
Tecnológico de Costa Rica

**Sistema Centro de Entrenamiento: Proyecto Fase
III
Software Architecture Document**

**Jose Pablo Espinoza López
Jose Pablo Fernández Cubillo
Roberto Vidal Patiño**

Tabla de Contenidos

Introducción	3
Propósito	3
Alcance	3
Definiciones, Acrónimos y Abreviaturas	3
Referencias	3
Visión General	3
Representación Arquitectónica	3
Metas y Restricciones	4
Vista de Casos de Uso	5
Vista Lógica	22
Descripción	22
Paquetes Significativos	22
Realizaciones de Casos de Uso	23
Vista de Procesos	34
Vista de Despliegue	35
Vista de Implementación	35
Descripción	35
Capas	35
Vista de Datos	36
Tamaño y Rendimiento	36
Calidad	36

Software Architecture Document

1. Introducción

La nueva normalidad, en una vida regida bajo nuevas reglas por la pandemia del Covid-19, esto ha cambiado la manera de interactuar con los servicios públicos como clínicas, restaurantes, tiendas de ropa etc. Por lo que se pretende desarrollar una aplicación web que trate de facilitar la manera de hacer uso de estos bienes. Para realizar esta plataforma se analizarán a fondo los requerimientos dados por el cliente para seleccionar la mejor forma de desarrollar el proyecto y la representación arquitectónica tomando en cuenta los objetivos, metas y propósitos.

Se declarará y explicará por medio de diagramas la arquitectura seleccionada y las tecnologías que se usarán para realizar este proyecto para que se realice de la manera más óptima y eficiente posible. En esta sección se muestra el propósito alcance, definiciones usadas en el proyecto referencias y una visión general.

Para que se puedan visualizar mejor los diagramas disponemos de este link donde se encuentran alojados los diagramas que fueron modificados en esta fase:

https://estudiantecr-my.sharepoint.com/:f/g/personal/robertovidal_estudiantec_cr/EhPFzlMfPnZJiInIVhVu6elBuka3Lnn_b8yRmXasiIXFuQ?e=ECGjNp

1.1 Propósito

El propósito de este proyecto es resolver por medio de una aplicación web el problema actual de algunos negocios que ofrecen servicios públicos y que por motivos de la pandemia deben tener el mínimo contacto posible entre los clientes y colaboradores.

Para llevar a cabo este objetivo se desarrollará un programa web utilizando una arquitectura 4+1 vistas vistas en donde se describe la arquitectura del sistema de software para explicar de una mejor manera las vistas de cómo funcionará .

También se busca implementar la arquitectura que mejor se adapte para brindar una flexibilidad en caso de que se tengan que hacer futuras modificaciones o agregar nuevas funcionalidades al programa.

1.2 Alcance

El alcance de este documento es la descripción de cómo se desea que sea desarrollado el sistema, intentando detallar lo máximo posible en cada parte de este. Cada sección de este documento describe cada parte del sistema final, para que a la hora de desarrollarlo se tenga una claridad de cómo debe ser su implementación.

1.3 Definiciones, Acrónimos y Abreviaturas

- Modelo 4 + 1: Organiza una descripción de arquitectura de software utilizando 5 vistas concurrentes, las cuales cada una tiene sus propios asuntos. Los arquitectos capturan sus decisiones de diseño en 4 vistas y usan la quinta para validarlas.

1.4 Referencias

P. B. Kruchten. (1995). The 4+1 view model of architecture

1.5 Visión General

Justo después de esta sección de introducción viene la parte de representación arquitectónica, donde se describe qué modelo se utilizará en el documento. Después va la parte de metas y restricciones, aquí se coloca dónde el sistema podría verse comprometido y los objetivos del sistema. En la vista de casos de uso se pone el diagrama de casos de uso y las plantillas de todos los casos de uso. En la vista lógica se coloca el diagrama de clases, su descripción y diagramas de secuencia con el funcionamiento de cada caso de uso. En la vista de procesos va un diagrama de actividad, con los diferentes procesos del sistema, para identificar dependencias entre procesos. En la vista de despliegue va el diagrama de despliegue con los diferentes dispositivos del sistema. En la vista de implementación viene un diagrama de componentes con las diferentes capas del sistema, cada una con su descripción. En la vista de datos viene el modelo de la base de datos.

2. Representación Arquitectónica

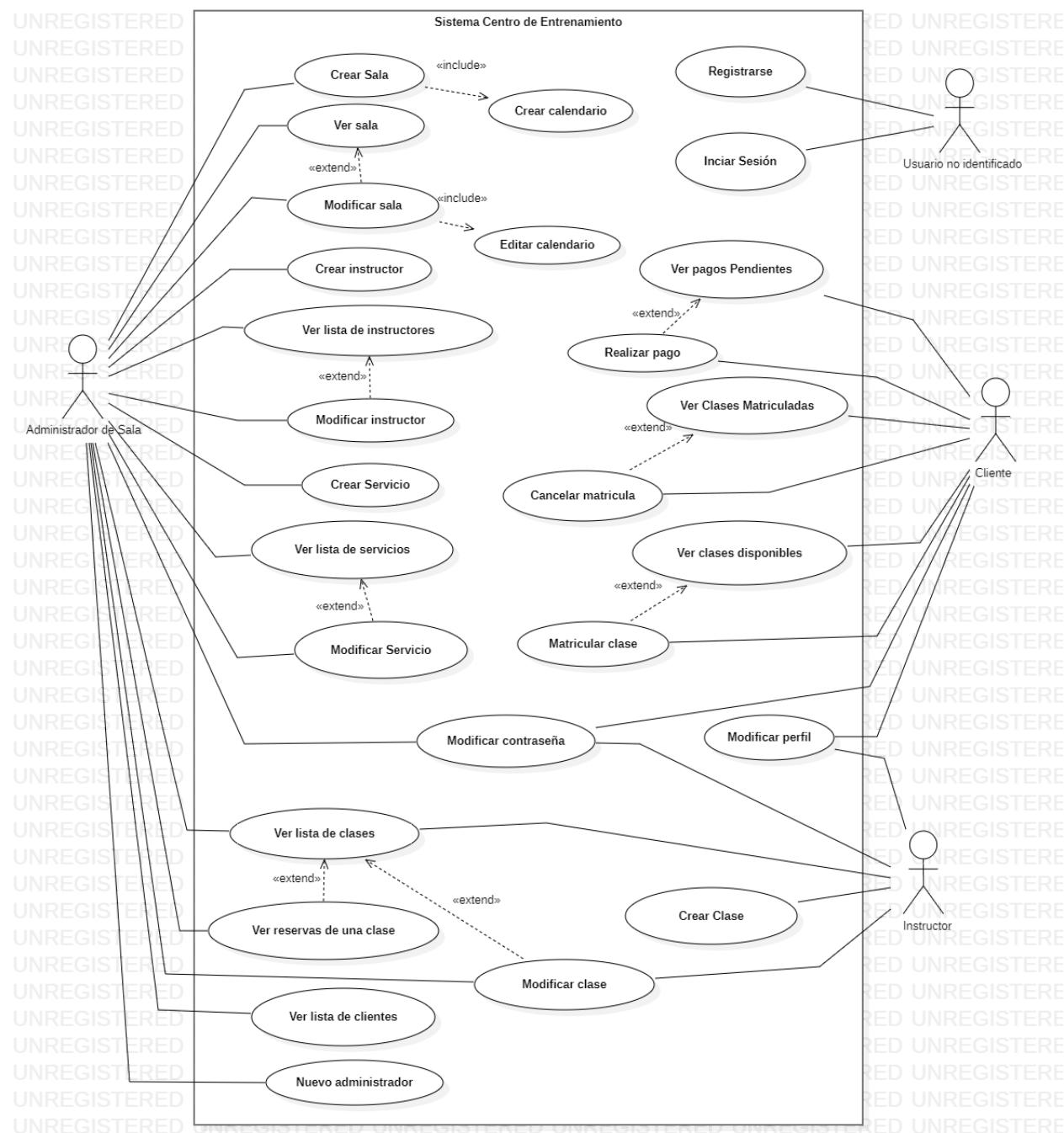
En este proyecto se hará uso del modelo 4+1 vistas para describir la arquitectura que se implementará en el sistema. Este modelo contiene las siguientes vistas:

- Vista Lógica: Requisitos funcionales del sistema y de lo que debe hacer, las funciones y servicios que se han definido. Se entra en detalle de qué exactamente se tiene que resolver. Se complementa con el Diagrama de Clases y el Diagrama de Paquetes
- Vista de Procesos: Se representan los flujos de trabajo paso a paso de los componentes del sistema. También se muestran requisitos no funcionales como ejecución, disponibilidad, tolerancia a fallos, integridad, seguridad, confiabilidad, etc. Se utilizan Diagramas de Actividad para saber cuáles procesos se ejecutan de manera independiente o si existe alguna dependencia entre procesos.
- Vista de Desarrollo: Se le llama también Vista de Despliegue. Se debe considerar si el sistema se puede pensar por subdivisiones por módulos, por componentes o por subsistemas. Se muestran las dependencias que hay entre ellos y la manera en que se comunican. Se puede complementar utilizando un Diagrama de Componentes y un Diagrama de Paquetes.
- Vista Física: Se le llama también Vista de Implementación. Se definen los componentes que se utilizan para que el sistema sea una realidad, pueden ser componentes tanto de software como de hardware. Se puede utilizar para hablar con el cliente sobre lo que se tiene que utilizar o lo que tiene que pagar adicionalmente. Normalmente se trabaja con Diagramas de Despliegue.
- Escenarios: Se juntan las 4 vistas para describir secuencias, interacciones, procesos, etc. Se realizan validaciones probando diferentes escenarios que pueden ocurrir en el sistema. Se utiliza el Diagrama de Casos de Uso.

3. Metas y Restricciones

Una de las restricciones más importantes a mencionar de este sistema es que cada administrador de sala solo puede administrar una sala. También, un instructor debe ser agregado por algún administrador a su sala para poder crear alguna clase. Los objetivos principales de este sistema es lograr crear un sistema útil tanto para los emprendedores que desean ingresar a su sala de entrenamiento, como para las personas que desean buscar un lugar para entrenar.

4. Vista de Casos de Uso



Use Case ID:	001
Use Case Name:	Registrar Usuario
Created By:	Roberto Vidal, Jose Espinoza, Jose Fernández

Actors:	Usuario no identificado
Description:	Se registrará un usuario en la Base de Datos

Trigger:	El usuario seleccionó en el menú inicial registrarse
Preconditions:	1. El usuario no está registrado en la Base de Datos
Postconditions:	1. El usuario estará en la Base de Datos
Normal Flow:	<ol style="list-style-type: none"> 1. El usuario llena el formulario con los datos 2. El usuario oprime el botón de registrar 3. El sistema revisa los datos y se guardan en la Base de Datos 4. Se muestra un mensaje al usuario de que los datos fueron ingresados correctamente
Alternative Flows:	<ol style="list-style-type: none"> 2.1 El usuario oprime el botón para volver 2.2 El sistema manda al usuario a la página de inicio de sesión
Exceptions:	<ol style="list-style-type: none"> 3.1 El Sistema avisa al usuario que llenó los datos de manera incorrecta 3.2 El usuario debe volver a llenar los datos
Includes:	N/A
Priority:	2
Frequency of Use:	Diario
Business Rules:	N/A
Special Requirements:	<ol style="list-style-type: none"> 1. Seguridad 2. Mantenibilidad 3. Escalabilidad
Assumptions:	N/A
Notes and Issues:	N/A

Use Case ID:	002
Use Case Name:	Iniciar Sesión
Created By:	Roberto Vidal, Jose Espinoza, Jose Fernández

Actors:	Usuario no identificado
Description:	Un usuario iniciará sesión en el sistema.
Trigger:	El usuario seleccionó en el menú inicial iniciar sesión
Preconditions:	1. El usuario está registrado en la Base de Datos
Postconditions:	1. El usuario ingresará al sistema
Normal Flow:	<ol style="list-style-type: none"> 1. El usuario ingresa su nombre de usuario y contraseña 2. El usuario oprime el botón 3. El sistema revisa los datos del usuario y se le avisa al usuario que se inició sesión de manera correcta
Alternative Flows:	<ol style="list-style-type: none"> 2.1 El usuario oprime el botón para registrarse 2.2 El sistema manda al usuario a la página de registro
Exceptions:	<ol style="list-style-type: none"> 3.1 El Sistema avisa al usuario que llenó los datos de manera incorrecta o que la contraseña o el usuario están incorrectos 3.2 El usuario debe volver a llenar los datos
Includes:	N/A
Priority:	2
Frequency of Use:	Diario
Business Rules:	N/A
Special Requirements:	<ol style="list-style-type: none"> 1. Seguridad 2. Mantenibilidad 3. Escalabilidad

Assumptions:	N/A
Notes and Issues:	N/A

Use Case ID:	003
Use Case Name:	Crear Sala
Created By:	Roberto Vidal, Jose Espinoza, Jose Fernández

Actors:	Administrador de Sala
Description:	Un administrador creará una sala de entrenamiento
Trigger:	El usuario escoge la opción de crear una sala.
Preconditions:	<ol style="list-style-type: none"> 1. El usuario está registrado en la Base de Datos 2. El usuario no tiene ninguna sala registrada
Postconditions:	<ol style="list-style-type: none"> 1. Se guardará la información de la sala en la base de datos
Normal Flow:	<ol style="list-style-type: none"> 1. El usuario llena el formulario con la información 2. El usuario oprime el botón de crear sala 3. El sistema revisa los datos y los guarda en la Base de Datos 4. El sistema avisa al usuario que los datos de la sala fueron registrados correctamente 5. El sistema manda al usuario a la página de crear calendario.
Alternative Flows:	<ol style="list-style-type: none"> 1.1 El usuario oprime el botón de cerrar sesión 1.2 Se manda al usuario a la página de inicio de sesión
Exceptions:	<ol style="list-style-type: none"> 4.1 El Sistema avisa al usuario que llenó los datos de manera incorrecta. 4.2 El usuario debe volver a llenar los datos
Includes:	Crear calendario
Priority:	1
Frequency of Use:	Diario
Business Rules:	N/A
Special Requirements:	<ol style="list-style-type: none"> 1. Seguridad 2. Mantenibilidad 3. Escalabilidad
Assumptions:	N/A
Notes and Issues:	Tomar en cuenta que además de los parámetros propios de la sala se crea el calendario y se añaden los servicios que da la sala

Use Case ID:	004
Use Case Name:	Ver sala
Created By:	Roberto Vidal, Jose Espinoza, Jose Fernández

Actors:	Administrador de Sala
Description:	Un administrador verá la configuración de su sala de entrenamiento
Trigger:	El usuario escoge la opción de ver la configuración en la página de administración de la sala
Preconditions:	<ol style="list-style-type: none"> 1. El usuario está registrado en la Base de Datos 2. El usuario tiene una sala registrada
Postconditions:	N/A
Normal Flow:	<ol style="list-style-type: none"> 1. El sistema presenta la configuración de la sala

	2. El usuario toca el botón de modificar sala
Alternative Flows:	1.1 El usuario oprime el botón de cerrar sesión 1.2 Se manda al usuario a la página de inicio de sesión 2.1 Se manda al usuario a la página de modificar sala
Exceptions:	N/A
Includes:	N/A
Priority:	1
Frequency of Use:	Diario
Business Rules:	N/A
Special Requirements:	1. Seguridad 2. Mantenibilidad 3. Escalabilidad
Assumptions:	N/A
Notes and Issues:	N/A

Use Case ID:	005
Use Case Name:	Modificar Sala
Created By:	Roberto Vidal, Jose Espinoza, Jose Fernández

Actors:	Administrador de Sala
Description:	Un administrador editará los servicios de su sala de entrenamiento
Trigger:	El usuario escoge la opción de modificar servicios en la página de administración de la sala
Preconditions:	1. El usuario está registrado en la Base de Datos 2. El usuario tiene una sala registrada
Postconditions:	1. Se editarán los parámetros de la sala
Normal Flow:	1. El usuario cambia los parámetros de la sala 2. El usuario puede cambiar el calendario o los servicios 3. El sistema revisa los datos y los guarda en la Base de Datos 4. El sistema avisa al usuario que los servicios se modificaron correctamente 5. El sistema manda al usuario a la página de modificar calendario.
Alternative Flows:	1.1 El usuario oprime el botón de cerrar sesión 1.2 Se manda al usuario a la página de inicio de sesión 1.1 El usuario oprime el botón de volver 1.2 Se manda al usuario a la página de ver sala
Exceptions:	4.1 El Sistema avisa al usuario que los datos se guardaron de forma incorrecta 4.2 El sistema devuelve al usuario a la página de administración de la sala
Includes:	Modificar calendario
Priority:	2
Frequency of Use:	Diario
Business Rules:	N/A
Special Requirements:	1. Seguridad

	<ol style="list-style-type: none"> 2. Mantenibilidad 3. Escalabilidad
Assumptions:	N/A
Notes and Issues:	Tomar en cuenta que además de los parámetros propios de la sala se puede modificar el calendario y los servicios de la sala

Use Case ID:	006
Use Case Name:	Crear calendario
Created By:	Roberto Vidal, Jose Espinoza, Jose Fernández

Actors:	Administrador de Sala
Description:	Un administrador creará el calendario de su sala de entrenamiento.
Trigger:	El usuario crea una sala.
Preconditions:	<ol style="list-style-type: none"> 1. El usuario está registrado en la Base de Datos 2. El usuario tiene una sala registrada
Postconditions:	<ol style="list-style-type: none"> 1. El sistema mantiene los datos del calendario
Normal Flow:	<ol style="list-style-type: none"> 1. El usuario selecciona los días y el horario que desea. 2. El usuario oprime el botón de confirmar
Alternative Flows:	<ol style="list-style-type: none"> 1.1 El usuario oprime el botón de cerrar sesión 1.2 Se manda al usuario a la página de inicio de sesión 1.1 El usuario oprime el botón de volver 1.2 Se manda al usuario a la página de crear sala
Exceptions:	N/A
Includes:	N/A
Priority:	1
Frequency of Use:	Diario
Business Rules:	N/A
Special Requirements:	<ol style="list-style-type: none"> 1. Seguridad 2. Mantenibilidad 3. Escalabilidad
Assumptions:	N/A
Notes and Issues:	N/A

Use Case ID:	007
Use Case Name:	Modificar calendario
Created By:	Roberto Vidal, Jose Espinoza, Jose Fernández

Actors:	Administrador de Sala
Description:	Un administrador modificará el calendario de su sala de entrenamiento.
Trigger:	El usuario modifica la sala.
Preconditions:	<ol style="list-style-type: none"> 1. El usuario está registrado en la Base de Datos 2. El usuario tiene una sala registrada
Postconditions:	<ol style="list-style-type: none"> 1. El sistema mantiene los datos del calendario
Normal Flow:	<ol style="list-style-type: none"> 1. El usuario selecciona los días y el horario que desea. 2. El usuario oprime el botón de confirmar
Alternative Flows:	<ol style="list-style-type: none"> 1.1 El usuario oprime el botón de cerrar sesión

	<p>1.2 Se manda al usuario a la página de inicio de sesión</p> <p>1.1 El usuario oprime el botón de volver</p> <p>1.2 Se manda al usuario a la página de modificar sala</p>
Exceptions:	N/A
Includes:	N/A
Priority:	1
Frequency of Use:	Diario
Business Rules:	N/A
Special Requirements:	<ul style="list-style-type: none"> 1. Seguridad 2. Mantenibilidad 3. Escalabilidad
Assumptions:	N/A
Notes and Issues:	N/A

Use Case ID:	008
Use Case Name:	Crear instructor
Created By:	Roberto Vidal, Jose Espinoza, Jose Fernández

Actors:	Administrador de Sala
Description:	Un administrador creará un instructor.
Trigger:	El usuario escoge la opción de añadir instructor.
Preconditions:	<ul style="list-style-type: none"> 1. El usuario está registrado en la Base de Datos 2. El instructor no está registrado en la Base de Datos
Postconditions:	<ul style="list-style-type: none"> 1. El instructor queda registrado
Normal Flow:	<ul style="list-style-type: none"> 1. El usuario ingresa la información del instructor 2. El usuario oprime el botón de crear instructor 3. El sistema revisa los datos y los guarda en la Base de Datos 4. El sistema avisa al usuario que el instructor fue agregado correctamente.
Alternative Flows:	<p>1.1 El usuario oprime el botón de cerrar sesión 1.2 Se manda al usuario a la página de inicio de sesión</p> <p>1.1 El usuario oprime el botón de volver 1.2 Se manda al usuario a la página de instructores</p>
Exceptions:	<ul style="list-style-type: none"> 3.1 El Sistema avisa al usuario que llenó los datos de manera incorrecta. 3.2 El usuario debe corregir los datos
Includes:	N/A
Priority:	2
Frequency of Use:	Diario
Business Rules:	N/A
Special Requirements:	<ul style="list-style-type: none"> 1. Seguridad 2. Mantenibilidad 3. Escalabilidad
Assumptions:	N/A
Notes and Issues:	N/A

Use Case ID:	009
Use Case Name:	Ver lista de instructores
Created By:	Roberto Vidal, Jose Espinoza, Jose Fernández

Actors:	Administrador de Sala
Description:	Un administrador verá la lista de instructores de su sala de entrenamiento.
Trigger:	El usuario escoge la opción de ver la lista de instructores en la página de administración de la sala.
Preconditions:	<ol style="list-style-type: none"> 1. El usuario está registrado en la Base de Datos 2. Hay al menos 1 instructor registrado
Postconditions:	<ol style="list-style-type: none"> 1. Se muestran los datos de los instructores
Normal Flow:	<ol style="list-style-type: none"> 1. La página muestra los instructores con un botón de eliminar y uno de modificar. 2. El usuario presiona el botón para eliminar los instructores que desee. 3. El usuario presiona el botón para modificar los instructores que desee.
Alternative Flows:	<ol style="list-style-type: none"> 1.1 El usuario oprime el botón de cerrar sesión 1.2 Se manda al usuario a la página de inicio de sesión 2.1 El sistema pide confirmación 2.2 El sistema realiza el cambio en la base de datos 3.3 El sistema manda al usuario a la página de modificar el instructor. 3.3 Luego se devuelve a la página de instructores.
Exceptions:	N/A
Includes:	N/A
Priority:	2
Frequency of Use:	Diario
Business Rules:	N/A
Special Requirements:	<ol style="list-style-type: none"> 1. Seguridad 2. Mantenibilidad 3. Escalabilidad
Assumptions:	N/A
Notes and Issues:	N/A

Use Case ID:	010
Use Case Name:	Modificar instructor
Created By:	Roberto Vidal, Jose Espinoza, Jose Fernández

Actors:	Administrador de Sala
Description:	Un administrador modificará un instructor.
Trigger:	El usuario escoge la opción de modificar instructor
Preconditions:	<ol style="list-style-type: none"> 3. El usuario está registrado en la Base de Datos 4. El instructor está registrado en la Base de Datos

Postconditions:	2. El instructor queda modificado
Normal Flow:	5. El usuario ingresa la información que quiere modificar del instructor 6. El usuario oprime el botón de modificar instructor 7. El sistema revisa los datos y los guarda en la Base de Datos 8. El sistema avisa al usuario que el instructor fue modificado correctamente.
Alternative Flows:	1.1 El usuario oprime el botón de cerrar sesión 1.2 Se manda al usuario a la página de inicio de sesión 1.1 El usuario oprime el botón de volver 1.2 Se manda al usuario a la página de instructores
Exceptions:	3.1 El Sistema avisa al usuario que llenó los datos de manera incorrecta. 3.2 El usuario debe corregir los datos
Includes:	N/A
Priority:	2
Frequency of Use:	Diario
Business Rules:	N/A
Special Requirements:	4. Seguridad 5. Mantenibilidad 6. Escalabilidad
Assumptions:	N/A
Notes and Issues:	N/A

Use Case ID:	011
Use Case Name:	Crear Servicio
Created By:	Roberto Vidal, Jose Espinoza, Jose Fernández

Actors:	Administrador de Sala
Description:	Un administrador creará un servicio.
Trigger:	El usuario escoge la opción de crear servicio en la página de editar servicios
Preconditions:	1. El usuario está registrado en la Base de Datos 2. El usuario tiene una sala registrada
Postconditions:	1. Se guardará el servicio en la Base de Datos
Normal Flow:	1. El usuario ingresa los datos del servicio. 2. El usuario oprime el botón de crear servicio 3. El sistema revisa los datos y los guarda en la Base de Datos 4. El sistema avisa al usuario que el servicio fue creado correctamente.
Alternative Flows:	1.1 El usuario oprime el botón de cerrar sesión 1.2 Se manda al usuario a la página de inicio de sesión 1.1 El usuario oprime el botón de volver 1.2 Se manda al usuario a la página de servicios.
Exceptions:	3.1 El Sistema avisa al usuario que llenó los datos de manera incorrecta. 3.2 El usuario debe corregir los datos

Includes:	N/A
Priority:	2
Frequency of Use:	Diario
Business Rules:	N/A
Special Requirements:	<ul style="list-style-type: none"> 1. Seguridad 2. Mantenibilidad 3. Escalabilidad
Assumptions:	N/A
Notes and Issues:	N/A

Use Case ID:	012
Use Case Name:	Ver lista de servicios
Created By:	Roberto Vidal, Jose Espinoza, Jose Fernández

Actors:	Administrador de Sala
Description:	Un administrador ve la lista de servicios que hay creados actualmente
Trigger:	El usuario escoge la opción de mostrar servicios
Preconditions:	<ul style="list-style-type: none"> 1. El usuario está registrado en la Base de Datos 2. Hay al menos 1 servicio registrado en la base de datos
Postconditions:	<ul style="list-style-type: none"> 1. Se muestran los datos de los servicios
Normal Flow:	<ul style="list-style-type: none"> 1. El sistema muestra los servicios con un botón de eliminar y uno de modificar 2. El usuario oprime el botón de eliminar un servicio 3. El usuario oprime el botón de modificar un servicio
Alternative Flows:	<ul style="list-style-type: none"> 1.1 El usuario oprime el botón de cerrar sesión 1.2 Se manda al usuario a la página de inicio de sesión 1.1 El usuario oprime el botón de volver 1.2 Se manda al usuario a la página de administración de la sala. 2.1 El usuario escoge la opción de eliminar servicio 2.2 El sistema elimina el servicio de la base de datos 3.1 El sistema manda al usuario a la pantalla de modificar servicio 3.2 Vuelve a la pantalla de servicios
Exceptions:	2.2.1 El sistema indica que no se pudo eliminar el servicio
Includes:	N/A
Priority:	2
Frequency of Use:	Diario
Business Rules:	N/A
Special Requirements:	<ul style="list-style-type: none"> 1. Seguridad 2. Mantenibilidad 3. Escalabilidad
Assumptions:	N/A
Notes and Issues:	N/A

Use Case ID:	013
--------------	-----

Use Case Name:	Modificar Servicio
Created By:	Roberto Vidal, Jose Espinoza, Jose Fernández

Actors:	Administrador de Sala
Description:	Un administrador modificará un servicio
Trigger:	El usuario escoge la opción de modificar un servicio
Preconditions:	<ol style="list-style-type: none"> 1. El usuario está registrado en la Base de Datos 2. El servicio está registrado en la Base de Datos
Postconditions:	<ol style="list-style-type: none"> 1. Se modificará el servicio creado en la Base de Datos
Normal Flow:	<ol style="list-style-type: none"> 1. El usuario modifica los datos del servicio 2. El usuario oprime el botón de modificar 3. El sistema revisa los datos y los guarda en la Base de Datos. 4. El sistema avisa al usuario que la clase fue modificada correctamente.
Alternative Flows:	<ol style="list-style-type: none"> 1.1 El usuario oprime el botón de cerrar sesión 1.2 Se manda al usuario a la página de inicio de sesión 1.1 El usuario oprime el botón de volver 1.2 Se manda al usuario a la página de lista de clases
Exceptions:	<ol style="list-style-type: none"> 3.1 El Sistema avisa al usuario que llenó los datos de manera incorrecta. 3.2 El usuario debe corregir los datos 4.1 El sistema indica que el servicio no se pudo modificar
Includes:	N/A
Priority:	1
Frequency of Use:	Diario
Business Rules:	N/A
Special Requirements:	<ol style="list-style-type: none"> 1. Seguridad 2. Mantenibilidad 3. Escalabilidad
Assumptions:	N/A
Notes and Issues:	El administrador puede cambiar el estado de la clase

Use Case ID:	014
Use Case Name:	Crear Clase
Created By:	Roberto Vidal, Jose Espinoza, Jose Fernández

Actors:	Instructor
Description:	Un instructor creará una clase.
Trigger:	El usuario escoge la opción de crear una clase.
Preconditions:	<ol style="list-style-type: none"> 1. El usuario está registrado en la Base de Datos 2. El debe estar registrado en una sala como instructor.
Postconditions:	<ol style="list-style-type: none"> 1. Se guardará la clase creada en la Base de Datos
Normal Flow:	<ol style="list-style-type: none"> 1. El usuario llena el formulario con los datos de la clase 2. El usuario oprime el botón de crear clase 3. El sistema revisa los datos y los guarda en la Base de Datos

	4. El sistema avisa al usuario que la clase fue creada correctamente
Alternative Flows:	1.1 El usuario oprime el botón de cerrar sesión 1.2 Se manda al usuario a la página de inicio de sesión 1.1 El usuario oprime el botón de volver 1.2 Se manda al usuario a la página de inicio
Exceptions:	3.1 El Sistema avisa al usuario que llenó los datos de manera incorrecta 3.2 El usuario debe corregir los datos
Includes:	N/A
Priority:	1
Frequency of Use:	Diario
Business Rules:	N/A
Special Requirements:	1. Seguridad 2. Mantenibilidad 3. Escalabilidad
Assumptions:	N/A
Notes and Issues:	N/A

Use Case ID:	015
Use Case Name:	Ver lista de clases
Created By:	Roberto Vidal, Jose Espinoza, Jose Fernández

Actors:	Administrador de Sala, Instructor
Description:	Un administrador o instructor abre la página del calendario para ver la lista de clases
Trigger:	El usuario escoge la opción de calendario
Preconditions:	1. El usuario está registrado en la Base de Datos 2. Debe haber al menos una clase registrada
Postconditions:	1. Se muestran las clases en el calendario
Normal Flow:	1. El sistema muestra las clases que la sala tiene y da la opción de editar y eliminar las clases. 2. El usuario oprime el botón de modificar alguna clase 3. El instructor elimina alguna clase
Alternative Flows:	1.1 El usuario oprime el botón de cerrar sesión 1.2 Se manda al usuario a la página de inicio de sesión 2.1 El sistema manda al usuario a la página de modificar clase
Exceptions:	1.1 El sistema indica que no hay clases por mostrar
Includes:	N/A
Priority:	2
Frequency of Use:	Diario
Business Rules:	N/A
Special Requirements:	4. Seguridad 5. Mantenibilidad 6. Escalabilidad
Assumptions:	N/A

Notes and Issues:	N/A
-------------------	-----

Use Case ID:	016
Use Case Name:	Ver lista de reservas de una clase
Created By:	Roberto Vidal, Jose Espinoza, Jose Fernández

Actors:	Administrador de Sala
Description:	Un administrador en la página de calendario ve los clientes que hicieron reserva en una clase
Trigger:	El usuario escoge la opción ver reservas de una clase
Preconditions:	<ol style="list-style-type: none"> 1. El usuario está registrado en la Base de Datos 2. La clase debe estar registrada en la base de datos
Postconditions:	<ol style="list-style-type: none"> 1. Se las personas que reservaron la clase
Normal Flow:	<ol style="list-style-type: none"> 1. El sistema muestra las personas que reservaron la clase
Alternative Flows:	<ol style="list-style-type: none"> 1.1 El usuario oprime el botón de cerrar sesión 1.2 Se manda al usuario a la página de inicio de sesión
Exceptions:	1.1 El sistema indica que no hay usuarios que hayan hecho una reserva
Includes:	N/A
Priority:	2
Frequency of Use:	Diario
Business Rules:	N/A
Special Requirements:	<ol style="list-style-type: none"> 1. Seguridad 2. Mantenibilidad 3. Escalabilidad
Assumptions:	N/A
Notes and Issues:	N/A

Use Case ID:	017
Use Case Name:	Modificar Clase
Created By:	Roberto Vidal, Jose Espinoza, Jose Fernández

Actors:	Administrador de Sala, Instructor
Description:	Un administrador o instructor modificará una clase
Trigger:	El usuario escoge la opción de modificar una clase en la lista de clases
Preconditions:	<ol style="list-style-type: none"> 1. El usuario está registrado en la Base de Datos 2. La sala está registrada en la Base de Datos 3. La clase está registrada en la Base de Datos
Postconditions:	1. Se modificará la clase creada en la Base de Datos
Normal Flow:	<ol style="list-style-type: none"> 1. El usuario modifica los datos de la clase 2. El usuario oprime el botón de confirmar 3. El sistema revisa los datos y los guarda en la Base de Datos 4. El sistema avisa al usuario que la clase fue modificada correctamente
Alternative Flows:	1.1 El usuario oprime el botón de cerrar sesión

	1.2 Se manda al usuario a la página de inicio de sesión
Exceptions:	3.1 El Sistema avisa al usuario que llenó los datos de manera incorrecta. 3.2 El usuario debe volver a llenar los datos 4.1 El sistema indica que la clase no se pudo modificar
Includes:	N/A
Priority:	1
Frequency of Use:	Diario
Business Rules:	N/A
Special Requirements:	4. Seguridad 5. Mantenibilidad 6. Escalabilidad
Assumptions:	N/A
Notes and Issues:	El administrador puede cambiar el estado de la clase

Use Case ID:	018
Use Case Name:	Ver la lista de clientes
Created By:	Roberto Vidal, Jose Espinoza, Jose Fernández

Actors:	Administrador de Sala
Description:	Un administrador verá la lista de clientes de su sala de entrenamiento
Trigger:	El usuario escoge la opción de ver la lista de clientes en la página de administración de la sala
Preconditions:	1. El usuario está registrado en la Base de Datos 2. El usuario tiene una sala registrada
Postconditions:	Se muestra la información de los clientes
Normal Flow:	1. El sistema muestra al usuario la lista de clientes. 2. El usuario oprime el botón de volver.
Alternative Flows:	2.1 El usuario oprime el botón de cerrar sesión 2.2 Se manda al usuario a la página de inicio de sesión
Exceptions:	N/A
Includes:	N/A
Priority:	2
Frequency of Use:	Diario
Business Rules:	N/A
Special Requirements:	1. Seguridad 2. Mantenibilidad 3. Escalabilidad
Assumptions:	N/A
Notes and Issues:	N/A

Use Case ID:	019
Use Case Name:	Ver la lista de clases disponibles
Created By:	Roberto Vidal, Jose Espinoza, Jose Fernández

Actors:	Cliente
---------	---------

Description:	Un cliente ve la lista de clases disponibles.
Trigger:	El usuario escoge la opción de ver la lista de clases disponibles en el calendario
Preconditions:	1. El usuario está registrado en la Base de Datos 2. Debe existir al menos una clase en la Base de Datos
Postconditions:	N/A
Normal Flow:	1. La página muestra las clases disponibles con su opción de matrícula 2. El usuario oprime el botón de matrícula en alguna clase
Alternative Flows:	2.1 El usuario oprime el botón de cerrar sesión 2.2 Se manda al usuario a la página de inicio de sesión 2.1 El usuario oprime el botón de volver. 2.2 El sistema manda al usuario a la página de inicio
Exceptions:	N/A
Includes:	N/A
Priority:	1
Frequency of Use:	Diario
Business Rules:	N/A
Special Requirements:	1. Seguridad 2. Mantenibilidad 3. Escalabilidad
Assumptions:	N/A
Notes and Issues:	N/A

Use Case ID:	020
Use Case Name:	Ver clases matriculadas
Created By:	Roberto Vidal, Jose Espinoza, Jose Fernández

Actors:	Cliente
Description:	Un cliente ve sus clases matriculadas.
Trigger:	El usuario escoge la opción de ver sus clases matriculadas en el calendario
Preconditions:	1. El usuario está registrado en la Base de Datos 2. Debe existir al menos una clase en la que el cliente esté matriculado
Postconditions:	Se muestran las clases en las que el cliente está matriculado
Normal Flow:	1. La página muestra las clases matriculadas con una opción para cancelar la matrícula 2. El usuario oprime el botón de cancelar matrícula 3. El sistema se encarga de cancelar la matrícula.
Alternative Flows:	1.1 El usuario oprime el botón de cerrar sesión 1.2 Se manda al usuario a la página de inicio de sesión
Exceptions:	N/A
Includes:	N/A
Priority:	1
Frequency of Use:	Diario
Business Rules:	N/A

Special Requirements:	1. Seguridad 2. Mantenibilidad 3. Escalabilidad
Assumptions:	N/A
Notes and Issues:	N/A

Use Case ID:	021
Use Case Name:	Cancelar Matrícula
Created By:	Roberto Vidal, Jose Espinoza, Jose Fernández

Actors:	Cliente
Description:	Un cliente cancela la matrícula a alguna clase
Trigger:	El usuario escoge la opción cancelar matrícula en sus clases matriculadas
Preconditions:	1. El usuario está registrado en la Base de Datos 2. Debe existir al menos una clase en la que el cliente esté matriculado
Postconditions:	Se elimina la matrícula del cliente de la base de datos.
Normal Flow:	1. El sistema revisa si la cancelación se hace máximo 8 horas previas a la clase 2. Se encuentra que sí se hace máximo 8 horas previas. 3. Se elimina la matrícula 4. Se guarda el monto a favor del cliente. 5. Se le avisa al usuario que la matrícula se canceló correctamente y que le quedó un monto a favor para próximas matrículas.
Alternative Flows:	2.1 El sistema encuentra que la cancelación se realiza posterior a las 8 horas previas. 2.2 Se elimina la matrícula 2.3 Se le avisa al usuario que la matrícula se canceló correctamente pero que perdió el dinero por hacerlo posterior a las 8 horas previas.
Exceptions:	N/A
Includes:	N/A
Priority:	1
Frequency of Use:	Diario
Business Rules:	N/A
Special Requirements:	1. Seguridad 2. Mantenibilidad 3. Escalabilidad
Assumptions:	N/A
Notes and Issues:	N/A

Use Case ID:	022
Use Case Name:	Ver pagos pendientes
Created By:	Roberto Vidal, Jose Espinoza, Jose Fernández

Actors:	Cliente
Description:	Un cliente ve la lista de pagos pendientes
Trigger:	El usuario escoge la opción para ver los pagos pendientes

Preconditions:	1. El usuario está registrado en la Base de Datos 2. El cliente tiene estado de moroso 3. El cliente tiene pagos pendientes
Postconditions:	1. El sistema muestra todos los pagos pendientes del cliente
Normal Flow:	1. El sistema muestra los pagos pendientes con un botón para realizar el pago 2. El usuario presiona el botón de realizar pago
Alternative Flows:	1.1 El usuario oprime el botón de cerrar sesión 1.2 Se manda al usuario a la página de inicio de sesión 2.1 Se manda al usuario a la página de realizar pago
Exceptions:	3.1 No existen pagos pendientes
Includes:	N/A
Priority:	1
Frequency of Use:	Diario
Business Rules:	N/A
Special Requirements:	1. Seguridad 2. Mantenibilidad 3. Escalabilidad
Assumptions:	N/A
Notes and Issues:	N/A

Use Case ID:	023
Use Case Name:	Realizar pago
Created By:	Roberto Vidal, Jose Espinoza, Jose Fernández

Actors:	Cliente
Description:	El cliente realiza uno de sus pagos pendientes
Trigger:	El usuario escogió la opción de realizar pago
Preconditions:	1. El usuario está registrado en la Base de Datos 2. El cliente tiene estado de moroso
Postconditions:	1. El pago realizado se guarda en la Base de Datos
Normal Flow:	1. El usuario selecciona el método de pago 2. El usuario presiona el botón de pagar 3. El sistema avisa que el pago se realizó correctamente 4. El sistema cambia el estado del pago
Alternative Flows:	1.1 El usuario oprime el botón de cerrar sesión 1.2 Se manda al usuario a la página de inicio de sesión 4.1 El sistema revisa si le quedan pagos pendientes al cliente 4.2 El sistema cambia el estado del cliente si no quedan pagos pendientes
Exceptions:	3.1 El sistema avisa que hubo un error con el pago
Includes:	N/A
Priority:	1
Frequency of Use:	Diario
Business Rules:	N/A
Special Requirements:	4. Seguridad 5. Mantenibilidad

	6. Escalabilidad
Assumptions:	N/A
Notes and Issues:	N/A

Use Case ID:	024
Use Case Name:	Modificar perfil
Created By:	Roberto Vidal, Jose Espinoza, Jose Fernández

Actors:	Cliente o Instructor
Description:	El usuario cambia sus datos personales
Trigger:	El usuario escogió la opción de perfil
Preconditions:	1. El usuario está registrado en la Base de Datos
Postconditions:	1. Se guarda la información modificada en la Base de Datos
Normal Flow:	1. El usuario modifica los valores del formulario 2. El usuario presiona el botón de enviar 3. El sistema hace la modificación en la Base de Datos 4. El sistema avisa que se modificó correctamente
Alternative Flows:	1.1 El usuario oprime el botón de cerrar sesión 1.2 Se manda al usuario a la página de inicio de sesión
Exceptions:	3.1 El sistema avisa que hubo un error con la modificación
Includes:	N/A
Priority:	2
Frequency of Use:	Diario
Business Rules:	N/A
Special Requirements:	1. Seguridad 2. Mantenibilidad 3. Escalabilidad
Assumptions:	N/A
Notes and Issues:	N/A

Use Case ID:	025
Use Case Name:	Modificar contraseña
Created By:	Roberto Vidal, Jose Espinoza, Jose Fernández

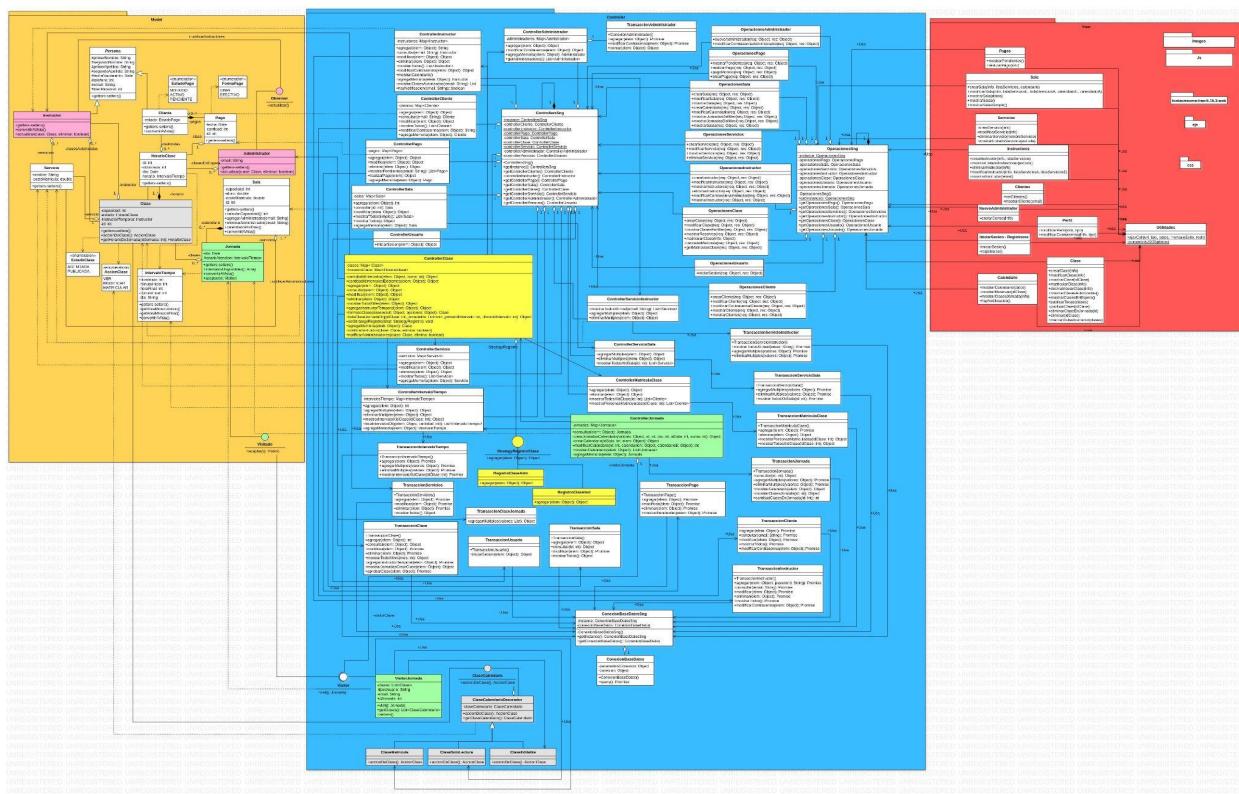
Actors:	Cliente, Instructor o Administrador
Description:	El usuario cambia su contraseña
Trigger:	El usuario escogió la opción de perfil
Preconditions:	1. El usuario está registrado en la Base de Datos
Postconditions:	1. Se guarda la información modificada en la Base de Datos
Normal Flow:	1. El usuario modifica los valores del formulario 2. El usuario presiona el botón de enviar 3. El sistema hace la modificación en la Base de Datos 4. El sistema avisa que se modificó correctamente
Alternative Flows:	1.1 El usuario oprime el botón de cerrar sesión 1.2 Se manda al usuario a la página de inicio de sesión
Exceptions:	3.1 El sistema avisa que hubo un error con la modificación
Includes:	N/A

Priority:	1
Frequency of Use:	Diario
Business Rules:	N/A
Special Requirements:	<ul style="list-style-type: none"> 1. Seguridad 2. Mantenibilidad 3. Escalabilidad
Assumptions:	N/A
Notes and Issues:	N/A

Use Case ID:	026
Use Case Name:	Nuevo administrador
Created By:	Roberto Vidal, Jose Espinoza, Jose Fernández

Actors:	Administrador
Description:	El usuario quiere agregar un nuevo administrador
Trigger:	El usuario escogió la opción de nuevo administrador
Preconditions:	<ul style="list-style-type: none"> 1. El usuario está registrado en la Base de Datos
Postconditions:	<ul style="list-style-type: none"> 1. Se guarda la información modificada en la Base de Datos 2. Se envía por correo una contraseña para que se pueda ingresar al sistema
Normal Flow:	<ul style="list-style-type: none"> 1. El usuario ingresa el correo al cual enviar la contraseña 2. El usuario presiona el botón de enviar 3. El sistema inserta los datos en la Base de Datos 4. El sistema avisa que se creó correctamente
Alternative Flows:	<ul style="list-style-type: none"> 1.1 El usuario oprime el botón de cerrar sesión 1.2 Se manda al usuario a la página de inicio de sesión
Exceptions:	<ul style="list-style-type: none"> 3.1 El sistema avisa que hubo un error con el proceso
Includes:	N/A
Priority:	1
Frequency of Use:	Diario
Business Rules:	N/A
Special Requirements:	<ul style="list-style-type: none"> 1. Seguridad 2. Mantenibilidad 3. Escalabilidad
Assumptions:	N/A
Notes and Issues:	N/A

5. Vista Lógica



5.1 Descripción

El sistema se organizó utilizando una arquitectura MVC porque nos permite realizar una separación entre las diferentes partes de la aplicación con el fin de lograr una separación entre los componentes que la forman.

5.2 Paquetes Significativos

5.2.1 Model

En este paquete están presentes las representaciones de los datos que maneja el sistema. Estos son actores como el instructor con los servicios que brinda y el cliente con sus pagos, la sala con su calendario, clase y servicios.

Instructor y Cliente heredan de la clase abstracta Persona, por tanto, tienen los atributos primerNombre, segundoNombre, primerApellido, segundoApellido, fechaNacimiento, telefono, email e identificacion. Cliente por su parte tiene el atributo estadoPago (tipo EstadoPago que es una enumeración) y está unida con la clase Pago por medio de una composición múltiple y Pago está unida a Clase por agregación. Instructor está unido con Sevicio por una agregación múltiple.

Clase tiene los atributos capacidad, estado, horarioClase (tipo EstadoClase que es una enumeración), instructorTemporal. Unida con Servicio, Instructor y Cliente con agregación, la última siendo múltiple.

Sala tiene los atributos capacidad, aforo, costoMatricula, está unida a Servicio por agregación múltiple y por composición con Jornada, que tiene los atributos dia, horarioAtencion y está unida por agregación múltiple con Clase.

5.2.2 View

En la vista se encuentran las diferentes pantallas que se le mostrarán a la persona que esté utilizando el sistema. Son un total de 9, con 5 de ellas solo para el administrador, 1 que solo se muestra al cliente y 3 que pueden ver todos los usuarios.

Las 5 del administrador son Sala, Servicios, Instructores, Clientes; donde se hacen las operaciones de crear, modificar y eliminar de cada uno; y NuevoAdministrador.

La que solo se muestra al cliente es Pagos, donde se muestran los pagos pendientes que tenga el usuario y aquí mismo realiza el pago.

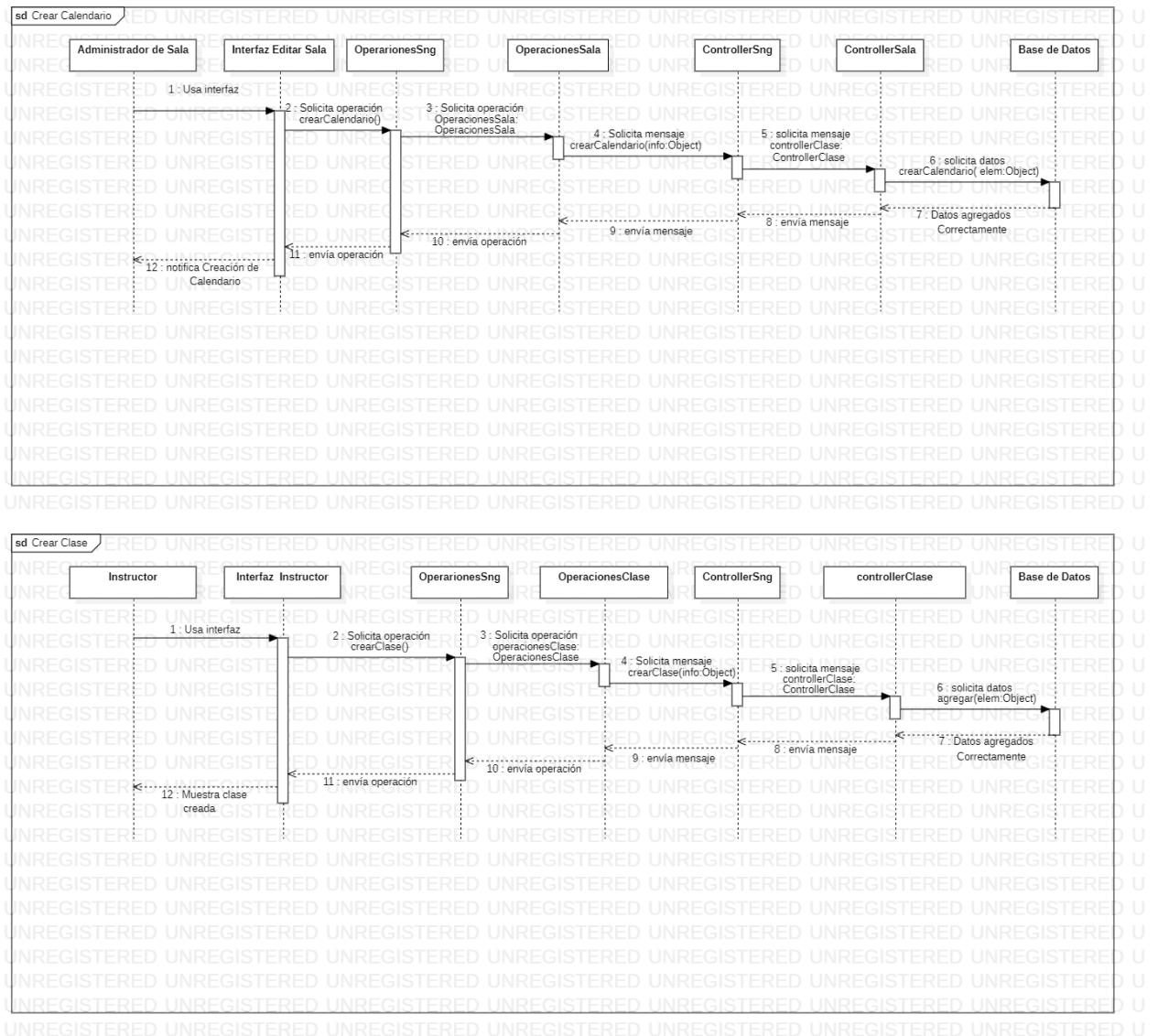
A las que todos pueden acceder son Perfil, que permite modificar los datos de la cuenta; Calendario e IniciarSesion-Registrarse, donde todos pueden iniciar sesión, pero solo los clientes se pueden registrar desde aquí.

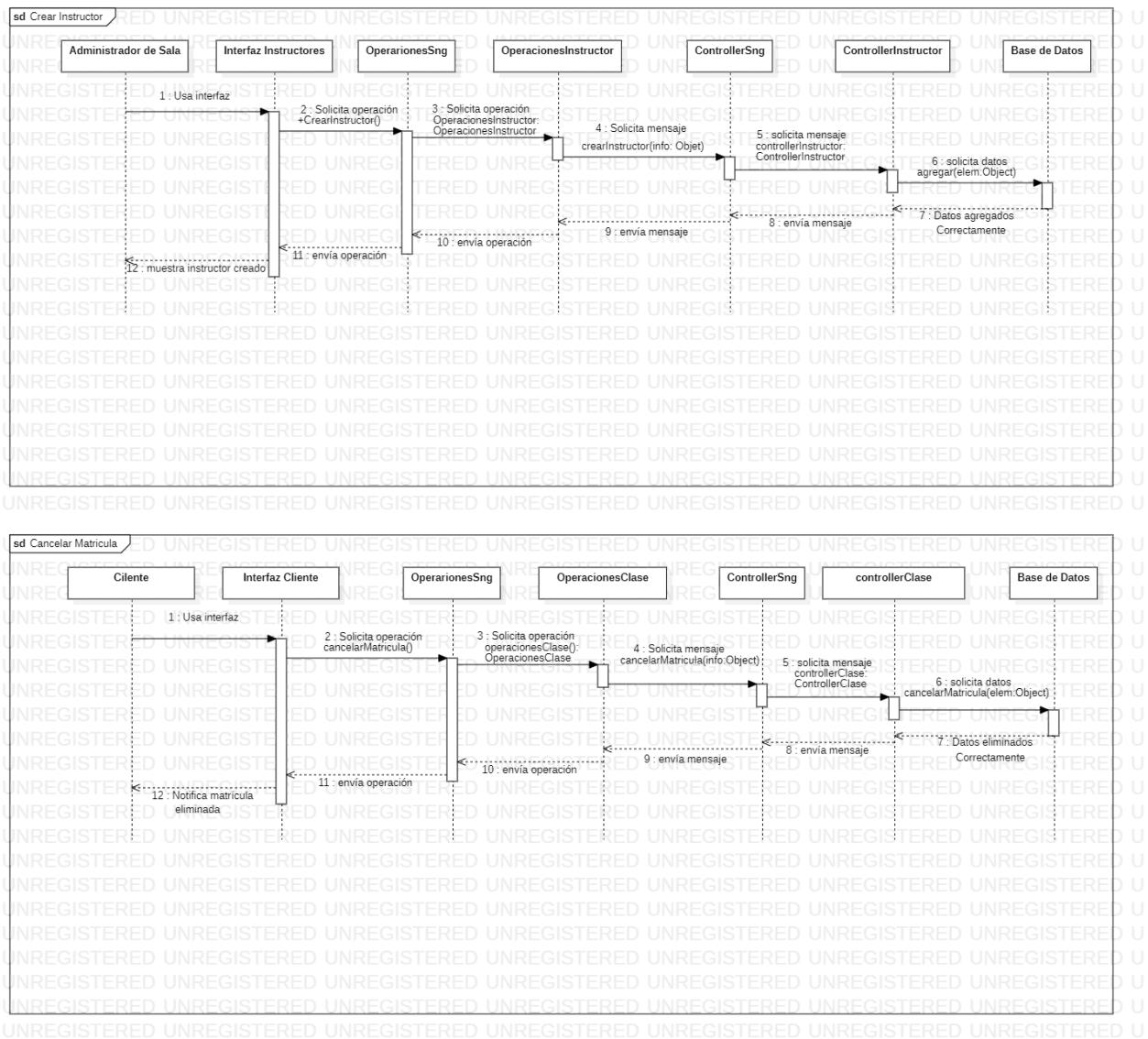
5.2.3 Controller

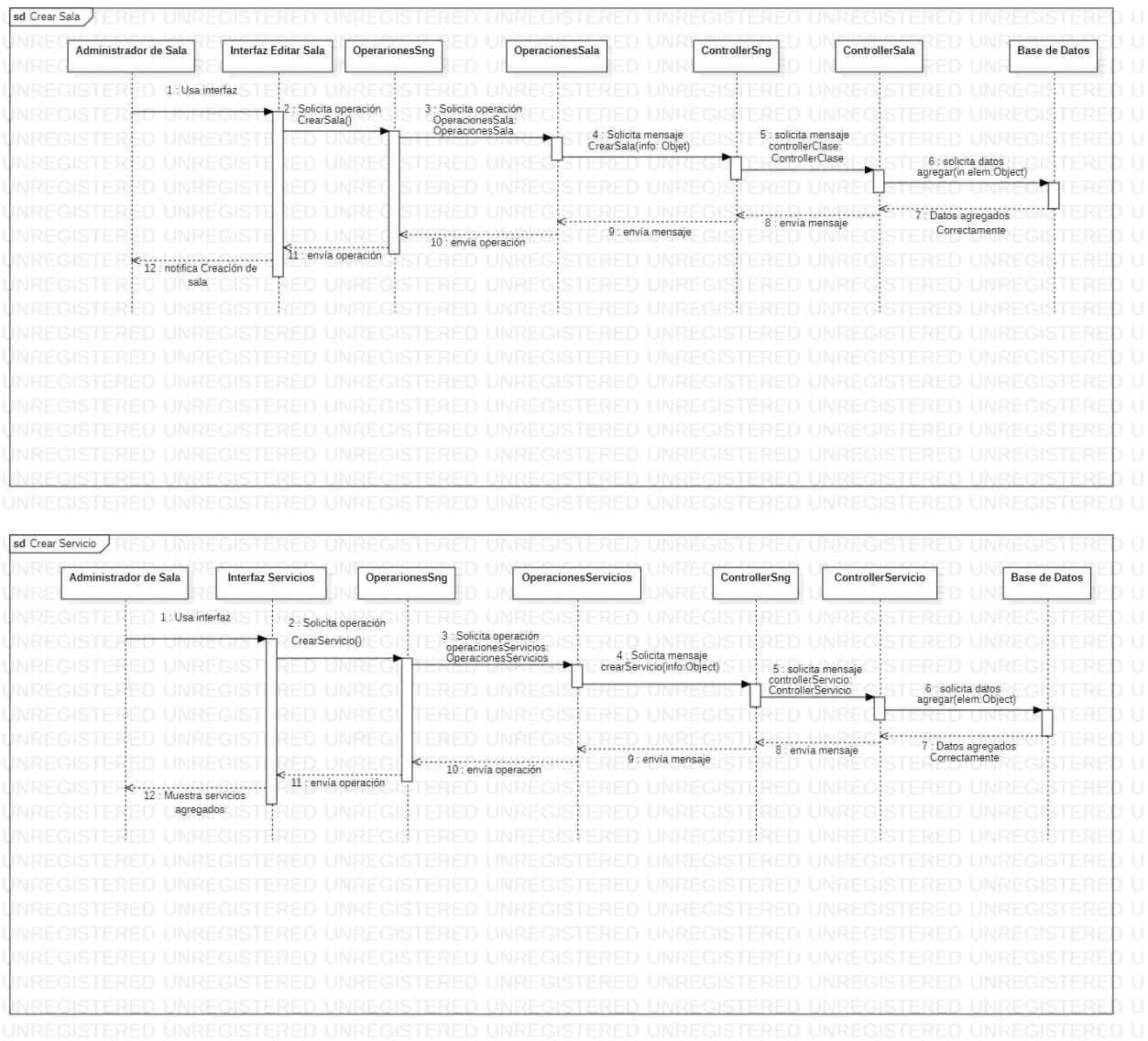
Por último el controller que se encarga de ser el intermediario entre la vista y el modelo, aquí se realizan todas las operaciones necesarias para que esta comunicación sea exitosa.

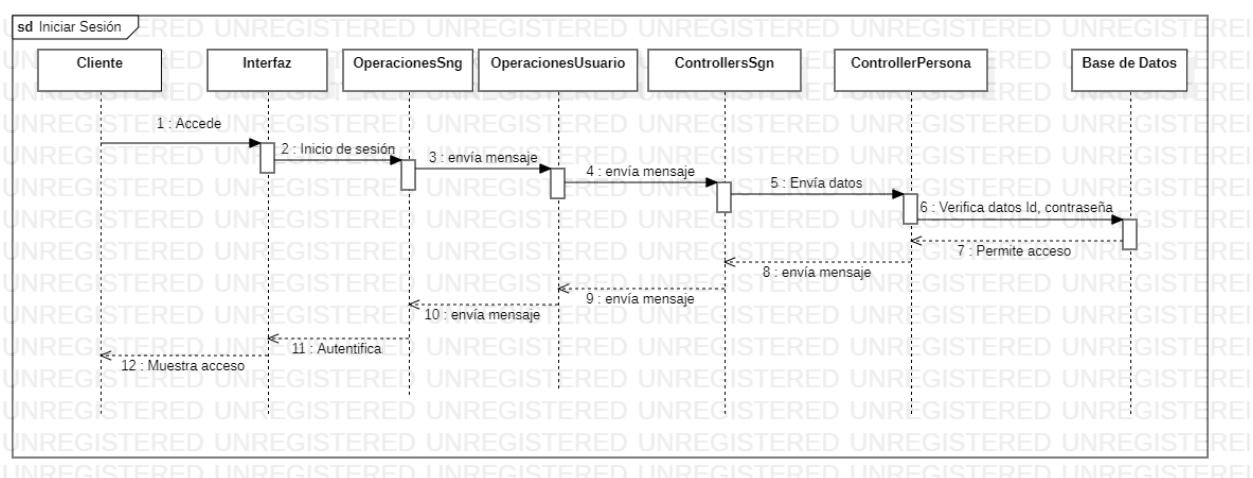
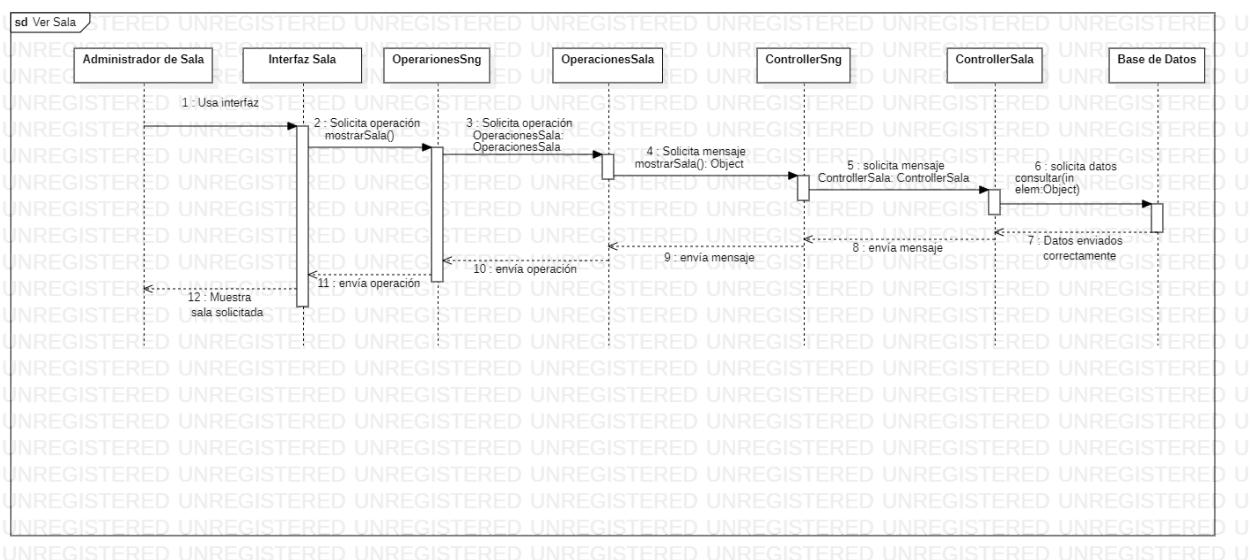
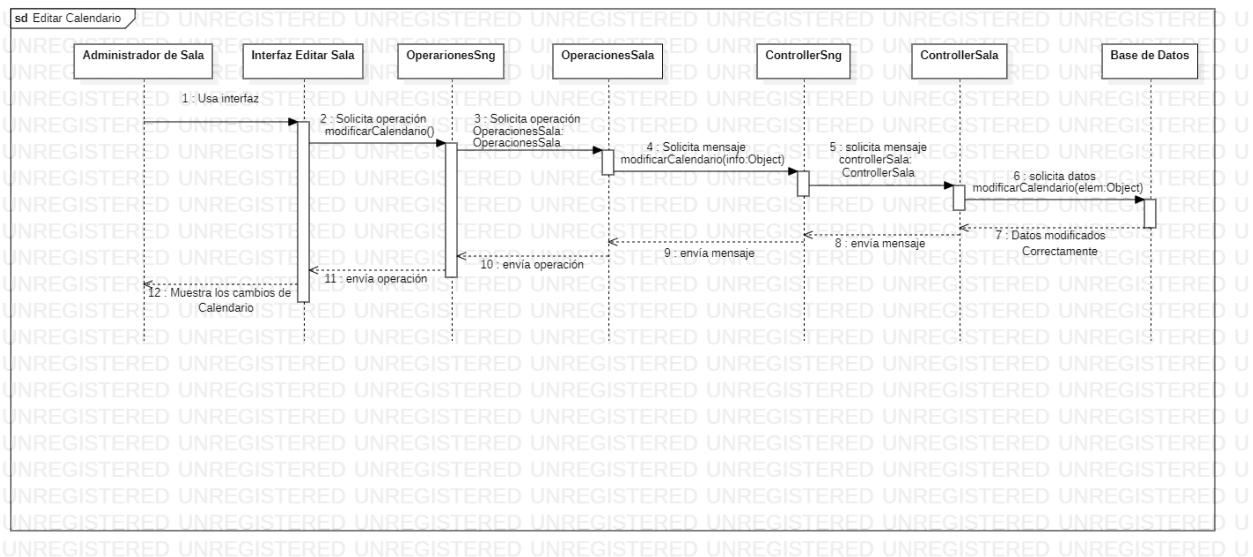
Tenemos dos Singletons uno para los controladores, que hacen las llamadas a la base de datos y conectan con la capa Model; y otro para conectar la capa View que designamos como operaciones.

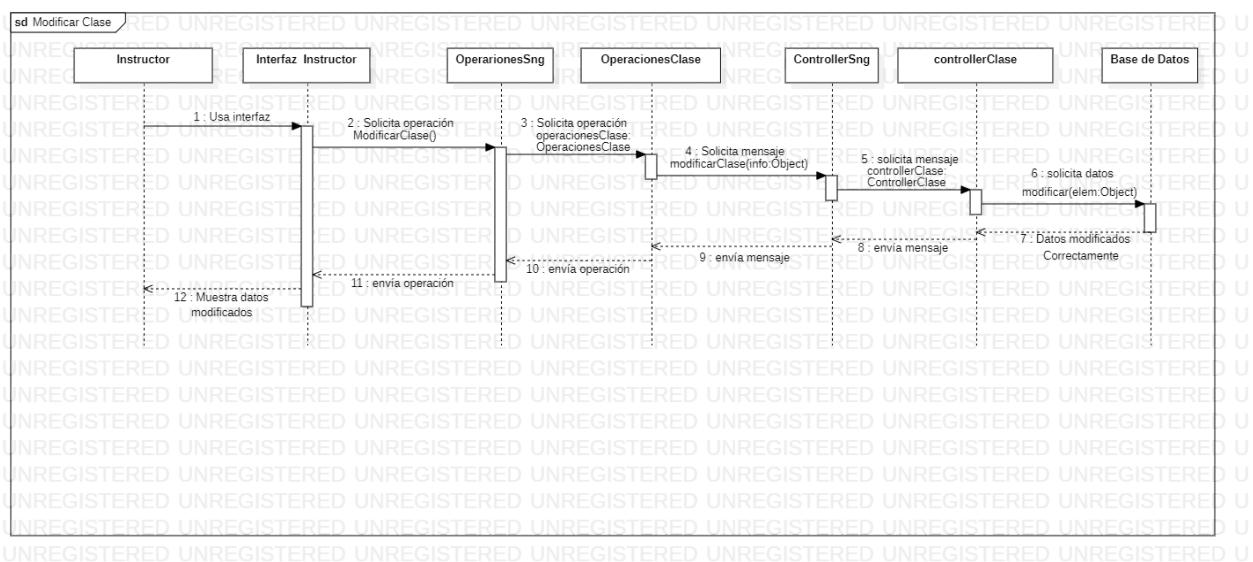
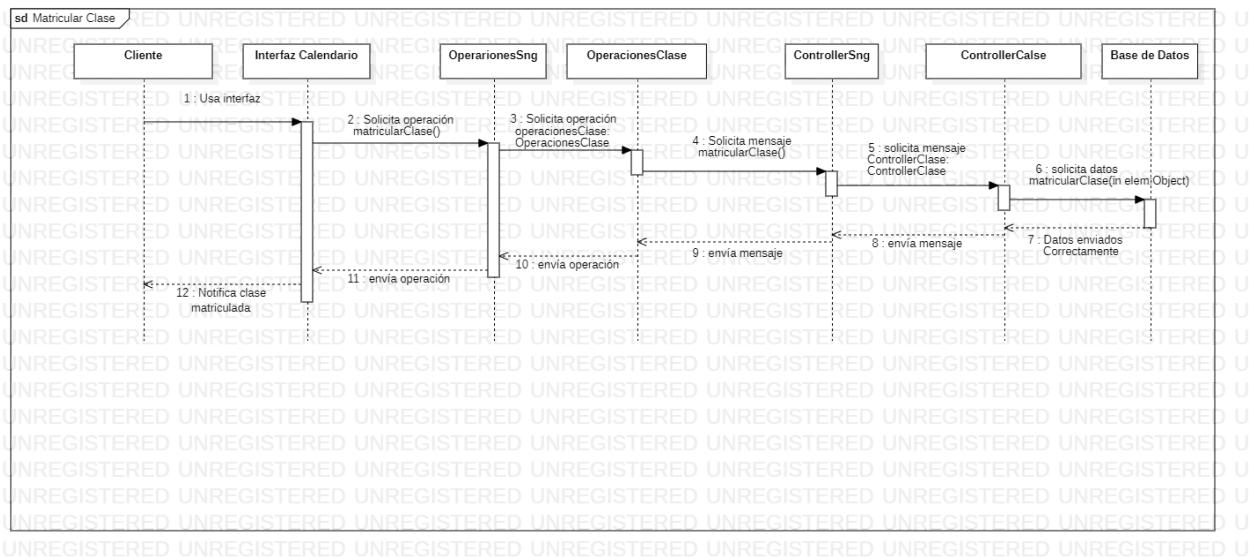
5.3 Realizaciones de Casos de Uso

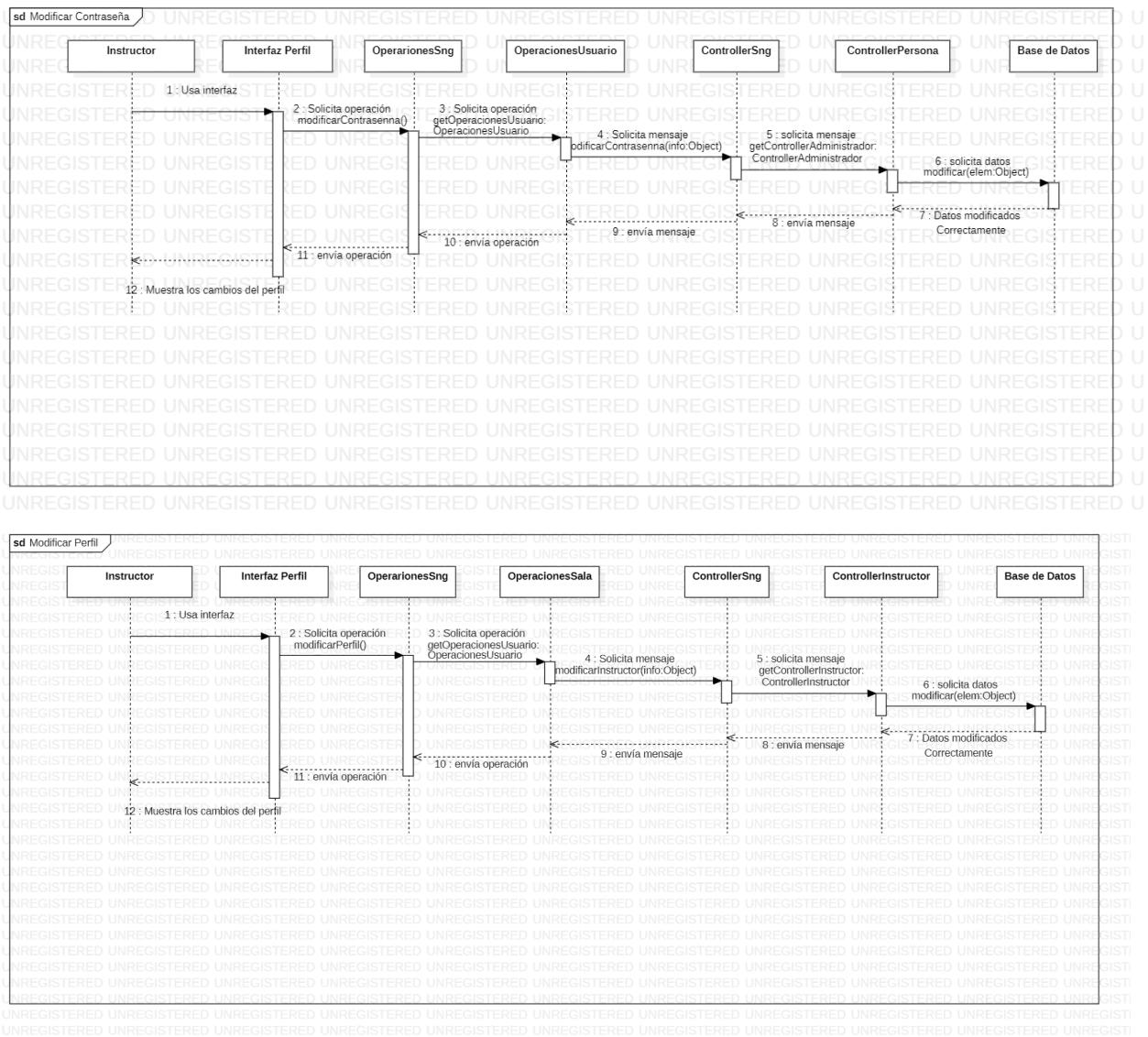


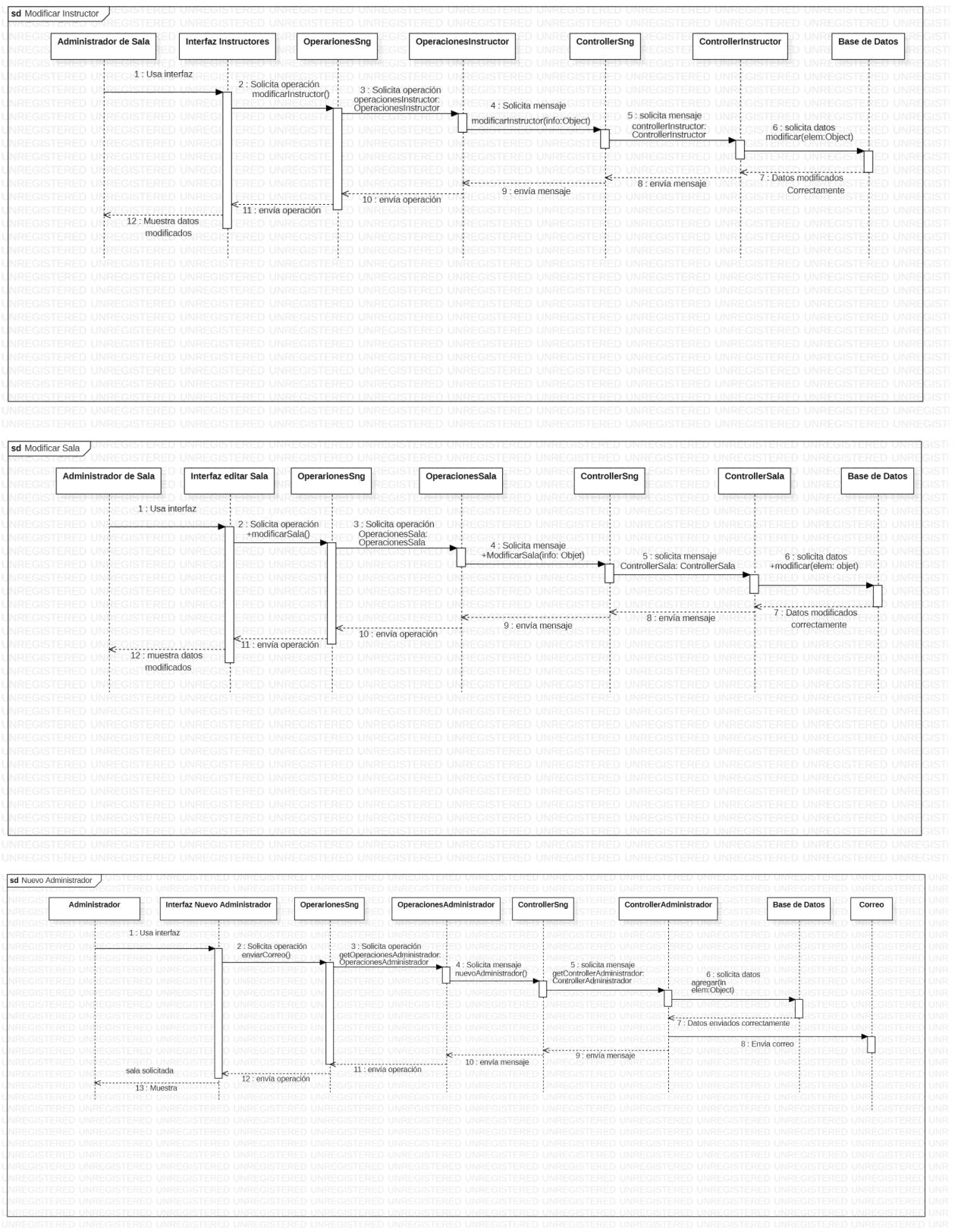


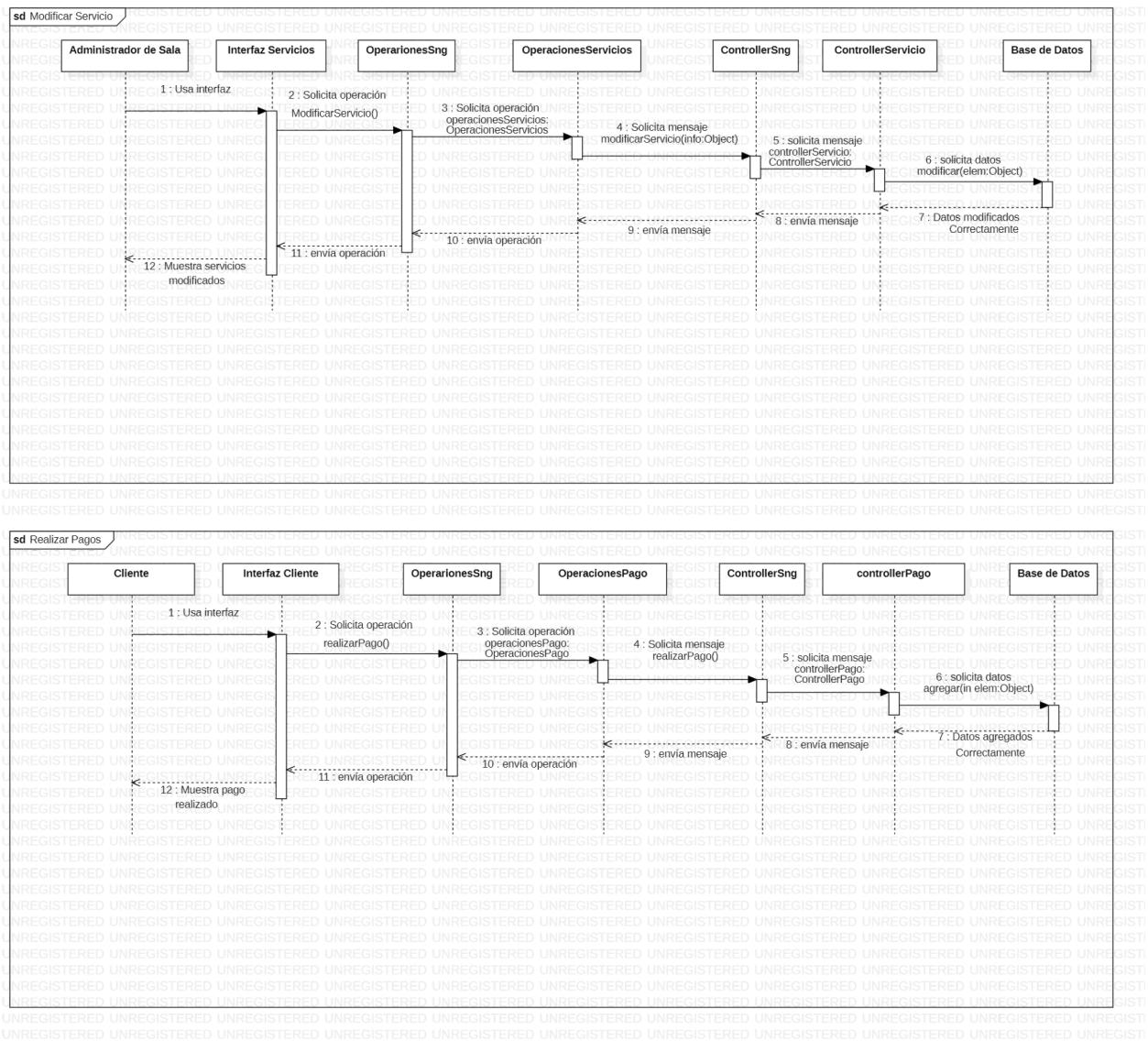


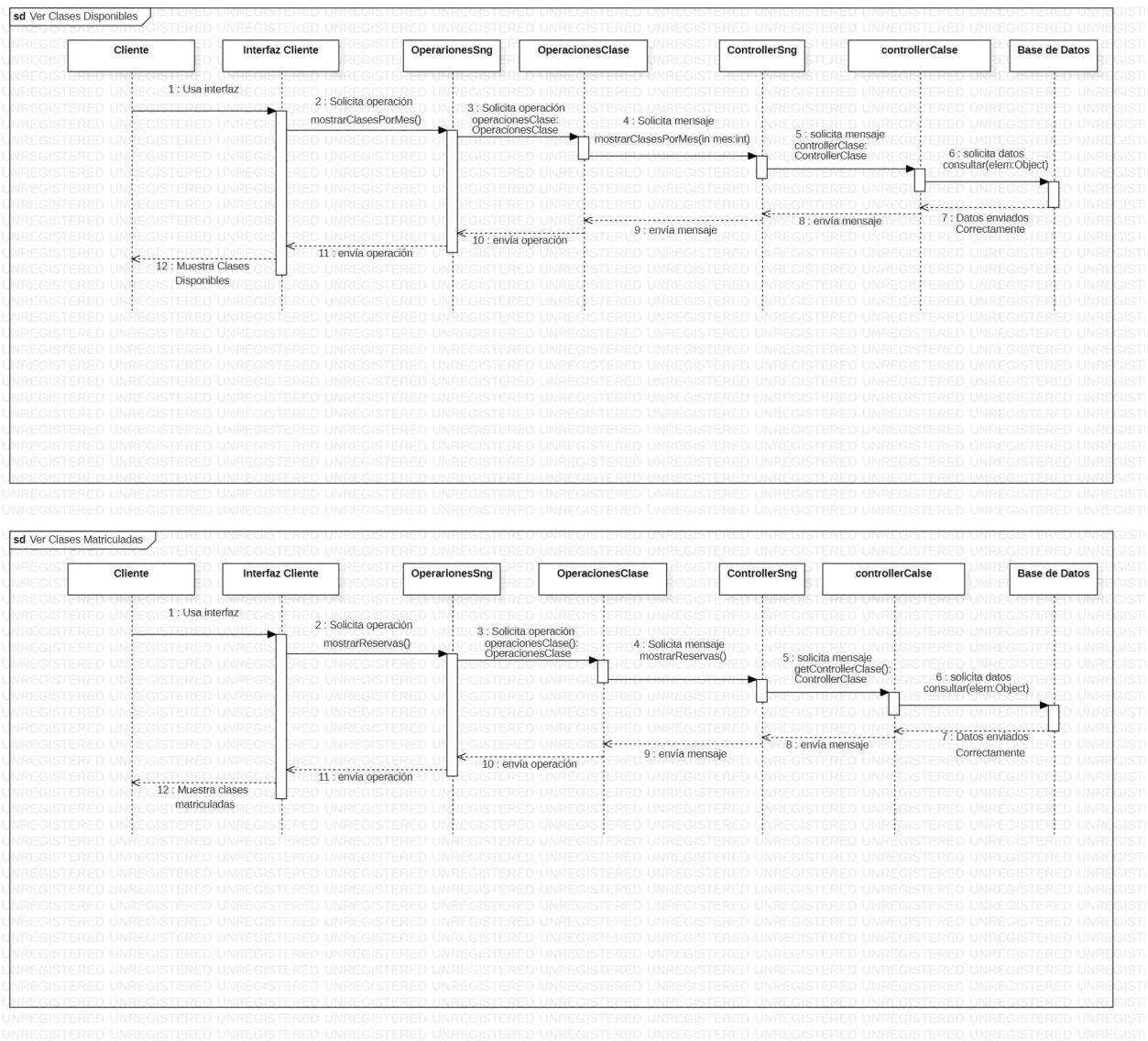


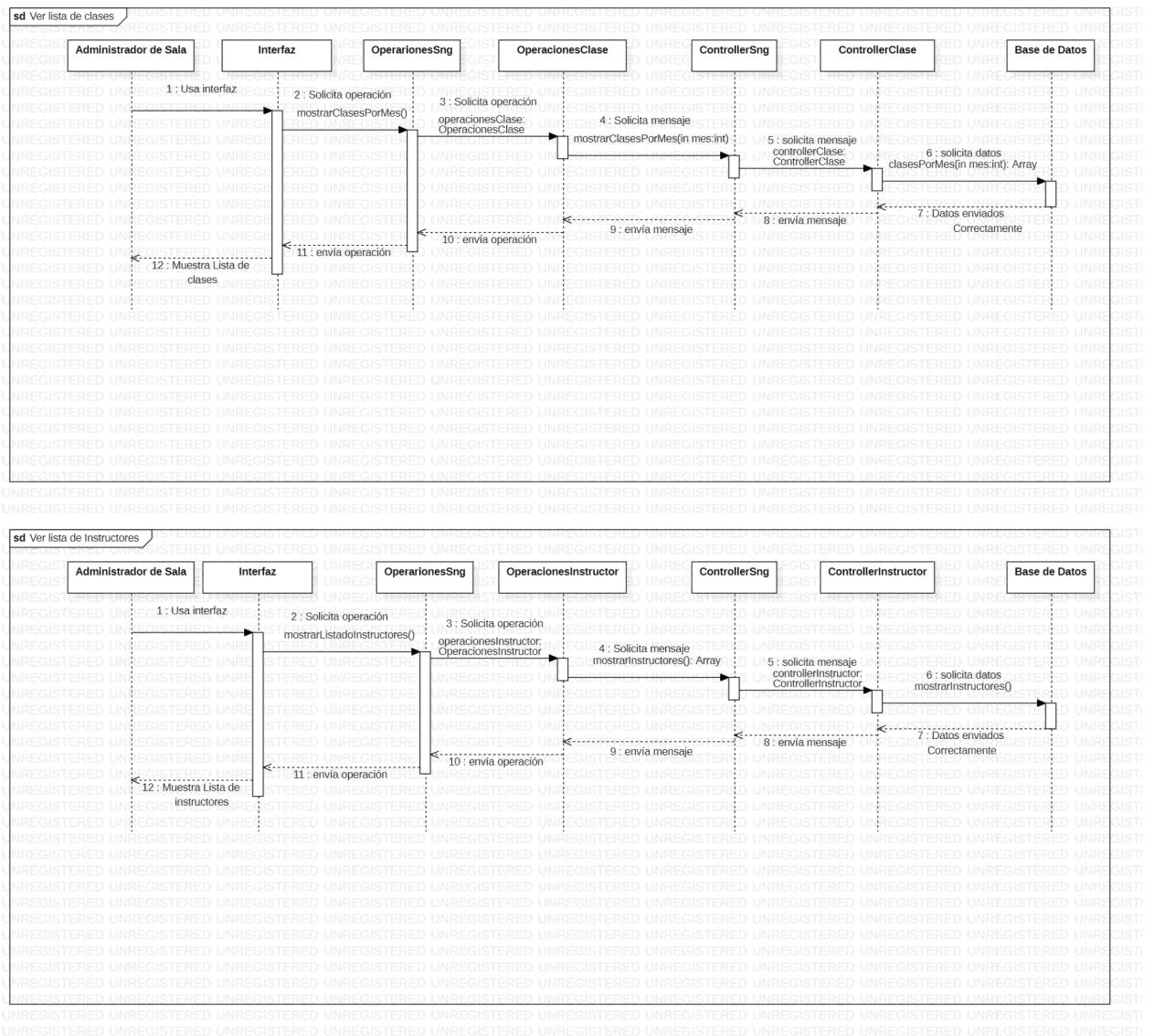


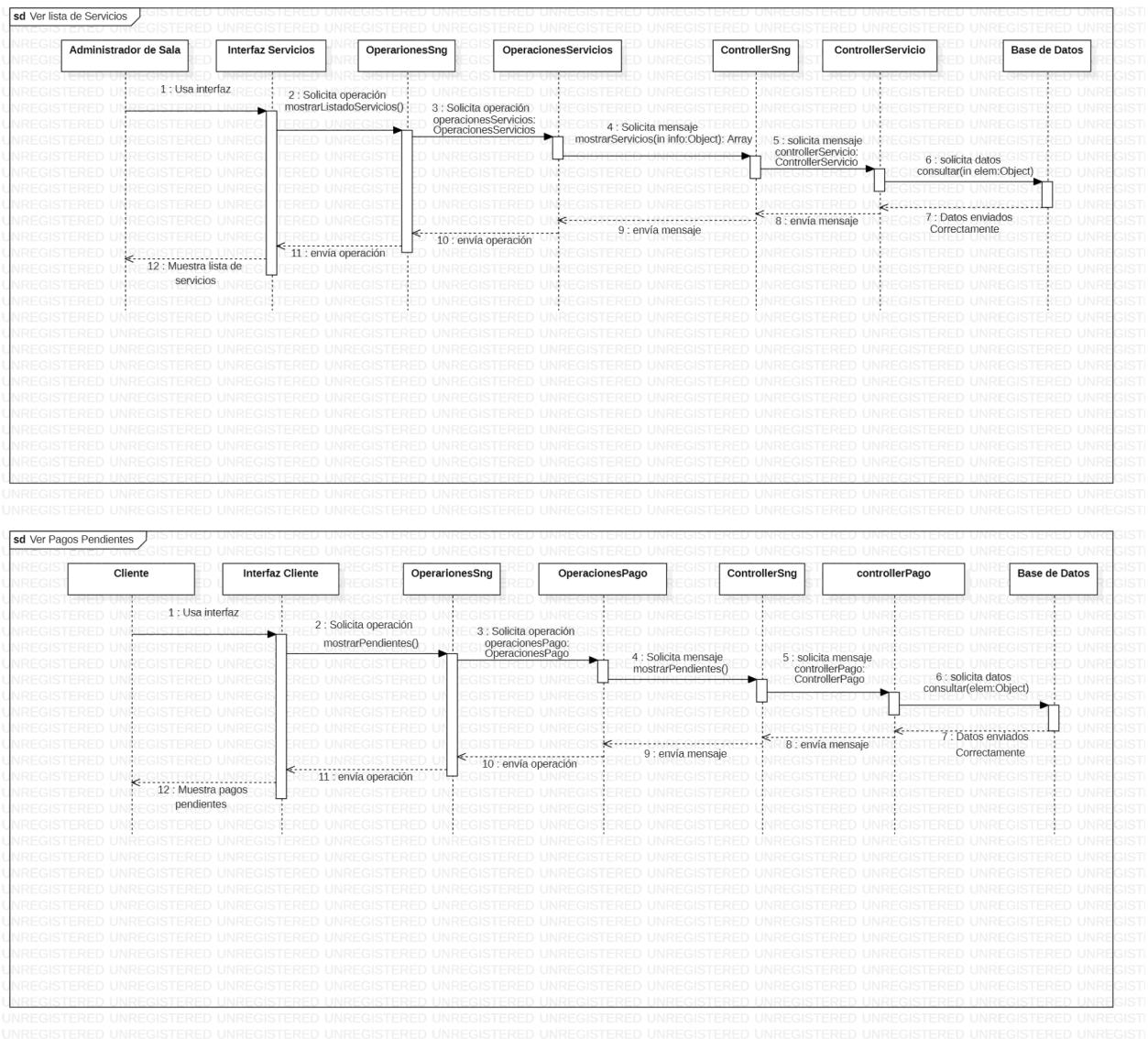


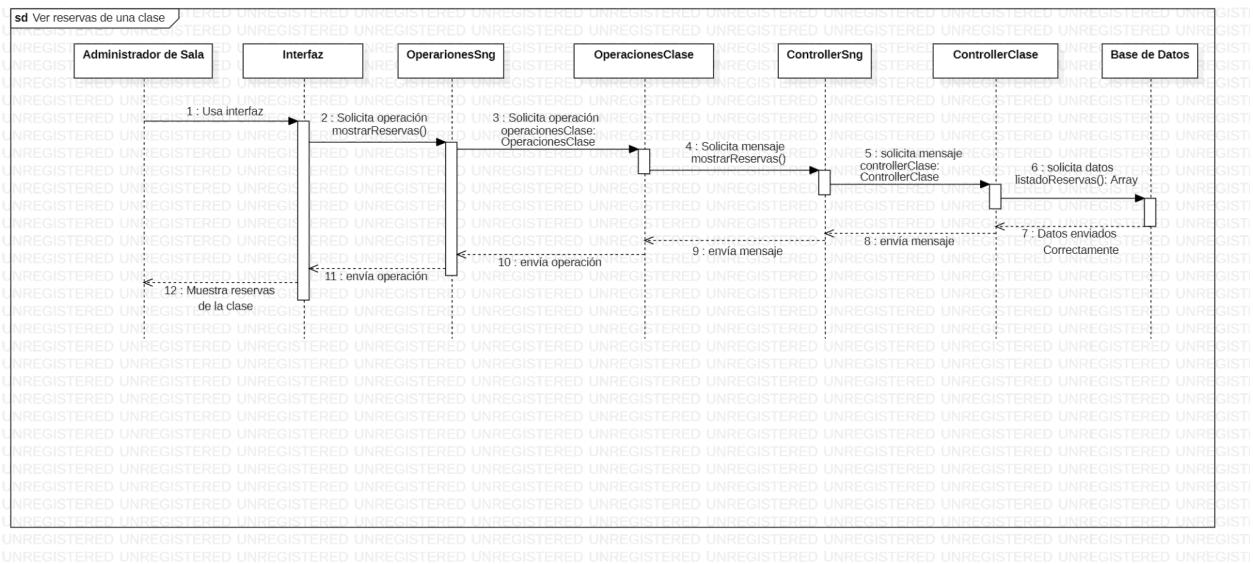




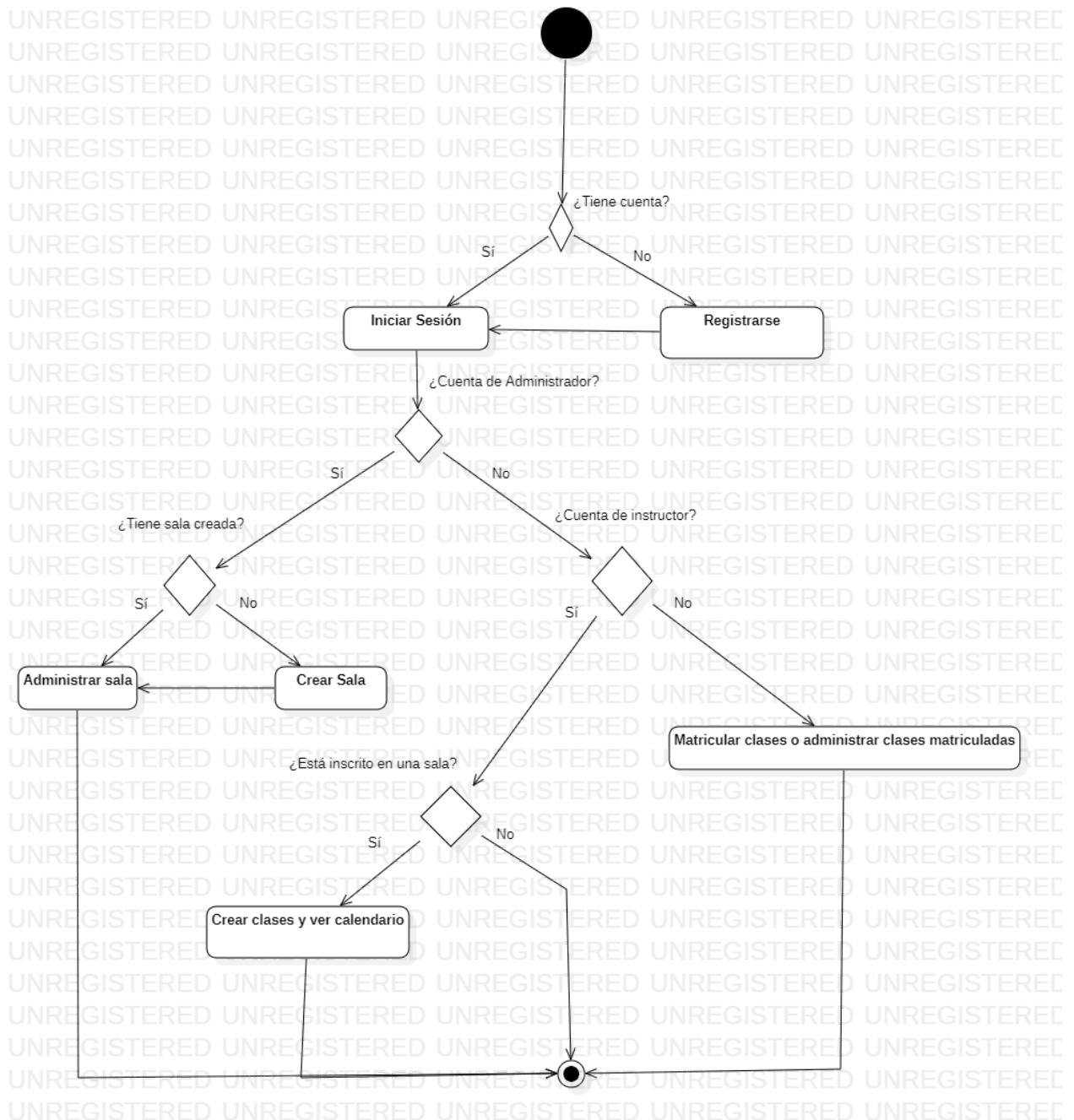




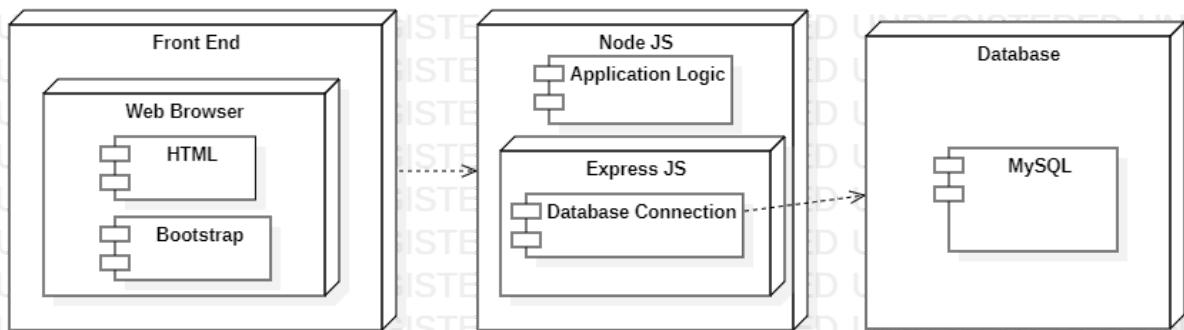




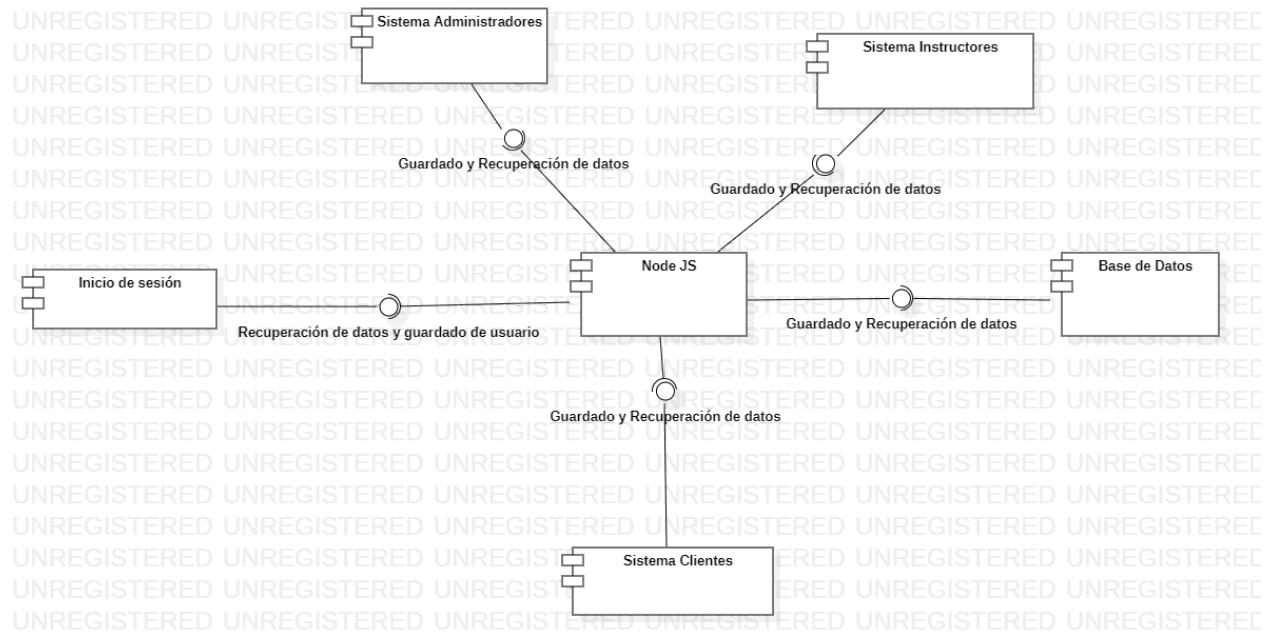
6. Vista de Procesos



7. Vista de Despliegue



8. Vista de Implementación



8.1 Descripción

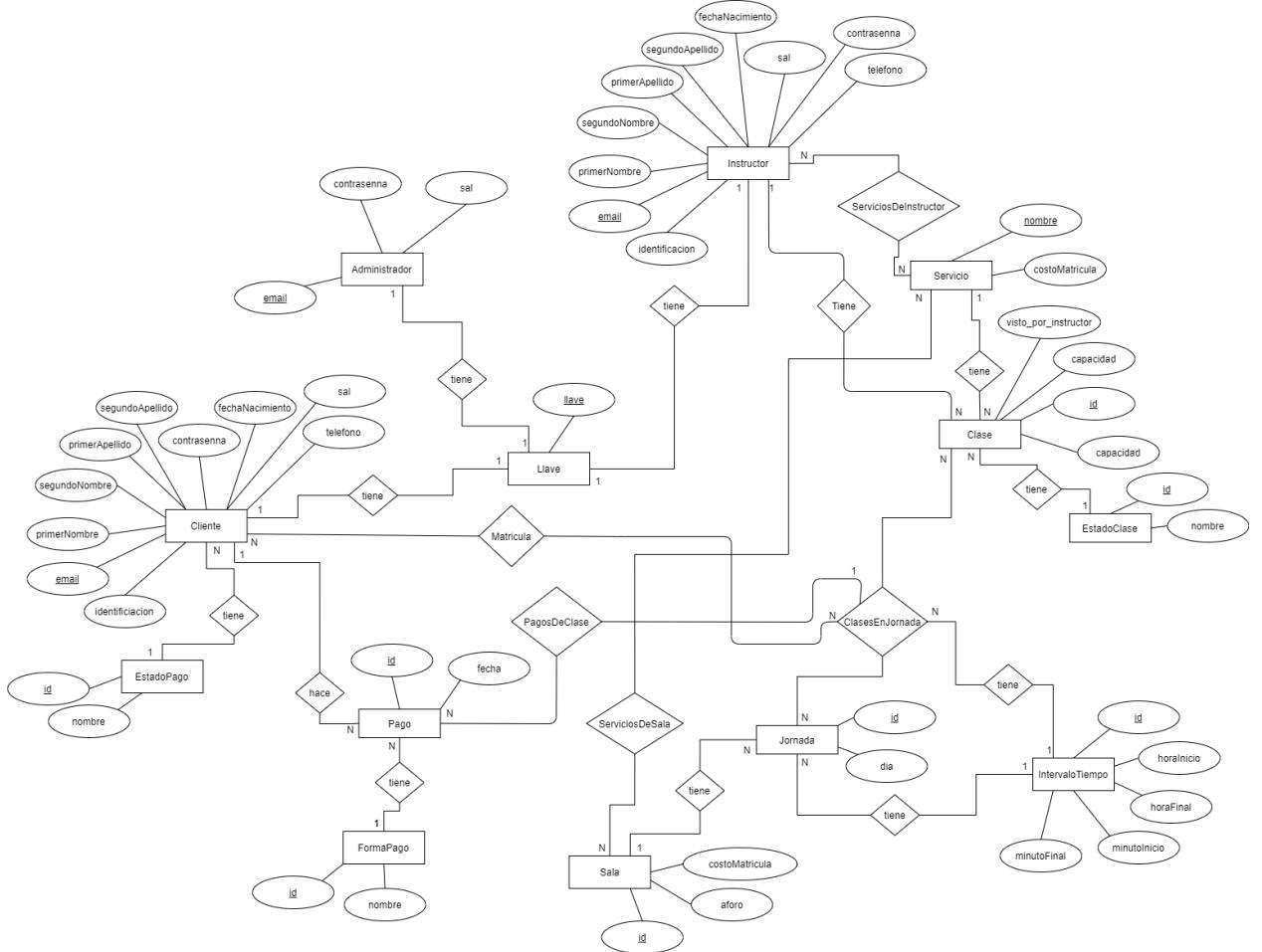
En el diagrama de componentes se pueden identificar específicamente 6 componentes, estos componentes se pueden dividir en interacción con el usuario e interacción con la base de datos y hay un intermedio entre estos dos. Los sistemas de administradores, clientes, instructores y el inicio de sesión equivalen a los sistemas que tienen interacción directa con algún usuario, el sistema de base de datos se encarga de guardar los datos y el sistema de Node JS es precisamente la conexión entre los sistemas de interacción con el usuario y la base de datos.

8.2 Capas

- **Inicio de sesión:** Esta capa incluye el proceso del usuario que desea registrarse en el sistema y el que desea iniciar sesión con alguna cuenta ya creada.
- **Sistema Administradores:** En esta capa se implementarían todos los casos de uso que corresponden al usuario Administrador de una Sala. Esta probablemente es la capa con mayores funcionalidades.
- **Sistema Instructores:** Aquí se incluyen los casos de uso del usuario Instructor, aquí cada instructor puede organizar sus clases.

- Sistema Clientes: En esta parte los clientes pueden ver las clases disponibles y matricular alguna clase, cancelar su matrícula, etc.
- Node JS: Esto es la conexión con la base de datos, se encarga de que todos los módulos anteriores, es decir, los de interacción con usuario, guarden sus cambios en la base de datos.
- Base de datos: Donde se guardarán todos los datos de usuarios, clases, salas, etc. Esta base de datos específicamente será desarrollada en MySQL.

9. Vista de Datos



10. Tamaño y Rendimiento

N/A

11. Calidad

Uno de los requerimientos es que este sistema tenga la propiedad de la extensibilidad, esto para que se le puedan agregar más atributos de una manera sencilla.