



Universidad Nacional del Nordeste



Facultad de Ciencias Exactas y Naturales y Agrimensura

Ingeniería del Software II

Año:2024

Proyecto: Gymsy

Profesores: Maria de los Angeles Ferraro,Laura Ines Gomez Solis.

Integrantes: Gonzalez Rodrigo Alejandro,
Gonzalez Jeremias Ezequiel,
Maciel Lautaro,
Lezana Mauricio Sebastian.

ÍNDICE

Índice de Tabla.....	3
Índice de Figuras.....	4
1. Introducción.....	5
1.1 Breve estado del arte.....	5
1.2 Objetivo.....	5
1.3 Fundamentación.....	5
2. Metodología.....	6
2.1 Ciclo de vida.....	6
2.2 Método de Educación.....	7
2.3 Arquitectura Utilizada.....	8
2.4 Especificación de Requerimientos de Software (ERS IEEE-STD-830-1998).....	9
2.4.1 Introducción.....	9
2.4.2 Propósito.....	9
2.4.3 Alcance.....	9
2.4.4 Personal Involucrado.....	9
2.4.5 Referencias.....	11
2.5 Requisitos específicos.....	11
2.5.1 Requerimientos Funcionales.....	11
2.5.2 Requerimientos no funcionales.....	12
2.5.3 Requerimientos no Funcionales Producto.....	12
2.5.4 Requerimientos no Funcionales Externos.....	12
2.5.5 Requerimientos no Funcionales Organizacional.....	13
2.5.6 Otros requisitos.....	13
2.6 Análisis de Riesgo.....	15
2.6.1 Riesgo de producto.....	15
2.6.2 Riesgo de proyecto.....	15
2.6.3 Riesgo de Negocio.....	16
2.6 Historias de Usuario.....	16
Crear Perfil de Instructor.....	19
2.7 Casos de uso.....	23
2.8 Conversación y diagrama de secuencia.....	24
2.8.1 Conversación: Caso de uso “Crear Perfil de Instructor”.....	24
2.8.1 Diagrama de secuencia “Crear Perfil Instructor”.....	25
2.8.2 Conversación: Caso de uso “Agregar Progreso”.....	27
2.8.2 diagrama de secuencia “Agregar Progreso”.....	28
2.8.3 Conversación: Caso de uso “Crear Plan”.....	29
2.8.3 diagrama de secuencia “Crear Plan”.....	30
2.9 Contratos de operaciones.....	31
2.10 Diagrama DER.....	32
2.10.1 Diccionario de datos.....	33
2.10.2 Mapa físico de la Base de datos.....	38
2.11 Diagrama de clases.....	39
3. Herramientas y/o Lenguajes de Programación.....	39

3.1 Descripción.....	39
4. Resultados.....	41
Plan de pruebas: Agregar Instructor.....	41
Plan de pruebas: Agregar Progreso.....	46
Plan de pruebas: Agregar Plan.....	48
Maquetado de pantallas.....	50
5. Referencia.....	54

Índice de Tabla

Tabla 1.....	12
Tabla 2.....	13
Tabla 3.....	13
Tabla 4.....	16
Tabla 5.....	17
Tabla 6.....	18
Tabla 7.....	18
Tabla 8.....	19
Tabla 9.....	19
Tabla 10.....	24
Tabla 11.....	27
Tabla 12.....	29
Tabla 13.....	33
Tabla 14.....	34
Tabla 15.....	35
Tabla 16.....	35
Tabla 17.....	36
Tabla 18.....	37
Tabla 19.....	38
Tabla 20.....	41
Tabla 21.....	46
Tabla 22.....	48

Índice de Figuras

Figura 1.....	14
Figura 2.....	14
Figura 3.....	20
Figura 4.....	20
Figura 5.....	20
Figura 6.....	20
Figura 7.....	21
Figura 8.....	21
Figura 9.....	21
Figura 10.....	21
Figura 11.....	22
Figura 12.....	24
Figura 13.....	24
Figura 14.....	25
Figura 15.....	25
Figura 16.....	27
Figura 17.....	27
Figura 18.....	28
Figura 19.....	29
Figura 20.....	29
Figura 21.....	30
Figura 22.....	31
Figura 23.....	37
Figura 24.....	38
Figura 25.....	51
Figura 26.....	52
Figura 27.....	52
Figura 28.....	53
Figura 29.....	53
Figura 30.....	54
Figura 31.....	54

1. Introducción

Gymsy es una aplicación para administrar tu gimnasio de manera eficiente y efectiva. Ya no se tendrá que lidiar con montañas de papeleo, largas horas de seguimiento manual o confusiones en la programación de pagos. Permite gestionar todos los aspectos de tu gimnasio en un solo lugar. Desde el registro y seguimiento de clientes hasta la programación y la gestión de pagos, nuestra aplicación simplifica cada paso del proceso.

1.1 Breve estado del arte

Es esencial que los servicios relacionados con la salud y el bienestar se adapten constantemente a las demandas de la población, aprovechando las tecnologías disponibles y su rápido avance. En este sentido, el uso de aplicaciones y software para la gestión de gimnasios se ha vuelto cada vez más común, permitiendo una administración más eficiente y efectiva de estos establecimientos.

En países desarrollados, la modernización de los servicios de salud y bienestar ha llevado a la adopción generalizada de aplicaciones y software especializados. Sin embargo, muchas de estas soluciones son costosas y pueden no adaptarse completamente a las necesidades de todos los gimnasios.

Gymsy surge como una solución para abordar esta problemática, ofreciendo una plataforma integral para la gestión de gimnasios que simplifica cada aspecto del proceso, desde el registro y seguimiento de clientes hasta la programación y gestión de pagos. Al aprovechar la información disponible y utilizarla de manera eficiente, Gymsy busca reducir los costos de desarrollo y ofrecer una solución accesible para gimnasios de todos los tamaños.

1.2 Objetivo

El objetivo principal de Gymsy es proporcionar una herramienta informática que permita una gestión eficiente de los gimnasios, simplificando las tareas administrativas y mejorando la experiencia tanto para los propietarios como para los clientes.

1.3 Fundamentación

La necesidad de una gestión más eficiente en los gimnasios se hace evidente en la falta de herramientas adecuadas para la comunicación y coordinación entre diferentes áreas dentro del establecimiento. Gymsy busca abordar esta problemática mediante el desarrollo de una plataforma que centre la información y facilite la comunicación entre los distintos departamentos del gimnasio, optimizando así el trabajo y mejorando la experiencia del cliente.

2. Metodología

2.1 Ciclo de vida

El proceso de producción de cualquier aplicación informática lleva consigo realizar una serie de tareas repartidas en cinco etapas, llamadas: Análisis, Diseño, Codificación, Pruebas. A estas cuatro etapas se las conoce como ciclo de vida de un producto software o, dicho de otra forma, el ciclo de vida de un programa son las distintas etapas por las que éste tiene que pasar durante su existencia. Para este proyecto utilizaremos la metodología Scrum, que es un marco de trabajo ágil para la gestión y desarrollo de proyectos. Se basa en la colaboración, la adaptabilidad y la entrega incremental del producto. Elegimos Scrum por encima de otras metodologías debido a que proporciona las siguientes ventajas:

1. **Flexibilidad:** Scrum es una metodología de gestión de proyectos muy flexible que permite adaptarse fácilmente a cambios en los requisitos del proyecto. Los sprints cortos y la capacidad de priorizar y planificar tareas durante el proyecto permiten a los equipos de Scrum responder rápidamente a cambios en el proyecto.
2. **Colaboración:** Scrum fomenta una colaboración efectiva entre los miembros del equipo. La estructura de equipo autoorganizado de Scrum fomenta una comunicación más abierta y una mayor colaboración entre los miembros del equipo, lo que puede resultar en una mejor calidad del trabajo y una mayor satisfacción del equipo.
3. **Entregas frecuentes:** Scrum promueve la entrega frecuente y continua de incrementos de trabajo completados y evaluables, lo que permite a los interesados obtener comentarios tempranos sobre el producto y realizar ajustes en consecuencia.
4. **Mayor control del proyecto:** Scrum proporciona una mayor visibilidad y control sobre el progreso del proyecto, ya que los equipos de Scrum realizan reuniones diarias, reuniones de revisión de sprint y reuniones de retrospectiva para revisar el trabajo completado y planificar el siguiente sprint.
5. **Mejora continua:** Scrum fomenta la mejora continua en el proceso de desarrollo de software, lo que puede llevar a una mayor eficiencia y una mayor calidad del producto a lo largo del tiempo.

2.2 Método de Educción

Se determinó que el método de educación más apropiado para el proyecto es "Entrevistas", ya que este método permite conocer el estado actual del sistema, metas organizacionales, identificar requisitos innecesarios, etc. Lo que a su vez simplifica el proceso de desarrollo y mejora la calidad del producto final.

La estructura de la entrevista es en forma de "Diamante" para la conformación del método de educación seleccionado lo que significa que las preguntas empezaran siendo específicas, a medida que avance la entrevista preguntas más generales y para finalizar preguntas específicas.

A continuación, un extracto de la entrevista llevada a cabo:

1. ¿En el sistema se registrará a los clientes?

- "Sí, el sistema debe contar con un formulario de registro para nuevos clientes, la información."

2. ¿Qué información debe tener un plan?

- "Debe tener el título, la descripción, precio e instructor a cargo."

3. ¿Cuando se ingrese un dato erróneo como quiere el sistema lo informe?

- "El sistema debe validar lo ingresado por el operador y mostrar un mensaje de error en caso de datos no válidos."

4. El sistema debe contar con una forma de identificación única ¿Prefiere que se haga automáticamente o manualmente?

- "Que el sistema lo genere automáticamente."

5. ¿Cómo se gestionan los roles de los usuarios del sistema?

- "Deben existir de forma predeterminada un administrador y un recepcionista, luego el administrador puede agregar más instructores y estos a los clientes algo automático."

6. ¿Cómo se debe visualizar a los clientes?

- "El sistema debe permitir la visualización de una lista de clientes asociados a un instructor específico."

7. ¿Puede el instructor agregar clientes y realizarles seguimiento?

- "Sí, el instructor tiene la capacidad de agregar clientes y hacerles seguimiento."

8. ¿Cómo se deben visualizar los usuarios?

- "El sistema proporciona la opción de visualizar una lista de todos los usuarios registrados, junto con sus roles asignados."

9. ¿Desde que sistema operativo cuentan para usar el sistema?

- "Windows 7 en adelante."

10. ¿Cómo le gustaría que fuera la aplicación?

- "La aplicación debe ser fácil de usar y contar con una interfaz intuitiva."

11. Respecto a la fluidez de las acciones del usuario ¿Como deberían ser?

- "La aplicación debe tener un tiempo de respuesta rápido para las acciones del usuario."

12. Sobre la fecha y hora que se manejen el sistema ¿Debe ser la del sistema operativo?

- "Sí, la aplicación debe integrarse con el calendario del sistema operativo."

13. ¿Cuáles son las normas que debe cumplir la aplicación?

- "La aplicación debe cumplir con legislación, regulaciones y leyes locales, principalmente sobre privacidad de datos y todo respecto a temas referentes a gimnasios."

14. ¿La aplicación debe garantizar la privacidad absoluta de la información de los clientes?

- "Sí, la aplicación garantiza la privacidad absoluta de la información de los clientes."

2.3 Arquitectura Utilizada

La arquitectura utilizada fue MVP(Modelo, Vista, Presentador)

Se emplea principalmente en entornos donde se requiere una clara separación de responsabilidades entre la lógica de negocio, la presentación de la interfaz de usuario y la gestión de eventos.

Aquí hay algunas razones por las que podrías considerar utilizar la arquitectura MVP:

- **Separación de responsabilidades:** MVP permite separar claramente la lógica de negocio (modelo), la presentación de la interfaz de usuario (vista) y la gestión de eventos (presentador). Esto facilita la comprensión del código y el mantenimiento a largo plazo.
- **Facilita las pruebas unitarias:** Al separar la lógica de presentación de la lógica de negocio, se facilita la escritura de pruebas unitarias para cada componente por separado. Esto mejora la calidad del código y facilita la detección de errores.
- **Reutilización del código:** La separación de la lógica de presentación y la lógica de negocio permite reutilizar los componentes de manera más efectiva. Por ejemplo, puedes utilizar la misma lógica de negocio con diferentes interfaces de usuario.

- **Adaptabilidad a cambios en la interfaz de usuario:** Si necesitas realizar cambios en la interfaz de usuario, la arquitectura MVP te permite hacerlo sin afectar la lógica de negocio subyacente. Esto es útil en situaciones donde la interfaz de usuario debe ser actualizada con frecuencia.
- **Escalabilidad:** MVP es una arquitectura escalable que puede adaptarse fácilmente a proyectos de diferentes tamaños y complejidades. Desde aplicaciones pequeñas hasta grandes sistemas empresariales, MVP puede proporcionar una base sólida para el desarrollo.

En resumen, la arquitectura Modelo-Vista-Presentador (MVP) es una opción sólida para proyectos donde se valora la separación de responsabilidades, la facilidad de prueba, la reutilización del código y la capacidad de adaptarse a cambios en la interfaz de usuario.

2.4 Especificación de Requerimientos de Software (ERS IEEE-STD-830-1998).

2.4.1 Introducción

El presente documento es una Especificación de Requisitos de Software perteneciente al desarrollo de un sistema para la gestión de clientes, ventas y pagos de la aplicación de Gymsy. Está estructurado según las directivas dadas por el estándar de IEEE 830. Esta especificación está dirigida a los desarrolladores del Sistema, al equipo de calidad, usuarios finales y tiene como objetivo analizar y documentar los requisitos funcionales y no funcionales del futuro sistema.

2.4.2 Propósito

El propósito de la propuesta es la de realizar una aplicación especialmente diseñada para la empresa “Gymsy” en el cual se brinda el servicio de compras online. El objetivo es informatizar el proceso de ventas, pagos a los empleados y administración de planes de entrenamiento a fin agilizar los procesos y reducir costos.

2.4.3 Alcance

El sistema permitirá el alta y baja de planes de entrenamiento, instructores y clientes. Generar informes de cantidad de clientes por instructor, ganancias en el tiempo al administrador. Agregar los progresos del cliente. Visualizar clientes, instructores, planes y pagos realizados dentro de Gymsy.

2.4.4 Personal Involucrado

Nombre	Gauna Octavio
Rol	Analista, diseñador y programador
Responsabilidad	Diseñador/Programador
Información contacto	gaunavictoroctavio@gmail.com

Nombre	Jeremías Ezequiel Gonzales
Rol	Analista, diseñador y programador
Responsabilidad	Diseñador/Programador
Información contacto	jeremiasgonzalez7464@gmail.com

Nombre	Gonzalez Rodrigo Alejandro
Rol	Analista, diseñador y programador
Responsabilidad	Diseñador/Programador
Información contacto	gonzlrodrigo@gmail.com

Nombre	Maciel Lautaro
Rol	Analista, diseñador y programador
Responsabilidad	Diseñador/Programador
Información contacto	lautaromaciel77@gmail.com

Nombre	Lezana Mauricio Sebastian
Rol	Analista, diseñador y programador
Responsabilidad	Diseñador/Programador
Información contacto	lezanamauricio86@gmail.com

2.4.5 Definiciones, acrónimos y abreviaturas

Nombre	Descripción
Usuario	Persona que usará el sistema para gestionar procesos
RF	Requerimiento Funcional
RNF	Requerimiento No Funcional

Usuario: Persona que usará el sistema para gestionar procesos

RF: Requerimiento Funcional

RNF: Requerimiento No Funcional

2.4.5 Referencias

Título del Documento	Referencia
Standard IEEE 830 1998	IEE
SIS-I	Sistema de Información Web para la Gestión de Procesos Administrativos y Académicos
RF	Requerimiento Funcional
RF	Requerimiento No Funcional

2.5 Requisitos específicos

2.5.1 Requerimientos Funcionales

RF#01: El sistema debe proporcionar un formulario para el registro de nuevos clientes.

RF#02: El formulario de creación de un nuevo plan debe incluir campos obligatorios, como el título, la descripción, precio, instructor a cargo, imagen (mínimo 2).

RF#03: El sistema debe validar los campos ingresados por el usuario y mostrar un mensaje de error si se ingresan datos inválidos.

RF#04: El sistema debe generar automáticamente un identificador único para cada nuevo usuario/plan creado.

RF#05: El sistema debe tener los roles administrador, recepcionista, instructor y cliente, los primeros dos son predeterminados, clientes e instructores se asignan cuando se crean sus perfiles automáticamente.

RF#08: El sistema debe permitir la visualización de una lista de todos los clientes asociados a un cierto instructor.

RF#10: El usuario instructor debe poder agregar clientes y hacerles un seguimiento.

RF#11: El sistema debe permitir la visualización de una lista de todos los usuarios registrados, con sus roles asignados.

2.5.2 Requerimientos no funcionales

2.5.3 Requerimientos no Funcionales Producto

RNF	Descripción del Requerimiento	Clasificación
RNF-01	La aplicación deberá ser compatible con Windows 7 en adelante.	Portabilidad
RNF-02	La aplicación deberá ser fácil de usar y con una interfaz intuitiva	Usabilidad
RNF-03	La aplicación deberá tener un tiempo de respuesta rápido para las acciones del usuario.	Rendimiento
RNF-04	La aplicación debe tener una arquitectura modular, permitiendo la integración de nuevas funcionalidades con facilidad	Portabilidad

Tabla 1

2.5.4 Requerimientos no Funcionales Externos

RNF	Descripción del Requerimiento	Clasificación
RNF-01	La aplicación deberá integrarse con el calendario del sistema operativo	Interoperabilidad
RNF-02	La aplicación deberá cumplir con las regulaciones y leyes locales sobre privacidad de datos	Legislativos
RNF-03	La aplicación debe cumplir con los estándares de seguridad.,	Seguridad
RNF-04	La aplicación deberá cumplir con la privacidad absoluta de la información de los clientes	Privacidad
RNF-05	La aplicación deberá regirse por normas y reglas basada en la ética -	Éticos

Tabla 2

2.5.5 Requerimientos no Funcionales Organizacional

RNF	Descripción del Requerimiento	Clasificación
RNF-01	La especificación de requerimientos de software deberá realizar según la norma IEEE 830	Estándares
RNF-02	El diseño de la aplicación debe desarrollarse con herramienta CASE, por ejemplo: StarUML	Estándares
RNF-03	El apartado front-end se deberá desarrollar con las tecnologías: .NET Framework con C#	Implementación
RNF-04	El apartado back-end se deberá desarrollar con las tecnologías: .NET Framework con C#	Implementación

Tabla 3

2.5.6 Otros requisitos

Es importante que la aplicación contenga imágenes descriptivas, instructores tanto de hombres como mujeres, con variedad de edades.

Requerimiento de gestión del dinero dentro del programa (Educado por método de entrevista):

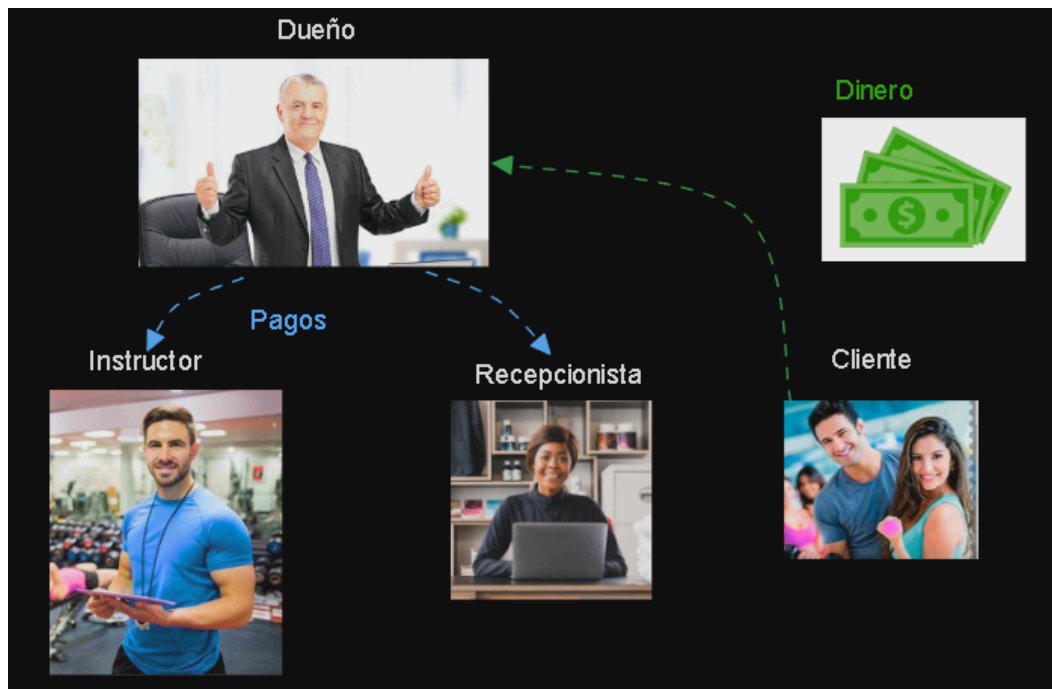


Figura 1

Bosquejo de vista del rol Recepcionista (Educado por método de entrevista):

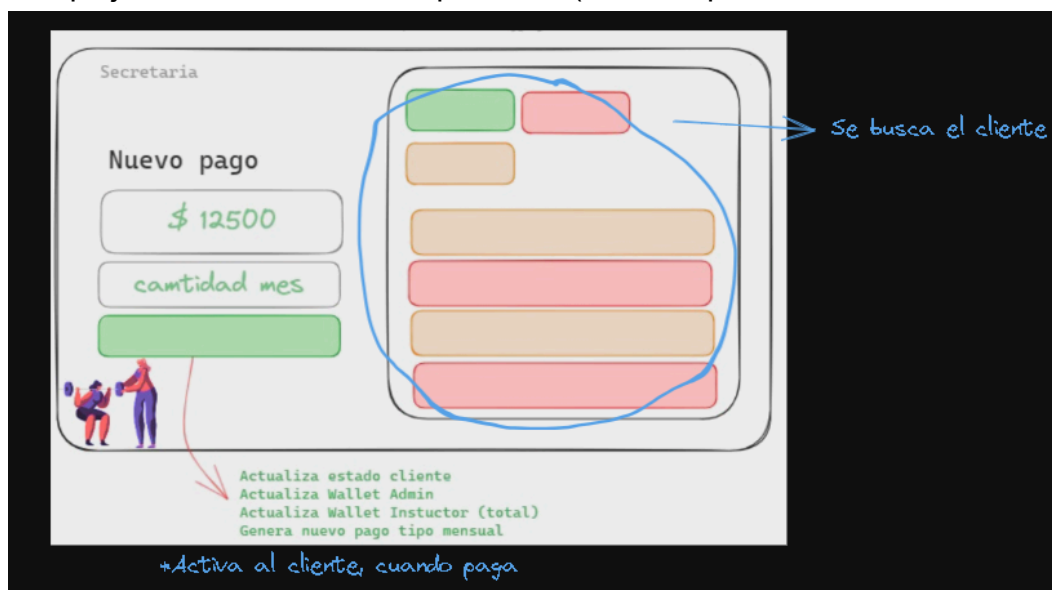


Figura 2

Paleta de Colores seleccionados en RGB:

Letras oscuras: 110; 118; 129

Fondo oscuro: 9; 0; 20

Fondo claro: 16; 8; 23

Fondo más Claro: 69; 34; 99

Fondo TextBox: 69; 34; 99

Botones Verdes: 41; 147; 45

Eliminaciones: 192; 0; 0

Ediciones: 0; 0; 192

Guardar: 192; 0; 0

Activar: 255;140;0

2.6 Análisis de Riesgo

2.6.1 Riesgo de producto

Problemas con la calidad del código y la seguridad:

- ❖ Probabilidad: Moderada
- ❖ Efecto: Crítico
- ❖ Estrategia: Implementar una política de control de calidad y pruebas de seguridad para garantizar que el código sea seguro y libre de errores antes de la implementación.

Requisitos incompletos o poco claros:

- ❖ Probabilidad: Baja
- ❖ Efecto: Moderado
- ❖ Estrategia: Establecer un proceso de revisión de requisitos para asegurarse de que sean claros y completos, y para identificar cualquier cambio necesario en el alcance del proyecto.

2.6.2 Riesgo de proyecto

El producto conlleva más tiempo de lo esperado (estimación):

- ❖ Probabilidad:moderada
- ❖ Efecto:Tolerable
- ❖ Estrategia: estimar los tiempos con nuevos cálculos basados en la experiencia que se obtuvo
- ❖ Subclasificación: estimación

Conflictos entre miembros del equipo de desarrollo:

- ❖ Probabilidad: Baja
- ❖ Efecto: Tolerable

- ❖ Estrategia: Promover la comunicación abierta y efectiva para prevenir y solucionar cualquier conflicto que pueda surgir.
- ❖ Subclasificación: personal

El personal involucrado no tiene experiencia lo que podría afectar los tiempos de entrega:

- ❖ Probabilidad: alta
- ❖ Efecto: moderado
- ❖ Estrategia: consultar a gente experimentada sobre resoluciones que toman un tiempo considerable
- ❖ Subclasificación: personal

2.6.3 Riesgo de Negocio

Solicitud de nuevos requerimientos por parte del cliente.

- ❖ Probabilidad: Baja
- ❖ Efecto: Alta
- ❖ Estrategia: Se tomará como actualización futura fuera del cronograma establecido.
- ❖ Subclasificación: Requerimientos

Cambios en las regulaciones o normativas:

- ❖ Probabilidad: Baja
- ❖ Efecto: Crítico
- ❖ Estrategia: Mantenerse actualizado sobre las regulaciones y normativas relevantes, y establecer un proceso para evaluar y ajustar el proyecto.
- ❖ Subclasificación: organizacional

2.6 Historias de Usuario

Los requisitos principales que se obtuvo en cuenta para desarrollar la aplicación fue:

Numero Requisito	#01	título	Crear y agregar usuarios
Estimación	9 días	Valor	10
RF	11		
Descripción	Cómo Admin Quiero que el sistema tenga un solo admin principal, una recepcionista, permita agregar varios instructores Para gestionar el negocio		

Dependencia	1
Pruebas	El Sistema DEBE contar con los cuatro perfiles anteriormente especificados El Sistema NO DEBE tener más perfiles de usuario

Tabla 4

Numero Requisito	#02	título	Crear planes de entrenamiento
Estimación	2 días	Valor	4
RF	2,3,4		
Descripción	Como instructor Quiero una vista en la cual se pueda crear planes personalizados Para asignarlos a un cliente		
Dependencia	1		
Pruebas	El instructor DEBE poder crear planes para sus clientes El sistema NO DEBE permitir planes con campos vacíos		

Tabla 5

Numero Requisito	#03	título	Agregar Clientes
Estimación	6 días	Valor	8
RF	1,4		
Descripción	Como Instructor Quiero una vista Para agregar clientes		
Dependencia	1,3		
Pruebas	El Instructor DEBE poder introducir un cliente ingresando: nombre,apellido,nickname,contraseña,g enero,dia de nacimiento, avatar, teléfono y cbu		

	<p>El sistema NO DEBE permitir ingresar clientes si se encuentra al menos un campo vacío</p> <p>El sistema NO DEBE permitir que se ingresen letras en los campos: teléfono, día de nacimiento y cbu</p> <p>El sistema NO DEBE permitir que se ingresen números en los campos: nombre, apellido</p>
--	--

Tabla 6

Numero Requisito	#04	título	Visualizar clientes
Estimación	6 días	Valor	7
RF	8,10		
Descripción	Como Instructor Quiero una vista Para visualizar los clientes		
Dependencia	1,3		
Pruebas	El Instructor DEBE poder visualizar a los clientes del sistema El sistema NO DEBE mostrar otros roles en esta tabla		

Tabla 7

Numero Requisito	#05	título	Generar gráficos estadísticos sobre el negocio
Estimación	5 días	Valor	8
RF	5		
Descripción	Cómo Admin Quiero ver gráficos estadísticos sobre la ganancia por mes y la proporción de cantidad de clientes por cada instructor del top con mas cantidad Para tener información más accesible y poder tomar mejores decisiones sobre		

	el negocio
Dependencia	1,3
Pruebas	El sistema DEBE tener un gráfico de líneas de ganancia por mes El Admin DEBE tener un de torta que muestre la proporción de cantidad de clientes por cada instructor del top con mas cantidad

Tabla 8

Planificación

Id	Historia de Usuario	puntos usados
01	Crear y agregar usuarios	8
02	Crear planes de entrenamiento	4
03	Agregar Clientes	8
4	visualizar Clientes	7
5	Generar gráficos estadísticos sobre el negocio	8
6	Crear Perfil de Instructor	16
7	Crear Progreso	16

Tabla 9

Tiempo disponible: 4 meses = 20 días por mes =80 días

Número de integrantes de desarrollo: 4

Número de Sprint: 4

Planificación de Sprint :

Se planifica un sprint de duración de 2 semanas

2 semanas = 10 días efectivos

5 integrantes

10 días efectivos * 4 integrantes = 40 días-hombre

Factor de dedicación = 7 días/80 días-hombre = 0.08

Velocidad estimada del nuevo sprint = 0.8 * 100 días-hombre= 8 puntos de historia

spring 1:

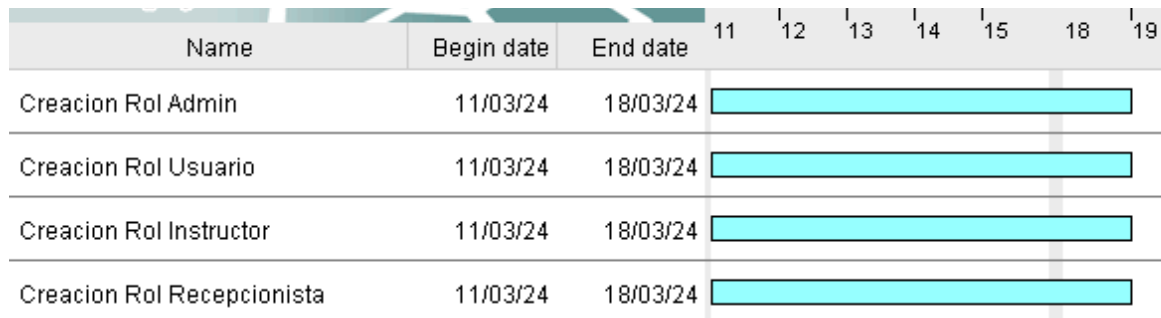


Figura 3

spring 2:

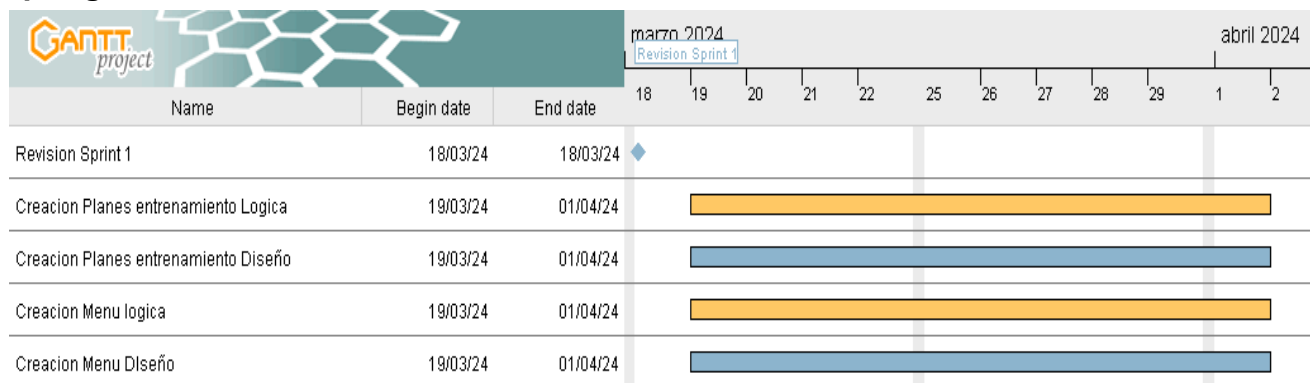


Figura 4

spring 3:

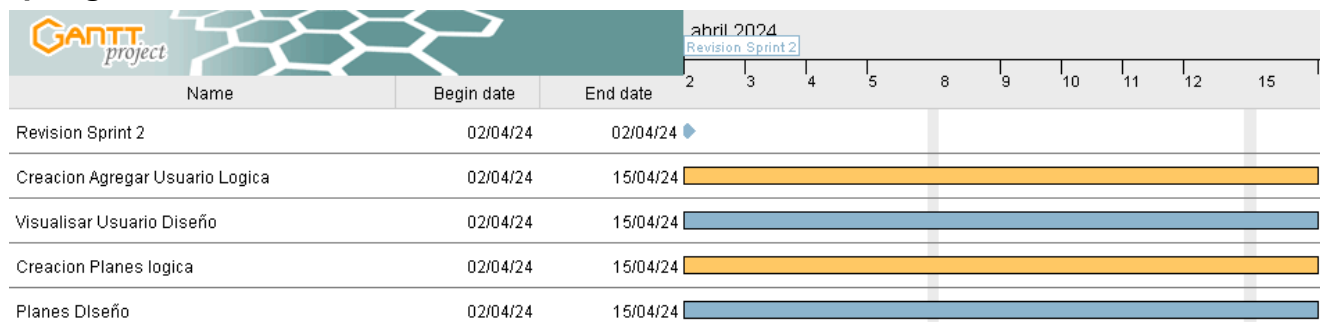


Figura 5

spring 4:

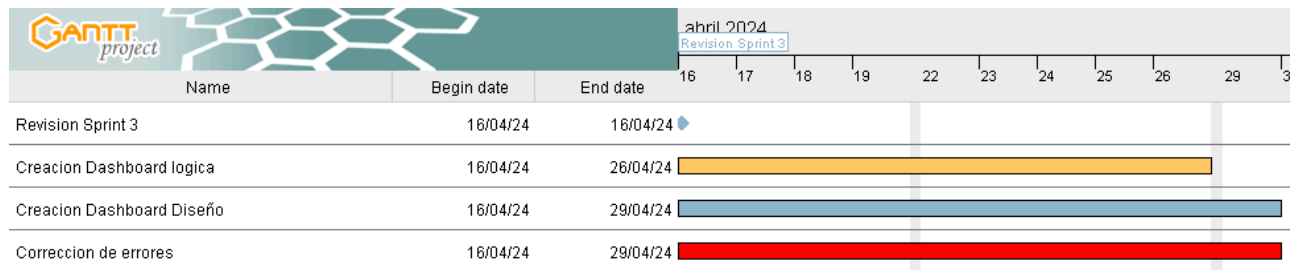


Figura 6

spring 5:

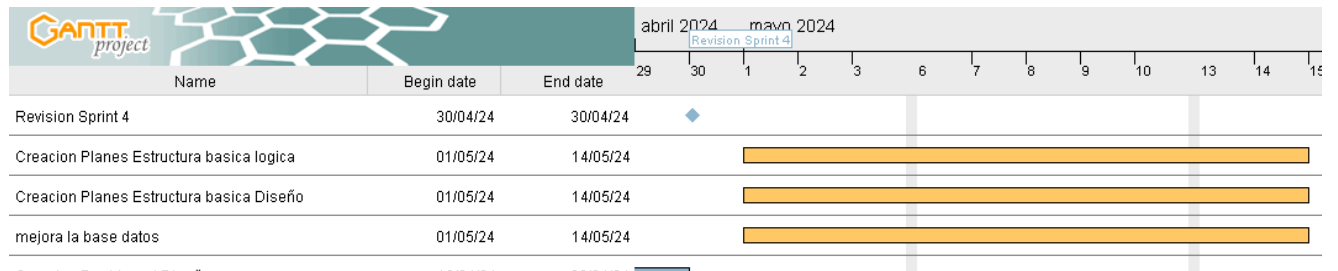


Figura 7

spring 6:

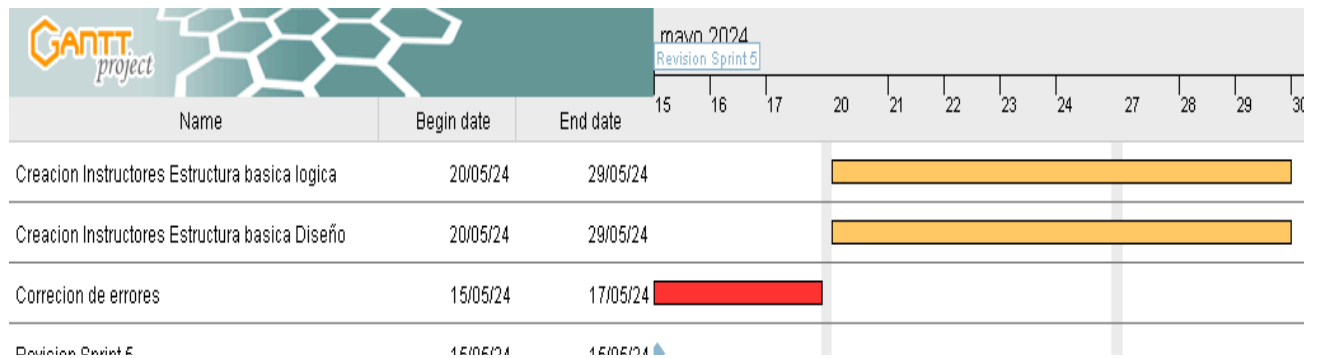


Figura 8

spring 7:

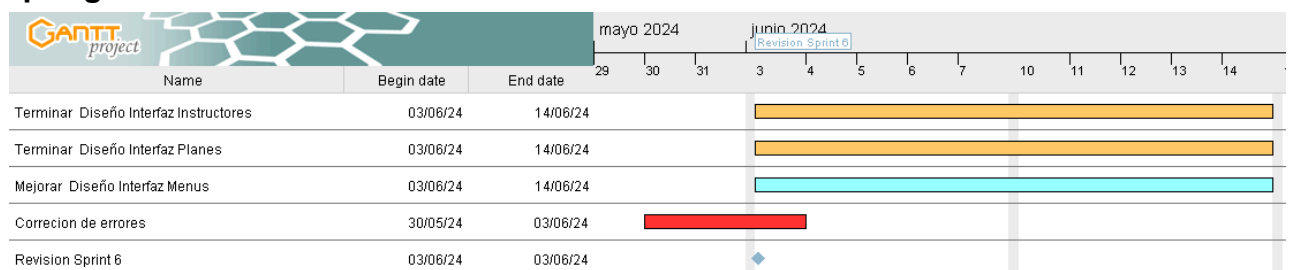


Figura 9

spring 8:

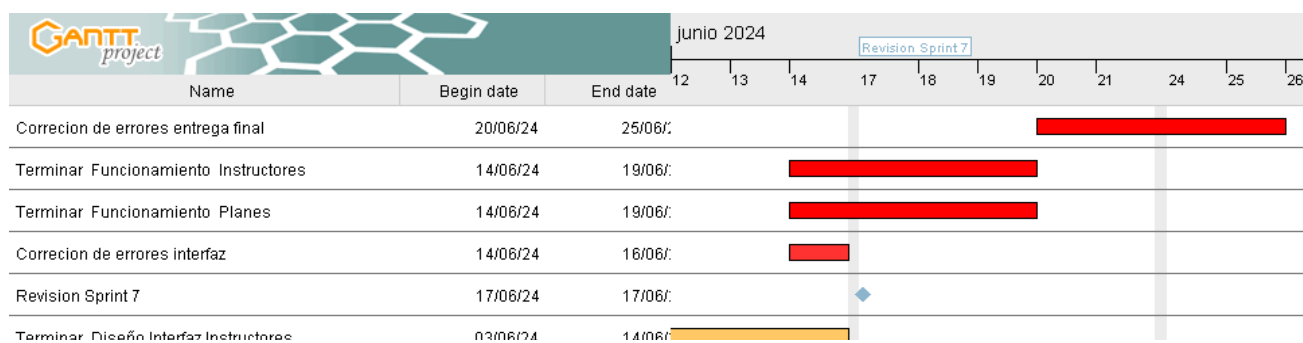


Figura 10

2.7 Casos de uso

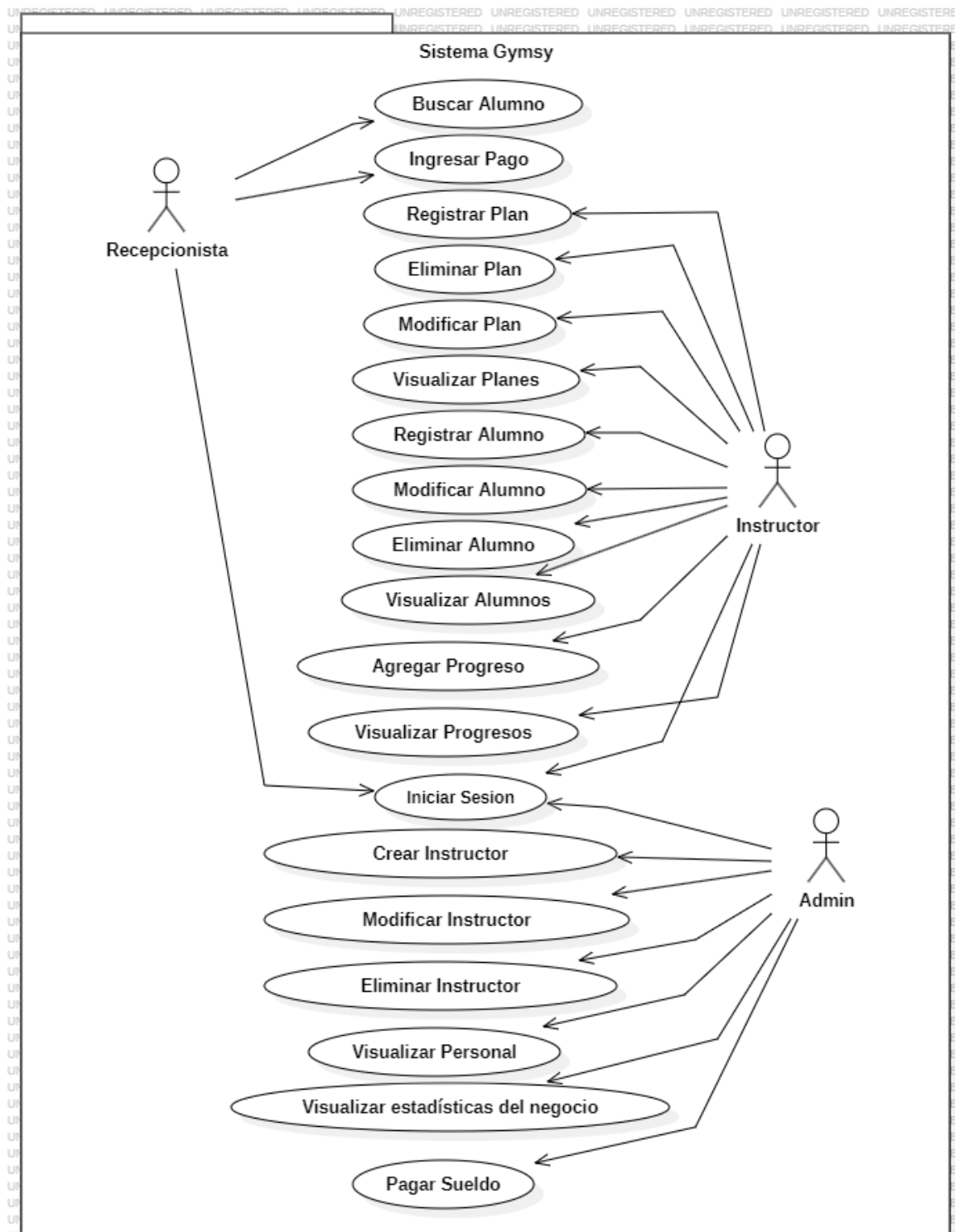


Figura11

2.8 Conversación y diagrama de secuencia

2.8.1 Conversación: Caso de uso “Crear Perfil de Instructor”

Actor: Admin

Acción	Curso Normal	Curso Alternativo
1. A: Ingresa el nombre, apellido, un avatar, numero de telefono, fecha de nacimiento, género y un apodo		
2. S: Se verifica que los campos no esten vacios validarCampos()	2.1 Los campos no estan vacios	2..1.1 Al Menos un campo esta vacio 2.1.2 Se muestra un mensaje “Revise y complete correctamente los campos.” 2.1.3 Volver a 1
3. S:se verifica la fecha de nacimiento verificarNacimiento(fecha de nacimiento)	3.1 La fecha de nacimiento es aceptable	3.1.1 El nacimiento es posterior al día de la inscripción 3.1.2 se muestra el Mensaje “Fecha de nacimiento no válida” 3.1.3 volver a 1
4. S: se verifica que el nombre de usuario no está registrado en el en el sistema esUnicoElApodo(apodo)	4.1 el nombre de usuario no esta registrado previamente	4.1.1 El nombre de usuario está registrado en el sistema. 4.1.2 Se muestra un mensaje “Error al verificar el nombre de usuario” 4.1.3 volver a 1
5. S:Se guarda el instructor guardarInstructor(nombre ,apellido,avatar,fecha de nacimiento,genero,apodo)		
6. S:se muestra el mensaje “Se guardaron correctamente los datos”		

6. Fin del caso de uso		
------------------------	--	--

tabla 10

2.8.1 Diagrama de secuencia “Crear Perfil Instructor”

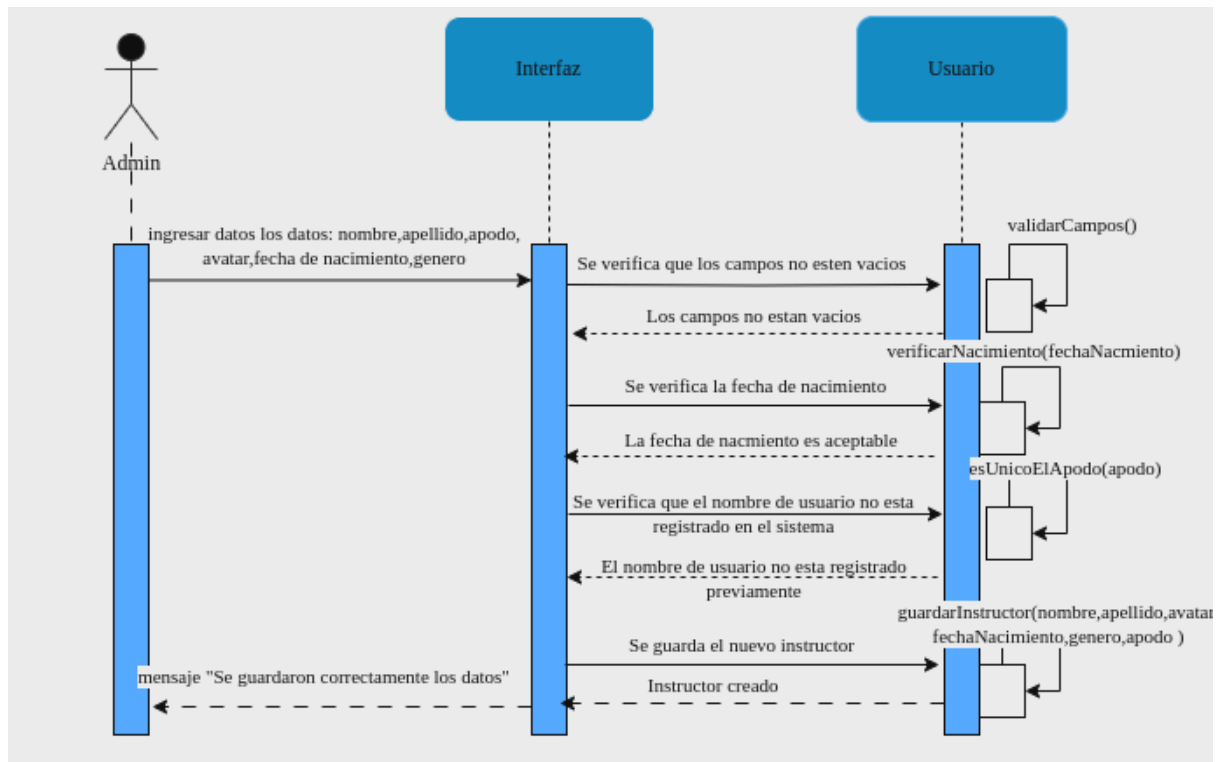


Figura 12

Diagrama de secuencia “Crear Perfil Instructor” alternativa 1

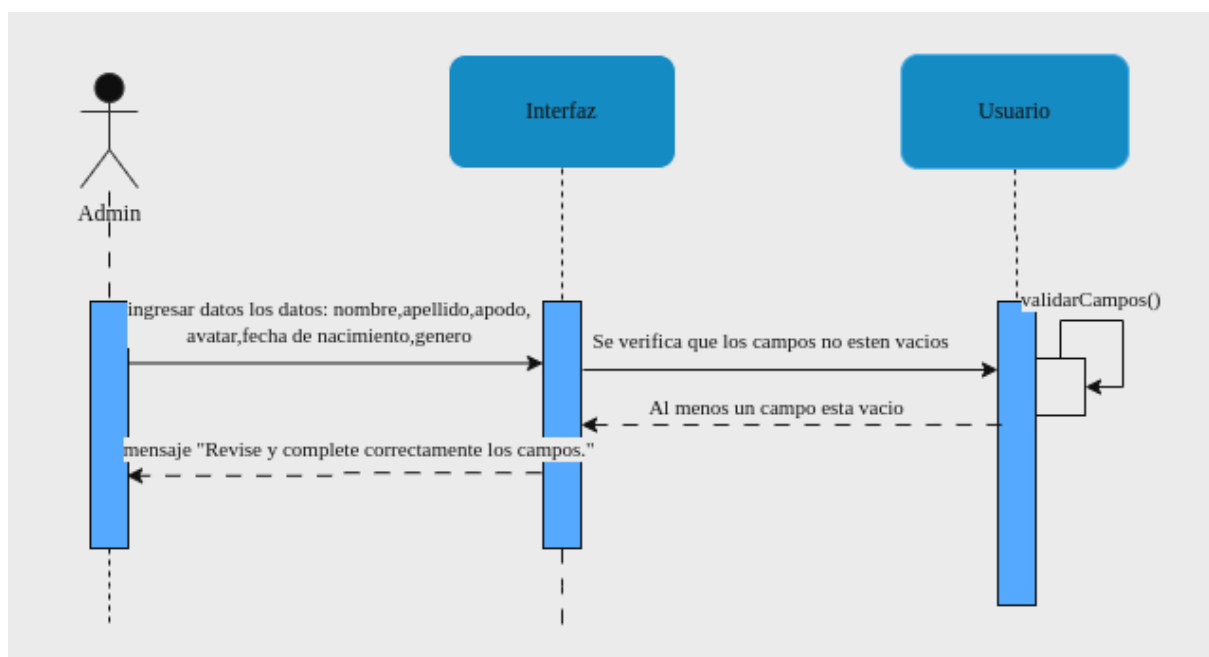


Figura 13

Diagrama de secuencia "Crear Perfil Instructor" alternativa 2

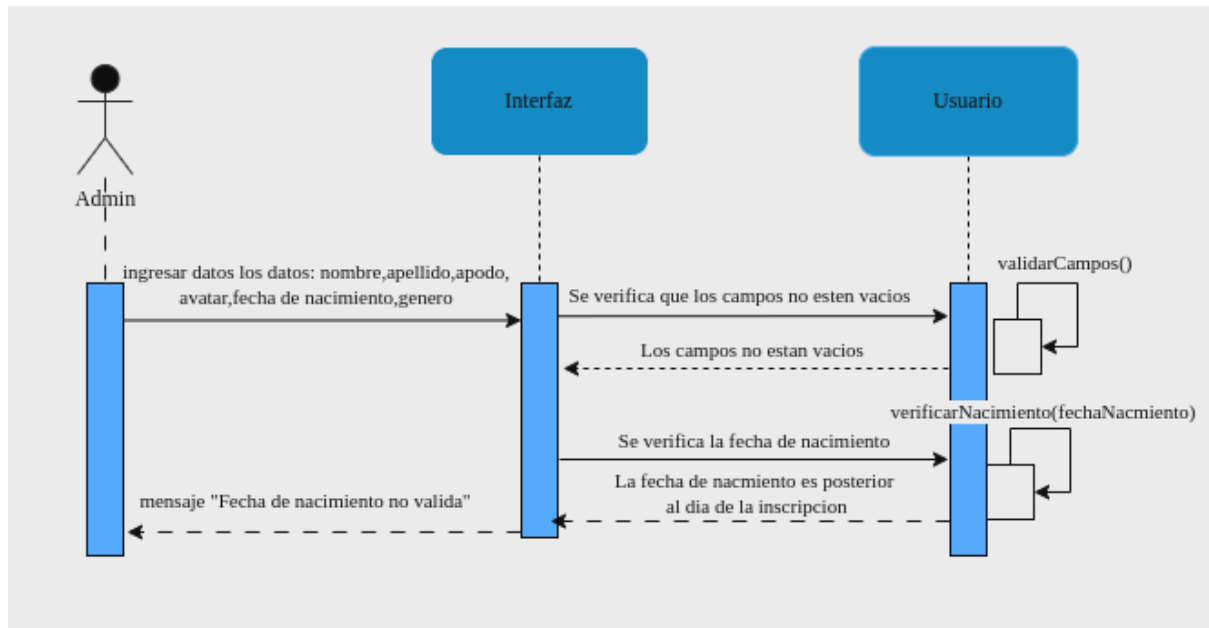


Figura 14

Diagrama de secuencia "Crear Perfil Instructor" alternativa 3

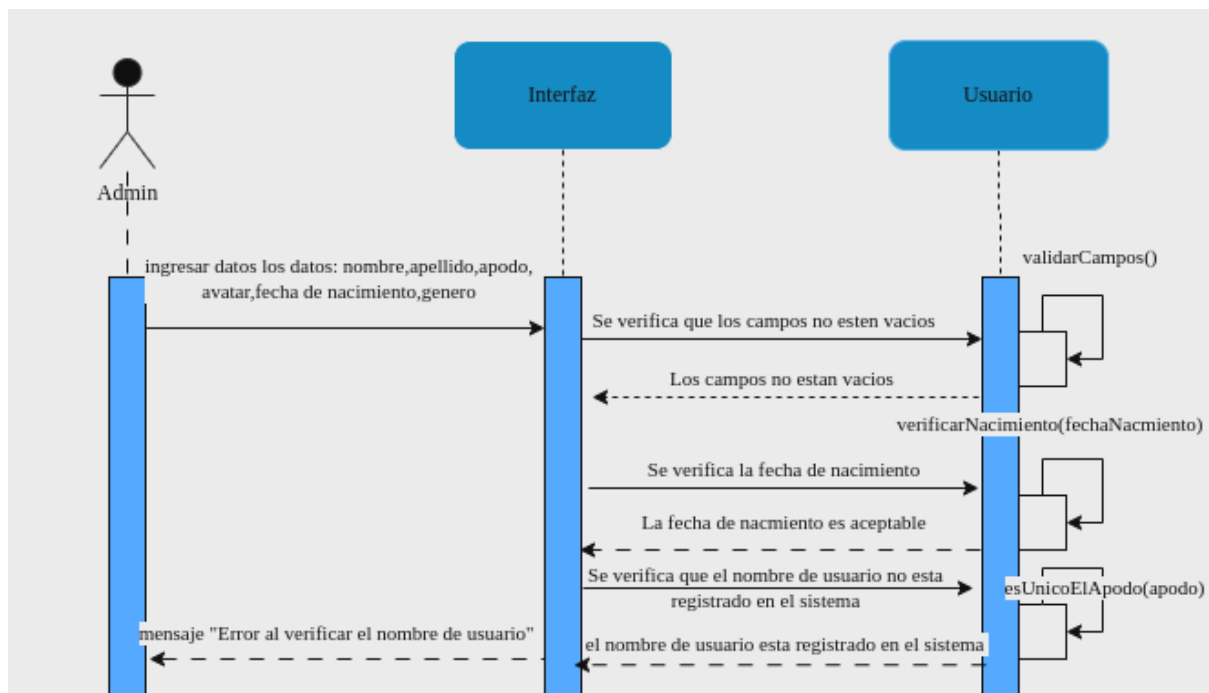


Figura 15

2.8.2 Conversación: Caso de uso “Agregar Progreso”

Actor: Instructor

Acción	Curso Normal	Curso Alternativo
1. A: Ingresa el título, el peso actual, la altura actual, notas sobre el progreso y una imagen		
2. S: se verifica que los campos no estén vacíos ValidarCampos(titulo,peso,altura,notas,imagen)	2.1 No hay campos vacíos	2.1.1 Se encontró al menos un campo vacío 2.1.2 Se muestra un mensaje “Verifique que los campos estén correctos.” 2.1.3 Ir a 1
3.S: se valida que no exista otro progreso con ese título TituloUnico(titulo)	3.1 No hay otros progresos con ese título	3.1.1 Se encontró un progreso con ese título 3.1.2 Se muestra un mensaje “Ya existe un progreso con ese título” 3.1.3 Ir a 1
4. S: Se guarda el progreso guardarProgreso(título,peso,altura,imagen)		
5. S:se muestra el mensaje “Se agregó correctamente”		
6. Fin del caso de uso		

Tabla 11

2.8.2 diagrama de secuencia “Agregar Progreso”

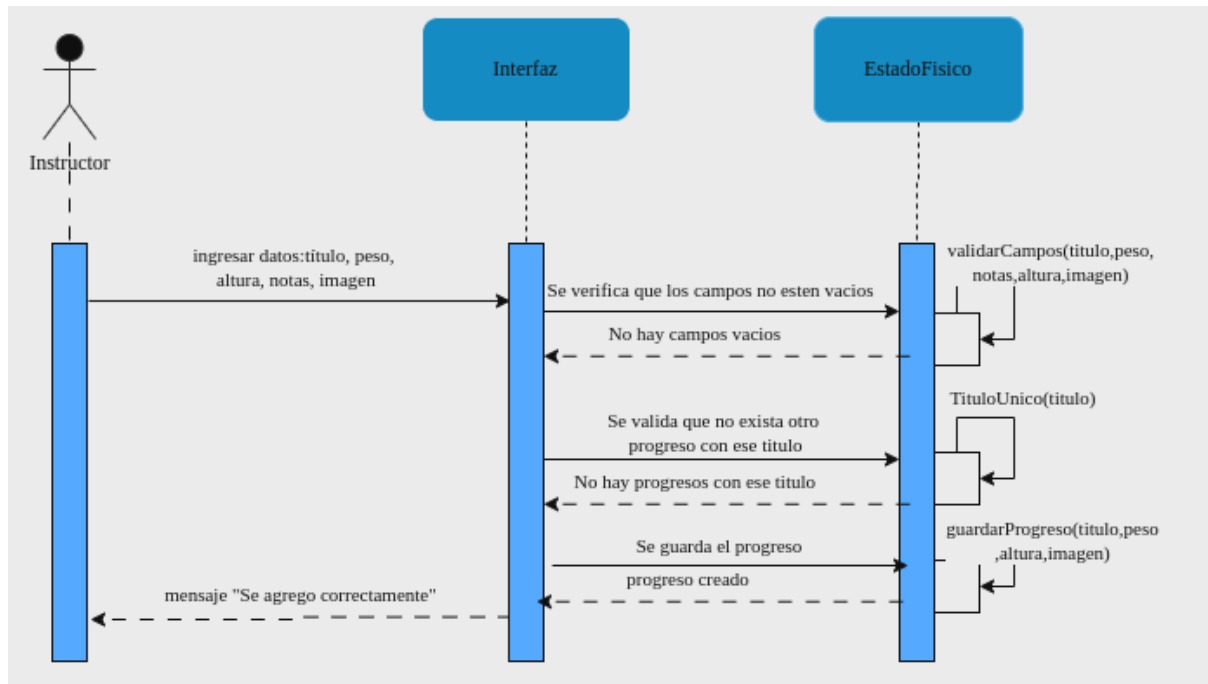


Figura 16

Diagrama de secuencia “Agregar Progreso” alternativa 1

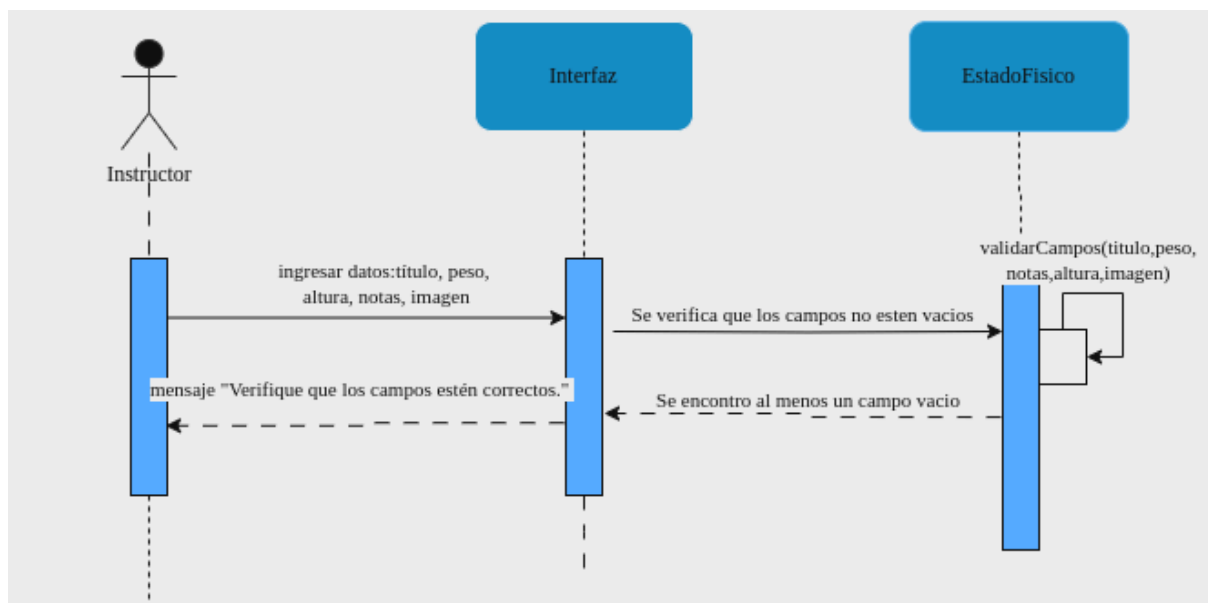


Figura 17

Diagrama de secuencia “Agregar Progreso” alternativa 2

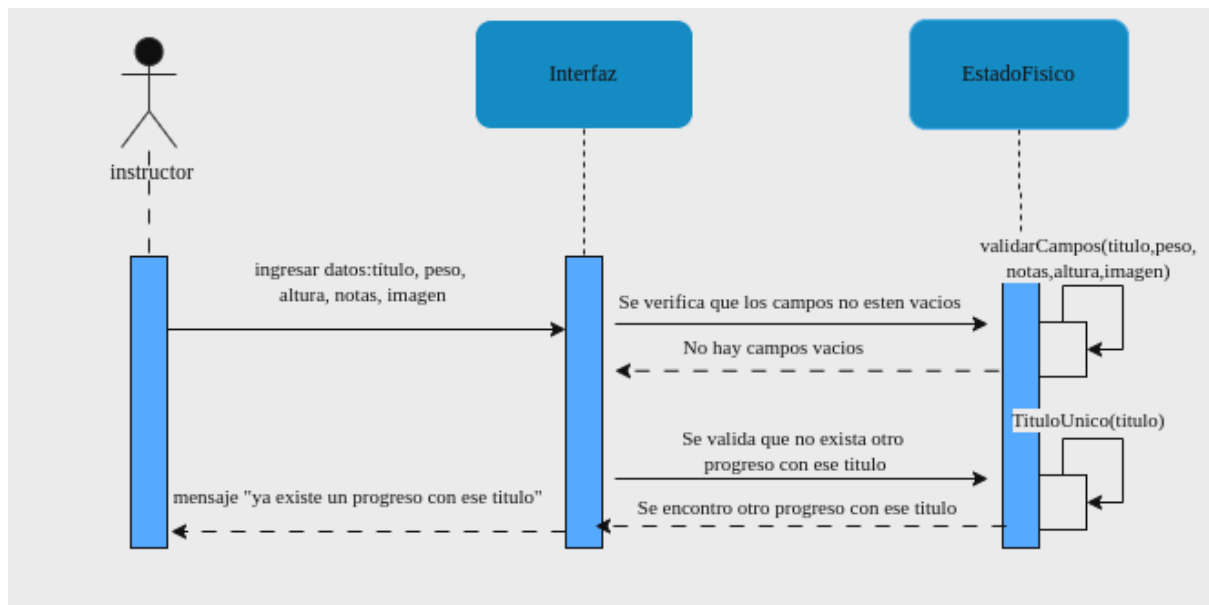


Figura 18

2.8.3 Conversación: Caso de uso “Crear Plan”

Actor: Instructor

Acción	Curso Normal	Curso Alternativo
1. A: Ingresa el precio y la descripción		
2. S: se verifica que los campos no esten vacios validarCampos(precio, descripción)	2.1 no hay campos vacios	2.1.1 Se encontró al menos un campo vacío 2.1.2 Se muestra un mensaje “Por favor, verifique que haya ingresado correctamente todos los campos.” 2.1.3 Ir a 1
3. S: se verifica que la descripción sea unica descripcionUnica(descripción)	3.1 no hay planes con esa descripción	3.1.1 Existe un plan con esa descripción 3.1.2 Se muestra un mensaje “Ya existe un plan con esa descripción” 3.1.3 Ir a 1
4. S: Se guarda el plan agregarPlan(descripción, p		

recio)		
5. S: Se muestra el mensaje “Plan guardado correctamente		
6. Fin del caso de uso		

Tabla 12

2.8.3 diagrama de secuencia “Crear Plan”

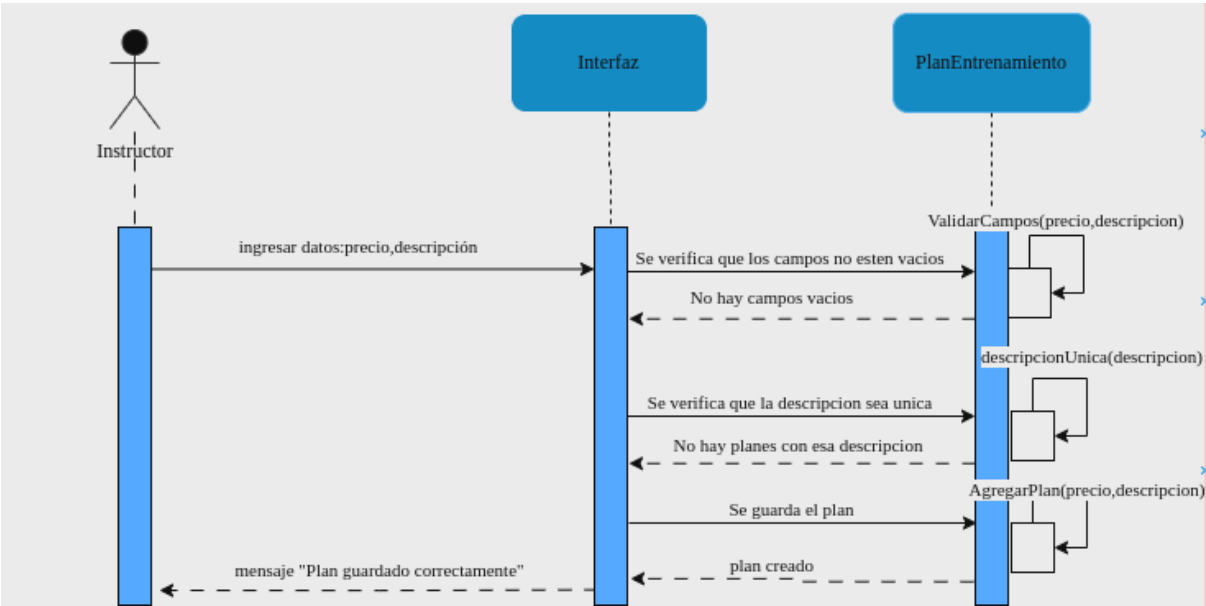


Figura 19

diagrama de secuencia “Crear Plan” alternativa 1

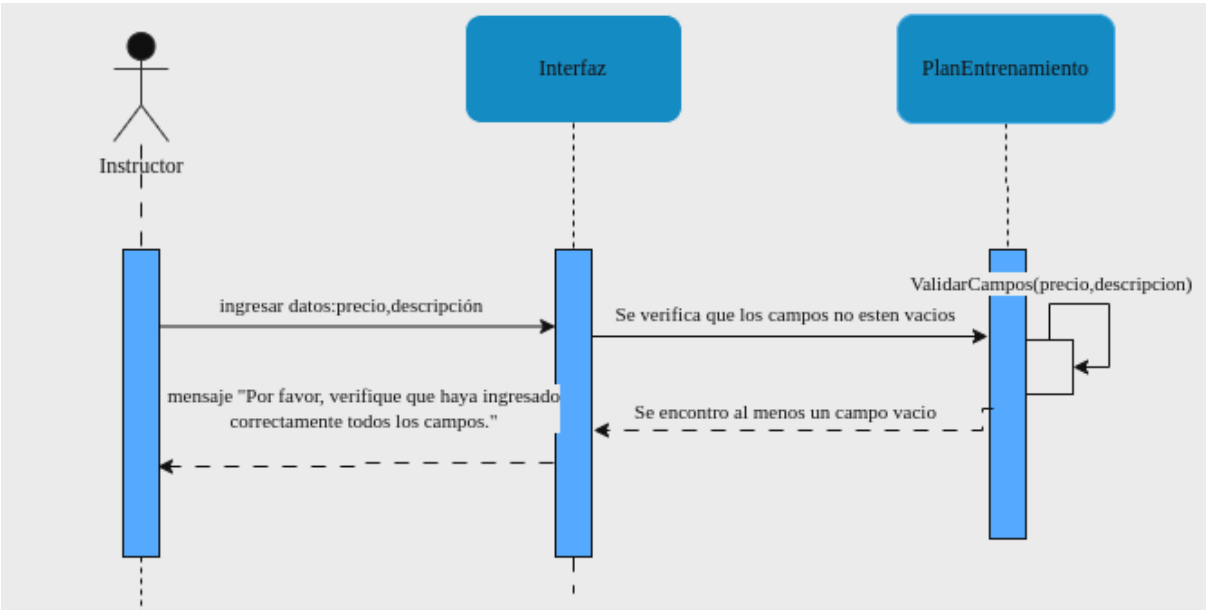


Figura 20

Diagrama de secuencia “Crear Plan” alternativa 2

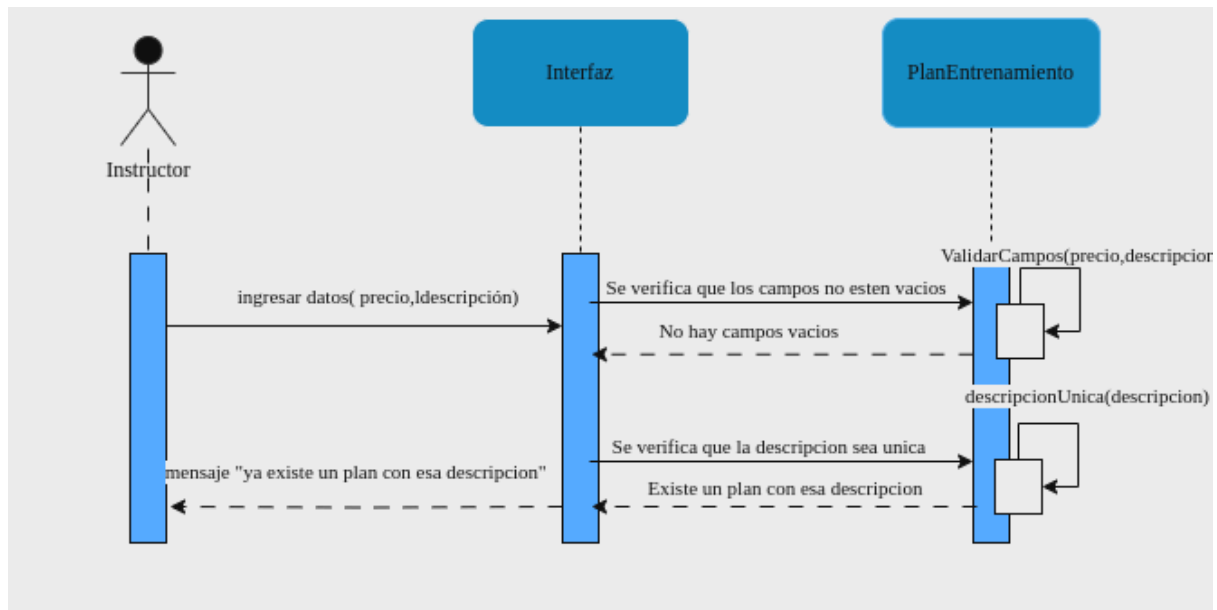


Figura 21

2.9 Contratos de operaciones

Nombre: `guardarInstructor(nombre, apellido, avatar, fecha_nacimiento, género, usuario)`

Referencia cruzada: Caso de uso: Agregar Perfil de instructor

Responsabilidades: Crear un nuevo usuario instructor.

Excepciones:

- ❖ Si algún campo se encuentra vacío debe mostrar un mensaje, y permitir que vuelva a cargar el dato.
- ❖ Si el apodo del instructor ya existe se debe informar con un mensaje

Precondición: el usuario debe estar logueado con el rol de Administrador.

Postcondición: se informa que se creó correctamente el instructor.

Nombre: `guardarProgreso(título, peso, altura, imagen)`

Referencia cruzada: Agregar Progreso

Responsabilidades: Crear un nuevo progreso para un cliente específico en base a los datos ingresados.

Excepciones:

- ❖ Si algún campo se encuentra vacío debe mostrar un mensaje, y permitir que vuelva a cargar el dato.
- ❖ Notificar a través de un mensaje si ya existe un progreso con el mismo título.

Precondición: debe existir una suscripción a la cual se le pueda agregar el progreso.

Postcondición: Se crea correctamente un progreso nuevo.

Nombre:agregarPlan(precio,descripción)

Referencia cruzada: Crear Plan

Responsabilidades: Crear un nuevo plan de entrenamiento en base a los datos ingresados.

Excepciones:

- ❖ Si algún campo se encuentra vacío debe mostrar un mensaje, y permitir que vuelva a cargar el dato.
- ❖ Se debe informar desde si ya existe un plan con la misma descripción.

Precondición:debe existir un usuario tipo entrenador para que dirija el plan de entrenamiento.

Postcondición: Se creará correctamente un nuevo plan de entrenamiento.

2.10 Diagrama DER

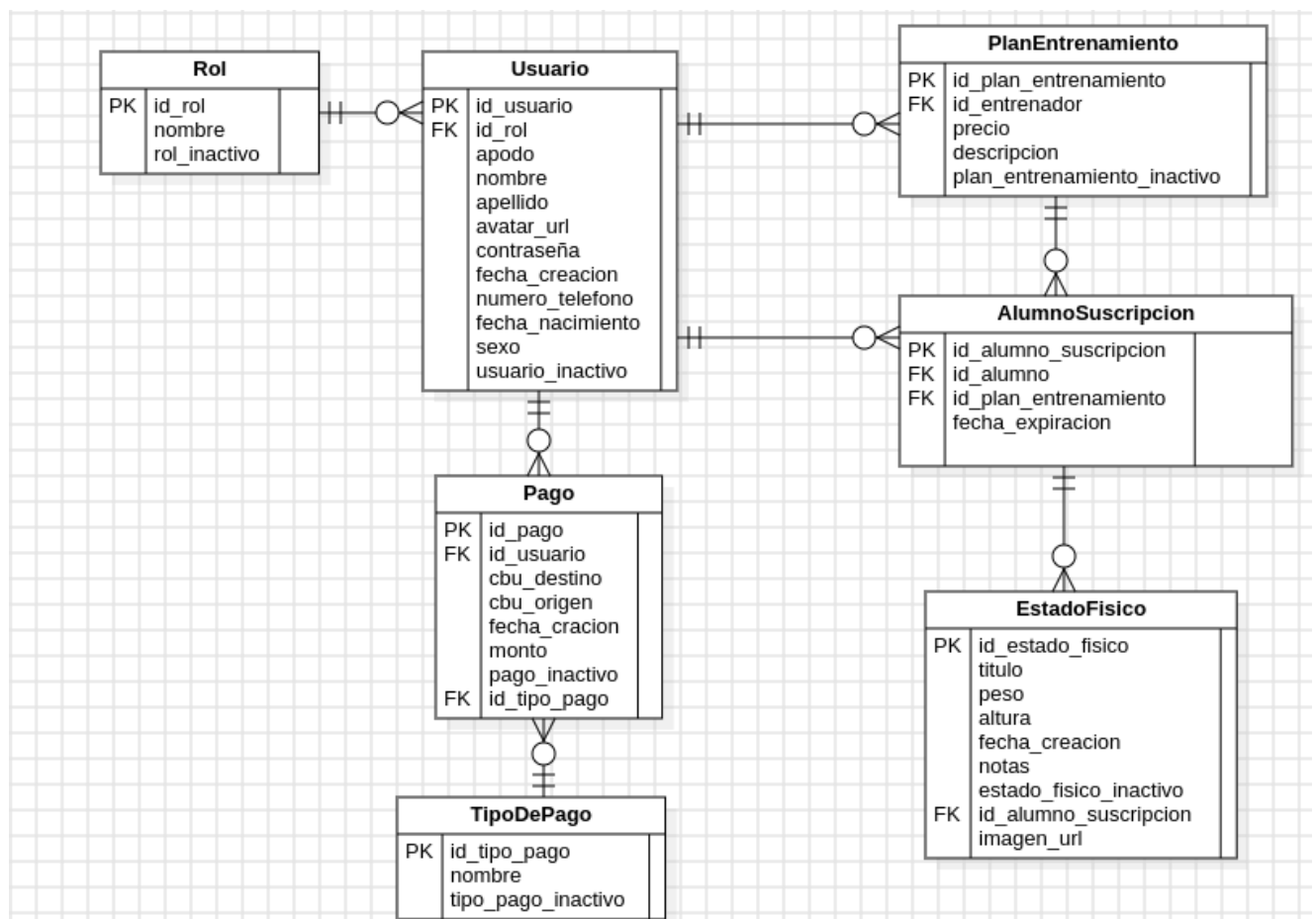


Figura 22

2.10.1 Diccionario de datos

Características de la tabla			
Nombre	Usuario		
Módulo	Usuario		
Descripción	Es la tabla que almacena los datos pertenecientes a los usuarios registrados en el sistema		
Características de los datos			
Campo	tipo	longitud	Significado
id_usuario	int	11	identificación única para cada usuario
id_rol	int	11	identificación del rol del usuario
apodo	varchar	20	apodo o nickname para acceder al sistema
nombre	varchar	20	nombre del usuario
apellido	varchar	20	apellido del usuario
avatar_url	varchar	100	url de la imagen del avatar del usuario
contraseña	varchar	100	contraseña para acceder al sistema del usuario
fecha_creacion	date	-	fecha de creación del usuario
numero_telefono	varchar	100	número de teléfono del usuario
fecha_nacimiento	date	-	fecha de nacimiento del usuario
sexo	char	1	sexo del usuario
usuario_inactivo	bit	-	si el usuario está activo en actualmente
Restricciones			
Campo		Tipo de Restricción	
id_usuario		PRIMARY KEY	
Claves Foráneas			
Campo		Entidad Asociada	
id_rol		Entidad Rol	

Tabla 13

Características de la tabla			
Nombre		Rol	
Módulo		Rol	
Descripción		Contiene los roles que puede tomar un usuario en el sistema	
Características de los datos			
Campo	tipo	longitud	Significado
id_rol	int	11	Identificación única para el rol
nombre	varchar	20	Nombre del rol
rol_inactivo	bit	-	Si el rol está activo actualmente
Restricciones			
Campo		Tipo de Restricción	
id_rol		PRIMARY KEY	

Tabla 14

Características de la tabla			
Nombre	Pago		
Módulo	Pago		
Descripción	Almacena toda la información de pagos del sistema que incluye los pagos de suscripciones y pagos de sueldos.		
Características de los datos			
Campo	tipo	longitud	Significado
id_pago	int	11	identificador único para el pago
id_usuario	int	11	identificador del usuario que realizó el pago
cbu_destino	varchar	22	cbu del destino de pago
cvu_origen	varchar	22	cbu del origen del pago

fecha_creacion	varchar	-	fecha creación del pago
monto	decimal	10,2	monto del pago
pago_inactivo	bit	-	si es pago está activo actualmente
id_tipo_pago	int	11	identificador del tipo de pago utilizado
Restricciones			
Campo		Tipo de Restricción	
id_pago		PRIMARY KEY	
Claves Foráneas			
Campo		Entidad Asociada	
id_usuario		Entidad Rol	
id_tipo_pago		Entidad TipoDePago	

Tabla 15

Características de la tabla			
Nombre	TipodePago		
Módulo	Tipodepago		
Descripción	Los diferentes tipos de pagos disponibles para realizar los pago		
Características de los datos			
Campo	tipo	longitud	Significado
id_tipo_pago	int	11	identificador del tipo de pago
nombre	varchar	50	nombre del tipo del pago
tipo_pago_inactivo	bit	-	si está activo actualmente el tipo de pago
Restricciones			
Campo		Tipo de Restricción	
id_tipo_pago		PRIMARY KEY	

Tabla 16

Características de la tabla	
Nombre	PlanEntrenamiento

Módulo	PlanEntrenamiento		
Descripción	Almacena los datos del plan de entrenamiento a los cuales se puede suscribir		
Características de los datos			
Campo	tipo	longitud	Significado
id_plan_entrenamiento	int	11	identificador del plan de entrenamiento
id_entrenador	int	11	identificador del entrenador que dirige el plan
precio	decimal	10,2	precio del plan
descripcion	varchar	max	información del plan de entrenamiento
plan_entrenamiento_in activo	bit	-	si está activo actualmente el plan de entrenamiento
Restricciones			
Campo		Tipo de Restricción	
id_plan_entrenamiento		PRIMARY KEY	
Claves Foráneas			
Campo		Entidad Asociada	
id_entrenador		Entidad usuario	

Tabla 17

Características de la tabla			
Nombre	AlumnoSuscripcion		
Módulo	AlumnoSuscripcion		
Descripción	Almacena la información de los diferentes alumno inscritos a un plan de entrenamiento		
Características de los datos			
Campo	tipo	longitud	Significado
id_alumno_suscripcion	int	11	identificador única de la suscripción
id_alumno	int	11	identificador única del alumno
id_plan_entrenamiento	int	11	identificador único del plan de entrenamiento relacionado

fecha_expiracion	datetime	-	fecha de expiración de la suscripción
Restricciones			
Campo			Tipo de Restricción
id_alumno_suscripcion			PRIMARY KEY
Claves Foráneas			
Campo			Entidad Asociada
id_alumno			Entidad Usuario
id_plan_entrenamiento			Entidad PlanEntrenamiento

Tabla 18

Características de la tabla			
Nombre		EstadoFisico	
Módulo		EstadoFisico	
Descripción		Se encargar de ir almacenando el progreso obtenido por un alumno en un determinado plan de entrenamiento	
Características de los datos			
Campo	tipo	longitud	Significado
id_estado_fisico	int	11	identificador único del estado físico
título	varchar	100	título del avance
peso	decimal	5,2	peso del alumno al momento del registro del progreso
altura	decimal	5,2	altura del alumno
fecha_creacion	datetime	-	fecha de creación del registro
notas	varchar	max	notas adicionales sobre el avance
estado_fisico_inactivo	bit	-	si está inactivo actualmente
id_alumno_suscripcion	int	11	suscripción a la cual pertenece
imagen_url	varchar	100	url de la imagen del alumno al momento del registro
Restricciones			
Campo			Tipo de Restricción

id_estado_fisico	PRIMARY KEY
Claves Foráneas	
Campo	Entidad Asociada
id_alumno_suscripcion	Entidad AlumnoSuscripcion

Tabla 19

2.10.2 Mapa físico de la Base de datos

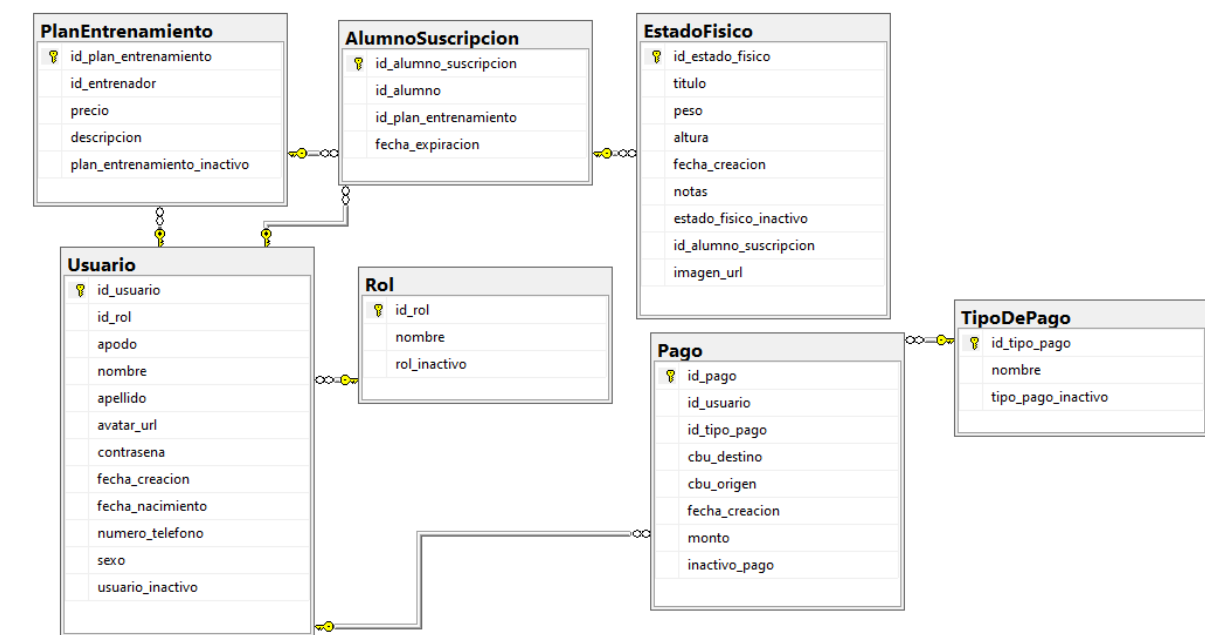


Figura 23

2.11 Diagrama de clases

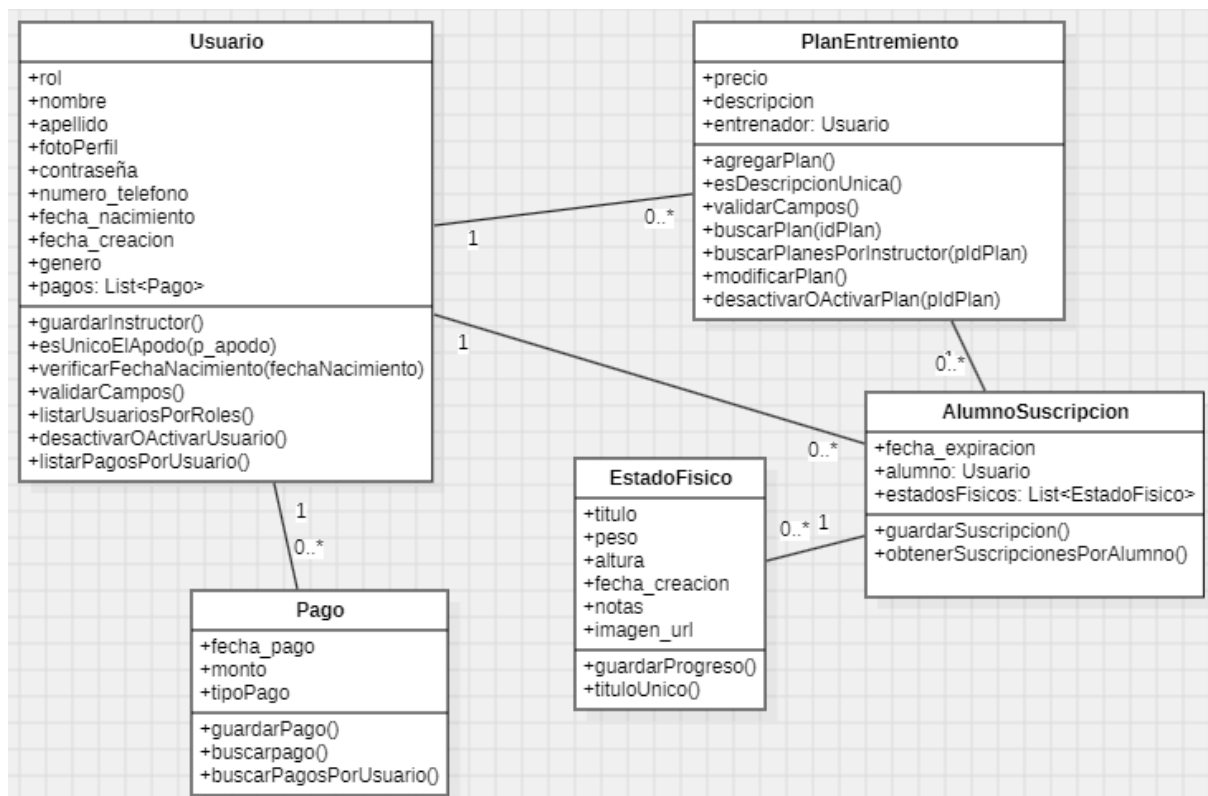


Figura 24

3. Herramientas y/o Lenguajes de Programación

Las tecnologías utilizadas durante el desarrollo del proyecto fueron: C#, .NET , .Net Framework, Sql Server como administrador de base de datos, Asana, Fligma

3.1 Descripción

La aplicación será desarrollada con la tecnología .NET en su versión .Net Framework y sus librerías asociadas, empleando el lenguaje C#, opcionalmente se podrá establecer conexiones con servicios de terceros

Asana: es una herramienta integral de gestión del trabajo diseñada para ayudar a personas y equipos a realizar un seguimiento de las tareas, delegar responsabilidades, monitorear el progreso y comunicarse en tiempo real.

Sql Server: Gestor de bases de datos utilizado en el proyecto.

Discord: Discord es una aplicación de chat gratuita que permite a los usuarios comunicarse en tiempo real mediante texto, voz o vídeo.

StarUML: una herramienta de modelado de software que permite a los desarrolladores crear diagramas UML (Unified Modeling Language) de manera intuitiva y eficiente. Los usuarios pueden diseñar diagramas de clases, diagramas de secuencia, diagramas de actividades, diagramas de casos de uso y otros tipos de diagramas utilizados en el desarrollo de software.

Draw.io: es una herramienta de diagramación en línea que permite a los usuarios crear una amplia variedad de diagramas, incluidos diagramas de flujo, diagramas de red, diagramas de clases, diagramas de secuencia y muchos más. Es una herramienta flexible y fácil de usar que ofrece una amplia gama de formas y símbolos que los usuarios pueden arrastrar y soltar para crear sus diagramas.

Chat GPT: Chat GPT es una tecnología desarrollada por OpenAI que utiliza inteligencia artificial para generar respuestas de texto en conversaciones humanas. Chat GPT se puede utilizar en una variedad de aplicaciones, como asistentes virtuales, chatbots, corrección de texto y generación de texto creativo. Es una herramienta poderosa para la reformulación de texto y la solución de problemas de lenguaje natural.

Figma: es una herramienta de diseño gráfico basada en la nube que permite a los usuarios colaborar en tiempo real. Se utiliza principalmente para el diseño de interfaces de usuario (UI) y experiencia de usuario (UX) para sitios web y aplicaciones móviles. Ofrece funcionalidades como prototipado, creación de wireframes, y sistemas de diseño, facilitando el trabajo en equipo y la coherencia en los proyectos de diseño.

4. Resultados

Plan de pruebas: Agregar Instructor

Id-nombre del Sistema: Sistema Gymsy

Versión del caso de Prueba:1

Autor del caso de Prueba: Maciel Lautaro

Fecha de Creación:15/6/24

Caso de uso:Agregar instructor

Nombre del probador: Jeremias Ezequiel Gonzalez

Fecha de ejecución: 24/06/24

CP	Objetivo	Datos de entrada	Resultado esperado	Resultado obtenido
1	Agregar un instructor- con datos válidos	<div>ingresar como admin y dar click en el botón "Instructor"</div> <div>click en el botón " agregar"</div> <div>nombre válido "agusto" "agusto "</div> <div>apellido válido "Gonzales" "Gonzales"</div> <div>numero telefono valido"3794""3794"</div> <div>fecha de nacimiento válido "02/02/2000"</div> <div>usuario valido "user1" "user1"</div> <div>contraseña válido "123" "123"</div> <div>imagen válida</div> <div>sexo marcado como masculino</div> <div>click en el botón "Guardar Instructor"</div>	<div>-El sistema registra el instructor</div> <div>-El sistema muestra el mensaje "se guardo correctamente los datos"</div>	-ok
2	Agregar un instructor - con nombre invalido, apellido valido, numero telefono valido, fecha nacimiento valido, usuario valido, contraseña valido, imagen valido, sexo valido	<div>ingresar como admin y dar click en el botón "Instructor"</div> <div>click en el botón "agregar"</div> <div>no ingresar nombre</div> <div>apellido válido "Gonzales"</div> <div>numero telefono valido"3794"</div> <div>fecha de nacimiento válido "02/02/2000"</div> <div>usuario valido "user1"</div>	-El sistema muestra el mensaje "Campo requerido" debajo del campo nombre	-ok

		contraseña válido "123"		
		imagen válida		
		sexo marcado como masculino		
		click en el botón "Guardar Instructor"		
3	Agregar un instructor - con nombre valido, apellido invalido, numero telefono valido, fecha nacimiento valido, usuario valido, contraseña valido, imagen valido, sexo valido	ingresar como admin y dar click en el botón "Instructor" click en el botón "agregar" nombre válido "agusto" no ingresar apellido " " numero telefono valido"3794" fecha de nacimiento válido "02/02/2000" usuario valido "user1" contraseña válido "123" imagen válida sexo marcado como masculino click en el botón "Guardar Instructor"	-El sistema muestra el mensaje "Campo requerido" debajo del campo apellido	-ok
4	Agregar un instructor - con nombre valido, apellido valido, numero telefono invalido, fecha nacimiento valido, usuario valido, contraseña valido, imagen valido, sexo valido	ingresar como admin y dar click en el botón "Instructor" click en el botón "agregar" nombre válido "agusto" apellido válido "Gonzales" no ingresar numero telefono fecha de nacimiento válido "02/02/2000" usuario valido "user1" contraseña válido "123" imagen válida sexo marcado como masculino click en el botón "Guardar Instructor"	-El sistema muestra el mensaje "Campo requerido" debajo del campo numero telefono	-ok
5	Agregar un instructor - con nombre valido, apellido valido, numero telefono valido, fecha nacimiento invalida,	ingresar como admin y dar click en el botón "Instructor" click en el botón "agregar" nombre válido "agusto"	-El sistema no debe dejar poner una fecha luego de la actual	-El sistema deja guardar una fecha luego de la actual

	usuario valido, contraseña valido, imagen valido, sexo valido	apellido válido "Gonzales" numero telefono valido"3794" fecha de nacimiento inválida"11/08/2025" usuario valido "user1" contraseña válido "123" imagen válida sexo marcado como masculino click en el botón "Guardar Instructor"		
6	Agregar un instructor - con nombre valido, apellido valido, numero telefono valido, fecha nacimiento invalido, usuario valido, contraseña valido, imagen valido, sexo valido	ingresar como admin y dar click en el botón "Instructor" click en el botón "agregar" nombre válido "agusto" apellido válido "Gonzales" numero telefono valido"3794" fecha de nacimiento invalido"11/08/2025" usuario valido "user1" contraseña válido "123" imagen válida sexo marcado como masculino click en el botón "Guardar Instructor"	-El sistema no debe dejar poner una fecha luego de la actual	--ok
7	Agregar un instructor - con nombre valido, apellido valido, numero telefono valido, fecha nacimiento valido, usuario invalido, contraseña valido, imagen valido, sexo valido	ingresar como admin y dar click en el botón "Instructor" click en el botón "agregar" nombre válido "agusto" apellido válido "Gonzales" numero telefono valido"3794" fecha de nacimiento válido "02/02/2000" no ingresar usuario " " contraseña válido "123" imagen válida sexo marcado como masculino	-El sistema muestra el mensaje "Campo requerido" debajo del campo usuario	-ok

		click en el botón "Guardar Instructor"		
8	Agregar un instructor - con nombre valido, apellido valido, numero telefono valido, fecha nacimiento valido, usuario valido, contraseña invalido, imagen valido, sexo valido	<div></div> <div>click en el botón "agregar"</div> <div>nombre válido "agusto"</div> <div>apellido válido "Gonzales"</div> <div>numero telefono valido"3794"</div> <div>fecha de nacimiento válido "02/02/2000"</div> <div>usuario valido "user1"</div> <div>no ingresar contraseña ""</div> <div>imagen válida</div> <div>sexo marcado como masculino</div> <div>click en el botón "Guardar Instructor"</div>	-El sistema muestra el mensaje "Campo requerido" debajo del campo contraseña	-ok
9	Agregar un instructor - con nombre valido, apellido valido, numero telefono valido, fecha nacimiento valido, usuario valido, contraseña valido, imagen invalido, sexo valido	<div>ingresar como admin y dar click en el botón "Instructor"</div> <div>click en el botón "agregar"</div> <div>nombre válido "agusto"</div> <div>apellido válido "Gonzales"</div> <div>numero telefono valido"3794"</div> <div>fecha de nacimiento válido "02/02/2000"</div> <div>usuario valido "user1"</div> <div>contraseña válido "123"</div> <div>no ingresar imagen</div> <div>sexo marcado como masculino</div> <div>click en el botón "Guardar Instructor"</div>	-El sistema muestra el mensaje "Campo requerido" debajo del campo imagen	-ok
10	Agregar un instructor - con nombre invalido, apellido valido, numero telefono valido, fecha nacimiento valido, usuario valido, contraseña valido, imagen valido, sexo valido	<div>ingresar como admin y dar click en el botón "Instructor"</div> <div>click en el botón " agregar"</div> <div>nombre inválido "123+"</div> <div>apellido válido "Gonzales"</div> <div>numero telefono valido"3794"</div> <div>fecha de nacimiento válido "02/02/2000"</div> <div>usuario valido "user1"</div> <div>contraseña válido "123"</div>	-El sistema no debe dejar de colocar signo especiales o números en el campo nombre	Ok

		imagen valido		
		csexo marcado como masculino		
		click en el botón "Guardar Instructor"		
11	Agregar un instructor - con nombre valido, apellido invalido, numero telefono valido, fecha nacimiento valido, usuario valido, contraseña valido, imagen valido, sexo valido	click en el botón "instructores"	-El sistema no debe dejar de colocar signo especiales o números en el campo Apellido	OK
		click en el botón " agregar"		
		nombre válido "agusto"		
		apellido Apellido"123+"		
		numero telefono valido"3794"		
		fecha de nacimiento válido "02/02/2000"		
		usuario valido "user1"		
		contraseña válido "123"		
		imagen valida		
		sexo marcado como masculino		
		click en el botón "Guardar Instructor"		
12	Agregar un instructor - con nombre valido, apellido valido, numero telefono invalido, fecha nacimiento valido, usuario valido, contraseña valido, imagen valido, sexo valido	ingresar como admin y dar click en el botón "Instructor"	-El sistema no debe dejar de colocar signo especiales o números en el campo numero telefono	Ok
		Click en el botón "Agregar"		
		nombre válido "agusto"		
		Apellido valido "Gonzales"		
		Numero de telefono invalido"asd"		
		fecha de nacimiento válido "02/02/2000"		
		usuario valido "user1"		
		contraseña válido "123"		
		Imagen valida		
		sexo marcado como masculino		
		click en el botón "Guardar Instructor"		

13	Agregar un instructor - con nombre valido, apellido valido,	ingresar como admin y dar click en el botón "Instructor"	-El sistema no debe dejar poner una fecha luego de la actual	-OK
		click en el botón "agregar"		

	numero telefono valido, fecha nacimiento invalido, usuario valido, contraseña valido, imagen valido, sexo valido	nombre válido "agusto"		
		apellido válido "Gonzales"		
		numero telefono valido"3794"		
		fecha de nacimiento invalida"11/08/2025"		
		usuario valido "user1"		
		contraseña válido "123"		
		imagen valida		
		sexo marcado como masculino		
		click en el botón "Guardar Instructor"		

Tabla 20

Plan de pruebas: Agregar Progreso

Id-nombre del Sistema: Sistema Gymsy

Versión del caso de Prueba: 2

Autor del caso de Prueba: Maciel Lautaro

Fecha de Creación:14/6/24

Caso de uso:Agregar Progreso

Nombre del probador: Jeremias Ezequiel Gonzalez

Fecha de ejecución: 24/06/24

CP	Objetivo	Datos de entrada	Resultado esperado	Resultado obtenido
1	Agregar un progreso al cliente - con todos los datos válidos	titulo "Primer día" valido	-El sistema verifica los campos	- ok
		peso actual "86" válido	-El sistema verifica que el título sea único	
		altura actual "175" válido	-El sistema agrega el progreso	
		nota "No hay notas" valido	-El sistema muestra el mensaje "Se agrego Correctamente"	
		imagen válida		
		click en el botón "Guardar Progreso"		
2	Agregar un progreso al cliente - con titulo vacío, peso actual válido , nota valido, imagen valida,	no agregar titulo	-El sistema verifica los campos	- ok
		peso actual válido "80"	-El sistema muestra el mensaje de error "Verifique que los campos esten correctos." y por debajo del botón guardar progreso una	
		altura actual válido "80"		
		nota valido "NN"		

	altura valida	imagen valida	leyenda "campo requerido"	
		click en el botón "Guardar Progreso"		
3	Agregar un progreso al cliente - con titulo valido, peso actual vacío, nota valido, imagen valida, altura valida	titulo valido "Tercer dia"	-El sistema verifica los campos	- ok
		no agrega peso actual	-El sistema muestra el mensaje de error "Verifique que los campos esten correctos." y por debajo del botón guardar progreso una leyenda	
		altura actual válido "80"	"campo requerido"	
		nota valido "nn"		
		imagen valida		
		click en el botón "Guardar Progreso"		
4	Agregar un progreso al cliente - con titulo valido, peso actual válido , nota vacío, imagen valida, altura valida	agregar titulo "Tercer dia" valido	-El sistema verifica los campos	- ok
		peso actual válido "80"	-El sistema muestra el mensaje de error "Verifique que los campos esten correctos." y por debajo del botón guardar progreso una leyenda	
		altura actual válido "80"	"campo requerido"	
		no agregar nota " "		
		imagen valida		
		click en el botón "Guardar Progreso"		
5	Agregar un progreso al cliente - con titulo valido, peso actual válido ,nota valido, imagen valida, altura vacia	agregar titulo valido "Cuarto dia"	-El sistema verifica los campos	- ok
		peso actual válido "80"	-El sistema muestra el mensaje de error "Verifique que los campos esten correctos." y por debajo del botón guardar progreso una leyenda	
		no agregar altura actual	"campo requerido"	
		nota valido "nn"		
		imagen valida		
		click en el botón "Guardar Progreso"		
6	Agregar un progreso al cliente - con titulo valido, peso actual válido ,nota valido, imagen vacia, altura valida	agregar titulo valido "Quinto dia"	-El sistema verifica los campos	- ok
		peso actual válido "80"	-El sistema muestra el mensaje de error "Verifique que los campos esten correctos." y por debajo del botón guardar progreso una leyenda	
		altura actual válido "80"	"campo requerido"	
		nota valido "asd"		
		no agregar imagen valida		

		click en el botón "Guardar Progreso"		
7	Agregar un progreso al cliente - con titulo duplicado	agregar titulo duplicado "Primer dia"	-El sistema verifica los campos	ok
		peso actual válido "80"	-El sistema verifica que el titulo sea único	
		altura actual válido "80"	-El sistema muestra el mensaje de error "Ya existe un progreso con ese titulo"	
		nota valido "asd"		
		imagen valida		
		click en el botón "Guardar Progreso"		

Tabla 21

Plan de pruebas: Agregar Plan

Pruebas

Id-nombre del Sistema: Sistema Gymsy

Versión del caso de Prueba: 4

Autor del caso de Prueba: Maciel Lautaro

Fecha de Creación: 13/6/24

Caso de uso: Agregar Plan

Nombre del probador: Jeremias Ezequiel Gonzalez

Fecha de ejecución: 24/06/24

CP	Objetivo	Datos de entrada	Resultado esperado	Resultado obtenido
1	Agregar un plan- precio valido y descripción valido	ingresar como instructor valido	-El sistema Verifica los campos	-ok
		click en el botón Plan	-El sistema verifica que la descripción sea valida	
		precio valido "100"	-El sistema agrega el plan	
		descripción valido "King Boxing"	-El sistema muestra el mensaje "Plan guardado correctamente."	
		click en el botón "Guardar plan"		
2	Agregar un plan - Precio invalido y descripción valido	ingresar como instructor valido	-El sistema Verifica los campos	-ok
		click en el botón plan	-El sistema muestra el mensaje "Por favor, verifique que haya ingresado correctamente todos los campos" y una leyenda debajo del campo "Error en el campo" debajo del campo precio	
		no ingresar precio		
		descripción valido "kickboxing"		
		click en el botón "Guardar plan"		

3	Agregar un plan - precio valido y Descripción invalido	ingresar como instructor valido	-El sistema Verifica los campos -El sistema muestra el mensaje “Por favor, verifique que haya ingresado correctamente todos los campos” y una leyenda debajo del campo "Error en el campo" debajo del campo descripcion	-ok
		click en el botón plan		
		precio valido "100"		
		descripción invalido “		
		click en el botón "Guardar plan"		
4	Agregar un plan- Precio invalido y descripción valido	ingresar como instructor valido	-El sistema Verifica los campos -El sistema muestra el mensaje “Por favor, verifique que haya ingresado correctamente todos los campos” y una leyenda debajo del campo "Error en el campo" debajo del campo precio	-ok
		click en el botón plan		
		no agregar precio		
		descripción valido "kickboxing"		
		click en el botón "Guardar plan"		
5	Agregar un plan- precio valido "100" y Descripción invalido	ingresar como instructor valido	-El sistema Verifica los campos -El sistema muestra el mensaje “Por favor, verifique que haya ingresado correctamente todos los campos” y una leyenda debajo del campo "Error en el campo" debajo del campo descripcion	-ok
		click en el botón plan		
		precio valido "100"		
		no agregar descripción		
		click en el botón "Guardar plan"		
6	Agregar un plan- Precio y Descripcion vacío	ingresar como instructor valido	-El sistema Verifica los campos -El sistema muestra el mensaje “Por favor, verifique que haya ingresado correctamente todos los campos” y una leyenda debajo del campo "Error en el campo" debajo de los campos precio y descripcion	-ok
		click en el botón plan		
		no agregar precio		
		no agregar descripción		
		click en el botón "Guardar plan"		

Tabla 22

Maquetado de pantallas

Pantalla de Inicio de sesión



Figura 25

Pantalla de control de alumnos del instructor

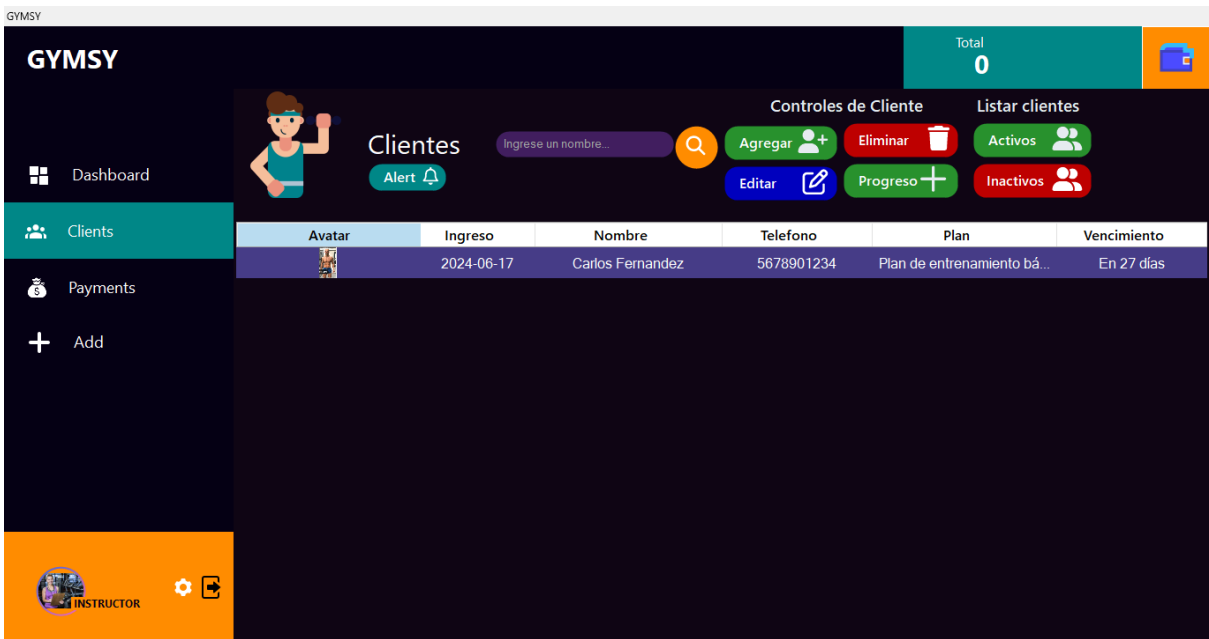


Figura 26

Pantalla de control de planes del instructor

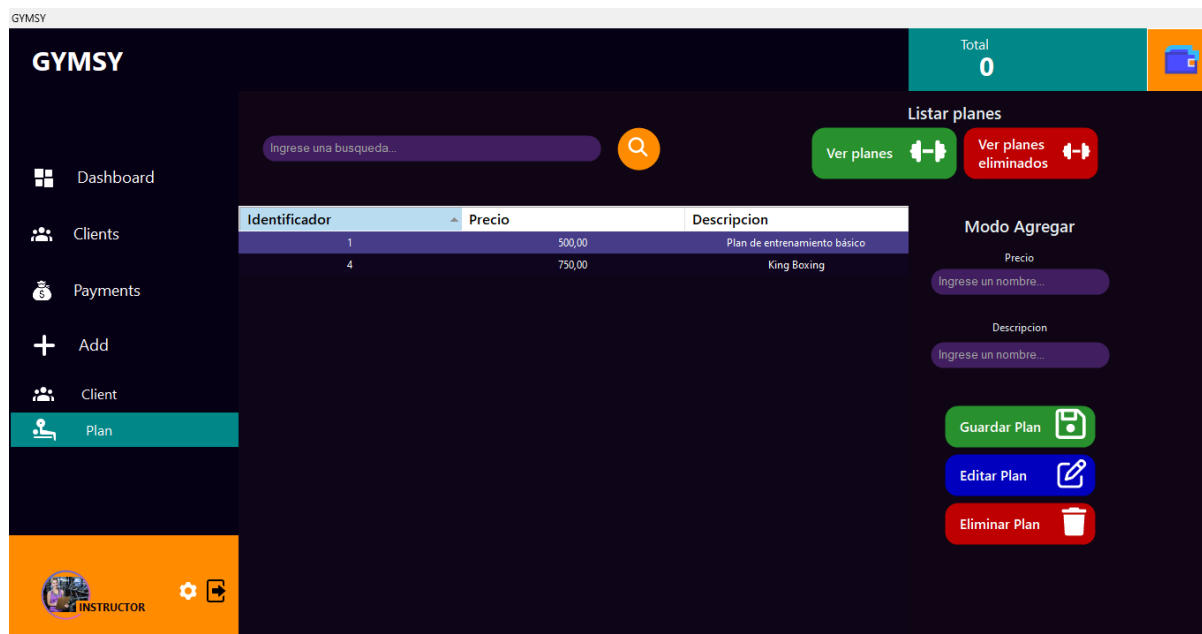


Figura 27

Pantalla de agregar progreso del instructor

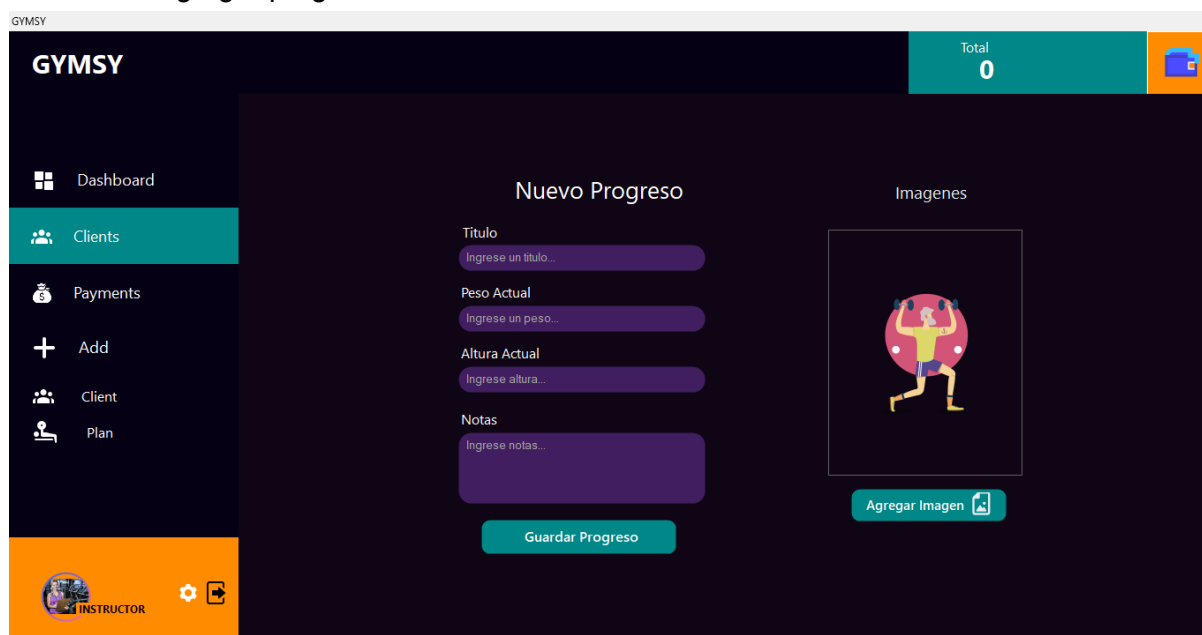


Figura 28

Pantalla de control de instructores del administrador

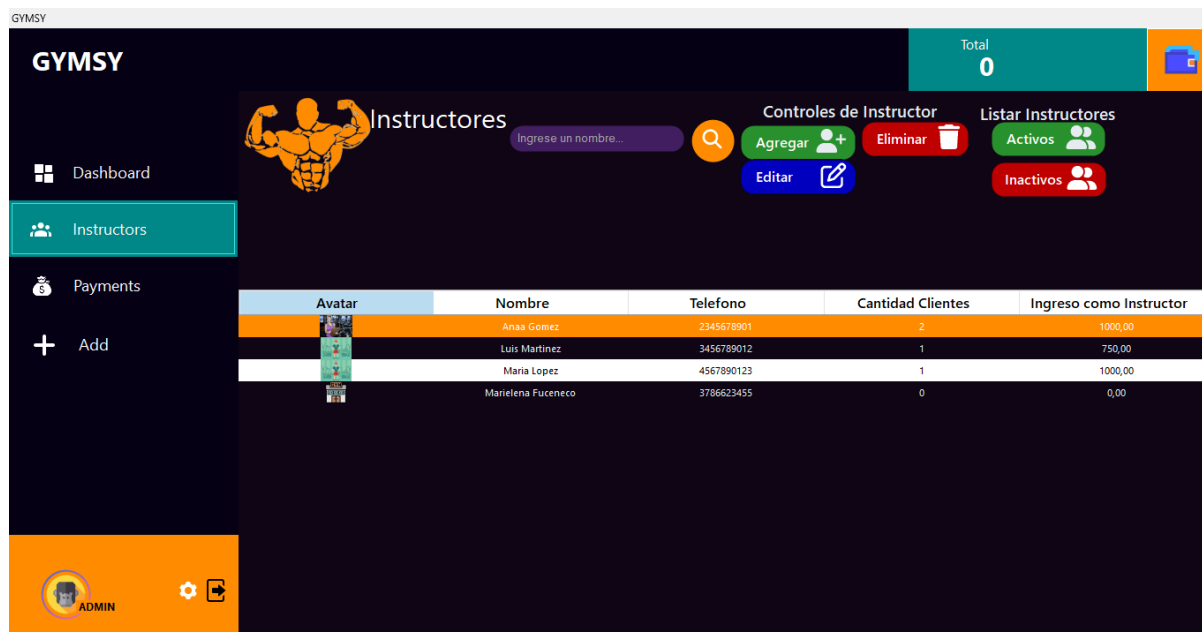


Figura 29

Pantalla de añadir instructor del administrador

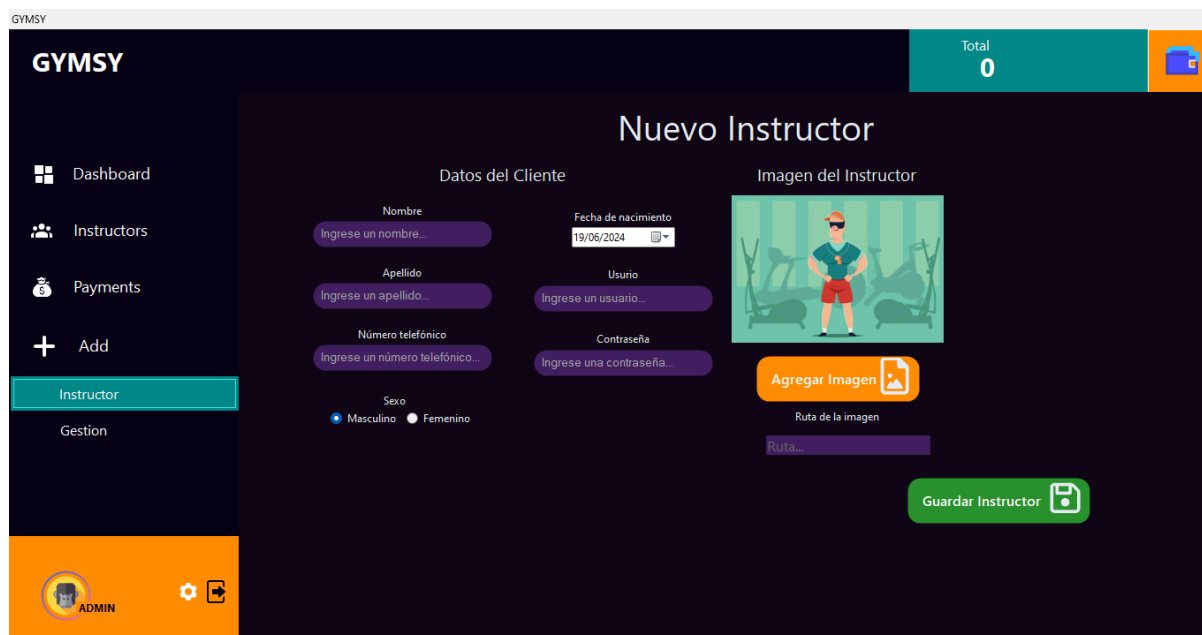


Figura 30

Pantalla de pago de la recepción

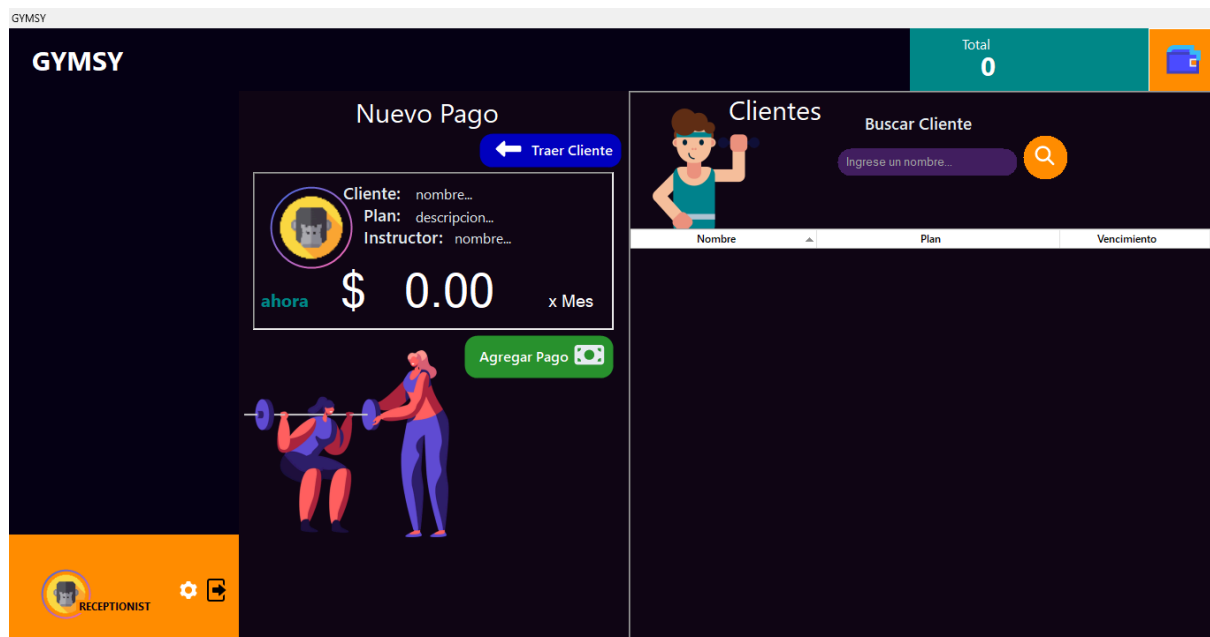


Figura 31

5. Referencia

- O. Vallejos, "Apunte de Análisis de Sistemas I", Facultad de Ciencias Exactas y Naturales y Agrimensura, Universidad Nacional del Nordeste. [Online]. Available: http://exa.unne.edu.ar/informatica/anasistem1/public_html/home.html.
- E. Yourdon, "Análisis estructurado Moderno", 2ª ed. Madrid: Prentice Hall, 1989.
- R. S. Pressman, "Ingeniería del Software: Un enfoque práctico", 5ª ed. Madrid: McGraw-Hill, 2002.
- A. Oliveros, "Importancia de los Requerimientos", material de Tópicos I, Maestría en Ingeniería de Software, Universidad Nacional del Nordeste, Corrientes, Argentina.
- I. Sommerville, "Ingeniería del Software", 7ª ed. Madrid: Pearson, 2005, Cap. 6, "Norma Estándar IEEE 830".
- K. Kendall y J. Kendall, "Análisis y diseño de sistemas", 6ª ed. Madrid: Pearson, 2005, Cap. 4.
- O. Vallejos, "Apuntes de la asignatura Ingeniería del Software I", Facultad de Ciencias Exactas y Naturales y Agrimensura, Universidad Nacional del Nordeste.
- Scrum Manager. [Online]. Available: <http://www.scrummanager.net>.
- M. Trigas, "Metodología Scrum TFC". [Online]. Available: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>.
- C. Larman, "UML y Patrones", 3ª ed. Madrid: Prentice Hall, 2004.
- D. Potel, "MVP: Model-View-Presenter (MVP) Design Pattern". [Online]. Available: <https://www.wildcrest.com/Potel/Portfolio/mvp.pdf>.
- "MVP Design Pattern", [Online]. Available: <https://help.windev.com/es-ES/?1000021496>.