



PROUESTA PROYECTO DAW

Módulo “Proyecto de Desarrollo de Aplicaciones Web”

Alumna: Elena Ardila Delgado

Fecha: 29 de Septiembre de 2025

Contenido

1. Título y descripción general del proyecto	2
2. Identificación del proyecto	2
3. Justificación y objetivos	2
4. Contenidos y aspectos principales.....	3
4.1 Requisitos funcionales.....	3
4.2 Requisitos no funcionales	5
4.3 Roles y permisos.....	7
5. Medios que se utilizarán	7
5.1 Sistema operativo.....	7
5.2 Lenguajes de programación usados.....	7
5.3 Frameworks y librerías	8
5.4 Entorno de desarrollo	8
6. Áreas del ciclo formativo	9



1. Título y descripción general del proyecto

FitMind AI es una aplicación web que genera rutinas de entrenamiento y tablas de dieta semanales personalizadas según edad, peso y objetivo del usuario. La IA funciona como un servicio externo (API) y la app no tiene backend propio: todo corre en el cliente (React + JavaScript) y usa Supabase para autenticación, base de datos y almacenamiento.

2. Identificación del proyecto

- Participante: Elena Ardila Delgado
- Ciclo formativo: Desarrollo de Aplicaciones Web
- Centro educativo: IES Albarregas, Mérida.

3. Justificación y objetivos

Las apps de fitness/nutrición suelen ser rígidas o exigir suscripción a servicios clínicos. Este proyecto propone un generador auto-servicio que combina reglas + IA como API, manteniendo un stack simple (front-only) y datos en Supabase. Se cubre así la necesidad de planes rápidos, editables y explicables, sin montar una infraestructura backend completa.

4. Contenidos y aspectos principales

4.1 Requisitos funcionales

RF1. Invitado: ver *landing* y registrarse

- Actor: Invitado.
- Descripción: Un usuario sin cuenta puede acceder a la página principal (*landing*) donde se presentan las características de la aplicación. Desde ahí, dispone de la opción de registrarse completando un formulario con email y contraseña.
- Resultado esperado: El invitado crea su cuenta y pasa a ser usuario registrado.

RF2. Registro/Login

- Actor: Invitado / Usuario registrado.
- Descripción: La aplicación utiliza Supabase Auth para gestionar de forma segura el registro, inicio de sesión y cierre de sesión de los usuarios. El sistema permite recuperar contraseña por correo electrónico.
- Resultado esperado: El usuario puede autenticarse en la aplicación y acceder a sus datos personales y planes generados.

RF3. Perfil editable

- Actor: Usuario registrado.
- Descripción: El usuario puede completar y actualizar en cualquier momento los datos de su perfil: edad, sexo, altura, peso, nivel de actividad y objetivo (perder, mantener o ganar peso). También puede añadir alergias o preferencias (ej. vegetariano) que se almacenan en un campo JSON. Además, el usuario puede subir y actualizar su avatar, almacenado en Supabase Storage.



- Resultado esperado: El perfil siempre refleja los datos actuales del usuario y se utilizan para personalizar los planes.

RF4. Generar plan semanal llamando a la API de IA

- Actor: Usuario registrado.
- Descripción: El sistema envía los datos del perfil y de la semana seleccionada a la API de IA, la cual devuelve una propuesta de rutina de entrenamiento y un menú semanal. Esta información se guarda en la base de datos Supabase para que el usuario pueda consultarla en cualquier momento.
- Resultado esperado: El usuario obtiene un plan semanal completo adaptado a sus objetivos y preferencias.

RF5. Sustituciones isocalóricas / mismo grupo muscular

- Actor: Usuario registrado.
- Descripción: El usuario puede solicitar el cambio de un plato de la dieta o de un ejercicio de la rutina. El sistema consulta la API de IA para proponer alternativas que cumplan con el mismo grupo muscular (en el caso de ejercicios) o que mantengan las calorías y macros (en el caso de platos).
- Resultado esperado: El usuario consigue flexibilidad en sus planes sin romper el equilibrio calórico o de entrenamiento.

RF6. Descargar PDF de la dieta (día o semana)

- Actor: Usuario registrado.
- Descripción: La aplicación permite exportar la dieta generada a un archivo PDF. El documento incluye una tabla con las comidas de cada día, sus valores calóricos y de macronutrientes, además de un pie con notas y un aviso legal.
- Resultado esperado: El usuario puede descargar, imprimir o compartir su dieta en un formato portable.

RF7. Administrador: CRUD de Ejercicio y Plato y gestión básica de usuarios

- Actor: Administrador.
- Descripción: El administrador puede crear, modificar o eliminar ejercicios y platos de la base de datos, manteniendo actualizado el catálogo que utiliza la IA. Además, puede consultar la lista de usuarios registrados y activar o desactivar sus cuentas si fuera necesario.
- Resultado esperado: La base de datos de ejercicios y platos se mantiene actualizada y los usuarios están correctamente gestionados.

4.2 Requisitos no funcionales

RNF1. Arquitectura: front-only (React + JS)

- Descripción: La aplicación está construida como Single Page Application (SPA) usando React y JavaScript. No se dispone de un backend propio: toda la lógica reside en el cliente, mientras que los servicios de datos y autenticación se delegan a Supabase y la generación de planes a la API de IA.
- Resultado esperado: El sistema es ligero, fácil de desplegar en servicios de hosting estático (Vercel).

RNF2. Seguridad: autenticación y control de acceso

- Descripción: Se utiliza Supabase Auth para la autenticación de usuarios, con contraseñas cifradas y soporte para recuperación vía correo. Las tablas en la base de datos cuentan con Row Level Security (RLS), que asegura que cada usuario solo pueda acceder a sus propios registros. Además, el almacenamiento de avatares en Supabase Storage tiene políticas de acceso restringido por usuario. En el cliente se aplican validaciones de datos y restricciones de formularios.
- Resultado esperado: El acceso a la información está protegido, y los datos de los usuarios no pueden ser manipulados ni consultados por terceros no autorizados.

RNF3. Privacidad: cumplimiento básico de protección de datos

- Descripción: La aplicación incluye un aviso legal y política de privacidad, visibles desde la landing. El usuario dispone de una opción para eliminar su cuenta, lo que implica borrar o anonimizar todos sus datos del sistema. Los datos personales (edad, peso, objetivos) se usan únicamente para generar los planes y nunca se comparten con terceros.
- Resultado esperado: Se garantiza la transparencia en el uso de los datos y se respeta el derecho del usuario a la supresión de su información.

RNF4. Rendimiento: optimización de cliente y llamadas externas

- Descripción: Los catálogos de ejercicios y platos se almacenan en caché local del cliente para evitar llamadas repetitivas a la base de datos. Las consultas a la API de IA se optimizan: si requieren clave privada, se canalizan a través de una Supabase Edge Function que actúa como proxy seguro. El tiempo de respuesta para la generación de un plan semanal no debe superar los 3 segundos en condiciones normales.
- Resultado esperado: El usuario percibe una aplicación ágil y fluida, con tiempos de carga reducidos incluso en conexiones limitadas.

RNF5. Calidad: pruebas y validaciones

- Descripción: El sistema incluye pruebas unitarias en las funciones críticas del cliente (cálculo de macros, adaptadores de datos de la IA). Se realizan pruebas end-to-end (E2E) con Cypress para validar los flujos principales (registro, login, generación de plan, descarga de PDF). Además, se verifican periódicamente las políticas de RLS para asegurar que la seguridad de los datos se cumple correctamente.
- Resultado esperado: La aplicación mantiene un nivel de calidad estable, evitando regresiones y asegurando el cumplimiento de los requisitos antes de cada entrega.

4.3 Roles y permisos

Acción	Invitado	Registrado	Admin
Ver landing	✓	✓	✓
Registro / Login	✓	✓	✓
Editar perfil		✓	
Generar/consultar rutina y dieta		✓	
Sustituir plato/ejercicio		✓	
Descargar dieta en PDF		✓	
CRUD ejercicios/platos, gestionar usuarios			✓

5. Medios que se utilizarán

5.1 Sistema operativo

Se utilizará Windows 11 como sistema principal de desarrollo, por ser el instalado en el equipo personal de la alumna.

5.2 Lenguajes de programación usados

- JavaScript: lenguaje principal para el desarrollo del cliente con React.
- SQL (PostgreSQL): utilizado en Supabase para la definición de tablas, relaciones y políticas de seguridad.
- JSON: formato de intercambio de datos con la API de IA y para almacenar preferencias de usuario.
- HTML5: base de la estructura de las vistas generadas por los componentes React, garantizando accesibilidad y semántica.



- CSS3: hoja de estilos usada para la personalización de los estilos sobre los componentes de Bootstrap.

5.3 Frameworks y librerías

- React: base del frontend, para construir la aplicación como una Single Page Application (SPA).
- Bootstrap 5: framework de estilos para maquetación rápida y responsive.
- Supabase: backend-as-a-service que proporciona base de datos Postgres, autenticación, almacenamiento de ficheros y funciones (Edge Functions).
- supabase-js: librería oficial para conectar React con los servicios de Supabase.
- jsPDF / html2canvas: generación de PDF de la dieta directamente en cliente.
- Cypress: framework de pruebas end-to-end para verificar los flujos completos de la aplicación.

5.4 Entorno de desarrollo

- Visual Studio Code como IDE principal.
- Node.js y npm como entorno de ejecución y gestor de dependencias.
- Git y GitHub para control de versiones, ramas de desarrollo y seguimiento del proyecto.
- Vite como bundler para desarrollo rápido y compilación optimizada de la aplicación.
- Navegadores como Chrome, Edge o Firefox para pruebas de ejecución y depuración.



6. Áreas del ciclo formativo

El proyecto FitMind AI (Front-Only) integra conocimientos y competencias de las diferentes áreas del ciclo formativo de Desarrollo de Aplicaciones Web (DAW):

Bases de datos

Diseño y gestión de un modelo relacional simplificado en PostgreSQL (Supabase), aplicación de consultas SQL y definición de políticas de seguridad mediante Row Level Security (RLS).

Lenguajes de marcas y sistemas de gestión de información

Empleo de HTML5 y CSS3 en combinación con Bootstrap para la maquetación, semántica y diseño responsive. Uso de JSON para intercambio de datos con la API y almacenamiento de preferencias.

Desarrollo web en entorno cliente

Implementación de la interfaz con React y Bootstrap, creación de componentes reutilizables, gestión de rutas (React Router) y generación de PDF en cliente.

Desarrollo web en entorno servidor

Aunque no se desarrolla un backend propio, se trabaja con Supabase como backend-as-a-service: autenticación, base de datos, almacenamiento y funciones. Se incluye la integración de una API externa de IA para la generación de planes.

Despliegue de aplicaciones web

Uso de Vite para compilación y optimización, despliegue en plataformas de hosting estático (Vercel) y gestión de variables de entorno para proteger claves de la API.

Diseño de interfaces web

Aplicación de principios de usabilidad y experiencia de usuario (UX), diseño responsive con Bootstrap y personalización visual mediante CSS.