

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA



ESCOLA TÉCNICA SUPERIOR DE ENXEÑARÍA

Aplicación móvil *Android* de ayuda
en la Enfermedad de Crohn

Autor:

Alberto Ruibal Ojea

Tutor:

Paulo Félix Lamas

Grado en Ingeniería Informática

Junio 2022

Trabajo de Fin de Grao presentado en la Escuela Técnica Superior de Ingeniería de la
Universidad de Santiago de Compostela para la obtención del Grado en Ingeniería
Informática



D. Paulo Félix Lamas, Profesor/a del Departamento de Electrónica y Computación de la Universidad de Santiago de Compostela,

INFORMA:

Que la presente memoria, titulada *Aplicación móvil Android de ayuda en la Enfermedad de Crohn*, presentada por D. Alberto Ruibal Ojea para superar los créditos correspondientes al Trabajo de Fin de Grado de la titulación de Grado en Ingeniería Informática, se ha realizado bajo mi tutoría en el Departamento de Electrónica y Computación de la Universidad de Santiago de Compostela.

Y para que así conste a los efectos oportunos, expiden el presente informe en Santiago de Compostela, a 14 de junio de 2022:

Tutor/a,

Alumno/a,

Paulo Félix Lamas

Alberto Ruibal Ojea

Agradecimientos

Me gustaría agradecer a todos los profesores dentro y fuera de la universidad que han alimentado mi curiosidad, aunque algunos más que otros, pero especialmente a mi tutor Paulo Félix Lamas, por haber sido mi guía y mi estímulo durante la planificación, organización y redacción de mi Trabajo de Fin de Grado.

Gracias también a mi familia y amigos, quienes me han hecho ser quien soy, a los cuales solamente puedo manifestar mi más sincero agradecimiento al haber dado su tiempo, su esfuerzo y sus recursos en apoyarme durante esta (larga) etapa académica.

Por último, gracias a la Unidad de Enfermedad Inflamatoria Intestinal del Complejo Hospitalario de Pontevedra por haberme formado tanto como paciente como profesional en la realización de mi Trabajo de Fin de Grado y a todas aquellos pacientes que aportaron un pequeña pero valiosa parte de su tiempo en esta idea.

Resumen

En este Trabajo de Fin de Grado se pretende abordar el desarrollo de una aplicación móvil para la plataforma *Android* que asista a los enfermos de Crohn mediante el registro y visualización de síntomas y alimentación, la detección temprana de un brote de la enfermedad y el acceso a recursos médicos de referencia de una forma centralizada desde la aplicación a desarrollar.

Índice general

1. Introducción.....	1
1.1. Objetivos del sistema.....	3
1.2. Entregables del proyecto	3
1.3. Exclusiones del proyecto.....	3
1.4. Restricciones del proyecto	4
1.5. Supuestos del proyecto.....	4
2. Gestión del proyecto	5
2.1. Análisis y gestión de riesgos.....	5
2.1.1. Identificación de riesgos.....	5
2.1.2. Análisis cualitativo de los riesgos	6
2.1.3. Planificación de la respuesta a los riesgos	7
2.2. Ciclo de vida del proyecto.....	8
2.3. Gestión de la configuración.....	8
2.4. Planificación.....	9
2.4.1. Esquema de descomposición de tareas (EDT).....	9
2.4.2. Estimación temporal en base al EDT	10
3. Análisis	11
3.1. Historias de Usuario.....	11
3.1.1. Lista de tareas	15
3.2. Especificación de requisitos.....	16
3.2.1. Requisitos funcionales del sistema.....	17
3.2.2. Requisitos no funcionales del sistema.....	18
3.3. Actores.....	18
4. Diseño	19
4.1. Arquitectura escogida	19
4.2. Diagrama de clases.....	21
4.3. Diagrama de diseño del sistema	22

4.4.	Análisis de tecnologías	23
4.4.1.	Almacenamiento de datos	23
4.4.2.	Base de datos de alimentos	23
4.4.3.	Tecnologías de desarrollo de sistemas expertos	24
5.	Desarrollo y pruebas	25
5.1.	Sprint 1.....	25
5.1.1.	Análisis	25
5.1.2.	Diseño	26
5.1.3.	Implementación.....	27
5.1.4.	Pruebas realizadas en el Sprint	28
5.2.	Sprint 2.....	29
5.2.1.	Análisis	29
5.2.2.	Diseño	29
5.2.3.	Implementación.....	30
5.2.4.	Pruebas realizadas en el Sprint	30
5.3.	Sprint 3.....	32
5.3.1.	Análisis	32
5.3.2.	Diseño	32
5.3.3.	Implementación.....	33
5.3.4.	Pruebas realizadas en el Sprint	33
5.4.	Sprint 4.....	34
5.4.1.	Análisis	34
5.4.2.	Diseño	34
5.4.3.	Implementación.....	37
5.4.4.	Pruebas realizadas en el Sprint	38
5.5.	Sprint 5.....	39
5.5.1.	Análisis	39
5.5.2.	Diseño	39
5.5.3.	Implementación.....	39
5.5.4.	Pruebas realizadas en el Sprint	41
5.6.	Evaluación de la usabilidad.....	42
5.6.1.	Evaluación según heurística de Nielsen	42
5.6.2.	Evaluación de la usabilidad con usuarios finales	43
6.	Conclusiones y ampliaciones	45
	Bibliografía	48
	APÉNDICE A. Manual Técnico	51
	APÉNDICE B. Manual de Usuario	54

Índice de Figuras

Figura 2.1: Esquema de Descomposición de Tareas (EDT).....	9
Figura 4.1: Modelo MVVM.....	19
Figura 4.2: Diagrama de clases.....	21
Figura 4.3: Diagrama de diseño del sistema.....	22
Figura 5.1: Guía de Inicio y Ventana Principal del Sprint 1.....	27
Figura 5.2: Módulo de Sintomatología del Sprint 1.....	27
Figura 5.3: Módulo de Alimentos y nueva visualización de síntomas.....	30
Figura 5.4: Ajustes, Privacidad y Alertas del Sprint 3.....	33
Figura 5.5: Síntomas y Salud del Sprint 4.....	37
Figura 5.6: Tipos de Recomendaciones del Sprint 5.....	40
Figura 5.7: Ejemplos de Recomendaciones del Sprint 5.....	40
Figura A.1: Estructura de ficheros del proyecto.....	52
Figura B.1: Guía de Inicio (I).....	54
Figura B.2: Guía de Inicio (II).....	55
Figura B.3: Ventana Principal.....	55
Figura B.4: Consulta de síntomas registrados y su significado.....	56
Figura B.5: Registro de síntomas.....	57
Figura B.6: Módulo de Alimentación.....	58
Figura B.7: Añadir alimentos y marcado de un no recomendado.....	58
Figura B.8: Añadir alimentos no recomendados y consulta.....	59
Figura B.9: Alerta de un alimento no recomendado.....	60

Figura B.10: Módulo de Salud y estado de ánimo.....	61
Figura B.11: Consulta y brote de la enfermedad.....	62
Figura B.12: Acceso a Recomendaciones (I).....	63
Figura B.13: Acceso a Recomendaciones (II)	63
Figura B.14: ajustes y Privacidad	64

Índice de Tablas

Tabla 2.1: Tabla de Identificación de Riesgos	6
Tabla 2.2: Tabla de Probabilidad de ocurrencia de los Riesgos.....	6
Tabla 2.3: Tabla de Valoración del Impacto de los Riesgos.....	6
Tabla 2.4: Tabla Evaluación Probabilidad-Impacto de los Riesgos	7
Tabla 2.5: Matriz de Probabilidad-Impacto de los Riesgos	7
Tabla 2.6: Tabla de Respuesta a los Riesgos.....	7
Tabla 2.7: Tabla Estimación Temporal.....	10
Tabla 3.1: HU-1. Registro de síntomas	12
Tabla 3.2: HU-2. Estado de ánimo.....	12
Tabla 3.3: HU-3. Registro de alimentos ingeridos	12
Tabla 3.4: HU-4. Registro de alimentos no recomendados	12
Tabla 3.5: HU-5. Visualización sencilla de los registros.....	13
Tabla 3.6: HU-6. Privacidad del usuario	13
Tabla 3.7: HU-7. Alerta de alimentos dañinos	13
Tabla 3.8: HU-8. Tipo de la enfermedad.....	14
Tabla 3.9: HU-9. Alerta de brotes.....	14
Tabla 3.10: HU-10. Recomendaciones médicas.....	14
Tabla 3.11: Tabla de la Lista de Tareas a realizar	16
Tabla 3.12: Tabla de Requisitos funcionales (I)	17
Tabla 3.13: Tabla de Requisitos funcionales (II)	18

Tabla 3.14: Tabla de Requisitos no funcionales	18
Tabla 3.15: Tabla Actores del Sistema.....	18
Tabla 4.1: Bases de Datos disponibles.....	23
Tabla 4.2: Tecnologías de Desarrollo de Sistemas Expertos en Android	24
Tabla 5.1: Resultados del cuestionario de Usabilidad.....	44

Capítulo 1

Introducción

La enfermedad de Crohn [1], en adelante, la enfermedad, es una Enfermedad Inflamatoria Intestinal (EII) crónica que surge por una respuesta inadecuada del sistema inmunitario hacia el organismo, desencadenando una respuesta inmunológica inflamatoria para “acabar” con agentes extraños (desconocido su porqué, aunque recientes estudios apuntan a un uso intensivo de antibióticos), haciendo que el revestimiento del tracto digestivo se inflame e irrite. Puede afectar a cualquier parte del tracto gastrointestinal, desde la boca hasta el final de este.

La afección de la enfermedad sobre la pared intestinal puede conllevar a la estenosis (estrechamiento del intestino) e incluso a su completa oclusión en situaciones críticas, ofreciendo una serie de síntomas con una frecuencia de aparición mayor o menor hasta llegar a la necesidad de realizar una cirugía en caso de que los diversos tratamientos no consigan estabilizarla. Es una enfermedad con un carácter sistémico, es decir, afecta al aparato digestivo y a otras localizaciones tales como las articulaciones, piel, boca, etc., siendo esto conocido como manifestaciones extraintestinales de la enfermedad (MEI) [1].

El tipo de la enfermedad [2] es el resultado del diagnóstico realizado por un especialista, pero puede cambiar en un mismo paciente con el tiempo dependiendo de su edad de diagnóstico, la localización del tramo afectado y su patrón evolutivo [3]. Por ello, se suele clasificar en los siguientes tipos, en donde se especifican sus síntomas [4] más relevantes:

- Ileítis: a nivel del íleon terminal, con una serie de síntomas asociados tales como diarrea, dolor abdominal, cansancio, fiebre y pérdida de peso.
- Colitis: afectación de una o varias zonas del colon, con una serie de síntomas asociados como diarrea, dolor abdominal, fiebre, sangrado y urgencia en ir al baño.

- Íleo-colitis: es el más frecuente (casi un 65% de los pacientes), afectando principalmente al íleon distal y al área cecal, el primer tramo del colon). Cuenta con los síntomas de los tipos anteriores, además de síntomas más generales como menos apetito, dolor articular y dolor de cabeza.
- Gastrointestinal alta: es el menos frecuente, afectando a la zona del esófago, estómago, duodeno, yeyuno e íleon proximal/medio, con unos síntomas tales como fiebre, náuseas, menos apetito y pérdida de peso.
- Perianal: anomalía o alteración de la zona del ano y recto, con una serie de síntomas asociados como diarrea, dolor anal, sangrado y heridas perianales.

Se denota como patrón evolutivo de la enfermedad [5] a la evolución de esta a lo largo del tiempo, cuya clasificación se realiza en base al grado de afectación y al daño estructural de la pared intestinal. Al igual que el tipo, el patrón no es independiente ni estable a lo largo del tiempo. Existen 3 patrones: comportamiento inflamatorio, comportamiento estenosante y comportamiento fistulizante.

Denominamos brote [5] a la reactivación de la enfermedad (se alternan periodos de actividad con periodos de remisión), el cual puede surgir por diversos factores como estrés, infecciones, no tomar la medicación o una mala alimentación, entre otros. Debido a su manifestación intermitente, en ocasiones los síntomas asociados pueden confundirse con una simple diarrea o gastroenteritis, siendo necesario recabar la mayor cantidad de información posible para su diferenciación.

Hoy en día, el especialista recolecta los diversos síntomas que han podido acaecer durante el último periodo para determinar el estado de la enfermedad, junto con pruebas diagnósticas (tales como análisis, resonancias magnéticas o endoscopias entre otras) para determinar mejor su estado.

Actualmente, existen una serie de aplicaciones ya creadas pensadas para ayudar en el transcurso de la enfermedad, pero cuentan con una serie de limitaciones:

- Aplicación de la UVigo y SERGAS [6]: en fase de pruebas, un Bot de *Telegram* con 25 unidades didácticas basadas en vídeos con pequeños cuestionarios.
- Crohn's and Colitis Support (*Android*): no disponible en España y es un foro y guía para seguir.
- My IBD Care: Manage Your Crohn's & Colitis Better (*Android*): no cuenta con una opción de añadir alimentos para poder cruzar datos y no disponible en español.
- Care4Today EII (*Android*): solo es un registro de peso y deposiciones.
- Crohns Food Tracker (*Android*): un diario para realizar un seguimiento no disponible en español.
- Atentum: interfaz tipo chat destinada al postoperatorio.

Las limitaciones existentes en dichas aplicaciones nos llevan a proponer una nueva aplicación para dispositivos móviles que empodere al paciente con una serie de herramientas.

1.1. Objetivos del sistema

El Trabajo de Fin de Grado tendrá un objetivo principal: el desarrollo de una aplicación *Android* que proporcione al paciente una ayuda en la gestión de la Enfermedad de Crohn, denominada “Asistente Crohn”. Este objetivo principal se desglosa en los siguientes objetivos secundarios:

1. Identificar y alertar de manera automática respecto a la aparición de un posible brote de la enfermedad de Crohn a partir de ciertos conjuntos de síntomas referidos por los especialistas. Esto conlleva la necesidad de: (1) implementar una funcionalidad de registro de síntomas; (2) implementar una herramienta que permita la identificación de distintos patrones de síntomas.
2. Proporcionar al usuario un acceso orientativo a recomendaciones de referencia sobre la gestión de la enfermedad, avaladas por sociedades médicas.
3. Proporcionar al usuario capacidades de autorregistro de la enfermedad, con especial atención a su alimentación, con el fin de servir de orientación sobre el origen de cada brote.

1.2. Entregables del proyecto

- Documentación y código fuente. Esta propia documentación, junto con el código fuente de la aplicación.
- Manual de usuario: en donde se especifique el funcionamiento del software y que se adjuntará a esta memoria.
- Programa ejecutable *Android*: estará disponible para su instalación en la tienda de aplicaciones de Google, Play Store.

1.3. Exclusiones del proyecto

- La aplicación no servirá para realizar un diagnóstico médico, servirá como una orientación.

1.4. Restricciones del proyecto

- La duración del proyecto deberá ser de 412,5 horas de acuerdo con la normativa de los Trabajos de Fin de Grado para la titulación del Grado en Ingeniería Informática [7].
- La aplicación móvil deberá desarrollarse en *Android* y publicarse en la tienda de aplicaciones.

1.5. Supuestos del proyecto

- El usuario deberá haber sido diagnosticado de enfermedad de Crohn por un especialista médico, quien habrá asignado al paciente un tipo de la enfermedad de los explicados en la Introducción de esta memoria.
- El usuario deberá disponer de un dispositivo móvil *Android* con versión 9.0 o superior y con conexión a Internet.

Capítulo 2

Gestión del proyecto

En este capítulo se abordará la organización y planificación del proyecto, partiendo de los objetivos propuestos (*ver apartado 1.1 del Capítulo de Introducción*).

En primer lugar, se tomará como punto de partida un Análisis de Riesgos en base a los objetivos del proyecto, continuando así con la elección del ciclo de vida a utilizar y la metodología de desarrollo. Además, se definirá la gestión de la configuración utilizada a lo largo del desarrollo del proyecto.

Por último, se especificará la planificación del proyecto, desglosando así desde una lista de tareas a realizar al esquema de descomposición de tareas (EDT) y las duraciones estimadas y finales.

2.1. Análisis y gestión de riesgos

El Análisis de Riesgos es vital para la realización del proyecto. Para ello, se empezará por indicar la identificación de los riesgos, determinando a su vez su probabilidad de aparición e impacto sobre el proyecto. Para finalizar, se indicará la planificación que se seguirá como respuesta a dichos riesgos.

2.1.1. Identificación de riesgos

CÓDIGO	NOMBRE DEL RIESGO	DESCRIPCIÓN
RSK-1	Falta de experiencia en el desarrollo de aplicaciones móviles.	La falta de conocimientos en el desarrollo de aplicaciones móviles y el estrés derivado de ello podría alterar el desarrollo de esta.
RSK-2	Incertidumbre en cuanto a los requisitos del proyecto.	La incertidumbre en el proyecto es conocida desde el principio.
RSK-3	Mala gestión de riesgos.	El alumno hace una mala gestión de los riesgos aquí presentes, bien sea no identificando riesgos importantes o restando importancia a los ya identificados.
RSK-4	Planificación optimista.	La poca experiencia por parte del alumno podría ser una dificultad para una planificación rigurosa.

CAPÍTULO 2. GESTIÓN DEL PROYECTO

RSK-5	Dificultades de comunicación con el tutor por agravamiento de la pandemia.	Debido a un agravamiento de la pandemia, la comunicación con el tutor se ve afectada (por enfermedad de ambos o por una modalidad telemática).
RSK-6	Pocos usuarios para pruebas.	Hay pocos usuarios dispuestos a probar la aplicación en fase de pruebas, bien sea por desconocimiento de la aplicación o falta de confianza en ella.

Tabla 2.1: Tabla de Identificación de Riesgos

2.1.2. Análisis cualitativo de los riesgos

A continuación, se ofrecen los parámetros para la evaluación de Probabilidad e Impacto de los Riesgos.

OCURRENCIA	PROBABILIDAD	DESCRIPCIÓN
Mayor o igual al 80%	Alta	El riesgo va a aparecer a lo largo del proyecto de un momento a otro.
Entre 30-80%	Media	El riesgo tiene una cierta probabilidad de aparecer.
Menor o igual al 30%	Baja	El riesgo tiene pocas opciones de aparecer en el proyecto, pero no se descarta.

Tabla 2.2: Tabla de Probabilidad de ocurrencia de los Riesgos

REPERCUSIÓN	PROBABILIDAD	DESCRIPCIÓN
Mayor o igual al 20%	Severo	El riesgo va a impactar gravemente en el proyecto y es necesaria una solución urgente.
Entre 10-20%	Moderado	El riesgo se puede resolver sin complicaciones, pero es considerado prioritario.
Menor o igual al 10%	Insignificante	El riesgo tendría una repercusión ínfima en el proyecto y no existe una gran necesidad de gestionarlo.

Tabla 2.3: Tabla de Valoración del Impacto de los Riesgos

CAPÍTULO 2. GESTIÓN DEL PROYECTO

Una vez determinadas las escalas de Probabilidad e Impacto, se muestra para cada riesgo su evaluación, junto con la Matriz de Probabilidad-Impacto:

RIESGO	PROBABILIDAD	IMPACTO
RSK-1	Alta	Severo
RSK-2	Alta	Moderado
RSK-3	Media	Severo
RSK-4	Media	Moderado
RSK-5	Baja	Moderado
RSK-6	Baja	Insignificante

Tabla 2.4: Tabla Evaluación Probabilidad-Impacto de los Riesgos

IMP./PROB.	ALTA	MEDIA	BAJA
Severo	RSK-1	RSK-3	RSK-6
Moderado	-	RSK-4	RSK-5
Insignificante	RSK-2	-	-

Tabla 2.5: Matriz de Probabilidad-Impacto de los Riesgos

2.1.3. Planificación de la respuesta a los riesgos

Una vez evaluados los riesgos en la Matriz de Probabilidad e Impacto según su incidencia, se plantea el tratamiento a seguir para cada uno. La adopción de una metodología ágil será la principal respuesta a los riesgos identificados.

RIESGO	TRATAMIENTO
RSK-1	Mitigar. Mediante una metodología ágil, se podrá reducir el impacto sobre el proyecto.
RSK-2	Mitigar. Mediante una metodología ágil, se priorizarán determinados requisitos en base a las necesidades establecidas.
RSK-3	Mitigar. Mediante una metodología ágil, se promueve la entrega frecuente y la revisión de código y documentación.
RSK-4	Prevención. Mediante una metodología ágil, la planificación se vuelve más flexible, contribuyendo así a una mejor gestión de riesgos.
RSK-5	Aceptar. Teniendo en cuenta la posibilidad muy reducida de que ocurra, se acepta el riesgo.
RSK-6	Mitigar. Mantener una comunidad de usuarios que deseen ayudar con las pruebas.

Tabla 2.6: Tabla de Respuesta a los Riesgos

2.2. Ciclo de vida del proyecto

El ciclo de vida escogido para la realización de este proyecto ha sido un ciclo de vida incremental mediante la metodología Scrum [8], ya que reduce la complejidad en el desarrollo al permitir una gran flexibilidad, manteniendo a su vez una participación por parte de todos los usuarios involucrados en su desarrollo.

En Scrum, cada bloque de trabajo se denomina Sprint y tiene una estructura y duración fijas. Al finalizar cada Sprint, se debe entregar un producto terminado al cliente (en este caso, el tutor). Cada Sprint cuenta con una serie de fases: análisis, diseño, implementación y pruebas.

Una vez finalizado cada Sprint, el cliente y el equipo de desarrollo se reunirán para definir el siguiente Sprint y realizar una retrospectiva del Sprint realizado, efectuando así un análisis del trabajo seguido para perpetuar una mejora continua a lo largo del desarrollo al registrar e incorporar los aprendizajes en el siguiente Sprint.

Para poder utilizar la metodología, se usará la herramienta de Notion [9], que cuenta con una plantilla que implementa la gestión de proyectos basada en Scrum mediante diferentes tableros y vistas: *backlog* (un listado de todas las tareas que se pretenden hacer durante el desarrollo del proyecto), *panel del Product Owner* (el alumno, líder del equipo Scrum al ser de una persona y responsable directo de maximizar el valor del producto), *Epics* (pequeños segmentos del trabajo que contribuyen a un hito), *Historias de Usuario* (un conjunto de Historias de Usuario forman los Epics), *retrospectiva* y los *diferentes Sprints*.

2.3. Gestión de la configuración

Sobre el software a desarrollar para la aplicación *Android*, se ha optado por incluir un proceso de gestión de la configuración, en donde se generará una nueva versión del software por cada Sprint.

Para ello, se utilizará la herramienta GitHub [10] para tener un control de los cambios realizados a lo largo del desarrollo, haciendo uso de la operación “push” para registrar los cambios y de comentarios identificativos de los cambios realizados.

Para asegurar la integridad del software, en este caso el único elemento de configuración a gestionar, se usará únicamente una cuenta de GitHub mediante los dispositivos personales autorizados para ello.

2.4. Planificación

En este apartado se incluye el EDT del proyecto según la metodología del PMBOK, partiendo de este para estimar la duración de las actividades y crear así una planificación del proyecto.

2.4.1. Esquema de descomposición de tareas (EDT)

Se muestra a continuación el EDT correspondiente luego de haber obtenido la lista de tareas según las Historias de Usuario. Se han agrupado las tareas en paquetes, siendo cada uno un módulo de la aplicación o entrega de documentación:

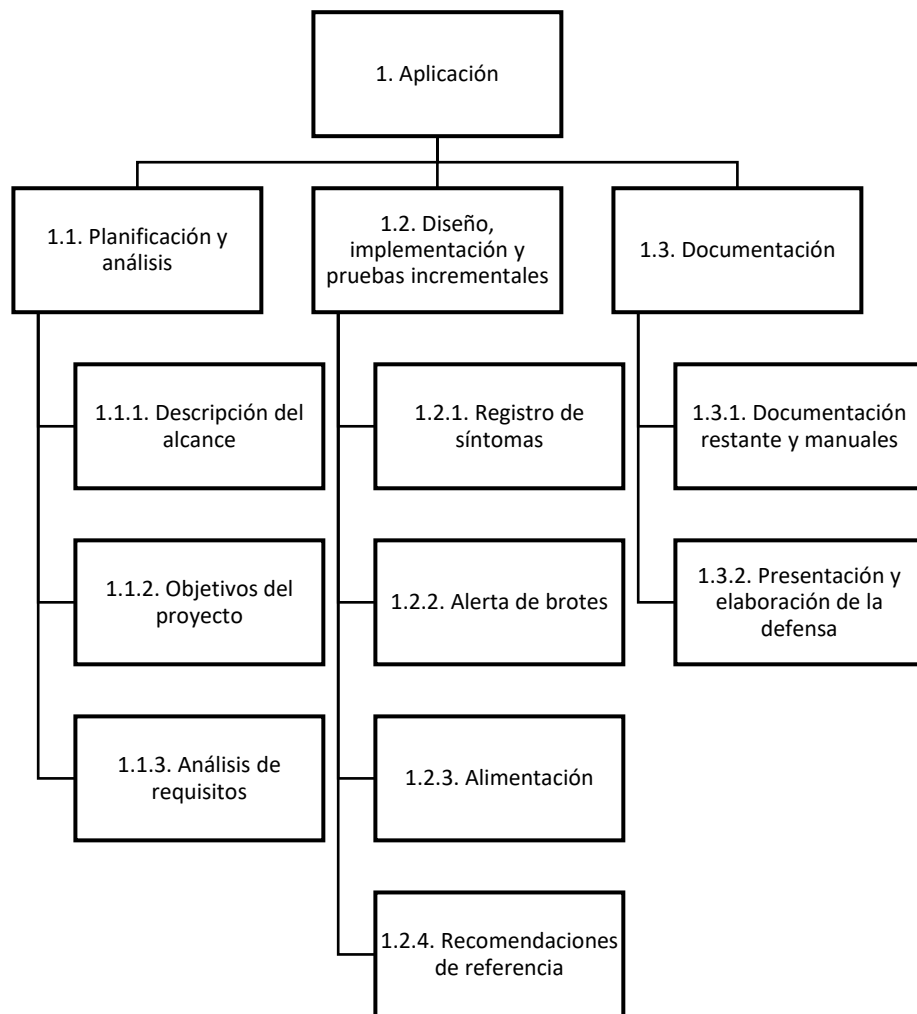


Figura 2.1: Esquema de Descomposición de Tareas (EDT)

2.4.2. Estimación temporal en base al EDT

Una vez obtenido el EDT, se obtiene la planificación temporal del total del proyecto. Se asume una carga de trabajo del alumno de 25 horas semanales teniendo en cuenta la carga que supone un Trabajo de Fin de Grado según la normativa vigente para el Grado en Ingeniería Informática [7]: 412,5 horas en total.

ESTIMACIÓN TEMPORAL	15,2 SEMANAS	412,5 HORAS
Planificación y análisis	3 semanas	75 horas
Descripción del alcance	2 días	10 horas
Objetivos del proyecto	3 días	15 horas
Análisis de requisitos	2 semanas	50 horas
Diseño, implementación y pruebas incrementales	10 semanas	277,5 horas
Registro de síntomas	17,0 días	85 horas
Alimentación	19,0 días	95 horas
Alerta de brotes	12,5 días	62,5 horas
Recomendaciones de referencia	7,0 días	35 horas
Documentación	2,2 semanas	60 horas
Documentación restante y manuales	2 semanas	50 horas
Presentación y elaboración de la defensa	2 días	10 horas

Tabla 2.7: Tabla Estimación Temporal

Dado que se va a utilizar la metodología ágil Scrum en el proyecto, no parece razonable establecer una planificación temporal en base a las tareas a realizar, ya que cada Sprint incluirá su propia planificación.

Capítulo 3

Análisis

En este capítulo se concretarán los detalles del sistema. Mediante la metodología Scrum, se ha optado por utilizar la técnica de Historias de Usuario como paso previo a la especificación de la lista de tareas para realizar.

Posteriormente, se ha optado por añadir la especificación de requisitos del sistema para así concretar de una forma más amplia la especificación del sistema.

3.1. Historias de Usuario

Se ha empleado la técnica de Historias de Usuario para determinar las funcionalidades del proyecto. Para ello, cada historia de usuario cuenta con una serie de campos, que se detallan a continuación.

- Sprint: Iteración asignada a dicha historia de usuario.
- EPIC: Objetivo o subobjetivo que se persigue.
- Prioridad: P0-P2, de mayor a menor prioridad.
- Estimación en Puntos: Puntos de Historia, partiendo de que 1 punto equivale a un día de trabajo.
- Criterios de aceptación: lista a completar para su finalización. Todas tendrán unos criterios comunes, que serán:
 - Código hecho para dichas funcionalidades
 - Pruebas hechas y aprobadas
 - Preguntas, respuestas y problemas solventados
 - Cambios de configuración comentados
 - Documentación actualizada
 - Revisión de código completa

HU-1	REGISTRO DE SÍNTOMAS
Descripción	Como usuario, quiero registrar mis síntomas para tener un histórico.
Sprint	Sprint 1
EPIC	Registro de síntomas
Prioridad	P0
Estimación en Puntos	14,5
Criterios de aceptación	Dada la interfaz, cuando el usuario marque sus síntomas, luego se guardarán y podrá acceder a ellos para su consulta.

Tabla 3.1: HU-1. Registro de síntomas

HU-2	ESTADO DE ÁNIMO
Descripción	Como usuario, quiero registrar mi estado de ánimo para determinar cómo me encuentro en mi día a día.
Sprint	Sprint 1
EPIC	Registro de síntomas
Prioridad	P1
Estimación en Puntos	2,5
Criterios de aceptación	Dado el usuario, cuando seleccione su estado de ánimo mediante los emojis, luego podrá acceder a su histórico de registros.

Tabla 3.2: HU-2. Estado de ánimo

HU-3	REGISTRO DE ALIMENTOS INGERIDOS
Descripción	Como usuario, quiero registrar los alimentos que tomo para tener un histórico.
Sprint	Sprint 2
EPIC	Registro de alimentación
Prioridad	P1
Estimación en Puntos	6
Criterios de aceptación	Dada la interfaz, cuando el usuario registre un alimento, luego tendrá la opción de consultarlo más tarde.

Tabla 3.3: HU-3. Registro de alimentos ingeridos

HU-4	REGISTRO DE ALIMENTOS NO RECOMENDADOS
Descripción	Como usuario, quiero poder registrar los alimentos que me hacen daño para poder consultarlos.
Sprint	Sprint 2
EPIC	Registro de alimentación
Prioridad	P0
Estimación en Puntos	2,5
Criterios de aceptación	Dada la interfaz, cuando el usuario busque un alimento y lo guarde en su lista de alimentos no recomendados, luego podrá acceder a esa lista y consultar los alimentos que le hacen daño.

Tabla 3.4: HU-4. Registro de alimentos no recomendados

HU-5	CALENDARIZACIÓN DE LOS REGISTROS
Descripción	Como usuario, quiero poder visualizar mis síntomas y alimentación de una forma sencilla
Sprint	Sprint 2
EPIC	Interfaz atractiva
Prioridad	P1
Estimación en Puntos	4,5
Criterios de aceptación	Dado un calendario, cuando el usuario acceda a él luego podrá ver sus síntomas y alimentación diferenciados por día. Dados los registros, cuando el usuario acceda al registro de síntomas luego podrá ver solamente sus síntomas (y viceversa para la alimentación).

Tabla 3.5: HU-5. Visualización sencilla de los registros

HU-6	PRIVACIDAD DEL USUARIO
Descripción	Como usuario, quiero que mis datos no salgan de mi dispositivo para proteger mi información.
Sprint	Sprint 3
EPIC	Privacidad
Prioridad	P0
Estimación en Puntos	1,5
Criterios de aceptación	Dados los ajustes del sistema, cuando el usuario acceda a ellos, luego podrá conocer cómo se tratan sus datos. Dado el almacenamiento de la información únicamente en el dispositivo, cuando el usuario acceda a ella o la borre, luego sus datos desaparecerán.

Tabla 3.6: HU-6. Privacidad del usuario

HU-7	ALERTA DE ALIMENTOS DAÑINOS
Descripción	Como usuario, me gustaría que la aplicación me alertase de los alimentos no recomendados en base a los síntomas registrados.
Sprint	Sprint 3
EPIC	Aviso de brotes
Prioridad	P1
Estimación en Puntos	4,5
Criterios de aceptación	Dado un síntoma, cuando el usuario presente ese síntoma después de ingerir el mismo alimento de forma reiterada, luego la aplicación le sugerirá que ese alimento puede ser el problema. Dada una alerta, cuando el usuario acepte la sugerencia, luego se añadirá el alimento a la sección de alimentos no recomendados.

Tabla 3.7: HU-7. Alerta de alimentos dañinos

HU-8	TIPO DE LA ENFERMEDAD
Descripción	Como usuario, quiero poder añadir el tipo de mi enfermedad y poder modificarlo.
Sprint	Sprint 4
EPIC	Aviso de brotes
Prioridad	P0
Estimación en Puntos	0,5
Criterios de aceptación	Dado el usuario, cuando acceda a la aplicación se le preguntará por su tipo de Crohn, luego la aplicación usará el tipo para determinar un posible brote de la enfermedad.

Tabla 3.8: HU-8. Tipo de la enfermedad

HU-9	ALERTA DE BROTES
Descripción	Como usuario, quiero que la aplicación me alerte de posibles brotes para poder actuar.
Sprint	Sprint 4
EPIC	Aviso de brotes
Prioridad	P0
Estimación en Puntos	12
Criterios de aceptación	Dada la interfaz, cuando el usuario tenga síntomas registrados, luego podrá ser notificado acerca de su estado de salud actual.

Tabla 3.9: HU-9. Alerta de brotes

HU-10	RECOMENDACIONES MÉDICAS
Descripción	Como usuario, quiero acceder a recomendaciones médicas para que me aconsejen en la gestión de mi enfermedad.
Sprint	Sprint 5
EPIC	Recomendaciones médicas
Prioridad	P0
Estimación en Puntos	7
Criterios de aceptación	Dada la interfaz, cuando el usuario acceda a recomendaciones, luego podrá consultar la información médica de la Plataforma G-EducaInflamatoria. Dada la interfaz, cuando las recomendaciones estén añadidas, luego se mencionará a la Plataforma G-EducaInflamatoria.

Tabla 3.10: HU-10. Recomendaciones médicas

Tal y como se puede observar en las Tablas 3.1 a 3.10, se han agrupado las Historias de Usuario en base a subsistemas funcionales según su afinidad y dependencias:

- SUB-1 - Subsistema General: HU-5, HU-6
- SUB-2 - Subsistema de Sintomatología: HU-1, HU-2
- SUB-3 - Subsistema de Alimentación: HU-3, HU-4
- SUB-4 - Subsistema de Recomendaciones Médicas: HU-10
- SUB-5 - Subsistema de Salud: HU-7, HU-8, HU-9

Según la HU-9 (Alerta de brotes), se ha de considerar el uso de un sistema experto (basado en reglas), para determinar si la persona está en brote o remisión de la enfermedad en base a los síntomas especificados en sus registros (en el que en base a su frecuencia de aparición se podrá establecer el estado actual de la enfermedad) y en donde se especificará el porqué de su resultado. El número de días por defecto serán de 3 días (así se diferencia del periodo de incubación de una gastroenteritis y un brote, el cual abarca más días entre sus otras características diferenciadoras).

Es decir, el sistema experto contará con una serie de reglas, formadas por un conjunto de síntomas (*ver apartado 1 del Capítulo de Introducción*) que ayudarán a determinar un posible brote.

3.1.1. Lista de tareas

En este apartado se incluyen las tareas asociadas a las Historias de Usuario descritas anteriormente.

HISTORIA DE USUARIO	TAREA	ESTIMACIÓN (DÍAS)
Como usuario, quiero acceder a recomendaciones médicas para que me aconsejen en la gestión de mi enfermedad	Citar recomendaciones de la Plataforma G-Educa Inflamatoria	4
	Implementar la interfaz de recomendaciones	1,5
	Estudiar la mejor forma de representar las recomendaciones	1
	Testear el módulo de recomendaciones	0,5
Como usuario, quiero poder registrar los alimentos que me hacen daño para poder consultarlos	Implementar la interfaz de registro de alimentos no recomendados	2,5
Como usuario, quiero asegurar los datos de mi dispositivo para proteger mi información	Crear un apartado de privacidad y términos de uso	1,5
Como usuario, quiero registrar mis síntomas para tener un histórico	Crear los objetos de representación de síntomas, alimentos, brotes y recomendaciones	2
	Implementar la interfaz de síntomas	7
	Testear el módulo de síntomas	3
	Estudiar la mejor forma de representar los síntomas en la interfaz	1
Como usuario, quiero registrar mi estado de ánimo para determinar cómo me encuentro en mi día a día	Implementar la interfaz del estado de ánimo	3
	Determinar cómo representar la escala de Likert para el ánimo	1

Como usuario, quiero registrar los alimentos que tomo para tener un histórico	Estudiar la mejor forma de representar el registro de alimentos	1
	Implementar la interfaz de registro de alimentos ingeridos	3
	Testear el módulo de alimentación	2
Como usuario, quiero que la aplicación me alerte de posibles brotes para poder actuar	Testear el módulo del estado actual	3
	Estudiar la mejor forma de representar el estado actual en la interfaz	1
	Implementar la interfaz del estado actual y su funcionalidad	4
	Crear las diversas reglas de patrones de síntomas	1
	Algoritmo para la detección de un posible brote de la enfermedad	3
Como usuario, quiero poder visualizar mis síntomas y alimentación de una forma sencilla	Crear una interfaz de calendario	1
	Implementar el calendario de modo que puedas ver tus registros de un día en concreto	3,5
Como usuario, me gustaría que la aplicación me alertase de los alimentos no recomendados en base a los síntomas registrados	Algoritmo para la detección de alimentos no recomendados	1,5
	Comparación de alimentos en busca de síntomas coincidentes	3
Comparación de alimentos en busca de síntomas coincidentes	Creación de una interfaz para añadir el tipo de Crohn	0,5

Tabla 3.11: Tabla de la Lista de Tareas a realizar

3.2. Especificación de requisitos

Una vez determinadas las funcionalidades del proyecto, se ha optado por incluir en este apartado la especificación de los requisitos de este. Para ello, cada requisito va a contar con nivel de importancia, el cual se detalla del siguiente modo:

- Alta: es un requisito indispensable para el correcto funcionamiento de la aplicación.
- Media: es un requisito que añadiría al sistema una característica deseable, pero que su ausencia no limitaría en gran medida a la aplicación.
- Baja: es un requisito prescindible, pero que a su vez aporta calidad al sistema.

3.2.1. Requisitos funcionales del sistema

ID	NOMBRE	DESCRIPCIÓN
RF-1	Crear un perfil de usuario	El usuario creará su perfil (Nombre y tablas de la base de datos) al iniciar la aplicación.
RF-2	Añadir síntomas	El usuario podrá registrar sus síntomas diarios, teniendo como restricción el registrar cada síntoma una vez al día.
RF-3	Ver síntomas registrados	El usuario podrá consultar los síntomas que ha registrado, mostrados por el día de registro en un calendario.
RF-4	Eliminar un síntoma registrado	El usuario podrá eliminar un síntoma registrado.
RF-5	Añadir el estado de ánimo diario	El usuario podrá registrar cómo se ha encontrado a lo largo del día.
RF-6	Ver el estado de ánimo diario	El usuario podrá ver su estado de ánimo para cada día en concreto en un calendario.
RF-7	Añadir alimentos ingeridos	El usuario podrá registrar los alimentos ingeridos según sus preferencias para determinar una dieta o aquellos que le han podido hacer daño.
RF-8	Ver alimentos ingeridos y no recomendados	El usuario podrá consultar sus alimentos ingeridos y no recomendados, mostrados por día de registro en un calendario.
RF-9	Eliminar un alimento ingerido	El usuario podrá eliminar un alimento ingerido.
RF-10	Añadir alimentos no recomendados	El usuario podrá añadir un alimento a la lista de alimentos que le hacen daño.
RF-11	Eliminar alimentos no recomendados	El usuario podrá eliminar un alimento de la lista de alimentos que le hacen daño.
RF-12	Alertar sobre un alimento dañino	El usuario será alertado de cuando un alimento le puede hacer daño en base a sus síntomas.
RF-13	Ver recomendaciones médicas	El usuario podrá acceder a recursos médicos sobre la gestión de su enfermedad.
RF-14	Añadir el tipo de la enfermedad	El usuario podrá añadir su tipo de la enfermedad para así conocer su estado de salud.
RF-15	Ver el estado de salud diario	El usuario podrá ver su estado de salud diario.
RF-16	Uso de un algoritmo de predicción para las reglas del estado de salud.	La aplicación asistirá al usuario mediante un algoritmo basado en los tipos de enfermedad y sus síntomas asociados.
RF-17	El usuario podrá borrar los datos contenidos en la aplicación en cualquier momento.	La aplicación permitirá borrar todos sus datos desde los ajustes del sistema.

Tabla 3.12: Tabla de Requisitos funcionales (I)

ID	DEPENDENCIAS	IMPORTANCIA	SUBSISTEMA
RF-1	-	Alta	SUB-1
RF-2	RF-1	Alta	SUB-2
RF-3	RF-2	Alta	SUB-2
RF-4	RF-3	Alta	SUB-2
RF-5	RF-1	Media	SUB-2
RF-6	RF-5	Media	SUB-2
RF-7	RF-1	Alta	SUB-3
RF-8	RF-7, RF-10	Alta	SUB-3
RF-9	RF-8	Alta	SUB-3
RF-10	RF-1, RF-8	Alta	SUB-3
RF-11	RF-10	Media	SUB-3
RF-12	RF-3, RF-10	Alta	SUB-3
RF-13	RF-1	Alta	SUB-4
RF-14	RF-1	Alta	SUB-5
RF-15	RF-3, RF-6, RF-8, RF-14	Alta	SUB-5
RF-16			
RF-17			

Tabla 3.13: Tabla de Requisitos funcionales (II)

3.2.2. Requisitos no funcionales del sistema

ID	NOMBRE	IMPORTANCIA
RR-1	Independencia entre módulos.	Media
RR-2	Interfaz atractiva y usable.	Alta
RR-3	Integración con las Recomendaciones de la Plataforma G-EducaInflamatoria de GETECCU.	Media
RR-4	El usuario conocerá cómo se usan sus datos médicos en la aplicación.	Alta

Tabla 3.14: Tabla de Requisitos no funcionales

3.3. Actores

ACT-1	USUARIO
Descripción	Representa al usuario de la aplicación.
Comentarios	Este usuario es quien accederá a la aplicación y a quien corresponderán los datos de salud de la base de datos.

Tabla 3.15: Tabla Actores del Sistema

Capítulo 4

Diseño

En este capítulo se concretará el diseño del sistema. Para ello, se comenzará con una justificación acerca de la arquitectura del sistema seleccionado, así como sus correspondientes diagramas de arquitectura y de clases.

4.1. Arquitectura escogida

Se ha optado por implementar el patrón MVVM (*Model - View - ViewModel*), al ser el que mejor se adecuía a la funcionalidad requerida, siendo el patrón de referencia en el contexto *Android* [11]. En él, se separa la lógica de presentación de datos de la lógica de negocio. Este cuenta con los siguientes 3 componentes principales:

- *Model*: Responsable de representar y almacenar la información.
- *View*: Responsable de manejar los componentes de la Interfaz de Usuario (UI). Es quien informa al *ViewModel* de las acciones del usuario.
- *ViewModel*: Responsable de manejar la información recibida del Modelo y pasarla a la Vista, actuando como enlace entre ambas capas.

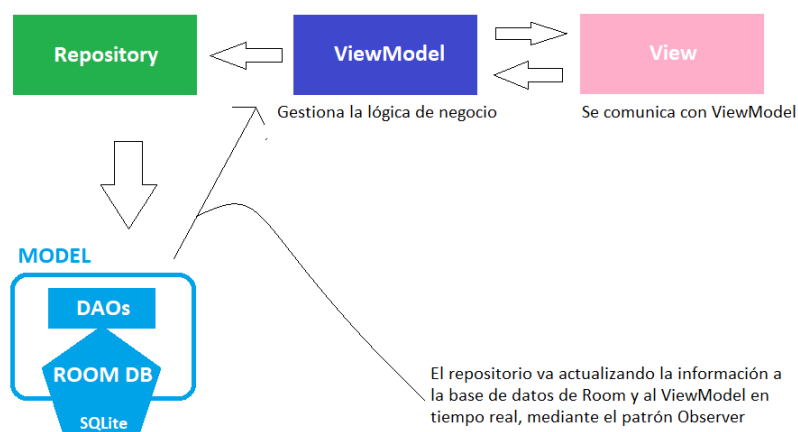


Figura 4.1: Modelo MVVM

Tal y como se puede percibir en la anterior imagen, en el Modelo se hará uso de *Room* [12]. *Room* es una librería de base de datos creada por el equipo de *Android*, que simplifica la tarea de trabajar con bases de datos en *Android* al estar ya integrada en el sistema, abstrayendo el uso de *SQLite* (un sistema de gestión de bases de datos relacional) al implementar una capa intermedia entre esta base de datos y el resto de la aplicación, haciendo uso de un DAO (un componente de software que proporciona una interfaz común entre la aplicación y *Room*, en este caso) por cada tipo de dato (especificado más adelante, en el diagrama de clases).

Además, se emplea el patrón *Observer*, en donde un objeto mantiene una lista con sus dependientes (los observadores) y les notifica en tiempo real cualquier cambio que ocurra, normalmente al llamar a uno de sus métodos. Esto nos permite actualizar la interfaz en tiempo de ejecución, mejorando así la visualización de datos en la Vista.

Una aplicación de *Android* consta de varios componentes, en donde cada tipo tiene un fin y ciclo de vida específico:

- **Actividades:** componente con el que el usuario interacciona con la aplicación, quien crea una ventana para mostrar los componentes de la interfaz.
- **Fragmentos:** una porción de la interfaz mostrada en una actividad. Se pueden combinar varios fragmentos en una sola actividad.
- **Servicios:** puntos de entrada para mantener la ejecución de la aplicación en segundo plano.
- **Proveedor de contenido:** administra un conjunto de datos de la aplicación que se almacena, en este caso, en *Room*.

Mediante la arquitectura MVVM, se consigue una serie de ventajas en el desarrollo y mantenimiento de la aplicación, tales como:

- Escalable y mantenible en el tiempo.
- Más fácil de probar y más ligera.
- Separación de problemas, modularización, sólida.

4.2. Diagrama de clases

A continuación, se incluye la relación existente en el sistema entre las diversas entidades y los atributos de las clases propias del Modelo. En cada Sprint, a medida que se vaya diseñando y desarrollando cada clase del Modelo, se especificarán sus atributos con más detalle.

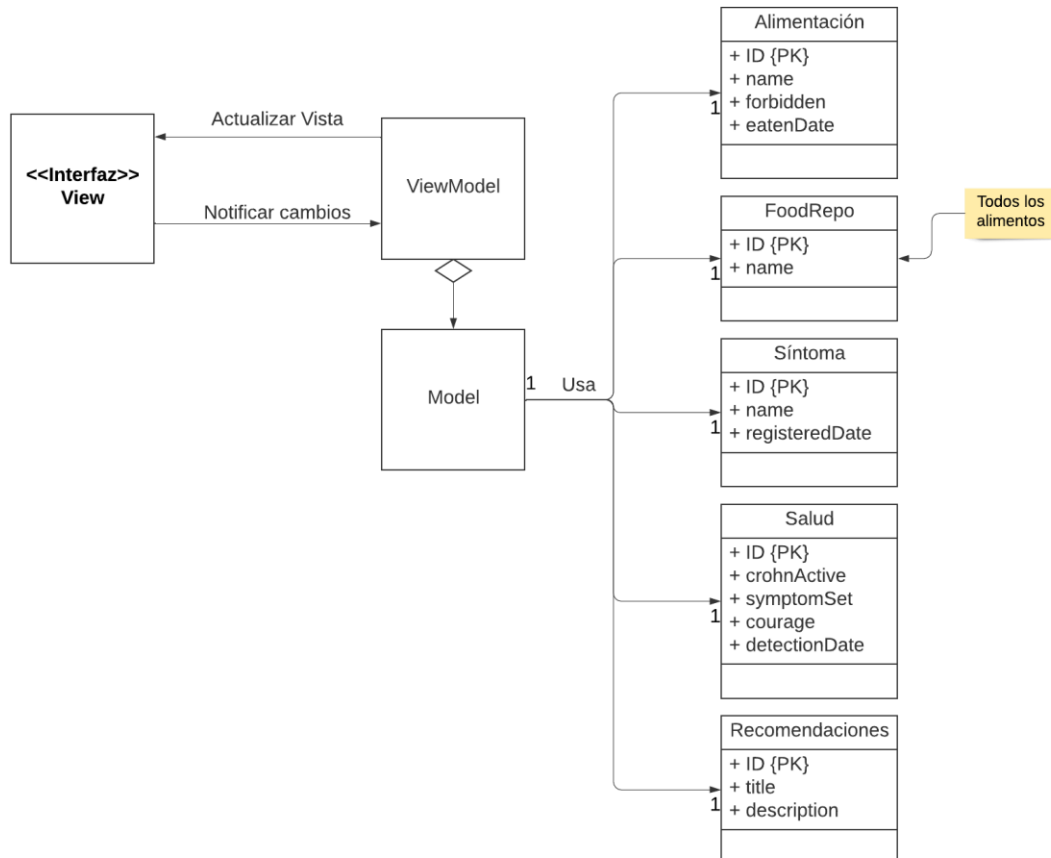


Figura 4.2: Diagrama de clases

4.3. Diagrama de diseño del sistema

Una vez especificadas las clases, se ha realizado el diagrama de la arquitectura del sistema, en donde se pueden ver las diferentes relaciones entre los componentes de este.

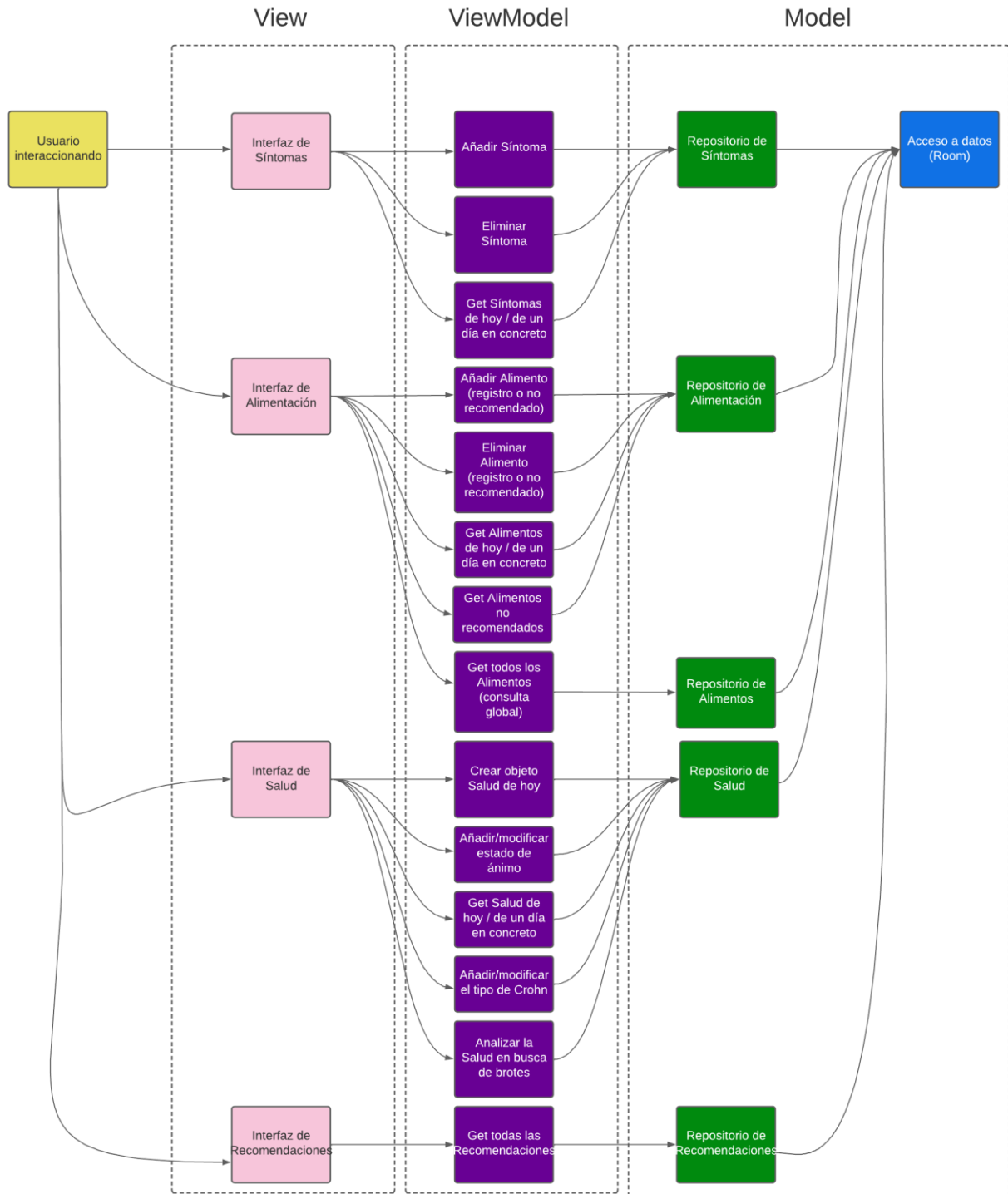


Figura 4.3: Diagrama de diseño del sistema

4.4. Análisis de tecnologías

Una vez realizado el diseño, es necesario confeccionar un análisis de tecnologías para usar en la implementación del proyecto, conociendo así las implicaciones de las diversas tecnologías en el desarrollo. Para ello, es necesario tener en cuenta una serie de criterios en base a los requisitos preestablecidos:

- Seguridad de los datos: no deben intervenir en los datos del usuario.
- Desacoplamiento: de cara a futuras implementaciones, es necesario que estas tecnologías permitan una fuerte independencia entre ellas por su futura actualización.
- Escalabilidad: la aplicación puede ser usada por una gran cantidad de usuarios, debe estar preparada para ello.

4.4.1. Almacenamiento de datos

Los datos de los usuarios deberán estar almacenados en sus dispositivos. Para su almacenamiento, tal y como se detalló, se hará uso de *Room*. De esta forma, se ha escogido *Room* y no otra base de datos debido a su tiempo de compilación de las consultas en *SQL*, junto con las anotaciones que permiten minimizar código y la capacidad de tener rutas de migración de bases de datos optimizadas, haciendo de *Room* una base de datos ideal para este desarrollo.

Además, para variables de la aplicación tales como el nombre del usuario, el tipo de la enfermedad y otros, se hará uso de *SharedPreferences* [13], una colección de preferencias de *Android* en donde se pueden almacenar datos de la forma clave-valor (evitando así tener que guardar en la base de datos estos elementos).

4.4.2. Base de datos de alimentos

Para poder mostrar alimentos al usuario para su registro, es necesario contar con una base de datos de alimentos. De estas bases de datos, únicamente se requieren nombres de los alimentos, no siendo necesario contar con la información nutricional de estos. Las bases de datos de alimentos con las cuales se ha contado son:

Nombre	Gratuidad	Actualizaciones	Propiedad
BEDCA [14]	Uso personal o investigador	Con frecuencia	Ministerio de Sanidad
BADALI [15]	Uso personal o investigador	Con frecuencia	UMH
EDAMAM [16]	Sí, pero dependiendo del número de usuarios y por cierto tiempo	Con frecuencia	EDAMAM
NUBENTOS [17]	No	Basada en la BD de EDAMAM	NUBENTOS

Tabla 4.1: Bases de Datos disponibles

En la tabla anterior se pueden ver las diferentes características de las bases de datos. Ya que únicamente se requieren nombres de alimentos, las mejores opciones son, sin duda, *BEDCA* y *BADALI*. Se ha decidido usar *BEDCA*, debido a que sus requisitos de uso se cumplen: se va a realizar un uso de la base de datos con un fin educacional y de ayuda al paciente.

Para ello, es necesario pedir autorización a los responsables de *BEDCA* para usar su base de datos y que acepten el uso de su base de datos en la aplicación. Al llegar al desarrollo del Sprint 2, no se obtuvo respuesta por parte de *BEDCA* ni de *BADALI* (las opciones existentes gratuitas), por lo que se optó por la creación de una tabla de alimentos en la base de datos. Se entrará más en detalle sobre esto en el apartado de diseño del Sprint 2 (*ver apartado 5.2.2 del Capítulo de Desarrollo y Pruebas*).

4.4.3. Tecnologías de desarrollo de sistemas expertos

En base a la HU-9 (Alerta de brotes), se ha optado por elegir un buen sistema experto basado en reglas para *Android*:

Nombre	Eficiencia móvil	Gratuidad	Actualizaciones
CLIPS [18]	Usa Rete	Sí	Con frecuencia
DROOLS [19]	Usa JESS, mejor preparado para Java	Sí	Con frecuencia

Tabla 4.2: Tecnologías de Desarrollo de Sistemas Expertos en Android

En la tabla anterior se pueden observar las diferentes características de los sistemas expertos seleccionados. Dado que el *fork* de *CLIPS* para *Android* lleva sin actualizarse desde 2014, *Drools* se posiciona como el mejor sistema de gestión de reglas a usar en la aplicación.

No obstante, *Android* emplea dependencias basadas en *Gradle*, mientras que *Drools* utiliza dependencias que solamente se encontraron en proyectos *Maven*, haciendo que determinadas clases necesarias para el empleo de *Drools* no se encontrasen disponibles para su integración en la aplicación. Por ello, no se ha desarrollado el sistema experto mediante ninguno de estos sistemas y se ha optado por la creación de un algoritmo que supla esta funcionalidad. Se entrará más en detalle sobre esto en el apartado de diseño del Sprint 4 (*ver apartado 5.4.2 del Capítulo de Desarrollo y Pruebas*).

Capítulo 5

Desarrollo y pruebas

En este capítulo se concretará el desarrollo y las pruebas del sistema. Para ello, se especificarán los diferentes Sprints realizados mediante la metodología de trabajo escogida (en este caso, Scrum) junto con las decisiones tomadas respecto a la implementación y/o desarrollo del proyecto, contando con la descripción y el grado de cumplimiento de las pruebas realizadas en cada Sprint según la metodología Scrum. El diseño de pruebas y su realización a lo largo de los Sprints nos van a proporcionar una cobertura de los diferentes requisitos del proyecto.

Otra parte fundamental del proyecto es crear un software usable (*ver Requisito no funcional RR-2 del Capítulo de Análisis*), por lo que al finalizar las descripciones del desarrollo de todos los Sprints se hará una evaluación heurística del sistema mediante los 10 heurísticos de Nielsen [20] y una evaluación de la usabilidad con los usuarios finales.

5.1. Sprint 1

En este primer Sprint, se llevó a cabo la implementación de la interfaz de introducción de la aplicación junto con la ventana principal y el módulo de Sintomatología.

5.1.1. Análisis

En este Sprint, se desarrollaron las Historias de Usuario siguientes: HU-1 (Registro de síntomas), HU-2 (Estado de ánimo).

Se persiguió una interfaz minimalista durante todo el desarrollo (*ver Requisito no funcional RR-2 del Capítulo de Análisis*), adoptando un diseño simple y atractivo. Para ello, se comenzó desarrollando unas ventanas que sirviesen de introducción o guía a lo que la aplicación permite hacer, haciendo uso de estas ventanas para mostrar los Términos de Uso de la aplicación (para así ceñirse a la Ley de Protección de Datos) y preguntar al usuario por su nombre para poder dirigirse a él/ella.

5.1.2. Diseño

La ventana de Inicio de la aplicación se ha diseñado en tarjetas, de modo que se pueda acceder a cada módulo de la aplicación de una forma fácil y rápida.

Se hizo uso de *SharedPreferences* para así guardar la aceptación de usuario de los Términos de Uso y su nombre de usuario. En la base de datos, se almacenan los diversos síntomas registrados por el Usuario. Estos tienen los siguientes campos:

- *ID*: la clave primaria autoincrementada identificativa del síntoma.
- *Name*: el nombre del síntoma.
- *RegisteredDate*: la fecha de registro del síntoma.

El registro de síntomas también está basado en tarjetas, con iconos característicos de cada síntoma y su nombre. Los síntomas existentes para su registro son, en base a lo especificado en el capítulo de Introducción de esta memoria:

- Diarrea
- Dolor abdominal
- Fiebre
- Pérdida de peso
- Dolor anal
- Heridas (perianales o relacionadas)
- Sangrado (anal u otras vías del tracto digestivo)
- Cansancio

Además, el estado de ánimo cuenta con una serie de emojis que representan 5 niveles según la escala de *Likert*. Para ello, se añadió el registro del estado de ánimo en un campo denominado “*courage*” en el objeto de Salud, el cual se expondrá con más detalle en el Sprint 4.

Para la visualización de los registros de síntomas, se hizo uso del widget *RecyclerView* y de una extensión de la clase *Adapter* [21]. De esta forma, mediante el patrón Observador utilizado para recabar información mediante el *ViewModel* (capítulo de diseño), se podrán crear listas dinámicas y así contar con una visualización eficiente de grandes conjuntos de datos, definiendo cómo se muestran mediante el Adaptador (en este caso, se muestran todos los síntomas almacenados con fecha, nombre y un icono de borrado).

5.1.3. Implementación

La implementación de dicho Sprint se ha llevado a cabo sin incidencia alguna, ejecutándose así conforme a la planificación realizada.

A continuación, se muestran unas capturas del resultado final, mostrando una captura de la Guía de Inicio, una de la elección de nombre, de la ventana principal y del módulo de Sintomatología:

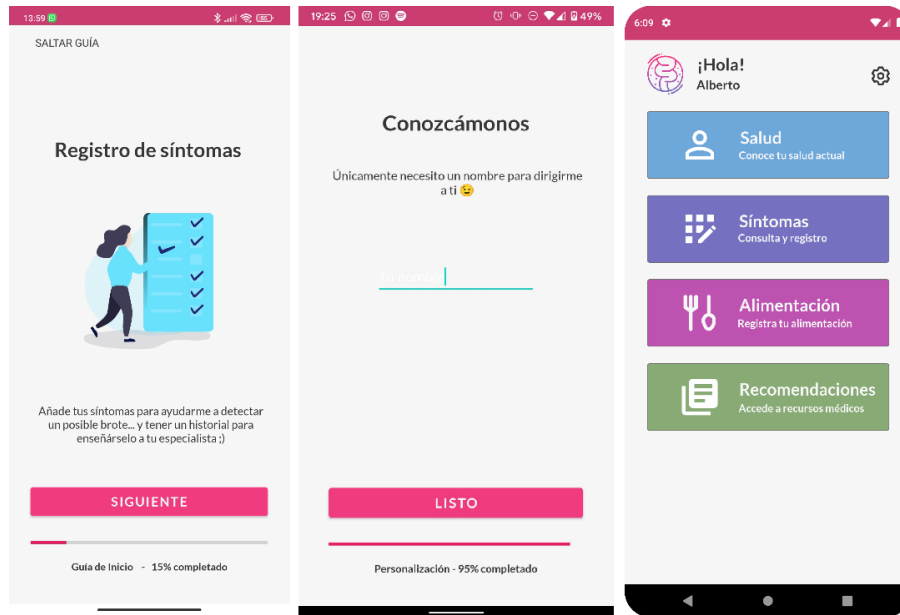


Figura 5.1: Guía de Inicio y Ventana Principal del Sprint 1

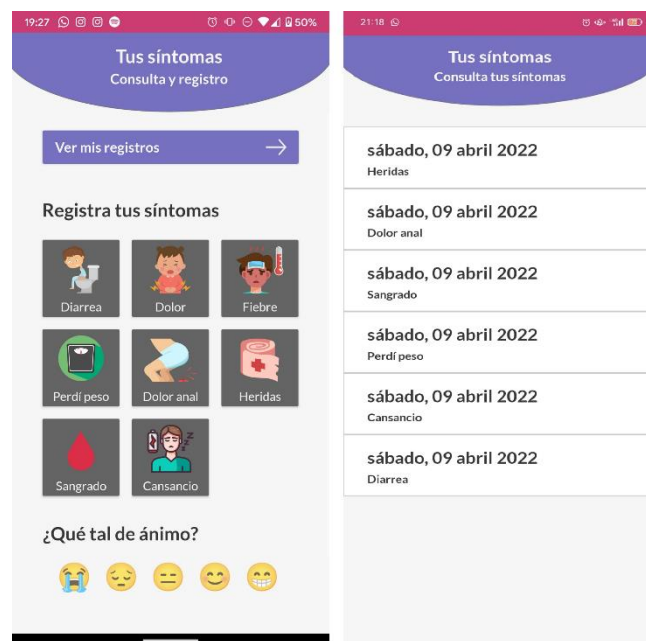


Figura 5.2: Módulo de Sintomatología del Sprint 1

5.1.4. Pruebas realizadas en el Sprint

Se ha hecho uso de las herramientas de desarrollo de *Android Studio* y de dispositivos móviles (junto con el emulador) para probar las funcionalidades implementadas, realizando así las siguientes pruebas:

Guía de la aplicación

- Aceptación de los Términos de Uso: VALIDADO
- No aceptación de los Términos de Uso: ERROR, necesaria su aceptación
- Creación de un nombre de usuario: VALIDADO
- Se deja el campo de usuario por defecto (etiqueta/*hint* del campo): ERROR, el nombre de usuario pasa a ser “Tu nombre”. Se procede a su corrección.
- No se especifica nada en el campo de usuario: VALIDADO, el nombre de usuario pasa a ser “Usuaría/o”.

Ventana de Inicio

- Se muestra el nombre de usuario: VALIDADO.
- Acceso correcto al módulo de Síntomas: VALIDADO
- Distribución en ventana de los elementos gráficos en diversos dispositivos: VALIDADO.

Módulo de Síntomas

- Distribución en ventana de los elementos gráficos en diversos dispositivos: VALIDADO.
- Registro a la vez de síntomas y estado de ánimo: VALIDADO.
- Visualización de los síntomas registrados: ERROR, las tarjetas no muestran del todo bien los datos, por lo que se plantea una modificación en un futuro Sprint.
- Registro de síntomas o estado de ánimo (solamente uno de los dos): ERROR, ya que por cómo se ha hecho es necesario especificar el estado de ánimo a la vez que se registran síntomas. No tiene mucho sentido, al ser un solo estado de ánimo diario.

Mediante los resultados de estas pruebas, se procede a dar el Sprint por finalizado con el cliente, añadiendo al siguiente Sprint aquellas partes que no han sido validadas para su mejora.

5.2. Sprint 2

En este segundo Sprint, se llevó a cabo la implementación del módulo de Alimentación y una mejora en la visualización de los datos mostrados mediante un Calendario.

5.2.1. Análisis

En este Sprint, se desarrollaron las Historias de Usuario siguientes: HU-3 (Registro De alimentos ingeridos), HU-4 (Registro de alimentos no recomendados) y HU-5 (Calendarización de los registros).

Para ello, se realizó una interfaz principal de búsqueda para añadir alimentos al registro, además de una pestaña auxiliar para poder determinar directamente un alimento que el usuario ya sepa que es dañino. Mediante el uso del *widget RecyclerView* y de una extensión de la clase *Adapter* se implementó la vista como en el Adaptador de síntomas, añadiendo a mayores un icono para agregar ese alimento a los alimentos no recomendados.

Además, se optó por incluir en este Sprint la Historia de Usuario HU-5. Para ello, se modificaron los módulos de Sintomatología y Alimentación, mostrando así en su ventana principal un Calendario en donde se puede seleccionar un día en concreto y ver los registros (síntomas, alimentos) de ese día mediante una mejora del Adaptador (se muestran los campos necesarios, la fecha ya no lo es al ser todos de la misma), a la vez que añadir registros para ese día en concreto.

5.2.2. Diseño

Se ha creado en la base de datos un objeto Alimentación con los siguientes campos:

- *ID*: la clave primaria autoincrementada identificativa del alimento.
- *Name*: el nombre del alimento.
- *EatenDate*: la fecha de registro del alimento.
- *Forbidden*: un booleano que indica si el alimento es dañino o no.

Dada una imposibilidad de uso de la base de datos *BEDCA* del Ministerio de Sanidad (ya que se necesitaba permiso explícito suyo para poder usarla y acceder a ella y no ocurrió), se ha decidido hacer en *Room* una nueva tabla denominada *FoodRepo*, con 800 alimentos, escritos en SQL mediante una tabla de alimentos [22].

- *ID*: la clave primaria autoincrementada identificativa del alimento.
- *Name*: el nombre del alimento.

5.2.3. Implementación

La implementación de este Sprint se ha llevado a cabo sin incidencia alguna, ejecutándose así conforme a la planificación realizada.

A continuación, se muestran unas capturas del resultado final, mostrando una captura de la ventana principal del módulo de Alimentación, del registro de alimentos y la modificación de la visualización en el módulo de Sintomatología:

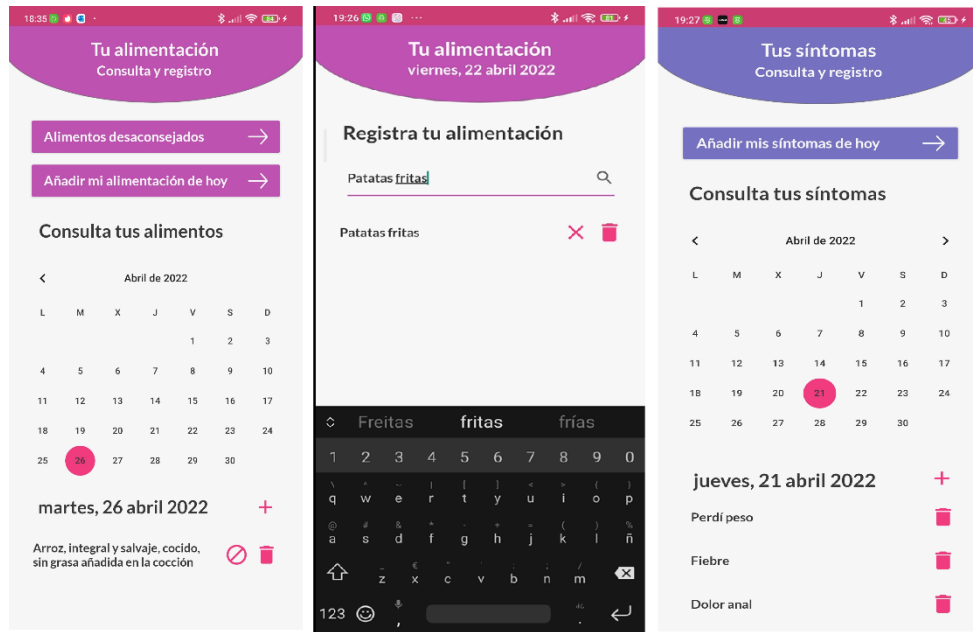


Figura 5.3: Módulo de Alimentos y nueva visualización de síntomas

5.2.4. Pruebas realizadas en el Sprint

En este Sprint se han llevado a cabo las siguientes pruebas mediante el empleo de las herramientas de *Android Studio* y dispositivos móviles:

Módulo de Alimentación

- Búsqueda de cualquier alimento y añadirlo a los registros: VALIDADO. Se decide limpiar el contenido de la barra de búsqueda una vez se seleccione un alimento.
- Búsqueda de un alimento por una parte (ej. Buscar “Pollo” para seleccionar “Alitas de Pollo”): ERROR, la búsqueda tiene que ser completa. Se modifica de modo que se haga un *contains* en vez de un *equalsIgnoreCase*. Se vuelve a ejecutar la prueba, VALIDADO.
- Añadir un alimento registrado a los alimentos no recomendados: VALIDADO. Se decide desactivar el icono de borrado al carecer de sentido en esta pestaña del módulo.

- Quitar un alimento registrado de los alimentos no recomendados: ERROR, al sacarlo se produce un *IndexOutOfBoundsException* al dejar el Adaptador vacío. Se modifica la vista del Adaptador para evitarlo. Se vuelve a ejecutar la prueba, VALIDADO.
- Añadir un alimento directamente a los alimentos no recomendados: VALIDADO.
- Registrar un alimento en un día diferente al actual: VALIDADO.
- Visualización de los alimentos en cualquier día con registros del Calendario: VALIDADO.

Módulo de síntomas

- Registrar un síntoma en un día diferente al actual: VALIDADO.
- Registro del estado de ánimo en un día diferente al actual: VALIDADO.
- Visualización de los síntomas desde el Calendario: VALIDADO.

Mediante los resultados de estas pruebas, se procede a dar el Sprint por finalizado con el cliente.

5.3. Sprint 3

En este tercer Sprint, se llevó a cabo la implementación de la pestaña de ajustes de la aplicación, con su correspondiente ventana de Privacidad, junto con alertas alimenticias, las cuales surgieron al hacer una retrospectiva en el Sprint anterior de funcionalidades que añadirían valor al proyecto.

5.3.1. Análisis

En este Sprint, se desarrollaron las Historias de Usuario siguientes: HU-6 (Privacidad del Usuario), HU-7 (Alerta de alimentos no recomendados).

Para comenzar, se desarrolló una sección en donde se explica de un modo amigable y sencillo el tratado de sus datos. La Política de Privacidad al completo se puede ver en la Tienda de aplicaciones de *Google, Play Store*, al ser necesaria adjuntarla.

Esta sección ha sido incluida dentro de la pestaña de ajustes, en donde el usuario también puede modificar su nombre y decidir si quiere recibir o no notificaciones *Push*, junto con modificar otras variables de la aplicación. En los ajustes también se incluirá una opción para borrar todos los datos almacenados.

Los datos de los usuarios están almacenados en *Room*, lo que permite que, aun estando en su propio dispositivo, no se vean expuestos al almacenar los datos de la base de datos y los archivos de la aplicación en un directorio privado.

Por otra parte, se ha incluido un algoritmo que permita una estimación de un alimento no recomendado para el usuario, siguiendo con la HU-7 (Alerta de alimentos no recomendados).

5.3.2. Diseño

Básicamente, se hará uso de *SharedPreferences* para así poder personalizar diferentes variables en los ajustes de la aplicación.

El algoritmo de alimentos no recomendados se realizará estudiando la correlación entre un alimento y los síntomas registrados para ese día, encontrándonos con una alerta vía notificación *Push* en caso de que ese alimento pudiese haber provocado esos síntomas durante varias veces (por defecto, se comprueba que exista una coincidencia entre los síntomas y el alimento en un total de 5 días con esos registros).

5.3.3. Implementación

La implementación de dicho Sprint ha concluido sin incidencia alguna, ejecutándose así conforme a la planificación realizada.

A continuación, se muestran unas capturas del resultado final, mostrando los ajustes, la ventana de Privacidad y las alertas de alimentos no recomendados:

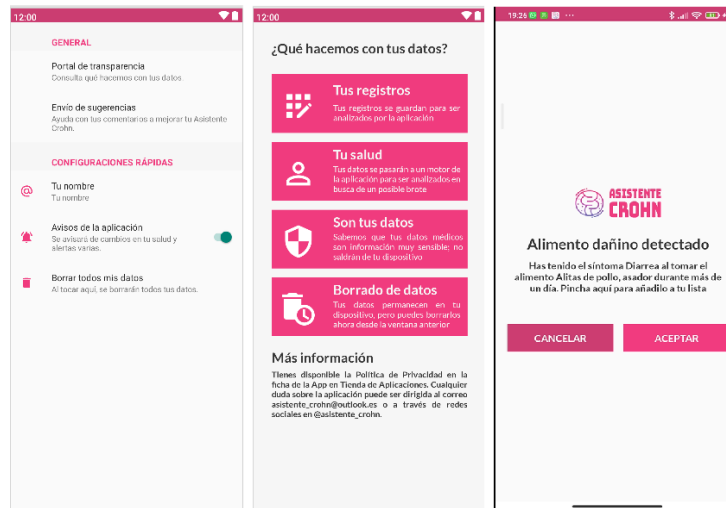


Figura 5.4: Ajustes, Privacidad y Alertas del Sprint 3

5.3.4. Pruebas realizadas en el Sprint

Se han ejecutado las siguientes pruebas:

Ajustes y Privacidad

- Acceso a la sección de Privacidad: VALIDADO
- Edición del nombre de usuario: VALIDADO
- Activación/desactivación de las alertas: VALIDADO

Alerta de alimentos no recomendados

- Comprobación de alertas antes de enviar notificaciones: VALIDADO
- Detección del alimento “Naranja” dañino al registrarlo junto con el síntoma “Diarrea”: VALIDADO
- Se indica que el alimento no es dañino: ERROR, la aplicación lo sigue mostrando, se propone añadir un array de alimentos que el usuario indica que no hacen daño. Se vuelve a ejecutar la prueba, VALIDADO.
- Se indica que el alimento hace daño: VALIDADO, se modifica el atributo a *forbidden*=true y se muestra en los alimentos no recomendados.

Mediante los resultados de estas pruebas, se procede a dar el Sprint por finalizado con el cliente.

5.4. Sprint 4

En este cuarto Sprint, se llevó a cabo la implementación del módulo de Salud con sus correspondientes vistas y modelado.

5.4.1. Análisis

Se desarrollaron las Historias de Usuario siguientes: HU-8 (tipo de la enfermedad), HU-9 (Alerta de brotes).

Según el tipo de Crohn, existen una serie de síntomas asociados que pueden determinar si el usuario puede estar en un brote de la enfermedad (*ver apartado 1 del Capítulo de Introducción*). Mediante el tipo de Crohn del usuario y los síntomas registrados, es posible estimar el estado de la enfermedad.

La frecuencia de aparición e importancia de cada síntoma no es la misma, de hecho, se ha establecido la escala CDAI (Crohn's Disease Activity Index, CDAI [5]) que permite medir el nivel de actividad de la enfermedad, la cual se ha adaptado según sus características clínicas para que, si en el tipo de la enfermedad se presenta una puntuación superior al 50% de la misma, se considere que para ese día en concreto la enfermedad puede estar ascendiendo a un brote. El tipo de la enfermedad y el número de días a analizar en busca de un brote se podrán modificar en *SharedPreferences* desde la ventana de ajustes de la aplicación, siendo 3 días por defecto en la búsqueda.

5.4.2. Diseño

En este Sprint, se comenzó por realizar un archivo de pruebas en *Drools* para conectarlo a modo de ejemplo a la aplicación, para comprobar su integración con esta al contar con cierta incertidumbre en su uso. Desgraciadamente, la aplicación de *Android* se crea mediante *Gradle*, pero *Drools* utiliza dependencias basadas en *Maven*, no estando así disponibles ciertas clases para su integración. En el análisis de tecnologías (*ver apartado 4.4.3 del Capítulo de Diseño*) se estudió tanto la integración de *Drools* como otros sistemas expertos, pero al revisar la documentación y ejemplos una vez empezado el desarrollo se encontró este problema.

Dado que la otra alternativa de sistema experto también estaba basada en *Maven* (*ver apartado 4.4.3 del Capítulo de Diseño*), se optó por la creación de un algoritmo para la detección temprana de un posible brote de la enfermedad. Es decir, anteriormente se iban a crear las diversas reglas con los tipos de la enfermedad y su Sintomatología en *Drools*, pero estas reglas se implementaron en *Java*.

Para ello, se ha hecho uso de la clase de Salud, con los siguientes campos, en donde se genera un objeto Salud por cada día en el que existan síntomas:

- *ID*: la clave primaria autoincrementada identificativa de la Salud.
- *crohnActive*: booleano que indica si la enfermedad está en brote.
- *Courage*: el estado de ánimo: Un *Integer* de 0-4 según la escala de *Likert*, siendo un 0 el estar desanimado y un 4 estar muy contento.
- *symptomSet*: booleano que indica si se presentan síntomas con una puntuación total mayor o igual a la mitad del umbral.
- *DetectionDate*: la fecha de creación del objeto Salud.

Como parte de la retrospectiva del anterior Sprint, se han añadido más síntomas al módulo de Sintomatología, en donde a continuación se muestra cada síntoma con su multiplicador asociado de cara a su implementación:

- Diarrea: 5 puntos
- Náuseas: 3 puntos
- Sin apetito: 3 puntos
- Dolor abdominal: 5 puntos
- Dolor articular: 4 puntos
- Dolor anal: 4 puntos
- Dolor de cabeza: 1 punto
- Fiebre: 3 puntos
- Cansancio: 3 puntos
- Urgencia por ir al baño: 4 puntos
- Pérdida de peso (mayor a un 10%): 2 puntos
- Sangrado (anal u otras vías del tracto digestivo): 4 puntos
- Aftas: 1 punto
- Heridas (perianales o relacionadas): 3 puntos
- Problema dermatológico: 2 puntos

Una vez realizado cada multiplicador en cada síntoma, los umbrales serían del siguiente modo:

- Ileítis: 21 puntos
- Colitis: 24 puntos
- Íleo-colitis: 29 puntos
- Gastrointestinal alta: 14 puntos
- Perianal: 18 puntos

El algoritmo de detección de brotes es el siguiente. En él, distinguimos una serie de constantes y un método:

- *PREV_DAYS*: el número de días a analizar, establecido a 3 por defecto. El usuario tendrá la opción de modificarlo desde los ajustes.
- *TIPO*: el tipo de la enfermedad (*ver apartado 1 del Capítulo de Introducción*).
- *crohnActiveCheck(días)*: método que comprueba si para el número de días anteriores al actual, la enfermedad se encuentra activa (atributo *crohnActive*).
- *umbral(TIPO)*: el umbral de puntuaciones totales asociadas a cada tipo de la enfermedad, descritos anteriormente.
- Además, cada día *D* está representado mediante una instancia de la clase *Salud*, utilizando así los atributos especificados en la anterior página.

Para cada día D CON REGISTROS de síntomas

Se analizan sus síntomas si $D.crohnActive == false \ \& \ D.symptomSet == false \ \& \ crohnActiveCheck(PREV_DAYs) == false$

1. *Se comprueban los síntomas de D según el TIPO establecido*
2. *Si la puntuación total de los síntomas es mayor o igual a la mitad del umbral(TIPO):*
 - a. *Se notifica que para el día D, $D.symptomSet = true$*
 - b. *Ahora es necesario ver los días anteriores*
 - i. *Si $D.symptomSet = true$ en $PREV_DAYs$ días $\Rightarrow BROTE = true$*
 1. *Se actualizan D y los $PREV_DAYs$ días a $D.crohnActive = true$*
 2. *Notificar al usuario*

Si resulta que $D.crohnActive == false$ o $D.crohnActiveCheck(PREV_DAYs) == TRUE$

1. *Se comprueban los síntomas de los $PREV_DAYs$ días según el TIPO establecido*
2. *Si resulta que la puntuación total de los síntomas sigue siendo mayor o igual al umbral(TIPO):*
 - a. *Se notifica que para el día D, $D.symptomSet = true$*
 - b. *Se actualiza D a $D.crohnActive = true$*
 - c. *Notificar al usuario*

La interfaz gráfica se desarrolló de un modo distinto a los dos módulos anteriores. En ella, aparece nada más entrar el registro del estado de ánimo (el cual se debe mover aquí para ser más coherente, eliminando así el error presenta en el primer Sprint al registrar el estado de ánimo) y una descripción del estado actual de la enfermedad con el tipo establecido por defecto, indicando si ese día se presentan síntomas compatibles con el tipo de enfermedad (el positivo) y también si está en brote.

Además, se permite conocer el estado al ver los registros anteriores en un Calendario en una pestaña auxiliar.

5.4.3. Implementación

La implementación de dicho Sprint ha concluido sin incidencia alguna, ejecutándose así conforme a la planificación realizada.

A continuación, se muestran unas capturas del resultado final, mostrando los nuevos síntomas y el módulo de Salud. De paso, se enseña en modo oscuro:

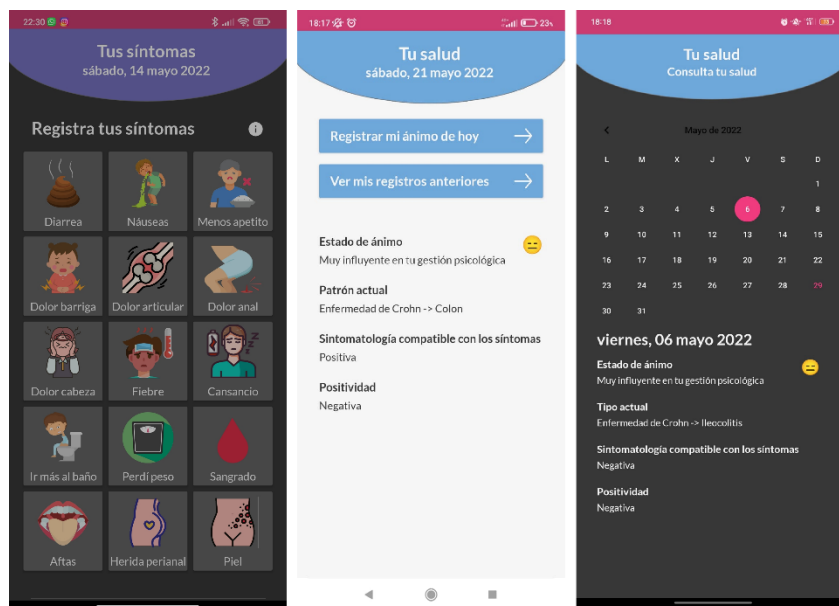


Figura 5.5: Síntomas y Salud del Sprint 4

5.4.4. Pruebas realizadas en el Sprint

Se han realizado las siguientes pruebas:

Estado de ánimo

- Registro correcto de un solo estado de ánimo por día: VALIDADO
- Visualización del estado de ánimo de cada día: VALIDADO

Módulo de síntomas

- Consulta de síntomas correcta después de añadir nuevos síntomas: ERROR, con muchos síntomas no se permite deslizar hacia abajo. Se corrige.
- Registro de síntomas correcto: ERROR 1, al deseleccionar síntomas el botón sigue activado, llevando a confusión. Se opta por activar el botón por defecto y renombrarlo a “Guardar cambios”. Se vuelve a ejecutar la prueba, VALIDADO.
- Registro de síntomas correcto: ERROR 2, al tener muchos síntomas seleccionados y deseleccionar todos menos uno, no lo hace. Se corrige el error al ser culpa de un “break;” dentro de un bucle. Se vuelve a ejecutar la prueba, VALIDADO.

Salud

- Selección del tipo actual de la enfermedad: ERROR, se ha creado la ventana de selección en un OnClickListener, por lo cual nunca se genera. Se corrige y se vuelve a ejecutar la prueba, VALIDADO.
- Modificación del tipo de la enfermedad: VALIDADO
- Cambio de los PREV_DAYS desde ajustes: VALIDADO
- Detección de Sintomatología compatible con el tipo de enfermedad definido: VALIDADO
- Alerta de un posible brote en caso de presentar Sintomatología compatible con el tipo durante PREV_DAYS: VALIDADO
- Fin del brote una vez los PREV_DAYS ya no presentan la Sintomatología: VALIDADO.

5.5. Sprint 5

En este quinto y último Sprint, se llevó a cabo la implementación del módulo de recomendaciones avaladas por sociedades médicas.

5.5.1. Análisis

En este Sprint, se desarrollará la Historia de Usuario siguiente: HU-10 (Recomendaciones médicas).

Para ello, la Plataforma de G-EducaInflamatoria ha aprobado la citación y referencia de su material. Se accede a una primera ventana, la cual clasifica el contenido que tiene la aplicación de la plataforma. Al acceder a cada subsección, esta muestra accesos directos al apartado correspondiente de su web.

5.5.2. Diseño

Se usa el objeto de Recomendaciones en *Room*, en el cual se incluyen los siguientes campos:

- *ID*: la clave primaria autoincrementada identificativa de la Recomendación.
- *Title*: título identificativo de la recomendación.
- *Description*: se almacenará la *URL* de dicha recomendación.

Se embebe cada recomendación en la aplicación mediante *WebView*, identificando cada tipo junto con el propio título. Los tipos de recomendaciones serán:

- Consulta y aprende: recursos médicos informativos.
- Nutrición: tablas nutricionales dependiendo del estado actual de la enfermedad.
- Infografías: recursos gráficos sobre diferentes aspectos de la enfermedad.
- Cuestionarios: acceso a cuestionarios para conocer el nivel de actividad, calidad de vida, causas de abandono de la medicación, niveles de adherencia, etc.

5.5.3. Implementación

La implementación de este Sprint ha concluido sin incidencia alguna, ejecutándose así conforme a la planificación realizada.

A continuación, se muestran unas capturas del resultado final, mostrando una captura de la vista principal del módulo y ejemplos varios:

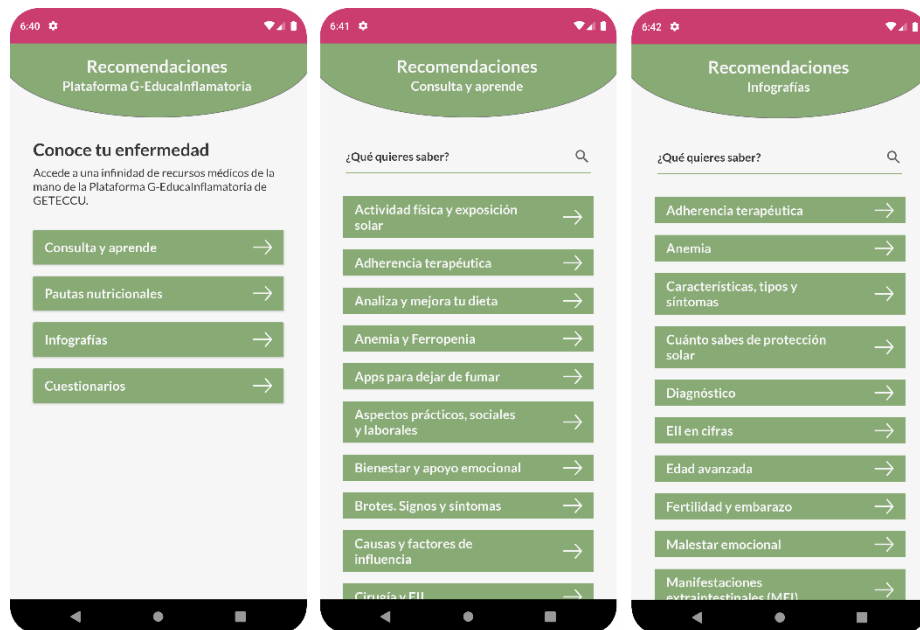


Figura 5.6: Tipos de Recomendaciones del Sprint 5

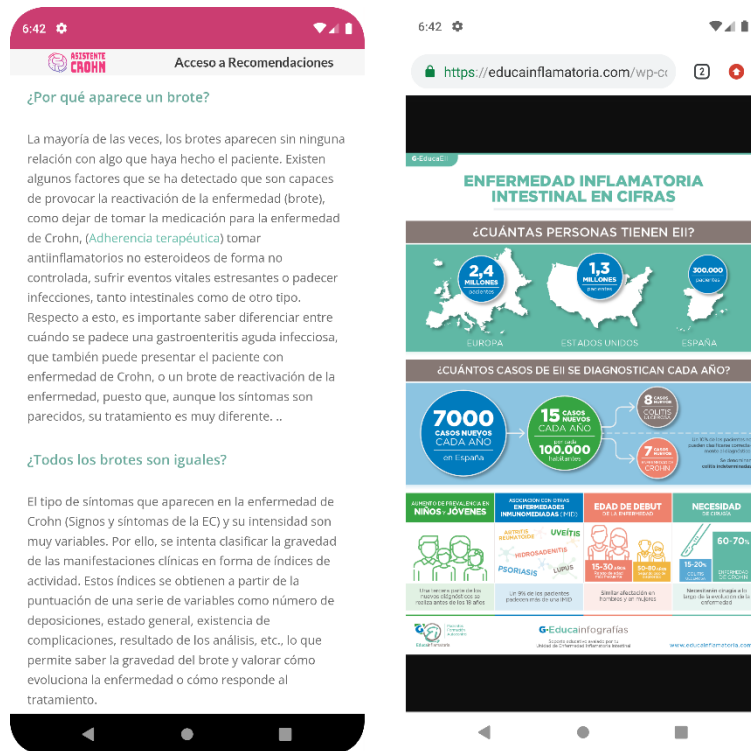


Figura 5.7: Ejemplos de Recomendaciones del Sprint 5

5.5.4. Pruebas realizadas en el Sprint

Se ha hecho uso de las herramientas de desarrollo de *Android Studio* y de dispositivos móviles (junto con el emulador) para probar las funcionalidades implementadas, realizando así las siguientes pruebas:

Recomendaciones

- Acceso a la sección de Consulta y Aprende: VALIDADO
- Acceso a la sección de Nutrición: ERROR, al embeber webs no se muestran las imágenes de modo correcto. Se decide mostrar el navegador en este apartado. Se vuelve a ejecutar la prueba, VALIDADO.
- Acceso a la sección de Infografías: ERROR, al embeber webs no se muestran las imágenes de modo correcto. Se decide abrir el navegador en este apartado. Se vuelve a ejecutar la prueba, VALIDADO.
- Acceso a la sección de Cuestionarios: VALIDADO

5.6. Evaluación de la usabilidad

El objetivo que se persigue mediante este apartado es evaluar la usabilidad de la aplicación una vez desarrollada toda la interfaz. Para ello, se dividirá en 2 secciones: Evaluación heurística del sistema mediante los 10 heurísticos de Nielsen [20] y una evaluación de la usabilidad con usuarios finales.

5.6.1. Evaluación según heurística de Nielsen

Visibilidad del estado del sistema

Esta primera heurística está presente en toda la aplicación, en las cabeceras de cada ventana, indicando mediante un título el módulo en el que se encuentra y un subtítulo con la acción a realizar en dicha pestaña del módulo. Además, cada módulo está identificado con un propio color, ayudando en su comprensión.

Utilizar el lenguaje de los usuarios

El lenguaje utilizado en la aplicación es cercano, sin tecnicismos; esto permite que los términos estén orientados al lenguaje de los usuarios.

Control y libertad para el usuario

La aplicación permite avanzar y retroceder entre las ventanas mediante el lenguaje de gestos de *Android*. Además, el usuario puede crear y eliminar elementos de una manera intuitiva.

Consistencia y estándares

La interfaz cuenta con un diseño común a toda la aplicación, siguiendo unos tamaños de texto, iconos y botones iguales y reconocibles para toda la aplicación. Se utilizan los colores para identificar el módulo actual, además de las acciones a realizar.

Prevención de errores

Se previene que el usuario accione de manera errónea en las diversas ventanas o en la introducción de datos. Además, en caso de surgir algún problema, se indica el error de un modo claro y sencillo.

Minimizar la carga de memoria del usuario

El lenguaje de diseño de la aplicación permite que el usuario sepa en todo momento qué está haciendo o cómo hacer la acción que desea. Para la consulta de

datos, se usa el Calendario en las interfaces requeridas, contando con una guía de los síntomas para saber qué significa cada uno.

Para acceder a cada módulo, basta con ir a la ventana principal, donde también se accede a los ajustes.

Flexibilidad y eficiencia de uso

La aplicación siempre se abre en la ventana principal. Una vez allí, el usuario selecciona el módulo al que ir, en donde puede registrar de una manera sencilla en todas las ventanas (exactamente los mismos pasos en todas las ventanas), ayudando así en su comprensión y navegación.

Los diálogos estéticos y diseño minimalista

Las ventanas y diálogos muestran únicamente aquella información relevante para su comprensión, eliminando redundancias y permitiendo así un diseño minimalista.

Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores

Las diferentes ventanas se actualizan una vez se modifican datos. Además, en caso de surgir algún error, se indica de una manera clara lo que ha sucedido.

Ayuda y documentación

La aplicación es intuitiva en cuanto a su uso, pero se incluye una guía de uso en Anexo B de esta memoria.

5.6.2. Evaluación de la usabilidad con usuarios finales

Una vez finalizado el desarrollo de la aplicación, y teniendo en cuenta que una de las restricciones del proyecto era su publicación, se ha procedido a lanzarla en la tienda de *Google, Play Store*, bajo el nombre de “Asistente Crohn” [23].

Una vez publicada, se ha elaborado un cuestionario de usabilidad [24]. Este cuestionario cuenta con una serie de preguntas sobre la escala de Likert (de 1-5), las cuales se pueden observar a continuación junto con el porcentaje de respuestas.

Este cuestionario fue realizado a pacientes que cuentan con la enfermedad y

CAPÍTULO 5. DESARROLLO Y PRUEBAS

que descargaron la aplicación, obteniendo 11 respuestas en el plazo de 24h en el que se dejó activo el formulario.

Pregunta / Valor	1	2	3	4	5
La aplicación me permite realizar las tareas de una forma fácil				45.5%	54.5%
La navegación entre las opciones resulta clara			9.1%	45.5%	45.5%
Sé en todo momento en qué parte de la aplicación me encuentro				27.3%	72.7%
La apariencia de la aplicación es agradable				45.5%	54.5%
Las imágenes en iconos, síntomas son representativas			9.1%	27.3%	63.3%
La interfaz de la aplicación está bien estructurada			18.2%	18.2%	63.6%
Los nombres de las opciones son claros y representativos			9.1%	54.5%	36.4%
La aplicación es fácil de usar				63.6%	36.4%
Me gustaría usar la aplicación a menudo para mantener un seguimiento de mi EII				45.5%	54.5%
Puedo registrar un síntoma o Alimento de una forma sencilla	9.1%			36.4%	54.5%
Sé cómo añadir un alimento que me hace daño		9.1%		54.5%	36.4%
La información de la enfermedad que veo en la aplicación es correcta y está actualizada			27.3%	45.5%	27.3%
Si quiero consultar información, es sencilla su búsqueda			9.1%	45.5%	45.5%
Estoy satisfecha/o con la aplicación			9.1%	54.5%	36.4%
La aplicación es útil para pacientes con Enf. de Crohn			9.1%	36.4%	54.5%

Tabla 5.1: Resultados del cuestionario de Usabilidad

En base a los resultados, podemos concluir que la aplicación satisface a los usuarios finales.

Capítulo 6

Conclusiones y ampliaciones

En este capítulo se especificará el grado de cumplimiento de los objetivos del proyecto y se discutirán posibles ampliaciones futuras de este.

Este Trabajo de Fin de Grado contaba con un único objetivo, desglosado en diferentes objetivos secundarios: el desarrollo de una aplicación *Android* que proporcione al paciente una ayuda en la gestión de la Enfermedad de Crohn. En base a la evaluación realizada, se puede afirmar que el objetivo se ha cumplido exitosamente.

El primer subobjetivo era el Identificar y alertar de manera automática respecto a la aparición de un posible brote de la enfermedad de Crohn a partir de ciertos patrones de síntomas referidos por los especialistas. Se ha implementado la funcionalidad de registro de síntomas y un algoritmo que permite la identificación de los diversos conjuntos de síntomas dentro de cada tipo de enfermedad: Ileítis, Colitis, Íleo-colitis, Gastrointestinal alta y Perianal.

El segundo subobjetivo era el proporcionar al usuario un acceso orientativo a recomendaciones de referencia sobre la gestión de la enfermedad, avaladas por sociedades médicas. Este subobjetivo se ha cumplido, proporcionando una interfaz de acceso a la Plataforma G-EducaInflamatoria tal y como se había previsto.

El último subobjetivo era el proporcionar al usuario capacidades de autorregistro de la enfermedad, con especial atención a su alimentación, con el fin de servir de orientación sobre el origen de cada brote. Este subobjetivo también se cumplió con creces. El inconveniente surgido durante el desarrollo debido a la no obtención de respuesta por parte de *BEDCA* para su uso en la aplicación no supuso ningún cambio en la aplicación: únicamente se necesitaban nombres de alimentos, pero en menos de una tarde se creó una base de datos de 800 alimentos sin intervenir en la planificación del Sprint.

Se ha seguido una metodología ágil (Scrum) para el desarrollo de este proyecto, permitiendo la ejecución de distintos cambios en los requisitos a lo largo del proyecto. La ejecución de un desarrollo incremental ha permitido priorizar determinados requisitos del sistema y llevar a cabo las modificaciones pertinentes conforme progresaba el desarrollo.

La adopción del patrón MVVM ha permitido que la aplicación sea sólida y que pueda ser escalable y mantenible en el tiempo, permitiendo así el poder añadir nuevos módulos de una forma sencilla sin afectar a los datos existentes de los usuarios.

Mediante la posibilidad que ofrece *Room* de migrar datos al actualizar la base de datos, bien sea por añadir una nueva tabla o por añadir nuevos datos precargados como la tabla de alimentos o las recomendaciones, la fiabilidad de la aplicación no se verá comprometida.

Gracias a este Trabajo de Fin de Grado, he podido aprender a desarrollar aplicaciones en *Android*, realizando así un aprendizaje complementario a lo aprendido durante el Grado en Ingeniería Informática. De él, se han puesto en práctica asignaturas tales como Programación Orientada a Objetos, Bases de Datos, Algoritmos y Estructuras de Datos, Ingeniería de Software, asignaturas de Diseño Web, Ingeniería de Servicios y Gestión de Proyectos, realizando así la construcción desde cero de un software siguiendo un proceso de ingeniería informática mediante la metodología Scrum.

A una semana de la puesta en producción en la tienda de aplicaciones de *Google* [23], cuenta con 50 usuarios activos. La iniciativa promovida mediante este Trabajo de Fin de Grado se ha hecho eco de diversas asociaciones:

- UNICEF España
- GETECCU (Grupo Español de Trabajo en Enfermedad de Crohn y Colitis Ulcerosa)
- FEAD (Fundación Española del Aparato Digestivo)
- SEPD (Sociedad Española de Patología Digestiva)
- ACCU España (Asociación de enfermos de Crohn y Colitis Ulcerosa)
- ASSEII (Asociación Socio Sanitaria de Enfermedades Inflamatorias Intestinales)

Además, varios hospitales, entre ellos el Hospital Victoria Eugenia Cruz Roja de Sevilla y las Unidades de Enfermedad Inflamatoria Intestinal de Pontevedra, Vigo y Santiago de Compostela, las cuales se postularon para una colaboración futura con sus pacientes y el uso de la herramienta en el seguimiento de las EII.

Bien es cierto que, contando con más tiempo, el proyecto podría haber contado con más características. Para empezar, podría haberse realizado un desarrollo utilizando el framework de *Ionic* en vez de *Android Studio*, contando así con una aplicación multiplataforma. Por desconocimiento de dicho framework, se planteó el uso de Java y *Android Studio*.

De cara al futuro, este proyecto puede verse mejorado con más características tales como:

- Gestión psicológica avanzada de la enfermedad, con consejos y formas de actuar en caso de que el usuario lo necesite.
- Recordatorio de medicación y citas médicas: en la mayoría de los estudios publicados, se describe una mala adherencia terapéutica en un 30-45%, llegando a veces hasta un 70% en pacientes con EII.
- Ciclo menstrual con Crohn: ya que puede afectar el ciclo menstrual normal debido a los cambios en niveles hormonales, siendo posible que los periodos sean más dolorosos, lleguen con más o menos frecuencia o incluso nunca.
- Registro de deposiciones: la cantidad y características de las deposiciones según la Escala de Heces de Bristol, la cual permite identificar posibles problemas derivados del tipo de heces.
- Un foro de pacientes: en donde puedan compartir experiencias, hacer preguntas a especialistas. Esto llevaría a un cambio en la arquitectura de la aplicación, seguramente una arquitectura REST. Este cambio se podría realizar sin ningún tipo de problema al tener ya desacoplada la interfaz de la lógica y el modelo gracias al patrón MVVM.
- Certificar la aplicación como una aplicación médica: sería necesario implementar el protocolo HL7 [25], además de más medidas de seguridad de datos. Este protocolo sería implementado a la vez que la arquitectura REST, garantizando así la seguridad de los datos de los usuarios.
- Nuevos idiomas: la aplicación *Android* se ha realizado de forma que únicamente es necesario modificar un archivo XML para añadir soporte a más idiomas.
- Comunidad de desarrollo: debido al interés suscitado por la aplicación, esta podría verse evolucionada por una comunidad de desarrolladores.
- Estudios clínicos y colaboraciones médicas: implementando HL7, REST y certificando la aplicación, se podrían llevar a cabo estudios clínicos con las Unidades EII de diversos hospitales como las ya postuladas.

Bibliografía

- [1] Llamas, Tania Fernández, Marín, Ana Victoria García y Ródenas, Gonzalo Antón, «Enfermedad inflamatoria intestinal,» de *Medicina y cirugía del aparato digestivo. Capítulo 21, 139-147*, Elsevier.
- [2] R. E. Stein y R. N. Baldassano, «Enfermedad inflamatoria intestinal,» de *Nelson. Tratado de pediatría, Capítulo 362, 1973-1981*, Elsevier.
- [3] «Educa Inflamatoria,» [En línea]. Available: <https://educainflamatoria.com/category/enfermedad-crohn/>. [Último acceso: 07 02 2022].
- [4] R. A. Uribe Olivares, «Síndromes colónicos,» de *Fisiopatología. La ciencia del porqué y el cómo. Capítulo 34, 357-369*, Elsevier.
- [5] J. Panés Díaz, «Enfermedad inflamatoria del intestino,» de *Farreras Rozman. Medicina Interna*, Elsevier.
- [6] «La UVigo desarrolla una "app" para mejorar la calidad de vida de los pacientes con Crohn y Colitis ulcerosa,» *Faro de Vigo*, 20 05 2022. [En línea]. Available: <https://www.farodevigo.es/gran-vigo/2022/05/20/universidad-vigo-desarrolla-app-mejorar-66290026.html>.
- [7] E. T. S. d. E. (USC), «Reglamento do Traballo de Fin de Grao en Enxeñaría Informática,» 2020.
- [8] «The 2020 Scrum Guide TM,» 01 01 2020. [En línea]. Available: <https://scrumguides.org/scrum-guide.html>.
- [9] «Notion. Seguimiento de Scrum,» 01 03 2022. [En línea]. Available: <https://www.notion.so/ASISTENTE-CROHN-APP-ed35ae0d516b470d96d067099354fba6>.
- [10] «GitHub Repository,» 01 03 2022. [En línea]. Available: https://github.com/albertoruibalojea/Asistente_Crohn.
- [11] «Guía de arquitectura de Android,» [En línea]. Available: https://developer.android.com/topic/architecture#fetching_data. [Último acceso: 31 05 2022].
- [12] «Android. Room Database,» [En línea]. Available: <https://developer.android.com/topic/libraries/architecture/room?hl=es-419>. [Último acceso: 31 05 2022].

- [13] «Android. SharedPreferences,» [En línea]. Available: <https://developer.android.com/reference/android/content/SharedPreferences>. [Último acceso: 31 05 2022].
- [14] «BEDCA Database,» [En línea]. Available: <https://www.bedca.net/>. [Último acceso: 31 05 2022].
- [15] «BADALI Database,» [En línea]. Available: <https://badali.umh.es/>. [Último acceso: 31 05 2022].
- [16] «EDAMAM Database,» [En línea]. Available: <https://api.edamam.com/>. [Último acceso: 31 05 2022].
- [17] «NUBENTOS,» [En línea]. Available: <https://www.nubentos.com/es/api/fooddatabaseapi/>. [Último acceso: 31 05 2022].
- [18] «CLIPS for Android,» [En línea]. Available: <https://github.com/gomezgoiri/CLIPS4Android>. [Último acceso: 31 05 2022].
- [19] «DROOLS,» [En línea]. Available: <https://www.drools.org/>. [Último acceso: 31 05 2022].
- [20] «The 10 Nielsen heuristics,» [En línea]. Available: <https://medium.com/zerotohero/the-10-nielsen-heuristics-d2797a4e5d5e>. [Último acceso: 31 05 2022].
- [21] «Android. RecyclerView and Adapter,» [En línea]. Available: <https://developer.android.com/guide/topics/ui/layout/recyclerview>. [Último acceso: 31 05 2022].
- [22] «Tabla de alimentos,» [En línea]. Available: <https://medicinainformacion.com/actualizacion-tabla>. [Último acceso: 31 05 2022].
- [23] «Play Store. Asistente Crohn,» 01 06 2022. [En línea]. Available: https://play.google.com/store/apps/details?id=com.aro.asistente_crohn.
- [24] «Resultados del formulario de Usabilidad,» 01 06 2022. [En línea]. Available: <https://forms.office.com/Pages/AnalysisPage.aspx?AnalyzerToken=BNHj2ZsOOXWFi8gYM6FLbTGbll6HHHqJ&id=LEUNj6S3ZEm4EIw5c3RHe1EC3t-2yVJHhCin4QeIiMZUM0tFTjNURDNCSkQzMTRBSEUzN01ROUMwOC4u>.
- [25] «HL7 Protocol,» [En línea]. Available: <https://es.wikipedia.org/wiki/HL7>. [Último acceso: 31 05 2022].
- [26] «DB Browser for SQLite,» [En línea]. Available: <https://sqlitebrowser.org/dl/>. [Último acceso: 31 05 2022].

Apéndice A

Manual Técnico

En este Capítulo se ofrecerá un manual de instalación de la aplicación para así permitir su despliegue y posible desarrollo.

A.1. Software necesario

Se requiere *Android Studio* para poder desplegar y desarrollar el proyecto. Al instalarlo, se instala junto con este el *SDK* necesario y *Gradle*. Además, está desarrollado bajo la *API 28 de Android* (el nombre de versión es *Android 9 Pie*).

Además, en caso de querer acceder a la base de datos, se necesitará el software *DB Browser for SQLite* [26]. Se adjuntan como código fuente los scripts *SQL* utilizados para su creación, bien no son necesarios al contar con el propio archivo *database.db* propio de *DB Browser for SQLite*.

Cabe destacar que, por seguridad, se ha establecido una clave de firma de la aplicación, por lo que únicamente puede ser desplegada en local o en un dispositivo móvil vía ADB, no siendo posible su actualización u conexión vía Play Store.

A.2. Estructura de ficheros

A continuación, se incluye la estructura de ficheros del proyecto. Esta estructura ha seguido el patrón MVVM comentado en el capítulo de diseño.

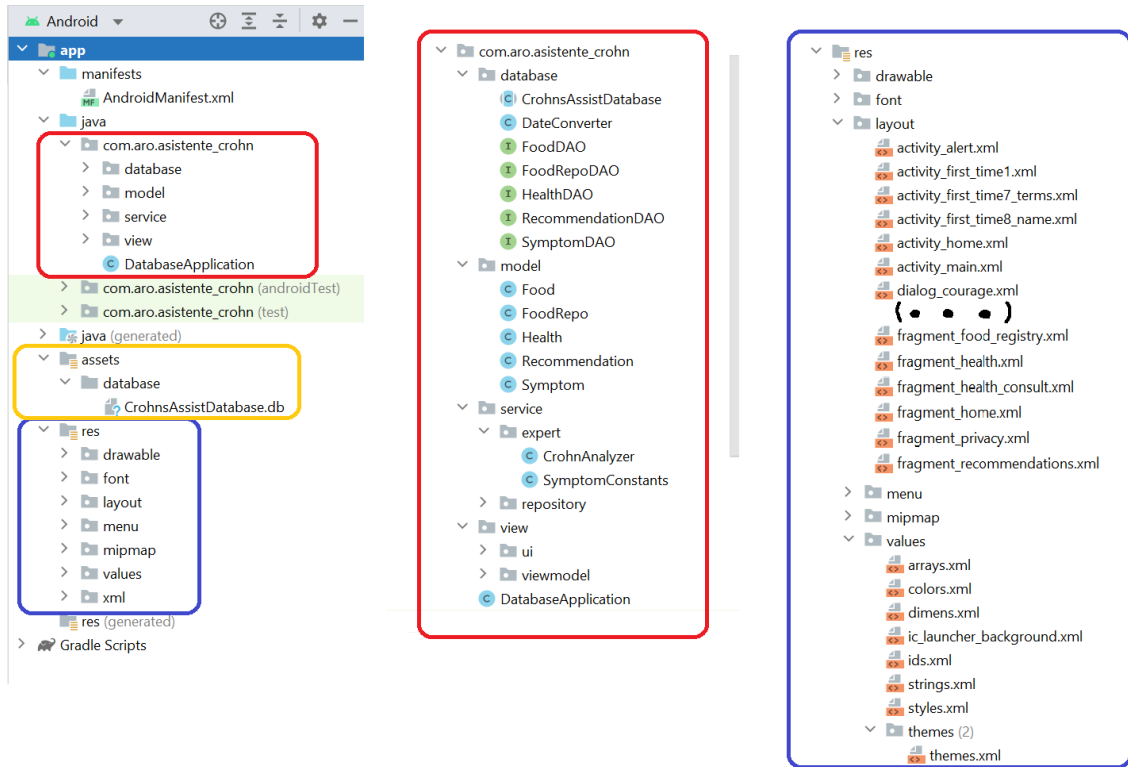


Figura A.1: Estructura de ficheros del proyecto

En ella, se divide el proyecto en tres partes diferenciadas: los archivos *Java*, la carpeta *assets* y los distintos archivos XML e imágenes de la carpeta *resources*.

En la carpeta de *Java* se encuentran los paquetes de la aplicación, en donde:

- En *View.ui* se almacenan las clases *Java* relativas a las *Actividades* y *Fragmentos*. Cada nombre de archivo se corresponde con un archivo de *res/layout* para una mayor comprensión. Aquí se puede ver que existen 5 *Actividades*, de ellas *MainActivity* se encarga de realizar la *Splash Screen*, *HomeActivity* es la usada en toda la aplicación por los diversos *Fragmentos* y *Diálogos* y el resto de *Actividades* son específicas de la Guía de la aplicación.
- En *View.ViewModel* se almacenan las clases pertenecientes al *ViewModel* (con los *Adaptadores*). Desde aquí, los objetos *LiveData* obtienen del repositorio los elementos a mostrar en las Vistas.
- En *database* se encuentran los *DAOs* de llamada a la base de datos (*Room*), junto con la clase de creación de la base de datos y quien se

encarga de migrar la base de datos de una versión a otra en caso de ser prepopulada con nuevos datos (tablas *FoodRepo* y *Recommendations*).

- En *model* se guardan los objetos de las clases.
- En *service.expert* se almacenan la clase del algoritmo para la detección de brotes y una clase (*SymptomsConstants.java*), dedicada a establecer las puntuaciones de cada síntoma y de cada tipo.
- En *repository* se almacenan las llamadas desde el *ViewModel* a los *DAOs* mediante objetos *LiveData*.

En *assets* se almacena el fichero de la base de datos, el cual se puede abrir desde *DB Browser for SQLite* tal y como se ha especificado anteriormente. Este archivo se usa en la clase específica de la base de datos, denominada *database.CrohnsAssistDatabase.java*.

Por último, en *res* se almacenan los recursos gráficos de la aplicación. En *drawable* se encuentran iconos e imágenes; en *layout* las ventanas de cada *Actividad*, *Fragmento* y *Diálogos*; en *values* se encuentran los valores preestablecidos como los colores a emplear, textos, dimensiones, los cuales son cambiados desde estos archivos para todo el proyecto; en *themes* se encuentran las configuraciones para los modos claro y oscuro de la aplicación; en *xml* se encuentra el archivo *root_preferences.xml*, el *layout* correspondiente a los ajustes.

Apéndice B

Manual de Usuario

En este capítulo se ofrecerá un manual de usuario de la aplicación desarrollada. Para ello, es necesario tener la aplicación instalada en un dispositivo, bien mediante la Play Store o mediante ADB como se especificó en el Manual Técnico.

B.1. Guía de Inicio

Una vez se inicia la aplicación por primera vez se muestra la Guía de Inicio, en donde el usuario puede conocer todo lo que le ofrece la aplicación.

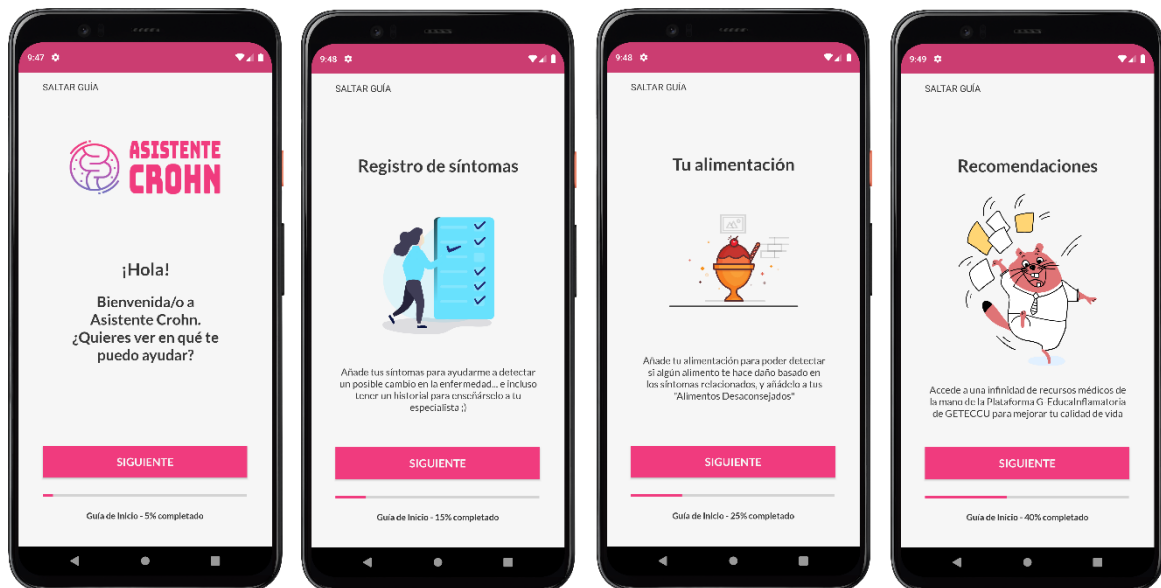


Figura B.1: Guía de Inicio (I)

Las dos últimas ventanas de la Guía son los Términos de Uso y la elección del nombre y tipo de enfermedad. En caso de haber accedido a esta guía con anterioridad y haber aceptado los Términos de Uso y haber escrito el nombre y tipo de enfermedad, estas ventanas ya no se muestran y el usuario es redirigido directamente a la ventana de Inicio.

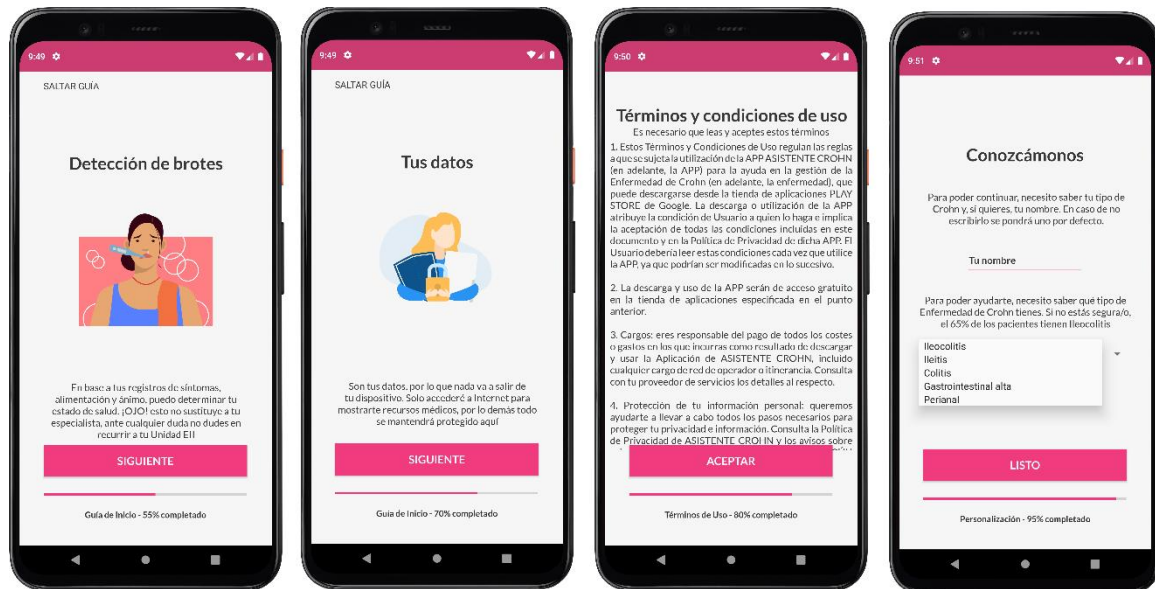


Figura B.2: Guía de Inicio (II)

B.2. Ventana Principal

Desde esta ventana, el usuario puede acceder a los diferentes módulos de la aplicación, diferenciados mediante tarjetas y colores.



Figura B.3: Ventana Principal

Tal y como se puede distinguir, desde aquí se accede a los módulos de Salud, Sintomatología, Alimentación y Recomendaciones, además de acceder a los ajustes desde el icono superior derecho “⚙️”.

B.3. Sintomatología

Al seleccionar el módulo de Sintomatología desde la ventana principal, nos encontramos con lo siguiente:

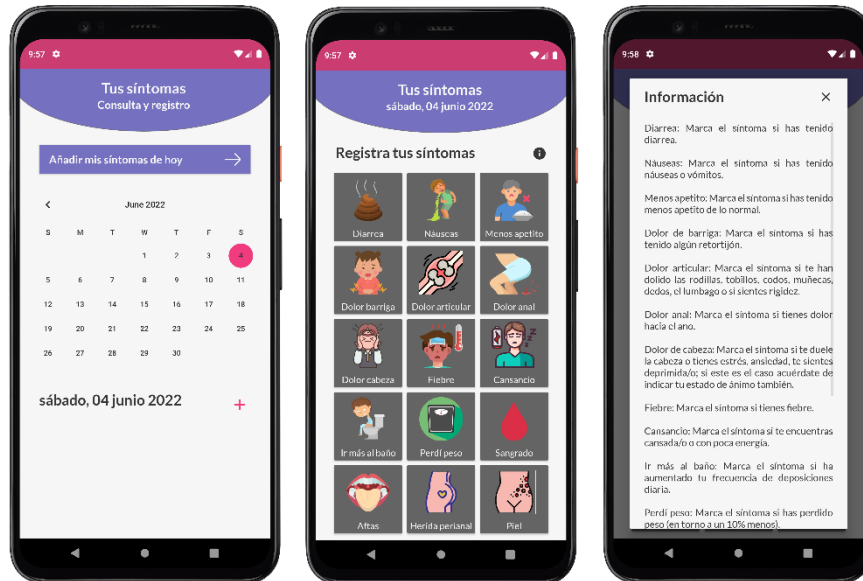


Figura B.4: Consulta de síntomas registrados y su significado

En ella, podemos observar un calendario para visualizar los síntomas registrados por días y una tarjeta en la zona superior para registrar los síntomas del día actual. Además, se pueden registrar síntomas de días anteriores en el icono “+” habilitado para ello.

Al acceder al registro de síntomas, nos encontramos con una tarjeta por cada síntoma, con una foto descriptiva y su nombre. En caso de tener dudas, se puede tocar sobre el icono “i” en la esquina superior derecha, el cual abre una ventana de información de cada síntoma.

Además, para conocer en qué día se están añadiendo registros, aparece en el cabecero de la ventana la fecha seleccionada (la actual o la del día seleccionado desde el calendario).

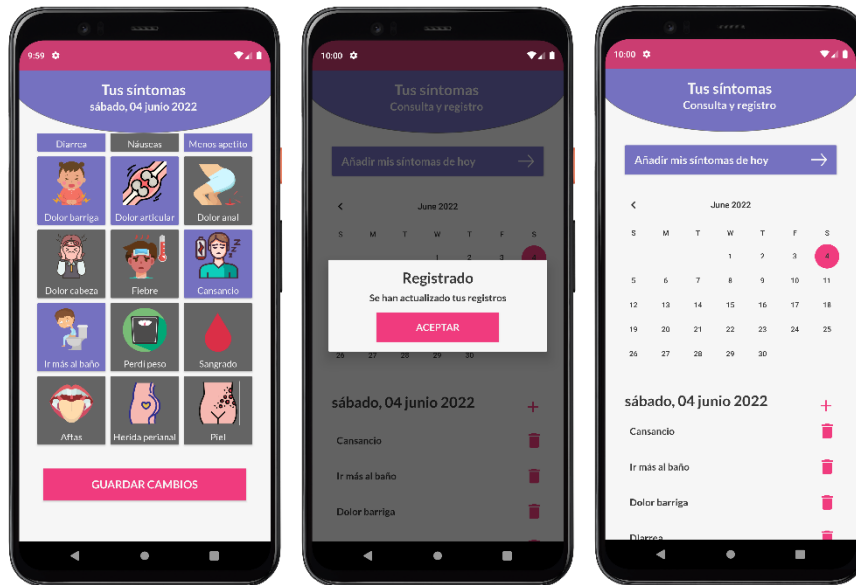


Figura B.5: Registro de síntomas

Una vez se selecciona el día en donde se quieren añadir registros, únicamente es necesario tocar sobre cada síntoma. Los síntomas ya guardados se marcan de color violeta como si estuviesen seleccionados, pudiendo deseleccionarlos si el usuario lo desea.

Al finalizar, el botón habilitado al final, “Guardar cambios”, se encarga de guardar los nuevos síntomas en la base de datos o de eliminar los ya existentes, mostrando a su vez una ventana informando de su registro exitoso.

Una vez añadidos, el calendario muestra los síntomas registrados, pudiendo eliminar un síntoma desde ahí sin tener que ir a la propia ventana de registro.

B.4. Alimentación

Al seleccionar el módulo de Alimentación desde la ventana principal, nos encontramos con lo siguiente:



Figura B.6: Módulo de Alimentación

Tal y como se observa, es la misma estructura que en el Módulo de Sintomatología. Se diferencia en que existen dos tarjetas, una para añadir la Alimentación del día de hoy y otra para añadir un nuevo alimento no recomendado.

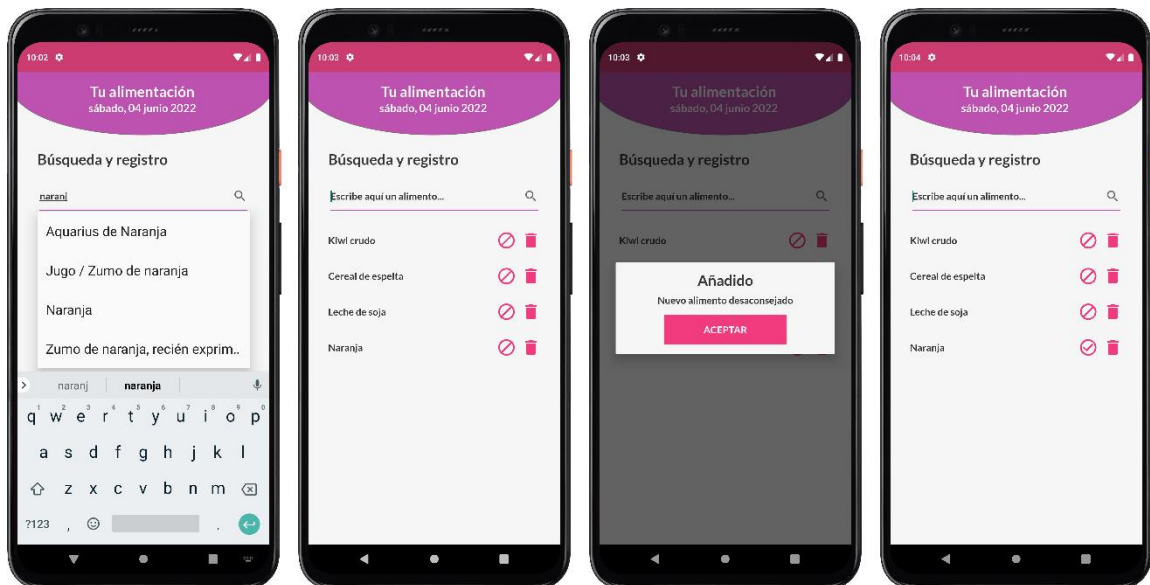


Figura B.7: Añadir alimentos y marcado de un no recomendado

En la anterior captura se puede ver lo que sucede al seleccionar la tarjeta “Añadir mi alimentación de hoy”. En esta ventana, se puede buscar cualquier alimento y añadirlo a los registros en cuestión. Estas ventanas se reutilizan al usar el icono “+” del calendario, modificando la fecha del cabecero.

En caso de querer eliminar un alimento, solamente es necesario pulsar sobre el icono de la papelera. Un alimento también puede ser marcado como no recomendado al presionar sobre el icono “⊘”. Se puede observar cómo este icono cambia una vez el alimento está marcado como no recomendado, en donde al pulsarlo únicamente deja de ser no recomendado para el usuario.

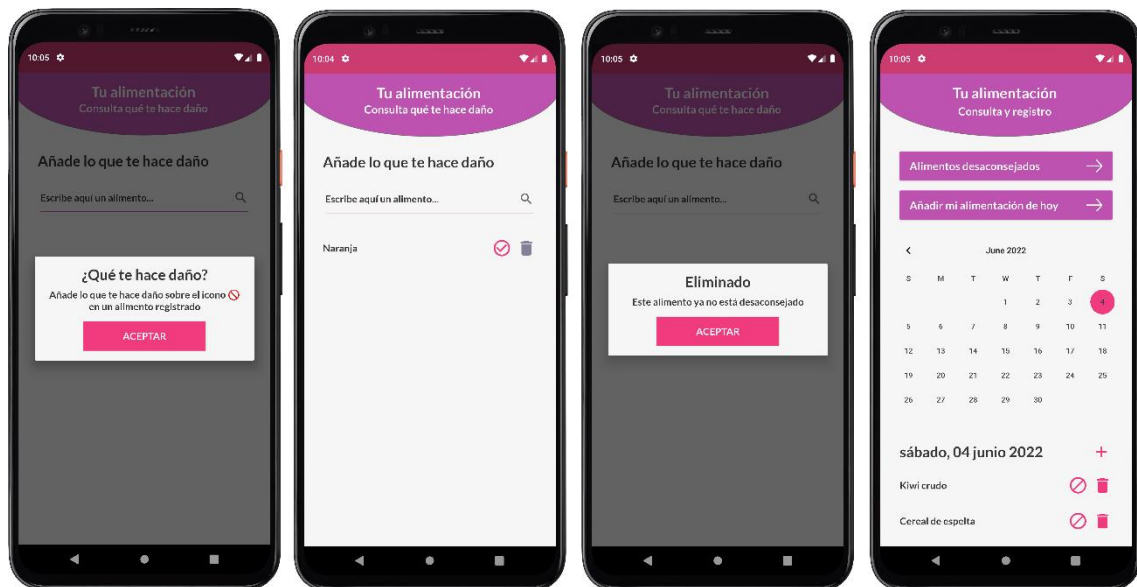


Figura B.8: Añadir alimentos no recomendados y consulta

En caso de seleccionar la tarjeta “Alimentos desaconsejados”, se abre una nueva ventana específica para ello. Si no hay alimentos, se indica un mensaje informando de cómo añadirlos.

En caso de tener alimentos registrados, como es el caso de la “Naranja” añadida en la figura B.7, se mostraría en dicha ventana. Podría marcarse como ya no recomendada al pulsar sobre el icono en sí, no siendo posible su eliminación para así evitar a la confusión.

Los registros se pueden observar, como en Sintomatología, en el calendario.

B.5. Alertas de alimentos no recomendados

Es posible alertar al usuario en caso de sospechar que un alimento le haga daño. Para ello, se necesita tener registros de síntomas y alimentos.

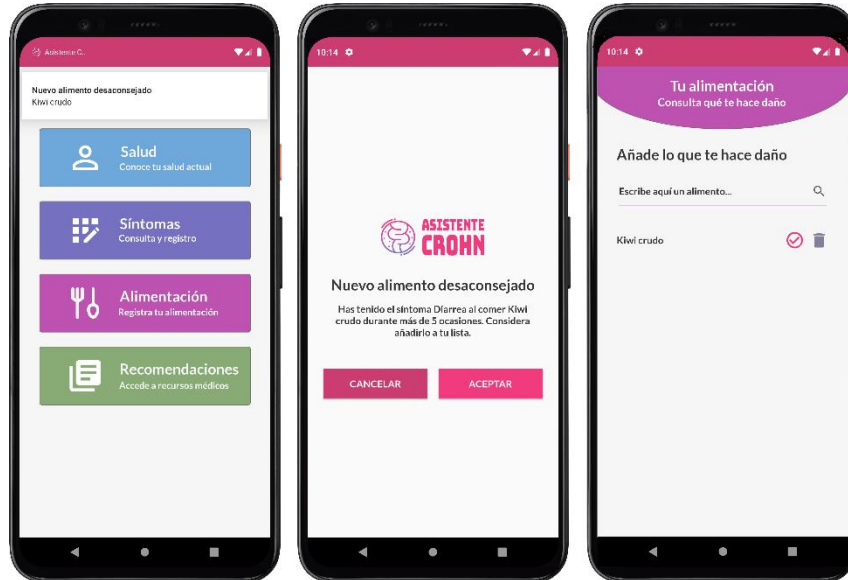


Figura B.9: Alerta de un alimento no recomendado

En caso de encontrar una relación entre un síntoma y un alimento durante repetidas ocasiones, la aplicación lo alerta mediante una notificación. Al pulsar la notificación, se abre una ventana indicando el síntoma, el alimento y el motivo.

Se puede ver que, una vez se acepta el alimento como no recomendado, este se añade a su ventana.

B.6. Módulo de Salud

Al seleccionar el módulo de Salud desde la ventana principal, nos encontramos con lo siguiente:

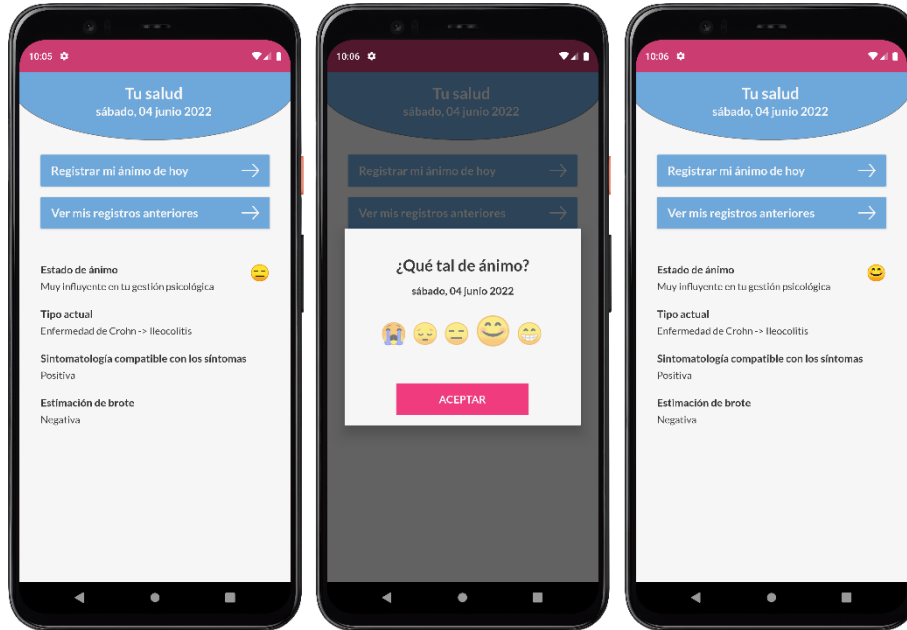


Figura B.10: Módulo de Salud y estado de ánimo

El módulo de Salud es algo distinto, mostrando nada más abrirlo el estado de Salud actual y dos tarjetas, una para añadir el estado de ánimo y otra para consultar los registros anteriores.

En caso de seleccionar el estado de ánimo (el cual está por defecto “😞” si no se selecciona otro), este se modifica y se muestra.

Se puede observar cómo la sintomatología del tipo de enfermedad es compatible con los síntomas registrados en la figura B.5 para el día actual.

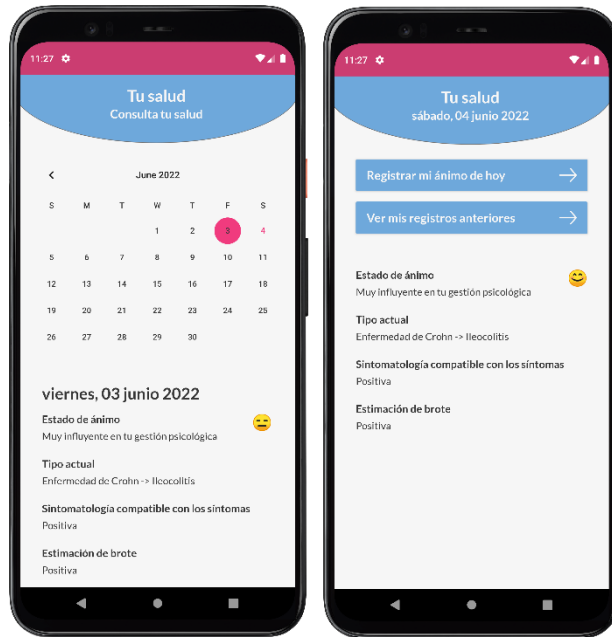


Figura B.11: Consulta y brote de la enfermedad

En la opción de consulta se observa el calendario. En él, se ve el estado de Salud para cada día.

Una vez la sintomatología es positiva durante los días establecidos en los ajustes de la aplicación (por defecto, 3 días), se indica que se encuentra en un brote, mostrando de esta forma el estado.

B.7. Módulo de Recomendaciones

Al seleccionar el módulo de Recomendaciones desde la ventana principal, nos encontramos con lo siguiente:

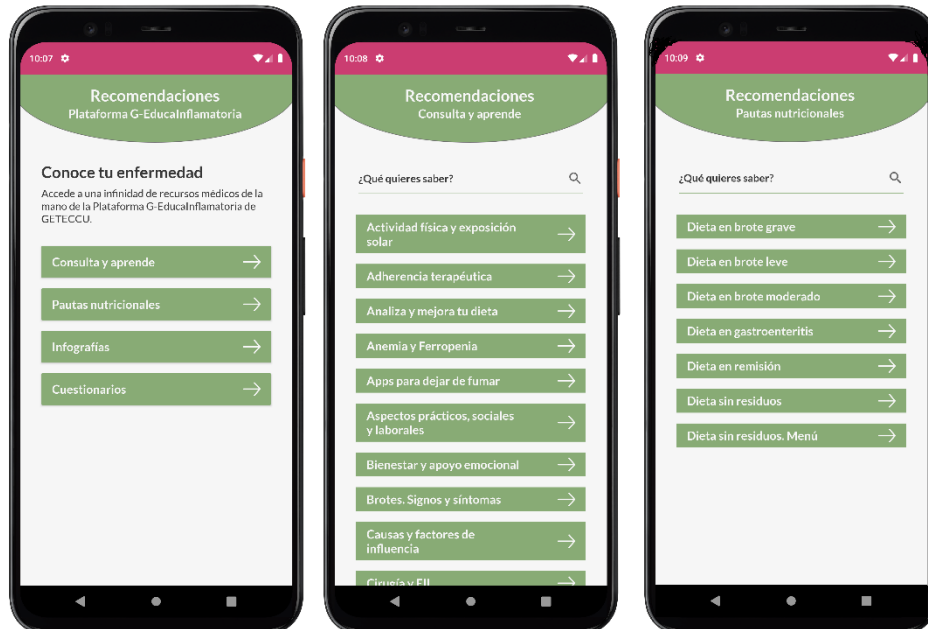


Figura B.12: Acceso a Recomendaciones (I)

Aquí se pueden contrar las Recomendaciones de la Plataforma G-EducaInflamatoria ordenadas según su tipo. En las capturas se muestran 2 secciones.

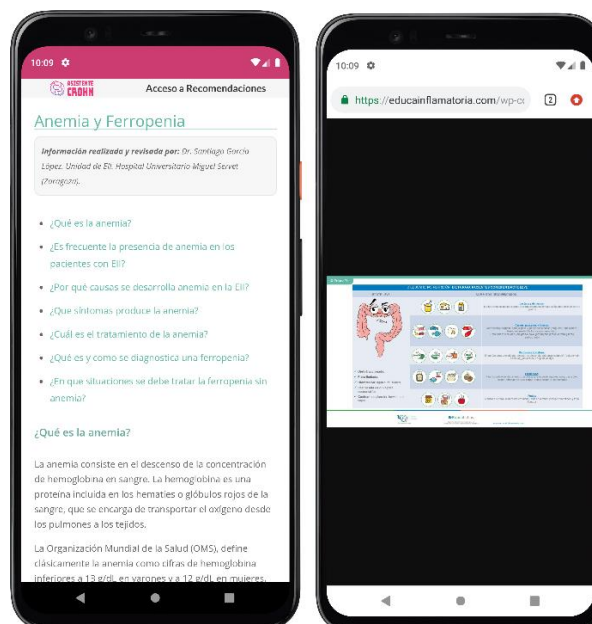


Figura B.13: Acceso a Recomendaciones (II)

Al pulsar sobre una Recomendación, esta se abre en la propia aplicación o desde el navegador según sea información, una imagen o un cuestionario.

B.8. Ajustes

Al seleccionar los ajustes desde la ventana principal, nos encontramos con lo siguiente:

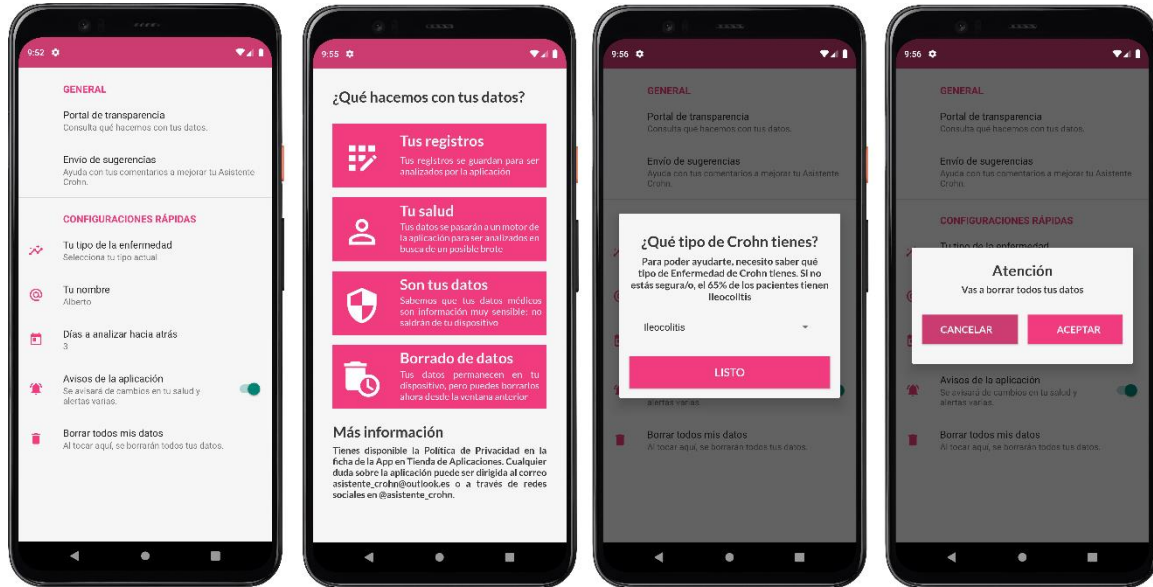


Figura B.14: Ajustes y Privacidad

Desde los ajustes, el usuario puede modificar su nombre, tipo de la enfermedad, los días a analizar, activar o desactivar alertas y borrar todos sus datos de memoria.

También puede acceder a una sección que explica cómo se tratan sus datos desde la aplicación.