



UNIVERSIDAD INTERNACIONAL SAN ISIDRO LABRADOR

FACULTAD DE INGENIERÍA DE SISTEMAS

PROYECTO DE GRADUACIÓN PARA OPTAR POR EL GRADO DE LICENCIATURA
EN INGENIERÍA EN SISTEMAS

SISTEMA INTEGRAL DE GESTIÓN PARA EL GIMNASIO EL KRAKEN (SIGEK)

PROFESOR TUTOR: ERIC CORELLA SOLIS

ELABORADO POR:

CARLOS FABRICIO AGÜERO ELIZONDO

RANDALL MORA SÁNCHEZ

PÉREZ ZELEDÓN, 2024

Declaración Jurada

Por este medio nosotros, Carlos Fabricio Agüero Elizondo portador de cédula de identidad número 3 0525 0270 y Randall Mora Sánchez portador de cédula 3 0526 0128 estudiantes de la Universidad Internacional San Isidro Labrador de la carrera de Licenciatura de Ingeniería en Sistemas, declaramos bajo fe de juramento y conscientes de las responsabilidades penales de este acto, que somos los autores intelectuales del proyecto de graduación titulado:

SISTEMA INTEGRAL DE GESTIÓN PARA EL GIMNASIO EL KRAKEN (SIGEK)

Juramos que este proyecto programado es original y que respetamos las leyes de los derechos de autor, por lo que liberamos a la Universidad Internacional San Isidro Labrador, de cualquier responsabilidad en caso de que nuestra declaración sea falsa.

Brindada en San Isidro, Pérez Zeledón, San José, Costa Rica el día 07 del mes de diciembre del año 2024.

Carlos Fabricio Agüero Elizondo

Cédula 3 0525 0270

Randall Mora Sánchez

Cédula 3 0526 0128

Tribunal examinador

Proyecto Programado grado de Licenciatura en Ingeniería de Sistemas, presentado el día 7 del mes de diciembre del año 2024, en la Universidad Internacional San Isidro Labrador ante el siguiente tribunal examinador.

Ing. Rudy Rodríguez Acuña

Director de La Escuela de Ingeniería en Sistemas

Ing. Eric Corella Solís

Profesor Tutor

MSc. Michael Corrales Oviedo

Profesor Lector

Carlos Fabricio Agüero Elizondo

Randall Mora Sánchez

Agradecimientos

Queremos agradecer a:

Dios porque nos brindó la salud, sabiduría y fuerza de voluntad para lograr esta meta a pesar del gran reto que significa trabajar y estudiar entre las múltiples circunstancias que se deben enfrentar día con día.

A nuestra familia y parejas por toda la motivación brindada a través de la carrera, por todos aquellos consejos que nos dieron fuerza para seguir cumpliendo con nuestros deberes como estudiantes, profesionales y personas.

A nuestros profesores que nos brindaron todas las herramientas y conocimientos necesarios para mejorar como profesionales. Su sabiduría y dedicación para enseñar han sido un ejemplo para superar los obstáculos.

Dedicatoria

Este proyecto está dedicado a todas aquellas personas que desean seguir mejorando como profesionales, a pesar del gran reto que implica trabajar y estudiar para financiar sus estudios. También está dedicado a todas las personas que nos aconsejaron y motivaron a lo largo de nuestra carrera universitaria.

Índice

1.	Introducción	10
1.1.	Antecedentes	11
1.1.1.	Antecedentes del negocio.....	11
1.1.2.	Antecedentes del problema	12
1.2.	Descripción del problema.....	13
1.3.	Justificación	14
1.4.	Formulación del problema.....	15
1.5.	Objetivos	16
1.5.1.	Objetivo general.....	16
1.5.2.	Objetivos específicos	16
1.6.	Delimitación del proyecto	16
1.6.1.	Alcances	17
1.6.2.	Limitaciones.....	17
1.7.	Tabla de operatividad de los objetivos	18
2.	Marco teórico	20
2.1.	Usuario	20
2.2.	Caso de uso.....	21
2.3.	Requerimiento funcional	21
2.4.	Requerimiento no funcional	21
2.5.	Diagrama UML.....	22
2.6.	Programación orientada a objetos	22
2.7.	Base de datos relacional.....	22
2.8.	Motor de base de datos	22
2.9.	Application Programming Interface (API).....	23
2.10.	TypeScript.....	23
2.11.	React	23
3.	Requerimientos	24
3.1.	Módulo de login.....	24
3.2.	Módulo de menú	24
3.3.	Módulo inicio (home).....	24
3.4.	Módulo de mi perfil.....	25
3.5.	Módulo de reserva	25
3.6.	Módulo de clase reservada	26

3.7.	Módulo de registro de asistencia	26
3.8.	Módulo de registro de resultados de clase	27
3.9.	Módulo de resultados de clases.....	27
3.10.	Módulo de Usuarios	28
3.11.	Módulo de creación de rutina.....	28
3.12.	Módulo de creación de clase	28
3.13.	Módulo de clase admin	29
3.14.	Módulo de membresías	29
4.	Solución propuesta	30
4.1.	Diagrama de base de datos	30
4.2.	Diagramas de casos de uso.....	31
4.2.1.	Módulo de reserva	31
4.2.2.	Módulo de resultados.....	32
4.2.3.	Módulo de mi perfil	33
4.2.4.	Módulo de mediciones.....	34
4.2.5.	Módulo de seguridad	35
4.2.6.	Módulo de personas	36
4.2.7.	Módulo de clase.....	37
4.2.8.	Módulo de enfermedades	38
4.2.9.	Módulo de actividades	39
4.2.10.	Módulo de membresía	40
4.2.11.	Módulo de usuarios	41
4.2.12.	Módulo de rutina	42
4.2.13.	Módulo de enfermedades de la persona.....	43
4.2.14.	Módulo de paquete	44
4.2.15.	Módulo de reserva	45
4.3.	Pantallas de la solución implementada.....	45
5.	Conclusiones.....	51
6.	Recomendaciones	52
7.	Bibliografía	53

Índice de tablas

Tabla 1.	Operatividad de los objetivos	20
Tabla 2.	Requerimiento login	24
Tabla 3.	Requerimiento menú	24
Tabla 4.	Requerimiento inicio	25
Tabla 5.	Requerimiento mi perfil.....	25
Tabla 6.	Requerimiento reserva	26
Tabla 7.	Requerimiento clase reservada	26
Tabla 8.	Requerimiento registro de asistencia	27
Tabla 9.	Requerimiento registro de resultados de clase	27
Tabla 10.	Requerimiento de resultados de clases	28
Tabla 11.	Requerimiento usuarios	28
Tabla 12.	Requerimiento creación de rutina	28
Tabla 13.	Requerimiento creación de clase.....	29
Tabla 14.	Requerimiento clase admin	29
Tabla 15.	Requerimiento membresías.....	29

Índice de figuras

Figura 1.	Diagrama entidad relación	30
Figura 2.	Caso de uso reserva.....	31
Figura 3.	Caso de uso resultados	32
Figura 4.	Caso de uso mi perfil	33
Figura 5.	Caso de uso mediciones.....	34
Figura 6.	Caso de uso seguridad	35
Figura 7.	Caso de uso personas	36
Figura 8.	Caso de uso clase	37
Figura 9.	Caso de uso enfermedades.....	38
Figura 10.	Caso de uso actividades.....	39
Figura 11.	Caso de uso membresía.....	40
Figura 12.	Caso de uso usuarios	41
Figura 13.	Caso de uso rutina.....	42
Figura 14.	Caso de uso enfermedades de la persona	43
Figura 15.	Caso de uso paquete.....	44
Figura 16.	Caso de uso reserva (administrador).....	45
Figura 17.	Pantalla login	46
Figura 18.	Pantalla bienvenida administrador	46
Figura 19.	Pantalla bienvenida usuario.....	47
Figura 20.	Menú administrador	47
Figura 21.	Pantalla gestión de usuarios.....	48
Figura 22.	Pantalla agregar rutina.....	48
Figura 23.	Pantalla gestión de clases	49
Figura 24.	Pantalla gestión de paquetes.....	49
Figura 25.	Pantalla gestión de membresía	50
Figura 26.	Pantalla gestión de reserva	50
Figura 27.	Pantalla marcar asistencia QR	51

1. Introducción

En el mundo moderno existen trabajos que no implican movimiento o esfuerzo físico, esto provoca que muchas personas dejen de lado la actividad física necesaria para que el cuerpo humano se mantenga saludable, para ello existen diversos espacios en los cuales las personas pueden realizar actividades recreativas o de ejercicio.

Hoy en día está a disposición de las personas, espacios denominados gimnasios en los que un profesional llamado entrenador nos indica las rutinas a seguir según nuestra condición física del sujeto, para ello se apoya en la teoría de su profesión y experiencia aplicándola. Junto con estas actividades, es común que el entrenador administre su propio gimnasio, para ello se propone una solución que apoye a tareas comunes realizadas en la gestión de negocios y además brindar una propuesta que ayude a realizar un trabajo personalizado a cada persona que haga uso del servicio.

Con el avance de la tecnología, surge la oportunidad de optimizar operaciones mediante el desarrollo de un sistema integral que administre personas, pagos, análisis de datos y herramientas para mejorar los entrenamientos personalizados. Este proyecto tiene como objetivo la entrega de un sistema de gestión de gimnasios que se adapte a las necesidades y oportunidades que tiene el gimnasio El Kraken, simplificando las tareas administrativas y mejore la experiencia del usuario mediante la implementación de un asistente virtual basado en Chat GPT para poder colaborar en la elaboración de rutinas y rutas de entrenamiento según las necesidades de la persona.

1.1. Antecedentes

En la presente sección se abordarán los antecedentes fundamentales que contextualizan su desarrollo, divididos en dos secciones principales: antecedentes del negocio y antecedentes del problema. Los antecedentes del negocio proporcionarán una visión general de la historia, misión, visión, y estructura de la organización, así como de su posición actual en el mercado y sus principales logros. Este contexto es esencial para comprender las capacidades y recursos con los que cuenta el gimnasio para enfrentar los desafíos presentes. Por otro lado, los antecedentes del problema se enfocarán en detallar la naturaleza del problema que se busca resolver, incluyendo su origen, evolución, y el impacto que ha tenido tanto en la operación como en su relación con los clientes y el mercado en general. Esta sección permitirá identificar las causas subyacentes del problema y la urgencia de su resolución, sentando las bases para el desarrollo de soluciones efectivas y sostenibles.

1.1.1. Antecedentes del negocio

El Kraken es una empresa que brinda un servicio de entrenamiento físico que busca mejorar el estado físico de las personas de todas las edades y prevenir enfermedades ligadas al sedentarismo.

Misión: En El Kraken, nos dedicamos a mejorar el estado físico y la salud de personas de todas las edades a través de un entrenamiento físico integral. Nos esforzamos por prevenir enfermedades relacionadas con el sedentarismo, proporcionando un servicio al cliente excepcional, programación enfocada en el rendimiento y resultados reales en menos tiempo. Nuestros valores de honestidad,

respeto, empatía, humildad, compromiso y responsabilidad guían cada interacción, garantizando que nuestros usuarios confíen plenamente en nuestra capacidad para cuidar de su salud y bienestar.

Visión: Ser reconocidos a nivel nacional como una cadena de centros de acondicionamiento físico de alta calidad, transformando el estilo de vida y la salud de las personas en todo el país. Nos proponemos expandir nuestro alcance, llevando nuestro enfoque único en rendimiento y servicio al cliente a nuevas zonas, y convirtiéndonos en líderes en la promoción de un estilo de vida activo y saludable.

Historia: El Kraken comenzó a finales del año 2016 en el pueblo de Juan Viñas como un pasatiempo que fue creciendo hasta volverse en lo que es hoy. Inicialmente se realizaban los ejercicios en una casa, luego de ello en un garaje y a la fecha ese garaje se amplió para poder tener más equipo y recibir más personas por sesión.

Durante la pandemia del COVID-19, el gimnasio pudo cumplir los reglamentos establecidos por el ministerio de salud y debido a esto pudo mantenerse brindando los servicios a las personas. Esto fue de suma importancia para la comunidad debido a que, durante el aislamiento social, las personas estaban limitadas en sus actividades físicas y El Kraken pudo ayudar a mejorar esta situación.

1.1.2. Antecedentes del problema

Actualmente se cuenta con un sistema que gestiona las personas y las suscripciones de cada una de ellas, además de poder meter datos de peso, IMC, estatura, ejercicio y demás, sin embargo, el sistema es poco interactivo y con una interfaz engorrosa para el usuario y administrador, esto deriva en una carencia al

momento de realizar las gestiones pertinentes por parte del administrador y poca analítica e interacción para el usuario.

Es necesario contar con un apoyo que facilite la elaboración de rutinas y rutas de entrenamiento para optimizar y agilizar la asignación de rutinas a las personas, esto según los parámetros ingresados por el entrenador y los objetivos que tiene la persona usuaria.

Además, es necesario de una interfaz de usuario para los clientes que motive a utilizar la aplicación y la experiencia de usuario sea la idónea para personas que se quieran mantener saludables.

1.2. Descripción del problema

El Gimnasio "El Kraken" ha reconocido una necesidad urgente de actualizar su sistema integrado de gestión debido a que el software actual no cumple con los estándares de eficiencia y robustez necesarios para una operación fluida. Esta situación ha derivado en múltiples problemas operativos que impactan negativamente tanto a los empleados como a los clientes, comprometiendo la calidad del servicio y la eficiencia del negocio. La gestión de clientes es un proceso central para cualquier gimnasio, pero el sistema actual presenta limitaciones significativas en esta área. La incapacidad para gestionar de manera eficiente el registro, y el seguimiento de pagos ha provocado errores frecuentes en el conteo y la actualización de datos de los miembros. Estos errores no solo generan insatisfacción entre los clientes, sino que también incrementan la carga administrativa para el personal. La ausencia de una funcionalidad robusta para crear rutinas ha provocado que los empleados pierdan

mucho tiempo en la gestión de estas, además del descontrolado uso del papel. Para abordar estos problemas y asegurar una operación eficiente y satisfactoria, es crucial desarrollar un nuevo software de gestión que sea moderno y robusto. Este nuevo sistema debe ofrecer una interfaz de usuario intuitiva y proveer análisis estadístico básico que permita facilitar la toma de decisiones en los diferentes procesos del gimnasio.

1.3. Justificación

Actualizar el sistema integrado de gestión del Gimnasio "El Kraken" es una necesidad urgente debido a las deficiencias del software actual que afectan negativamente tanto a empleados como a clientes. Un nuevo software moderno y robusto es crucial para mejorar la eficiencia operativa y aumentar la satisfacción del cliente. La gestión de membresías, el manejo de pagos y creación de rutinas reducirá la carga de trabajo manual para el personal, permitiéndoles enfocarse en tareas más estratégicas y en brindar un mejor servicio. Además, un sistema actualizado minimizará los errores en facturación y registro de asistencias, incrementando la precisión de los datos y la confiabilidad del sistema, resultando en una operación más fluida y efectiva.

Una interfaz de usuario intuitiva y moderna mejorará la experiencia tanto de los empleados como de los miembros, quienes podrán gestionar sus reservas y pagos de manera más fácil y rápida, incrementando su satisfacción y fidelidad. La capacidad de generar informes detallados y estadísticas permitirá al gimnasio tomar decisiones informadas con base al análisis estadístico, optimizando el uso de recursos y mejorando la gestión del tiempo.

La implementación de medidas de seguridad avanzadas protegerá los datos sensibles de los miembros y de la operación del gimnasio, asegurando la privacidad y cumpliendo con las normativas legales vigentes. Además, un sistema moderno y flexible permitirá al gimnasio escalar sus operaciones de manera eficiente conforme crece el número de miembros y se expanden los servicios ofrecidos, asegurando que el software pueda adaptarse a las necesidades cambiantes del negocio sin requerir cambios significativos.

La adopción de un software avanzado y eficiente posicionará al Gimnasio "El Kraken" como un líder en la industria fitness de la zona, ofreciendo una experiencia superior a sus clientes y diferenciándose de la competencia. Este proyecto no sólo resolverá las deficiencias del sistema actual, sino que también proporcionará una base sólida para el crecimiento y éxito sostenido del gimnasio.

1.4. Formulación del problema

Según lo indignado por medio de entrevistas al dueño del gimnasio, se ha validado que la gestión realizada mediante la aplicación actual es engorrosa y poco atractiva para él como administrador y para sus clientes. Además, no se adapta como debería ser a su negocio, ya que, existen funcionalidades que se desean poseer, pero no están disponibles, como lo es la capacidad de contar con la herramienta de creación de rutinas y rutas de entrenamiento según los parámetros de cada persona.

1.5. Objetivos

1.5.1. Objetivo general

Desarrollar un sistema de gestión integral para el gimnasio El Kraken el cual automatiza procesos administrativos, proporcionará análisis de datos y la mejora la experiencia del usuario mediante la implementación de un asistente virtual para la creación de rutinas y rutas de entrenamiento.

1.5.2. Objetivos específicos

- Definir las características y funciones que el cliente desea que tenga el SIGEK, mediante la conversación y demostración de ilustraciones que permitan validar los requerimientos.
- Construir una base de datos mediante diagramas y un motor de base de datos para soportar las necesidades del negocio y garantizar un rendimiento óptimo, escalabilidad y seguridad de sus datos.
- Programar el front-end y back-end del SIGEK que resuelva las necesidades del negocio previamente identificadas y estar en constante validación con el cliente.
- Aplicar el despliegue e integración del SIGEK a los usuarios finales mediante capacitaciones y documentación que permita adquirir el conocimiento necesario para el uso del sistema.

1.6. Delimitación del proyecto

La siguiente sección, describe por medio de los alcances y limitaciones, las delimitaciones que tiene el proyecto y justifica cada una de ellas.

1.6.1. Alcances

El proyecto se centrará en el desarrollo de un sistema de gestión de gimnasios que incluya las siguientes funcionalidades principales:

- Gestión de personas y membresía.
- Estadísticas básicas de datos.
- Implementación de un asistente para la creación de rutinas.

El sistema estará diseñado para ser escalable y adaptable a diferentes tipos y tamaños de gimnasios, con una interfaz de usuario intuitiva y amigable.

Con este proyecto, se busca mejorar la eficiencia operativa del gimnasio El Kraken, proporcionar una experiencia más personalizada a los usuarios y fomentar un estilo de vida saludable mediante el acceso a herramientas y recursos de entrenamiento de calidad.

1.6.2. Limitaciones

Actualmente no se cuenta con la infraestructura necesaria para mantener un sistema propio del gimnasio, sin embargo, es posible adquirir lo necesario por medio del cloud computing o la nube. Agregado a ello, el tiempo y personal es limitado debido a que para el desarrollo del sistema se tiene a disposición 4 meses y 2 personas que laboran 40 horas semanales en promedio.

1.7. Tabla de operatividad de los objetivos

Objetivos específicos	Herramientas y metodologías	Actividades
<p>Definir las características y funciones que el cliente desea que tenga el SIGEK, mediante la conversación y demostración de ilustraciones que permitan validar los requerimientos.</p>	<p>Uso de la metodología SCRUM para la toma de requerimientos.</p> <p>Entrevistas con el dueño del gimnasio.</p> <p>Observación de la operación normal del gimnasio.</p>	<p>Redacción de las historias de usuario.</p> <p>Validación de las historias de usuario con el dueño del gimnasio.</p> <p>Realizar bocetos ilustrativos del sistema.</p>
<p>Construir una base de datos mediante diagramas y un motor de base de datos para soportar las necesidades del negocio y garantizar un rendimiento óptimo, escalabilidad y seguridad de sus datos.</p>	<p>Uso de diagramas relacionales.</p> <p>Validación de la funcionalidad de la solución.</p> <p>Definir el motor de base de datos.</p> <p>Construcción de la base de datos.</p>	<p>Pruebas de inserción y almacenamiento.</p> <p>Inserción de registros con escenarios comunes dentro del gimnasio.</p>

<p>Programar el front-end y back-end del SIGEK que resuelva las necesidades del negocio previamente identificadas y estar en constante validación con el cliente.</p>	<p>Definir la arquitectura de la aplicación frontend</p> <p>Construcción del frontend mediante el uso de un Framework actual y eficiente para las necesidades del sistema</p> <p>Construcción del backend mediante el uso de una tecnología actual para la programación de servicios o APIs</p>	<p>Construcción de métodos de programación en las diferentes capas de lógica.</p> <p>Pruebas unitarias en métodos de programación.</p>
<p>Aplicar el despliegue e integración del SIGEK a los usuarios finales mediante capacitaciones y documentación que permita adquirir el conocimiento necesario para el uso del</p>	<p>Identificar el medio por el cual los usuarios puedan hacer acceso a la aplicación final</p> <p>Validar el buen</p>	<p>Implementación del sistema en un servicio de despliegue web.</p> <p>Pruebas de integración y rendimiento.</p>

sistema.	comportamiento de la aplicación una vez desplegado Construir la documentación necesaria para hacer un uso correcto de la aplicación	
----------	--	--

Tabla 1. Operatividad de los objetivos

2. Marco teórico

En esta sección, se presenta y discute el conjunto de teorías, conceptos, y antecedentes que son relevantes para la comprensión del trabajo. Además, permite comprender conceptos y tecnologías clave que va a incorporar o será parte de la solución que se construirá.

2.1. Usuario

Dentro de un sistema existen diferentes entidades, parte de ellas son las personas que están incluidas dentro de los flujos necesarios para que funcione o este mismo tenga un propósito de existencia. Para ello definimos a un usuario como.

Persona que utiliza un producto o servicio de forma habitual. El concepto de usuario se aplica a diferentes ámbitos y plataformas, pero su esencia es la misma: representa a aquel individuo o entidad que hace uso e interactúa con algo para obtener

algún beneficio o cumplir con una función determinada. (De Enciclopedia Significados, 2024)

Tomando en cuenta esto, podemos definir al usuario como la persona que utiliza un producto o servicio, en este caso va a ser la persona que va a utilizar el sistema SIGEK.

2.2. Caso de uso

IBM (2024) define que los casos de uso se desarrollan para perfeccionar un conjunto de requisitos centrados en un rol o tarea específica. Los casos de uso describen las acciones que los usuarios o roles llevan a cabo en la solución, mientras que un proceso de negocio define cómo se ejecutan estas funciones.

2.3. Requerimiento funcional

Los requisitos funcionales de un sistema especifican las tareas que este debe realizar. Dicho de otra forma, describen cómo debe comportarse o qué funciones debe cumplir el sistema o software bajo determinadas condiciones (pmoinformatica, 2017)

2.4. Requerimiento no funcional

Los requisitos no funcionales describen las características generales y las limitaciones del sistema o aplicación en desarrollo. Según pmoinformatica (2017), ejemplos de estos requisitos incluyen atributos como la eficiencia, seguridad, confiabilidad y usabilidad del sistema. Además, se proporcionan ejemplos de requisitos no funcionales a nivel organizacional y externo.

2.5. Diagrama UML

Se desarrolló para proporcionar un lenguaje visual de modelado que sea común, rico en semántica y sintaxis, destinado a la arquitectura, diseño e implementación de sistemas de software complejos, abarcando tanto su estructura como su comportamiento. (Lucidchart, s.f)

2.6. Programación orientada a objetos

La Universidad de Europea (2022) en su sitio web define la programación orientada a objetos es un enfoque de desarrollo de software que se centra en los datos y objetos, en lugar de en las funciones y la lógica. Un objeto se describe como una entidad que contiene datos, junto con atributos y comportamientos específicos.

2.7. Base de datos relacional

Una base de datos relacional es un sistema que almacena y permite el acceso a datos que están interconectados. Estas bases de datos utilizan el modelo relacional, que representa los datos de manera clara y sencilla mediante tablas (Oracle, s.f)

2.8. Motor de base de datos

Una parte indispensable del proyecto es el motor de base de datos, para poder definirlo es necesario entenderlo como “El Motor de base de datos es el servicio principal para almacenar, procesar y proteger los datos. Asimismo, proporciona acceso controlado y procesamiento rápido de transacciones para cumplir los requisitos de las aplicaciones de base de datos más exigentes.” (Pérez, 2011, p. 13)

2.9. Application Programming Interface (API)

La gran mayoría de soluciones de software utilizan las API para la comunicación entre tecnologías, para ello entendemos que “Las API son mecanismos que permiten a dos componentes de software comunicarse entre sí mediante un conjunto de definiciones y protocolos.” (Amazon Web Services, s.f.). Estas están construidas con lenguajes de programación, para el presente trabajo se utilizará un API basada en ASP.NET Core, para la comunicación de las partes del sistema.

2.10. TypeScript

Para la solución propuesta, es necesario conocer de qué es TypeScript(TS), para ello lo definimos como “TypeScript es una extensión al lenguaje de programación JavaScript que se caracteriza por ampliar su sintaxis en el ámbito de los tipos.” (Casero, 2024). Por lo que podemos decir que TS es un lenguaje de programación que tiene como base el JavaScript y extiende sus propiedades como la definición de tipos para la programación orientada a objetos.

2.11. React

Para realizar la interfaz de usuario, con la que mayormente el usuario va a interactuar con el sistema, podemos describir de la siguiente manera a React, “Creada por Facebook, React contiene una colección de fragmentos de código JavaScript reutilizables utilizados para crear interfaces de usuario (UI) llamadas componentes.” (A & A, 2023). Esto nos quiere decir que React es una biblioteca principalmente creada para facilitar a los desarrolladores la creación de interfaces de usuario para los sistemas.

3. Requerimientos

En esta sección se describen los requerimientos acordados, que son la base para elaborar el sistema y satisfacer las necesidades del cliente. Cada uno de ellos fue previamente revisado y aprobado.

3.1. Módulo de login

Requerimiento	REQ-01	Estado	Aprobado
Nombre del módulo	Módulo de Login		
Descripción			
Acceder al inicio de sesión donde se pueda ingresar usuario y contraseña para poder acceder a la aplicación			
Especificaciones			
<ul style="list-style-type: none"> • Usuario • Contraseña 			
Importancia	1	Dependencia	NA
Justificación y comentarios			
Esta pantalla permite al cliente, poder ingresar usuario y contraseña para poder acceder a las funcionalidades de la aplicación.			

Tabla 2. Requerimiento login

3.2. Módulo de menú

Requerimiento	REQ-02	Estado	Aprobado
Nombre del módulo	Módulo de Menú		
Descripción			
Opción en el menú para poder realizar reservas nuevas, ver resultados de clases, el perfil, clase reservada, entre otros			
Especificaciones			
<ul style="list-style-type: none"> • Menú • Ítem que permite ir a la pantalla de reserva 			
Importancia	1	Dependencia	REQ-01
Justificación y comentarios			
Permite al usuario navegar entre las diferentes opciones del sistema disponibles para su rol			

Tabla 3. Requerimiento menú

3.3. Módulo inicio (home)

Requerimiento	REQ-03	Estado	Aprobado
---------------	--------	--------	----------

Nombre del módulo	Módulo Inicio (home)		
Descripción			
Pantalla de inicio donde se observe las reservas activas y suscripción que se adquirió.			
Especificaciones			
<ul style="list-style-type: none"> • Información de las reservas activas de la semana • Suscripción que se posee 			
Importancia	1	Dependencia	REQ-02
Justificación y comentarios			
Esta pantalla permite ver de manera general, datos que son importantes para el cliente, en este caso se muestra solamente las reservas que se tienen activas durante la semana y la suscripción que se adquirió.			

Tabla 4. Requerimiento inicio

3.4. Módulo de mi perfil

Requerimiento	REQ-04	Estado	Aprobado
Nombre del módulo	Módulo de mi perfil		
Descripción			
Consultar la información básica del usuario en sesión en el sistema			
Especificaciones			
<ul style="list-style-type: none"> • Nombre • Fecha nacimiento • Edad • Peso • Estatura • Membresía Activa • Plan • Enfermedades 			
Importancia	1	Dependencia	REQ-01
Justificación y comentarios			
Esta pantalla permite que los clientes puedan consultar su información las veces que sea necesario			

Tabla 5. Requerimiento mi perfil

3.5. Módulo de reserva

Requerimiento	REQ-05	Estado	Aprobado
Nombre del módulo	Módulo de reserva		
Descripción			
Reservar y cancelar asistencia a clases según la fecha y hora disponible para poder tener un campo en el box			
Especificaciones			
<ul style="list-style-type: none"> • Persona • Rutina 			

<ul style="list-style-type: none"> • Fecha • Descripción • Campos disponibles • Campos 			
Importancia	1	Dependencia	REQ-01
Justificación y comentarios			
El presente módulo permitirá que los clientes puedan realizar reservas de las clases en el box, así como cancelarlas. Es importante ya que este es el único medio en el que un cliente puede asistir a la clase.			

Tabla 6. Requerimiento reserva

3.6. Módulo de clase reservada

Requerimiento	REQ-06	Estado	Aprobado
Nombre del módulo	Módulo de clase reservada		
Descripción			
También observar los detalles de la clase reservada, así como los campos disponibles y totales, las personas que asistirán, en el menor tiempo posible. El usuario podrá ver observar la rutina solo si consulta en el tiempo de clase			
Especificaciones			
<ul style="list-style-type: none"> • Personas • Rutina • Fecha • Campos disponibles y totales 			
Importancia	1	Dependencia	REQ-05
Justificación y comentarios			
Esta pantalla es muy importante porque el cliente tiene oportunidad de cancelar la asistencia a una clase, así como observar cuales personas y también para llevar el control de asistencias en las diferentes clases			

Tabla 7. Requerimiento clase reservada

3.7. Módulo de registro de asistencia

Requerimiento	REQ-07	Estado	Aprobado
Nombre del módulo	Módulo de registro de asistencia		
Descripción			
Registrar la asistencia a la clase (reserva)			

Especificaciones			
<ul style="list-style-type: none"> Reserva asociada Marcar asistencia 			
Importancia	1	Dependencia	REQ-06
Justificación y comentarios			
Esta pantalla permite que los clientes puedan marcar la asistencia a la clase y el administrador del gimnasio esté pendiente de los resultados			

Tabla 8. Requerimiento registro de asistencia

3.8. Módulo de registro de resultados de clase

Requerimiento	REQ-08	Estado	Aprobado
Nombre del módulo	Módulo de registro de resultados de clase		
Descripción			
Registrar los resultados obtenidos de la clase asistida			
Especificaciones			
<ul style="list-style-type: none"> Reserva asociada Ejercicio Peso Repeticiones Minutos y segundos 			
Importancia	1	Dependencia	REQ-07
Justificación y comentarios			
Esta pantalla permite que los clientes digiten los resultados de la clase para llevar trazabilidad de los logros de la clase			

Tabla 9. Requerimiento registro de resultados de clase

3.9. Módulo de resultados de clases

Requerimiento	REQ-09	Estado	Aprobado
Nombre del módulo	Módulo de resultados de clases		
Descripción			
Observar los resultados de las clases asistidas tales como repeticiones, pesos, ejercicios, tiempos			
Especificaciones			
<ul style="list-style-type: none"> Reserva asociada Ejercicio Peso Repeticiones Minutos y segundos 			
Importancia	1	Dependencia	REQ-09
Justificación y comentarios			
Esta pantalla permite que los clientes llevar una trazabilidad de sus resultados y récords en los diferentes ejercicios para mejorar sus marcas en las clases			

Tabla 10. Requerimiento de resultados de clases**3.10. Módulo de Usuarios**

Requerimiento	REQ-10	Estado	Aprobado
Nombre del módulo	Módulo de Usuarios		
Descripción			
Registrar nuevos usuarios en el sistema, así como modificar o eliminar los registros			
Especificaciones			
<ul style="list-style-type: none"> • Rol • Nombre de usuario • Contraseña • Es temporal • Teléfono 			
Importancia	1	Dependencia	NA
Justificación y comentarios			
Esta pantalla permite que el administrador pueda registrar nuevos usuarios para permitir el ingreso al sistema a nuevas personas o potenciales clientes.			

Tabla 11. Requerimiento usuarios**3.11. Módulo de creación de rutina**

Requerimiento	REQ-11	Estado	Aprobado
Nombre del módulo	Módulo de Creación de Rutina		
Descripción			
Poder crear rutinas para poder ser utilizadas en las clases			
Especificaciones			
<ul style="list-style-type: none"> • Rutina • Ejercicios 			
Importancia	1	Dependencia	NA
Justificación y comentarios			
Esta pantalla permite al usuario administrador de la aplicación, poder crear rutinas nuevas con ejercicios específicos y así se pueda utilizar en las clases.			

Tabla 12. Requerimiento creación de rutina**3.12. Módulo de creación de clase**

Requerimiento	REQ-12	Estado	Aprobado
Nombre del módulo	Módulo de Creación de clase		
Descripción			
Poder crear clases para que los clientes puedan reservar			
Especificaciones			

<ul style="list-style-type: none"> • Fecha • Hora • Rutina • Ejercicios 			
Importancia	1	Dependencia	NA
Justificación y comentarios			
Esta pantalla permite al usuario administrador de la aplicación, poder crear nuevas clases para que los clientes puedan asistir.			

Tabla 13. Requerimiento creación de clase

3.13. Módulo de clase admin

Requerimiento	REQ-13	Estado	Aprobado
Nombre del módulo	Módulo de Clase Admin		
Descripción			
Ver las clases y los clientes que reservan			
Especificaciones			
<ul style="list-style-type: none"> • Fecha • Hora • Clases 			
Importancia	1	Dependencia	REQ-05
Justificación y comentarios			
Esta pantalla permite al usuario administrador de la aplicación, poder observar todas las clases disponibles, modificarla, y especificar en cada una de ellas los usuarios que reservan.			

Tabla 14. Requerimiento clase admin

3.14. Módulo de membresías

Requerimiento	REQ-14	Estado	Aprobado
Nombre del módulo	Módulo de Membresías		
Descripción			
Gestionar paquetes de membresías			
Especificaciones			
<ul style="list-style-type: none"> • Membresías disponibles 			
Importancia	1	Dependencia	NA
Justificación y comentarios			
Esta pantalla permite al usuario administrador de la aplicación, poder hacer la gestión de todo lo relacionado con los paquetes disponibles en la aplicación.			

Tabla 15. Requerimiento membresías

4. Solución propuesta

En esta sección se expondrá de manera ilustrativa por medio de figuras la solución propuesta.

4.1. Diagrama de base de datos

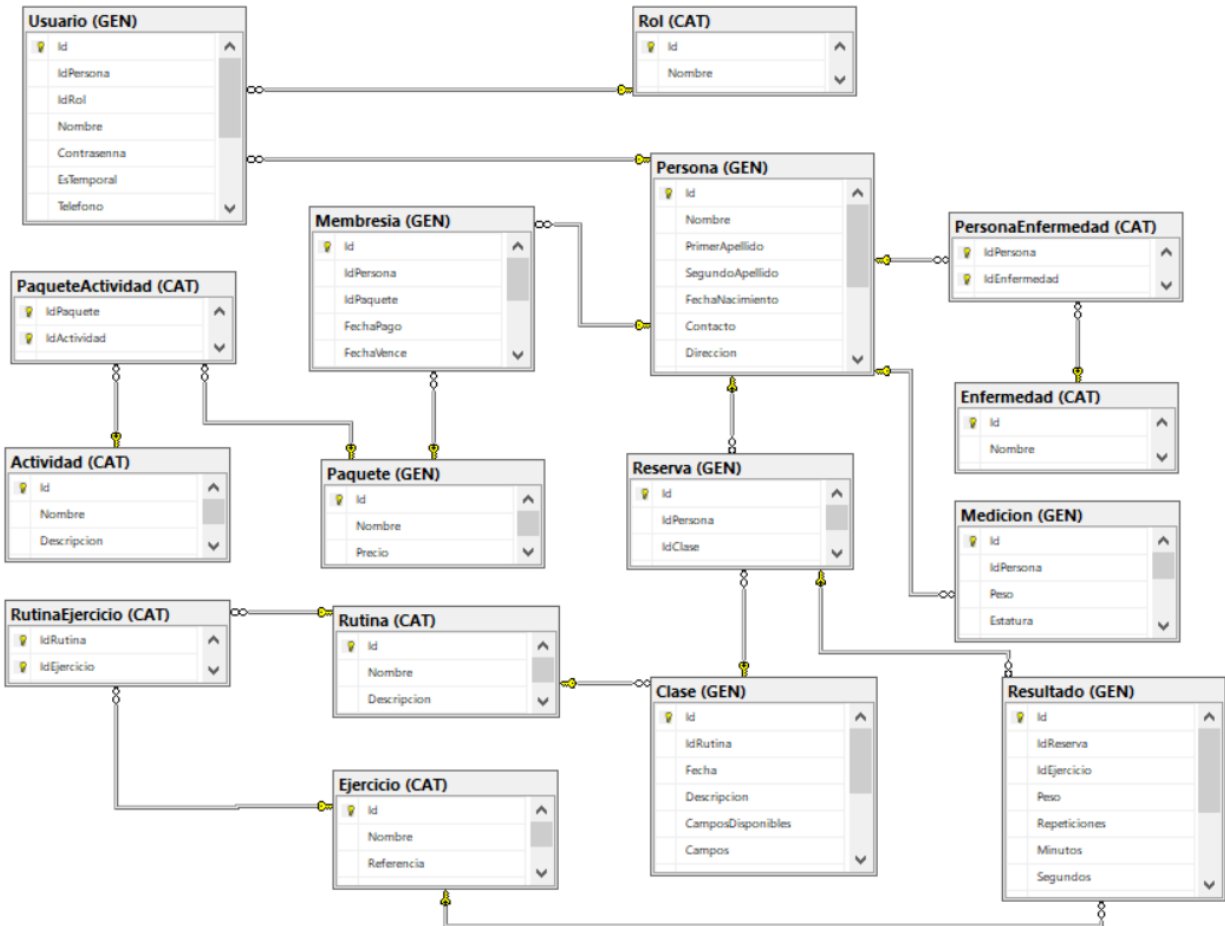


Figura 1. Diagrama entidad relación

4.2. Diagramas de casos de uso

4.2.1. Módulo de reserva

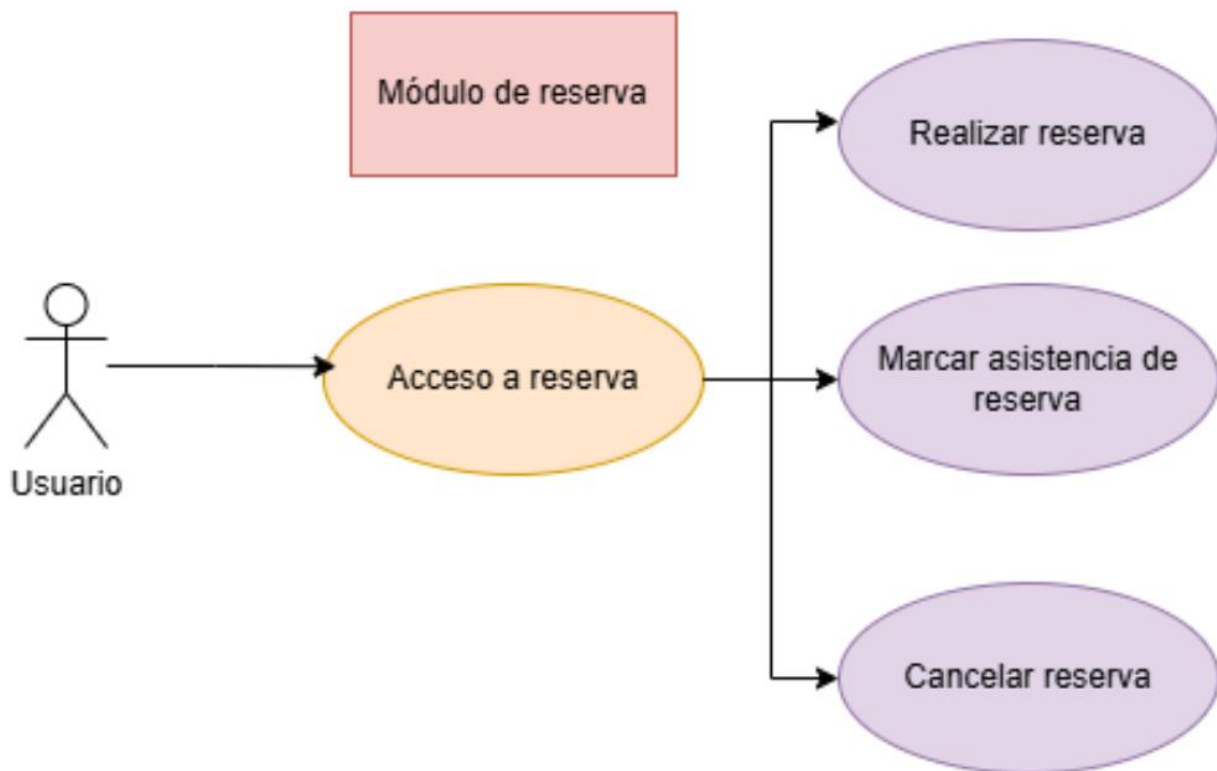


Figura 2. Caso de uso reserva

4.2.2. Módulo de resultados

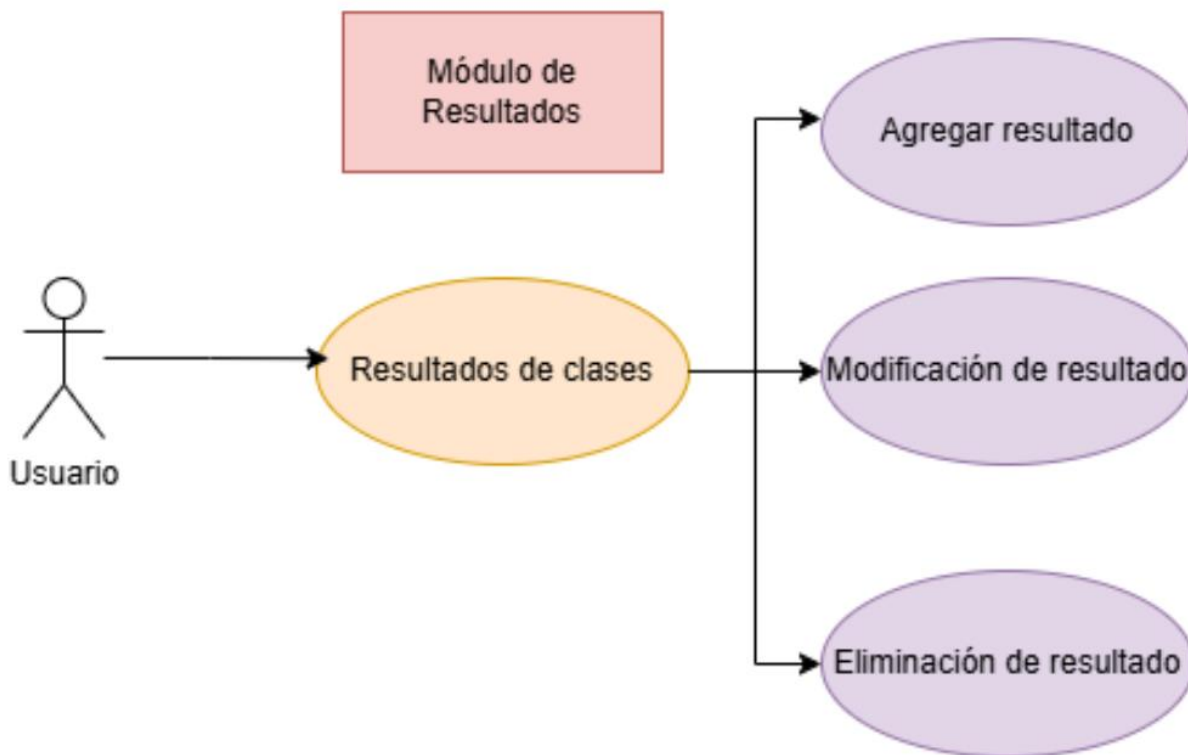


Figura 3. Caso de uso resultados

4.2.3. Módulo de mi perfil

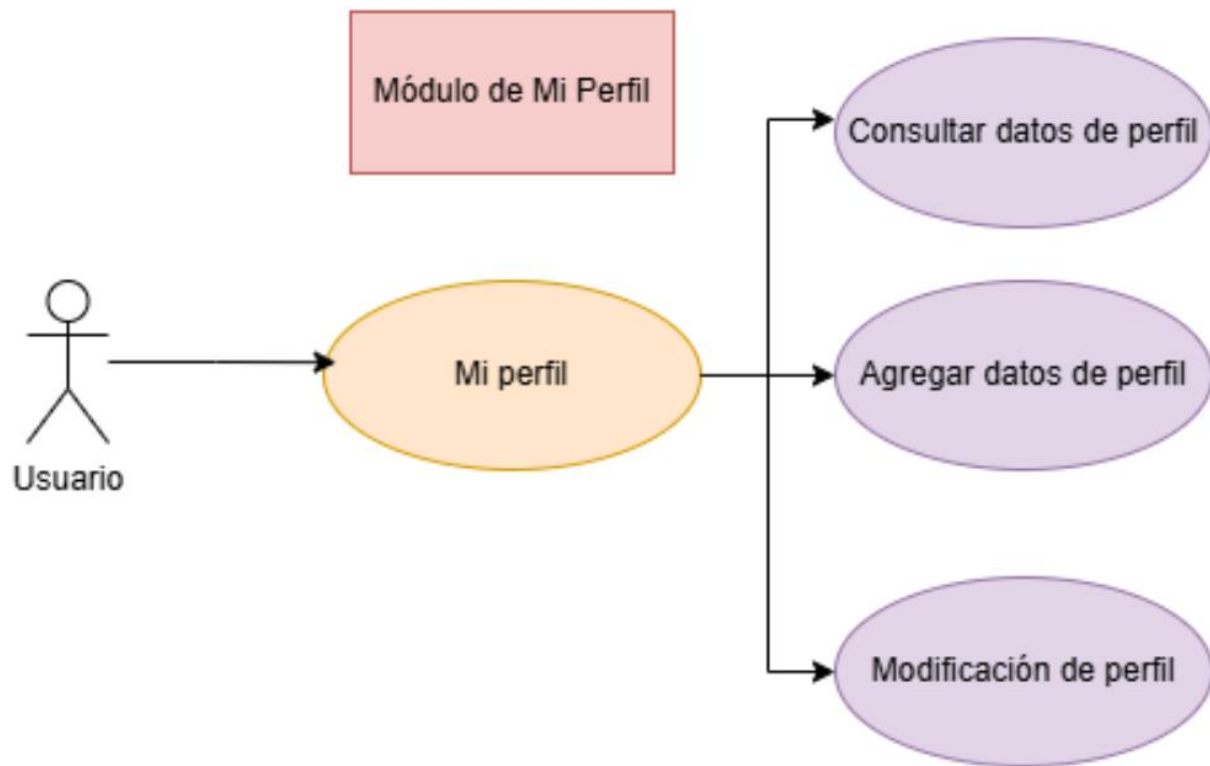


Figura 4. Caso de uso mi perfil

4.2.4. Módulo de mediciones

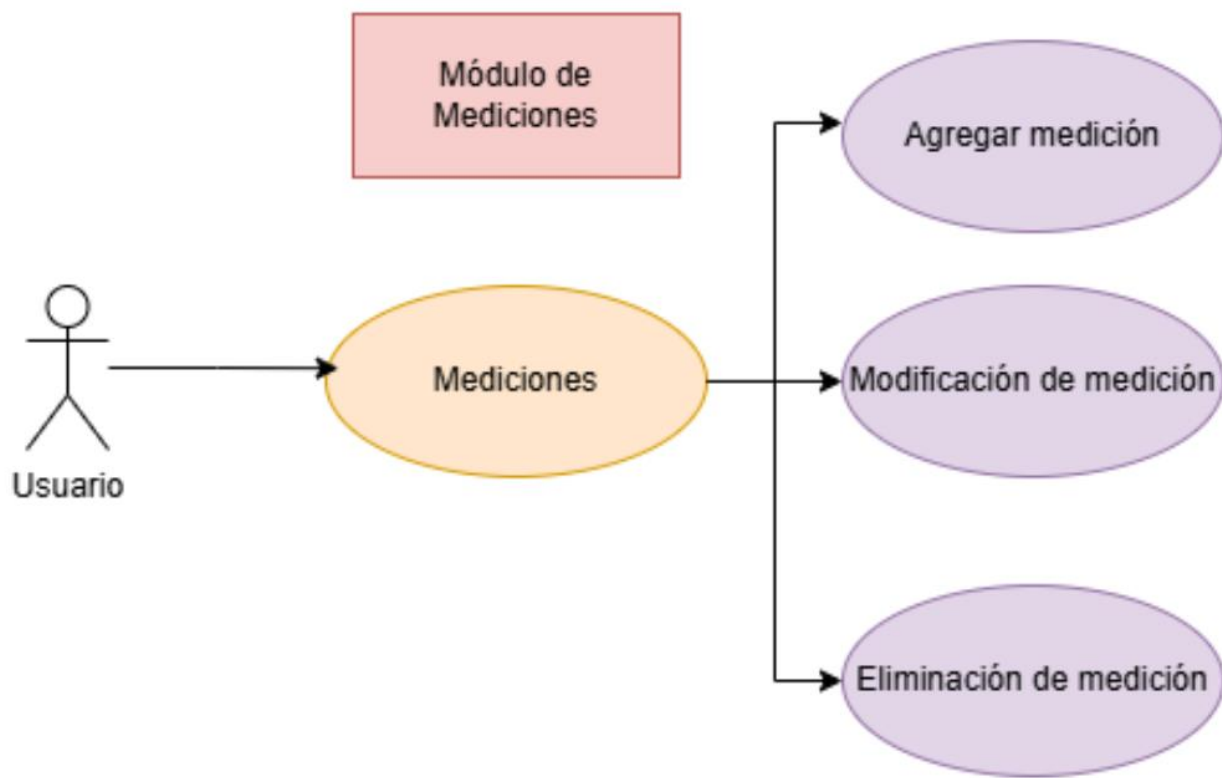


Figura 5. Caso de uso mediciones

4.2.5. Módulo de seguridad

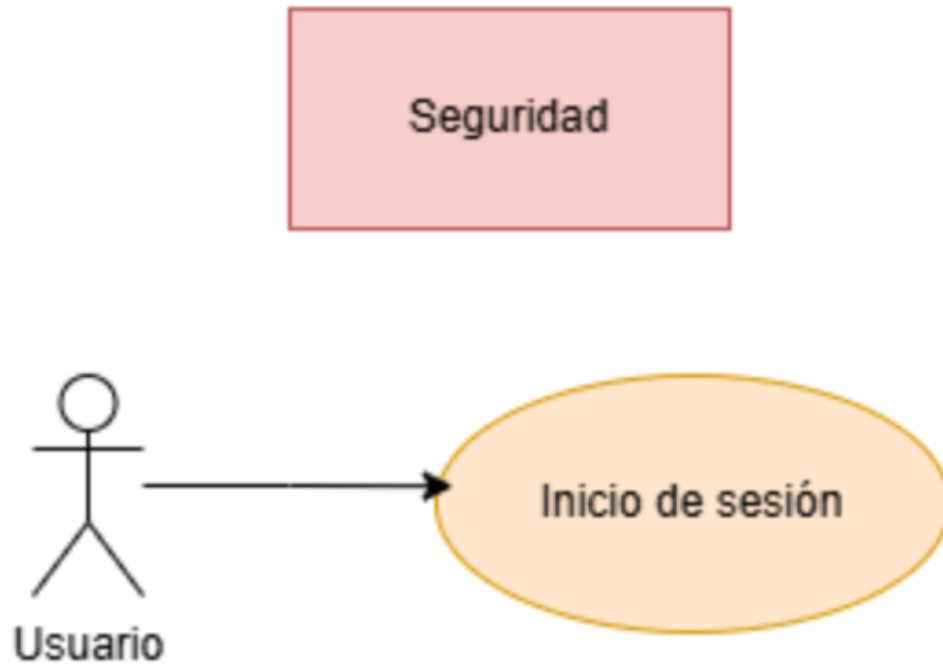


Figura 6. Caso de uso seguridad

4.2.6. Módulo de personas

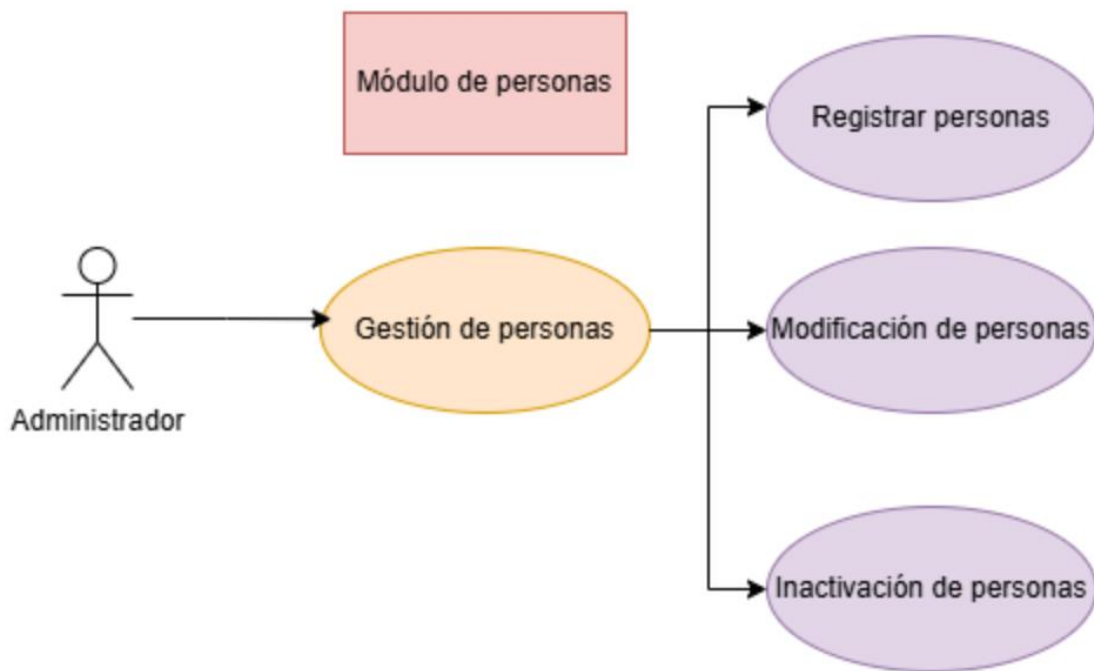


Figura 7. Caso de uso personas

4.2.7. Módulo de clase

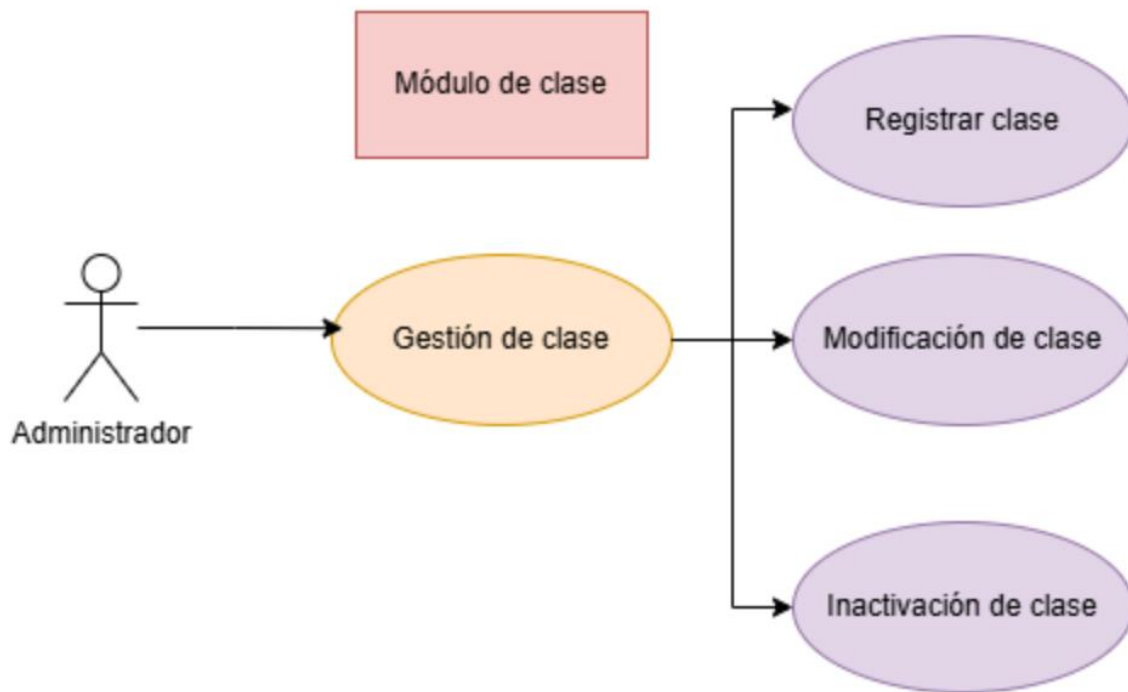


Figura 8. Caso de uso clase

4.2.8. Módulo de enfermedades

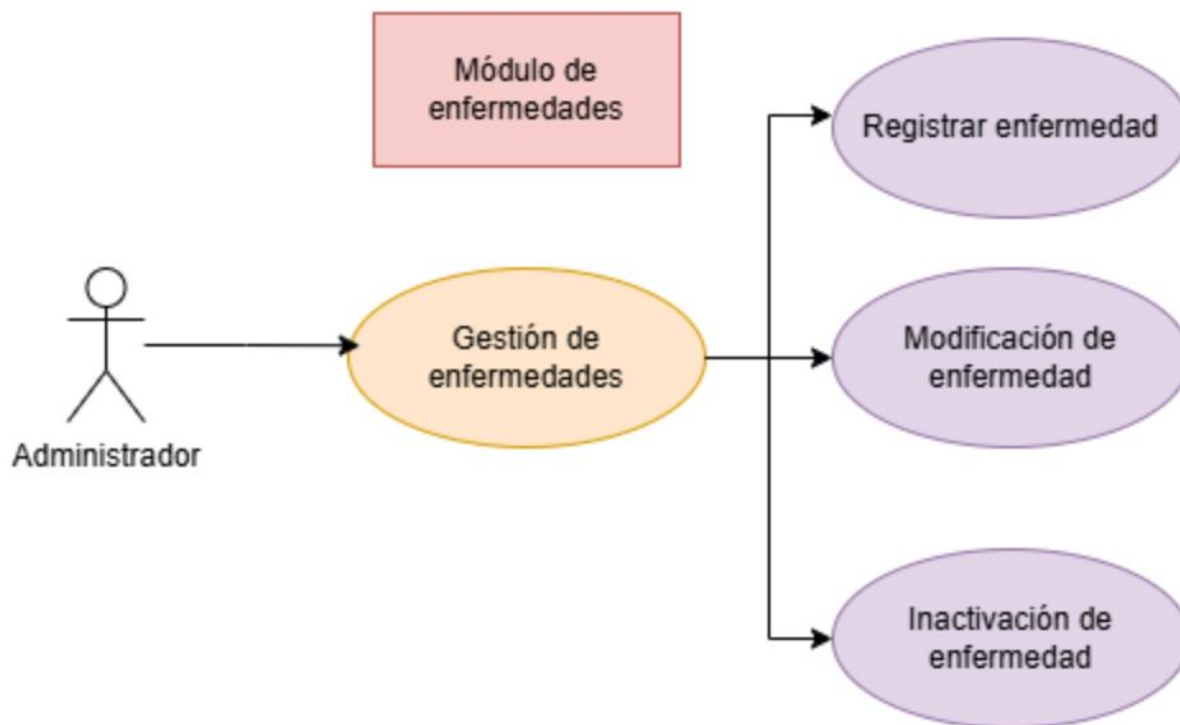


Figura 9. Caso de uso enfermedades

4.2.9. Módulo de actividades

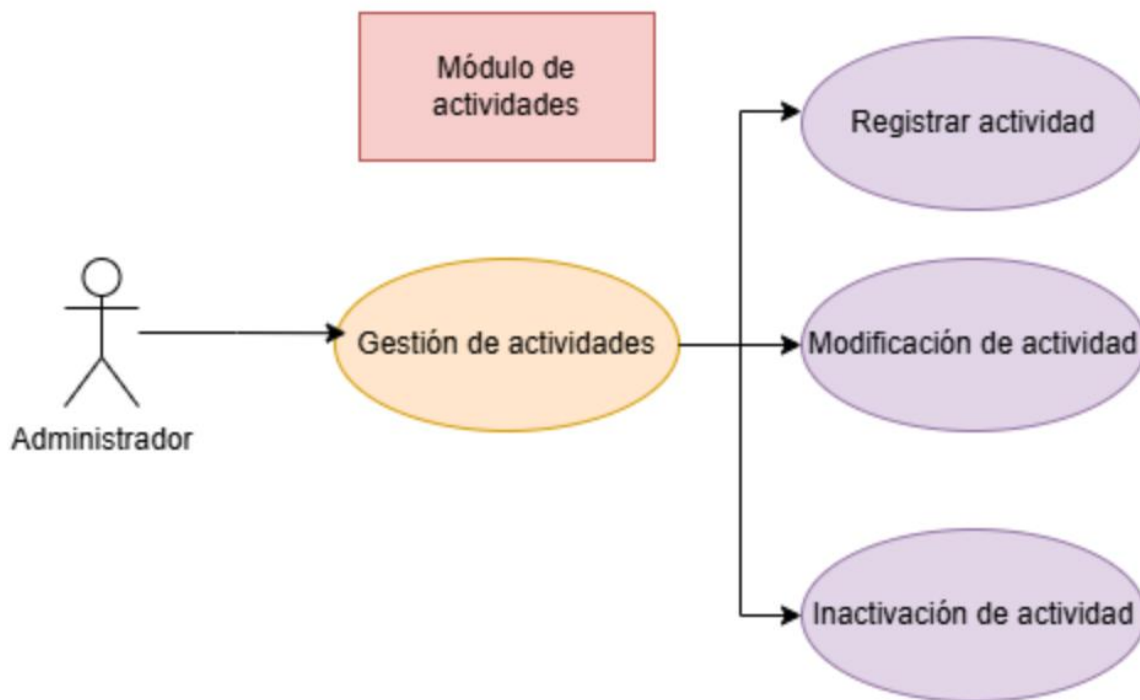


Figura 10. Caso de uso actividades

4.2.10. Módulo de membresía

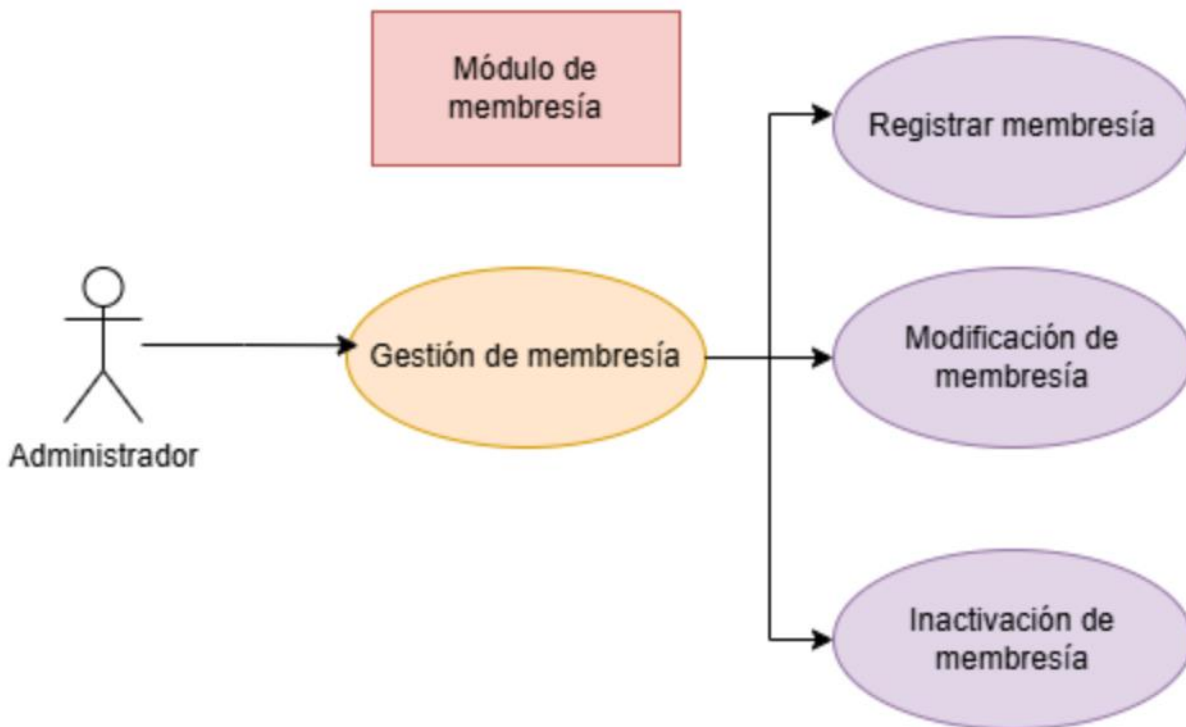


Figura 11. Caso de uso membresía

4.2.11. Módulo de usuarios

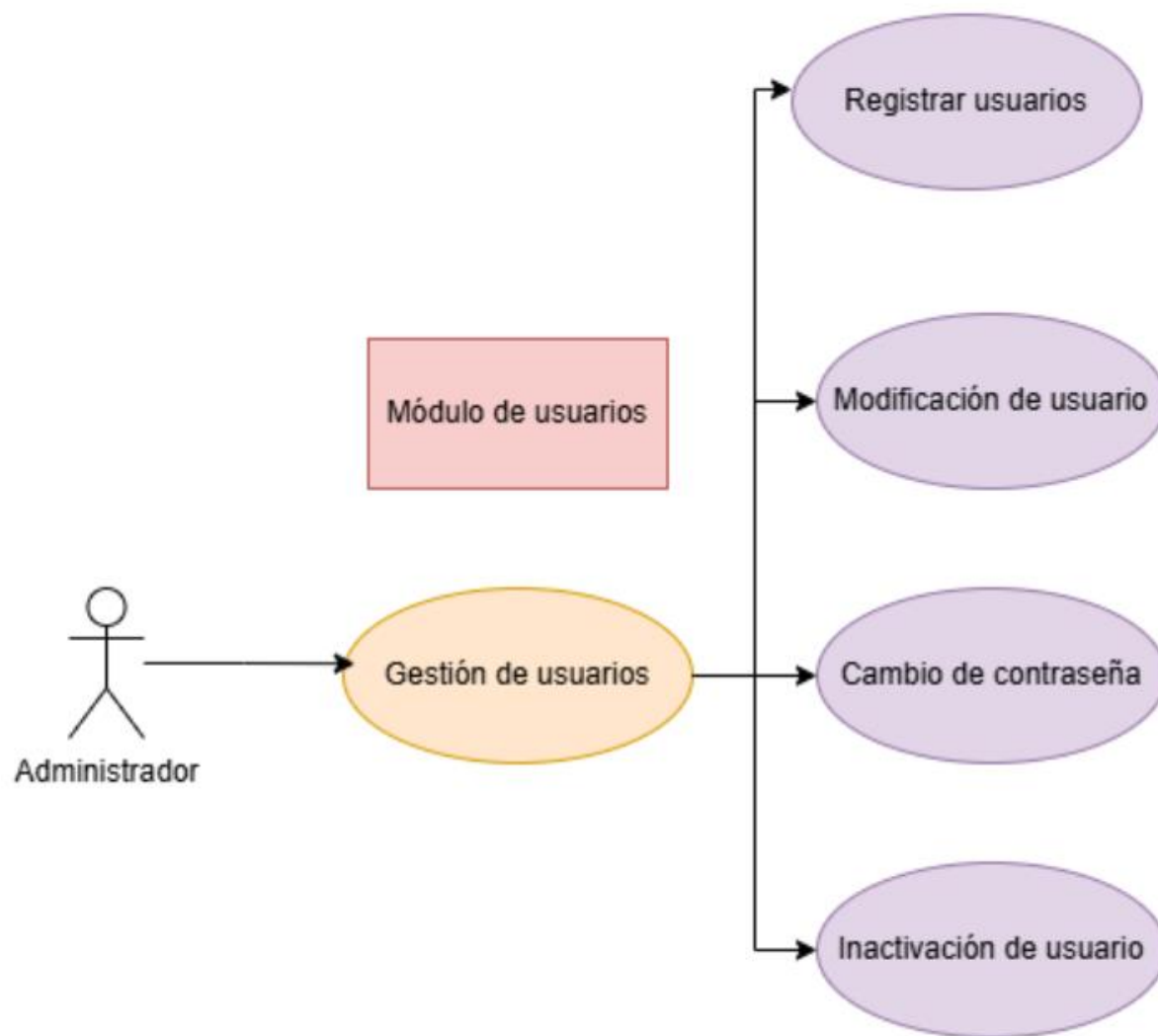


Figura 12. Caso de uso usuarios

4.2.12. Módulo de rutina

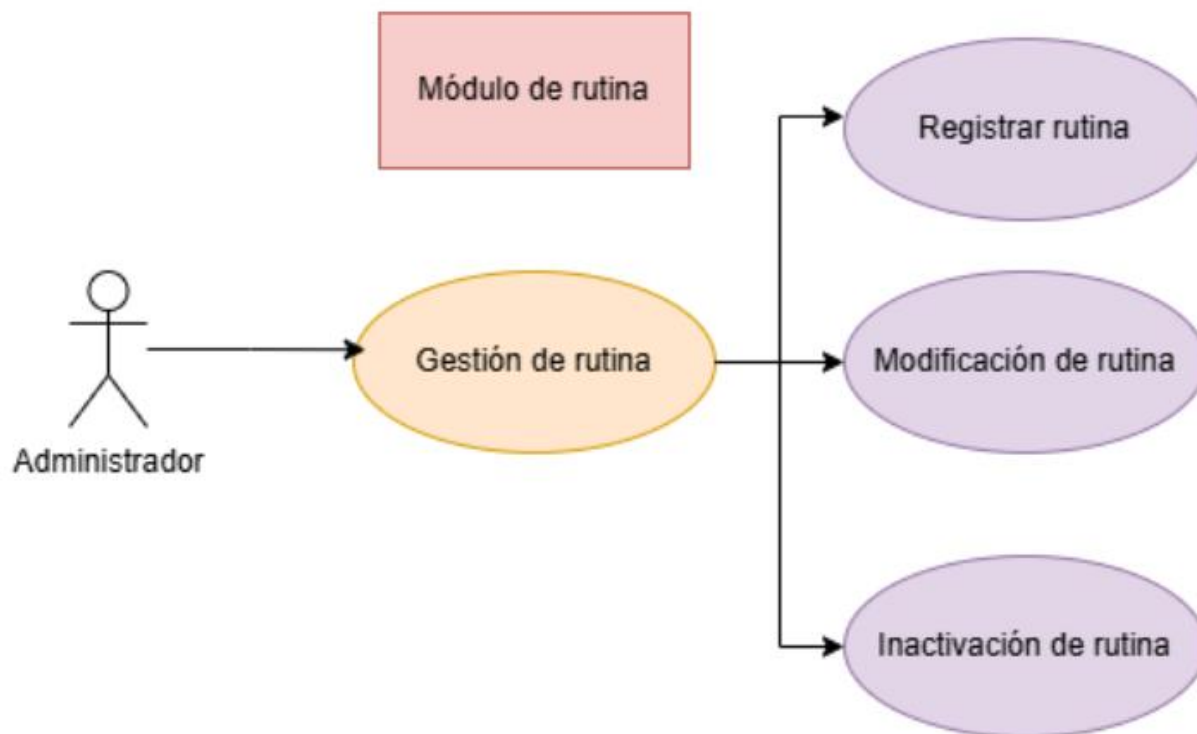


Figura 13. Caso de uso rutina

4.2.13. Módulo de enfermedades de la persona

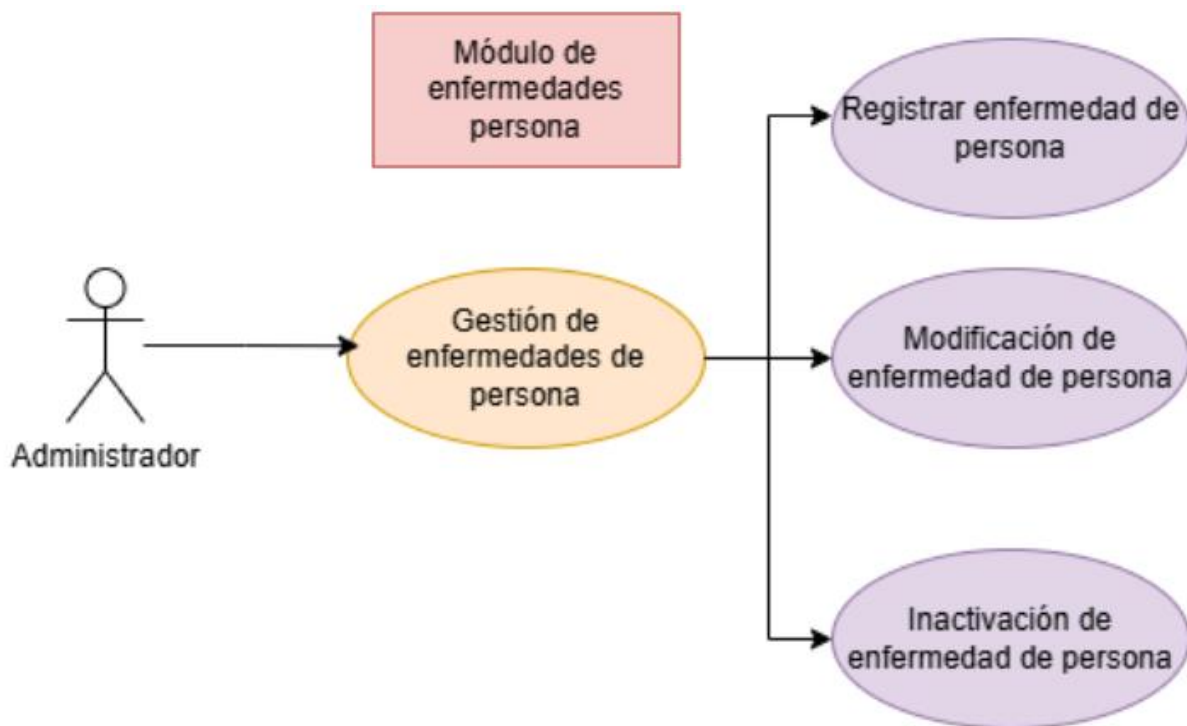


Figura 14. Caso de uso enfermedades de la persona

4.2.14. Módulo de paquete

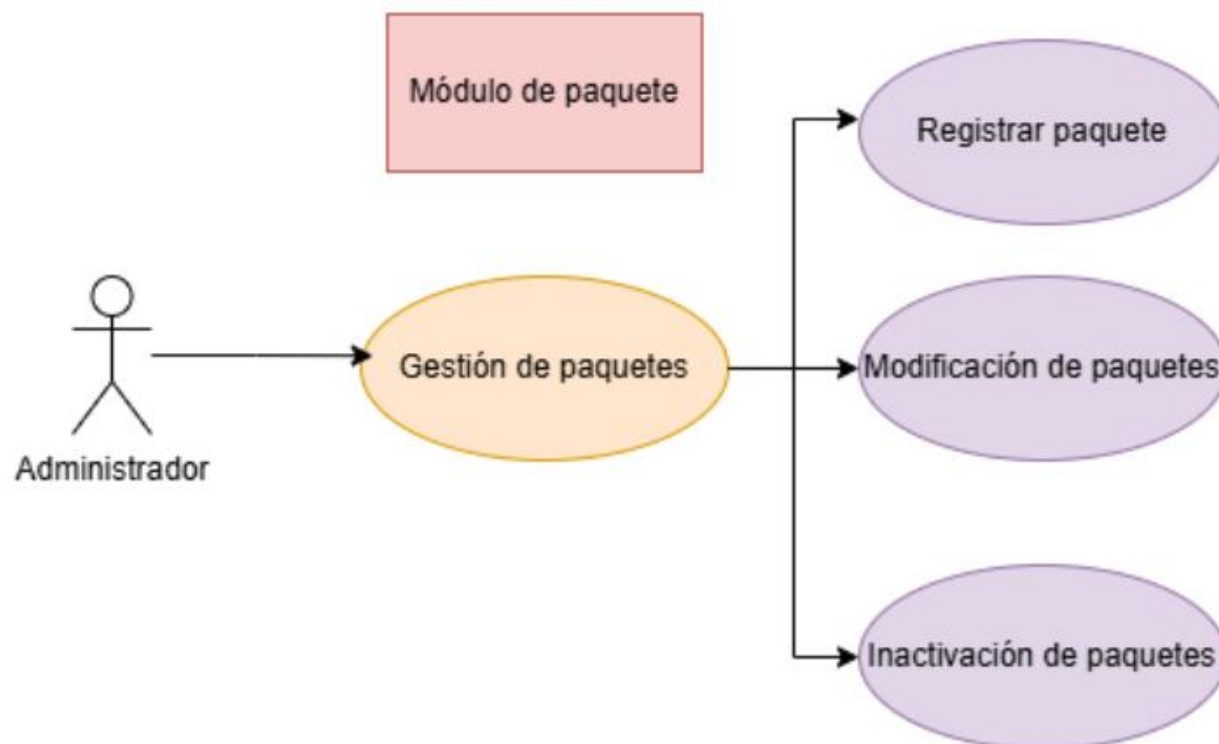


Figura 15. Caso de uso paquete

4.2.15. Módulo de reserva

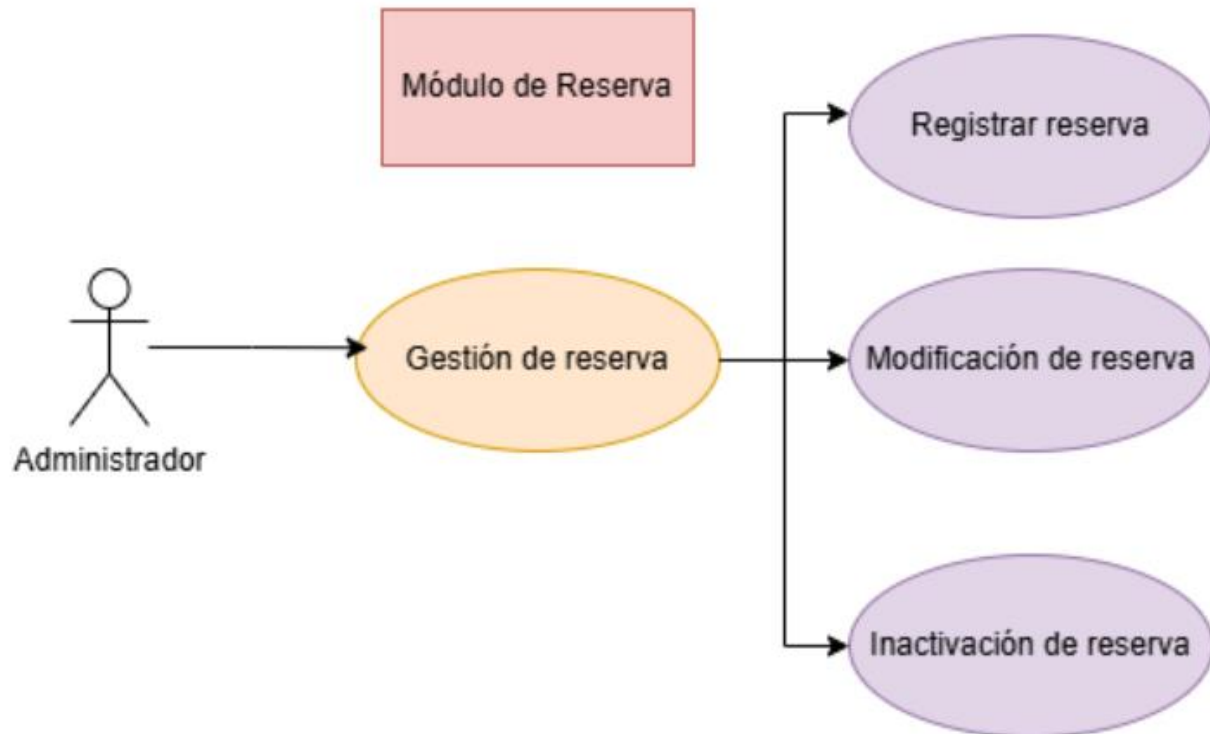


Figura 16. Caso de uso reserva (administrador)

4.3. Pantallas de la solución implementada

Completando la solución propuesta, se presenta pantallas que forman parte del SIGEK que cumplen con las necesidades establecidas por la organización.

Iniciando con el login del sistema que permite a los usuarios acceder a las funcionalidades ya sean de clientes o administradores, esto lo podemos observar en la Figura 17.

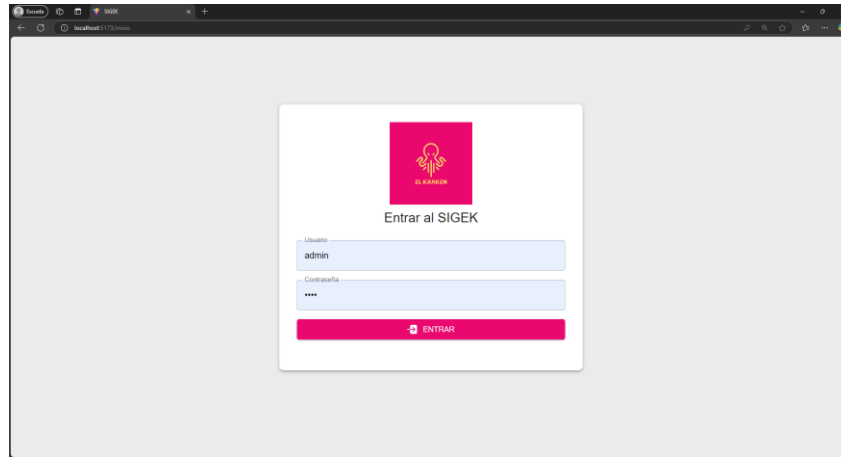


Figura 17. Pantalla login

Al iniciar sesión como usuario administrador, se nos muestra la pantalla de bienvenida, la cual se puede observar en la Figura 18.

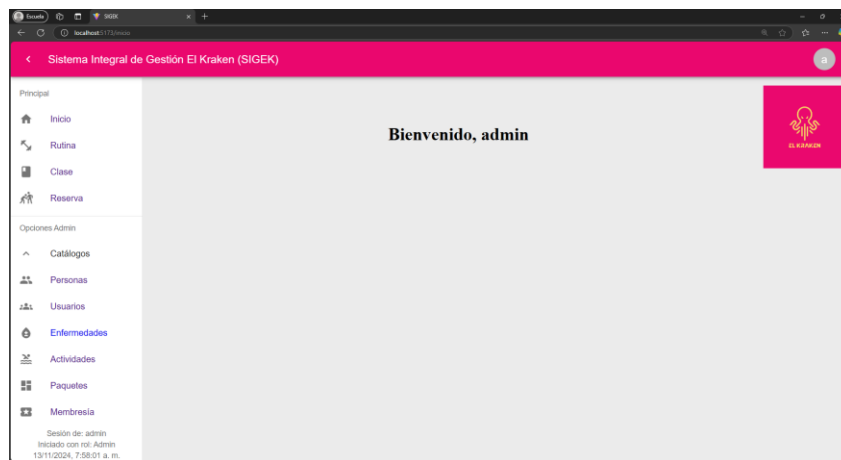


Figura 18. Pantalla bienvenida administrador

Además, al iniciar sesión como cliente se muestra la pantalla de inicio mostrada en la Figura 19.

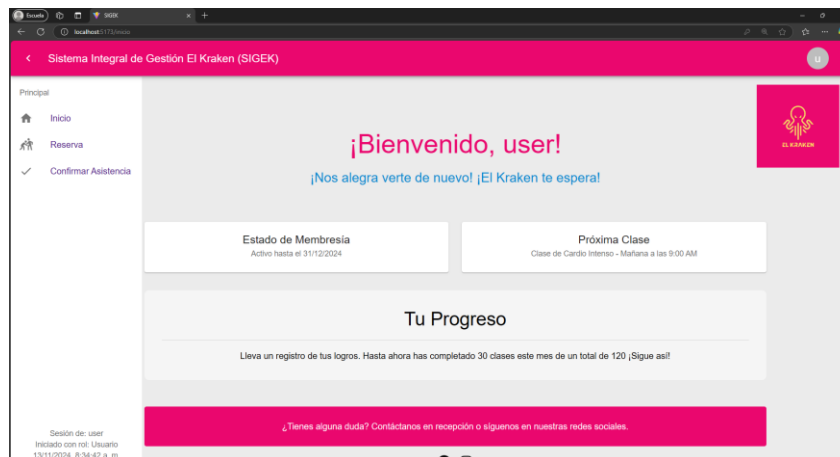


Figura 19. Pantalla bienvenida usuario

Seguidamente, podemos observar al costado izquierdo el menú ilustrado en la Figura 20 que muestran las funcionalidades a las que puede acceder para realizar las gestiones necesarias.

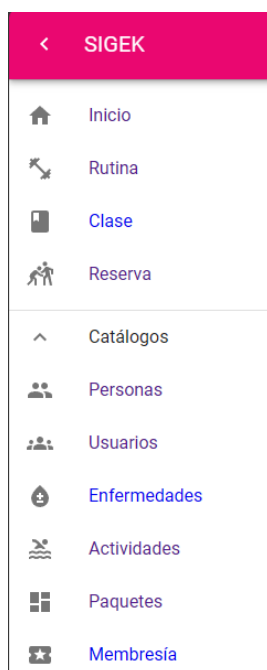
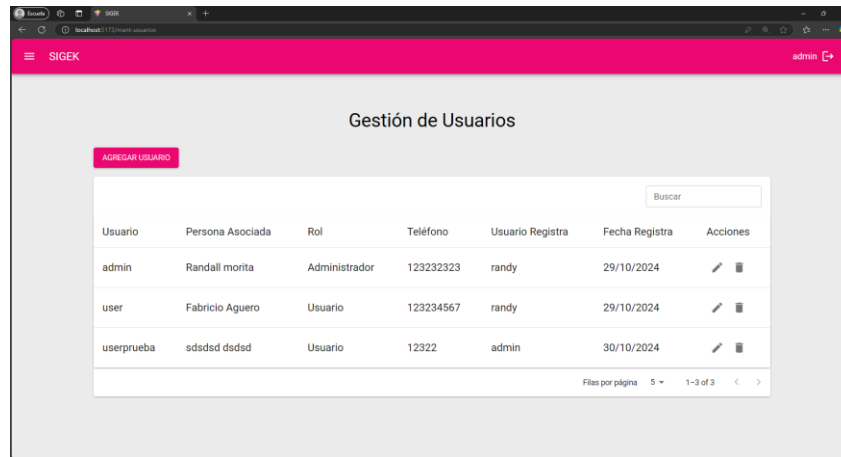


Figura 20. Menú administrador

Como usuario administrador, es posible realizar la gestión de los usuarios que pueden acceder al sistema, para ello se cuenta con el módulo de usuario como se ilustra en la Figura 21 que permite realizar dichas acciones.



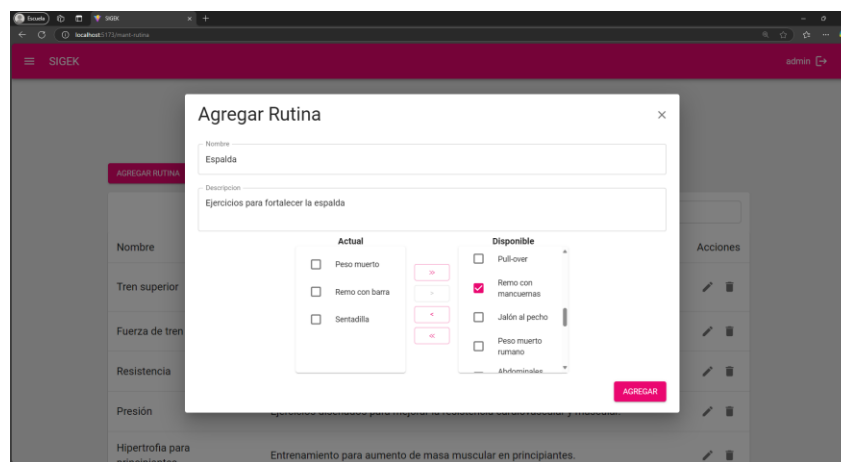
The screenshot shows the 'Gestión de Usuarios' (User Management) page in the SIGEX system. It features a search bar and a table with the following data:

Usuario	Persona Asociada	Rol	Teléfono	Usuario Registra	Fecha Registra	Acciones
admin	Randall morita	Administrador	123232323	randy	29/10/2024	[Edit] [Delete]
user	Fabricio Agüero	Usuario	123234567	randy	29/10/2024	[Edit] [Delete]
userprueba	sdsdsd dsdsd	Usuario	12322	admin	30/10/2024	[Edit] [Delete]

At the bottom of the table, there is a pagination control showing 'Filas por página: 5' and '1-3 of 3'.

Figura 21. Pantalla gestión de usuarios

Para la gestión de rutinas, como se observa en la Figura 22, es posible manipular rutinas que posteriormente pueden ser implementadas en las clases.



The screenshot shows the 'Agregar Rutina' (Add Routine) form. It includes the following fields and options:

- Nombre:** Espalda
- Descripción:** Ejercicios para fortalecer la espalda
- Actual:**
 - Peso muerto
 - Remo con barra
 - Sentadilla
- Disponibles:**
 - Pullover
 - Remo con mancuernas
 - Jalón al pecho
 - Peso muerto rutinario

Navigation arrows are present between the 'Actual' and 'Disponibles' sections. A red 'AGREGAR' button is located at the bottom right of the form.

Figura 22. Pantalla agregar rutina

Relacionado con lo anterior, es posible gestionar clases como se observa en la Figura 23, permitiendo así definir clases que contienen una rutina.

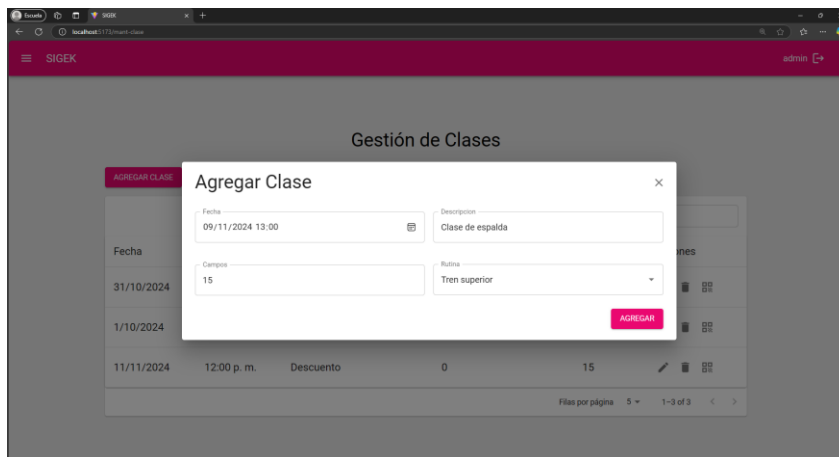


Figura 23. Pantalla gestión de clases

Es posible definir paquetes establecidos como se muestra en la Figura 24, esto para asignar actividades que van a tener los paquetes.

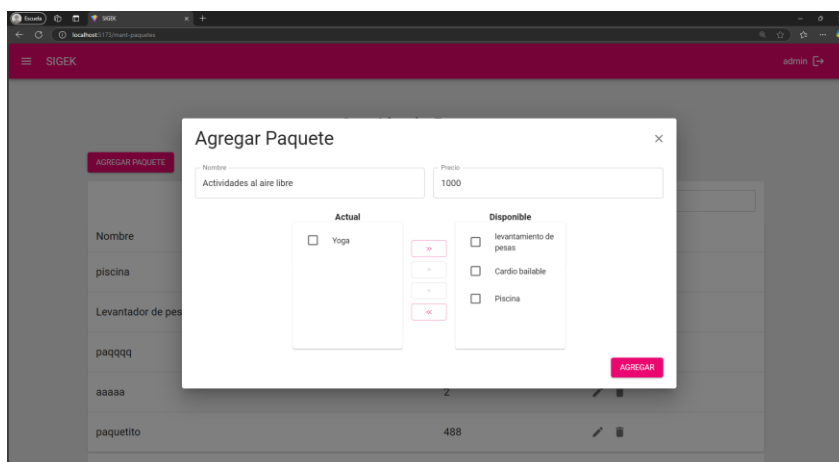


Figura 24. Pantalla gestión de paquetes

Que estas a su vez van a poder ser utilizadas en sus correspondientes membresías como se observa en la Figura 25, para poder ser asignadas a los usuarios correspondientes.

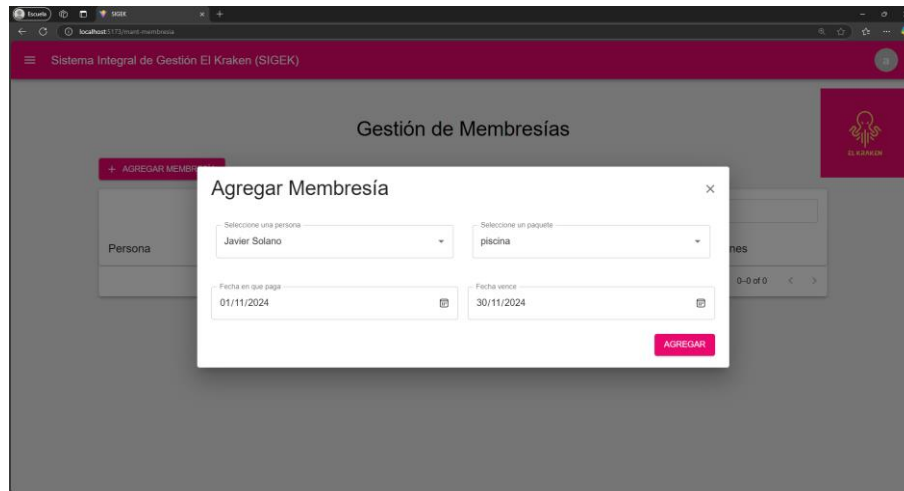


Figura 25. Pantalla gestión de membresía

Como cliente se pueden realizar la gestión de reservas de clases previamente establecidas, como se muestra en la Figura 26.

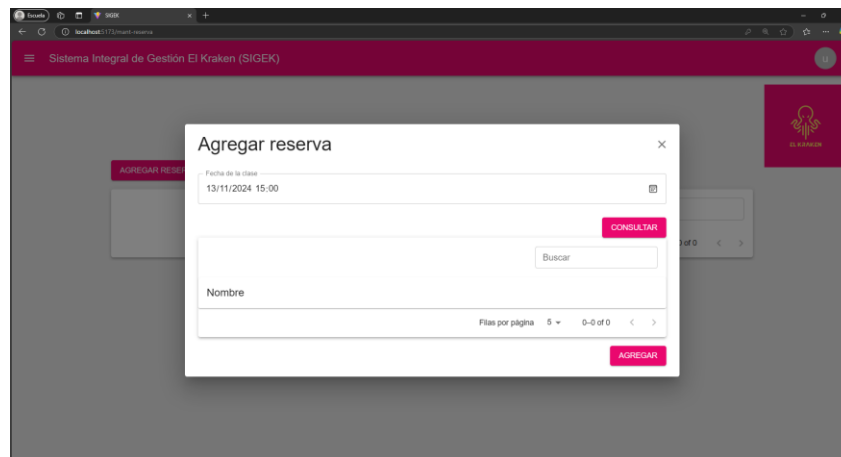


Figura 26. Pantalla gestión de reserva

Luego en el momento de llegar a la clase reservada podemos marcar la asistencia mediante el código QR que el administrador puede generar al consultar la clase y el cliente escanear en la pantalla que se muestra en la Figura 27.

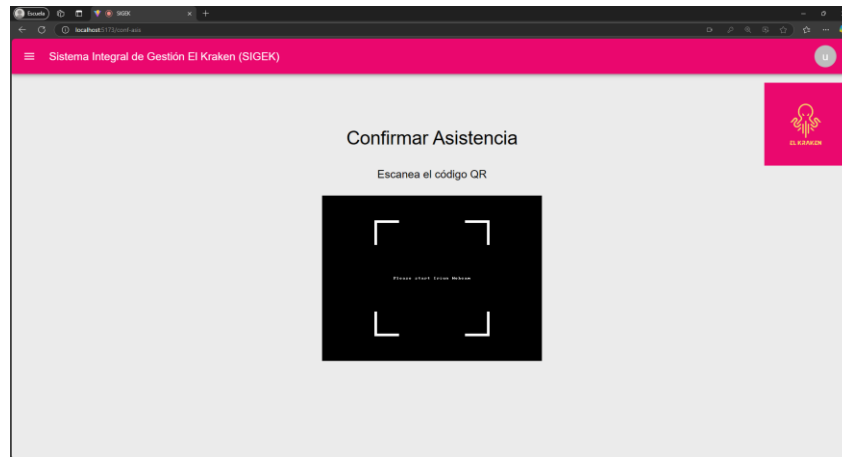


Figura 27. Pantalla marcar asistencia QR

5. Conclusiones

En conclusión, el proceso de definir las características y funciones que el cliente desea para el SIGEK permitió obtener una comprensión detallada de requisitos del cliente, así como la demostración frecuente de los avances del sistema los cuales fueron fundamentales para validar las expectativas y asegurar que fueran coherentes con las necesidades reales del cliente

La construcción de una base de datos utilizando diagramas y un motor de base de datos ha permitido estructurar una solución robusta para el control y manejo de la información capaz de satisfacer las necesidades del negocio. Esta estructura es capaz de soportar el crecimiento del negocio, asegurando confidencialidad, disponibilidad de la información y una base para la evolución posterior del sistema.

Utilizar las tecnologías de React y .Net ha facilitado la elaboración de un sistema robusto en respuesta y manejo de los datos. React en la parte del front-end, es versátil al permitir la creación de componentes reutilizables que permiten manejar las entidades necesarias para el despliegue de la información. Por otro lado, .Net, se utiliza una

arquitectura N-capas, que facilita el manejo de la información proveniente del front-end y base de datos.

El constante contacto con el cliente al aplicar la metodología ágil scrum, permitió realizar una integración del sistema desde etapas tempranas del desarrollo, por lo que, en la etapa de implementación, se pudo observar un gran conocimiento del cliente sobre la solución entregada.

6. Recomendaciones

Se recomienda mantener una comunicación abierta con reuniones periódicas con el cliente durante todo el proceso de incorporación de mejoras y nuevas funcionalidades en el sistema, de la mano de prototipos visuales y herramientas que permitan validar los requerimientos e implementar ajustes en un tiempo prudente.

Para construir la base de datos de manera efectiva acorde a las necesidades del negocio se recomienda empezar minuciosamente con el diseño de diagramas conocidos que se consideren necesarios, que ayuden a identificar todas las entidades posibles del negocio y sus respectivas relaciones para validar el buen entendimiento de este, así como posibles retos futuros.

Tener en cuenta las tecnologías óptimas para realizar la solución desde etapas tempranas del proyecto ayuda a elaborar una solución que se adapte a las necesidades del negocio, por lo que, se recomienda investigar de diversas tecnologías para tener una noción más amplia y poder responder de la mejor manera a los requerimientos que se presenten.

Es recomendable aplicar las metodologías según diga su manual, debido a que, tener claras las bases es una gran ventaja para resolver las situaciones que se presenten durante el desarrollo del proyecto, además estas metodologías sirven de apoyo estructural para poder realizar de manera exitosa proyectos de diversa índole.

7. Bibliografía

- IBM. (2024). *Defining use cases*. IBM Documentation.
<https://www.ibm.com/docs/es/product-master/12.0.0?topic=processes-defining-use-cases>
- PMO Informatica. (2017). *Requerimientos funcionales: Ejemplos*. PMO Informatica.
<https://www.pmoinformatica.com/2017/02/requerimientos-funcionales-ejemplos.html>
- PMO Informatica. (2015). *Requerimientos no funcionales: Ejemplos*. PMO Informatica.
<https://www.pmoinformatica.com/2015/05/requerimientos-no-funcionales-ejemplos.html>
- Oracle. (s.f.). *What is a relational database?* Oracle.
<https://www.oracle.com/ar/database/what-is-a-relational-database/>
- Universidad Europea. (2022). *Programación orientada a objetos*. Universidad Europea.
<https://universidadeuropea.com/blog/programacion-orientada-objetos/>
- Lucidchart. (s.f.). *¿Qué es el Lenguaje Unificado de Modelado (UML)?* Lucidchart.
<https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml>

Pérez, M. (2011). Microsoft SQL Server 2008 R2. Motor de base de datos y administración. RC Libros.

¿Qué es una API? - Explicación de interfaz de programación de aplicaciones - AWS. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/api/>

De Enciclopedia Significados, E. (2024, 6 junio). *Qué es un Usuario (concepto y definición)*. Enciclopedia Significados. <https://www.significados.com/usuario/>

Casero, A. (2024, 8 abril). ¿Qué es TypeScript? | KeepCoding Bootcamps. *KeepCoding Bootcamps*. https://keepcoding.io/blog/typescript/#%C2%BFQue_es_TypeScript

A, D., & A, D. (2023, 29 junio). *Qué es React: definición, características y funcionamiento*. Tutoriales Hostinger. <https://www.hostinger.es/tutoriales/que-es-react>