Documento de requerimientos de software

***MiniFit***

***Fecha: 03-09-2025***

**Integrantes:** Dyland Martinez Cisternas

Polet Adasme Peñailillo

Fernanda Barra Reyes

**Sección:** Capstone-007v

**Docente:** Fernando Herrera Francesconi

**Tabla de contenido**

[Historial de Versiones 3](#_heading=h.am7428ymxwr9)

[Información del Proyecto 3](#_heading=h.30j0zll)

[Aprobaciones 3](#_heading=h.1fob9te)

[1. Propósito 4](#_heading=h.57ntnhdlyzji)

[2. Alcance del producto / Software 5](#_heading=h.pi1c83ijncx3)

[3. Referencias 7](#_heading=h.d3pty1wy5les)

[4. Funcionalidades del producto 8](#_heading=h.ebbhz2h54pkw)

[5. Clases y características de usuarios 9](#_heading=h.d4f7yyvttbk4)

[6. Entorno operativo 11](#_heading=h.rr9r2v3wntka)

[7. Requerimientos funcionales 12](#_heading=h.yswug8eyfojf)

[8. Reglas de negocio 17](#_heading=h.2q86wcii9wt)

[9. Requerimientos de interfaces externas 18](#_heading=h.eip8u592qz3v)

[Interfaces de usuario 18](#_heading=h.oohl81x8ohyw)

[Interfaces de hardware 20](#_heading=h.a4mxdk6kztpm)

[Interfaces de software 21](#_heading=h.u60km7diev56)

[10. Requerimientos no funcionales 23](#_heading=h.dx2xcczcqmos)

[11. Otros requerimientos 25](#_heading=h.y9thbrxd1pck)

[12. Glosario 26](#_heading=h.cypdk8ywaznv)

### 

### Historial de Versiones

| **Fecha** | **Versión** | **Autor** | **Organización** | **Descripción** |
| --- | --- | --- | --- | --- |
| 03-09-25 | v1.0 | Fernanda Barra | - | Primera versión del MVP para revisión interna |
|  |  |  |  |  |

### Información del Proyecto

| Empresa / Organización | DUOC UC |
| --- | --- |
| Proyecto | MiniFit |
| Fecha de preparación | 03-09-2025 |
| Cliente | Municipalidad la Pintana / Tutores de niños |
| Patrocinador principal | Supermercado Lider - Walmart |
| Gerente / Líder de Proyecto | Dyland Martínez |
| Gerente / Líder de Análisis de negocio y requerimientos | Polet Adasme |

### Aprobaciones

| **Nombre y Apellido** | **Cargo** | **Departamento u Organización** | **Fecha** | **Firma** |
| --- | --- | --- | --- | --- |
| Dyland Martínez Cisternas | Lider de Proyecto | - | 03-09-35 |  |
| Polet Adasme Peñailillo | Lider de Negocios | - | 03-09-35 |  |
| Fernanda Barra Reyes | QA/UX | - | 03-09-35 |  |
|  |  |  |  |  |

### Propósito

En el presente documento se especifica los requerimientos del MVP de MiniFit , una app móvil y PWA a padres, apoderados y tutores para registro simple de alimentación y actividad física para niños, con visualizaciones básicas, recomendaciones educativas para padres/tutores y consumo de APIs externas. En el documento se cubre el MVP completo, no incluye futuras extensiones, pagos ni integraciones. El proyecto utilizará herramientas y tecnologías accesibles y fáciles de implementar; Ionic, Angular y Firebase.

### 

### Alcance del producto / Software

El producto tiene como objetivo apoyar a padres y tutores en la concientización y seguimiento en los hábitos de sus hijos, con la finalidad de hacerlos más saludables a corto plazo.

**Beneficios del producto:**

* Patrocinador: En este caso el supermercado Lider (Walmart) se le dará visibilidad, donde se le dará una diferenciación de sus productos y como marca responsable en apoyar el proyecto de mejorar la salud. Se anuncia textual que es patrocinador.
* Municipalidad: Fortalecimiento de programas vida saludable, alineación con estrategias de MINSAL y JUNAEB, institución innovadora, comprometida con la infancia y posibilidad de integrar la app en escuelas.
* Usuarios: acceso gratuito a los usuarios de la comuna de la municipalidad permitiendo monitorear hábitos alimenticios y actividad física.

**Objetivos del software**

* Proveer una aplicación híbrida (PWA + Android) para registrar, visualizar y reflexionar sobre hábitos de alimentación y actividad física infantil.
* Integrar OpenFoodFacts para datos nutricionales; garantizar seguridad de datos en Firebase; implementar dashboard con gráficos; entregar un MVP en 16 semanas.
* Fomentar la concientización familiar sobre obesidad infantil; entregar recomendaciones educativas simples; abrir la puerta a escalabilidad hacia colegios o instituciones.

**Relación con objetivos corporativos**

* Supermercado patrocinador: estrategia corporativa de promover vida saludable en familias y niños. MiniFit actúa como canal digital y educativo, reforzando el posicionamiento responsable.

**Fuera de alcance**

* Módulos de monetización, pagos o suscripciones.
* Gamificación avanzada, rankings o comunidad.
* Integración con wearables (Fitbit, Apple Watch).
* Soporte iOS en MVP inicial

### 

### Referencias

* Mapa Nutricional JUNAEB: <https://www.mineduc.cl/mapa-nutricional-junaeb-registra-leve-alza-en-malnutricion-por-exceso-y-disminucion-de-grupos-con-deficit-alimentario/>
* OMS — Ending childhood obesity: <https://www.who.int/publications/i/item/9789241510066>
* OpenFoodFacts API Docs: <https://world.openfoodfacts.org/data>
* WCAG 2.2 — Web Content Accessibility Guidelines.

### 

### Funcionalidades del producto

1. Autenticación y gestión de sesión (Firebase Auth).
2. Perfil del apoderado (datos básicos y preferencia de recordatorios).
3. Gestión de niño(s) (alta/edición y historial de peso/estatura).
4. Registro de alimentación diaria (slots: desayuno, colaciones, almuerzo, cena; categorías simples; notas).
5. Registro de actividad física (tipo de actividad y minutos por día).
6. Reportes y gráficos (resumen semanal y mensual de alimentación y minutos activos).
7. Integración con OFF (búsqueda por texto/categoría y consulta por código de barras).
8. Ficha nutricional del alimento (valores por 100 g/ml y por porción; Nutri-Score cuando exista).
9. Calificación de salud del alimento (reglas educativas basadas en umbrales MINSAL; alertas de energía/azúcares/grasas sat./sodio).
10. Tips educativos (catálogo administrable por Admin; sin referencias a marcas).
11. Patrocinio con nota una nota visible “Patrocinado”.
12. Notificaciones locales (recordatorios básicos de registro; Android y PWA cuando aplique).
13. PWA instalable y build Android (Capacitor) para distribución de pruebas.
14. Panel Admin (semilla) para gestión de tips, banderas de features y datos de patrocinio.

[Matriz de trazabilidad de requisitos](https://docs.google.com/spreadsheets/d/13o5kubqDqNw1BWKs33lt-n9o7rDexv9q/edit?usp=sharing&ouid=115065065507976570629&rtpof=true&sd=true) - Matriz explicativa

### 

### Clases y características de usuarios

* **Administrador:** encargado de gestionar la plataforma a nivel global. Sus funciones incluyen mantener el catálogo de alimentos y actividades, revisar la consistencia de la información, y administrar los perfiles de usuarios registrados. Este rol es esencial para asegurar la calidad de los datos y la continuidad operativa de la aplicación.

Privilegio de seguridad: Alta

Nivel de experiencia: Media/Alta

* **Apoderado (usuario principal):** constituye el rol central del proyecto, ya que el sistema está diseñado principalmente para los padres y tutores responsables. El apoderado será quien registre la información de sus hijos (alimentación, actividad física, parámetros de crecimiento) y quien consulte los reportes y gráficos de progreso. Este rol refleja la convicción del proyecto: son los adultos quienes deben asumir la responsabilidad de guiar y acompañar la formación de hábitos saludables en los niños.

Privilegio de seguridad: Baja, acceso solo a los datos que el mismo ingresó y reportes generados de estos.

Nivel de experiencia: Básica con dispositivo móvil android o navegador móvil.

* **Niño (usuario pasivo):** aunque el beneficiario directo del proyecto es el niño, este no tendrá un acceso activo al sistema en la primera versión (MVP). Su rol será indirecto: a través de los registros hechos por el apoderado, el niño verá reflejado su progreso en los gráficos y estadísticas, fomentando la motivación y el diálogo con los padres. Esta decisión de diseño responde tanto a la factibilidad técnica como a la importancia de mantener la supervisión adulta como filtro principal.

Privilegio de seguridad: Inexistente

Nivel de experiencia: No aplica.

**Funcionalidades por usuario:**

**Apoderado:**

1. Crear cuenta e iniciar sesión.
2. Crear niño y cargar métricas iniciales.
3. Registrar alimentación/actividad diaria con ayuda de las herramientas.
4. Consultar reportes semanales/mensuales y tips.
5. Activar recordatorios.

**Administrador:**

1. Gestionar tips y categorías.
2. Configurar patrocinio y banderas.
3. Verificar consistencia de catálogos.

**Niño:** Sin acceso (beneficiario indirecto de los registros y reportes).

### 

### Entorno operativo

* Frontend: Ionic 7 + Angular 17.
* Backend: Firebase (Auth, Firestore, Hosting, opcional Functions).
* API externa: OpenFoodFacts (REST JSON, gratuita).
* Distribución: PWA + Android (Capacitor).
* Herramientas complementarias: GitHub para control de versiones, Chart.js para gráficos.

### 

### Requerimientos funcionales

**F1 - Autenticación y gestión de sesión**

* REQ-AUTH-01: Crear cuenta con email/contraseña.  
   Descripción: El apoderado ingresa correo y clave para registrarse y quedar autenticado.
* REQ-AUTH-02: Iniciar y cerrar sesión.  
   Descripción: Permite entrar a la app y salir cuando lo desee, mostrando mensajes claros.
* REQ-AUTH-03: Proteger pantallas internas.  
   Descripción: Solo usuarios con sesión activa pueden ver y usar las funciones privadas.

**F2 - Perfil del apoderado**

* REQ-PROFILE-01: Editar datos básicos (nombre y teléfono opcional).  
   Descripción: El usuario puede mantener su información actualizada desde “Perfil”.
* REQ-PROFILE-02: Configurar recordatorios.  
   Descripción: Permite activar/desactivar recordatorios y definir una hora preferida.

**F3 - Gestión de niño(s)**

* REQ-CHILD-01: Crear/editar niño(s) con datos básicos.  
   Descripción: Se registra nombre y datos mínimos para poder llevar seguimiento.
* REQ-CHILD-02: Historial de peso y estatura.  
   Descripción: Se guardan mediciones por fecha para ver evolución en el tiempo.
* REQ-CHILD-03: Límite razonable de niños por cuenta (5)  
   Descripción: Evita sobrecarga del sistema manteniendo un número máximo (configurable).

**F4 - Registro de alimentación diaria**

* REQ-MEAL-01: Registrar por franjas (desayuno, colaciones, almuerzo, cena).  
   Descripción: Facilita anotar qué comió el niño en cada momento del día.
* REQ-MEAL-02: Categoría simple y notas.  
   Descripción: Se etiqueta cada registro (ej.: fruta, lácteo) y se pueden agregar comentarios.
* REQ-MEAL-03: Editar/eliminar del día.  
   Descripción: Permite corregir o borrar registros recientes de forma sencilla.

**F5 - Registro de actividad física**

* REQ-ACT-01: Registrar tipo de actividad y minutos.  
   Descripción: Se anota lo realizado (caminar, jugar, deporte) y su duración.
* REQ-ACT-02: Editar/eliminar del día.  
   Descripción: Se corrigen o quitan actividades registradas por error.

**F6 - Reportes y gráficos**

* REQ-REP-01: Resumen semanal.  
   Descripción: Muestra, de forma simple, categorías de comida y minutos activos por semana.
* REQ-REP-02: Tendencia mensual.  
   Descripción: Permite ver el progreso durante el mes con un gráfico claro.
* REQ-REP-03: Compartir/guardar gráfico.  
   Descripción: Genera una imagen básica del reporte para enviar o guardar.

**F7 - Integración con OpenFoodFacts (OFF)**

* REQ-OFF-01: Búsqueda por texto/categoría.  
   Descripción: Al escribir el nombre del alimento, se muestran resultados de OFF.
* REQ-OFF-02: Consulta por código de barras.  
   Descripción: Permite ingresar o escanear el código para traer la ficha del producto.
* REQ-OFF-03: Registro manual si no hay resultados o falla la red.  
   Descripción: El apoderado puede ingresar el alimento a mano para no detener el registro.

**F8 - Ficha nutricional del alimento**

* REQ-NUTRI-01: Mostrar valores por 100 g/ml (kcal, azúcares, grasas saturadas, sodio).  
   Descripción: Se presentan los datos clave que ayudan a entender el producto.
* REQ-NUTRI-02: Calcular por porción indicada.  
   Descripción: El usuario escribe la porción y la app estima los valores para esa cantidad.
* REQ-NUTRI-03: Mostrar Nutri-Score cuando exista.  
   Descripción: Si OFF lo entrega, se muestra la letra A–E como referencia general.

**F9 - Calificación educativa del alimento**

* REQ-HEALTH-01: Mostrar alertas (energía/azúcares/grasas sat./sodio) por 100 g/ml.  
   Descripción: Si superan límites de referencia, se muestran avisos simples (“Azúcares altos”, etc.).
* REQ-HEALTH-02: Etiquetar Saludable/Moderado/Limitar.  
   Descripción: La app resume la evaluación con una etiqueta y una explicación breve.

**F10 - Tips educativos (sin marcas)**

* REQ-TIPS-01: Mostrar 1–3 tips en la pantalla inicial.  
   Descripción: Consejos cortos, rotativos, para apoyar hábitos saludables.
* REQ-TIPS-02: Gestión de tips por Administrador.  
   Descripción: Permite crear, editar o desactivar tips desde un panel simple.

**F11 - Módulo de patrocinio (MVP)**

* REQ-SPONSOR-01: Nota visible “Patrocinado”.  
   Descripción: Toda tarjeta o sección del sponsor se identifica claramente como patrocinada.
* REQ-SPONSOR-02: Activar/desactivar patrocinio desde una bandera remota.  
   Descripción: Permite prender o apagar el bloque del patrocinador sin cambiar la app.
* REQ-SPONSOR-03 (opcional): Botón “Dónde comprar” con enlace.  
   Descripción: Dirige a la tienda del sponsor cuando corresponde.
* REQ-SPONSOR-04: Separar patrocinio de tips y reportes.  
   Descripción: Evita confusiones manteniéndolo en secciones distintas.

\* En el MVP no se filtra el contenido patrocinado por su calidad nutricional; solo se identifica con la nota “Patrocinado”.

**F12 - Notificaciones locales**

* REQ-NOTIF-01: Activar/desactivar recordatorios diarios.  
   Descripción: El apoderado decide si quiere avisos para registrar hábitos.
* REQ-NOTIF-02: Definir hora preferida.  
   Descripción: Se fija un horario conveniente para recibir el aviso.

**F13 - Distribución (PWA + Android)**

* REQ-DIST-01: PWA instalable.  
   Descripción: La app se puede “instalar” desde el navegador y usar como app.
* REQ-DIST-02: Se genera APK para pruebas en Android.  
   Descripción: Se crea un instalador sencillo para probar en dispositivos reales.

**F14 - Panel Admin (semilla)**

* REQ-ADMIN-01: Gestionar tips y banderas de funciones.  
   Descripción: Permite a un administrador ajustar contenido y encender/apagar módulos.
* REQ-ADMIN-02: Sin acceso a datos personales de familias.  
   Descripción: El panel no incluye información privada de apoderados ni niños.

### 

### Reglas de negocio

1. El apoderado es el único que registra información del niño; el niño no accede en el MVP.
2. MiniFit es educativo, no clínico; no prescribe dietas ni tratamientos.
3. Los tips son neutrales (sin marcas); el patrocinio se muestra solo con badge y en módulos separados.
4. No se promueven productos que no cumplan filtros saludables.
5. Los datos de menores son privados para el apoderado; no se comparten con terceros.
6. La municipalidad, si participa, solo accede a datos agregados y anonimizados (futuro).
7. Límite de 5 niños por cuenta (configurable).
8. Límite de registros diarios (p. ej., 25 entradas por niño/día).
9. Las banderas permiten activar/desactivar patrocinio y funciones sin redeploy.
10. Toda aparición de patrocinio debe ir con etiqueta visible “Patrocinado”.

### Requerimientos de interfaces externas

#### Interfaces de usuario

MiniFit es una PWA y app Android pensada para padres/apoderados. Las pantallas son sencillas, con textos breves y botones grandes.

**Tipos / áreas con interfaz distinta**

* Acceso: Login/Registro, Recuperar contraseña.
* Hogar (inicio): accesos rápidos a registrar comida/actividad y ver tips.
* Registro:
  + *Alimentación* (desayuno, colación, almuerzo, cena).
  + *Actividad física* (tipo + minutos).
* Niños: alta/edición y métricas (peso/estatura).
* Reportes: semanal y mensual (gráficos simples).
* Alimentos: búsqueda/escáner, ficha nutricional y calificación educativa.  
  Patrocinio: bloques y tarjetas con nota visible “Patrocinado” (no se filtra contenido por salud).
* Perfil: datos del apoderado y recordatorios.
* Admin (semilla): gestión de tips y textos del patrocinio (sin datos personales).

**Guías y estándares GUI**

* Estructura: encabezado con título y botón “atrás”; contenido en tarjetas; acciones principales visibles.
* Botones: principal “Guardar/Agregar”, secundario “Cancelar/Editar/Eliminar”. Estados: normal, deshabilitado, cargando.
* Formularios: etiquetas claras, validaciones al enviar, mensajes simples; campos obligatorios marcados.
* Listas: paginadas o con “ver más”.
* Mensajería: toasts para éxito/errores; estados “sin datos” con texto guía.
* Accesibilidad: contrastes legibles, tamaños táctiles amplios, foco visible, soporte lector de pantalla.
* Lenguaje: español claro; sin tecnicismos.  
  Patrocinio: cada pieza muestra badge “Patrocinado”; no se mezcla con tips ni con gráficos.

*Ejemplos de pantalla:* Login, Inicio, Registrar alimentación, Reporte semanal, Ficha de producto, Bloque de patrocinio, Admin de tips.

#### 

#### Interfaces de hardware

**Dispositivos soportados**

* Computadores (navegador moderno) para la PWA.
* Dispositivos móviles Android (APK) y PWA en móviles.
* iOS: uso vía navegador (PWA); *app nativa fuera de este MVP*.
* Impresoras / otros periféricos: no aplican en el MVP.

**Interacciones de datos y control**

* Cámara / escáner de código de barras: lectura del código para consultar productos.
  + Solicitud de permiso solo al usar el escáner.
  + No se guardan fotos; se captura solo el código leído.
* Notificaciones locales: programación de recordatorios en el sistema operativo.
* Vibración / sonido: comportamiento por defecto del sistema (sin personalización especial).

**Protocolos de comunicación con hardware**

* Conectividad por Wi-Fi/4G/5G; no se usa Bluetooth/NFC.
* Acceso a cámara y notificaciones mediante plugins de Capacitor/Ionic autorizados por el SO.

#### 

#### Interfaces de software

**Componentes y sistemas externos**

* OpenFoodFacts (API): búsqueda por texto/categoría y consulta por código de barras.
  + Formato: JSON.
  + Campos clave (cuando existan): code, product\_name, brands, nutrition\_grades (Nutri-Score), nutrientes por 100 g/ml (energía, azúcares, grasas saturadas, sodio).
* Firebase Authentication: creación de cuentas, inicio/cierre de sesión.
* Firestore (Firebase): almacenamiento de perfiles, niños, comidas, actividades, tips, textos de patrocinio.
* Hosting/Deploy: Firebase Hosting (PWA) y build Android con Capacitor.
* Librerías cliente: Ionic/Angular, Chart.js (gráficos).

**Requisitos de integración**

* No enviar datos personales a OpenFoodFacts.
* Si OFF no responde o faltan datos, permitir registro manual del alimento.
* Reglas de Firestore: cada documento tiene dueño; solo el dueño lee/escribe lo suyo.
* Manejo de errores y mensajes claros (reintentar / flujo manual).

**Interfaces de comunicación**

**Funciones de comunicación necesarias**

* Navegador web para la PWA y conectividad de red para consultas externas.
* Enlaces externos para el botón “Dónde comprar” del patrocinador (abre fuera de la app).
* Correo electrónico: no se envía correo desde MiniFit (solo podría usarse el flujo estándar de recuperación de Firebase si se habilita).

**Estándares y formatos**

* Protocolos: HTTPS (TLS 1.2 o superior) para todo el tráfico.
* Estilo de API: REST sobre HTTP; JSON como formato de intercambio.
* CORS: habilitado para la PWA desde el dominio de despliegue.  
  Idiomas: preferencia es-CL cuando el servicio externo lo soporte.

**Seguridad en las comunicaciones**

* Todo viaja cifrado (HTTPS).
* Sesión manejada por el SDK de Firebase; no se comparten tokens con terceros.
* No se registran ni transmiten datos sensibles de niños a servicios externos.
* Enlaces del patrocinador se abren en navegador/app externa sin adjuntar datos del usuario.

**Tiempos y fallos**

* Si una consulta externa supera un tiempo razonable, mostrar aviso y permitir continuar manualmente.
* Reintentos limitados y mensajes simples.
* La falta de red no bloquea el registro de comidas/actividades (se permite anotar manualmente y continuar).

### 

### Requerimientos no funcionales

**Seguridad y privacidad (RNF-SEC)**

* RNF-SEC-01: Cada usuario solo ve sus propios datos.
* RNF-SEC-02: Minimizar datos personales solicitados.
* RNF-SEC-03: Onboarding con explicación y aceptación del uso de datos.
* RNF-SEC-04: Opción de eliminar cuenta y datos.
* RNF-SEC-05: No compartir datos personales con el patrocinador.
* RNF-SEC-06: Reportes futuros a municipios solo como resúmenes sin nombres.

**Rendimiento (RNF-PERF)**

* RNF-PERF-01: App liviana y rápida en celulares comunes y 4G.
* RNF-PERF-02: Listas y gráficos muestran solo lo necesario.
* RNF-PERF-03: Consultas externas eficientes y reutilización de resultados recientes cuando sea posible.

**Confiabilidad (RNF-REL)**

* RNF-REL-01: Si falla OFF o no hay internet, permitir registro manual.
* RNF-REL-02: Mensajes claros ante errores y reintentos automáticos.
* RNF-REL-03: Evitar duplicados por toques repetidos.

**Usabilidad y accesibilidad (RNF-UX)**

* RNF-UX-01: Interfaz simple (botones grandes, textos cortos).
* RNF-UX-02: Buena lectura (contraste y tamaño legible).
* RNF-UX-03: Formularios con etiquetas y mensajes sencillos.
* RNF-UX-04: Uso cómodo en pantallas pequeñas y con lector de pantalla.

**Mantenibilidad (RNF-MAINT)**

* RNF-MAINT-01: Código ordenado por pantallas/funciones, nombres claros.
* RNF-MAINT-02: Pruebas básicas en login, registros y ficha nutricional.
* RNF-MAINT-03: Guía breve para instalar/publicar y facilitar traspaso.

**Portabilidad (RNF-PORT)**

* RNF-PORT-01: Funciona como PWA y como APK Android.
* RNF-PORT-02: Si no hay cámara, se puede ingresar alimento/código a mano.
* RNF-PORT-03: Diseño responsive para distintos tamaños de pantalla.

**Legal y ética (RNF-ETH)**

* RNF-ETH-01: Contenido educativo; no sustituye consejo médico.
* RNF-ETH-02: Prioridad a la privacidad de los niños.
* RNF-ETH-03: Patrocinio siempre identificado con “Patrocinado”.
* RNF-ETH-04: En el MVP no se filtra el contenido patrocinado; solo se identifica.
* RNF-ETH-05: Mantener el patrocinio separado de tips y reportes.

### 

### 

### Otros requerimientos

Requerimientos no cubiertos en ninguna otra sección del documento de requerimientos de software, por ejemplo: Requerimientos de bases de datos, internacionalización, legales y objetivos de reúso de componentes de software.

### Glosario

* **Apoderado:** Adulto responsable que usa la app para registrar y revisar hábitos del niño.
* **Niño (beneficiario):** Menor cuyos hábitos se registran; no usa la app en el MVP.
* **Administrador (Admin):** Persona que gestiona tips y textos del patrocinio; no ve datos personales.
* **MVP (Producto Mínimo Viable):** Primera versión funcional y acotada del proyecto.
* **PWA (Aplicación Web Progresiva):** Sitio que se puede instalar como app desde el navegador.
* **APK:** Archivo instalable de Android para probar la app en teléfonos.
* **Ionic / Angular:** Frameworks que usamos para construir la interfaz y la lógica de la app.
* **Capacitor (plugins):** Puente que permite usar funciones del teléfono (cámara, notificaciones) desde Ionic.
* **Chart.js:** Librería para mostrar gráficos simples en la app.
* **OpenFoodFacts (OFF):** API pública con información de alimentos (nombre, marca, nutrientes, etc.)
* **Nutri-Score (A–E):** Letra que resume la calidad nutricional de un producto (A es mejor, E es peor).
* **Código de barras:** Identificador del producto que se puede leer con cámara o escribir a mano.
* **Ficha nutricional:** Vista que muestra calorías, azúcares, grasas saturadas y sodio (por 100 g/ml y por porción).
* **Calificación educativa:** Etiqueta que la app asigna a un alimento: Saludable / Moderado / Limitar, con explicación breve.
* **Firebase Authentication (Auth):** Servicio para crear cuentas, iniciar y cerrar sesión.
* **Firestore:** Base de datos en la nube donde guardamos perfiles, niños, comidas, actividades, tips y textos del patrocinio.
* **Reglas de seguridad (Firestore Rules):** Permisos que aseguran que cada usuario solo vea sus propios datos.
* **Emuladores de Firebase:** Entorno local para probar sin tocar datos reales.
* **Patrocinio (MVP):** Bloque con contenido identificado con la **nota “Patrocinado”** y botón “Dónde comprar”.
* **Interfaz de usuario (UI):** Pantallas que ve el usuario (Login, Inicio, Registrar, Reportes, Ficha alimento, etc.).
* **Accesibilidad:** Buen contraste, textos claros, tamaños táctiles amplios y soporte para lector de pantalla.
* **Guía de estilo:** Botones principales visibles, formularios con etiquetas, mensajes simples y estados “sin datos”.
* **Interfaces de hardware:** Uso de **cámara/escáner** para leer códigos y **notificaciones locales** del sistema.
* **Interfaces de software:** Conexiones con OFF, Firebase Auth, Firestore, y librerías cliente (Ionic/Angular, Chart.js).
* **Interfaces de comunicación:** Navegador, red, y apertura de enlaces externos (por ejemplo, del patrocinador).
* **REST:** Forma de acceder a servicios por web usando URLs y métodos (GET, POST, etc.).
* **JSON:** Formato de datos liviano que usan las APIs para enviar y recibir información.
* **HTTPS / TLS:** Capa de seguridad para cifrar las comunicaciones en red.
* **CORS:** Permiso que permite a la PWA consultar servicios externos desde su dominio.
* **Notificaciones locales:** Avisos que programa el teléfono para recordar registrar hábitos.
* **Permisos (cámara/notificaciones):** Autorizaciones del sistema que el usuario acepta o rechaza.
* **Fallback manual:** Alternativa para seguir registrando cuando no hay internet o la API no responde.  
  **Datos personales (PII):** Información de una persona (como nombre y contacto). Se minimiza y no se comparte con terceros.
* **Criterios de aceptación:** Condiciones claras que deben cumplirse para decir que un requisito está logrado.
* **Caso de prueba (TC):** Pasos y resultado esperado para comprobar un requisito.
* **Requerimientos no funcionales (RNF):** Condiciones de calidad (seguridad, velocidad, accesibilidad, etc.).