

**Diseño de**

**Software**

**Proyecto**

**Segundo Parcial**

**Integrantes:**

**David Roberto Mendoza Loor**

**Kevin Antonio Cevallos Pilay**

**Marlon Eddie Lindao Varas**

**Profesor:**

**David Alonso Jurado Mosquera**

**Paralelo:**

**1**

Contenido

[Propuesta 3](#_Toc31601003)

[Patrones de Diseño utilizados dentro del proyecto 3](#_Toc31601004)

[Diagrama casos de uso y especificaciones 4](#_Toc31601005)

[Diagrama de clases 10](#_Toc31601006)

[Diagramas de secuencias 11](#_Toc31601007)

[Diagrama normalizado 13](#_Toc31601008)

[Docker 14](#_Toc31601009)

[Reporte SonarQube 15](#_Toc31601010)

# Propuesta

Se diseñará un sistema automatizado con conexión a una base datos para un quiosco de una Inmobiliaria, el cual permita a sus clientes, diseñar sus casas, ver el costo de sus bosquejos, y ver el precio de los detalles de estas viviendas. Además, el sistema permitirá a los vendedores, poder consultar datos de los clientes de la empresa, y de los diseños realizados.

# Patrones de Diseño utilizados dentro del proyecto

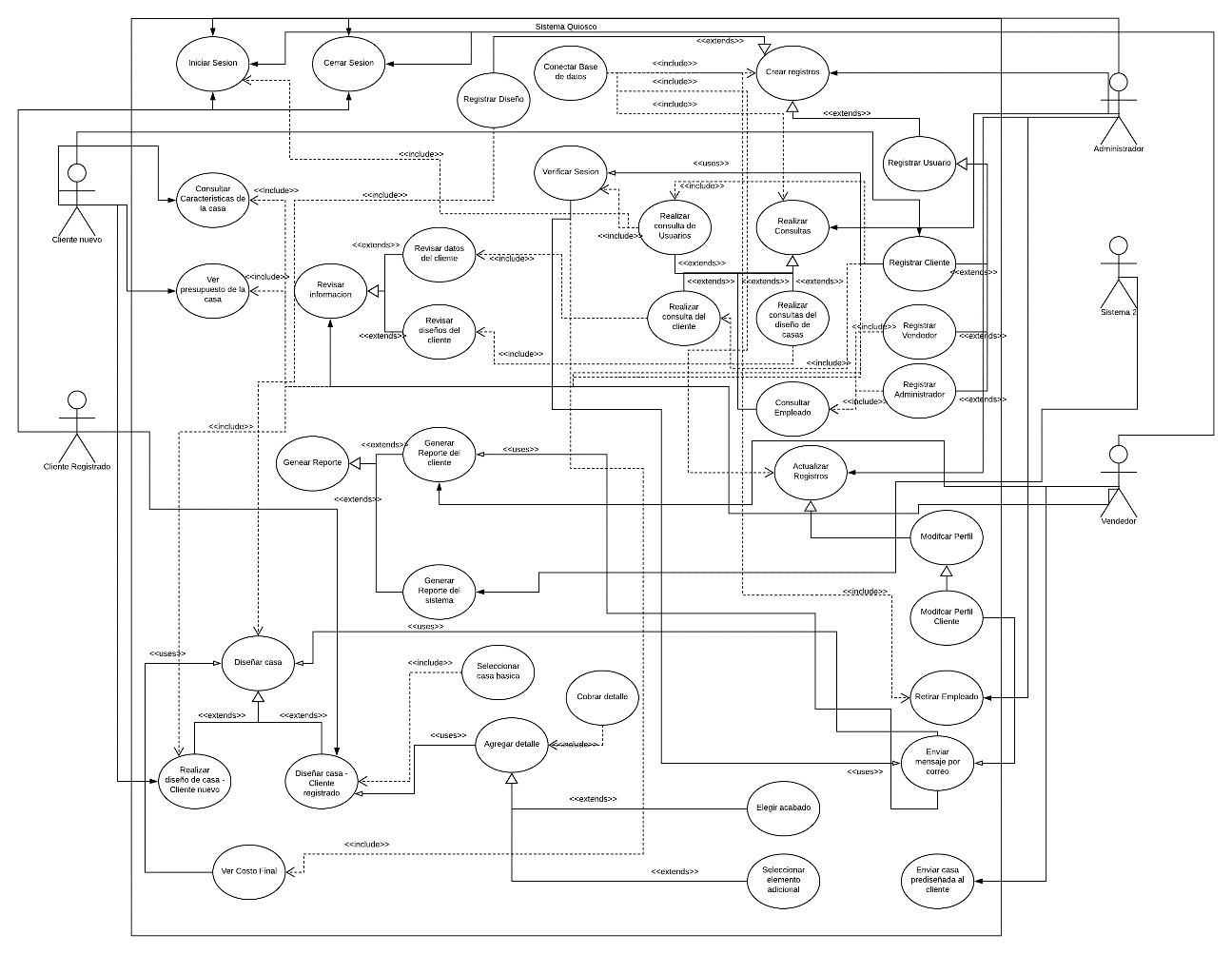
**Singleton:** Se utilizo el patrón Singleton para crear una clase de conexión a la base de datos que contenga una sola instancia, y por medio de esta sola instancia haremos las solicitudes y peticiones necesaria a la base de datos a lo largo del resto de clases.

**Builder:** Las especificaciones del proyecto nos indica que existen diseños de casas básicas que tienen un patrón de construcción igual para cada una, estas poseen muchos pasos para su construcción e instanciarlas directamente nos llevaran a tener un constructor demasiado largo, para ello usamos el patrón Builder para que la clase que necesite la instancia de la casa básica se abstraiga totalmente de la construcción de la casa básica que la construcción es extensa.

**Decorator:** Los usuarios una vez hayan elegido un modelo de casa básica tienen la potestad de añadir elementos adicionales en diseño, estos cambios se darán en tiempo de ejecución por lo que el patrón que nos permite realizar dichos cambios es Decorator, dependiendo de los elementos adicionales el diseño generado por el cliente variará en su precio.

**Bridge:** Usaremos este patrón para manejar el envió de emails y generar reportes en formato pdf. Enviar y generar pdf podría estar en una sola clase, pero podría causar que esta clase sea muy larga y que tenga más de una responsabilidad, por lo que usando el patrón Bridge podemos separar estas responsabilidades y en el caso de que en un futuro queramos generar emails con archivos en otros formatos a parte de pdf será más fácil integrarlo a nuestro proyecto.

# Diagrama casos de uso y especificaciones



|  |  |
| --- | --- |
| **Caso de Uso** | |
| **ID** | **CDU- P 01** |
| **Titulo** | **Diseñar Casa – Cliente Registrado** |
| **Autor** | Marlon Eddie Lindao Varas |
| **Actor principal** | Cliente Registrado |
| **Actor secundario** |  |
| **Usa** | Seleccionar casa básica, Agregar Detalle, Ver Costo Adicional, Enviar mensaje por correo, Registrar Diseño. |
| **Extiende** | Diseñar casa |
| **Descripción** | Cliente registrado: quiere diseñar su casa, por lo que deberá seleccionar una casa modelo, y agregarle detalles en caso de quererlo. |
| **Precondición** | * El cliente debe estar registrado en el sistema. * El cliente debe haber iniciado sesión en el sistema. |
| **Secuencia normal** | 1. El sistema le muestra al cliente las casas básicas que ofrece el sistema. 2. El sistema le muestra al usuario, un mensaje al cliente, diciendo que seleccione una de las casas básicas mostradas. 3. El cliente selecciona la casa, y el sistema le muestra la opción para agregar detalles a la casa. 4. El cliente escoge los detalles que desea para su casa, y el usuario presiona en finalizar. 5. El sistema guarda el diseño del cliente en la base de datos. 6. El sistema le muestra la casa con los detalles, y le presenta la opción para ver el costo final de la casa y enviar el diseño por email.    * 1. El cliente presiona en la opción para ver el costo final, y el sistema calcula su valor, con el precio de la casa básica, y con los precios de cada detalle agregado a la casa.      2. El sistema le muestra el precio final de la casa al cliente.   7.2.1 El cliente selecciona enviar el diseño por email, y el sistema procesa el diseño en un archivo, y se lo envía a la dirección registrado en su perfil. |
| **Secuencia alternativa** | 7.2.1 El cliente no tiene registrado su email  7.2.1.1 El sistema le indica al cliente que no tiene registrado su e-mail  7.2.1.2 El sistema le muestra la pantalla para modificar su registro  7.2.1.3 El cliente registra su email, y presiona en guardar, en donde el sistema procede a actualizar su registro en la base de datos. |
| **Postcondición** | El diseño de la casa ha sido guardado en la base de datos correctamente. |
|  |  |

|  |  |
| --- | --- |
| **Caso de Uso** | |
| **ID** | **CDU- P 02** |
| **Titulo** | **Revisar Datos del Cliente** |
| **Autor** | Marlon Eddie Lindao Varas |
| **Actor principal** | Vendedor |
| **Actor secundario** |  |
| **Usa** | Realizar consulta del cliente |
| **Extiende** | Revisar Información. |
| **Descripción** | Vendedor: quiere revisar los datos del cliente, por lo que debe digitar en el sistema, la cedula o pasaporte del cliente para realizar la consulta en el sistema. |
| **Precondición** | * El cliente debe estar registrado en la base de datos. * El vendedor debe haber iniciado sesión en el sistema. |
| **Secuencia normal** | 1. El sistema se conecta a la base de datos. 2. El sistema le pide al vendedor que ingrese la cedula o pasaporte del cliente. 3. El sistema valida si la cedula o pasaporte está ingresada correctamente. 4. El sistema realiza una consulta en la base de datos, para validar si el cliente se encuentra en la base. 5. El sistema recibe la respuesta de la base de datos, con el registro del cliente. 6. El sistema le muestra al vendedor, los datos del cliente. |
| **Secuencia alternativa** | 3. La cedula o pasaporte no ha sido ingresado correctamente.  3.1 El sistema muestra una notificación al vendedor que la cedula o pasaporte no es válido.  5. El cliente no se encuentra registrado en la base de datos.  5.1 El sistema le muestra una notificación al vendedor de que el cliente no se encuentra registrado en el sistema.  5.2 El sistema regresa a la ventana principal del sistema para el vendedor. |
| **Postcondición** | El sistema le presenta los datos del cliente al vendedor |

|  |  |
| --- | --- |
| **Caso de Uso** | |
| **ID** | **CDU- P 03** |
| **Titulo** | **Revisar Diseños del Cliente** |
| **Autor** | Marlon Eddie Lindao Varas |
| **Actor principal** | Vendedor |
| **Actor secundario** |  |
| **Usa** | Realizar consultas del diseño de casas |
| **Extiende** | Revisar Información. |
| **Descripción** | Vendedor: quiere revisar los diseños realizados por el cliente, por lo que deberá consultarlos en la base de datos. |
| **Precondición** | * El cliente debe estar registrado en la base de datos. * El vendedor debe haber iniciado sesión en el sistema. |
| **Secuencia normal** | 1. El sistema se conecta a la base de datos. 2. El sistema le pide al vendedor que ingrese la cedula o pasaporte del cliente. 3. El sistema valida si la cedula o pasaporte está ingresada correctamente. 4. El sistema realiza una consulta en la base de datos, para validar si el cliente se encuentra en la base. 5. El sistema recibe la respuesta de la base de datos, con el registro del cliente. 6. Una vez validado al cliente, el sistema hace una consulta a la base de datos, para tener los diseños realizados por el cliente. 7. El sistema recibe los diseños del cliente de la base de datos. 8. El sistema le muestra al vendedor, los diseños realizados por el cliente. |
| **Secuencia alternativa** | 3. La cedula o pasaporte no ha sido ingresado correctamente.  3.1 El sistema muestra una notificación al vendedor que la cedula o pasaporte no es válido.  5. El cliente no se encuentra registrado en la base de datos.  5.1 El sistema le muestra una notificación al vendedor de que el cliente no se encuentra registrado en el sistema.  5.2 El sistema regresa a la ventana principal del sistema para el vendedor.  7.El cliente no tiene diseños realizados.  7.1 El sistema le muestra una notificación al vendedor, indicándole que el cliente no tiene diseños presentes en la base de datos. |
| **Postcondición** | Los diseños del cliente son presentados al vendedor en pantalla por el sistema. |

|  |  |
| --- | --- |
| **Caso de Uso** | |
| **ID** | **CDU- P 04** |
| **Titulo** | **Registrar Vendedor** |
| **Autor** | Marlon Eddie Lindao Varas |
| **Actor principal** | Administrador |
| **Actor secundario** |  |
| **Usa** | Consultar Empleado |
| **Extiende** | Registrar Usuario |
| **Descripción** | Administrador: quiere crear una cuenta para un vendedor nuevo al sistema, por lo que deberá registrar sus datos en la base de datos. |
| **Precondición** | El vendedor no debe tener una cuenta ya creada. |
| **Secuencia normal** | 1. El sistema se conecta a la base de datos. 2. El sistema le pide al administrador que ingrese los datos del vendedor. 3. El sistema valida que los datos se encuentren correctamente digitados. 4. El sistema realiza una consulta a la base de datos con la cedula o pasaporte del vendedor. 5. El sistema recibe la respuesta de la base de datos, diciendo que el vendedor no se encuentra registrado. 6. El sistema procede a crear la cuenta del vendedor, creando su registro en la base de datos. 7. El sistema le muestra una notificación al administrador, que la cuenta ha sido creado correctamente. |
| **Secuencia alternativa** | 3. Los datos no tienen valores válidos.  3.1 El sistema le muestra al administrador el o los campos que están incorrectos.  3.2 El administrador debe corregir los campos señalados por el sistema.  5. El vendedor ya se encuentra registrado en la base de datos  5.1. El sistema le muestra una notificación al administrador de que el vendedor ya se encuentra en la base de datos. |
| **Postcondición** | La cuenta del vendedor ha sido creada correctamente. |

|  |  |
| --- | --- |
| **Caso de Uso** | |
| **ID** | **CDU-P05** |
| **Titulo** | **Iniciar Sesión** |
| **Autor** | Marlon Eddie Lindao Varas |
| **Actor principal** | Cliente Registrado, Vendedor, Administrador |
| **Actor secundario** |  |
| **Usa** | Realizar consulta de Usuarios |
| **Extiende** |  |
| **Descripción** | Cliente Registrado, Vendedor, Administrador: quiere ingresar al sistema, usando sus credenciales. |
| **Precondición** | Los usuarios deben estar registrados en la base de datos. |
| **Secuencia normal** | 1. El sistema se conecta a la base de datos. 2. El sistema le pide al usuario (Cliente Registrado, Vendedor o Administrador) que ingrese su usuario y contraseña para iniciar sesión. 3. El sistema realiza una consulta a la base de datos con los datos ingresados. 4. El sistema recibe la respuesta de la base de datos. 5. El sistema le muestra al usuario, la pantalla principal con las opciones, según su perfil. |
| **Secuencia alternativa** | 4. El usuario o contraseña son inválidos.  4.1 El sistema muestra una notificación al usuario que su usuario o contraseña no son válidos.  4.2 El sistema le vuelve a mostrar la pantalla de inicio de sesión.  4.3 Si el usuario es un cliente nuevo, debe seleccionar la opción “Crear cuenta”.  4.4 Si el usuario es un nuevo empleado, debe pedir a un administrador que lo registre en el sistema |
| **Postcondición** | El usuario ha ingresado al sistema de forma correcta |

# Diagrama de clases

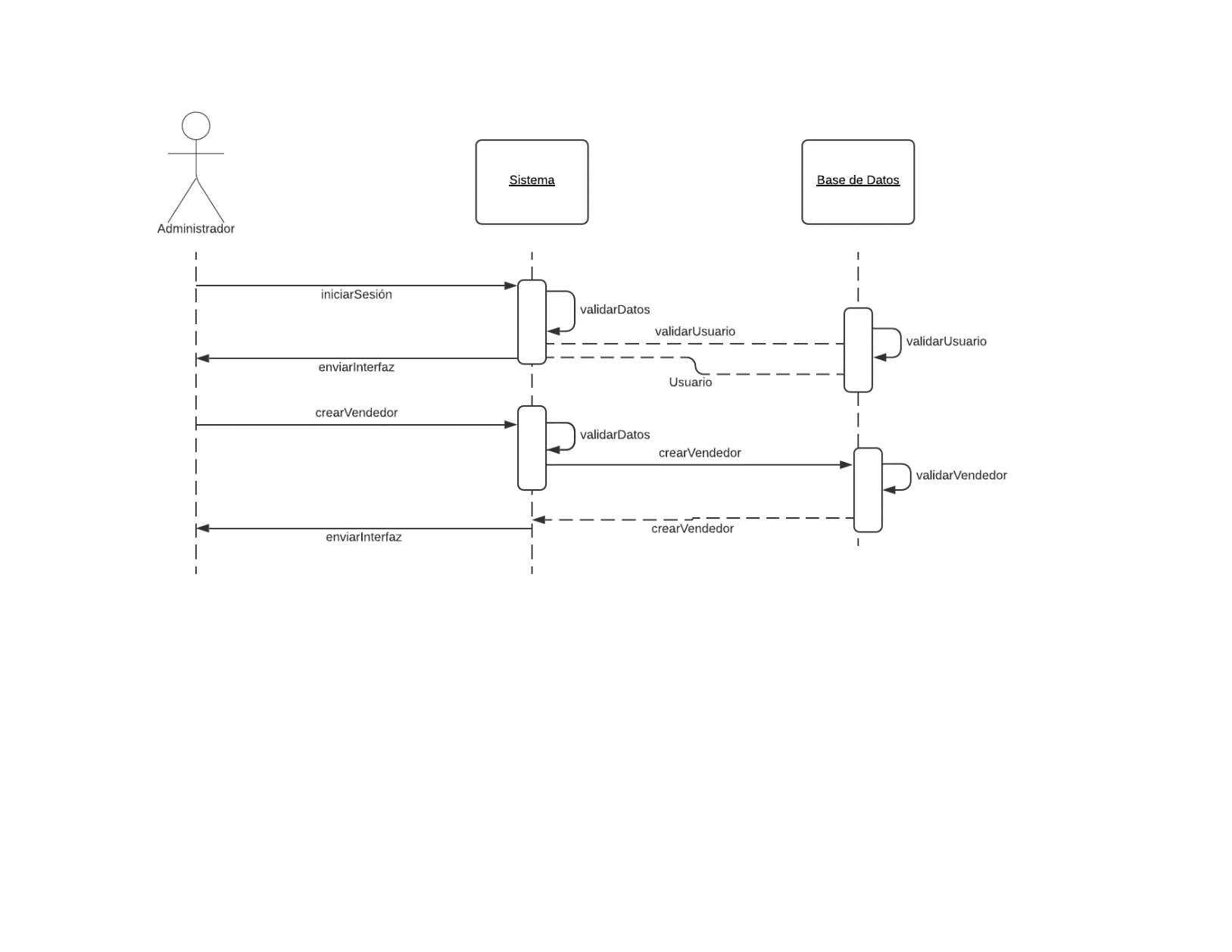
Imagen que contiene texto

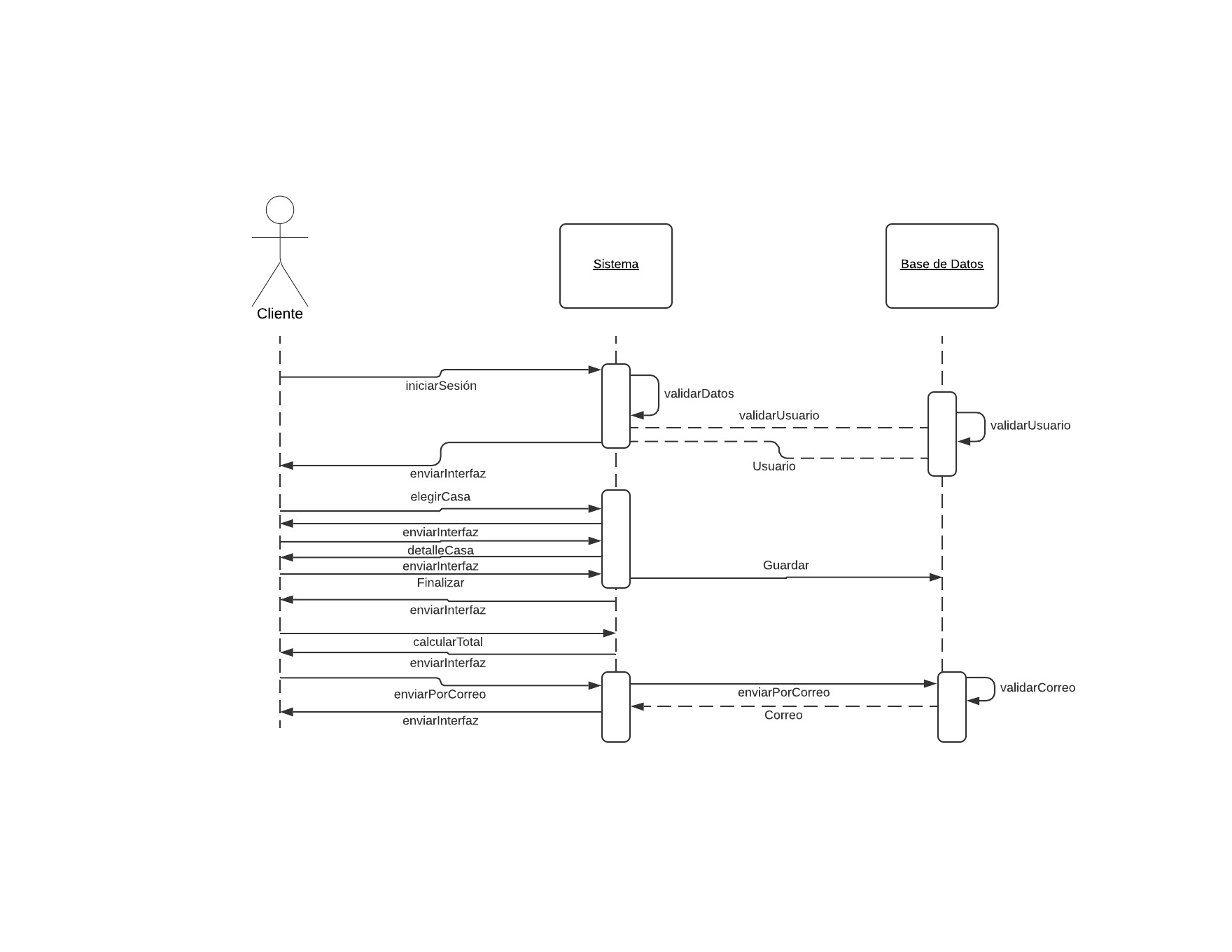
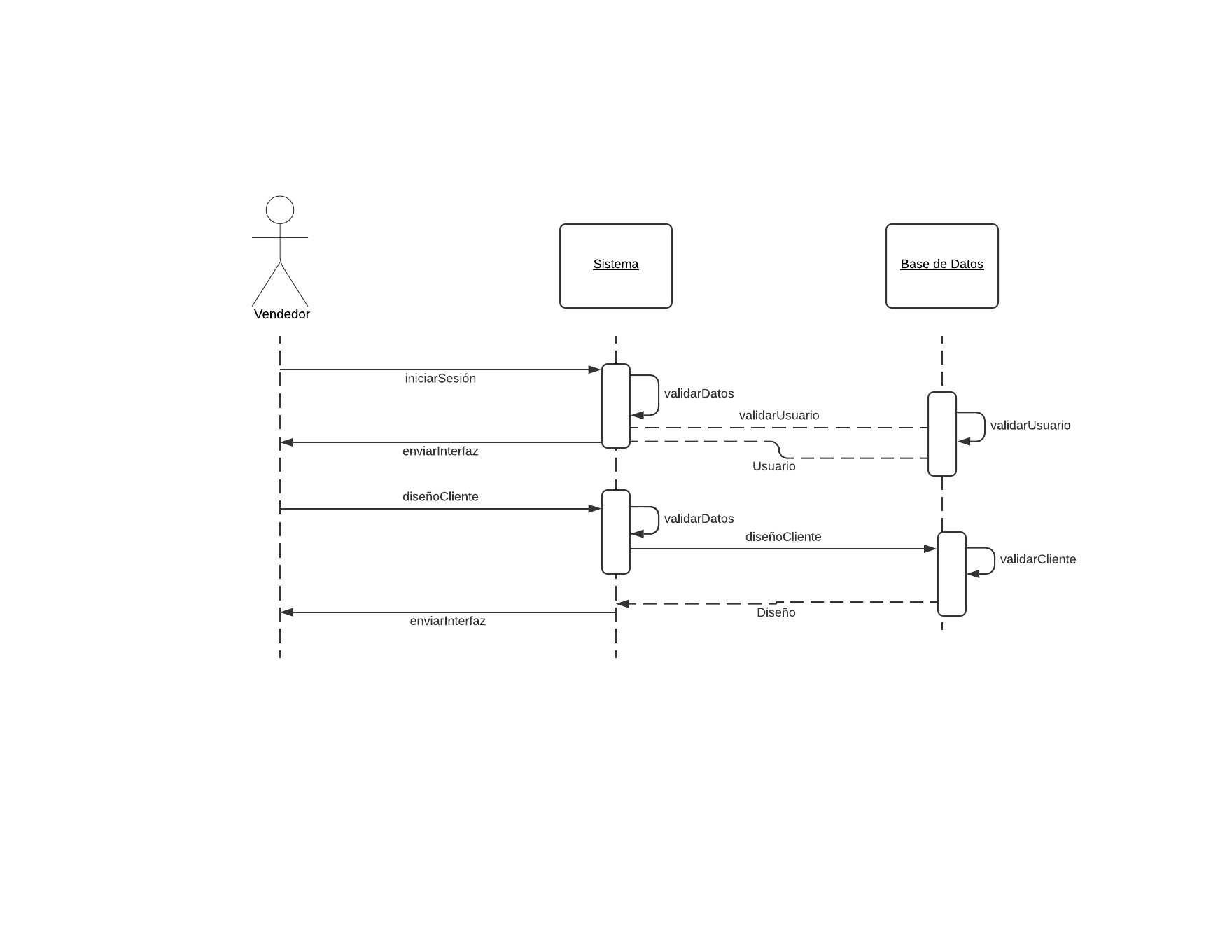
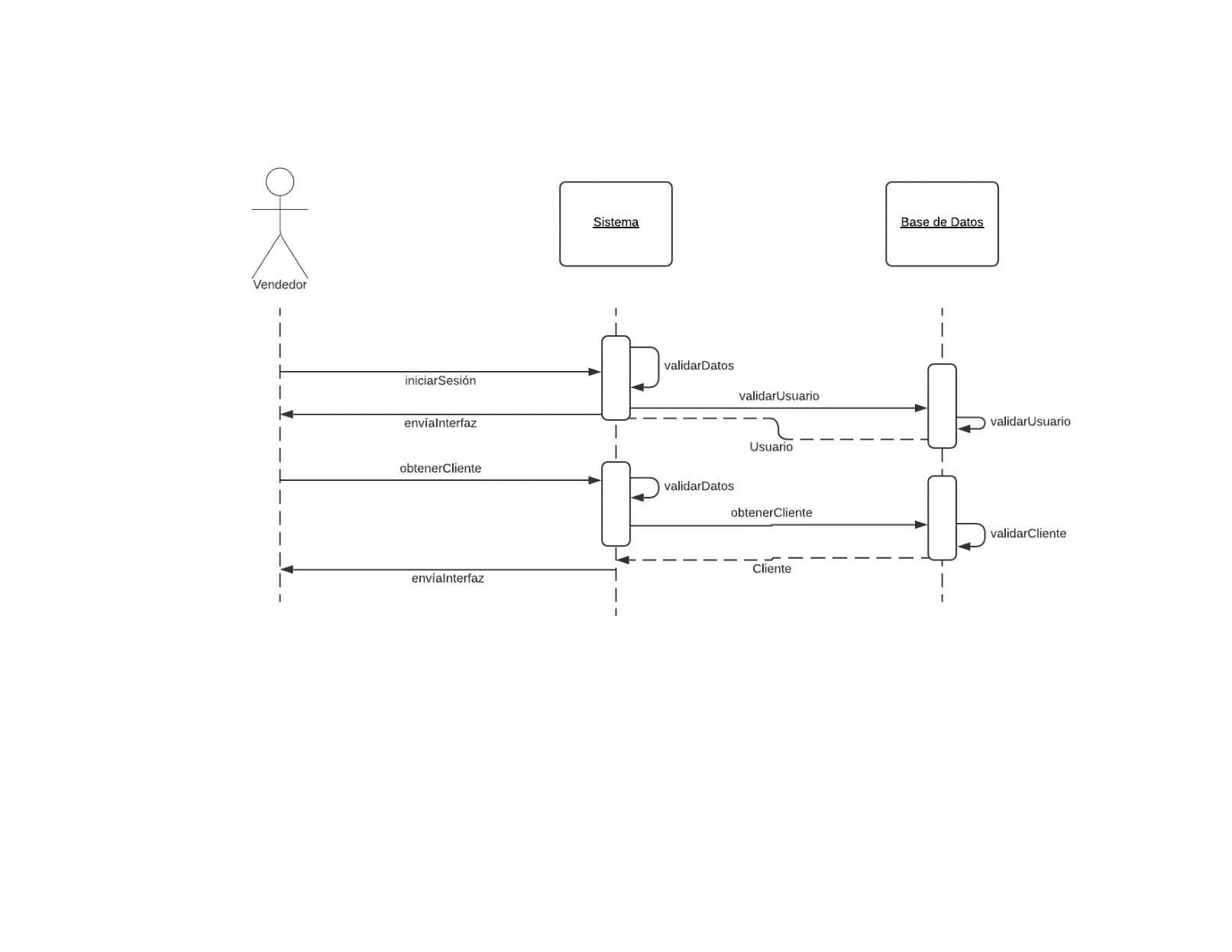
Descripción generada automáticamente

# Diagramas de secuencias

Imagen que contiene captura de pantalla, mapa

Descripción generada automáticamente





# Diagrama normalizado

Imagen que contiene captura de pantalla

Descripción generada automáticamente

# Docker

**Conexión a base de datos de Docker**

Requerimientos

-Tener instalado Docker-compose (Teniendo instalado Docker Desktop ya se instala automáticamente Docker-compose)

Pasos

1. Desde la terminal ir a la carpeta dentro del proyecto llamada dockerFiles mediante el comando cd.

2. Si se realiza desde MacOs o Ubuntu ignorar este paso, en caso de utilizar Windows hacer click derecho en icono de Docker en la barra de tareas en el escritorio y hacer click en la opción Switch to Linux Containers.

3. Ejecutar el comando docker-compose up con el cual se instala la imagen de mysql en caso de no tenerla instalada y una vez instalada se crea el contenedor y empieza a ejecutarse.

4. Conectarse a la base de datos con el usuario root y contraseña password. La dirección y puerto de enlace es localhost:3308.

# Reporte SonarQube

**Link:** http://200.126.12.164/dashboard/index/CLM