1. **Stack base (Next.js + Supabase)**
2. **Librerías recomendadas para cada página/sección**
3. **IA pendiente → qué ya existe y puedes integrar rápido**

**1️⃣ Stack base**

* **Frontend:** Next.js (con App Router)
* **Backend-as-a-Service:** Supabase (Auth, DB, Storage, Realtime)
* **Estilos/UI:** TailwindCSS + shadcn/ui (rápido y moderno)
* **Gráficas:** Recharts o Chart.js
* **Pasarela de pago:** LemonSqueezy o Paddle (tienen SDK + webhooks)

**2️⃣ Librerías por página/contenido**

**1. Onboarding / Registro**

* **Supabase Auth Helpers** → @supabase/auth-helpers-nextjs
* **OAuth** integrado en Supabase (Google, Apple, GitHub, etc.)
* **Formularios** → react-hook-form + zod (validación).
* **UI** → @headlessui/react + shadcn/ui.

**2. Home / Dashboard**

* **Charts** → recharts (más personalizable) o react-chartjs-2.
* **Estado global** → zustand o jotai para manejar progreso.
* **Notificaciones** → react-hot-toast.

**3. Entrenamientos**

* **Videos/animaciones** → react-player o lottie-react para ejercicios animados.
* **Player optimizado** (para rutinas paso a paso) → video.js.
* **Personalización de rutinas** → lógica en DB de Supabase (tablas de planes, ejercicios).

**4. Planificación**

* **Calendario** → react-big-calendar o fullcalendar-react.
* **Integración Google Calendar/iCal** → googleapis (Node SDK).

**5. Nutrición (futuro)**

* **Macros y calorías** → tablas Supabase + cálculo en backend.
* **Scanner de código de barras (IA/ML)** → react-qr-barcode-scanner o integración con **Open Food Facts API** (gratis).

**6. Seguimiento de Progreso (IA futuro)**

* **Gráficas** → recharts.
* **Comparación de fotos antes/después** → aquí sí IA → Computer Vision modelo ya existente (ej: **Roboflow**, **TensorFlow.js**, o API de **Replicate**).

**7. Comunidad / Social**

* **Chat y feed** → Supabase Realtime (canales tipo chat).
* **Foro básico** → Tablas en Supabase (posts, likes, comments).
* **Ranking global** → queries SQL en Supabase.

**8. Perfil de Usuario**

* **Storage de imágenes** → Supabase Storage (fotos de perfil).
* **Formularios de datos** → react-hook-form.
* **Notificaciones push** → OneSignal o Firebase Cloud Messaging (FCM).

**9. Premium / Tienda**

* **Pagos** → SDK oficial de **LemonSqueezy** (@lemonsqueezy/lemonsqueezy.js) o **Paddle**.
* **Gestión de suscripción** → webhook que actualiza rol en Supabase (ej: free → premium).

**3️⃣ IA pendiente (qué ya existe para integrar)**

**🔹 Entrenador con IA**

* Ya existen **modelos generativos de entrenamiento** que puedes consumir vía API:
  + **OpenAI (GPT-4o-mini)** → generar rutinas personalizadas en texto.
  + **Replicate** → algunos modelos de planificación fitness.
  + **LangChain.js** → para orquestar reglas con IA y datos del usuario.

👉 Aquí no necesitas entrenar un modelo propio, puedes integrar **ChatGPT API** y darle contexto del usuario (objetivo, tiempo, equipo).

**🔹 Adaptación diaria**

* Se puede resolver con **reglas + IA ligera**:
  + Si ayer no entrenó → bajar carga.
  + Si durmió poco (integración con wearable) → IA recomienda rutina ligera.
  + GPT puede generar ese plan dinámico.

**🔹 Scanner de alimentos (Nutrición)**

* No necesitas IA desde cero.
* Usa **Open Food Facts API** o **Nutritionix API** → devuelven calorías/macros al escanear código de barras.
* IA solo para “recomendar dieta” (GPT).

**🔹 Seguimiento con fotos**

* Ya hay modelos de **pose estimation** y **body measurement** que puedes integrar:
  + **TensorFlow.js PoseNet / MoveNet** (gratis y corre en navegador).
  + **MediaPipe de Google** (detectar posturas, contar repeticiones).

**🔹 AR / Corrección postural (a futuro)**

* **MediaPipe + Three.js** → ya hay demos de corrección postural en tiempo real.
* No necesitas entrenar IA propia.

**🚀 Resumen**

* **Supabase + Next.js** cubre el 80% (Auth, DB, Storage, Realtime).
* **UI/UX rápido** → Tailwind + shadcn/ui.
* **Charts / calendario / video** → librerías React ya maduras.
* **Pagos** → LemonSqueezy o Paddle (webhooks → Supabase roles).
* **IA**:
  + Rutinas → GPT-4o-mini (ya disponible).
  + Nutrición (scanner) → Open Food Facts API.
  + Seguimiento fotos/posturas → TensorFlow.js / MediaPipe.

👉 Te conviene empezar con un **MVP**:

1. Auth + Perfil + Dashboard básico.
2. Entrenamientos predefinidos.
3. Comunidad (retos simples).
4. Luego agregar IA y Nutrición.

**📌 Roadmap de desarrollo por fases**

**🟢 Fase 1 – MVP (3 a 6 meses)**

🎯 Objetivo: lanzar una app **funcional y atractiva**, validar que la gente la use.

**Páginas/Secciones clave**

1. **Onboarding / Registro**
   * Supabase Auth (email + Google/Apple).
   * Preferencias iniciales (objetivo, tiempo, equipo disponible).
2. **Dashboard simple**
   * Resumen diario (entreno planificado).
   * Gráfico de progreso básico (ej: entrenamientos completados).
3. **Entrenamientos básicos**
   * Catálogo de rutinas (estáticas, predefinidas).
   * Ejercicios con video/animaciones.
   * Quick workout.
4. **Perfil de usuario**
   * Datos básicos (edad, peso, altura).
   * Foto de perfil (Supabase Storage).
5. **Comunidad básica**
   * Retos grupales (ej: “Entrena 3 veces esta semana”).
   * Ranking simple (quién entrenó más días).

**Tech/Librerías mínimas**

* nextjs + @supabase/auth-helpers-nextjs
* tailwindcss + shadcn/ui
* react-hook-form + zod
* recharts (gráficos)
* react-player (videos de ejercicios)

**🟡 Fase 2 – Beta (6 a 12 meses)**

🎯 Objetivo: aumentar retención con **personalización + gamificación**.

**Nuevas funciones**

1. **Dashboard avanzado**
   * Progreso semanal/mensual con métricas más completas.
   * Recordatorios automáticos (notificaciones push → OneSignal o FCM).
2. **Planificación**
   * Calendario de entrenamientos.
   * Ajuste de días de descanso.
3. **Entrenador con IA (v1)**
   * Usar GPT-4o-mini para generar rutinas personalizadas según preferencias.
   * Adaptación diaria con reglas simples (si no entrenó ayer → rutina ligera).
4. **Gamificación**
   * Logros y medallas (ej: “100 flexiones completadas”).
   * Niveles de experiencia.
5. **Suscripción Premium**
   * Integrar pagos con **LemonSqueezy** o **Paddle**.
   * Plan mensual → desbloquea rutinas avanzadas y comunidad pro.

**Extra**

* Guardar estadísticas de entrenamientos (sets, reps, tiempo).
* Ranking global y entre amigos.

**🔵 Fase 3 – Escalado Global (12 a 24 meses)**

🎯 Objetivo: diferenciarse con **IA avanzada, nutrición y social fuerte**.

**Nuevas funciones**

1. **Nutrición**
   * Registro de macros y calorías.
   * Scanner de alimentos (Open Food Facts API).
   * IA para sugerencias de dieta básica.
2. **Seguimiento de progreso con IA**
   * Subida de fotos → comparación antes/después.
   * Pose detection con **TensorFlow.js / MediaPipe** → corrección postural y rep counting.
3. **Entrenamientos en vivo**
   * Sesiones grupales vía video (WebRTC o integración con Daily.co/LiveKit).
   * Retos diarios en tiempo real.
4. **Social fuerte**
   * Feed estilo Instagram para compartir progreso.
   * Retos globales estilo “Maratón Virtual”.
5. **Recomendaciones inteligentes**
   * Recordatorios de hidratación y descanso.
   * IA ajusta carga según sueño/pasos (si conectas con wearables: Apple Watch, Fitbit, Garmin, Strava).

**🟣 Fase 4 – Diferenciación avanzada (24+ meses)**

🎯 Objetivo: destacar frente a apps grandes (Nike, Freeletics, Fitbod).

* **Realidad aumentada (AR)**
  + Corrección postural con cámara.
  + Guías visuales proyectadas en el entorno del usuario.
* **Marketplace global**
  + Entrenadores independientes pueden subir rutinas.
  + Sistema de revenue sharing (tu app como plataforma).
* **IA de salud integral**
  + Personalización no solo en entreno, sino en sueño, estrés, recuperación.
  + Integración con datos biométricos (API Apple Health, Google Fit, Whoop).

**🚀 Estrategia de implementación**

* **Fase 1 (MVP)** = 20% de features → 80% de valor.
* **Fase 2** = enganchar al usuario → IA básica + gamificación.
* **Fase 3** = escalar global con nutrición + social.
* **Fase 4** = diferenciar con AR + marketplace.

👉 Así aseguras que no te quemas de inicio, pero desde la Fase 1 ya puedes lanzar algo que **enganche y valide mercado**.

¿Quieres que te arme también la **estructura de base de datos en Supabase (tablas y relaciones)** según este roadmap para que empieces ya bien organizado?

|  |  |  |  |
| --- | --- | --- | --- |
| Proveedor | Requerimientos principales | Dificultad | Comentarios |
| Google | - Cuenta en Google Cloud Console - Crear un proyecto - Activar “OAuth consent screen” - Configurar “Authorized redirect URI” (https://<project-ref>.supabase.co/auth/v1/callback) | ⭐⭐ (fácil) | Documentación clara. Validación rápida si es app de pruebas. Ideal como primer OAuth. |
| GitHub | - Cuenta GitHub - Crear una “OAuth App” - Configurar “Authorization callback URL” | ⭐ (muy fácil) | Uno de los más simples de implementar. Popular para devs. |
| Twitch | - Cuenta Twitch - Activar 2FA - Crear aplicación en [Twitch Dev Console](https://dev.twitch.tv/console/apps) - Configurar redirect URI | ⭐⭐ (medio) | UI sencilla pero obliga a tener 2FA activo. Suele dar problemas con SMS. |
| Facebook | - Cuenta en [Facebook Developers](https://developers.facebook.com/) - Crear “App” - Activar producto “Facebook Login” - Verificación de app si es pública - Redirect URI | ⭐⭐⭐⭐ (difícil) | Proceso engorroso. A veces bloquea funciones sin verificación de negocio. Puede pedir revisión de privacidad. |
| LinkedIn | - Cuenta LinkedIn con [LinkedIn Developers](https://www.linkedin.com/developers/) - Crear aplicación - Configurar redirect URI - Solicitar permisos adicionales (ej. email, perfil completo) | ⭐⭐⭐ (medio-alto) | Su API es más restrictiva. Para datos de perfil avanzados necesitas aprobación. |
| Twitter/X | - Cuenta en [Developer Portal](https://developer.twitter.com/) - Crear app + proyecto - Definir redirect URI - Acceso a nivel “Elevated” si quieres datos de email | ⭐⭐⭐ (medio) | API cambió bastante desde que es “X”. Obtener email requiere aprobación. |

**🗄️ Estructura de Base de Datos – Supabase**

**Usuarios y Perfil**

-- Usuarios base (Supabase ya maneja la tabla auth.users)

create table profiles (

id uuid primary key references auth.users(id) on delete cascade,

username text unique,

full\_name text,

age int,

height numeric(5,2), -- en cm

weight numeric(5,2), -- en kg

goal text, -- bajar peso, ganar masa, rendimiento, salud

equipment text[], -- lista de equipo disponible

preferences jsonb, -- tiempo disponible, lesiones, etc.

avatar\_url text,

created\_at timestamp default now()

);

**Entrenamientos y Ejercicios**

create table exercises (

id uuid primary key default gen\_random\_uuid(),

name text not null,

category text, -- fuerza, cardio, movilidad

video\_url text,

description text,

created\_at timestamp default now()

);

create table workouts (

id uuid primary key default gen\_random\_uuid(),

name text not null,

type text, -- rutina predefinida, AI, quick workout

duration int, -- en minutos

difficulty text, -- beginner, intermediate, advanced

created\_by uuid references profiles(id), -- entrenador o sistema

created\_at timestamp default now()

);

-- Relación n a n entre workout y ejercicios

create table workout\_exercises (

id uuid primary key default gen\_random\_uuid(),

workout\_id uuid references workouts(id) on delete cascade,

exercise\_id uuid references exercises(id) on delete cascade,

sets int,

reps int,

rest int -- segundos de descanso

);

**Progreso y Seguimiento**

create table user\_workouts (

id uuid primary key default gen\_random\_uuid(),

user\_id uuid references profiles(id) on delete cascade,

workout\_id uuid references workouts(id),

completed boolean default false,

completed\_at timestamp

);

-- Para métricas (peso, % grasa, medidas, etc.)

create table progress (

id uuid primary key default gen\_random\_uuid(),

user\_id uuid references profiles(id) on delete cascade,

metric text, -- weight, fat\_percentage, chest, waist, etc.

value numeric(10,2),

recorded\_at timestamp default now()

);

**Comunidad / Social**

create table challenges (

id uuid primary key default gen\_random\_uuid(),

name text,

description text,

start\_date date,

end\_date date,

reward text,

created\_at timestamp default now()

);

create table user\_challenges (

id uuid primary key default gen\_random\_uuid(),

user\_id uuid references profiles(id) on delete cascade,

challenge\_id uuid references challenges(id) on delete cascade,

progress int default 0, -- % completado

completed boolean default false

);

create table posts (

id uuid primary key default gen\_random\_uuid(),

user\_id uuid references profiles(id) on delete cascade,

content text,

image\_url text,

created\_at timestamp default now()

);

create table post\_likes (

id uuid primary key default gen\_random\_uuid(),

post\_id uuid references posts(id) on delete cascade,

user\_id uuid references profiles(id) on delete cascade

);

create table post\_comments (

id uuid primary key default gen\_random\_uuid(),

post\_id uuid references posts(id) on delete cascade,

user\_id uuid references profiles(id) on delete cascade,

comment text,

created\_at timestamp default now()

);

**Pagos y Suscripciones**

create table subscriptions (

id uuid primary key default gen\_random\_uuid(),

user\_id uuid references profiles(id) on delete cascade,

status text, -- active, canceled, trial

provider text, -- lemonsqueezy, paddle

external\_id text, -- id de la pasarela

started\_at timestamp,

expires\_at timestamp

);

**(Futuro) Nutrición**

create table foods (

id uuid primary key default gen\_random\_uuid(),

name text,

brand text,

calories int,

protein numeric(5,2),

carbs numeric(5,2),

fat numeric(5,2),

barcode text unique

);

create table meals (

id uuid primary key default gen\_random\_uuid(),

user\_id uuid references profiles(id) on delete cascade,

food\_id uuid references foods(id),

quantity numeric(5,2), -- gramos

consumed\_at timestamp default now()

);