

Pontificia Universidad Católica del Ecuador

Facultad De Ingeniería

Escuela de Sistemas



TEMA:

DISEÑO Y DESARROLLO DE UN PROTOTIPO FUNCIONAL WEB PARA LA
GESTIÓN DE USUARIOS DE UN GIMNASIO.

AUTOR:

MATEO ALEJANDRO LINCANGO DELGADO

TRABAJO PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SISTEMAS Y
COMPUTACIÓN

QUITO, 2023

DEDICATORIA

Primeramente, quiero dedicar este trabajo de tesis a mis padres, Alejandro y Editha ya que ellos fueron una parte fundamental en mi vida, dándome la oportunidad de estudiar en instituciones privadas desde que tengo memoria. Ellos han sido una guía en mi crecimiento diario. También quiero agradecer a mis abuelos, Rosa y Alejandro a los cuales los considero como mis segundos padres. Ellos fueron el pilar para ser la persona que soy hoy apoyándome desde que era un niño. A mis amigos de carrera que me han apoyado a lo largo de mi vida universitaria, explicándome un tema que no entiendo o guiándome para obtener la solución a algún problema.

A mi hermana Valentina por haber estado presente, dándome ánimos o compartiendo experiencias. Aunque a veces nos peleemos espero siempre contar con tu apoyo.

Finalmente, a mis profesor, quienes me han brindado la guía necesaria para aplicarla en mi futura cara profesional.

AGRADECIMIENTO

Agradezco a mis padres por haberme guiado en todo mi vida escolar y haberme brindado su apoyo. A mi familia por siempre estar conmigo en las buenas y malas. A mi primo Sebastián que me prestaba su computadora para programar en tiempos de pandemia. A mi abuelo Alejandro, que ya no está, por haberme brindado tantas enseñanzas desde que era solo un niño. A mi abuela Rosa por siempre estar pendiente de mí. Agradezco a mis compañeros por aclararme dudas en ciertos temas del desarrollo de este sistema.

RESUMEN

Este trabajo tiene como objetivo desarrollar un sistema web para la gestión de clientes de un gimnasio, con el fin de agilizar los procesos dentro del gimnasio.

A través de entrevistas con Eduardo León se detectó la necesidad de un sistema para automatizar los procesos tales como el registro de un cliente, la asignación de las rutinas a los clientes, en caso de que el usuario desee brindarle un plan alimenticio.

Además, se implementó SCRUM para el desarrollo de la aplicación ya que se tuvo que cumplir con un cronograma de 4 meses para la planificación, desarrollo y pruebas del sistema. Gracias a seguir la planificación se logró cumplir con las fechas acordadas desde un inicio.

ÍNDICE

DEDICATORIA	I
AGRADECIMIENTO	I
RESUMEN	I
ÍNDICE DE FIGURAS, GRÁFICOS Y TABLAS	III
CAPÍTULO I: INTRODUCCIÓN	1
1. MARCO DE REFERENCIA	1
1.1. JUSTIFICACIÓN.....	1
1.2. Planteamiento del problema	2
1.3. Objetivo General.....	3
1.4. Objetivos Específicos.....	3
1.5. Antecedentes.....	3
1.6. Alcance.....	4
CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA.....	5
2. Marco Teórico.....	5
2.1. Generalidades	5
2.2. Framework.....	6
2.3. JavaScript.....	6
2.4. Asp.Net Core	6
2.5. PostgreSQL.....	7
2.6. Programación Orientada a objetos.....	7
2.7. Metodología Informática	7
2.7.2. Artefactos de SCRUM	8
2.8. Fases de Scrum.....	10
CAPÍTULO III: METODOLOGÍA	12
3. Metodología de desarrollo del plan de tesis.....	12
3.1. Investigación Cualitativa	12
3.2. Investigación Aplicativa.....	13
3.3. Metodología de desarrollo de software	13
3.4. Modelo que aplica.....	14
CAPÍTULO IV: DESARROLLO DE LA INVESTIGACIÓN.....	16
4. Desarrollo de la Metodología Scrum	16
4.1. Asignación de Roles	16

4.2.	Historias de Usuario.....	17
4.3.	Requerimientos Funcionales.....	22
4.4.	Requerimientos No Funcionales	22
4.5.	Casos de uso de la aplicación	23
4.6.	Product Backlog.....	39
4.7.	Arquitectura de software	41
4.8.	Épicas de usuario	43
4.9.	Planificación de Sprints.....	46
4.10.	Base de datos	50
CAPÍTULO V: IMPLEMENTACIÓN.....		53
5.	Implementación de la aplicación.....	53
5.1.	Estándares de codificación	53
5.2.	Implementación de la tesis.....	54
5.3.	Interfaces Gráficas.....	61
5.4.	Pruebas funcionales.	105
5.5.	Resultados.....	116
CONCLUSIONES Y RECOMENDACIONES		116
Bibliografía		119
ANEXOS		1211

ÍNDICE DE FIGURAS, GRÁFICOS Y TABLAS

ÍNDICE DE FIGURAS

Figura 1.	Figura de Caso de uso de Cliente.....	24
Figura 2.	Figura de Caso de uso de Entrenador	25
Figura 3.	Figura de Caso de uso de Administrador	26
Figura 4.	Figura de Caso de uso iniciar sesión y registrarse.....	26
Figura 5.	Figura de Caso de uso: Administrar Usuarios.....	27
Figura 6.	Figura de Caso de uso: Administrar Roles y permisos.....	30
Figura 7.	Figura de Caso de uso: Administrar rutinas.....	34
Figura 8.	Figura de Caso de uso: Administrar planes alimenticios.....	37
Figura 9.	Figura de Arquitectura de software	43
Figura 10.	Figura de Modelo Conceptual.....	50
Figura 11.	Figura de Modelo Lógico.....	51
Figura 12.	Figura de Modelo Físico	51
Figura 13.	Figura de controlador de Cliente.....	54
Figura 14.	Figura de Interfaz de Cliente.....	54
Figura 15.	Figura Organización de la solución.....	55
Figura 16.	Figura Organización de la solución.....	56
Figura 17.	Interfaz de inicio de sesión	61
Figura 18.	Interfaz Home	62
Figura 19.	Interfaz de menú de Cliente.....	63
Figura 20.	Interfaz de Menú Entrenador	64
Figura 21.	Interfaz de Plan Alimenticio Asignado	65
Figura 22.	Interfaz de Rutina Asignada	66
Figura 23.	Interfaz de listado de Roles	67
Figura 24.	Interfaz de crear un nuevo Rol.....	68
Figura 25.	Interfaz de detalle de Rol.....	69
Figura 26.	Interfaz de eliminar Rol.....	70
Figura 27.	Interfaz editar rol.....	71
Figura 28.	Interfaz de lista de usuarios	72
Figura 29.	Interfaz crear nuevo usuario	73
Figura 30.	Interfaz de editar usuario seleccionado.....	74
Figura 31.	Interfaz de detalle de usuario seleccionado	75
Figura 32.	Interfaz de eliminar usuario seleccionado	76

Figura 33.	Interfaz de Lista de membresías.....	77
Figura 34.	Interfaz de crear membresía.....	78
Figura 35.	Interfaz de editar membresía seleccionada	78
Figura 36.	Interfaz de Detalle membresia	79
Figura 37.	Interfaz de lista de alimentos	81
Figura 38.	Interfaz de crear un nuevo alimento.....	82
Figura 39.	Interfaz de editar alimento	83
Figura 40.	Interfaz de detalle de alimento.....	84
Figura 41.	Interfaz de eliminar alimento.....	85
Figura 42.	Interfaz de lista de Clientes.....	86
Figura 43.	Interfaz de crear un nuevo cliente.....	87
Figura 44.	Interfaz de editar cliente.....	88
Figura 45.	Interfaz de detalle de cliente	89
Figura 46.	Interfaz de eliminar cliente	90
Figura 47.	Interfaz de lista de ejercicios.....	91
Figura 48.	Interfaz de crear un nuevo ejercicio	92
Figura 49.	Interfaz de editar ejercicio.....	93
Figura 50.	Interfaz de detalle de ejercicio	94
Figura 51.	Interfaz de eliminar ejercicio	95
Figura 52.	Interfaz de lista de planes alimenticios.....	96
Figura 53.	Interfaz de crear un nuevo plan alimenticio.....	97
Figura 54.	Interfaz de editar plan alimenticio	98
Figura 55.	Interfaz de detalle de plan alimenticio	99
Figura 56.	Interfaz de eliminar plan alimenticio	100
Figura 57.	Interfaz de lista de rutinas.....	101
Figura 58.	Interfaz de crear una nueva rutina	102
Figura 59.	Interfaz de editar rutina.....	103
Figura 60.	Interfaz de detalle de rutina	104
Figura 61.	Interfaz de eliminar plan alimenticio	105

ÍNDICE DE TABLAS

Tabla 1.....	Tabla de Roles
.....	16

Tabla 2.....	Tabla Historia de inicio de sesión	17
Tabla 3.....	Tabla Historia de Visualización de Rutinas	18
Tabla 4.....	Tabla Historia de Acceso a Plan Alimenticio	18
Tabla 5.....	Tabla Historia de Actualización de Perfil	19
Tabla 6.....	Tabla Historia de Recepción de Notificaciones	19
Tabla 7.....	Tabla Historia de Inicio de Sesión	20
Tabla 8.....	Tabla Historia de Asignación de Rutinas	20
Tabla 9.....	Tabla Historia de Asignación de Planes Alimenticios:	21
Tabla 10.....	Tabla Historia de Gestión de Clientes	21
Tabla 11.....	Tabla de Product Backlog	39
Tabla 12.....	Tabla de lista de prioridades	43
Tabla 13.....	Tabla de Épicas de usuario	44
Tabla 14.....	Tabla de planificación Sprint 1	46
Tabla 15.....	Tabla de Planificación Sprint 2	47
Tabla 16.....	Tabla de Planificación Sprint 3	47
Tabla 17.....	Tabla de Planificación Sprint 4	47
Tabla 18.....	Tabla de Planificación Sprint 5	48
Tabla 19.....	Tabla de Planificación Sprint 6	48
Tabla 20.....	Tabla de Planificación Sprint 7	48
Tabla 21.....	Tabla de Planificación Sprint 9	49

Tabla 22.	Tabla de Planificación Sprint 10	49
Tabla 23.	Tabla de Sprint	56
Tabla 24.	Revisión Sprint 1	57
Tabla 25.	Revisión Sprint 2	57
Tabla 26.	Revisión Sprint 3	57
Tabla 27.	Revisión Sprint 4	58
Tabla 28.	Revisión Sprint 5	58
Tabla 29.	Revisión Sprint 6	59
Tabla 30.	Revisión Sprint 7	59
Tabla 31.	Revisión Sprint 9	60
Tabla 32.	Revisión Sprint 10	60
Tabla 33.	Tabla de prueba de humo Sistema Web Edu's Gym	106

CAPÍTULO I: INTRODUCCIÓN

1. MARCO DE REFERENCIA

1.1. JUSTIFICACIÓN

En la actualidad, los gimnasios son destinos populares para personas de todas las edades que buscan mejorar su salud y bienestar físico. La creciente demanda de servicios de acondicionamiento físico enfatiza la importancia de cumplir con las expectativas del público. Por lo tanto, la implementación de un sistema web para el gimnasio es crucial. Este sistema no solo hará que la gestión y las operaciones internas sean más fáciles, sino que también mejorará la experiencia del usuario y satisfacer las crecientes necesidades de una población que busca un estilo de vida activo. El sistema web se erige como una herramienta clave para satisfacer las demandas de desde la administración de clientes hasta la personalización de rutinas.

Al implementar un sistema web en el gimnasio, se proporcionará al usuario la capacidad de recibir notificaciones anticipadas sobre la fecha de vencimiento de su pago mensual a través de alertas enviadas a su correo. Esta aplicación no solo facilitará la gestión de pagos, sino que también permitirá a los clientes acceder a información detallada sobre rutinas y dietas personalizadas, brindándoles las herramientas necesarias para alcanzar sus objetivos de fitness. Además, el sistema permitirá al dueño del gimnasio utilizar un registro detallado de los números reales de clientes para desarrollar estrategias efectivas, como promociones y descuentos, con el objetivo de atraer nuevos clientes y aumentar las ganancias del establecimiento. Los clientes también podrán gestionar sus horarios de manera eficiente, evitando la congestión en el gimnasio durante las horas pico.

Con información precisa sobre preferencias y hábitos, los entrenadores podrán personalizar las dietas y rutinas, brindando un servicio más adaptado a las metas individuales de cada cliente.

Contar con este sistema proporcionará al gimnasio la capacidad de medir y evaluar los resultados obtenidos. Identificará áreas de mejora y permitirá establecer un nuevo plan para corregir errores, atrayendo así a nuevos clientes y manteniendo la lealtad de los existentes. Además de los beneficios para los clientes, la implementación del sistema resultará en una reducción significativa de costos, disminuyendo el uso de documentos impresos y optimizando el tiempo empleado en tareas administrativas.

1.2. Planteamiento del problema

Al implementar un sistema web, va a permitir que el usuario pueda saber por medio de notificación previas cuando se culminará su fecha de pago mensual con una alerta que dará su correo, además esta aplicación le permitirá al cliente saber las rutinas y dietas que tenga que cumplir para lograr los resultados que el cliente desee. Al tener un registro de los números reales de clientes el dueño del gimnasio puede establecer estrategias para como promociones y descuentos para atraer nuevos clientes y que de esta manera genere un aumento en las ganancias del gimnasio. También el cliente podrá reservar un horario o reagendar para evitar que el gimnasio este muy lleno a ciertas horas.

Además, el entrenador podrá llevar los avances de los clientes sabiendo a mayor detalle sus preferencias y hábitos mejorando la experiencia del usuario estableciendo dietas y rutinas personalizadas para cada uno dependiendo de que resultado es el que busca.

Al contar con el sistema el gimnasio puede medir los resultados de este y ver en qué aspectos está fallando para establecer un nuevo plan para corregir los errores y así conseguir

nuevos clientes y mantener a los que ya se tiene. Algunas de las ventajas que se tiene al implementar este sistema es la reducción de costos y dinero reduciendo el número de documentos impresos y reduciendo el tiempo de tareas administrativas.

1.3. Objetivo General

Analizar, crear y desarrollar un sistema web para la gestión de usuario de un gimnasio que permita automatizar varios procedimientos de este.

1.4. Objetivos Especificos

- Determinar los procesos de control de usuarios que generan perdida de información, con la finalidad de abolirlos.
- Cambiar el funcionamiento de la gestión de usuarios a un sistema administrativo automatizado que da más comodidad al administrador del gimnasio y a los clientes.
- Presentar una aplicación que contenga interfaces interactivas y organizadas con el fin de que esta sea fácil de usar para los usuarios.

1.5. Antecedentes

En la actualidad la tecnología está al alcance de todos y esta se ha convertido en algo vital en nuestra rutina diaria. Los gimnasios locales por mucho tiempo han ofrecido el servicio de entrenamiento con pesas, sin embargo, son pocos los gimnasios que cuentan con un sistema para una mejor gestión de esto y es esta la razón por la que estos también deben acoplarse a este cambio.

El propósito del proyecto es mejorar el servicio que se brinda a los clientes ofreciéndoles una nueva tecnología por medio la cual ellos tendrán acceso a una página web que ayudara al entrenador a tener una mejor gestión de las rutinas y dietas de sus clientes, mejorando la experiencia tanto del cliente como del entrenador. Lo que se espera obtener al implementar este nuevo sistema en el gimnasio es un aumento en la eficiencia del entrenador reduciendo el tiempo

en que el planea las rutinas y dietas de los clientes, llevando un seguimiento personalizado de cada uno.

1.6. Alcance

En este trabajo, se desarrollará un sistema web que cuente con los módulos necesarios para la correcta gestión de clientes, otro modulo que ayude a los clientes a saber la rutina general que deben cumplir y el plan alimenticio al que se deben regir. También, el sistema contará con interfaz amigable con los usuarios basándose en la elección de colores Negro (Que significa elegancia), blanco (Que significa paz) y verde (Significa salud y energía.). También esta será muy interactiva para que los usuarios no se confundan al momento de navegar a través de ella.

CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA

2. Marco Teórico

2.1. Generalidades

En este proyecto vamos a trabajar con una metodología aplicada porque el conocimiento que se aplicara está enmarcado en la lógica del negocio y en el desarrollo Web

Se aplicarán técnicas que nos faciliten la identificación de los requerimientos funcionales para lo que será necesario diagramas de proceso y usaremos entrevistas a los involucrados

La lógica de negocio del Edu's Gym empieza por el registro de un nuevo cliente en la libreta del dueño. Ahí se cancela la primera mensualidad junto con la inscripción. Después el entrenador te pesa y mide para poder crear un plan que el cliente debe seguir. Este plan incluye las rutinas que el cliente realizara en cada sesión junto con el plan alimenticio que llevará ya sea para bajar de peso o aumentar musculatura. El entrenador llevará un seguimiento de cada cliente para ir actualizando sus rutinas y plan alimenticio. El dueño entrega un papel donde está el número de sesiones máximas que el cliente puede ir en un mes que son 26 sesiones, también incluye el nombre y la fecha de pago. Este papel se lo presenta al entrar del gimnasio para llevar un registro de que días viene cada cliente. Este proceso es uno de los que se quiere automatizar con este proyecto.

Para el desarrollo del sistema se usará la metodología ágil SCRUM ya que esta nos permite realizar el proyecto en un corto periodo de tiempo.

Para el modelo de datos usaremos un modelo entidad-relación que nos permite visualizar las relaciones que existen entre las entidades.

Para el desarrollo del backend usaremos ASP.net core donde crearemos los métodos para realizar el CRUD (Create, Read, Update , Delete) de nuestro sistema.

Además, utilizaremos lenguaje de marcado HTML para el desarrollo de la página web. Utilizaremos Bootstrap que nos facilitara el dar estilos a nuestra página web y que esta sea adaptable. El patrón de arquitectura que se utilizará es modelo-vista-controlador.

Las pruebas que se realizaran al sistema son: pruebas funcionales, pruebas de comunicaciones, pruebas de rendimiento, pruebas de facilidad de uso.

2.2. Framework

Son las herramientas utilizadas para el desarrollo de aplicaciones que nos facilitan el desarrollo de estas. Un framework es un conjunto de componentes reutilizables para construir una aplicación, este cuenta con extensiones, librerías y plugin que disminuyen la carga de trabajo del programador.

2.3. JavaScript

Es un lenguaje de programación interpretado, basado en prototipos que soporta programación orientada a objetos. Es utilizado para hacer páginas webs interactivas. Algunas de las funciones con las que cuenta JavaScript pueden mejorar la experiencia del usuario (anónimo, 2022).

2.4. Asp.Net Core

Es un marco teórico multiplataforma que tiene como fin compilar aplicaciones en internet. Es de código abierto y de alto rendimiento (Anderson, 2022). Es un framework orientado a aplicaciones web que fue desarrollado por Microsoft y su comunidad.

2.5. PostgreSQL

El sistema gestor de base de datos que utilizaremos será el de PostgreSQL ya que es código abierto lo que nos ayudara a no realizar gastos en este punto. Además, está diseñado para soportar sintaxis SQL. Es por este motivo que cuenta con características como control de concurrencia, funciones y consultas. (Herrera, 2016). Al ser un framework de código abierto existe una comunidad grande que nos permitirá a solventar dudas fácilmente a través de internet.

2.6. Programación Orientada a objetos

Es un paradigma de programación que nos ayuda a desarrollar aplicaciones complejas con un código claro y manejable ya que nos permite la reutilización de código.

2.7. Metodología Informática

Para realizar el proyecto de titulación se escogió la metodología ágil SCRUM la cual nos permitirá realizar el proyecto en un corto periodo de tiempo, también podemos separar el proyecto en bloque más pequeños que los llamaremos sprints, estos nos ayudan a que el proyecto sea más gestionable. Además, se logra la integración de todas las partes involucradas en el proyecto es decir que se mantiene contacto en todas las fases con el cliente (Feliciano, 2017).

2.7.1. Definición de SCRUM

Scrum es una metodología ágil de trabajo iterativo e incremental para la gestión de proyectos. Es usado en el desarrollo de software. SCRUM se centra en el software funcional y la flexibilidad para cambiar junto con los requisitos del negocio (anónimo, 2022).

2.7.2. Artefactos de SCRUM

- **Producto Backlog**

Producto backlog es una lista que contiene las nuevas funciones, las tareas, la corrección de errores y las mejoras para crear el producto. Estos datos se obtienen a partir de reuniones con el cliente, las demandas del mercado. Este artefacto se lo actualiza a medida que se obtiene nueva información.

- **Sprint Backlog**

Es un conjunto que tiene las tareas del producto backlog que se desarrollaran durante el siguiente sprint. Dichos sprint backlogs son creados con el fin de planificar las entregas en futuros Sprint. Para generar un sprint backlog se escoge una tarea del producto backlog y se lo divide en tareas más pequeñas. Los Sprint Backlogs se actualizan en la fase de planificación de sprints.

- **Incremento del producto**

Es la entrega del producto que se realizará el cliente y tiene lugar una vez completadas las tareas del backlog que fueron desarrolladas durante un sprint. Esto también va a incluir a las tareas desarrolladas en los anteriores sprints. Una vez concluido un sprint siempre hay un incremento el cual es decidido durante la fase de planificación de SCRUM. Los incrementos ocurren cuando el equipo decide mostrar un avance del proyecto al cliente.

- **Sprint**

Es un periodo de 1 a 4 semanas en la cual un equipo trabaja para completar las tareas definidas en la etapa de planificación de SCRUM. Los sprints ayudaran al equipo a realizar

un mejor software. En cada sprint de trabajo obtendremos un incremento del trabajo el cual fue definido previamente.

2.7.3. Roles de SCRUM

- **Equipo de desarrollo**

Normalmente está formado entre 3 a 9 profesionales los cuales son los encargados de desarrollar el software, los cuales deberán entregar un incremento del trabajo al final de cada ciclo de desarrollo. Ellos se encargan de cada incremento a partir de los elementos definidos en el producto backlog durante la fase de planeación.

- **SCRUM Master**

Es el líder del equipo. Su objetivo principal es que el equipo alcance los objetivos al final del proyecto definidos en la fase de planeación eliminando dificultades que puedan complicar cumplir la entrega del producto (Canal, 2022).

- **Product Owner**

Es la persona que se encarga de maximizar y optimizar el valor del producto. Es la persona intermediaria entre los involucrados y los sponsors del producto. Una de sus tareas es mantener el Product backlog bien estructurado. En cada sprint debe marcar cual es la meta acordada para que el producto incremente su valor. También, el Product Owner debe ser el encargado hacer conocer al equipo las demandas del cliente y de la retroalimentación hecha por los mismos.

2.8. Fases de Scrum

Scrum cuenta con 5 fases y cada una de estas busca satisfacer las necesidades planteadas por el SCRUM Master y al mismo tiempo debe cumplir los plazos de entrega del proyecto.

- **Inicio**

La primera fase es la fase donde vamos a analizar el proyecto, aquí identificamos cuales son las necesidades del sprint. En esta fase se responde preguntas de Que Como y cuando lo queremos.

Los primeros pasos que realizamos son:

1. Crear la visión del proyecto
2. Identificar al SCRUM Master y los involucrados
3. Formar los equipos de SCRUM
4. Crear las listas de requerimientos
5. Planificar el proyecto.

- **Planificación y estimación**

En esta fase se incluyen estos pasos:

1. Crear las historias de usuarios
2. Estimar e identificar las tareas
3. Crear el sprint backlog

Esta fase es clave ya que nos ayudara a establecer metas fijas y cumplir con las fechas de entrega de cada sprint. Es una de las tareas más importantes ya que el SCRUM Master debe delegar las tareas a cada grupo y estimar los tiempos de entrega.

- **Implementación**

Debemos cumplir con los siguientes procesos

1. Crear los entregables
2. Realizar una reunión semanal.
3. Refinanciamiento del backlog priorizado del producto.

- **Revisión y retrospectiva**

Una vez implementado lo planificado se debe hacer una revisión del proceso, lo que quiere decir una evaluación propia del equipo de trabajo.

- 2 Validar y demostrar el sprint

- 3 Revisión del sprint

- **Lanzamiento**

Esta es la última fase de la metodología ágil SCRUM. Este es el desenlace del proyecto donde entregamos el producto al cliente y se debe cumplir con 2 tareas.

1. Enviar los entregables
2. Retroalimentación del proyecto.

3. Metodología de desarrollo del plan de tesis

3.1. Investigación Cualitativa

Este análisis se realizará empleando un enfoque de investigación aplicada, dado que su finalidad es utilizar los conocimientos adquiridos a lo largo de la carrera para abordar una problemática concreta en el gimnasio objetivo.

La investigación cualitativa consiste en comprender el comportamiento de un grupo de personas en base a sus experiencias, por lo tanto, en este proyecto realizaremos un estudio de las personas relacionadas con el gimnasio llamado Edu's Gym. Se espera obtener información a base de experiencias de los clientes de dicho gimnasio.

La investigación cualitativa se enfoca en el análisis detallado, descripción e interpretación de fenómenos sociales complejos a través de la recopilación y evaluación de datos que no involucran cifras numéricas. Su propósito es explorar y profundizar en la comprensión de estos fenómenos mediante enfoques cualitativos y tácticas como entrevistas, observación participante y revisión de contenidos. De esta manera, se facilita el análisis de los datos recolectados utilizando métodos como el análisis temático, narrativo y del discurso, junto con otras estrategias cualitativas, para identificar patrones y temas emergentes.

La metodología cualitativa que seleccionamos es la metodología tradicional ya que necesitamos de las personas que asisten al gimnasio para resolver un problema, es decir que con la ayuda de ellos podremos mejorar y facilitar las tareas principales para un mejor funcionamiento de Edu's Gym.

3.2. Investigación Aplicativa

Según Ortega(2022) la investigación aplicada busca utilizar los conocimientos y hallazgos adquiridos para abordar problemas o situaciones específicas en la práctica. En lugar de centrarse en la teoría o la generación de nuevos conceptos, la investigación aplicada se orienta hacia la resolución de problemas prácticos y la implementación de soluciones en el mundo real. Este enfoque busca aplicar los resultados de la investigación para mejorar situaciones, tomar decisiones informadas y generar un impacto práctico en diversos campos. La utilización de métodos y procedimientos de ingeniería se vuelve esencial al considerar la implementación de metodologías de desarrollo, ya que estas brindan una guía fundamental para el proceso de desarrollo y garantizan la calidad del producto final en términos de seguridad, funcionalidad y coherencia. En la actualidad, al emprender un proyecto de software, se debe tomar la decisión entre optar por una metodología ágil o una tradicional, y en este caso, hemos optado por emplear una metodología ágil.

Después de concluir la investigación aplicada, se logrará disponer de un prototipo o herramienta práctica y funcional para la gestión del gimnasio, lo que puede tener un efecto positivo en la organización y administración de las principales actividades del gimnasio comunidades. Para lograr esto, es necesario aplicar los principios y metodologías de la investigación científica con el propósito de diseñar, desarrollar y poner a prueba un sistema web que permita una gestión eficaz y eficiente del gimnasio.

3.3. Metodología de desarrollo de software

Previamente a definir la metodología que vamos a utilizar en el desarrollo de este proyecto hay que dejar claro la definición de metodología de desarrollo de software.

La metodología de desarrollo de software son los métodos y técnicas utilizados para diseñar un software. Esta nos permitirá controlar el desarrollo del mismo anticipando los errores que pueden ocurrir, ahorrar tiempo y gestionar de una mejor manera los recursos de los que disponemos.

Existen varios tipos de metodología, pero en el cual nos vamos a basar es en la metodología ágil.

3.4. Modelo que aplica

3.4.1. Metodología ágil

En los últimos años, hemos observado un crecimiento en la adopción de esta metodología, lo cual ha sido impulsado por la necesidad de adaptar nuestras prácticas de trabajo y enfoque de desarrollo a las particularidades de cada proyecto. El objetivo primordial de esta tendencia es optimizar la productividad, permitiendo la gestión de proyectos de manera versátil y eficiente. Algunas de las ventajas más destacadas de esta metodología incluyen:

- Eleva la calidad del producto final.
- Fomenta la participación del cliente a lo largo del proceso de desarrollo, lo que se traduce en una mayor satisfacción por parte del cliente.
- Contribuye a una mejor organización y dinámica de trabajo en equipo.
- Facilita un mayor control sobre los aspectos temporales y financieros del proyecto.

He observado cómo esta metodología se ha vuelto esencial en la industria, marcando una diferencia significativa en la forma en que abordamos proyectos de desarrollo.

3.4.2. Metodología SCUM

Scrum, como metodología, representa un enfoque que destila valores esenciales como colaboración, transparencia y flexibilidad en el contexto de proyectos y desarrollo de software. Esta radica en la creencia de que, a través de la entrega por etapas y la retroalimentación constante, se logra la satisfacción del cliente. Este método inculca un alto sentido de responsabilidad individual en los miembros del equipo, promueve la comunicación efectiva y aboga por la inspección y adaptación en curso. Sin duda, Scrum emerge como una filosofía de trabajo que persigue la excelencia en calidad y eficiencia, respaldando una visión de mejora constante y entrega de productos innovadores que se ajustan a los cambios continuos del entorno del proyecto.

En base a lo investigado hemos elegido trabajar con SCRUM ya que tenemos a un experto en el área que tendrá el rol de Producto Owner que será Eduardo León mientras que los demás roles del equipo los cumplirá mi persona que son Scrum master, diseñador, evaluador y desarrollar.

Mantendremos reuniones con Eduardo León y sus clientes con el objetivo de realizar el backlog para cada Sprint.

3.4.3. Técnica de recolección de datos

Para recopilar datos, se empleará el enfoque de entrevista, que se considera la elección idónea para esta situación de investigación. Esta decisión se basa en la disponibilidad de un experto en el campo, como se mencionó anteriormente. Las entrevistas brindarán una oportunidad única para obtener información detallada y precisa sobre el objeto de estudio a través de la interacción directa con los participantes, siguiendo las pautas de Hernández Sampieri (2014). Es importante destacar que la entrevista es una técnica ampliamente empleada en

investigaciones cualitativas debido a su capacidad para explorar en profundidad temas complejos y adquirir una comprensión más profunda de la información.

CAPÍTULO IV: DESARROLLO DE LA INVESTIGACIÓN

4. Desarrollo de la Metodología Scrum

El objetivo principal de este capítulo reside en proporcionar una exposición minuciosa del proceso de desarrollo de la aplicación web, empleando la metodología Scrum como marco de trabajo. En las páginas que siguen, se detallarán las historias de usuario que guiarán el diseño y desarrollo, los casos de uso derivados de estas historias, así como la arquitectura del software. Además, se asignarán los roles y responsabilidades a los miembros del equipo de desarrollo. La finalidad última de este enfoque es dirigir la atención hacia cada tarea con miras a cumplir exitosamente con el Sprint y realizarlo en el plazo estipulado, asegurando que los entregables satisfagan las necesidades y expectativas del usuario.

4.1. Asignación de Roles

Tabla 1. Tabla de Roles

Rol	Responsable
Líder del proyecto:	Ing. Mora Edison

Product Owner:	León Eduardo
Scrum Master:	Lincango Mateo
Diseñador:	Lincango Mateo
Desarrollador:	Lincango Mateo
Tester:	Lincango Mateo

Nota. La tabla presentada proporciona una visión clara de los miembros del equipo y los roles que desempeñarán en el proyecto. Cada rol se asocia con diversas actividades y responsabilidades específicas, lo que garantiza una distribución eficiente de tareas. .La tabla de roles y responsabilidades del equipo se convierte en una herramienta esencial para la gestión y coordinación efectiva del proyecto, permitiendo un enfoque claro en la ejecución de tareas y el logro de los objetivos del proyecto.

4.2. Historias de Usuario

Las historias de usuario se obtuvieron a través de una reunión con el entrenador del gimnasio Edu's Gym. Además, se llevó a cabo una pequeña reunión con algunos de los clientes del gimnasio. Una vez finalizadas estas reuniones se identificaron los actores y procesos en los que se estableció el cronograma para realizar este proyecto.

4.2.1. Historias de usuario para el cliente

Tabla 2. Tabla Historia de inicio de sesión

ID	H001
Nombre	Historia de inicio de sesión
Prioridad	Alta

Riesgo	Baja
Descripción	Permitir a los cliente iniciar sesión para acceder a sus datos personalizados
Validación	Verificar que los clientes pueden iniciar sesión correctamente y acceder a sus paneles de control.

Tabla 3. Tabla Historia de Visualización de Rutinas

ID	H002
Nombre	Historia de Visualización de Rutinas
Prioridad	Media
Riesgo	Bajo
Descripción	Habilitar la visualización de rutinas asignadas para el día actual.
Validación	Se debe confirmar que los clientes pueden ver claramente sus rutinas diarias.

Tabla 4. Tabla Historia de Acceso a Plan Alimenticio

ID	H003
Nombre	Historia de Acceso a Plan Alimenticio
Prioridad	Media
Riesgo	Bajo
Descripción	Permitir a los clientes acceder a sus planes alimenticios personalizados.

Validación	Verificar que los clientes pueden acceder a la información de sus planes alimenticios.
-------------------	--

Tabla 5. Tabla Historia de Actualización de Perfil

ID	H004
Nombre	Historia de Actualización de Perfil
Prioridad	Media
Riesgo	Bajo
Descripción	Facilitar la actualización de la información del perfil del cliente.
Validación	Confirmar que los clientes pueden realizar cambios en su información de perfil.

Tabla 6. Tabla Historia de Recepción de Notificaciones

ID	H006
Nombre	Historia de Recepción de Notificaciones
Prioridad	Media
Riesgo	Bajo
Descripción	Enviar notificaciones por correo electrónico a los clientes sobre vencimientos de membresías.
Validación	Confirmar que los clientes reciben notificaciones oportunas.

4.2.2. Historias de usuario para el entrenador

Tabla 7. Tabla Historia de Inicio de Sesión

ID	H007
Nombre	Historia de Inicio de Sesión
Prioridad	Alta
Riesgo	Bajo
Descripción	Permitir que los entrenadores inicien sesión para acceder a las herramientas de gestión.
Validación	Verificar que los entrenadores pueden iniciar sesión y acceder a su panel de control.

Tabla 8. Tabla Historia de Asignación de Rutinas

ID	H008
Nombre	Historia de Asignación de Rutinas
Prioridad	Alta
Riesgo	Moderado
Descripción	Facilitar la asignación de rutinas de ejercicio a clientes específicos.
Validación	Se debe confirmar que los entrenadores pueden asignar rutinas efectivamente.

Tabla 9. Tabla Historia de Asignación de Planes Alimenticios:

ID	H009
Nombre	Historia de Asignación de Planes Alimenticios
Prioridad	Alta
Riesgo	Moderado
Descripción	Permitir que los entrenadores asignen planes alimenticios personalizados a sus clientes.
Validación	Verificar que los entrenadores pueden asignar planes alimenticios de manera exitosa.

Tabla 10. Tabla Historia de Gestión de Clientes

ID	H011
Nombre	Historia de Gestión de Clientes
Prioridad	Alta
Riesgo	Moderado
Descripción	Facilitar la gestión de clientes, incluyendo la adición y eliminación de clientes.
Validación	Verificar que los entrenadores pueden gestionar efectivamente la información de los clientes.

4.3. Requerimientos Funcionales

Los requerimientos funcionales fueron obtenidos a partir de las historias de usuario. Estos requerimientos funcionales cumplen un papel fundamental para el correcto funcionamiento de la página web. Estos requerimientos cumplen con las necesidades del usuario.

Una vez analizadas las historias de usuario se identificó los requerimientos para el correcto funcionamiento de la página web para el gimnasio.

- Iniciar sesión
- Gestión de Clientes
- Asignación de Rutinas y Planes Alimenticios
- Gestión de Rutinas
- Seguimiento de membresías
- Gestión de Usuarios
- Notificaciones por Correo Electrónico
- Gestión de planes alimenticios
- Gestión de roles y permisos
- Panel de Administración

4.4. Requerimientos No Funcionales

En base a las historias de usuario se ha obtenido los requerimientos no funcionales. Dichos requerimientos son necesarios para garantizar el buen funcionamiento de la aplicación web los cuales vienen de las necesidades de los usuarios. Los requerimientos no funcionales se detallan a continuación.

- Compatibilidad
- Responsive
- Arquitectura MVC

- Se usará asp.net Core para el desarrollo del backend y del frontend.

4.5. Casos de uso de la aplicación

En base a los requisitos funcionales, los requisitos no funcionales y las historias de usuario, se elaborará los casos de uso para nuestro aplicativo web.

Los casos de uso representan detalladamente las interacciones entre los clientes, el administrador y el entrenador con el sistema web. Se describe los escenarios y las acciones que se pueden dar al interactuar con la aplicación web.

Figura 1. Figura de Caso de uso de Cliente

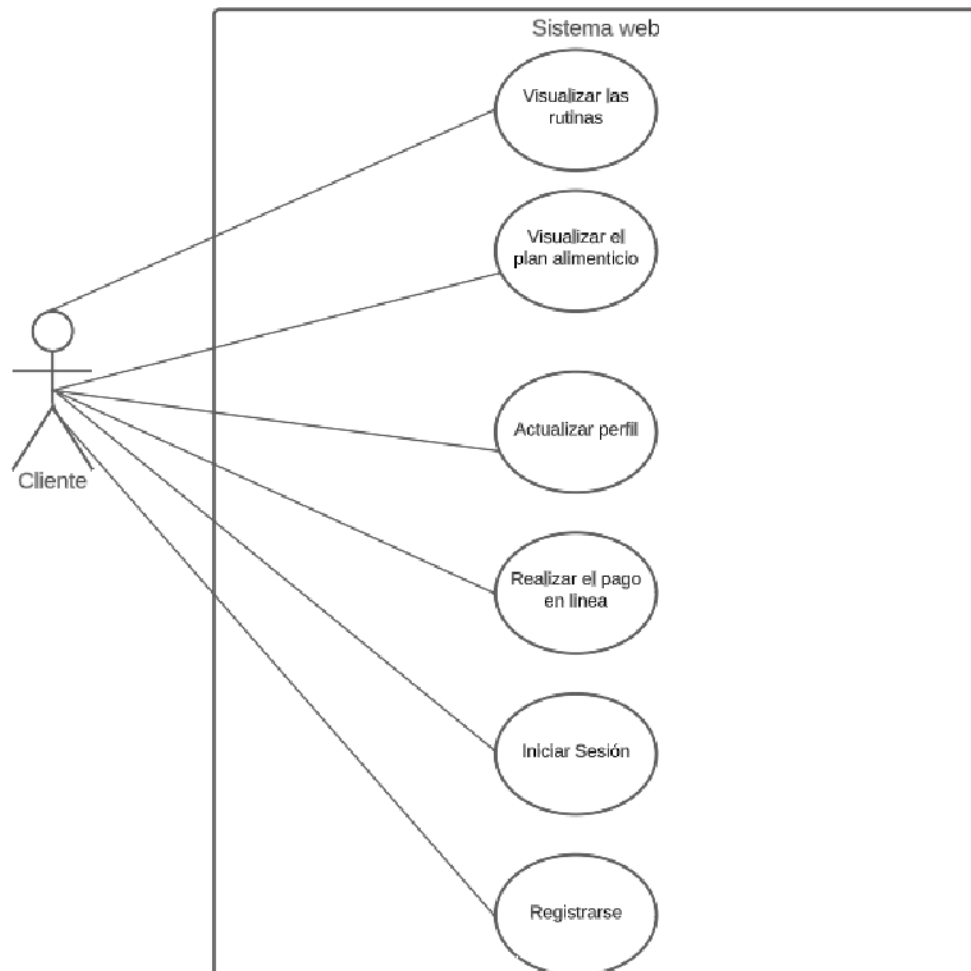


Figura 2. Figura de Caso de uso de Entrenador

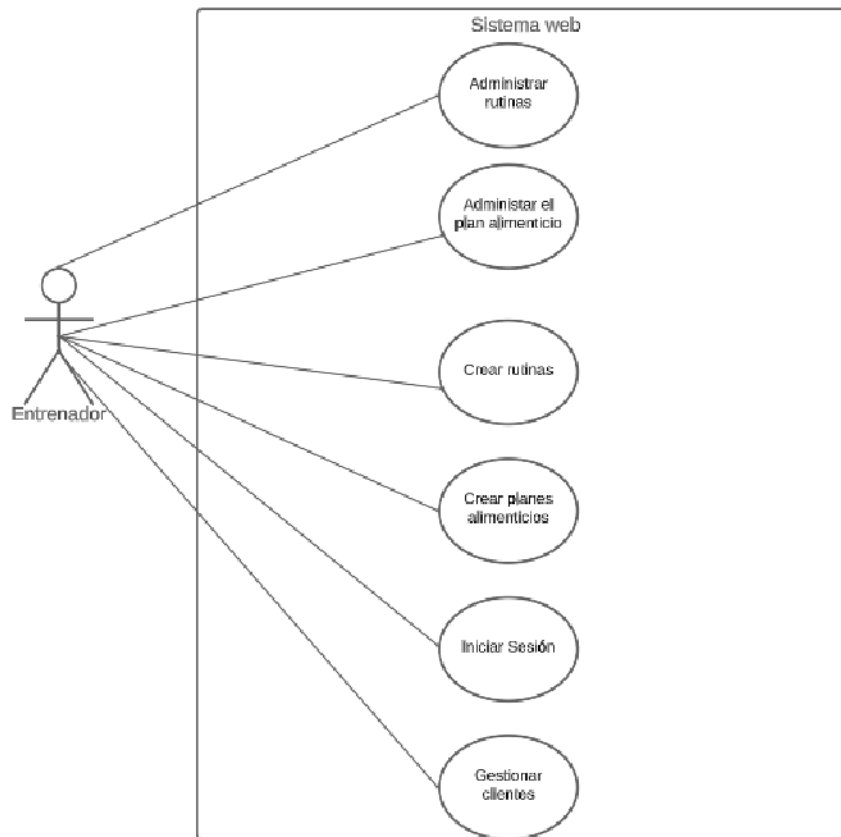


Figura 3. Figura de Caso de uso de Administrador

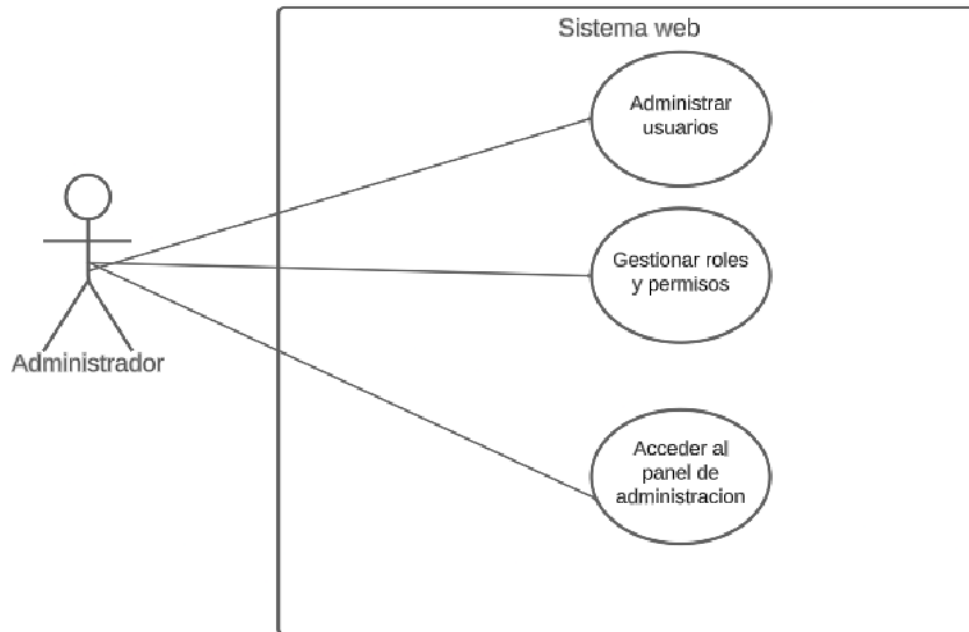
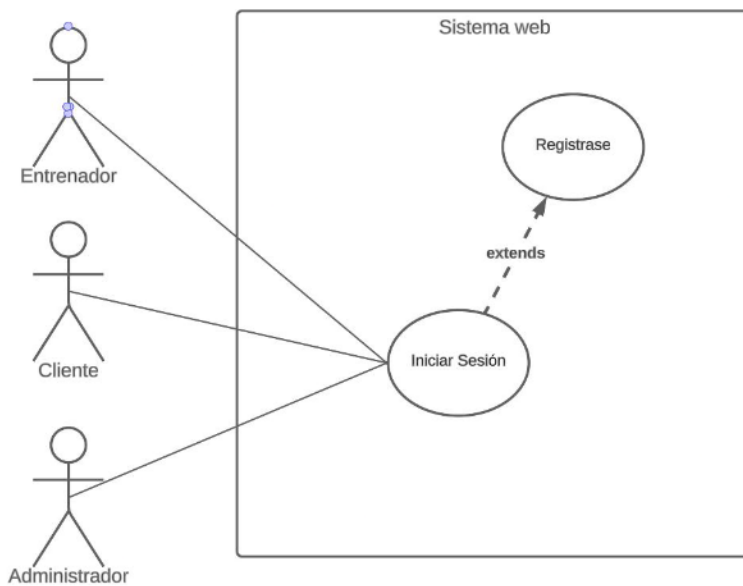


Figura 4. Figura de Caso de uso iniciar sesión y registrarse



Descripción: Los actores pueden ingresar al sistema

Actores: Entrenador, Cliente, Administrador

Flujo Principal:

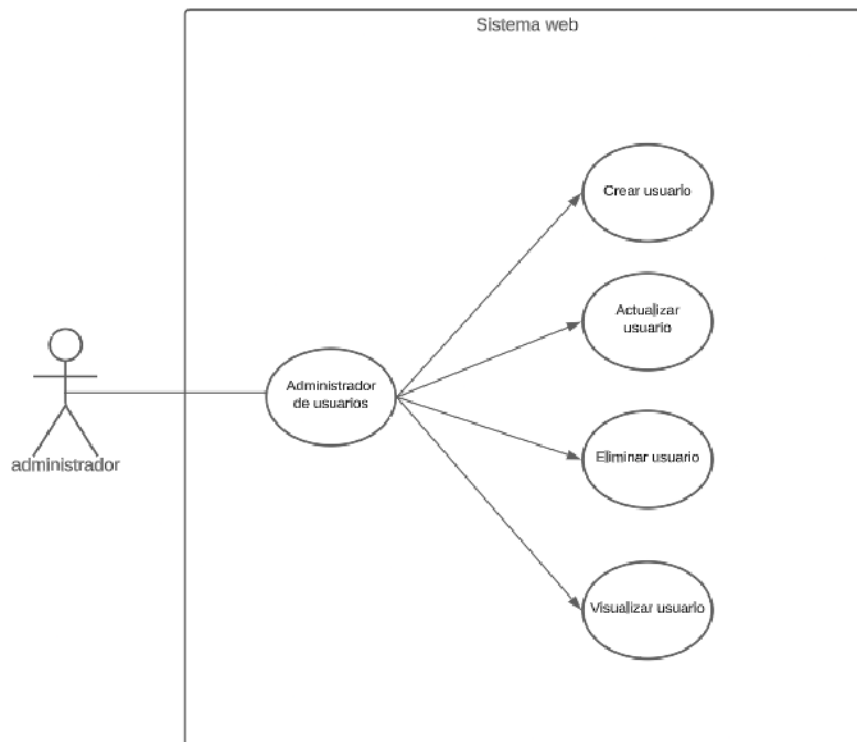
El actor crear un usuario para ingresar al sistema.

Usando los credenciales previamente creados el actor ingresa al sistema.

El sistema valida los credenciales ingresados.

En caso de que el ingreso sea exitoso se redirige a la página principal del sistema.

Figura 5. Figura de Caso de uso: Administrar Usuarios.



Caso de uso: Crear usuario

Descripción: Los actores pueden crear a los usuarios.

Actores: Administrador

Flujo Principal:

El actor selecciona el apartado de usuarios

El sistema presenta la lista de usuarios.

El actor selecciona el botón de crear usuario.

El sistema carga un formulario para crear un usuario nuevo.

El sistema crea un nuevo usuario

El sistema presenta la lista de usuarios actualizada

Caso de uso: Actualizar usuario

Descripción: Los actores pueden actualizar a los usuarios.

Actores: Administrador

Flujo Principal:

El actor selecciona el apartado de usuarios

El sistema presenta la lista de usuarios.

El actor selecciona el botón de actualizar.

El sistema carga un formulario con los datos del usuario.

El actor modifica el usuario seleccionado.

El actor presiona el botón guardar.

El sistema actualiza el registro del usuario seleccionado.

El sistema presenta la lista de usuarios actualizada

Caso de uso: Eliminar usuario

Descripción: Los actores pueden eliminar a los usuarios.

Actores: Administrador

Flujo Principal:

El actor selecciona el apartado de usuarios

El sistema presenta la lista de usuarios.

El actor selecciona el botón de eliminar usuario en el usuario que desea eliminar.

El sistema muestra un mensaje para confirmar.

El actor presiona el botón confirmar.

El sistema elimina el usuario deseado.

El sistema muestra un mensaje de éxito o error.

El sistema presenta la lista de usuarios actualizada

Caso de uso: Visualizar usuario

Descripción: Los actores pueden visualizar la lista de usuarios.

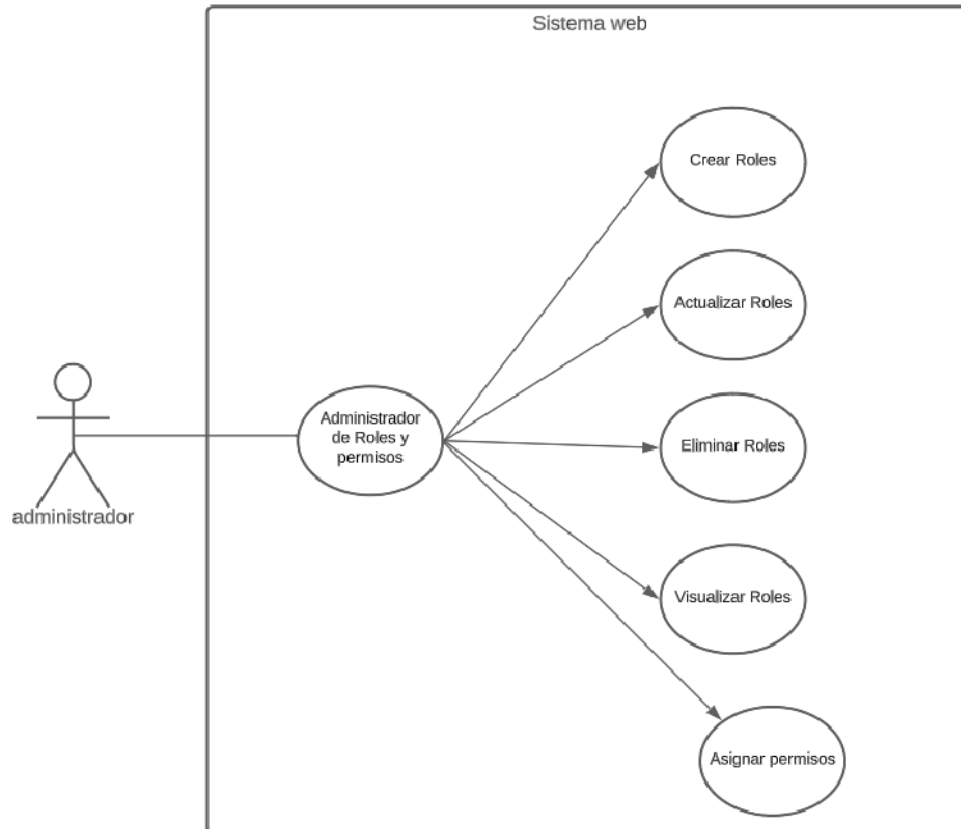
Actores: Administrador

Flujo Principal:

El actor selecciona el apartado de usuarios

El sistema presenta la lista de usuarios.

Figura 6. Figura de Caso de uso: Administrar Roles y permisos.



Caso de uso: Crear rol

Descripción: Los actores pueden crear roles.

Actores: Administrador

Flujo Principal:

1. El actor selecciona el apartado de roles

2. El sistema presenta la lista de roles.
3. El actor selecciona el botón de crear rol.
4. El sistema carga un formulario para crear un rol nuevo.
5. El sistema crea un nuevo rol.
6. El sistema presenta la lista de roles actualizada.

Caso de uso: Actualizar roles

Descripción: Los actores pueden actualizar a los roles.

Actores: Administrador

Flujo Principal:

1. El actor selecciona el apartado de roles
2. El sistema presenta la lista de roles.
3. El actor selecciona el botón del rol a actualizar.
4. El sistema carga un formulario con los datos del rol.
5. El actor modifica el rol seleccionado.
6. El actor presiona el botón guardar.
7. El sistema actualiza el registro del rol seleccionado.
8. El sistema presenta la lista de roles actualizada

Caso de uso: Eliminar rol

Descripción: Los actores pueden eliminar a los roles.

Actores: Administrador

Flujo Principal:

1. El actor selecciona el apartado de roles
2. El sistema presenta la lista de roles.
3. El actor selecciona el botón de eliminar rol en el rol que desea eliminar.
4. El sistema muestra un mensaje para confirmar.
5. El actor presiona el botón confirmar.
6. El sistema elimina el rol deseado.
7. El sistema muestra un mensaje de éxito o error.
8. El sistema presenta la lista de roles actualizada

Caso de uso: Visualizar roles

Descripción: Los actores pueden visualizar la lista de roles.

Actores: Administrador

Flujo Principal:

1. El actor selecciona el apartado de roles
2. El sistema presenta la lista de roles.

Caso de uso: Asignar permiso a roles

Descripción: Los actores pueden visualizar la lista de roles.

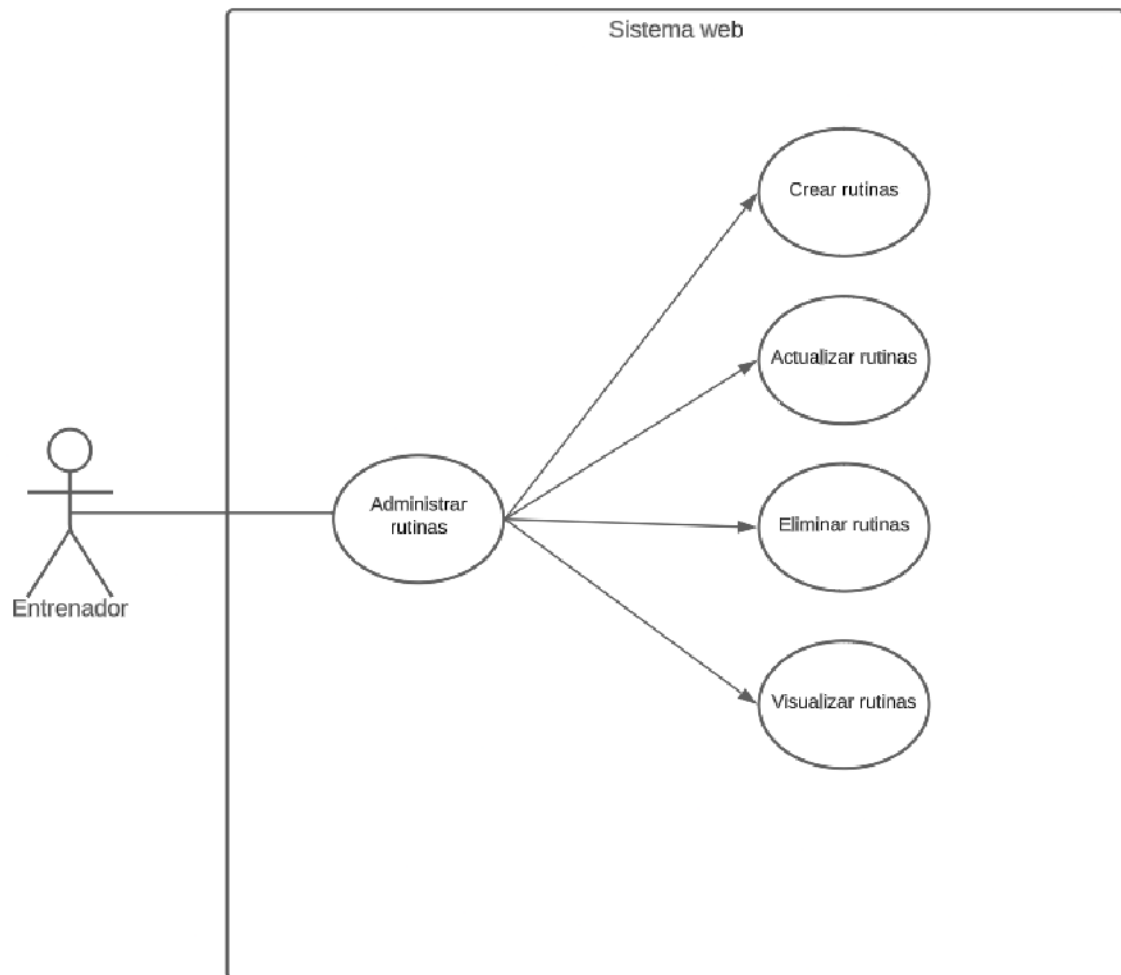
Actores: Administrador

Flujo Principal:

1. El actor selecciona el apartado de roles

2. El sistema presenta la lista de roles.
3. El actor selecciona el rol al que desea asignar permisos.
4. El sistema muestra un mensaje para confirmar.
5. El actor presiona el botón guardar.
6. El sistema guardar el rol con los permisos otorgados.
7. El sistema muestra un mensaje de éxito o error.

Figura 7. Figura de Caso de uso: Administrar rutinas.



Caso de uso: Crear rutina

Descripción: Los actores pueden crear rutinas.

Actores: Entrenadores

Flujo Principal:

1. El actor selecciona el apartado de rutinas.
2. El sistema presenta la lista de rutinas.

3. El actor selecciona el botón de crear rutina.
4. El sistema carga un formulario para crear una rutina nueva.
5. El sistema crea una nueva rutina.
6. El sistema presenta la lista de rutina actualizada

Caso de uso: Actualizar rutina

Descripción: Los actores pueden actualizar a los rutinas.

Actores: Entrenador

Flujo Principal:

1. El actor selecciona el apartado de rutinas.
2. El sistema presenta la lista de rutinas.
3. El actor selecciona el botón de editar en la rutina que desea actualizar.
4. El sistema carga un formulario con los datos de la rutina.
5. El actor modifica la rutina seleccionada.
6. El actor presiona el botón guardar.
7. El sistema actualiza el registro de la rutina seleccionada.
8. El sistema presenta la lista de rutinas actualizada

Caso de uso: Eliminar rutina

Descripción: Los actores pueden eliminar a los rutinas.

Actores: Entrenador

Flujo Principal:

1. El actor selecciona el apartado de rutinas.
2. El sistema presenta la lista de rutinas.
3. El actor selecciona el botón de eliminar rutina en la rutina que desea eliminar.
4. El sistema muestra un mensaje para confirmar.
5. El actor presiona el botón confirmar.
6. El sistema elimina la rutina deseado.
7. El sistema muestra un mensaje de éxito o error.
8. El sistema presenta la lista de rutinas actualizada

Caso de uso: Visualizar rutinas.

Descripción: Los actores pueden visualizar la lista de rutinas.

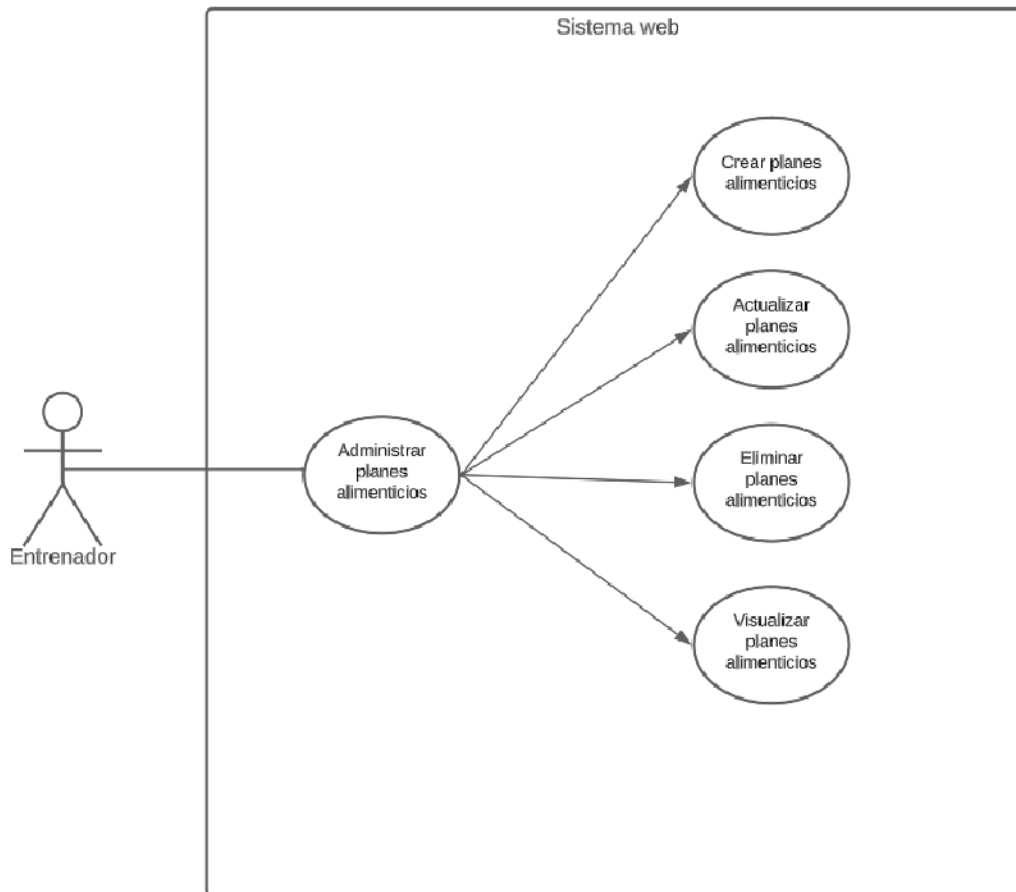
Actores: Entrenador

Flujo Principal:

El actor selecciona el apartado de rutinas.

El sistema presenta la lista de rutinas.

Figura 8. Figura de Caso de uso: Administrar planes alimenticios.



Caso de uso: Crear plan alimenticio

Descripción: Los actores pueden crear planes alimenticios.

Actores: Entrenadores

Flujo Principal:

1. El actor selecciona el apartado de planes alimenticios.
2. El sistema presenta la lista de planes alimenticios.
3. El actor selecciona el botón de crear plan alimenticio.

4. El sistema carga un formulario para crear un plan alimenticio nuevo.
5. El sistema crea un nuevo plan alimenticio.
6. El sistema presenta la lista de planes alimenticios actualizada

Caso de uso: Actualizar plan alimenticio

Descripción: Los actores pueden actualizar a los planes alimenticios.

Actores: Entrenador

Flujo Principal:

1. El actor selecciona el apartado de planes alimenticios.
2. El sistema presenta la lista de planes alimenticios.
3. El actor selecciona el botón de editar en plan alimenticio que desea actualizar.
4. El sistema carga un formulario con los datos del plan alimenticio.
5. El actor modifica el plan alimenticio seleccionado.
6. El actor presiona el botón guardar.
7. El sistema actualiza el registro del plan alimenticio seleccionado.
8. El sistema presenta la lista de planes alimenticios actualizada

Caso de uso: Eliminar plan alimenticio

Descripción: Los actores pueden eliminar a los planes alimenticios.

Actores: Entrenador

Flujo Principal:

1. El actor selecciona el apartado de planes alimenticios.

2. El sistema presenta la lista de planes alimenticios.
3. El actor selecciona el botón de eliminar plan alimenticio en el plan alimenticio que desea eliminar.
4. El sistema muestra un mensaje para confirmar.
5. El actor presiona el botón confirmar.
6. El sistema elimina el plan alimenticio deseado.
7. El sistema muestra un mensaje de éxito o error.
8. El sistema presenta la lista de planes alimenticios actualizada

Caso de uso: Visualizar planes alimenticios.

Descripción: Los actores pueden visualizar la lista de planes alimenticios.

Actores: Entrenador

Flujo Principal:

El actor selecciona el apartado de planes alimenticios.

El sistema presenta la lista de planes alimenticios.

4.6. Product Backlog

El Product Backlog nos permite observar las funcionalidades y los requisitos que debe cumplir el sistema web. El Product backlog se puede actualizar en la vida útil del sistema.

Tabla 11. Tabla de Product Backlog

Product Backlog		
Código	Nombre	Descripción
Requerimientos Funcionales		
RF01	Iniciar Sesión	Los usuarios registrados deben poder iniciar sesión en sus cuentas utilizando sus credenciales.
RF02	Gestión de Clientes	Los administradores deben poder agregar, modificar y eliminar registros de clientes.
RF03	Asignación de Rutinas y Planes Alimenticios	Los entrenadores deben poder asignar rutinas de ejercicios y planes alimenticios a cada cliente.
RF04	Gestión de Rutinas	Los entrenadores deben poder agregar, modificar y eliminar registros de rutinas.
RF05	Gestión de usuarios	Los administradores deben poder agregar, modificar y eliminar registros de usuarios.
RF06	Notificaciones por Correo Electrónico	El sistema debe enviar notificaciones automáticas a los clientes sobre vencimientos de membresías.
RF07	Gestión de planes alimenticios	Los entrenadores deben poder agregar, modificar y eliminar registros de planes alimenticios.

RF08	Gestión de roles y permisos	<p>Los administradores deben poder asignar roles específicos (cliente, entrenador, administrador) a los usuarios.</p> <p>Los administradores deben poder ajustar los permisos de cada rol según sea necesario.</p>
Requerimientos no funcionales		
RNF01	Usabilidad	La interfaz debe ser fácil de usar e intuitiva
RNF02	Compatibilidad	La aplicación debe ser compatible con los últimos navegadores (Chrome, Firefox, Safari).
RNF03	Responsive	La interfaz de usuario debe ser responsiva y adaptarse a diferentes tamaños de pantalla.
RNF04	Arquitectura MVC	El sistema debe seguir el patrón de diseño Modelo-Vista-Controlador (MVC) para garantizar una separación clara de las responsabilidades
RNF05	Desarrollo en ASP.NET Core	La aplicación debe estar diseñada y optimizada para garantizar un rendimiento rápido y eficiente utilizando la plataforma ASP.NET Core.

4.7. Arquitectura de software

Tres componentes principales componen la arquitectura Modelo-Vista-Controlador (MVC) que se implementará en el proyecto de titulación destinado a la gestión de un gimnasio. El Modelo

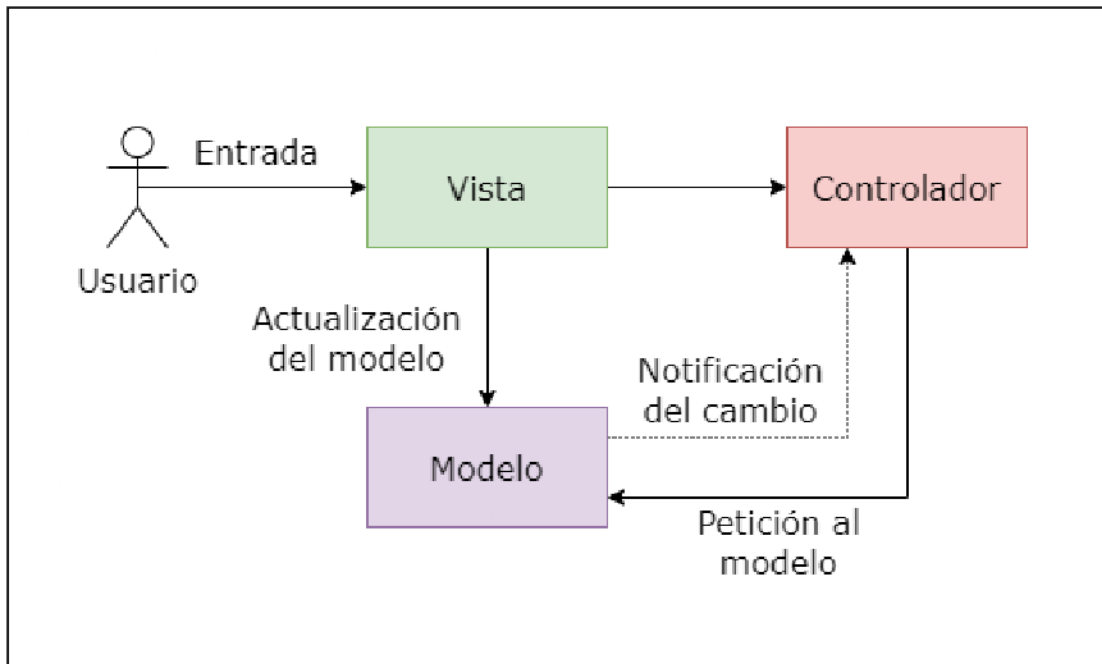
está en el centro del sistema y es responsable de administrar la lógica comercial, que incluye operaciones básicas con clientes, rutinas, planes de alimentación y control de membresías. El mantenimiento y la manipulación de datos, así como la implementación de la lógica de negocios central, como la asignación de rutinas y el seguimiento del estado de las membresías, son sus principales responsabilidades.

La Vista se encargará de presentar la información de manera comprensible y facilitar la interacción. Aquí, se organizarán elementos visuales y formularios para mostrar datos sobre clientes, rutinas, planes de alimentación y detalles de membresía. Proporcionar una interfaz amigable que recolecte datos de entrada del usuario de manera eficiente es su tarea principal.

El Controlador, el tercer componente, coordinará las interacciones del usuario al recibir y procesar solicitudes HTTP de Vista. Para garantizar una actualización adecuada del Modelo en respuesta a las acciones del usuario, como la asignación de rutinas y la actualización de membresías, actuará como intermediario entre la lógica comercial y la interfaz de usuario.

La estructura lógica de los archivos y carpetas facilita la identificación y modificación de cada componente. Al adherirse a los principios fundamentales de la modularidad y la separación de preocupaciones, esta implementación de MVC facilita la flexibilidad, el mantenimiento y la escalabilidad del sistema.

Figura 9. Figura de Arquitectura de software



Nota. Es importante entender el funcionamiento de la arquitectura MVC. Imagen tomada de (Huet, 2022)

4.8. Épicas de usuario

Las épicas de usuario son cruciales al ofrecer descripciones detalladas de las funcionalidades clave que se buscarán desarrollar. Para organizar, planificar y priorizar las tareas, estas épicas dividen las metas extensas en historias de usuario más manejables.

Tabla 12. Tabla de lista de prioridades

Prioridad	Valoración
Baja	1

Media	2
Alta	3
Muy Alta	4

Tabla 13. Tabla de Épicas de usuario

Epica	Código	Nombre	Prioridad (1-4)
01	001	✓ Descargar e instalar Visual Studio o Visual Studio Code.	4
	002	✓ Configurar las herramientas esenciales para el desarrollo en ASP.NET Core.	3
	003	✓ Crear el proyecto base de la aplicación.	4
02	004	✓ Descargar e integrar las dependencias necesarias para el desarrollo.	3
	005	✓ Instalar y configurar librerías externas relevantes para el proyecto.	3
	006	✓ Gestionar las actualizaciones y versiones de las dependencias.	2
03	007	✓ Descargar y aplicar plantillas de diseño para la interfaz de usuario.	3
	008	✓ Personalizar la apariencia inicial de la aplicación según los requisitos de diseño.	3

	009	V	Configurar y adaptar las plantillas para ser compatibles con el enfoque del gimnasio.	2
04	010	V	Descargar e instalar el sistema de gestión de bases de datos	4
	011	V	Configurar la base de datos inicial del sistema.	3
	012	V	Realizar las migraciones iniciales y establecer las relaciones entre las tablas.	4
05	013	V	Establecer la conexión entre la aplicación y la base de datos.	3
	014	V	Configurar los parámetros de conexión y las cadenas de conexión.	3
	015	V	Realizar pruebas de conexión para garantizar la comunicación efectiva.	3
06	016	V	Desarrollar un módulo y componente dedicado a la gestión de usuarios.	3
	017	V	Implementar la interfaz para el registro y modificación de perfiles.	3
	018	V	Configurar el sistema de autenticación para usuarios.	4
07	019	V	Configurar roles de usuario (cliente, entrenador, administrador).	3
	020	V	Establecer permisos y restricciones según el rol del usuario.	4

	021	V	Desarrollar interfaces para la administración de roles y permisos.	3
08	022	V	Desarrollar un módulo y componente dedicado a la administración de membresías.	3
	023	V	Implementar la interfaz para la creación y asignación de membresías.	4
10	024	V	Crear interfaces adaptadas a las necesidades de clientes y entrenadores.	3
	025	V	Diseñar paneles de control específicos para cada rol.	4
	026	V	Implementar lógica personalizada para acciones de clientes y entrenadores.	3

4.9. Planificación de Sprints

Con la ayuda del Product Backlog y las épicas de usuario planeamos nuestros Sprints.

Tabla 14. Tabla de planificación Sprint 1

Sprint 1	
Código	Descripción
V01	Descargar e instalar Visual Studio o Visual Studio Code.
V02	Configurar las herramientas esenciales para el desarrollo en ASP.NET Core.
V03	Crear el proyecto base de la aplicación.

Tabla 15. Tabla de Planificación Sprint 2

Sprint 2	
Código	Descripción
V04	Descargar e integrar las dependencias necesarias para el desarrollo.
V05	Instalar y configurar librerías externas relevantes para el proyecto.
V06	Gestionar las actualizaciones y versiones de las dependencias.

Tabla 16. Tabla de Planificación Sprint 3

Sprint 3	
Código	Descripción
V07	Descargar y aplicar plantillas de diseño para la interfaz de usuario.
V08	Personalizar la apariencia inicial de la aplicación según los requisitos de diseño.
V09	Configurar y adaptar las plantillas para ser compatibles con el enfoque del gimnasio.

Tabla 17. Tabla de Planificación Sprint 4

Sprint 4	
Código	Descripción

V10	Descargar e instalar el sistema de gestión de bases de datos.
V11	Configurar la base de datos inicial del sistema.
V12	Realizar las migraciones iniciales y establecer las relaciones entre las tablas.

Tabla 18. Tabla de Planificación Sprint 5

Sprint 5	
Código	Descripción
V13	Establecer la conexión entre la aplicación y la base de datos.
V14	Configurar los parámetros de conexión y las cadenas de conexión.
V15	Realizar pruebas de conexión para garantizar la comunicación efectiva.

Tabla 19. Tabla de Planificación Sprint 6

Sprint 6	
Código	Descripción
V16	Desarrollar un módulo y componente dedicado a la gestión de usuarios.
V17	Implementar la interfaz para el registro y modificación de perfiles.
V18	Configurar el sistema de autenticación para usuarios.

Tabla 20. Tabla de Planificación Sprint 7

Sprint 7

Código	Descripción
V19	Configurar roles de usuario (cliente, entrenador, administrador).
V20	Establecer permisos y restricciones según el rol del usuario.
V21	Desarrollar interfaces para la administración de roles y permisos.

Tabla 21. Tabla de Planificación Sprint 9

Sprint 9	
Código	Descripción
V22	Desarrollar un módulo y componente dedicado a la planificación de rutinas.
V23	Implementar la interfaz para la creación y asignación de rutinas.

Tabla 22. Tabla de Planificación Sprint 10

Sprint 10	
Código	Descripción
V24	Crear interfaces adaptadas a las necesidades de clientes y entrenadores.
V25	Diseñar paneles de control específicos para cada rol.
V26	Implementar lógica personalizada para acciones de clientes y entrenadores.

4.10. Base de datos

Figura 10. Figura de Modelo Conceptual

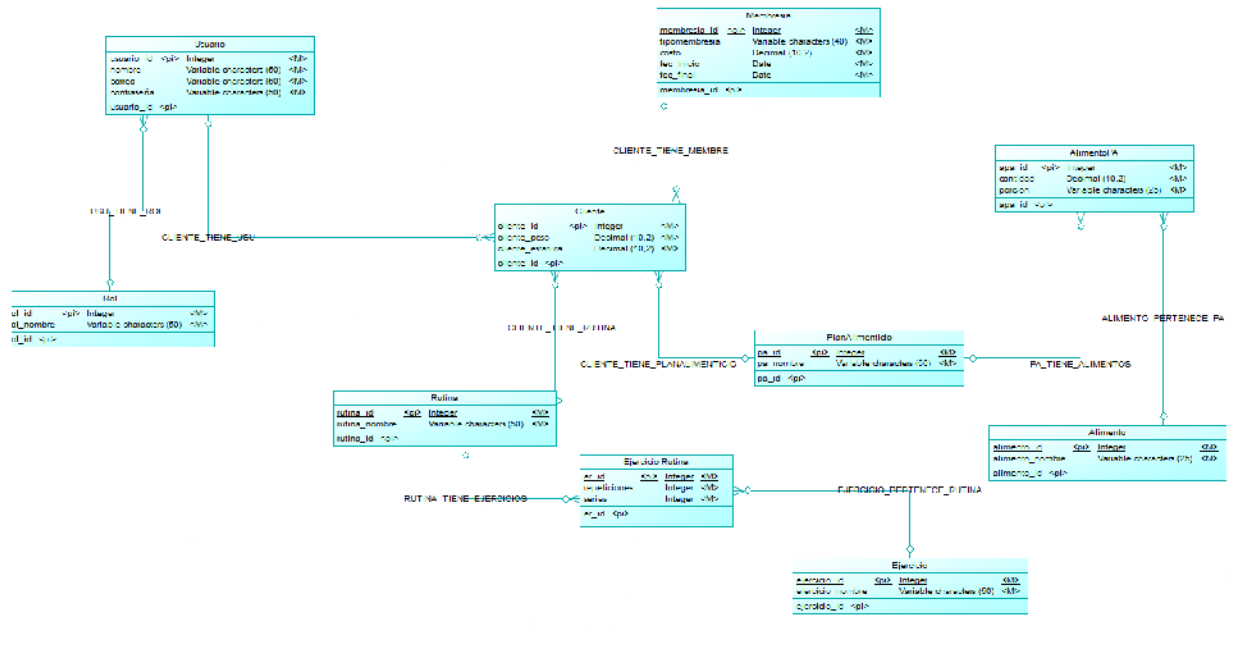


Figura 11. Figura de Modelo Lógico

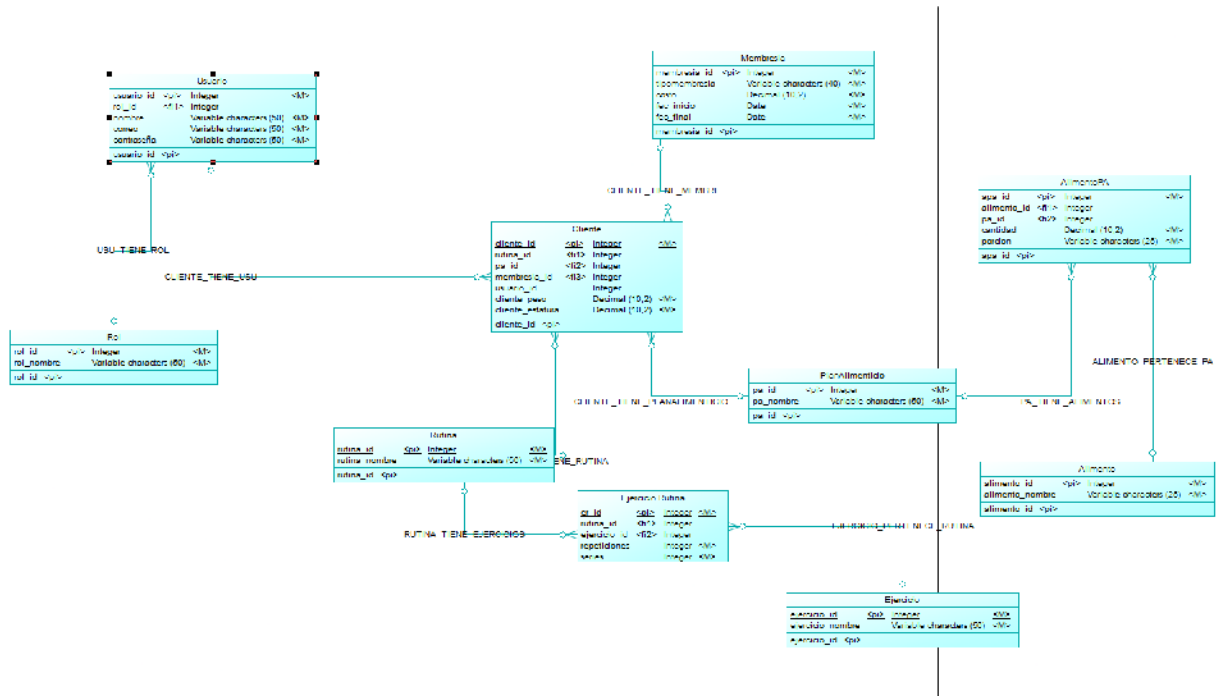
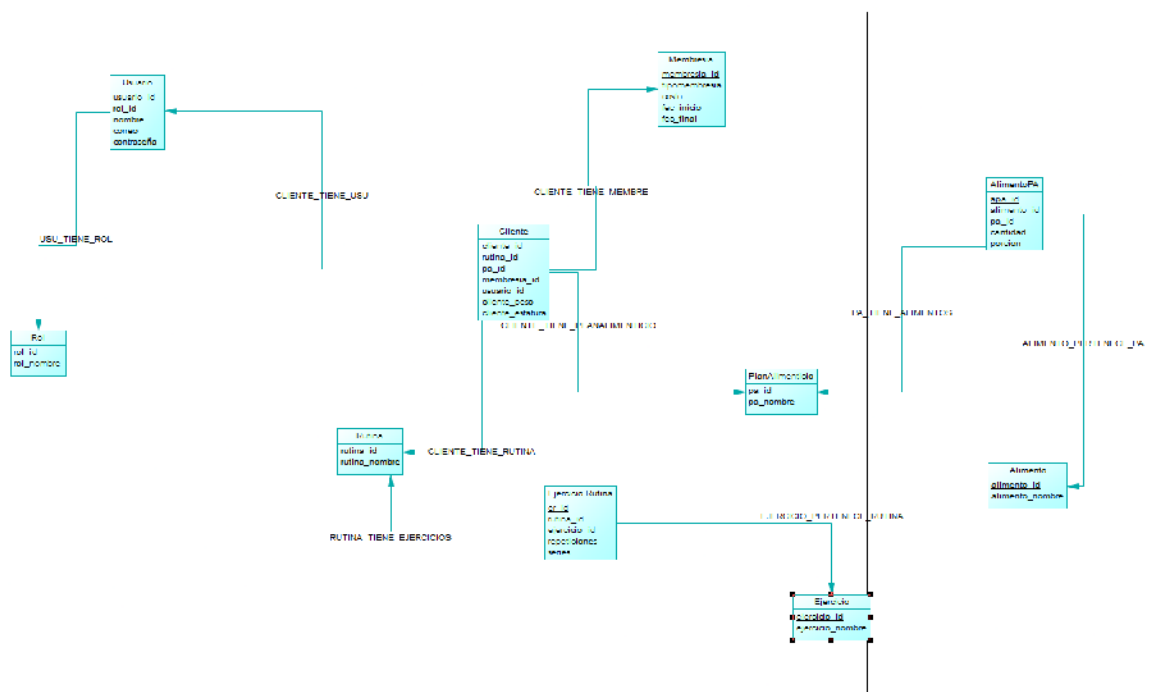


Figura 12. Figura de Modelo Físico



CAPÍTULO V: IMPLEMENTACIÓN

5. Implementación de la aplicación

La implementación de una aplicación es una parte crucial de cualquier proyecto de desarrollo de software, y esto también se aplica a las aplicaciones web de gestión del gimnasio. En esta etapa, se escribe el código requerido y la integración de múltiples módulos ya establecidos. Además, se llevan a cabo pruebas detalladas para asegurar la calidad y eficacia de la aplicación.

Dado que contribuye, la implementación adquiere gran importancia directamente al logro de los objetivos del proyecto, tanto específicos como generales. Esta es la etapa actual donde se materializan los diseños y especificaciones anteriores y se produce el producto final.

Al completar la implementación con éxito, la aplicación alcanza la funcionalidad deseada y sus componentes se integran adecuadamente. Además, se realizan pruebas rigurosas para identificar y corregir posibles errores.

5.1. Estándares de codificación

En primer lugar, abordaremos el estándar de declaración de variables, métodos, clases e interfaces CamelCase. La primera letra de cada palabra dentro de una variable debe escribirse con mayúscula según esta convención. Esta práctica ayuda a que el código sea más fácil de leer y comprender, especialmente cuando se trabaja en equipo y se revisa en el futuro. A continuación se presenta un ejemplo.

5.2. Implementación de la tesis

En la implementación de tesis llevamos a cabo al revisión de los Sprints y de las interfaces gráficas del sistema web con sus funciones finales.

Figura 13. Figura de controlador de Cliente

```
[Authorize(Roles = "Administrador, Entrenador")]  
1 reference  
public class ClienteController : Controller  
{  
    private readonly IClienteRepository _clienteRepository;  
    private readonly IMembresiaRepository _membresiaRepository;  
    private readonly IPlanalimentoRepository _planalimentoRepository;  
    private readonly IRutinaRepository _rutinaRepository;  
    private readonly IUsuarioRepository _usuarioRepository;  
}
```

Figura 14. Figura de Interfaz de Cliente

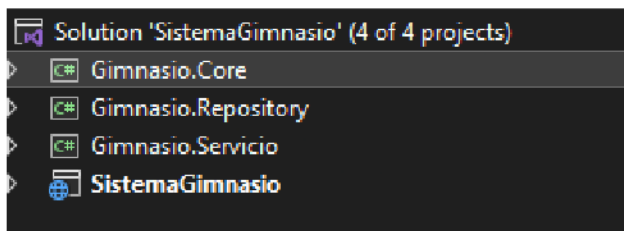
```
public interface IClienteRepository  
{  
    public List<Cliente> GetClientes ();  
    public Cliente GetClienteById(int? id);  
    public Cliente GetClienteByUsername(int id);  
    public void CreateCliente(Cliente cliente);  
    public void UpdateCliente(Cliente cliente);  
    public void DeleteCliente(Cliente cliente);  
    5 references  
    public bool ClienteExists(int id);  
}
```

En la figura presentada previamente, se puede observar la interfaz de “Cliente” donde los métodos presentan el estándar CamelCase

5.2.1. Organización de módulos

La organización de módulos es fundamental para que el código este mas limpio y legible. Se han ubicado los módulos en distintos modelos. Se presenta a continuación una figura.

Figura 15. Figura Organización de la solución



Al separar el proyecto en estos módulos, se obtiene un mayor control en el desarrollo del proyecto.

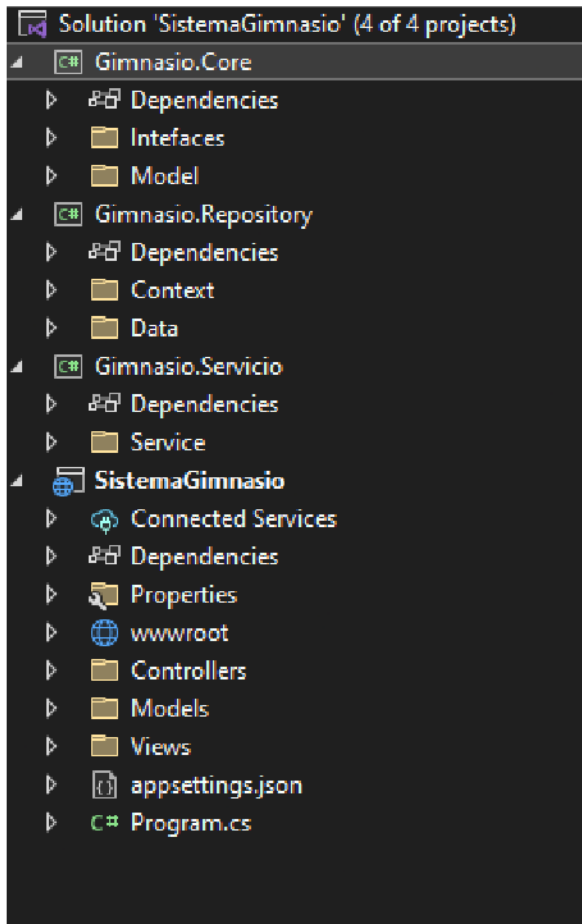
El módulo .Core se centra en la creación de interfaces y de clases. En caso de necesitar un servicio adicional, este se le incluirá en la carpeta de interfaces.

El módulo .Repository se enfoca en la conexión con la base de datos y la implementación de los métodos para interactuar con la misma.

El módulo .Servicio contiene los servicios necesarios para el sistema web funcione correctamente.

El SistemaGimnasio es el módulo de front-end asp net.Core MVC donde se desarrollan las vistas usando Razor Pages.

Figura 16. Figura Organización de la solución



5.2.2. Revisión Sprints

Tabla 23. Tabla de Sprint

Status	
No completado	Green
En proceso	Yellow
Completado	Red

Tabla 24. Revisión Sprint 1

Sprint 1			
Código	Descripción	Estatus	
V01	Descargar e instalar Visual Studio o Visual Studio Code.	Completado	
V02	Configurar las herramientas esenciales para el desarrollo en ASP.NET Core.	Completado	
V03	Crear el proyecto base de la aplicación.	Completado	

Tabla 25. Revisión Sprint 2

Sprint 2			
Código	Descripción	Estatus	
V04	Descargar e integrar las dependencias necesarias para el desarrollo.	Completado	
V05	Instalar y configurar librerías externas relevantes para el proyecto.	Completado	
V06	Gestionar las actualizaciones y versiones de las dependencias.	Completado	

Tabla 26. Revisión Sprint 3

Sprint 3			
-----------------	--	--	--

Código	Descripción	Estatus
V07	Descargar y aplicar plantillas de diseño para la interfaz de usuario.	Completado
V08	Personalizar la apariencia inicial de la aplicación según los requisitos de diseño.	Completado
V09	Configurar y adaptar las plantillas para ser compatibles con el enfoque del gimnasio.	Completado

Tabla 27. Revisión Sprint 4

Sprint 4		
Código	Descripción	Estatus
V10	Descargar e instalar el sistema de gestión de bases de datos.	Completado
V11	Configurar la base de datos inicial del sistema.	Completado
V12	Realizar las migraciones iniciales y establecer las relaciones entre las tablas.	Completado

Tabla 28. Revisión Sprint 5

Sprint 5		
Código	Descripción	Estatus
V13	Establecer la conexión entre la aplicación y la base de datos.	Completado

V14	Configurar los parámetros de conexión y las cadenas de conexión.	Completado
V15	Realizar pruebas de conexión para garantizar la comunicación efectiva.	Completado

Tabla 29. Revisión Sprint 6

Sprint 6		
Código	Descripción	Estatus
V16	Desarrollar un módulo y componente dedicado a la gestión de usuarios.	Completado
V17	Implementar la interfaz para el registro y modificación de perfiles.	Completado
V18	Configurar el sistema de autenticación para usuarios.	Completado

Tabla 30. Revisión Sprint 7

Sprint 7		
Código	Descripción	Estatus
V19	Configurar roles de usuario (cliente, entrenador, administrador).	Completado
V20	Establecer permisos y restricciones según el rol del usuario.	Completado

V21	Desarrollar interfaces para la administración de roles y permisos.	Completado	
-----	--	------------	--

Tabla 31. Revisión Sprint 9

Sprint 9			
Código	Descripción	Estatus	
V22	Desarrollar un módulo y componente dedicado a la planificación de rutinas.	Completado	
V23	Implementar la interfaz para la creación y asignación de rutinas.	Completado	

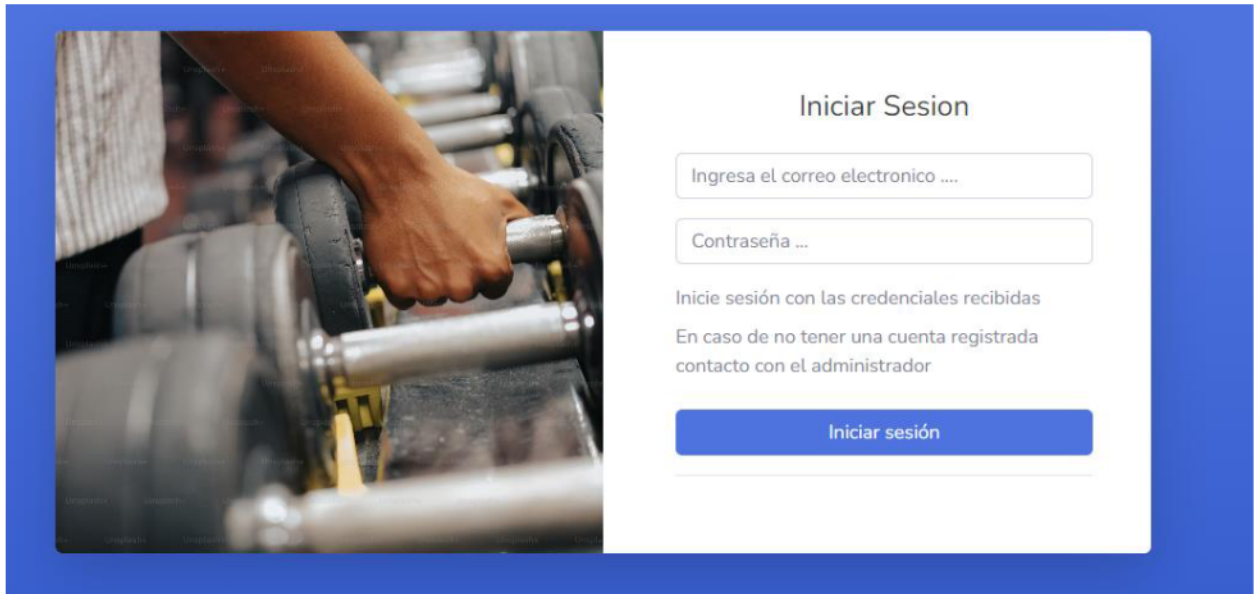
Tabla 32. Revisión Sprint 10

Sprint 10			
Código	Descripción	Estatus	
V24	Crear interfaces adaptadas a las necesidades de clientes y entrenadores.	Completado	
V25	Diseñar paneles de control específicos para cada rol.	Completado	
V26	Implementar lógica personalizada para acciones de clientes y entrenadores.	Completado	

5.3. Interfaces Gráficas

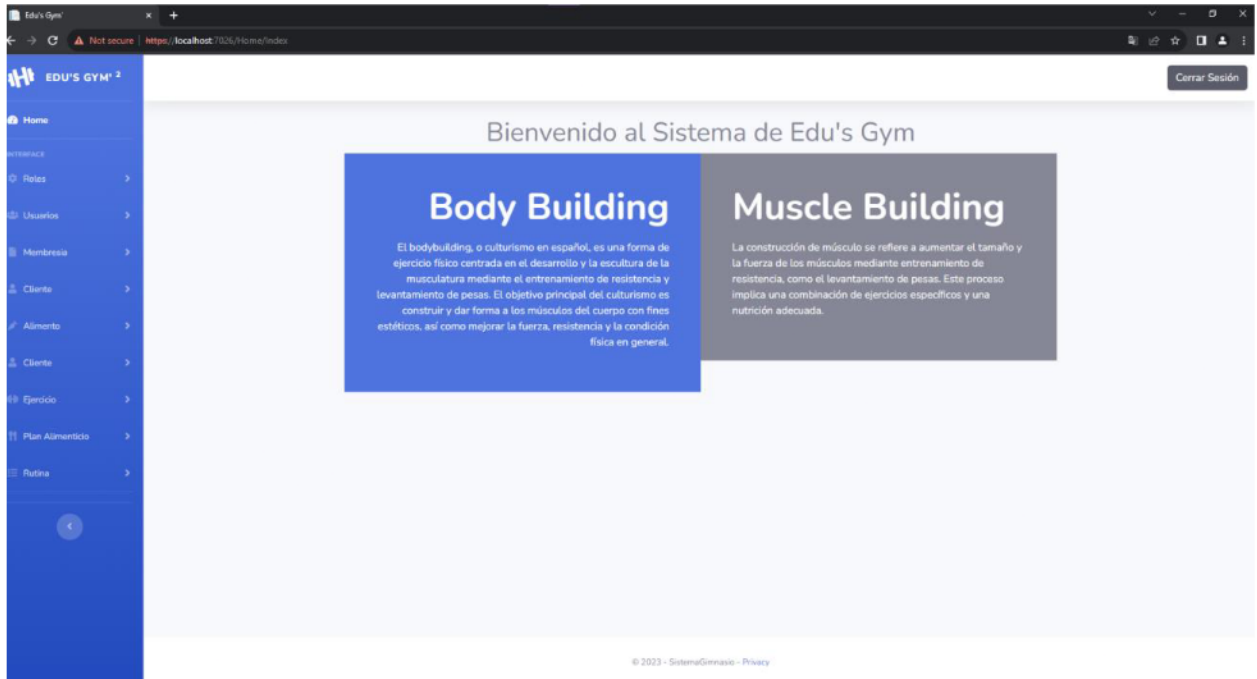
A continuación, se presenta las interfaces del sitio web para el gimnasio “Edu’s Gym”, donde se muestra la funcionalidad de cada módulo del sistema web.

Figura 17. Interfaz de inicio de sesión



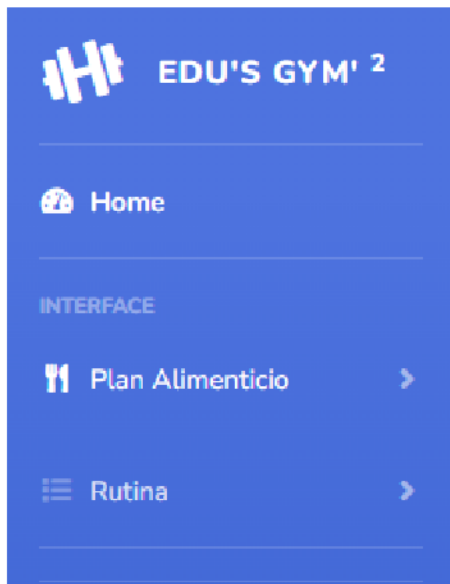
La figura 17 nos muestra la interfaz de inicio sesión de la aplicación. Este formulario solicita las credenciales del usuario, en el cual es el correo y la contraseña correspondiente. Se proporciona una breve descripción para conseguir el usuario y contraseña.

Figura 18. Interfaz Home



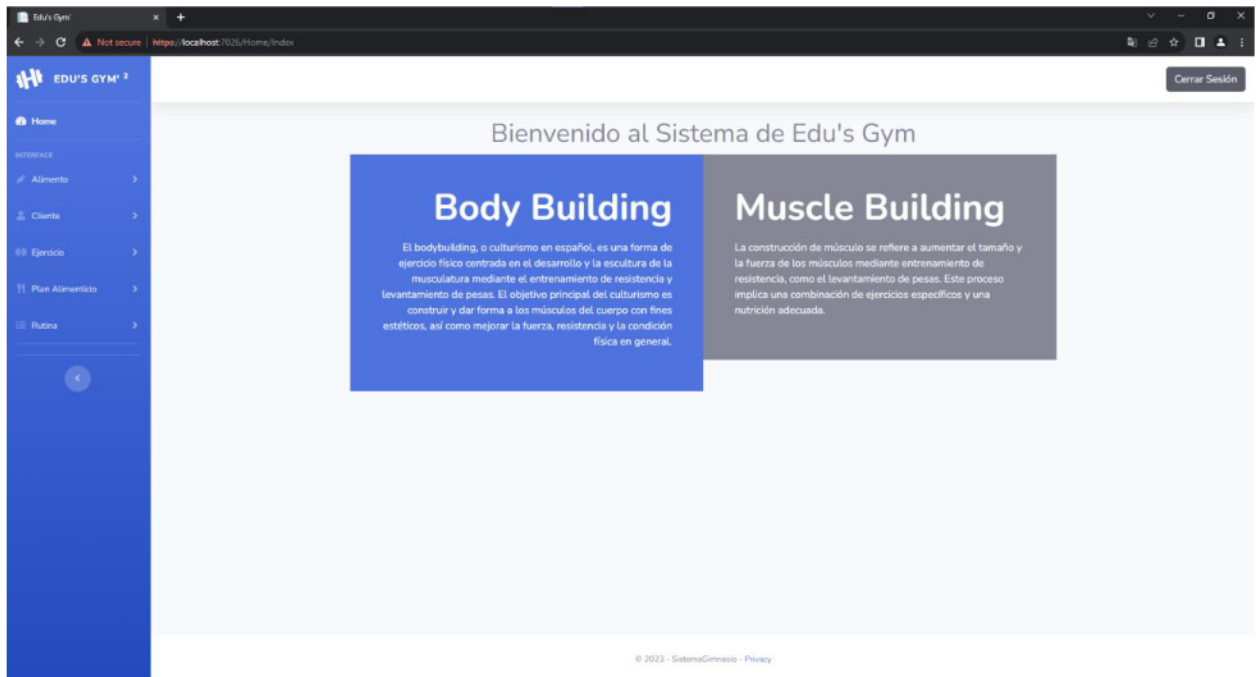
Nota. Una vez que el usuario haya ingresado sesión entrará automáticamente a la vista de home. El menú desplegable de la derecha cambiará según el Rol. El Rol de administrador tiene acceso a todas las interfaces.

Figura 19. Interfaz de menú de Cliente



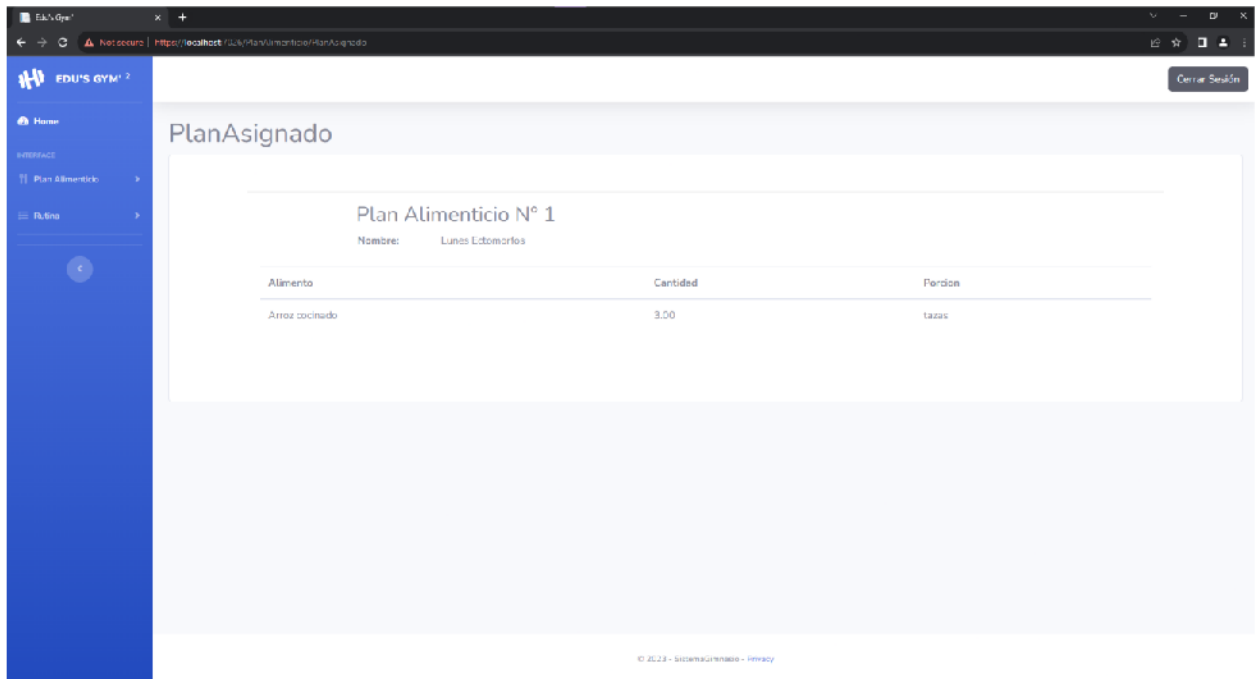
Nota. El cliente tiene acceso a la vista de Plan Alimenticio que debe seguir ese día y a la rutina que debe realizar.

Figura 20. Interfaz de Menú Entrenador



Nota. El entrenador tiene acceso a los CRUD para las tablas de alimento, cliente, ejercicio, plan alimenticio y Rutina.

Figura 21. Interfaz de Plan Alimenticio Asignado



Nota. La figura 21 muestra el plan alimenticio que el cliente debe seguir. La tabla nos muestra el alimento la cantidad de alimento que debe consumir junto a porción.

Figura 22. Interfaz de Rutina Asignada

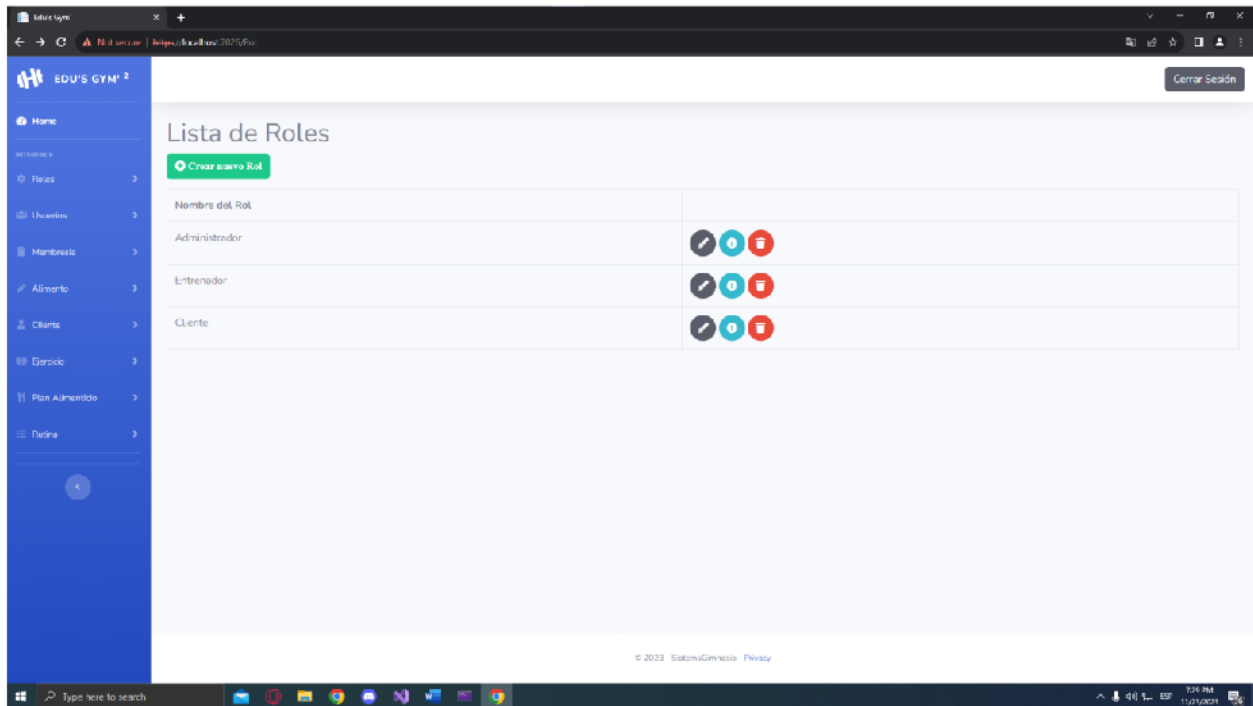
Rutina N° 9
Nombre: prueba

Ejercicio	Series	Repeticiones
Press Banca	4	12
Aperturas	4	12
Sentadilla	7	8
Press Declinado	4	4

Nota. La figura 22 muestra la rutina que el cliente debe seguir. La tabla nos muestra el ejercicio que debe realizar junto con las series y repeticiones de cada uno.

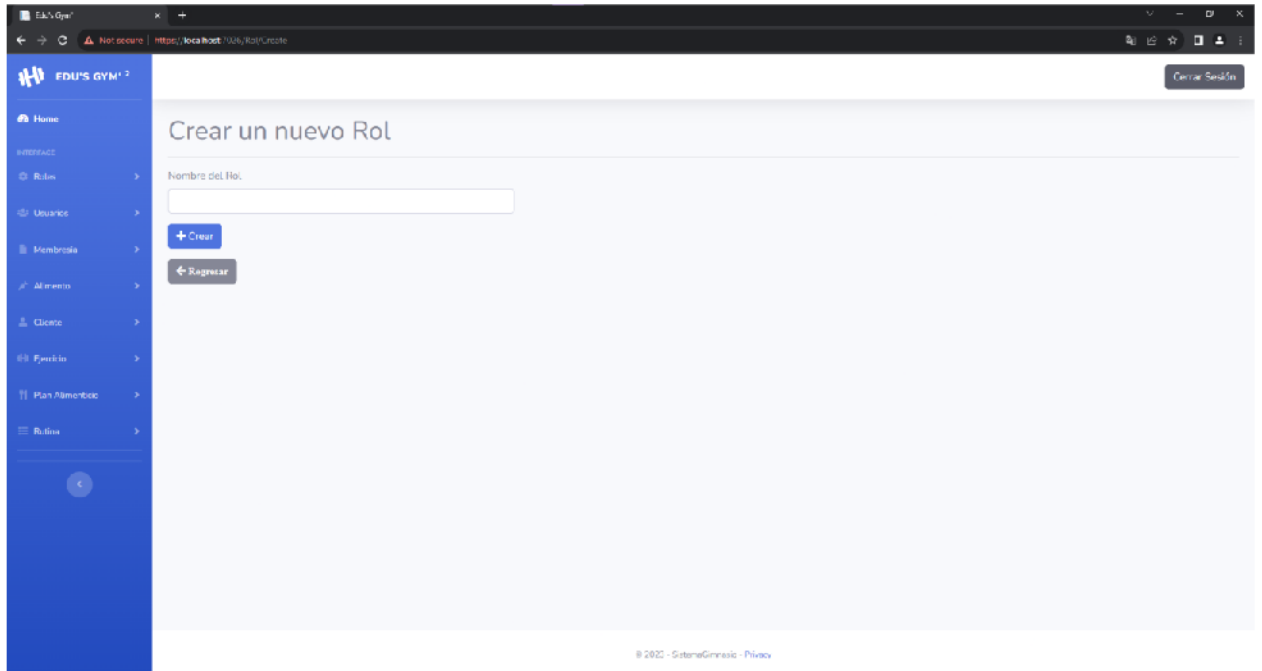
5.3.1. Gestión de Roles

Figura 23. Interfaz de listado de Roles



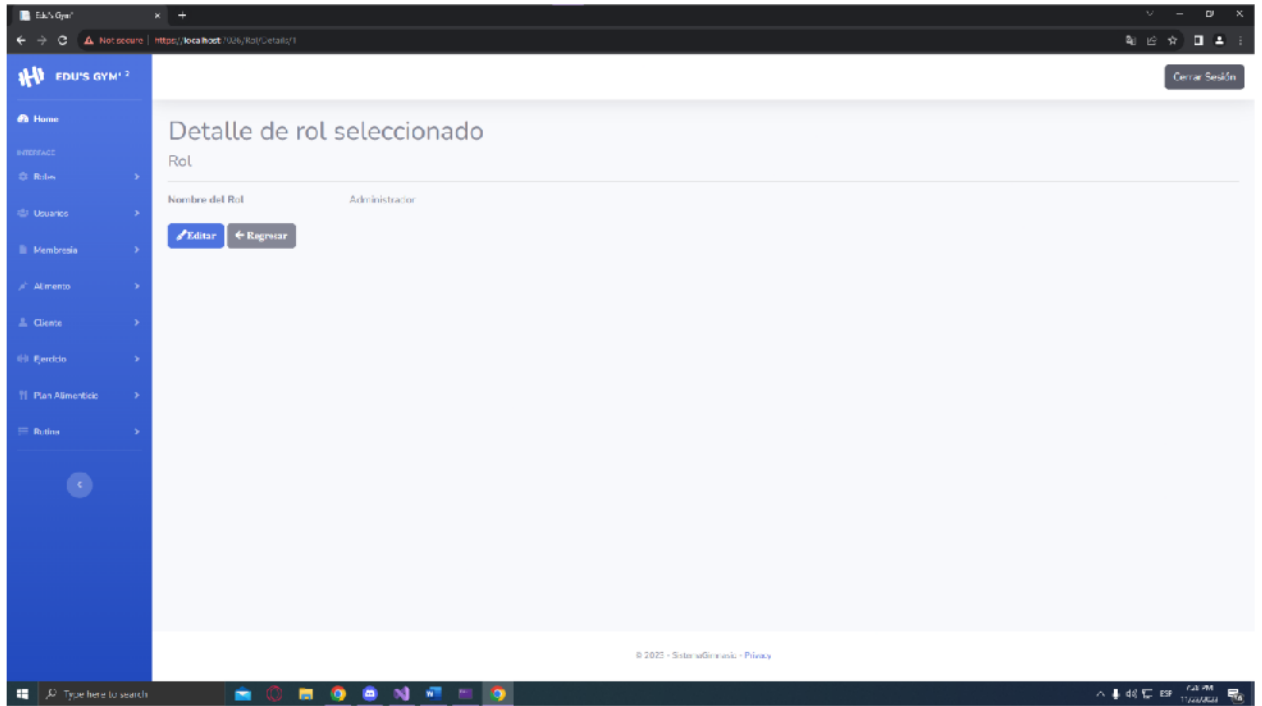
Nota. Se muestra el listado de los roles existentes. Los iconos de la derecha permiten editar, ver la información o eliminar el registro seleccionado.

Figura 24. Interfaz de crear un nuevo Rol



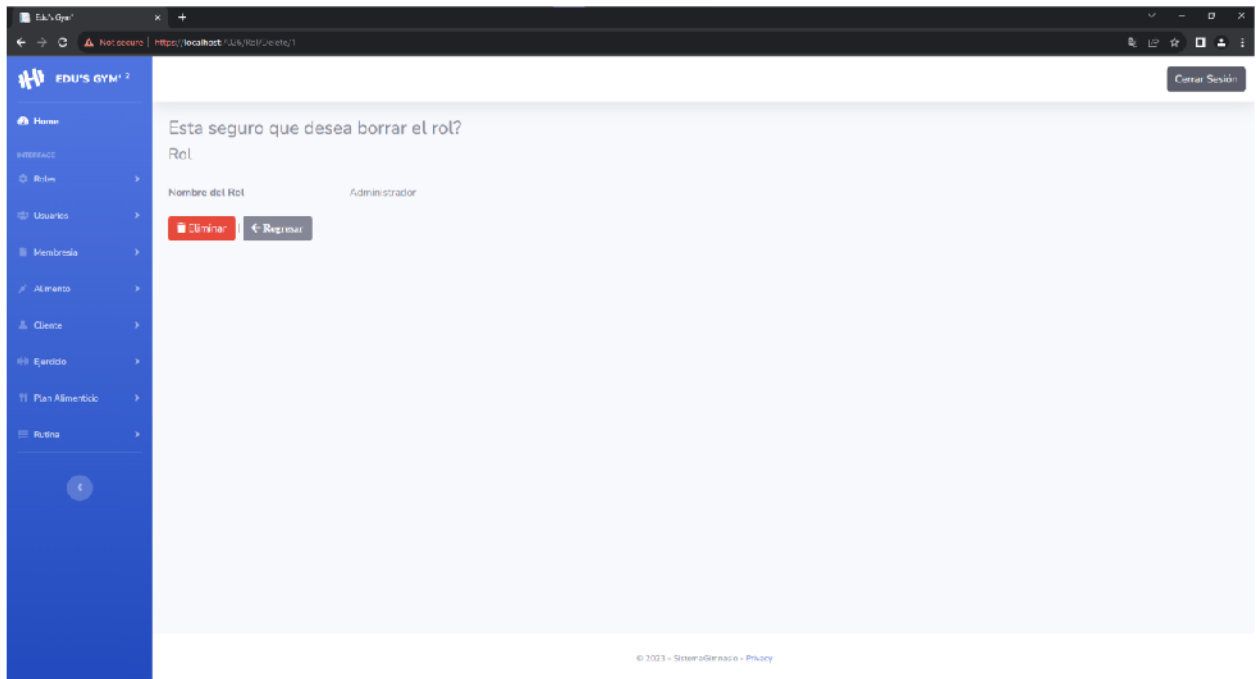
Nota. El Administrador puede ingresar un nuevo Rol y guardarlo en la tabla de roles

Figura 25. Interfaz de detalle de Rol



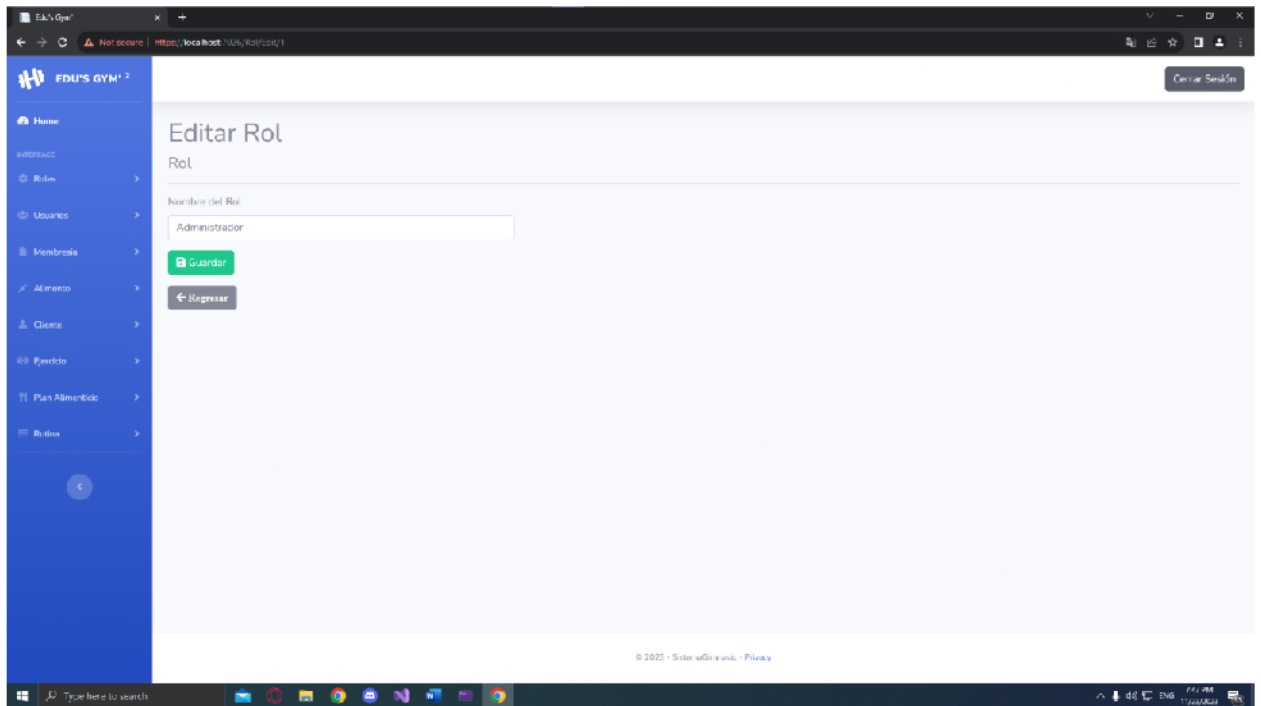
Nota. El administrador puede editar un rol que ya existe y guardarlo en la tabla de roles

Figura 26. Interfaz de eliminar Rol



Nota. El administrador puede eliminar un rol ya existente.

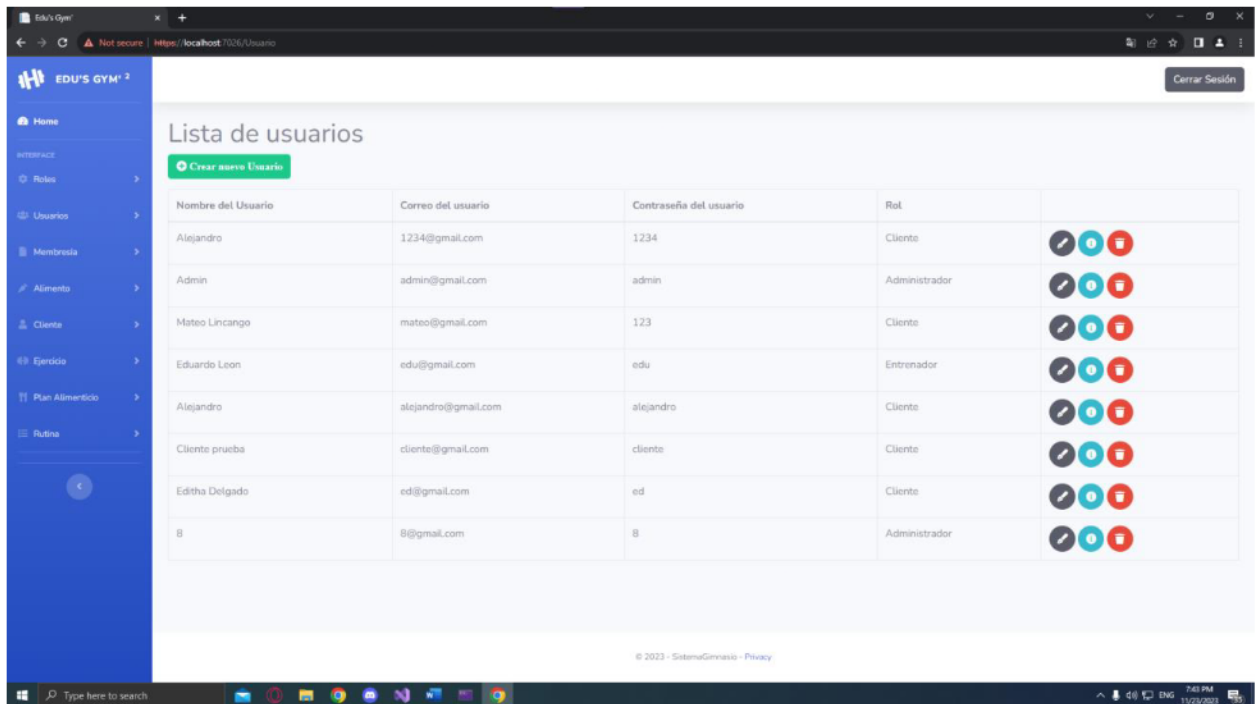
Figura 27. Interfaz editar rol



Nota. El administrador puede editar un rol ya existente

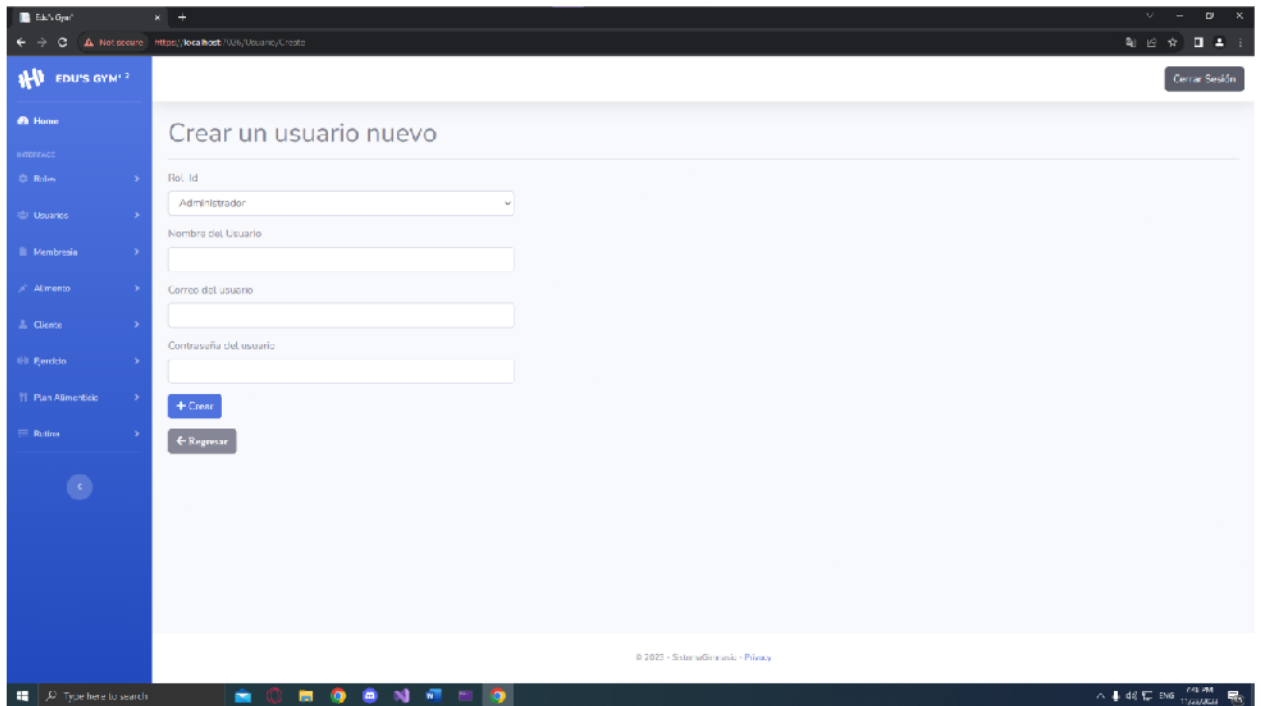
5.3.2. Gestión Usuarios

Figura 28. Interfaz de lista de usuarios



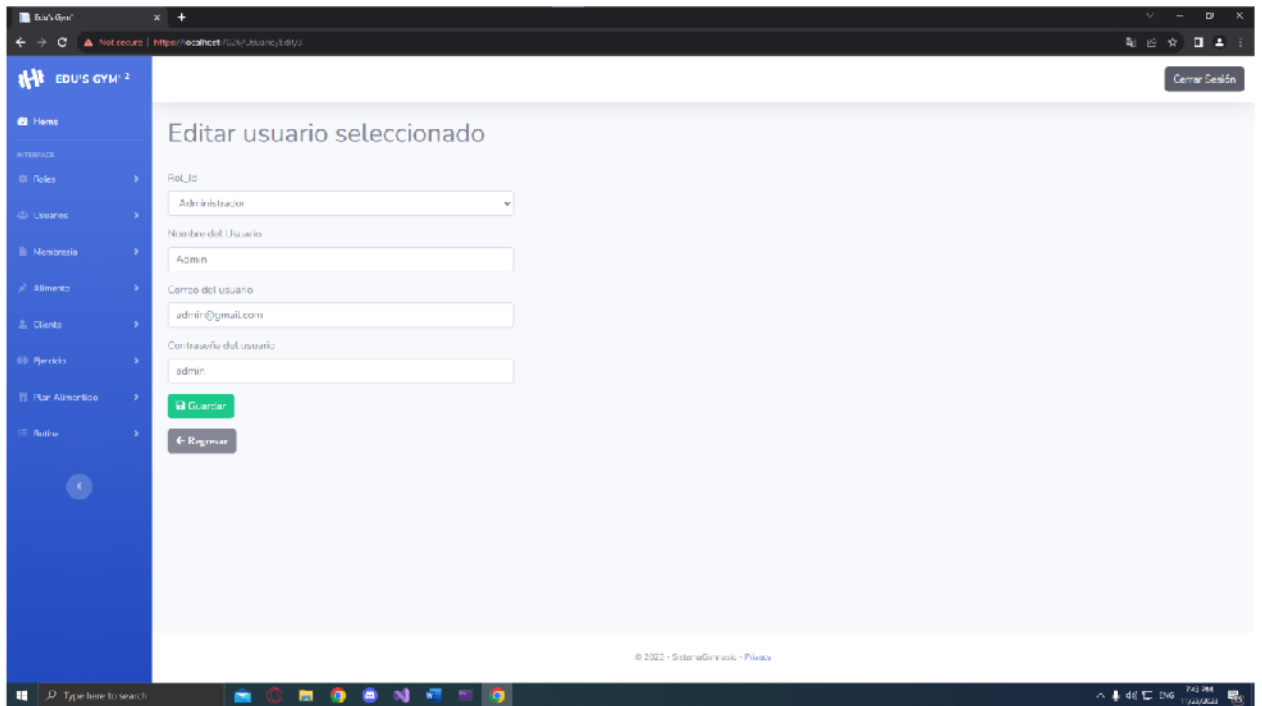
Nota El administrador puede observar la lista de clientes del sistema. Los iconos de la derecha permiten editar, ver la información o eliminar el registro seleccionado.

Figura 29. Interfaz crear nuevo usuario



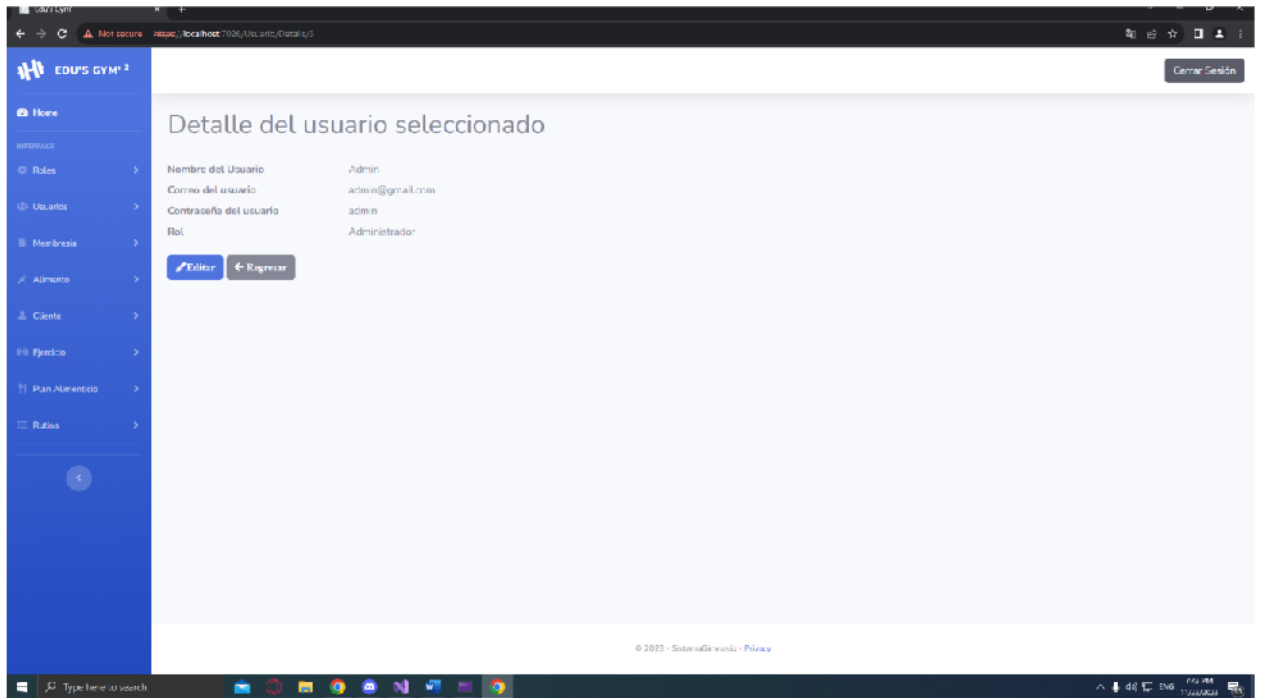
Nota. El administrador puede crear un usuario nuevo

Figura 30. Interfaz de editar usuario seleccionado



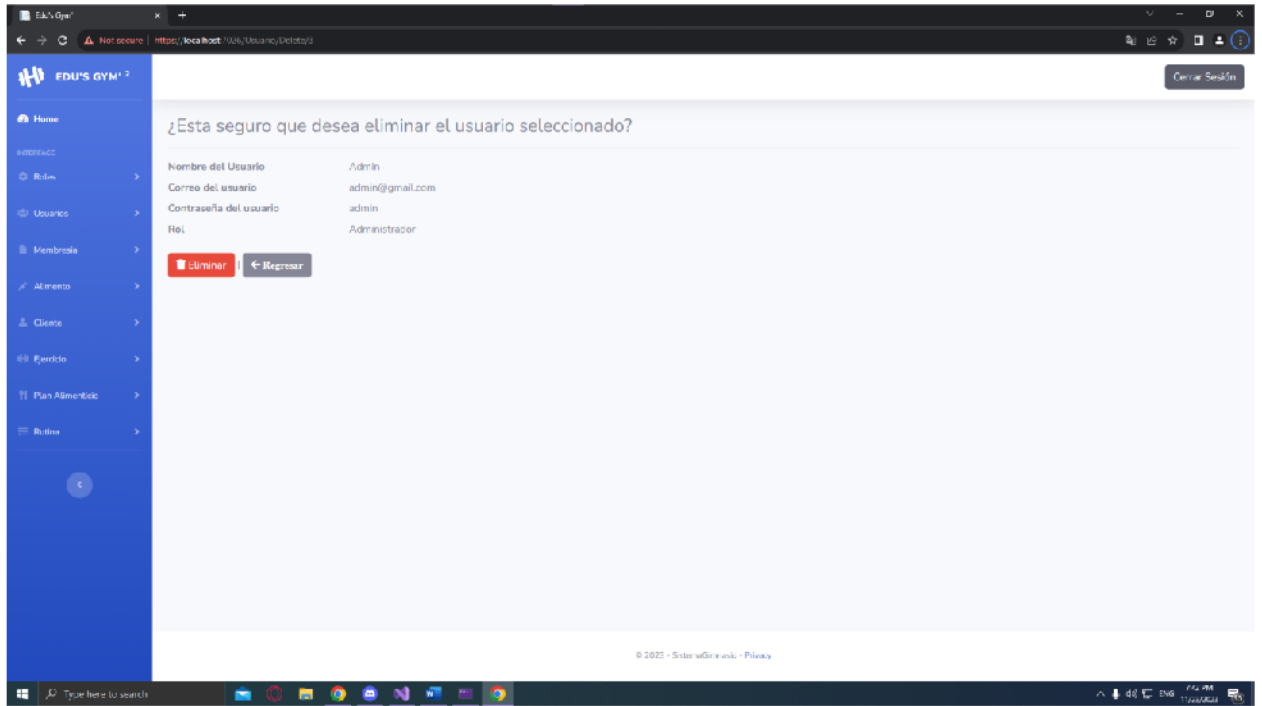
Nota. El administrador puede editar a un usuario existente

Figura 31. Interfaz de detalle de usuario seleccionado



Nota. El administrador puede observar los detalles de un usuario que el seleccione

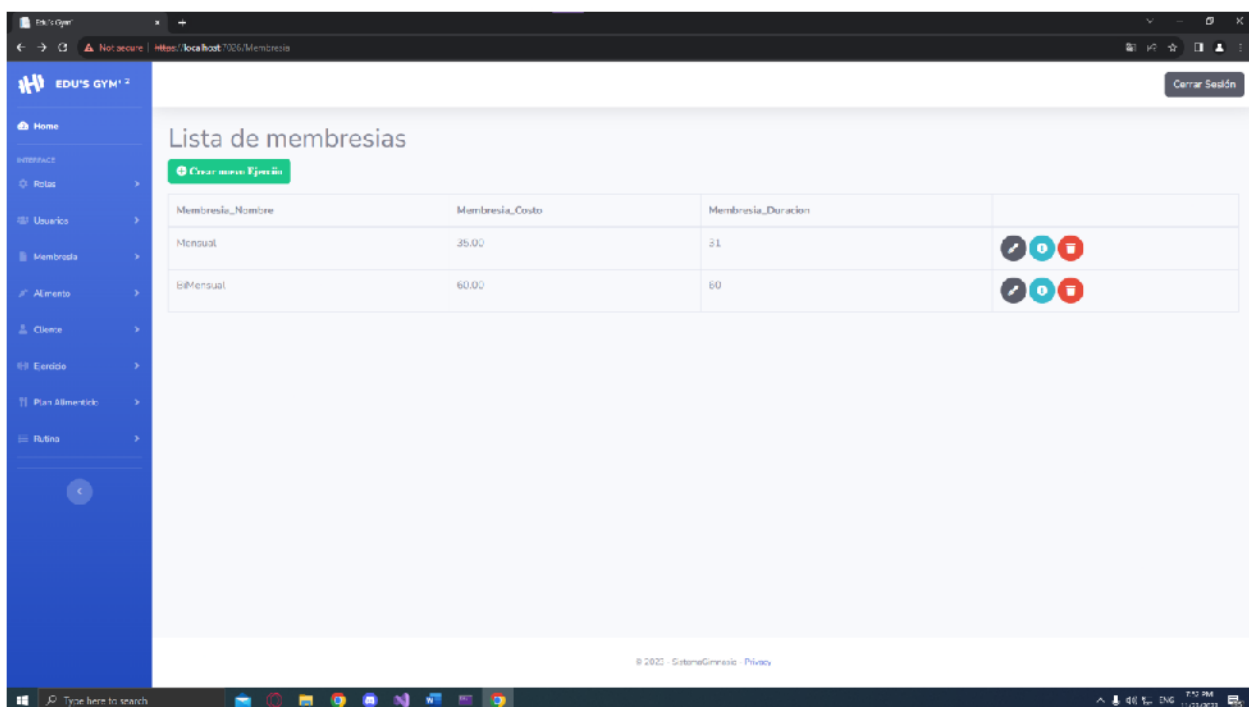
Figura 32. Interfaz de eliminar usuario seleccionado



Nota. El usuario puede eliminar a un cliente existente

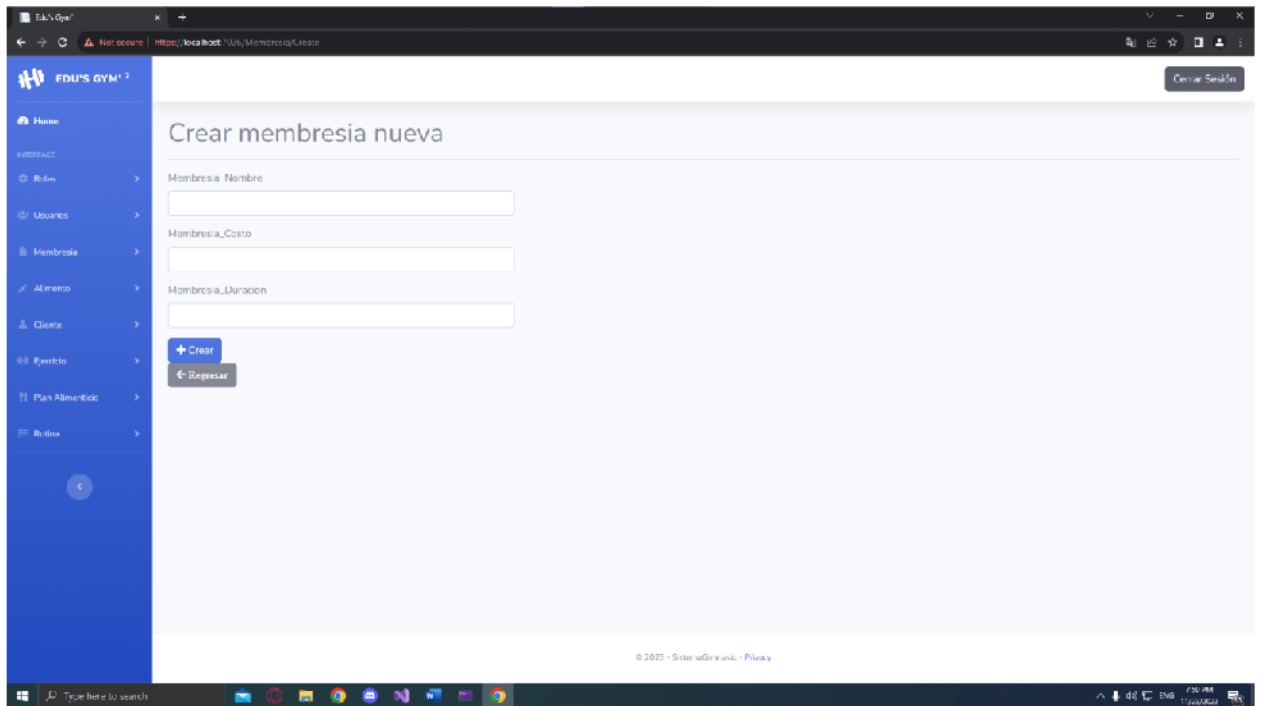
5.3.3. Gestión de Membresías

Figura 33. Interfaz de Lista de membresías



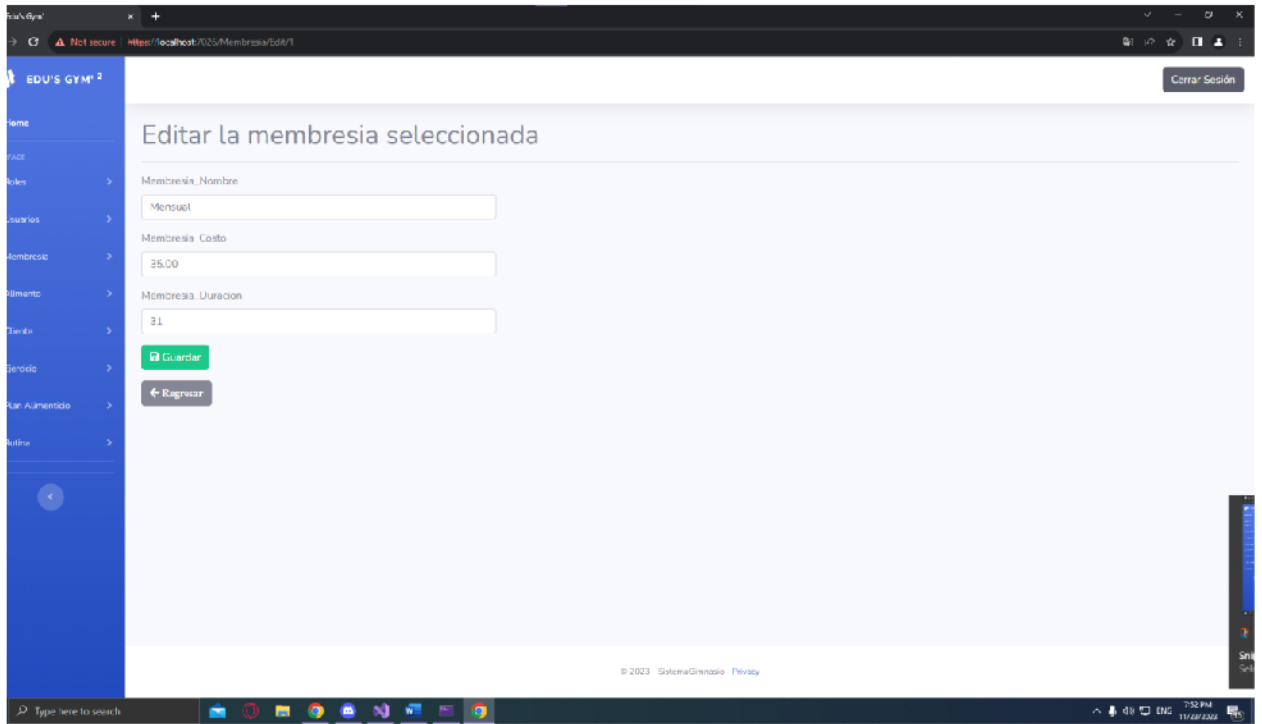
Nota. El administrador puede acceder al listado de membresías existen. Los iconos de la derecha permiten editar, ver la información o eliminar el registro seleccionado.

Figura 34. Interfaz de crear membresía



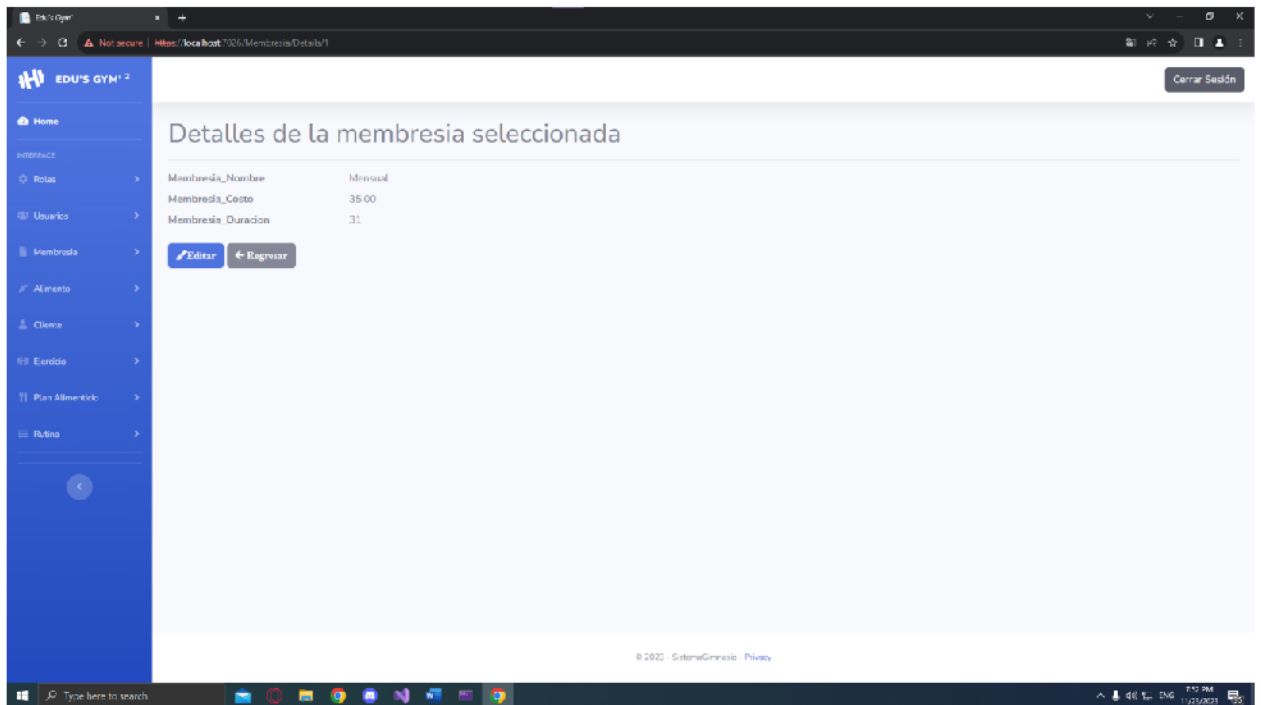
Nota. El administrador puede crear una nueva membresía.

Figura 35. Interfaz de editar membresía seleccionada

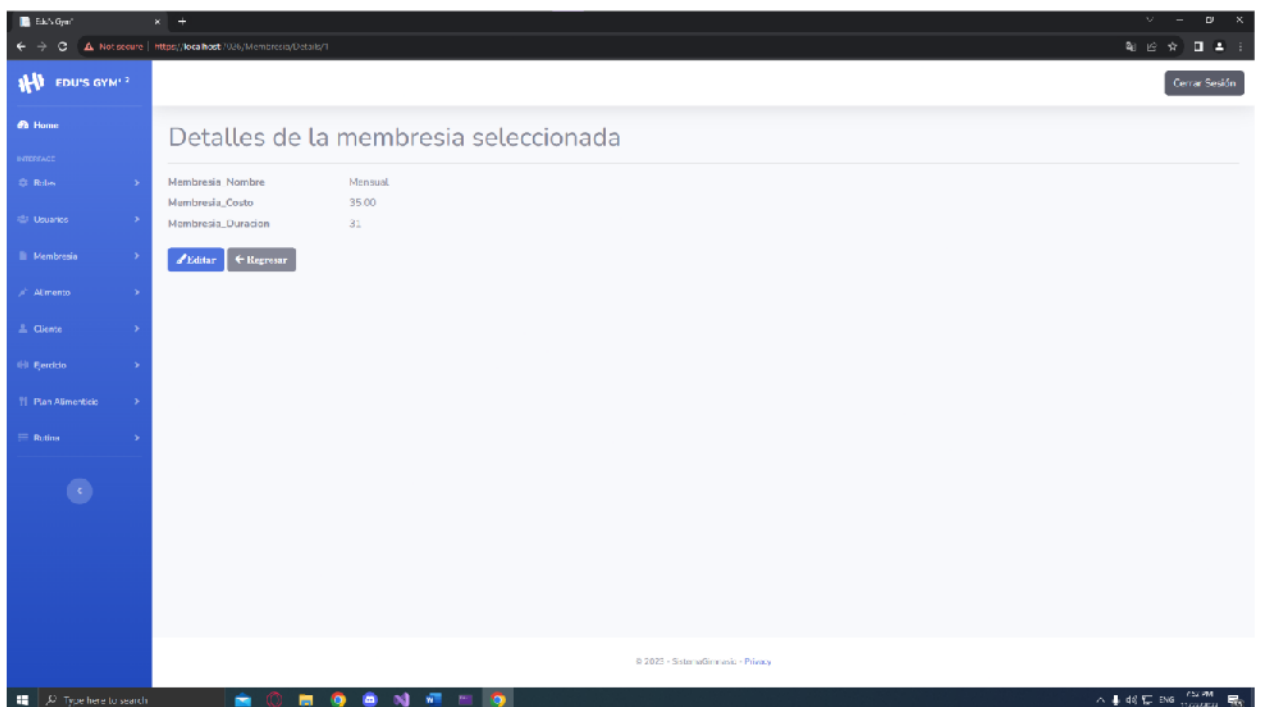


Nota. El administrador puede editar una membresía existente

Figura 36. Interfaz de Detalle membresia

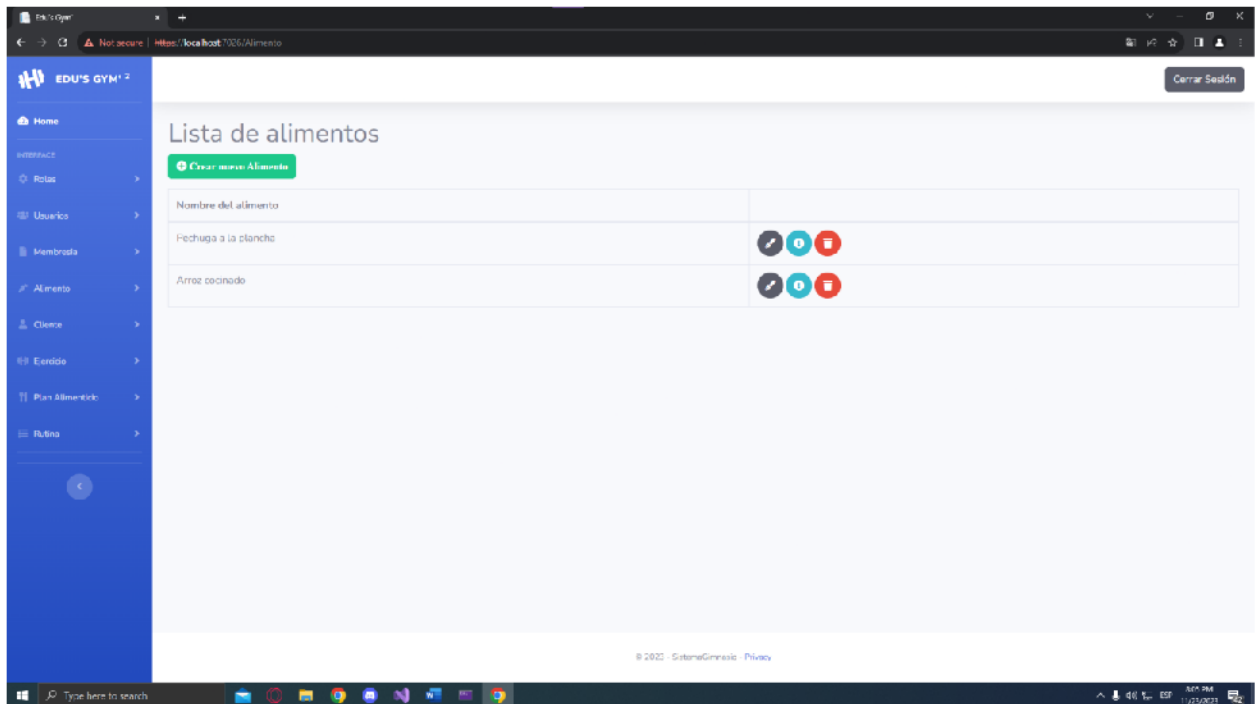


Nota. El administrador puede eliminar la membresia seleccionada.



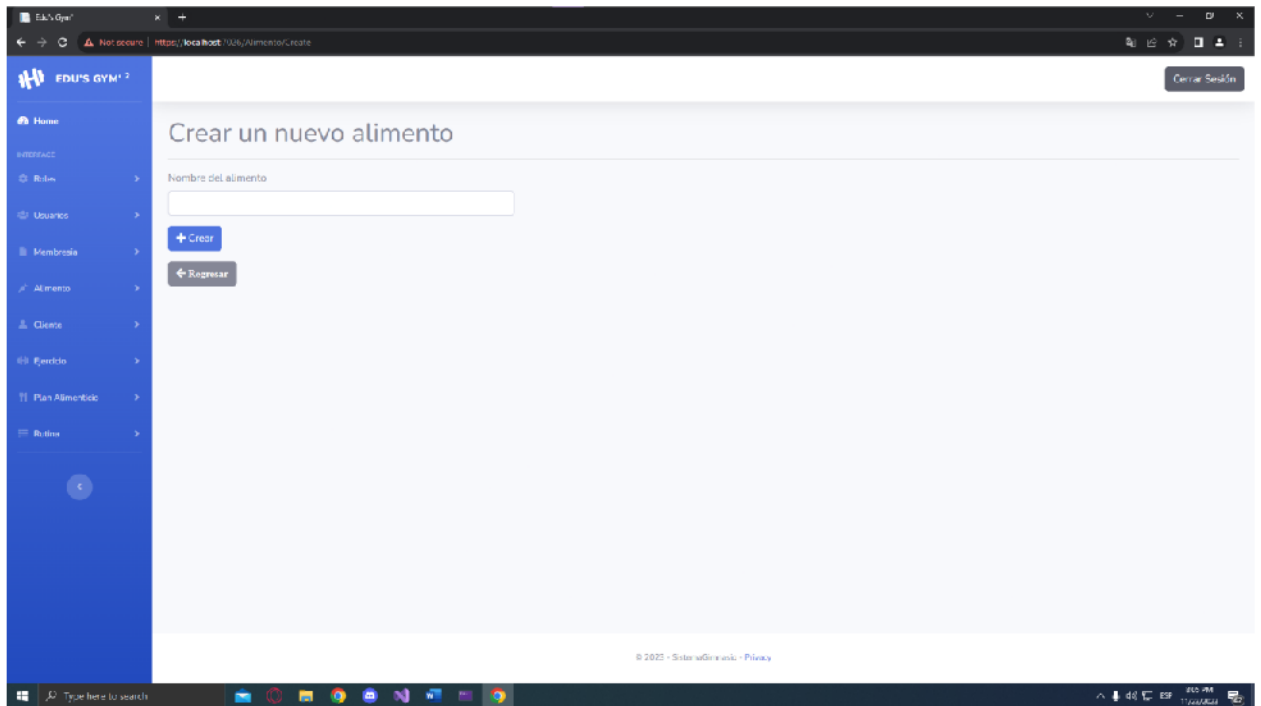
5.3.4. Gestión de Alimentos

Figura 37. Interfaz de lista de alimentos



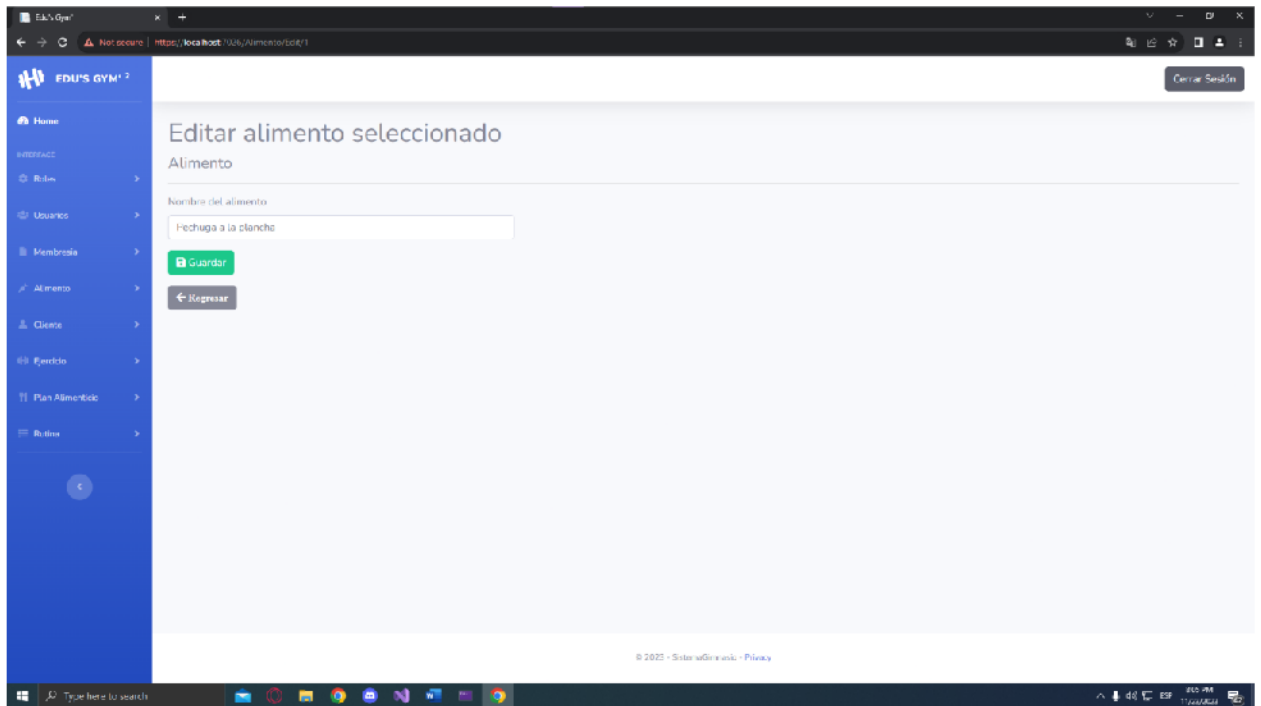
Nota. El administrador y entrenador pueden ver los alimentos existen. Las acciones de la derecha nos permiten entrar en nuestras operaciones CRUD. Estos alimentos pueden ser añadidos a un plan alimenticio.

Figura 38. Interfaz de crear un nuevo alimento



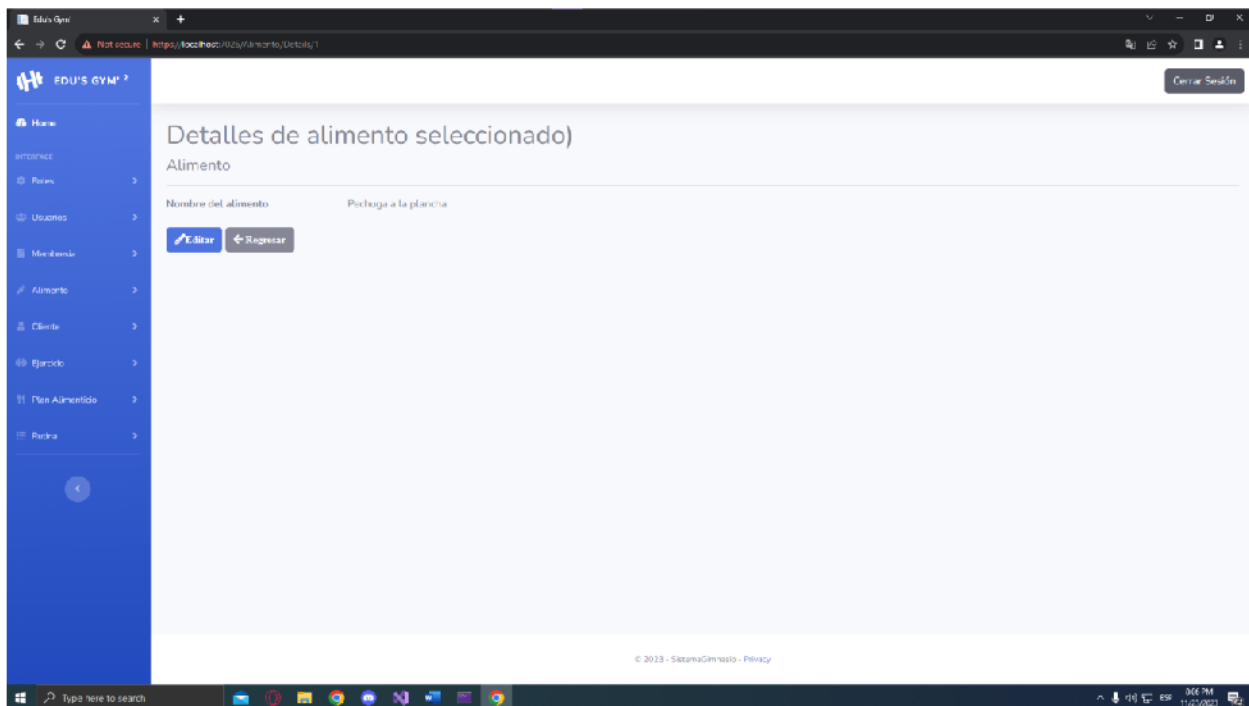
Nota. El administrador y entrenador pueden crear un nuevo alimento.

Figura 39. Interfaz de editar alimento



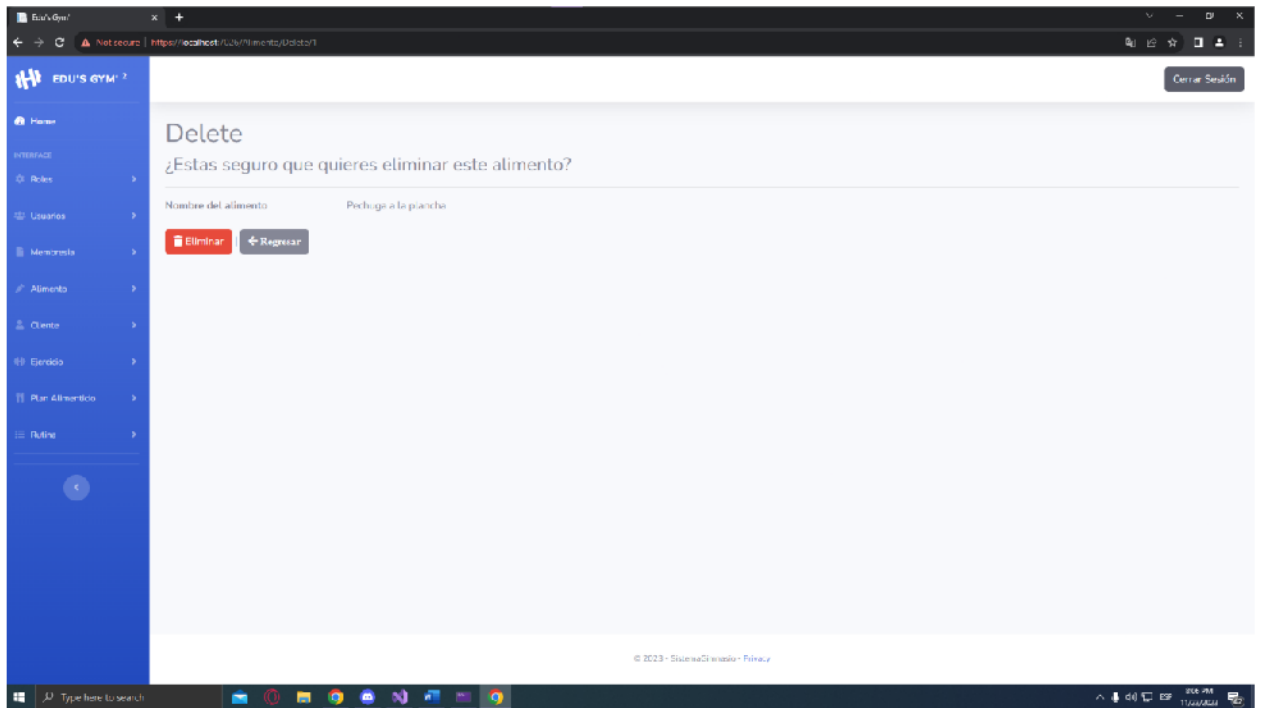
Nota. El administrador y entrenador pueden editar el alimento seleccionado.

Figura 40. Interfaz de detalle de alimento



Nota. El administrador y entrenador pueden ver los del alimento seleccionado.

Figura 41. Interfaz de eliminar alimento



Nota. El administrador y entrenador pueden eliminar el alimento seleccionado.

5.3.5. Gestión de Clientes

Figura 42. Interfaz de lista de Clientes

Peso del cliente(kg)	Estatura del cliente(m)	Imagen del cliente	ClienteFecIni	ClienteFecFin	Membresía	Pa	Rutina	Usuario	Acciones
8.00	8.00	ed	11/23/2023	11/22/2023	Mensual	Lunes Ectomorfos	Pecho y Triceps1	ed@gmail.com	[Edit] [Add] [Delete]
6.00	6.00	cliente	11/15/2023	11/16/2023	Mensual	Lunes Ectomorfos	Pierna3	cliente@gmail.com	[Edit] [Add] [Delete]
5.00	1.73	123	11/15/2023	11/30/2023	Mensual	Lunes Ectomorfos	Espalda y biceps2	1234@gmail.com	[Edit] [Add] [Delete]
6.00	6.00	6	11/23/2023	11/21/2023	Mensual	Lunes Ectomorfos	prueba7	aljandro@gmail.com	[Edit] [Add] [Delete]
1.00	1.73	Mateo	11/20/2023	12/26/2023	Mensual	Lunes Ectomorfos	prueba	mateo@gmail.com	[Edit] [Add] [Delete]

Nota. El administrador y entrenador pueden ver los clientes existentes. Las acciones de la derecha nos permiten entrar en nuestras operaciones CRUD.

Figura 43. Interfaz de crear un nuevo cliente

The screenshot shows a web browser window with the URL `http://localhost:7000/Clientes/Crear`. The page title is "Crear nuevo Cliente". On the left, there is a blue sidebar with a "Home" button and a list of menu items: "Roles", "Usuarios", "Membresía", "Alimento", "Cliente", "Estado", "Plan Alimenticio", and "Rutina". The main content area contains a form with the following fields:

- Rutina diaria:
- Plan alimenticio diario:
- Membresía del cliente:
- Correo:
- Peso del cliente:
- Estatura del cliente:
- Imagen del cliente:
- ClienteEctm:
- ClienteEctm:

At the bottom of the form is a blue button labeled "+ Crear".

Nota. El administrador y entrenador pueden crear un nuevo cliente. Se puede asignar al cliente una rutina, un plan alimenticio, membresía y un usuario.

Figura 44. Interfaz de editar cliente

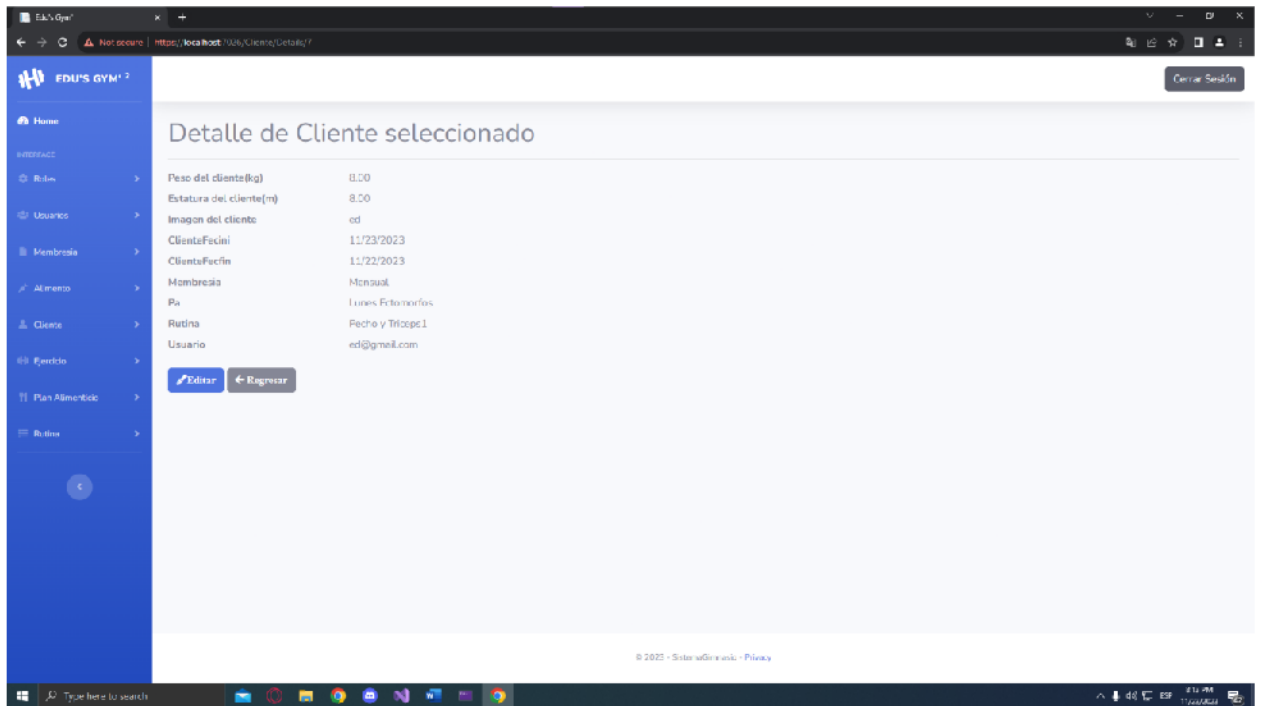
The screenshot shows a web browser window with the URL `https://localhost:3000/cliente/edit/7`. The page title is "Editar cliente seleccionado". On the left, there is a blue sidebar menu with the following items: "Inicio", "Roles", "Usuarios", "Membresías", "PAGO", "Clientes", "Ejercicios", "Plan Alimenticio", and "Huban". The main content area contains a form with the following fields and values:

- Huban diaria: Pecho y Tríceps 1
- Plan alimenticio diario: Lunes-Ectomorfo
- Membresía del cliente: Mensual
- Correo: ed1@gmail.com
- Peso del cliente(kg): 8.00
- Estatura del cliente(m): 8.00
- Imagen del cliente: ed
- Cliente:Inicio: 11/23/2023
- Cliente:Fin: 11/22/2023

At the bottom of the form, there are two buttons: "Guardar" (green) and "Regresar" (grey).

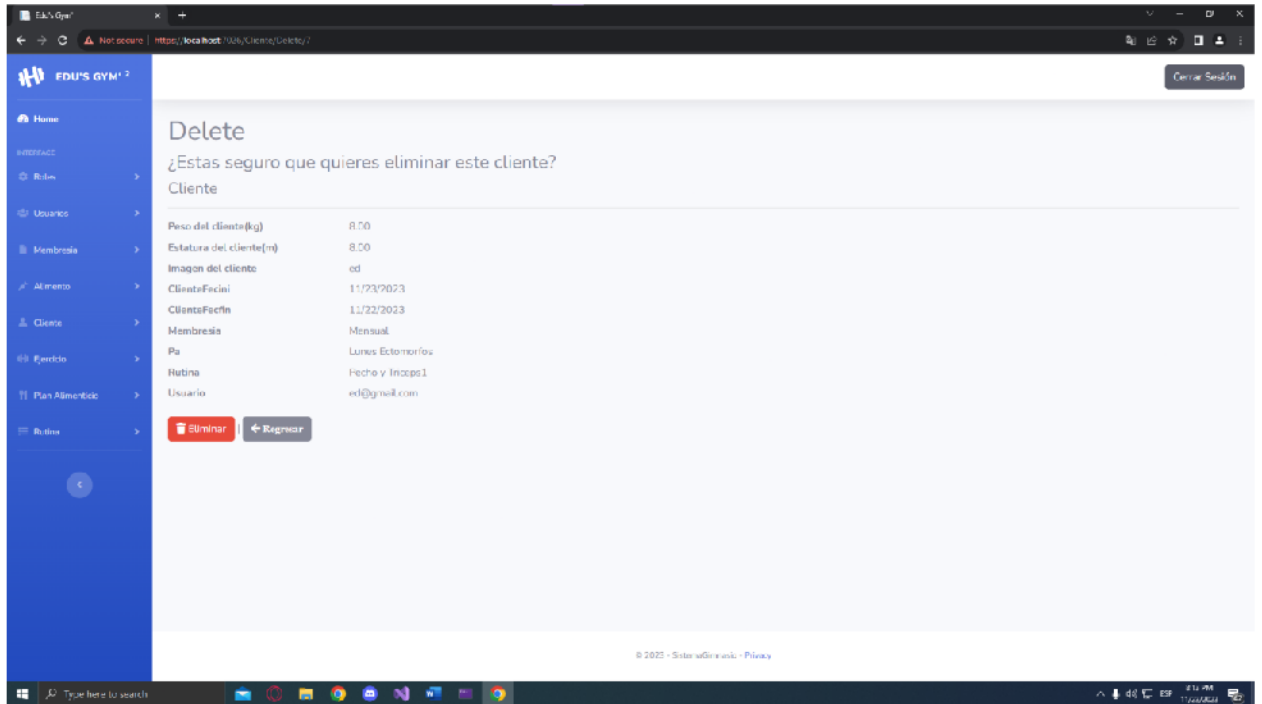
Nota. El administrador y entrenador pueden editar la información del cliente seleccionado.

Figura 45. Interfaz de detalle de cliente



Nota. El administrador y entrenador pueden ver los del cliente seleccionado.

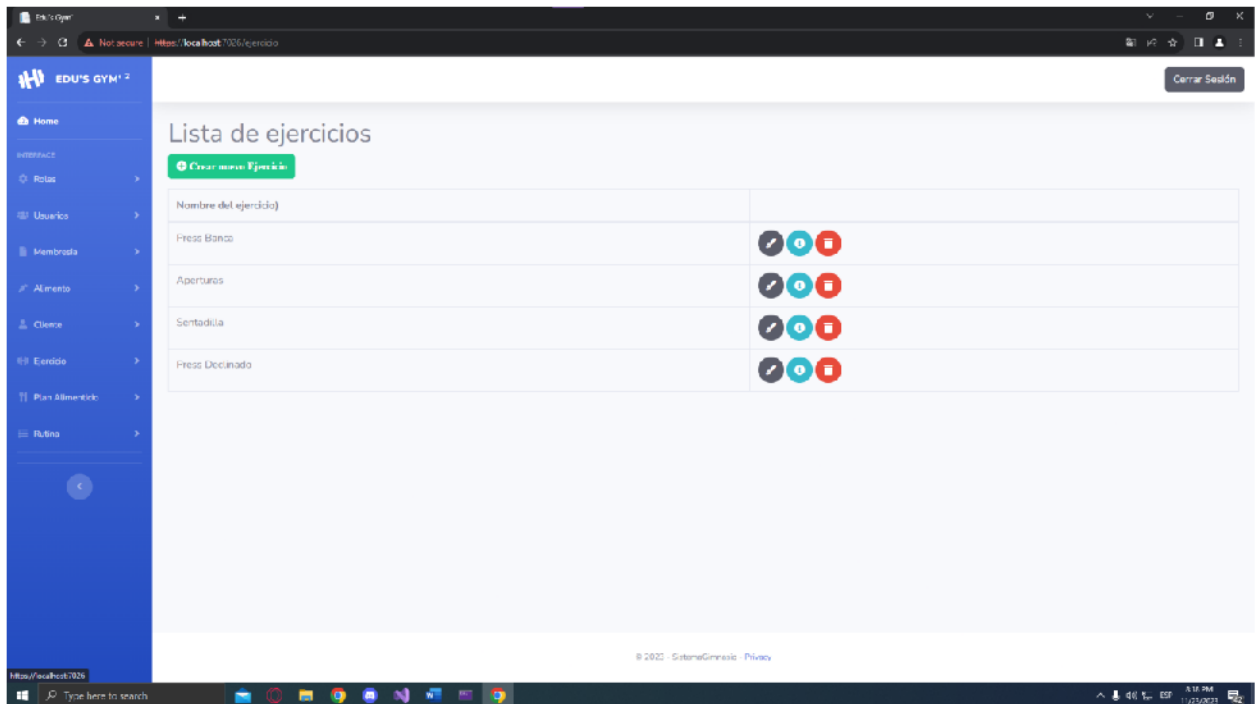
Figura 46. Interfaz de eliminar cliente



Nota. El administrador y entrenador pueden eliminar el cliente seleccionado.

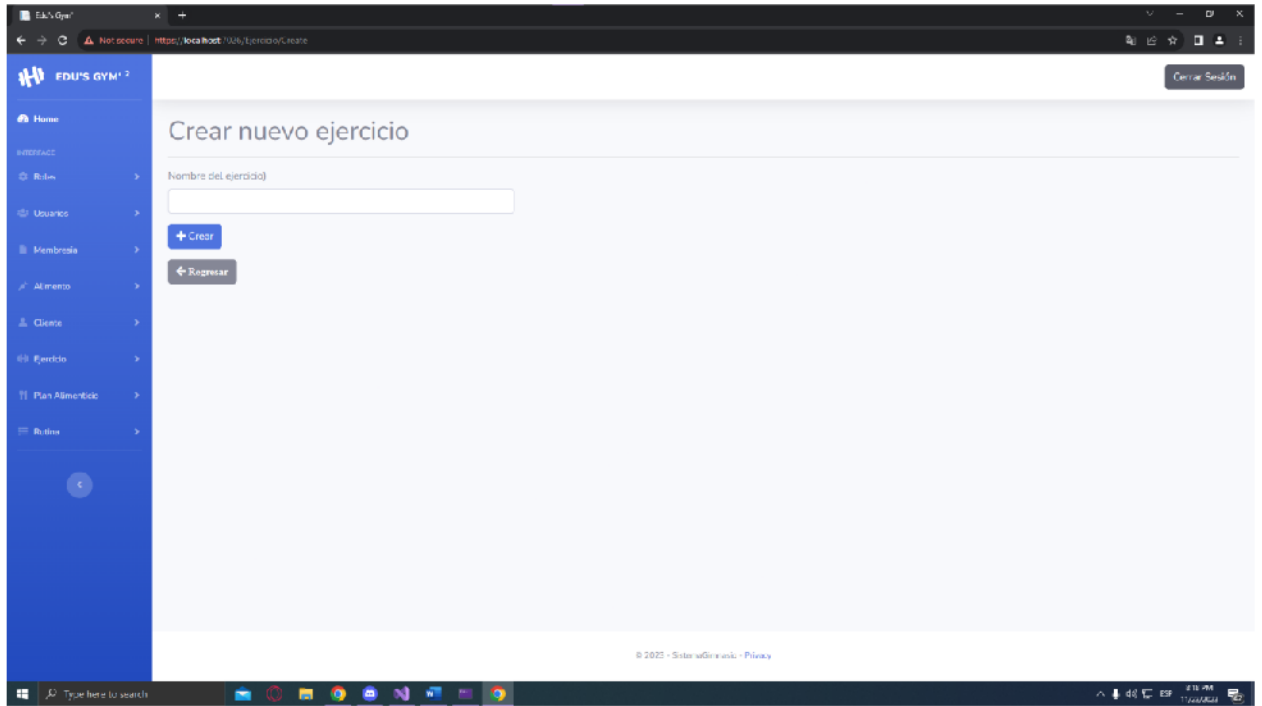
5.3.6. Gestión de Ejercicios

Figura 47. Interfaz de lista de ejercicios



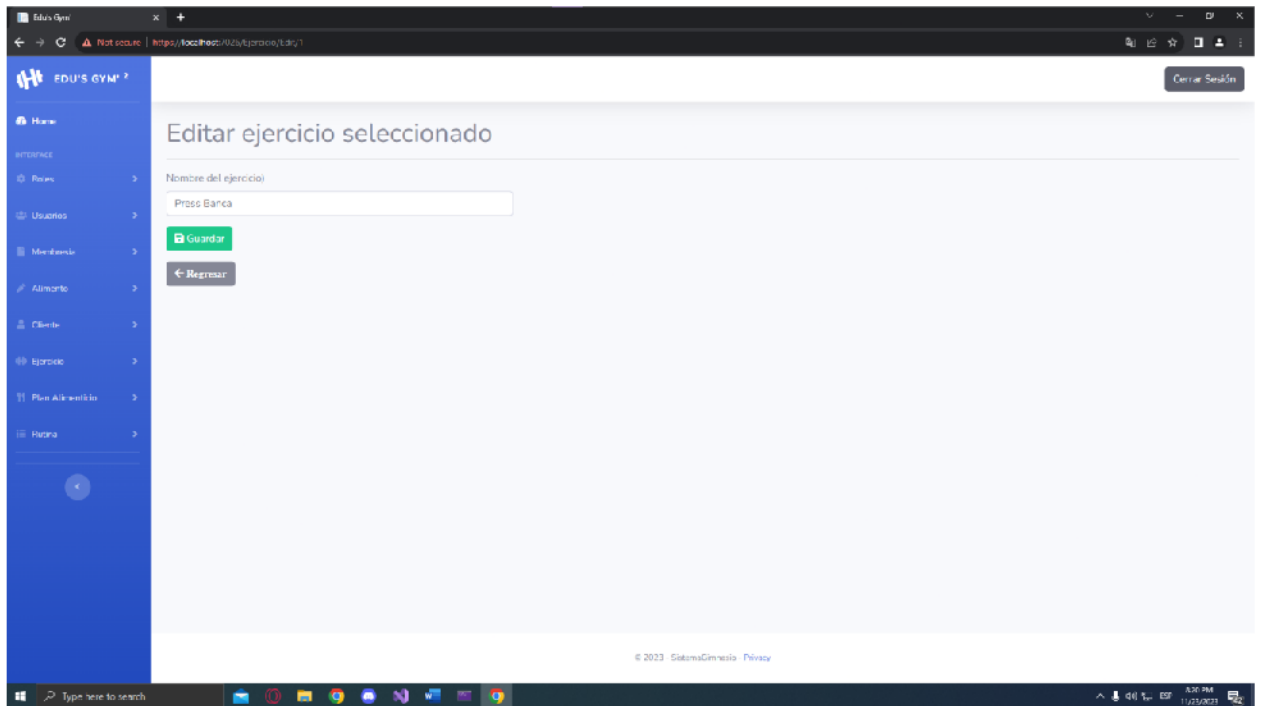
Nota. El administrador y entrenador pueden ver los ejercicios existen. Las acciones de la derecha nos permiten entrar en nuestras operaciones CRUD. Los ejercicios pueden ser añadidos a una rutina

Figura 48. Interfaz de crear un nuevo ejercicio



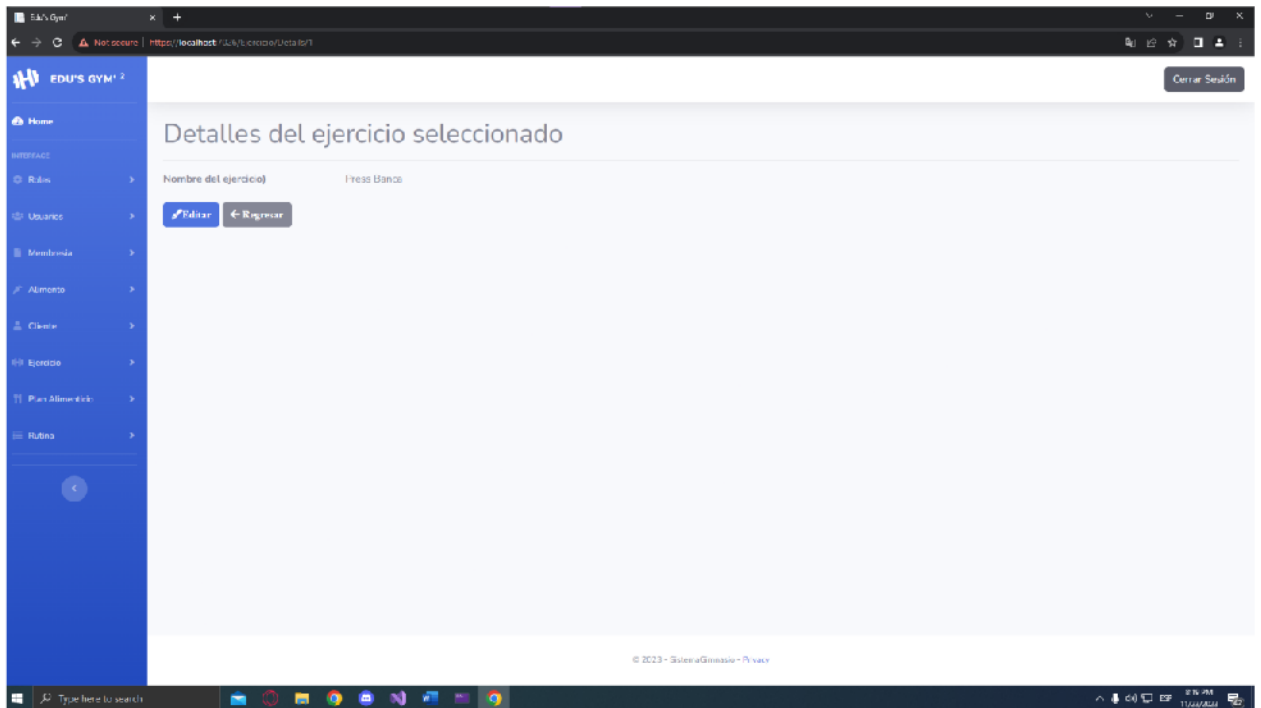
Nota. El administrador y entrenador pueden crear un nuevo ejercicio.

Figura 49. Interfaz de editar ejercicio



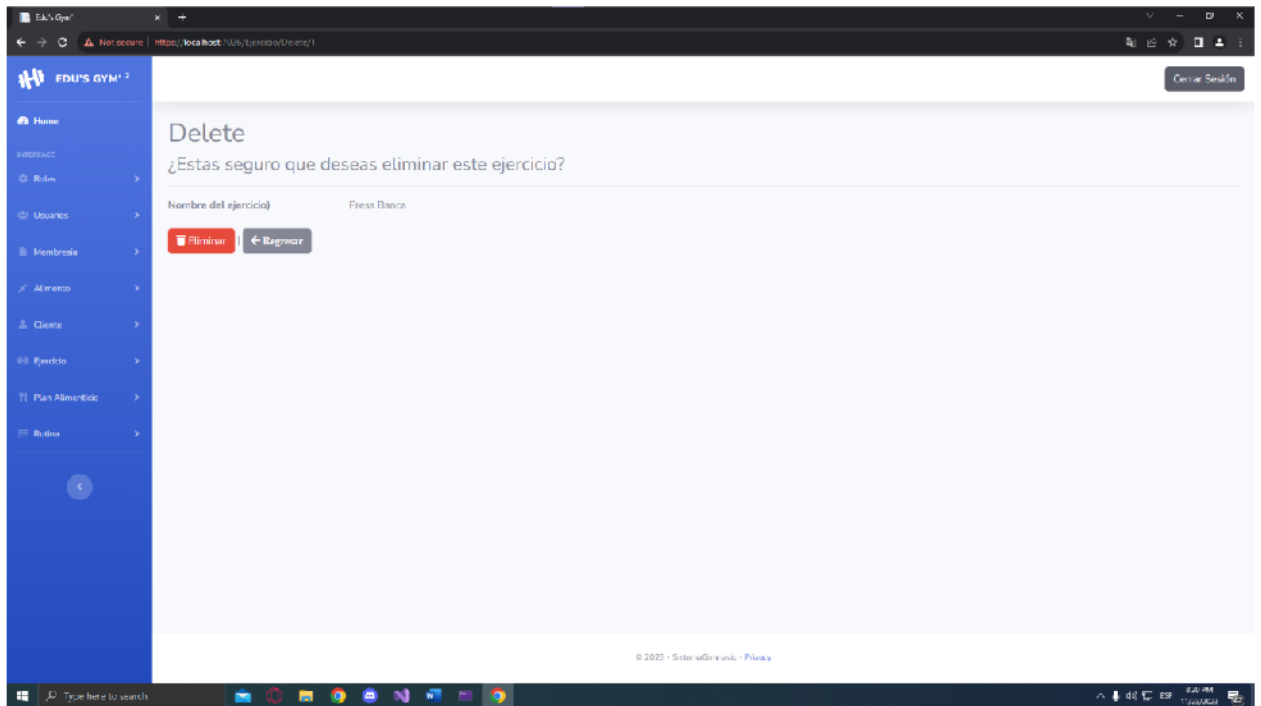
Nota. El administrador y entrenador pueden editar el ejercicio seleccionado.

Figura 50. Interfaz de detalle de ejercicio



Nota. El administrador y entrenador pueden ver los detalles del ejercicio seleccionado.

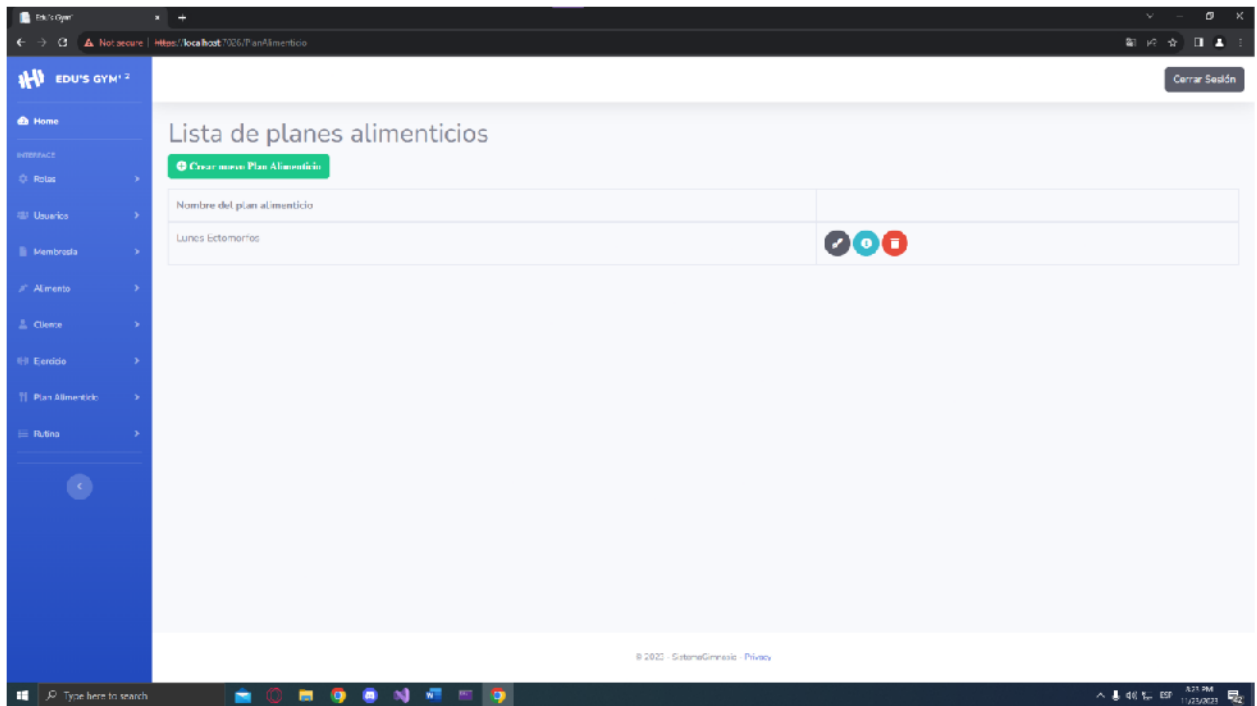
Figura 51. Interfaz de eliminar ejercicio



Nota. El administrador y entrenador pueden eliminar el ejercicio seleccionado.

5.3.7. Gestión de Planes de Alimenticios

Figura 52. Interfaz de lista de planes alimenticios



Nota. El administrador y entrenador pueden ver los planes alimenticios que existen. Las acciones de la derecha nos permiten entrar en nuestras operaciones CRUD,

Figura 53. Interfaz de crear un nuevo plan alimenticio

EDU'S GYM

Inicio

Crear un nuevo Plan Alimenticio

Nombre del plan alimenticio

El campo no puede estar vacío

Detalles del pedido

Alimento	Porcion	Cantidad
Pechuga o la planche	<input type="text"/>	<input type="text"/>
Amar cocinado	<input type="text"/>	<input type="text"/>

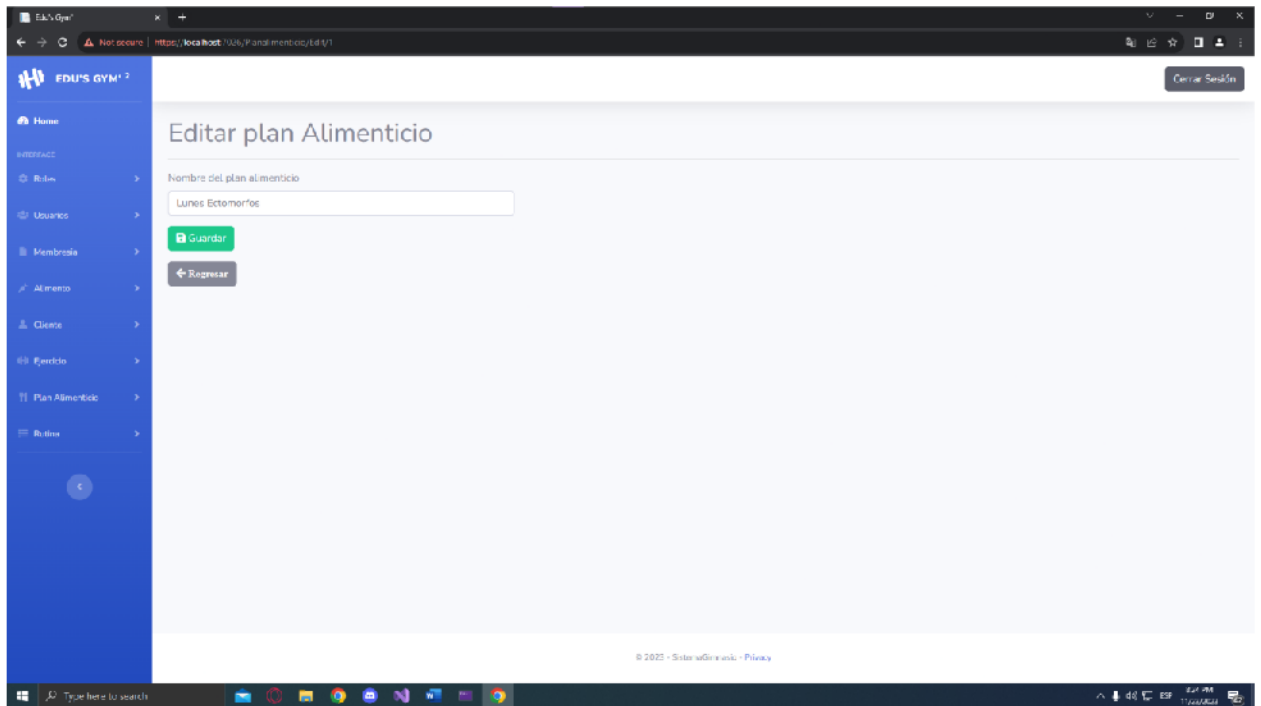
+ Crear

< Regresar

© 2021 - SistemaGym - Privado

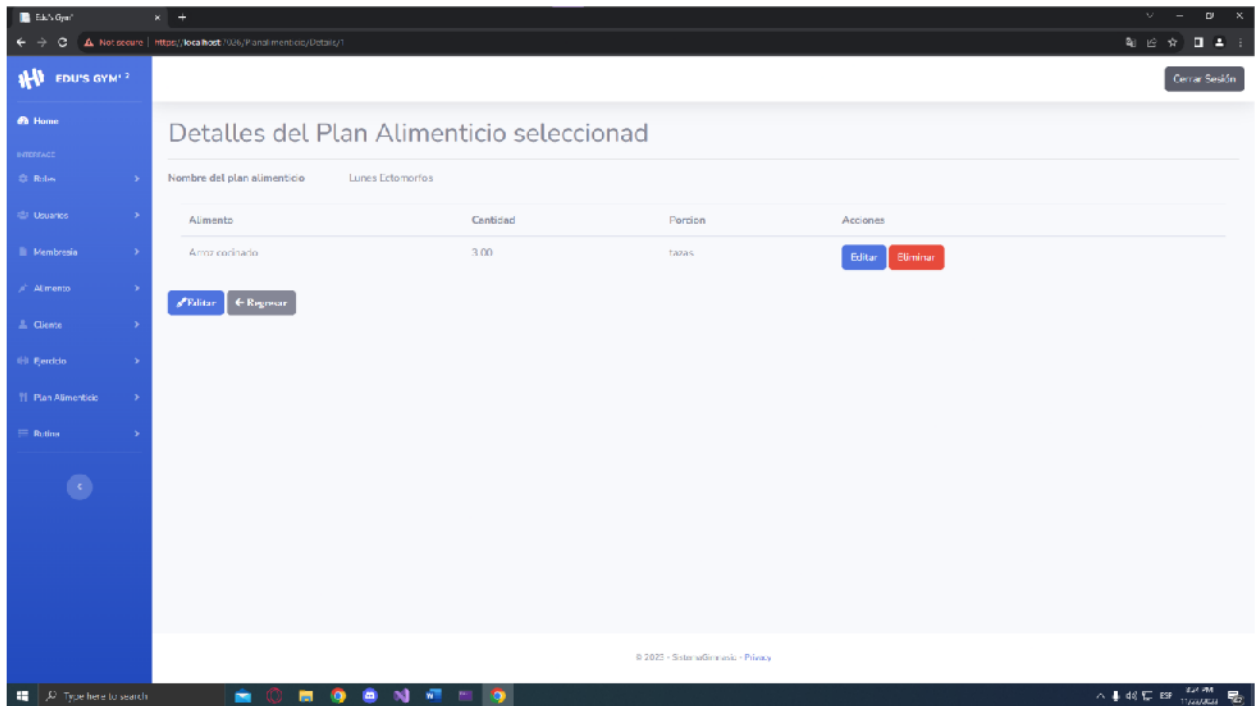
Nota. El administrador y entrenador pueden crear un nuevo plan alimenticio el cual va a llevar los alimentos previamente creados, cada alimento tiene una porción (por ejemplo gramos) y una cantidad(1,2,3....).

Figura 54. Interfaz de editar plan alimenticio



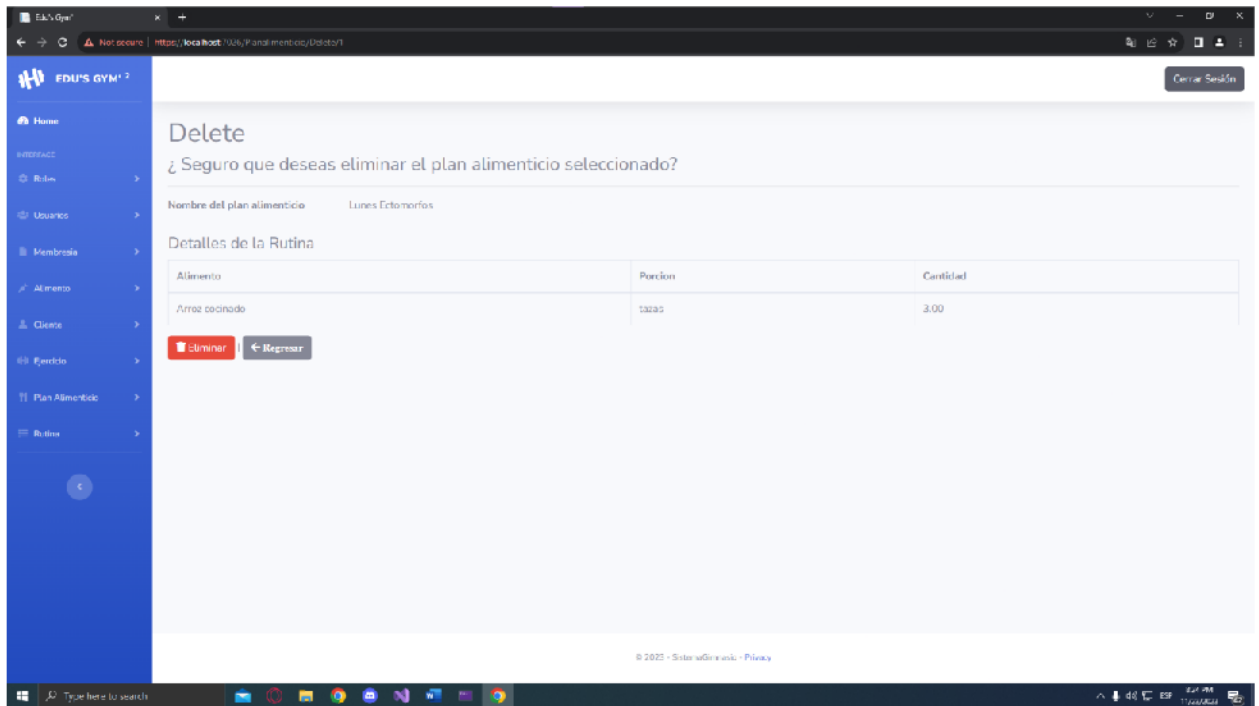
Nota. El administrador y el entrenador puede editar el nombre del plan alimenticio

Figura 55. Interfaz de detalle de plan alimenticio



Nota. El administrador y entrenador puede ver los alimentos que contiene ese plan alimenticio.

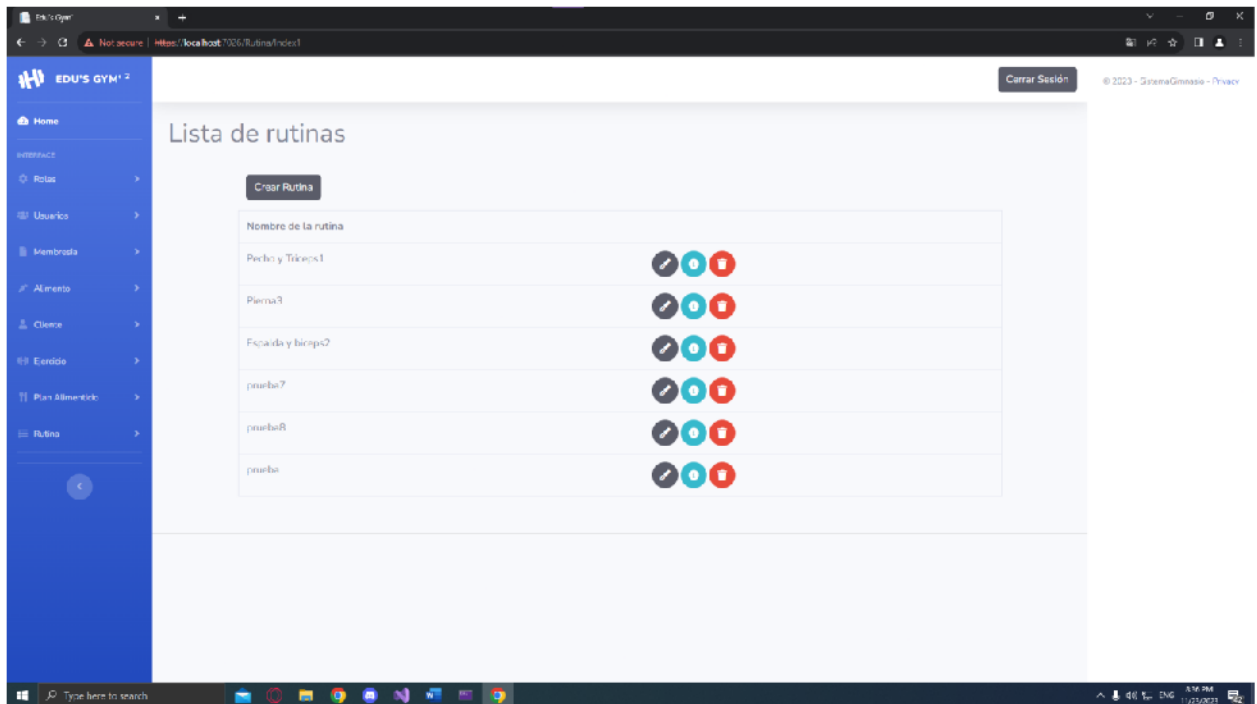
Figura 56. Interfaz de eliminar plan alimenticio



Nota. El administrador y entrenador puede eliminar el plan alimenticio seleccionado. Se muestra los alimentos que contiene este plan alimenticio

5.3.8. Gestión de Rutinas

Figura 57. Interfaz de lista de rutinas



Nota. El administrador y entrenador pueden ver las rutinas que existen. Las acciones de la derecha nos permiten entrar en nuestras operaciones CRUD.

Figura 58. Interfaz de crear una nueva rutina

EDU'S GYM

Home

Roles

Usuarios

Membresía

Alimento

Clases

Ejercicios

Price Almacenado

Rutina

Crear una nueva Rutina

Nombre de la rutina

El campo no puede estar vacío

Detalles del pedido

Ejercicio	Series	Repeticiones
Press Banca	<input type="text"/>	<input type="text"/>
Aperturas	<input type="text"/>	<input type="text"/>
Sentadilla	<input type="text"/>	<input type="text"/>
Press Declinado	<input type="text"/>	<input type="text"/>

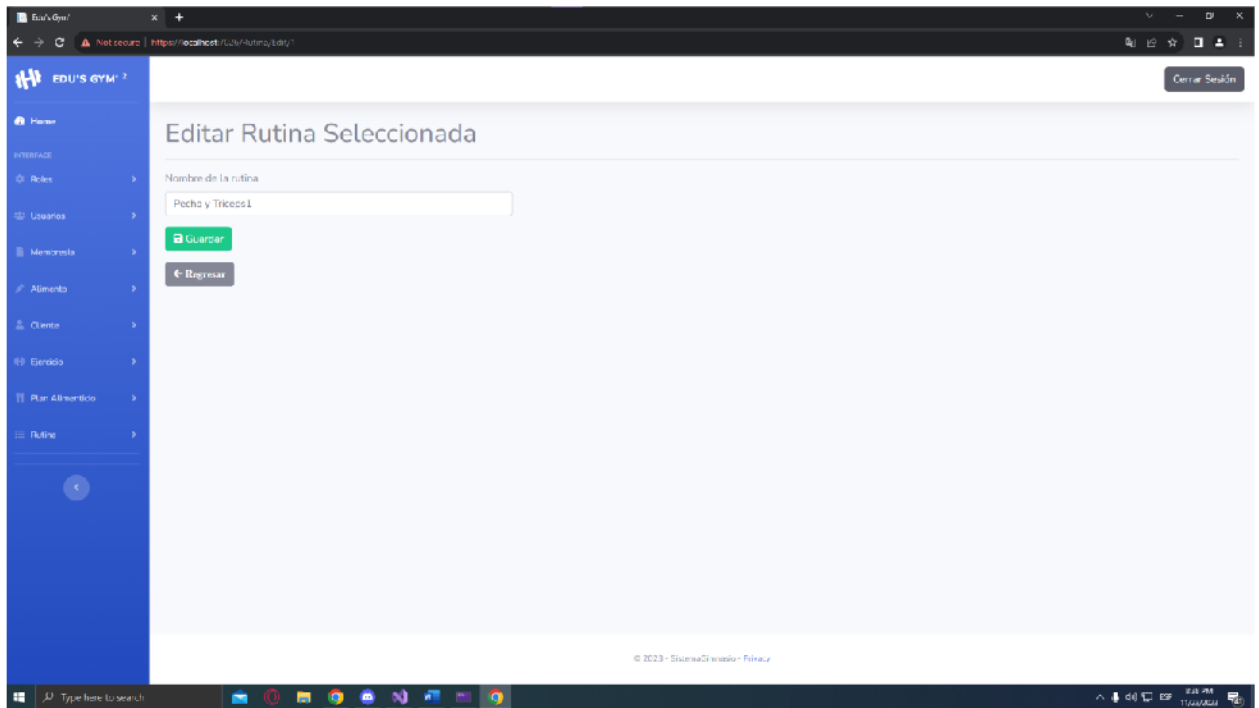
+ Crear

← Regresar

© 2023 - Sistema Dinámico - Privacy

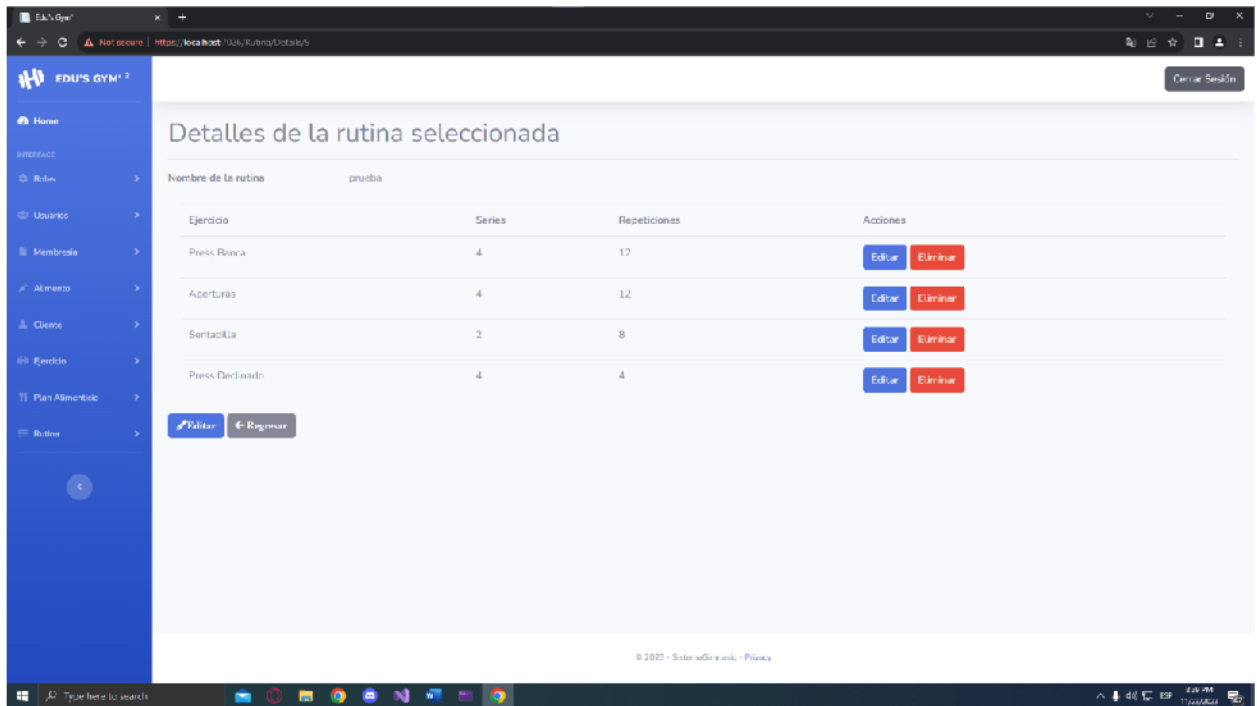
Nota. El administrador y entrenador pueden crear una nueva rutina la cual va a llevar los ejercicios previamente creados, cada ejercicio tiene un numero de repeticiones y un numero de series.

Figura 59. Interfaz de editar rutina



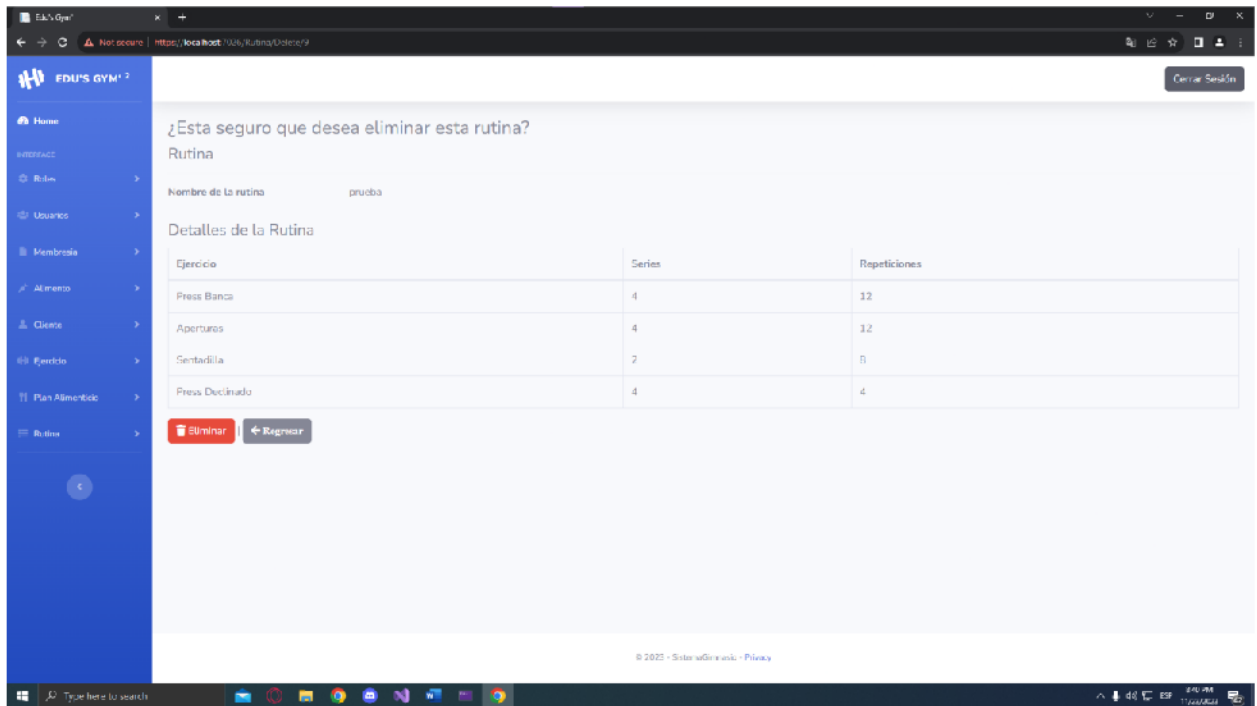
Nota. El administrador y el entrenador puede editar el nombre de la rutina.

Figura 60. Interfaz de detalle de rutina



Nota. El administrador y entrenador puede ver los ejercicios que contiene esa rutina.

Figura 61. Interfaz de eliminar plan alimenticio



Nota. El administrador y entrenador puede eliminar la rutina seleccionada. Se muestra los ejercicios que contiene esta rutina.

5.4. Pruebas funcionales.

Las pruebas funcionales son una forma de determinar si el software funciona correctamente. Hay varios tipos de pruebas funcionales, incluidas unitarias, integradas, componentes y de humo. (Vargas, s.f)

5.4.1. Pruebas de humo.

Esta prueba se realiza manualmente para verificar que las funcionalidades críticas e importantes del sistema funcionan correctamente. Esta prueba es crucial para garantizar que el software sea estable y funcional. Realizar este tipo de pruebas es beneficioso porque es

realizado por desarrolladores, ahorra tiempo, tarda poco y comprueba completamente todo el software. (Testbytes, 2019)

5.4.2. Prueba de Humo para Sistema Web para el gimnasio de Edu's Gym

Tabla 33. Tabla de prueba de humo Sistema Web Edu's Gym

ID	Descripción	Pasos	Resultado esperado	Resultado final	Estado
1	Inicio de Sesión	<ol style="list-style-type: none"> Llenar cada campo con las credenciales correspondientes. Pulsar el botón en "Ingresar". 	Se muestra la home del sitio web.	El usuario ingresa correctamente al sistema.	Aprobado.
2	Visualización de clientes	<ol style="list-style-type: none"> Iniciar sesión como administrador 	Visualizar la lista de cliente.	El usuario visualizo la lista de los	Aprobado.

		<p>dor o como entrenador.</p> <p>2. Seleccionar en el menu "Cliente".</p> <p>3. Se muestra la lista de cliente</p>		<p>clientes existen.</p>	
3	<p>Creación de nuevo Cliente</p>	<p>1. Iniciar sesión como entrenador o administrador.</p> <p>2. Selecciona en el menú cliente.</p> <p>3. Presiona el botón Crear nuevo cliente</p>	<p>Crear un nuevo cliente</p>	<p>El usuario fue capaz de crear un nuevo cliente</p>	<p>Aprobado.</p>

		<p>4. Se despliega una nueva interfaz para agregar un cliente</p> <p>5. Se llena los campos correspondientes para crear un nuevo usuario.</p> <p>6. Pulsar el botón de “guardar” para guardar el cliente en la tabla.</p>			
--	--	---	--	--	--

4	<p>Actualizar peso de cliente</p>	<ol style="list-style-type: none"> 1. Iniciar sesión. 2. Escoger cliente en el menú. 3. Se despliega la lista de clientes 4. El usuario presiona el botón editar en el cliente que desea editar 5. Se despliega una interfaz con los campos cargados de todos los 	<p>Editar el peso del cliente</p>	<p>Se actualizo el peso del cliente correctamente</p>	<p>Aprobado.</p>
---	-----------------------------------	--	-----------------------------------	---	------------------

		<p>datos del cliente</p> <p>6. El usuario edita el peso del cliente</p> <p>7. El usuario guarda el cliente actualizado</p> <p>8. Se despliega la lista con el cliente actualizado</p>			
5	Eliminar cliente.	<p>1. Iniciar sesión.</p> <p>2. Escoger cliente en el menú.</p> <p>3. Se despliega la</p>	<p>Clien te eliminado</p> <p>Correctamen te</p>	<p>Corre cto</p> <p>funcionamie nto la</p> <p>interfaz</p> <p>eliminar cliente</p>	<p>Aprob ado.</p>

		<p>lista de los clientes.</p> <p>4. Pulsa el botón eliminar el cliente que desea eliminar.</p> <p>5. Se despliega una nueva interfaz para confirmar que desea eliminar el cliente</p> <p>6. Se confirma que desea eliminar al cliente</p>			
--	--	---	--	--	--

		7. Se despliega la lista de clientes sin el cliente el cual fue eliminado			
6	Ver rutina asignada	<ol style="list-style-type: none"> 1. Iniciar sesión como cliente. 2. Seleccionar rutina y luego rutina asignada 3. Se despliega una tabla con la rutina a realizar. 	Ver la rutina a realizar	Ciente visualiza la rutina que debe realizar ese día	Aprobado.

7	<p>Asignar ejercicio a rutina</p>	<p>1. Iniciar sesión con las credenciales correspondientes al administrador.</p> <p>2. Seleccionar el apartado de “Registro nuevo de producto”.</p> <p>3. Llenar todos los campos correspondientes.</p> <p>4. Seleccionar la imagen</p>	<p>Registro del producto exitoso.</p>	<p>Corrección al funcionamiento al agregar un producto.</p>	<p>Aprobado.</p>
---	-----------------------------------	---	---------------------------------------	---	------------------

		del producto. 5. Pulsar en el botón “Guardar”.			
8	Registro de usuario.	1. Iniciar sesión con las credenciales correspondientes al administrador. 2. Seleccionar el apartado de “Crear notificaciones” 3. Ingresar los datos necesarios	Registro de notificación creada con éxito.	Correo electrónico funcionamiento de los botones al agregar una notificación.	Aprobado.

		4. Pulsar el botón de “guardar”			
0	1 Cerrar sesión.	1. Iniciar sesión con las credenciales correspondientes de un usuario normal o un administrador. 2. Seleccionar el botón “Cerrar sesión”. 3. Notificación de salir de sesión exitoso.	Cerrar sesión exitosa.	Correcto funcionamiento del botón al salir de la sesión.	Aprobado.

5.5. Resultados

El resultado de este proyecto fue la creación exitosa de un prototipo de sistema de gestión para un gimnasio que utilizaba ASP.NET Core MVC, la arquitectura Onion y PostgreSQL. Una interfaz de usuario atractiva y fácil de usar permite a clientes y administradores administrar rutinas, planes alimenticios y membresías de manera eficiente. La experiencia fue mejorada con la implementación de una navegación fácil de usar en la página de inicio, la edición de detalles en rutinas y planes alimenticios, así como el proceso de gestión de membresías.

La concepción del sistema de gestión se basó en un análisis completo de los requisitos funcionales y no funcionales para garantizar que el producto final cumpliera con las expectativas. Además, para garantizar un desarrollo consistente y eficiente, se seleccionó cuidadosamente la tecnología más adecuada, incluidos marcos y lenguajes de programación. Este enfoque completo fue esencial para el éxito del proyecto y sienta las bases para futuras expansiones y mejoras en la gestión de gimnasios.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

- 1) La elección de la metodología SCRUM ha sido fundamental para una gestión de proyectos efectiva, ya que me permite adaptarme de manera ágil a los cambios en los requisitos y garantizar entregas incrementales y regulares. Esta estrategia ha aumentado significativamente la calidad y la flexibilidad del producto final.
- 2) El sistema implementado satisface los requisitos funcionales, permitiendo la gestión integral de rutinas, planes alimenticios y membresías en un entorno de gimnasio. La

atención especial a la experiencia del usuario ha resultado en una interfaz intuitiva que facilita la edición de detalles en rutinas y planes alimenticios, mejorando la usabilidad de la aplicación.

- 3) La aplicación de medidas de seguridad como la autenticación y autorización, junto con la validación de datos, garantiza que los datos del usuario estén seguros y seguros. La optimización de consultas SQL y la selección de PostgreSQL como base de datos mejoran el rendimiento del sistema.
- 4) La arquitectura Onion ha demostrado ser efectiva para la escalabilidad y la mantenibilidad del sistema, ofreciendo una estructura modular que facilita las expansiones y actualizaciones posteriores. Mi compromiso con la calidad del software y la facilitación de futuros desarrollos se reflejan en la inclusión de pruebas unitarias y documentación clara del código.

Recomendaciones

- 1) Experimentar y utilizar técnicas de diseño de experiencia de usuario (UX) avanzadas para mejorar aún más la interfaz del sistema. Para hacer que la aplicación sea más fácil de usar y atractiva, esto podría incluir estudios de usabilidad, pruebas de usuario e implementación de características centradas en el usuario.
- 2) Explorar la integración de las tecnologías emergentes, como el aprendizaje automático o la inteligencia artificial (IA), pueden adaptarse aún más a las preferencias individuales de los usuarios en cuanto a sus rutinas de ejercicio y planes de dieta. Esto podría resultar en un sistema más inteligente y adaptable a las necesidades únicas de cada cliente.
- 3) Considerar la creación de una aplicación móvil adicional que permita a los clientes acceder y administrar sus hábitos alimenticios y rutinas desde sus dispositivos móviles. Esto permitiría a los usuarios interactuar con el sistema en cualquier momento y lugar, mejorando la accesibilidad y la comodidad.

Bibliografía

- Huet, P. (24 de Agosto de 2022). *Open Webinars*. Obtenido de Open Webinars:
<https://openwebinars.net/blog/arquitectura-de-software-que-es-y-que-tipos-existen/>
- Herrera, L. (2016). Comparación del desempeño de los Sistemas Gestores de Bases de Datos MySQL y PostgreSQL. Obtenido de
<http://repositorio.puce.edu.ec/bitstream/handle/22000/19728/Tesis%20-Pablo-Santacruz-V10.pdf?sequence=1&isAllowed=y#page=36&zoom=100,92,31>
- Castillo, A. A. (2017). Curso de Programación Web: JavaScript, Ajax y jQuery. 2ª Edición.
- Feliciano, C. (2017, junio 6). 4 razones para utilizar SCRUM a la hora de desarrollar software. INVID. <https://invidgroup.com/es/4-razones-para-utilizar-scrum-la-hora-de-desarrollar-software/>
- Qué es SCRUM. (s/f). Arimetrics. Recuperado el 17 de noviembre de 2022, de <https://www.arimetrics.com/glosario-digital/scrum>
- Canal, P. (2022). Definición y características del Scrum Master. Thinking for Innovation. <https://www.iebschool.com/blog/definicion-y-caracteristicas-del-scrum-master-agile-scrum/>
- Rehkopf, D. M. (s/f). Todo lo que necesitas saber sobre los sprints de scrum. Atlassian. Recuperado el 17 de noviembre de 2022, de <https://www.atlassian.com/es/agile/scrum/sprints>
- Scrum: roles y responsabilidades. (2017, noviembre 29). Deloitte Spain. <https://www2.deloitte.com/es/es/pages/technology/articles/roles-y-responsabilidades-scrum.html>
- Mancuzo, G. (2020, julio 10). Fases de la Metodología Scrum. Blog - ComparaSoftware; ComparaSoftware. <https://blog.comparasoftware.com/fases-metodologia-scrum/>
- (S/f). Amazon.com. Recuperado el 17 de noviembre de 2022, de <https://aws.amazon.com/es/what-is/javascript/>
- Rick-Anderson. (s/f). Información general de Asp.net Core. Microsoft.com. Recuperado el 17 de noviembre de 2022, de <https://learn.microsoft.com/es-es/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-7.0>
- MVC - Glosario de MDN Web Docs: Definiciones de términos relacionados con la Web. (s/f). Mozilla.org. Recuperado el 17 de noviembre de 2022, de <https://developer.mozilla.org/es/docs/Glossary/MVC>
- <http://repositorio.puce.edu.ec/bitstream/handle/22000/19728/Tesis%20-Pablo-Santacruz-V10.pdf?sequence=1&isAllowed=y#page=36&zoom=100,92,31>
- <http://repositorio.puce.edu.ec/bitstream/handle/22000/19728/Tesis%20-Pablo-Santacruz-V10.pdf?sequence=1&isAllowed=y#page=36&zoom=100,92,31>
- Raftcart. (23 de abril 2021). RafCart - Plantilla HTML multipropósito Bootstrap5 de comercio

electrónico. Templatemonster. <https://www.templatemonster.com/es/plantillas-web-tipo181186.html>

Metronic. (s.f). Metronic Dashboard. Metronic. <https://keenthemes.com/metronic>

ANEXOS

Anexo A: Cronograma Plan de Titulación

Capítulos	Agosto				Septiembre				Octubre				Noviembre				Diciembre			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Capítulo 1: Introducción																				
Capítulo 2: Marco Teórico																				
Capítulo 3: Metodología																				
Capítulo 4: Desarrollo de la Investigación																				
Capítulo 5: Implementación																				
Conclusiones y Recomendaciones																				