



DiMo (Diabetes Monitoring). PEC 3

Fernando López Tejada

Máster Universitario en Desarrollo de Aplicaciones para Dispositivos
Móviles

Nombre Consultor/a: Francesc D'Assís Giralt Queralt

Profesor/a responsable de la asignatura: Carles Garrigues Olivella

Diciembre 2017

C) Copyright

© (Fernando López Tejeda)

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>DiMo (Diabetes Monitoring).</i>
Nombre del autor:	<i>Fernando López Tejada</i>
Nombre del consultor/a:	<i>Francesc D'Assís Giralt Queralt</i>
Nombre del PRA:	<i>Carles Garrigues Olivella</i>
Fecha de entrega (mm/aaaa):	10/2017
Titulación:	Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles
Idioma del trabajo:	<i>Castellano.</i>
Palabras clave	<i>Diabetes, Monitorización y Seguimiento</i>

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.*

La diabetes es una enfermedad que padecen cientos de millones de personas en la actualidad, y se espera que este número aumente durante los próximos años debido a múltiples parámetros relacionados no solo con factores genéticos, sino también con un cambio en los hábitos de vida y el incremento sedentarismo, según la OMS. La diabetes es un importante problema de salud pública cuya intervención requiere un carácter prioritario.

El proyecto presentado en esta memoria tiene como objetivo principal mejorar la calidad de vida de pacientes con diabetes Tipo 1 y Tipo 2, permitiendo un seguimiento integral, personalizado e inteligente de los parámetros más relevantes relacionados con la enfermedad, posibilitando el análisis del comportamiento y evolución de la misma tanto por el propio paciente como por especialistas y familiares.

Para ello, se ha desarrollado un sistema compuesto por una aplicación móvil y un conjunto de librerías y servicios necesarios para el estudio de los distintos factores que influyen en el transcurso de la enfermedad. A través de la aplicación móvil el usuario dispondrá de una herramienta sencilla para la obtención de información asociada a la actividad física, ingesta nutricional, niveles de glucosa y demás parámetros relevantes, de la forma más natural y menos intrusiva posible, haciendo uso de sensores del propio Smartphone y

dispositivos externos, así como de forma manual. Gracias al conjunto de librerías desarrolladas y a los servicios de almacenamiento en la nube, los datos son almacenados para su posterior análisis, permitiendo al paciente llevar un auto-control efectivo de su diabetes.

Dicho sistema supone un complemento fundamental a la atención médica tradicional, apoyándose de las nuevas tecnologías y facilitando el autocontrol y seguimiento de la enfermedad, en gran medida gracias al estudio evolutivo que puede derivarse, así como un medio esencial de prevención de las principales consecuencias a un control erróneo o insuficiente de la enfermedad.

Abstract (in English, 250 words or less):

Millions of people are suffering from Diabetes Mellitus today. This amount is expected to increase over the next few years due to multiple factors, not only genetic ones, but also because of our sedentary lifestyle according to the World Health Organization (WHO). Diabetes is an important public health issue whose intervention is a priority.

The main aim of this research work is to improve the quality of life of patients with Diabetes (Type I and II), providing an integral, personalised and smart management and monitoring of relevant parameters related to this disease. As well as enabling the analysis of behaviour and evolution by the own diabetic patients and experts, through new technologies and devices. 9

We have developed a software system consisting of a a mobile application and a set of libraries and cloud services to facilitate the study of the whole group of considered influential parameters of the disease. By means of a mobile application, user has a simple tool to gather data from physical activity, nutritional intake, glucose levels and other parameters in a natural a non-intrusive way, making use of Smartphone sensors and external devices, and also manually. Software libraries and cloud services provide a trustworthy and secure storage of patient information which is analysed and displayed by the system applications, providing patients an essential mechanism to carry out effective self-control of their Diabetes.

The developed system is an essential complement to traditional medical care, by using current technologies to facilitate diabetes self-control and monitoring. Patients can observe the evolution of their disease helping to prevent and reduce the main consequences from a wrong control of Diabetes.

Índice

1. INTRODUCCIÓN	1
1.1. Contexto	1
1.2. Breve descripción de los capítulos de la memoria	7
2. OBJETIVOS DEL TRABAJO	9
2.1. Descripción de objetivos	9
2.2. Ejemplo de usuarios y escenarios	11
2.2.1. Ejemplo 1	11
2.2.2. Ejemplo 2	12
3. ENFOQUE Y MÉTODO SEGUIDO	15
4. PLANIFICACIÓN DEL TRABAJO	17
4.1. Herramientas y tecnologías	17
4.1.1. Herramientas de Software	17
4.1.2. Herramientas de Hardware	18
4.2. Planificación de las tareas	19
5. ETAPA INICIAL DEL DESARROLLO	23
5.1. Captura de requisitos	24
5.1.1. Requisitos funcionales	24
5.1.2. Requisitos no funcionales	26
5.2. Navegación por la aplicación	29
5.1. Casos de uso	31
5.2. Prototipo inicial	34
5.3. Evaluación	38
6. Implementación	41
6.1. Estructura del proyecto	41
6.2. Funcionalidades, herramientas y recursos	43
6.2.1. Google Sign-In	43
6.2.2. MPAndroid Chart	45
6.2.3. Android Volley.....	47
6.2.4. Material Design	48

6.2.5. Google Fit – Monitorización Actividad Física	50
6.2.6. Problemas en la autenticación con API REST DiMo.....	52
7. Conclusiones.....	53
7.1. Propuestas de mejora.....	54
Anexo 1 Manual de Usuario	55
1. Guía de instalación y puesta en marcha.....	55
2. Guía de uso de la App	58
Anexo 2 Test de Evaluación.....	65
1. Tareas “Think aloud”	65
2. Cuestionario de perfil y usabilidad	66
Bibliografía.....	69

Lista de figuras

Ilustración 1 Diagrama General del sistema.....	11
Ilustración 2 Diagrama de etapas del Prototipado Evolutivo	16
Ilustración 3 Planificación Semanal	19
Ilustración 4 Etapas PEC1	20
Ilustración 5 Etapas PEC2	20
Ilustración 6 Etapas PEC3	21
Ilustración 7 Etapas PEC4	22
Ilustración 8 Estructura de la aplicación	29
Ilustración 9 Árbol de navegación de la aplicación	30
Ilustración 10 Diagrama de casos de uso general del cliente Android.....	31
Ilustración 11 Diagrama de caso de uso de la sección estadísticas	32
Ilustración 12 Diagrama de caso de uso de recibir alarma	33
Ilustración 13 Logo de la aplicación.....	34
Ilustración 14 Pantalla de inicio de sesión.....	35
Ilustración 15 Pantalla de resumen.....	36
Ilustración 16 Menú desplegable	36
Ilustración 17 Pantalla de perfil.....	37
Ilustración 18 Pantalla de estadísticas	38
Ilustración 19 Estructura carpetas del cliente Android.....	41
Ilustración 20 Configuración Google Sign-In	44
Ilustración 21 Función de Autenticación Google	45
Ilustración 22 Configuración del gráfico.....	46
Ilustración 23 Código XML LineChart	46
Ilustración 24 Petición HTTP con Volley.....	48
Ilustración 25 Ejecutar petición Volley	48
Ilustración 26 CardView y RecyclerView	49
Ilustración 27 NavigationView	50
Ilustración 28 Conectar GoogleApiClient.....	51
Ilustración 29 Método OnDataPoint.....	51
Ilustración 30 Instalación de la App	55
Ilustración 31 Login a través de DiMo App.....	56

Ilustración 32 Cookie de sesión	57
Ilustración 33 DiMo Inicio de sesión	58
Ilustración 34 Menu desplegable	59
Ilustración 35 Pantalla de resumen.....	60
Ilustración 36 Botón flotante.....	60
Ilustración 37 Mi perfil	61
Ilustración 38 Pantalla de estadísticas	62
Ilustración 39 Introducción de nutrición	63
Ilustración 40 Citas médicas	63

1. INTRODUCCIÓN

1.1. Contexto

La diabetes es un conjunto de enfermedades metabólicas caracterizadas por una hiperglucemia (aumento de azúcar en sangre), bien por la falta de producción de insulina en el páncreas (diabetes tipo 1), o bien porque la insulina no actúa adecuadamente (diabetes tipo 2), e incluso por la combinación de ambos casos. El paciente diabético debe tener cuidado también con episodios de hipoglucemia que deben considerarse.

- Diabetes mellitus tipo 1. Aparece generalmente en la infancia de la persona y requiere el tratamiento permanente con insulina, ya que esta no se produce en el organismo de dicha persona.

- Diabetes mellitus tipo 2. También llamada “de adulto”. Aunque exista una escasez de insulina, se produce una resistencia de las células a dicha hormona, desarrollándose un déficit en su producción por el páncreas.

- Diabetes gestacional. Es la diabetes que se produce durante el embarazo.

- Otros tipos de diabetes. También existen otras diabetes poco frecuentes como diabetes tipo MODY (diabetes del adulto en la infancia), mitocondrial, neonatal, etc.

Hoy en día, la prevalencia de diabetes en el mundo supone que más de 420 millones de personas sufren esta enfermedad, según la Organización Mundial de la Salud (OMS), afectando a un 15% de la población. La diabetes de tipo 2 es la más extendida (alrededor de un 95% de la población diabética) y suele estar vinculada al estilo de vida y los hábitos nutricionales. En general, se

calcula que en el año 2030, 552 millones de personas estén afectadas por algún tipo de diabetes [1].

Por otra parte, se estima que la diabetes y sus complicaciones son los responsables de la ocupación del 20-30% de las camas hospitalarias y del 6,5-7,5% del gasto sanitario nacional. Según algunos estudios realizados por la London School of Economics, los costes directos en España procedentes de tratamientos y hospitalizaciones a causa de la diabetes alcanzan los 5.450 millones de euros al año, lo que supone un coste anual por paciente de 1.708 euros.

La calidad del servicio asistencial recibido, los hábitos de vida del paciente, el tratamiento que recibe y la educación entorno a la enfermedad conforman algunos de los ejes de los nuevos estudios sobre diabetes en España.

Tanto la diabetes tipo 1 como la tipo 2 están influenciadas por una serie de factores, clínicos en su mayoría. El seguimiento y control de determinados factores incide directamente en la calidad de vida de la persona diabética. Estos factores están incluidos en distintos aspectos o dominios de actuación, a destacar: el aspecto clínico, nutricional, actividad física, el entorno familiar, etc.

El creciente y masivo uso de dispositivos Smartphone en nuestra sociedad y la continua aparición de dispositivos inteligentes capaces de comunicarse entre sí, posibilitan el desarrollo de soluciones de monitorización continua y no intrusiva de enfermedades como la diabetes. El hecho de mantener un control autónomo apoyado por estas tecnologías y bajo la supervisión del propio paciente, médico y familiares, mejoraría considerablemente la calidad de vida de los pacientes y su entorno, reduciendo diversos costes médicos asociados.

En la actualidad existen multitud de aplicaciones, tanto móviles como de otra naturaleza para el control y la gestión de la diabetes. Habitualmente estas aplicaciones tratan algunos de los factores relevantes asociados a la

enfermedad y no realizan un análisis centralizado de todos esos factores, tal y como propone el proyecto DiMo. Además, los valores de la totalidad de los parámetros manejados por dichas herramientas son introducidos directamente por el usuario, induciendo a errores humanos. Cada vez más, se están realizando esfuerzos en lograr una monitorización continua y automatizada, pero, aun así, la mayoría de las aplicaciones necesitan de una dedicación e interacción casi constante por parte del paciente.

A continuación se presentan algunos de los sistemas más representativos hasta la fecha en el ámbito de la monitorización y gestión de la diabetes llevados a cabo por la comunidad científica y desarrolladores independientes:

- Nightscout: Es un proyecto “open source” surgido en Estados Unidos que posibilita el uso e integración, de forma no oficial, de distintos sistemas de monitorización continua de glucosa comerciales en un sistema propio con el objetivo de que familiares y padres tengan acceso a las medidas de glucosa de sus hijos sin necesidad de estar alrededor de ellos, gracias a dispositivos Smartphone y servicios en la nube [2].

- Glooko [3] y Diabeto [4]: A partir de sendos proyectos de investigación surgen estos sistemas comerciales hoy en día. Ambos plantean una solución similar basada en la obtención de datos de glucosa de glucómetros Bluetooth compatibles a través de dispositivos hardware propios y haciendo uso de aplicaciones de gestión de diabetes. Estos sistemas están basados en planes mensuales de servicios en los cuales, además de posibilitar la obtención de valores de glucosa, posibilitan el seguimiento nutricional a través de expertos, así como otras funciones.

- Aplicaciones móviles: Existen bastantes aplicaciones móviles con multitud de objetivos bajo el nexo común de la gestión de la diabetes y la ayuda al paciente diabético. Aunque algunas son capaces de automatizar la obtención de medidas, gracias a proyectos como los mencionados anteriormente, la mayoría requieren que el usuario introduzca los datos necesarios para poder

llevar a cabo un seguimiento de la diabetes. Casos como Glimp [5] capaz de trabajar con monitores de glucosa comerciales y Glup! [6] para una gestión sencilla de los valores de glucosa y ejercicio físico.

Por otra parte, existe un gran número de compañías del campo de la salud que ofrecen sus productos de forma comercial a todo el colectivo de pacientes diabéticos. Dichos productos suelen estar licenciados clínicamente, consistiendo en dispositivos hardware y aplicaciones software de seguimiento y gestión privativa, es decir, cerrados al resto de usuarios diabéticos que opten por otro tipo de soluciones. Igualmente son productos caros no subvencionados en los que la venta de consumibles que hacen funcionar estos sistemas constituye la principal fuente de ingresos de estas compañías. Dexcom [6], es una de las principales compañías que ofrecen este tipo de productos y servicios.

Para la realización del proyecto es imprescindible conocer la enfermedad de la Diabetes Mellitus, así como identificar todo aquello potencialmente útil, desde el punto de vista clínico, que permita definir los requisitos del sistema a desarrollar.

El proyecto propone la realización de un sistema para facilitar el seguimiento integral de la enfermedad tanto por parte de facultativo y familiares como del propio paciente, tal y como se especificó anteriormente. Para ello, se deben tener en cuenta diferentes aspectos relacionados con la Diabetes tales como actividad física, niveles de glucosa, perfil nutricional o educación relativa a la enfermedad. Así mismo, gran parte de esta información es proporcionada por los diferentes dispositivos externos como smartphones, sensores, relojes inteligentes, brazaletes de actividad, etc., y almacenada para su posterior análisis y procesamiento. El paciente recibe en tiempo real toda la información de interés sobre su enfermedad, manteniendo ésta bajo control, haciendo uso de su Smartphone y de cualquier dispositivo inteligente vinculado, así como de las indicaciones del especialista. El médico, teniendo en cuenta el perfil del paciente y los datos procedentes de los dispositivos externos, puede establecer unos objetivos y una línea de tratamiento, pudiendo analizar la evolución del

paciente en tiempo real desde cualquier lugar, como complemento al tratamiento clásico de la enfermedad.

Después de realizar un estudio de mercado y observando la cuota actual del uso de dispositivos smartphones, se ha elegido desarrollar una aplicación nativa con sistema operativo Android como la opción más óptima, teniendo en cuenta la idea de poder abarcar el mayor número de usuarios posibles, ya que 4 de cada 5 teléfonos móviles que se usan en Europa son smartphones, y además, el sistema operativo Android es el más utilizado por los usuarios, situándose 70 puntos por encima del perseguidor más próximo, iOS de Apple, destacando también el precio más asequible y variedad de dispositivos que utilizan el sistema operativo de Google. [9]

El hecho de mejorar la calidad de vida de pacientes diabéticos a través de la integración de nuevas tecnologías, supone un reto y una motivación para el desarrollo del proyecto.

Del conocimiento detallado de la diabetes y del paciente diabético y a partir de la literatura [8] y la consulta a expertos del entorno cercano, se han obtenido los factores más relevantes en el seguimiento de la enfermedad y por consiguiente los parámetros a tener en cuenta en el desarrollo de la aplicación:

- Nivel de glucemia en sangre. El conocimiento que aporta el valor de glucosa en sangre es determinante para el control de la enfermedad, y del que dependerán el resto de factores.
- Dieta. Es importante destacar que seguir una dieta saludable y bien equilibrada es uno de los aspectos más importantes en la diabetes para tener la enfermedad bajo control. En este caso, se deben tener en cuenta las raciones de hidratos de carbono², índice glucémico de los alimentos, calorías, y otros datos relacionados con el aporte nutricional (grasas, proteínas, etc.).

- Actividad y ejercicio físico. El control de la actividad diaria y el ejercicio físico es otro de los factores a considerar en el seguimiento del paciente diabético. Conocer las calorías quemadas, los pasos dados en una caminata, el tiempo realizando una actividad física, e incluso los lugares frecuentados son de gran utilidad para un seguimiento y detallado.
- Nivel de estrés y desarrollo de emociones. Ante situaciones que provocan estrés y cambios emocionales, el paciente diabético debe responder realizando un control más exhaustivo. En este caso, el estudio de las horas de sueño y su estado, así como la frecuencia cardíaca, pueden ser estudiados para vislumbrar situaciones anormales que influyan en los cambios producidos durante el control de la enfermedad.
- Educación diabetológica. El paciente diabético es la persona que mejor deberá conocer su estado y respuestas del organismo en todo momento. Saber lo que hay que hacer en determinados casos, conocer la administración de insulina, y los hábitos instaurados de una vida saludable son algunos elementos que deben aprender estos pacientes desde el momento en que debutan con diabetes.
- Tipo y dosis de insulina. Estos datos son de especial relevancia, que el paciente diabético deberá conocer para una correcta administración. No solamente es importante e interesante su conocimiento por dicha persona, sino que estos datos deben ser conocidos por familiares y facultativos.
- Hemoglobina glicosilada. Este dato obtenido a través del análisis de sangre, ofrece información, sobre todo al experto, de cómo ha sido el control glucémico de una persona con diabetes en los últimos tres meses. El resultado suele ser un porcentaje y los valores normales oscilan entre el 4 y el 6%, recomendando a las personas con diabetes

que lo mantengan por debajo del 6,5%. Resultados anormales indican que los niveles de glucosa no han estado bien regulados y controlados.

- Debut diabético. Es importante conocer la edad a la que debutó el paciente diabético. Este dato aporta una información de vital importancia en el seguimiento de la diabetes y su tipo.
- Parámetros personales y antropométricos. Entre los datos personales de una persona que son de interés en el estudio de la diabetes, pueden incluirse aquellos como el peso, la talla, edad, Índice de masa corporal (IMC), etc.

1.2. Breve descripción de los capítulos de la memoria

Este documento está estructurado en siete capítulos, un anexo y un apartado con las referencias utilizadas. A continuación se van a listar cada uno de ellos junto a una descripción sobre su contenido:

Capítulo 1, Introducción: Se expone cual es el problema a resolver, explicando el contexto en el que se va a desarrollar el proyecto, justificando las decisiones tomadas y presentando la solución a realizar.

Capítulo 2, Objetivos: Se describe los objetivos principales que se quieren alcanzar al finalizar este trabajo fin de máster, así como ejemplos de escenarios y usuarios que potencialmente utilizarán la aplicación.

Capítulo 3, Enfoque y método seguido: Se expone la metodología de trabajo a seguir para llevar a cabo la realización del proyecto.

Capítulo 4, Planificación del trabajo: se describen las tecnologías y herramientas software y hardware utilizadas y se comenta, por medio de tablas, la planificación de todo el ciclo de desarrollo del proyecto.

Capítulo 5, Etapa inicial del desarrollo: Se detallan todas las tareas realizadas en los inicios del proyecto: descripción de actores que intervienen, captura de requisitos, funcionalidad de la aplicación, diagramas y prototipado.

Capítulo 6, Implementación: Se describe los puntos más significativos a la hora de codificar la aplicación: *frameworks*, librerías, APIs, guías de diseño, etc.

Capítulo 7, Conclusiones: Se realiza una evaluación de los objetivos propuestos en el capítulo 2 y se comenta las posibles mejoras que se podrían realizar en futuras versiones de la aplicación.

Anexo A, Manual de usuario: Guía que describe una serie de pautas que deben de seguir los usuarios para conocer y entender las funcionalidades que es capaz de realizar la aplicación, así como los pasos a seguir para instalar el software en sus dispositivos.

2. OBJETIVOS DEL TRABAJO

2.1. Descripción de objetivos

Este proyecto supone el desarrollo e integración de un sistema inteligente adaptado, para el seguimiento de pacientes diabéticos haciendo uso de dispositivos Smartphone y tecnologías de la información como complemento a la atención médica tradicional, con los siguientes objetivos:

- a) Monitorizar los aspectos relacionados con la enfermedad (niveles de glucosa en sangre, actividad física, parámetros nutricionales, etc.).
- b) Favorecer el intercambio y almacenamiento de la información monitorizada de forma estructurada.
- c) Facilitar el autocontrol del paciente, y asentar las bases en el seguimiento y estudio asíncrono de los datos relevantes por parte de familiares y expertos.
- d) Permitir el análisis y visualización del comportamiento y evolución de los parámetros contemplados y relacionados con la diabetes a través del uso de gráficas, tablas, listas y otros elementos que permitan la visualización de los datos.
- e) Favorecer la escalabilidad del sistema, permitiendo tener en cuenta futuros aspectos y requisitos por parte de los especialistas.
- f) Fomentar el uso de nuevas tecnologías para el seguimiento y control eficiente de la diabetes.

- g) Conseguir que los usuarios que van a hacer uso de la aplicación no sean solo diabéticos tipo dos, si no usuarios generales, que valoren la monitorización de aspectos de su vida cotidiana.

El cumplimiento de los objetivos anteriores, posibilitan el estudio continuado del comportamiento de la enfermedad y la tendencia de los niveles de glucosa en el tiempo, así como del resto de factores relacionados con la diabetes, pudiendo ayudar a expertos a mejorar el tratamiento sobre el paciente, y a este a ejercer un autocontrol, mejorando su calidad de vida.

La Ilustración 1, muestra el diagrama general del sistema y las interacciones entre componentes y actores. De modo que el sistema se compondrá de un espacio de almacenamiento propio en la nube (DiMo Store) dónde se almacenan todos los datos relevantes del paciente diabético; una librería de servicios para facilitar la comunicación entre los espacios en la nube, DiMo REST API y la aplicación móvil correspondiente, capaz de obtener datos de dispositivos externos, así como de los sensores del propio Smartphone que favorecen el autocontrol de la enfermedad. DiMo REST API y DiMo Store son herramientas que ya se encuentran activas, y que no será necesario comenzar a desarrollar desde cero, si no que simplemente se adaptarán los datos y algunos de los *endpoints* según las necesidades del proyecto y de la aplicación Android.

De forma general, el proyecto se ha denominado con el acrónimo DiMo (Diabetes Monitoring).

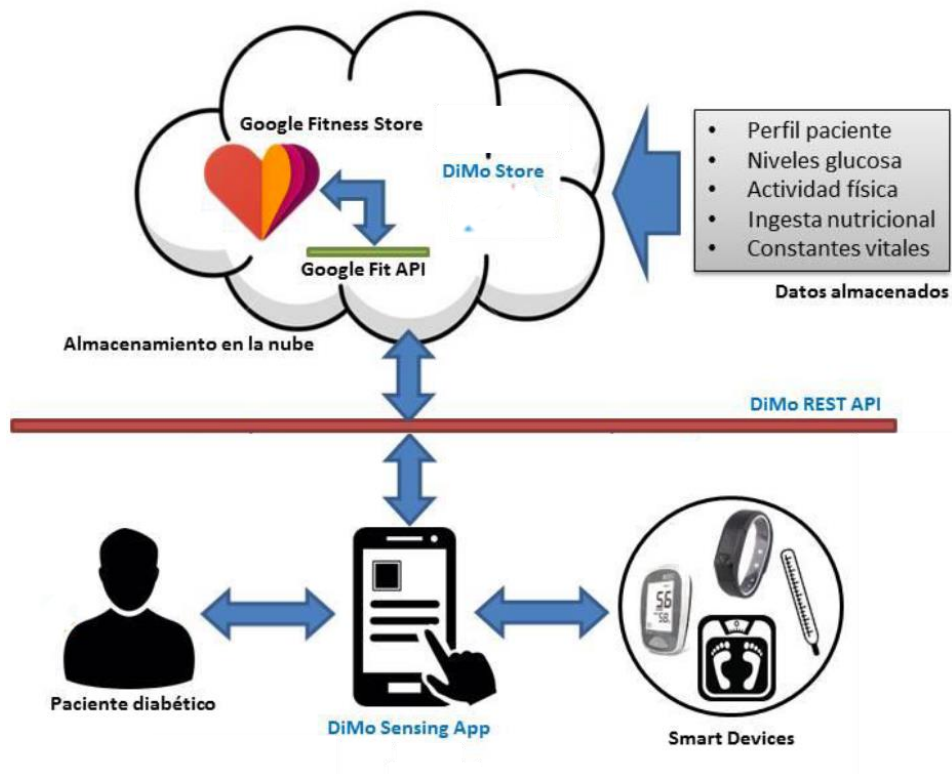


Ilustración 1 Diagrama General del sistema

2.2. Ejemplo de usuarios y escenarios

En este apartado se van a describir dos escenarios distintos en el uso de la aplicación, donde, dadas las características y el modo de vida de los usuarios, van a encontrar en DiMo una herramienta realmente útil en su día a día.

2.2.1. Ejemplo 1

Nombre: Isabel

Edad: 70 años

Profesión: Pensionista

Descripción de la persona: Isabel, madre de 5 hijos, abuela de 7 y viuda que vive en Puertollano (Ciudad Real). Desde hace más de 25 años, se le detectó una diabetes tipo 2, lo que le implica tener un gran cuidado en su alimentación, llevando un continuo registro de los alimentos ingeridos, así como

un control de glucosa en sangre, esfuerzos físicos y periódicas revisiones médicas.

En su día a día, suele hacer la compra por las mañanas antes de ir al centro de ocio de mayores, donde realiza diversas actividades como juegos de mesa, bailes de salón, yoga, etc. Por las tardes, suele dar paseos de una hora con amigos y visitar a los familiares más cercanos. Poco antes de la hora de cenar, vuelve a casa y se suele acostar sobre las 23:30 aproximadamente.

En el ámbito tecnológico, Isabel tiene conocimientos básicos en el uso de smartphones y ordenadores personales. El uso principal que realiza con ellos, es consultar los sitios web de noticias, páginas sobre recetas de cocina, video llamadas y redes sociales como Facebook.

Descripción del escenario: Martes, Isabel se despierta sobre las 7:30 AM, realiza su aseo personal, desayuna controlando su glucosa en sangre antes de la ingesta de cualquier alimento y se inyecta una dosis de insulina para empezar a comer. Al finalizar esta comida, apunta en un cuaderno los alimentos y la cantidad ingerida para tener el seguimiento actualizado.

Después se dirige a hacer la compra y al centro de ocio. Vuelve a hacerse un análisis antes de la comida principal y vuelve a registrar los alimentos consumidos. Este proceso se volverá a repetir después de la comida, y antes y después de la cena.

2.2.2. Ejemplo 2

Nombre: Antonio

Edad: 28 años

Profesión: Ingeniero Químico

Descripción de la persona: Antonio, vive con sus padres y sus dos hermanos pequeños en Ciudad Real. Trabaja como ingeniero químico en un laboratorio de una compañía petrolera. Realiza mucho deporte, monitorizando

su actividad física siempre que realiza algún ejercicio, principalmente atletismo, y lleva un control rígido con su alimentación.

En su vida diaria, tiende a tener situaciones de estrés, ya que el trabajo ocupa la mayor parte de su tiempo, y hace del ejercicio físico su principal *hobbie*, en el que invierte la mayoría de su tiempo libre.

Tecnológicamente, es un enamorado de todas las novedades que estén dentro del ámbito del deporte. Suele llevar pulseras de monitorización de actividad física, y está al tanto de los lanzamientos de nuevos dispositivos. Además, es propietario de un blog donde comenta su evolución y da consejos sobre cómo llevar una vida sana y activa.

Descripción del escenario: Viernes, Antonio se despierta sobre las 6:30 AM, y se prepara para hacer *footing* durante media hora. Una vez finalizada la actividad, realiza su higiene personal y se prepara un desayuno voluminoso con alto contenido en hidratos de carbono y proteínas.

A las 8:30 AM entra a trabajar hasta las 18:30 de la tarde. Durante este periodo no realiza mucha actividad física, ya que la mayor parte del tiempo se encuentra sentado en una mesa de oficina. Sobre las 12:00 toma un pequeño almuerzo, y a las 15:00 ingiere la comida principal.

Sobre las 19:30, realiza una merienda pre-entreno y va al gimnasio de al lado de su casa, invirtiendo una hora donde realiza ejercicios de musculación y aeróbicos. Sobre las 21:30 toma una cena ligera y las 23:30 se dispone a dormir.

3. ENFOQUE Y MÉTODO SEGUIDO

La metodología a utilizar para el desarrollo de este proyecto es la llamada **Prototipado Evolutivo**. [8]

A la hora de crear un producto software, el desarrollador encuentra dificultades para ser preciso en la captura de los requisitos solicitados por el cliente. Una herramienta de gran utilidad para resolver este problema son los prototipos, que se define como una versión incompleta de un desarrollo software utilizada para establecer de una forma clara los requerimientos del cliente, evaluar las opciones de diseño y obtener soluciones para mejorar el producto.

La metodología del Prototipado Evolutivo se basa en la construcción de un prototipo inicial en un corto espacio de tiempo, teniendo en cuenta unos requisitos básicos del sistema y procediendo a su evaluación al mismo tiempo que se van recogiendo nuevos requisitos para el siguiente prototipo. Se iniciará con los requerimientos más claros y fundamentales y se irán añadiendo los de prioridad más baja conforme el sistema y el cliente lo vayan solicitando.

Esta metodología tiene las ventajas de reducir el tiempo de desarrollo, no tener un coste significativo y conseguir una gran involucración del usuario en el proceso, lo que implica que se puedan obtener con más precisión los requisitos finales del usuario y por lo tanto generar un mayor grado de satisfacción una vez está completada la aplicación.

A continuación se describen cada una de las etapas que conforman esta metodología y que tendrán que ser recorridas por el prototipo hasta la consecución del sistema final (Ilustración 2).

- a) Definición del objetivo y funcionalidad: Se recogen los requisitos iniciales del usuario o provenientes de la última versión del prototipo del ciclo anterior, y se define como se plasmarán en el prototipo a desarrollar.
- b) Desarrollo: Se procede a la construcción del prototipo basándose en los requisitos anteriormente recogidos.
- c) Evaluación: Se realizan una serie de pruebas para validar que se están cumpliendo todos los requerimientos definidos y para observar todos los errores o mejoras que se tendrán en cuenta en futuras versiones del prototipo.
- d) Entrega del prototipo final: Cuando una vez analizados los resultados obtenidos en la etapa anterior, se considera que se han cumplido todos los requerimientos definidos en cada una de las versiones del sistema, se dará por terminado el desarrollo y se entregará la aplicación en su versión final.

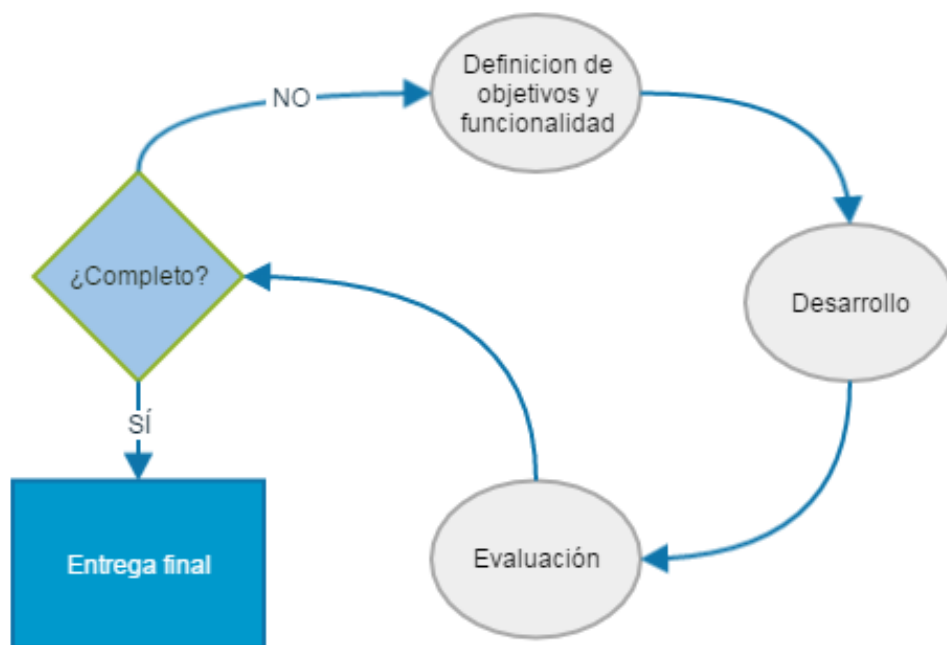


Ilustración 2 Diagrama de etapas del Prototipado Evolutivo

4. PLANIFICACIÓN DEL TRABAJO

4.1. Herramientas y tecnologías

A continuación, se van a listar las herramientas y tecnologías que se han utilizado o se utilizarán para realizar la aplicación y la documentación:

4.1.1. Herramientas de Software

- a) Android SDK, Android Studio: Kit de desarrollo oficial de aplicaciones para Android que contiene las herramientas necesarias para la creación de aplicaciones para el sistema operativo de Google. Android Studio será el entorno de desarrollo integrado utilizado para el desarrollo del cliente Android [13].
- b) Visual Paradigm 13.0 Community Edition: Herramienta utilizada para el desarrollo de aplicaciones a través del modelado UML, creada por Visual Paradigm International. Sus usos en este proyecto se enfocarán en la elaboración de diagramas de clases, diagramas de secuencia y casos de uso [14].
- c) Lenguaje Java 1.8: Lenguaje de programación orientado a objetos y multiplataforma desarrollado por Sun Microsystems, se utilizará para desarrollar la parte cliente del proyecto [15].
- d) XML: Lenguaje de marcas creado por World Wide Web Consortium [16] cuyo uso se centra en el almacenamiento e intercambio de datos.

- e) JSON: Formato de intercambio de datos a través de cadenas de texto que está formado por una serie desordenada de colecciones pares-valor. Utilizado para formar los mensajes que se intercambian el cliente y la API de la aplicación [17].
- f) Microsoft Word 2017: Procesador de textos del paquete Microsoft Office creado por la compañía Microsoft que se utilizará en la elaboración de la documentación del Trabajo Fin de Máster [18].
- g) GIMP: Software libre y gratuito de edición gráfica que se utilizará para la creación de los iconos, logotipos y fondos de la aplicación [19].
- h) Google APIs: Conjunto de comandos, funciones y protocolos que pueden ser utilizados para añadir funcionalidad a las aplicaciones. En este proyecto se hará uso del API de Google Fit para obtener y almacenar los datos del usuario [20].

4.1.2.Herramientas de Hardware

- a) Ordenador personal: Utilizado para el desarrollo completo del proyecto y de la documentación:

Modelo: MacBook Pro 2015

Sistema operativo: MacOS Sierra.

Procesador: Intel Core i7

RAM: 16GB.

- b) Smartphones: Utilizados para la realización de pruebas y evaluación de los requisitos.

Xiaomi Mi5s: Procesador Snapdragon 821, 3GB RAM.

BQ M5: Procesador Snapdragon 615, 3GB RAM

4.2. Planificación de las tareas

El autor de este proyecto, compagina su vida laboral con la elaboración de este trabajo fin de master. Debido a esta circunstancia, se observa, gracias a la tabla mostrada en la Ilustración 3, las horas que se han invertido en cada una de las semanas existentes desde el inicio a la finalización del mismo.

	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
11:00 12:00							
12:00 13:00							
16:00 17:00							
17:00 18:00							
18:00 19:00							
19:00 20:00							
19:00 20:00							
20:00 21:00							
21:00 22:00							

Ilustración 3 Planificación Semanal

Ilustración 4, se observa el tiempo invertido (en horas) para la elaboración de cada una de las tareas de la primera entrega del trabajo, PEC 1.

Teniendo en cuenta las futuras entregas, se han elaborado otras tablas que informan de la planificación a seguir hasta la consecución final del proyecto (Ilustración 5, Ilustración 6 e Ilustración 7).

Nombre de la tarea	Fecha de inicio	Fecha final	Duración (horas)
Selección e investigación de la temática	20/9/17	25/9/17	6
Desarrollo e investigación sobre el tema elegido	26/9/17	28/9/17	10
Elaborar requisitos del sistema	28/9/17	30/9/17	4
Elegir herramientas y tecnologías a utilizar	29/9/17	1/10/17	4
Documentación de PEC1	2/10/17	10/10/17	15
Entrega PEC1	11/10/17	11/10/17	1

Ilustración 4 Etapas PEC1

Nombre de la tarea	Fecha de inicio	Fecha final	Duración (horas)
Captura de requisitos y definición de usuarios del sistema	12/10/17	15/10/17	12
Análisis de los casos de uso	16/10/17	20/10/17	10
Elaboración de los diagramas de flujo	21/10/17	24/10/17	10
Construcción de prototipos	25/10/17	27/10/17	8
Evaluación de los prototipos	27/10/17	28/10/17	3
Documentación de PEC2	28/10/17	1/11/17	15
Entrega PEC2	1/11/17	1/11/17	1

Ilustración 5 Etapas PEC2

Nombre de la tarea	Fecha de inicio	Fecha final	Duración (horas)
Diseño iconografía e imágenes	2/11/17	7/11/17	10
Creación de pantallas de la aplicación	8/11/17	19/11/17	30
Implementación de la navegación entre componentes	20/11/17	26/11/17	25
Conexión con la API	27/11/17	1/11/17	20
Evaluación de etapas beta	2/11/17	5/12/17	8
Documentación de PEC3	6/10/17	12/12/17	15
Entrega PEC3	13/12/17	13/12/17	1

Ilustración 6 Etapas PEC3

Nombre de la tarea	Fecha de inicio	Fecha final	Duración (horas)
Documentar introducción y contexto de la aplicación	2/11/17	7/11/17	10
Explicar los objetivos conseguidos	8/11/17	19/11/17	30
Explicar las etapas del proceso de desarrollo	20/11/17	26/11/17	25
Destacar aspectos técnicos del desarrollo	27/11/17	1/11/17	20
Evaluación y Usabilidad del producto	2/11/17	5/12/17	8
Manual de usuario, manual de instalación y conclusiones finales	6/10/17	12/12/17	15
Retoques finales en la	13/12/17	30/12/17	30

implementación del código			
Presentación	25/12/17	03/01/18	10

Ilustración 7 Etapas PEC4

5. ETAPA INICIAL DEL DESARROLLO

En este apartado se va a describir cada una de las etapas que se han llevado a cabo en el proceso de desarrollo de la aplicación utilizando la metodología del prototipado evolutivo.

Los roles de los actores que componen los integrantes del proyecto son los siguientes:

- a) **Fernando López Tejeda:** Desarrollador del proyecto.

- b) **Director del Trabajo fin de Máster:** Cliente, encargado de realizar la evaluación de los prototipos y documentación, así como la corrección y la definición de nuevas propuestas de mejora para futuras versiones de la aplicación.

- c) **Usuarios testing:** Actores que no tienen por qué disponer de un perfil tecnológico, hacen uso de la aplicación en cada una de las etapas e informan al desarrollador de las sensaciones obtenidas al utilizarlas y de las posibles mejoras que se podrían implementar. Principalmente, este grupo está formado por personas cercanas al desarrollador: familiares, amigos, compañeros de trabajo, etc.

5.1. Captura de requisitos

Partimos de la base que el sistema a desarrollar consiste en una aplicación para el sistema operativo Android, con el principal objetivo de poder ser utilizada por la gran mayoría de usuarios, independientemente de que sus conocimientos tecnológicos no sean elevados. A continuación, se van a detallar los requisitos del sistema, categorizándolos en funcionales y no funcionales:

5.1.1. Requisitos funcionales

Identificador	Requisito Funcional 1 (RF1)
Nombre	Registro de usuarios
Descripción	El sistema deberá permitir registrarse a los usuarios de la aplicación a través de su email y una contraseña de la plataforma Google.

Identificador	Requisito Funcional 2 (RF2)
Nombre	Identificación de usuarios
Descripción	La aplicación deberá poder identificar a los usuarios que previamente se han registrado para utilizar sus servicios.

Identificador	Requisito Funcional 3 (RF3)
Nombre	Cerrar sesión
Descripción	El sistema permitir a los usuarios cerrar sesión en cualquier momento del uso de la aplicación.

Identificador	Requisito Funcional 4 (RF4)
Nombre	Menú desplegable
Descripción	El sistema dispondrá de un menú desplegable colocado en el margen izquierdo de la pantalla, que podrá ser desplegado desde cualquiera de las pantallas principales de la aplicación

Identificador	Requisito Funcional 5 (RF5)
Nombre	Pantalla Resumen
Descripción	El sistema deberá contener una pantalla de resumen, que se visualizará una vez iniciada la sesión del usuario y que contendrá información de los parámetros del usuario en los tres últimos días de uso de la aplicación.

Identificador	Requisito Funcional 6 (RF6)
Nombre	Botón flotante
Descripción	El usuario podrá insertar datos en la aplicación a través de un botón flotante, colocado en la parte inferior derecha, que se desplegará para poder elegir el tipo de dato a registrar.

Identificador	Requisito Funcional 7 (RF7)
Nombre	Menú Mi Perfil
Descripción	El sistema dispondrá de una sección desde la que se podrá gestionar la información personal del usuario.

Identificador	Requisito Funcional 8 (RF8)
Nombre	Menú Estadísticas
Descripción	El sistema dispondrá de una sección denominada "Estadísticas" desde la que se podrá filtrar por una serie de parámetros para crear gráficas, listas, diagramas de sectores, etc.

Identificador	Requisito Funcional 9 (RF9)
Nombre	Menú Citas Médicas
Descripción	El sistema dispondrá de una sección denominada "Citas médicas" desde la que se registrar alarmas para recibir información sobre las citas en un tiempo determinado.

Identificador	Requisito Funcional 10 (RF10)
Nombre	Inserción de datos
Descripción	El sistema debe permitir al usuario insertar datos relacionados con la diabetes, estatura, peso y nutrición del usuario.

Identificador	Requisito Funcional 11 (RF11)
Nombre	Monitorización física
Descripción	El sistema debe monitorizar la actividad física del usuario de forma automática y transparente.

5.1.2. Requisitos no funcionales

Identificador	Requisito No Funcional 1 (RNF1)
Nombre	Sistema operativo y dispositivos
Descripción	La aplicación se debe ejecutar en dispositivos <i>smartphone</i> que cuente con el sistema operativo Android de Google.

Identificador	Requisito No Funcional 2 (RNF2)
Nombre	Versión del sistema operativo
Descripción	La versión mínima que debe tener instalada cada dispositivo que vaya a utilizar la aplicación es Android 5.0 (Lollipop)

Identificador	Requisito No Funcional 3 (RNF3)
Nombre	Interfaz simple e intuitiva
Descripción	El sistema debe ser fácil de utilizar y muy intuitivo durante la interacción con las distintas pantallas de la aplicación.

Identificador	Requisito No Funcional 4 (RNF4)
Nombre	Resolución de pantalla
Descripción	La aplicación será compatible con diferentes resoluciones de pantalla.

Identificador	Requisito No Funcional 5 (RNF5)
Nombre	Tamaño de la aplicación
Descripción	La aplicación necesitará un máximo de 40 Megabytes para ser instalada con el fin de que el almacenamiento del terminal no sea un problema a la hora de poder ejecutar la aplicación.

Identificador	Requisito No Funcional 6 (RNF6)
Nombre	Pantallas en la aplicación
Descripción	Las pantallas principales que debe contener la aplicación son: inicio de sesión, resumen, mi perfil, estadísticas y citas médicas

Identificador	Requisito No Funcional 7 (RNF7)
Nombre	Parámetros mi perfil
Descripción	La sección mi perfil debe permitir gestionar los parámetros: Nombre, apellidos, e-mail, país, edad, fecha de nacimiento, género, peso, altura, debut diabético y rango ideal de glucosa.

Identificador	Requisito No Funcional 8 (RNF8)
Nombre	Parámetros diabetes
Descripción	El sistema debe permitir insertar los siguientes datos relacionados con la diabetes: Glucosa en sangre y dosis de insulina.

Identificador	Requisito No Funcional 9 (RNF9)
Nombre	Parámetros físicos
Descripción	El sistema debe permitir insertar los siguientes datos relacionados con la actividad física: Pasos, distancia, calorías, tipo de actividad y velocidad alcanzada.

Identificador	Requisito No Funcional 10 (RNF10)
Nombre	Subsecciones de sección estadísticas
Descripción	La sección de estadísticas deberá tener 3 subsecciones: Glucosa, actividad física, nutrición y una que relacione todas las anteriores.

Identificador	Requisito No Funcional 11 (RNF11)
Nombre	Citas médicas
Descripción	La alarma de las citas médica debe lanzar la notificación una hora antes de la fecha determinada. La información que debe registrar es: fecha, hora, título, descripción y lugar de la cita.

Identificador	Requisito No Funcional 12 (RNF12)
Nombre	Conexión
Descripción	El sistema deberá estar conectado a internet para poder hacer uso de la aplicación.

5.2. Navegación por la aplicación

En la Ilustración 8 y en la Ilustración 9 se representa la estructura y el árbol de navegación de la aplicación a desarrollar, creado por la aplicación online *draw.io* [22]. Se observan las principales pantallas por las que navegarán los usuarios y las acciones a realizar para llegar a cada una de ellas.

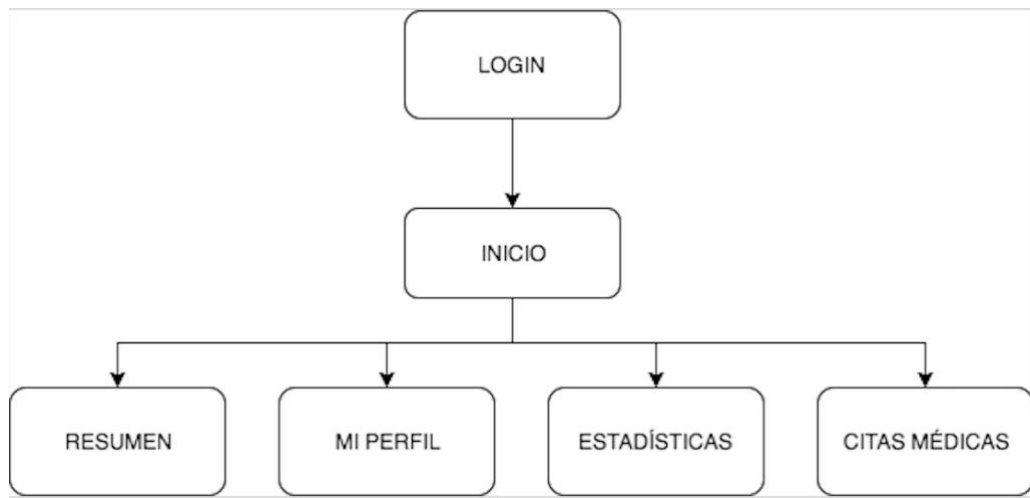


Ilustración 8 Estructura de la aplicación

Como se puede observar, al ejecutar la aplicación se mostrará una **pantalla de inicio** de sesión a través de una cuenta de Google. En ella se pedirá la autenticación para poder dirigirse a la pantalla principal que nos dará acceso a las demás ventanas disponibles.

En la **pantalla inicial** se podrá visualizar un resumen de la información del usuario, centrándose en los parámetros principales que se han registrado en los últimos tres días. Además, se dispondrá de un menú desplegable que permitirá navegar por las distintas secciones de la aplicación, y estará disponible en todas las pantallas secundarias, dando la opción de moverse de una sección a otra de una forma rápida, directa y sencilla.

Teniendo en cuenta al resumen o pantalla de inicio como la primera sección, la opción de Mi Perfil se establece como la segunda pantalla, donde

se gestionarán los datos del usuario identificado, que ayudarán a conocer al paciente y a generar las estadísticas de una forma más completa.

El apartado de **estadísticas** estará compuesto por numerosos tipos de gráficos y listas que, ayudados de filtros, permitirán tanto al usuario como a los expertos hacer un seguimiento del paciente de una forma precisa y visual.

El menú se compondrá en último lugar por la sección de **citas médicas**, donde se mostrarán la información de las diferentes citas que tiene registrado el usuario.

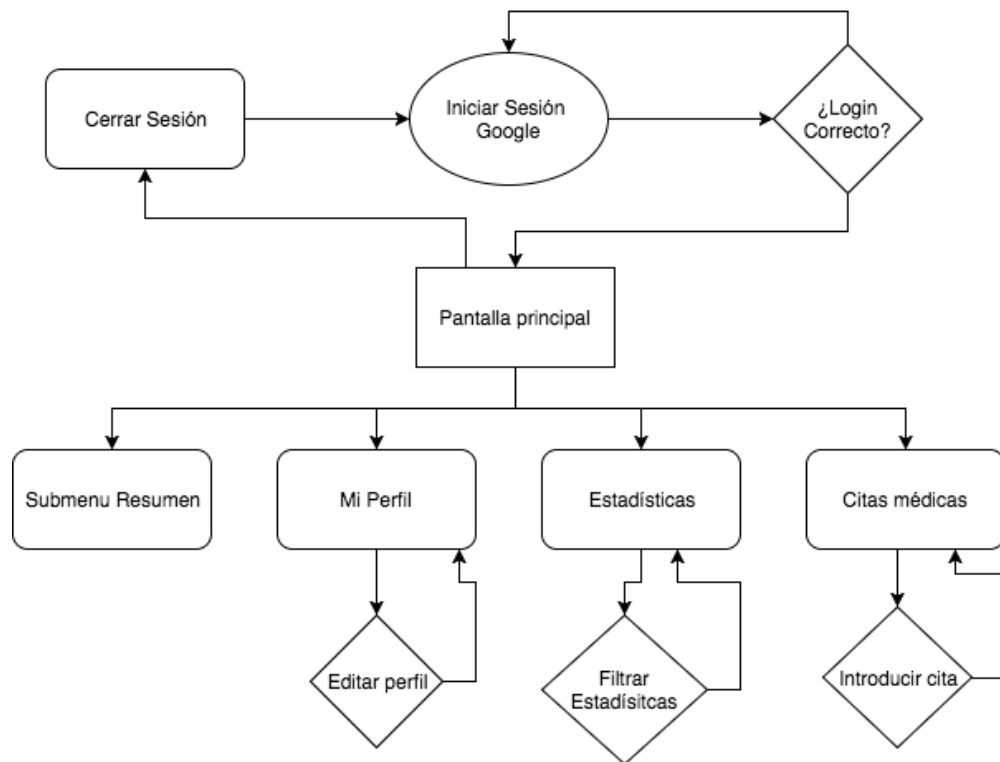


Ilustración 9 Árbol de navegación de la aplicación

5.1. Casos de uso

A continuación se van a describir algunos de los escenarios que se pueden dar utilizando la aplicación, describiendo, por medio de casos de uso, los detalles de cada uno de ellos.

En la Ilustración 10 se representa el sistema desde el punto de vista del cliente Android, donde el actor externo, denominado usuario, puede realizar una serie de acciones. Al inicio, deberá registrarse en la aplicación por medio de las credenciales de Google, que es la plataforma a utilizar en todo el ecosistema. Una vez realizado, el usuario podrá ver el resumen semanal que ha sido generado a través de los datos enviados por el servidor. Se utiliza la etiqueta *<<include>>* para indicar que será necesario estar identificado en el sistema para realizar dicha acción.

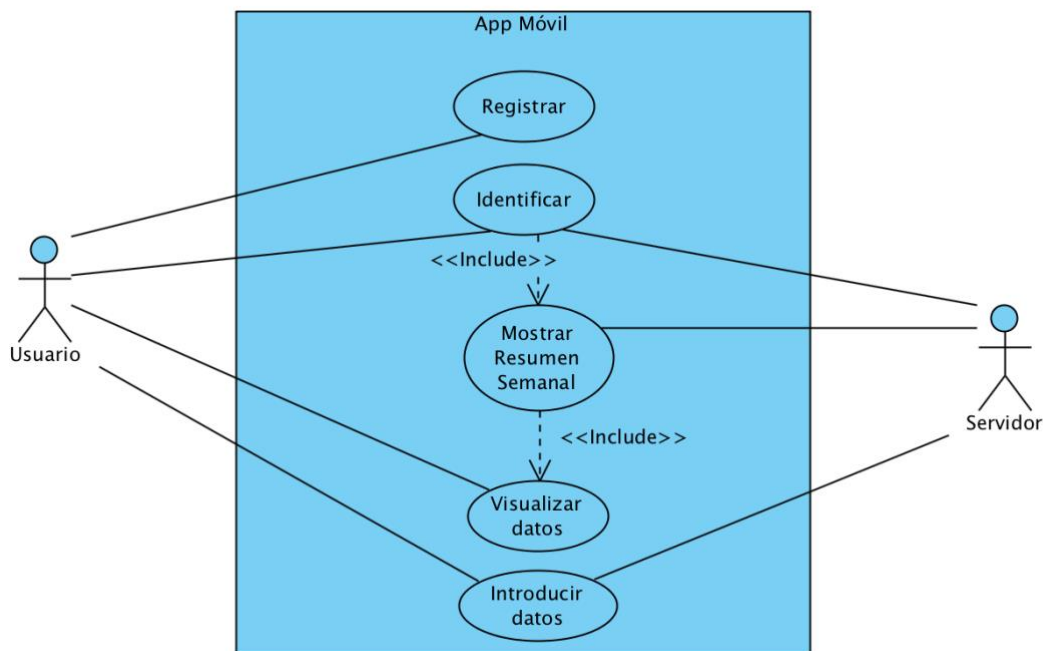


Ilustración 10 Diagrama de casos de uso general del cliente Android

Para llegar a la situación reflejada en la Ilustración 11 el usuario deberá haber pulsado en la pantalla principal el botón flotante, que permite la introducción de datos de forma manual. Una vez desplegado, seleccionará la opción de introducir nuevo dato de nutrición y añadirá el alimento determinado.

Cuando se registre la información, el actor seleccionará la sección de estadísticas del menú desplegable lateral y elegirá los filtros correspondientes a los parámetros de nutrición, y así poder visualizar un informe de todo lo ingerido en el periodo de tiempo determinado, mostrando la información en forma de gráfica, lista, etc.

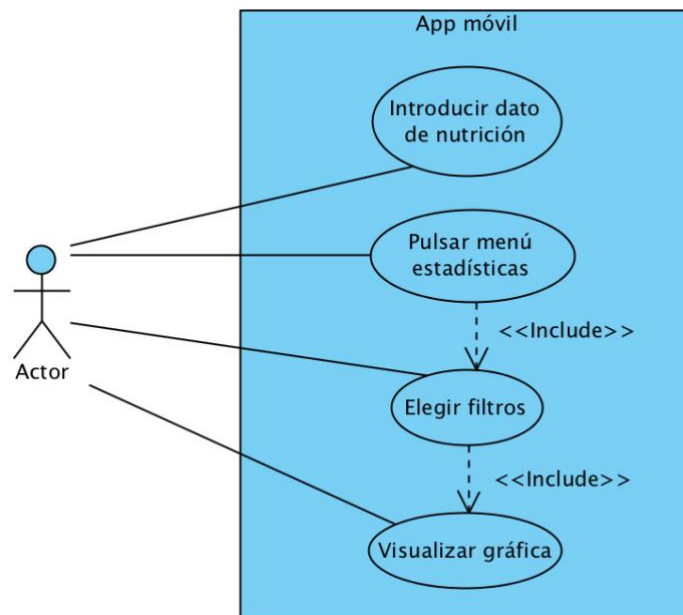


Ilustración 11 Diagrama de caso de uso de la sección estadísticas

En la Ilustración 12 se observa el escenario en el que el usuario recibirá una alarma, por parte de la aplicación, informando de la fecha y hora de la cita médica correspondiente. En primer lugar, el actor debe pulsar el botón de añadir cita acompañándola de una descripción que la detalle. Cuando llegue el momento en el que se configuro la alarma, la app enviará una notificación al usuario una hora antes del tiempo introducido previamente.

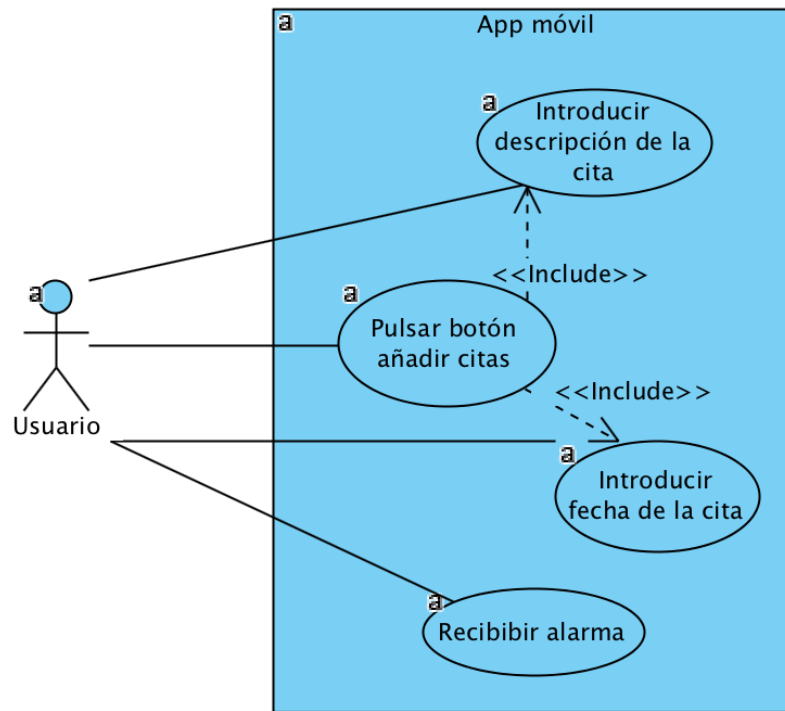


Ilustración 12 Diagrama de caso de uso de recibir alarma

5.2. Prototipo inicial

En las fases iniciales del desarrollo y diseño de la aplicación, una de las más importantes es la de construcción de los prototipos, ya que disminuye el tiempo de desarrollo, previene de fallos o de futuras modificaciones en la app, y permitir hacerse una idea global de cómo será el producto final, sin tener que emplear un esfuerzo muy significativo.

En primer lugar, se tuvieron en cuenta las características del proyecto a la hora de crear, mediante la herramienta Gimp [21], un **logo** descriptivo con la funcionalidad de la aplicación. Por ello, después de pensar diferentes colores e imágenes, se escogió la representación presente en la Ilustración 13 como el logotipo final. En él se puede observar como hay una mano en la parte superior de la que emana una representación de una gota de sangre. Con ello se quiere relacionar el logo con el acto de análisis de sangre que suelen realizar los diabéticos en su día a día. Además, en la gota, se observan unas líneas que representan una gráfica, haciendo alusión a los informes de los datos de los usuarios. Por último, aparece un logo de Android, informando de la plataforma para la que se ha desarrollado la aplicación.



Ilustración 13 Logo de la aplicación

Después de la creación del logotipo de DiMo, se van a diseñar las pantallas del prototipo inicial que se han llevado a cabo mediante el uso de la herramienta NinjaMock [23].

En la Ilustración 14 se observa la **pantalla de inicio** de sesión de DiMo a través de la cuenta de Google, que será la plataforma a utilizar para autenticar a los usuarios en la aplicación.



Ilustración 14 Pantalla de inicio de sesión

Una vez iniciada la sesión, aparece el escenario representado en la Ilustración 15, donde se puede observar el aspecto de la **pantalla resumen de la aplicación**, que contendrá, a base de elementos como gráficas, diagramas, tablas, etc., la información registrada en el sistema.

La aplicación dispondrá de un **menú desplegable** desde donde se podrá navegar por las diferentes secciones. Este elemento estará disponible en todas las pantallas, y se accederá a él a través del botón de “hamburguesa”

existente en la parte superior izquierda o deslizando desde el margen izquierdo hacia el centro de la pantalla.

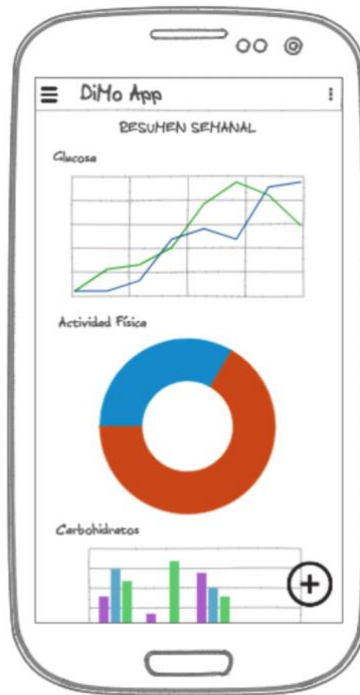


Ilustración 15 Pantalla de resumen



Ilustración 16 Menú desplegable

Otra de las pantallas destacables en la aplicación es la de **Mi Perfil**, cuyo diseño se puede observar en la Ilustración 17. En ella se gestionan los

datos del usuario identificado, que serán de gran utilidad a la hora de mostrar la información y realizar los cálculos para la elaboración de los informes y estadísticas.



Ilustración 17 Pantalla de perfil

La pantalla de **estadísticas** (Ilustración 18), es una de la más importantes de la aplicación, donde el usuario puede filtrar entre varios tipos de parámetros para que el sistema genere, de forma automática, diferentes gráficas, tablas, listas, etc. El objetivo de esta ventana consiste en realizar un seguimiento de la situación del usuario, de una forma representativa, para conseguir una más fácil evaluación de los expertos o de los mismos usuarios.



Ilustración 18 Pantalla de estadísticas

5.3. Evaluación

Al terminar cada ciclo de la metodología de desarrollo, se obtiene un prototipo provisional, que deberá ser evaluado de cara a supervisar si los requisitos capturados se están cumpliendo y para definir nuevos que impliquen mejoras y correcciones en posteriores versiones del prototipado.

En las etapas iniciales, los prototipos elaborados, tenían un carácter no funcional, y para su evaluación, se tuvo en cuenta principalmente a usuarios del entorno del desarrollador. Las correcciones y mejoras realizadas en esta fase se centran en aspectos de la interfaz gráfica: uso de iconos, colores, posición de los elementos, facilidad de uso, etc. También se llegó a la conclusión de que era conveniente la inclusión de un nuevo parámetro relacionado con la diabetes tipo 2: el pulso cardiaco, variable que tiene verdadera importancia tanto en el análisis de la enfermedad, como en la monitorización de la actividad física, y que no se había tenido en cuenta en las capturas de requisitos previas.

En etapas más avanzadas del desarrollo, donde los prototipos ya demostraban las funcionales de la aplicación, se puso mucho más énfasis en la fase de evaluación. En ella volvieron a participar personas cercanas al desarrollador, que contaban con diversos niveles de conocimientos tecnológicos y dos de ellos, diabéticos tipo dos.

Las pruebas de evaluación realizadas se basaban en dos tareas. La primera utilizaba la técnica “*Think Aloud*”, donde los usuarios evaluados realizaban una serie de tareas predefinidas. La segunda consistía en un cuestionario con preguntas sobre su perfil personal y la usabilidad de la aplicación. En el [Anexo 2](#) se pueden ver detalladas los enunciados de cada una de las pruebas.

Durante el proceso “*Think Aloud*”, el usuario iba realizando las tareas, comentando en voz alta cada una de las acciones y pensamientos que estaba realizando, justificándolos para que el evaluador, en este caso el desarrollador, tomase nota sobre las dificultades, éxitos, o mejoras a realizar en el futuro.

La siguiente tarea a realizar por parte de los usuarios consistía en un cuestionario, que deberían cumplimentar una vez utilizada la aplicación durante un periodo considerable, y así, poder evaluar la usabilidad de la aplicación, el grado de satisfacción y tener un registro del perfil de cada uno de los usuarios.

6. Implementación

6.1. Estructura del proyecto

En este apartado se va a describir cómo se encuentra organizado el proyecto en el lado del cliente Android. Para estructurar el proyecto, se distribuyen los archivos en carpetas y paquetes, como se puede apreciar en la Ilustración 19, y se guardan en cada una de ellas según la función que desempeñen.

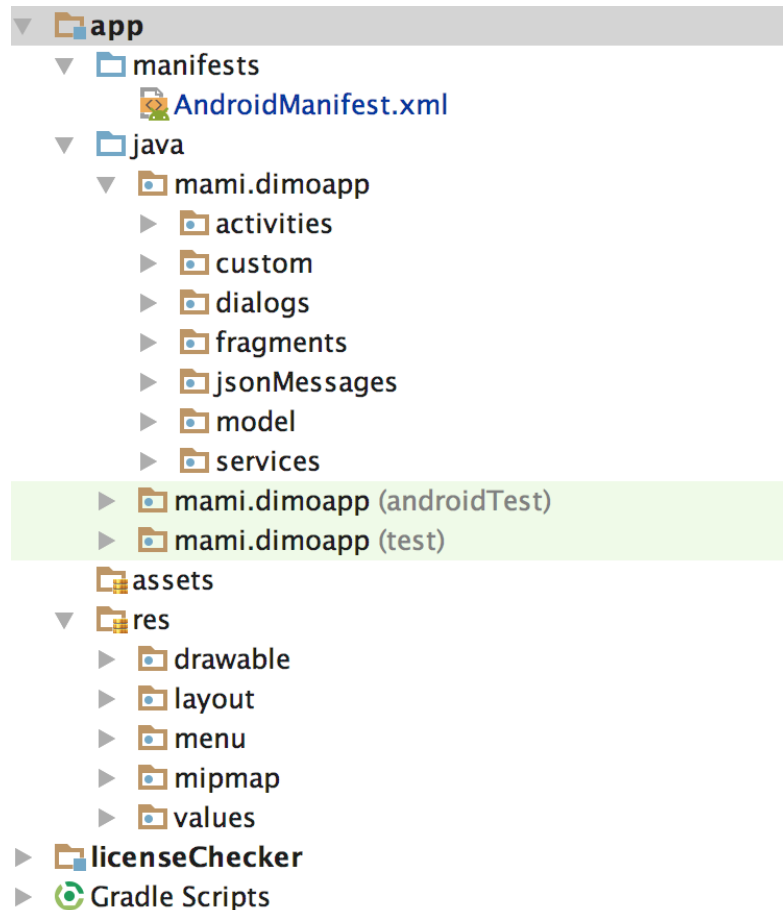


Ilustración 19 Estructura carpetas del cliente Android

- a) Android Manifest: Archivo XML que realiza una descripción de la aplicación, indica los permisos y servicios que se pueden implementar en ella, e informa del paquete Java y de la versión mínima del sistema operativo Android a utilizar, en este caso la 5.0, Lollipop (API v21), ya que con ello se alcanzaría el 80% del mercado Android actual. [24]
- b) Activities: Almacena las clases Java que extienden de la clase *Activity*, cuya función principal es la implementación de las funcionalidades que se realizan en las *activities* de la aplicación, que en Android se entiende como las pantallas o ventanas que se muestran en el dispositivo.
- c) Custom: En esta carpeta se archivan las clases que han sido creadas para modificar funcionalidades de las librerías utilizadas: Dar formato a los valores de los ejes en las gráficas, mostrar número de notificaciones en la parte inferior de los iconos (Badge), convertidores de fechas, peticiones http customizadas, etc.
- d) Dialogs: Paquete donde se alojan las clases que lanzan los diferentes diálogos que se muestran en la aplicación.
- e) Fragments: Archiva las clases que contienen todos los componentes *fragments* a utilizar.
- f) Model: Alberga las clases de dominio: Perfil del usuario, puntos de nutrición, insulina, glucosa, actividad física, etc.
- g) Services: Contiene los procesos que se ejecutarán en segundo plano.
- h) Res: Carpeta que contiene los recursos que utiliza la aplicación ordenados por funcionalidad según en qué subcarpeta se encuentren. Se describen en la tabla siguiente:

Recursos	Descripción
Drawable	Contiene imágenes con distintos formatos y para distintos tamaños de pantalla y descriptores de imágenes XML.
Layout	Contiene los ficheros XML donde se configuran las interfaces de usuario de las distintas pantallas de la aplicación.
Menu	Ficheros donde se configuran los diferentes menús de la aplicación.
Mipmap	Contiene las distintas resoluciones para el logotipo de la app.
Values	Contiene ficheros XML donde se describen distintos tipos de estilos, colores o valores de <i>String</i> .

6.2. Funcionalidades, herramientas y recursos

En esta sección se van a comentar los puntos que han sido claves a la hora de desarrollar esta aplicación, describiendo los pasos seguidos para completar sus funcionalidades.

6.2.1. Google Sign-In

Esta aplicación utiliza Google como sistema de autenticación de usuarios. Se ha elegido este proceso debido a que Google es una plataforma muy utilizada por el público general, lo que implica que beneficie tanto al usuario, ya que no tiene que recordar nuevas contraseñas, y al desarrollador por no tener que crear un nuevo sistema de cuentas y autenticación. Además, es a través de ella donde se recoge la información del perfil del usuario: nombre, fecha de nacimiento, foto de perfil, género, etc.

Para integrar este servicio en la aplicación se deben seguir los siguientes pasos:

1. Entrar en la **consola de desarrolladores de Google** [25], seleccionar la plataforma *Android App* e introducir el nombre del paquete de la app y un nombre de aplicación (Ilustración 20).

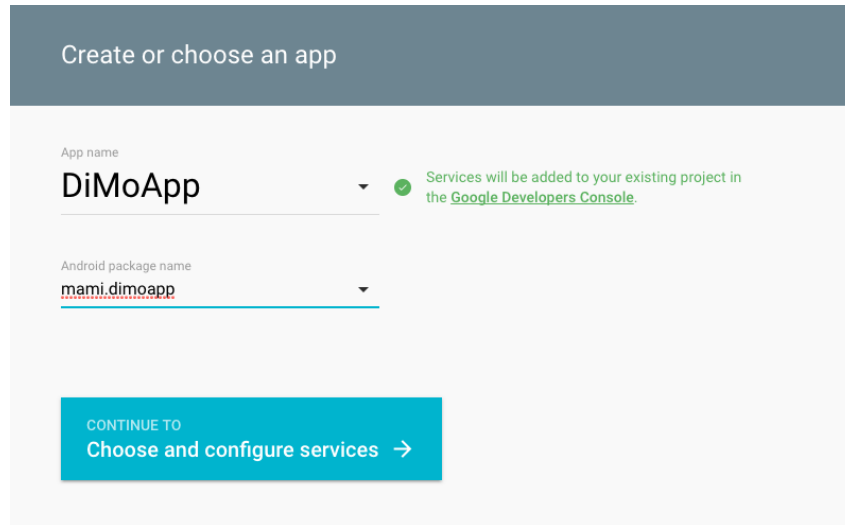


Ilustración 20 Configuración Google Sign-In

2. Seleccionar el servicio que se va a configurar en la app, en este caso Google Sign-In. Para ello nos pedirá la **clave SHA-1** de la aplicación, una función encriptada que utiliza Google para controlar el acceso y evitar malas prácticas de sus servicios. La forma de obtener dicha función es ejecutando el comando mostrado a continuación, donde la variable *USER* será la ubicación del fichero *debug.keystore* en el equipo.

Linux y OS X:

```
keytool -list -v -keystore .android/debug.keystore -alias androiddebugkey -storepass android -keypass android
```

Windows:

```
keytool -list -v -keystore "C:\USER\.android\debug.keystore" -alias androiddebugkey -storepass android -keypass android
```

3. Una vez realizado, se descargará el fichero ***google-services.json***, que contiene la información necesaria para configurar la app y asociarla al proyecto que se está desarrollando. La ubicación de este archivo debe ser en la carpeta raíz *app*.

4. Se deberán añadir las dependencias de **google play services** y **google play auth** en los ficheros **.gradle** del proyecto
5. Crear la interfaz de usuario donde el usuario pueda interactuar para proceder a llamada de autenticación de la app por medio de Google.
6. En términos de código, la llamada consistiría en invocar a la función **getSignInIntent()** existente en la clase *SignInActivity.java* del proyecto que se puede observar en la Ilustración 21.

```
// [START signIn]
private void signIn() {
    Intent signInIntent = Auth.GoogleSignInApi.getSignInIntent(mGoogleApiClient);
    startActivityForResult(signInIntent, RC_SIGN_IN);
}
// [END signIn]
```

Ilustración 21 Función de Autenticación Google

7. Una vez lanzado el **intent** e identificado el usuario, se usará la función **onActivityResult** para recibir la respuesta de la acción, y así poder realizar una acción determinada teniendo en cuenta si la identificación se ha realizado con éxito o no.

6.2.2. MPAndroid Chart

MPAndroid Chart es una librería utilizada para generar diferentes tipos de gráficas en la aplicación: diagrama de líneas, de barras, de sectores, etc. A continuación, se va describir un ejemplo de un diagrama de líneas existente en el desarrollo de este proyecto:

1. Añadir las dependencias de la librería en los archivos **.gradle** del proyecto (**com.github.PhilJay:MPAndroidChart:v3.0.1**).
2. Recoger los datos a representar en una lista que contiene objetos de la clase **Entry**.

3. Crear una instancia de un objeto **LineChart**, y una vez obtenidos los datos, se procede a cambiar el aspecto del gráfico para modificar los parámetros definidos por defecto. Se cambiará las líneas a color rojo, marcado los puntos también con ese color, se indicará el valor que aparecerá en la leyenda, el tipo de zoom, tamaños de los textos, duración de la animación, etc. (Ilustración 22)

```
chart.getDescription().setEnabled(false);
//chart.setVisibleXRangeMaximum(5);
chart.getAxisRight().setEnabled(false);
chart.setPinchZoom(true);

LineDataSet dataSet = new LineDataSet(entries, "Glucosa");
dataSet.setMode(LineDataSet.Mode.CUBIC_BEZIER);
dataSet.setColor(Color.RED);
dataSet.setCircleColor(Color.RED);
dataSet.setCircleRadius(5f);
dataSet.setValueTextSize(12f);
dataSet.setLineWidth(3);
dataSet.setAxisDependency(YAxis.AxisDependency.LEFT);
dataSet.setHighlightEnabled(true);

List<ILineDataSet> dataSets = new ArrayList<>();
dataSets.add(dataSet);

LineData data = new LineData(dataSets);
chart.setData(data);
chart.animateY(3000);
```

Ilustración 22 Configuración del gráfico

4. En el archivo **XML** donde se define la interfaz declarar una instancia de **LineChart** para que se puede visualizar en el dispositivo.

```
<com.github.mikephil.charting.charts.LineChart
    android:id="@+id/chartGlucose"
    android:layout_width="match_parent"
    android:layout_height="300dp"
    android:layout_marginTop="10dp"/>
```

Ilustración 23 Código XML LineChart

6.2.3.Android Volley

Es una librería desarrollada por Google utilizada en el proyecto para enviar peticiones **HTTP**. Su implementación resulta sencilla para el desarrollador, ya que el manejo de hilos se realiza de forma transparente e independiente del hilo principal.

Un ejemplo de utilización de esta librería sería cuando el usuario se realiza un análisis de la glucosa en sangre y registra ese dato a través del formulario existente en la pantalla de introducción de datos de la aplicación. Para realizar la petición **GET**, que recupera los datos, se requiere una **URL** del servidor, un periodo de tiempo en formato milisegundos y tomar las **cookies** de sesión recogidas al iniciar sesión cuando se ejecuta por primera vez la app. **(*Los problemas encontrados en este proyecto a la hora de recoger estas cookies son comentados en la sección 6.2.6. de este documento)**

En la Ilustración 24, se puede observar la definición de la URL y la toma de las *cookies* para su posterior inclusión en la petición **HTTP**. En este caso se implementa una petición ligeramente modificada por el desarrollador, donde sí se realiza con éxito, se recibe como respuesta un objeto **JSON**, que contendrá la información sobre el histórico de los datos de glucosa en sangre registrados por el usuario en un periodo de tiempo determinado.

Para lanzar la petición, se requiere ejecutar la función *Volley.newRequestQueue()*, la cual se encarga de encolar la llamada, y ya será el sistema el que realice el proceso cuando tenga los recursos necesarios (Ilustración 25).

```

//Server URL
String url = "http://mamilab.esi.uc1m.es:5050/api/v2.0/glucose?start="+start*1000000+"&end="+end*1000000;

//Create the list of the cookies, i.e. key=value
List<String> cookies = new ArrayList<>();
cookies.add(Store.get().getCookie());
datos = new ArrayList<>();

CustomRequest customRequest = new CustomRequest(Request.Method.GET, url,null,
    new Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject jsonArray) {
            // Display the first 500 characters of the response string.
            Log.i("BIEN","Response is: "+ jsonArray.toString());
        }
    }
);

```

Ilustración 24 Petición HTTP con Volley

```

customRequest.setCookies(cookies);
// and finally add the request to the queue
Volley.newRequestQueue(getContext()).add(customRequest);

```

Ilustración 25 Ejecutar petición Volley

6.2.4. Material Design

A partir de la versión 5.0 Lollipop de Android (API v21), Google lanzó una guía de diseño denominada Material Design [26], que disponía de nuevos componentes y funcionalidades que afectaban directamente a la interfaz gráfica.

Algunos de los nuevos añadidos que se han utilizado en el desarrollo de esta aplicación son:

- a) *CardView* y *RecyclerView*: Son dos nuevos widgets, utilizados en el proyecto para la creación de listas que contiene información sobre las ingestas de alimentos, dosis de insulina, análisis de glucosa en sangre, etc. (Ilustración 26)
- b) Elevación en las vistas: Es una propiedad que tienen muchos de los elementos de la interfaz gráfica, que da la impresión de profundidad a las vistas, ya que el eje Z de las vistas esta elevado, proyectando una sombra que produce esa sensación.

c) Formulario: Elementos como *EditText*, *TextView*, *Spinner* y botones han sido modificados y tienen un aspecto más minimalista con esta guía de diseño.

d) *NavigationView*: El menú desplegable de la aplicación tiene un aspecto Material Design que se puede observar en la Ilustración 27.



Ilustración 26 CardView y RecyclerView

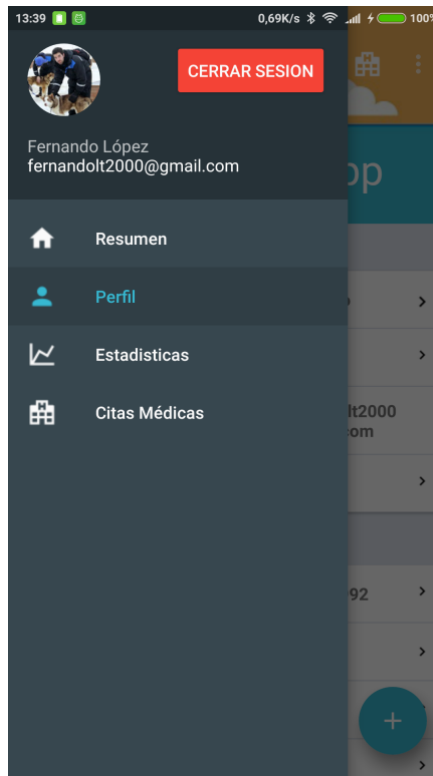


Ilustración 27 NavigationView

6.2.5. Google Fit – Monitorización Actividad Física

Para monitorizar la actividad física en la aplicación se ha utilizado la plataforma Google Fit. A continuación se va a describir cómo se ha implementado en el proyecto, describiendo los pasos a seguir y comentando un ejemplo de cálculo de pasos.

1. Habilitar la API de Google Fit en la consola de desarrolladores de Google [25].
2. Crear un id de cliente Android de OAuth para el control de peticiones, utilizando la clave SHA-1 conseguida de la misma forma que en el apartado anterior, integración con Google Sign-in.
3. Añadir las dependencias de la librería al archivo `.gradle`. ('com.google.android.gms:play-services:11.6.0')

4. Crear una clase Java que extienda de la clase *Service* y que implemente *OnDataPointListener*, *ConnectionCallback* y *OnConnectionFailedListener* de *GoogleApiClient*.
5. Iniciar la instancia *GoogleApiClient* añadiendo el *scope de Fitness Sensor* y configurando los *callbacks* (Ilustración 28).

```
mApiClient = new GoogleApiClient.Builder(this)
    .addApi(Fitness.SENSORS_API)
    .addScope(new Scope(Scopes.FITNESS_ACTIVITY_READ_WRITE))
    .addConnectionCallbacks(this)
    .addOnConnectionFailedListener(this)
    .build();

mApiClient.connect();
```

Ilustración 28 Conectar GoogleApiClient

6. Una vez conectado, el dispositivo comenzará a monitorizar los pasos, que se podrá registrar a través de las llamadas al método *OnDataPoint()*, cuya implementación se puede observar en la Ilustración 29.

```
@Override
public void onDataPoint(DataPoint dataPoint) {
    for( final Field field : dataPoint.getDataType().getFields() ) {
        final Value value = dataPoint.getValue( field );
        Log.i("pasos","Field: " + field.getName() + " Value: " + value);
        //Get values from type Value
        String add = "Value: "+value;
        String [] parts = add.split("Value: ");
        String stepsPart = parts[1];
        int steps = Integer.parseInt(stepsPart);
        Store.get().setSteps(Store.get().getSteps()+ steps);
        Log.i( "StepsService", "Pasos: "+Store.get().getSteps());
    }
}
```

Ilustración 29 Método OnDataPoint

6.2.6. Problemas en la autenticación con API REST DiMo

DiMo App utiliza el API REST DiMo para realizar las **peticiones HTTP** a sus servidores. El modo de autenticación de la aplicación se basaba en utilizar un elemento **WebView** incrustado en un *activity* de la aplicación, y así, registrar al usuario con los datos de la cuenta de Google a través de un formulario web, que utilizaba el protocolo **OAuth2**. Una vez identificado con éxito, la app guardaba la **cookie de sesión** para incluirla en todas las peticiones HTTP que realizaba contra la API, con la intención de no tener que identificarse de nuevo en cada una de las llamadas.

A partir de **finales de 2017**, Google ha comenzado a no permitir elementos *WebViews* incrustados en la aplicación a la hora de realizar peticiones OAuth a servidores propios. Como alternativas a implementar para solventar este inconveniente, propone el uso de **Chrome Custom Tabs**, que permiten mostrar contenido web en otra ventana sin tener que integrar en la App su propio navegador web, y dando la sensación de encontrarse en la propia aplicación.

El problema que se encuentra el desarrollador, es que estas *Custom Tabs*, no permiten a la aplicación gestionar esas *cookies*, ya que depende de la App *Google Chrome*, y no es posible recuperarlas a través de una aplicación externa por motivos de seguridad.

Dado esta situación, convendría realizar una modificación del API, que permita el uso de *tokens*, tal y como recomienda Google en su web para desarrolladores [27]. El no tener en cuenta este problema en la planificación del proyecto y el desconocimiento por parte del desarrollador del lenguaje *Python*, que es en el que está desarrollado esta API, ha provocado que se haya tenido que proceder a tomar una alternativa provisional hasta que se realice la modificación del API REST en un futuro inmediato.

En el manual de usuario adjunto en el [Anexo 1](#) de este documento se explica el procedimiento a seguir para la ejecución de la aplicación.

7. Conclusiones

En este capítulo se analizan los objetivos planteados antes de comenzar el desarrollo del proyecto. Además, se expondrán una serie de propuestas de mejora que se podrían implementar en una versión futura de la aplicación que añada nuevas funcionalidades y mejore la experiencia de usuario.

El objetivo principal de este proyecto se basaba en conseguir desarrollar una aplicación móvil para sistemas operativos Android. Para ello, se ha tenido que hacer frente a todas las etapas del ciclo de desarrollo de software, empleando los conocimientos adquiridos en las asignaturas impartidas en el máster.

Otro de los fines principales de este desarrollo consistía en conseguir monitorizar los aspectos relacionados con la enfermedad de la diabetes. Para ello, se ha realizado un análisis completo de todos los parámetros relacionados, teniendo en cuenta sus características a la hora de representarlos.

Los pacientes y/o expertos que utilicen la aplicación han conseguido tener un mayor control de la evolución de la enfermedad, teniendo un histórico detallado de toda actividad, que permite ser más consciente de cualquier síntoma que pueda aparecer, facilitando el seguimiento de la enfermedad y, por lo tanto, haciendo su día a día más cómodo.

Además de usuarios diabéticos, se ha podido observar como otras personas ajenas a la enfermedad pueden ver utilidad en la aplicación, ya que la monitorización de aspectos como la nutrición o la actividad física, resultan muy interesantes para una considerable parte de la sociedad actual.

7.1. Propuestas de mejora

DiMo es una aplicación que continua en etapa de desarrollo. El desarrollador es consciente de la potencia que puede llegar a tener y por ello se han numerado una serie de correcciones y mejoras que completarían mucho más el sistema.

- a) Utilización de dispositivos externos, como glucómetros bluetooth, que permitan registrar datos del usuario de una forma más automática y directa.
- b) Mejora de la autenticación con la API y el guardado de *cookies* de sesión para que toda identificación sea transparente al usuario final.
- c) Incluir más elementos de animación durante la navegación por la aplicación, de forma que se produzca un mayor dinamismo en la pantalla. También sería interesante implementar “un toque” de gamificación al producto que resulte más atractivo al usuario.
- d) Internacionalizar la aplicación, dando soporte en varios idiomas para conseguir una mayor cuota de mercado. Para añadir un nuevo idioma, se deberá crear un nuevo archivo de tipo *values* similar al ya existente *strings.xml*, pero con los valores en el nuevo idioma a configurar. Al iniciar la aplicación, el sistema se encargará de elegir el archivo a tener en cuenta dependiendo del lenguaje predeterminado que tenga el dispositivo.
- e) Despliegue de la aplicación en un mercado de aplicaciones, como puede ser Google Play Store, para conseguir una difusión mucho mayor del producto.
- f) Creación de una App para otras plataformas, véase iOS de Apple.

Anexo 1 Manual de Usuario

1. Guía de instalación y puesta en marcha

Para poder utilizar la aplicación DiMo se debe tener en posesión el archivo con la extensión *.apk* "DiMo.apk".

Una vez localizado, se deberá almacenar en un dispositivo Android con una versión mayor a la 5.0 Lollipop. Acto seguido, se ejecutará un administrador de archivos con el que podamos navegar hasta la ubicación del archivo y así llegar a ejecutarlo.

Una vez lanzado aparecerá una imagen semejante a la mostrada en la Ilustración 30, donde al pulsar en instalar, comenzará el proceso de instalación de la app y se podrá comenzar a utilizarla.

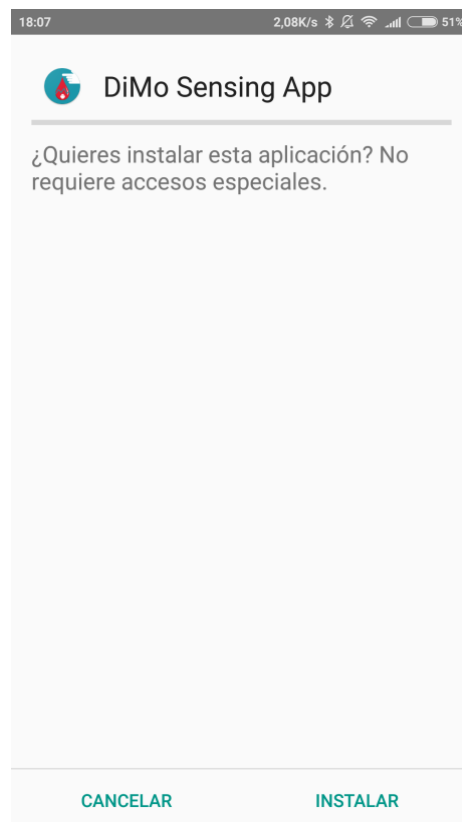


Ilustración 30 Instalación de la App

Debido a los problemas encontrados la hora de autentificar por medio de *WebViews* comentado en el apartado 6.2.6, hay que realizar un paso más en el proceso de autenticación, que será transparente para el usuario en futuras versiones de la aplicación. Estos serán los pasos a seguir:

1. En primer lugar, habrá que ejecutar desde el navegador del PC la siguiente dirección:

<http://mamilab.esi.uclm.es:5050/login>

2. Una vez ejecutada, sí es la primera vez que accedemos, aparecerá la ventana mostrada en la Ilustración 31, se deberá pulsar el botón “Login” y se accederá a través de la cuenta de Google.



Ilustración 31 Login a través de DiMo App

3. Se mostrará la aplicación Web del ecosistema DiMo. El siguiente paso a realizar consiste en conseguir la cookie de sesión del usuario. Para ello se deberá pulsar el botón derecho del ratón y hacer *click* en inspeccionar. Nos dirigimos a la pestaña de *Network* y pulsamos en actualizar la página, observaremos diversas llamadas, recogemos una de ellas, por ejemplo, *login*, y se recoge el valor de la *cookie*, cuya selección se puede observar en la Ilustración 32. Este valor lo deberemos guardar porque será necesario pegarlo al iniciar la App.

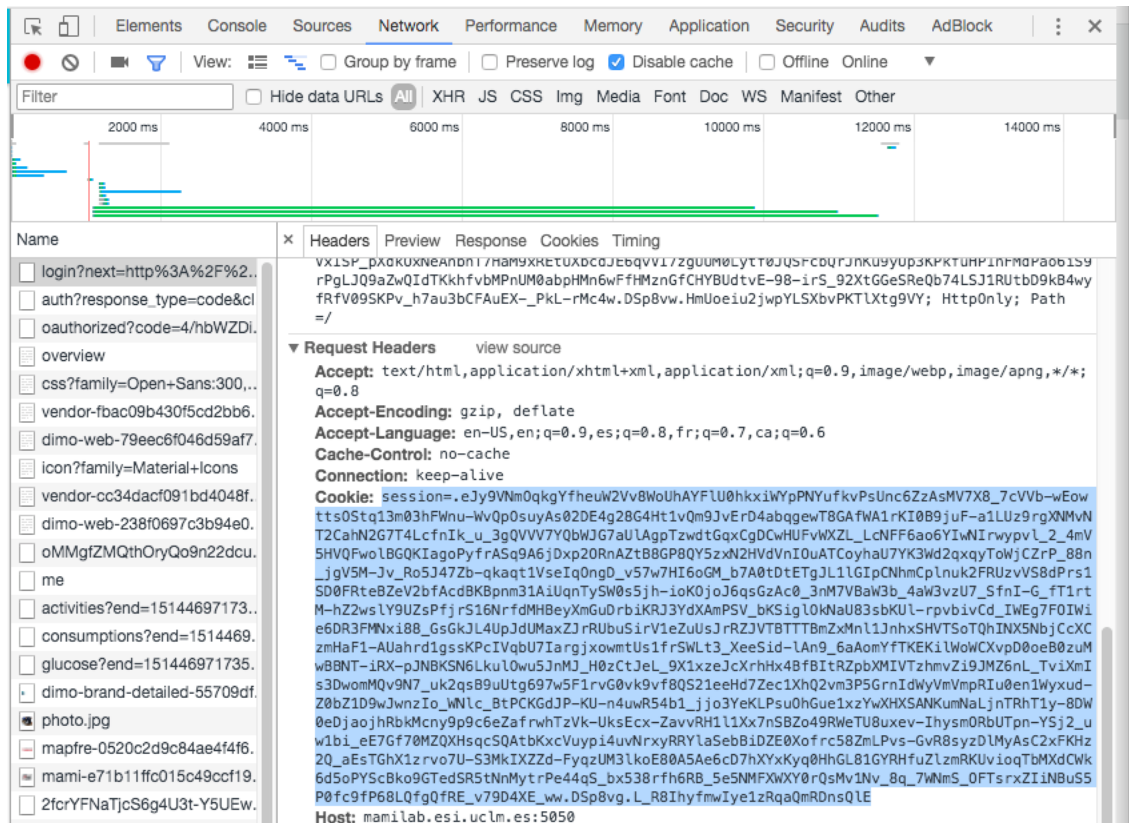


Ilustración 32 Cookie de sesión

4. Ejecutamos la aplicación instalada y procedemos a iniciar sesión con sesión con nuestra cuenta de Google pulsando al botón de “Iniciar sesión”. El siguiente paso lógico sería la navegación hasta la pantalla inicial de la App, pero en este caso habrá que realizar un paso más.
5. Pegamos la cookie de sesión copiada en el navegador en la ventana que nos aparece en la Ilustración 33 y pulsamos aceptar. La aplicación se lanzará con normalidad hasta que caduque esta cookie de sesión.

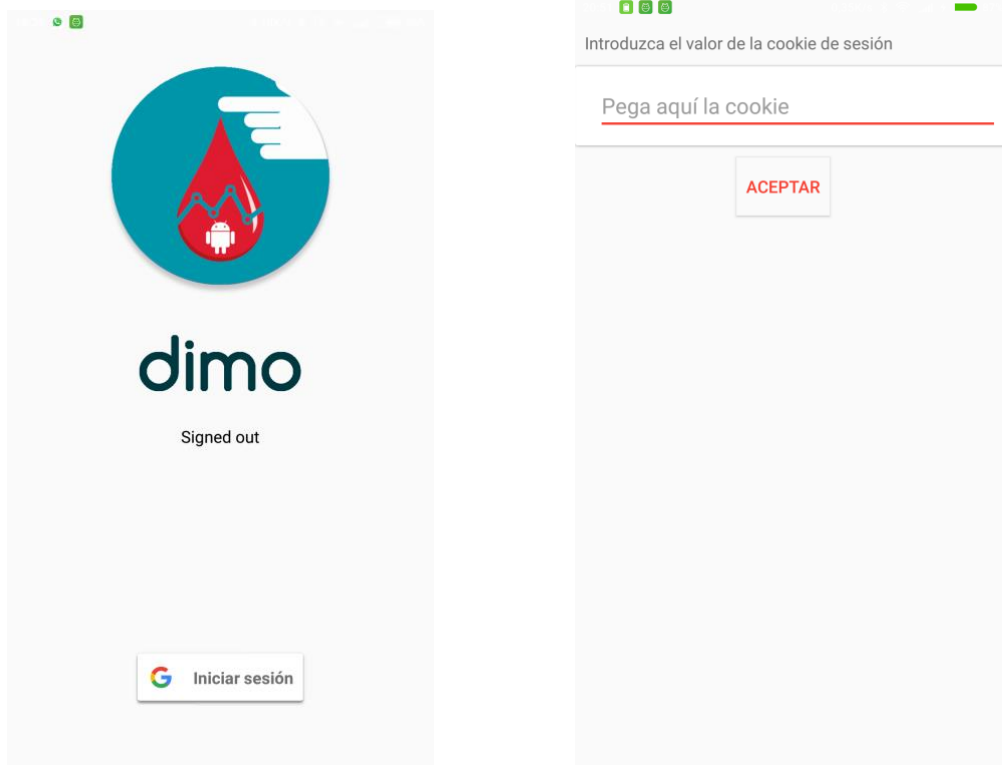


Ilustración 33 DiMo Inicio de sesión

2. Guía de uso de la App

Una vez ejecutada la aplicación y realizado todo el proceso anterior, nos encontraremos en la pantalla inicial o de resumen. En primer lugar, deberemos saber cómo **navegar por el sistema**, para ello se utiliza el **menú desplegable**, posicionado en el lado izquierdo de la pantalla, al que se puede acceder a través del “botón de hamburguesa” o deslizando el dedo desde el borde izquierdo hacía el centro de la pantalla (**Error! Reference source not found.**). Las secciones que comprenden la aplicación son las de resumen, perfil, estadísticas y citas médicas.

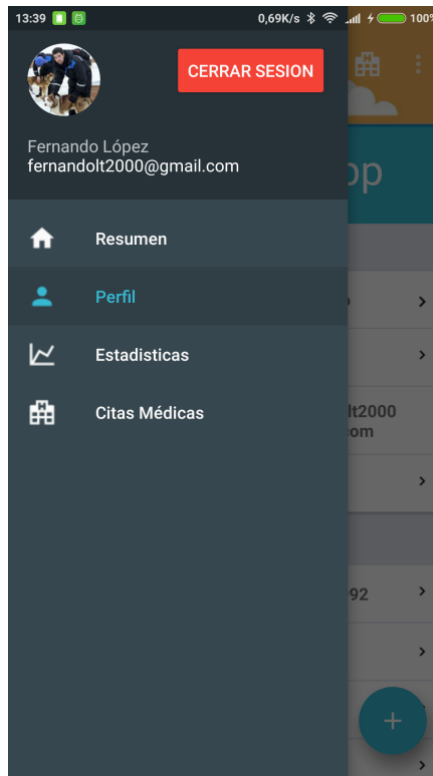


Ilustración 34 Menu desplegable

A continuación, se van a detallar cada una de las secciones citadas:

a) Pantalla de resumen:

La pantalla de resumen observada en la Ilustración 35, muestra un informe de los datos principales registrados en la App en los últimos tres días, teniendo en cuenta el índice glucémico, la nutrición y la actividad física. Para ello, se utiliza un diagrama de líneas, una tabla y un diagrama de sectores respectivamente.

Un elemento muy importante a tener en cuenta será el botón flotante posicionado en la parte inferior derecha de la pantalla, que se utilizará para añadir de forma manual datos como puede ser la dosis de insulina ingerida, los alimentos tomados, modificar el peso, añadir una nueva medida de pulso cardiaco, etc. En la Ilustración 36 se observa cómo se despliega ese botón al pulsarlo y visualizar la animación.



Ilustración 35 Pantalla de resumen



Ilustración 36 Botón flotante

b) Pantalla de perfil:

Al entrar en la sección de Mi Perfil, se muestra toda la información correspondiente al usuario, y se permite gestionarlo para que sus resultados se vean reflejados en los informes que se generan posteriormente en la aplicación



Ilustración 37 Mi perfil

c) Pantalla de Estadísticas:

En esta pantalla se observan 4 sub-apartados, y cada uno de ellos muestran los análisis de unos determinados parámetros relacionados con la diabetes. Para visualizar la información, se debe filtrar por fecha para conocer el intervalo en el que se quieren conocer los datos. En el apartado Relación, se permite filtrar además por dos tipos de parámetros en una misma gráfica, consiguiendo así un análisis más profundo para el usuario. Un ejemplo que relaciona el índice glucémico con la ingesta de calorías se puede observar en la Ilustración 38.

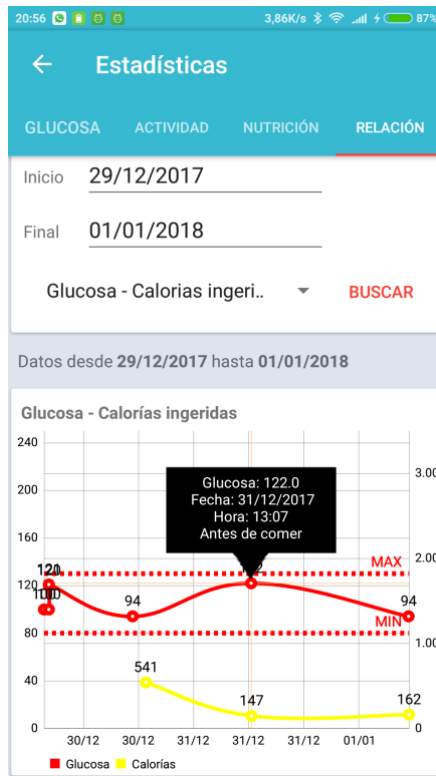


Ilustración 38 Pantalla de estadísticas

Para llegar a conseguir que aparezcan datos en cada uno de las gráficas, se deben añadir previamente a través de los diálogos que se activan con el botón flotante comentado anteriormente. Un ejemplo de dialogo de introducción de un dato de nutrición se puede observar en la Ilustración 39.

a) Pantalla de citas médicas:

El sistema permite al usuario recibir alarmas para que pueda utilizar la aplicación a modo de recordatorio de eventos como revisiones médicas, análisis, medición de algún parámetro, etc. En la Ilustración 40 se puede observar el aspecto de esa ventana cuando se te tiene registrada una cita médica.



Ilustración 39 Introducción de nutrición



Ilustración 40 Citas médicas

Anexo 2 Test de Evaluación

1. Tareas “*Think aloud*”

1. Realizar el registro en la aplicación con su cuenta de Google.
2. Terminar de completar el perfil, modificando aquellos campos provenientes de la configuración de la cuenta de Google que no estén debidamente actualizados y proporcionando además la información adicional requerida (peso, altura, fecha del debut diabético, rango ideal de glucosa)
3. Añadir una medición de glucosa. Simulando que se hizo previa al desayuno, el mismo día de esta prueba.
4. Comprobar que la medición de glucosa del paso 2 aparece en el índice de glucosa del resumen.
5. Añadir una dosis de insulina. Simulando que se hizo previa al desayuno de ese mismo día y momentos después de la medición de glucosa del paso 2.
6. Añadir una ingesta de comida, por ejemplo, el desayuno que se tomó tras la dosis de insulina inyectada a la que hace referencia el paso 4. Como requisito debe estar integrada por al menos dos elementos ej.: leche + cereales, leche + bizcocho, etc.
7. Añadir una nueva medición de glucosa. Simulando que se hizo posterior al desayuno del paso 5 (darle un valor diferente a la anterior)
8. Buscar la estadística que relaciona las medidas de glucosa con la insulina administrada. Mostrarla para el periodo comprendido entre el día

de la prueba y el día anterior (Interpretar los valores del gráfico que se muestra en pantalla)

2. Cuestionario de perfil y usabilidad

Rellene el siguiente post-cuestionario de evaluación de aplicación DiMo.

1. Indique su sexo:

- Mujer
- Hombre

2. Indique su rango de edad:

3. Con relación a su profesión:

- Estudiante
- Trabajador (activo o parado)
- Jubilado

4. Fecha de debut diabético:

5. ¿Cómo considera sus conocimientos en el manejo de un *smartphone*?

- Alto
- Medio
- Bajo
- Ninguno

6. ¿Con qué frecuencia accede a Internet?

- Cada día
- Varias veces por semana
- Alguna vez
- Nunca

7. Los diferentes paneles gráficos estadísticos de la aplicación móvil (Índice de glucosa, carbohidratos por ingesta, calorías por ingesta, etc.) ¿Proporcionan información suficiente para su rápida interpretación?
- Soy capaz de identificar, de manera eficaz, todas las mediciones en un determinado periodo de tiempo, así como sus valores y el momento concreto en el que se realizaron.
 - Por lo general, identifico las mediciones sobre las gráficas en el periodo concreto indicado, aunque los ejes de éstas, no me proporcionan toda la información necesaria o su rango es inadecuado.
 - No, he tenido varios problemas recurrentes para localizar e identificar mediciones sobre las gráficas.
8. Prestando atención a las denominaciones de las secciones de la aplicación móvil, los títulos de los diferentes paneles y gráficas y los nombres de las etiquetas utilizadas ¿Qué respuesta considera más adecuada?
- He entendido el significado de cada uno de los títulos en los diferentes paneles estadísticos, los nombres de las secciones, etc., sin ningún tipo de confusión.
 - Al principio, he tenido problemas para interpretar qué representa alguna gráfica a partir de su título o he encontrado inconsistencias entre términos con significados parecidos que me crearon confusión, aunque más adelante, una vez completada la evaluación, entiendo y sé interpretar cada gráfica y panel de la aplicación móvil.
 - Completada esta evaluación, todavía hay gráficas y/o paneles de la aplicación móvil que no sé interpretar o no entiendo.
9. Cuando un usuario realiza una acción en la aplicación móvil espera cierta respuesta por parte de ésta sobre el estado de su acción (Por

ejemplo, respuestas textuales: “perfil actualizado”, “consumición registrada”, “valor registrado correctamente”. Seleccione la respuesta que más se ajusta al feedback que usted percibe al utilizar la aplicación móvil.

- El feedback es bueno, de modo que tras cada acción que realizo con la aplicación móvil obtengo una respuesta bastante descriptiva que me ayuda a “no perderme” mientras manejo la aplicación.
- El feedback existe pero es mejorable, bien porque algunas acciones están carentes de feedback o porque la respuesta de la aplicación a mi acción no es suficientemente descriptiva.
- El feedback es malo, apenas existe, o las respuestas son inadecuadas forzando situaciones en las que me desoriento al manejar la aplicación.

10. ¿Qué momentos (experiencias de usuario) durante la interacción con las dos aplicaciones considera que fueron satisfactorios para usted? Enumérelos razonando brevemente el porqué, y detallando a cuál de las dos aplicaciones hacen referencia:

11. ¿Qué momentos (experiencias de usuario) durante la interacción con las dos aplicaciones considera que fueron poco o nada satisfactorios para usted? Enumérelos razonando brevemente el porqué, y detallando a cuál de las dos aplicaciones hacen referencia.

12. Comentarios:

Bibliografía

- [1] Organización Mundial de la Salud. “Informe Mundial sobre la Diabetes”. 2016;
<http://www.who.int/diabetes/global-report/es/>
- [2] Nightscout Project. ,2017;
<http://www.nightscout.info/>
- [3] Glooko, Remote Patient Monitoring for Diabetes. Mountain View, USA, 2017. <https://www.glooko.com/>
- [4] Diabeto Management System, Maharashtra, India. 2017;
<https://diabe.to/>
- [5] Glimp Application. Play store, Google;
<https://play.google.com/store/apps/details?id=it.ct.glicemia&hl=es>
- [6] Glup App. Play store, Google;
<https://play.google.com/store/apps/details?id=com.nurun.glup&hl=es>
- [7] Kantar Worldpanel Studio, 2017
<http://es.kantar.com/tech/m%C3%B3vil/2017/agosto-2017-cuota-de-mercado-de-smartphones-en-espa%C3%B1a-mayo-2017/>
- [8] Diabetes para Dummies, 2007, Wiley Publishing, Inc. Dr. Alan L. Rubin
- [9] Cuota de mercado de smartphones en España, 05/2017. Kantar World Panel
<https://www.kantarworldpanel.com/es/Noticias/Cuota-de-mercado-de-smartphones-en-Espana>

- [10] DEXCOM: Continuous Glucose Monitoring. USA; 2017;
<https://www.dexcom.com/es-ES>
- [11] Abbott Company.USA; 2017;
<http://www.abbott.com/>
- [12] Ian Sommerville, Ingeniería de Software. Prentice Hall Mexico 6^a Edición. Capítulo 8.
- [13] Google, Android Studio 2.3.3, 2017;
<http://developer.android.com/intl/es/sdk/index.html>
- [14] Hong Kong Institute of Vocational Education, Visual Paradigm 13.0 Community Edition, 2017;
<https://www.visual-paradigm.com>
- [15] Sun Microsystems, Java 1.8;
<https://www.java.com/en/>
- [16] World Wide Web Consortium, XML 1.1.;
<https://www.w3.org/XML/>
- [17] Douglas Crockford, JSON Format;
<http://www.json.org/>
- [18] Microsoft, Microsoft Word, 2017;
<http://products.office.com/es>
- [19] Equipo de desarrollo de GIMP, GIMP 2.8.16, 2017;
www.gimp.org.es/
- [20] Google, Google Fit Api, 2017;
<https://developers.google.com/fit/>

- [21] Gimp 2.9.8., Fundacion GNOME, 2017;
<http://www.gimp.org.es/>
- [22] Draw.io, JGraph Ltd, 2017;
<https://www.draw.io/>
- [23] NinjaMock, 2017;
<https://ninjamock.com/>
- [24] Estado actual de las versiones Android, 2017, Android developers;
<https://developer.android.com/about/dashboards/index.html>
- [25] Consola de desarrolladores de Google, 2017,
<https://developers.google.com/mobile/add>
- [26] Material Design Guide Line, Google, 2017,
<https://material.io/guidelines/>
- [27] Google Sign-In for Android, Google Developers, 2017,
<https://developers.google.com/identity/sign-in/android/start-integrating>