

Documento de Requisitos — NoraFit (MVP)

1. Resumen del producto

NoraFit es una aplicación móvil destinada a permitir a usuarios registrar, gestionar y monitorizar sus rutinas de gimnasio de forma simple y gratuita. El MVP debe incluir las funcionalidades esenciales para crear/usar rutinas, registrar ejercicios (peso, repeticiones), temporizadores (incluido modo HIIT), historial de sesiones, y una biblioteca de ejercicios con visualizaciones GIF que muestren los músculos activados. El objetivo es ofrecer seguimiento real del entrenamiento sin necesidad de suscripciones.

2. Personas (2-3)

- Usuario Principiante (Ana)**

Perfil: Persona que asiste al gimnasio de forma irregular, busca organizar sus rutinas y ver progreso claro. Necesita una interfaz sencilla y guías visuales del ejercicio.

- Usuario Intermedio (Carlos)**

Perfil: Entrena con regularidad, utiliza rutinas estructuradas y busca llevar registros detallados de peso y repeticiones para progresar.

- Administrador / Soporte (Equipo NoraFit)**

Perfil: Personal técnico que gestiona contenidos (ejercicios), revisa reportes de uso y mantiene la plataforma.

3. Requisitos Funcionales (priorizados)

Notación: RF = Requisito Funcional

RF-1 — Gestión de cuentas / autenticación (High)

Descripción: Registro, inicio de sesión y gestión básica de cuentas de usuario. **Criterios de aceptación:**

- El usuario puede registrarse con email y contraseña.
- El usuario puede iniciar sesión con email y contraseña.
- Sesión persistente (token JWT) tras login.

Estimación: S

Dependencias: Base de datos de usuarios, endpoints de auth.

RF-2 — Crear / editar / eliminar rutinas (High)

Descripción: Permitir al usuario crear rutinas compuestas por ejercicios ordenables.

Criterios de aceptación:

- El usuario puede crear una rutina con nombre y lista de ejercicios (mín. 1).
- El usuario puede editar nombre, ejercicios y orden.
- El usuario puede eliminar una rutina con confirmación.

Estimación: M

Dependencias: RF-4 (biblioteca de ejercicios).

RF-3 — Registrar ejercicio en una sesión (High)

Descripción: En una sesión activa, el usuario registra sets con peso y repeticiones por ejercicio.

Criterios de aceptación:

- El usuario selecciona el ejercicio y guarda sets con peso y repeticiones.
- Los registros se almacenan vinculados a la sesión y fecha.
- Se muestra resumen de la sesión tras finalizar.

Estimación: M

Dependencias: RF-2, RF-6 (historial).

RF-4 — Biblioteca de ejercicios (High)

Descripción: Base de datos de ejercicios con nombre, categoría, GIF de ejecución y músculos activados.

Criterios de aceptación:

- Se pueden buscar y filtrar ejercicios por categoría.
- Cada ejercicio tiene un GIF o animación y una lista de músculos activados.
- Admin puede añadir/editar ejercicios (endpoint protegido).

Estimación: M

Dependencias: Backend media storage.

RF-5 — Modo HIIT (High)

Descripción: Temporizador configurable para entrenamientos HIIT (work/rest/rounds).

Criterios de aceptación:

- Usuario configura duración de trabajo y descanso y número de rondas.
- Temporizador ejecuta cuenta regresiva y notifica cambios (vibración/push).

- Registro de sesión HIIT guardado en historial.

Estimación: M

Dependencias: RF-3 (registro de sesiones), notificaciones.

RF-6 — Historial y progreso (High)

Descripción: Visualización del historial de sesiones y evolución por ejercicio. **Criterios de aceptación:**

- El usuario ve historial con fecha, ejercicios y sets.
- El usuario puede ver tendencia (por ejemplo: máxima carga por ejercicio por mes).
- Exportar historial en CSV (opcional).

Estimación: M

Dependencias: RF-3, DB.

RF-7 — Temporizadores de descanso por set (Medium)

Descripción: Temporizador simple para descanso entre series en una sesión. **Criterios de aceptación:**

- Usuario inicia temporizador en pantalla de sesión.
- Temporizador puede pausar/reiniciar. **Estimación:** S

Dependencias: RF-3.

RF-8 — Configuración del perfil (Medium)

Descripción: Datos básicos del usuario (altura, peso, objetivo) para personalizar métricas.

Criterios de aceptación:

- Usuario puede editar sus datos básicos.
- Interfaz muestra progreso relativo a objetivos. **Estimación:** S

RF-9 — Backup/restore básico de datos (Low)

Descripción: Opción para exportar/importar datos de usuario (JSON/CSV). **Criterios de aceptación:**

- Usuario puede exportar su historial en CSV.
- Importar sólo formato validado. **Estimación:** L

RF-10 — Sistema de feedback / reporte de errores (Low)

Descripción: Enviar reportes por correo o ticket (simple) al equipo. **Criterios de aceptación:**

- Usuario puede abrir un formulario de reporte con descripción.
- Se genera registro en DB o se envía un correo al equipo. **Estimación:** S

RF-11 — Sin funciones sociales (informativo) (High)

Descripción: No se implementan feeds ni social sharing en el MVP. Se documenta explícitamente qué queda fuera. **Criterios de aceptación:**

- No existen endpoints / pantallas para posts, likes o comentarios. **Estimación:** 0 (decisión de diseño)

RF-12 — Visualización GIF y resaltado de músculos (Medium)

Descripción: Mostrar GIFs con overlay que destaque músculos activos (visual). **Criterios de aceptación:**

- GIF reproducible en pantalla de ejercicio.
- Overlay o leyenda indica músculos principales.

Estimación: M

4. Requisitos No Funcionales

- **RNF-1 — Rendimiento:** Las llamadas principales a la API (listar rutinas, obtener ejercicio, guardar set) deben responder en < 300 ms bajo carga normal (usuarios concurrentes < 1000).
- **RNF-2 — Disponibilidad:** 99.5% uptime para servicios críticos (auth, DB).
- **RNF-3 — Seguridad:**
 - Autenticación con JWT y HTTPS obligatorio.
 - Contraseñas almacenadas con hashing (bcrypt o Argon2).
 - Control de acceso a endpoints admin.
- **RNF-4 — Privacidad:** Registro de datos mínimos; opción para eliminar cuenta (Right to be forgotten). Cumplir buenas prácticas de protección de datos.
- **RNF-5 — Compatibilidad:** App compatible con Android 8+ e iOS 12+ (según alcance inicial).

- **RNF-6 — Escalabilidad:** Backend modular en Spring Boot con posibilidad de escalar servicios y usar CDN para assets (GIFs).
 - **RNF-7 — Mantenibilidad:** Código documentado, pruebas unitarias básicas en backend y CI/CD en pipeline.
 - **RNF-8 — Localización:** Soporte inicial en español; arquitectura preparada para i18n.
-

5. Casos de uso / User Flows (breves)

1. Registro y primer uso

- Usuario se registra → crea perfil mínimo → explora biblioteca de ejercicios → crea primera rutina → inicia sesión de entrenamiento.

2. Sesión de fuerza

- Usuario selecciona rutina → inicia sesión → por cada ejercicio registra sets (peso/reps) → usa temporizador de descanso → finaliza y guarda sesión → consulta resumen.

3. Sesión HIIT

- Usuario configura work/rest/rounds → inicia temporizador HIIT → sigue intervalos → app notifica cambios → guarda la sesión como HIIT en historial.

4. Revisión de progreso

- Usuario abre historial → filtra por ejercicio → observa tendencia de cargas y repeticiones en periodo seleccionado.

6. Notas del Product Owner (Supuestos, alcance MVP y riesgos)

Supuestos importantes

- El MVP será una app móvil nativa híbrida en Flutter (una base para Android e iOS).
- Backend en Java + Spring Boot con base de datos relacional (PostgreSQL recomendada).
- Los GIF de ejercicios pueden alojarse inicialmente en un bucket S3 o storage similar.
- No se implementan funciones sociales en este MVP.

Criterios de MVP (qué queda dentro) •

Registro y autenticación básica.

- CRUD de rutinas.
- Registro de sesiones y sets.
- Biblioteca de ejercicios con GIF.
- Temporizadores (descanso + HIIT).
- Historial y tendencias básicas.

Fuera del MVP (se planifica para futuras versiones)

- Feed social, compartición pública, planes de entrenamiento pagados (premium) — opcional en futuro.
- Integraciones con wearables (futuro).
- Machine learning para recomendaciones automáticas.

Riesgos conocidos

- **Riesgo técnico:** Almacenamiento y carga de GIFs puede impactar rendimiento y consumo de datos.
- **Riesgo de adopción:** Usuarios acostumbrados a apps con entrenadores pueden preferir features sociales.
- **Riesgo de privacidad:** Manejo de datos personales exige políticas claras y medidas de seguridad.

7. Recomendaciones técnicas rápidas (para comenzar implementación)

- **Stack recomendado:** Flutter (frontend), Java + Spring Boot (backend), PostgreSQL, AWS S3 para GIFs, JWT para auth.
- **CI/CD:** Pipeline mínimo para builds automáticos (ej. GitHub Actions) y pruebas unitarias en backend.
- **Estrategia de despliegue:** Empezar con una demo en entorno de staging; usar feature flags para activar/desactivar funcionalidades.
- **Medición de uso:** Implementar telemetría básica (eventos: sesión iniciada, rutina completada, ejercicio registrado) para analítica de adopción.

8. Conversión rápida a Issues (ejemplo listo para pegar) Issue template (copiar dentro de un Issue nuevo):

Título: [US-04] Registrar sets en sesión **Descripción (user story):**

Como **usuario**, quiero **registrar sets** (peso y repeticiones) durante una sesión para llevar el control de mi progreso. **Acceptance criteria:**

- Formulario para agregar sets por ejercicio.
- Datos almacenados y visibles en el resumen de la sesión.
- Validaciones: peso y reps numéricos y > 0 . **Prioridad:** High

Estimación: M

Labels: backend, frontend, api, priority:high (Repetir

para cada user story del backlog).