

ANALISIS Y REQUISITOS

Joan Casanovas | Rene Oviedo
Jerrarquixs 2024-25

ÍNDICE

ÍNDICE	2
INTRODUCCIÓN	3
TIPOS DE USUARIOS	4
REQUISITOS PARA EL FUNCIONAMIENTO DE LA APLICACIÓN	5
Requisitos funcionales	5
Requisitos no funcionales	5
INFORMACIÓN A GUARDAR EN LA BASE DE DATOS	6
MODELO ENTIDAD/RELACIÓN	7
MODELO RELACIONAL	8
CASOS DE USO	9

INTRODUCCIÓN

Este documento proporciona una visión general del proceso de recopilación, análisis y validación de requisitos para JoreFIT, asegurando que el producto final cumpla con los objetivos planteados.

La identificación y documentación precisa de los requisitos es fundamental para evitar malentendidos y errores costosos en etapas posteriores del desarrollo. En este análisis, se abordarán tanto requisitos funcionales como no funcionales, proporcionando una base sólida para la planificación, diseño y construcción del sistema. Los puntos expuestos y tratados en este documento, serán los siguientes:

- ★ Objetivo del proyecto
- ★ Tipos de usuarios.
- ★ Funcionalidades a desarrollar para el correcto funcionamiento del programa.
- ★ Requisitos funcionales y no funcionales.
- ★ Información a guardar en nuestra base de datos.



OBJETIVO DEL PROYECTO

El proyecto JoreFIT, una aplicación web destinada a ofrecer rutinas de entrenamiento personalizadas según los objetivos y nivel físico de los usuarios.

El objetivo principal de esta aplicación es permitir a los usuarios registrarse, elegir un plan de entrenamiento (como ganancia muscular, definición, pérdida de peso o ejercicios cardiovasculares) y recibir ejercicios adaptados a su nivel.



TIPOS DE USUARIOS

En la aplicación JoreFIT existen dos tipos de usuarios principales, cada uno con funcionalidades específicas que se detallan a continuación:

Administrador

Este perfil tiene acceso total a todas las funcionalidades de la plataforma. Sus principales tareas son:

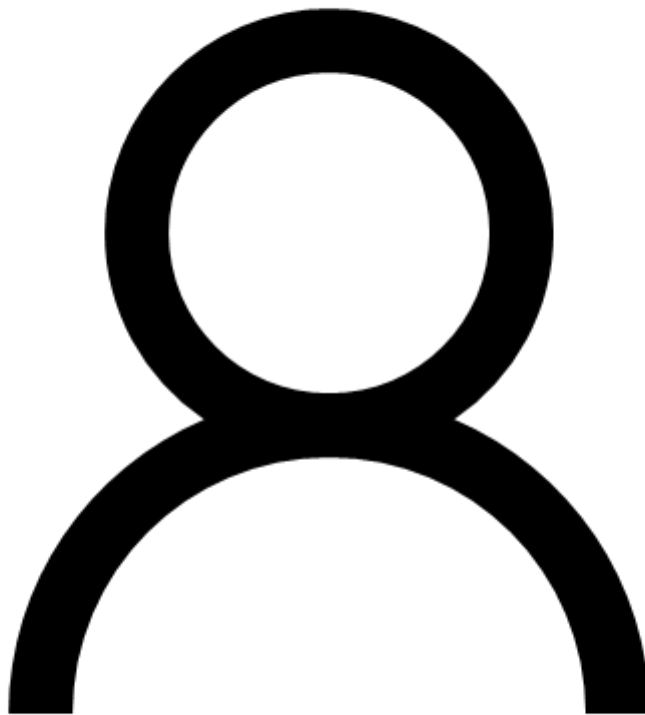
- ★ Gestionar los usuarios:
 - Visualizar todos los usuarios registrados.
 - Eliminar usuarios.
 - Modificar datos de cualquier usuario.
- ★ Gestionar los planes de entrenamiento:
 - Crear nuevos planes.
 - Editar información de planes existentes.
 - Eliminar planes que ya no se usen.
- ★ Gestionar ejercicios:
 - Crear ejercicios para los diferentes planes e intensidades.
 - Modificar y actualizar ejercicios.
 - Eliminar ejercicios obsoletos.
- ★ Gestionar planes y ejercicios asignados:
 - Eliminar planes o ejercicios asignados a los usuarios si es necesario.
- ★ Acceder a todas las rutas protegidas por rol de administrador.

Usuario

Este perfil está pensado para los clientes de la plataforma, quienes podrán personalizar su experiencia de entrenamiento. Sus funcionalidades incluyen:

- ★ Registro en la plataforma.
- ★ Inicio y cierre de sesión.
- ★ Recuperación de contraseña mediante verificación por correo.
- ★ Modificación de sus datos personales.

- ★ Consulta y contratación de planes de entrenamiento.
- ★ Visualización de sus propios planes contratados.
- ★ Visualización de los ejercicios asignados a cada plan.
- ★ Modificación de los ejercicios asignados (según disponibilidad).
- ★ Marcar ejercicios como completados para llevar el seguimiento del progreso.
- ★ Ver si su plan está en curso o finalizado.



REQUISITOS PARA EL FUNCIONAMIENTO DE LA APLICACIÓN

La aplicación JoreFIT contará con las siguientes funcionalidades, organizadas según los diferentes módulos que conforman el sistema. Estas funcionalidades definen todo lo que se debe programar para cumplir con los objetivos del proyecto y cubrir tanto las necesidades del usuario como las tareas del administrador.

Gestión de usuarios

- ★ Registro de nuevos usuarios.
- ★ Inicio de sesión y cierre de sesión.
- ★ Verificación de correo electrónico y recuperación de contraseña mediante código.
- ★ Modificación de datos personales.
- ★ Visualización de perfil.
- ★ Eliminación de usuarios (solo administrador).
- ★ Listado completo de usuarios (solo administrador).

Gestión de planes de entrenamiento

- ★ Visualización de todos los planes disponibles.
- ★ Consulta detallada de un plan específico.
- ★ Contratación de un plan por parte del usuario según su nivel y objetivo.
- ★ Creación, modificación y eliminación de planes (solo administrador).

Gestión de planes asignados

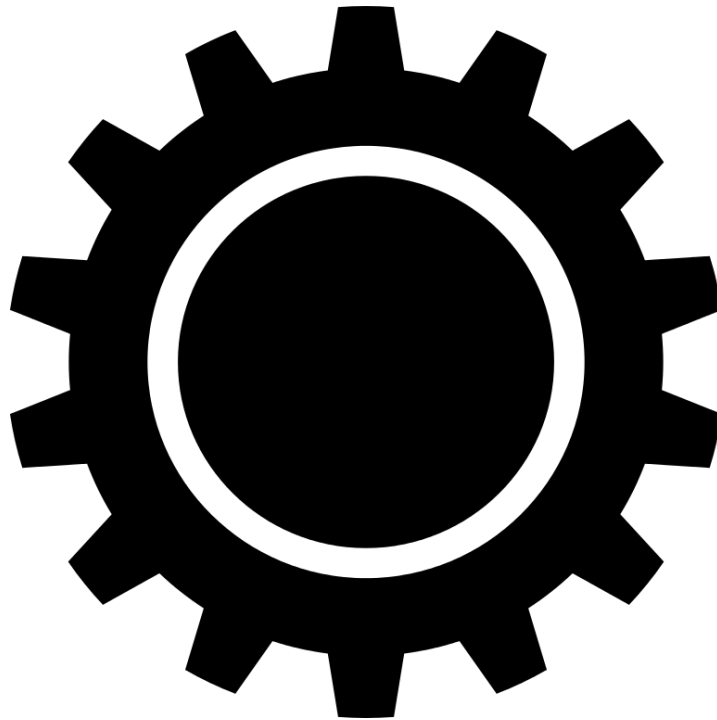
- ★ Asignación de un plan personalizado a un usuario.
- ★ Modificación y eliminación de planes asignados.
- ★ Visualización de los planes asignados por usuario.
- ★ Visualización de los propios planes contratados por cada usuario.
- ★ Consulta individual de cada plan asignado.

Gestión de ejercicios

- ★ Consulta de todos los ejercicios disponibles o de uno específico.
- ★ Creación, modificación y eliminación de ejercicios (solo administrador).
- ★ Asociación de ejercicios a los distintos tipos de planes e intensidades.

Gestión de ejercicios asignados

- ★ Asignación de ejercicios a un usuario según el plan contratado y su nivel.
- ★ Visualización de los ejercicios correspondientes a un plan específico.
- ★ Modificación o reemplazo de ejercicios asignados.
- ★ Actualización del estado del ejercicio (por ejemplo, marcar como completado).
- ★ Eliminación de ejercicios asignados.
- ★ Visualización del progreso del plan (en curso o completado).



INFORMACIÓN A GUARDAR EN LA BASE DE DATOS

La aplicación JoreFit gestionará una gran cantidad de información para poder ofrecer una experiencia personalizada a los usuarios y un control total al administrador. Por ello, será necesario almacenar los siguientes tipos de datos en la base de datos:

- ★ **Datos de los usuarios:** Se incluirán sus datos personales (nombre, apellidos, correo electrónico, fecha de nacimiento, peso, altura), así como sus credenciales de acceso. También se guardará información relacionada con su nivel físico y sus objetivos personales.
- ★ **Datos de pago:** La aplicación permitirá a los usuarios introducir opcionalmente sus datos de tarjeta para contratar servicios o planes. No obstante, también se podría considerar el uso de una pasarela de pago externa para mayor seguridad.
- ★ **Catálogo de planes de entrenamiento:** Cada plan contará con una descripción, duración, precio e imagen. Esta información permitirá a los usuarios elegir el plan que mejor se adapte a sus necesidades.
- ★ **Catálogo de ejercicios:** Cada ejercicio tendrá una imagen, descripción, el grupo muscular que trabaja, la intensidad (leve, media o alta), series, repeticiones y estará vinculado a un tipo de plan específico.
- ★ **Planes asignados a los usuarios:** Se registrará qué plan ha contratado cada usuario, así como si ese plan ha sido completado o no.
- ★ **Ejercicios asignados a cada plan y usuario:** Se almacenará qué ejercicios han sido asignados a cada usuario dentro de un plan, junto con su estado de realización. Esta funcionalidad es clave para hacer un seguimiento del progreso de cada usuario.

Tabla	Campos	Descripción
Usuarios	<ul style="list-style-type: none"> ● id: BIGINT, auto incremental, PK, NN. ● correo: STRING, único (máx. 255), UK, NN. Validación: formato email. ● contraseña: STRING, NN. Validación: mínimo 8 caracteres. ● nombre: STRING(50), NN. ● apellido: STRING(50), NN. ● fecha_nacimiento: DATE, NN. ● altura: DOUBLE, NULLABLE. ● pesoActual: DOUBLE, NN. ● pesoObjetivo: DOUBLE, NN. ● nivel: STRING, NN. Validación: debe ser uno de ['aficionado', 	Esta tabla almacena información básica de los usuarios que interactúan con el sistema.

	<ul style="list-style-type: none"> • 'intermedio', 'avanzado']. • rol: STRING, NN. Validación: debe ser 'cliente' o 'admin'. • numero_tarjeta: STRING(20), NULLABLE. Validación: numérica de 16 dígitos. • nombre_titular: STRING(100), NULLABLE. • cvv: STRING(3), NULLABLE. Validación: 3 dígitos numéricos. • fecha_vencimiento: STRING, NULLABLE. Validación: formato MM/AA. • recovery_code: STRING, NULLABLE. • code_expires_at: TIMESTAMP, NULLABLE. 	
Planes	<ul style="list-style-type: none"> • id: BIGINT, auto incremental, PK, NN. • nombre: STRING(100), NN. • descripcion: STRING(255), NN. • duracion: INTEGER sin signo, NN. Validación: mayor que 0. • precio: DOUBLE(8,2), NN. Validación: ≥ 0. • imagen: TEXT, NN. 	Esta tabla define los diferentes planes de ejercicios disponibles.
Ejercicios	<ul style="list-style-type: none"> • id: BIGINT, auto incremental, PK, NN. • imagen: TEXT, NN. • nombres: STRING(100), NN. • descripcion: TEXT, NN. • series: INTEGER sin signo, NN. Validación: > 0. • repeticiones: STRING, NN. Validación: formato "X reps" o rango tipo "10-12". • tipo: STRING(100), NN. Validación: debe coincidir con un plan. • musculo: STRING(50), NN. • intensidad: STRING, NN. Validación: debe ser uno de ['leve', 'media', 'alta']. 	Esta tabla contiene la información sobre cada ejercicio disponible.

planesAsignados	<ul style="list-style-type: none">● id: BIGINT, auto incremental, PK, NN.● idPlan: BIGINT, FK, NN → referencia a planes.id.● idUsu: BIGINT, FK, NN → referencia a usuarios.id.● completado: BOOLEAN, NN, valor por defecto false.	Esta tabla registra los planes asignados a cada usuario y su estado.
ejerciciosAsignados	<ul style="list-style-type: none">● id: BIGINT, auto incremental, PK, NN.● idPlan: BIGINT, FK, NN → planes.id.● idEjer: BIGINT, FK, NN → ejercicios.id.● idUsu: BIGINT, FK, NN → usuarios.id.● completado: BOOLEAN, NN, valor por defecto false.	Esta tabla registra los ejercicios asignados a los usuarios en un plan específico, junto con su estado.

REQUERIMIENTOS DEL PROYECTO

Para garantizar el desarrollo y puesta en marcha de la aplicación web, es imprescindible disponer de ciertos recursos tecnológicos y operativos. A continuación, se detallan los requerimientos necesarios para su correcta implementación.

Recursos para el Desarrollo y Despliegue

Para la construcción y funcionamiento óptimo de la aplicación, se utilizarán las siguientes tecnologías:

- **Frontend:** Angular 18, proporcionando una interfaz moderna y dinámica para mejorar la experiencia del usuario.
- **Backend:** Laravel (PHP), encargado de gestionar la lógica de negocio y la comunicación con la base de datos.
- **Base de datos:** XAMPP, que integra MySQL como sistema de gestión de datos, facilitando la administración y el mantenimiento.
- **Contenerización:** Uso de **Docker**, permitiendo una implementación eficiente en distintos entornos (desarrollo, pruebas y producción).
- **Alojamiento:** Servidor o VPS compatible con Docker y Docker Compose, para el despliegue estable de la aplicación.
- **Navegador:** El cliente deberá contar con un navegador web moderno (Chrome, Firefox, Edge) para acceder a la aplicación.

Requerimientos Adicionales

Además de los recursos tecnológicos, se contemplan otros aspectos clave para el correcto funcionamiento del sistema:

- **Dominio y Certificado SSL:** Se recomienda la adquisición de un dominio propio y la implementación de un certificado SSL para garantizar una comunicación segura.
- **Infraestructura del Cliente:** Es necesario confirmar si el cliente dispone del servidor adecuado para el despliegue o si debe incluirse en el presupuesto.
- **Acceso al Servidor:** En caso de que el despliegue no lo realice directamente el equipo de desarrollo, el cliente deberá proporcionar acceso al servidor.
- **Fecha de Entrega:** 19/05/2025

Este apartado garantiza que todos los elementos necesarios para el desarrollo y puesta en marcha del proyecto estén contemplados desde el inicio.