	175
109.	CU-NTF: Diagrama de Casos de Uso . . . . .	176
110.	CU-FOOD: Diagrama de Casos de Uso . . . . .	178
111.	Diag. Secuencia de Acceso al Listado de Usuarios (CU-USR-01) . . . . .	181
112.	Diag. Secuencia de Intercambio de Usuario (CU-USR-02) . . . . .	182
113.	Diag. Secuencia de Registro de Usuarios (CU-USR-03) . . . . .	183
114.	Diag. Secuencia de Eliminación de Usuario (CU-USR-04) . . . . .	184
115.	Diag. Secuencia de Consulta de Datos de un Usuario (CU-USR-05) . . . . .	185
116.	Diag. Secuencia de Edición de Datos del Perfil (CU-USR-06) . . . . .	186
117.	Diag. Secuencia de Consulta del Listado de Medicinas (CU-USR-07) . . . . .	187
118.	Diag. Secuencia de Gestión de Medicinas (CU-USR-08, CU-USR-09, CU-USR-10)	188
119.	Diag. Secuencia de Consulta del Listado de Recomendaciones (CU-USR-11) . .	189
120.	Diag. Secuencia de Adición de Sugerencia (CU-USR-12) . . . . .	190

121.	Diag. Secuencia de Consulta de una Sugerencia Concreta (CU-USR-13) . . . . .	191
122.	Diag. Secuencia de Edición de Sugerencia (CU-USR-14) . . . . .	192
123.	Diag. Secuencia de Borrado de Sugerencia (CU-USR-15) . . . . .	193
124.	Diag. Secuencia de Acceso al Listado de Mediciones (CU-MSR-01) . . . . .	194
125.	Diag. Secuencia de Adición de una Medición (CU-MSR-02) . . . . .	195
126.	Diag. Secuencia de Consulta Detallada de una Medición (CU-MSR-03) . . . . .	196
127.	Diag. Secuencia de Edición de una Medición (CU-MSR-04) . . . . .	197
128.	Diag. Secuencia de Borrado de una Medición (CU-MSR-05) . . . . .	198
129.	Diag. Secuencia de Consulta de Sugerencias para una Medición (CU-MSR-06) .	198
130.	Diag. Secuencia de Consulta de Mediciones Tomadas para una Medición (CU-MSR-07) . . . . .	199
131.	Diag. Secuencia de Gestión de Mediciones Tomadas en una Medición (CU-MSR-08) . . . . .	200
132.	Diag. Secuencia de Visualización de Gráficas (CU-INFO-01) . . . . .	201
133.	Diag. Secuencia de Consulta de Notificaciones Programadas (CU-NTF-01) . . .	202
134.	Diag. Secuencia de Programar una Nueva Notificación (CU-NTF-02) . . . . .	203
135.	Diag. Secuencia de Borrado de una Notificación (CU-NTF-03) . . . . .	204
136.	Diag. Secuencia de Búsqueda de Información Nutricional (CU-FOOD-01) . . .	205



# Índice de tablas

1.	Modelo de Base de Datos. Tabla <i>Users</i> . Explicación. . . . .	72
2.	Modelo de Base de Datos. Tabla <i>Measures</i> . Explicación. . . . .	73
3.	Modelo de Base de Datos. Tabla <i>Medicines</i> . Explicación. . . . .	73
4.	Modelo de Base de Datos. Tabla Intermedia <i>Measures-Medicines</i> . Explicación. . . . .	74
5.	Modelo de Base de Datos. Tabla <i>Suggestions</i> . Explicación. . . . .	74
6.	Modelo de Base de Datos. Tabla Intermedia <i>Medicines-Suggestions</i> . Explicación. . . . .	75
7.	Modelo de Base de Datos. Tabla <i>Exercises</i> . Explicación. . . . .	75
8.	Modelo de Base de Datos. Tabla <i>Symptoms</i> . Explicación. . . . .	76
9.	Modelo de Base de Datos. Tabla <i>Notifications</i> . Explicación. . . . .	77
10.	DGR. Historial de Versiones Realizadas . . . . .	139
11.	DGR. Información del Proyecto . . . . .	139
12.	DGR. Aprobaciones del Proyecto . . . . .	139
13.	DGR. Participantes en el Proyecto . . . . .	145
14.	DGR. Usuarios en el Proyecto . . . . .	145
15.	DGR. Representante de la UMA (1) . . . . .	146
16.	DGR. Representante de la UMA (2) . . . . .	146
17.	DGR. Representante de AEPD . . . . .	146
18.	DGR. Perfil de Usuario Joven . . . . .	147
19.	DGR. Perfil de Usuario Estándar . . . . .	147
20.	DGR. Perfil de Usuario Mayor . . . . .	148
21.	DGR. Actores identificados en los Casos de Uso . . . . .	157
22.	CU-USR-01: Acceso al Listado de Usuarios . . . . .	158
23.	CU-USR-02: Intercambio de Usuario . . . . .	158
24.	CU-USR-03: Registro de Usuarios . . . . .	159
25.	CU-USR-04: Eliminación de Usuario . . . . .	160
26.	CU-USR-05: Consulta de Datos de un Usuario . . . . .	161
27.	CU-USR-06: Edición de Datos del Perfil . . . . .	162
28.	CU-USR-07: Consulta del Listado de Medicinas . . . . .	162



29.	CU-USR-08: Adición de Medicinas . . . . .	163
30.	CU-USR-09: Edición de Medicinas . . . . .	164
31.	CU-USR-10: Borrado de Medicinas . . . . .	165
32.	CU-USR-11: Consulta del Listado de Recomendaciones . . . . .	166
33.	CU-USR-12: Adición de Sugerencia . . . . .	167
34.	CU-USR-13: Consulta de una Sugerencia Concreta . . . . .	167
35.	CU-USR-14: Edición de Sugerencia . . . . .	168
36.	CU-USR-15: Borrado de Sugerencia . . . . .	169
37.	CU-MSR-01: Acceso al Listado de Mediciones . . . . .	170
38.	CU-MSR-02: Adición de una Medición . . . . .	171
39.	CU-MSR-03: Consulta Detallada de una Medición . . . . .	171
40.	CU-MSR-04: Edición de una Medición . . . . .	172
41.	CU-MSR-05: Borrado de una Medición . . . . .	173
42.	CU-MSR-06: Consulta de Sugerencias para una Medición . . . . .	173
43.	CU-MSR-07: Consulta de Medicinas Tomadas en una Medición . . . . .	174
44.	CU-MSR-08: Gestión de Medicinas Tomadas en una Medición . . . . .	174
45.	CU-INFO-01: Visualización de Gráficas . . . . .	175
46.	CU-NTF-01: Consulta de Notificaciones Programadas . . . . .	176
47.	CU-NTF-02: Programar una Nueva Notificación . . . . .	177
48.	CU-NTF-03: Borrado de una Notificación . . . . .	178
49.	CU-FOOD-01: Búsqueda de Información Nutricional . . . . .	179
50.	DGP. Versiones del Documento . . . . .	214
51.	DGP. Información del Proyecto . . . . .	214
52.	DGP. Aprobaciones del Proyecto . . . . .	214
53.	DGP. Simbología escogida para validar las pruebas . . . . .	218
54.	DGP. Pruebas de Acceso a la Aplicación (PR-01) . . . . .	221
55.	DGP. Pruebas de Revisión de Usuarios (PR-02) . . . . .	221
56.	DGP. Pruebas de Gestión de Medicinas (PR-03) . . . . .	222
57.	DGP. Pruebas de Gestión de Sugerencias (PR-04) . . . . .	223
58.	DGP. Pruebas de Gestión de Mediciones (PR-05) . . . . .	223
59.	DGP. Pruebas de Gráficos (PR-06) . . . . .	224

60.	DGP. Pruebas de Gestión de Notificaciones (PR-07)	224
61.	DGP. Pruebas de Información Nutricional (PR-08)	225
62.	DGP. Pruebas de Revisión de Documentos (PR-09)	225



# 1

# Introducción

En esta primera sección se pretende comentar aspectos de interés para la memoria de manera que se apoye con datos la finalidad del propio TFG así como detallar sus objetivos de una manera más extensa y precisa. Se añade un sumario de los entregables finales en este proyecto, así como de la estructura de este documento y todo lo que contiene.

## 1.1. Motivación

La diabetes es una enfermedad que aparece por diversos motivos. Se origina principalmente por problemas en el páncreas al no producir esta la cantidad adecuada de insulina o no haciendo un buen uso de la que se produce. Depende de cuál de estos motivos sea podemos distinguir entre diferentes tipos de diabetes, siendo los más conocidos los Tipo 1 o Tipo 2. Sin embargo, aunque la teoría sea la mencionada, la vida de cada una de las personas es propensa a padecer estas enfermedades en función de sus hábitos de vida. Si bien hay casos que se comienza a padecer como consecuencia de un embarazo u otra enfermedad como puede ser una hepatitis, la mayoría de veces se origina por el ritmo de vida personal. No sólo la mala alimentación influye como se podría llegar a pensar, sino que acompañada de una vida sedentaria puede ser causa directa de que una persona acabe padeciéndola a lo largo de su vida.

Debido a un aumento de casos en los últimos años y a la cantidad de problemas que una persona pueden llegar a producirse por una diabetes mal gestionada, el motivo del TFG es ayudar al paciente de la misma de manera que se brinde un entorno sencillo que resulte útil y fácil de manejar con el que llevar el registro diario de la enfermedad a través del desarrollo de una aplicación de carácter móvil en base a las decisiones tomadas gracias al estudio de mercado que se verá en las páginas siguientes; concretamente en la Sección 4.1.

Mediante el uso de tecnologías novedosas como Flutter o Dart se busca dar soporte a esta idea, mostrando de manera adicional la capacidad de dichas herramientas para ser implementadas en aplicaciones actuales.

## 1.2. Objetivos

El presente TFG pretende finalizar con el desarrollo de una aplicación móvil desde la que un usuario pueda consultar y añadir datos importantes en su evolución diaria de la diabetes independientemente del dispositivo y su sistema operativo. La gestión de la enfermedad y la ayuda al paciente diabético es el motivo principal de este proyecto.

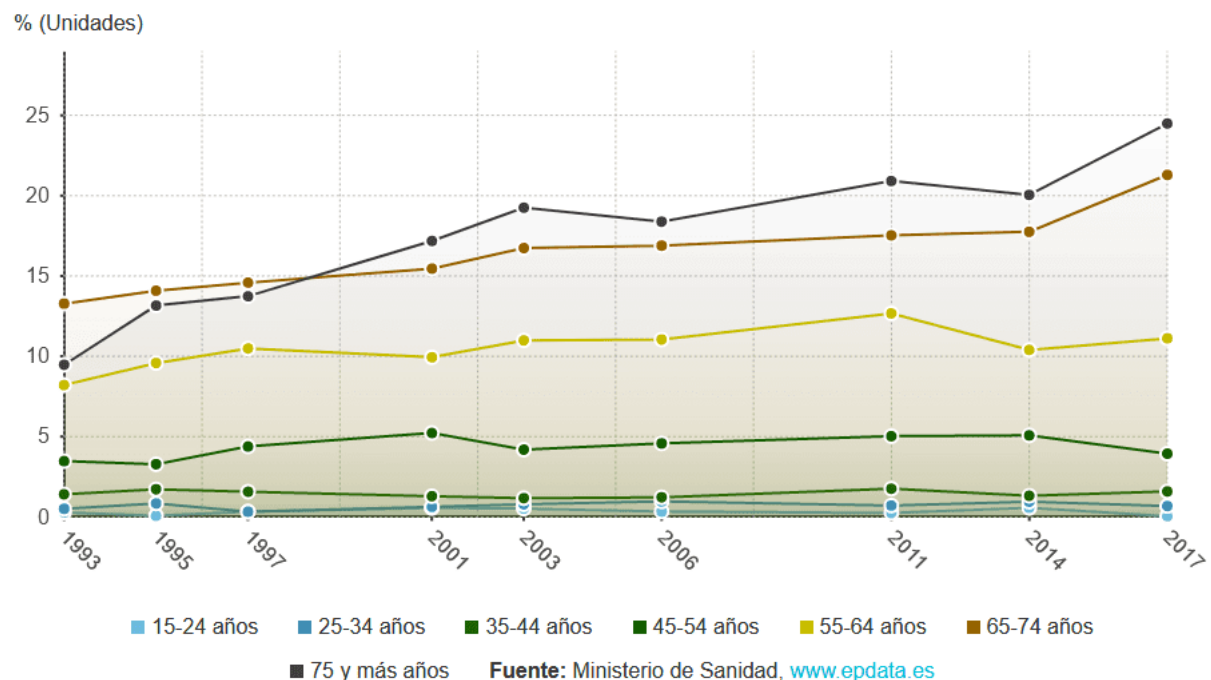


Figura 1: Avance de la Diabetes por Grupos de Edad. 1993-2017 [1]

Además, la aplicación pretende ser usable por el grupo mayoritario de personas que sufren diabetes como lo es el grupo compuesto por personas mayores a 65 años [Fig.1]. Debido a los problemas que pueden sufrir dichas personas, se pretende adaptar en la medida de lo posible la aplicación para que su usabilidad sea la mejor posible, suponiendo un nivel de aprendizaje y conocimiento básico para manejar la aplicación. Sumado a esto, se encuentran aspectos adicionales como:

- Proporcionar conocimiento acerca de ciertas comidas y elaboraciones que faciliten la elección de una dieta saludable para el usuario.
- La gestión de varios usuarios desde un mismo dispositivo haciendo posible el control de algún familiar o persona cercana.

Toda esta información concierne a la aplicación desarrollada que, de manera concisa, permite al usuario gestionar su información diaria y obtener con ella datos para controlar la enfermedad a fin de que su salud sea la mejor posible.

De manera adicional, la aplicación pretende mostrar la ventaja de un lenguaje y tecnología novatas en este sector como lo son Flutter y Dart. Tecnologías tempranas que, aun sin usarse ni enseñarse en el desempeño del grado, han destacado por su variedad de opciones y facilidad para el desarrollo no sólo de aplicaciones móviles, sino por su rápida extensión a páginas web o aplicaciones de escritorio. Con el fin de aprender esta materia emergente y trabajando en un mercado cada vez más amplio como el móvil, se desempeña el siguiente TFG dando lugar a un proceso de Ingeniería de Software completo que abarque todas sus fases en una metodología ágil que, más detalladamente, se explicará en la Sección 3.

### **1.3. Estructura del Documento**

El presente documento dispone de varios capítulos (o secciones) donde se tratarán aspectos concretos de todo el proceso abarcado para la realización de la aplicación en cuestión. A lo largo del mismo se encontrarán diversas figuras y tablas que servirán de ejemplo y modelo para validar la información mostrada en la misma. Los contenidos con los que contará esta memoria abarcarán:

#### **1. Introducción**

Se introduce el proyecto con un resumen de lo que ofrece, así como sus objetivos y motivo de existencia. Se añade la actual estructura del documento así como una breve explicación de los entregables que acompañarán al mismo.

#### **2. Tecnologías y Herramientas**

En este capítulo se tratará con carácter informativo las tecnologías usadas para la ela-

boración del proyecto. También se tratarán las herramientas para la realización de la aplicación, incluyendo su gestión, desarrollo y documentación en todos sus niveles a fin de conocer los requerimientos de cada una de las partes.

### **3. Metodología**

Una breve aclaración sobre el método de trabajo establecido desde el origen de la idea hasta llegar al final del desarrollo que acabará con la entrega de la memoria redactada. Tareas realizadas, metodología y trabajos similares tenidos en cuenta para el proyecto.

### **4. Análisis y Modelado**

Contará con un primer acercamiento a la documentación previa al desarrollo del proyecto como lo es la obtención y análisis de requisitos o el desarrollo de los casos de uso presentes en la aplicación. Se añaden también diagramas de navegación, clase y secuencia. De manera adicional, y como parte del diagrama de navegación, se incorpora la visión de la interfaz gráfica del producto final a fin de ayudar y evaluar la misma.

### **5. Desarrollo e Implementación**

Se tratará de explicar la estructura interna de la aplicación desarrollada a través de modelos como el de la base de datos usada y añadiendo información sobre las *Application Programming Interfaces* (API) usadas y cómo se aplicarán. Las iteraciones planeadas en el proyecto se descompondrán aquí en un intento por explicar cada una de las tareas realizadas en ellas y dando un orden al desarrollo.

### **6. Conclusiones y Líneas Futuras**

Una vez finalizado el proyecto se expondrán aquí diversas críticas y aclaraciones sobre el mismo con lo que se ha realizado y no se ha explicado en ninguna otra sección o ampliando la misma. Se valorará la importancia del proyecto en un futuro atendiendo a sus posibles actualizaciones e innovaciones dentro del sector.

### **7. Bibliografía**

Para finalizar se mostrarán todos los enlaces y documentos que han sido de utilidad a lo largo de la memoria y que aparecen citados como tal. Servirá así de apoyo a la explicación de conceptos e información provista. En caso de que sea posible se proporcionará acceso a dichos documentos.

Como complemento a toda esta información se adherirá al final del documento una sección dedicada exclusivamente a apéndices con información adicional que, a pesar de constituir un documento independiente, se añaden a este proporcionando datos sobre diferentes informes y manuales realizados sobre la aplicación como puede ser un Documento General de Requisitos (DGR), un manual de usuario o un resumido Informe de Pruebas de la misma.

## 1.4. Sumario de Entregables

Como se ha mencionado previamente, el desarrollo de este proyecto pretende ser una elaboración al completo realizada siguiendo el marco de trabajo y procesos propios de la Ingeniería Software. Por este motivo, los entregables proporcionados, algunos añadidos como apéndices, como conclusión del proyecto serán:

- **Producto Software** que actuará como conclusión del proyecto. Una aplicación móvil resultado de la elaboración de todo el proceso de desarrollo plasmado en la memoria de manera que pueda probarse.
- **Código Fuente** de la aplicación, permitiendo consultar aspectos de diseño interno, elaboración y recursos usados que, aunque ya detallados en la memoria, permitan evaluar la aplicación de conceptos al proyecto.
- **Bases de Datos** preparadas para poder usar la aplicación. Se incluirán dos archivos diferentes que iniciarán la aplicación en diferentes modos. Estos serán:
  - ◇ ***bd\_empty.db***: Esta estructura es la que usa la aplicación cuando se instala de manera normal. Contiene un usuario por defecto, sin nada más en su interior.
  - ◇ ***bd\_demo.db***: Base de datos que contiene datos suficientes como para emular un uso temporal de la aplicación. Contiene diferentes usuarios, así como medicinas y mediciones de algunos de ellos para visualizar posibles escenarios de la aplicación.
- **Instaladores (APK)** con los que poder instalar la aplicación directamente en un dispositivo Android cualquiera. Se añaden dos archivos de este tipo:
  - ◇ ***mesuri\_init.apk***: Aplicación por defecto que emula una instalación desde la *Play Store*. Usa la base de datos *bd\_empty.db*.



- ◊ *mesuri\_demo.apk*: Instalador que emula la aplicación después de un tiempo en funcionamiento. Usa la base de datos *bd\_demo.db*.
- **Memoria** actual con la que se proporcionará toda la información disponible y necesaria para entender cómo se ha realizado durante todo el proceso, mostrando cualquier aspecto que el alumno haya visto adecuado reflejar.
- **Documento General de Requisitos (DGR)**. Un apéndice más que contará con todo el proceso de requisitos de la aplicación y que mostrará de manera más detallada todo el proceso de obtención de los mismos. Contendrá la información referente a casos de uso (CU) y diagramas de secuencia (DS) relacionados.
- **Documento General de Pruebas (DGP)**. Un apéndice donde se evaluará cómo la aplicación cumple (o no) los requisitos definidos para la misma, mostrando procesos que se han realizado para comprobarlo. Este se abarcará de manera breve.
- **Manual de Instalación** de la aplicación. Apéndice que incluirá una explicación con los diferentes métodos para probar la aplicación, desde el propio código o a través de una APK instalable en un dispositivo.
- **Manual de Usuario** que mostrará de manera sencilla y mediante un flujo de acciones básico cómo realizar las diferentes acciones dentro de la aplicación. Esto permitirá reflejar cómo la aplicación puede usarse por usuarios inexpertos contribuyendo así a un manejo más rápido e intuitivo de la misma. Se añadirá como apéndice.

# 2

## Tecnologías y Herramientas

Un capítulo centrado en las tecnologías elegidas para el desarrollo de la aplicación y las diferentes herramientas que se usarán en otras áreas del proyecto como la gestión o la documentación.

### 2.1. Tecnologías

#### 2.1.1. Dart



Figura 2: Logo de Dart

Dart [2] es un lenguaje de programación *open-source* diseñado por Google con la intención de agilizar y facilitar la programación en diferentes entornos (móvil, web, escritorio, etc.). Aunque permite usarse con diferentes frameworks como AngularDart, a día de hoy se ha visto conocido principalmente por ser el lenguaje usado en Flutter.

Entre sus características principales destaca su simpleza y semejanza a lenguajes ya existentes como Java, JavaScript o C++, permitiendo entre otras cosas crear variables dinámicas que se adapten al tipo del valor que le asignemos o, lo que se considera su mayor logro, permitir desarrollos nativos en los sistemas donde se implementa. Con los años ha incluido soporte de numerosas librerías que le permiten actuar de manera independiente en multitud de proyectos y ha recibido soporte para trabajar con lenguajes más comunes como lo son SQL y XML.

### 2.1.2. Flutter

Flutter [3] es un framework de código abierto diseñado por Google con la característica principal de crear aplicaciones nativas para cualquier entorno de una manera sencilla. Basa toda su actuación en esta idea proporcionando un rendimiento casi perfecto en cualquier plataforma que se decida, independientemente de la que sea. De esta manera le proporciona capacidad para evitar el desarrollo de diferentes aplicaciones pudiendo de una sola vez, por ejemplo, obtener una aplicación Android e iOS.



Figura 3: Logo de Flutter

De manera adicional cabe destacar que posee diferentes versiones para cada uno de los entornos, aunque el más avanzado actualmente es el móvil (uno de los motivos para desarrollar la aplicación en este entorno). En un carácter más general, proporciona gran variedad de widgets que poder aplicar al diseño gráfico, así como la posibilidad de incorporar paquetes y otros añadidos que surgen dentro de la comunidad de usuarios.

### 2.1.3. SQLite



Figura 4: Logo de SQLite

Como sistema gestor de base de datos se ha seleccionado SQLite [4]. Este entorno proporciona una base de datos que permite gestionar de manera sencilla y local una base de datos SQL. Este motor de bases de datos es muy usado en entornos móviles, por lo que Flutter adapta su uso de manera nativa a través de un paquete conocido como *sqflite* [5] que puede añadirse al proyecto fácilmente.

La elección de la misma se ha debido a que no es necesario que los datos de usuario se almacenen en una plataforma cloud debido a la naturaleza de la aplicación. Los datos podrían ser considerados de carácter delicado y dado que no hay un red de aplicaciones web que se conecten a la aplicación, la elección de una base de datos local parece la mejor opción.

## 2.2. Herramientas de Desarrollo

### 2.2.1. Visual Studio Code



Figura 5: Logo de VSCode

Visual Studio Code (VSCode) [6] es un editor de código abierto con soporte para la gran mayoría de lenguajes. Permite adherir gran cantidad de plugins y complementos que facilitan sugerencias de código para cada uno de los lenguajes, además de integrar una consola y compilador propios.

Su incorporación en este proyecto se ha considerado gracias a que incorpora plugins tanto para Dart como para Flutter facilitando el arranque y testeado de la aplicación y permitiendo usar funciones propias de Flutter como el *hot-reload*, para visualizar los cambios programados rápidamente sin tener que compilar de nuevo la aplicación; o el uso de las DevTools, con las que ver el rendimiento y estructura de la aplicación durante su ejecución.

### 2.2.2. Android Studio



Figura 6: Logo de AndroidStudio

AndroidStudio [7] es un Entorno Integrado de Desarrollo (o IDE) pensado para elaborar aplicaciones nativas en el sistema operativo Android. Su uso en este proyecto no es el de IDE, sino que se limita a la creación de emuladores gracias al uso del SDK de Android. Si bien VSCode podría generar emuladores, el proceso es mucho más complejo que con AndroidStudio, ya que este último proporciona una interfaz gráfica para ello. Con él se podrán crear nuevos emuladores de distintos dispositivos para probar la aplicación en diferentes situaciones.

### 2.2.3. VirtualBox

VirtualBox [16] es un entorno que permite la creación de máquinas virtuales que permiten ejecutar así en cualquier equipo un sistema operativo independientemente del que tenga el equipo base. Su uso en el proyecto ha sido clave para poder ejecutar los emuladores creados con AndroidStudio y visualizar así la aplicación desarrollada sin perder tiempo en el momento de instalar y reiniciar la aplicación a cada cambio realizado.



Figura 7: Logo de VirtualBox

### 2.2.4. MoonModeler



Figura 8: Logo de MoonModeler

MoonModeler [17] es una herramienta de modelado de bases de datos que permite el desarrollo sencillo y visual de bases de datos tanto SQL como NoSQL. Su elaboración sencilla facilita la creación y posterior funcionamiento, pues ayuda a generar código entendible por los principales gestores como MongoDB, SQLite o MariaDB, entre otros. Ha sido clave para la modificación de la BD durante el desarrollo sin perder demasiado tiempo en su implementación.

### 2.2.5. Postman

Postman [18] provee una cantidad inmensa de operaciones posibles para la creación y comprobación de APIs realizando llamadas u ofreciendo datos. El propósito de su uso en la aplicación ha sido en complemento a *OpenFoodFacts*, la cual es una API explicada posteriormente a la que invocamos para obtener datos de alimentos, por lo que Postman ha sido clave para conocer su funcionamiento y datos que provee.



Figura 9: Logo de Postman

### 2.2.6. DB Browser

DBBrowser [19] está pensada para proporcionar un entorno gráfico sencillo para la creación, modelado y visualización de BBDD realizadas con SQLite. Una vez generada la BBDD, este programa ha sido usado para la gestión de la misma, comprobando la exactitud de las operaciones realizadas sobre ella.



Figura 10: Logo DB Browser

## 2.3. Herramientas de Gestión

### 2.3.1. Git / Github



Figura 11: Git / GitHub Logo

Git [8] es una herramienta que permite el control de versiones permitiendo restaurar una versión anterior o crear ramas alternativas de un mismo proyecto. Esta se usará en complemento con Github [9], un repositorio de código online (entre otras cosas) con el cual se dispondrá en todo momento de una copia segura en la nube con la que poder restaurar estas versiones

antiguas y generar seguridad en el mantenimiento del proyecto para evitar conflictos.

### 2.3.2. Microsoft Office: Teams



Figura 12: Microsoft Teams Logo

Microsoft Teams [10] es una zona de encuentro donde es posible organizar reuniones entre usuarios a fin de mejorar el trabajo entre usuarios. Su uso en este proyecto ha sido clave para organizar reuniones con los tutores y compartir archivos y cuestiones entre nosotros de manera que se pudieran solventar problemas de manera rápida y comunicativa.

## 2.4. Herramientas de Documentación

### 2.4.1. Microsoft Office: Word

Word [11] es una herramienta de procesamiento de textos creada por Microsoft donde elaborar documentos de manera sencilla y rápida. Esta aplicación se ha usado en este trabajo como bloc de notas para recopilar información que después se ha trasladado a LaTeX. Otra tarea realizada con Word ha sido la elaboración de apéndices que, posteriormente, han sido añadidos a este caso de estudio.



Figura 13: Microsoft Word Logo

### 2.4.2. LaTeX: MiKTeX con Texmaker

Latex [12] es un sistema por el cual se pueden generar documentos de alta calidad en diversos formatos como artículos, libros, reportes o tesis. Es similar a un lenguaje de programación en sí mismo. Como complemento se ha usado MiKTeX [13] que permite la actualización de paquetes y versiones LaTeX mediante una interfaz gráfica que simplifica estos pasos. Además, incorpora de base numerosos paquetes aplicables a nuestros textos en LaTeX.

Por último, se usa Texmaker [14] como editor para escribir y generar estos textos usando las herramientas que ofrece MiKTeX para compilar y trabajar con LaTeX.

### 2.4.3. MagicDraw

MagicDraw [20] es un software que proporciona un entorno y herramientas útiles para la generación de código y la elaboración de maquetas y modelos basados en el estándar UML. Su uso en el proyecto ha sido muy importante para la generación de los diferentes diagramas de casos de uso o secuencia, entre otros.



Figura 14: MagicDraw Logo

## **2.5. Herramientas Adicionales**

### **2.5.1. Microsoft Forms**

Microsoft Forms [15] es una herramienta de Microsoft que permite realizar y compartir cuestionarios online de manera sencilla suministrando un tratamiento de los resultados con gráficos y respuestas sencillas.

El uso en el proyecto ha sido útil para realizar la encuesta utilizada en el estudio de mercado obteniendo de ella datos e imágenes para realizar la memoria.





# 3

## Metodología

La metodología define el proyecto y planta las bases de actuación a lo largo de todo el proceso. En esta sección se explicarán las decisiones tomadas en este contexto y detallarán aspectos sobre ellas.

### 3.1. Enfoque

El estudio de mercado explicado en la Sección 4.1 establece el enfoque que recibirá el proyecto en su desarrollo, así como la justificación de las decisiones tomadas. El mercado actual se mueve cada vez más hacia las plataformas móviles con la intención de que el usuario permanezca conectado en todo momento. Esto no deja de lado las aplicaciones web o de escritorio, pero debido al límite de tiempo para el TFG del que se dispone se ha priorizado el diseño móvil en virtud del tipo de aplicación.

El usuario evita de esta forma disponer de un equipo portátil o sobremesa con el que actualizar su información, siendo una manera más rápida de actualizar y añadir registros a la aplicación.

### 3.2. Metodología

Se planteará para la realización del proyecto el uso de una metodología ágil en la que poder adaptar los cambios que aparezcan durante el desarrollo. En una tecnología temprana como lo es Flutter y Dart, los cambios de versión son habituales y continuos. Estos pueden provocar que algunas características o formas de realizar algo ya establecido cambie al momento. Esto puede provocar que los requisitos sufran variaciones durante el desarrollo y haya que adaptar elementos de código al nuevo método.

Como metodología se ha escogido Scrum [21] adaptada a la situación de un proyecto individual realizado por una única persona que ocupará todos los roles referentes al equipo de trabajo como:

- Analista.
- Diseñador.
- Desarrollador.
- Verificador.
- Documentalista.

El único rol que no se ocupará por el estudiante es el de *Product Owner*, cubierto por ambos tutores del proyecto. Ambos participarán en el proceso verificando a través de diferentes *sprints* el refinamiento de la aplicación y ampliando las funcionalidades del producto desarrollado. Esta verificación se realizará mediante reuniones quincenales con los tutores para evaluar el estado y decidir los siguientes pasos a realizar.

En todo momento se mantendrá esta metodología teniendo que escoger en cada iteración las funcionalidades a implementar, así como realizar y revisar las pruebas de las mismas. La documentación asociada se realizará de manera previa y final a la iteración a fin de tener siempre un acercamiento lo más exacto posible con el resultado que se busca.

### **3.3. Iteraciones Realizadas**

El proyecto pretende mostrar todo el proceso de Ingeniería Software aplicado a un proyecto. En este existen diferentes fases de trabajo que se tendrán detallarán a continuación y durante todo el documento.

#### **3.3.1. Tareas Previas**

Antes de comenzar con las fases preestablecidas por SCRUM se han realizado etapas previas con fines de formación y aprendizaje de las tecnologías usadas. Varias de ellas son:

- **Documentación y Análisis de Viabilidad:** La decisión de realizar la aplicación móvil, tanto en temática como tecnologías ha conllevado un trabajo de estudio del mercado que pudiera confirmar la idea inicial planteada.
- **Configuración y Aprendizaje de Tecnologías:** Flutter y Dart son tecnologías con guías de configuración fáciles de seguir, pero aunque están enfocadas a desarrollos rápidos, el poco conocimiento previo de ellas ha requerido el aprendizaje básico de las mismas hasta permitir empezar a trabajar.

### 3.3.2. Iteraciones Posteriores

Una vez realizado el trabajo previo se plantearán las etapas e iteraciones que se desarrollarán a lo largo del proyecto siguiendo el modelo de SCRUM. Estas iteraciones comenzarán y terminarán una tras otra en períodos de tiempo de quince días y revisando en cada una de ellas lo realizado en las anteriores. El objetivo final será el de cumplir con todos los requisitos planteados para la aplicación. En total se realizarán 5 iteraciones.

En la primera iteración, llamada Iteración 0, se recopilarán y analizarán los datos previos al desarrollo. Conocer qué se va a desarrollar a través del **análisis de requisitos** realizando el correspondiente Documento General de Requisitos (DGR) que aparecerá más adelante como apéndice de la memoria. Además de obtener los requisitos, se realizarán los correspondientes casos de uso de la aplicación y se establecerá cómo será su implementación a lo largo de las demás fases. En esta iteración también se realizarán los correspondientes modelos de la base de datos y de clases usadas en la aplicación, así como los mockups básicos relacionados con la interfaz del producto, correspondientes a la fase SCRUM de **diseño y arquitectura**.

La iteración 1 comenzará el trabajo con la aplicación una vez se disponga ya de un primer acercamiento a la base de datos que se usará. Se incorporará en esta iteración aspectos como la edición de usuarios, así como la gestión de sus medicinas y sugerencias. En cada uno de estos procesos se desarrollará la interfaz de las pantallas necesarias para cada una de estas acciones.

En la iteración 2, y con los aspectos de usuario finalizados, se comenzará la inclusión de

las respectivas mediciones de glucosa y toda su gestión en la base de datos. Aspectos como la gestión de síntomas, ejercicio físico, medicinas por medición se realizarán en esta parte.

Para la iteración 3 se realizarán algunos aspectos como la posibilidad y gestión de diferentes usuarios en la aplicación, así como el trabajo con funcionalidades adicionales como lo son el trabajo con la API de *OpenFoodFacts* y la realización del sistema de notificaciones.

En la última iteración se terminará de cerrar el proyecto revisando el cumplimiento de todos los requisitos y la aceptación de las pruebas realizadas. Se redactará la documentación restante que quede por elaborar y se prepararán los entregables finales del TFG.

Como se ha mencionado, en cada una de estas iteraciones se revisarán los requisitos y se obtendrá información sobre si están o no siendo cumplidos. De la misma forma se crearán en caso de necesidad nuevas versiones de los modelos cumpliendo con las variaciones que puedan sufrir aprovechando así la metodología ágil utilizada. En esta memoria sólo aparecerán las últimas versiones de cada uno de ellos.

### **3.4. Estudio de Trabajos Similares**

Al ser Flutter una tecnología novedosa no había trabajos que lo usaran como base para su realización. Esto ha hecho que la utilidad de los mismos se redujera. Sin embargo, sí que hay muchos que controlan tanto aplicaciones móviles. Estos son los que he usado como inspiración para el aspectos de la interfaz, principalmente. De manera adicional, para la realización de este proyecto se han revisado diversos TFG de años anteriores con la intención de conocer aspectos sobre la redacción, en concreto algunos de los disponibles en el portal RiUMA <sup>1</sup>. A fin de todo, los que he acabado usando y sirviendo de utilidad se comprenden en los los siguientes UNICAR: una plataforma para compartir vehículos entre alumnos y personal de la UMA [22] y Gestión de rutas turísticas para Málaga [23].

Me he centrado en estos por estar tutorizados por mi propio tutor, además de por los motivos explicados previamente.

---

<sup>1</sup>Acceso a RIUMA: <https://riuma.uma.es>

# 4

## Análisis y Modelado

Las primeras fases de la Ingeniería del Software son de gran importancia ya que aspectos como los requisitos o el modelado de la aplicación son el camino a seguir para la fase de desarrollo, en la cual toda la información recabada ayudará a elaborar de manera fiable la misma. En esta sección se tratarán todos los aspectos referentes a la documentación previa al desarrollo que tratará las fases de **Obtención y Análisis de Requisitos**, así como la posterior de **Diseño y Arquitectura del Proyecto**.

### 4.1. Estudio de Mercado

Previo a detallar los requisitos de la aplicación se realizará un estudio de mercado para dar a conocer el por qué de las decisiones tomadas en ámbitos del tema principal de la aplicación: la diabetes; así como la decisión de desarrollar una aplicación móvil como medio para ello.

#### 4.1.1. Información sobre la Diabetes

Si atendemos a los gráficos proveídos por la propia Organización Mundial de la Salud (OMS) que muestran la evolución de dicha enfermedad se puede ver cómo su incremento en los últimos 40 años ha sido muy grande pasando de unos 108 millones de casos conocidos en 1980 y llegando a los 422 millones en 2014 [24]. Cifra que no ha dejado de aumentar desde entonces. Esto nos deja en la situación de que una de cada 11 personas padecen esta dolencia.

En el caso de España, los datos no mejoran. Unos datos más actualizados suministrados por el Ministerio de Sanidad muestran que también se ha producido un aumento de casos en los últimos años pasando de un 4 % a un 7.8 % [Fig. 16] en el país.

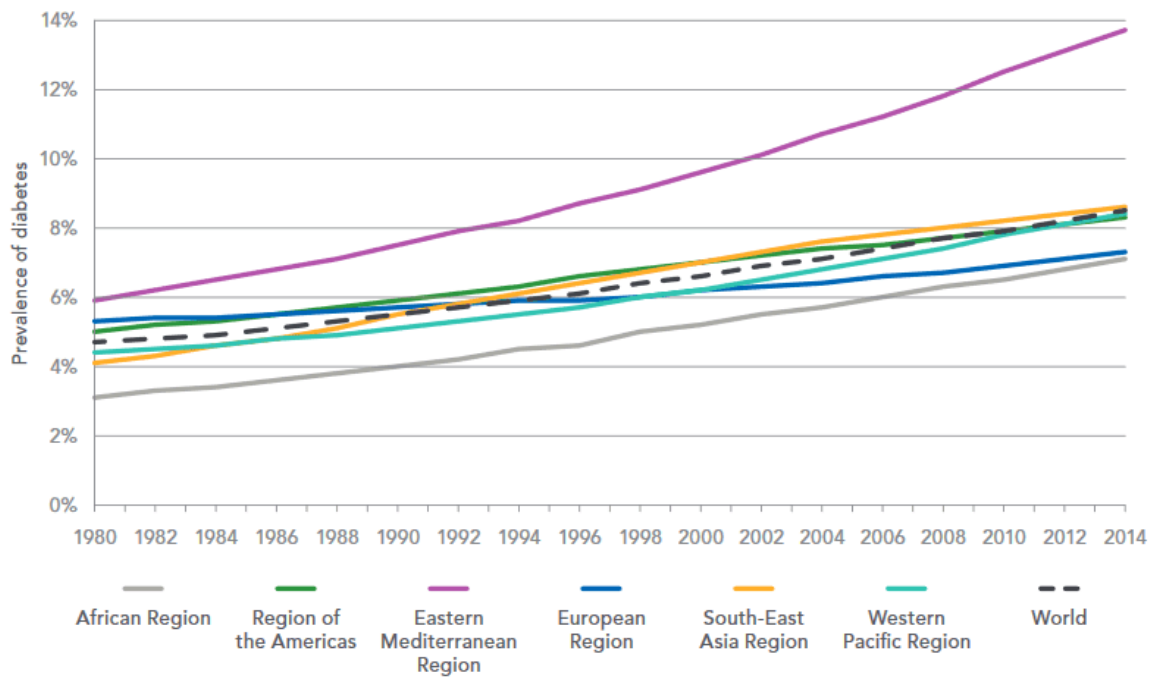


Figura 15: Evolución de la Diabetes(%). 1980 - 2014.

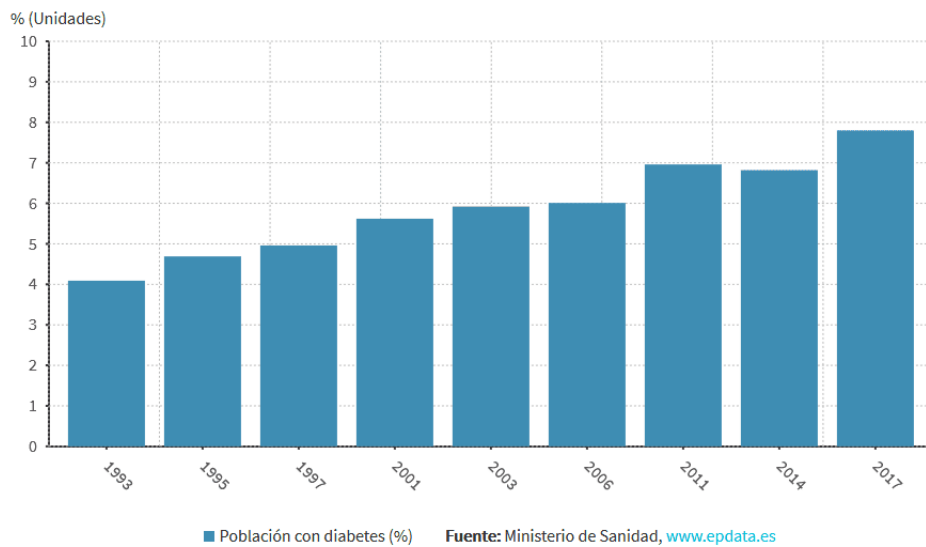


Figura 16: Evolución de la Diabetes en España(%). 1993 - 2017. [1]

Con estas estadísticas y acuerdo a diversos estudios realizados por la *International Diabetes Federation* (IDF) se prevee que aproximadamente 700 millones de personas se vean afectadas por esta enfermedad en torno al año 2045 aumentando así en un 51 % los casos respecto al día de hoy[Fig.17], lo que resulta en un dato preocupante si tenemos en cuenta que, de acuerdo a

la OMS, la diabetes ha sido la novena enfermedad que más muertes ha causado en este pasado año 2019[26].

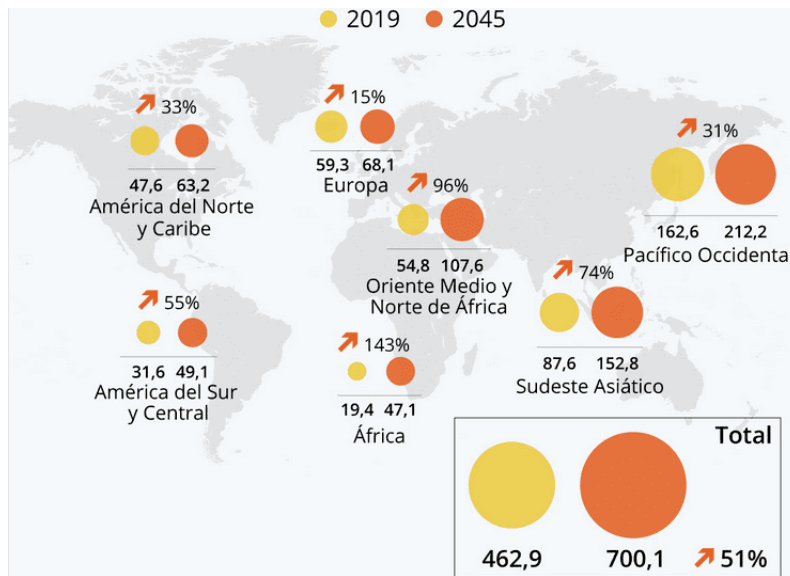


Figura 17: Diabetes en el Futuro(%). 2019 - 2045. [25]

Dado que los problemas de padecer esta enfermedad se pueden agravar si el tratamiento a la misma no es adecuado (ceguera, hambre intensa, sudor, fatiga...)[27], lo que motiva este TFG es proporcionar al usuario una herramienta sencilla y eficaz para evitar en la medida de lo posible casos de mala gestión que lleve a empeorar su situación.

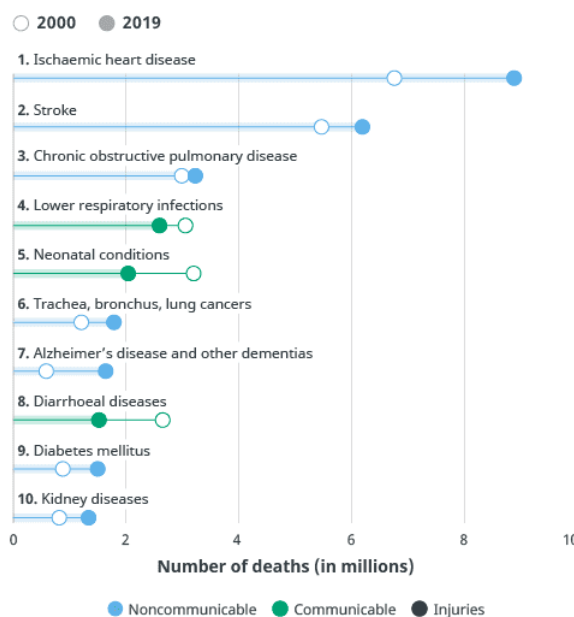


Figura 18: Enfermedades Mortales. 2000-2019 [26]



Adicionalmente, tras una encuesta realizada de manera anónima y a la que han respondido cerca de 80 personas [28] se ha obtenido información útil para este estudio. Si bien el tamaño de la muestra es pequeño, nos deja ver datos evidentes de la incidencia de la enfermedad en las personas que nos rodean. Tras una criba de las respuestas se puede apreciar como la gran mayoría conoce a personas que sufren diabetes [Fig. 19] a las que han visto cambiar su alimentación [Fig. 20] o tomar mediciones de glucosa en público para luego apuntarlas [Fig. 21]. Son datos que aunque pareciendo imprecisos dejan ver de manera sencilla la incidencia de la enfermedad en un entorno cercano a cualquier persona.

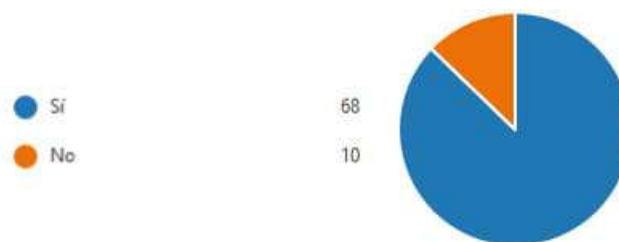


Figura 19: Encuesta [28]: ¿Conoces a Personas con Diabetes?



Figura 20: Encuesta [28]: ¿Ha cambiado su alimentación al padecer Diabetes?

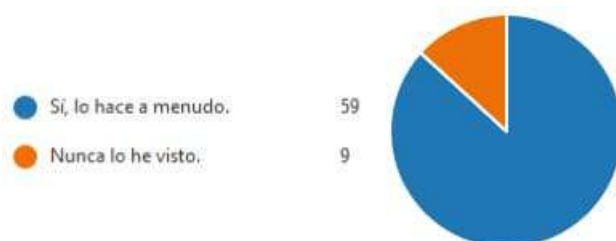


Figura 21: Encuesta [28]: ¿Has visto a esta persona tomar mediciones de glucosa?

A raíz de estas preguntas se han tomado decisiones acerca de varios de los objetivos de la aplicación como por ejemplo la adición de una sección en la que el usuario puede consultar la

información alimentaria de diversas comidas a través del uso de una API o, por otra parte, la incorporación de algunos datos que se almacenarán en la base de datos y que en un comienzo no estaban planteados. La unión de estos aspectos con la necesidad de llevar un registro han sido claves para tomar la decisión de la temática de la aplicación.

#### 4.1.2. Decisión de Aplicación Móvil

La decisión de llevar la aplicación exclusivamente al ámbito móvil proviene de diversos aspectos y en parte es influida por la elección de Dart y Flutter como herramientas principales para ello. Flutter es un framework de desarrollo con poca evolución en el mercado (apenas tres años). En este tiempo, ha centrado su evolución en el campo de las tecnologías móviles y adentrándose cada vez más en la posibilidad y compatibilidad de ser usado en ámbitos de página web o aplicación de escritorio. Sin embargo, este factor ha sido crucial para que se priorice la versión móvil respecto a las demás reduciendo así el número de problemas que una tecnología tan temprana pudiera provocar. Ha sido recientemente cuando, durante la revisión de esta memoria, ha surgido el anuncio de ambas plataformas de manera estable con Flutter 2 [29].

Dentro de esta información, otro factor a tener en cuenta es el deseo del usuario de estar conectado en todo momento lo que combina perfectamente con el carácter de la aplicación como registro de datos. El usuario puede estar fuera de casa cuando toma sus mediciones y no disponer así de un equipo, siendo más cómodo apuntar información en su dispositivo móvil. En términos estadísticos se puede ver también la tendencia al alta en el uso de dispositivos móviles para acceder a Internet frente al uso en PCs [Fig. 22].

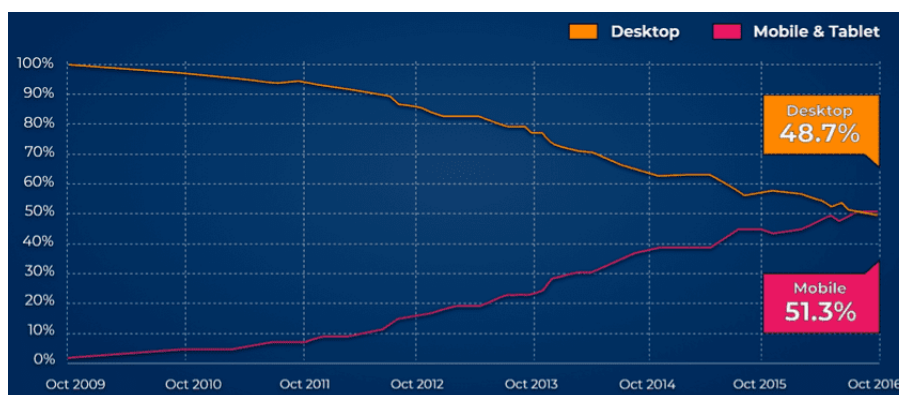


Figura 22: Evolución en uso de Móviles y PC globalmente. 2009 - 2016 [30]

Un dato que permanece aún en 2020, al menos en España, donde un confinamiento y un acercamiento al teletrabajo después ha provocado que el móvil sea el gran beneficiado de esta pelea [31] [32].

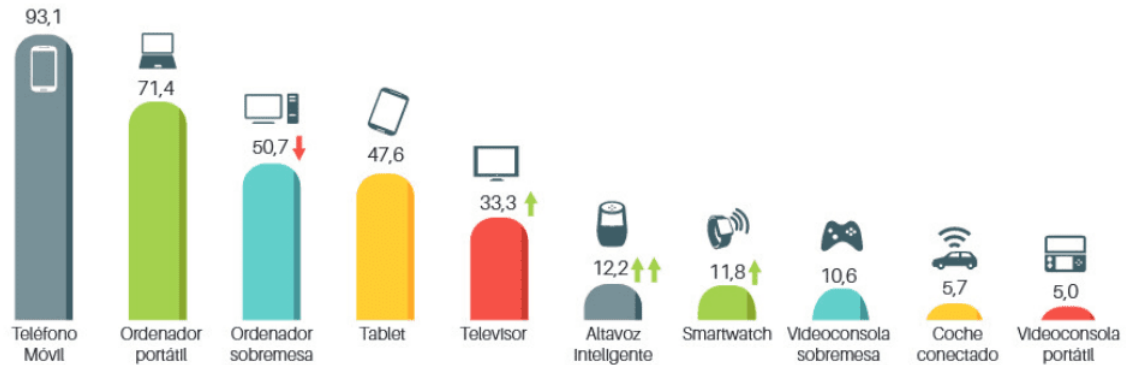


Figura 23: Dispositivos de Acceso a Internet (%). 2020 [32]

Además, es importante conocer el dispositivo desde el cual se accede, conociendo así que el desarrollo en Android e iOS no es el mismo. Dos sistemas suponen dos desarrollos, por lo que optamos a usar Flutter para el mismo garantizando así que una misma aplicación permita el despliegue de manera nativa en uno y otro dispositivo, abarcando así una gama amplia de dispositivos (casi su totalidad [Fig. 24]).

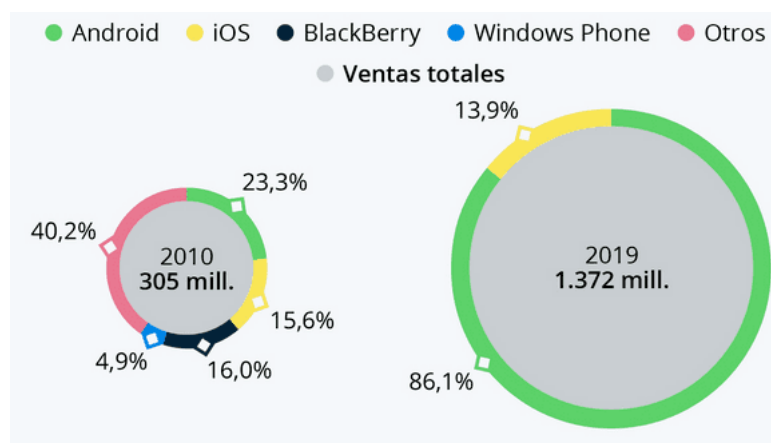


Figura 24: Venta de Disps. Móviles según su Sistema Operativo (%). 2010-2019 [33]

Estos factores son los que han provocado que la aplicación priorice el desarrollo móvil frente a otros dispositivos, así como el uso de las tecnologías elegidas para ello.

## 4.2. Requisitos Definidos

Como se ha mencionado previamente en el punto 1.2 correspondiente a los objetivos del proyecto, los requisitos de la aplicación pasan por desarrollar una manera eficaz y rápida, además de sencilla para que cualquier usuario, independientemente de sus habilidades para el manejo de dispositivos móviles pueda acceder y registrar todas sus mediciones, síntomas y medicinas que diariamente debe tomarse. Se reúnen con esto aspectos relativos a la información nutricional de diversos productos con los que el usuario querría informarse, así como la gestión de diferentes usuarios para entornos familiares o la incorporación de notificaciones para el aviso diario del usuario en la aplicación.

Todos estos requisitos, junto a sus respectivos casos de uso y diagramas de secuencia se encuentran en el **apéndice C**, correspondiente al **Documento General de Requisitos**, donde se explican con detalle cada uno de los requisitos y la interacción del usuario con la aplicación para cumplirlos.

## 4.3. Diagrama de Navegación

Durante la navegación, el usuario puede navegar por diferentes pantallas con las que interacciona para cumplir cada uno de los requisitos definidos. La navegación en esta se ha procurado precisa y sencilla, alcanzado un total de 21 interfaces diferentes que podrían considerarse pantallas. El flujo de navegación [Fig. 25] muestra los avances entre cada una de las pantallas, siendo posible ir hacia atrás en cualquiera de ellos.

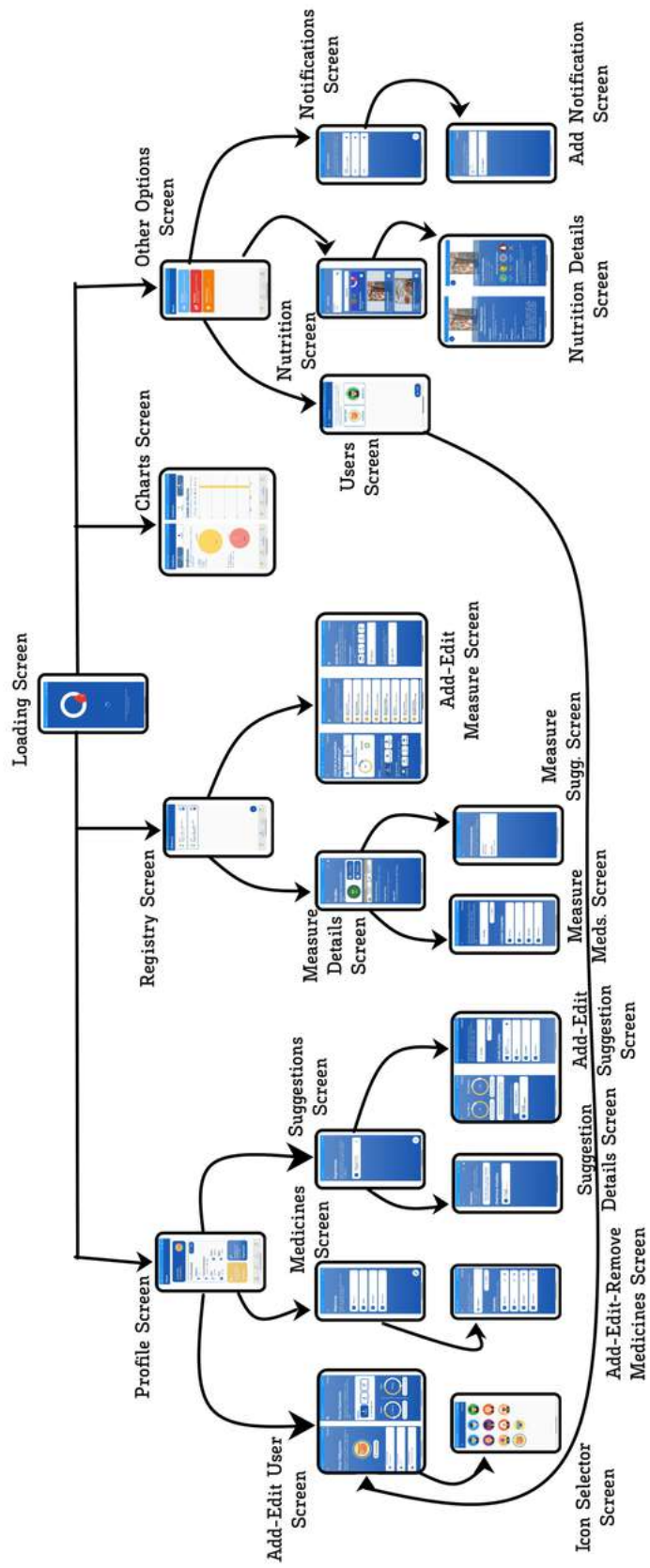


Figura 25: Diagrama de Navegación de Mesuri

## 4.4. Diagramas de Clases

Se han usado diferentes clases tanto para el manejo de objetos de la base de datos como para facilitar el funcionamiento de la aplicación. Esto ha hecho que existan no sólo modelos, sino también servicios para la interacción con el cliente o para el manejo de aspectos como el idioma o las notificaciones dentro de Flutter. Es importante mencionar aquí que los diagramas se encuentran realizados siguiendo el modelo UML, adheriendo los datos y tipos propios de Dart. Aclarado esto, podemos entender como clases:

### 4.4.1. Modelos de Objetos

Estos son los objetos que manejamos dentro de la aplicación y que provienen de la base de datos con la que se enlaza la aplicación. Cada uno de estos modelos contiene referencias a cada uno de los campos de sus respectivas tablas en la base de datos usada, así como un método para transformar objetos del formato que se recibe de las consultas (podría decirse que es un *JSON*) para adaptarlo a estos modelos. El proceso inverso también se encuentra contemplado. En algunos casos, especialmente en aquellos que actúan como objetos independientes se pueden encontrar constructores sin parámetros cuya referencia en la lógica de la aplicación creará una instancia donde todos los parámetros serán nulos. Se pueden considerar modelos básicos de la aplicación.

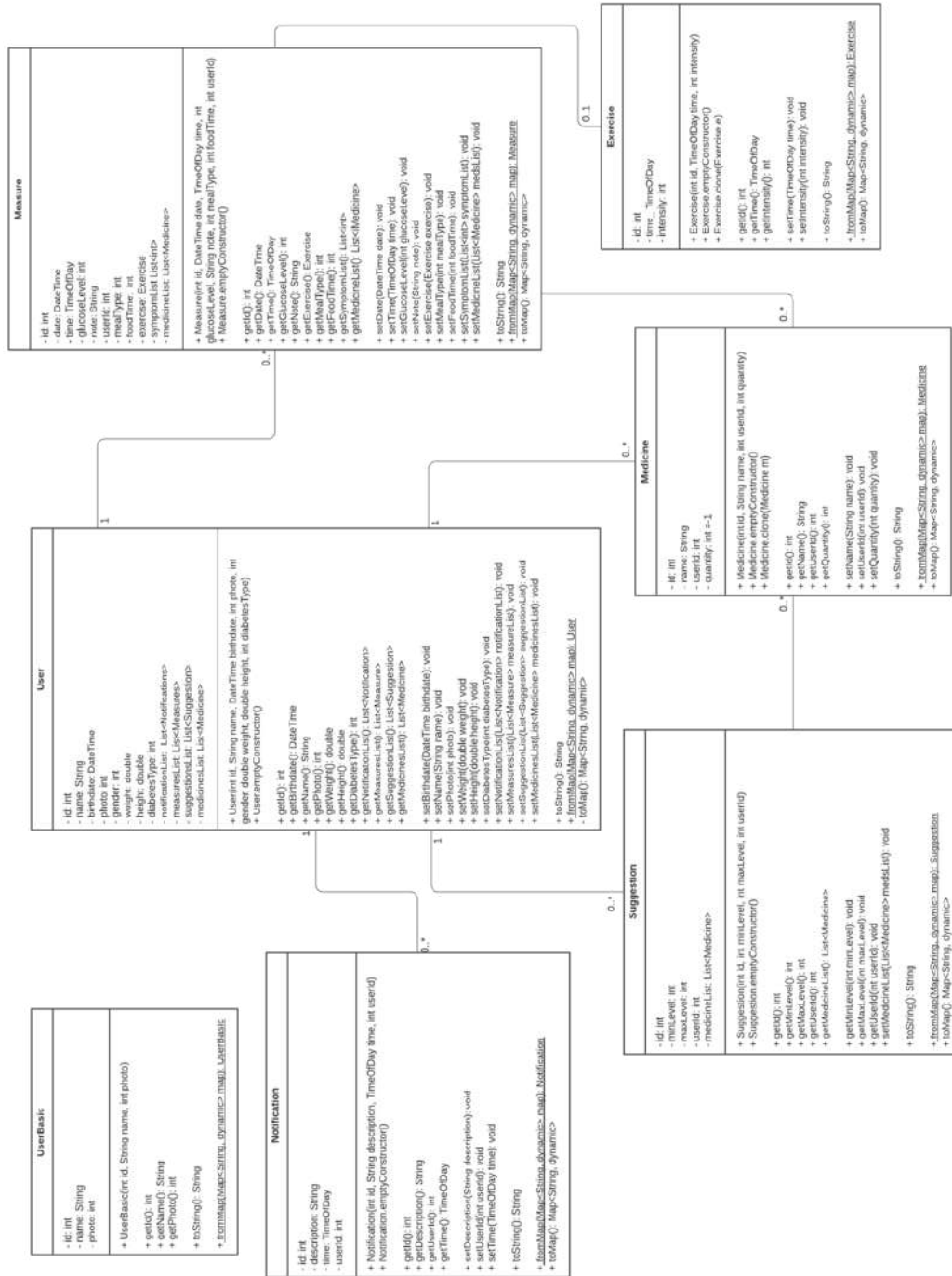


Figura 26: Diagrama de clases de los modelos usados en la aplicación

#### **4.4.2. Servicios de Base de Datos**

Son diferentes clases que actúan a modo de trámites para la gestión de los datos de un formulario o la interacción con la propia base de datos. Son necesarios como instancia para realizar operaciones de control y mantenimiento para el primer caso con formularios, mientras que contienen métodos estáticos que son llamados directamente y que actualizan los registros de la base de datos. Cada uno de los servicios hace referencia a un modelo de los vistos anteriormente.



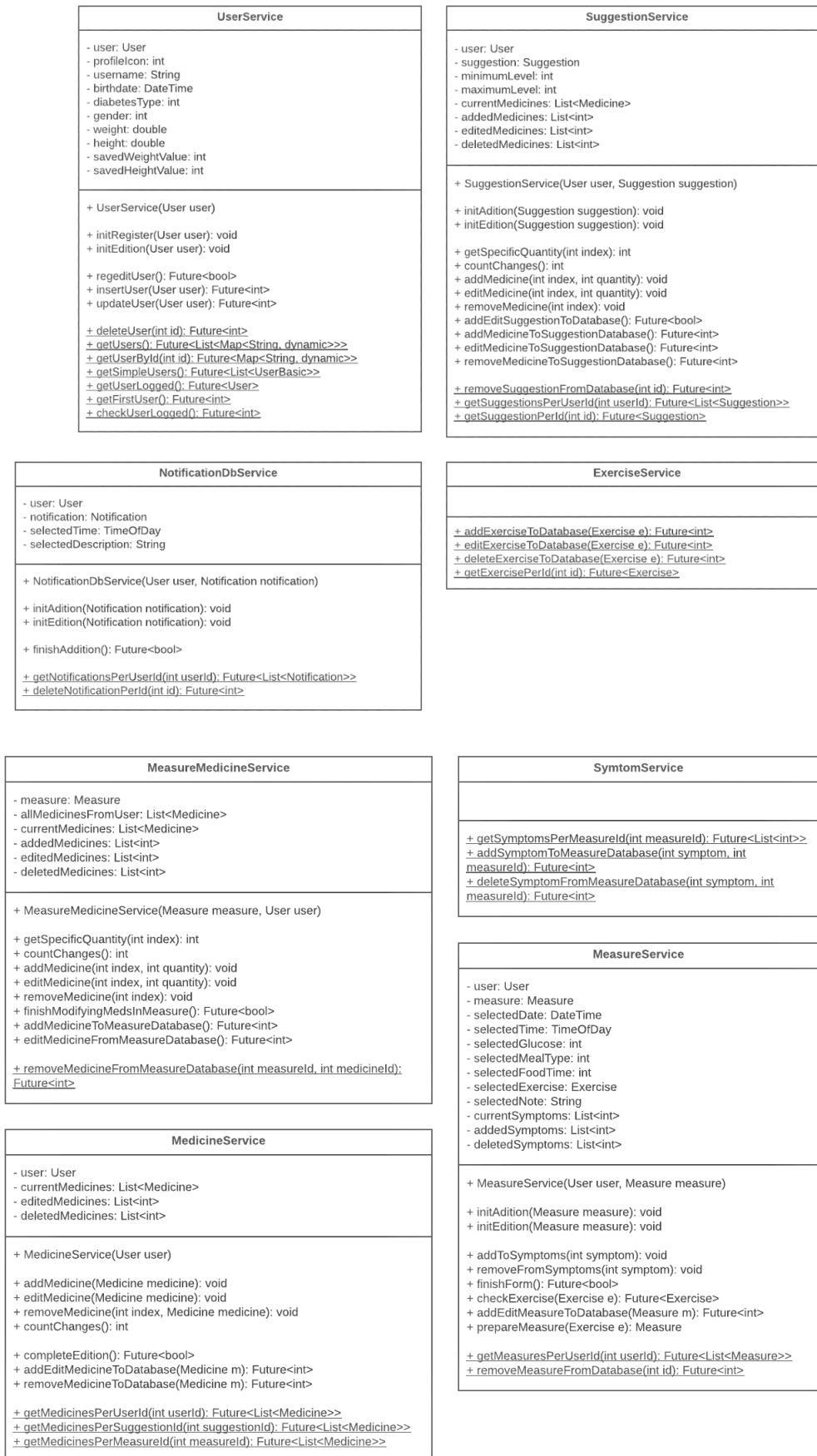


Figura 27: Diagrama de los servicios que gestionan Formularios y Base de Datos

### 4.4.3. Servicios para Librerías

Un servicio muy importante y que es referenciado continuamente en cada uno de los mencionados anteriormente es el que se encarga de la comunicación la propia base de datos de la aplicación. El funcionamiento se concentra en dos servicios con funciones relacionadas, pero diferentes. En el caso del servicio *DatabaseConnections* la actuación que tiene es cargar la base de datos adherida a la aplicación como *asset* o recurso para hacer así que el resto de la aplicación funcione correctamente. Es entonces cuando la instancia del servicio *DBHelper* es útil pues es la que contiene las operaciones para mantener la comunicación con la base de datos en todo momento, tanto para obtener la información como para modificar los datos que contiene. Por ello contiene diferentes funciones a las que pasándole el nombre de la tabla, así como el objeto a añadir, actualizar o eliminar consigue realizar las consultas necesarias para todos los métodos.

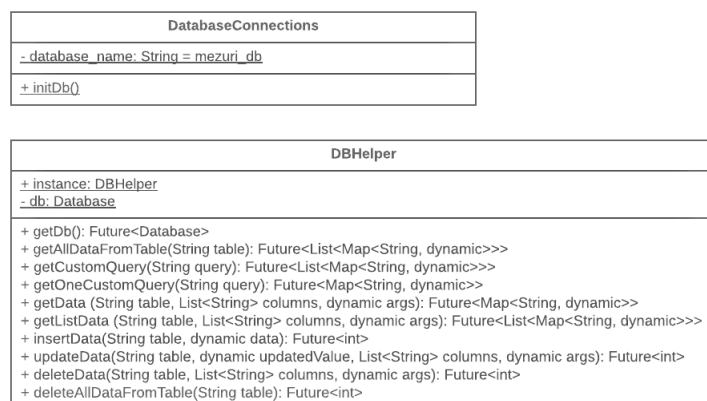


Figura 28: Modelo de los servicios para establecer la comunicación con la base de datos

Por otro lado, Flutter contiene un amplio registro de librerías desarrolladas por el propio Google o por otros usuarios en pos de facilitar el desarrollo de operaciones comunes dentro de la aplicación. La base de datos y los servicios vistos son un punto de acceso a ellos. Algo a priori complejo como trabajar con notificaciones o con los registros internos de un dispositivo son solucionados por estas librerías que aportan un enfoque similar al que, por ejemplo, provee Android cuando trabajamos directamente con él. En este caso, y aunque se explicará con mayor

detalle en la sección de Desarrollo, se encuentran aquí servicios que favorecen el trabajo con APIs, así como los mencionados de registros y notificaciones del dispositivo.

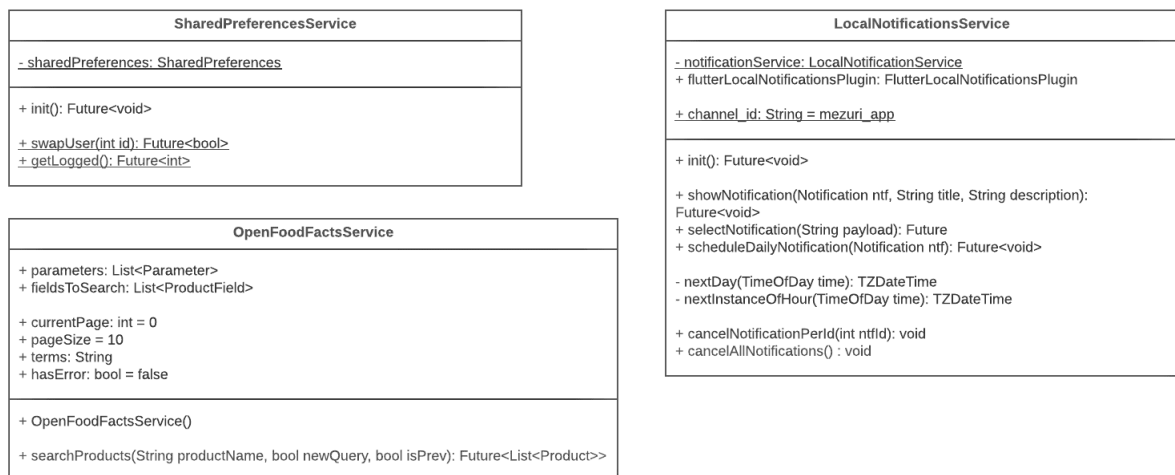


Figura 29: Clases que gestionan librerías incorporadas en la aplicación

#### 4.4.4. Útiles

Son clases creadas específicamente para contener métodos estáticos con los que realizar transformaciones de diferentes datos en diferentes ámbitos como mostrar datos dentro de la aplicación, convertirlos de uno a otro tipo para ser almacenados o simplificar la recurrente tarea de escoger un mensaje predeterminado correcto para ser mostrado. Incorporan ayudas por ejemplo para realizar traducciones dentro de la aplicación o para configurar el tamaño visual de ciertos elementos en virtud de un diseño *responsive*.

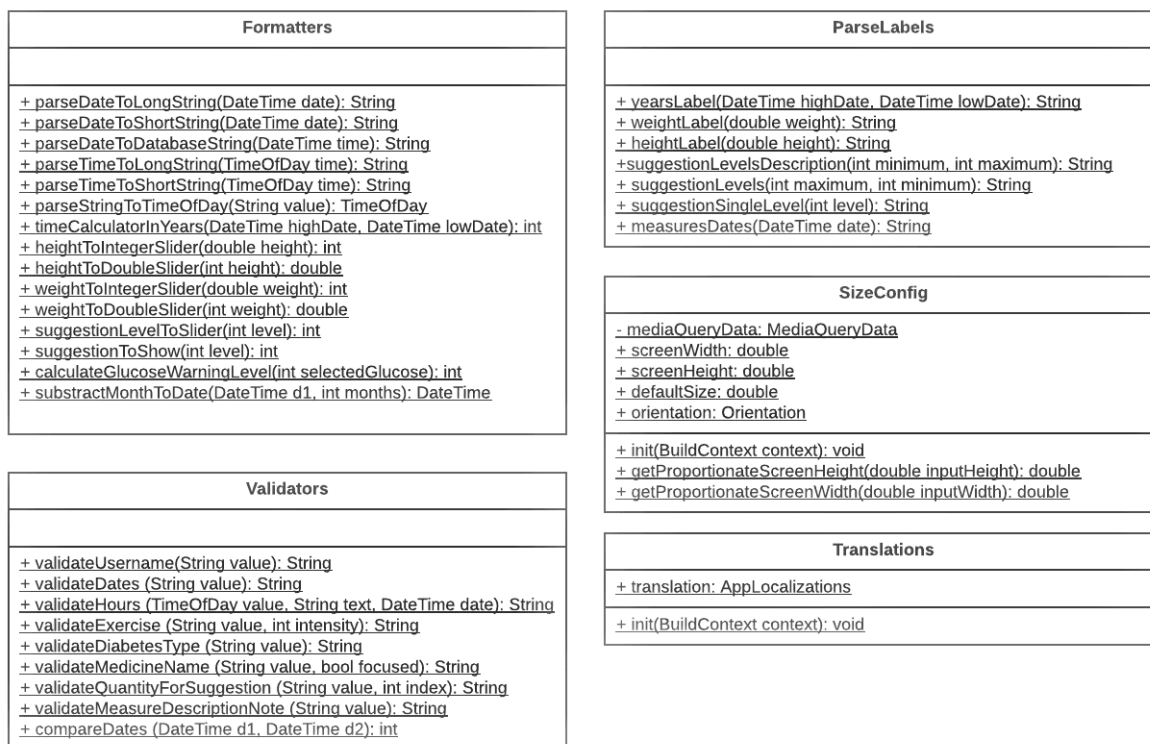


Figura 30: Modelo de las clases que ayudan a la gestión de información y transformación de datos

También pueden considerarse de este tipo la clase nombrada como *Validators* [Fig. 30], la cual incluye diferentes métodos aplicables a formularios para gestionar que los datos que se introducen son correctos. En Flutter, estos validadores devuelven siempre una cadena de texto si se ha producido un error, mientras que devuelven un valor nulo si todo está correcto, permitiendo así manejar también los errores que deben mostrar en cada uno de los casos.

Por último, para facilitar la gestión de la navegación se han desarrollado dos clases nombradas como *MesuriRoutes* y *RouteGenerator* que contienen la ruta que da nombre a cada una de las ventanas sobre las que nos moveremos y el proceso de creación y paso de argumentos de una a otra, respectivamente. Usando las constantes definidas en la primera, se establece la segunda como mecanismo por defecto de enrutamiento de la aplicación, haciendo que sólo llamando a dicha ruta, y pasando los argumentos, podremos tener una navegación propia y eficiente.

MesuriRoutes
+ loadingScreen: String = '/' + mainScreen: String = '/main-screen' + profileRegeditScreen: String = '/profile-edition' + medicinesScreen: String = '/medicines' + medicinesRegistryScreen: String = '/registry-medicines' + suggestionsScreen: String = '/suggestions' + suggestionDetailScreen: String = '/suggestion-details' + suggestionAddEditScreen: String = '/suggestion-addedit' + measureAddEditScreen: String = '/measure-addedit' + measureDetailsScreen: String = '/measure-details' + usersScreen: String = '/users-screen' + nutritionScreen: String = '/nutrition-screen' + nutritionDetailsScreen: String = '/nutrition-details-screen' + notificationScreen: String = '/notification-screen' + notificationDetailsScreen: String = '/notification-details-screen' + notificationAddEditScreen: String = '/notification-addedit-screen'
RouteGenerator
+ generateRoute(RouteSettings settings): Route<dynamic>

Figura 31: Clases para gestionar el sistema de navegación de la aplicación

#### 4.5. Constantes

Mesuri también cuenta en su código diferentes constantes con valores que se usan para que el almacenamiento en la base de datos y la traducción sea más sencilla. Es así por ejemplo en el caso de diferentes mensajes como el tipo de diabetes, las imágenes que pueden seleccionarse como iconos del usuario o los tipos de comida. El valor de los mismos se almacena dentro de la aplicación como enteros, siendo este el que se almacenará más tarde en la base de datos. Al mostrarlo, hay presentes métodos declarados también como constantes que recibiendo dicho valor, devuelven la cadena correcta en el idioma correcto. Precisamente, en el apartado de nutrición, muchas categorías que provienen y diferentes tipos de alimentos se ha realizado de esta manera.

También se consideran constantes los aspectos relacionados con el tema que usa la aplicación. Si bien permite el modo claro y oscuro, ambos han sido definidos a partir de un valor fuera de alguna clase, de manera que desde cualquier punto de la aplicación sea posible acceder a ellos.



Figura 32: Colores usados en la aplicación declarados como constantes



Figura 33: Géneros declarados como constantes enteras para su procesamiento

## 4.6. Interfaz Gráfica

En el diagrama de navegación [Fig. 25] ya se ha visto el diseño de la propia interfaz de la aplicación. Flutter ha sido la elección para elaborar este proyecto gracias a su facilidad para el diseño de interfaces vistosas y agradables. Cada una de las pantallas que se mencionan son lo que se llama en Flutter como Widgets. Por hacer un breve resumen de la tecnología, todo lo que usamos en Flutter es un Widget, ya sea un texto, un botón o una vista entera (no deja de ser una combinación de diferentes widgets). Cada uno de estos es lo que en otros frameworks como Ionic o Angular se llaman componentes.

Cada uno de estos componentes permite la comunicación de unos a otros y la composición para crear vistas fácilmente modificables que incorporan tanto aspectos de diseño gráfico, evidenciado en la creación de la interfaz, acompañado por lógica que permite actualizar la aplicación o la propia vista en función de nuestras acciones sobre la misma. Esto ha favorecido a que la aplicación no contenga demasiadas pantallas para facilitar la comprensión de la misma de cara al uso por usuarios inexpertos.

### 4.6.1. Línea de Diseño

El diseño se ha centrado principalmente en eso: acciones sencillas y comprensibles con un sólo vistazo de la aplicación. Iconos reconocibles y elementos que se visualicen correctamente para evitar errores en el trato de la misma.

#### 4.6.1.1. Icono de la aplicación

Toda la aplicación se ha desarrollado como es evidente teniendo en mente el color azul, un hecho que puede verse reflejado en las interfaces que aparecen en el anterior diagrama de navegación; sin embargo, la decisión de esto, aunque dudosa al principio, fue rápidamente escogida como la mejor. Este color azul se relaciona en gran medida con el propio icono de *Mesuri* [Fig. 34] y a su vez, ambos provienen del conocido y establecido **Símbolo Mundial de la Diabetes** [34] [Fig. 35] que podemos comprobar a continuación. Este icono es representado y conocido a partir de esta enfermedad. A raíz de esta idea, se modificó el diseño del propio símbolo de la diabetes a fin de no adueñarse comercialmente de él incorporando una gota en color rojo que vendría a simbolizar la sangre que tenemos que usar cuando tomamos una medición, además de hacerlo blanco para usar un tono azulado como color principal.

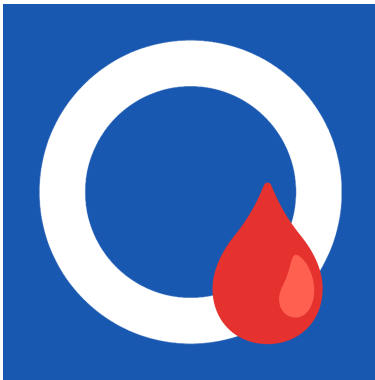


Figura 34: Icono de Mesuri

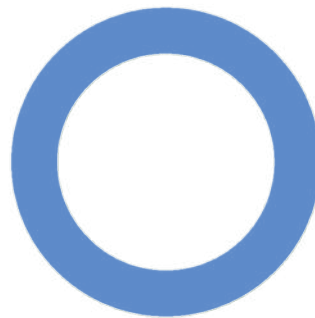


Figura 35: Símbolo Mundial de la Diabetes

#### 4.6.1.2. Colores Principales y Adaptación de Modos

Los colores principales escogidos han sido un tono azul oscuro (`#1958B1`) acompañado por un tono más claro (`#5D89C8`) y un tercero que hace de contraste con ambos (`#EAC468`). Estos se modifican ligeramente para la incorporación del modo oscuro dentro de la aplicación [Fig. 36], el cual ha sido añadido atendiendo a que todo elemento sea fácilmente visible en ambos modos. En parte, esto ha sido realizado gracias a la facilidad de Flutter para la elaboración de temas nuevos, incorporando por defecto la posibilidad de configurar un tema que abarca textos, iconos, barras de navegación y otros elementos de la aplicación para cada uno de los modos (oscuro y claro). Esto, asociado a un archivo con constantes donde se han declarado

los colores, permite que tan sólo modificando un pequeño valor, actúe sobre cada uno de los elementos de la aplicación que lo usan.

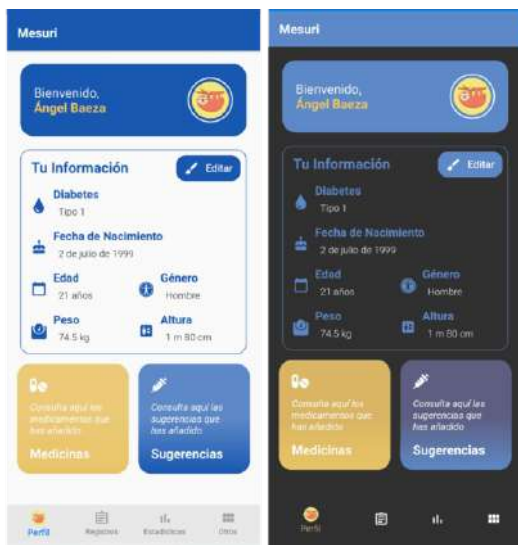


Figura 36: Comparación entre Modo Claro y Oscuro



Figura 37: Comparación entre tamaño de letra S y XL

Se ha intentado también combinar la interfaz de manera que sea fácilmente visible de acuerdo a la configuración del dispositivo referente al tamaño de letra. Cada dispositivo permite configurar el texto base del SO en función de si lo queremos más grande o más pequeño. Personas con mala visión y personas mayores podrían tener problemas con un entorno con el que un usuario normal no lo tendría. Por esto, se ha elaborado la interfaz de manera que acepte estos cambios de tamaño adaptando elementos al tamaño del texto que contienen [Fig. 37].

#### 4.6.1.3. Explicación del Nombre: Mesuri

Por último, es conveniente explicar el propio nombre de la aplicación. *Mesuri* proviene del esperanto. Su significado original es *Medida*, algo conveniente para lo que es esta aplicación. El esperanto es una idioma mundial que surgió como la unión de los pueblos en un significado de esperanza, por ello la primera intención al buscar un nombre fue usar algo en este. Por esto, buscando palabras relacionadas apareció *Mesuri*, que si bien significa lo dicho, parece aunar los conceptos de *medir* y *azúcar* (*sugar*). Esto hizo que se estableciese como el nombre definitivo.



## 4.7. Estructura del Proyecto

Cuando se inicia un proyecto en Flutter es importante conocer que no hay una estructura correcta para el mismo. Hay varios modelos que se pueden seguir, por lo que se explicará aquí la estructura del que se ha usado, pero este puede variar y ser adaptado a diferentes tipos. En concreto, el seguido para este proyecto centraliza cada funcionalidad en carpetas individuales que contienen todos los *widgets* destinados a esa característica. Explicando únicamente los directorios y archivos relevantes, podemos encontrar:

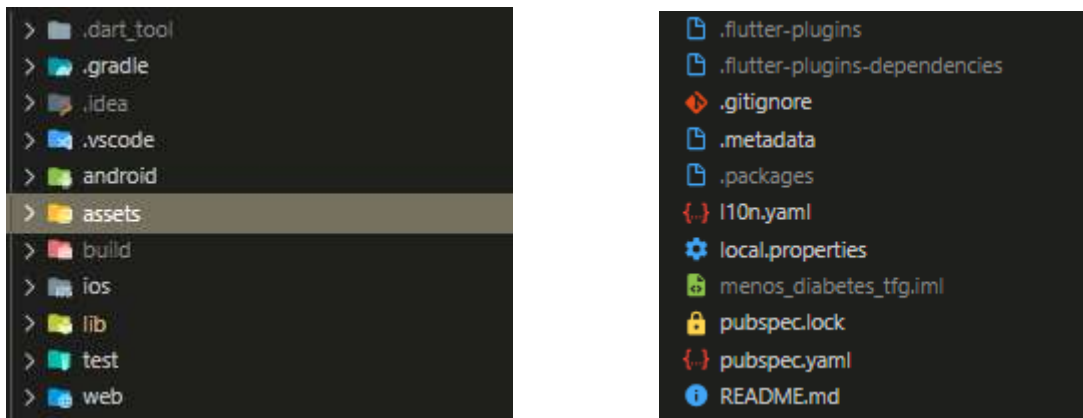


Figura 38: Estructura General Completa del Proyecto

- **android:** Carpeta que contiene la configuración de la aplicación para dispositivos Android. Si se desea incluir algún permiso o configuración debe hacerse desde aquí.
- **ios:** Carpeta que cumple la misma función que la anterior, pero para la configuración de dispositivos iOS.
- **assets:** Carpeta donde se incluyen todos los recursos propios de la aplicación como imágenes, vídeos... Cualquier cosa utilizable en la aplicación.
- **lib:** En esta carpeta se incluye el código principal del proyecto incluyendo configuración, pantallas, componentes... Es la carpeta principal.
- **pubspec.yaml:** Archivo donde se incluyen todas las librerías que quieren añadirse, así como los recursos que se usarán para la misma.

#### 4.7.1. Directorio *assets*

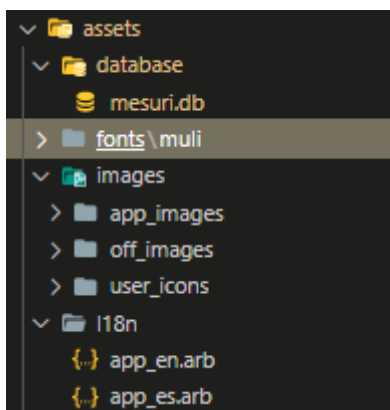


Figura 39: Estructura del Proyecto. Directorio '*assets*'

- **database:** En esta carpeta se encuentra la base de datos que se inicializa cuando la aplicación se instala en el dispositivo.
- **fonts:** Contiene fuentes / tipos de letra adicionales que se incluyen en la aplicación para mostrar ciertos textos. Las fuentes usadas deben ir añadidas al archivo *pubspec.yaml*.
- **images:** Son las imágenes que sirven de apoyo a la aplicación ya sea los iconos de usuario o el icono de la propia aplicación, entre otros.
- **i18n:** Carpeta donde se encuentran los archivos necesarios para establecer diferentes idiomas en la aplicación.

#### 4.7.2. Directorio *lib*

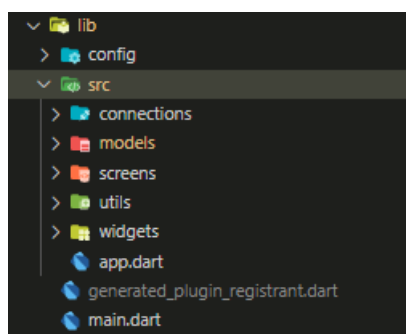


Figura 40: Estructura del Proyecto. Directorio '*lib*'

- **config:** Contiene archivos que sirven de configuración para el proyecto como pueden ser los temas establecidos para los modos claro y oscuro, así como las diferentes rutas existentes en la aplicación.

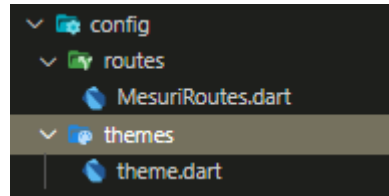


Figura 41: Estructura del Proyecto. Directorio 'config'

- **src:** Es la carpeta principal en la que se almacena cada una de las pantallas, clases y archivos adicionales para ejecutar la aplicación. Ofrece diferentes opciones en su interior.

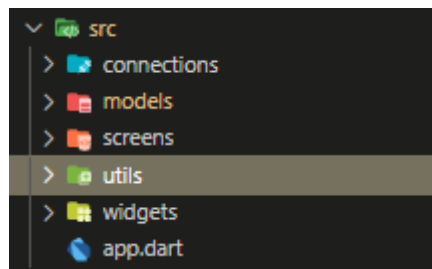


Figura 42: Estructura del Proyecto. Directorio 'src'

- ◇ **connections:** Archivos relativos a la creación e inicialización de la base de datos.
- ◇ **models:** Contiene todos los modelos de objetos que se usan en la aplicación.

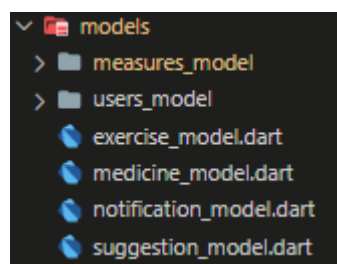


Figura 43: Estructura del Proyecto. Directorio 'models'

- ◇ **screens:** Aquí se contienen cada una de las pantallas diseñadas para la aplicación divididas en carpetas. De esta manera si se quiere modificar una en concreto es más fácil de acceder a ella.

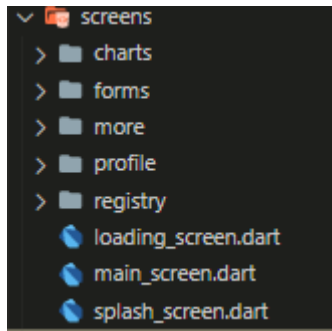


Figura 44: Estructura del Proyecto. Directorio 'screens'

- ◇ **utils:** Contiene archivos útiles para el desempeño de la aplicación como constantes usadas para visualizar información, así como *helpers* que ayudarán a la transformación de datos y *services* que contienen la lógica de formularios y base de datos.



Figura 45: Estructura del Proyecto. Directorio 'utils'

- ◇ **widgets:** Existe una carpeta con el mismo nombre en el interior de cada una de las carpetas correspondientes a cada ventana de *screens*. En este caso se engloban widgets generales usados a lo largo de toda la aplicación.



# 5

## Desarrollo e Implementación

Una vez la planificación se realiza completamente, o en gran medida dependiendo de la metodología usada para el proyecto, el desarrollo del proyecto pasa por la programación del mismo para entrar en la fase de **Implementación, Pruebas e Instalación y Despliegue** de la aplicación en cuestión. En esta sección se tratarán el modelo de aspectos internos del desarrollo así como las diferentes iteraciones realizadas y en qué requisitos se han centrado.

### 5.1. Base de Datos

Para la base de datos se ha decidido usar **SQLite** exclusivamente. Esta es una herramienta sencilla y sobre todo ligera que permite ser incluida y gestionada por la propia aplicación. El tema principal de la aplicación es ser sencilla para cualquier usuario y ya que no se aspira a tener una complicada red de información ni, como se explica al principio de esta memoria, diferentes versiones de la aplicación para otros dispositivos, la necesidad de una base de datos remota no existe.

Flutter además es un framework que no es precisamente ligero en sus versiones, haciendo que cualquier aplicación, ya sea sencilla o compleja, pueda resultar más pesada que si es realizada en otro lenguaje o framework. Buscar una opción simple, fácil de gestionar y que suponga poco peso añadido al producto ha sido algo clave en la elección de esta tecnología. Además, entre todas las opciones disponibles para Flutter, combinando bases de datos tanto SQL como NoSQL, SQLite es una herramienta testada y muy extendida para aplicaciones móviles. El tipo de datos que se guardan, pudiendo ser considerados datos sensibles, favorecen al trato local de los mismos para no caer en daños personales, sumado a que la aplicación está

pensada como un apoyo al paciente con diabetes y no una solución final para la gestión de cada uno de los elementos de la misma.

El problema de usar esta tecnología es que requiere de una gestión más correcta y estricta de los datos que si se usase cualquier alternativa NoSQL. Además, en SQLite concretamente hay ciertos tipos de datos que no están presentes y no son opciones disponibles a la hora de almacenar la información. La omisión o falta de estos, que además son tipos recurrentes en el momento de guardar la información como por ejemplo, para almacenar fechas o tiempos, o incluso un valor condicional, provoca que no puedan ser almacenados directamente usando esta plataforma, sino que deben tratarse tanto en la entrada como salida de la misma. Sin embargo, este problema ha sido solucionado mediante ciertos métodos visibles en las clases explicadas anteriormente que formatean los datos en ambos sentidos a fin de tener un funcionamiento lo más óptimo posible.

### **5.1.1. Modelo**

El modelo representado [Fig. 46] corresponde al último de los diseños de la base de datos. Cuando se comenzó a realizar la base de datos, esta incluía mucha más información y eso daba lugar a muchas tablas adicionales. Cada una de estas tablas se usaban principalmente para la inclusión de traducciones dentro de la misma; sin embargo, estas tablas adicionales modificaban cómo debía ser la gestión de la información incluyendo una capa realmente innecesaria de complejidad a la misma. Además, al incluir idiomas, sería mucho más difícil realmente introducir nuevos idiomas en el futuro. Finalmente, la inclusión de constantes en el código, así como la experiencia usando Flutter permitió ir aprendiendo y modificando poco a poco el diseño en el que basamos esta estructura permitiendo que las tablas trataran no sólo menos datos, sino además datos mucho más sencillos como lo son enteros y, al mismo tiempo, reducir la complejidad de las llamadas dentro de la propia aplicación, así como el número de ellas.

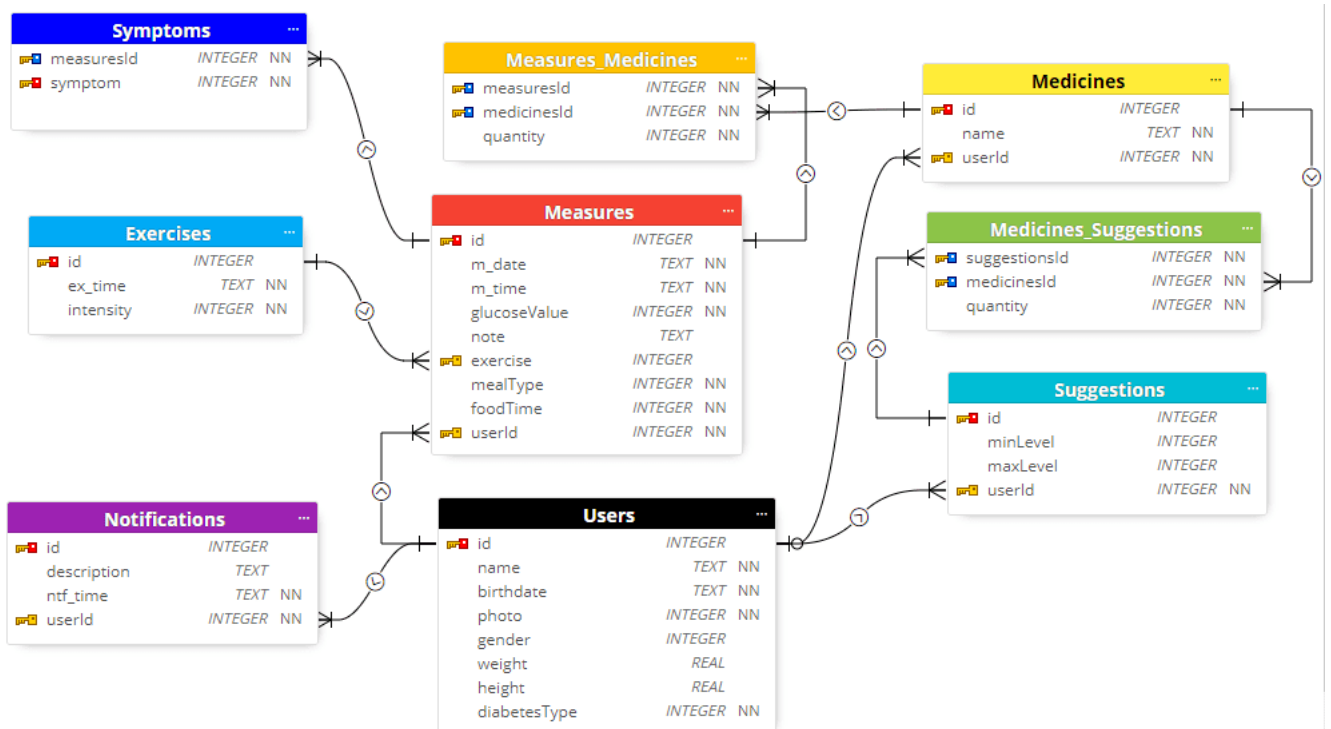


Figura 46: Modelo de Base de Datos. Última versión.

### 5.1.2. Explicación de las Tablas

El modelo final de la base de datos [Fig. 46] contiene un total de 9 tablas, de las cuales dos son tablas intermedias para la elaboración de listados (*Measures\_Medicines*, *Medicines\_Suggestions*). Cada una de estas tablas se explicará en las siguientes secciones, resumiendo sus parámetros para dejar claro el uso y propiedades de cada uno de ellos.

A modo de resumen, cabe destacar que ninguno de los parámetros nombrado como *id* y que hace referencia como se puede intuir al *id* de cada registro (considerado clave principal o **Primary Key (PK)**) aparece marcado como **Not Null (NN)**, siendo posible que sean nulos o sin valor asociado (no obligatorios). El motivo de esto, dado que no aparece en el modelo, es porque cada uno de ellos está definido como **Autoincrement (AI)**, haciendo que no sea necesario pasar este valor para añadir un registro, sino que internamente se crea sólo aumentando por sí mismo.



### 5.1.2.1. Users (Usuarios)

La tabla *Users* se puede considerar una de las principales del modelo que se presenta. Se encarga de almacenar toda la información de cada uno de los usuarios y que será de utilidad para el manejo posterior del mismo en la aplicación. A modo de *Primary Key* contiene un valor numérico que se autoincrementa con cada usuario añadido al sistema.

Parámetro	PK	NN	Comentarios
<b>id</b>	X		Referencia a un usuario en otras tablas con las que tiene relación.
<b>name</b>		X	Nombre del usuario. Útil a la hora de mostrar información y gestionar diferentes usuarios
<b>birthdate</b>		X	Fecha de nacimiento. Útil para conocer al paciente. Debido a los tipos soportados por SQLite es almacenado como texto.
<b>photo</b>		X	Icono de cada usuario. Entero que referencia un icono de la aplicación para aliviar el almacenamiento.
<b>gender</b>			Género del paciente. No obligatorio. Almacenado como entero para facilitar la gestión del mismo y su traducción.
<b>weight</b>			Peso del paciente. No obligatorio. Almacenado como número real incluyendo así decimales.
<b>height</b>			Altura del paciente. No obligatorio. Número real almacenado para incluir decimales.
<b>diabetesType</b>		X	Tipo de diabetes padecida. Motivo principal de la aplicación. Valor entero referenciado en la aplicación para su tratamiento.

Tabla 1: Modelo de Base de Datos. Tabla *Users*. Explicación.

### 5.1.2.2. Measures (Mediciones)

Las mediciones son también una de las claves de la aplicación y su referencia en la base de datos es muy importante. Representan el nivel y estado del paciente cada vez que mide su azúcar. Cada una de estas debe ir asociada a un usuario. Para esto se utiliza el *id* del mismo que actúa como un campo en el mismo.

Parámetro	PK	NN	Comentarios
<b>id</b>	X		Referencia la medición en otras tablas con las que tiene relación.
<b>m_date</b>		X	Fecha en que se realizó. Mantener un registro adecuado.
<b>m_time</b>		X	Hora y minuto en que se tomó. Mantener un registro adecuado.
<b>glucoseValue</b>		X	Nivel de glucosa tomado en la medición.
<b>note</b>			Información adicional adherible a la medición.
<b>exercise</b>			FK. Referencia al registro de actividad física realizado por el usuario previo a la medición.
<b>mealType</b>		X	Estado en que se tomó la comida. Entero para ser tratado internamente.
<b>foodTime</b>		X	Comida en la que se toma la medición (desayuno, comida...). Entero para ser tratado.
<b>userId</b>		X	FK. Referencia al usuario al que pertenece esta medición.

Tabla 2: Modelo de Base de Datos. Tabla *Measures*. Explicación.

### 5.1.2.3. Medicines (Medicinas)

Esta tabla contiene las **medicinas** de cada uno de los usuarios y que se usan tanto en las mediciones para dar lugar a las medicinas que ha tomado el paciente como en la sugerencias que le son recomendadas al mismo tras cada medición.

Parámetro	PK	NN	Comentarios
<b>id</b>	X		Referencia la medicina en otras tablas con las que tiene relación.
<b>name</b>		X	Nombre de la medicina. Identifica la medicina concreta.
<b>userId</b>		X	FK. Referencia al usuario que ha añadido esta medicina.

Tabla 3: Modelo de Base de Datos. Tabla *Medicines*. Explicación.

### 5.1.2.4. Measures\_Medicines

Esta tabla contiene información sobre las medicinas que son añadidas por el usuario a cada medición y que se consideran como tomadas, por lo que para llevar un registro de las mismas,

se hace necesario incluir una cantidad de la misma para ser almacenado.

Dado que cada medición puede incluir diferentes medicinas que el usuario haya tomado, se ha optado por usar como clave principal la combinación del identificador de cada medición y medicina, configurando la relación de manera que si cualquiera de estas se borra, esta también sea eliminada del sistema.

Parámetro	PK	NN	Comentarios
<b>measuresId</b>	X	X	FK. Referencia la medición a la que se añade la medicina.
<b>medicinesId</b>	X	X	FK. Identificador de la medicina que se ha tomado.
<b>quantity</b>		X	Cantidad tomada de la medicina en la medición.

Tabla 4: Modelo de Base de Datos. Tabla Intermedia *Measures-Medicines*. Explicación.

#### 5.1.2.5. Suggestions (Sugerencias)

Las **sugerencias** son configuraciones del usuario para que, entre unos niveles establecidos como mínimo y máximo, le sean recomendadas unas medicinas concretas con unas cantidades adecuadas de cada una. Cada usuario posee las suyas propias, por lo que se referencia al mismo a través de su *id*.

En este caso, es necesario añadir una gestión extra para ambos niveles, pues ninguno es obligatorio, a pesar de que siempre debe existir al menos uno. Este control se realiza desde la aplicación.

Parámetro	PK	NN	Comentarios
<b>id</b>	X		Identificador de la sugerencia para ser referenciada con otros elementos.
<b>minLevel</b>			Nivel mínimo de glucosa sobre el que actuará la sugerencia.
<b>maxLevel</b>			Nivel máximo de glucosa sobre el que actuará la sugerencia.
<b>userId</b>		X	FK. Identificador del usuario al que pertenece la sugerencia.

Tabla 5: Modelo de Base de Datos. Tabla *Suggestions*. Explicación.

### 5.1.2.6. Medicines\_Suggestions

Al igual que las mediciones, cada sugerencia puede incluir diferentes medicinas con una cantidad adecuada que el usuario debe tomar. Para hacer esto, se establecen como claves primarias los identificadores de cada uno, sugerencia y medicina.

Esta relación está configurada de manera que borrando una medicina, se borrará de las sugerencias en las que esté presente. Esto también se da en el caso contrario, donde eliminar una sugerencia eliminará los registros en los que esté presente.

Parámetro	PK	NN	Comentarios
<b>suggestionsId</b>	X	X	FK. Referencia la sugerencia a la que se añade la medicina.
<b>medicinesId</b>	X	X	FK. Identificador de la medicina que será sugerida.
<b>quantity</b>		X	Cantidad recomendada de la medicina para la sugerencia.

Tabla 6: Modelo de Base de Datos. Tabla Intermedia *Medicines-Suggestions*. Explicación.

### 5.1.2.7. Exercises (Ejercicios)

Los ejercicios hacen referencia a la actividad física realizada por el usuario en el momento previo de la medición. De esta manera, es fácil controlar si su actividad física influye en las mediciones. Es utilizado por ello en las mediciones.

Incluye un tiempo con la cantidad de ejercicio realizado, así como una intensidad que se almacena como entero para ser tratado posteriormente por la aplicación.

Parámetro	PK	NN	Comentarios
<b>id</b>	X		Identificador del ejercicio. Usado para ser referenciado en la medición a la que se añade.
<b>ex_time</b>		X	Tiempo de ejercicio realizado.
<b>intensity</b>		X	Intensidad del ejercicio realizado.

Tabla 7: Modelo de Base de Datos. Tabla *Exercises*. Explicación.

### 5.1.2.8. Symptoms (Síntomas)

Los síntomas también son comunes en las mediciones. De hecho, es habitual tomarse mediciones por el simple hecho de padecer síntomas de glucosa elevada o baja. Por este motivo se considera añadir un listado de síntomas a cada una de ellas.

La clave principal es la combinación de la medición a la que se añade y del propio síntoma, no pudiendo añadir dos veces el mismo síntoma a la medición. Si la medición fuera borrada, todos los registros se eliminarían consecutivamente.

Los síntomas se almacenan como valores enteros para facilitar el tratamiento, siendo contenido su significado dentro de la propia aplicación y usando dicho valor para ser mostrado.

Parámetro	PK	NN	Comentarios
<b>measuresId</b>	X	X	FK. Identificador de la medición a la que se añade el síntoma.
<b>symptom</b>	X	X	Identificador del síntoma que se añade a la medición.

Tabla 8: Modelo de Base de Datos. Tabla *Symptoms*. Explicación.

### 5.1.2.9. Notifications (Notificaciones)

Cada usuario puede programar sus propias notificaciones que le servirán de aviso a determinadas horas del día. Cada una de ellas puede ser añadida o eliminada, de manera que el usuario tenga un control de las mismas adecuado con su ritmo de vida.

La descripción es un valor opcional para que el usuario identifique la notificación rápidamente tanto en la aplicación como al ser mostrada, pero no es obligatorio, pues el mensaje que se mostrará en la aplicación será predefinido si esta no existe.

Parámetro	PK	NN	Comentarios
<b>id</b>	X		Identificador de la notificación. Útil para identificar y programar o borrar la notificación del dispositivo.
<b>description</b>			Descripción para la medición. Será mostrada como texto de la notificación.

<b>ntf_time</b>		<b>X</b>	Momento en el que la notificación aparecerá en el dispositivo.
<b>userId</b>		<b>X</b>	FK. Identificador del usuario que programa la notificación.

Tabla 9: Modelo de Base de Datos. Tabla *Notifications*. Explicación.

## 5.2. Librerías Adicionales Utilizadas

Flutter y Dart proveen de muchas funciones y características para favorecer el desarrollo de la interfaz y lógica de la aplicación. Son lenguajes que ayudan a que la programación sea rápida. Uno de los factores que llevan a esto es la comunidad que contiene detrás que continuamente se encuentra desarrollando librerías y *widgets* que pueden ser usados en diferentes proyectos sin mayor complicación.

Para el desarrollo de ciertos elementos de la aplicación, así como algunas funcionalidades que se han incorporado, se han buscado librerías útiles para ello y seleccionado entre un grupo con características similares para cada una. Algunas de las que se han usado y que resultan más interesantes a nivel de desarrollo pueden ser *OpenFoodFacts*, *SQLite*, *Local Notifications*, *Flutter Native TimeZone*, *Flutter Localizations* y *Charts Flutter*. No son las únicas, pero sí las que a nivel de desarrollo son más relevantes. El resto han sido usadas para aspectos visuales como el selector circular que aparece, por ejemplo, cuando añadimos una medición o la incorporación de numerosos iconos adicionales.

### 5.2.1. Open Food Facts

*Open Food Facts* [35] es un proyecto colaborativo entre usuarios de todo el mundo sin ánimo de lucro. Su función principal es recopilar información nutricional de alimentos de todo el mundo en diferentes idiomas y con información útil para el usuario interesado como la procedencia, los alérgenos, el valor nutricional o simplemente imágenes del mismo.



Figura 47: Logo de *OpenFoodFacts*

Como proyecto es muy recomendable echarle un vistazo y se decidió incorporar a la aplicación como elemento principal para la sección de **Nutrición**. Si bien es una API que es fácilmente consultable y que, de hecho, ha sido consultada en el proceso de desarrollo a través de Postman para conocer la información recibida, en un trato que se tuvo con uno de los principales desarrolladores del proyecto se conoció la existencia de una librería [36] para Flutter en la que se podía facilitar la consulta de estos alimentos y sus características a través de funciones y tipos ya establecidos por defecto en cada una de ellas, lo que simplificaba el trato de información. Por ello, la decisión de escoger esta librería resultó interesante para el proyecto.

### 5.2.2. **SQLite**

**SQLite** [5] es una librería que incorpora la base de datos SQLite a nuestras aplicaciones móviles adaptada a Flutter incorporando los tipos y métodos necesarios para poner en marcha la aplicación. A día de hoy es compatible con Android, iOS y MacOS, dejando fuera la versión web. Permite la ejecución de transacciones y ayuda a realizar funciones básicas como las inserciones, actualizaciones o borrados de la base de datos que asociemos. De igual manera, provee métodos tanto para crear y configurar la BD desde cero como para inicializarla desde un recurso añadido previamente a la aplicación.

En la aplicación su uso está claro: establecer la base de datos sobre la que opera el sistema y hacer funcionar las llamadas a la misma para la gestión. Previamente ya se ha explicado el modelo usado en la misma.

### 5.2.3. **Local Notifications**

Esta librería es la encargada de manejar y gestionar los eventos producidos por la aplicación a través de sus notificaciones. El trato de notificaciones siempre ha sido algo complicado

y Flutter / Dart no da soporte directo a ellas. Esta librería [37] provee de funciones y características propias para generar notificaciones y programarlas a horarios específicos para que aparezcan cuando el desarrollador quiera hacerlo. En este caso, es el propio usuario quien establece la fecha para las mismas, pudiendo eliminarlas y añadirlas fácilmente.

#### **5.2.4. Flutter Native TimeZone**

Esta librería ha sido útil precisamente para la gestión de las notificaciones. Cuando se programa una notificación es necesario tener en cuenta la hora en la que se establecerá la misma, y esta depende de la zona horaria del dispositivo.

Usando *Native Timezone* [38] se puede consultar la zona horaria en la que el usuario está para asegurar que la notificación se muestra exactamente cuando el usuario desee. El soporte actual es ofrecido para Android e iOS únicamente.

#### **5.2.5. Flutter Localizations**

Tener una aplicación multilinguaje es ideal en los tiempos que corren. No sólo ayuda a que tus usuarios comprendan mejor la aplicación, sino que también favorece a que se extienda la misma. Por este motivo, tener un medio para agregar idiomas de manera sencilla es conveniente y es lo que proporciona esta librería [39].

Con ella, al igual que en Android nativo, basta con tener un archivo por cada idioma que se quiera añadir en el que se contienen todos los textos de los que se disponga traducción. Así, sólo haciendo referencia a estos textos o identificadores de cada uno, nuestra aplicación obtendrá el idioma del sistema operativo para escoger el idioma adecuado. Si no existiera el idioma, se puede configurar uno por defecto; en este caso es el inglés.

#### **5.2.6. Flutter Charts**

Es una librería temprana que permite el desarrollo de gráficos de diferentes tipos (barras, sectores, líneas, conjuntos...) [40] y ayudan a dibujar la interfaz gráfica de los mismos. Si bien hay que configurar los datos que se van a mostrar y cómo queremos hacerlo, así como la leyenda, colores o formato, se simplifica mucho el dibujado de los mismos y es relativamente



fácil mostrar la información tal y cómo la queremos.

### 5.3. Iteraciones Realizadas

Como se comentó al inicio, el proyecto ha seguido una metodología Scrum adaptada a la situación de un estudiante en la que todos los roles han sido ocupados por el alumno, mientras que los tutores han ocupado el rol del *Product Owner* siendo los que finalmente evaluaban la aplicación. Como es usual en Scrum, las fases del desarrollo del proyecto se han distribuido en diferentes iteraciones desde el inicio del aprendizaje hasta la entrega de esta memoria.

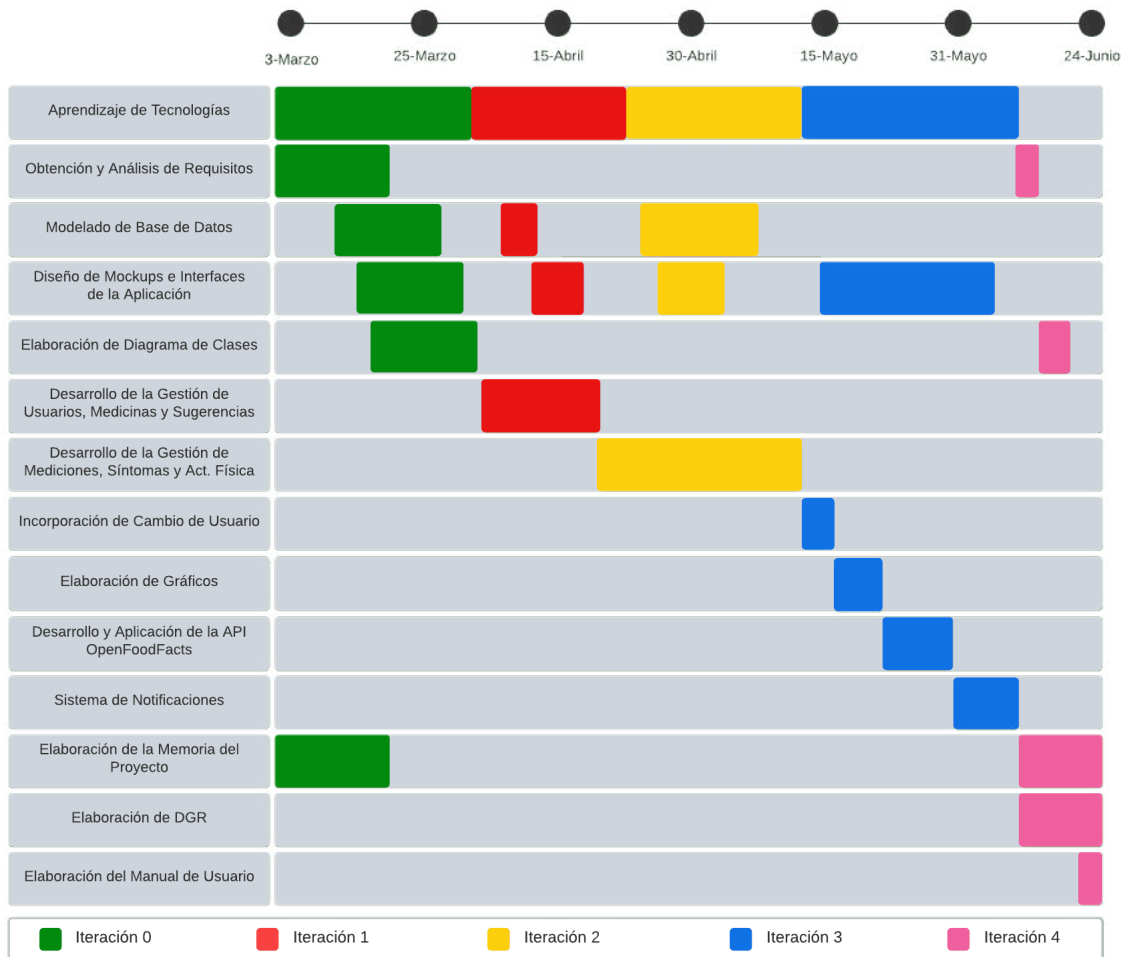


Figura 48: Diagrama de Gantt del Trabajo Realizado

El esquema de trabajo [Fig. 48] deja ver que el desarrollo se ha centrado en diversas iteraciones, concretamente 5 a lo largo cuatro meses que abarcan desde Marzo a Junio. Con este

plan, cada iteración se ha planificado de manera que abarque tareas independientes una de otra, siendo necesario y posible que, en algunos casos, se vuelve a echar el ojo atrás para comprobar o corregir elementos que se perciban incorrectos o que deban seguir tratándose. Es el caso por ejemplo del aprendizaje de tecnologías que, si bien se tuvo al inicio un período de aprendizaje, ha sido proceso continuo a lo largo de todos estos meses hasta acabar la aplicación.

### 5.3.1. Iteración 0

La iteración inicial se centró en aprender e investigar las tecnologías usadas. Dart es un lenguaje similar a *JavaScript* pero nunca había trabajado directamente con él, así como con Flutter, que aunque su aprendizaje pueda considerarse más sencillo que el de otros framework y a primera vista puedan realizarse aplicaciones con menor esfuerzo, un trabajo más completo como el que pretende ser esta aplicación requiere de conocer diversas librerías y aspectos adicionales para la comunicación entre ventanas y *widgets*.

Al mismo tiempo que se investigaban las tecnologías, se avanzaba en la elaboración de la memoria con elementos que ya se conocían y que pudieran ahorrar trabajo posterior. El proyecto ya se conocía en cuanto al público que iría destinado y se había realizado ya una encuesta y diferentes estudios de mercado sobre el enfoque que se tendría finalmente. Esto favoreció que las primeras secciones de la memoria fueran rápidamente completadas.

Con el estudio de mercado y unas cuantas decisiones de cómo debería ser la aplicación, se manejó la posibilidad y resultado final de enfocar la aplicación de cara a que cualquier usuario pudiera usarla sin necesidad de estar familiarizado con estas tecnologías mediante una interfaz sencilla e intuitiva. A lo largo de diferentes bocetos se planteó una aproximación a lo que sería la interfaz final.

Con todo este trabajo realizándose, la primera fase de la Ingeniería Software realizada, la **Obtención de Requisitos**, permitió avanzar rápidamente. El estudio de mercado proveyó un conocimiento añadido a lo que debería incluirse y se pedía, por lo que se comenzó a elaborar el Documento General de Requisitos a partir de ellos. Una vez analizados y revisados los re-

quisitos, los diagramas correspondientes a casos de uso y secuencia de cada uno de ellos se realizaron y se incluyeron en el propio DGR.

Por último, se comenzó la fase de **Diseño y Modelado** de los elementos que finalmente resultarían útiles para comenzar la elaboración de las diferentes funcionalidades. La primera versión de la base de datos surgió en este momento con diferentes tablas que dieron lugar a un diagrama de clases aproximado de lo que se usaría posteriormente en la aplicación.

### 5.3.2. Iteración 1

La iteración 1 comienza cuando todo el proceso de documentación inicial ha terminado, o al menos, gran parte de su elaboración. Algunas tareas se revisarán, pero la iteración se centra en desarrollar aspectos básicos del usuario dentro de la aplicación.

#### 5.3.2.1. Elaboración e Implementación de la Base de Datos

Lo primero que se realiza en este caso es elaborar la base de datos que se usará dentro de la aplicación. Esta actuará a modo de recurso de la misma y contará con un usuario establecido por defecto desde el inicio. Se incorpora a la aplicación y se prueba que se crea correctamente a partir de la misma. Una vez comprobado que los datos se crean bien y que pueden ser llamados de manera eficiente se crean las primeras interfaces de la aplicación, ya que Flutter plantea un desarrollo paralelo entre lo que sería la lógica y el aspecto visual de la aplicación.

El primer caso a solucionar es que el usuario cargado al inicio se mantenga durante todas las ventanas de la aplicación. Las llamadas a la base de datos se perciben como métodos asíncronos que hacían que se tardara unos segundos, en función del dispositivo, en obtener la información. Si cada vez que cambiáramos de pantalla esto debiera ser así, hacía inviable llamar al usuario una y otra vez, no sólo por el tiempo perdido, sino por la cantidad de llamadas que debieran realizarse.

Para solucionar esto se desarrolló una ventana de carga (*Loading Screen* en fig. 25) en la que se llamaría al usuario inicial para que una vez que fuera cargado sin errores, avanzase a la siguiente ventana. A partir de aquí se tendría la instancia del usuario desde el inicio y se

compartiría entre las diferentes ventanas mediante paso por parámetro entre ellas.

En esta iteración, además, se comenzó a tratar las primeras ventanas de la aplicación, así como la posibilidad de **gestionar usuarios** de manera individual con las acciones de añadir, eliminar y editar. Esto se tratará con mayor detalle en la siguiente sección, pero surgía un problema con la base de datos y es que ciertos datos no estaban bien definidos. Fue necesario revisar la base de datos y adaptarla de manera que se eliminaban diferentes tablas que no eran necesarias y se cambiaba el tipo de diferentes parámetros dentro de tablas como *Users*, *Medicines* y *Suggestions*.

### 5.3.2.2. Gestión de Usuarios

Cuando se planteó la gestión de usuarios se pensó en la elaboración de interfaces vistosas que dinámicamente y con el flujo de interacción del usuario se fueran actualizando. Flutter plantea el **estado** de la aplicación como una de sus características principales y permite que, mediante su **gestión**, se puedan producir resultados de cambios dinámicos de la interfaz. Esto al principio supuso un problema hasta conocer la mejor forma de controlar estos estados.

Mostrar la información del usuario una vez que ya era cargado era sencillo, y gracias al manejo que se hacía de la navegación, estableciendo rutas predefinidas para cada transición y un método general para ser gestionadas, las interfaces no se hacían complicadas. El paso de parámetros para conocer al usuario que se estaba añadiendo o editando se hacía a través de estos métodos y en este punto sólo preocupaba la creación de formularios para realizar las diferentes tareas. Ambos son realmente las mismas ventanas reutilizadas para ser útiles en ambas posibilidades.

Cada una de las ventanas que componían el formulario incluía diferentes tipos de datos. Para simplificar, se decidió establecer una ventana para cada uno de los datos en función de si eran obligatorios o no. De esta forma, la primera ventana sería la de datos obligatorios y la segunda la de datos opcionales. Si bien la lógica se realiza en Flutter sobre las ventanas de interfaz, gestionar los datos del usuario con el que se trata es complejo. Al cambiar entre estas dos ventanas que se mencionan se perdía la referencia al mismo y se reiniciaba cada una de

ellas perdiendo lo que ya hubiera sido introducido. Por esto, se creó el primer servicio que contendría las operaciones y parámetros fijos que se usarían en el formulario.

Ambas ventanas estarían dentro de un *PageView* que simplemente cambia entre pestañas. Así, creando una instancia del servicio para inicializar tanto al usuario editado como añadido según el caso, se usarían los datos del servicio para mostrarlos en los campos del formulario evitando así perder los mismos al cambiar de ventana. Sería en este servicio también donde se contienen las operaciones para confirmar los cambios realizados y realizar operaciones propias con la base de datos.

Aspectos interesantes de este caso es que como se ha visto en el modelo de la base de datos presentado, muchas variables son realmente enteros como el género o el icono. Mediante funciones a las que sólo hay que pasarle este entero, podemos usarlos para obtener los correspondientes iconos y elementos propios. De la misma forma, en los selectores de peso y altura encontramos una limitación y es que el propio eje sólo cuenta con 300 campos seleccionables, por lo que se realiza la operación de transformar cada uno de esos 300 campos para establecer un límite superior y poder trabajar con metros o kilos evitando detectar que existe esa limitación.

Mencionar que existe un desplegable o un diálogo como son llamados en dispositivos móviles tanto para el selector de icono, que aun pareciendo una pantalla nueva no es más que un *modal* que aparece y desaparece cuando seleccionamos algo. Igual pasa con diálogos más sencillos para establecer la fecha de nacimiento o el tipo de diabetes.

Finalmente, cuando aceptamos los cambios aparece un diálogo que nos marca si el proceso ha dado error o todo ha ido bien. Si ha dado error, podremos volver atrás, mientras que si todo ha ido bien, cerraremos la ventana y se gestionará el estado de la aplicación para aplicar los cambios realizados consultando de nuevo la base de datos para traer de nuevo los datos necesarios.

### 5.3.2.3. Mediciones y Sugerencias

Una vez que la parte de usuario fue terminada, las mediciones y sugerencias, como partes fundamentales de la aplicación fueron también incluidas en esta iteración. El modelo de interfaces, aunque conllevó un tiempo para ser elaborado se realizó de manera sencilla. El trabajo real estaba en la implementación y optimización de las operaciones con la base de datos.

Cuando se carga al usuario al inicio de la aplicación ya contiene todos los datos con los que se relaciona: mediciones, sugerencias, mediciones... Sin embargo, cuando se quiere añadir una medicina, por ejemplo, hacer una consulta por cada operación que se hace es muy ineficiente. Por ejemplo, si se desea editar una medicina que ya esté añadida realizamos una operación, pero si en el mismo momento el usuario se arrepiente y decide que quiere eliminarla, puede hacerlo, pero ya son dos operaciones las que se han realizado. En este caso son dos, pero podrían ser flujos que supongan mayor cantidad. Para omitir esto y realizar sólo una, o al menos, las mínimas posibles, se ha desarrollado la interfaz y servicio relacionado con un *buffer* en el que se registran qué medicinas han ido editadas, eliminadas o añadidas, de manera que si por ejemplo se realiza la operación de editar y eliminar seguidamente, sólo se hace la operación de eliminar en la base de datos. Este mecanismo se aplica igual para cuando queremos añadir las medicinas dentro de cada sugerencia.

Las sugerencias, por su parte, se realizan de manera sencilla. El usuario selecciona los niveles de glucosa en los que quiere aplicar la sugerencia mediante selectores circulares más fáciles y rápidos de manejar y finalmente añade las medicinas que quiere que se le recomienden.

### 5.3.2.4. Configuración Inicial de Aplicación

Cuando comenzamos una aplicación en Flutter hay muchos elementos que realizar para configurarla de manera correcta como lo son los temas de la aplicación y colores que se usarán como principales. Esta tarea fue realizada también en esta iteración, aunque se vería completada a lo largo de todo el desarrollo.

### 5.3.3. Iteración 2

Esta iteración, con la parte de usuarios ya terminada se centró en desarrollar el listado y aspectos propios de las mediciones de glucosa que mantiene el usuario y que es la parte principal de la aplicación realmente. No sólo operaciones básicas como añadir, eliminar o editar, sino también la posibilidad de asociar síntomas o actividad física realizada a cada una.

Además, se incorporan también las medicinas que el usuario ha tomado para dicha medición y se permite ver las sugerencias que coinciden con el nivel de glucosa seleccionado. Finalmente, se realizan los gráficos con los datos que se toman en cada uno de estos procesos.

#### 5.3.3.1. Gestión de Mediciones

El diseño de interfaces se simplifica mucho por tener ya ciertos elementos diseñados de la parte de usuarios y que son reutilizables en este caso. El funcionamiento del formulario es el mismo pidiendo primero los datos obligatorios y posteriormente los opcionales. En este caso se organizan en tres ventanas que funcionan de la misma manera dentro de un *PageView*. Al iniciar la edición o adición se crea una instancia del servicio asociado donde se realizará la lógica principal del formulario y que servirá de copia de los datos cada vez que cambiemos de ventana yendo hacia adelante o hacia atrás dentro del formulario.

Al igual que con el formulario de usuarios, mucha información se recopila con valores simples de enteros como el tipo de comida o el momento de la medición y gracias a funciones de transformación se simplifica el trato de la información. El funcionamiento es realmente parecido con el anterior de usuarios, por lo que no hay demasiados detalles relevantes que incluir aquí.

Mencionar la posibilidad de seleccionar síntomas de un listado previamente definido en la aplicación y que contiene todos los síntomas posibles de la diabetes obtenido tras una búsqueda e información de los mismos en diferentes portales especializados.

Una vez que la medición se ha añadido, se vuelve al listado de las mismas el cual estará or-

denado por fecha y hora de las mediciones apareciendo primero las mediciones más recientes. Se ha optado por contener la mayor parte de información en la vista externa como el tipo de comida o el momento, para ser identificado rápidamente desde fuera.

### **5.3.3.2. Detalles de Medición: Sugerencias y Medicinas**

Cuando entramos a visualizar los detalles de una medición es posible ver dos secciones. La primera de ellas nos permite ver las sugerencias que coinciden con el nivel marcado de glucosa y que indica la cantidad y medicinas que deberían tomarse. Este proceso se hace dinámicamente sin necesidad de asociar la medición a la sugerencia, haciendo que si modificamos el nivel de glucosa se cambien automáticamente las posibles sugerencias recomendadas.

En cuanto a las medicinas, el funcionamiento es idéntico a cuando añadimos medicinas a una sugerencia o al usuario en sí. Se almacenan los cambios en un *buffer* para que, hasta no ser aceptados, no se realicen.

### **5.3.4. Iteración 3**

En la última iteración referente a la lógica de la aplicación y a las funcionalidades de la misma se pueden incluir todas las tareas que no se han realizado hasta este momento siendo quizás la iteración más compleja que se ha realizado hasta ahora. Los gráficos, el cambio de usuario, el apartado de nutrición o las notificaciones son los elementos a implementar en este último *sprint*.

Finalmente, el modelo de la base de datos ya es el definitivo, por lo que no es necesario revisarlo ni modificarlo para realizar estas tareas y de igual manera, la variedad de interfaces de las que ya se disponen agilizan el proceso para elaborar el resto faltante en las ventanas necesarias.

#### **5.3.4.1. Creación de Gráficos**

Los gráficos se realizan de manera sencilla gracias a la librería mencionada anteriormente *flutter\_charts*. Su generación es dinámica, siendo actualizados con cada medición que el usuario añade al sistema. Se ha optado así por dos tipos de diagramas: circular y barras.



El diagrama circular obtiene de las mediciones del usuario los tipos de comida que realiza y el momento en el que toma las mediciones. De esta manera se puede llevar un registro de cuáles son los momentos más activos del usuario a la hora de tomar las mediciones. Para cada uno de estos se dibuja un diagrama diferente en favor de tener una organización mejor de los datos.

El de barras, más interesante en lo personal, se centra en analizar las mediciones de los tres últimos meses obteniendo el momento del día en el que se han realizado y teniendo en cuenta el nivel de glucosa que se ha obtenido. De esta manera se puede ver una evolución en un período temporal suficiente para ver en qué momentos del día el usuario suele tener más problemas con respecto a la enfermedad. El análisis que se hace es seleccionar las mediciones referentes a los tres últimos meses de manera que una vez las tengamos, individualmente se obtienen cuatro grupos: uno por cada momento del día. Con un total de 12 agrupaciones diferentes, se realiza la media de los niveles de glucosa de cada una de ellas para dibujar las barras correctamente.

#### **5.3.4.2. Cambio de Usuario**

La aplicación estaba originalmente pensada para un usuario; sin embargo, una familia puede tener varios casos de diabetes y si por ejemplo un niño o una persona mayor tiene la enfermedad pero no acceso a un dispositivo móvil, es conveniente que sea otra la que lleve su control. Por ello, incluir varios usuarios y la posibilidad de cambiar entre ellos se consideró un aspecto útil para incluir.

Desde el mismo momento en que se inicia la aplicación también lo hace el servicio respectivo a las *SharedPreferences*, un método para almacenar dentro del sistema del dispositivo información permanente. Esta variable que se almacena referencia al identificador del último usuario que se estableció en la aplicación, por lo que cada vez que se cambia de usuario se modifica el valor de la variable, volviendo a la ventana inicial de carga que usará este nuevo identificador para cargar al usuario seleccionado.

El tratamiento de esta información no es realmente complicado y se incluyó por las ven-

tajas que añade respecto al tiempo que supone implementarlo.

#### 5.3.4.3. Apartado de Nutrición

Como se ha explicado, a través de la librería y web *openfoodfacts* se accede a un sistema de búsqueda que permite encontrar información nutricional de multitud de alimentos conociendo de primera mano si contiene ciertos elementos perjudiciales para una persona como posibles elementos alérgenos o trazas de cualquier ingrediente.

El funcionamiento de esta web es curioso en el sentido de que cualquier usuario del mundo puede incorporar nuevos alimentos con todos sus datos. Son revisados, claramente, pero el hecho de poder incluir alimentos de todo el mundo genera un conflicto respecto a las categorías y cierta información adicional que provee. Por ejemplo, cada alimento existente en la web tiene establecido un lenguaje principal que incluye un código de idioma como lo son *es* (España), *en* (Inglaterra) o *fr* (Francia), entre otros. Sin embargo, aunque el producto sea en ese idioma, muchos de sus datos, sean por defecto registrados por la web o añadidos por otros usuarios, pueden llegar en idiomas diferentes al principal. Ejemplo de esto es la existencia de productos españoles cuyas categorías y ciertos alérgenos llegan en francés. El número total de categorías sumando las que incorpora la web y las que son añadidas por usuarios supera las 81000, por lo que controlar su traducción no ha sido posible. Sí se controla, por ejemplo, cuando llega un alérgeno o traza que en la web está registrado, que en cuyo caso, es sustituido a su idioma correspondiente de los posibles de la aplicación a través de una transformación y análisis realizado en la misma.

Otro aspecto que es controlado es que el sistema operativo permite conocer el idioma principal y su código, por lo que en función del idioma en el que se encuentre la aplicación recibiremos productos cuyo idioma principal sea el que está seleccionado. Sigue ocurriendo el problema anterior, pero se filtra de manera que no aparezcan elementos en un idioma no controlado como puede ser el chino cuando la aplicación establece otro.

Realizando estos filtros, y con tan sólo añadir una palabra a la búsqueda, el usuario tendrá acceso a diferentes productos para consultar su información y saber de lo que contiene.

#### 5.3.4.4. Sistema de Notificaciones

Se ha considerado interesante también añadir la posibilidad de que el usuario sea el que configure sus propias notificaciones y cuando quiere que se reciban. Para ello, se incorpora un formulario con que añadir la notificación e incluir una descripción que será el mensaje que se muestre cuando esta aparece en el dispositivo.

Una vez añadida, a la hora marcada se mostrará la notificación. Para ello se ha usado la librería *local\_notifications* de Flutter en la que se permite establecer el día y hora que se emitirá la notificación, así como el mensaje y título que tendrá los sistemas operativos Android e iOS.

#### 5.3.5. Iteración 4

En la última iteración se terminará de cerrar el proyecto revisando el cumplimiento de todos los requisitos. Para ello se volverá al Documento General de Requisitos en el que se modificarán los diferentes diagramas de uso y secuencia, así como la información que aparece tras haber considerado muchos de los aspectos planteados al inicio y que durante el desarrollo han sido modificados por la aparición de nuevas ideas.

Además de revisar cada uno de los diagramas, se realizarán los diagramas de clases correspondientes a clases y servicios auxiliares que no se hubieran contemplado, así como corregir los que ya se tenían. Elaborar la memoria a raíz de estos y completar las secciones correspondientes tanto al **Análisis y Modelado** como al **Desarrollo e Implementación**. Se redactará la documentación restante referente a los apéndices como el **Manual de Instalación** haciendo referencia a cómo hacer funcionar la aplicación a través de una *apk* generada y por último, al **Manual de Usuario** que explicará con mayor detalle cada uno de los pasos y acciones que pueden realizarse en la aplicación para su correcto uso.

### 5.4. Informe de Pruebas

El proceso de pruebas si bien no se ha añadido al diagrama de trabajo ni a ninguna iteración es porque no se ha seguido una planificación fija para el mismo. Cada proceso de pruebas se ha realizado posterior a cada iteración o incluso en mitad de la misma a partir de la itera-

ción 2, cuando la aplicación tenía contenido suficiente para ser interesante realizar pruebas. El proceso seguido ha sido el de incorporar pruebas de **regresión** e **integración** principalmente, de manera que tras cada trabajo realizado en la aplicación pudiera hacerse una comprobación con el sistema funcionando en tiempo real de todos los elementos. Con esta metodología, no sólo los elementos nuevos se probaban, sino también los anteriores en complemento a las novedades.

El resultado de las pruebas es favorable, cumpliendo así el desarrollo y producto final con todos los requisitos propuestos al inicio. Toda esta información puede encontrarse más ampliada y detallada, así como ver los diferentes criterios seguidos o los casos de prueba que se han tenido en cuenta en el **Apéndice D**, el **Documento General de Pruebas**.



# 6

## Conclusiones y Líneas Futuras

### 6.1. Conclusiones

El proyecto ha terminado y es el momento de reflexionar acerca de cómo ha sido su avance y si ha cumplido con las expectativas que se tenían del mismo.

En primer lugar, fijándonos en las funcionalidades que se requerían de la aplicación, se consideran cumplidas. Si bien pueden existir errores no detectados en las pruebas, las funcionalidades básicas están bien incorporadas sin funcionamientos extraños. Además, con este proyecto no sólo se atendía la creación de la aplicación, sino también la presentación de un framework tan dinámico como lo es Flutter. En lo que se ha podido consultar de manera previa a la realización del TFG, Flutter no es una tecnología con la que se haya trabajado mucho en estos aspectos, seguramente por la poca madurez de la misma; sin embargo, tras haber aprendido desde una base sencilla y haber ido avanzando desde ella hasta dominar y tener un conocimiento casi se podría decir amplio de la misma se considera que Flutter es una herramienta perfecta para este tipo de desarrollos. La realización de interfaces no es extremadamente compleja haciéndose versátil y fluida, pero sobre todo mostrando rápidamente lo que se quiere hacer; sumado a un código intuitivo que, aun necesitando aprender las bases de Dart, opta a ser un lenguaje, bajo mi punto de vista, muy interesante para el futuro.

Este aprendizaje ha sido lo que se buscaba al realizar el trabajo con estas herramientas, ya que en un principio no me sentía cómodo trabajando con Java en Android Studio. Me parecía un desarrollo costoso y poco intuitivo en algunos aspectos; sin embargo, Flutter ha tenido un

interés real casi desde el principio y aprenderlo se considera de utilidad finalmente.

Por otro lado, hablando un poco sobre el tiempo dedicado y los plazos establecidos, cabe mencionar que a pesar de usar una metodología de trabajo ágil y dinámica en el sentido de poder adaptarse a la situación, ha resultado muy complicado ser estrictamente seguidor de la misma. Los plazos se han cumplido, acabando este proyecto previo a la fecha límite que se estableció, pero compaginar estudios, prácticas y este proyecto no ha hecho las cosas más fáciles. A pesar de esto, se considera que la planificación seguida ha sido buena cumpliendo a tiempo con todos los objetivos de la misma.

A nivel de aplicación se considera que los objetivos, tanto personales como de proyecto se han cumplido, diseñando una aplicación, a mi manera de ver agradable, que es usable por personas de cualquier edad. El factor sanitario que posee, así como la idea principal de ayudar a las personas ha sido una motivación suficiente para seguir con el proyecto y el resultado, aunque hubiera estado bien realizar alguna que otra funcionalidad más, se considera aceptable. A nivel personal se podría decir que estoy orgulloso del resultado conseguido siendo capaz tanto de hacer una aplicación desde cero como de aprender unas tecnologías de las que no tenía ni idea previamente para ello.

## **6.2. Líneas Futuras**

Hay muchas ideas que se han quedado en el tintero realmente. Al inicio de este proyecto las funcionalidades y requisitos que se pedían a la aplicación eran las básicas que se mencionaban en el *Anteproyecto*; sin embargo, a medida que se desarrollaba la aplicación final comenzaron a pensarse muchos requisitos que si bien no daría tiempo a implementarlos, sería interesante de verlos en un futuro. Esto no quita que el proyecto haya cumplido con los requisitos establecidos al inicio; sin embargo, siempre se quiere hacer algo más, sobretodo en dos situaciones: cuando inviertes tanto tiempo quieres que todo quede perfecto y por otro lado, cuando no sabes de algo, se sueña demasiado con cosas que no se saben hacer.

A pesar de esto, no se abandonan estas ideas que se tuvieron, sino que se adaptan para incluir las como posibles actualizaciones de la aplicación, ofreciendo al usuario mayor capacidad

y opciones para usar la aplicación:

- Incorporación de **mapas** a la aplicación de manera que una vez el usuario entrase en esta ventana y teniendo en cuenta su posición se pudiera mostrar el lugar más cercano tanto de hospitales, como centros de salud y farmacias. Se pretendía incluir esto con compatibilidad con Google Maps de manera que en todo momento pudiera enviarse el objetivo a la aplicación para indicar el camino hasta el lugar seleccionado.
- La aplicación trabaja a nivel local en lo que respecta a datos, por eso se pensó que el usuario perdería sus datos si cambia de dispositivo al no poder transferir sus datos al nuevo. Esto no es precisamente conveniente, por lo que se plantearon diferentes opciones con las que se pudieran tanto **exportar** como **importar** datos a la aplicación. De esta manera se consideraría la generación de un fichero o la comunicación *bluetooth* para realizar este paso, así ningún usuario perdería sus datos.
- A colación de lo anterior de migrar datos, otra posibilidad es incorporar directamente una vía online con la que poder compartir usuarios entre diferentes dispositivos. Así, si una persona mayor con la que no se convive está incapacitada, no tiene móvil o simplemente quiere comprobarse que su estado es bueno, podría tenerse acceso desde otro dispositivo para mantener un seguimiento.
- Por último, mejorar el sistema de información nutricional de manera que pudiera establecerse una búsqueda no sólo de alimentos, sino de platos preparados que pudieran valorarse nutricionalmente ofreciendo vías para obtener recetas. A raíz de esto, incluir un escáner de código de barras con el que poder filtrar y encontrar alimentos más fácilmente.

Además de esto, que son funciones que no están presentes, sobre lo que sí se ha implementado también hay sugerencias e ideas para mejorar, como hacer más sencillo el sistema de adición de medicinas en cada medición, pudiendo añadir directamente las medicinas de una sugerencia seleccionada, o modificar el perfil del usuario incorporando más posibilidades para los datos que se incluyen, entre otras cosas.



Las ideas no eran pocas y el deseo de incorporarlo a la aplicación final era tan deseable como imposible con el tiempo del que se ha dispuesto. Sin embargo, quedan anotados de manera que en futuras actualizaciones del producto puedan incorporarse tras haber tenido tiempo de trabajar en ellas hasta considerar un funcionamiento óptimo de las mismas.

# Referencias

- [1] “*La diabetes en España, en datos y gráficos*”. epData, 2020. [Online] Disponible: <https://www.epdata.es/datos/diabetes-espana-datos-graficos/472>
- [2] Dart. *Programming Language*. [Online] Disponible: <https://dart.dev>
- [3] Flutter. *Framework*. [Online] Disponible: <https://flutter.dev>
- [4] SQLite. *SQL Database*. [Online] Disponible: <https://www.sqlite.org>
- [5] sqflite. *SQLite for Flutter*. [Online] Disponible: <https://pub.dev/packages/sqflite>
- [6] Visual Studio Code. *Code Editor*. [Online] Disponible: <https://code.visualstudio.com>
- [7] Android Studio. *Integrated Development Environment (IDE)*. [Online] Disponible: <https://developer.android.com/studio>
- [8] Git. *Herramienta de Control de Versiones*. [Online] Disponible: <https://git-scm.com>
- [9] GitHub. *Repositorio de Código Online*. [Online] Disponible: <https://github.com>
- [10] Microsoft Teams. *Aplicación de Reuniones*. [Online] Disponible: <https://www.microsoft.com/es-es/microsoft-teams/group-chat-software>
- [11] Microsoft Word. *Procesamiento de Texto*. [Online] Disponible: <https://www.microsoft.com/es-es/microsoft-365/word>
- [12] LaTeX. *Sistema de Composición de Textos*. [Online] Disponible: <https://www.latex-project.org>
- [13] MiKTeX. *LaTeX Distribution*. [Online] Disponible: <https://miktex.org>
- [14] TexMaker. *Editor de Código Latex*. [Online] Disponible: <https://www.xm1math.net/texmaker>
- [15] Microsoft Forms. *Herramienta de Encuestas Online*. [Online] Disponible: <https://forms.office.com/>

- [16] VirtualBox VM. *Herramienta para ejecutar Máquinas Virtuales*. [Online] Disponible: <https://www.virtualbox.org/>
- [17] MoonModeler. *Herramienta de modelado de BBDD*. [Online] Disponible: <https://www.datansen.com/data-modeling/moon-modeler-for-databases.html>
- [18] Postman. *Herramienta para el desarrollo de APIs*. [Online] Disponible: <https://www.postman.com/>
- [19] DB Browser. *Herramienta para la gestión y visualización de BBDD*. [Online] Disponible: <https://sqlitebrowser.org/>
- [20] MagicDraw. *Herramienta CASE*. [Online] Disponible: <https://www.nomagic.com/products/magicdraw>
- [21] Hein Smith. “*Scrum: The Ultimate Beginner’s Guide To Learn and Master Scrum Agile*”. CreateSpace Independent Publishing Platform, 2018.
- [22] Marcos Sepúlveda Romero. “*UNICAR: una plataforma para compartir vehículos entre alumnos y personal de la UMA*”. RiUMA, 26-11-2020. [Online] Disponible: <https://riuma.uma.es/xmlui/handle/10630/20479>
- [23] Alejandro Luna Criado. “*Gestión de rutas turísticas para Málaga*”. RiUMA, 20-12-2018. [Online] Disponible: <https://riuma.uma.es/xmlui/handle/10630/17113>
- [24] “*Global Report On Diabetes*”. World Health Organization, 2016.
- [25] M. Mena. “*463 millones de personas viven con diabetes en el mundo, cifra que va en aumento*”. statista, Noviembre 2020. [Online] Disponible: <https://es.statista.com/grafico/6698/la-expansion-de-la-diabetes/>
- [26] “*The Top 10 Causes of Death*”. World Health Organization, Diciembre 2020. [Online] Disponible: <https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death>
- [27] “*Diabetes*”. World Health Organization, Junio 2020. [Online] Disponible: <https://www.who.int/news-room/fact-sheets/detail/diabetes>

- [28] Ángel B. E. *Opiniones Diabéticas*. Office Forms, Noviembre 2020. [Online] Disponible: <https://forms.office.com/r/cExURzh2ww>
- [29] Chris Sells. “*What’s New in Flutter 2*”. Medium, Marzo 2021. [Online] Disponible: <https://medium.com/flutter/whats-new-in-flutter-2-0-fe8e95ecc65>
- [30] Anónimo. “*Mobile Vs. Desktop Internet Usage (Latest 2020 Data)*”. BroadbandSearch. Actualizado en 2020. [Online] Disponible: <https://www.broadbandsearch.net/blog/mobile-desktop-internet-usage-statistics>
- [31] “*23º Navegantes en la Red*”. AIMC. Marzo 2021. [Online] Disponible: <http://download.aimc.es/aimc/cc8ke5T/macro2020/>
- [32] Laura S. “*Creemos que el móvil nos escucha, pero cada vez lo utilizamos más: así ha marcado la pandemia el uso de Internet*”. XatakaMóvil. Marzo 2021. [Online] Disponible: <https://www.xatakamovil.com/movil-y-sociedad/creemos-que-movil-nos-escucha-cada-vez-utilizamos-asi-ha-marcado-pandemia-uso-internet>
- [33] Mónica M. “*Android e iOS dominan el mercado de los smartphones*”. Statista. Julio 2020. [Online] Disponible: <https://es.statista.com/grafico/18920/cuota-de-mercado-mundial-de-smartphones-por-sistema-operativo/>
- [34] Fundación Diabetes. “*¿Qué es el día mundial de la diabetes?*”. Fundaciondiabetes.org. [Online] Disponible: <https://www.fundaciondiabetes.org/diamundial/328/que-es-el-dia-mundial-de-la-diabetes>
- [35] “*Open Food Facts*”. World.openfoodfacts.org. [Online] Disponible: <https://world.openfoodfacts.org/>
- [36] “*openfoodfacts*”. Dart packages. [Online] Disponible: <https://pub.dev/packages/openfoodfacts>
- [37] “*flutter\_local\_notifications*”. Dart packages. [Online] Disponible: [https://pub.dev/packages/flutter\\_local\\_notifications](https://pub.dev/packages/flutter_local_notifications)
- [38] “*flutter\_native\_timezone*”. Dart packages. [Online] Disponible: [https://pub.dev/packages/flutter\\_native\\_timezone](https://pub.dev/packages/flutter_native_timezone)

- [39] “*flutter\_localizations*”. Dart packages. [Online] Disponible:  
[https://pub.dev/packages/flutter\\_localizations](https://pub.dev/packages/flutter_localizations)
- [40] “*charts\_flutter*”. Dart packages. [Online] Disponible:  
[https://pub.dev/packages/charts\\_flutter](https://pub.dev/packages/charts_flutter)
- [41] “*EU General Data Protection Regulation (EU-GDPR)*”. Mayo 2016. [Online] Disponible:  
<https://www.privacy-regulation.eu/en/index.htm>
- [42] Klimaszewski Szymon. “*Diabetes - Diario de Glucosa*”. Google Play Store. [Online] Disponible:  
<https://play.google.com/store/apps/details?id=com.szyk.diabetes&hl=es&gl=US>
- [43] Ascensia Diabetes Care. “*Contour Diabetes App*”. [Online] Disponible:  
<https://www.diabetes.ascensia.es/products/contour-diabetes-app/>

# Apéndice A

# Manual de Usuario

En este apéndice se explicará cómo usar la aplicación una vez que está instalada incluyendo cada uno de los aspectos adicionales que incluye. El usuario podrá leer dicho apéndice de manera que se conozca cómo usar la aplicación en cada una de las funcionalidades que ofrece al mismo tiempo que visualiza imágenes complementarias para mejorar el entendimiento.

Como aclaración, mencionar que este apéndice no pretende explicar aspectos de diseño y/o desarrollo, sino mostrar las funcionalidades de la aplicación de manera sencilla.

## A.1. Introducción y Conceptos

Bienvenido/a al manual de usuario de la aplicación móvil *Mesuri* usado para transmitir conocimientos básicos acerca del uso de la misma. La aplicación *Mesuri* provee una vía sencilla para que cualquier persona de cualquier edad paciente de diabetes pueda disfrutar de un entorno sencillo para la gestión de sus mediciones diarias. De esta manera, con esta aplicación se podrá:

- Gestionar mediciones de uno o varios usuarios al mismo tiempo.
- Gestionar las medicinas tomadas en cada una de las mediciones.
- Añadir sugerencias por nivel de glucosa que serán recomendadas para cada una de las mediciones.
- Control de los síntomas padecidos en el período entre mediciones.
- Visualización de gráficos y estadísticas de las mediciones añadidas.
- Gestión de notificaciones para cada uno de los usuarios.
- Consulta de diferentes alimentos para conocer su valor nutricional.

En este momento en el que las personas quizás encuentran hábitos más sedentarios y las opciones para alimentarse son más variadas (y no por ello más sanas), *Mesuri* permite ayudar a gestionar los diferentes problemas que pueden derivar en diabetes.

## A.2. ¿Cómo acceder a Mesuri?

Mesuri es una aplicación sencilla que no requiere de registro o inicio de sesión ya que es el propio usuario el que se gestiona a sí mismo o, en su defecto, es gestionado por otra persona. Sin necesidad de establecer una conexión a Internet para prácticamente toda funcionalidad de Mesuri, el entorno que provee está preparado desde el inicio para usarse desde el propio momento de instalación.

### A.3. Funcionalidades

Las acciones que se pueden realizar con Mesuri son muy variadas. Desde la gestión de usuarios hasta la búsqueda de información nutricional de alimentos se contemplan entre ellas. Para explicar el funcionamiento de cada una de ellas se intentará explicar una ejecución normal de la aplicación.

#### A.3.1. Inicio y Primera Vista

Cuando Mesuri esté instalada en el dispositivo aparecerá con su respectivo icono que podrá ser pulsado para ser iniciada. Acto seguido, lo primero que se presentará es una pantalla de carga en la que se obtienen los datos del usuario mostrando de nuevo el icono de la aplicación en un tamaño mayor, así como un círculo que indicará el proceso de carga de la información inicial.

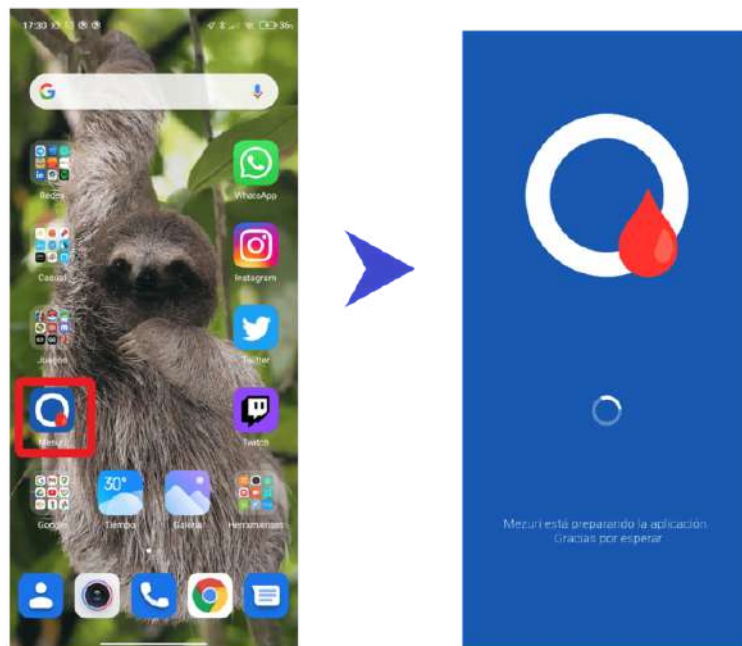


Figura 49: Icono y Pantalla de Carga de Mesuri

Cuando la aplicación finalmente está iniciada, se mostrará la que es considerada su ventana principal desde la que podremos acceder a diferentes aspectos de la aplicación: **Perfil**, **Registros**, **Estadísticas** y **Otros**. De ente todas estas vistas, la que mostrará información del



**Perfil** será la primera que se visualizará. Desde ella podremos acceder a diferentes opciones.



Figura 50: Barra de Navegación Principal

### A.3.2. Perfil de Usuario: Medicinas y Sugerencias

El perfil de usuario será la ventana en la que se visualizará la información del usuario activo y desde la que podremos tanto modificarlo como gestionar cada una de las medicinas y recomendaciones que tenga asociadas. En esta ventana encontraremos diferentes aspectos:



Figura 51: Acciones desde el Perfil del Usuario en Mesuri

- 1. Información:** Datos añadidos al perfil del usuario visibles desde esta ventana.
- 2. Botón Editar:** Botón que lleva al usuario hasta el formulario de edición de sus datos.
- 3. Medicinas:** Acceso al listado de medicinas añadidas por el usuario.
- 4. Sugerencias:** Acceso al listado de Sugerencias añadidas por el usuario.

### A.3.2.1. Edición de Perfil

Cuando se presiona el botón de **Editar**, Mesuri conduce al usuario a un formulario desde el que poder modificar sus datos. El formulario cuenta con dos secciones separadas: una para datos obligatorios y otra para datos opcionales.

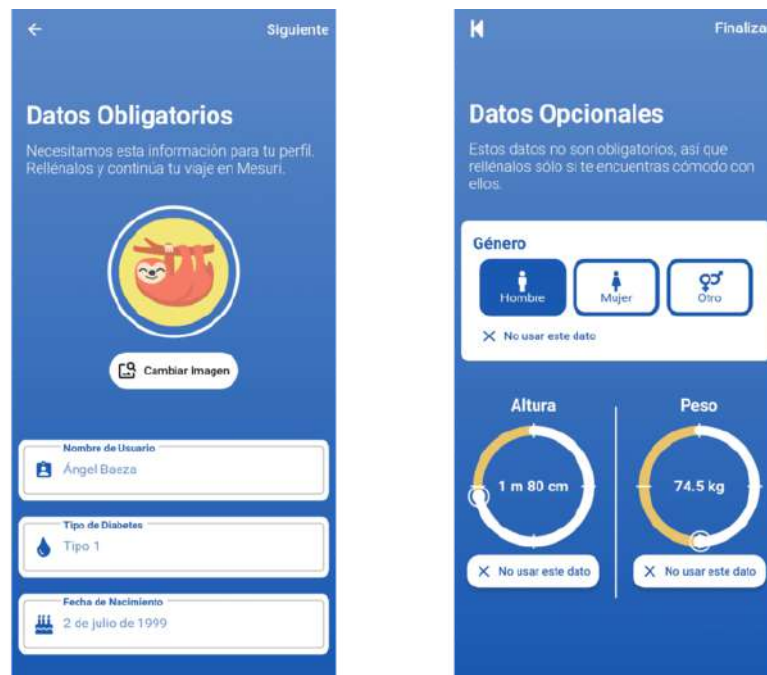


Figura 52: Edición de Usuario. Datos Obligatorios y Opcionales

El formulario de datos obligatorios incluye información como el nombre de usuario, el icono, el tipo de diabetes o la fecha de nacimiento. Cada uno de estos datos es, como es normal, obligatorio de introducir en la aplicación ofreciendo facilidades para cada uno de ellos.

Desde esta ventana [Fig. 53] se tendrán diferentes opciones para modificar los datos. El icono, por ejemplo, se modifica accediendo a una nueva ventana de selección [Fig. 54] a través del botón 'Cambiar Imagen' (3). Una vez abierta esta ventana, se podrá intercambiar la imagen aceptando o cancelando el cambio mediante los botones superiores.



Figura 53: Edición de Usuario. Acciones en Datos Obligatorios



Figura 54: Edición de Usuario. Ventana de Selección de Icono

Al igual que esta opción, también se abren diálogos y ventanas con las que seleccionar el tipo de diabetes (4) [Fig. 55], mostrando un listado de los diferentes tipos y haciendo que puedan seleccionarse, o la fecha de nacimiento (5), que permite seleccionar un día [Fig. 56].



Figura 55: Edición de Usuario. Selección de Diabetes



Figura 56: Edición de Usuario. Selección de Fecha

En todo momento se podrá cancelar la edición del perfil presionando el botón 'Volver Atrás'

(1) para cancelar la edición del perfil. Cuando los datos estén correctos se podrá presionar el botón ‘*Siguiente*’ donde se comprobará la validez de los datos y se pasará a la siguiente ventana: la de datos opcionales [Fig. 57].



Figura 57: Edición de Usuario. Acciones en Datos Opcionales

Desde esta ventana se tendrán siempre dos opciones principales: el botón de ‘*Ir Atrás*’ (6), para volver a la ventana de datos obligatorios, y el botón de ‘*Finalizar*’ para confirmar la edición de los datos. En cuanto a los datos, desde aquí se podrá seleccionar tanto el género (7) como el peso y altura (8) pudiendo en todo momento presionar el botón de ‘*No usar este dato*’ para dejarlo sin definir.

#### A.3.2.2. Listado de Medicinas

Si se accede desde la pestaña de **Perfil** [Fig. 51] a la sección de **Medicinas**, se mostrará en pantalla un listado con todas las medicinas. Desde aquí se podrá siempre ‘*Ir Atrás*’ (1) así como acceder al formulario de edición de medicinas (2) [Fig. 59] desde el que podremos añadir, editar o eliminar medicinas de nuestra selección.



Figura 58: Listado de Medicinas



Figura 59: Gestión de Medicinas. Acciones

Desde este formulario se podrá rellenar el campo de texto con el nombre de la medicina para ser añadida (3), así como editar (4) o eliminar (5) alguna que ya exista. En el caso de la edición [Fig. 60] se dispondrá de dos botones para las acciones correspondientes de 'Cancelar' (6) y 'Confirmar' (7), que guardará el cambio realizado en el nombre.



Figura 60: Gestión de Medicinas. Edición



Figura 61: Gestión de Medicinas. Borrado

Por último, si se presiona el icono de 'Eliminar' (5) de una medicina aparecerá una pequeña ventana de confirmación [Fig. 61] explicando las consecuencias del mismo: desaparecer

de cualquier sugerencia o medición en la que esté incluida. Cuando todos los cambios deseados estén realizados, se presionará el botón de 'Finalizar' (6), desde el que se confirmarán los cambios y serán aplicados.

### A.3.2.3. Listado de Sugerencias

Si se accede desde la pestaña de **Perfil** [Fig. 51] a la sección de **Sugerencias**. Esta, por las características que incluye se explicará en la sección siguiente.

### A.3.3. Sugerencias / Recomendaciones

Al acceder a la sección de **Sugerencias**, se mostrará en pantalla un listado con todas las sugerencias añadidas por el usuario. Si el listado no está vacío, es posible acceder a los **Detalles de una Sugerencia** presionando el botón correspondiente de la que se quiera consultar (1). Además, desde aquí es posible acceder al formulario para añadir alguna recomendación nueva (2) [Fig. 65].



Figura 62: Listado de Sugerencias

#### A.3.3.1. Detalles de una Sugerencia

Desde esta ventana [Fig. 63] se podrá ver en detalle aspectos de la sugerencia en cuestión, tanto un mensaje con los niveles de glucosa entre los que actuaría como el listado de medicinas

acompañado de las cantidades que se han indicado para cada una de ellas en dicha sugerencia.



Figura 63: Detalles de Sugerencia. Acciones



Figura 64: Detalles de Sugerencia. Borrado

Si en algún momento se quiere modificar la sugerencia se presiona el botón con el icono de 'Editar' (a) en la parte superior, accediendo al formulario de edición [Fig. 65], mientras que si la sugerencia o recomendación no fuese ya necesaria se puede eliminar con el botón marcado con el icono correspondiente (b). Si se elige esta última opción, aparecerá en pantalla un mensaje advirtiéndole de lo que se está haciendo [Fig. 64]. Si presionamos 'Eliminar', la sugerencia y toda su información será eliminada volviendo al listado anterior.

### A.3.3.2. Adición / Edición

Tanto si se presiona el botón del listado para añadir una sugerencia como si se presiona el de editar dentro de los detalles, la pantalla que se mostrará será un formulario con el que realizar estas acciones [Fig. 65].



Figura 65: Gestión de Sugerencias. Formulario Principal

Desde este formulario se seleccionan los niveles para la sugerencia (3), añadiendo un mensaje de texto (4) que ayudará a entender qué rango se está estableciendo. Cada sugerencia, además, debe incluir un listado de medicinas que serán las que se mostrarán posteriormente. Para acceder y modificar este listado se pulsa el botón de '*Modificar Medicinas*' (5) dejando ver la siguiente ventana con todas las medicinas añadidas [Fig. 66] previamente por el usuario.



Figura 66: Sugerencia. Gestión de Medicinas



Figura 67: Sugerencia. Gestión de Medicinas.  
Acciones



Una vez el formulario para añadir medicinas a una sugerencia está abierto, se puede encontrar un listado con las medicinas con dos tipos de medicinas: las que ya están añadidas, marcadas con un icono (7), y las que no están añadidas. En el momento de añadir una medicina nueva lo primero que debe hacerse es seleccionar dicha medicina quedando esta marcada (6) y escribir una cantidad adecuada en el campo de texto. Una vez hecho esto se presiona el botón 'Añadir' (8).

Para editar una de ellas debe seleccionarse una medicina añadida apareciendo así dos iconos [Fig. 67]. El primero de ellos (10) marcará la medicina (12) e incluirá dos acciones bajo el campo de texto [Fig. 68] que permitirán tanto 'Cancelar' (14) como 'Confirmar' (13) la edición.

Para eliminar, se selecciona el botón correspondiente (11) apareciendo así un mensaje explicando lo que sucederá [Fig. 69]. Si se confirma, la medicina será eliminada de la sugerencia.



Figura 68: Sugerencia. Gestión de Medicinas.

Edición

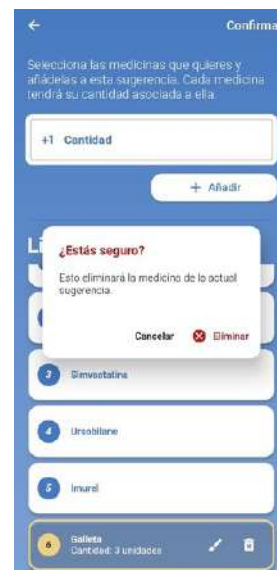


Figura 69: Sugerencia. Gestión de Medicinas.

Borrado

Cuando el proceso de gestión de las medicinas termina, se presiona el botón 'Confirmar' (9) volviendo al formulario de la sugerencia [Fig. 65] y desde el cual se podrá tanto 'Volver Atrás' como 'Finalizar' la adición o edición de la sugerencia.

### A.3.4. Mediciones

Otra de las secciones que se encuentra en la barra de navegación principal [Fig. 50] es la dedicada a los **Registros** o **Mediciones** de glucosa registrados por un usuario. Cuando se inicie la aplicación o al registrar un nuevo usuario este aparecerá vacío [Fig. 70]. Con tan sólo añadir uno, el listado mostrará diferentes mediciones detalladas de manera simple [Fig. 71].



Figura 70: Registro de Mediciones. Listado Vacío



Figura 71: Registro de Mediciones. Listado con Elementos

Desde este listado también es posible añadir nuevas mediciones a través del botón flotante inferior, con el que pasamos a un nuevo formulario [Fig. 75].

#### A.3.4.1. Detalles de Medición

Si se presiona en cualquiera de las mediciones añadidas al listado se accede a una vista en mayor detalle [Fig. 72] desde la que se accede a diversas opciones de la misma. En un inicio, aparecen datos visibles como el nivel de glucosa o el momento en el que se tomó la medición. También los síntomas, actividad física o la nota descriptiva en caso de que se haya añadido alguno de estos datos.

Desde esta vista en detalle se dispone de dos botones en la parte superior desde los que puede '*Editar*' (1) o '*Borrar*' (2) la medición mostrada.



Figura 72: Mediciones. Vista en Detalle

Por otra parte, cuando una medición es añadida es posible que se encuentre una sugerencia añadida para el valor de glucosa establecido. Para acceder a esta ventana se debe presionar el botón 'Sugerencias Específicas' (4) que dará acceso a un listado de las recomendaciones coincidentes [Fig. 73].



Figura 73: Medición en Detalle. Vista de las Sugerencias



Figura 74: Medición en Detalle. Gestión de Medicinas Tomadas

También es posible gestionar las medicinas tomadas en cada medición accediendo a través

del botón ‘*Medicinas Tomadas*’ al listado [Fig. 74]. Desde aquí, el proceso de adición, edición y borrado de medicinas tomadas es el mismo que el explicado para las **Sugerencias** en la **sección A.3.3.2**, por lo que no se explicará en detalle aquí.

#### A.3.4.2. Adición / Edición de Medición

Cuando se añade o se edita una medición la ventana mostrada es un formulario desde el que poder completar la información de la misma. Este se divide en tres pantallas: **datos obligatorios** [Fig. 75], **síntomas padecidos** [Fig. 76] y **datos opcionales** [Fig. 77].

Figura 75: Formulario de Medición. Datos Obligatorios

Figura 76: Formulario de Medición. Listado de Síntomas

En la ventana de datos obligatorios se rellenará información como el nivel de glucosa, así como el momento en el que se tomó la medición, entre otros. En todo momento podrá tanto ‘*Ir Atrás*’ (5), como avanzar al siguiente formulario, el de síntomas, con el botón ‘*Siguiete*’ (6).

El listado de síntomas permite al usuario escoger entre un listado los síntomas que ha padecido en el momento de tomar la medición. Se selecciona el que se desea incluir y se presiona el botón con el icono ‘*Más*’ (7) que aparece. Para ser eliminado, se sigue el mismo proceso.

Figura 77: Formulario de Medición. Datos Opcionales

Una vez se termine de incorporar síntomas a la medición y se presione el botón marcado como ‘*Siguiente*’ (8) se podrá avanzar al formulario con datos opcionales [Fig. 77] desde el que se podrá añadir información relativa a una actividad física como la intensidad y el tiempo dedicado a la misma (9). De manera adicional se podrá escribir una nota (10) que describa algún dato que el usuario vea necesario incorporar a la misma. En el caso de la actividad física es posible eliminarla mediante el botón ‘*No añadir ejercicio*’.

Para finalizar la edición o adición se presiona el botón ‘*Finalizar*’ (11) de la barra superior para verificar los cambios que hemos realizado. En cada uno de los casos se dirigirá a la vista en detalle o al listado, respectivamente. Ambos con los datos actualizados.

### A.3.4.3. Borrado de Mediciones

Desde la vista en detalle [Fig. 72] es posible eliminar una medición a través del botón ‘Eliminar’ de la parte superior. Cuando se presiona, se muestra un mensaje de confirmación [Fig. 78]. Al confirmar el borrado, la vista volverá al listado de mediciones.



Figura 78: Mediciones. Borrado de una Medición

### A.3.5. Gráficos y Estadísticas

Los **Gráficos** y la visualización de los mismos son parte de la tercera sección de la barra de navegación principal [Fig. 50]. Desde esta ventana, el usuario podrá ver estadísticas construidas a partir de sus mediciones basadas en el momento del día y el estado en el que las toma [Fig. 79], así como construir una comparación de los tres últimos meses en cuanto a la media de sus niveles de glucosa [Fig. 80].



Figura 79: Estadísticas. Diagrama de Sectores de Tiempo

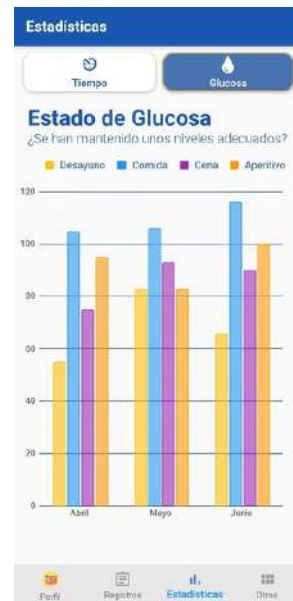


Figura 80: Estadísticas. Diagrama Comparativo de los Últimos Meses

Es importante saber que si no es posible construir los gráficos por la falta de mediciones se mostrará un mensaje al respecto mencionando el problema [Fig. 81].



Figura 81: Estadísticas. Falta de Mediciones

### A.3.6. Otros

Por último, la sección final presente en la barra de navegación principal [Fig. 50] corresponde al apartado '**Otros**', desde el cual se puede acceder a otras opciones de la aplicación no tan comunes para un usuario normal. Concretamente se puede acceder al **listado de usuarios** (1) presentes en la aplicación, una sección en la que establecer **notificaciones propias** (3) y por último, la sección desde la cual consultar el **valor nutricional** (2) de varios alimentos.

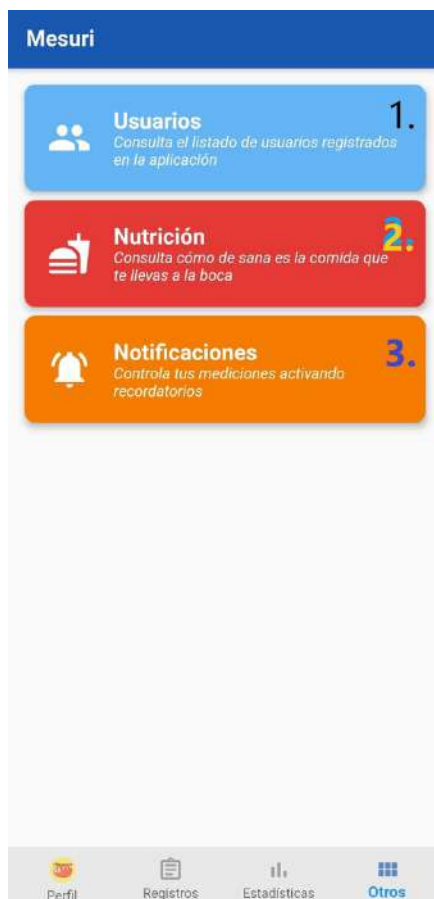


Figura 82: Otros. Listado de Opciones



### A.3.7. Usuarios

Desde la sección '*Usuarios*' se tiene acceso al listado de usuarios de la aplicación de manera que puedan añadirse, eliminarse o intercambiarse entre ellos dentro de la aplicación. Al iniciarse aparecerá un indicador de cuál es el usuario seleccionado actualmente.



Figura 83: Usuarios. Listado

#### A.3.7.1. Adición de Usuario

Desde este listado se pueden añadir nuevos usuarios a la aplicación. Para ello se presiona el botón inferior '*Añadir*' de manera que al presionarlo se accede al formulario, esta vez vacío, visto en la **sección A.3.2.1**, con un funcionamiento idéntico al mismo.

#### A.3.7.2. Intercambio de Usuario

Dentro del listado de usuarios [Fig. 83], cuando un usuario es seleccionado aparecen sobre él dos iconos que permiten interactuar con el mismo (1). Cada icono tiene un comportamiento y función diferente, siendo el de la izquierda (2) el que permite cambiar el usuario principal de la aplicación. Una vez se seleccione este usuario se dispondrá de nuevo el inicio de la aplicación (pestaña '*Perfil*') con el nuevo usuario y sus datos cargados.

*Nota. Si se selecciona el usuario actual, no es posible realizar esta acción sobre él.*

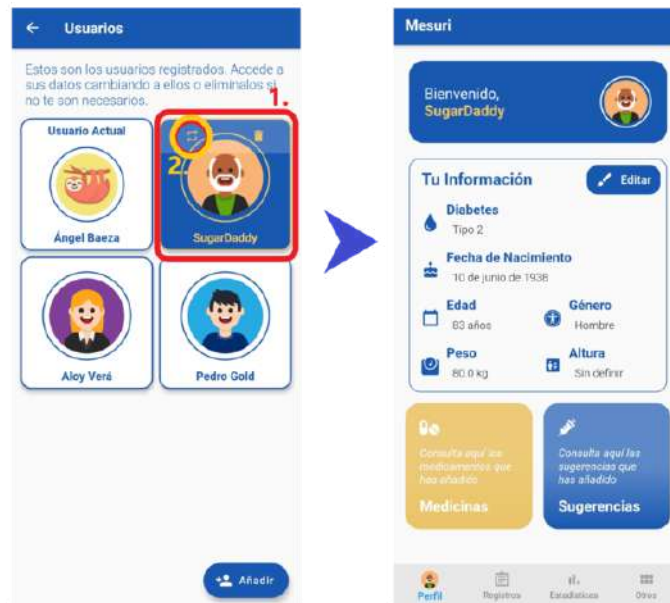


Figura 84: Usuarios. Intercambio de Usuario Principal

### A.3.7.3. Eliminación de un Usuario

De manera similar al intercambio de usuario, es posible eliminar usuarios desde el listado de usuarios [Fig. 83]. Al seleccionar un usuario (1) aparecen dos iconos sobre él. En este caso, el icono de la derecha es el encargado de 'Eliminar'. Una vez presionado aparecerá un mensaje de confirmación advirtiendo de que todos los datos del usuario se borrarán. Si confirmamos esta eliminación, el listado de usuarios se actualizará eliminando el usuario.

*Nota. Si se selecciona el usuario actual, no es posible realizar esta acción sobre él.*

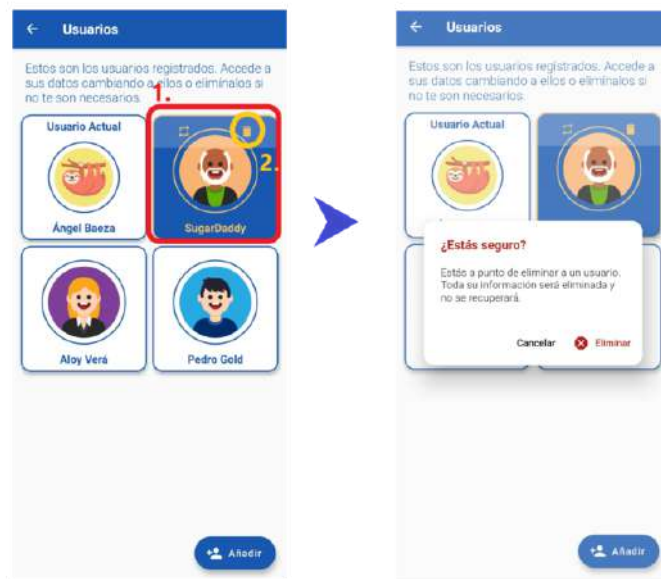


Figura 85: Usuarios. Eliminación de un Usuario

### A.3.8. Notificaciones

Las **Notificaciones** son una sección más de Mesuri en la que el usuario podrá definir sus propios avisos para que mediante una notificación del dispositivo, este le avise de que es momento de realizar una medición (o lo que el usuario prefiera). Al acceder a esta ventana desde '*Otros*' [Fig. 82], se muestra el listado con las notificaciones activas.

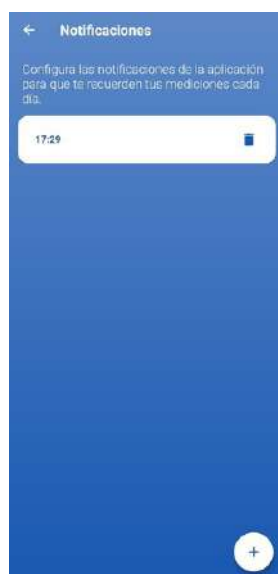


Figura 86: Notificaciones. Listado

### A.3.8.1. Adición de una Notificación

Desde el propio listado, si se desea añadir una nueva notificación basta con presionar el botón inferior con el que abrir el formulario de notificaciones [Fig. 87]. Este formulario es algo simple donde el usuario debe introducir la hora a la que quiere la medición, así como una descripción para la misma, siendo esta última opcional.

Una vez que la notificación haya sido añadida, la aplicación nos redirigirá al listado, de manera que la notificación se mostrará en el momento adecuado.

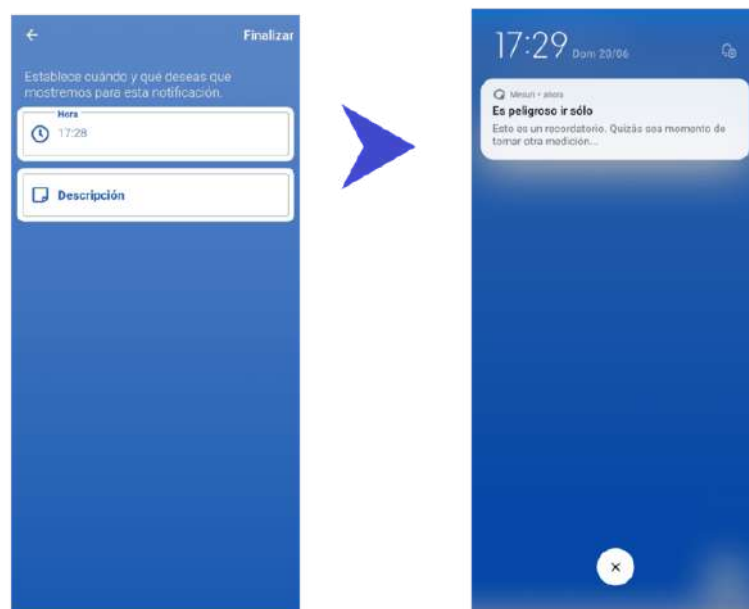


Figura 87: Notificaciones. Adición de una Notificación

### A.3.8.2. Borrado de una Notificación

Desde el propio listado, cada notificación añadida muestra un icono con el cual puede eliminarse fácilmente (1). Presionando este se mostrará un mensaje de confirmación con el que, una vez confirmado se eliminará la notificación haciendo que no muestre más dicho aviso.

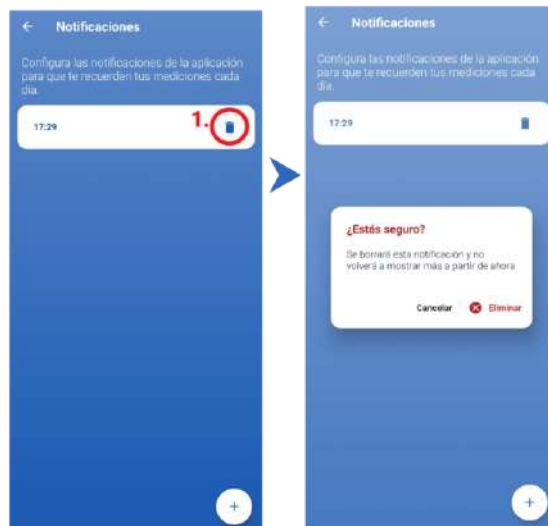


Figura 88: Notificaciones. Borrado de una Notificación

### A.3.9. Nutrición

La última sección accesible pertenece a '**Nutrición**', un lugar dentro de Mesuri que, mediante conexión a Internet, permite acceder a información nutricional de numerosos alimentos gracias a sencillas búsquedas.



Figura 89: Nutrición. Ventana Principal

### A.3.9.1. Búsqueda de Alimentos

Desde la ventana principal [Fig. 90] el funcionamiento es tan sencillo como escribir una palabra en el campo de texto y presionar el botón de ‘*Buscar*’ (1). Tras unos segundos de búsqueda (depende de la conexión a internet del dispositivo aparecerán en pantalla resultados de los alimentos que coincidan con lo que se ha buscado.

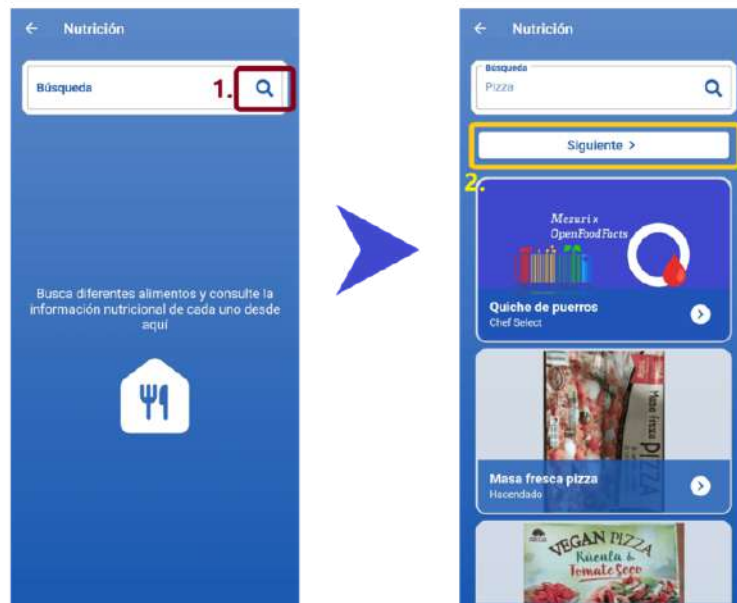


Figura 90: Nutrición. Búsqueda de Alimentos

Sin embargo, es común que nos lleguen gran cantidad de alimentos resultado de la búsqueda, de manera que podremos navegar entre diferentes páginas presionando el botón ‘*Siguiente*’ (2) con el que avanzaremos en el listado [Fig. 91]. Siempre es posible ir a una página anterior desde el botón ‘*Anterior*’ (3).

Es posible además que una búsqueda realizada no tenga resultados, lo que mostrará un mensaje de aviso al usuario acerca del problema [Fig. 92].



Figura 91: Nutrición. Búsqueda de Alimentos en Varias Páginas

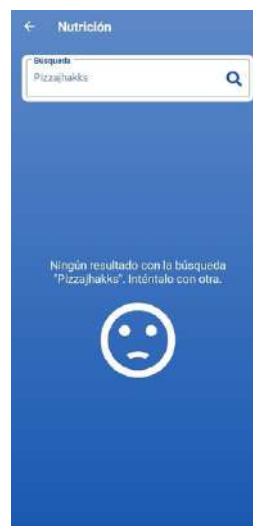


Figura 92: Nutrición. Búsqueda sin Resultados

### A.3.9.2. Consulta de Información Nutricional de un Alimento

Cuando se encuentre el alimento que se buscaba o sobre el que hay algún interés basta con seleccionarlo en el listado para abrir una vista detallada del mismo que incluye tanto ingredientes e información básica como aspectos nutricionales (grasas, azúcares...).

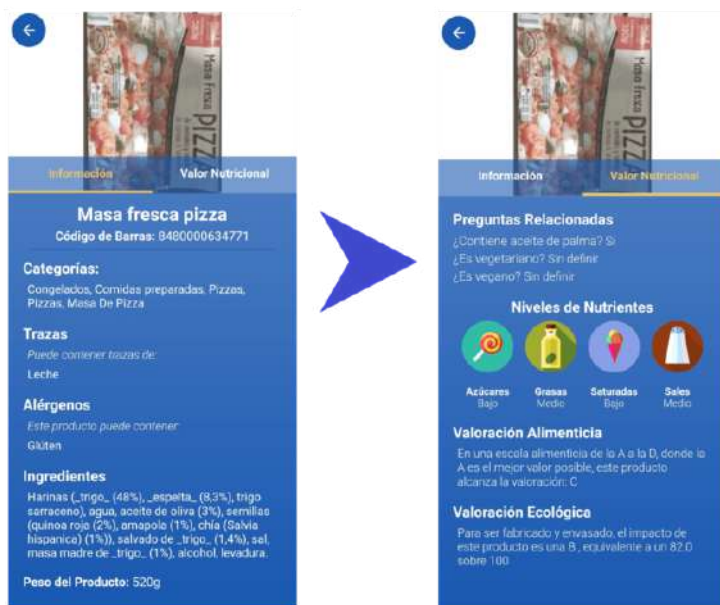


Figura 93: Nutrición. Información Detallada de un Alimento

## A.4. Detalles Adicionales

Las funcionalidades principales de la aplicación son las mostradas previamente; sin embargo, Mesuri añade ciertos elementos a nivel de interfaz y capacidades que deben conocerse.

### A.4.1. Soporte para Diferentes Idiomas

Mesuri está preparado totalmente para ser una aplicación multilingüaje. Aunque tenga soporte para español, su idioma por defecto ha sido configurado en inglés y ofrece traducción directa para cada uno de los elementos y textos que incluye.

Esta traducción se acoge al idioma en el que se encuentra el dispositivo y su sistema operativo, de manera que si el dispositivo se configura con una región demográfica fuera de España, la misma se mostrará en inglés. Toda la aplicación seguirá funcionando correctamente.

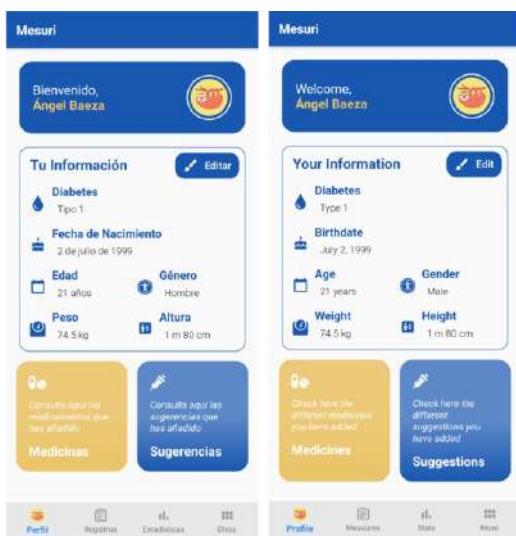


Figura 94: Multilingüaje. Comparación del Perfil



Figura 95: Multilingüaje. Comparación de Estadísticas



#### A.4.2. Soporte para Modo Oscuro

Prácticamente cada dispositivo hoy en día permite alternar entre los modos claro y oscuro ya sea dependiendo de la hora del día o del gusto del usuario. Por este motivo, Mesuri incluye un modo oscuro propio establecido cada vez que el sistema se encuentra con este activado.

Si bien se han querido mantener los colores propios de Mesuri, el modo oscuro usa unos tonos más apagados de los mismos manteniendo la apariencia usual de la aplicación. Este modo puede resultar cómodo y agradable para muchos usuarios que no quieran disponer de un brillo excesivo en sus dispositivos.

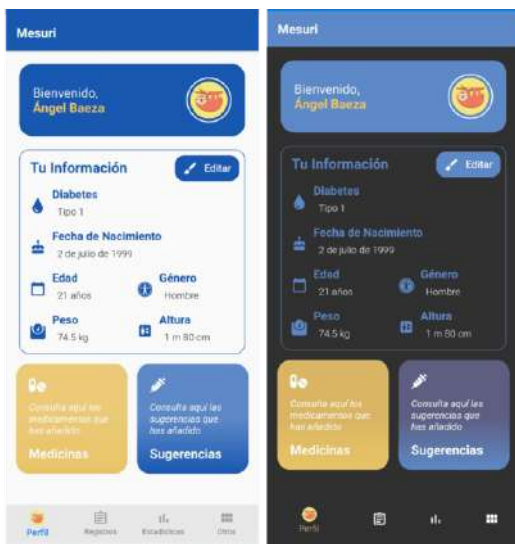


Figura 96: Modo Oscuro. Comparación del Perfil



Figura 97: Modo Oscuro. Comparación de Estadísticas

### A.4.3. Adaptación al Tamaño de Texto del Dispositivo

De igual forma que con el modo oscuro, muchos dispositivos incluyen la posibilidad de adaptar el tamaño de texto a diferentes situaciones. Una persona a la que le cueste un esfuerzo mayor visualizar elementos puede ampliar el tamaño de letra del dispositivo adaptando así cualquier elemento.

Mesuri cuenta por este motivo con un sistema de adaptación a esta característica nativa de cualquier entorno móvil para ajustar su interfaz en lo posible a todos los tamaños posibles.

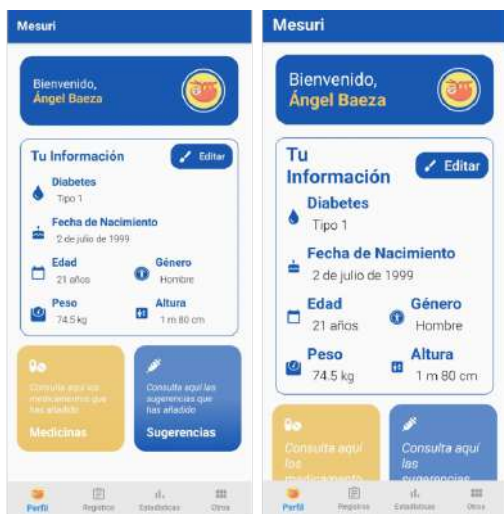


Figura 98: Tamaño de Texto. Comparación del Perfil



Figura 99: Tamaño de Texto. Comparación de Nutrición

## A.5. Consideraciones Finales

**Mesuri** es una herramienta sencilla que pretende servir de ayuda a personas que lo necesitan. Si encuentra algún error o quiere comunicarse con el desarrollador por algún otro motivo es posible hacerlo a través del correo asociado: [estecorreonoexiste@gmail.com](mailto:estecorreonoexiste@gmail.com).

Esperamos que disfrute de la aplicación.



# Apéndice B

# Manual de Instalación

En este apéndice se describirá el proceso de instalación de la aplicación contando con los entregables de la misma. Como se ha mencionado, la entrega contiene no sólo el código de la aplicación, sino también bases de datos y paquetes de aplicaciones (*apk*) de Android tanto vacías como de prueba que pueden usarse para visualizar diferentes versiones de la aplicación.

## B.1. Aspectos Importantes

Para facilitar la instalación se manejará principalmente la versión de Android instalable a través de la *apk* adjunta. Aunque el código de la aplicación estará presente y puede probarse directamente desde ahí. Sin embargo, no se explicará la instalación de Flutter o Dart, sino que se tendrá en cuenta un entorno donde esté configurado para ello.

## B.2. Recursos

Con la extracción de la aplicación encontramos diferentes ficheros y archivos que serán necesarios para ejecutar la instalación de la misma. A modo de referencia para las siguientes secciones y procesos que se explicarán se detallarán los archivos junto a una breve descripción de los mismos:

- **Código Fuente / Proyecto:** Carpeta con todos los archivos necesarios para ejecutar el proyecto. Contiene tanto código como recursos necesarios para iniciar la aplicación en cualquier dispositivo, así como en un emulador.
- **Bases de Datos:** Estructura con los datos que la aplicación usa para su funcionamiento. Se incluyen dos:
  - ◇ *bd\_empty.db*: Esta estructura es la existente en la aplicación por defecto. Cuando la aplicación se instale se configura esta base de datos que contiene un usuario por defecto, sin nada más en su interior.
  - ◇ *bd\_demo.db*: Base de datos con datos suficientes como para emular un uso temporal de la aplicación. Contiene diferentes usuarios, así como medicinas y mediciones de algunos de ellos para visualizar posibles escenarios de la aplicación.
- **Instalables (*apk*):** Archivo ejecutable para el sistema operativo Android que permite una instalación sencilla de aplicación con todos los aspectos ya configurados, así como la base de datos. Se entregan dos:
  - ◇ *mesuri\_init.apk*: Aplicación por defecto emulando una instalación desde la *Play Store*. Usa la base de datos *bd\_empty.db*.

- ◇ *mesuri\_demo.apk*: Aplicación emulando una aplicación que lleva un tiempo en funcionamiento. Usa la base de datos *bd\_demo.db*.

## B.3. Instalación

A partir de este momento la instalación o la prueba de la aplicación puede realizarse de diferentes maneras: **ejecutando el código fuente**, ya sea en dispositivo físico o emulador, o **instalando una *apk*** en un dispositivo con sistema operativo Android.

### B.3.1. Ejecución del Código Fuente

*Nota. Como se ha mencionado, no se explicará la instalación y puesta a punto de Flutter y Dart. Se considera que no entra dentro del concepto del manual. Su instalación para poder ejecutar este método se puede encontrar en la página oficial de Dart<sup>2</sup> y Flutter<sup>3</sup>, respectivamente.*

*Nota Adicional. Se recomienda que si se procede a este método se configure desde Visual Studio Code o Android Studio, ambos explicados dentro de la guía de Flutter.*

Cuando se descomprime la aplicación y se obtiene la carpeta con todo el código fuente de la misma es necesario realizar varios pasos. El primero es disponer de toda la configuración de Dart y Flutter en el equipo en el que se esté probando. Una vez todo el entorno esté configurado, tendremos acceso a comandos de Flutter y habrá que manejar cómo es la aplicación que queremos mostrar.

#### B.3.1.1. Configuración de la Base de Datos

La aplicación puede mostrarse con una configuración por defecto o con unos datos pre-establecidos para ver el uso de la misma. Esto depende de la base de datos que usemos de las definidas previamente. Para añadir la base de datos que se desee es necesario conocer la estructura del proyecto. Los pasos para configurar la base de datos son:

---

<sup>2</sup>Instalación del SDK de Dart: <https://dart.dev/get-dart>

<sup>3</sup>Instalación de Flutter: <https://flutter.dev/docs/get-started/install>

1. Dentro de la carpeta del proyecto, acceder a la carpeta marcada como **assets**, donde se encuentran los recursos de la aplicación.
2. En el interior de esta carpeta se encuentra una carpeta llamada **database**. En esta carpeta es donde situaremos la base de datos que se quiera mostrar.
3. Escoger la base de datos de las disponibles con la entrega (*bd\_empty.db*, *bd\_demo.db*) y copiarla en la carpeta **database**.
4. Cuando la base de datos seleccionada se encuentre en la carpeta hay que cambiar su nombre a '**mesuri**', siendo este el nombre que está configurado en los archivos.



Figura 100: Instalación. Configuración de la Base de Datos

### B.3.1.2. Ejecución en Emulador

Para este paso es esencial disponer de un emulador Android configurado en el equipo. Si no se encuentra instalado, debe configurarse antes de ser ejecutado. Android Studio provee de herramientas sencillas para ello <sup>4</sup>.

### B.3.1.3. Ejecución en Dispositivo Físico

Para poder ejecutar la aplicación en un dispositivo físico es necesario que el mismo se encuentre con las opciones de desarrollador activadas. Junto a esto, debe estar activa la de-

---

<sup>4</sup>Configuración de emulador en Android Studio: <https://developer.android.com/studio/run/emulator?hl=es-419>

puración en el mismo. Cada dispositivo podría tener las opciones en un lugar diferente de los ajustes, por lo que sería necesario buscarlas para poder realizar este paso.

En cualquier caso, si el método de ejecución será este, se recomienda instalar directamente las APK bajo el proceso descrito en la sección **B.3.2, Instalación de una APK**.

#### **B.3.1.4. Ejecución de la Aplicación**

Cuando el emulador esté activo o el dispositivo físico conectado al equipo y configurado, dependiendo del programa desde el que estemos ejecutando podrían diferir las opciones para hacerlo funcionar; sin embargo, a favor de ser lo más neutro posible, se plantea:

1. Acceder a la consola de comandos del equipo desde la carpeta principal del proyecto.
2. Ejecutar el comando **flutter run**.

Este método comenzará a hacer una versión de la aplicación para instalarla en el emulador o dispositivo. En el caso del emulador sólo podrá ejecutarse en modo *debug*, mientras que en el dispositivo físico se podrá añadir opcionalmente el flag - - **release** para una versión final. Posteriormente, debe aceptarse la instalación y la aplicación estará funcionando correctamente en el lugar escogido.

#### **B.3.2. Instalación de una APK**

Este proceso se considera más sencillo que los descritos anteriormente debido a que únicamente hay que pasar el paquete comprimido deseado, que es la propia **APK**. En la entrega se han adjuntado dos de estos archivos (*mesuri\_init.apk*, *mesuri\_demo.apk*) que contienen la aplicación en las diferentes versiones. Los pasos a seguir son:

1. Con el dispositivo conectado, copiar la APK que se quiera probar en cualquier directorio interno de este.
2. Una vez dentro y comprobado que se ha copiado bien el archivo, desconectar el dispositivo. No es obligatorio este paso, pero es recomendable.



3. Desde el gestor de archivos del dispositivo, acceder al directorio al que se ha añadido y seleccionar el archivo comprimido APK.

4. Seleccionada la APK, comenzar su instalación.

◇ Es posible que este paso no sea realizable si no se activa la opción de *'Permitir aplicaciones de origen desconocido'*. ES conveniente así activar esta hasta que la aplicación esté instalada. Posteriormente, puede volver a desactivarse.

◇ Cuando la aplicación se abre para instalar es común que aparezcan avisos sobre que no se reconoce al desarrollador. Estos avisos aparecen debido a que la aplicación está firmada, pero no subida a la PlayStore. Dicho esto, simplemente actuar bajo el procedimiento mostrado a continuación [Fig. 101].



Figura 101: Proceso de Instalación de una APK. Pasos a seguir en el Dispositivo

# Apéndice C

# Documento General de Requisitos

Apéndice donde se tratarán aspectos relacionados con los requisitos de la aplicación desarrollada de manera extensa y clara. Se consta aquí del **Documento General de Requisitos** (DGR) donde se extiende la información mostrada con anterioridad en la sección 4 del documento principal.

## DOCUMENTO GENERAL DE REQUISITOS

---

**Clave:** TFG-DIABETES-DGR

**Nº Páginas:** 74

Aplicación para Control y

**Título:** Ayuda al Paciente de Diabetes

DOCUMENTO GENERAL DE REQUISITOS

## C.1. Introducción

### C.1.1. Historial de Versiones

Fecha	Versión	Autor	Organización	Descripción
13/03/2021	1.0	Ángel Baeza	Universidad de Málaga	Captura de Requisitos
18/06/2021	2.0	Ángel Baeza	Universidad de Málaga	Revisión de Requisitos

Tabla 10: DGR. Historial de Versiones Realizadas

### C.1.2. Información del Proyecto

<b>Empresa / Organización</b>	Universidad de Málaga
<b>Proyecto</b>	Aplicación Móvil para el Control y Ayuda al Paciente de Diabetes
<b>Fecha de Preparación</b>	13 de Marzo de 2021
<b>Gerente / Líder de Proyecto</b>	Ángel Baeza
<b>Líder de Análisis</b>	Ángel Baeza

Tabla 11: DGR. Información del Proyecto

### C.1.3. Aprobaciones

Nombre y Apellido	Cargo	Departamento	Fecha	Firma
Ángel Baeza	Responsable	Ingeniería Software	xx/xx/2021	
Gabriel Jesús Luque	Tutor	LCC	xx/06/2021	
Christian Cintrano	Cotutor	LCC	xx/06/2021	

Tabla 12: DGR. Aprobaciones del Proyecto

#### C.1.4. Objetivo del Documento

El objetivo de este documento es recoger, analizar y definir las características y necesidades de alto nivel del sistema **ApMovCAD** en una versión inicial del mismo. Se centrará en describir las expectativas de cada una de las partes del proyecto y de los usuarios finales y por qué estas necesidades existen. Los detalles de cómo **ApMovCAD** cumple con estas necesidades se detallan en los casos de uso y en las especificaciones adicionales.

#### C.1.5. Alcance del Producto

El principal objetivo del desarrollo de una **Aplicación Móvil para el Control y Ayuda al Paciente de Diabetes** (*ApMovCAD*) es facilitar un sistema de registros y seguimiento de la enfermedad de la Diabetes a personas o familiares que sufran de esta brindándoles la ayuda necesaria para ello.

Otro objetivo del proyecto es el desarrollo de una aplicación móvil centrada en tecnologías novedosas y en auge como lo es el framework Flutter a través del uso del lenguaje Dart para su elaboración.

Ambos objetivos aportarán beneficio tanto en el sector académico como en el médico. A nivel académico este proyecto podrá verse como un primer referente en el desarrollo móvil con tecnologías ya mencionadas; por otro lado, en el sector sanitario, la aparición de una aplicación que aporte y ayude a pacientes de diabetes.

#### C.1.6. Definiciones, Acrónimos y Abreviaturas

Algunas de los términos usados en este documento aparecen mediante sus siglas o son demasiado técnicos para su entendimiento. Estos son:

- **UMA.** Universidad de Málaga. Institución académica situada en Málaga representante del alumno desarrollador del producto.
- **RGPD.** Reglamento General de Protección de Datos. Documento que reúne las leyes y restricciones respecto al tratamiento de datos de carácter privado o personal en el ámbito

digital. En vigor desde 2016 bajo la redacción y acuerdo de los países pertenecientes a la Unión Europea.

- **AEPD.** Agencia Española de Protección de Datos. Autoridad encargada de controlar el cumplimiento de las leyes sobre protección de datos en España. Esto incluye el RGPD.
- **DGR.** Documento General de Requisitos. Documento propio de la Ingeniería de Requisitos que reúne aquellos conceptos y funcionalidades que un producto en desarrollo debe cumplir así como el entorno en el que se produce y entra en funcionamiento el mismo.
- **UE.** Unión Europea. Entidad política que reúne a gran parte de los países europeos creando una asociación económica y política unida en favor de Europa.
- **ApMovCAD.** Aplicación Móvil para el Control y Asistencia al paciente de Diabetes. Producto que se quiere desarrollar.
- **Bluetooth.** Tecnología que usa redes inalámbricas para la transmisión de información entre dos dispositivos.
- **Android.** Sistema operativo presente en la mayoría de dispositivos que permite el funcionamiento de los mismos.
- **iOS.** Sistema operativo propio de dispositivos Apple. Competencia de Android.
- **Diabetes.** Enfermedad donde los niveles de glucosa alcanzan valores elevados en la sangre ante la falta o mal uso de la insulina producida por el cuerpo que puede derivar en problemas más graves de no ser tratada.

### C.1.7. Resumen

El presente Documento General de Requisitos (DGR) pretende mostrar todos los aspectos y requerimientos del proyecto desarrollado. Para ello se estructura en diferentes secciones donde se detalla cada uno de los aspectos importantes.

El documento comienza con una introducción al proyecto realizado donde se incluirá un glosario y se comentarán los objetivos del mismo. Estos conceptos permitirán comprender cuál

es la vía del proyecto y los participantes que estarán involucrados tanto en el uso como en el desarrollo. Esto se explicará en los puntos 3 y 4.

La sección 4 explicará elementos del producto final en torno al despliegue o coste del mismo. De manera breve contendrá las características que posteriormente se explicarán en las secciones siguientes que van de la 5 a la 8. Todas estas serán un listado y explicación de los requisitos que requerirá el producto final en términos de funcionalidades, uso, mantenimiento o documentación.

Finalmente, como resultado de todos los requisitos se elaborarán diagramas que simulen un entorno inicial para el dominio de la aplicación y se explicarán las características funcionales (Sección 11). Como última sección habrá un espacio para las cuestiones que hayan surgido durante la elaboración del documento y que no hayan podido resolverse.

## **C.2. Directivas del Proyecto**

### **C.2.1. Oportunidades de Negocio**

El proyecto en sí mismo no se ajusta a una filosofía de negocio debido a la naturaleza sanitaria y académica del mismo. Si bien podría hacerse dando lugar a entornos médicos que ocupasen esta línea, no es la finalidad principal del mismo.

### **C.2.2. Descripción del Problema**

El problema de la Diabetes afecta cada día a más personas en unos rangos de edad que van reduciéndose cada vez más. Si bien el paciente estándar hoy día de la enfermedad es una persona mayor a los 65 años, los números se están invirtiendo considerablemente. Y no sólo eso, sino que su aumento también es muy elevado en los últimos años [25].

El impacto del problema es evidente: una población joven con problemas en su organismo a los que una diabetes podría derivar en otros problemas mayores [27] llegando incluso a alcanzar una muerte prematura por su mala gestión o desconocimiento de la misma [26].

Una solución satisfactoria sería la existencia de una aplicación, independientemente del entorno usado para ella, que provea de las herramientas necesarias para que cualquier persona que padezca la enfermedad sea capaz de gestionarla y obtener información sobre la misma que le pueda ser de utilidad.

### **C.2.3. Descripción del Producto**

Teniendo siempre en cuenta el punto de vista de usuarios que padezcan esta enfermedad, así como de personas cercanas encargadas de controlarla, el producto conocido como ApMovCAD pretende ser una herramienta sencilla que permita ayudar a controlar la diabetes de manera diaria con un manejo rápido, eficaz y, sobre todo, sencillo.

Frente a otros productos del mercado que permiten el registro de mediciones de glucosa, explicados en la sección 3.5, ApMovCAD ofrece aspectos diferenciadores concentrando en



una única aplicación el registro de mediciones, tratamiento, información nutricional y mapa de referencias para el usuario.

### C.3. Descripción de Participantes y Usuarios

#### C.3.1. Resumen de los Participantes

Nombre	Rol
UMA	Cliente Principal.
AEPD	Participante Indirecto. Legislación.

Tabla 13: DGR. Participantes en el Proyecto

#### C.3.2. Resumen y Entorno de los Usuarios

La aplicación permite la gestión de información propia únicamente. Esto lleva a que sólo haya un tipo de usuario con capacidad de borrar, añadir, editar y buscar información dentro de la aplicación. Aún así, el usuario se puede dividir en función de su rango de edad.

Nombre	Descripción	Participante
Usuario	Acceso y registro de datos en la aplicación. Capacidad de editar, borrar y buscar información de la aplicación.	-

Tabla 14: DGR. Usuarios en el Proyecto

### C.3.3. Perfiles de los Participantes

#### C.3.3.1. UMA

<b>Representante</b>	Gabriel Jesús Luque Polo
<b>Tipo</b>	Evaluador y Asistente del Proyecto.
<b>Responsabilidades</b>	Proveer asistencia al alumno encargado del TFG.
<b>Criterio de Éxito</b>	Superación del TFG por el alumno.
<b>Entregables</b>	Conocimientos. Evaluación de producto.
<b>Comentarios</b>	-

Tabla 15: DGR. Representante de la UMA (1)

<b>Representante</b>	Cristian Cintrano López
<b>Tipo</b>	Evaluador y Asistente del Proyecto.
<b>Responsabilidades</b>	Proveer asistencia al alumno encargado del TFG.
<b>Criterio de Éxito</b>	Superación del TFG por el alumno.
<b>Entregables</b>	Conocimientos. Evaluación de producto.
<b>Comentarios</b>	-

Tabla 16: DGR. Representante de la UMA (2)

#### C.3.3.2. AEPD

<b>Representante</b>	-
<b>Tipo</b>	Institución. Legislativo.
<b>Responsabilidades</b>	Asegurar y facilitar el cumplimiento de la normativa de protección de datos.
<b>Criterio de Éxito</b>	-
<b>Entregables</b>	Legislaciones y Reglas sobre la protección de datos.
<b>Comentarios</b>	Como participante indirecto, sus requisitos son implícitos.

Tabla 17: DGR. Representante de AEPD

### C.3.4. Perfiles de Usuario

#### C.3.4.1. Usuario Temprano (menor a 16 años)

<b>Representante</b>	Usuario de edad temprana.
<b>Descripción</b>	Perteneciente al grupo de edad menor a los 16 años.
<b>Tipo</b>	-
<b>Responsabilidades</b>	Correcto uso del sistema.
<b>Criterios de Éxito</b>	Correcta adaptación de los usuarios de esta aplicación al uso básico. Comodidad en el uso del SW. Variedad de opciones.
<b>Comentarios</b>	Se espera que el usuario tenga conocimientos algo menores en el uso de aplicaciones móviles pudiendo acceder así a los contenidos íntegros de la aplicación, pero centrándose en elementos básicos. Existe la posibilidad de no disponer de móvil y ser monitorizado por una persona de mayor edad.

Tabla 18: DGR. Perfil de Usuario Joven

#### C.3.4.2. Usuario Estándar (16 a 55 años)

<b>Representante</b>	Usuario Estándar.
<b>Descripción</b>	Perteneciente al grupo de edad comprendido entre los 16 y 35 años.
<b>Tipo</b>	-
<b>Responsabilidades</b>	Correcto uso del sistema.
<b>Criterios de Éxito</b>	Correcta adaptación de los usuarios de esta aplicación. Comodidad en el uso del SW. Variedad de funciones.
<b>Comentarios</b>	Se espera que el usuario tenga conocimientos avanzados en el uso de aplicaciones móviles pudiendo acceder así a los contenidos íntegros de la aplicación.

Tabla 19: DGR. Perfil de Usuario Estándar

### C.3.4.3. Usuario de Edad Avanzada (mayor a 55 años)

<b>Representante</b>	Usuario de edad avanzada.
<b>Descripción</b>	Perteneciente al grupo de edad que supera los 55 años.
<b>Tipo</b>	-
<b>Responsabilidades</b>	Correcto uso del sistema.
<b>Criterios de Éxito</b>	Correcta adaptación de los usuarios de esta aplicación al uso básico. Comodidad en el uso del SW.
<b>Comentarios</b>	Se espera que el usuario tenga conocimientos bajos en el uso de aplicaciones móviles pudiendo acceder así a los contenidos íntegros de la aplicación, pero centrándose en elemento básicos.

Tabla 20: DGR. Perfil de Usuario Mayor

### C.3.5. Alternativas y Competencia

Antes de comenzar el desarrollo del producto se ha investigado sobre las opciones actuales en el mercado que podrían servir de competencia bajo el caso de acabar con la comercialización del proyecto. Al atender a estas opciones han surgido elementos clave dentro del presente producto llegando a barajar diferentes vías de desarrollo al encontrar en ellas opciones que han dado lugar a diferentes requisitos a valorar para la inclusión en este producto.

En la búsqueda de ambos criterios se han encontrado varias aplicaciones con características similares al producto que se intenta desarrollar. Además de añadir ciertas referencias sobre las mismas, se explicarán algunos conceptos.

#### C.3.5.1. Contour Diabetes App

Contour Diabetes App [43] es una aplicación móvil que permite añadir datos referentes a comidas, mediciones y ejercicio físico por cada registro, mostrando además gráficas construidas con estos valores. Está pensada para ser sincronizada con dispositivos *Contour* con los que obtener valores más fiables, pero funciona también sin necesidad de ellos. Almacena los datos

del usuario en la nube para su posterior tratamiento médico.

Su diseño colorido sirve de apoyo a usuarios de edad avanzada para localizar bien los elementos incluyendo además textos en gran tamaño en favor de una buena lectura de los mismos. Su uso es sencillo y eficiente; sin embargo, se han encontrado elementos a mejorar como que transita entre pantallas de manera rápida sin cuestionar, lo que podría llevar a errores del usuario.



Figura 102: ContourApp. Principal

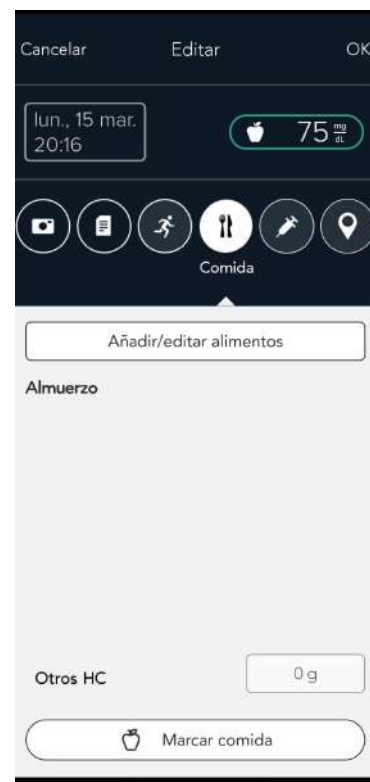


Figura 103: CountourApp. Añadir Medición

Se ha considerado interesante esta aplicación para obtener ideas acerca de las funcionalidades y sobre los datos que pueden recogerse al realizar las mediciones, así como del diseño y la manera de mostrar la información. El registro de comidas o de medicación se han considerado útiles a pesar de contar ya con una idea aproximada del mismo por la forma en que permite usar esta información de manera útil tanto para usuario como para médico.

### C.3.5.2. Diabetes - Diario de Glucosa [42]

Es una aplicación básica con la que tomar registros de glucosa. Sus opciones son bajas; sin embargo, su interfaz es mucho más visual para cualquier usuario siendo sencilla y eficaz. Permite añadir cantidad de glucosa en sangre, notas y etiquetas dentro de la aplicación de manera que permite consultar así un listado de las que hemos añadido o filtrar por estas etiquetas. Incluye una sección de estadísticas donde poder ver la media de las mediciones tomadas en rangos diferentes de tiempo (incluyendo mínima y máxima).

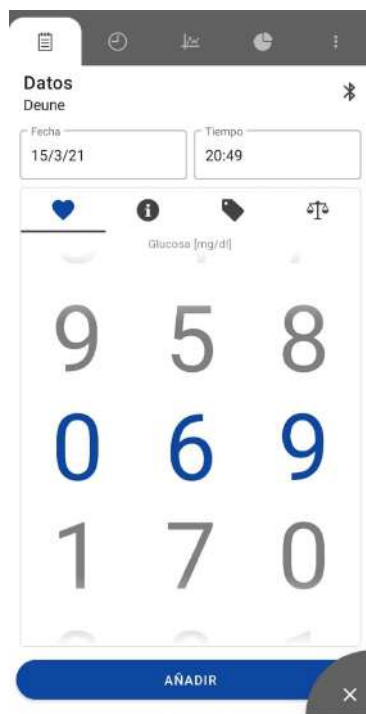


Figura 104: DiabetesApp. Principal



Figura 105: DiabetesApp. Listado de Registros

La aplicación trabaja a nivel local con la posibilidad de añadir diferentes usuarios con datos diferentes sin necesidad de registro. Soluciona la portabilidad de datos a través de exportar/importar archivos de texto generados desde la propia aplicación que pueden enviarse por cualquier aplicación de mensajería o incluso *Bluetooth*. Precisamente son estas ideas las que se han tomado para la aplicación ya que al estar pensada para personas mayores, no tienen por qué contar con una dirección de correo electrónico pudiendo así transferir sus datos de uno a otro dispositivo de manera rápida en caso de que fuese necesario.

## **C.4. Visión General del Producto**

### **C.4.1. Entorno de Despliegue**

#### **C.4.1.1. Entorno para la Implementación del Sistema Actual**

La aplicación a desarrollar se implementará en dispositivos móviles cuyo sistema operativo sea tanto Android como iOS. El sistema debe así hacer frente a un desarrollo en diferentes entornos proporcionando un rendimiento nativo al mismo. Cualquier usuario con un dispositivo móvil debe poder acceder a la instalación y uso de la aplicación independientemente del modelo o marca del mismo.

La aplicación debe funcionar en entornos sin conexión estable a una red. En caso de que no haya conexión, la aplicación permitirá el uso de la aplicación en cuanto a sus funcionalidades básicas. Si existe esta conexión, varias de las funcionalidades se activarán para el usuario para su uso.

#### **C.4.1.2. Aplicaciones Colaboradoras**

El entorno de la aplicación no requiere de aplicaciones externas para su funcionamiento; sin embargo, en el acceso a diversas funcionalidades se hará uso de diversas API y librerías que proporcionen servicios dentro del producto final. La API principal será:

- **OpenFoodFacts.** Dispone una base de datos completada por usuarios de todo el mundo desde la cual se puede consultar información nutricional de varios alimentos acercando así una guía al usuario en cuanto a la comida que puede comer.

#### **C.4.1.3. Paquetes Comerciales**

Con el fin de lograr alcanzar los requisitos del producto se hará uso de aplicaciones o paquetes comerciales que permitan el desarrollo de la misma.

Dentro del entorno de desarrollo se codificará con Visual Studio Code[6], un entorno con capacidad multilenguaje que añade la posibilidad de incorporar plugins afines que faciliten



el desarrollo. En este período se hará uso de herramientas que favorezcan el testing como lo son Android Studio [7], quien permitirá crear y gestionar emuladores en un equipo para la visualización de la aplicación previa al uso de un dispositivo real. Estos emuladores se cargarán y ejecutarán usando VirtualBox [16], creando así una máquina virtual Android.

#### **C.4.2. Suposiciones y Dependencias**

##### **C.4.2.1. Factores Externos que tienen un Efecto en el Producto**

Durante el desarrollo y posterior mantenimiento del producto pueden aparecer diferentes factores que obliguen a cambiar o realizar un nuevo análisis de tecnologías y aplicaciones usadas para ello.

Por un lado, la API para obtener información sobre diferentes alimentos puede verse alterada o modificada incluyendo elementos de pago o un uso limitado en el momento de realizar peticiones sobre la misma. De igual manera, Flutter [3] y Dart [2] son lenguajes tempranos que varían muy rápidamente, lo que puede suponer que varios de los elementos desarrollados queden sin soporte o impliquen un cambio en la aplicación para que sigan funcionando correctamente.

##### **C.4.2.2. Suposiciones que asume el Equipo en torno al Proyecto**

Las mayores suposiciones realizadas durante el proyecto corresponde al rango de edad de los usuarios y su predisposición al utilizar la aplicación. Si bien hay estudios e información que apoyan los rangos establecidos, las personas pueden cambiar sus hábitos rápidamente.

#### **C.4.3. Precio y Coste**

De acuerdo a la naturaleza de la aplicación, no se presupone un coste económico de la misma. Las herramientas usadas son de carácter gratuito al no ser usadas con fines comerciales y/o al ser licenciadas por la institución de la Universidad de Málaga. A nivel temporal se espera que el producto quede realizado en un tiempo de 4 meses con una estimación de 296 horas de trabajo realizado dividido en diferentes iteraciones o fases.

En cuanto al precio, al ser este producto el resultado de un Trabajo de Fin de Grado, no se valora la posibilidad de cobrar por él.

## **C.5. Requisitos Funcionales**

La aplicación desarrollada debe cumplir con ciertos requisitos de acuerdo al funcionamiento esperado de la aplicación. Estos irán reflejados bajo el indicador **RF** y agrupados en diferentes categorías según su implicación en el producto.

### **C.5.1. Datos de Usuario (USR)**

**RF-USR-1.** El sistema permite la existencia de varios usuarios.

**RF-USR-1.1.** El sistema permite cambiar entre usuarios.

**RF-USR-1.2.** Un usuario puede ser añadido al sistema.

**RF-USR-1.3.** Un usuario puede ser eliminado del sistema.

**RF-USR-2.** El usuario puede gestionar sus datos del perfil.

**RF-USR-2.1.** El usuario puede introducir datos opcionales.

**RF-USR-2.2.** El usuario puede modificar sus datos previamente introducidos.

**RF-USR-2.3.** El usuario puede eliminar ciertos datos no obligatorios.

**RF-USR-2.4.** El usuario puede consultar sus datos previamente introducidos.

**RF-USR-3.** El usuario puede gestionar su propio listado de medicinas.

**RF-USR-3.1.** El usuario puede añadir medicinas.

**RF-USR-3.2.** El usuario puede consultar el listado de medicinas introducidas previamente.

**RF-USR-3.3.** El usuario puede modificar las medicinas añadidas previamente.

**RF-USR-3.4.** El usuario puede eliminar sus medicinas introducidas previamente.

**RF-USR-4.** El usuario puede gestionar su propio listado de sugerencias de medicinas.

**RF-USR-4.1.** El usuario puede añadir nuevas recomendaciones.

**RF-USR-4.2.** El usuario puede consultar el listado recomendaciones.

**RF-USR-4.3.** El usuario puede modificar las recomendaciones añadidas previamente.

**RF-USR-4.4.** El usuario puede eliminar sus recomendaciones introducidas previamente.

**RF-USR-4.5.** El usuario puede consultar las medicinas añadidas a cada recomendación.

**RF-USR-4.6.** El usuario puede gestionar las medicinas añadidas a cada recomendación.

### **C.5.2. Mediciones (MSR)**

**RF-MSR-5.** El usuario puede gestionar las mediciones de glucosa en relación a la enfermedad.

**RF-MSR-5.1.** El usuario puede añadir estas mediciones incorporando datos opcionales.

**RF-MSR-5.2.** El usuario puede consultar un listado de las mediciones introducidas.

**RF-MSR-5.3.** El usuario puede consultar información de mediciones individuales.

**RF-MSR-5.4.** El usuario puede introducir datos nuevos tras la inserción de mediciones.

**RF-MSR-5.5.** El usuario puede modificar datos de las mediciones introducidas previamente.

**RF-MSR-5.6.** El usuario puede eliminar las mediciones.

**RF-MSR-6.** El usuario puede consultar las recomendaciones asociadas a una medición de acuerdo al nivel de glucosa.

**RF-MSR-7.** El usuario puede gestionar las medicinas de cada medición.

**RF-MSR-7.1.** El usuario puede añadir medicinas al listado de medicinas tomadas de una medición.

**RF-MSR-7.2.** El usuario puede consultar el listado de medicinas tomadas de una medición.

**RF-MSR-7.3.** El usuario puede modificar medicinas del listado de medicinas tomadas de una medición.

**RF-MSR-7.4.** El usuario puede eliminar medicinas del listado de medicinas tomadas de una medición.

### **C.5.3. Información (INFO)**

**RF-INFO-8.** El usuario puede consultar gráficas acerca de sus mediciones dentro del sistema.

### **C.5.4. Notificaciones (NTF)**

**RF-NTF-9.** El usuario puede gestionar notificaciones que quiera recibir de la aplicación.

**RF-NTF-9.1.** El usuario puede añadir alarmas / notificaciones programadas para introducir mediciones.

**RF-NTF-9.2.** El usuario puede eliminar las alarmas / notificaciones previamente establecidas.

### **C.5.5. Alimentación y Nutrición (FOOD)**

**RF-FOOD-10.** El usuario tiene acceso a un listado de alimentos para obtener información nutricional.

**RF-FOOD-10.1.** El usuario puede realizar búsquedas para obtener estos alimentos.

**RF-FOOD-10.2.** El usuario puede consultar los alimentos resultado de una búsqueda.

## C.6. Casos de Uso (CU)

### C.6.1. Actores

Para el funcionamiento del sistema se han identificado diferentes tipos de usuarios que podrían acceder a la aplicación desarrollada. Se considera a estos usuarios como **actores** en los siguientes casos de uso.

Nombre	Rol
Usuario	Representa al usuario que accede a la aplicación ya sea por primera vez o reiteradamente.

Tabla 21: DGR. Actores identificados en los Casos de Uso

### C.6.2. Casos de Uso

Todos los casos de uso representan escenarios de la aplicación a desarrollar de manera que, además de los casos de uso explicados, se incluye un diagrama de los mismos agrupados por temática de los mismos.

#### C.6.2.1. Datos de Usuario (USR)



Figura 106: CU-USR: Diagrama de Casos de Uso

<b>ID</b>	CU-USR-01
<b>Título</b>	Acceso al Listado de Usuarios
<b>Descripción</b>	Visualización del listado de usuarios presentes en la aplicación.
<b>Pre-Condición</b>	-
<b>Post-Condición</b>	Listado de Usuarios mostrado en pantalla.
<b>Prioridad</b>	Alta
<b>Reqs. Tratados</b>	RF-USR-1.1, RF-USR-1.2, RF-USR-1.3
<b>Escenario Principal</b>	
<ol style="list-style-type: none"> <li>1. El usuario selecciona la pestaña ‘<i>Otros</i>’ en la barra de navegación.</li> <li>2. El usuario presiona el botón ‘<i>Usuarios</i>’.</li> <li>3. El sistema muestra el listado de usuarios añadidos.</li> </ol>	

Tabla 22: CU-USR-01: Acceso al Listado de Usuarios

<b>ID</b>	CU-USR-02
<b>Título</b>	Intercambio de Usuario
<b>Descripción</b>	El usuario actual quiere cambiar de perfil en la aplicación para cambiar de usuario y ver sus datos.
<b>Pre-Condición</b>	Varios usuarios añadidos al sistema.
<b>Post-Condición</b>	Aplicación mostrando los datos de un usuario diferentes a los que mostraba al inicio.
<b>Prioridad</b>	Media
<b>Reqs. Tratados</b>	RF-USR-1.1
<b>Escenario Principal</b>	
<ol style="list-style-type: none"> <li>1. El usuario realiza los pasos del <b>CU-USR-01</b> para acceder al listado de usuarios.</li> <li>2. El usuario presiona sobre el usuario al que quiere cambiar.</li> <li>3. El usuario presiona el botón de ‘<i>Cambiar Usuario</i>’.</li> <li>4. El sistema modifica los datos que se están mostrando.</li> <li>5. El sistema retorna a la página principal de la aplicación con los datos del nuevo usuario.</li> </ol>	

Tabla 23: CU-USR-02: Intercambio de Usuario

<b>ID</b>	CU-USR-03
<b>Título</b>	Registro de Usuarios
<b>Descripción</b>	El usuario quiere añadirse a sí mismo (o a otra persona) a la aplicación para comenzar la gestión de sus datos.
<b>Pre-Condición</b>	Existencia previa de algún usuario en el sistema.
<b>Post-Condición</b>	Nuevo usuario creado en el sistema.
<b>Prioridad</b>	Media
<b>Reqs. Tratados</b>	RF-USR-1.2
<b>Escenario Principal</b>	
<ol style="list-style-type: none"> <li>1. El usuario realiza los pasos del <b>CU-USR-01</b> para acceder al listado de usuarios.</li> <li>2. El usuario presiona el botón 'Añadir'.</li> <li>3. El sistema muestra al usuario un formulario y pide los datos obligatorios para el nuevo usuario.</li> <li>4. El usuario rellena el formulario a través de los campos nombre de usuario, fecha de nacimiento y tipo de diabetes.</li> <li>5. El usuario presiona el botón 'Siguiete'.</li> <li>6. El sistema valida los datos introducidos.</li> <li>7. El sistema presenta un formulario con datos opcionales a rellenar.</li> <li>8. El usuario rellena los campos opcionales como género, altura y peso.</li> <li>9. El usuario presiona el botón 'Finalizar'.</li> <li>10. El sistema valida los datos introducidos.</li> <li>11. El sistema lleva al usuario al listado de usuarios.</li> </ol>	
<b>Escenario Alternativo 1</b>	
<ol style="list-style-type: none"> <li>6. El sistema registra un error en los datos.</li> <li>7. El sistema lanza un mensaje para avisar al usuario y le permite corregirlos.</li> </ol> <p><i>Retorno al paso 3 del escenario principal.</i></p>	
<b>Escenario Alternativo 2</b>	
<ol style="list-style-type: none"> <li>10. El sistema registra un error en los datos.</li> <li>11. El sistema lanza un mensaje para avisar al usuario y le permite corregirlos.</li> </ol> <p><i>Retorno al paso 7 del escenario principal.</i></p>	

Tabla 24: CU-USR-03: Registro de Usuarios



<b>ID</b>	CU-USR-04
<b>Título</b>	Eliminación de Usuario
<b>Descripción</b>	Un usuario quiere borrarse a él mismo o a otro usuario de la aplicación.
<b>Pre-Condición</b>	Existencia de algún usuario en el sistema.
<b>Post-Condición</b>	Usuario no existente en el sistema y datos del usuario borrado eliminados. Usuario actual en el listado de usuarios.
<b>Prioridad</b>	Media
<b>Reqs. Tratados</b>	RF-USR-1.3
<b>Escenario Principal</b>	
<ol style="list-style-type: none"> <li>1. El usuario realiza los pasos del <b>CU-USR-01</b> para acceder al listado de usuarios.</li> <li>2. El usuario selecciona el perfil que quiere eliminar.</li> <li>3. El usuario presiona el botón 'Eliminar'.</li> <li>4. El sistema lanza un mensaje de confirmación.</li> <li>5. El usuario presiona la opción 'Confirmar'.</li> <li>6. El sistema comprueba los usuarios del sistema.</li> <li>7. El sistema elimina el usuario seleccionado y todos sus datos.</li> <li>8. El sistema retorna al listado de usuarios.</li> </ol>	
<b>Escenario Alternativo 1</b>	
<ol style="list-style-type: none"> <li>5. El usuario presiona el botón de 'Cancelar'.</li> <li>6. El sistema muestra el listado de usuarios.</li> </ol> <p><i>Retorno al paso 1 del escenario principal.</i></p>	
<b>Escenario Alternativo 2</b>	
<ol style="list-style-type: none"> <li>2. El usuario selecciona el único usuario presente en el sistema.</li> <li>3. El sistema no muestra el botón 'Eliminar'.</li> </ol>	

Tabla 25: CU-USR-04: Eliminación de Usuario

<b>ID</b>	CU-USR-05
<b>Título</b>	Consulta de Datos de un Usuario
<b>Descripción</b>	El usuario quiere consultar sus datos introducidos para la elaboración de su perfil.
<b>Pre-Condición</b>	Existencia del usuario en el sistema. Usuario del que se quieren consultar los datos seleccionado en la aplicación.

<b>Post-Condición</b>	Datos del usuario mostrados en pantalla.
<b>Prioridad</b>	Alta
<b>Reqs. Tratados</b>	RF-USR-2.4
<b>Escenario Principal</b>	
<ol style="list-style-type: none"> <li>1. El usuario presiona sobre el botón <i>'Perfil'</i> en la barra de navegación principal.</li> <li>2. El sistema muestra los datos del usuario.</li> </ol>	

Tabla 26: CU-USR-05: Consulta de Datos de un Usuario

<b>ID</b>	CU-USR-06
<b>Título</b>	Edición de Datos del Perfil
<b>Descripción</b>	El usuario quiere gestionar (añadir, modificar o eliminar) información añadida a su perfil.
<b>Pre-Condición</b>	Existencia del usuario en el sistema. Usuario del que se quieren gestionar los datos seleccionado en la aplicación.
<b>Post-Condición</b>	Datos del usuario cambiados correctamente. Usuario en su perfil visualizando los datos personales.
<b>Prioridad</b>	Alta
<b>Reqs. Tratados</b>	RF-USR-2.1, RF-USR-2.2, RF-USR-2.3
<b>Escenario Principal</b>	
<ol style="list-style-type: none"> <li>1. El usuario realiza el caso de uso <b>CU-USR-05</b> accediendo así a su información personal.</li> <li>2. El usuario presiona el botón <i>'Editar'</i>.</li> <li>3. El sistema abre un formulario con los datos del usuario.</li> <li>4. El usuario añade, elimina o modifica los datos que desea cambiar.</li> <li>5. El usuario presiona el botón <i>'Aceptar'</i>.</li> <li>6. El sistema valida los campos del formulario.</li> <li>7. El sistema retorna a la página de perfil del usuario.</li> </ol>	
<b>Escenario Alternativo 1</b>	
<ol style="list-style-type: none"> <li>5. El usuario presiona el botón <i>'Cancelar'</i>.</li> </ol> <p><i>Retorno al paso 1 del escenario principal.</i></p>	
<b>Escenario Alternativo 2</b>	

6. El sistema registra un error en los datos.

7. El sistema lanza un mensaje para avisar al usuario y le permite corregirlos.

*Retorno al paso 4 del escenario principal.*

Tabla 27: CU-USR-06: Edición de Datos del Perfil

<b>ID</b>	CU-USR-07
<b>Título</b>	Consulta del Listado de Medicinas
<b>Descripción</b>	El usuario quiere consultar las distintas medicinas introducidas en su perfil.
<b>Pre-Condición</b>	Existencia del usuario en el sistema. Usuario del que se quieren consultar las medicinas seleccionado en la aplicación.
<b>Post-Condición</b>	Listado de medicinas del usuario mostrados en pantalla.
<b>Prioridad</b>	Alta
<b>Reqs. Tratados</b>	RF-USR-3.2
<b>Escenario Principal</b>	
<ol style="list-style-type: none"><li>1. El usuario presiona sobre el botón 'Perfil'.</li><li>2. El sistema muestra los datos del usuario.</li><li>3. El usuario selecciona la sección de 'Medicinas'.</li><li>4. El sistema abre la ventana de 'Medicinas' mostrando el listado del usuario.</li></ol>	

Tabla 28: CU-USR-07: Consulta del Listado de Medicinas

<b>ID</b>	CU-USR-08
<b>Título</b>	Adición de Medicinas
<b>Descripción</b>	El usuario quiere añadir medicinas a su perfil de la aplicación o a una sugerencia.
<b>Pre-Condición</b>	Existencia del usuario en el sistema. Usuario del que se quieren gestionar las medicinas seleccionado en la aplicación.
<b>Post-Condición</b>	Listado de medicinas actualizado correctamente. Usuario visualizando el listado de medicinas propias (o de una sugerencia).
<b>Prioridad</b>	Alta
<b>Reqs. Tratados</b>	RF-USR-3.1, RF-USR-4.6
<b>Escenario Principal</b>	

1. El usuario realiza el caso de uso *CU-USR-07* accediendo así a su listado de medicinas.
2. El usuario presiona el botón *'Editar'* marcado con un icono.
3. El sistema abre un formulario con el listado y un campo para añadir.
4. El usuario añade el nombre de la medicina.
5. El usuario presiona el botón *'Añadir'*.
6. El sistema valida los campos del formulario.
7. El sistema añade la medicina al listado.
8. El usuario presiona el botón *'Finalizar'*.
9. El sistema guarda los cambios.
10. El sistema retorna al listado de medicinas.

#### Escenario Alternativo 1

8. El usuario presiona el botón *'Cancelar'*.

*Retorno al paso 1 del escenario principal.*

#### Escenario Alternativo 2

6. El sistema registra un error en los datos.
7. El sistema lanza un mensaje para avisar al usuario y le permite corregirlos.

*Retorno al paso 4 del escenario principal.*

Tabla 29: CU-USR-08: Adición de Medicinas

<b>ID</b>	CU-USR-09
<b>Título</b>	Edición de Medicinas
<b>Descripción</b>	El usuario quiere modificar medicinas de su perfil (o de una sugerencia).
<b>Pre-Condición</b>	Existencia del usuario en el sistema. Usuario del que se quieren gestionar las medicinas seleccionado en la aplicación.
<b>Post-Condición</b>	Listado de medicinas actualizado correctamente. Usuario visualizando el listado de medicinas del perfil (o de una sugerencia).
<b>Prioridad</b>	Alta
<b>Reqs. Tratados</b>	RF-USR-3.3, RF-USR-4.6
<b>Escenario Principal</b>	

<ol style="list-style-type: none"> <li>1. El usuario realiza el caso de uso <i>CU-USR-07</i> accediendo así a su listado de medicinas.</li> <li>2. El usuario presiona el botón <i>'Editar'</i> marcado con un icono.</li> <li>3. El sistema abre un formulario con el listado y un campo para añadir.</li> <li>4. El usuario presiona el icono de <i>'Editar'</i> de una medicina del listado.</li> <li>5. El sistema rellena el campo de texto con el nombre actual de la medicina.</li> <li>6. El usuario modifica el nombre de la medicina.</li> <li>7. El usuario presiona el botón <i>'Confirmar'</i>.</li> <li>8. El sistema valida los campos del formulario.</li> <li>9. El sistema modifica la medicina del listado y limpia el campo de texto.</li> <li>10. El usuario presiona el botón <i>'Finalizar'</i>.</li> <li>11. El sistema guarda los cambios.</li> <li>12. El sistema retorna al listado de medicinas.</li> </ol>
<b>Escenario Alternativo 1</b>
<ol style="list-style-type: none"> <li>7. El usuario presiona el botón <i>'Cancelar'</i>.</li> </ol> <p><i>Retorno al paso 3 del escenario principal.</i></p>
<b>Escenario Alternativo 2</b>
<ol style="list-style-type: none"> <li>10. El usuario presiona el botón <i>'Cancelar'</i>.</li> </ol> <p><i>Retorno al paso 1 del escenario principal.</i></p>
<b>Escenario Alternativo 3</b>
<ol style="list-style-type: none"> <li>8. El sistema registra un error en los datos.</li> <li>9. El sistema lanza un mensaje para avisar al usuario y le permite corregirlos.</li> </ol> <p><i>Retorno al paso 5 del escenario principal.</i></p>

Tabla 30: CU-USR-09: Edición de Medicinas

<b>ID</b>	CU-USR-10
<b>Título</b>	Borrado de Medicinas
<b>Descripción</b>	El usuario quiere eliminar medicinas de su perfil (o de una sugerencia).
<b>Pre-Condición</b>	Existencia del usuario en el sistema. Usuario del que se quieren gestionar los datos seleccionado en la aplicación.
<b>Post-Condición</b>	Listado de medicinas actualizado correctamente. Usuario visualizando el listado de medicinas propias (o de una sugerencia).

<b>Prioridad</b>	Alta
<b>Reqs. Tratados</b>	RF-USR-3.4, RF-USR-4.6
<b>Escenario Principal</b>	
<ol style="list-style-type: none"> <li>1. El usuario realiza el caso de uso <b>CU-USR-07</b> accediendo así a su listado de medicinas.</li> <li>2. El usuario presiona el botón <i>'Editar'</i> marcado con un icono.</li> <li>3. El sistema abre un formulario con el listado y un campo para añadir.</li> <li>4. El usuario presiona el icono de <i>'Eliminar'</i> de una medicina del listado.</li> <li>5. El sistema muestra una ventana de confirmación al usuario.</li> <li>6. El usuario presiona el botón <i>'Confirmar'</i>.</li> <li>7. El sistema elimina la medicina del listado.</li> <li>8. El usuario presiona el botón <i>'Finalizar'</i>.</li> <li>9. El sistema guarda los cambios.</li> <li>10. El sistema retorna al listado de medicinas.</li> </ol>	
<b>Escenario Alternativo 1</b>	
<ol style="list-style-type: none"> <li>6. El usuario presiona el botón <i>'Cancelar'</i>.</li> </ol> <p><i>Retorno al paso 3 del escenario principal.</i></p>	
<b>Escenario Alternativo 2</b>	
<ol style="list-style-type: none"> <li>8. El usuario presiona el botón <i>'Cancelar'</i>.</li> </ol> <p><i>Retorno al paso 1 del escenario principal.</i></p>	

Tabla 31: CU-USR-10: Borrado de Medicinas

<b>ID</b>	CU-USR-11
<b>Título</b>	Consulta del Listado de Recomendaciones
<b>Descripción</b>	El usuario quiere consultar las recomendaciones introducidas en su perfil.
<b>Pre-Condición</b>	Existencia del usuario en el sistema. Usuario del que se quieren consultar los datos seleccionado en la aplicación.
<b>Post-Condición</b>	Listado de recomendaciones del usuario mostrados en pantalla.
<b>Prioridad</b>	Alta
<b>Reqs. Tratados</b>	RF-USR-4.2
<b>Escenario Principal</b>	

1. El usuario presiona sobre el botón 'Perfil'.
2. El sistema muestra los datos del usuario.
3. El usuario selecciona la sección de 'Sugerencias'.
4. El sistema abre la ventana de 'Sugerencias' mostrando el listado del usuario.

Tabla 32: CU-USR-11: Consulta del Listado de Recomendaciones

<b>ID</b>	CU-USR-12
<b>Título</b>	Adición de Sugerencia
<b>Descripción</b>	El usuario quiere añadir una sugerencia.
<b>Pre-Condición</b>	Existencia del usuario en el sistema. Usuario al que se quiere añadir la sugerencia seleccionado en la aplicación.
<b>Post-Condición</b>	Listado de sugerencias actualizado correctamente. Usuario visualizando el listado de sugerencias.
<b>Prioridad</b>	Alta
<b>Reqs. Tratados</b>	RF-USR-4.1
<b>Escenario Principal</b>	
<ol style="list-style-type: none"> <li>1. El usuario realiza el caso de uso <b>CU-USR-11</b> accediendo así a su listado de sugerencias.</li> <li>2. El usuario presiona el botón 'Añadir' marcado con un icono.</li> <li>3. El sistema abre un formulario con los datos a seleccionar.</li> <li>4. El usuario añade los niveles de glucosa.</li> <li>5. El usuario presiona el botón 'Añadir Medicinas'.</li> <li>6. El sistema abre una ventana con las medicinas de la sugerencia.</li> <li>7. El usuario realiza un caso de uso similar al <b>CU-USR-8</b>, <b>CU-USR-9</b> y <b>CU-USR-10</b> para la gestión de medicinas en la sugerencia.</li> <li>8. El usuario presiona el botón 'Finalizar'.</li> <li>9. El sistema valida la información introducida.</li> <li>10. El sistema retorna al listado de sugerencias.</li> </ol>	
<b>Escenario Alternativo 1</b>	
<ol style="list-style-type: none"> <li>8. El usuario presiona el botón 'Cancelar'.</li> </ol> <p><i>Retorno al paso 1 del escenario principal.</i></p>	
<b>Escenario Alternativo 2</b>	

5. El usuario presiona el botón *Finalizar* con ninguna medicina añadida.
6. El sistema lanza un mensaje para avisar al usuario y le impide continuar.

*Retorno al paso 4 del escenario principal.*

Tabla 33: CU-USR-12: Adición de Sugerencia

<b>ID</b>	CU-USR-13
<b>Título</b>	Consulta de una Sugerencia Concreta
<b>Descripción</b>	El usuario quiere consultar los detalles de una sugerencia.
<b>Pre-Condición</b>	Existencia del usuario en el sistema. Usuario del que se quiere consultar la sugerencia seleccionado en la aplicación.
<b>Post-Condición</b>	Detalles de la sugerencia mostrados en pantalla.
<b>Prioridad</b>	Alta
<b>Reqs. Tratados</b>	RF-USR-4.5
<b>Escenario Principal</b>	
<ol style="list-style-type: none"> <li>1. El usuario realiza el caso de uso <b>CU-USR-11</b> accediendo así a su listado de sugerencias.</li> <li>2. El usuario selecciona una sugerencia del listado.</li> <li>3. El sistema abre en detalle la sugerencia mostrando su información.</li> </ol>	

Tabla 34: CU-USR-13: Consulta de una Sugerencia Concreta

<b>ID</b>	CU-USR-14
<b>Título</b>	Edición de Sugerencia
<b>Descripción</b>	El usuario quiere modificar una sugerencia.
<b>Pre-Condición</b>	Existencia del usuario en el sistema. Usuario del que se modificar la sugerencia seleccionado en la aplicación.
<b>Post-Condición</b>	Sugerencia actualizada correctamente. Usuario visualizando los detalles de la sugerencia.
<b>Prioridad</b>	Alta
<b>Reqs. Tratados</b>	RF-USR-4.3
<b>Escenario Principal</b>	



1. El usuario realiza el caso de uso **CU-USR-13** accediendo así a una sugerencia.
2. El usuario presiona el botón *'Editar'* marcado con un icono.
3. El sistema abre un formulario con los datos a seleccionar.
4. El usuario modifica los niveles de glucosa.
5. El usuario presiona el botón *'Añadir Medicinas'*.
6. El sistema abre una ventana con las medicinas de la sugerencia.
7. El usuario realiza un caso de uso similar al **CU-USR-8**, **CU-USR-9** y **CU-USR-10** para la gestión de medicinas en la sugerencia.
8. El usuario presiona el botón *'Finalizar'*.
9. El sistema valida la información introducida.
10. El sistema retorna al listado de sugerencias.

#### Escenario Alternativo 1

8. El usuario presiona el botón *'Cancelar'*.

*Retorno al paso 1 del escenario principal.*

#### Escenario Alternativo 2

5. El usuario presiona el botón *'Finalizar'* con ninguna medicina añadida.
6. El sistema lanza un mensaje para avisar al usuario y le impide continuar.

*Retorno al paso 4 del escenario principal.*

Tabla 35: CU-USR-14: Edición de Sugerencia

<b>ID</b>	CU-USR-15
<b>Título</b>	Borrado de Sugerencia
<b>Descripción</b>	El usuario quiere eliminar una sugerencia.
<b>Pre-Condición</b>	Existencia del usuario en el sistema. Usuario del que se quieren modificar la sugerencia seleccionado en la aplicación.
<b>Post-Condición</b>	Sugerencia eliminada correctamente. Usuario visualizando el listado actualizado de sugerencias.
<b>Prioridad</b>	Alta
<b>Reqs. Tratados</b>	RF-USR-4.4
<b>Escenario Principal</b>	

1. El usuario realiza el caso de uso *CU-USR-13* accediendo así a una sugerencia.
2. El usuario presiona el botón *'Eliminar'* marcado con un icono.
3. El sistema abre una ventana de confirmación.
4. El usuario presiona el botón *'Confirmar'*.
5. El sistema elimina la sugerencia.
6. El sistema retorna al listado de sugerencias.

#### Escenario Alternativo 1

4. El usuario presiona el botón *'Cancelar'*.

*Retorno al paso 1 del escenario principal.*

Tabla 36: CU-USR-15: Borrado de Sugerencia

#### C.6.2.2. Mediciones (MSR)



Figura 107: CU-MSR: Diagrama de Casos de Uso

<b>ID</b>	CU-MSR-01
<b>Título</b>	Acceso al Listado de Mediciones
<b>Descripción</b>	Visualización de las mediciones añadidas al sistema pertenecientes al usuario seleccionado.
<b>Pre-Condición</b>	Existencia del usuario en el sistema. Usuario del que se quieren consultar las mediciones seleccionado.
<b>Post-Condición</b>	Listado de mediciones del usuario mostrado en pantalla.
<b>Prioridad</b>	Alta
<b>Reqs. Tratados</b>	RF-MSR-5.2

### Escenario Principal

1. El usuario presiona el botón 'Registro' de la barra de navegación principal.
2. El sistema muestra el listado con todas las mediciones guardadas.

Tabla 37: CU-MSR-01: Acceso al Listado de Mediciones

<b>ID</b>	CU-MSR-02
<b>Título</b>	Adición de una Medición
<b>Descripción</b>	Añadir a los datos del usuario una nueva medición de glucosa.
<b>Pre-Condición</b>	Usuario al que se quiere añadir seleccionado en la aplicación.
<b>Post-Condición</b>	Nueva medición añadida en los datos del usuario.
<b>Prioridad</b>	Alta
<b>Reqs. Tratados</b>	RF-MSR-5.1
<b>Escenario Principal</b>	
<ol style="list-style-type: none"><li>1. El usuario realiza el caso de uso <b>CU-MSR-01</b> accediendo así al listado de mediciones.</li><li>2. El usuario presiona el botón 'Añadir'.</li><li>3. El sistema muestra una pantalla con los datos obligatorios a añadir: hora, fecha, momento de la medición y situación en la que se toma (ayuno, antes o después de comer).</li><li>4. El usuario selecciona las opciones adecuadas y presiona el botón 'Siguiente'.</li><li>5. El sistema valida los campos y pasa a la siguiente página del formulario.</li><li>6. El usuario selecciona los síntomas que ha padecido previo a la medición y presiona el botón 'Siguiente'.</li><li>7. El sistema valida los campos.</li><li>8. El sistema muestra un formulario con los datos adicionales que el usuario puede introducir: nota y actividad física realizada.</li><li>9. El usuario completa los campos adecuados para su medición.</li><li>10. El usuario presiona 'Finalizar'.</li><li>11. El sistema valida los campos y muestra un mensaje de confirmación al usuario.</li><li>12. El sistema retorna a la página principal de la aplicación.</li></ol>	
<b>Escenario Alternativo 1</b>	
<ol style="list-style-type: none"><li>4. El usuario presiona el botón 'Cancelar'.</li></ol> <p><i>Retorno al paso 1 del escenario principal volviendo a la ventana principal.</i></p>	

**Escenario Alternativo 2**

5. El sistema detecta un error en los datos y lanza un mensaje de advertencia permitiendo corregirlos.

*Retorno al paso 4 del escenario principal.*

Tabla 38: CU-MSR-02: Adición de una Medición

<b>ID</b>	CU-MSR-03
<b>Título</b>	Consulta Detallada de una Medición
<b>Descripción</b>	Visualización en detalle de la medición seleccionada perteneciente al usuario actual.
<b>Pre-Condición</b>	Usuario del que se quieren consultar la medición seleccionado.
<b>Post-Condición</b>	Detalles de una medición mostrados en pantalla.
<b>Prioridad</b>	Alta
<b>Reqs. Tratados</b>	RF-MSR-5.3
<b>Escenario Principal</b>	
<ol style="list-style-type: none"> <li>1. El usuario realiza el caso de uso <b>CU-MSR-01</b> accediendo así al listado de mediciones.</li> <li>2. El usuario presiona una medición del listado.</li> <li>3. El sistema muestra los detalles de la medición presionada.</li> </ol>	

Tabla 39: CU-MSR-03: Consulta Detallada de una Medición

<b>ID</b>	CU-MSR-04
<b>Título</b>	Edición de una Medición
<b>Descripción</b>	Editar la medición de glucosa de un usuario.
<b>Pre-Condición</b>	Usuario que quiere modificar su medición seleccionado en la aplicación.
<b>Post-Condición</b>	Nueva medición añadida en los datos del usuario.
<b>Prioridad</b>	Alta
<b>Reqs. Tratados</b>	RF-MSR-5.4, RF-MSR-5.5
<b>Escenario Principal</b>	

1. El usuario realiza el caso de uso **CU-MSR-03** accediendo así a una medición.
2. El usuario presiona el botón *'Editar'* marcado con un icono.
3. El sistema muestra un formulario con los datos obligatorios a añadir: hora, fecha, momento de la medición y situación en la que se toma (ayuno, antes o después de comer).
4. El usuario modifica las opciones deseadas y presiona el botón *'Siguiente'*.
5. El sistema valida los campos y pasa a la siguiente página del formulario.
6. El usuario modifica los síntomas que ha padecido previo a la medición y presiona el botón *'Siguiente'*.
7. El sistema valida los campos.
8. El sistema muestra un formulario con los datos adicionales que el usuario puede introducir: nota y actividad física realizada.
9. El usuario completa los campos deseados para su medición.
10. El usuario presiona *'Finalizar'*.
11. El sistema valida los campos y muestra un mensaje de confirmación al usuario.
12. El sistema retorna a la página principal de la aplicación.

#### Escenario Alternativo 1

4. El usuario presiona el botón *'Cancelar'*.

*Retorno al paso 1 del escenario principal volviendo a la ventana principal.*

#### Escenario Alternativo 2

5. El sistema detecta un error en los datos y lanza un mensaje de advertencia permitiendo corregirlos.

*Retorno al paso 4 del escenario principal.*

Tabla 40: CU-MSR-04: Edición de una Medición

<b>ID</b>	CU-MSR-05
<b>Título</b>	Borrado de una Medición
<b>Descripción</b>	Eliminar la medición de glucosa de un usuario.
<b>Pre-Condición</b>	Usuario que quiere eliminar su medición seleccionado en la aplicación.
<b>Post-Condición</b>	Listado de mediciones actualizado en los datos del usuario.
<b>Prioridad</b>	Alta
<b>Reqs. Tratados</b>	RF-MSR-5.6
<b>Escenario Principal</b>	

<ol style="list-style-type: none"> <li>1. El usuario realiza el caso de uso <b>CU-MSR-03</b> accediendo así a una medición.</li> <li>2. El usuario presiona el botón <i>'Eliminar'</i> marcado con un icono.</li> <li>3. El sistema muestra un mensaje de confirmación al usuario.</li> <li>4. El usuario presiona el botón <i>'Confirmar'</i>.</li> <li>5. El sistema elimina la medición y retorna al listado de mediciones.</li> </ol>
<b>Escenario Alternativo 1</b>
<ol style="list-style-type: none"> <li>4. El usuario presiona el botón <i>'Cancelar'</i>.</li> </ol> <p><i>Retorno al paso 1 del escenario principal volviendo a la ventana principal.</i></p>

Tabla 41: CU-MSR-05: Borrado de una Medición

<b>ID</b>	CU-MSR-06
<b>Título</b>	Consulta de Sugerencias para una Medición
<b>Descripción</b>	Visualizar el listado de medicinas recomendado para una medición de glucosa de un usuario.
<b>Pre-Condición</b>	Usuario que quiere ver la recomendación seleccionado en la aplicación.
<b>Post-Condición</b>	Listado de sugerencias filtrado por nivel de glucosa mostrado por pantalla.
<b>Prioridad</b>	Alta
<b>Reqs. Tratados</b>	RF-MSR-6
<b>Escenario Principal</b>	
<ol style="list-style-type: none"> <li>1. El usuario realiza el caso de uso <b>CU-MSR-03</b> accediendo así a una medición.</li> <li>2. El usuario presiona el botón <i>'Sugerencias Recomendadas'</i>.</li> <li>3. El sistema muestra el listado de sugerencias referente al nivel de glucosa de la medición seleccionada.</li> </ol>	

Tabla 42: CU-MSR-06: Consulta de Sugerencias para una Medición

<b>ID</b>	CU-MSR-07
<b>Título</b>	Consulta de Medicinas Tomadas en una Medición
<b>Descripción</b>	Visualizar el listado de medicinas tomadas en una medición de glucosa de un usuario.
<b>Pre-Condición</b>	Usuario que quiere consultar las medicinas tomadas de una medición seleccionado.
<b>Post-Condición</b>	Listado de medicinas mostrado en pantalla.

<b>Prioridad</b>	Alta
<b>Reqs. Tratados</b>	RF-MSR-7.2
<b>Escenario Principal</b>	
<ol style="list-style-type: none"> <li>1. El usuario realiza el caso de uso <b>CU-MSR-03</b> accediendo así a una medición.</li> <li>2. El usuario presiona el botón '<i>Medicinas Tomadas</i>'.</li> <li>3. El sistema muestra el listado de medicinas completo marcando como seleccionadas las que se ha tomado.</li> </ol>	

Tabla 43: CU-MSR-07: Consulta de Medicinas Tomadas en una Medición

<b>ID</b>	CU-MSR-08
<b>Título</b>	Gestión de Medicinas Tomadas en una Medición
<b>Descripción</b>	Modificar el listado de medicinas tomadas en una medición añadiendo, editando o eliminando medicinas.
<b>Pre-Condición</b>	Usuario que quiere modificar las medicinas tomadas de una medición seleccionado en la aplicación.
<b>Post-Condición</b>	Listado de medicinas actualizado mostrado en pantalla.
<b>Prioridad</b>	Alta
<b>Reqs. Tratados</b>	RF-MSR-7.1, RF-MSR-7.3, RF-MSR-7.4
<b>Escenario Principal</b>	
<ol style="list-style-type: none"> <li>1. El usuario realiza el caso de uso <b>CU-MSR-07</b> accediendo así al listado de medicinas tomadas de una medición.</li> <li>2. El usuario realiza un caso de uso similar al <b>CU-USR-8</b>, <b>CU-USR-9</b> y <b>CU-USR-10</b> para la gestión de las medicinas.</li> <li>3. El usuario presiona el botón '<i>Confirmar</i>'.</li> <li>4. El sistema actualiza la información de las medicinas tomadas y retorna a la ventana de detalles.</li> </ol>	
<b>Escenario Alternativo 1</b>	
<ol style="list-style-type: none"> <li>3. El usuario presiona el botón '<i>Cancelar</i>'.</li> </ol> <p><i>Retorno al paso 1 del escenario principal volviendo a la ventana de detalle de una medición.</i></p>	

Tabla 44: CU-MSR-08: Gestión de Medicinas Tomadas en una Medición

### C.6.2.3. Información (INFO)

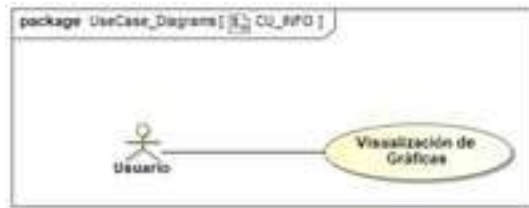


Figura 108: CU-INFO: Diagrama de Casos de Uso

<b>ID</b>	CU-INFO-01
<b>Título</b>	Visualización de Gráficas
<b>Descripción</b>	Visualización del usuario de gráficas construidas con las mediciones añadidas al sistema.
<b>Pre-Condición</b>	Usuario cuyas mediciones se toman de referencia seleccionado en la aplicación.
<b>Post-Condición</b>	Usuario visualizando las gráficas construidas con sus datos.
<b>Prioridad</b>	Media
<b>Reqs. Tratados</b>	RF-INFO-8
<b>Escenario Principal</b>	
<ol style="list-style-type: none"> <li>1. El usuario selecciona en la barra de navegación principal la opción <i>'Estadísticas'</i>.</li> <li>2. El sistema consulta las mediciones del usuario y construye la gráfica.</li> <li>3. El sistema muestra las gráficas al usuario.</li> </ol>	
<b>Escenario Alternativo 1</b>	
<ol style="list-style-type: none"> <li>2. El sistema detecta que no hay mediciones.</li> <li>3. El sistema muestra un mensaje de <i>'No hay suficiente información'</i>.</li> </ol>	

Tabla 45: CU-INFO-01: Visualización de Gráficas



#### C.6.2.4. Notificaciones (NTF)



Figura 109: CU-NTF: Diagrama de Casos de Uso

<b>ID</b>	CU-NTF-01
<b>Título</b>	Consulta de Notificaciones Programadas
<b>Descripción</b>	Visualización de cuáles son las alarmas / notificaciones que un usuario ha almacenado en el sistema como recordatorios.
<b>Pre-Condición</b>	-
<b>Post-Condición</b>	Listado de las notificaciones / alarmas mostrándose en pantalla.
<b>Prioridad</b>	Baja
<b>Reqs. Tratados</b>	RF-NTF-9
<b>Escenario Principal</b>	
<ol style="list-style-type: none"> <li>1. El usuario selecciona la pestaña 'Otros' en la barra de navegación.</li> <li>2. El usuario presiona el botón 'Notificaciones'.</li> <li>3. El sistema muestra un listado de las notificaciones programadas previamente por el usuario.</li> </ol>	

Tabla 46: CU-NTF-01: Consulta de Notificaciones Programadas

<b>ID</b>	CU-NTF-02
<b>Título</b>	Programar una Nueva Notificación
<b>Descripción</b>	Configuración de una nueva notificación para el usuario que la añade.
<b>Pre-Condición</b>	-
<b>Post-Condición</b>	Notificación nueva programada en la aplicación.
<b>Prioridad</b>	Baja
<b>Reqs. Tratados</b>	RF-NTF-9.1
<b>Escenario Principal</b>	

1. El usuario realiza el caso de uso **CU-NTF-01** accediendo así al listado de notificaciones ya existentes.
2. El usuario presiona el botón 'Añadir'.
3. El sistema presenta un formulario con los datos para programar una notificación: descripción y hora.
4. El usuario completa los datos del formulario.
5. El usuario presiona el botón 'Finalizar'.
6. El sistema valida los campos.
7. El sistema añade la nueva notificación al listado previo.
8. El sistema retorna al listado de notificaciones.

#### Escenario Alternativo 1

5. El usuario presiona el botón 'Cancelar'.

*Retorno al paso 1 del escenario principal.*

#### Escenario Alternativo 2

6. El sistema detecta un error en los datos introducidos.
7. El sistema lanza una advertencia y y permite al usuario corregirlos.

*Retorno al paso 3 del escenario principal.*

Tabla 47: CU-NTF-02: Programar una Nueva Notificación

<b>ID</b>	CU-NTF-03
<b>Título</b>	Borrado de una Notificación
<b>Descripción</b>	Eliminación de una notificación que se había programada con anterioridad.
<b>Pre-Condición</b>	Existencia de una notificación previamente programada.
<b>Post-Condición</b>	Notificación eliminada del sistema.
<b>Prioridad</b>	Baja
<b>Reqs. Tratados</b>	RF-NTF-9.2
<b>Escenario Principal</b>	

1. El usuario realiza el caso de uso *CU-NTF-01* accediendo así al listado de notificaciones ya existentes.
2. El usuario presiona el botón 'Eliminar' de una notificación.
3. El sistema presenta un mensaje de confirmación.
4. El usuario presiona el botón 'Aceptar'.
5. El sistema elimina la notificación.
6. El sistema retorna a la ventana anterior.

**Escenario Alternativo 1**

4. El usuario presiona el botón 'Cancelar'.

*Retorno al paso 1 del escenario principal.*

Tabla 48: CU-NTF-03: Borrado de una Notificación

**C.6.2.5. Alimentación / Nutrición (FOOD)**



Figura 110: CU-FOOD: Diagrama de Casos de Uso

<b>ID</b>	CU-FOOD-01
<b>Título</b>	Búsqueda de Información Nutricional
<b>Descripción</b>	El usuario usa la API OpenFoodFacts para obtener información nutricional de ciertas comidas.
<b>Pre-Condición</b>	Conexión a internet del dispositivo establecida.
<b>Post-Condición</b>	Información de alimento buscado mostrándose en pantalla.
<b>Prioridad</b>	Media
<b>Reqs. Tratados</b>	RF-FOOD-10.1, RF-FOOD-10.2
<b>Escenario Principal</b>	

1. El usuario selecciona la pestaña 'Otros' en la barra de navegación.
2. El usuario presiona el botón 'Nutrición'.
3. El sistema muestra al usuario un formulario de búsqueda.
3. El usuario rellena con el nombre del producto.
4. El usuario presiona 'Buscar'.
5. El sistema realiza la búsqueda consultando el recurso.
6. El sistema muestra al usuario los resultados de la búsqueda.
7. El usuario accede al producto que le interese consultar.
8. El sistema muestra con mayor detalle los datos del producto.

#### **Escenario Alternativo 1**

6. La búsqueda en OpenFoodFacts no obtiene resultados.
7. El sistema muestra un mensaje de 'No se han encontrado resultados'.

*Retorno al paso 2 del escenario principal.*

#### **Escenario Alternativo 2**

5. El sistema sufre un error al hacer la búsqueda.
6. El sistema muestra al usuario un mensaje de error.

*Retorno al paso 2 del escenario principal.*

Tabla 49: CU-FOOD-01: Búsqueda de Información Nutricional

## **C.7. Diagramas de Secuencia**

Cada uno de los diagramas de secuencia siguientes se corresponderá con uno o varios casos de uso definidos en la sección anterior de manera que pueda verse de manera más precisa la interacción del usuario con el sistema y el orden de las acciones de las mismas.

### **C.7.1. Datos de Usuario (USR)**

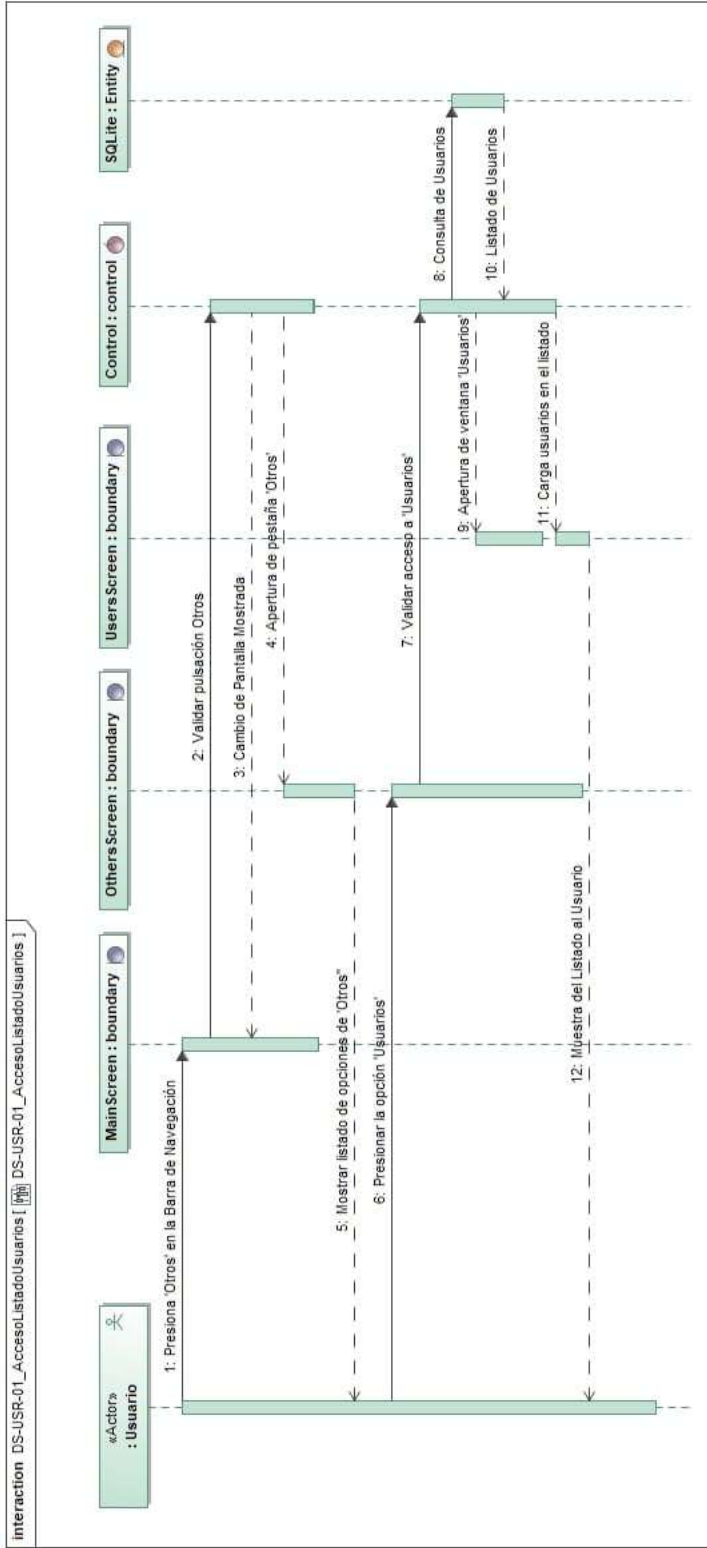


Figura 111: Diag. Secuencia de Acceso al Listado de Usuarios (CU-USR-01)

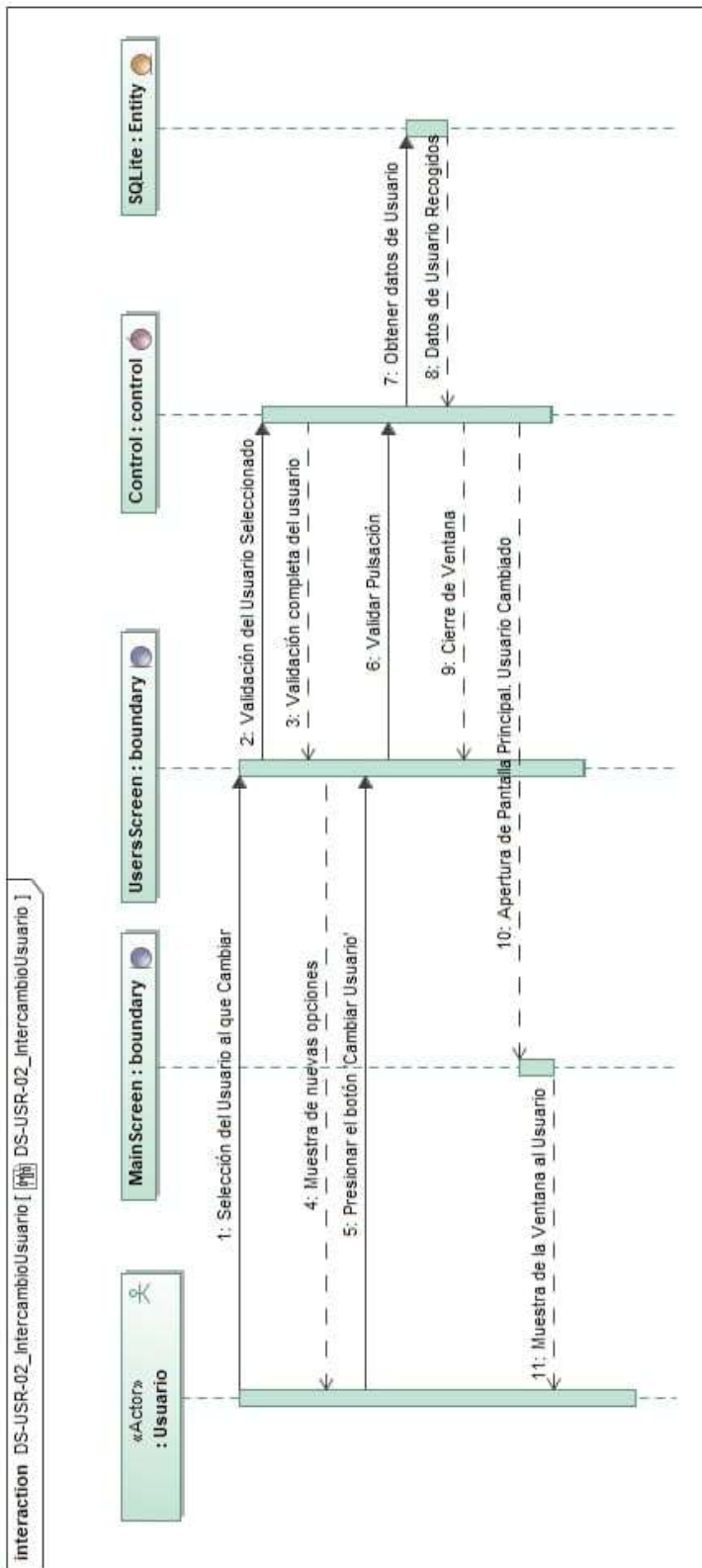


Figura 112: Diag. Secuencia de Intercambio de Usuario (CU-USR-02)

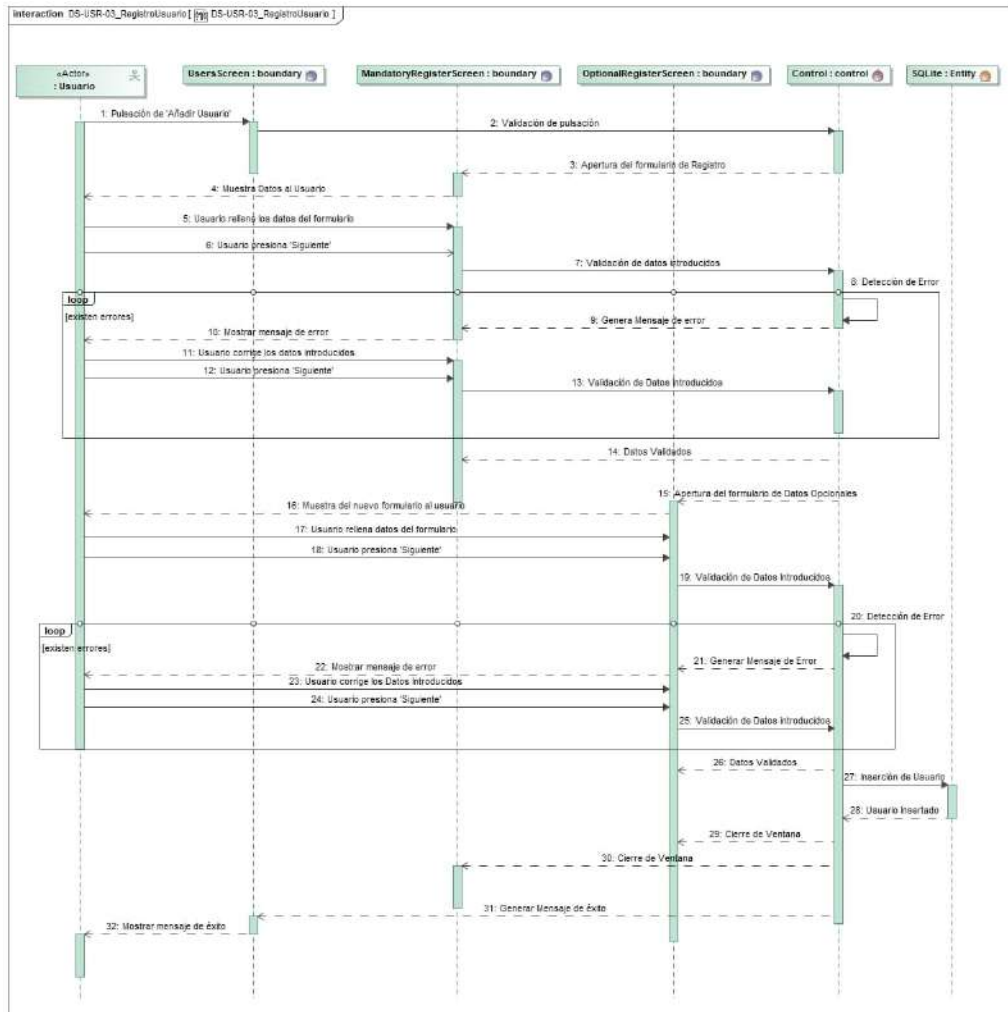


Figura 113: Diag. Secuencia de Registro de Usuarios (CU-USR-03)



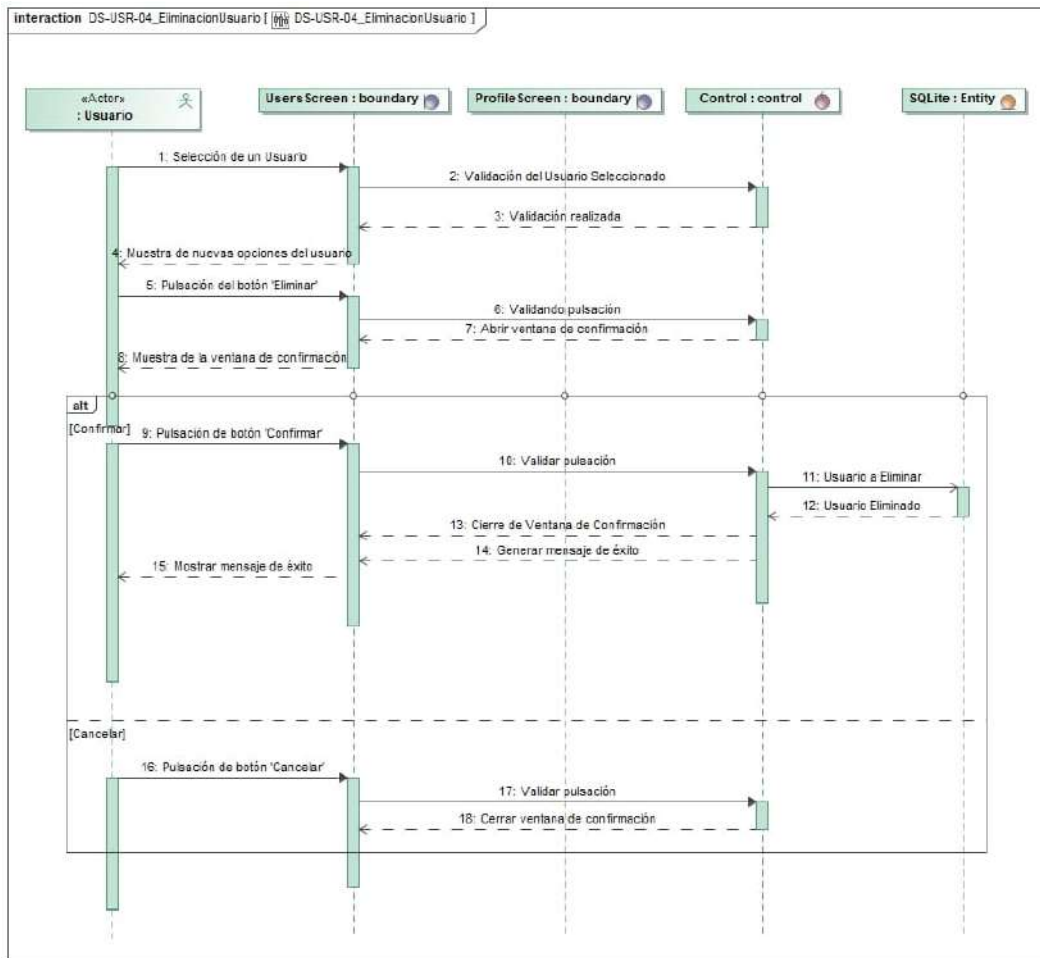


Figura 114: Diag. Secuencia de Eliminación de Usuario (CU-USR-04)

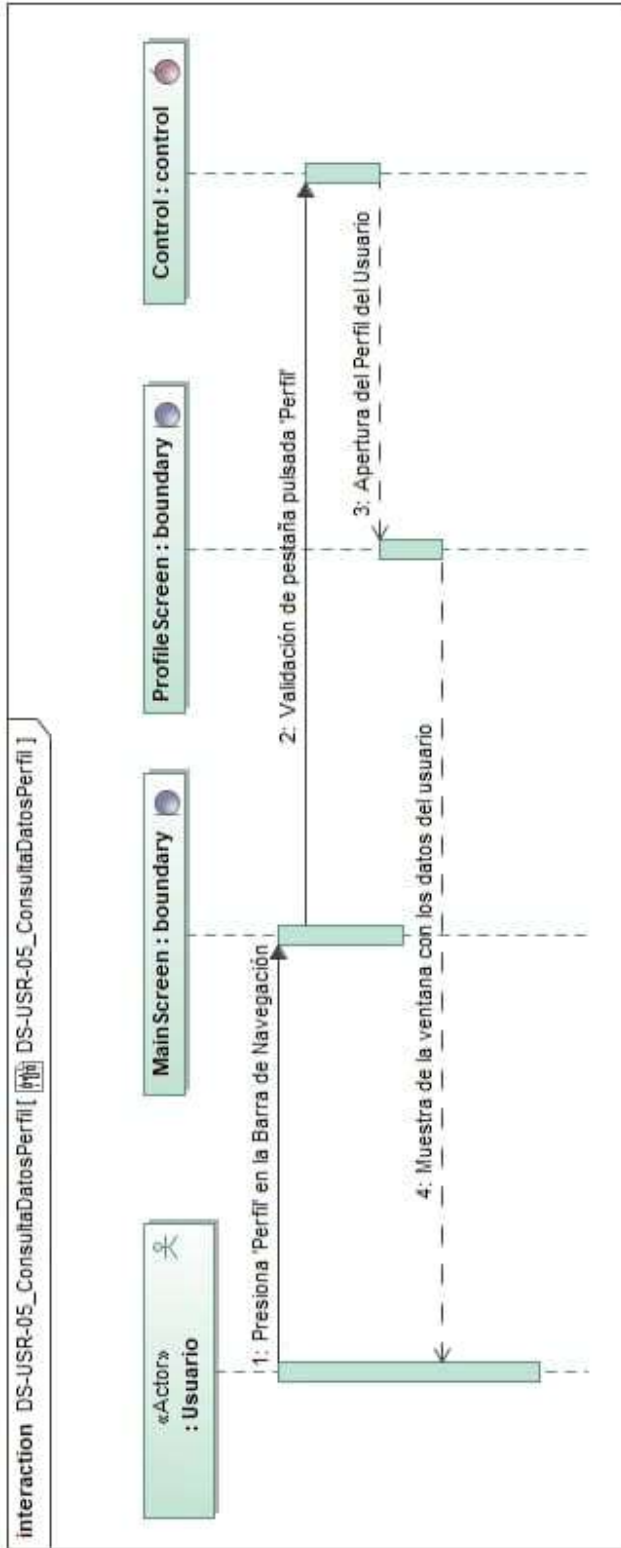


Figura 115: Diag. Secuencia de Consulta de Datos de un Usuario (CU-USR-05)

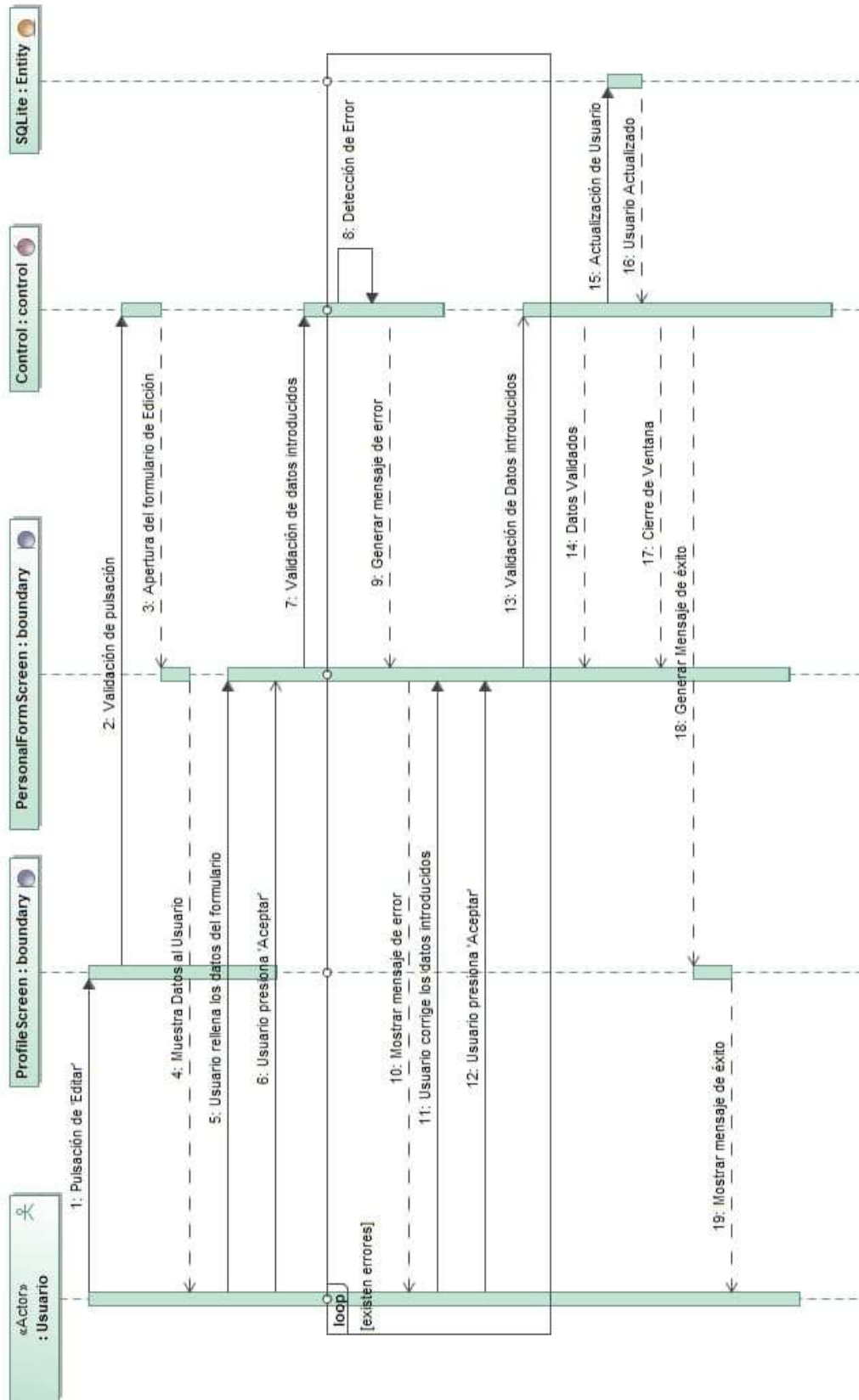


Figura 116: Diag. Secuencia de Edición de Datos del Perfil (CU-USR-06)

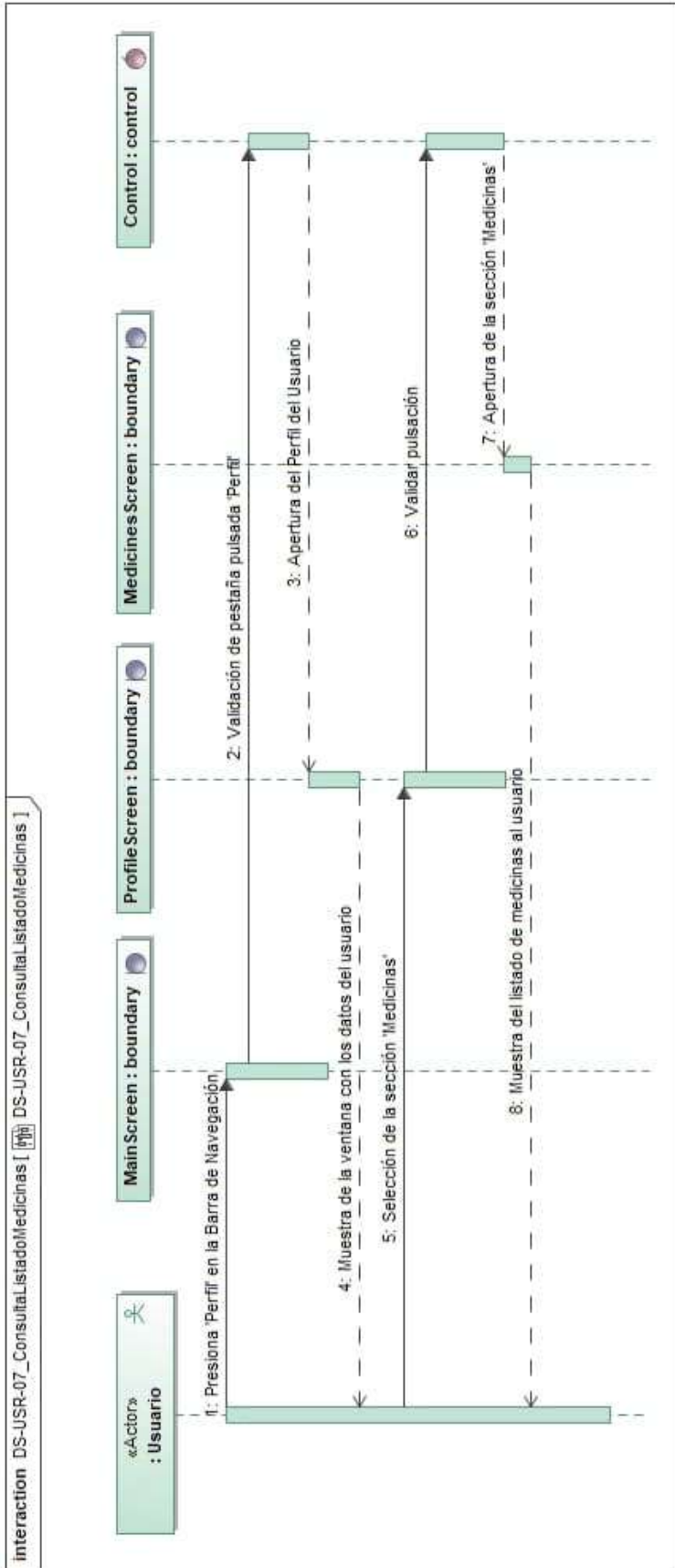


Figura 117: Diag. Secuencia de Consulta del Listado de Medicinas (CU-USR-07)

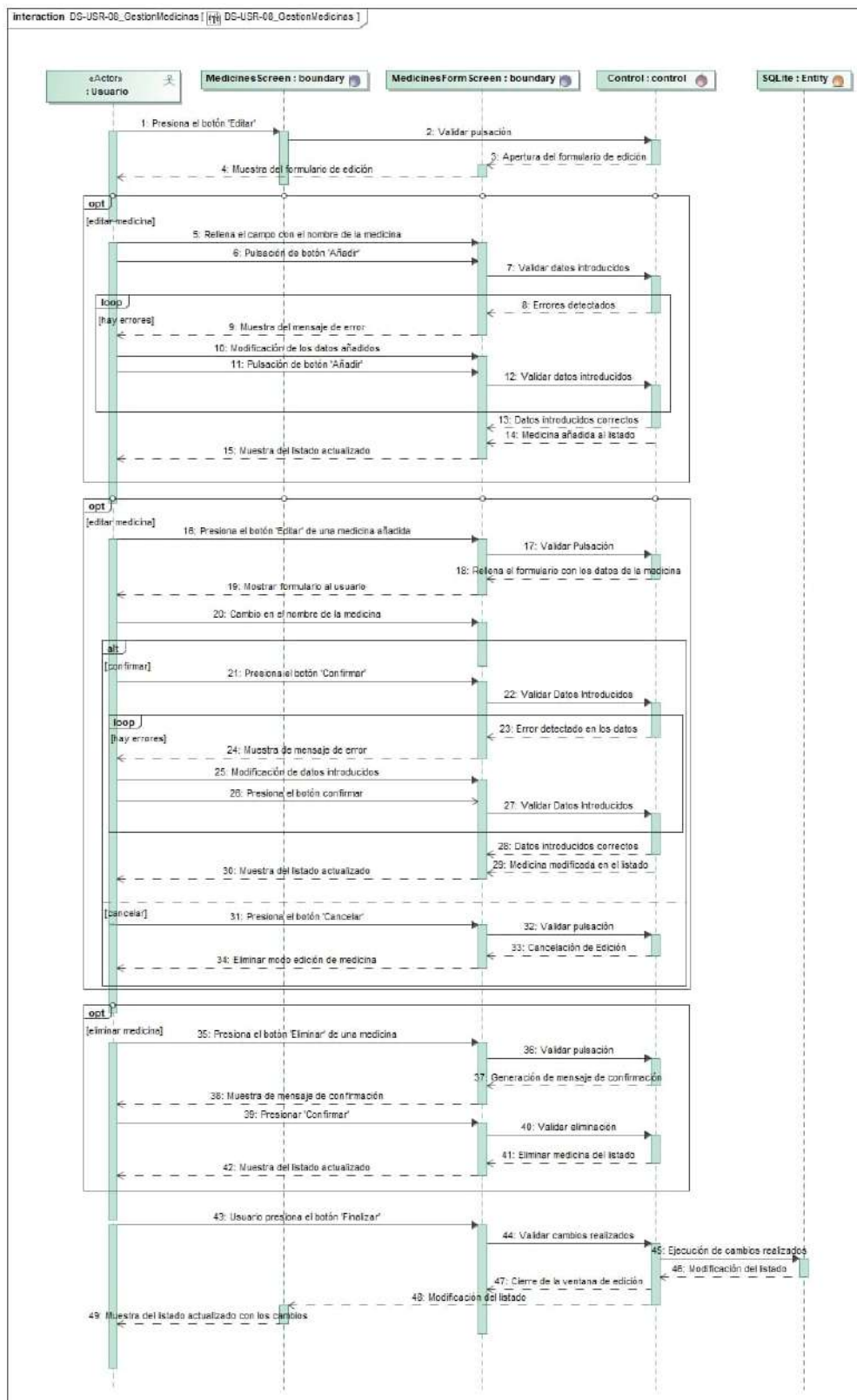


Figura 118: Diag. Secuencia de Gestión de Medicinas (CU-USR-08, CU-USR-09, CU-USR-10)

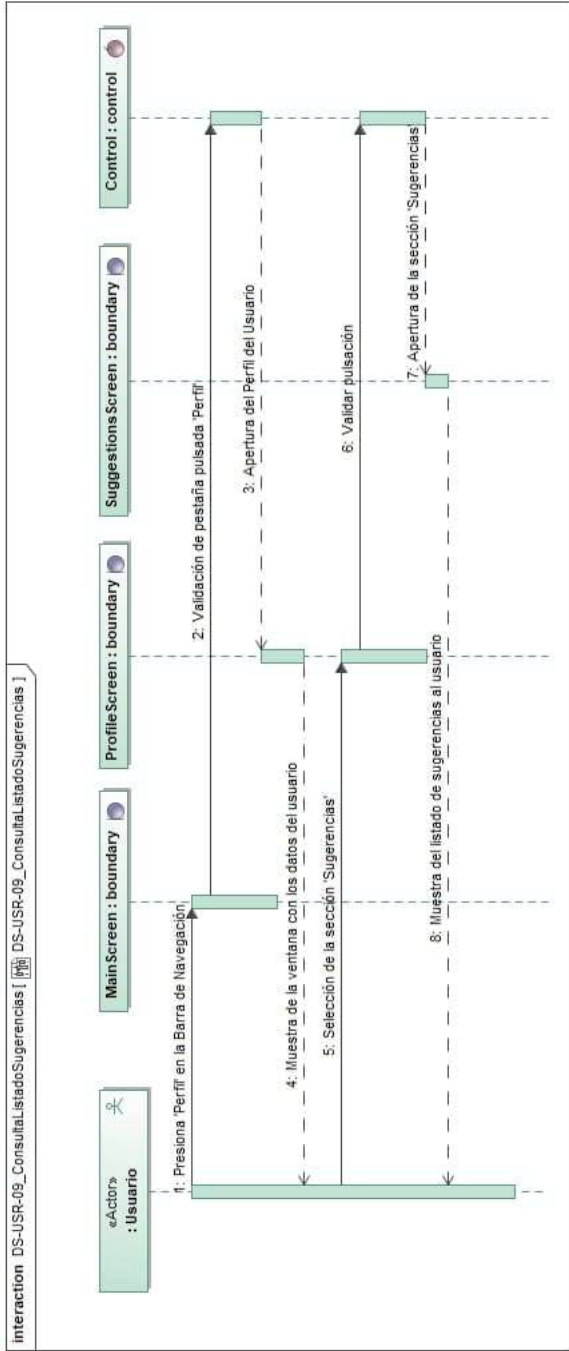


Figura 119: Diag. Secuencia de Consulta del Listado de Recomendaciones (CU-USR-11)

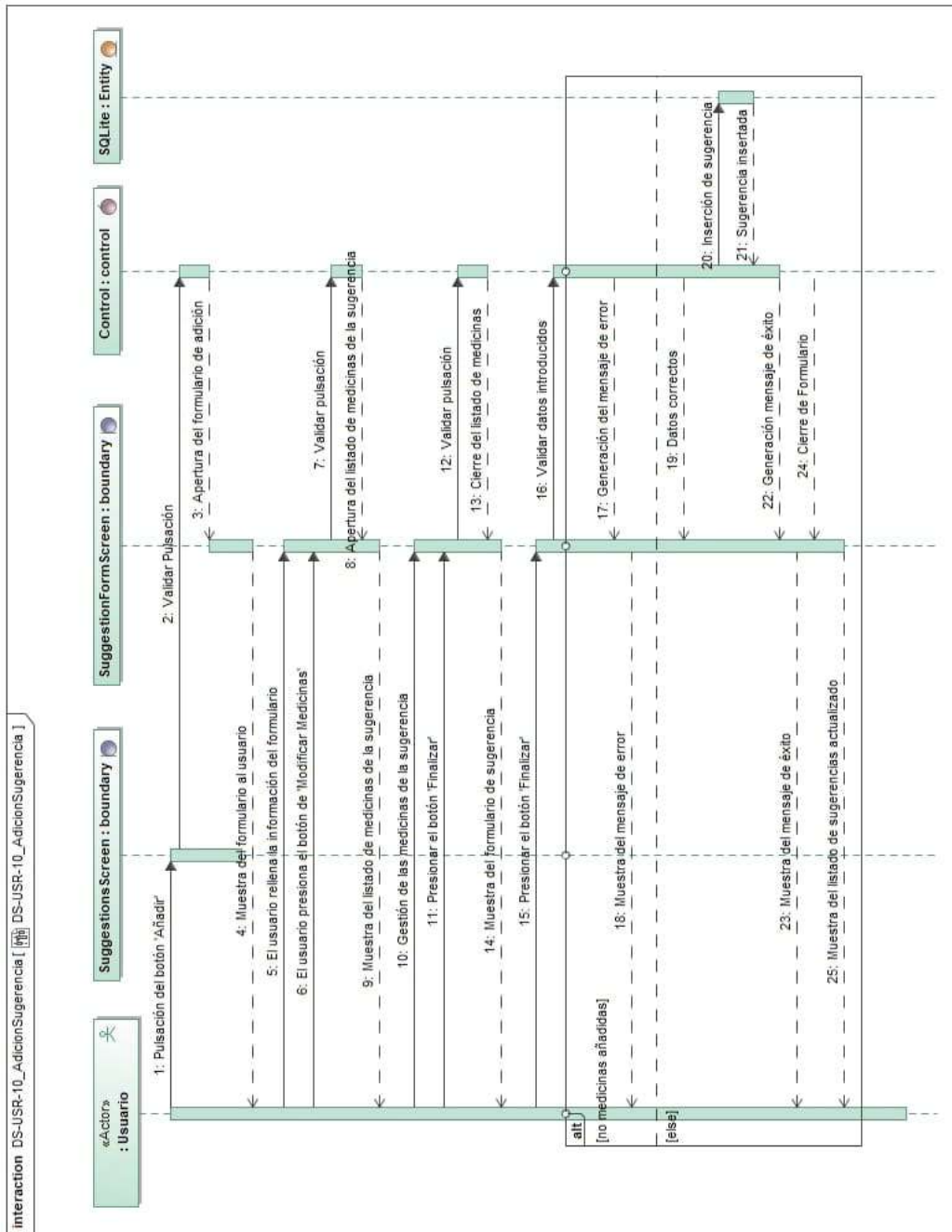


Figura 120: Diag. Secuencia de Adición de Sugerencia (CU-USR-12)

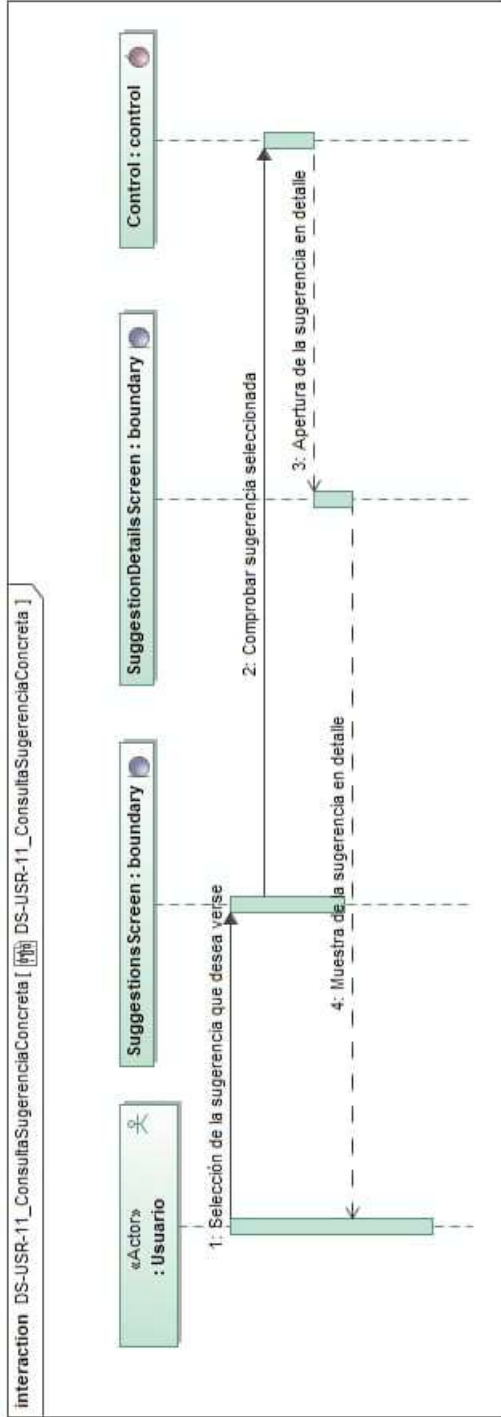


Figura 121: Diag. Secuencia de Consulta de una Sugerencia Concreta (CU-USR-13)



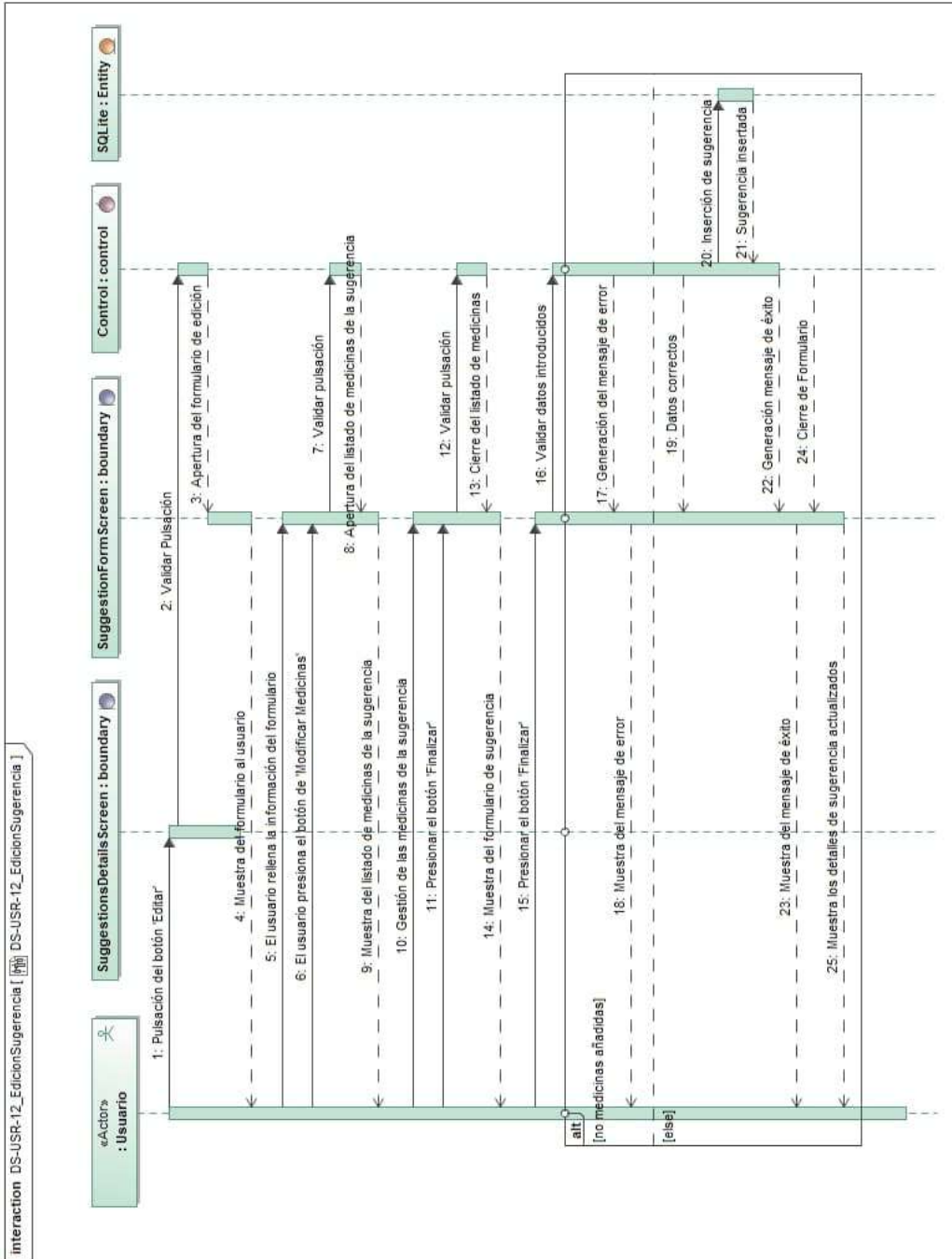


Figura 122: Diag. Secuencia de Edición de Sugerencia (CU-USR-14)

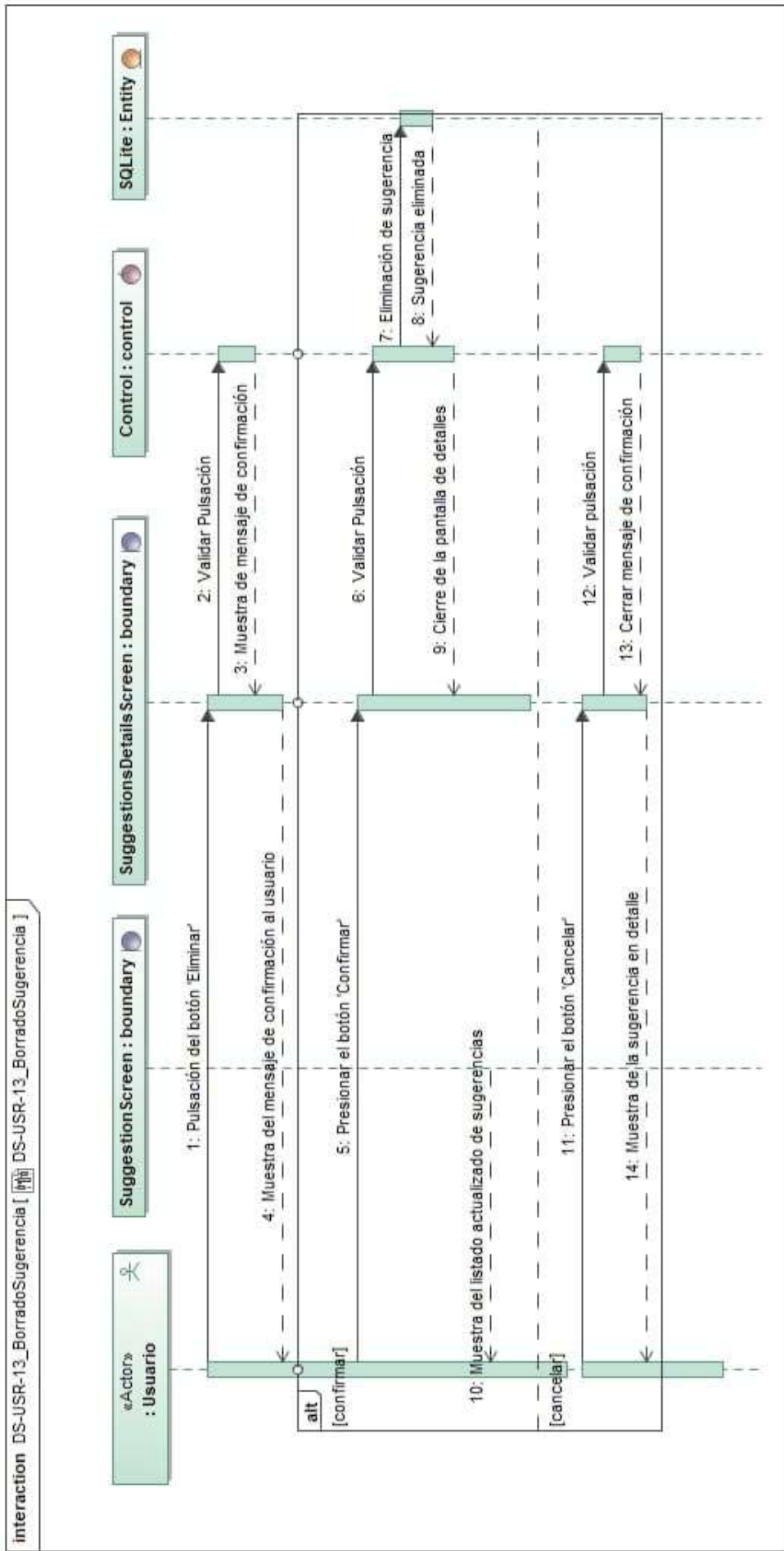


Figura 123: Diag. Secuencia de Borrado de Sugerencia (CU-USR-15)

### C.7.2. Mediciones (MSR)

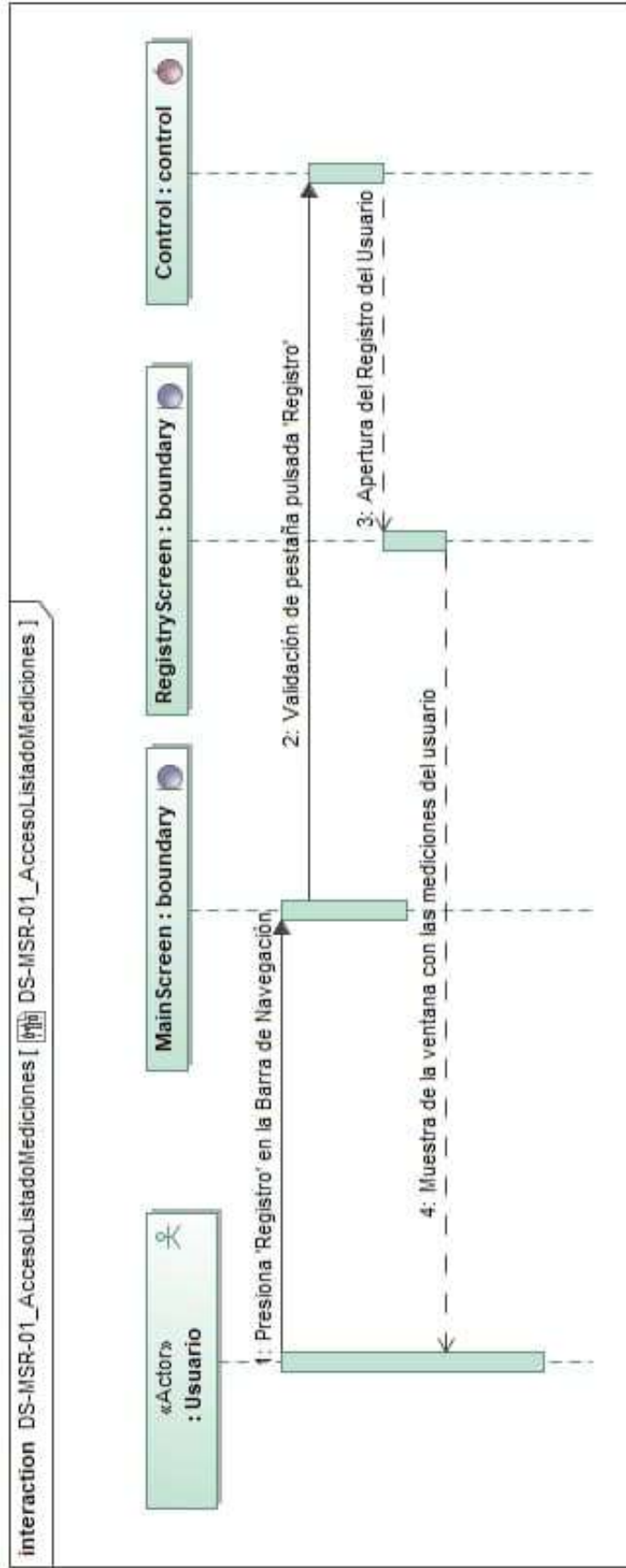


Figura 124: Diag. Secuencia de Acceso al Listado de Mediciones (CU-MSR-01)

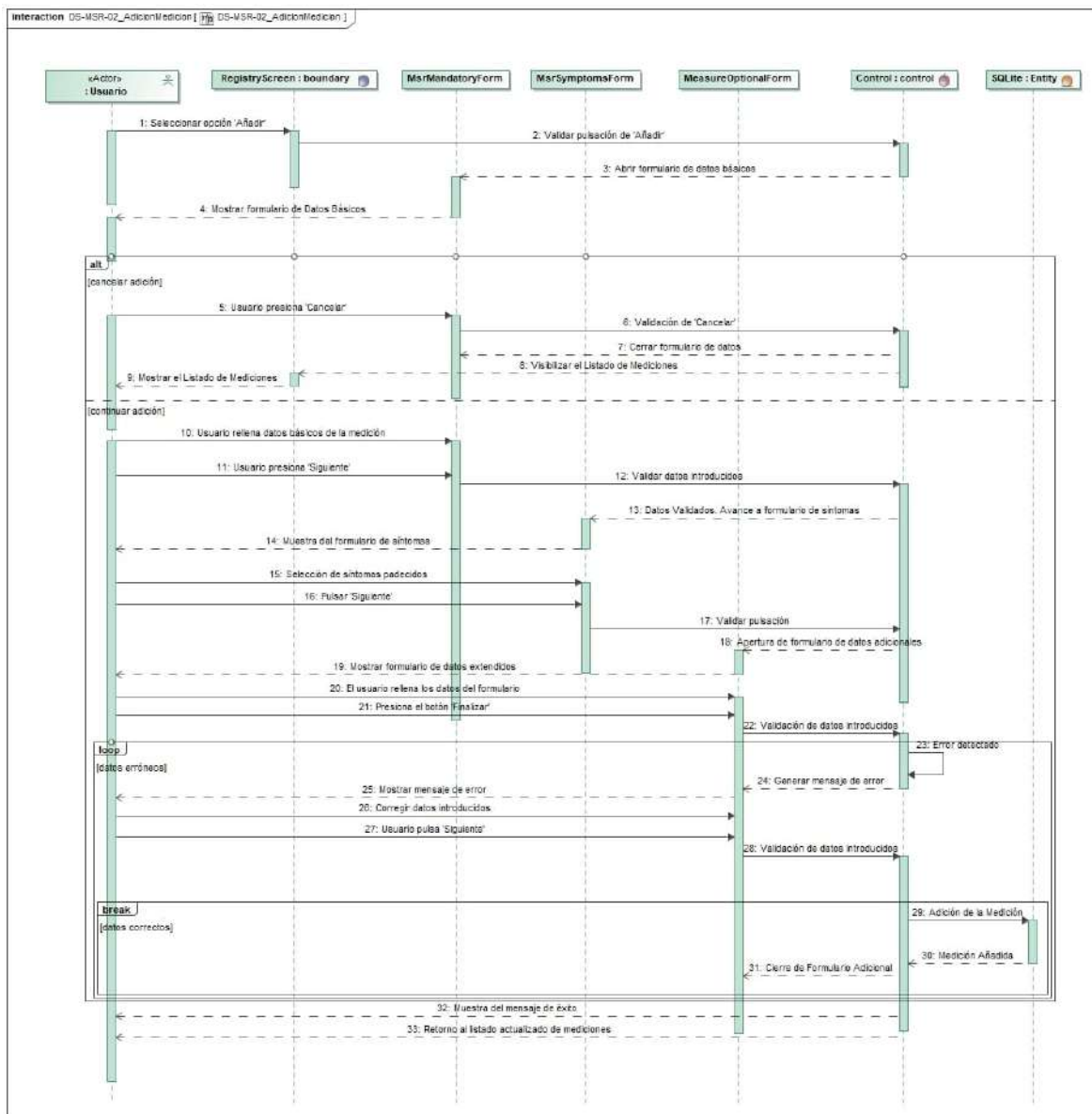


Figura 125: Diag. Secuencia de Adición de una Medicor (CU-MSR-02)

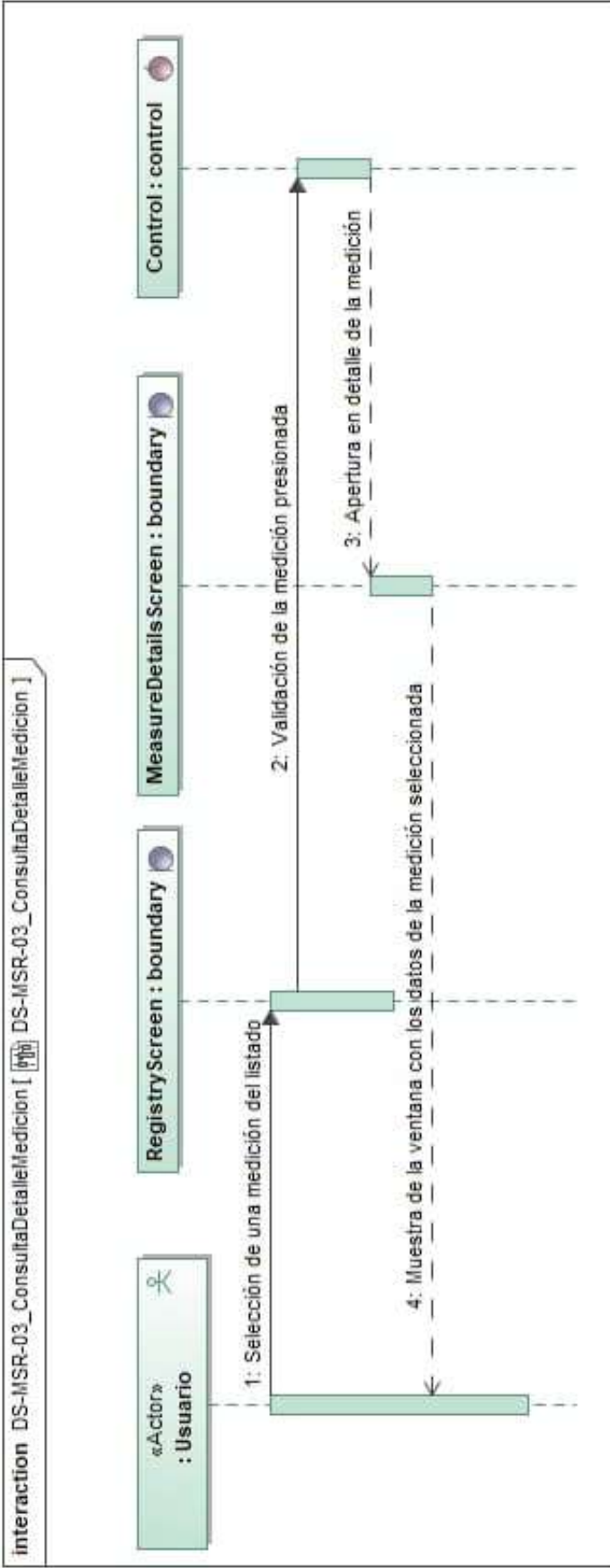


Figura 126: Diag. Secuencia de Consulta Detallada de una Medición (CU-MSR-03)

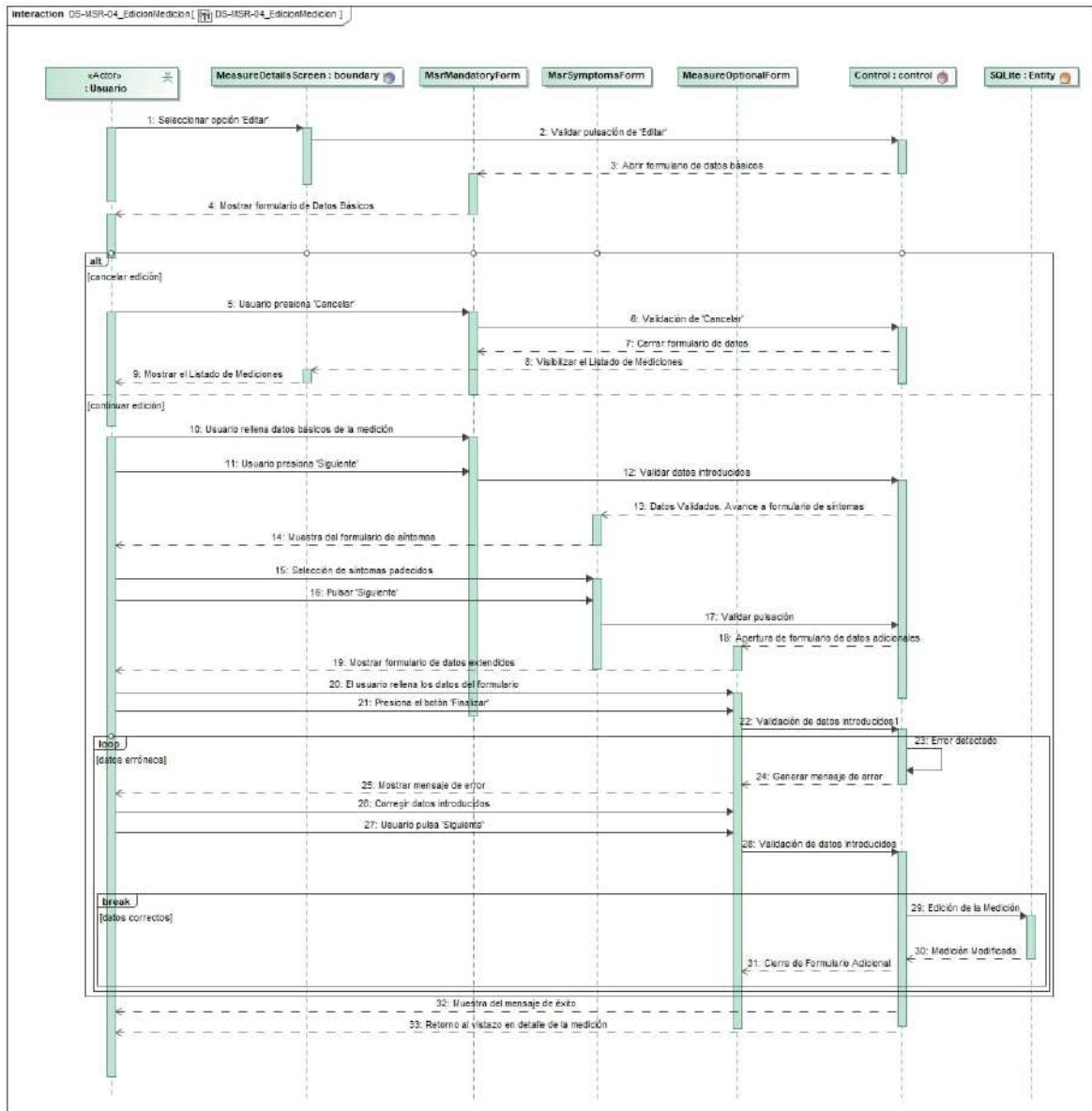


Figura 127: Diag. Secuencia de Edición de una Medición (CU-MSR-04)

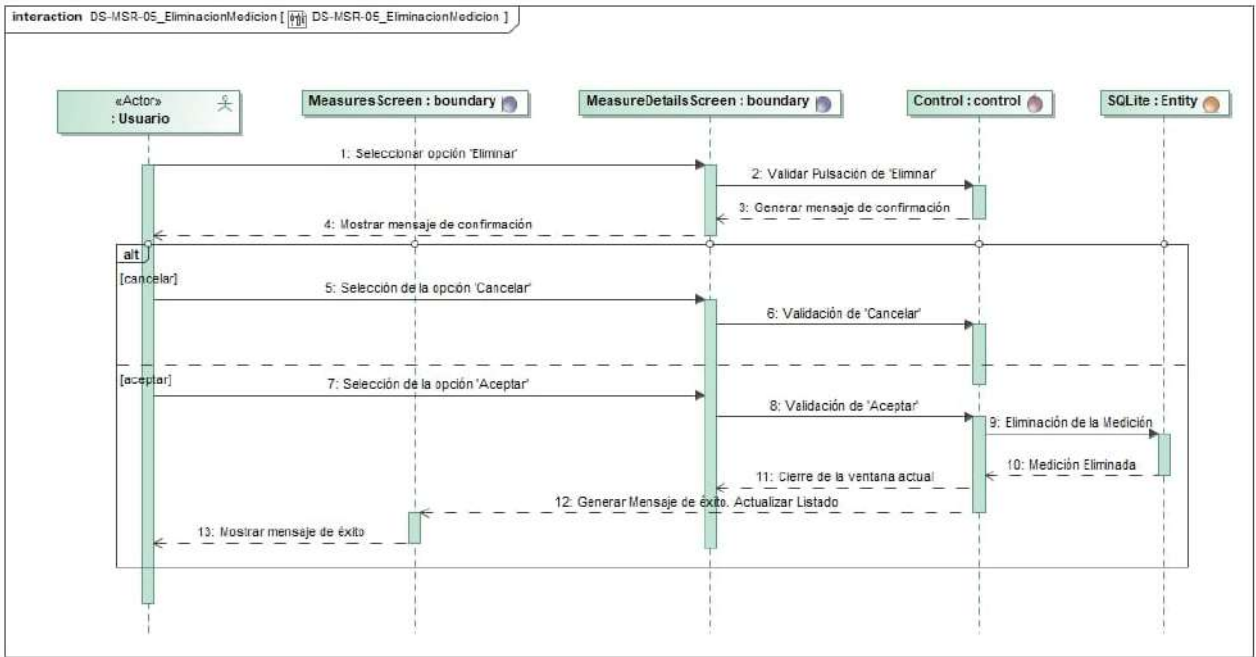


Figura 128: Diag. Secuencia de Borrado de una Medición (CU-MSR-05)

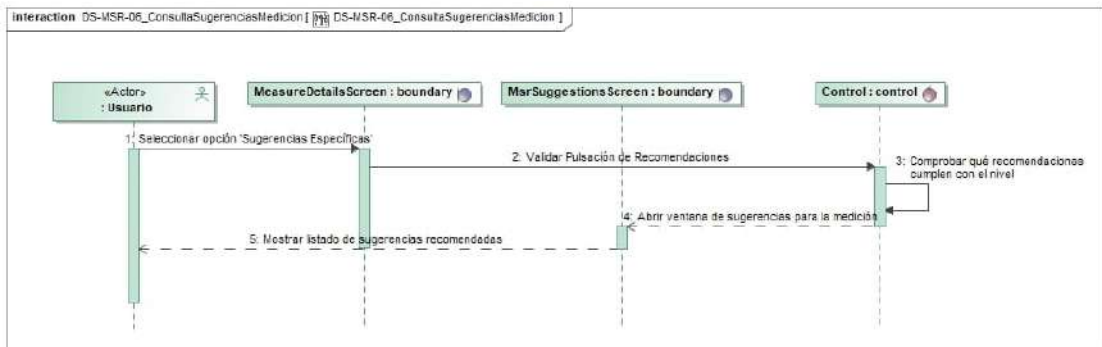


Figura 129: Diag. Secuencia de Consulta de Sugerencias para una Medición (CU-MSR-06)

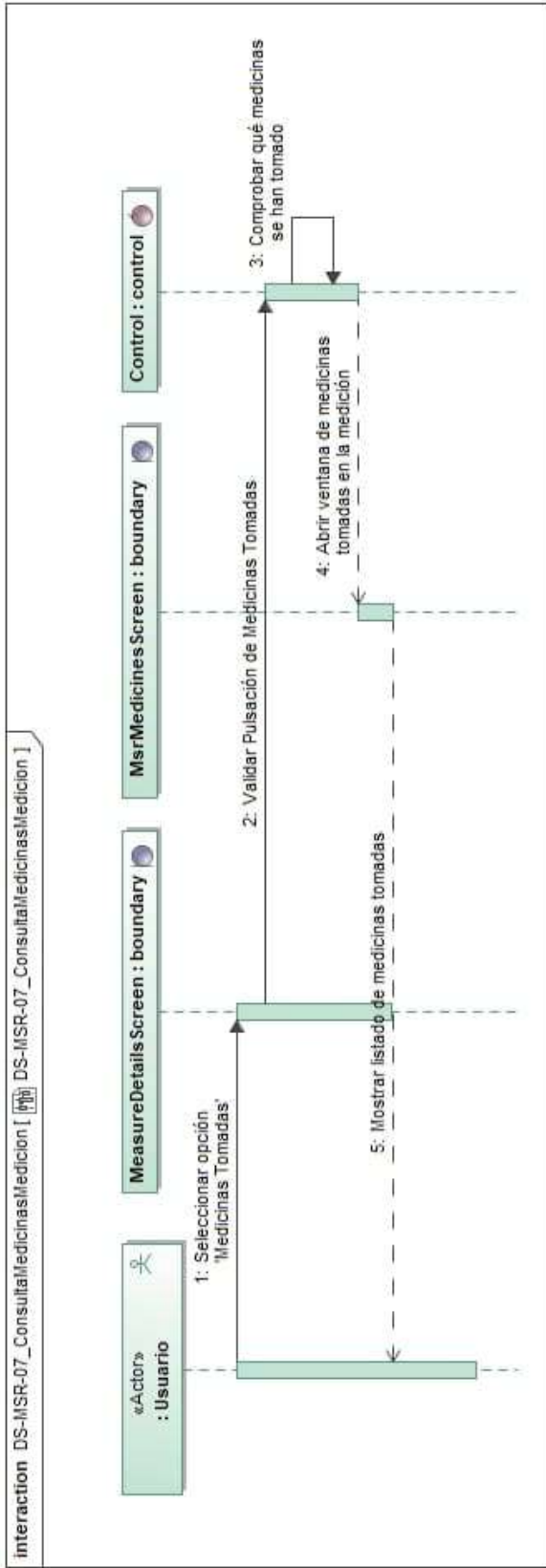


Figura 130: Diag. Secuencia de Consulta de Mediciones Tomadas para una Medición (CU-MSR-07)



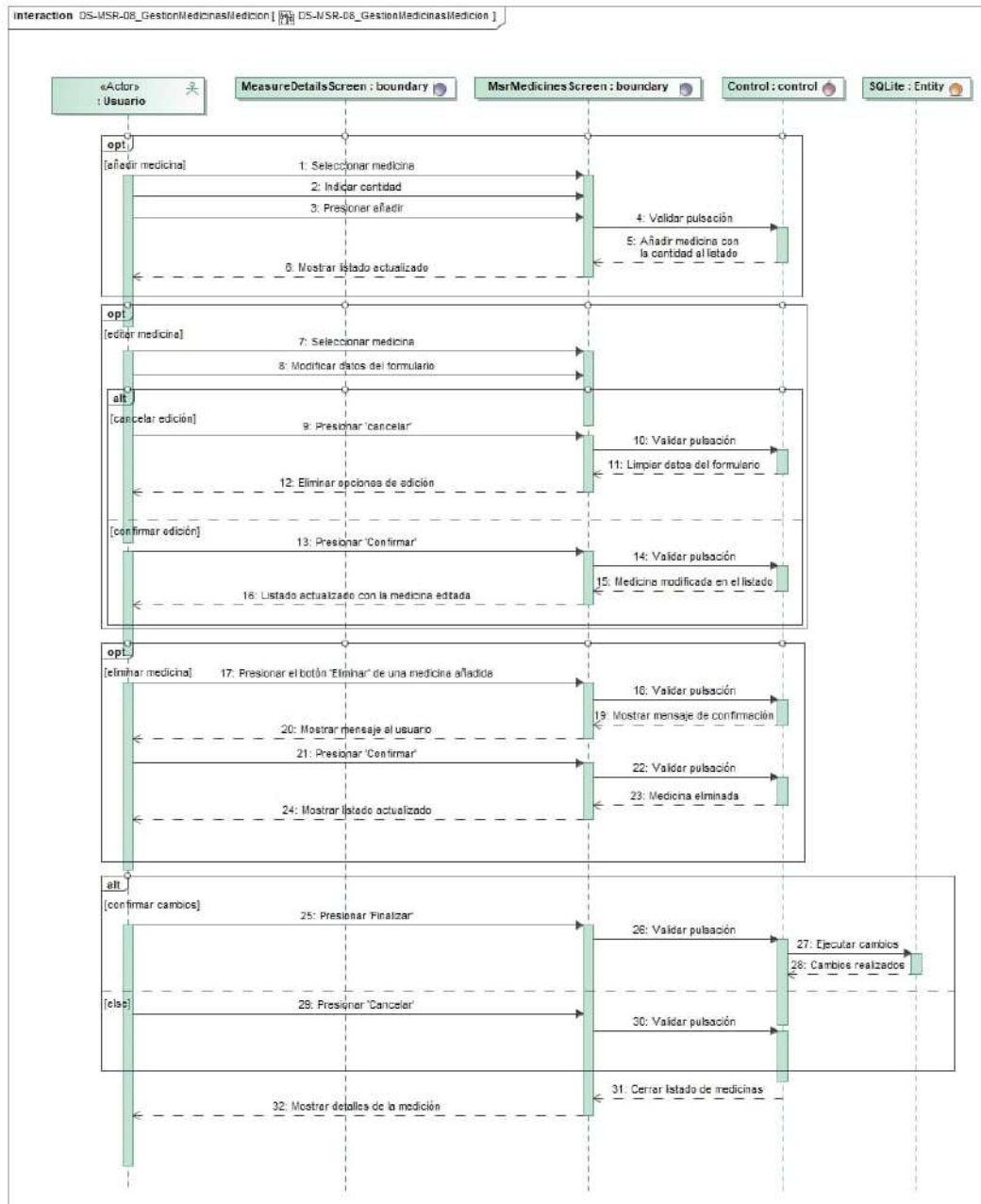


Figura 131: Diag. Secuencia de Gestión de Mediciones Tomadas en una Medicación (CU-MSR-08)



### C.7.4. Notificaciones (NTF)

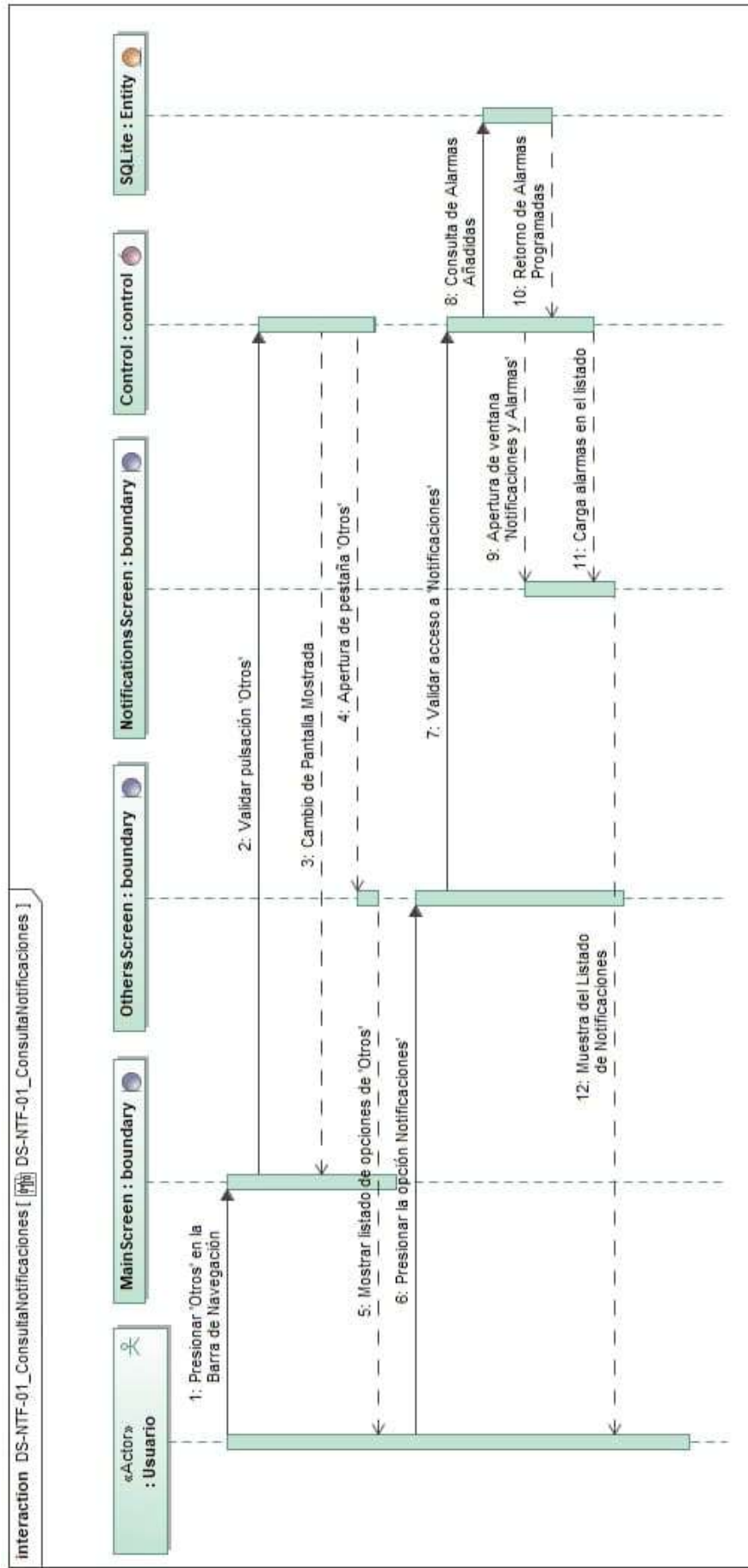


Figura 133: Diag. Secuencia de Consulta de Notificaciones Programadas (CU-NTF-01)

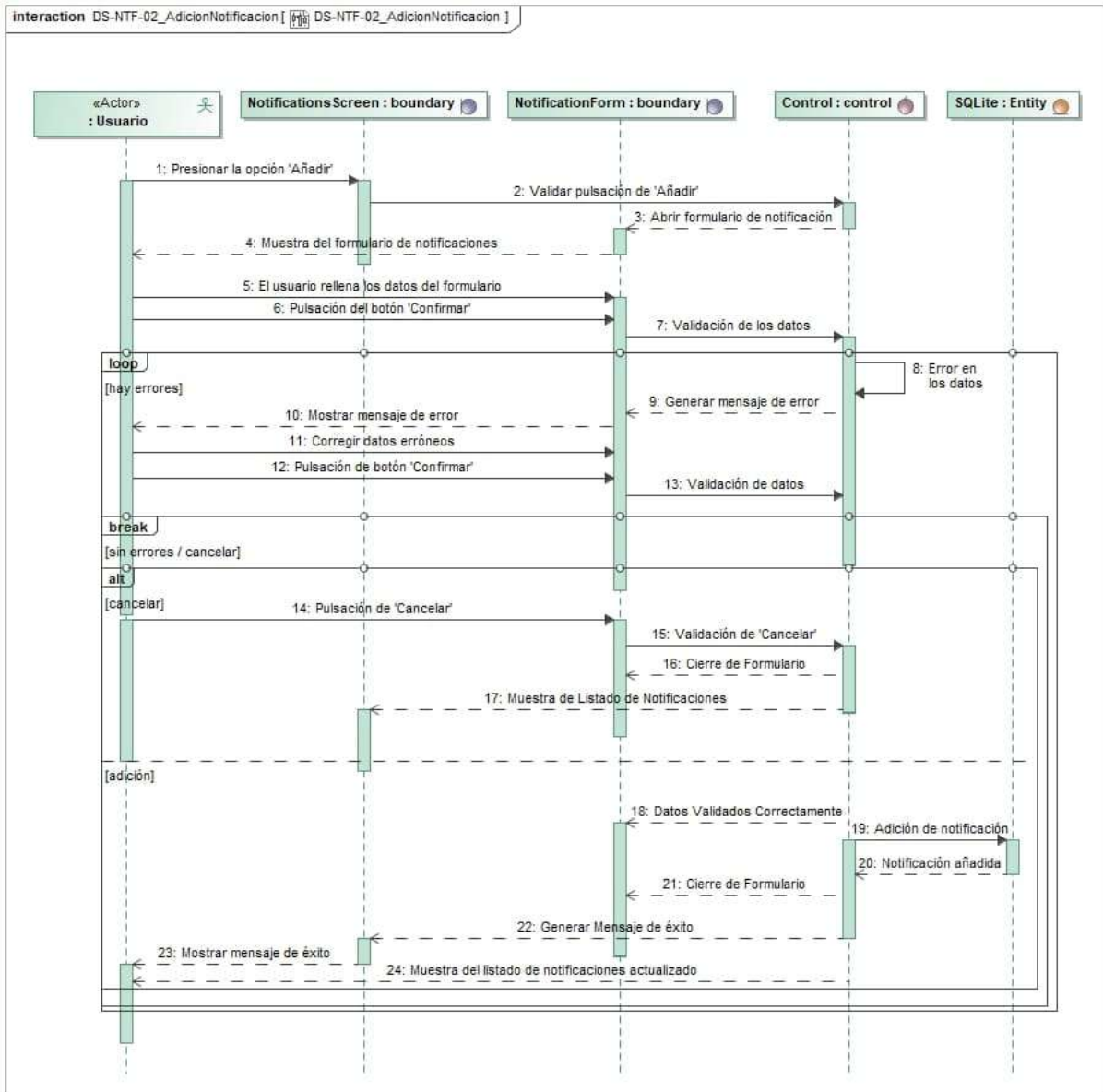


Figura 134: Diag. Secuencia de Programar una Nueva Notificación (CU-NTF-02)

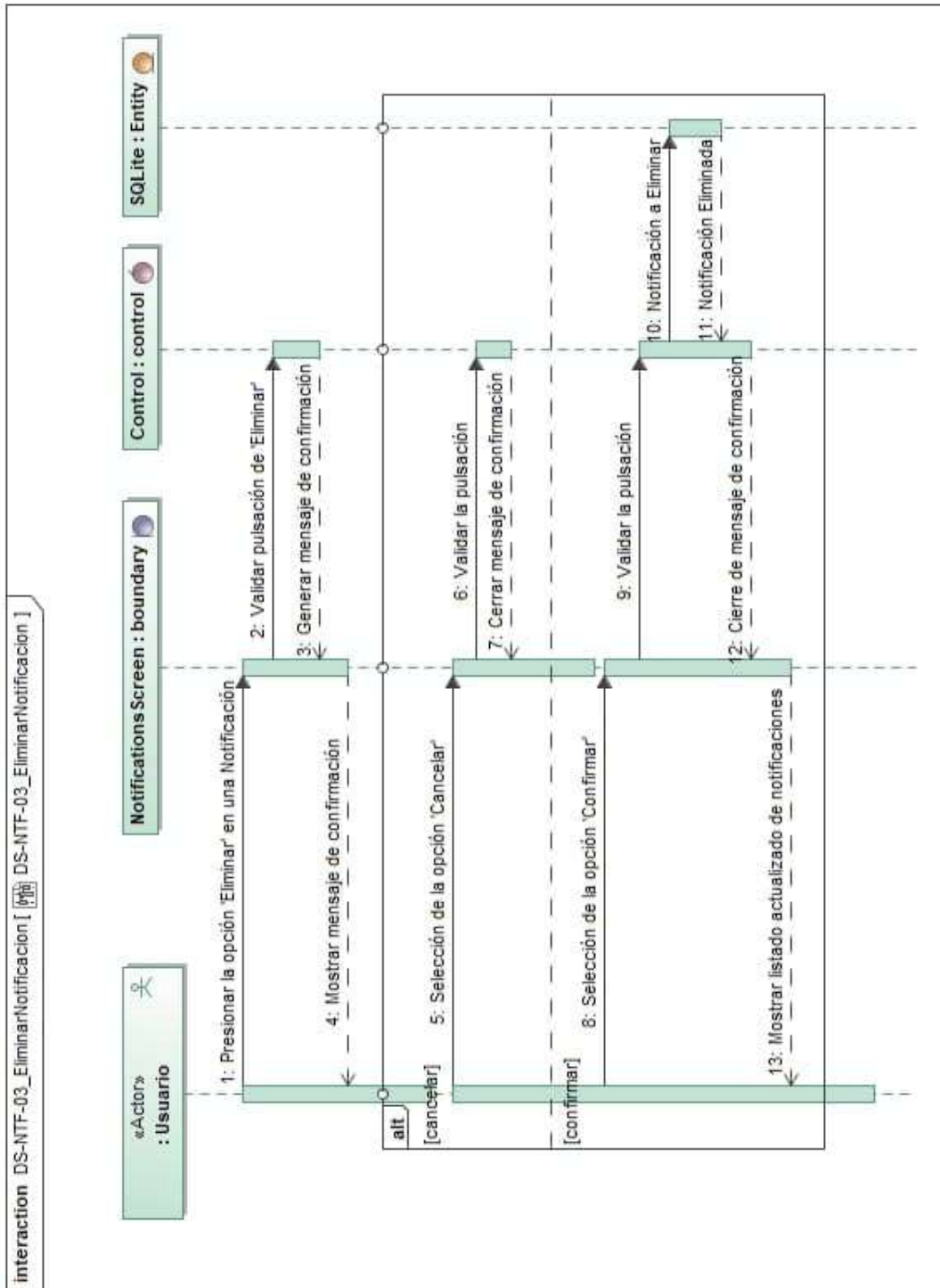


Figura 135: Diag. Secuencia de Borrado de una Notificación (CU-NTF-03)

### C.7.5. Alimentación / Nutrición (FOOD)

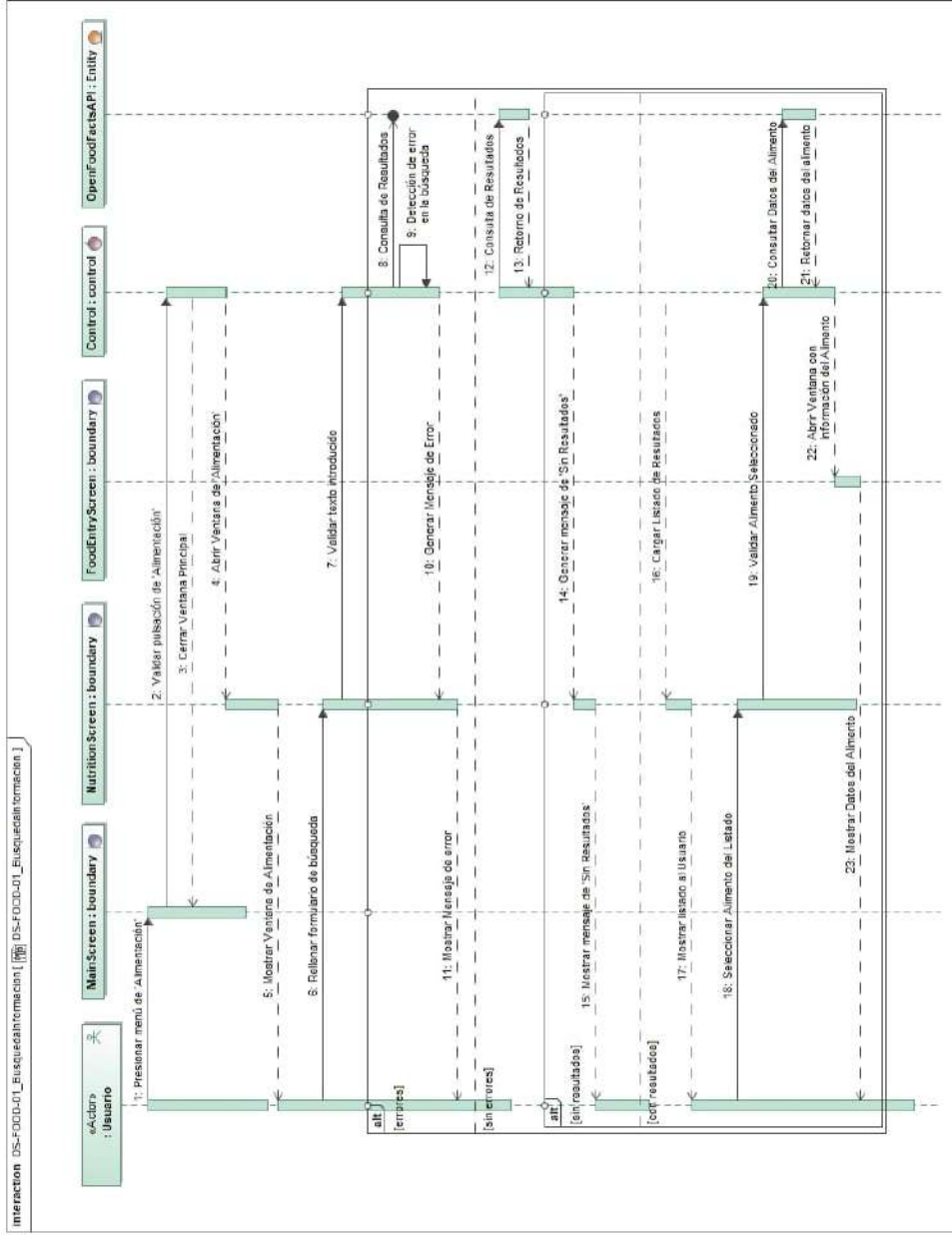


Figura 136: Diag. Secuencia de Búsqueda de Información Nutricional (CU-FOOD-01)

## **C.8. Requisitos No Funcionales**

Requisitos en relación con la manera en la que el sistema funciona y actúa durante la interacción del usuario con el mismo. Se nombrarán bajo la etiqueta **RNF**.

### **C.8.1. Requisitos de Aspecto**

#### **C.8.1.1. Interfaz**

**RNF1.** El sistema debe poder ser usado por personas con diferentes problemas de visión.

**RNF1.1.** El sistema debe seguir patrones reconocibles de colores.

**RNF1.2.** El sistema debe ser adaptado de manera que sea funcional bajo cualquier tamaño de letra del dispositivo.

**RNF1.3.** El sistema debe incluir un modo oscuro.

**RNF2.** El sistema debe ser asociado rápidamente con su aspecto principal: diabetes.

**RNF2.1.** El sistema debe poseer un icono reconocible para pacientes de diabetes.

**RNF2.2.** El sistema debe poseer un nombre fácil de recordar.

### **C.8.2. Requisitos de Facilidad de Uso y Aprendizaje**

#### **C.8.2.1. Facilidad de Uso**

**RNF3.** El sistema debe poder ser usado por personas de cualquier rango de edad.

**RNF3.1.** El sistema debe ser fácil de usar para usuarios menores de 16 años.

**RNF3.2.** El sistema debe ser fácil de usar para usuarios entre 16 y 55 años de edad.

**RNF3.3.** El sistema debe ser fácil de usar para usuarios mayores de 55 años.

**RNF4.** El sistema debe motivar al usuario para ser usado.

**RNF5.** El sistema debe poder ser utilizado de manera sencilla por cualquier persona.

**RNF5.1.** El sistema debe ser fácil de usar para usuarios sin formación sanitaria.

**RNF5.2.** El sistema debe ser fácil de usar para usuarios con poca experiencia en el uso de aplicaciones móviles.

### **C.8.2.2. Facilidad de Aprendizaje**

RNF6. El sistema debe proporcionar herramientas para ayudar al usuario a su uso.

RNF7. El sistema debe ser intuitivo para el usuario.

### **C.8.3. Requisitos de Funcionamiento**

#### **C.8.3.1. Requisitos de Velocidad**

RNF8. Se optimizará el sistema para que tareas de gestión en la base de datos se realicen rápidamente.

RNF8.1. Se optimizará el sistema para agilizar tareas de obtención de datos.

RNF8.2. Se optimizará el sistema para agilizar tareas de inserción de datos.

RNF8.3. Se optimizará el sistema para agilizar tareas de actualización de datos.

RNF8.4. Se optimizará el sistema para agilizar tareas de eliminación de datos.

#### **C.8.3.2. Requisitos de Seguridad Crítica**

RNF9. Los recursos del dispositivo deben tratarse de manera óptima.

#### **C.8.3.3. Requisitos de Precisión**

RNF10. Se formatearán unidades de medición en números sin decimales.

RNF10.1. No se permitirán valores de medición mayores a 500 unidades.

RNF11. Las unidades de glucosa en sangre se medirán en miligramo por decilitro (mg/dl).

RNF12. Las unidades de cada una de las medicinas se medirán en unidades (U).

#### **C.8.3.4. Requisitos de Fiabilidad y Disponibilidad**

RNF13. El sistema permitirá su uso 24 horas por día, 365 días por año.

RNF14. Las tareas que hagan uso de una red sólo estarán disponibles al establecerse una conexión estable.



#### **C.8.3.5. Requisitos de Capacidad**

*No se contemplan requisitos de capacidad.*

#### **C.8.3.6. Requisitos de Escalabilidad**

*No se contemplan requisitos de escalabilidad.*

#### **C.8.4. Requisitos Operacionales**

##### **C.8.4.1. Entorno Físico**

*No se contemplan requisitos destacables referentes al entorno físico.*

##### **C.8.4.2. Entorno Tecnológico**

*No se contemplan requisitos destacables referentes al entorno operacional.*

##### **C.8.4.3. Aplicaciones Recomendadas**

*No se contemplan requisitos destacables referentes aplicaciones recomendadas para el uso del producto.*

##### **C.8.4.4. Soporte**

*No se contemplan requisitos de soporte.*

#### **C.8.5. Requisitos de Mantenimiento y Portabilidad**

##### **C.8.5.1. Requisitos de Mantenimiento**

*No se contemplan requisitos de mantenimiento.*

##### **C.8.5.2. Requisitos de Portabilidad**

**RNF15.** Se espera que el sistema pueda ejecutarse en dispositivos móviles con sistema operativo Android o iOS.

**RNF16.** Se contempla la opción de incluir idioma inglés dentro del sistema.

#### **C.8.6. Requisitos de Seguridad**

**RNF17.** El acceso a los datos almacenados puede darse por cualquier usuario con acceso al dispositivo.

#### **C.8.7. Requisitos Culturales y Políticos**

*No se han encontrado motivos por los que el producto pueda ser inaceptable en otros lugares.*

#### **C.8.8. Requisitos Legales**

**RNF18.** El almacenamiento de datos en el sistema cumple con las leyes establecidas en el Artículo 6 del RGPD establecido por la UE en 2016.<sup>5</sup>

---

<sup>5</sup>"Lawfulness of Processing".Art. 6 RGPD [41].

## **C.9. Requisitos de Documentación**

### **C.9.1. Manual de Usuario**

El Manual de Usuario documentará desde que el sistema ha sido instalado en el dispositivo y cubrirá todos los aspectos y funcionalidades del mismo. Todos los ejemplos y pasos se ilustrarán con imágenes obtenidas directamente del producto final. Estas imágenes podrán ser modificadas o contar con dibujos para mostrar con exactitud los pasos a realizar.

Este manual seguirá un índice y un nivel de detalle lo más preciso posible para que el usuario entienda en todo momento la variedad de opciones que el sistema ofrece llegando a dominar la aplicación. Éste irá incluido en la memoria como apéndice.

### **C.9.2. Ayuda en Línea**

*El sistema no proporcionará ayuda en línea.*

### **C.9.3. Guía de Instalación y Configuración**

La instalación de la aplicación se documentará en un manual que servirá de apéndice en la memoria. Este contemplará todos los aspectos necesarios para probar la aplicación en un dispositivo externo, preferiblemente Android. Seguirá los mismos criterios que el Manual de Usuario en términos de recopilar información e imágenes con las que explicar los pasos necesarios.

## **C.10. Atributos de las Características Funcionales**

### **C.10.1. Esfuerzo**

La prioridad del proyecto se ha considerado alta debido a las complicaciones del mismo y a la fecha límite establecida para su fin. Esto ha provocado el trabajo activo en el mismo durante el período de desarrollo.

Se espera así que el coste temporal del proyecto sea de 296 horas distribuidas en las diferentes iteraciones a lo largo de tres meses. Esto proporciona un resultado aproximado de 24 h a la semana.

### **C.10.2. Riesgo**

Bajo el criterio del equipo, el proyecto consta de riesgos referentes a la constante actualización de las tecnologías usadas en el mismo. Se advierten de posibles retrasos eventuales en las iteraciones establecidas, que se corregirán rápidamente trabajando durante más tiempo del planeado inicialmente.

No se esperan riesgos significativos más allá de los ya establecidos. Se espera entregar el sistema desarrollado en la fecha establecida al inicio del mismo.

### **C.10.3. Estabilidad**

Atendiendo al ámbito del sistema, se esperan pocos cambios en las funcionalidades definidas. Este planteamiento hará que todo el desarrollo y cada iteración se mantengan acuerdo a la planificación hasta el final.

### **C.10.4. Asignada A**

El proyecto corre a cargo del alumno que lo desarrolla debido al interés existente en la aprobación del mismo.



# Apéndice D

# Documento General de Pruebas

En este apéndice se concretarán detalles de las pruebas realizadas planteando tanto el detalle de las mismas como el método usado para realizarlas. Esto se comentará de manera sencilla concretando los aspectos importantes de ellas, así como el proceso.

## D.1. Historial de Versiones

Fecha	Versión	Autor	Organización	Descripción
22/04/2021	1.0	Ángel Baeza	Universidad de Málaga	Pruebas. 2ª Iteración
13/05/2021	2.0	Ángel Baeza	Universidad de Málaga	Pruebas. 3ª Iteración
20/06/2021	3.0	Ángel Baeza	Universidad de Málaga	Pruebas. 4ª Iteración

Tabla 50: DGP. Versiones del Documento

## D.2. Información del Proyecto

<b>Empresa / Organización</b>	Universidad de Málaga
<b>Proyecto</b>	Aplicación Móvil para el Control y Ayuda al Paciente de Diabetes
<b>Fecha de Preparación</b>	22 de Abril de 2021
<b>Gerente / Líder de Proyecto</b>	Ángel Baeza
<b>Gerente / Líder de Pruebas Software</b>	Ángel

Tabla 51: DGP. Información del Proyecto

## D.3. Aprobaciones

Nombre y Apellido	Cargo	Departamento	Fecha	Firma
Ángel Baeza	Responsable	Ingeniería Software	xx/06/2021	
Gabriel Jesús Luque	Tutor	LCC	xx/06/2021	
Christian Cintrano	Cotutor	LCC	xx/06/2021	

Tabla 52: DGP. Aprobaciones del Proyecto

## **D.4. Resumen Ejecutivo**

El presente documento pretende mostrar las pruebas realizadas sobre el producto '*Aplicación Móvil para el Control y Ayuda al Paciente de Diabetes*'. El objetivo del mismo es mostrar de manera óptima y eficiente las capacidades del sistema y sus funcionalidades.

## **D.5. Alcance de las Pruebas**

### **D.5.1. Elementos de las Pruebas**

En este documento se tratarán las pruebas a todas aquellas ventanas y funcionalidades incorporadas al producto final comprobando la correcta completitud de los requisitos negociados al inicio del desarrollo. Los componentes que se probarán serán:

- Acceso a la aplicación.
- Ventanas presentes en la aplicación:
  - ◊ Perfil.
  - ◊ Registros.
  - ◊ Estadísticas.
  - ◊ Notificaciones.
  - ◊ Usuarios.
- Compatibilidad de la base de datos.
- Programación de notificaciones.
- Funcionalidad del módulo nutricional.
- Comprobación de documentos.

### **D.5.2. Funcionalidades a Probar**

Las pruebas se centrarán en comprobar en detalle cada uno de los módulos anteriores verificando que la aplicación está correctamente implementada en cada uno de los casos establecidos. Para ello, se establecerán una serie de acciones posibles en la aplicación contemplando:



### **PR-01. Acceso a la Aplicación**

- ◇ PR-01-A. Acceso a diferentes ventanas.
- ◇ PR-01-B. Navegación entre ventanas.
- ◇ PR-01-C. Visualización correcta de ventanas.

### **PR-02. Gestión de Usuarios**

- ◇ PR-02-A. Acceso al listado.
- ◇ PR-02-B. Adición.
- ◇ PR-02-C. Edición.
- ◇ PR-02-D. Borrado.
- ◇ PR-02-E. Intercambio.
- ◇ PR-02-F. Visualización correcta de datos en el Perfil.

### **PR-03. Gestión de Medicinas**

- ◇ PR-03-A. Acceso al listado.
- ◇ PR-03-B. Adición.
- ◇ PR-03-C. Edición.
- ◇ PR-03-D. Borrado.

### **PR-04. Gestión de Sugerencias**

- ◇ PR-04-A. Acceso al listado.
- ◇ PR-04-B. Adición.
- ◇ PR-04-C. Edición.
- ◇ PR-04-D. Borrado.
- ◇ PR-04-E. Gestión de medicinas añadidas.

### **PR-05. Gestión de Mediciones**

- ◇ PR-05-A. Acceso al listado.

- ◇ **PR-05-B.** Adición.
- ◇ **PR-05-C.** Edición.
- ◇ **PR-05-D.** Borrado.
- ◇ **PR-05-E.** Gestión de medicinas tomadas.
- ◇ **PR-05-F.** Presencia de sugerencias recomendadas.
- ◇ **PR-05-G.** Visualización correcta de datos.

#### **PR-06. Construcción y Visualización de Gráficos**

- ◇ **PR-06-A.** Acceso a gráficos sin mediciones añadidas.
- ◇ **PR-06-B.** Acceso a gráficos con mediciones añadidas.

#### **PR-07. Gestión de Notificaciones•**

- ◇ **PR-07-A.** Acceso al listado.
- ◇ **PR-07-B.** Adición.
- ◇ **PR-07-C.** Borrado.
- ◇ **PR-07-D.** Muestra de notificaciones.

#### **PR-08. Búsqueda de información nutricional**

- ◇ **PR-08-A.** Búsqueda sin resultados.
- ◇ **PR-08-B.** Búsqueda con resultados.
- ◇ **PR-08-C.** Mostrar mensaje de error en la búsqueda.
- ◇ **PR-08-D.** Muestra de alimentos seleccionados en detalle.
- ◇ **PR-08-E.** Muestra correcta de elementos no disponibles de un alimento.

#### **PR-09. Revisión de documentos realizados**

- ◇ **PR-09-A.** Comprobación del informe de requisitos.
- ◇ **PR-09-B.** Comprobación de la memoria y sus diferentes secciones.

### D.5.3. Funcionalidades a no Probar

Algunos de los elementos implementados se basan en el uso de librerías y *plugins* externos de los cuales únicamente se tiene acceso a sus funciones. Por este motivo, y debido a que es imposible o el trabajo que conlleva detrás, no se realizarán pruebas específicas para ellos, sino que se testeará su funcionamiento en integración con la aplicación.

### D.5.4. Enfoque de Pruebas (Estrategia)

La estrategia seguida para este informe contempla la realización de pruebas **estáticas** para la revisión y comprobación de documentos y de pruebas **dinámicas** para referirse al comportamiento de la aplicación. Dentro de las pruebas dinámicas se valorarán aquellas sin entrar en un diseño de código estricto:

- **Integración:** Para comprobar el funcionamiento de la aplicación cada vez que se incorporaba un elemento o funcionalidad nueva.
- **Humo:** Comprobando en todo momento que la aplicación no tuviera ningún defecto visual ni problemas al ser usada.
- **Regresión:** Fundado un poco sobre las anteriores, a fin de conseguir visualizar posibles problemas de la aplicación que interrumpían su correcto funcionamiento o diseño visual.

Para cada una de estas pruebas no se han realizado test unitarios a nivel de código, sino que las pruebas han sido elaboradas desarrollando la aplicación y ejecutando flujos de acciones que pudieran provocar errores; es decir, pruebas sobre un **entorno real**.

Las pruebas presentes en este mismo documento serán actualizadas con cada iteración, pues en cada momento se revisará no sólo lo nuevo implementado, sino también todo lo anterior. No se entrará a valorar el diseño programático de las mismas. Cada una de las pruebas se valorará mediante tres símbolos concretos con posibilidad de hacer comentarios en la misma. Los símbolos son:

[V] **Válido**

[X] **Error**

[-] **Intermedio**

Tabla 53: DGP. Simbología escogida para validar las pruebas

## **D.6. Criterios de Aceptación y Rechazo**

### **D.6.1. Criterios de Aceptación**

Debido al tipo de proyecto y el interés dedicado a que salga bien, se valorará la validez de los requisitos basándonos en un porcentaje de éxito del 80 % en el momento de la prueba.

### **D.6.2. Criterios de Rechazo**

Un requisito será rechazado como incompleto o no-cumplido cuando no esté presente en la aplicación o su funcionamiento no es el esperado teniendo en cuenta los requisitos definidos.

### **D.6.3. Criterios de Reanudación**

A lo largo del documento se han detallado diferentes iteraciones, lo que implica que tras cada una se reanudará la prueba de las funcionalidades tanto de cada una como de las anteriores a favor de que nada haya cambiado en el transcurso de la misma.

## **D.7. Recursos**

### **D.7.1. Requerimientos de Entornos - Hardware**

Las pruebas realizadas se harán sobre una copia del sistema en un dispositivo móvil real con sistema operativo Android MIUI 11. Este será el modelo **Xiaomi Redmi Note 9 Pro**. Se contará además con conexión a internet para ciertas pruebas.

### **D.7.2. Requerimientos de Entornos - Software**

Dado que el entorno de desarrollo del producto es móvil, y por la complicación en casos de modelos *Apple con iOS* para ejecutar aplicaciones de prueba, se requerirá un dispositivo que contenga Android como sistema operativo principal.

### **D.7.3. Personal**

Debido a que el proyecto sólo cuenta con una persona en el equipo de desarrollo, el personal encargado de realizar las pruebas será esta, el líder y ejecutor del proyecto Ángel Baeza.

## D.8. Pruebas Realizadas

### D.8.1. PR-01: Acceso a la Aplicación

Prueba	Validez	Observaciones
<i>Acceso a Ventanas</i> (PR-01-A)	V	Acceso correcto a todas las ventanas.
<i>Navegación entre Ventanas</i> (PR-01-B)	V	La navegación sigue el diagrama de navegación correctamente.
<i>Visualización de Ventanas</i> (PR-01-C)	V	Todas las ventanas se visualizan correctamente.

Tabla 54: DGP. Pruebas de Acceso a la Aplicación (PR-01)

### D.8.2. PR-02: Gestión de Usuarios

Prueba	Validez	Observaciones
<i>Acceso al Listado</i> (PR-02-A)	V	Acceso correcto al listado de usuarios. Visualización correcta de usuarios.
<i>Adición</i> (PR-02-B)	V	Usuario añadido correctamente. Retorno al listado de usuarios actualizado correcto.
<i>Edición</i> (PR-02-C)	V	Usuario editado correctamente. Retorno al perfil actualizado correctamente.
<i>Borrado</i> (PR-02-D)	V	Usuario eliminado correctamente. Retorno al listado de usuarios actualizado correctamente.
<i>Intercambio</i> (PR-02-E)	V	Usuario intercambiado correctamente. Retorno al perfil con los datos del nuevo usuario cargados correctamente.
<i>Visualización de Perfil</i> (PR-02-F)	V	Datos del usuario en perfil mostrados correctamente (con y sin definir).

Tabla 55: DGP. Pruebas de Revisión de Usuarios (PR-02)

### D.8.3. PR-03: Gestión de Medicinas

Prueba	Validez	Observaciones
<b>Acceso al Listado</b> <i>(PR-03-A)</i>	V	Acceso correcto al listado de medicinas. Visualización correcta de medicinas.
<b>Adición</b> <i>(PR-03-B)</i>	V	Medicina añadida correctamente.
<b>Edición</b> <i>(PR-03-C)</i>	V	Medicinas editada correctamente.
<b>Borrado</b> <i>(PR-03-D)</i>	V	Medicina eliminada correctamente.

Tabla 56: DGP. Pruebas de Gestión de Medicinas (PR-03)

### D.8.4. PR-04: Gestión de Sugerencias

Prueba	Validez	Observaciones
<b>Acceso al Listado</b> <i>(PR-04-A)</i>	V	Acceso correcto al listado de sugerencias. Visualización correcta de sugerencias.
<b>Adición</b> <i>(PR-04-B)</i>	V	Sugerencia añadida correctamente. Retorno al listado de sugerencias actualizado correcto.
<b>Edición</b> <i>(PR-04-C)</i>	V	Sugerencia editada correctamente. Retorno a los detalles de sugerencia actualizados correctamente.
<b>Borrado</b> <i>(PR-04-D)</i>	V	Sugerencia eliminada correctamente. Retorno al listado de sugerencias actualizado correcto.
<b>Gestión de Medicinas Añadidas</b> <i>(PR-04-E)</i>	V	Medicinas visualizadas correctamente. Medicinas añadidas correctamente. Medicinas editadas correctamente. Medicinas eliminadas correctamente. Visualización actualizada de medicinas correcta.

Tabla 57: DGP. Pruebas de Gestión de Sugerencias (PR-04)

**D.8.5. PR-05: Gestión de Mediciones**

<b>Prueba</b>	<b>Validez</b>	<b>Observaciones</b>
<b><i>Acceso al Listado (PR-05-A)</i></b>	V	Acceso correcto al listado de mediciones. Visualización correcta de mediciones.
<b><i>Adición (PR-05-B)</i></b>	V	Medición añadida correctamente. Retorno al listado de mediciones actualizado correcto.
<b><i>Edición (PR-05-C)</i></b>	V	Medición editada correctamente. Retorno a los detalles de medición actualizados correctamente.
<b><i>Borrado (PR-05-D)</i></b>	V	Medición eliminada correctamente. Retorno al listado de mediciones actualizado correcto.
<b><i>Gestión de Medicinas Tomadas (PR-05-E)</i></b>	V	Medicinas visualizadas correctamente. Medicinas añadidas correctamente. Medicinas editadas correctamente. Medicinas eliminadas correctamente. Visualización actualizada de medicinas correcta.
<b><i>Presencia de Sugerencias Recomendadas (PR-05-F)</i></b>	V	Acceso correcto al listado de sugerencias específicas. Visualización correcta de información (con y sin sugerencias).
<b><i>Visualización de Datos (PR-05-G)</i></b>	V	Acceso correcto a los detalles de la medición. Visualización correcta de información (añadida y sin definir).

Tabla 58: DGP. Pruebas de Gestión de Mediciones (PR-05)

**D.8.6. PR-06: Construcción y Visualización de Gráficos**



Prueba	Validez	Observaciones
<i>Acceso sin Mediciones Añadidas</i> (PR-06-A)	V	Gráfico no construido correctamente. Visualización correcta de mensaje 'No posible'.
<i>Acceso con Mediciones Añadidas</i> (PR-06-B)	V	Gráfico construido correctamente. Visualización correcta de gráfico.

Tabla 59: DGP. Pruebas de Gráficos (PR-06)

#### D.8.7. PR-07: Gestión de Notificaciones

Prueba	Validez	Observaciones
<i>Acceso al Listado</i> (PR-07-A)	V	Acceso correcto al listado de notificaciones. Visualización correcta de notificaciones.
<i>Adición</i> (PR-07-B)	V	Notificación añadida correctamente.
<i>Borrado</i> (PR-07-C)	V	Notificación eliminada correctamente.
<i>Muestra de Notificaciones</i> (PR-07-D)	-	Notificación mostrada una única vez al cumplir su período.

Tabla 60: DGP. Pruebas de Gestión de Notificaciones (PR-07)

#### D.8.8. PR-08: Búsqueda de Información Nutricional

Prueba	Validez	Observaciones
<i>Búsqueda sin Resultados</i> (PR-08-A)	V	Búsqueda realizada correctamente. Visualización correcta de mensaje 'Sin Resultados'.
<i>Búsqueda con Resultados</i> (PR-08-B)	V	Búsqueda realizada correctamente. Visualización correcta de resultados.
<i>Mostrar Mensaje de Error en Búsqueda</i> (PR-08-C)	V	Búsqueda no realizada correctamente. Visualización correcta de mensaje 'Error'.

<b><i>Muestra en Detalle de Alimentos Seleccionados (PR-08-D)</i></b>	V	Detalles de alimentos mostrados correctamente.
<b><i>Muestra Correcta de Datos no Disponibles (PR-08-E)</i></b>	V	Detalles no disponibles de alimentos mostrados correctamente.

Tabla 61: DGP. Pruebas de Información Nutricional (PR-08)

#### D.8.9. PR-09: Revisión de Documentos

<b>Prueba</b>	<b>Validez</b>	<b>Observaciones</b>
<b><i>Informe de Requisitos (PR-09-A)</i></b>	V	Informe de requisitos actualizado correctamente. Requisitos disponibles refinados.
<b><i>Memoria Realizada (PR-09-B)</i></b>	V	Memoria completa. Memoria actualizada correctamente

Tabla 62: DGP. Pruebas de Revisión de Documentos (PR-09)



UNIVERSIDAD  
DE MÁLAGA

| [uma.es](http://uma.es)

E.T.S de Ingeniería Informática  
Bulevar Louis Pasteur, 35  
Campus de Teatinos  
29071 Málaga

E.T.S. DE INGENIERÍA INFORMÁTICA