

+

MEMORIA

Carmen García Rodríguez 2º DAW



LIFELIFT

ÍNDICE

1. INTRODUCCIÓN.....	4
1.1 Presentación y objetivos.....	4
1.2 Contexto.....	4
1.3 Planteamiento del Problema (la idea).....	4
1.4 Análisis de costes.....	5
1.5 Plan de Financiación.....	5
1.6 Plan de recursos humanos.....	5
1.7 Plan de prevención de riesgo.....	6
2. REQUISITOS.....	8
2.1 Introducción.....	8
2.2 Descripción general.....	8
2.3 Requisitos Específicos.....	8
2.3.1 Requerimientos Funcionales.....	8
2.3.2 Requerimientos de Interfaces Externas.....	9
2.3.2.1 Interfaces de los Usuario.....	9
2.3.2.2 Interfaces Hardware.....	9
2.3.2.3 Interfaces Software.....	9
2.3.3 Requerimientos de Rendimiento.....	9
2.3.4 Obligaciones del Diseño.....	10
2.3.4.1 Estándares Cumplidos.....	10
2.3.4.2 Limitaciones Hardware.....	10
2.3.5 Atributos.....	10
2.3.5.1 Seguridad.....	10
2.3.5.2 Facilidades de Mantenimiento.....	10
2.3.5.3 Portabilidad.....	10
2.3.5.4 Otros Requerimientos.....	11
3.1 Introducción.....	12
3.2 Diagrama de Clases.....	12
3.3 Diagrama de Casos de Uso.....	12
4. DISEÑO.....	14
4.1 Introducción.....	14
4.2 Capa de Presentación.....	14
4.3 Capa de Negocio o Lógica de la Aplicación.....	14
4.4 Capa de Persistencia o Datos.....	15
5. IMPLEMENTACIÓN.....	17
5.1 Tecnologías utilizadas en el desarrollo del proyecto.....	17
5.2 Descripción del Proyecto.....	17

5.2.1 Capa de Presentación.....	18
5.2.2 Capa de Negocio o Lógica de la Aplicación.....	18
5.2.3 Capa de Persistencia o de Datos.....	18
6. EVALUACIÓN.....	20
6.1 Introducción.....	20
6.2 Validaciones de páginas de Estilo.....	20
6.3 Validación de enlaces.....	20
6.4 Validación de la Resolución.....	20
6.5 Validación de Navegadores.....	20
6.6 Otras validaciones.....	20
7. CONCLUSIÓN.....	21
7.1 Valoración Personal del Trabajo Realizado (análisis DAFO + análisis CAME).....	21
7.1.1 Análisis DAFO.....	21
7.1.1 Análisis CAME.....	22
7.2 Posibles ampliaciones.....	22
8. BIBLIOGRAFÍA.....	23

1. INTRODUCCIÓN

1.1 Presentación y objetivos

LifeLift es una aplicación web que proporciona una plataforma online de entrenamientos y nutrición, que permite a los clientes obtener entrenamientos personalizados a sus objetivos y estilo de vida.

El objetivo de este proyecto es interiorizar los conocimientos adquiridos en el ciclo formativo de Desarrollo de aplicaciones Web y ser capaz de aplicarlo a un proyecto real. En mi caso personal, además de utilizar las herramientas aprendidas en el módulo también añadí la herramienta de Ruby on Rails, por lo que mi otro objetivo era familiarizarme más con su uso, ya que antes de realizar el proyecto mis conocimientos sobre ella eran bastante limitados. Además mi objetivo también es aumentar los conocimientos aprendidos en clase probando nuevas tecnologías y maneras de desarrollar proyectos.

El objetivo de la plataforma en sí, suponiendo que se sacase al mercado, los objetivos son:

- Proveer una plataforma accesible y fácil de usar para usuarios interesados en mejorar su condición física y hábitos alimenticios.
- Facilitar a los entrenadores la gestión de sus clientes y la creación de rutinas personalizadas.
- Permitir a los administradores una gestión integral de la plataforma, optimizando la asignación de recursos y servicios.

1.2 Contexto.

El auge de las tecnologías digitales y la creciente conciencia sobre la importancia de un estilo de vida saludable han creado un entorno favorable para las aplicaciones de fitness y nutrición. LifeLift se desarrolla en un contexto donde la demanda por soluciones de bienestar personalizadas y accesibles está en constante aumento.

1.3 Planteamiento del Problema (la idea).

La idea principal para el desarrollo de la aplicación era crear una plataforma online de entrenamiento que permita a los clientes obtener entrenamientos personalizados a sus objetivos y estilo de vida, además de proporcionarles recomendaciones nutricionales en función de los entrenamientos que realice.

Después de desarrollar en detalle la idea resultó en una aplicación donde los usuarios pueden iniciar sesión como usuario, entrenador o administrador. Los usuarios llenarán un formulario de inscripción donde se les preguntará sobre sus objetivos y tendrán un espacio personal donde verán las rutinas de entrenamiento que se les asignen. Los entrenadores tendrán un espacio personal donde puedan ver a los usuarios a los que entrena y crearles rutinas personalizadas. Y los administradores podrán gestionar los ejercicios y asignarles usuarios a los

entrenadores para que les creen las rutinas. Además el administrador también podrá acceder a todos los contenidos a los que tenga acceso un usuario cualquiera.

Después del planteamiento detallado de la idea teniendo en cuenta las funcionalidades que le podría realizar con las herramientas que iba a utilizar, creo que la idea final se aproxima lo suficiente a la idea planteada en la propuesta.

1.4 Análisis de costes.

El análisis de costes de LifeLift contempla una serie de gastos estimados necesarios para poner en marcha la plataforma. Estos incluyen un servidor web con un coste aproximado de 100€ al año, aportaciones en efectivo de los socios que ascienden a 20.000€, y aportaciones en especie valoradas en 4.000€. Además, se considera un préstamo bancario de 5.000€, y gastos de publicidad y promoción estimados en 5.000€. Las nóminas mensuales para cada empleado se proyectan en 1.500€, con un costo adicional de 400€ por empleado para la seguridad social y una cuota mensual de autónomos de 1.000€.

Sin embargo, si el proyecto se lanzase en este momento, los costes iniciales serían mínimos. Solo se necesitaría la inversión para el servidor y el pago de los salarios para el desarrollador y el entrenador. Dado que inicialmente asumiría ambas funciones, me asignaría el sueldo más bajo permitido por convenio, optimizando así el uso de los recursos disponibles.

1.5 Plan de Financiación.

El plan de financiación de LifeLift se compone de una combinación de financiación propia y ajena para cubrir los costes estimados del proyecto. La financiación propia incluye aportaciones en efectivo de los socios, que ascienden a 20.000€, y aportaciones en especie valoradas en 4.000€. Además, se ha considerado un préstamo bancario de 5.000€ para complementar la inversión inicial. Estos fondos se destinan a cubrir los gastos de servidor web, publicidad y promoción, nóminas, seguridad social, y la cuota de autónomos.

Inicialmente, los requisitos de financiación serían mínimos. Solo se necesitaría cubrir la inversión del servidor y los salarios del desarrollador y entrenador. En esta fase, asumiría ambas funciones, optimizando los recursos financieros disponibles. A medida que la plataforma crezca y sea necesario contratar más entrenadores, se revisará y ajustará el plan de financiación para garantizar la sostenibilidad y expansión del proyecto.

1.6 Plan de recursos humanos.

El plan de recursos humanos para LifeLift está diseñado para garantizar que el proyecto cuente con el personal adecuado desde su inicio y a lo largo de su crecimiento. Inicialmente, el equipo estará compuesto por dos miembros clave: el desarrollador web y el profesional en Ciencias de la Actividad Física y el Deporte con especialización en nutrición.

El desarrollador web, función que desempeñará personalmente, será responsable de crear y mantener la estructura y funcionalidad de la plataforma, así como de proporcionar soporte técnico continuo. Esta persona será fundamental en la implementación y actualización de las características técnicas de la aplicación, asegurando que los usuarios tengan una experiencia fluida y eficiente.

El profesional en Ciencias de la Actividad Física y el Deporte será encargado de diseñar y supervisar las rutinas de entrenamiento y planes de nutrición personalizados para los usuarios. Este rol es vital para ofrecer un servicio de alta calidad y cumplir con los objetivos de salud y fitness de nuestros clientes. Dada la etapa inicial del negocio, esta posición requerirá una capacidad multifuncional para abarcar todos los aspectos relacionados con el entrenamiento y la nutrición.

Conforme la empresa crezca y la base de usuarios se expanda, se planificará la contratación de más entrenadores y nutricionistas para mantener la calidad del servicio personalizado. Además, se considerará la incorporación de personal adicional en áreas como marketing, atención al cliente y administración, asegurando así una operación eficiente y un crecimiento sostenible. Este plan de expansión se basará en el desempeño financiero y la demanda del mercado, permitiendo a LifeLift adaptarse y evolucionar según las necesidades del negocio.

1.7 Plan de prevención de riesgo

El Plan de Prevención de Riesgos para LifeLift es esencial para identificar, evaluar y mitigar los posibles riesgos que podrían afectar el desarrollo y operación de la plataforma. Este plan se centra en varios tipos de riesgos: técnicos, operacionales, financieros, legales y de mercado.

- **Riesgos Técnicos:** Para mitigar los riesgos asociados a fallos técnicos, se implementará un riguroso plan de pruebas y mantenimiento. Esto incluye pruebas regulares de la plataforma, actualizaciones de seguridad y copias de seguridad periódicas de todos los datos. Se establecerá un protocolo de respuesta rápida para resolver cualquier problema técnico que pueda surgir, minimizando así el tiempo de inactividad y los inconvenientes para los usuarios.
- **Riesgos Operacionales:** Dado que inicialmente el equipo es reducido, la dependencia de un número limitado de personal clave representa un riesgo significativo. Para abordar esto, se desarrollarán procedimientos documentados para todas las operaciones críticas. Además, se fomentará la formación continua para asegurar que más miembros del equipo puedan asumir responsabilidades clave en caso de necesidad.
- **Riesgos Financieros:** La empresa se enfrentará a riesgos financieros relacionados con la falta de flujo de caja y la necesidad de financiamiento adicional. Para mitigar estos riesgos, se llevará a cabo una planificación financiera detallada y se mantendrá una reserva de emergencia. También se explorarán diversas fuentes de financiación, como préstamos bancarios y aportaciones de capital, para asegurar la estabilidad financiera en los primeros años de operación.

- **Riesgos Legales:** Para evitar problemas legales, se cumplirá estrictamente con todas las normativas locales e internacionales relacionadas con la privacidad de los datos, el comercio electrónico y la publicidad. Se contratará un asesor legal para revisar todos los términos de servicio y políticas de privacidad, garantizando que la empresa opere dentro del marco legal correspondiente.
- **Riesgos de Mercado:** El mercado del fitness digital es altamente competitivo y cambiante. Para mitigar los riesgos de mercado, se realizará un análisis continuo de la competencia y de las tendencias del sector. Además, se adoptará una estrategia de marketing flexible que permita ajustarse rápidamente a las condiciones del mercado y a las preferencias de los consumidores.

2. REQUISITOS

2.1 Introducción.

La especificación de requisitos es un paso crucial en el desarrollo de una aplicación, ya que define las necesidades y expectativas de los usuarios. En esta sección detallaré los requisitos necesarios para el correcto funcionamiento de la aplicación, asegurando que se cumplan los objetivos del proyecto y se brinde una experiencia de usuario satisfactoria.

2.2 Descripción general.

El proyecto consiste en el desarrollo de una aplicación web para el ámbito del fitness, que permitirá a los usuarios acceder a rutinas de entrenamiento personalizadas y recibir asesoramiento nutricional. La aplicación contará con diferentes tipos de usuarios, incluyendo clientes, entrenadores y administradores, cada uno con sus propias funcionalidades y privilegios. El objetivo principal es proporcionar una plataforma intuitiva y eficiente que facilite la interacción entre usuarios y mejore su experiencia en el ámbito del fitness y la salud.

2.3 Requisitos Específicos.

Los requisitos específicos del sistema se dividen en diferentes categorías para comprender mejor las necesidades y funcionalidades del proyecto. A continuación, se detallan los requisitos funcionales, que describen las acciones que el sistema debe ser capaz de realizar para satisfacer los objetivos del usuario.

2.3.1 Requerimientos Funcionales.

- Gestión de Usuarios:
 - Registro de nuevos usuarios como clientes, entrenadores o administradores.
 - Inicio de sesión para acceder a las funcionalidades del sistema.
- Gestión de Entrenamientos:
 - Creación y asignación de rutinas de entrenamiento por parte de los entrenadores.
 - Visualización de las rutinas asignadas para los clientes.
 - Registro de progreso y seguimiento de las actividades realizadas.
- Gestión Administrativa:
 - Gestión de ejercicios por parte de los administradores.
 - Gestión de usuarios que permita asignarles el rol de entrenador o administrador.
 - Asignación de entrenador a los clientes.
 - Acceso y supervisión de todas las funcionalidades del sistema.
- Interfaz de Usuario:
 - Diseño intuitivo y fácil de usar para todas las funciones del sistema.
 - Visualización de las rutinas de entrenamiento.
 - Posibilidad de marcar como completado cada uno de los ejercicios.

2.3.2 Requerimientos de Interfaces Externas.

2.3.2.1 Interfaces de los Usuario.

Las interfaces de los usuarios constituyen la parte visible y accesible del sistema. Deben ser intuitivas y amigables para facilitar su uso por parte de los clientes, entrenadores y administradores. Los requisitos específicos son:

- **Registro de Usuarios:** Debe haber un formulario de registro que permita a los usuarios crear una cuenta proporcionando la información necesaria, como nombre, correo electrónico y contraseña. Todos los usuarios se registrarán inicialmente como clientes. Luego, el administrador del sistema podrá otorgar el rol de entrenador o administrador a los usuarios seleccionados a través de una interfaz de administración.
- **Inicio de Sesión:** Se debe proporcionar una interfaz de inicio de sesión que permita a los usuarios acceder al sistema introduciendo su correo electrónico y contraseña previamente registrados.
- **Perfil de Usuario:** Cada tipo de usuario debe tener un perfil personalizado que muestre la información relevante, como nombre, correo electrónico, tipo de usuario y cualquier otro detalle específico según el rol.
 - Cliente: Historial de sus rutinas de entrenamiento
 - Entrenador: Listado de sus clientes y por cada cliente las rutinas que tiene asignadas.
 - Administrador: listado de clientes y de ejercicios.

2.3.2.2 Interfaces Hardware.

Aunque el proyecto se centra en una aplicación web, es importante considerar los dispositivos hardware en los que los usuarios pueden acceder a la aplicación. Lo más habitual es que estos sean dispositivos móviles, por lo que la aplicación debe ser compatible y responsive con una variedad de dispositivos móviles, como teléfonos inteligentes y tabletas, para garantizar una experiencia de usuario consistente y óptima en diferentes plataformas. No se requiere hardware específico más allá de un dispositivo con capacidad de acceder a internet.

2.3.2.3 Interfaces Software.

Las interfaces software son los sistemas operativos y navegadores web compatibles con la aplicación. La aplicación debe ser compatible con todos los navegadores, pero principalmente con los más usados, como son Google Chrome, Mozilla Firefox, Safari y Microsoft Edge, para garantizar que los usuarios puedan acceder a ella sin problemas independientemente del navegador que utilicen.

El sistema operativo recomendado para desplegar la aplicación es Linux ya que Rails se instala más fácilmente y funciona en él, aunque se podría igualmente en cualquier otro.

2.3.3 Requerimientos de Rendimiento.

- **Tiempo de Respuesta:** La aplicación debe cargar la página principal en menos de 3 segundos.

- Escalabilidad: El sistema debe poder llegar a manejar hasta 10,000 usuarios activos simultáneamente sin degradación del rendimiento. Este número de usuarios es importante para que la aplicación llegue a ser totalmente rentable.
- Disponibilidad: La aplicación debe estar operativa y accesible para los usuarios el 99.9% del tiempo durante un año

2.3.4 Obligaciones del Diseño.

2.3.4.1 Estándares Cumplidos.

La aplicación debe cumplir con las normas de accesibilidad web ([WCAG 2.1](#)). Por las que debe ser perceptible, operable, comprensible y robusta

Uso de estándares de seguridad web para proteger datos sensibles ([OWASP](#)), que ayudan a evitar vulnerabilidades y a realizar pruebas de seguridad.

2.3.4.2 Limitaciones Hardware.

No existen limitaciones significativas en cuanto a hardware, siempre que los dispositivos sean capaces de ejecutar un navegador web moderno. En cuanto a memoria, ningún dispositivo tendrá problemas ya que la carpeta de despliegue de la aplicación sólo ocupa 100Mb.

2.3.5 Atributos.

2.3.5.1 Seguridad.

El registro de la aplicación consta de protección de datos personales, incluidas contraseñas, mediante cifrado. Esta acción se realiza mediante [Strong Parameters](#), método que nos facilita tener control y seguridad al procesar datos. Se llaman así porque nos permite filtrar los parámetros que recibimos, de modo que solamente los parámetros que indicamos son tomados en cuenta por nuestro controlador.

La aplicación también cuenta con un sistema de paginación que no permite aumentar mediante la url el número de elementos por página para disminuir la apropiación de datos por [web scraping](#)

2.3.5.2 Facilidades de Mantenimiento.

Las facilidades de mantenimiento de la aplicación se basan en partir de un código limpio y ordenado, además de estar programado en un lenguaje de bajo nivel y por tanto fácilmente comprensible. Además cuenta con un control de versiones en Git para gestionar el desarrollo del proyecto.

2.3.5.3 Portabilidad.

La aplicación deberá ser portable entre distintos servidores y ser fácilmente desplegable en distintas plataformas de hosting.

2.3.5.4 Otros Requerimientos.

Usabilidad: La interfaz debe ser intuitiva y fácil de usar, con una curva de aprendizaje mínima.

Fiabilidad: El sistema debe ser robusto y capaz de recuperarse rápidamente de fallos.

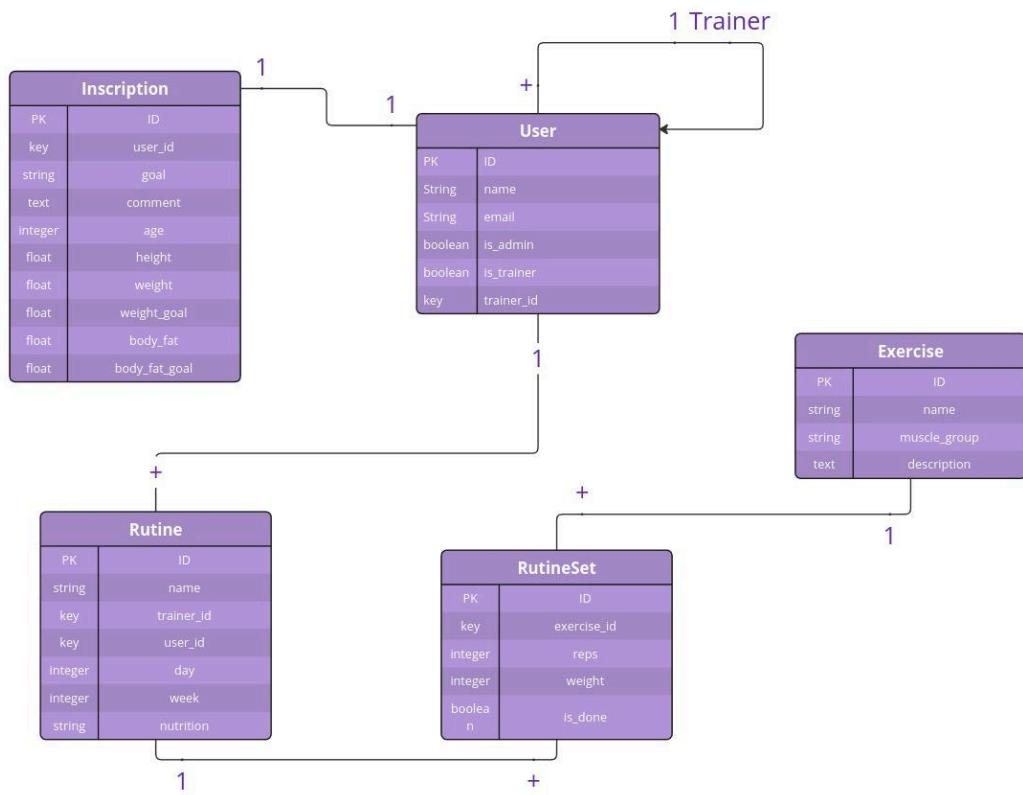
3. ANÁLISIS

3.1 Introducción.

En este apartado se incluyen dos diagramas que representan la estructura y funcionamiento de la aplicación.

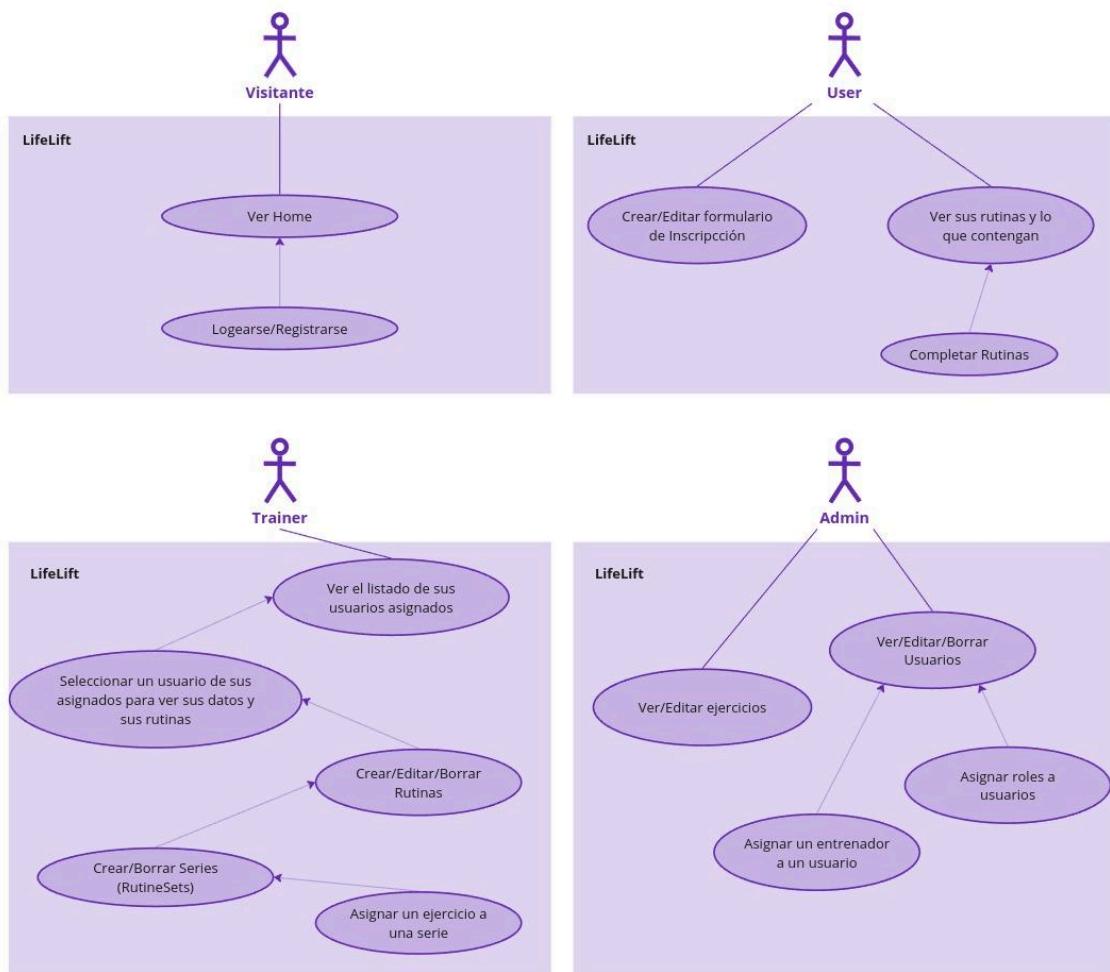
3.2 Diagrama de Clases.

Este diagrama muestra las tablas creadas en la base de datos y las respectivas relaciones que tienen entre sí.



3.3 Diagrama de Casos de Uso.

En este diagrama se representan las distintas vistas y funcionalidades que tiene la aplicación dependiendo del tipo de usuario con el que se acceda.



4. DISEÑO

4.1 Introducción.

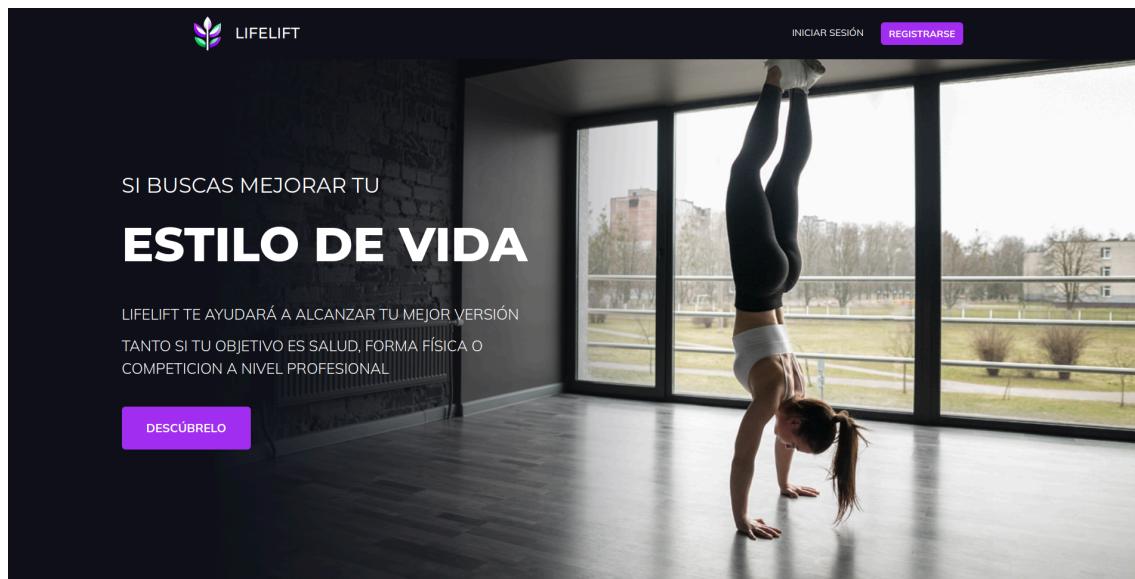
En esta sección se detalla el diseño arquitectónico y funcional de la aplicación. El diseño está estructurado en capas para facilitar el desarrollo, mantenimiento y escalabilidad de la aplicación. La arquitectura de capas incluye la capa de presentación, la capa de negocio o lógica de la aplicación, y la capa de persistencia o datos.

4.2 Capa de Presentación.

La capa de presentación es la interfaz de usuario de la aplicación. Es la capa con la que los usuarios interactúan directamente y está diseñada para ser intuitiva y fácil de usar. Esta capa incluye:

- **Diseño de la Interfaz de Usuario (UI):** Se han utilizado principios de diseño de UX/UI para crear una experiencia de usuario amigable y accesible. La interfaz incluye formularios de registro e inicio de sesión, paneles de control para usuarios, entrenadores y administradores, y vistas para visualizar y gestionar rutinas de entrenamiento.
- **Navegación y Usabilidad:** La navegación es clara y consistente, permitiendo a los usuarios moverse fácilmente entre las diferentes secciones de la aplicación.

Home



Interfaz usuario



Área de entrenamiento de Carmen

[EDITAR INSCRIPCIÓN](#)

Rutinas

[Semana 4](#)
[Semana 3](#)
[Semana 2](#)
[Semana 1](#)
Nombre Rutina
LUNES PECHO Y HOMBRO
Día
4
Nutrition
Día alto en hidratos
MARTES CIRCUITO
2
4
Día alto en hidratos
JUEVES PIerna Y GLUTEO
3
4
Día alto en hidratos
VIERNES ESPALDA Y BICEPS
4
4
Día bajo en grasa
SÁBADO ABDOMINALES
5
4
Día bajo en grasa

Nutrición

Dieta Baja en Hidratos de Carbono

- Consumo de Proteínas de Calidad: Incluye carnes magras, pescado, huevos, y productos lácteos bajos en grasa. Las

Dieta Alta en Hidratos de Carbono

- Carbohidratos Complejos: Prioriza fuentes de carbohidratos complejos como cereales integrales (arroz integral, quinoa, avena).

Dieta Baja en Grasas

- Proteínas Magras: Incluye fuentes de proteínas bajas en grasa como pollo sin piel, pavo, pescado blanco, y productos lácteos bajos en grasa o sin grasa.


[CERRAR SESIÓN](#)

Rutina Lunes Pecho y Hombro

Series

Ejercicio
Repeticiones
Peso
Completado

Flexiones
25
0 Kg

Flexiones
25
0 Kg

Press Banca
15
30 Kg

Press Banca
12
40 Kg
COMPLETAR

Press Banca
8
50 Kg
COMPLETAR

Interfaz Entrenador:

The screenshot shows the 'Mis Usuarios' (My Users) section of the LifeLift Trainer dashboard. At the top, there is a search bar labeled 'Buscar por nombre:' with a 'BUSCAR' button. Below the search bar, a table displays a single user entry:

Nombre	Email
Carmen	carmen@gmail.com

Next to the user entry is a 'VER USUARIO' (View User) button.

The screenshot shows the 'Rutinas' (Routines) section of the LifeLift Trainer dashboard. At the top, there is a navigation bar with tabs: 'Semana 4', 'Semana 3', 'Semana 2', and 'Semana 1'. Below the tabs, a table lists five routines:

Nombre Rutina	Día	Semana	Nutrition	EDITAR	BORRAR
Lunes Pecho y Hombro	1	4	Día alto en hidratos	EDITAR	BORRAR
Martes Circuito	2	4	Día alto en hidratos	EDITAR	BORRAR
Jueves Pierna y Glúteo	3	4	Día alto en hidratos	EDITAR	BORRAR
Viernes Espalda y Biceps	4	4	Día bajo en grasa	EDITAR	BORRAR
Sábado Abdominales	5	4	Día bajo en grasa	EDITAR	BORRAR

Below the table is a 'CREAR RUTINA' (Create Routine) button.

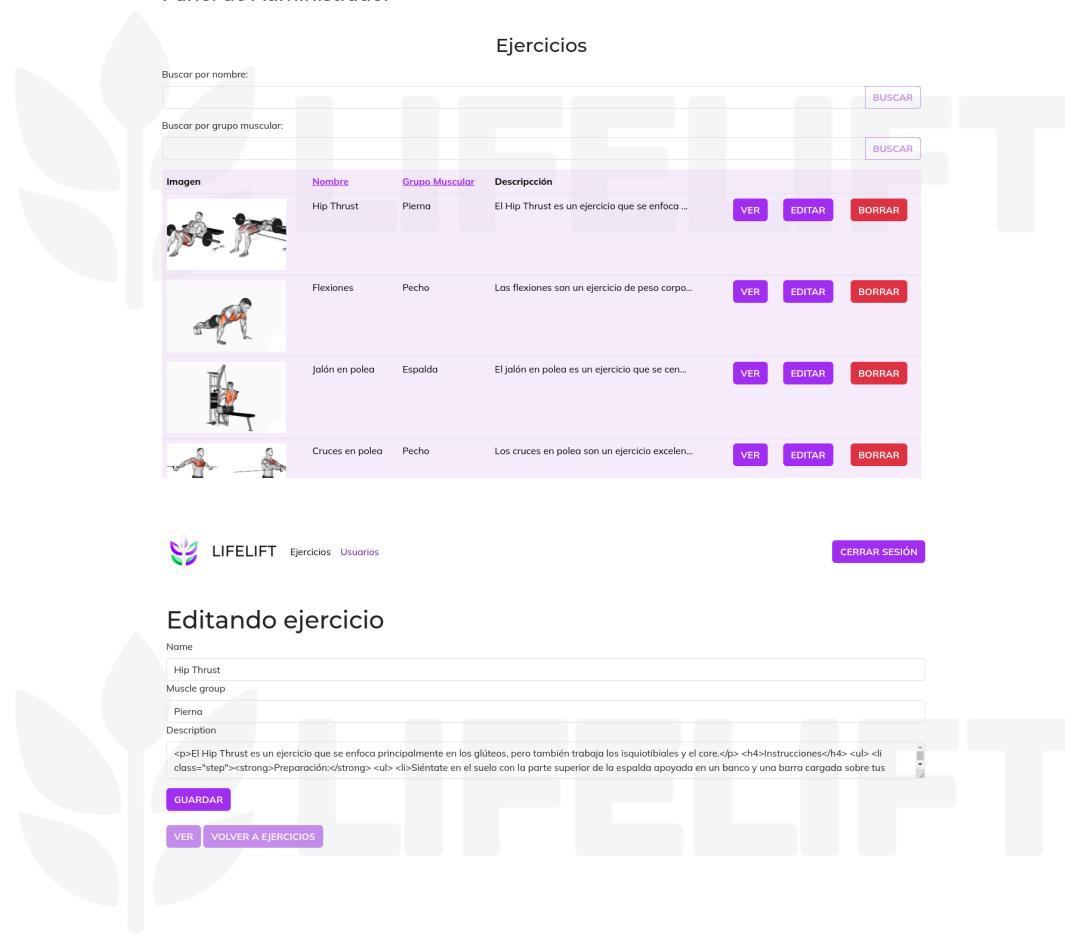
At the bottom of the page, there is an 'Inscripción' (Registration) section containing the following information:

Objetivos: Ganar masa muscular
Edad: 20
Altura: 1.58
Peso: 51.0
Objetivo: Bajar 51.0

Editar Rutina Lunes Pecho y Hombro

Series

Ejercicio	Repeticiones	Peso	
	Flexiones	25	0 Kg BORRAR
	Flexiones	25	0 Kg BORRAR
	Press Banca	15	30 Kg BORRAR
	Press Banca	12	40 Kg BORRAR
	Press Banca	8	50 Kg BORRAR

Interfaz Administrador:


LIFELIFT Ejercicios Usuarios CERRAR SESIÓN

Panel de Administrador

Ejercicios

Buscar por nombre: BUSCAR

Buscar por grupo muscular: BUSCAR

Imagen	Nombre	Grupo Muscular	Descripción	VER	EDITAR	BORRAR
	Hip Thrust	Pierna	El Hip Thrust es un ejercicio que se enfoca ...	VER	EDITAR	BORRAR
	Flexiones	Pecho	Las flexiones son un ejercicio de peso corpo...	VER	EDITAR	BORRAR
	Jalón en polea	Espalda	El jalón en polea es un ejercicio que se cen...	VER	EDITAR	BORRAR
	Cruces en polea	Pecho	Los cruces en polea son un ejercicio excel...	VER	EDITAR	BORRAR

LIFELIFT Ejercicios Usuarios CERRAR SESIÓN

Editando ejercicio

Name: Hip Thrust
Muscle group: Pierna
Description:

El Hip Thrust es un ejercicio que se enfoca principalmente en los glúteos, pero también trabaja los isquiotibiales y el core.

Preparación: Siéntate en el suelo con la parte superior de la espalda apoyada en un banco y una barra cargada sobre tus hombros.

The screenshot displays two views of the LifeLift application:

- Panel de Administrador (Admin Panel):** This view shows a table of users with columns: Nombre (Name), Email, Es Admin (Is Admin), Es Entrenador (Is Trainer), Entrenador (Trainer), and Inscripción (Registration). Each user row has three buttons: VER (View), EDITAR (Edit), and BORRAR (Delete).

Nombre	Email	Es Admin	Es Entrenador	Entrenador	Inscripción			
Trainer	trainer@lifelift.com	No	Si	No tiene	No	VER	EDITAR	BORRAR
Admin	admin@lifelift.com	Si	Si	No tiene	No	VER	EDITAR	BORRAR
Carmen	carmen@gmail.com	No	No	Trainer	Si	VER	EDITAR	BORRAR

- Editando Usuario (Editing User):** This view shows a form for editing a user named "Trainer". It includes fields for Nombre (Name), Email (Email), and checkboxes for Es Admin (Is Admin) and Es Entrenador (Is Trainer). Below the form are buttons for GUARDAR (Save) and VOLVER A USUARIOS (Return to Users).

Para obtener este resultado, la aplicación consta de las siguientes vistas:

- Vistas únicas para el usuario Admin: Aquí podemos encontrar vistas para los usuarios y los ejercicios y vistas para ambos como es el menú.

```
└── admin
    ├── exercises
    │   ├── _form.html.erb
    │   ├── _search.html.erb
    │   ├── edit.html.erb
    │   ├── index.html.erb
    │   ├── new.html.erb
    │   └── show.html.erb
    ├── shared
    │   └── _menu.html.erb
    └── users
        ├── _form.html.erb
        ├── _search.html.erb
        ├── edit.html.erb
        ├── index.html.erb
        └── show.html.erb
```

- Vistas del login y registro: Creadas por la gema de Devise, modifiqué las que necesité y las demás las dejé para usarlas en posibles mejoras.

```
└── devise
    ├── confirmations
    ├── mailer
    ├── passwords
    ├── registrations
    ├── sessions
    ├── shared
    └── unlocks
```

- Vistas de la Home: Aquí podemos encontrar los menús que se ven con y sin iniciar sesión.

```
└── home
    ├── _menu_logged.html.erb
    ├── _menu.html.erb
    └── index.html.erb
```

- Vistas únicas para el usuario cliente: como el formulario de inscripción y mi cuenta.

```

    < inscriptions
      <> _form.html.erb
      <> edit.html.erb
      <> my_account.html.erb
      <> new.html.erb
      <> view_exercise.html.erb
      <> view_routine.html.erb
  
```

- Vistas únicas para entrenador: consta de vistas como mi cuenta donde ve sus usuarios asignados y vistas para crear y editar las rutinas.

```

    < trainer
      < rutines
        <> _form.html.erb
        <> edit.html.erb
        <> new.html.erb
      < trainers
        <> _search.html.erb
        <> my_account.html.erb
        <> user.html.erb
  
```

- Layouts: plantillas para todas las vistas de la aplicación. En este caso tenemos una para administradores y otra que comparten entrenadores usuarios y visitantes. Los layouts de email se crean por defecto y no los he utilizado.

```

    < layouts
      <> admin.html.erb
      <> application.html.erb
      <> mailer.html.erb
      <> mailer.text.erb
  
```

4.3 Capa de Negocio o Lógica de la Aplicación.

La capa de negocio o lógica de la aplicación es donde se gestionan las reglas de negocio y los procesos internos. Esta capa incluye:

- **Gestión de Usuarios:** Implementación de funcionalidades para registrar nuevos usuarios, gestionar sesiones de inicio y cierre de sesión, y asignar roles de usuario (clientes, entrenadores, administradores) por parte de los administradores.

- **Gestión de Entrenamientos:** Creación, asignación y seguimiento de rutinas de entrenamiento personalizadas. Los entrenadores pueden crear rutinas y asignarlas a los usuarios, y los usuarios pueden registrar su progreso.
- **Gestión Administrativa:** Funcionalidades para que los administradores gestionen ejercicios, asignen entrenadores a usuarios, y supervisen todas las actividades de la aplicación.
- **Validación y Seguridad:** Verificación de la entrada de datos, control de acceso basado en roles y protección contra vulnerabilidades comunes como inyecciones de código o SQL

4.4 Capa de Persistencia o Datos.

La capa de persistencia se encarga de almacenar y recuperar datos desde la base de datos.

La base de datos consta de 5 tablas: Usuarios(User), Inscripciones(Inscription), Ejercicios(Exercise), Rutinas(Routine) y Series(RoutineSet).

Como se puede ver en el diagrama del apartado [3.2 Diagrama de clases](#), las tablas constan de los siguientes campos:

- Usuarios(User)
 - id
 - name: String (nombre del usuario)
 - email: String (email del usuario)
 - is_admin: boolean (true si el usuario tiene el rol de Administrador)
 - is_trainer: boolean (true si el usuario tiene el rol de Entrenador)
 - trainer_id: foreign key (id del usuario con rol de entrenador que se le haya asignado al cliente)

No es necesario añadir un campo “contraseña” ni especificar que el email debe ser único ya que se eso se encarga la gema de Device
- Inscripciones(Inscription)
 - id
 - user_id: foreign key (id del usuario al que pertenece la inscripción)
 - goal: String (texto con el objetivo principal del usuario, tendrá 3 opciones para escoger)
 - comment: text (comentario del usuario acerca de sus objetivos o lo que considere relevante)
 - age: integer (edad del usuario)
 - height: float (altura del usuario)
 - weight: float (peso del usuario)
 - weight_goal: float (peso que usuario quiere alcanzar)
 - body_fat: float (porcentaje de grasa corporal del usuario)
 - body_fat_goal: float (porcentaje de grasa corporal que usuario quiere alcanzar)
- Ejercicios(Exercise)
 - id

- name: String (nombre del ejercicio)
- muscle_group: String (grupo muscular que trabaja el ejercicio, en esta versión se puede seleccionar entre 4 opciones)
- description: text (Texto que explica información sobre el ejercicio y como realizarlo correctamente)
- Rutinas(Routine)
 - id
 - name: String (nombre de la rutina)
 - trainer_id: foreign key (id del usuario con rol de entrenador que crea la rutina)
 - user_id: foreign key (id del usuario recibirá la rutina)
 - day: integer (dia de la semana que corresponde al entrenamiento. Ej: en una semana pueden realizarse cuatro entrenamientos, entonces los días serán 1, 2, 3 y 4)
 - week: integer(semana a la que pertenece la rutina)
 - nutrition: String (recomendación nutricional para este día de entrenamiento, en esta versión se puede seleccionar entre 4 opciones)
- Series(RoutineSet)
 - id
 - exercise_id: foreign key (id del ejercicio que se realiza en esta serie de la rutina)
 - reps: integer (número de repeticiones que se realizan del ejercicio)
 - weight: integer (peso con el que se realiza el ejercicio)
 - is_done: boolean (es true si el usuario ha marcado la serie como completada)

5. IMPLEMENTACIÓN

En este apartado se detallan las tecnologías utilizadas.

5.1 Tecnologías utilizadas en el desarrollo del proyecto

Para el desarrollo de la aplicación LifeLift se han utilizado diversas tecnologías que permiten una implementación eficiente y escalable. A continuación, se detallan las principales tecnologías y herramientas empleadas:

- **Ruby on Rails:** Framework principal para el desarrollo de la aplicación, utilizado por su facilidad de uso, rapidez en el desarrollo y fuerte enfoque en la convención sobre la configuración lo que facilita mucho el desarrollo de proyectos sencillos.
- **SQLite:** Sistema de gestión de bases de datos relacional, elegido por su sencillez y porque es el que Rails utiliza por defecto.
- **HTML5, CSS3:** Tecnologías fundamentales para el desarrollo de la interfaz de usuario.
- **Bootstrap:** Framework CSS para el diseño responsivo y atractivo de la interfaz de usuario.

A continuación se detallan las gemas de Rails utilizadas:

1. **sassc-rails:** Esta gema integra la implementación en C de Sass, llamada LibSass, en el pipeline de activos de Rails. Mejora significativamente la velocidad de compilación de archivos SCSS.
2. **bootstrap:** Bootstrap es un popular framework CSS que te permite crear interfaces web responsivas y con un diseño atractivo. Con esta gema, puedes usar Bootstrap en tu aplicación Rails para estilizar tus componentes y páginas.
3. **ransack:** Ransack es una gema que facilita la búsqueda y filtrado de registros en tu aplicación. Proporciona una interfaz sencilla para crear consultas de búsqueda complejas y personalizables. Ideal para implementar formularios de búsqueda avanzada.
4. **kaminari:** Kaminari es un paginador sofisticado y personalizable para aplicaciones web en Ruby. Te permite dividir grandes conjuntos de datos en páginas más pequeñas para una mejor experiencia de usuario. Funciona con varios ORMs y frameworks web.
5. **devise:** Devise es una solución flexible de autenticación para Rails basada en Warden. Proporciona características como inicio de sesión, registro, recuperación de contraseña y más. Es ampliamente utilizado para gestionar la autenticación de usuarios en aplicaciones Rails.
6. **brakeman:** Brakeman es una herramienta de seguridad estática que analiza tu código en busca de posibles problemas de seguridad. Detecta vulnerabilidades y te ayuda a proteger tu aplicación contra ataques.

7. **erb_lint:** Esta gema verifica la sintaxis y el formato de las plantillas ERB (Embedded Ruby) en tus vistas. Ayuda a mantener un código limpio y legible.
8. **standard y standard-rails:** Estas gemas aplican reglas de estilo consistentes al código Ruby. Pueden ayudarte a mantener un código limpio y fácil de leer, siguiendo las mejores prácticas.
9. **error_highlight:** Proporciona una forma más visual y atractiva de mostrar errores en tu aplicación. Mejora la experiencia del usuario al resaltar los problemas de manera efectiva.
10. **rails-i18n:** Esta gema se utiliza para gestionar las traducciones en tu aplicación Rails. Facilita la internacionalización y localización de tu sitio web.

5.2 Descripción del Proyecto

La implementación del proyecto se ha dividido en tres capas principales: presentación, lógica de la aplicación y persistencia de datos. Cada capa se detalla a continuación. Sigue una arquitectura en capas que asegura modularidad, facilidad de mantenimiento y escalabilidad.

5.2.1 Capa de Presentación.

- **Ruby on Rails**
- **HTML5 y CSS3:** Para estructurar y estilizar las páginas web.
- **Bootstrap:** Para facilitar el diseño responsive, asegurando que la aplicación sea accesible desde cualquier dispositivo, ya sea un ordenador, una tableta o un teléfono móvil.
- **JavaScript:** Para añadir interactividad y mejorar la funcionalidad del lado del cliente.

5.2.2 Capa de Negocio o Lógica de la Aplicación.

La capa de negocio es donde reside la lógica de la aplicación. Aquí se implementan las reglas de negocio y la funcionalidad principal de la aplicación:

- **Gestión de Usuarios:** Utilizando Devise para la autenticación, se gestionan las sesiones de usuario, incluyendo el registro, inicio de sesión y la asignación de roles por parte del administrador.
- **Gestión de Entrenamientos:** Los entrenadores pueden crear y asignar rutinas de entrenamiento a los clientes. Los clientes pueden visualizar estas rutinas y registrar su progreso.
- **Gestión Administrativa:** Los administradores pueden crear y gestionar ejercicios, asignar entrenadores a clientes y supervisar todas las funcionalidades de la aplicación.
- **Validación y Seguridad:** Implementación de validaciones de datos y control de acceso para asegurar que solo los usuarios autorizados puedan realizar ciertas acciones.

5.2.3 Capa de Persistencia o de Datos.

La capa de persistencia se encarga de almacenar los datos de la aplicación de manera segura y eficiente:

- **SQLite:** Base de datos relacional utilizada para almacenar toda la información relacionada con usuarios, entrenadores, rutinas de entrenamiento, ejercicios y seguimiento del progreso.
- **Migraciones de Rails:** Utilizadas para gestionar y versionar los cambios en la estructura de la base de datos de manera controlada.
- **Fixtures de rails:** Utilizadas para añadir datos por defecto a la base de datos que se crean cada vez que se lanza la misma, de este modo se pueden hacer pruebas sin tener que introducir una y otra vez los mismos datos.
- **Seguridad de Datos:** Implementación de técnicas de cifrado para contraseñas y protección de datos sensibles.

6. EVALUACIÓN

6.1 Introducción.

En este apartado se comprueba que la aplicación cumple con los requisitos establecidos de cumplimiento de estándares web y que funciona de manera óptima acorde a los mismos.

6.2 Validaciones de páginas de Estilo.

La aplicación está maquetada en su mayoría con Bootstrap 5, cuyas hojas de estilo no es necesario validar ya que se supone que Bootstrap cumple con todas las normativas y recomendaciones de estilos.

Además de Bootstrap también he usado una hoja de estilos propia que será la que se validará. Esta se encuentra en app/assets/stylesheets.

Resultados del Validador CSS del W3C para TextArea (CSS versión 3 + SVG)

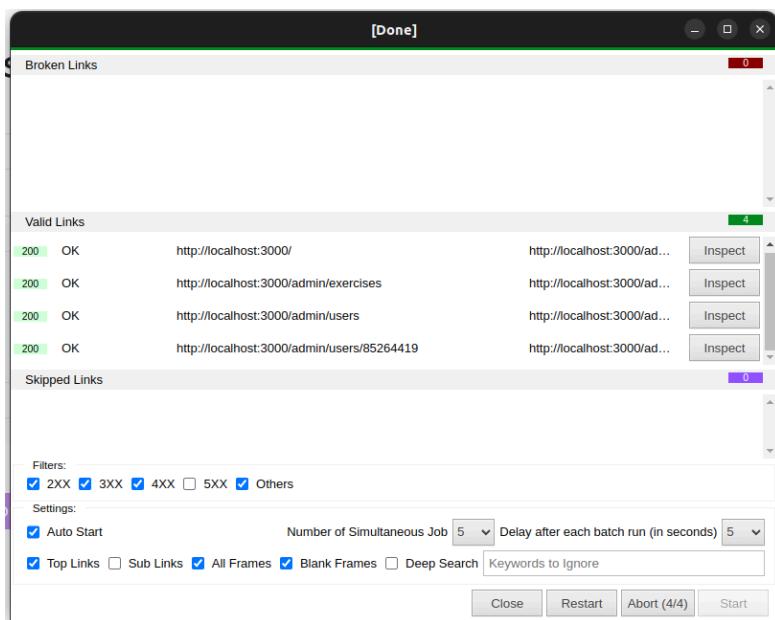
¡Enhorabuena! No error encontrado.

Este documento es [CSS versión 3 + SVG](#) válido!

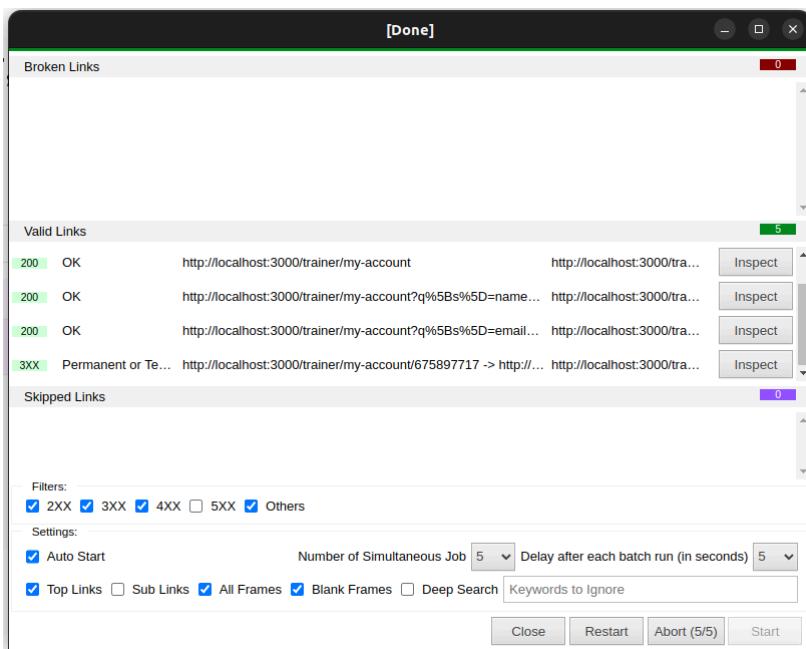
6.3 Validación de enlaces.

Para validar los enlaces he utilizado la extensión de chrome Broken link checker

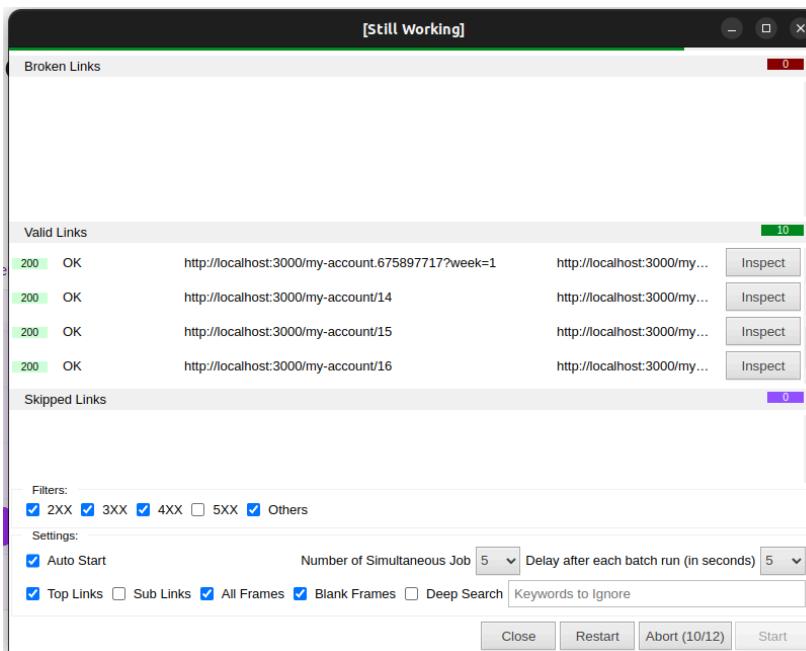
Enlaces en Admin:



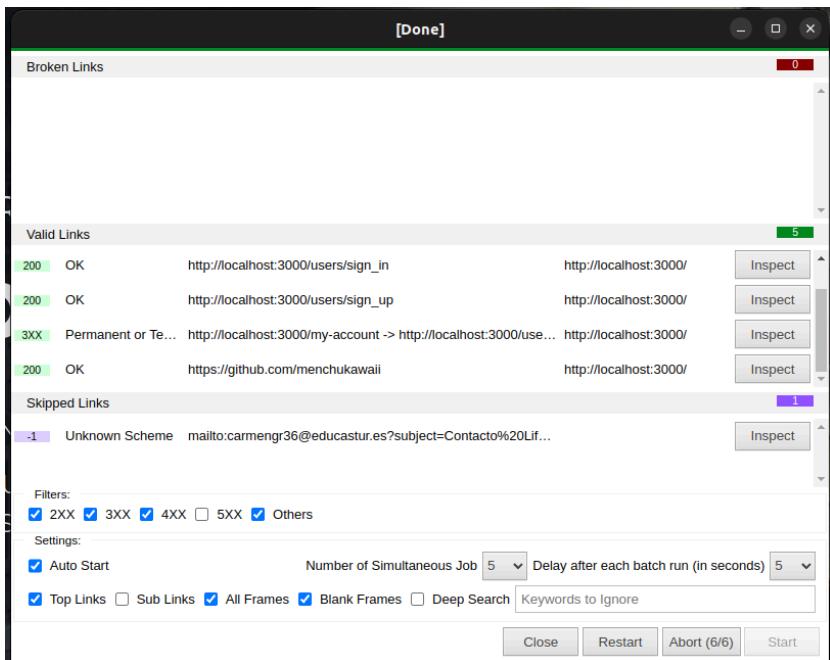
Enlaces Entrenador:



Enlaces Usuario:



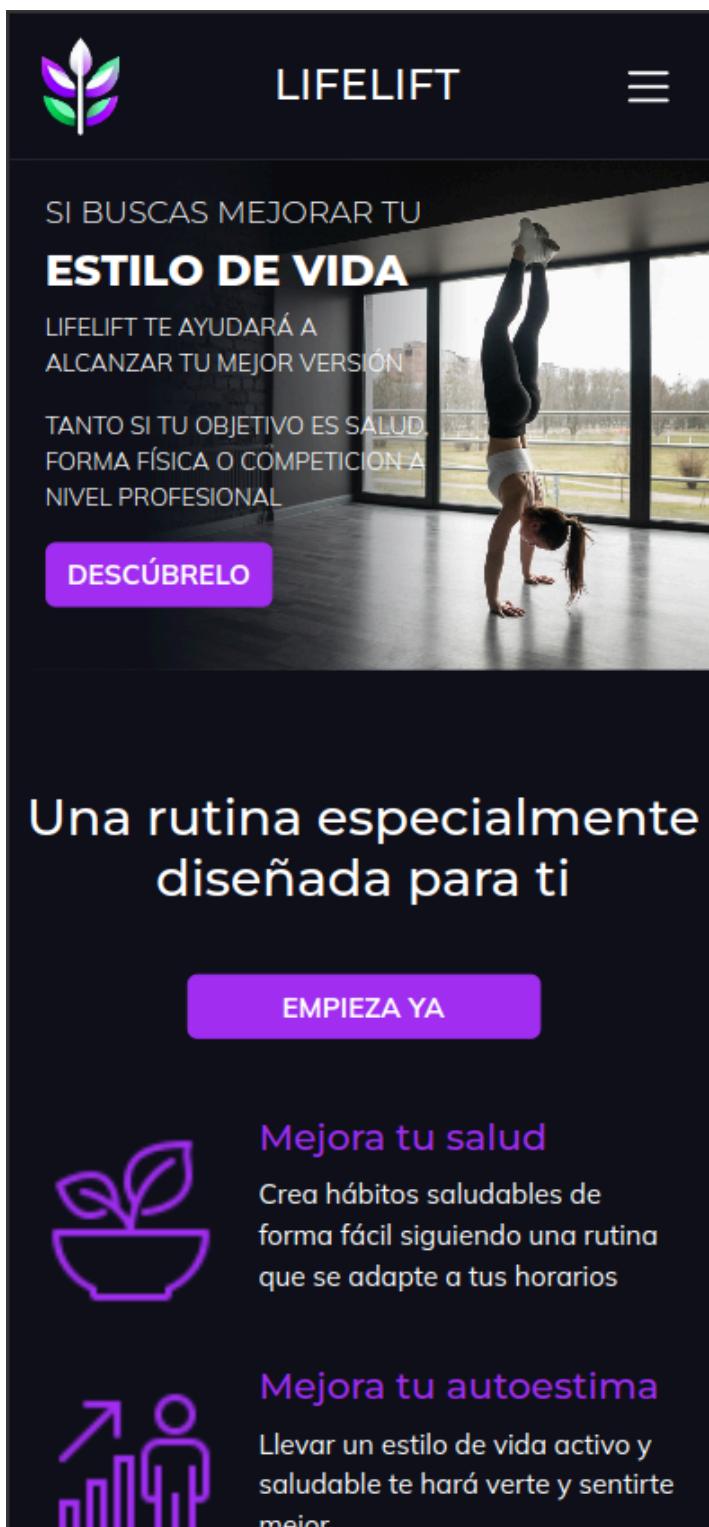
Enlaces Home:

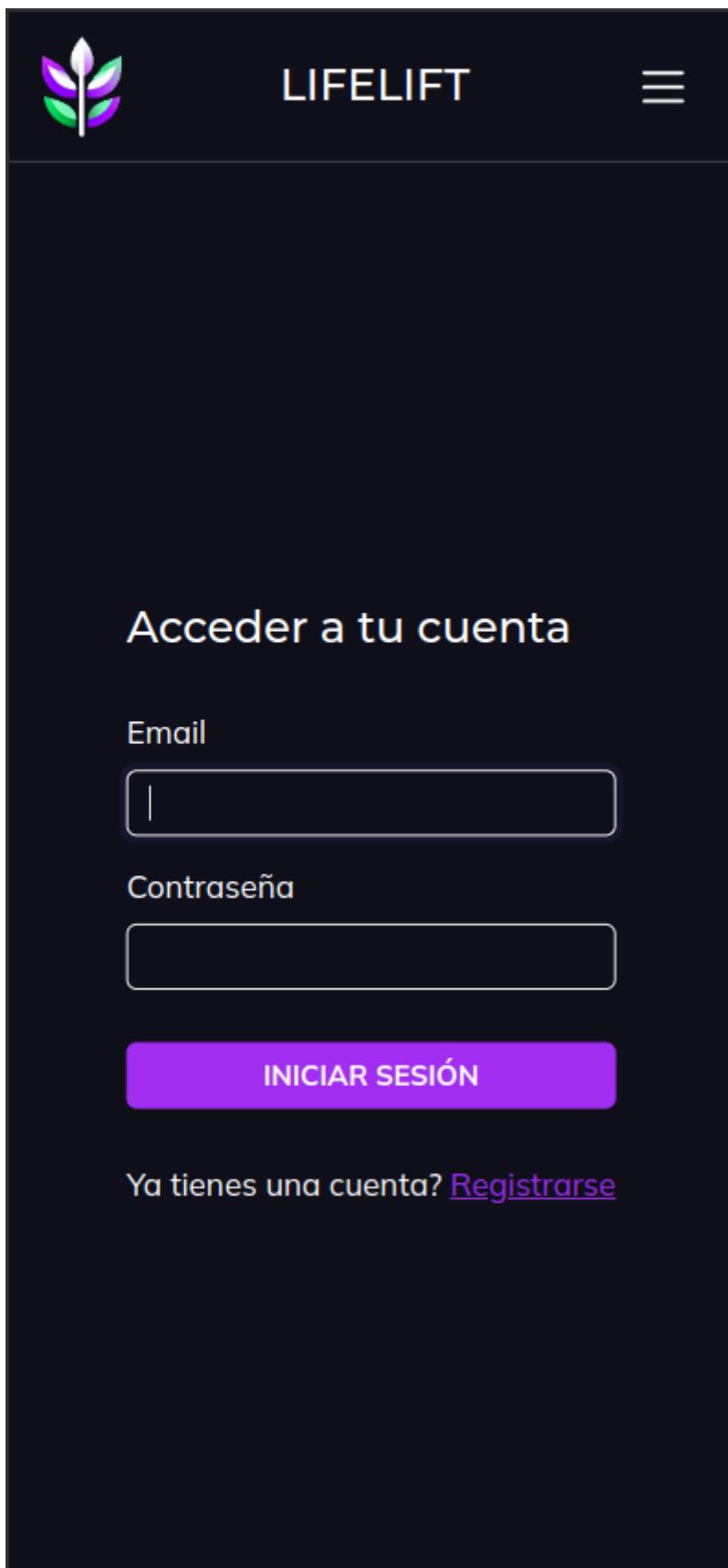


6.4 Validación de la Resolución.

La aplicación está maquetada de manera que se puede usar en cualquier dispositivo, sin importar su resolución. A continuación se muestran unas capturas de como se ve en un Iphone XR y en un Ipad PRO

Iphone XR







LIFELIFT



Hola Trainer, estás en el
Área de trabajo para
entrenadores

Mis Usuarios

Buscar por nombre:

 BUSCAR

Nombre Email

Carmen carmen@gmail.com

**VER
USUARIO**



LIFELIFT



Rutinas de Carmen: carmen@gmail.com

Rutinas

Semana 4	Semana 3	Semana 2			
Nombre	Rutina	Día	Semana	Nutrition	
Lunes Pecho y Hombro	1	4		Día alto en hidratos	<button>EDITAR</button>
Martes Circuito	2	4		Día alto en hidratos	<button>EDITAR</button>
Jueves Pierna y Gluteo	3	4		Día alto en hidratos	<button>EDITAR</button>
Viernes Espalda y Biceps	4	4		Día bajo en grasa	<button>EDITAR</button>
Sábado Abdominales	5	4		Día bajo en grasa	<button>EDITAR</button>

CREAR RUTINA



Área de entrenamiento de Carmen

[EDITAR INSCRIPCIÓN](#)

Rutinas

Nombre Rutina	Día	Semana	Nutrition
LUNES PECHO Y HOMBRO	1	4	Día alto en hidratos
MARTES CIRCUITO	2	4	Día alto en hidratos
JUEVES PIERNA Y GLUTEO	3	4	Día alto en hidratos
VIERNES ESPALDA Y BICEPS	4	4	Día bajo en grasa
SÁBADO ABDOMINALES	5	4	Día bajo en grasa



Rutina Lunes Pecho y Hombro

Series

Ejercicio	Repeticiones
	Flexiones 25
	Flexiones 25
	Press Banca 15
	Press Banca 12
	Press 8

Ipad PRO

Área de entrenamiento de Carmen

[EDITAR INSCRIPCIÓN](#)

Rutinas

Semana 4 Semana 3 Semana 2

Nombre Rutina	Día	Semana	Nutrition
LUNES PECHO Y HOMBRO	1	4	Día alto en hidratos
MARTES CIRCUITO	2	4	Día alto en hidratos
JUEVES PIERNAS Y GLUTEOS	3	4	Día alto en hidratos
VIERNES ESPALDA Y BICEPS	4	4	Día bajo en grasa
SÁBADO ABDOMINALES	5	4	Día bajo en grasa

Nutrición

Dieta Baja en Hidratos de Carbono

- Consumo de Proteínas de Calidad:** Incluye carnes magras, pescado, huevos, y productos lácteos bajos en grasa. Las proteínas ayudan a mantener la masa muscular y proporcionan saciedad.
- Verduras No Almidonadas:** Enfócate en vegetales como espinacas, brócoli, coliflor, y pimientos. Son bajas en carbohidratos y ricas en nutrientes esenciales.
- Grasas Saludables:** Incorpora aguacates, nueces, semillas, y aceites saludables como el aceite

Dieta Alta en Hidratos de Carbono

- Carbohidratos Complejos:** Prioriza fuentes de carbohidratos complejos como cereales integrales (arroz integral, quinoa, avena), legumbres (lentejas, garbanzos) y tubérculos (patatas, batatas). Estos alimentos liberan energía de forma sostenida.
- Frutas y Verduras:** Consumir una amplia variedad de frutas y verduras frescas. Son ricas en vitaminas, minerales y fibra, que apoyan la digestión y la salud general.

Dieta Baja en Grasas

- Proteínas Magras:** Incluye fuentes de proteínas bajas en grasa como pollo sin piel, pavo, pescado blanco, y productos lácteos bajos en grasa o sin grasa.
- Carbohidratos Complejos y Fibra:** Enfócate en carbohidratos integrales y ricos en fibra, como granos enteros (arroz integral, avena), legumbres, frutas y verduras. Estos alimentos ayudan a mantener niveles estables de energía y promueven la saciedad.
- Limitación de Grasas Saturadas y Trans:** Evita alimentos altos en grasas saturadas y trans, como



Rutina Lunes Pecho y Hombro

Series

Ejercicio		Repeticiones	Peso	Completado
	Flexiones	25	0 Kg	
	Flexiones	25	0 Kg	
	Press Banca	15	30 Kg	
	Press Banca	12	40 Kg	COMPLETAR
	Press Banca	8	50 Kg	COMPLETAR
	Cruces en polea	20	10 Kg	COMPLETAR
	Cruces en polea	20	10 Kg	COMPLETAR

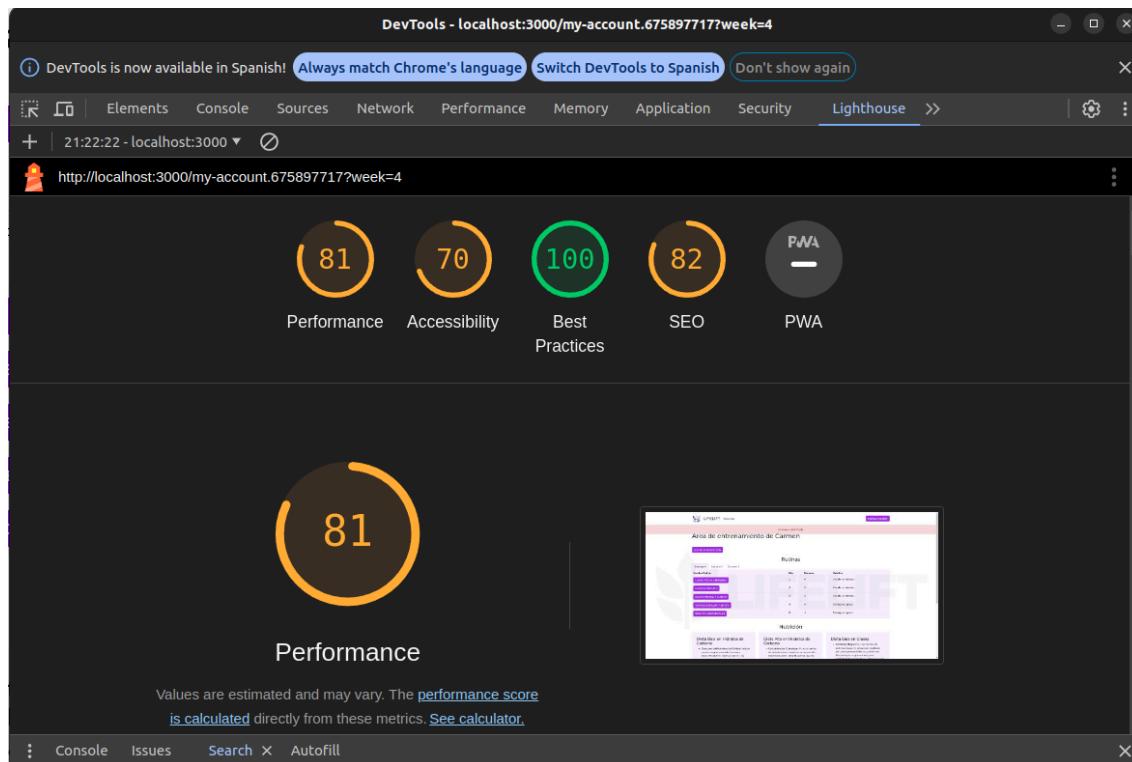
[VOLVER A MI CLASE](#)

6.5 Validación de Navegadores.

La aplicación se ve de la misma manera en los navegadores más usados: chrome, firefox, edge, opera y safari. Al usar Bootstrap nos aseguramos de que es compatible con todos los navegadores, ya que este es uno de sus principios.

6.6 Otras validaciones.

Como validación extra he añadido la herramienta de lighthouse chrome que analiza distintos aspectos de una web, como rendimiento, accesibilidad, buenas prácticas y SEO.



7. CONCLUSIÓN

7.1 Valoración Personal del Trabajo Realizado (análisis DAFO + análisis CAME).

Durante la planificación y ejecución de la aplicación web, puede identificar una serie de factores clave que influyen en el éxito del proyecto, los cuales se han analizado utilizando las metodologías DAFO y CAME.

7.1.1 Análisis DAFO



7.1.1 Análisis CAME



7.2 Posibles ampliaciones.

Ya que este ha sido un proyecto durante el cual he tenido que aprender mucho, tanto de las herramientas utilizadas como de desarrollo web en sí, hay muchas posibles mejoras.

La primera sería añadir muchísimos más ejercicios a la base de datos, ya que para ahorrar tiempo por ahora solo he creado 12 ejercicios diferentes, como si se tratase de una versión demo, ya que para demostrar el funcionamiento de la aplicación es más que suficiente.

Otra mejora que haré seguro en un futuro es añadir un avatar o foto de perfil a los usuarios, que ayudará a identificarlos mucho mejor.

Como última mejora que me he planteado por el momento, me gustaría añadir un apartado de red social, donde pudieras compartir los resultados de tus entrenamientos, gráficos de progreso o fotos con los demás usuarios de la aplicación. Esto me parece una mejora sustancial que aumentaría notablemente la usabilidad de la aplicación pero que por el momento me parece demasiado compleja para el tiempo del que se disponía y mis conocimientos actuales.

8. INSTRUCCIONES DE DESPLIEGUE

Para desplegar el proyecto necesitamos instalar Ruby y Ruby on Rails. Recomiendo instalarlo en Linux ya que está diseñado para este sistema operativo y es más sencillo de instalar, aunque funciona también en Windows y Mac.

Lo más sencillo es instalar RVM , un administrador de versiones de Ruby que nos permitirá tener la versión que necesitemos para cada proyecto.

En esta página muestran paso por paso como instalar Ruby on Rails y Ruby mediante rvm.

[Instalar RVM](#)

Una vez instalado RVM abrimos en una terminal la carpeta del proyecto y ejecutamos el comando: "rvm use". Debería darnos este resultado:

```
carmen@carmen-portatil:~/Documentos/ejerciciosRuby/Webs/lifelift$ rvm use  
Using /home/carmen/.rvm/gems/ruby-3.2.3
```

Si nos devuelve otra versión de Ruby debemos ejecutar rvm use ruby-3.2.3 para seleccionar la versión que queremos.

Ahora debemos crear la base de datos con los siguientes comandos:

- rails db:create
- rails db:migrate
- rails db:seed

Con esto ya podríamos lanzar el comando para lanzar el servidor: rails s y se desplegará la aplicación.

9. BIBLIOGRAFÍA

El sector del fitness en España - Datos estadísticos. Statista. [URL](#)

Las apps de fitness generarán 97 millones de euros en España este año. CMD Sport. [URL](#)

Pirámide de población Asturias. DatosMacro. [URL](#)

APPS google play:

- [Hevy: Gym Workout Tracker](#)
- [Entrenamiento en casa: Fitness. Leap Fitness Club](#)
- [Fitness & Bodybuilding. VGFIT LLC](#)
- [Fitness Ejercicios en casa. Fitness Online LLC](#)
- [FitOn Entrenamientos y Fitness](#)

Plataformas web:

- INDYA. Nutricionistas Deportivos Online. [URL](#)
- The Sayan Kiwi. [URL](#)
- Fuertafit. Entrenamiento online con planes calendarizados. [URL](#)

Tecnologías

- [Ruby](#)
- [Rails](#)
- [Hotwired](#)

Bootstrap

- Bootstrap Ruby Gem - GitHub. [URL](#)
- Bootstrap docs - Bootstrap [URL](#)

Rails

- Ruby Version Manager - RVM [URL](#)
- Rails Guides
 - The Asset Pipeline - Rails Guides. [URL](#)
 - Rails Routing from the Outside In - Rails Guides. [URL](#)
 - Action View Form Helpers - Rails Guides [URL](#)
 - Active Record Associations - Rails Guides [URL](#)
 - Form Helpers - Rails Guides [URL](#)
 - Rails Routing from the Outside In - Rails Guides [URL](#)
 - Active Record Validations - Rails Guides [URL](#)
 - Rails Internationalization (I18n) API - Rails Guides [URL](#)
 - Rails Routing from the Outside In - Rails Guides [URL](#)
 - Testing Rails Applications - Rails Guides [URL](#)
- [Apydock](#)
 - belongs_to - Apydock [URL](#)

- collection_select - Apydock [URL](#)

- Devise Gem - GitHub [URL](#)
- Traducciones I18n devise.es.yml [URL](#)
- Include blank for first item in select list in options_for_select tag - StackOverflow [URL](#)
- Ransack - GitHub [URL](#)
- Ransack Gem - Search Matchers - Ransack [URL Example](#)
- Kaminari - GitHub [URL](#)