



# Aplicación ExtreFit

José Luis Moreno Torés

Sergio Ramírez Pla

20/10/2025

— **Repositorio GitHub:**

[Repositorio GitHub](#)

2º DAM B

— **Tablero Tello:**

[Tablero de Tello](#)

IES Albarregas

— **Repositorio Figma:**

[Tablero figma](#)

Profesora: Mercedes

## ÍNDICE

<b>ÍNDICE.....</b>	<b>1</b>
<b>1. Resumen ejecutivo.....</b>	<b>2</b>
1.1. Problema/oportunidad.....	2
1.2. Propuesta de solución (1–2 párrafos).....	2
1.3. Objetivo de producto y objetivo de E1.....	2
1.4. Alcance de la demo de E1 (qué se muestra y qué no).....	2
<b>2. Justificación.....</b>	<b>3</b>
2.1. Beneficios esperados (técnicos, educativos y de uso).....	3
2.2. Integración con la industria extremeña (sectores, clústeres, casos de uso).....	3
2.3. Análisis de productos similares (benchmark breve: 3–5 referencias).....	4
2.4. Participación en ODS (mapa ODS ↔ funcionalidad/impacto).....	5
<b>3. Historias de usuario.....</b>	<b>6</b>
3.1. Convenciones de numeración y formato (p. ej., HU-001, HU-002...).....	6
3.2. Ejemplos de historias (3–5) con Criterios de Aceptación (Given/When/Then).....	7
3.3. Backlog priorizado (MoSCoW/Kano o similar).....	7
Must/Should/Could → Imprescindible/Importante/Conveniente.....	7
3.4. Trazabilidad HU ↔ CA ↔ Pruebas ↔ Figma.....	8
<b>4. Arquitectura.....</b>	<b>8</b>
4.1. Diagrama (C1/C2: contexto y contenedores).....	8
4.2. Decisiones de arquitectura (ADR) y numeración (p. ej., ADR-001, ADR-002).....	10
4.2.1. ADR-001: Elección de arquitectura cliente-servidor modular.....	10
4.2.2. ADR-002: Elección de herramientas y tecnologías.....	11
4.3. Integraciones, datos y dependencias.....	12
4.4. Riesgos técnicos y mitigación.....	12
<b>5. Requisitos no funcionales (NFR).....</b>	<b>12</b>
5.1. Convención de numeración (p. ej., NFR-001, NFR-002...).....	12
5.2. Rendimiento y capacidad (latencia, throughput).....	13
5.3. Seguridad (authZ/authN, cifrado, logs).....	13
5.4. Accesibilidad y UX (pautas mínimas).....	13
5.5. Calidad del código y mantenibilidad (estándares, lint, pruebas).....	14
5.6. Observabilidad (logs, métricas).....	14
5.7. Verificación de NFR (métricas y cómo medirlas).....	14
<b>6. Prototipo Figma y Anexos.....</b>	<b>15</b>
6.1. Prototipo navegable (lista de pantallas/flows y estados).....	15
6.2. Guía de diseño (tokens: color, tipografía, espacio; componentes y variantes).....	15
<b>7. Anexos.....</b>	<b>15</b>
7.3.1. Plan de pruebas y evidencias (capturas, vídeos, enlaces).....	15
7.3.2. KPIs iniciales (aceptación HIU, defectos, lead time).....	15
7.3.3. Enlaces y referencia de artefactos (repo, tablero, ADR, NFR).....	15
7.3.4. Changelog de E1 (tabla de cambios relevantes).....	15

## 1. Resumen ejecutivo

### 1.1. Problema/oportunidad

En la actualidad, muchas personas desean mejorar su alimentación y su forma física, pero no cuentan con herramientas sencillas que les orienten de forma clara y práctica. Las aplicaciones existentes suelen ser demasiado técnicas, de pago o centradas solo en deportistas.

Por ello, existe la oportunidad de crear una aplicación accesible para cualquier usuario, que combine nutrición, rutinas de ejercicio y consejos básicos para llevar un estilo de vida saludable.

### 1.2. Propuesta de solución (1–2 párrafos)

ExtreFit se plantea como una aplicación enfocada en la nutrición y el bienestar general. Permitirá calcular el índice de masa corporal (IMC), ofrecer recomendaciones alimenticias personalizadas y proponer rutinas básicas de ejercicio adaptadas a cada persona.

La idea es que el usuario tenga en un mismo lugar toda la información necesaria para empezar a mejorar su salud, sin necesidad de conocimientos previos. En futuras versiones se podrán incluir más funciones, pero esta primera etapa se centra en construir una base sencilla y funcional.

### 1.3. Objetivo de producto y objetivo de E1

El objetivo general del producto es ofrecer una herramienta útil que ayude a cualquier persona a adquirir hábitos saludables mediante una aplicación fácil de usar, visual y motivadora.

El objetivo específico de la iteración E1 es mostrar el funcionamiento básico del sistema, incluyendo el registro de usuario, los cálculos principales (como el IMC) y la generación de recomendaciones o menús simples.

### 1.4. Alcance de la demo de E1 (qué se muestra y qué no)

En la demo de la primera entrega (E1) se mostrará el flujo básico de la aplicación: el registro o inicio de sesión de usuario, el cálculo del IMC y la visualización de recomendaciones alimenticias y rutinas de ejercicio básicas.

No se incluirán todavía funciones avanzadas como el seguimiento de progresos a largo plazo o la personalización completa de menús. El objetivo principal de esta demo es demostrar la estructura y funcionalidad base del proyecto.

## 2. Justificación

### 2.1. Beneficios esperados (técnicos, educativos y de uso)

El desarrollo de ExtreFit aportará una serie de beneficios tanto en el ámbito técnico como educativo y práctico. Desde el punto de vista técnico, el proyecto nos permitirá diseñar una arquitectura cliente-servidor funcional, sencilla y escalable, capaz de ofrecer una experiencia fluida tanto en dispositivos móviles como en navegadores web.

La aplicación contará con un sistema de autenticación de usuarios, almacenamiento de datos personales en una base de datos y la posibilidad de realizar cálculos desde el backend, como el índice de masa corporal (IMC).

Además, se prevé la integración de APIs que permitan acceder a bases de datos de alimentos y, en futuras versiones, la incorporación de un lector de códigos de barras que facilite el registro de productos de manera rápida y práctica.

Todo esto se complementará con una interfaz *responsive*, adaptada a distintos tamaños de pantalla, con un diseño moderno y accesible que priorice la comodidad del usuario.

En el ámbito educativo, este proyecto representa una oportunidad para aplicar en un contexto real los conocimientos adquiridos durante el ciclo de Desarrollo de Aplicaciones Multiplataforma.

A través del desarrollo de ExtreFit, pondremos en práctica conceptos como el trabajo en equipo, la gestión del control de versiones mediante GitHub, la planificación ágil de tareas y el uso de herramientas profesionales de diseño como Figma.

Asimismo, nos permitirá mejorar nuestra capacidad para resolver problemas reales de programación, tomar decisiones técnicas y organizar el proceso de desarrollo de manera autónoma y eficiente. Más allá del aspecto técnico, este proyecto también fomenta la colaboración, la comunicación y la gestión del tiempo, habilidades fundamentales en cualquier entorno profesional.

Por último, en cuanto al uso por parte del público, ExtreFit ofrecerá una herramienta útil y accesible que ayudará a los usuarios a cuidar su salud y bienestar general. La aplicación proporcionará cálculos básicos como el IMC, rutinas de ejercicio adaptadas y recomendaciones nutricionales sencillas, pensadas para todo tipo de personas, no solo deportistas o expertos en nutrición. Su principal beneficio será la facilidad de uso: una app práctica, cercana y visual que motive a los usuarios a mejorar sus hábitos diarios.

## 2.2. Integración con la industria extremeña (sectores, clústeres, casos de uso)

ExtreFit se desarrolla en el IES Albarregas, en Mérida, dentro de la comunidad autónoma de Extremadura, una región con un importante potencial en el ámbito agroalimentario y turístico. El proyecto busca aprovechar esta conexión territorial incluyendo guiños y vínculos con la industria local.

Extremadura es conocida por la calidad de sus productos agroalimentarios y por su compromiso con la promoción de hábitos de vida saludables. En este sentido, la aplicación podría servir como un punto de encuentro entre tecnología y producto local, fomentando el consumo de alimentos típicos de la región y ofreciendo menús o recomendaciones basados en productos de temporada y proximidad.

A nivel práctico, se podrían establecer colaboraciones con empresas del sector agroalimentario, gimnasios o centros de nutrición locales, creando sinergias entre el mundo educativo y el profesional.

Asimismo, la aplicación podría tener utilidad dentro de programas institucionales orientados a la promoción de la salud o la educación alimentaria en la región. De esta manera, ExtreFit no se limita a ser un proyecto académico, sino que puede convertirse en una herramienta con un valor real dentro del ecosistema extremeño, uniendo tecnología, educación y desarrollo local.

## 2.3. Análisis de productos similares (benchmark breve: 3–5 referencias)

Para definir las características de ExtreFit se ha realizado un análisis comparativo de varias aplicaciones consolidadas en el ámbito de la salud y la nutrición, entre las que destacan MyFitnessPal, Samsung Health, Lose It! y My Enjoy. Cada una de estas herramientas aporta ventajas y limitaciones que permiten identificar un hueco en el mercado que ExtreFit puede cubrir.

**MyFitnessPal** es una de las aplicaciones más conocidas para el registro de alimentos y ejercicios. Su principal fortaleza radica en la amplitud de su base de datos, que permite registrar prácticamente cualquier alimento, así como en la integración con múltiples dispositivos y apps de fitness. Además, ofrece una comunidad activa que motiva al usuario a cumplir sus objetivos. No obstante, su complejidad puede resultar excesiva para principiantes, y algunas entradas de alimentos generadas por los propios usuarios pueden no ser fiables. Asimismo, la versión gratuita presenta limitaciones importantes en funciones avanzadas, lo que puede restringir la experiencia de aquellos usuarios que buscan un seguimiento completo sin coste adicional. La lección para ExtreFit es clara: es posible tener una base de datos robusta, pero la prioridad debe ser la usabilidad, ofreciendo un registro integral de nutrición y ejercicio sin sobrecargar al usuario con funcionalidades técnicas o innecesarias.

**Samsung Health**, por su parte, adopta un enfoque integral de bienestar. Permite registrar actividad física, sueño, ingesta de agua y alimentación, proporcionando así una visión global de la salud del usuario. Su integración con dispositivos Samsung es muy potente, aunque puede

limitar su eficacia en usuarios que no poseen estos dispositivos. Además, la planificación nutricional avanzada es limitada y algunas integraciones con terceros son complicadas. ExtreFit puede aprender de Samsung Health la importancia de ofrecer un seguimiento holístico de la salud, incorporando distintas métricas y permitiendo que muchas funciones básicas sean gratuitas, para fomentar la adopción inicial.

**Lose It!** es una aplicación muy conocida internacionalmente que destaca por su interfaz clara y amigable y por facilitar el registro de alimentos mediante un escáner de código de barras. Permite personalizar objetivos calóricos y nutricionales y ofrece reportes de progreso semanales, así como integración con dispositivos como Fitbit o Apple Watch. Sin embargo, algunas funciones avanzadas, como el seguimiento de macronutrientes o estadísticas detalladas, están bloqueadas en la versión gratuita, y la entrada de alimentos generada por los usuarios puede causar errores puntuales. Asimismo, ciertas funciones automáticas, como la suma de calorías quemadas, pueden resultar confusas si no se ajustan correctamente. De Lose It! se desprende la importancia de ofrecer funcionalidades prácticas y visuales desde el inicio, con un registro rápido de alimentos, personalización de objetivos y seguimiento de progreso, evitando que la app resulte compleja o confusa para el usuario.

Finalmente, **My Enjoy** se caracteriza por su sencillez y enfoque en rutinas de ejercicio básicas y consejos saludables accesibles. Su ventaja principal es la baja barrera de entrada para cualquier usuario, sin complicar la experiencia con métricas avanzadas o bases de datos extensas. No obstante, su limitación reside en la falta de seguimiento nutricional detallado y en la ausencia de integraciones con dispositivos o funciones avanzadas. Este enfoque minimalista muestra que ofrecer una versión ligera de ExtreFit, centrada en lo esencial —ejercicio básico y recomendaciones sencillas—, puede ser muy atractivo para usuarios que buscan iniciar hábitos saludables sin complicaciones.

En conjunto, el análisis de estas aplicaciones revela un hueco importante en el mercado: ninguna ofrece una combinación equilibrada de sencillez, acceso gratuito, enfoque educativo y adaptabilidad a distintos niveles de usuario. ExtreFit pretende cubrir esta necesidad, ofreciendo una herramienta clara, visual y accesible que integre registro nutricional básico, cálculo del IMC, rutinas de ejercicio adaptadas y recomendaciones personalizadas, sin abrumar al usuario con opciones técnicas avanzadas o funcionalidades premium excesivas. La combinación de lecciones extraídas de estas aplicaciones consolidadas permitirá construir una app inicial sólida y fácil de usar, con potencial de crecimiento hacia versiones más completas y personalizadas en el futuro.

## 2.4. Participación en ODS (mapa ODS ↔ funcionalidad/impacto)

El proyecto ExtreFit está alineado con varios de los Objetivos de Desarrollo Sostenible establecidos por la ONU, especialmente con el ODS 3, centrado en garantizar una vida sana y promover el bienestar para todos.

La aplicación contribuye directamente a este objetivo al fomentar hábitos saludables mediante la alimentación equilibrada y la práctica regular de ejercicio físico.

A través de sus funciones de cálculo del IMC, recomendaciones personalizadas y rutinas básicas, ExtreFit pretende mejorar la salud de sus usuarios y concienciar sobre la importancia del autocuidado y la prevención de enfermedades relacionadas con la mala alimentación o el sedentarismo.

Además, también se vincula con el ODS 4, Educación de calidad, al incorporar contenidos educativos sobre nutrición y bienestar que ayudan a mejorar la cultura alimentaria de los usuarios. Igualmente, ExtreFit se relaciona con el ODS 12, Producción y consumo responsables, al promover el uso de productos locales y de temporada dentro de sus recomendaciones, apoyando así al sector agroalimentario extremeño y fomentando un consumo más sostenible.

De esta forma, el proyecto no solo tiene un valor tecnológico y formativo, sino también un impacto positivo a nivel social, educativo y ambiental, contribuyendo a la mejora de la calidad de vida y al desarrollo sostenible en su entorno más cercano.

## 3. Historias de usuario

### 3.1. Convenciones de numeración y formato (p. ej., HU-001, HU-002...)

En el desarrollo de ExtreFit se ha seguido una metodología ágil, centrada en la entrega continua de valor al usuario y en la adaptación a sus necesidades reales. En este contexto, las historias de usuario (HU) constituyen el elemento central de la planificación funcional, ya que representan de manera clara y concisa cada necesidad o requerimiento desde la perspectiva de quien utilizará la aplicación. Cada historia describe no solo qué debe hacer el sistema, sino también el beneficio concreto que proporciona al usuario, asegurando que cada desarrollo técnico tenga un impacto tangible y perceptible.

Para mantener una estructura ordenada y trazable, cada historia recibe un identificador en el formato HU-XXX, donde “XXX” es un número incremental. Así, la primera historia es HU-001, la segunda HU-002, y así sucesivamente hasta cubrir la totalidad de funcionalidades previstas. Esta numeración facilita relacionar cada historia con el prototipo de Figma, los criterios de aceptación, las pruebas funcionales y los commits en el repositorio de GitHub, garantizando un control completo del desarrollo.

La redacción de cada historia sigue la convención estándar en metodologías ágiles: “Como [tipo de usuario], quiero [funcionalidad o acción] para [lograr un beneficio concreto]”. Además, cada HU incluye criterios de aceptación en formato Dado / Cuando / Entonces (D/C/E), que permiten verificar de forma objetiva cuándo la historia se considera completada. Este enfoque asegura que las historias sean claras, verificables y directamente relacionadas con la experiencia del usuario.

### 3.2. Ejemplos de historias (3–5) con Criterios de Aceptación (Given/When/Then)

Para garantizar que la primera versión de ExtreFit (iteración E1) sea funcional y útil, se han definido 15 historias de usuario, que cubren desde el registro hasta funcionalidades prospectivas, pasando por los elementos esenciales de seguimiento de salud. Por ejemplo, HU-001 permite al usuario registrarse en la aplicación mediante correo electrónico y contraseña, asegurando que los datos personales se almacenen de manera segura. La HU-002 aborda el inicio de sesión, de modo que el usuario pueda recuperar su sesión y acceder a información previamente guardada. Estas historias se clasifican como **Must Have**, ya que sin ellas el flujo funcional mínimo de la aplicación no sería posible.

Otras historias se centran en el núcleo funcional de ExtreFit. La HU-004 calcula el índice de masa corporal (IMC) cuando el usuario introduce peso y altura, mostrando además una clasificación interpretativa de su estado físico. La HU-005 interpreta estos resultados, permitiendo al usuario entender su situación de manera sencilla y visual. A partir de estos datos, HU-006 y HU-007 proporcionan recomendaciones nutricionales y rutinas de ejercicio, respectivamente, ofreciendo orientación personalizada y fomentando hábitos saludables.

Posteriormente, se incluyen funcionalidades orientadas a la gestión y personalización del perfil, como la HU-009, que permite modificar peso, altura y edad, y la HU-010, que muestra un historial de cálculos anteriores para hacer un seguimiento de la evolución.

Complementariamente, HU-012 introduce la opción de marcar menús y rutinas favoritas, mientras que HU-014 activa un modo oscuro para mejorar la experiencia visual del usuario. Por último, HU-015 contempla la integración con dispositivos inteligentes (smartwatches) para registrar automáticamente la actividad física, reflejando la escalabilidad futura del proyecto.

Cada historia incluye criterios de aceptación que definen de manera verificable cuándo se considera completada. Por ejemplo, la HU-004 se da por cumplida cuando, dado que el usuario introduce su peso y altura, el sistema calcula correctamente el IMC y lo clasifica según los estándares de salud. En conjunto, las 15 historias proporcionan un flujo completo de interacción, asegurando que el prototipo inicial sea funcional y demostrable.

### 3.3. Backlog priorizado (MoSCoW/Kano o similar)

**Must/Should/Could → Imprescindible/Importante/Conveniente**

La priorización de las historias de usuario se ha realizado siguiendo la metodología MoSCoW, que clasifica las funcionalidades en **Must Have**, **Should Have**, **Could Have** y **Won't Have**, según su relevancia para la primera iteración.

Las historias Must Have son imprescindibles: registro, inicio de sesión, cálculo e interpretación del IMC, y cierre de sesión. Las Should Have, aunque no críticas, añaden valor significativo, como la visualización de rutinas y recomendaciones nutricionales o la edición de datos de perfil. Las Could Have aportan mejoras estéticas o complementarias, como favoritos, modo oscuro e historial de progresos. Finalmente, Won't Have agrupa funcionalidades planificadas para futuras versiones, como la integración con dispositivos inteligentes.

Esta priorización permite concentrar los esfuerzos en lo esencial, garantizando un prototipo funcional y demostrable, mientras se mantiene la posibilidad de incorporar mejoras de forma escalonada. Además, facilita el seguimiento del progreso del equipo, evaluando la finalización de las historias prioritarias como indicador de cumplimiento de objetivos.

### 3.4. Trazabilidad HU ↔ CA ↔ Pruebas ↔ Figma

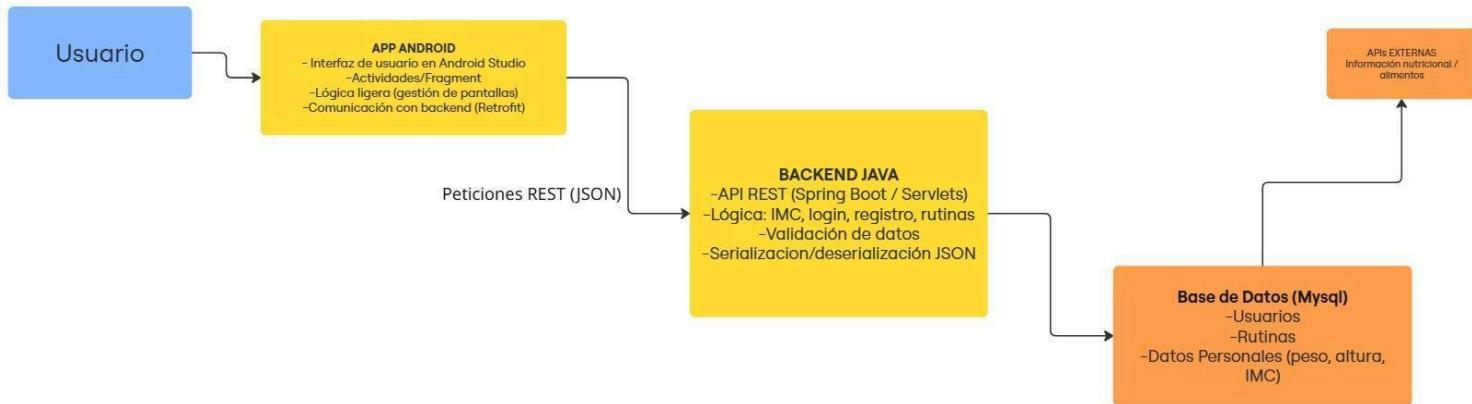
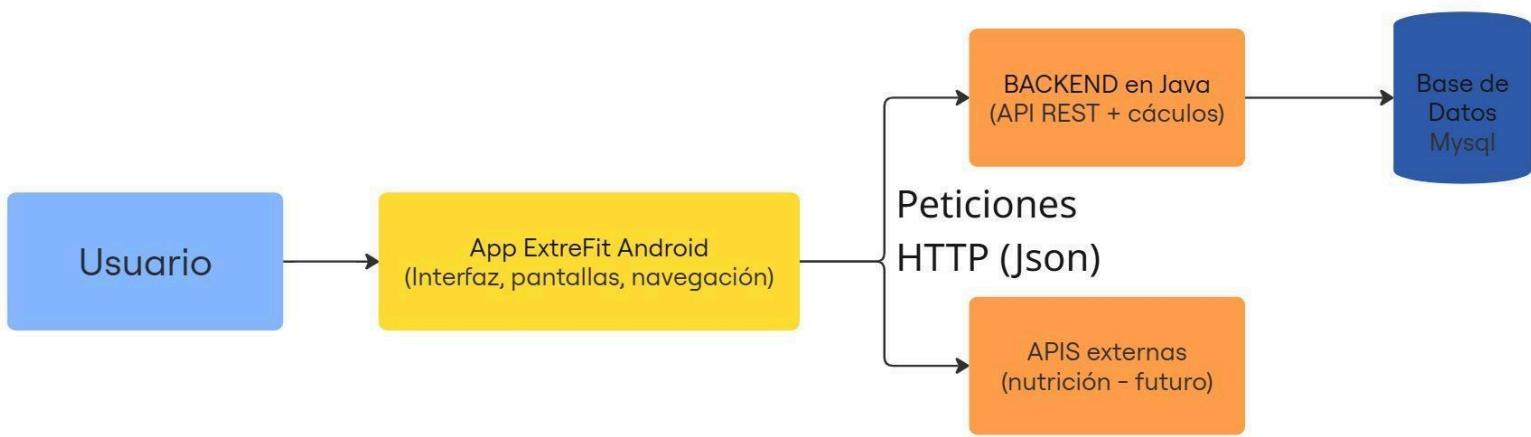
En ExtreFit, la trazabilidad es un elemento fundamental del control de calidad. Cada historia de usuario está directamente vinculada con criterios de aceptación, pruebas funcionales y pantallas del prototipo de Figma. Por ejemplo, la HU-001 se asocia con la pantalla de registro, y la HU-004 con el panel de salud donde se visualiza el IMC. Esta relación asegura que lo que se desarrolla corresponde exactamente con lo planificado, y que cualquier desviación puede detectarse y corregirse con rapidez.

## 4. Arquitectura

### 4.1. Diagrama (C1/C2: contexto y contenedores)

La aplicación ExtreFit se entiende mejor cuando se analiza dentro de su propio contexto tecnológico, ya que su funcionamiento depende de la interacción entre diferentes componentes que se encargan de cumplir funciones concretas. En el nivel más general, que corresponde al modelo C1 o nivel de contexto, ExtreFit se sitúa como un sistema digital destinado a ofrecer servicios de salud y bienestar a personas que desean mejorar su alimentación y su forma física. El actor principal dentro de este contexto es el propio usuario, que accede a la aplicación a través de su dispositivo móvil Android. Desde el momento en que abre la aplicación, todo el flujo del sistema gira en torno a esta relación directa entre el usuario y la App de ExtreFit, que actúa como punto de entrada a todas las funciones de la herramienta.

Una vez el usuario inicia la aplicación en su móvil, esta se convierte inmediatamente en el centro de su interacción con el sistema. La App Android no funciona de manera aislada, sino que se conecta de manera continua con un servidor backend desarrollado en Java, que representa la otra pieza fundamental del sistema. Todo lo que el usuario solicita desde la interfaz, ya sea calcular el IMC, iniciar sesión, registrarse, obtener recomendaciones de alimentación o ver rutinas de ejercicio, se envía desde la aplicación móvil hacia el backend. Este servidor no solo procesa la petición, sino que también toma decisiones internas sobre cómo debe gestionarse la información, ya sea consultando datos previamente almacenados o accediendo a servicios externos. Además de la comunicación con el backend, la aplicación mantiene conexión con una serie de APIs externas que pueden proporcionar información complementaria según las necesidades del proyecto, como datos nutricionales o información relacionada con hábitos saludables. Finalmente, el backend se comunica con la base de datos MySQL, que funciona como el lugar en el que se almacenan los datos persistentes del sistema, tales como usuarios registrados, históricos, configuraciones o cualquier dato necesario para que la aplicación mantenga continuidad en su funcionamiento. De esta forma, el contexto general del sistema queda formado por la relación entre el usuario, la App Android, el backend Java, las APIs externas y la base de datos MySQL, cada uno cumpliendo un papel concreto dentro del funcionamiento global.



Cuando se desciende al nivel C2, que es el nivel más orientado a los contenedores o a la estructura interna del sistema, se puede observar con mayor claridad cómo se organizan las diferentes partes del proyecto. El usuario continúa siendo el punto de inicio, pero en este nivel cobra más importancia la forma en la que cada contenedor se comunica con los demás. La aplicación Android funciona como un contenedor independiente que, además de presentar la interfaz al usuario, actúa como intermediario entre él y el backend. Cada acción que el usuario realiza dentro de la App se traduce en una petición REST que se envía al servidor Java. Este backend, que constituye otro contenedor diferenciado, es el encargado de recibir las peticiones, interpretarlas, procesarlas y decidir si necesita consultar la base de datos o conectarse a un servicio externo antes de generar una respuesta.

El contenedor correspondiente al backend en Java se comunica directamente con otro contenedor que es esencial para el correcto funcionamiento de la aplicación: la base de datos MySQL. Cada vez que se necesita almacenar información sobre un usuario, recuperar sus datos, validar su inicio de sesión o guardar resultados como su IMC, es el backend quien

realiza las consultas necesarias en la base de datos. La comunicación se realiza siguiendo las prácticas habituales de persistencia, pero se mantiene encapsulada dentro de este contenedor, evitando que la App Android tenga acceso directo a los datos. Paralelamente, el backend también mantiene comunicación con contenedores externos que representan las APIs de terceros. Estas APIs permiten obtener datos complementarios, principalmente relacionados con nutrición o ejercicio, que enriquecen las recomendaciones ofrecidas por la aplicación sin necesidad de almacenar toda esa información localmente.

Así, la estructura de contenedores en el nivel C2 queda definida por una interacción clara y ordenada: el usuario actúa sobre la aplicación Android, la aplicación realiza peticiones REST al backend Java, y el backend se encarga tanto de consultar la base de datos MySQL como de comunicarse con las APIs externas. Cada contenedor cumple un rol diferenciado, pero todos trabajan de manera coordinada para ofrecer al usuario una experiencia fluida y coherente. Esta separación en contenedores permite que cada pieza del sistema pueda evolucionar, actualizarse o escalar independientemente sin afectar negativamente al resto del sistema, manteniendo una arquitectura clara, modular y adecuada para un proyecto que irá ampliándose en iteraciones posteriores.

## 4.2. Decisiones de arquitectura (ADR) y numeración (p. ej., ADR-001, ADR-002)

Dentro del desarrollo de ExtreFit ha sido necesario tomar una serie de decisiones técnicas importantes que afectarán directamente a cómo evoluciona el proyecto en el futuro. Para que estas decisiones queden bien documentadas y sean comprensibles tanto para nosotros como para cualquier otra persona que trabaje en el proyecto más adelante, se ha utilizado el formato ADR, que consiste en explicar el contexto en el que se tomó la decisión, la opción elegida, las alternativas consideradas y las consecuencias que tendrá esa elección. Este tipo de documentación resulta especialmente útil en proyectos educativos como el nuestro, ya que permite justificar de forma razonada cada avance y da una visión muy clara del camino que hemos seguido para construir la arquitectura del sistema.

### 4.2.1. ADR-001: Elección de arquitectura cliente-servidor modular

**Contexto:** Para el proyecto era necesario definir una estructura sencilla pero flexible, que permitiera trabajar de forma colaborativa y distribuir responsabilidades entre interfaz, lógica de negocio y almacenamiento de datos.

**Decisión:** Se adoptó una arquitectura cliente-servidor modular en tres capas: presentación (Android), lógica de negocio (Java) y datos (MySQL).

**Alternativas descartadas:** Se consideró desarrollar la aplicación solo con HTML y JavaScript, guardando datos en LocalStorage, pero esta opción limitaba enormemente las funcionalidades y la escalabilidad. Otra alternativa fue usar desde el principio un servicio en la nube externo que gestionara toda la lógica, pero resultaba excesivo para una primera versión desarrollada por dos alumnos.

**Consecuencias:** Esta decisión garantiza escalabilidad, mantenimiento sencillo y posibilidad de añadir mejoras sin afectar al resto del sistema. Cada capa puede evolucionar de forma independiente, facilitando la integración de nuevas funcionalidades o APIs externas en versiones futuras.

**Fecha/Estado:** 25/11/2025 – Aprobado y en uso en E1.

#### 4.2.2. ADR-002: Elección de herramientas y tecnologías

**Contexto:** El equipo cuenta con conocimientos en Java, SQL y desarrollo web básico, por lo que se necesitaba seleccionar herramientas que garantizaran rapidez y comprensión.

**Decisión:** Se optó por las siguientes tecnologías: para el frontend Android, desarrollo nativo en Java/Kotlin; para el backend, Java; y para la base de datos, MySQL. El control de versiones se realiza mediante GitHub.

**Alternativas descartadas:** Se evaluó el uso de frameworks complejos como Angular o Spring Boot, pero se descartaron por la curva de aprendizaje y la complejidad innecesaria para la primera iteración. También se consideró el uso de SQLite como base de datos única en el dispositivo móvil, pero se desechó por limitar la capacidad de sincronización de datos y crecimiento del sistema.

**Consecuencias:** La elección de herramientas conocidas facilita el desarrollo rápido y comprensible, garantiza mantenimiento sencillo y permite que otros profesores o compañeros puedan entender y evaluar el código sin dificultades.

**Fecha/Estado:** 25/11/2025 – Aprobado y en uso en E1.

#### 4.3. Integraciones, datos y dependencias

En la primera iteración, ExtreFit almacenará información esencial del usuario, como nombre, correo, peso, altura e IMC, en la base de datos MySQL. Las recomendaciones y menús se generarán a partir de datos internos, pero la arquitectura permite conectar con APIs de nutrición en el futuro. Las dependencias técnicas incluyen librerías para validación de formularios, gestión de peticiones REST y posible representación gráfica de información con Chart.js. Para la gestión del proyecto se usarán herramientas externas como Figma para el diseño de pantallas y Trello para organizar tareas y seguimiento de trabajo en equipo.

#### 4.4. Riesgos técnicos y mitigación

Se identificaron varios riesgos que podrían afectar a la estabilidad y funcionalidad de ExtreFit. Entre ellos destacan la seguridad de datos, limitaciones de tiempo por la reducida plantilla, errores de integración entre frontend y backend, diseño de interfaz y conflictos de código al usar GitHub. Cada riesgo cuenta con estrategias de mitigación: cifrado y validación de datos para proteger la información personal, priorización MoSCoW para centrarse en lo esencial,

pruebas incrementales para detectar problemas de integración, prototipos en Figma para asegurar usabilidad, y trabajo en ramas separadas con revisión de cambios para evitar conflictos. Estas medidas garantizan que la arquitectura se mantenga estable y preparada para futuras iteraciones.

## 5. Requisitos no funcionales (NFR)

Los requisitos no funcionales establecen las propiedades esenciales que ExtreFit debe cumplir para ofrecer una experiencia de usuario de calidad, garantizar la seguridad y permitir que la aplicación sea mantenible y escalable en el tiempo. A diferencia de los requisitos funcionales, que describen acciones específicas de la aplicación, los NFR definen cómo el sistema se comporta y qué estándares de calidad debe mantener. Desde el inicio del proyecto, estos requisitos han guiado la toma de decisiones técnicas y arquitectónicas, reflejadas en los ADR, asegurando que cada elección de tecnología y diseño respalde la satisfacción de estos criterios esenciales.

### 5.1. Convención de numeración (p. ej., NFR-001, NFR-002...)

Para mantener un control claro y preciso de los requisitos no funcionales, se ha adoptado la convención de numeración NFR-XXX, en la que cada requisito recibe un identificador incremental que facilita su trazabilidad y referencia en futuras versiones del proyecto. Esta numeración permite relacionar cada requisito con decisiones técnicas concretas, con historias de usuario y con las evidencias de prueba, garantizando un seguimiento riguroso. Por ejemplo, NFR-001 corresponde al rendimiento básico de la aplicación, NFR-002 a la seguridad y cifrado de datos, NFR-003 a la accesibilidad y diseño responsive, y así sucesivamente hasta cubrir todos los aspectos relevantes. Esta convención asegura que los desarrolladores, evaluadores y cualquier miembro del equipo puedan referirse de manera inequívoca a cada requisito durante todo el ciclo de vida de ExtreFit, desde la implementación hasta las pruebas y la documentación.

### 5.2. Rendimiento y capacidad (latencia, throughput)

ExtreFit se ha diseñado con la prioridad de ofrecer un rendimiento ágil y eficiente. Esto significa que cada acción realizada por el usuario, desde navegar entre pantallas hasta calcular su índice de masa corporal (IMC), debe ejecutarse de manera casi instantánea. La arquitectura cliente-servidor modular, documentada en ADR-001, respalda directamente este requisito al permitir que la carga de trabajo se distribuya entre la aplicación Android, el backend Java y la base de datos MySQL. Gracias a esta separación de responsabilidades, la aplicación puede procesar operaciones en paralelo y reducir tiempos de espera, evitando que la interfaz se vuelva lenta o que el usuario perciba retardos en la respuesta del sistema. Las decisiones técnicas también incluyen optimización de recursos, evitando enviar datos innecesarios en cada petición REST y manteniendo las imágenes y scripts ligeros para acelerar la carga de

pantallas. Estas medidas garantizan un rendimiento adecuado en dispositivos con diferentes capacidades de hardware, asegurando que la aplicación sea ágil incluso en terminales modestos.

### 5.3. Seguridad (authZ/authN, cifrado, logs)

La seguridad de los datos del usuario es un requisito no funcional esencial que ha guiado varias decisiones arquitectónicas. La autenticación básica (authN) permite que cada usuario se registre y acceda con su correo electrónico y contraseña, mientras que la autorización (authZ), planificada para versiones futuras, garantizará que los usuarios solo puedan acceder a sus propios datos. ADR-002, sobre la elección de herramientas y tecnologías, respalda este requisito al permitir seleccionar soluciones conocidas y seguras que faciliten la implementación de cifrado de contraseñas y validación de entradas. Las contraseñas se almacenarán cifradas mediante algoritmos robustos, como bcrypt, y todas las entradas se validarán para prevenir inyecciones SQL o errores de formulario. Además, se mantendrán registros detallados de eventos y errores para detectar posibles accesos no autorizados o fallos del sistema, facilitando tanto la auditoría como la mejora continua de la seguridad. Este enfoque integral garantiza que los datos del usuario estén protegidos desde la primera iteración del proyecto, alineándose con los estándares de seguridad modernos y las buenas prácticas de desarrollo.

### 5.4. Accesibilidad y UX (pautas mínimas)

La accesibilidad y la experiencia de usuario son fundamentales para ExtreFit, y todas las decisiones de diseño se han tomado pensando en la facilidad de uso y la claridad visual. La aplicación cuenta con una interfaz intuitiva, colores suaves y tipografía legible, que permite que cualquier persona pueda navegar por la aplicación sin dificultad. El diseño es responsive, adaptándose automáticamente a móviles, tabletas y ordenadores. La elección de un backend modular y un frontend Android, según ADR-001, facilita la implementación de pantallas adaptables y rápidas, mientras que las tecnologías seleccionadas en ADR-002 aseguran que se pueda desarrollar una interfaz coherente y fácil de mantener. Además, se han considerado pautas mínimas de accesibilidad, como botones suficientemente grandes y flujos de navegación claros, para que la aplicación sea usable por personas con distintos niveles de experiencia tecnológica. Esto no solo mejora la experiencia del usuario, sino que también incrementa la adopción de la aplicación al hacerla accesible para un público más amplio.

### 5.5. Calidad del código y mantenibilidad (estándares, lint, pruebas)

Desde la concepción del proyecto, se han establecido criterios rigurosos de calidad del código para garantizar que ExtreFit sea mantenible y escalable. Se utilizan nombres de variables y funciones claros y descriptivos, una indentación consistente y comentarios explicativos en secciones complejas del código. Además, la gestión del código mediante GitHub permite que cada desarrollador trabaje en ramas separadas, facilitando la revisión de cambios antes de fusionarlos a la rama principal y evitando conflictos. Las herramientas de análisis estático (lint)

aseguran que el código cumpla con los estándares establecidos, y se planifican pruebas unitarias y de integración que permiten verificar que las funcionalidades se ejecutan correctamente. Estas decisiones están directamente relacionadas con ADR-002, ya que la elección de tecnologías conocidas garantiza que el equipo pueda aplicar buenas prácticas de desarrollo y mantener un código limpio y comprensible, incluso cuando el proyecto crezca o sea revisado por terceros.

## 5.6. Observabilidad (logs, métricas)

La observabilidad es un aspecto clave para poder monitorizar el comportamiento de la aplicación y garantizar que se cumplen los NFR. ExtreFit implementará mecanismos de registro de eventos, errores y métricas de rendimiento en el backend, de manera que el equipo pueda analizar cualquier incidencia, detectar problemas de latencia o fallos en la ejecución y tomar decisiones informadas sobre mejoras. Esta capacidad de observación se apoya en la arquitectura modular definida en ADR-001, que facilita centralizar los logs y recopilar métricas de forma estructurada sin afectar al rendimiento de la aplicación. La observabilidad no solo permite la detección temprana de problemas, sino que también respalda la verificación de otros NFR, como la seguridad y la experiencia de usuario, al proporcionar datos concretos sobre el comportamiento real del sistema.

## 5.7. Verificación de NFR (métricas y cómo medirlas)

Para cada requisito no funcional se han definido métodos de verificación claros y medibles. El rendimiento se comprobará evaluando la rapidez de la carga de pantallas y la eficiencia en el cálculo del IMC, documentando los resultados con capturas y vídeos. La seguridad se validará mediante pruebas de acceso, comprobación del cifrado de contraseñas y simulación de entradas maliciosas para confirmar que no se producen vulnerabilidades. La accesibilidad se medirá observando la adaptabilidad de la interfaz a diferentes dispositivos y evaluando la facilidad de uso en distintos escenarios. La calidad del código se verificará mediante revisiones de GitHub y herramientas de análisis estático, y la observabilidad se comprobará revisando que los logs registran correctamente los eventos y que las métricas se recopilan de manera adecuada. Cada verificación está vinculada a los ADR: ADR-001 asegura que la arquitectura permite la monitorización y el rendimiento óptimo, mientras que ADR-002 garantiza que las tecnologías seleccionadas permiten implementar seguridad, mantenibilidad y calidad de manera efectiva. Con este enfoque, los NFR no solo se documentan, sino que se pueden evaluar de manera objetiva, asegurando que ExtreFit cumpla con los estándares de calidad, seguridad y usabilidad desde la primera iteración.

## 6. Prototipo Figma y Anexos

### 6.1. Prototipo navegable (lista de pantallas/flows y estados)

<https://github.com/Beethovencompositor21/Proyecto-Intermodular/blob/main/Dокументos%20Proyecto/Figma/Pantallas%20ExtreFit.fig>

### 6.2. Guía de diseño (tokens: color, tipografía, espacio; componentes y variantes)

#### Tokens de diseño

##### Colores principales:

- Verde (#5DCE33) → Salud, energía, progreso.
- Blanco (#FFFFFF) → Claridad y limpieza.
- Negro (#000000) → Legibilidad y contraste.

##### Colores funcionales de botones:

- Entrar → Verde (#5DCE33)
- Volver → Azul (#29657B)
- Registrarse → Naranja (#B95621)
- Cerrar sesión → Rosa (#A422B8)
- Guardar → Morado (#7B29B9)

##### Tipografía:

- **Títulos:** Inter, moderna, clara, deportiva.
- **Cuerpo de texto:** Inter, alta legibilidad en pantallas pequeñas.
- **Accesibilidad asegurada:** Tamaño mínimo 12 px, contraste suficiente, evitar grises claros.

## Espaciado y layout:

- Estructura limpia y minimalista.
- Botones: altura mínima 48 px, anchura mínima 90 px, esquinas redondeadas 20 px, sombreado ligero.
- Formularios: bordes finos 1 px, etiquetas visibles, feedback visual al enfocar.
- Layout general: navegación fija en parte superior o inferior, espaciado consistente para legibilidad.

## Componentes y variantes

### 1. Botones

- Variantes: Entrar, Volver, Registrarse, Cerrar sesión, Guardar.
- Estado normal, hover, activo, deshabilitado.
- Colores definidos por función.

### 2. Campos de formulario

- Tipos: texto, contraseña, checkbox.
- Feedback visual: borde y sombra al enfocar, color de error para validaciones.

### 3. Círculos de progreso

- Uso: calorías, ejercicios, comidas.
- Variantes: completado, parcial, vacío.
- Colores: verde progresivo, gris para vacío.

### 4. Notificaciones y mensajería

- Bandeja de entrada y mensajes leídos.
- Indicadores visuales de nuevos mensajes.

### 5. Calendario

- Ejercicios y nutrición: día destacado, días completados.
- Enlaces a videos o detalles de comida.

## 6. Iconografía

- Calendario → rutina y nutrición.
- Sobre → mensajería.
- Engranaje → configuración.
- Estilo consistente y minimalista, tamaño uniforme.

## 7. Anexos

### 7.3.1. Plan de pruebas y evidencias (capturas, videos, enlaces)

<https://github.com/Beethovencomposer21/Proyecto-Intermodular/blob/main/Documento%20Proyecto/MatrizPruebasHU/Matriz%20de%20Pruebas%20HU%20ExtreFit.pdf>

### 7.3.2. KPIs iniciales (aceptación HU, defectos, lead time)

### 7.3.3. Enlaces y referencia de artefactos (repo, tablero, ADR, NFR)

**Repositorio GitHub:**

<https://github.com/Beethovencomposer21/Proyecto-Intermodular/tree/main>

**Tablero:**

<https://trello.com/b/3jKuQ5kX/pidam2b-extrefit-extrefit>

**Lista de Historias de Usuario:**

### 7.3.4. Changelog de E1 (tabla de cambios relevantes)