**4. Codificación**

**a. Entornos**

* **Servidor Web**: Apache/XAMPP
* **Base de Datos**: MySQL/MariaDB 10.4.28+
* **Versión PHP**: 8.2.4+
* **Sistema Operativo**: Windows 10 (entorno de desarrollo)

**b. Lenguajes de Programación**

* **Backend**: PHP (con arquitectura MVC)
* **Frontend**:
* HTML5
* Tailwind CSS
* JavaScript
* **Base de Datos**: SQL (MySQL/MariaDB)

**c. Aspectos relevantes en la implementación**

1. **Arquitectura MVC**:

* Separación clara de responsabilidades en controladores, modelos y vistas
* Estructura organizada en directorios específicos para cada capa

1. **Seguridad**:

* Implementación de sistema de autenticación seguro
* Gestión de sesiones nativas de PHP
* Middleware de autenticación

1. **Integración de IA**:

* Implementación de chatbot utilizando la API Google Gemini
* Integración de asistentes inteligentes para la atención al usuario

1. **Gestión de Datos**:

* Sistema de seguimiento de progreso de usuarios
* Base de datos nutricional
* Gestión de rutinas y clases

1. **Interfaz de Usuario**:

* Diseño responsive con Tailwind CSS
* Sistema de navegación intuitivo
* Panel de administración y panel de miembros

1. **Características Técnicas Destacadas**:

* Sistema de enrutamiento personalizado
* Gestión de sesiones robusta
* Integración con servicios externos (API Google Gemini)
* Sistema de gestión de clases y entrenadores
* Seguimiento de progreso y métricas físicas

Este resumen refleja los aspectos más relevantes de la implementación técnica del proyecto FITPro, manteniendo el enfoque en los elementos más significativos para el desarrollo y funcionamiento del sistema.

**4. Codificación**

**a. Entornos**

El sistema FITPro se desarrolla en un entorno de producción robusto y escalable. La infraestructura se basa en Apache/XAMPP como servidor web principal, proporcionando un entorno estable para el despliegue de aplicaciones PHP. La gestión de datos se realiza mediante MySQL/MariaDB en su versión 10.4.28+, ofreciendo un rendimiento óptimo para operaciones de base de datos complejas. El desarrollo se ejecuta sobre PHP 8.2.4+, aprovechando las últimas características y mejoras de rendimiento del lenguaje. El entorno de desarrollo se configura en Windows 10, proporcionando una plataforma consistente para el desarrollo y pruebas.

**b. Lenguajes de Programación**

La arquitectura del sistema se fundamenta en una combinación estratégica de tecnologías. El backend se desarrolla en PHP, implementando el patrón de diseño MVC (Modelo-Vista-Controlador) para garantizar una separación clara de responsabilidades y un mantenimiento eficiente del código. El frontend se construye utilizando HTML5 como base estructural, complementado con Tailwind CSS para el diseño y estilizado, y JavaScript para la implementación de funcionalidades dinámicas e interactivas. La capa de persistencia de datos se gestiona mediante SQL, específicamente MySQL/MariaDB, permitiendo operaciones eficientes de almacenamiento y recuperación de información.

**c. Aspectos relevantes en la implementación**

1. **Arquitectura MVC**

* Implementación de una estructura modular que separa la lógica de negocio, la presentación y el control de flujo
* Organización jerárquica de directorios que facilita la navegación y mantenimiento del código
* Separación clara de responsabilidades entre controladores, modelos y vistas

1. **Sistema de Seguridad**

* Implementación de un sistema de autenticación multi-capa
* Gestión de sesiones mediante mecanismos nativos de PHP
* Implementación de middleware de autenticación para control de acceso
* Validación y sanitización de datos en todas las capas de la aplicación

1. **Integración de Inteligencia Artificial**

* Implementación de un sistema de chatbot utilizando la API Google Gemini
* Desarrollo de asistentes inteligentes para la atención al usuario
* Procesamiento de lenguaje natural para la interpretación de consultas
* Sistema de aprendizaje continuo para mejorar las respuestas

1. **Gestión de Datos**

* Sistema de seguimiento de progreso de usuarios con métricas personalizadas
* Base de datos nutricional con información detallada de alimentos
* Sistema de gestión de rutinas y clases con programación dinámica
* Implementación de caché para optimizar el rendimiento

1. **Interfaz de Usuario**

* Diseño responsive utilizando Tailwind CSS
* Sistema de navegación intuitivo con breadcrumbs
* Panel de administración con dashboard personalizable
* Panel de miembros con funcionalidades específicas por rol

1. **Características Técnicas Destacadas**

* Sistema de enrutamiento personalizado con manejo de URLs amigables
* Gestión de sesiones robusta con mecanismos de seguridad
* Integración con servicios externos mediante APIs
* Sistema de gestión de clases y entrenadores con programación dinámica
* Implementación de seguimiento de progreso y métricas físicas
* Sistema de notificaciones en tiempo real
* Gestión de recursos multimedia
* Implementación de logs para auditoría y debugging

1. **Optimización y Rendimiento**

* Implementación de caché en múltiples niveles
* Optimización de consultas a base de datos
* Minificación de recursos estáticos
* Lazy loading de componentes
* Compresión de datos en transferencia

1. **Mantenibilidad y Escalabilidad**

* Documentación detallada del código
* Implementación de patrones de diseño
* Estructura modular para facilitar actualizaciones
* Sistema de versionado para control de cambios
* Implementación de pruebas automatizadas

Este documento técnico proporciona una visión detallada de la implementación del sistema FITPro, destacando los aspectos más relevantes de su arquitectura, seguridad, funcionalidades y características técnicas. La documentación sirve como referencia para el mantenimiento, actualización y escalabilidad del sistema, asegurando su continuidad y evolución.

**📌 Sección 4: Codificación**

**a. Entornos (completamente ampliable)**

Debes detallar **herramientas concretas** y su uso en el desarrollo. Aquí algunos puntos clave a desarrollar:

* **IDE utilizado**: por ejemplo, Visual Studio Code o PHPStorm. Mencionar extensiones relevantes (PHP Debug, Live Server, etc.).
* **Servidor local**: ya has indicado XAMPP, pero podrías incluir herramientas como **phpMyAdmin** y su uso en el desarrollo.
* **Gestión de versiones**: ¿Usaste Git o GitHub? ¿Cómo? Por ejemplo, para backups o versionado.
* **Despliegue**: ¿Cómo y dónde se despliega la app? ¿Usas algún servicio de hosting (como DonDominio, cPanel, etc.)?
* **Sistemas adicionales**: ¿Usaste Postman para probar APIs? ¿Alguna herramienta de documentación como Swagger?

**b. Lenguajes de programación (tienes la base, pero falta el enfoque aplicado)**

Incluye no solo los lenguajes usados, sino cómo se aplican específicamente en tu sistema:

* **PHP (MVC)**: ¿cómo estructuras los controladores, modelos y vistas? ¿Sigues alguna convención?
* **JavaScript + Ajax**: ¿qué funciones cumple? ¿Qué mejoras de UX aporta? ¿Hay validación de formularios en tiempo real, actualizaciones asincrónicas, etc.?
* **Tailwind CSS**: justifica por qué lo elegiste (modularidad, velocidad, responsive, etc.).
* **Integración con la API de IA (Google Gemini)**: indica qué lenguaje o librería usas para conectar con esta API, cómo se consume y qué funcionalidades inteligentes provee.
* **SQL**: qué tipo de consultas se realizan: ¿hay procedimientos almacenados? ¿joins complejos? ¿se controla la seguridad?

**c. Aspectos relevantes en la implementación (deberías ampliar en profundidad)**

Aquí deberías estructurarlo por temas clave y justificar cada decisión. Puedes seguir este esquema, alineado con el ejemplo NetLibros:

**4. Codificación**

**a. Entornos**

El sistema **FITPro** se desarrolla en un entorno sólido y pensado para. Está montado sobre una infraestructura que combina estabilidad y flexibilidad: utilizamos **Apache/XAMPP** como servidor web principal, lo que nos da una base confiable para desplegar nuestras aplicaciones PHP.

La gestión de datos esta desarrollada en **MySQL** (versión 10.4.28), la cual nos permite realizar operaciones complejas de base de datos. Ya teníamos experiencia con ella, y eso nos marcó una gran diferencia. Cada vez que surgía una nueva idea/problema o había que cambiar algo de fondo, el hecho de conocerla de antes nos dio agilidad y seguridad para avanzar sin mas complicaciones. Para el desarrollo, utilizamos **PHP 8.2.4+**, una versión moderna del lenguaje que no solo mejora la velocidad, sino que también nos abre la puerta a nuevas posibilidades técnicas.

Todo está desarrollado en **Windows 10**, lo que nos garantiza una experiencia consistente durante el proceso de creación, pruebas y ajustes.

Lo que realmente hace especial al desarrollo de FITPro es su **modularidad, basada en el Modelo Vista Controlador (MVC)**. Nos permite mantener todo bien ordenado: los controladores se encargan del flujo de la aplicación, los modelos manejan la lógica de negocio y las vistas le dan vida a la interfaz.

La seguridad es una prioridad. Implementamos un sistema de **autenticación por capas**, gestionamos sesiones con mecanismos nativos de PHP, y usamos middleware para asegurarnos de que cada usuario solo accede a lo que le corresponde. También ponemos especial atención en la **validación y sanitización de datos**, sin importar si vienen del cliente, del servidor o de cualquier punto intermedio.

Además, algo que nos emociona particularmente es la **integración de inteligencia artificial**. A través de la **API de Google Gemini**, hemos implementado un **chatbot inteligente** que no solo responde preguntas, sino que también aprende con el tiempo. Este asistente virtual puede interpretar lenguaje natural, entender las consultas de los usuarios y mejorar sus respuestas día a día. Es como tener un entrenador digital personalizado con cada conversación.

**b. Lenguajes de Programación**

La arquitectura de FITPro está basada en una combinación estratégica de tecnologías, elegidas no solo por su potencia, sino por cómo funcionan entre sí.

En el **backend**, trabajamos con **PHP** usando el ya mencionado patrón **MVC**, lo que nos da una estructura limpia y manejable, casi como construir con bloques que encajan a la perfección. En el **frontend**, usamos **HTML5**, estilizando con el framework open-source **Tailwind CSS** para lograr un diseño moderno y adaptándose a cualquier dispositivo necesario, y por último, con **JavaScript**, que se encarga de las interacciones y movimientos dinámico vía **AJAX** que hacen que la app se sienta fluida y viva.

La **capa de acceso a los datos** está gestionada mediante **SQL**, concretamente MySQL, permitiendo una gestión eficiente y ordenada de toda la información.

Hemos diseñado un **sistema de navegación intuitivo**, que ayuda al usuario a no perderse, y paneles personalizados. Es una experiencia pensada para sentirse cómoda, rápida y humana.

Y es que la mantenibilidad es clave en este tipo de proyectos. Por eso nos aseguramos de dejar una **documentación detallada**, aplicar **patrones de diseño** cuando corresponde, y organizar el código de forma **modular**, para que las futuras no sean difíciles de aplicar. También contamos con un sistema de **versionado** y, por supuesto, pruebas automatizadas para detectar cualquier problema antes de que llegue a los usuarios.  
  
Hemos utilizado **GitHub como plataforma central para gestionar el código**, y la verdad es que ha sido determinador. Nos ha permitido **controlar versiones de forma precisa**, colaborar sin pisarnos los pies, y mantener un historial limpio de cada avance, prueba o corrección. Ese nivel de trazabilidad nos ha dado mucha tranquilidad durante todo el desarrollo llegando incluso a salvarnos gran parte de una funcionalidad gracias a su historial de backups.