**Manual del Desarrollador del Sitio Web WorldFit**

**Introducción**

Este manual técnico tiene como objetivo proporcionar una guía detallada sobre la estructura, configuración, y funcionamiento del sitio web **WorldFit**. En este documento, se describen los componentes clave del sistema, incluyendo el desarrollo frontend y backend, la integración de bases de datos, y las funcionalidades implementadas para ofrecer una experiencia óptima a los usuarios.

**WorldFit** es una plataforma en línea diseñada para ofrecer orientación personalizada en rutinas de ejercicio y brindar información detallada sobre gimnasios. Este manual está dirigido a desarrolladores, administradores de sistemas, y cualquier otra persona encargada de mantener y actualizar la plataforma.

**Tabla de contenido.**

**1. Tecnologías Utilizadas ……………………………………………………………………………………………………………..**

1.1 Lenguajes de Programación

1.2 Base de Datos

1.3 Entorno de Desarrollo

**2. Librerías y Frameworks ……………………………………………………………………………………………………………**

2.1 Front-end

2.2 CSS

2.3 JavaScript

**3. Diagramas UML ………………………………………………….……………………………………………………………………**

3.1 Diagrama de Clases

3.2 Diagrama de Actividades

3.3 Diagrama de MER

**4. Patrones de Diseño ………………………………………………………………………………………………………………….**

4.1 MVC

1. **Tecnologías Utilizadas**

**1.1 Lenguajes de programación.**

Para la realización de este proyecto, se han utilizado diversos lenguajes de programación tanto en el frontend como en el backend:

**Front-end:**

* **HTML:** Utilizado para crear toda la estructura del contenido visual del proyecto, asegurando la compatibilidad con diferentes navegadores.

*Figura 1. autor: https://icons8.com/icons*

* **CSS:** Implementado para el diseño y la personalización de colores y estilos, mejorando la apariencia visual y la estética del proyecto.



*Figura 2. autor: https://icons8.com/icons*

* **JavaScript:** Esta tecnología se utilizó para proporcionar interactividad en el sitio, permitiendo una experiencia de usuario más amigable.



*Figura 3. autor: https://icons8.com/icons*

**Back-end:**

* **PHP:** Este lenguaje de programación se utilizó para toda la lógica implementada en el proyecto y como intermediario entre el servidor y el cliente. Gracias a su facilidad de uso y amplia gama de funcionalidades, PHP resultó ser una opción adecuada para el desarrollo del backend.

*Figura 4. autor: https://icons8.com/icons*

**1.2 Base de datos.**

Para la realización de la base de datos, se optó por el uso de MySQL, una tecnología de gestión de bases de datos relacional. MySQL se eligió debido a su robustez, escalabilidad y facilidad de integración con otros componentes del proyecto. Este sistema de gestión de bases de datos permite almacenar, organizar y recuperar la información de manera eficiente, proporcionando un rendimiento óptimo para el proyecto.

*Versión 8.0.36*

*Figura 5. autor: https://icons8.com/icons*

**1.3 Entorno de Desarrollo y control de Versiones.**

Para el desarrollo del proyecto, se utilizó el entorno de servidor local XAMPP. Gracias a su facilidad de configuración, XAMPP facilitó el manejo de pruebas y la gestión del entorno de desarrollo en un servidor local, permitiendo la ejecución del proyecto de manera eficiente y simplificada.

*Figura 6. autor: https://www.pngwing.com/es/search?q=xampp*

Además, para el control de versiones, se empleó el sistema de control de versiones Git, junto con un repositorio en GitHub. Estas herramientas permitieron un manejo eficaz de las versiones del código y facilitaron la colaboración entre el equipo.



*Versión 2.45.2*

*Figura 7. autor: https://github.com/bryan2811/Curso-de-Git-y-Github-Platzi*

1. **Librerías y Frameworks**

**2.1. Front-end.**

Se utilizó el framework jQuery para facilitar la manipulación del DOM y mejorar la interactividad del sitio web. jQuery simplifica la implementación de funcionalidades dinámicas, permitiendo una integración eficiente y una experiencia de usuario enriquecida.

*Figura 8. autor: https://icon-icons.com/es/icono/jquery-logo/169491*

*Versión 3.6.0*

**2.2. CSS.**

Para el diseño y la estética, se empleó la biblioteca Bootstrap. Esta herramienta proporciona una interfaz responsive que se adapta a diferentes tamaños de pantalla, así como una amplia gama de componentes y estilos que contribuyen a la apariencia profesional y funcional del proyecto.

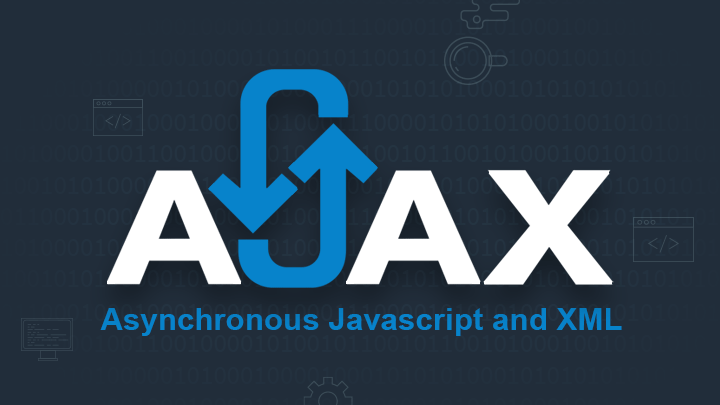


*Versión 5.3*

*Figura 9. autor: https://devcode.la/blog/que-es-bootstrap/*

*Figura 10. autor: https://icons8.com/icons*

**2.3. JavaScript.**

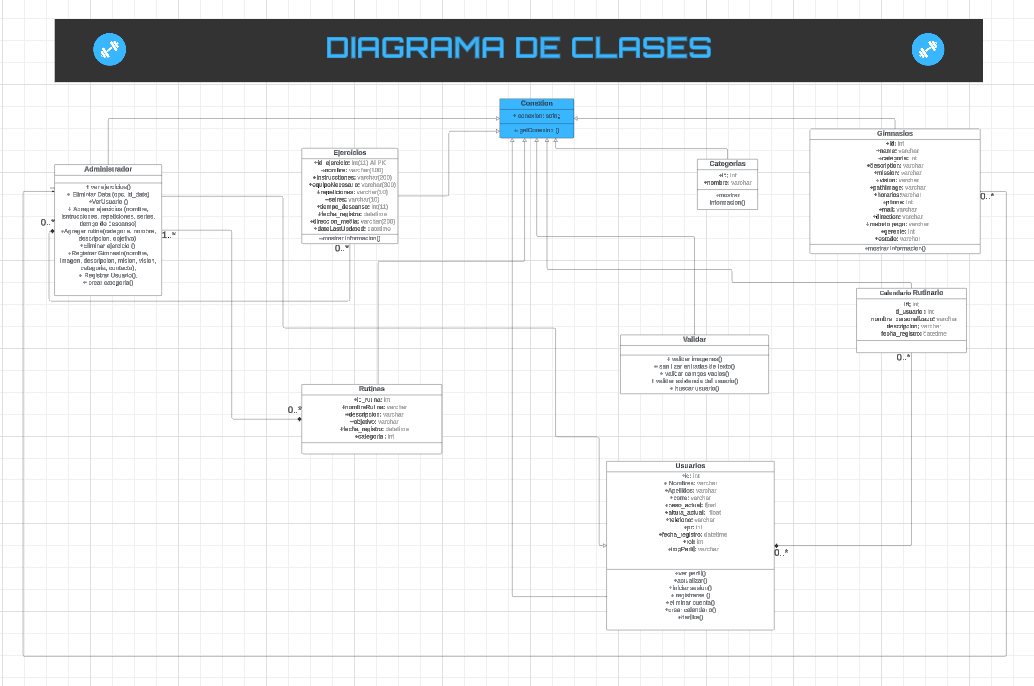
En cuanto a JavaScript, se utilizó AJAX junto con jQuery para realizar solicitudes al servidor de manera asíncrona. Esta combinación permite actualizar el contenido de la página sin necesidad de recargar el navegador, ofreciendo una experiencia de usuario más fluida y eficiente.

*Figura 11. autor: https://velog.io/@tabi4645/AJAX*

1. **Diagramas UML**

**3.1 Diagrama de Clases**

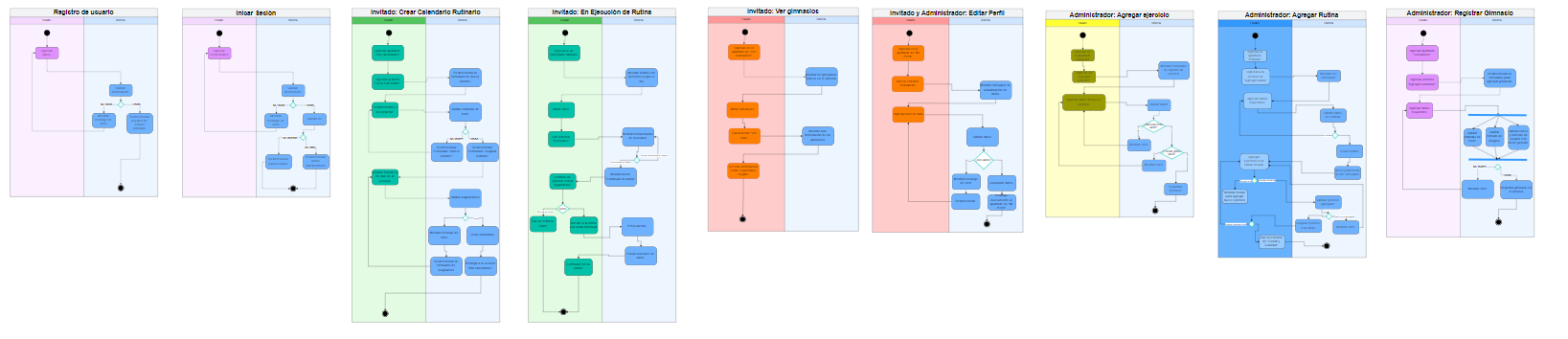
A continuación, se presenta el diagrama de clases que muestra las clases utilizadas en el proyecto. Este diagrama proporciona una visión clara de la estructura del sistema.[**Ver Mejor Aquí**](https://lucid.app/lucidchart/2c459e70-d4c3-48dd-be5d-208d4d31c4d7/edit?viewport_loc=-4612%2C-937%2C12118%2C5687%2CHWEp-vi-RSFO&invitationId=inv_ea0ec6ca-5b8e-42c6-b7e9-af93c7adf127).



*Figura 12. autor: Fernando Alfonso*

**3.2 Diagrama de Actividades**

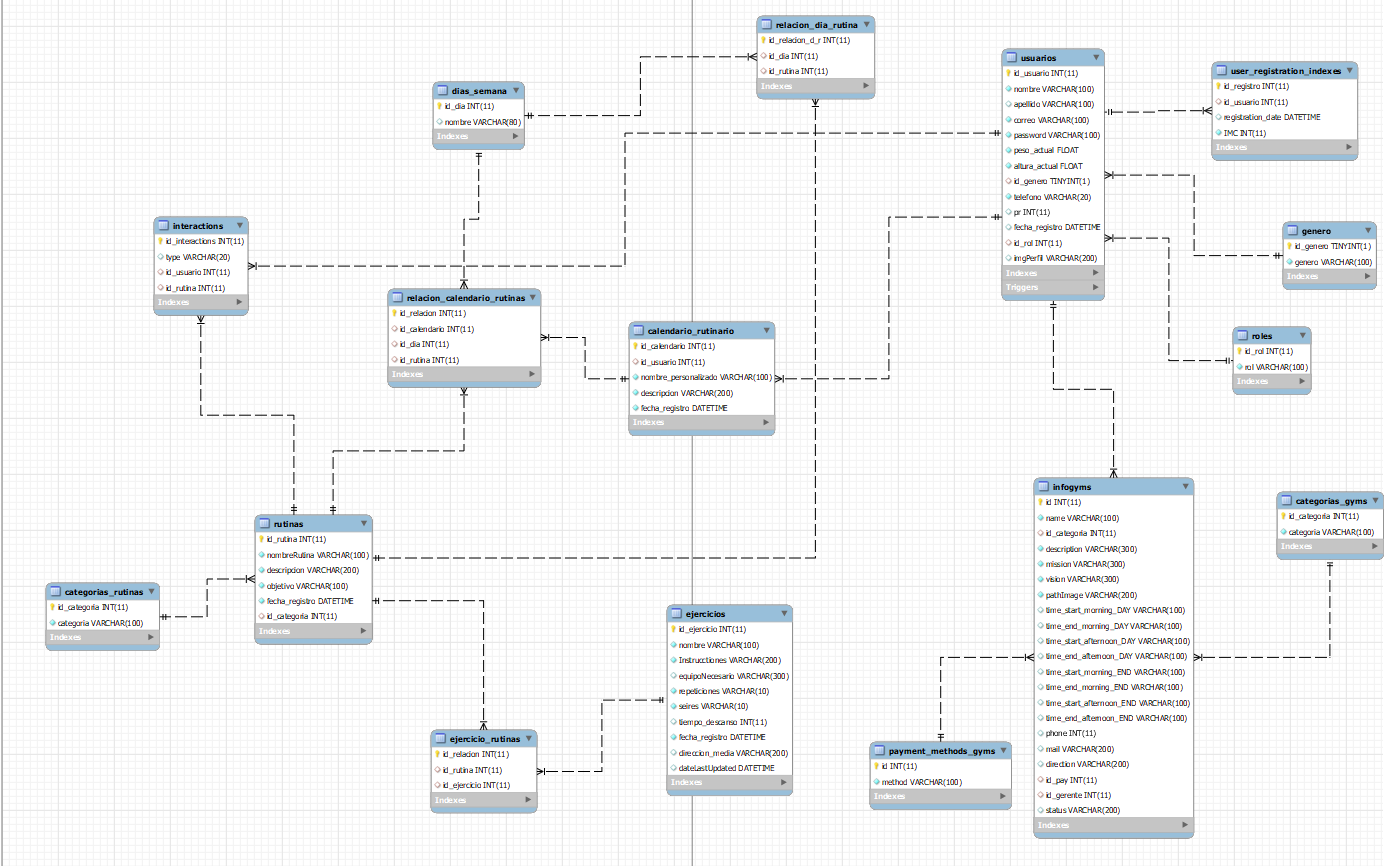
A continuación, se presenta el diagrama de actividades que muestra las actividades utilizadas en el proyecto. Este diagrama proporciona una visión clara de la estructura del sistema. [Ver mejor Aquí](https://drive.google.com/file/d/15KSr-LRW6wgR87t1Fdo9jfuwQjdFyUdL/view?usp=sharing).



*Figura 13. autor: Fernando Alfonso*

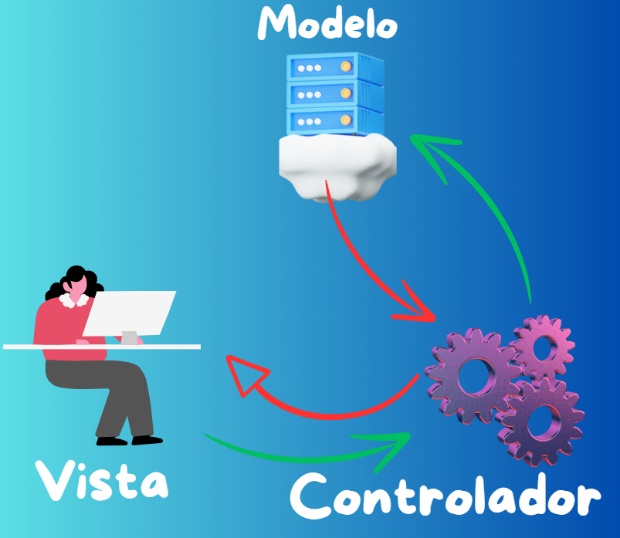
**3.3 Diagrama Entidad relación**

A continuación, se presenta el diagrama MER de la base de datos que muestra la estructura de la base de datos utilizada en el proyecto. Este diagrama proporciona una visión clara de la organización de los datos. [Ver Mejor aquí.](Diagrama%20MER.PNG)



*Figura 14. autor: Fernando Alfonso*

1. **Patrón de Diseño**

El proyecto se basó en el patrón de diseño **Modelo-Vista-Controlador** (MVC) debido a su facilidad de mantenimiento en caso de errores y su estructura clara para el desarrollo. Este patrón divide la aplicación en tres componentes principales: el Modelo, que gestiona los datos; la Vista, que presenta la interfaz de usuario; y el Controlador, que maneja la lógica de aplicación y las interacciones del usuario. Esta separación facilita la gestión del código, la resolución de problemas y el desarrollo eficiente.

*Figura 15. autor: Fernando Alfonso*