

**UNIVERSIDAD AUTÓNOMA DE SINALOA**  
**FACULTAD DE INFORMÁTICA**

**PROYECTO:**

**FITMATCH**

**MAESTRO:**

° FIDEL CASTRO BOJORQUEZ

**INTEGRANTES:**

Alan Tristan Briseño Escobar

Paul Vazquez Larreta

Adán Alfonso Saucedo Espinoza

Justin Ismael Alcantara Angulo

José Manuel Solar Rodríguez

**GRUPO:**

2-2

**Proyecto de Sistema: FITMATCH**

**Link a repositorio:** <https://github.com/Alantaural/FITMATCH>

# Sistema Web FitMatch: Gestión de Rutinas de Ejercicio Personalizadas

## 1. Introducción

Bienvenido al sistema web **FitMatch**. Esta aplicación está diseñada para ayudar a los usuarios a crear, gestionar y visualizar sus rutinas de ejercicio personalizadas de una manera sencilla e intuitiva. Ideal para entusiastas del fitness, entrenadores personales o cualquier persona que busque organizar su entrenamiento.

## 2. Resumen del Sistema

FitMatch es una aplicación web que permite a los usuarios autenticarse para acceder a funcionalidades de gestión de rutinas. El sistema proporciona una interfaz amigable para:

- **Crear nuevas rutinas:** Definir el nombre de la rutina y añadir ejercicios con sus respectivas series y repeticiones.
- **Visualizar rutinas existentes:** Acceder y revisar las rutinas guardadas.
- **Gestión de usuarios:** Registro e inicio de sesión para asegurar la privacidad y personalización de las rutinas.

El objetivo principal es ofrecer una herramienta eficiente para que los usuarios mantengan un registro estructurado y accesible de sus planes de entrenamiento.

## 3. Requisitos

### a. Requisitos Funcionales y No Funcionales

#### Requisitos Funcionales:

- El sistema debe permitir a los usuarios registrarse con un nombre de usuario y contraseña.
- El sistema debe permitir a los usuarios iniciar y cerrar sesión.
- El sistema debe permitir a los usuarios crear nuevas rutinas de ejercicio, especificando un nombre para la rutina y su descripción.
- El sistema debe permitir a los usuarios añadir múltiples ejercicios a una rutina, incluyendo el ejercicio (selección de una lista predefinida), número de series, número de repeticiones, factor de intensidad y series de calentamiento.
- El sistema debe permitir a los usuarios guardar las rutinas creadas en la base de

datos.

- El sistema debe permitir a los usuarios ver una lista de todas sus rutinas guardadas, asociadas a su usuario.
- El sistema debe permitir a los usuarios ver los detalles de una rutina específica (ejercicios que la componen, series, repeticiones, etc.).
- El sistema debe permitir a los usuarios eliminar rutinas existentes.
- El sistema debe validar los datos de entrada en los formularios (ej. campos obligatorios, formato de email, hash de contraseña).

### Requisitos No Funcionales:

- **Usabilidad:** La interfaz de usuario debe ser intuitiva y fácil de usar, con un diseño limpio y moderno.
- **Rendimiento:** El sistema debe responder de manera eficiente a las interacciones del usuario, con tiempos de carga y respuesta mínimos.
- **Seguridad:** Las credenciales de usuario (contraseñas) deben manejarse de forma segura mediante hash. La comunicación con la base de datos debe prevenir inyecciones SQL.
- **Fiabilidad:** El sistema debe ser estable y minimizar caídas o errores, proporcionando mensajes claros al usuario en caso de fallos.
- **Compatibilidad:** La aplicación debe ser compatible con los navegadores web modernos (Chrome, Firefox, Edge, Safari).

### b. Requisitos Técnicos

- **Servidor Web:** Apache HTTP Server (suministrado por XAMPP).
- **Lenguaje de Programación Backend:** PHP.
- **Base de Datos:** MySQL (suministrado por XAMPP).
- **Lenguajes Frontend:** HTML, CSS, JavaScript.
- **Entorno de Desarrollo Local:** XAMPP (Apache, MySQL, PHP, phpMyAdmin).
- **Gestión de Contraseñas:** Funciones nativas de PHP (password\_hash() y password\_verify()).
- **Comunicación Cliente-Servidor:** AJAX (usando la API fetch de JavaScript).

### c. Requisitos de Arquitectura del Sistema

El sistema FitMatch sigue una arquitectura **Cliente-Servidor** con un enfoque **Monolítico**, donde los componentes de la interfaz de usuario, la lógica de negocio y la capa de acceso a datos residen en el mismo servidor.

- **Capa de Presentación (Frontend):**
  - **HTML:** Proporciona la estructura y el contenido de las páginas web.
  - **CSS:** Define el estilo visual y la apariencia de la interfaz de usuario, siguiendo

un tema oscuro y minimalista.

- **JavaScript:** Maneja la interactividad del lado del cliente, las validaciones de formularios y las comunicaciones asíncronas (AJAX) con el backend.
- **Capa de Lógica de Negocio (Backend):**
  - Implementada en **PHP (api.php)**, es el punto de entrada para todas las solicitudes del cliente.
  - Se encarga de procesar las peticiones (registro, login, creación/eliminación de rutinas), interactuar con la base de datos, aplicar la lógica de la aplicación y devolver respuestas en formato JSON al frontend.
- **Capa de Datos:**
  - Gestionada por **MySQL**, donde se almacenan persistentemente todos los datos de la aplicación.
  - La interacción se realiza a través de PHP utilizando la extensión mysqli para consultas seguras (sentencias preparadas).

La comunicación entre el frontend y el backend se realiza a través de peticiones HTTP/HTTPS (GET, POST, DELETE) que envían y reciben datos en formato JSON.

## 2. Instalación (Configuración del Entorno de Desarrollo)

Para poner en marcha FitMatch en tu entorno de desarrollo local, sigue estos pasos:

1. **Descarga e Instala XAMPP:**
  - Obtén la versión más reciente compatible con tu sistema operativo desde <https://www.apachefriends.org/es/download.html>.
  - Sigue el asistente de instalación, preferiblemente instalando en la ruta por defecto (ej. C:\xampp). Asegúrate de seleccionar los componentes **Apache**, **MySQL** y **PHP**.
2. **Inicia Apache y MySQL:**
  - Abre el **XAMPP Control Panel**.
  - Haz clic en "Start" junto a "Apache" y "MySQL". Asegúrate de que ambos servicios se muestren en color verde. Si Apache no inicia, verifica si el puerto 80 está en uso por otra aplicación (ej. Skype) y considera cambiar el puerto de Apache a 8080 en su archivo de configuración (httpd.conf).
3. **Clona o Descarga el Proyecto:**
  - Si el proyecto está en un repositorio Git, clónalo: `git clone https://github.com/Alantaural/FITMATCH`
  - Si tienes el proyecto como un archivo ZIP, descárgalo y descomprímelo.
4. **Coloca la Carpeta del Proyecto:**
  - Mueve la carpeta principal de tu proyecto (que debe llamarse `fitmatch_app`) al directorio `htdocs` dentro de tu instalación de XAMPP (ej. C:\xampp\htdocs\).

- La estructura de tu proyecto debe ser similar a:

```
C:\xampp\htdocs\  
└─ fitmatch_app\  
    │  
    └─ css\  
        └─ style.css  
    │  
    └─ js\  
        └─ script.js  
    └─ api.php  
    └─ db_connection.php  
    └─ index.html
```

### 5. Configura la Base de Datos:

- Abre tu navegador y ve a <http://localhost/phpmyadmin/>.
- En el panel izquierdo, haz clic en "New" o "Nueva" y crea una nueva base de datos con el nombre **fitmatch**.
- Con la base de datos fitmatch seleccionada, ve a la pestaña "SQL".
- Copia y pega el contenido de tu archivo SQL con la estructura de las tablas (ejercicios, rutinas, rutina\_ejercicios, usuarios) y los datos iniciales. Ejecuta la consulta.
- **Importante:** Asegúrate de que la columna Contra en la tabla usuarios sea VARCHAR(255) para almacenar contraseñas hasheadas.

### 6. Configura la Conexión a la Base de Datos en PHP:

- Abre el archivo C:\xampp\htdocs\fitmatch\_app\db\_connection.php.
- Verifica que la variable \$dbname esté configurada como "fitmatch" y que \$username y \$password correspondan a la configuración por defecto de XAMPP (root y "" - cadena vacía, respectivamente).

### 7. Accede a la Aplicación:

- Abre tu navegador.
- Navega a la URL de tu proyecto: [http://localhost/fitmatch\\_app/](http://localhost/fitmatch_app/) (o [http://localhost:8080/fitmatch\\_app/](http://localhost:8080/fitmatch_app/) si configuraste Apache en el puerto 8080).

## 3. Uso del Sistema

### Flujo de Usuario Básico:

1. **Acceso Inicial:** Al acceder a la aplicación, si no hay una sesión activa, se presentará la interfaz de "Iniciar Sesión" y "Registrarse".
2. **Registro de Nuevo Usuario:**
  - Completa el formulario "Registrarse" con un nombre, correo, contraseña,

descripción (opcional) e imagen de perfil (opcional).

- Haz clic en "Registrar". El sistema almacenará tus datos de forma segura.

### 3. Inicio de Sesión:

- Ingresa tu correo y contraseña en el formulario "Iniciar Sesión".
- Haz clic en "Iniciar Sesión". Si las credenciales son correctas, el sistema te redirigirá a la interfaz principal de la aplicación, mostrando tu nombre de usuario y las secciones de gestión de rutinas.
- Si ya iniciaste sesión previamente, la aplicación te llevará directamente a la interfaz principal al recargar, gracias al almacenamiento local de la sesión.

### 4. Creación de Rutinas:

- Dentro de la sección "Crear Nueva Rutina", ingresa un nombre y descripción para tu rutina.
- En la sección "Añadir Ejercicios a la Rutina", selecciona un ejercicio de la lista desplegable, introduce las series, repeticiones, factor de intensidad y series de calentamiento. Haz clic en "Añadir a la rutina" para agregarlo a la lista temporal de la rutina actual. Puedes añadir múltiples ejercicios.
- Una vez que todos los ejercicios deseados han sido añadidos, haz clic en "Guardar Rutina" para almacenarla en la base de datos.

### 5. Visualización de Mis Rutinas:

- Navega a la sección "Mis Rutinas". Aquí verás una lista de todas las rutinas que has creado.
- Haz clic en el botón "Ver" junto a cualquier rutina para acceder a sus detalles, mostrando la lista de ejercicios que la componen con sus parámetros específicos.
- Haz clic en el botón "Eliminar" para borrar una rutina completa. El sistema solicitará confirmación.

### 6. Cerrar Sesión:

- En la parte superior de la interfaz, junto a tu nombre de usuario, encontrarás el botón "Cerrar Sesión".
- Haz clic en este botón para finalizar tu sesión activa, borrar los datos de sesión del navegador y regresar a la pantalla de inicio de sesión.

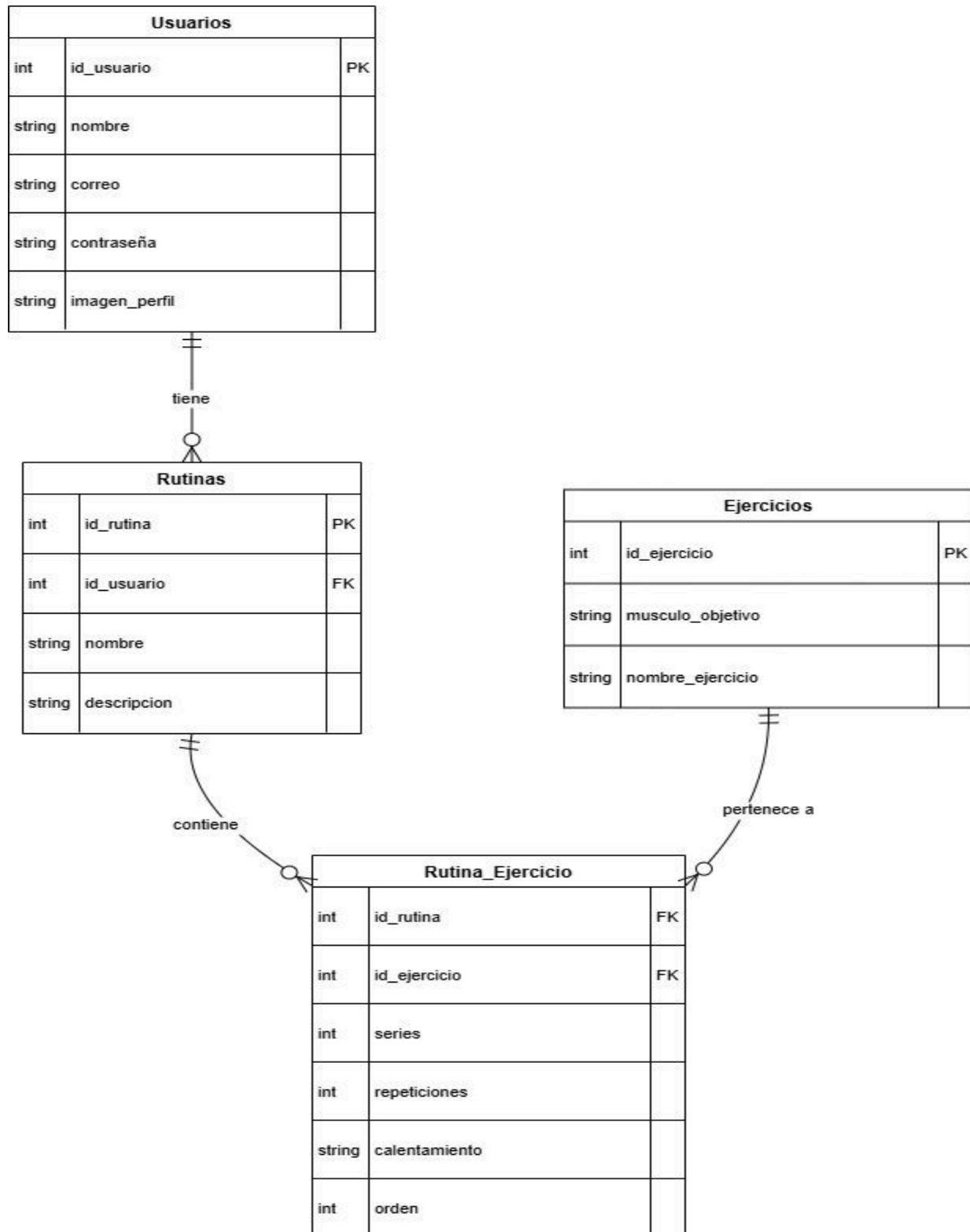
## 4. Base de Datos (Modelado)

El sistema FitMatch utiliza una base de datos **fitmatch** en MySQL, organizada en cuatro tablas principales relacionadas entre sí:

- **ejercicios:** Almacena una lista maestra y predefinida de ejercicios físicos.
  - Id\_Ejercicio (INT, PRIMARY KEY, AUTO\_INCREMENT): Identificador único del ejercicio.

- Musculo\_Objetivo (VARCHAR(50)): Músculo principal al que se dirige el ejercicio (ej. 'Pecho', 'Espalda', 'Hombro').
- Nombre\_Ejercicio (VARCHAR(50)): Nombre descriptivo del ejercicio (ej. 'Press Banca', 'Dominadas').
- **rutinas:** Contiene la información general de las rutinas creadas por los usuarios.
  - Id\_Rutina (INT, PRIMARY KEY, AUTO\_INCREMENT): Identificador único de la rutina.
  - Id\_Usuario (INT, FOREIGN KEY): Identificador del usuario que creó la rutina (referencia a usuarios.Id\_Usuario).
  - Nombre (VARCHAR(50)): Nombre asignado por el usuario a la rutina (ej. 'Rutina de Fuerza PPL').
  - Descripcion\_Rutina (VARCHAR(200)): Breve descripción de la rutina.
- **rutina\_ejercicios:** Esta es una tabla intermedia que establece una relación de "muchos a muchos" entre rutinas y ejercicios. Define qué ejercicios pertenecen a una rutina específica y sus detalles únicos para esa rutina.
  - Id\_Rutina (INT, FOREIGN KEY): Referencia a la rutina a la que pertenece el ejercicio.
  - Id\_Ejercicio (INT, FOREIGN KEY): Referencia al ejercicio específico.
  - Series (INT): Número de series para este ejercicio en esta rutina.
  - Repeticiones (INT): Número de repeticiones para este ejercicio en esta rutina.
  - Factor\_Intensidad (VARCHAR(10)): Factor de intensidad (ej. 'Rir 1', '%RM').
  - Calentamiento (INT): Series de calentamiento para este ejercicio.
  - Orden (INT): El orden en que este ejercicio aparece en la rutina.
  - **Clave Primaria Compuesta:** (Id\_Rutina, Id\_Ejercicio)
- **usuarios:** Almacena la información de registro de cada usuario.
  - Id\_Usuario (INT, PRIMARY KEY, AUTO\_INCREMENT): Identificador único del usuario.
  - Nombre (VARCHAR(50)): Nombre de usuario.
  - Correo (VARCHAR(50), UNIQUE): Correo electrónico del usuario (utilizado para iniciar sesión y debe ser único).
  - Contra (VARCHAR(255)): Contraseña del usuario, almacenada como un hash seguro (NO texto plano).
  - Descripcion (VARCHAR(200)): Descripción personal del usuario.
  - Imagen\_Perfil (VARCHAR(50)): URL o nombre de archivo de la imagen de perfil del usuario.

## Diagrama Entidad-Relación (Conceptual):





## Relaciones:

- Un usuario puede crear muchas rutinas (1:N de usuarios a rutinas).
- Una rutina puede contener muchos ejercicios, y un ejercicio puede estar en muchas rutinas (N:M entre rutinas y ejercicios, resuelta por rutina\_ejercicios).

*(Nota: Este es un diagrama ASCII conceptual. Para un diagrama visual completo, se recomienda usar una herramienta de modelado de bases de datos.)*

Archivo SQL de la Base de Datos:

El script SQL completo para la creación de la base de datos y sus tablas, incluyendo los datos iniciales de ejercicios, se encuentra en:

[ruta\_relativa\_a\_tu\_archivo\_sql\_si\_lo\_exportas\_ej. sql/fitmatch\_db.sql]

## 5. Mantenimiento y Actualizaciones

- **Copias de Seguridad:** Es fundamental realizar copias de seguridad periódicas de la base de datos (fitmatch) para prevenir la pérdida de datos.
- **Actualizaciones del Entorno:** Mantener XAMPP (Apache, PHP, MySQL) y el sistema operativo actualizados a las últimas versiones estables para beneficiarse de mejoras de seguridad, rendimiento y compatibilidad.
- **Gestión de Contenido:** Actualizar la lista de ejercicios en la tabla ejercicios o añadir nuevas características a las rutinas según las necesidades de los usuarios.
- **Optimización del Código:** Realizar refactorización del código PHP y JavaScript para mejorar la legibilidad, la eficiencia y la modularidad a medida que la aplicación crece.
- **Monitoreo de Errores:** Implementar un sistema de registro de errores (logging) en el backend de PHP para facilitar la depuración y el mantenimiento.

## 6. Pruebas

Para asegurar la calidad y funcionalidad del sistema, se recomienda realizar los siguientes tipos de pruebas:

- **Pruebas Unitarias:**
  - Validación de datos de entrada en formularios (cliente y servidor).
  - Funciones de hash y verificación de contraseñas.
  - Funciones auxiliares en JavaScript y PHP.
- **Pruebas de Integración:**
  - **Flujo de Registro:** Confirmar que un nuevo usuario puede registrarse y sus datos se guardan correctamente en la base de datos.
  - **Flujo de Inicio de Sesión:** Verificar que un usuario puede iniciar sesión con credenciales correctas e incorrectas.

- **Creación de Rutina:** Asegurar que se puede crear una rutina con múltiples ejercicios y que se guarda correctamente en rutinas y rutina\_ejercicios.
- **Visualización y Eliminación de Rutinas:** Confirmar que las rutinas se listan correctamente y se pueden ver sus detalles, así como eliminarlas.
- **Comunicación Cliente-Servidor:** Asegurar que las peticiones AJAX se envían y reciben correctamente y las respuestas JSON se procesan.
- **Pruebas de Usabilidad (UX):**
  - Evaluar la facilidad de navegación y la claridad de la interfaz de usuario.
  - Recopilar feedback de usuarios para identificar puntos de mejora en la experiencia.
- **Pruebas de Seguridad:**
  - Intentar inyecciones SQL en los puntos de entrada de datos para verificar la protección de las sentencias preparadas.
  - Probar la robustez del manejo de sesiones y la protección contra accesos no autorizados.
- **Pruebas de Responsividad:**
  - Verificar que la interfaz se adapta y es funcional en diferentes tamaños de pantalla (escritorio, tablet, móvil) y orientaciones.
- **Pruebas de Rendimiento (Carga):** (Opcional) Simular múltiples usuarios realizando operaciones para evaluar la capacidad de respuesta del servidor bajo carga.

**Autores:**

- Alcántara Angulo Justin Ismael
- Briseño Escobar Alan Tristan
- Saucedo Espinoza Adán Alfonso
- Solar Rodríguez José Manuel
- Vazquez Larreta Paul Alberto.

**Versión:** 1.0.0

**Fecha:** [Fecha: 8 de Junio de 2025]