

Chapter 1

Diseño

El sistema permite el procesamiento en tiempo real de imágenes de vacas, facilitando tanto la observación como la calificación de su condición corporal. La interfaz de usuario comienza con una pantalla de inicio de sesión, donde los usuarios se autentican. Dependiendo del rol asignado (calificador u observador), el sistema presenta diferentes funcionalidades: los calificadores se enfocan primero en iniciar o finalizar una sesión de calificador y luego en evaluar el BCS de las vacas mediante un proceso de etiquetado manual, mientras que los observadores visualizan las imágenes en tiempo real y ejecutan el modelo predictivo de BCS de forma automática con un modelo pre-entrenado. El sistema utiliza un esquema distribuido, donde el procesamiento de las imágenes se gestiona a través de una cola de mensajes (JMS), permitiendo un flujo de imágenes capturadas en tiempo real de las vacas que están pasando por la manga. Estas imágenes se envían al servidor y se asocian a una vaca y a una locación específica. La cola de mensajes asegura el orden adecuado de las imágenes, mientras que el componente receptor almacena las imágenes en función de su metadata asociada. Además, el servidor gestiona las interacciones de calificadores y observadores, facilitando la creación de sesiones y el registro de calificaciones para los calificadores, así como la transmisión de datos de predicción para los observadores y, asimismo, provee reportes accesibles para ambos roles.

1.1 Login

El flujo de creación de usuario en la aplicación móvil se implementa mediante el envío de una solicitud POST al servidor. El módulo de creación de usuario del servidor procesa esta solicitud, generando el nuevo usuario en el sistema y asegurando que el nombre sea único a través de una validación.

Para la autenticación, cuando un usuario ingresa su nombre en la aplicación, se envía una solicitud POST, la cual es manejada por el módulo de login del servidor. El sistema verifica la identidad del usuario únicamente a través de su nombre, sin necesidad de una contraseña. Una vez identificado, el sistema

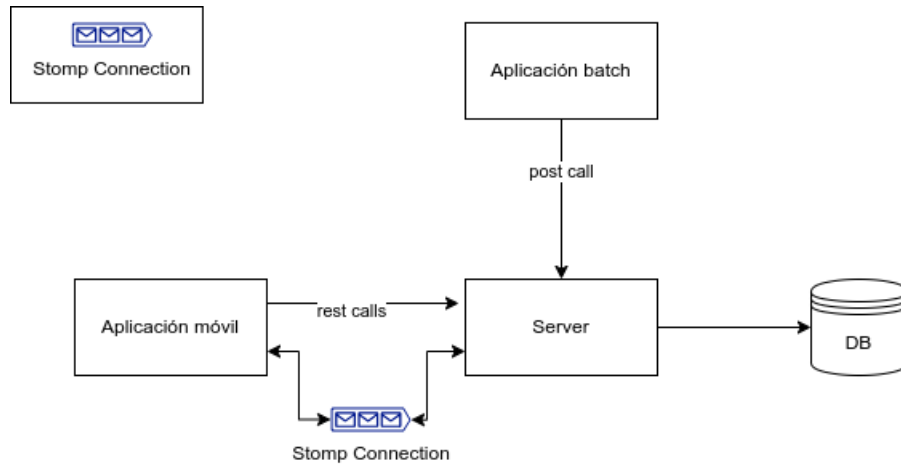


Figure 1.1: Componentes del sistema

clasifica al usuario como calificador u observador, habilitando o restringiendo funciones según su rol.

Este diseño del módulo de login, junto con la funcionalidad de creación de usuarios y la validación de nombres únicos, se alinea con los objetivos del prototipo. Se prioriza la simplicidad y la funcionalidad básica, dejando de lado la implementación de medidas de seguridad avanzadas que podrían ser consideradas en versiones futuras.

1.2 Sesión de calificación

Cada sesión de calificación está vinculada a una locación específica, predefinida en los scripts de inicialización del sistema. Este prototipo establece una sesión de calificación por defecto para simplificar la operación, pero el diseño del servidor permite la posibilidad de múltiples sesiones simultáneas en distintas locaciones. Esto significa que diferentes calificadores pueden trabajar en paralelo, ya sea en diferentes locaciones o en la misma. El hecho de permitir sesiones en diferentes locaciones en paralelo, abre la posibilidad a que el componente servidor de servicio desde nodos a los que se accede mediante internet.

El servidor genera automáticamente un identificador único para cada sesión de calificación, junto con las marcas de fecha y hora de inicio y finalización al comenzar o concluir la sesión. Aunque la aplicación móvil actualmente está limitada a la gestión de la sesión por defecto, el sistema permite que varios calificadores trabajen simultáneamente en diferentes locaciones o en la misma locación, aunque esta funcionalidad no está completamente implementada en la aplicación móvil en esta etapa.

En su estado actual, solo el calificador tiene la capacidad de iniciar y finalizar

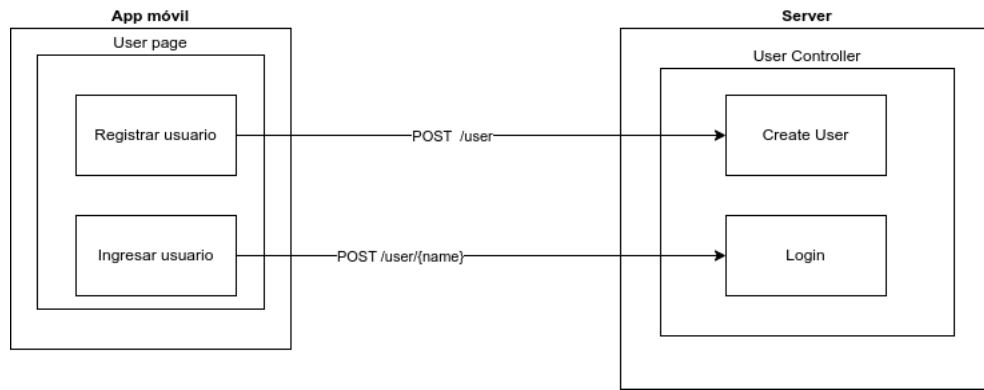


Figure 1.2: Diagrama de componentes - login

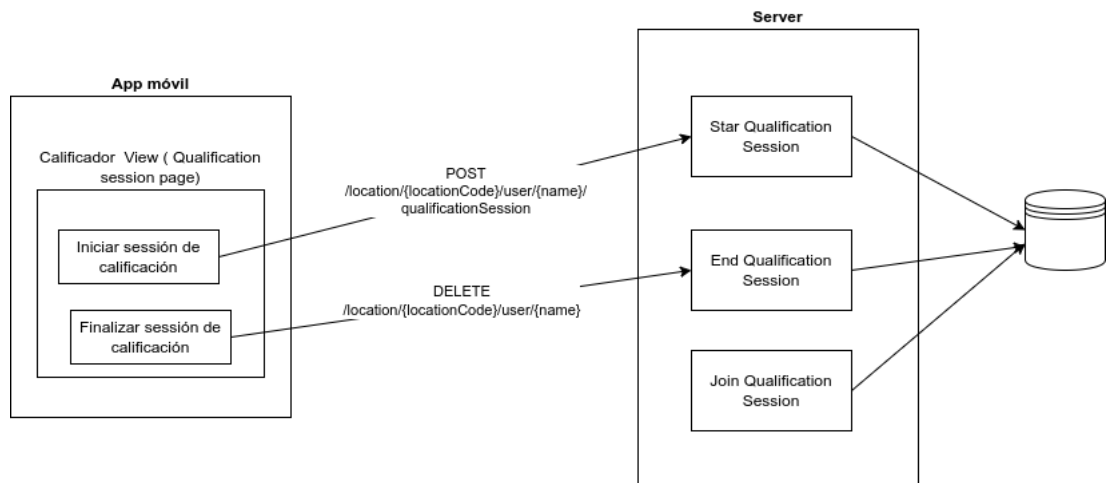


Figure 1.3: Diagrama de componentes - sesión de calificación

una sesión, lo que implica que no puede haber una sesión sin la intervención del calificador. No obstante, es posible iniciar una sesión de calificación sin la presencia física de un calificador mediante la invocación directa de la misma llamada REST que este utilizaría para gestionar la sesión. Esto permite que el sistema mantenga sesiones de calificación sin requerir la intervención directa del calificador en todo momento. En el futuro, el sistema podría adaptarse para que los observadores también tengan la capacidad de iniciar y finalizar sesiones, lo que ofrecería mayor flexibilidad, permitiendo que tanto observadores como calificadores trabajen de manera independiente, sin depender necesariamente de la presencia del otro.

La base de datos registra cada sesión de calificación, lo que es esencial para el módulo de reportes que genera informes basados en estas sesiones. Los componentes claves incluyen:

- Inicio de Sesión de Calificación: El servidor gestiona el inicio de las sesiones de calificación, garantizando que solo exista una sesión activa por locación. Además, se encarga de asociar a la nueva sesión todas las imágenes previamente almacenadas que aún no estén vinculadas a una sesión. De este modo, se asegura que ninguna imagen quede sin procesar si no hay una sesión de calificación activa en el momento de su recepción.
- Finalización de Sesión de Calificación: El servidor cierra la sesión, permitiendo el uso de datos en reportes.
- Unirse a una Sesión de Calificación: Aunque el servidor soporta esta funcionalidad, la aplicación móvil actual no permite que un calificador se una a una sesión iniciada por otro.
- Locación de la Sesión de Calificación: Actualmente, sólo permite iniciar o finalizar sesiones en la locación por defecto, pero es extensible para permitir más funcionalidades en futuras versiones.

1.3 Sistema de notificación de imágenes

El diseño del sistema para notificar las imágenes de las vacas se centra en varios componentes clave que interactúan para garantizar el procesamiento adecuado de las imágenes y su integración con el flujo de calificación y observación. A continuación, se detalla el proceso de recepción y almacenamiento de imágenes, la gestión de mensajes, y el rol específico de los calificadores y observadores en la transmisión y visualización de estos datos.

1.3.1 Recepción y Almacenamiento de Imágenes

El sistema cuenta con una interfaz REST que recibe las imágenes capturadas junto con la identificación de la vaca, la locación de la manga y la posición de la vaca. Este último dato corresponde al orden en el que se registran los eventos, es decir, la posición de la vaca en la caravana dentro del sistema de identificación.

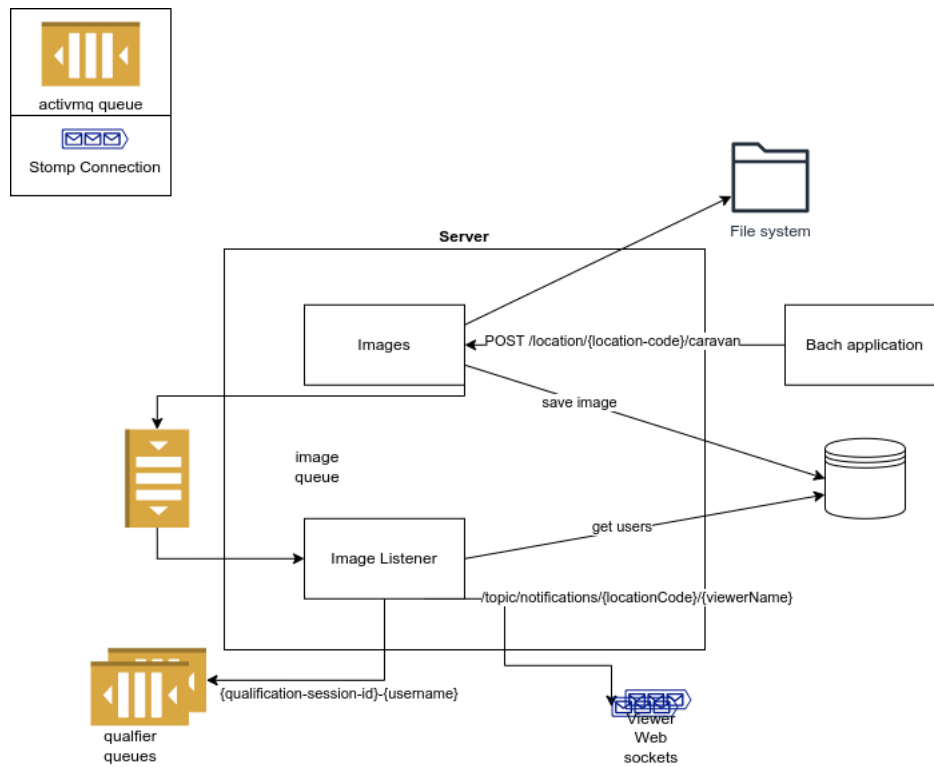


Figure 1.4: Diagrama de componentes - imágenes

Este mecanismo asegura que las imágenes estén asociadas a la vaca correcta, la posición de esta dentro de la caravana a la locación específica desde donde se envían. La metadata relacionada (identificación de la vaca, locación, posición etc.) se almacena en la base de datos, mientras que las imágenes en sí se guardan en el sistema de archivos.

Este proceso ocurre de manera independiente a la activación de una sesión de calificación, ya que las imágenes se almacenan de inmediato y se asocian a la sesión correspondiente cuando esta se inicia. Dado que no puede haber más de una sesión de calificación por locación y que las imágenes están asociadas tanto a una vaca como a una locación, es sencillo determinar qué imágenes deben ser utilizadas por los observadores para realizar la predicción de BCS.

Posteriormente, cada conjunto de imágenes genera un mensaje JMS (Java Message Service) que es enviado a una cola de procesamiento. Cada conjunto se asocia a un único animal, asegurando que la información enviada para su procesamiento corresponda exclusivamente a una vaca en particular.

1.3.2 Gestor de mensajes de imágenes

El Gestor de Mensajes de Imágenes distribuye los mensajes generados por la llegada de imágenes a los componentes correspondientes que incluyen tanto la gestión de las imágenes que serán calificadas por expertos como las que se visualizarán en tiempo real por los observadores. Este proceso se realiza mediante dos rutas principales:

- Mensajes a calificadores: El sistema verifica la locación de las imágenes para determinar si hay una sesión de calificación activa. Si es así, los mensajes JMS se envían a las colas de los calificadores asociados a esa sesión, permitiendo que puedan realizar la evaluación. Es imprescindible que exista una sesión activa para que los calificadores accedan a las imágenes.
- Mensajes a observadores: Los observadores reciben las imágenes a través de una conexión websocket, independientemente de la existencia de una sesión de calificación. Las imágenes son transmitidas en tiempo real para que los observadores las visualicen.

Esta arquitectura asegura una distribución eficiente de las imágenes: los calificadores trabajan en sus respectivas sesiones y los observadores pueden ver las imágenes y ejecutar el algoritmo predictivo de BCS de manera simultánea, incluso sin una sesión de calificación en curso.

1.3.3 Calificadores

Los calificadores reciben las imágenes mediante un sistema de colas JMS, que les permite acceder a los trabajos de calificación pendientes. El sistema gestiona estos trabajos de manera robusta, asegurando que no se pierdan datos en caso de interrupciones del sistema.

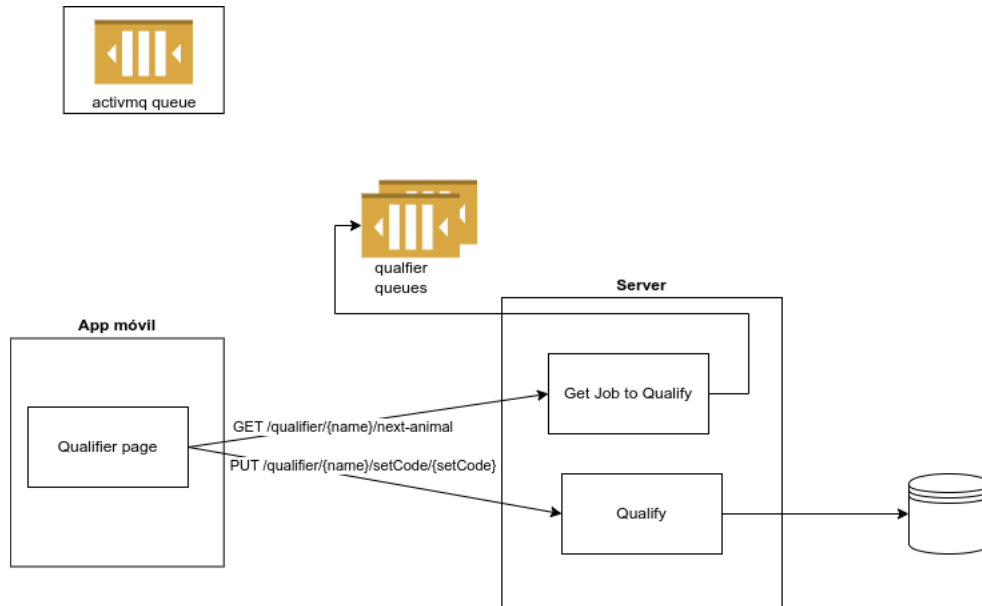


Figure 1.5: Diagrama de componentes: calificador

Para evaluar, los calificadores deben estar asociados a una sesión de calificación activa. Las imágenes de las vacas se envían únicamente si hay calificadores disponibles. El calificador accede a las imágenes mediante solicitudes GET a su cola asignada. Una vez evaluada una imagen, el puntaje BCS es enviado al servidor mediante una solicitud PUT, y el trabajo queda registrado en el sistema.

Este proceso asegura que las evaluaciones de los calificadores estén organizadas y almacenadas correctamente para su uso posterior, como el entrenamiento o mejoramiento de modelos predictivos de BCS.

1.3.4 Observador

En el diseño actual del prototipo, los observadores reciben las imágenes en tiempo real a través de websockets, conectados por defecto a una locación específica. A diferencia de los calificadores, los observadores no necesitan estar asociados a una sesión de calificación para visualizar las imágenes y ejecutar de forma automática el algoritmo predictivo BCS.

Actualmente, el cálculo del puntaje BCS se lleva a cabo exclusivamente en los dispositivos móviles de los observadores. Esto es debido a que el servidor no está diseñado para realizar este procesamiento. El sistema utiliza un enfoque de round-robin para seleccionar un observador de la lista de aquellos que tienen conexiones websocket abiertas, permitiendo que el dispositivo seleccionado realice la predicción del puntaje BCS. El resultado de esta predicción se envía al servidor

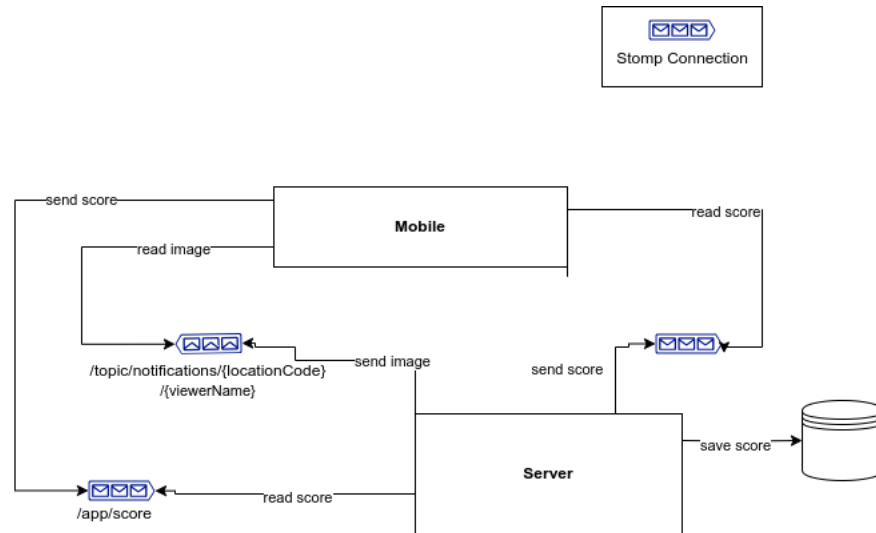


Figure 1.6: Diagrama de componentes - Observadores

y se distribuye a otros observadores en la misma locación a través de websocket, lo que permite que el puntaje BCS calculado sea visualizado en tiempo real en los dispositivos de todos los observadores activos.

Sin embargo, el diseño del sistema podría ser adaptado para que, en futuras versiones, el servidor pueda realizar estos cálculos, siempre y cuando se cuente con un hardware adecuado para el procesamiento en tiempo real. En escenarios donde el servidor sea de bajo costo y con capacidades limitadas, se seguiría dependiendo de los dispositivos móviles para la ejecución del algoritmo. En cambio, si el servidor fuera actualizado a un hardware más potente, todo el procesamiento podría centralizarse allí, eliminando la necesidad de utilizar los dispositivos de los observadores.

En la versión actual, el puntaje calculado en los dispositivos móviles se envía al servidor y se distribuye a otros observadores en la misma locación a través de websocket. Así, el puntaje BCS se visualiza en tiempo real en los dispositivos de todos los observadores activos. Si no hay observadores disponibles al momento de capturar las imágenes, la vaca no obtendrá un puntaje predictivo, ya que el algoritmo de predicción debe ejecutarse en los dispositivos de los observadores.

Los observadores asignados ejecutan el algoritmo de predicción en sus dispositivos, y el puntaje calculado se envía a:

- Su pantalla: Para que puedan visualizar el resultado de inmediato.
- El servidor: Donde se almacena en la base de datos para su posterior análisis.
- Los demás observadores conectados: Para que también puedan ver el puntaje obtenido.

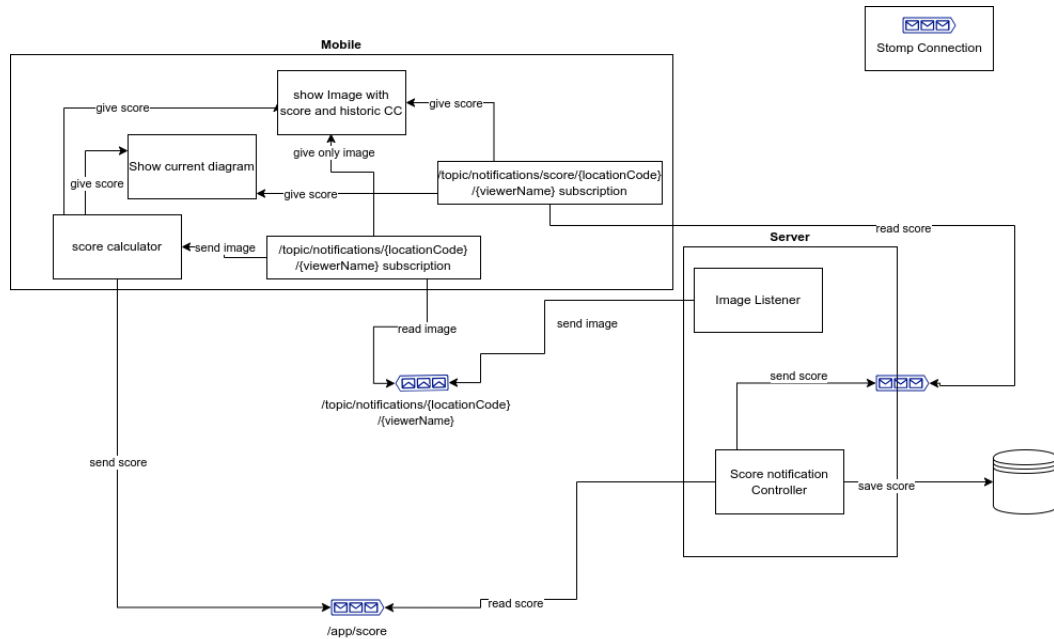


Figure 1.7: Diagrama de componentes - Observadores en detalle

Si el planificador designa la tarea de predicción a otro observador, el dispositivo del observador inicial espera el resultado del puntaje BCS. Una vez que llega, el puntaje se muestra automáticamente en su pantalla, actualizando simultáneamente los reportes y visualizaciones.

Esta estructura garantiza que el flujo de imágenes, la predicción de BCS y la visualización por parte de los observadores se lleven a cabo de manera eficiente y sin interrupciones, independientemente de si hay sesiones activas.

Verificar y aclarar que solo se puede ver si no hay sesiones activas el número de predicción de la vaca y su historial dentro del sistema. Pero no se puede ver el reporte de la sesión hasta que no inicie esta..

1.4 Reportes

Cada opción de visualización de reportes, ya sea para sesiones actuales o reportes de sesiones anteriores, se accede mediante llamadas REST a la base de datos. Esta estructura permite que tanto observadores como calificadores obtengan la información necesaria cuando lo deseen. Es importante señalar que los reportes solo contienen los valores de las predicciones generadas por el sistema. Los calificadores no tienen acceso a los valores que ellos mismos u otros calificadores asignan a las vacas durante el proceso de calificación.

- Actualización de estado en Tiempo Real: En las sesiones actuales, sólo los

observadores reciben actualizaciones de las predicciones en tiempo real, lo que les permite actualizar el diagrama de torta automáticamente. Por otro lado, los calificadores no reciben estas actualizaciones en tiempo real, ya que su función principal es calificar las vacas. Sin embargo, pueden acceder a los datos en cualquier momento, ingresando a la sección de la aplicación donde se visualizan los reportes.

- Historial de BCS: Exclusivo para los observadores, el historial de BCS se obtiene mediante una llamada REST a la base de datos. Este diagrama se actualiza automáticamente con el valor predicho una vez que el cálculo del BCS para la sesión actual finaliza, garantizando que los observadores tengan acceso a la información más precisa y actualizada.

Esta arquitectura no solo facilita la visualización y el acceso a los reportes, sino que también asegura que tanto los observadores como los calificadores tengan la información necesaria para tomar decisiones informadas sobre la condición corporal de las vacas. En el futuro, la implementación de un apartador automático podría jugar el rol de observador y actuar habilitando diferentes puertas para apartar a las vacas de un rodeo que requieren una dieta o tratamiento especial basado en su valor de BCS.

1.4.1 Observador

En el caso de los observadores, como se mencionó anteriormente, dado que es posible que los observadores estén mirando la pantalla de diagramas y dado que siempre recibe el BCS calculado (o por él mismo o por otro observador) se actualiza el diagrama automáticamente utilizando websocket y el protocolo STOMP.

Los observadores pueden ver los diagramas de torta que se actualizan automáticamente en tiempo real al recibir nuevas predicciones de BCS, ya sea realizadas por ellos mismos o por otros observadores. Además, pueden consultar el historial de BCS de vacas en sesiones anteriores cuando reciben una nueva imagen, lo que les permite comparar los datos actuales con los registros previos.

1.4.2 Calificador

En el caso de los calificadores, su función principal es evaluar las vacas, por lo que los diagramas se actualizan únicamente cuando deciden consultar los reportes a través de una llamada REST. Este enfoque permite que realicen las solicitudes cuando lo consideren necesario, evitando la necesidad de mantener una conexión websocket abierta para recibir notificaciones sobre las predicciones.

1.5 Implementación

Esta sección describe las tecnologías, bibliotecas y configuraciones empleadas para desarrollar e implementar el prototipo. A continuación, se detallan los

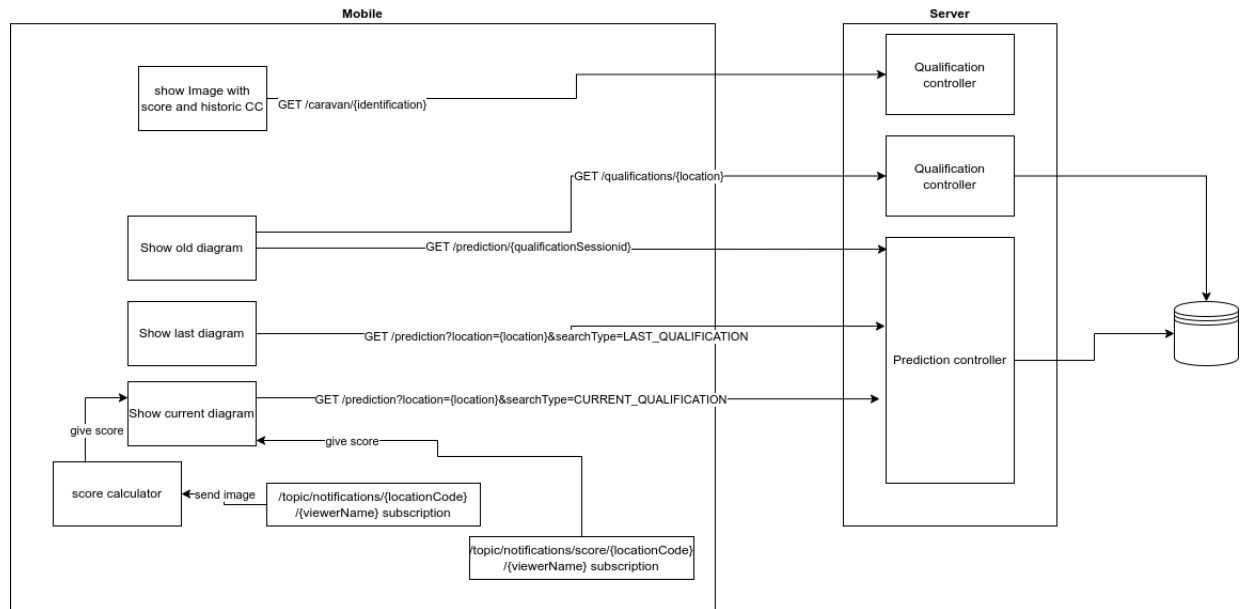


Figure 1.8: Diagrama de componentes - reporte de observadores

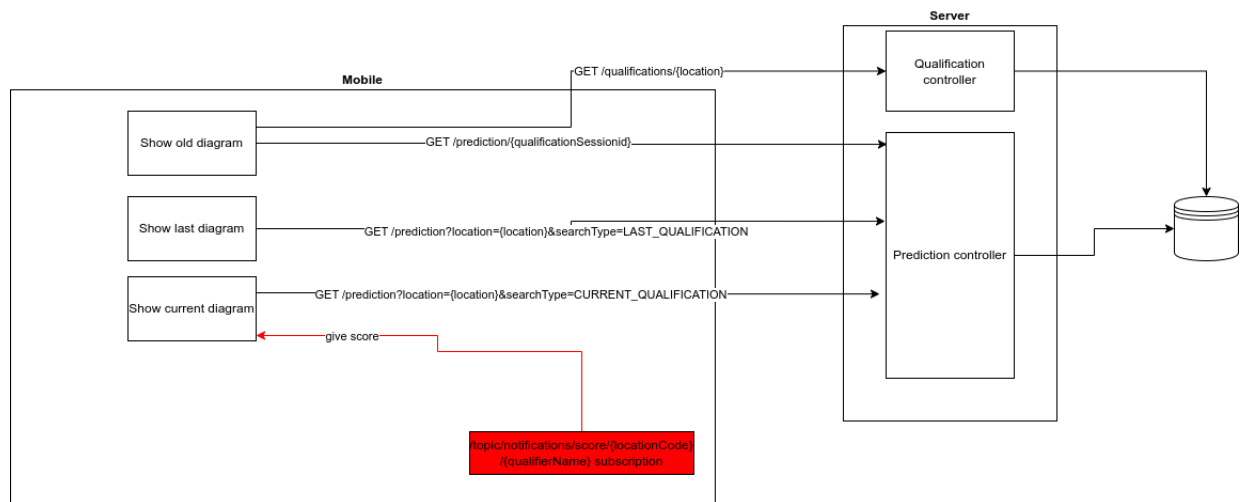


Figure 1.9: Diagrama de componentes - reporte de calificadores

componentes principales, la configuración del sistema y el modelo de base de datos.

1.5.1 Tecnologías y Bibliotecas Utilizadas

- Servidor: El backend está desarrollado con Spring Boot para la creación del servidor, aprovechando Spring Web para implementar los servicios REST y el protocolo STOMP para la comunicación a través de WebSockets.
- Base de Datos: Se utiliza MySQL como base de datos relacional y JDBC para interactuar con ella.
- Mensajería: La integración de ActiveMQ con JMS facilita la comunicación asincrónica y garantiza la disponibilidad de los mensajes entre los distintos componentes.
- Aplicación Móvil: La aplicación está desarrollada en Flutter para Android, con capacidad de expansión hacia iOS. Esto permite la posibilidad de adaptaciones para dispositivos iOS en el futuro.

1.5.2 Configuración del sistema

1.5.2.1 Configuración del Servidor

1. Requisitos previos
 - (a) java 11: Es necesario para ejecutar la aplicación: `java -version`
 - (b) Maven(opcional si se compilará desde el código fuente)
2. Configuración de ActiveMQ Artemis
 - (a) Descarga la última versión de ActiveMQ Artemis desde su sitio oficial, descromprime el archivo y crea una instancia del broker: `./artemis create mybroker --user admin --password admin`
 - (b) Inicia el brojer en el puerto 61616: `./artemis run`
3. Configuración de MySQL
 - (a) Instala MySQL 8.0 y arranca el servicio
 - i. En macOS/Linux: `mysql.server start`
 - ii. en window: `net start mysql`
 - (b) Accede a MySQL y crea la base de datos server: `CREATE SCHEMA server;`

1.5.2.2 Configuración de docker

Opcionalmente, puedes configurar todos los servicios a través de contenedores Docker para simplificar la instalación y configuración (recomendado para ambientes de desarrollo).

Para iniciar el servicio de mensajería en un contenedor Docker con ActiveMQ Artemis, ejecuta el siguiente comando:

```
docker run -p 61616:61616 -e AMQUSER=admin-eAMQPASSWORD=adminquay.io/artemiscloud/activemq-artemis-broker
```

Este comando configura el broker de mensajería en el puerto 61616.

Luego, inicia la base de datos MySQL en un contenedor Docker con el siguiente comando:

```
docker run --name=mysql-db -e MYSQL_ROOT_PASSWORD=root -p 3306:3306 mysql:8.0
```

Una vez que el contenedor esté en funcionamiento, accede a MySQL y crea el esquema necesario para el sistema:

```
docker exec -it mysql-db mysql -p
```

Cuando se te solicite la contraseña, introduce root y luego ejecuta el siguiente comando para crear el esquema:

```
CREATE SCHEMA server;
```