

**TEMA:**

“DESARROLLO DE UNA PLATAFORMA DE GESTIÓN CON UN MÓDULO DE NUTRICIÓN PERSONALIZADA PARA EL GIMNASIO FORCA & FITNESS”

**AUTORES:**

ASTO AIQUIPA, Gustavo Blas

ICHPAS CLEMENTE, Luis Angel

FLORES CÓRDOVA, César Augusto

**CURSO**: Herramientas de Desarrollo

**DOCENTE**:

AMACHE SÁNCHEZ, Milton Freddy

**SECCIÓN**: 31652

**TURNO**:

Tarde

**Lima - Perú, 2025**

Dedicatoria

Dedico este trabajo con todo mi cariño y gratitud:

A nuestras familias, por su comprensión, paciencia y apoyo incondicional durante todo este trayecto académico.

A nuestros amigos, por estar siempre presentes para animarnos y motivarnos en los momentos difíciles.

A nuestro docente, por su guía, experiencia y valiosos aportes que nos ayudaron a enriquecer y culminar esta primera etapa del proyecto.

De manera muy especial, dedico mi esfuerzo a mi enamorada, [Nombre de ella], por su apoyo incondicional, comprensión y motivación constante durante la realización de este proyecto. Sin su compañía este camino hubiera sido mucho más difícil.

Agradecimiento

Al momento del desarrollo de este proyecto, muchas personas han sido fundamentales en el proceso de investigación de este trabajo. En primer lugar, nuestras familias cuya comprensión y apoyo emocional fueron importantes durante este trayecto. Nuestros amigos, quienes están presentes para brindarnos ánimos.

Un agradecimiento especial a nuestro docente, quien con su experiencia y orientación ha sido una guía esencial para el desarrollo de este proyecto en su primera etapa, por los valiosos aportes que nos ayudaron a enriquecer este estudio.

Finalmente, quiero agradecer a todas las personas que contribuyeron de alguna forma a la realización de esta tesis, sin su apoyo no hubiera sido posible.

**Agradecimientos personales**

Gustavo Asto: De manera muy especial quiero agradecer a mi enamorada, Joselyn, por su apoyo incondicional, comprensión y motivación durante el desarrollo de este proyecto.

Resumen

En la actualidad, la utilización de plataformas digitales se ha vuelto indispensable para todo tipo de negocios, incluidos los del sector físico-deportivo, por los beneficios que aportan en eficiencia operativa y experiencia del cliente. Casos como Action Black en Colombia y Smart Fit en Brasil demuestran que la integración de tecnología y módulos de nutrición incrementa la competitividad y satisfacción de los usuarios.

El gimnasio Forca & Fitness, con dos sedes en Lima, actualmente presenta ciertas ineficiencias en la centralización de la información, además en la comunicación de horarios y eventos, esto generado por el uso de registros manuales o herramientas poco optimas. Asi mismo, carece de un valor agregado como vendría a ser un modulo de nutrición, el cual complemente sus servicios deportivos. Esto representando un efecto adverso en su operatividad y consumo de recursos, pero que al mismo tiempo representa una oportunidad para modernizar sus procesos y marcar la diferencia frente a sus competidores.

Abstract

Today, the use of digital platforms has become indispensable for all types of businesses, including those in the fitness and sports sector, due to the benefits they bring in operational efficiency and customer experience. Cases such as Action Black in Colombia and Smart Fit in Brazil demonstrate that the integration of technology and nutrition modules increases competitiveness and user satisfaction.

The Forca & Fitness gym, with two locations in Lima, currently experiences certain inefficiencies in centralizing information and communicating schedules and events, due to the use of manual records or suboptimal tools. Furthermore, it lacks added value, such as a nutrition module, which would complement its sports services. This has an adverse impact on its operations and resource consumption, but at the same time represents an opportunity to modernize its processes and differentiate itself from its competitors.

Índice

[Dedicatoria 1](#_Toc211806266)

[Agradecimiento 2](#_Toc211806267)

[Resumen 3](#_Toc211806268)

[Abstract 4](#_Toc211806269)

[Realidad Problemática 7](#_Toc211806270)

[Justificación del proyecto 9](#_Toc211806271)

[Objetivos General y Especifico 11](#_Toc211806272)

[Objetivo general 11](#_Toc211806273)

[Objetivos específicos 11](#_Toc211806274)

[Marco teórico 12](#_Toc211806275)

[Fundamento teórico 12](#_Toc211806276)

[Herramientas y tecnologías de desarrollo 12](#_Toc211806277)

[Metodología 18](#_Toc211806278)

[Enfoque metodológico 18](#_Toc211806279)

[Roles y responsabilidades 19](#_Toc211806280)

[Artefactos de Scrum 21](#_Toc211806281)

[Ciclo de desarrollo (Sprints) 30](#_Toc211806282)

[Modelado del sistema 33](#_Toc211806283)

[Desarrollo del proyecto 42](#_Toc211806284)

[Base de datos 42](#_Toc211806285)

[Bibliografía 47](#_Toc211806286)

**índice de Ilustraciones**

[Ilustración 1 Logo Spring Boot 14](#_Toc211806287)

[Ilustración 2 Logo React JS 14](#_Toc211806288)

[Ilustración 3 Logo GitHub 15](#_Toc211806289)

[Ilustración 4 Logo VS CODE 16](#_Toc211806290)

[Ilustración 5 Logo MariaDB 16](#_Toc211806291)

[Ilustración 6 Logo Git 17](#_Toc211806292)

[Ilustración 7 logo Java 18](#_Toc211806293)

[Ilustración 8 Logo Ms Project 18](#_Toc211806294)

[Ilustración 9 Diagrama de caso de uso – Recepcionista 34](#_Toc211806295)

[Ilustración 10 Diagrama de Casos de Usos - Alumno 35](#_Toc211806296)

[Ilustración 11 Diagrama de Casos de Usos – Alumno 36](#_Toc211806297)

[Ilustración 12 Diagrama de Casos de Usos – Administrador 37](#_Toc211806298)

[Ilustración 13 Diagrama de Casos de Actividad – Proceso de Registro 38](#_Toc211806299)

[Ilustración 14 Diagrama de Casos de Actividad – Proceso de plan Nutricional 39](#_Toc211806300)

[Ilustración 15 Diagrama de Casos de Actividad – Horario 40](#_Toc211806301)

[Ilustración 16 Diagrama de Casos de Actividad – Gestionar una Nueva Clase 41](#_Toc211806302)

[Ilustración 17 Diagrama de Casos de Actividad – Gestionar nuevo trabajador 42](#_Toc211806303)

[Ilustración 18 Modelo físico 47](#_Toc211806304)

**índice de tablas**

[Tabla 1 Product Backlog 23](#_Toc211806305)

[Tabla 2 Nivel de prioridad 26](#_Toc211806306)

[Tabla 3 Estructura del Sprint Backlog 27](#_Toc211806307)

[Tabla 4 Tabla de Incremento 30](#_Toc211806308)

[Tabla 5 Tabla resumen de Sprints 32](#_Toc211806309)

**Indice de la tabla de Contenidos: Memoria Técnica del Proyecto Forca&Fitness**

Realidad Problemática

En los últimos años el empleo de plataformas digitales se ha ido volviendo una necesidad en todo tipo de negocios, ya sean grandes o pequeños, esto debido a los beneficios que esta trae consigo al agilizar diversos procesos internos en la empresa. Además de servir como un atractivo llamativo para los usuarios que se interesen en contratar el servicio o producto que se ofrezca. En este caso tratándose del sector físico-deportivo, como viene a ser un gimnasio.

A nivel internacional se tiene el caso de empresas como Action Black en Colombia, la cual se fundo en el año de 2015, el cual tuvo un crecimiento escalado, el cual se vio potenciado cuando incluyeron la tecnología en el negocio, esto de la mano de Trainingym, una empresa que ayudo en desarrollo de software para el negocio, desde plataformas hasta aplicaciones para los usuarios, esto trayendo beneficios como la eficiencia operativa, aumento satisfacción del cliente. Por otra parte, otro caso es Smart Fit en Brasil, el cual además de invertir en equipamientos en las sedes, también integro plataformas digitales para el control de los clientes (*Smart Fit - Sustentabilidad*, s. f.), también incluye la alimentación en sus programas, brindando una herramienta que emplea la tecnología para informar sobre correctas dietas para sus clientes, mejorando los resultados y la satisfacción de estos.

Es en ese sentido que nos encontramos con el gimnasio Forca & Fitness, la cual en la actualidad posee dos sedes repartidas en Lima, es en ellas que se denotan ciertas ineficiencias operativas, las cuales generan dificultades en el tema de centralización de la información. Esto debido a la falta de una plataforma unificada para el tema de registros en general de trabajadores como alumnos, debido a que estos se realizan a mano o con herramientas que no son las más optimas. Así como para la difusión de información sobre cambios en los horarios o eventos especiales, estos afectando en la coordinación entre las filiales y la experiencia de los estudiantes. Todo esto impactando negativamente en la operatividad del negocio, además de consumir recursos valiosos que podrían destinarse a la enseñanza y el desarrollo de nuevas clases. Asimismo, Forca & Fitness carece de un módulo de nutrición que complemente sus servicios deportivos, lo que representa una oportunidad para ofrecer un valor diferencial frente a otros gimnasios del rubro.

Justificación del proyecto

La justificación del proyecto se da ante la necesidad del gimnasio Forca & Fitness de modernizar los procesos internos. Esto dado a la dependencia actual a los métodos manuales y poco organizados para el registro y comunicación, los cuales generaron ineficiencias, limitando el potencial del gimnasio, y afectando en la experiencia de los usuarios. El desarrollo de una plataforma de gestión centralizado con un módulo de nutrición personalizada traerá beneficios directos para la administración, alumnos y el negocio, de tal manera posicionándola como una institución completa en deporte y salud.

El desarrollo de una plataforma de gestión centralizado con un módulo de nutrición personalizada traerá los siguientes beneficios.

**Administración:** Entre los beneficios se encuentra la centralización de la información entre las dos sedes, de tal manera se evitarían la duplicidad de registros, el riesgo de pérdida de datos. Además, facilitara a los Administradores en la gestión de inscripciones, acceso a reportes generales, los cuales ayudaran en la toma de decisiones.

**Alumnos:** Los beneficios serian inmediatos, tomando en cuenta una mejora en la comunicación entre los horarios, eventos, los cuales serán instantáneos y organizados, de tal manera mejorando la experiencia. Además, al facilitar un área ahora acceder en rutinas e información sobre alimentación, el cual añadir un valor diferencial.

**Para la empresa:** Una plataforma digital proyectará una imagen de modernidad y profesionalismo, aumentando su competitividad en el mercado. La integración del módulo de alimentación permitirá ofrecer planes alimenticios vinculados al progreso físico del estudiante, generando una propuesta de valor única frente a otros gimnasios que solo se centran en la parte deportiva.

En consecuencia, el desarrollo de este proyecto resolvería los problemas administrativos, además de permitir a Forca & Fitness posicionarse como una institución integral en el ámbito del deporte y la salud.

Objetivos General y Especifico

Objetivo general

- Desarrollar una plataforma de gestión centralizado con un módulo de nutrición personalizada

Objetivos específicos

- Detectar los procesos actuales dentro de F orca & Fitness para identificar áreas de mejora.

- Diseñar la arquitectura y los módulos de la plataforma, incluyendo una base de datos centralizada.

- Desarrollar la plataforma con funcionalidades para el registro de nuevos alumnos y la actualización de datos existentes.

- Integrar un módulo para incluir las sugerencias en planes de alimentación personalizados según el progreso del alumno en los entrenamientos.

Marco teórico

Fundamento teórico

En la actualidad, la transformación digital ha llevado a que varios sectores lo implementen con el fin de mejorara su gestión operativa y la experiencia de usuario. La plataforma propuesta se basa en una arquitectura cliente-servidor, en el cual el cliente maneja la interfaz y el servidor gestiona la lógica. Por otra parte, el backend se desarrollará con SpringBoot y Java, mientras que el frontend se desarrollará con React. La información se almacenará en MariaDB, junto con JPA e Hibernate. El desarrollo del proyecto seguirá una metodología ágil, siendo esta Scrum. En conjunto, todas estas tecnologías permitirán desarrollar una plataforma segura i eficiente para la gestión integral del gimnasio Forca & Fitness.

Herramientas y tecnologías de desarrollo

SPRING BOOT

Spring Boot es un framework basado en Java que facilita la creación de aplicaciones web y microservicios. Su principal ventaja es que simplifica la configuración y el desarrollo, permitiendo crear aplicaciones de manera rápida y con un mínimo esfuerzo de configuración. Spring Boot elimina la necesidad de configurar manualmente componentes como servidores de aplicaciones, gestionando automáticamente dependencias y configuraciones predeterminadas.

Ilustración 1 Logo Spring Boot

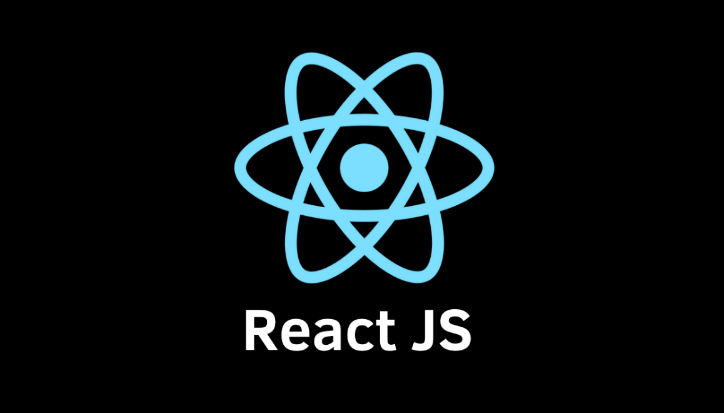


Fuente: Spring Boot

REACT

React.js es una biblioteca de JavaScript enfocada en el desarrollo de interfaces de usuario dinámicas y eficientes, especialmente útil en aplicaciones web modernas. Gracias a su estructura basada en componentes reutilizables, permite construir una interfaz clara, ordenada y fácil de mantener, lo cual es clave en sistemas que manejan gran cantidad de datos como perfiles.

Ilustración 2 Logo React JS



Fuente: React JS

GITHUB

GitHub es una plataforma de desarrollo colaborativo que permite a los usuarios subir, almacenar y gestionar proyectos con código fuente. Los desarrolladores pueden alojar sus proyectos en repositorios, donde se guarda el código y su historial de cambios, ya sean públicos o privados. Además, GitHub emplea un sistema de control de versiones que facilita el seguimiento de modificaciones y permite la colaboración simultánea entre varios desarrolladores sin conflictos.

Ilustración 3 Logo GitHub



Fuente: GitHub

VS CODE

Visual Studio Code (VS Code) es un editor de código fuente desarrollado por Microsoft, diseñado para ser ligero, rápido y altamente personalizable. A diferencia de otros editores de código, como los IDE completos (Entornos de Desarrollo Integrados), VS Code se enfoca en ofrecer una experiencia ágil para la escritura y edición de código, sin sacrificar características avanzadas. Lo que lo diferencia es su capacidad de ser extendido con una gran cantidad de extensiones y herramientas, adaptándose a las necesidades específicas de cada desarrollador.

Ilustración 4 Logo VS CODE



Fuente: VS CODE

MariaDB

MariaDB Server es un sistema de gestión de bases de datos relacionales de código abierto. Es uno de los servidores de bases de datos más populares del mundo, con usuarios notables como Wikipedia, WordPress.com y Google. MariaDB Server se publica bajo la licencia de código abierto GPLv2 y se garantiza entonces que seguirá siendo abierto.

Se puede utilizar para datos de transacciones de alta disponibilidad, análisis de datos, como servidor integrado, y una amplia gama de herramientas y aplicaciones soportan MariaDB Server.

Ilustración 5 Logo MariaDB



Fuente: MariaDB

GIT

Git es un sistema de control de versiones distribuido, lo que significa que un clon local del proyecto es un repositorio de control de versiones completo. Estos repositorios locales plenamente funcionales permiten trabajar sin conexión o de forma remota con facilidad. Los desarrolladores confirman su trabajo localmente y, a continuación, sincronizan la copia del repositorio con la del servidor. Este paradigma es distinto del control de versiones centralizado, donde los clientes deben sincronizar el código con un servidor antes de crear nuevas versiones.

Ilustración 6 Logo Git



Fuente: Git

JAVA WEB

Java es un lenguaje de programación ampliamente utilizado para codificar aplicaciones web. Ha sido una opción popular entre los desarrolladores durante más de dos décadas, con millones de aplicaciones Java en uso en la actualidad. Java es un lenguaje multiplataforma, orientado a objetos y centrado en la red que se puede utilizar como una plataforma en sí mismo. Es un lenguaje de programación rápido, seguro y confiable para codificarlo todo, desde aplicaciones móviles y software empresarial hasta aplicaciones de macrodatos y tecnologías del servidor.

Ilustración 7 logo Java



Fuente: Java

MICROSOFT PROJECT

Microsoft Project es un software diseñado para planificar, organizar y gestionar proyectos y portafolios. Esta herramienta puede usarse tanto en la nube (Software as a Service) como instalada localmente (On Premises). Su principal objetivo es facilitar la administración de proyectos mediante funcionalidades como la programación de tareas, la asignación de recursos y la generación de reportes.

Ilustración 8 Logo Ms Project



Fuente: Microsoft

Metodología

La metodología se define como el conjunto de estrategias y procedimientos orientados al desarrollo del proyecto, de tal manera garantizando orden, eficiencia y cumplimiento de los objetivos. Para este trabajo se opto por un enfoque ágil, específicamente el marco de trabajo Scrum, dado a su capacidad de adaptarse a los cambios, además de fomentar la colaboración constante entre los integrantes del equipo y el cliente.

Enfoque metodológico

El proyecto empleara una metodología ágil, siendo esta Scrum porque durante el desarrollo de software permite trabajar de manera iterativa, y con una mejora continua. Asi mismo, se elegio este enfoque por su capacidad de adaptabilidad a los requerimientos de los clientes, además de permitir una comunicación constante entre los miembros del equipo, garantizando un producto final que cumpla con las necesidades del gimnasio Forca & Fitness.

Según Pressman y Maxim (2020), las metodologías agiles tienen como prioridad la entrega temprana y continua de componentes funcionales del software, de tal manera fomentando la interacción con el cliente, minimizando los riesgos de incumplimiento de objetivos. De tal manera, diferenciándose de modelos tradicionales como en cascada, el cual necesita finalizar una etapa antes de avanzar a la siguiente. En tal sentido, Scrum permite trabajar mediante Sprints, en los cuales se planifica, desarrolla, prueba y evalúan los avances.

Este enfoque ágil es el más considerado para este proyecto debido a su naturaleza, el cual es modular. La plataforma contempla el desarrollo en módulos (gestión de usuarios, administración, nutrición, comunicación), por lo que un desarrollo iterativo facilita el desarrollo independiente de cada uno. Asi mismo, Scrum promueve un trabajo colaborativo en el que se definen los roles específicos, los cuales son el Product Owner, Scrum Master y el Equipo de Desarrollo, permitiendo mejorar la coordinación de tareas y control de progreso. Para el desarrollo de este proyecto cada Sprint tomará un tiempo aproximado de dos semanas por Sprint, siendo seis en total, además de que se realizaran reuniones de seguimiento al iniciar cada Sprint, que abarca desde le inicio al final del proyecto.

Roles y responsabilidades

En el marco de desarrollo de la metodología ágil Scrum, se realizó la organización de los roles para facilitar la colaboración, comunicación y la entrega de avances, según Schwaber y Sutherland (2020), Scrum establece tres roles esenciales: Product Owner, Scrum Master y Equipo de desarrollo, estos se adaptan según las necesidades a cumplir en el proyecto,

1. **Product Owner (Propietario del Producto)**

Responsable: jefe de Proyecto

Encargado de analizar los requisitos del producto, buscando el cumplimiento de los requerimientos de la manera óptima. Asi mismo, supervisar los objetivos del negocio y las entregas técnicas, como de validar los resultados al final de cada sprint.

1. **Scrum Master**

Responsable: Analista Funcional

Actúa como facilitador del proceso Scrum, su función principal es eliminar impedimentos, garantizar que el equipo siga los principios ágiles y promover la mejora continua. Para ello, coordina las reuniones de seguimiento, revisiones y retrospectivas de Sprint. Además, vela por el cumplimiento del cronograma (según el diagrama de Gantt) y la comunicación fluida entre las partes involucradas.

1. **Equipo de Desarrollo**

Responsables: Programadores Backend y Frontend, Analista de Base de Datos, Diseñador UX/UI, QA Tester. Son los encargados de convertir los requerimientos definidos por el Product Owner en incrementos funcionales del sistema. Cada miembro contribuye desde su área de especialización:

**Programador Backend:** desarrolla la lógica de negocio y gestiona la comunicación con la base de datos.

**Programador Frontend:** implementa la interfaz de usuario y las funcionalidades interactivas.

**Analista de Base de Datos:** diseña, normaliza e implementa los modelos de datos.

**Diseñador UX/UI:** construye prototipos, diseña la experiencia del usuario y valida la usabilidad del sistema.

**QA Tester:** realiza las pruebas unitarias, integrales y funcionales en los Sprints finales, asegurando la calidad del producto.

Cada Sprint finaliza con una revisión y retrospectiva, donde los roles colaboran para evaluar resultados, identificar mejoras y planificar la siguiente iteración.

Artefactos de Scrum

Los artefactos Scrum son documentos e informaciones creados y utilizados como parte del marco de trabajo Scrum. Proporcionan transparencia sobre el trabajo, el progreso y los objetivos del proyecto. Estos artefactos incluyen el Product Backlog (backlog del producto), el sprint backlog y los incrementos, cada artefacto define el trabajo de manera clara y concisa, permitiendo que los equipos Scrum planifiquen y prioricen tareas en los próximos Sprints. También ofrecen una visión del progreso de un proyecto Scrum, lo que ayuda a los equipos a seguir el progreso y Re priorizar tareas si es necesario.

Product Backlog:

El Product Backlog es el centro de planificación, en este punto se procesan los requisitos funcionales y no funcionales del sistema. Para ello, se representa en una lista dinámica y priorizada en ítems que describen el trabajo pendiente para alcanzar la visión completa del producto.

Su mantenimiento y priorización están bajo la responsabilidad del Product Owner, quien lo adapta conforme avanza el desarrollo y se obtiene retroalimentación del cliente o del entorno de negocio. En este proyecto, el Product Backlog servirá como base para la planificación de los Sprints definidos en el diagrama de Gantt.

Cada elemento del Backlog se formula como una historia de usuario o requisito específico, con criterios de aceptación, prioridad y responsable designado dentro del equipo Scrum.

**Estructura del Product Backlog**

Cada ítem del Product Backlog incluirá la siguiente información:

* + - ID: Código único del ítem (PB1, PB2, etc.)
    - Historia de usuario o requisito: Descripción breve siguiendo el formato “Como [rol], quiero [función], para [beneficio]”.
    - Criterios de aceptación: Condiciones objetivas que determinan si el ítem está completo.
    - Prioridad: Clasificación (Alta, Media, Baja) según valor para el producto.
    - Responsable: Rol asignado dentro del equipo (Product Owner, Desarrollador, QA Tester, etc.).
    - Estado: Situación actual del ítem (Pendiente, En desarrollo, Completado).

Tabla 1 Product Backlog

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Historia de Usuario / Requisito | Criterios de Aceptación | Prioridad | Responsable | Estado |
| PB1 | Como usuario, quiero registrarme y autenticarme en la plataforma para acceder a mis datos personales. | El sistema valida datos, genera sesión y protege contraseñas. | Alta | Product Owner / Dev Backend | Pendiente |
| PB2 | Como usuario, deseo gestionar mi perfil nutricional para registrar peso, edad y objetivos alimenticios. | Se permite editar datos y se almacenan correctamente en la BD. | Alta | Dev Frontend / Analista BD | Pendiente |
| PB3 | Como administrador, necesito un panel centralizado de control para gestionar usuarios y estadísticas. | Acceso solo autorizado; muestra métricas básicas de usuarios activos. | Media | Jefe de Proyecto / Dev Backend | Pendiente |
| PB4 | Como usuario, quiero recibir notificaciones y recordatorios de mis horarios de comida y ejercicios. | Sistema envía alertas automáticas y registra el historial. | Media | Dev Frontend / UX Designer | Pendiente |
| PB5 | Como sistema, deseo integrar una API externa para obtener información nutricional de alimentos. | Se conecta con API y retorna datos precisos al usuario. | Alta | Analista BD / Dev Backend | Pendiente |
| PB6 | Como usuario, quiero visualizar mi progreso nutricional en gráficos y reportes semanales. | Se generan gráficos dinámicos con datos históricos. | Media | Dev Frontend / QA Tester | Pendiente |
| PB7 | Como usuario, deseo consultar un módulo de ayuda o soporte técnico desde la plataforma. | Se habilita un módulo accesible con preguntas frecuentes. | Baja | UX Designer / QA Tester | Pendiente |
| PB8 | Como administrador, necesito generar reportes automáticos sobre usuarios y planes nutricionales. | Se exportan reportes en formato PDF y Excel. | Media | Dev Backend / Analista BD | Pendiente |
| PB9 | Como usuario, quiero cerrar sesión de forma segura para proteger mi información. | Logout funcional y eliminación de token de sesión. | Alta | Dev Backend | Pendiente |

El Product Backlog será gestionado de forma evolutiva e incremental, para ello cada Sprint tomará un conjunto de ítems según su prioridad y complejidad.

El Product Owner evaluará continuamente los avances y reordenará la lista en función del valor entregado al cliente.

Tabla 2 Nivel de prioridad

|  |  |  |
| --- | --- | --- |
| Nivel de prioridad | Criterios | Ejemplo de ítems |
| Alta | Funcionalidades esenciales para la operación del sistema. | PB1, PB2, PB5, PB9 |
| Media | Mejoras funcionales que aportan valor al usuario. | PB3, PB4, PB6, PB8 |
| Baja | Elementos complementarios o estéticos. | PB7 |

Sprint Backlog

En el Sprint Backlog es donde se juntan los elementos seleccionados en el Product Backlog, los cuales el equipo se compromete a desarrollar, es por ello que constituye una visión detallada del trabajo a realizar en un lapse de dos a cuatro semanas, incluyendo las tareas necesarias para presentar un incremento funcional del producto. Este se actualiza de

En el contexto del proyecto “Plataforma de gestión centralizada con módulo de nutrición personalizada”, el Sprint Backlog está alineado con las fases del diagrama de Gantt, donde cada fase (Fase 1 a Fase 6) equivale a un Sprint, permitiendo mantener un ritmo de desarrollo iterativo y controlado.

**Estructura del Sprint Backlog**

Cada Sprint incluye:

* Objetivo del Sprint (Sprint Goal)
* Historias de usuario seleccionadas del Product Backlog
* Tareas técnicas necesarias para implementarlas
* Responsables asignados
* Estado de avance
* Duración estimada

Tabla 3 Estructura del Sprint Backlog

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sprint** | **Objetivo del Sprint** | **Historias seleccionadas (PBs)** | **Tareas principales** | **Responsables** | **Estado** |
| **Sprint 1** (Fase 1: Inicio y Planificación) | Definir el alcance, requerimientos y planificación del proyecto. | PB1 | - Reunión inicial- Levantamiento de requerimientos- Definición de cronograma | Jefe de Proyecto, Analista Funcional | Completado |
| **Sprint 2** (Fase 2: Diseño) | Diseñar la arquitectura del sistema y la base de datos. | PB2, PB3 | - Diseño frontend/backend- Modelado de BD- Prototipo de interfaz | Analista BD, UX/UI Designer, Programador 2 | En desarrollo |
| **Sprint 3** (Fase 3: Desarrollo Inicial) | Implementar módulo de registro y autenticación. | PB1, PB9 | - Desarrollo backend- Desarrollo frontend- Pruebas iniciales | Programador 1, Programador 2, QA Tester | Pendiente |
| **Sprint 4** (Fase 4: Desarrollo Funcional) | Crear funcionalidades de comunicación y notificaciones. | PB4, PB5 | - Desarrollo módulo de notificaciones- Integración de APIs externas- Reuniones de seguimiento | Dev Backend, Analista BD | Pendiente |
| **Sprint 5** (Fase 5: Integración y Pruebas) | Integrar módulos y realizar pruebas funcionales completas. | PB3, PB6, PB8 | - Integración de módulos- Pruebas unitarias e integrales- Corrección de errores | QA Tester, Dev Backend, Dev Frontend | Pendiente |
| **Sprint 6** (Fase 6: Cierre) | Documentar y entregar el producto final al cliente. | PB7, PB8 | - Documentación técnica- Elaboración de manual de usuario- Revisión final | Analista Funcional, QA Tester, jefe de Proyecto | Pendiente |

El Sprint 1 (Fase 1: Inicio y Planificación) fue completado con éxito, cumpliendo el objetivo establecido de definir el alcance, los requerimientos y el cronograma del proyecto. Durante este sprint se realizaron las siguientes actividades principales: la reunión inicial de arranque, el levantamiento y análisis de requerimientos funcionales y no funcionales, y la definición del cronograma de trabajo.

Incremento

El Incremento es el resultado funcional y potencialmente entregable al finalizar cada sprint. Según la Guía de Scrum (Schwaber & Sutherland, 2020), el incremento debe representar una parte del producto que esté completamente integrada, probada y que cumpla con la Definición de Hecho (Definition of Done) establecida por el equipo. Cada nuevo incremento se suma a los anteriores, asegurando que el sistema evolucione de forma continua y coherente hacia el producto final.

En el contexto del proyecto “DESARROLLO DE UNA PLATAFORMA DE GESTIÓN CON UN MÓDULO DE NUTRICIÓN PERSONALIZADA PARA EL GIMNASIO FORCA & FITNESS”, el incremento refleja el avance acumulativo de la plataforma, dividido en seis iteraciones (Sprints), cada una generando una versión funcional validada del sistema.

Tabla 4 Tabla de Incremento

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sprint** | **Objetivo del Incremento** | **Entregables principales** | **Criterios de aceptación (Definition of Done)** | **Validación** |
| **Sprint 1** | Definir los requerimientos y plan del proyecto. | Documento de requerimientos, cronograma, acta de inicio, backlog inicial. | Documentación completa, validación del Product Owner. | Reunión de revisión y acta de validación. |
| **Sprint 2** | Diseñar la arquitectura y la base de datos del sistema. | Diagramas de arquitectura, modelo ER normalizado, prototipo de interfaz. | Diseño aprobado, coherencia con los requerimientos. | Validado por el Analista Funcional y el Jefe de Proyecto. |
| **Sprint 3** | Desarrollar módulo de registro y autenticación. | Backend funcional, interfaz de registro operativa, pruebas unitarias básicas. | Código ejecutable sin errores críticos, pruebas unitarias superadas. | QA Tester y Product Owner. |
| **Sprint 4** | Desarrollar módulo de comunicación y notificaciones. | API de notificaciones, integración de APIs externas. | Flujo funcional comprobado, integración completa. | QA Tester y Dev Backend. |
| **Sprint 5** | Integrar módulos y realizar pruebas completas. | Sistema integrado, informe de pruebas, corrección de errores. | Cumplimiento del 100 % de casos de prueba, integración estable. | Aprobación conjunta del equipo Scrum. |
| **Sprint 6** | Documentar y entregar el producto final. | Manual de usuario, documentación técnica, versión final del sistema. | Entrega sin defectos críticos, documentación verificada. | Validación final del Product Owner y cierre del proyecto. |

Ciclo de desarrollo (Sprints)

El ciclo de desarrollo del proyecto “DESARROLLO DE UNA PLATAFORMA DE GESTIÓN CON UN MÓDULO DE NUTRICIÓN PERSONALIZADA PARA EL GIMNASIO FORCA & FITNESS”, elaborado bajo la metodología ágil Scrum cuenta con Sprints; o ciclo de desarrollo, los cuales tienen una duración aproximada de 2 a 4 semanas por Sprint, siendo un total de seis hasta el final del proyecto, durante su desarrollo el equipo aborda las historias de usuario seleccionadas en el Backlog.

En el contexto del presente proyecto, el ciclo de desarrollo está alineado con las fases definidas en el diagrama de Gantt, de modo que cada fase corresponde a un sprint independiente. Esto garantiza una gestión ágil, transparente y controlada del progreso. Las revisiones de sprint se realizarán los lunes, con el propósito de evaluar los avances, identificar posibles bloqueos y definir los ajustes necesarios antes de iniciar las siguientes iteraciones.

Relación con el diagrama de Gantt

En referencia al diagrama de Gantt del proyecto, se realizaron seis Sprints, los cuales se alinean con las fases del plan de trabajo. Cada Sprint cuenta con las actividades a realizar, sus respectivos roles, además de contar con reuniones de revisión de avances.

Tabla 5 Tabla resumen de Sprints

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sprint / Fase | Duración | Fechas | Objetivo principal | Entregables (Incrementos) | Responsables principales |
| Sprint 1: Inicio y Planificación | 8 días | 9/08/25 – 9/16/25 | Definir requerimientos, alcance y cronograma del proyecto. | Documento de requerimientos, cronograma aprobado. | Jefe de Proyecto, Analista Funcional |
| Sprint 2: Diseño | 9 días | 9/17/25 – 9/26/25 | Diseñar la arquitectura del sistema y la base de datos. | Modelos de BD, prototipo de interfaz, arquitectura general. | Analista BD, UX/UI Designer |
| Sprint 3: Desarrollo Inicial | 11 días | 9/27/25 – 10/09/25 | Desarrollar el módulo de registro y autenticación. | Módulo funcional de registro. | Programadores, QA Tester |
| Sprint 4: Desarrollo Funcional | 12 días | 10/10/25 – 10/23/25 | Implementar funcionalidades de comunicación y notificaciones. | Módulo de notificaciones e integración de APIs. | Dev Backend, Analista BD |
| Sprint 5: Integración y Pruebas | 21 días | 10/24/25 – 11/18/25 | Integrar módulos y ejecutar pruebas funcionales. | Sistema integrado probado. | QA Tester, Dev Backend/Frontend |
| Sprint 6: Cierre | 11 días | 11/19/25 – 12/01/25 | Documentar y entregar el producto final al cliente. | Documentación técnica y manual de usuario. | Analista Funcional, Jefe de Proyecto |

Validación de cada Sprint

Al finalizar cada Sprint, se realiza una revisión junto con el Product Owner, Scrum Master y el equipo de desarrolladores, con el fin de verificar se cumplan los requisitos definidos en el Product Backlog.

Modelado del sistema

Diagrama de Casos de Uso

**Diagrama de caso de uso – Recepcionista**

El diagrama muestra las cuatro funcionalidades principales del recepcionista en el sistema: actualizar datos del alumno, gestionar inscripciones y renovaciones, registrar nuevos alumnos, y procesar pagos. Estas operaciones permiten al recepcionista administrar eficientemente la atención diaria y mantener actualizada la información de los clientes del gimnasio.

Ilustración 9 Diagrama de caso de uso – Recepcionista

Diagrama

El contenido generado por IA puede ser incorrecto.

Fuente: Elaboración propia

**Diagrama de Casos de Usos - Alumno**

El diagrama muestra las cuatro funcionalidades disponibles para el alumno en el sistema: consultar horarios de clases disponibles, consultar su plan nutricional personalizado, ver notificaciones sobre cambios o eventos, y consultar información general del gimnasio. Estas operaciones permiten al alumno acceder a los servicios y mantenerse informado sobre su programa de entrenamiento y las actividades del gimnasio.

Ilustración 10 Diagrama de Casos de Usos - Alumno

Diagrama

El contenido generado por IA puede ser incorrecto.

Fuente: Elaboración propia

**Diagrama de Casos de Usos – Alumno**

El diagrama muestra un Instructor como actor principal que interactúa con el sistema mediante cuatro funciones. Puede generar planes de alimentación nutricional, los cuales se relacionan con consultar planes existentes y actualizar recomendaciones. También puede diseñar rutinas de ejercicio y registrar incidencias. Es un sistema básico de gestión para entrenadores que combina aspectos nutricionales y de ejercicio físico.

Ilustración 11 Diagrama de Casos de Usos – Alumno

Diagrama

El contenido generado por IA puede ser incorrecto.

Fuente: Elaboración propia

**Diagrama de Casos de Usos – Administrador**

El diagrama muestra un Administrador como actor principal con seis funcionalidades del sistema. Puede gestionar clases y publicar cambios de instructor, asignar horarios de clases, notificar eventos especiales, registrar trabajadores y gestionar información del personal. Es un sistema administrativo para la gestión operativa de un centro de entrenamiento o gimnasio, enfocado en la organización de clases, personal y comunicaciones.

Ilustración 12 Diagrama de Casos de Usos – Administrador

Diagrama

El contenido generado por IA puede ser incorrecto.

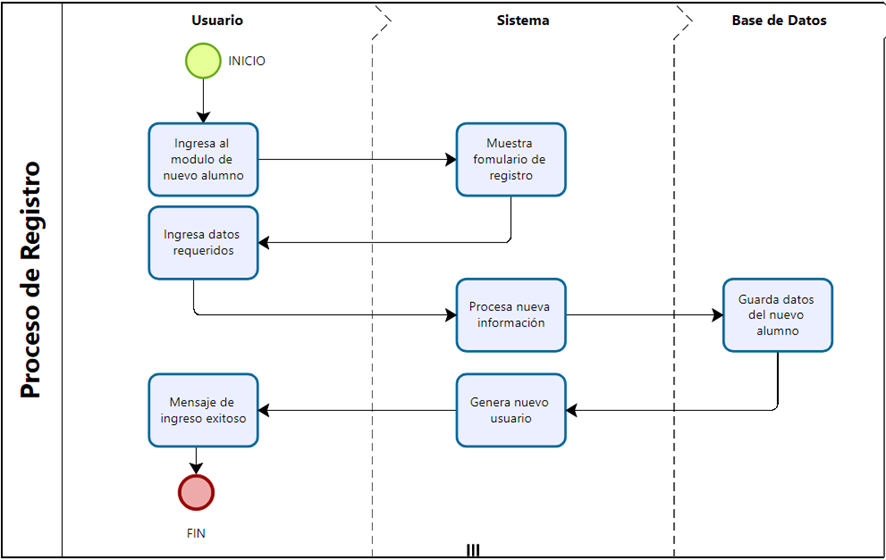
Fuente: Elaboración propia

**Diagrama de Casos de Actividad**

**Diagrama de Casos de Actividad – Proceso de Registro**

Este diagrama de actividad muestra el proceso de registro de un nuevo alumno. Inicia cuando el usuario ingresa al módulo de registro, el sistema muestra el formulario correspondiente y el usuario ingresa los datos requeridos. Luego el sistema procesa la información y la base de datos guarda los datos del nuevo alumno. Finalmente, el sistema genera un nuevo usuario y envía un mensaje de ingreso exitoso al usuario, donde termina el proceso. Es un flujo secuencial simple que representa la interacción entre usuario, sistema y base de datos durante el registro

Ilustración 13 Diagrama de Casos de Actividad – Proceso de Registro



Fuente: Elaboración propia

**Diagrama de Casos de Actividad – Proceso de plan Nutricional**

Este diagrama de actividad representa el proceso de selección de un plan nutricional. Inicia cuando el usuario ingresa a la plataforma de plan nutricional, el sistema muestra la interfaz y la base de datos carga los planes disponibles. El sistema permite seleccionar un plan y el usuario visualiza sus detalles. Existe un punto de decisión donde se pregunta si es agrado del alumno, si la respuesta es no, regresa a seleccionar otro plan. Si es sí, el sistema registra la nueva suscripción y la base de datos guarda el nuevo suscriptor. Finalmente, el sistema notifica al usuario, quien recibe la notificación y el proceso termina.

Ilustración 14 Diagrama de Casos de Actividad – Proceso de plan Nutricional

Diagrama

El contenido generado por IA puede ser incorrecto.

Fuente: Elaboración propia

**Diagrama de Casos de Actividad – Horario**

Inicia cuando el usuario accede al módulo de horario, el sistema muestra la interfaz y la base de datos carga el listado de clases disponibles. El usuario selecciona un horario y el sistema muestra los detalles de la clase mientras la base de datos obtiene sus datos. El usuario selecciona el horario deseado y hay un punto de decisión preguntando si es agrado del alumno. Si responde no, regresa a mostrar los detalles para elegir otro. Si responde sí, el usuario se inscribe al horario, el sistema registra al alumno y la base de datos guarda el nuevo registro. Finalmente, el sistema notifica al usuario, quien recibe el mensaje y el proceso termina.

Ilustración 15 Diagrama de Casos de Actividad – Horario

Diagrama

El contenido generado por IA puede ser incorrecto.

Fuente: Elaboración propia

**Diagrama de Casos de Actividad – Gestionar una Nueva Clase**

Este diagrama de actividad representa el proceso de gestión de una nueva clase por parte del administrador. Comienza cuando el usuario administrador accede al módulo de clases y el sistema muestra la interfaz de gestión. El administrador ingresa el nombre y descripción de la nueva clase, luego el sistema la abre en el sistema. A continuación, el administrador selecciona un instructor y la base de datos carga el listado de instructores disponibles. El administrador asigna un horario a la clase, el sistema registra el nuevo horario y la base de datos almacena la nueva clase. Finalmente, el administrador notifica el nuevo horario a los usuarios y el proceso termina.

Ilustración 16 Diagrama de Casos de Actividad – Gestionar una Nueva Clase

Diagrama

El contenido generado por IA puede ser incorrecto.

Fuente: Elaboración propia

**Diagrama de Casos de Actividad – Gestionar nuevo trabajador**

Este diagrama de actividad muestra el proceso de gestión de registro de un nuevo trabajador. Inicia cuando el administrador accede al módulo de clases y el sistema muestra la interfaz de registro. El administrador ingresa los datos del trabajador, el sistema procesa los datos ingresados y el administrador selecciona si será instructor o recepcionista. La base de datos guarda los nuevos datos y genera credenciales según el cargo asignado. Finalmente, el sistema genera un nuevo usuario, envía un mensaje de confirmación al administrador y el proceso termina. Es un flujo simple para dar de alta empleados con sus respectivos roles y credenciales de acceso.

Ilustración 17 Diagrama de Casos de Actividad – Gestionar nuevo trabajador

Diagrama

El contenido generado por IA puede ser incorrecto.

Fuente: Elaboración propia

Desarrollo del proyecto

Base de datos

Una base de datos como lo indica Kosinski (2025) se refiere a “un repositorio digital para almacenar, gestionar y proteger colecciones organizadas de datos”. Asi mismo tenemos dos tipos de base de datos, en este caso utilizaremos una base de datos relacional porque ofrece una estructura sólida y flexible para mantener la información organizada.

Por otra parte, también se elabora tres tipos de modelo de datos.

Modelo Conceptual:

El modelo conceptual representa la estructura lógica del sistema de gestión del gimnasio, orientado a la administración de usuarios, roles, clases, planes nutricionales e incidencias. Según Modelado de Datos Conceptual | Erwin, Inc. (2021), este modelo permite definir las entidades, relaciones y restricciones que organizan la información clave del sistema.

El sistema se centra en la entidad USUARIO, núcleo principal que almacena credenciales y datos de acceso. Cada usuario “Pertenece” a un ROL mediante una relación (N:1), donde el rol puede ser Administrador, Instructor, Recepcionista o Alumno. Esta entidad sirve como base para las entidades derivadas ADMINISTRADOR, INSTRUCTOR y ALUMNO, cada una vinculada mediante una relación 1:1 con USUARIO, asegurando que un usuario solo adopte un rol específico.

En cuanto a ALUMNO, se registran datos personales, físicos y de contacto, además de la fecha de inscripción y estado de membresía. La entidad INSTRUCTOR contiene información profesional y de contratación, vinculándose con ALUMNO mediante la tabla intermedia ALUMNO\_INSTRUCTOR, que establece una relación N:M entre ambos, con atributos como fecha de asignación y estado.

La gestión de clases se modela con la entidad CLASE, donde cada clase “Es impartida” por un INSTRUCTOR (N:1) e “Incluye” a múltiples ALUMNOS mediante la entidad intermedia INSCRIPCION\_CLASE, que establece la relación N:M entre ambas entidades y almacena el estado y la fecha de inscripción.

En la parte nutricional, la entidad PLAN\_NUTRICIONAL “Es creado” por un INSTRUCTOR (N:1) y “Asignado” a un ALUMNO (N:1). Contiene detalles sobre objetivos, calorías diarias, personalización y fecha de creación, asegurando seguimiento personalizado en la dieta de cada alumno.

Respecto a la gestión de incidencias, la entidad INCIDENCIA “Es registrada” por un INSTRUCTOR (N:1) y “Afecta” a un ALUMNO (N:1), permitiendo documentar problemas o eventos relevantes dentro de las actividades del gimnasio.

El modelo integra además la tabla ROL, que clasifica los tipos de usuarios y se relaciona directamente con USUARIO (1:N). La tabla ADMINISTRADOR permite la gestión del sistema y control general de usuarios, instructores y alumnos.

Finalmente, las restricciones de integridad garantizan coherencia en los datos:

Claves primarias únicas y autoincrementales.

Claves foráneas que aseguran correspondencia entre entidades dependientes (por ejemplo, Usuario.id\_rol → Rol.id\_rol, Clase.id\_instructor → Instructor.id\_instructor).

Restricciones UNIQUE en campos críticos como dni, email y username.

Valores por defecto que refuerzan consistencia lógica (ej. estado activo, fechas automáticas).

Este modelo conceptual proporciona una estructura robusta y coherente para gestionar usuarios, clases, planes y eventos dentro del sistema del gimnasio, asegurando trazabilidad, integridad de datos y escalabilidad funcional.

Modelo lógico

Este es un diagrama Entidad-Relación (ER) que muestra la estructura de una base de datos para un sistema de gestión académico.

Tablas Principales

Administrator Almacena los datos de los administradores del sistema. Contiene información como ID, usuario, nombre, apellidos, DNI y otros datos de contacto. Solo hay un índice asociado.

Rol Una tabla simple que define los roles disponibles en el sistema. Cada rol tiene un nombre.

Usuario Tabla central que gestiona todos los usuarios del sistema. Incluye credenciales (nombre de usuario, contraseña), email, ID de rol y fecha de registro. Es la base para la autenticación.

Instructor Contiene información específica de los instructores: datos personales, especialidad, fecha de contratación. Se relaciona con la tabla usuario para la autenticación.

Alumno Almacena datos de los estudiantes: nombre, apellidos, DNI, teléfono, dirección, género y peso/altura. Se vincula con usuario.

Clase Define las clases disponibles: nombre, descripción, duración, día de la semana, horarios de inicio y fin, instructor asignado y fecha de creación.

Incidencia Registro de problemas o incidentes: incluye descripción, tipo, instructor y alumno involucrados, con fecha/hora del incidente.

Alumno\_Instructor Tabla de relación que vincula alumnos con instructores.

Plan\_Nutricional Planes personalizados de nutrición para alumnos: descripción, objetivo, calorías diarias y notas de personalización.

Inscripción\_Clase Registra las inscripciones de alumnos en clases específicas, con fecha de inscripción y estado.

Relaciones Principales

Las líneas punteadas conectan las tablas mostrando cómo se relacionan entre sí. Por ejemplo, un alumno puede inscribirse en múltiples clases, un instructor puede tener múltiples alumnos, y cada alumno puede tener un plan nutricional personalizado.

Modelo físico

En el contexto del desarrollo de un proyecto que incluya una base de datos relacional, la elaboración de un modelo físico es un paso importante, esto debido a que como lo señala Team (2015) “Modelo físico de datos en DBMS: lo que necesita saber.”, Este modelo físico puede convertirse directamente en el diseño real de la base de datos, lo que permite seguir evolucionando en la gestión de la información.

Es por ello por lo que se elaboró el modelo físico del proyecto, este modelo muestra cómo es que se implementa en la base de datos del proyecto. Es por esa razón que incluye las tablas, columnas, llaves, tipos de datos.

La elaboración de este modelo nos permitió identificar y diseñar de una manera más estructurada la base de datos para almacenar la información referente a los candidatos, las postulaciones, evaluaciones, preguntas, y más tablas con las que trabajaremos.

Ilustración 18 Modelo físico

**Escala de tiempo

El contenido generado por IA puede ser incorrecto.**

Fuente: Elaboración propia

**Arquitectura Técnica del Sistema de Gestión de Gimnasio**

1. Visión General de la Arquitectura

El sistema de gestión de gimnasio sigue un modelo de Arquitectura de Tres Capas (3-Tier), separando claramente la presentación (Frontend), la lógica de negocio (Backend) y la persistencia de datos. Esta estructura garantiza escalabilidad, mantenibilidad y una alta cohesión entre los módulos.

Frontend (Cliente): Desarrollado con React, responsable de la Interfaz de Usuario (UI) y la experiencia de usuario (UX). Se comunica con el Backend mediante peticiones RESTful a través de HTTP.

Backend (Servidor): Implementado con Spring Boot, gestiona la lógica de negocio, la seguridad y el acceso a la base de datos.

Base de Datos (Persistencia): Utiliza MariaDB, una base de datos relacional robusta y de alto rendimiento.

2 Detalle de Dependencias y Componentes Clave (Backend)

La funcionalidad del servidor se construye sobre las siguientes dependencias de Maven, agrupadas por su función específica:

2.1. Persistencia y Base de Datos  
Texto

El contenido generado por IA puede ser incorrecto.

2.2. Seguridad y Autenticación (JWT)

Este módulo es fundamental para la protección de la API, implementando un sistema de autenticación sin estado (stateless) esencial para la comunicación con React.  
Texto

El contenido generado por IA puede ser incorrecto.

2.3. Funcionalidad de la Aplicación  
Texto

El contenido generado por IA puede ser incorrecto.

3. Implementación del Módulo de Seguridad (JWT)

Este es el paso más crítico para la comunicación React-Spring Boot.

Entidades de Usuario y Rol: Define las clases User (o Socio) y Role con las anotaciones de JPA.

Clases de Seguridad: Implementa tu JwtAuthenticationFilter, JwtTokenProvider y el UserDetailsService para manejar el login, registro y validación de tokens.

Controladores de Autenticación: Crea un AuthController para los endpoints /api/auth/login y /api/auth/register.

3. Desarrollo de los Módulos de Gestión

* Una vez que la seguridad funcione, puedes crear los módulos principales:
* Módulo de Socios/Clientes:
* Entidad: Socio (nombre, email, fecha\_registro, estado).
* Repositorio, Servicio y SocioController.
* Módulo de Membresías:
* Entidad: Membresia (tipo, precio, duración).
* Módulo de Clases/Actividades:
* Entidad: Clase (nombre, hora, instructor).

4. Análisis de Dependencias de tu Proyecto Spring Boot

Tu configuración de Maven incluye todos los componentes clave para un sistema de gestión moderno:  
Texto

El contenido generado por IA puede ser incorrecto.

4. Diagramas y Tablas

4.1 Vista Página de Inicio (Elaboración Propia)

Interfaz de usuario gráfica, Texto, Aplicación, Sitio web

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Aplicación

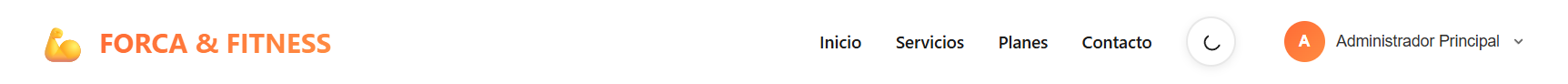
El contenido generado por IA puede ser incorrecto.  
Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

4.2 Vista Footer del Sistema (Elaboración Propia)

Captura de pantalla de computadora

El contenido generado por IA puede ser incorrecto.

4.3 Vista Header de Navegación (Elaboración Propia)  
  
4.4 Vista Sección de Servicios (Elaboración Propia)  
Escala de tiempo

El contenido generado por IA puede ser incorrecto.

4.4.1 Entrenamiento Personalizado (Elaboración Propia)  
Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

4.4.2 Planes Nutricionales (Elaboración Propia)  
Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

4.4.3 Clases Grupales (Elaboración Propia)  
Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

4.4.4 Seguimiento de Progreso (Elaboración Propia)  
Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

4.5 Vista Detalle de Entrenamiento Personalizado (Elaboración Propia)  
  
Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.  
Interfaz de usuario gráfica, Aplicación, Teams

El contenido generado por IA puede ser incorrecto.  
4.5.1 Plan Básico (Elaboración Propia)  
Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.  
4.5.2 Plan Premiun (Elaboración Propia)  
Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

4.5.1 Plan VIP (Elaboración Propia)  
Interfaz de usuario gráfica, Aplicación, Teams

El contenido generado por IA puede ser incorrecto.

4.6 Vista Página de Contacto (Elaboración Propia)  
Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.  
4.7 DASHBOARD (Elaboración Propia)  
Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

4.7.1 DASHBOARD - Perfil (Elaboración Propia)  
Interfaz de usuario gráfica, Texto, Aplicación, Word, Correo electrónico

El contenido generado por IA puede ser incorrecto.  
  
  
4.7.2 DASHBOARD - Sede (Elaboración Propia)  
Sitio web

El contenido generado por IA puede ser incorrecto.  
  
  
4.7.3 DASHBOARD – Equipo (Elaboración Propia)  
Interfaz de usuario gráfica, Sitio web

El contenido generado por IA puede ser incorrecto.

4.7.4 DASHBOARD - Usuarios (Elaboración Propia)  
Imagen que contiene Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.  
  
4.7.5 DASHBOARD - Membresía (Elaboración Propia)  
Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

4.7.7 DASHBOARD - Pago (Elaboración Propia)  
Interfaz de usuario gráfica, Sitio web

El contenido generado por IA puede ser incorrecto.  
  
4.7.8 DASHBOARD - Promociones (Elaboración Propia)  
  
Interfaz de usuario gráfica, Aplicación, Sitio web

El contenido generado por IA puede ser incorrecto.  
  
4.7.9 DASHBOARD - Promociones (Elaboración Propia)  
  
Imagen que contiene Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.4.7.9 DASHBOARD – Gimnasio (Elaboración Propia)  
  
Tabla, Sitio web

El contenido generado por IA puede ser incorrecto.

Bibliografía

Obtén más información sobre qué y cuándo comer para aprovechar al máximo tu entrenamiento físico. (s. f.). Mayo Clinic. <https://www.mayoclinic.org/es/healthy-lifestyle/fitness/in-depth/exercise/art-20045506>

Smart Fit - Sustentabilidad. (s. f.). <https://www.smartfit.com.pe/sustentabilidad>

Guest. (n.d.). Software engineering: a practitioner's approach - PDFCOFFEE.COM. *pdfcoffee.com*. <https://pdfcoffee.com/software-engineering-a-practitioner39s-approach-pdf-free.html>

Schwaber, K., & Sutherland, J. (2020). *La Guía Scrum: Las reglas del juego* (Traducción al español). Scrum.org. [https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf](https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-Spanish-Latin-South-American.pdf?utm_source=chatgpt.com)

Kosinski, M. (2025, 2 abril). Base de datos. *¿Qué es una base de datos?* <https://www.ibm.com/es-es/think/topics/database>