

Universidad Nacional
del Centro de la Provincia De Buenos Aires



Facultad de Ciencias Exactas
Carrera de Ingeniería de Sistemas

Un prototipo de ambiente Dew para etiquetado y clasificación de ganado

por
Cristian David Roldan

Director: Dr. Matías Hirsch

Co-Director: Dr. Juan Manuel Toloza

Tandil, Buenos Aires, Argentina

Trabajo final de carrera presentado como requisito parcial
para optar por el título de
Ingeniero de Sistemas

Agradecimientos

A mis padres, cuyo apoyo incondicional me permitió mudarme de ciudad y dedicarme a mis estudios, brindándome siempre la seguridad y el respaldo necesarios para alcanzar esta meta.

A mi mujer, por su paciencia, aliento y apoyo moral en los momentos más desafiantes, impulsándome a finalizar esta tesis después de tantos años.

A mi director, por su guía y acompañamiento a lo largo de todo el proceso de desarrollo.

A todos ellos, mi más sincero y profundo agradecimiento.

Resumen

El objetivo fue desarrollar un prototipo de computación Dew que aproveche capacidad de cómputo presente en la cercanía de un usuario para ejecutar tareas de clasificación y etiquetado en tiempo real sin necesidad de conexión a internet. Las tareas de clasificación emplearán un modelo tensorflow previamente desarrollado en el marco de un trabajo publicado en una revista científica de alto impacto en el área de las tecnologías aplicadas al agro Alvarez et al. [2018]. El prototipo a desarrollar tiene múltiples propósitos que pueden resumirse en:

El prototipo busca mejorar el modelo mediante un reentrenamiento optimizado gracias a un proceso ágil de etiquetado de imágenes. Además, automatiza la clasificación del ganado con dispositivos móviles a través de una aplicación Android, que permite a expertos etiquetar imágenes y a usuarios no expertos visualizar el BCS. Esta aplicación se comunica con un servidor que gestiona tareas de etiquetado y clasificación, diferenciando roles de usuario. También se implementó un módulo de reportes para visualizar el BCS calculado por el algoritmo.

El sistema desarrollado es funcional y escalable, mejorando la precisión y eficiencia en la predicción del BCS en entornos ganaderos y permitiendo futuras ampliaciones, como la integración con IoT y la optimización de modelos predictivos. Su eficiencia se evaluó midiendo tiempos de predicción en dispositivos móviles comparándola con una versión preliminar, optimizando el flujo de datos entre servidor y dispositivos en un entorno realista.

Contents

1	Introducción	1
1.1	Uso de tecnología de la información en granjas	1
1.2	Importancia de BCS y la necesidad de mejorar la alimentación de las vacas	2
1.3	Motivación	2
1.3.1	Escasez de prototipos	3
1.4	Propuesta	4
1.4.1	Objetivo	4
1.4.2	Contexto	5
1.4.3	Desafíos actuales	5
1.4.4	Solución propuesta	5
2	Trabajos relacionados	7
2.1	Importancia y avances de la IA para calcular BCS	7
2.2	Importancia de dew computing	8
2.3	Comparaciones	9
2.3.1	Limitaciones halladas en trabajos relacionados	11
2.3.2	Mejoras de mi propuesta	12
3	Requerimientos	15
3.1	Requerimientos funcionales	15
3.2	Casos de uso	16

CONTENTS

3.2.1	Login	16
3.2.2	Sesión de calificación	17
3.2.3	Sistema de notificación de imágenes	18
3.2.4	Calificadores	18
3.2.5	Observadores	19
3.2.6	Reportes	19
4	Diseño e implementación	21
4.1	Login	21
4.2	Sesión de calificación	22
4.3	Sistema de notificación de imágenes	24
4.3.1	Recepción y Almacenamiento de Imágenes	25
4.3.2	Gestor de mensajes de imágenes	25
4.3.3	Calificadores	26
4.3.4	Observador	26
4.4	Reportes	28
4.4.1	Observador	29
4.4.2	Calificador	30
4.5	Implementación	30
4.5.1	Tecnologías y Bibliotecas Utilizadas	30
4.5.2	Pasos para la configuración del sistema	31
4.5.2.1	Configuración del componente Servidor	31
4.5.2.2	Configuración de la aplicación móvil.	32
4.5.2.3	Configuración de docker	33
4.5.3	Modelo de base de datos	34
5	Resultados	39
5.1	Pantalla de Inicio de Sesión (Login)	39
5.2	Calificador	40
5.3	Sistema de Notificación de Imágenes	42
5.4	Reportes Generados	42
5.5	Rendimientos en dispositivos móviles	44
5.6	Rendimientos en el servidor	47

CONTENTS

6 Conclusiones, Limitaciones y Trabajos Futuros	49
6.1 Conclusión	49
6.2 Limitaciones	50
6.2.1 Seguridad	50
6.2.2 Tolerancia a fallos	50
6.2.3 Mejora en la Gestión de Jobs para Calificadores	50
6.2.4 Observadores y Algoritmo de Predicción	51
6.3 Trabajos futuros	51
6.3.1 Múltiples calificadores	51
6.3.2 Sistema generador de Jobs	51
6.3.3 Procesamiento de imágenes	52
6.3.4 Integración con IoT	52
6.3.5 Múltiples modelos	52
6.3.6 Múltiples plataformas	52
6.3.7 Configuración de la participación de observadores en el cálculo de BCS .	53
Bibliography	55

CONTENTS

List of Tables

2.1	Uso de dispositivos móviles e IA en tareas de ganadería	10
5.1	Transferencia de datos	45
5.2	Tiempos de ejecución móvil	46
5.3	Tiempo de procesamiento total	47

LIST OF TABLES

List of Figures

4.1	Componentes del sistema	22
4.2	Diagrama de componentes - login	22
4.3	Diagrama de componentes - sesión de calificación	23
4.4	Diagrama de componentes - imágenes	24
4.5	Diagrama de componentes: calificador	26
4.6	Diagrama de componentes - Observadores	27
4.7	Diagrama de componentes - Observadores en detalle	28
4.8	Diagrama de componentes - reporte de observadores	29
4.9	Diagrama de componentes - reporte de calificadores	30
5.1	App móvil: caso de uso de login	40
5.2	App móvil: caso de uso calificación	41
5.3	App móvil: Arribo de imágenes	41
5.4	App móvil: menú	43
5.5	App móvil: reportes	43
5.6	Tiempo de detección	45

Introducción

1.1 Uso de tecnología de la información en granjas

La adopción de tecnología en las granjas tiene como objetivo transformar la forma en que se gestionan y optimizan las operaciones agropecuarias, permitiendo una mayor eficiencia, sostenibilidad y precisión en la producción. En el ámbito de la ganadería lechera, por ejemplo, se han diseñado sistemas avanzados para la detección del estro en vacas, como lo describe el artículo de Li et al. [2010]. Este sistema utiliza tecnología inalámbrica Zigbee y sensores de infrarrojos para monitorear la temperatura corporal y el movimiento del ganado, tratando de mejorar la precisión en la detección del estro y reduciendo la carga laboral.

Por otro lado, en un contexto más amplio, la adopción de tecnologías como la Internet de las Cosas (IoT) y Edge Computing (EC) son cruciales en la industria láctea para poder adaptarse a un mercado globalizado. Según Alonso et al. [2020], estas tecnologías permiten digitalizar toda la cadena de valor, optimizando procesos, garantizando la trazabilidad y asegurando la calidad del producto final, lo que es fundamental en un entorno de agricultura inteligente.

Finalmente, la aplicación de técnicas de Deep Learning (DL) también está revolucionando la ganadería. En Mahmud et al. [2021] destacan cómo DL se está utilizando para la automatización de tareas como la identificación y el monitoreo de la salud del ganado, aunque aún existen desafíos significativos, especialmente en el monitoreo en tiempo real y el rastreo de movimientos en condiciones de granja.

En conjunto, estos avances subrayan el papel esencial de la tecnología en la modernización de la agricultura y la ganadería, impulsando mejoras en la productividad.

1.2 Importancia de BCS y la necesidad de mejorar la alimentación de las vacas

Entre varias variables de importancia productiva, económica y ambiental, existe consenso en que la identificación de la condición corporal (BCS, por sus siglas en inglés) y los cambios en esta son los mejores indicadores de las reservas nutricionales de una vaca, superando al peso vivo o al cambio de peso vivo. Esto se debe a que estas últimas variables pueden verse afectadas por factores como el peso fetal y el llenado del rumen, como se describió tempranamente en la lechería Wildman et al. [1982] y en el ganado bovino Richards et al. [1986]. A lo largo del tiempo, se han acumulado evidencias sobre la efectividad del sistema visual de condición corporal como un fuerte predictor de la capacidad productiva, reproductiva y del bienestar animal Roche et al. [2009].

Por lo tanto, el BCS es una herramienta fundamental en la gestión del ganado lechero, ya que permite estimar la movilización de reservas energéticas en grasa y músculo, siendo crucial para asegurar el bienestar animal y la eficiencia productiva. Según Yukun et al. [2019], el BCS es ampliamente utilizado para monitorear la ingesta de alimentos y el estado de salud del ganado, además de optimizar la gestión de la curva de lactancia. Sin embargo, la evaluación manual tradicional del BCS, aunque útil, es subjetiva y propensa a errores, dado que debe tenerse en cuenta la variabilidad entre observadores Morin et al. [2017].

Por otra parte, la mejora de la alimentación del ganado es esencial para maximizar la producción de leche, y en este contexto, la precisión en la evaluación del BCS se vuelve aún más crítica. Çevik [2020] destacan la importancia del BCS como el indicador principal de una alimentación adecuada en el ganado lechero, y señalan que agrupar a los animales según su puntuación de BCS permite una gestión más eficiente de la alimentación.

1.3 Motivación

A pesar del valor indiscutido del BCS, existe una subutilización de este valioso indicador en los sistemas comerciales por múltiples causas, entre las que se encuentran la no disponibilidad regular del/la experto/a en los movimientos de hacienda, superposición de actividades que relega su medición regular, falta local de especialistas entre otros De Hagen et al. [2010], Maresca et al. [2008], Selk et al. [1988]. Por este motivo, la principal motivación de este proyecto es la de desarrollar un prototipo de ambiente Dew con aplicación en el sector de producción tamero. La razón por la que se elige este dominio es porque se cuenta con avances obtenidos en el marco de una tesis doctoral sobre un modelo que aplica aprendizaje profundo para clasificación de ganado bovino basado en el indicador de condición corporal. La idea es realizar un desarrollo que permita agilizar el etiquetado de imágenes –aumento del conjunto de datos de entrenamiento– que se realiza con una cámara de profundidad. También se busca realizar

una clasificación en tiempo real in situ, es decir, a medida que los animales pasan por lo que en el campo se conoce con el nombre de manga donde se realizan diversas prácticas veterinarias. A partir del prototipo generado y planteando la tarea de clasificación a partir de imágenes como un cómputo en el que intervienen varios dispositivos de usuario, también se busca validar la performance de heurísticas de planificación de tareas estudiadas mediante simulación.

1.3.1 Escasez de prototipos

Si bien existe abundante investigación sobre algoritmos e inteligencia artificial aplicados en granjas Bao and Xie [2022], este artículo destaca que persiste la necesidad de desarrollar aplicaciones sistemáticas que integren dispositivos de recolección de datos, procesamiento y algoritmos inteligentes. Según estos autores, esto es especialmente relevante en el contexto de la ganadería inteligente, donde se requieren sistemas de bajo costo y modelos de IA eficientes. Sin embargo desafíos como la precisión de los sensores y el costo de los dispositivos, continúan siendo un obstáculo que debe superarse para lograr una implementación efectiva.

Además, los estudios sobre machine learning y reconocimiento de patrones han mostrado avances en áreas como la detección de enfermedades y el monitoreo del comportamiento animal. Sin embargo, como se observa en los artículos de Kumar et al. [2018] y Riekert et al. [2020], muchos de estos estudios se centran en marcos teóricos o modelos de detección sin una implementación práctica clara. Por ejemplo, el primero presenta un modelo de identificación de vacas mediante deep learning, pero no aborda cómo se implementará en la práctica.

En el ámbito del reconocimiento de patrones y procesamiento de imágenes, se han desarrollado diversos modelos teóricos sin implementación efectiva que pueden procesar y analizar información proveniente de imágenes, videos y sonidos de manera similar a la inteligencia cerebral. Estos modelos permiten describir e identificar comportamientos individuales, evaluar la calidad de la carne y medir el rendimiento en el crecimiento animal. Por ejemplo, Masferrer et al. [2018] propone un método de clasificación automática, combinados con datos intrínsecos de los cerdos, para simular un sistema de clasificación manual. Del mismo modo, Masferrer et al. [2018] introducen un algoritmo inteligente para el monitoreo del comportamiento alimentario mediante el reconocimiento de patrones y aprendizaje automático.

En el campo de las redes neuronales artificiales (ANN), los desarrollos recientes se han enfocado principalmente en la creación de modelos y algoritmos dirigidos a la detección de enfermedades animales, la estimación del rendimiento de crecimiento y el monitoreo y control de las condiciones ambientales. Por ejemplo, Nasirahmadi et al. [2019] desarrollaron un algoritmo para monitorear automáticamente la postura de acostado lateral y esternal de los cerdos, proporcionando información crucial sobre su salud y bienestar.

Aunque todos estos modelos y algoritmos representan avances significativos en teoría, su implementación práctica en entornos reales aún es limitada. Esto subraya la necesidad de

traducir estas innovaciones en soluciones tangibles que puedan ser aplicadas efectivamente en la ganadería, llevando los desarrollos teóricos al uso práctico en la producción animal.

1.4 Propuesta

La propuesta consiste en desarrollar un prototipo de ambiente de computación Dew compuesto de un componente servidor y una aplicación móvil diseñada para predecir el BCS en vacas lecheras, combinando tecnologías de inteligencia artificial, aprendizaje automático y computación en el borde. Este prototipo no sólo servirá para predecir el BCS utilizando un modelo de red neuronal previamente entrenado para vacas lecheras de raza Holstein-Freisian, sino que también permitirá almacenar la información obtenida durante las evaluaciones realizadas por expertos, dando la posibilidad de reentrenamiento continuo del modelo.

1.4.1 Objetivo

El objetivo general de este proyecto es el de desarrollar un prototipo de ambiente de computación Dew que aproveche capacidad de cómputo presente en la cercanía de un usuario para ejecutar tareas de clasificación y etiquetado in-situ. Desde el punto de vista práctico, la capacidad de cómputo estará representada por dispositivos de bajo costo tipo SBC (Single Board Computers) y/o dispositivos móviles (smartphones) que podrán asumirse ya presentes en el contexto de utilización (tambo lechero) o en su defecto factibles de ser desplegados con mínimos requerimientos para llevar a cabo tareas de clasificación de ganado en tiempo real. Las tareas de clasificación propiamente dichas, emplearán un modelo tensorflow previamente desarrollado en el marco de un trabajo publicado en una revista científica de alto impacto en el área de las tecnologías aplicadas al agro Alvarez et al. [2018]. El prototipo a desarrollar tiene múltiples propósitos que pueden resumirse en:

- 1- Contribuir a mejorar el modelo tensorflow a partir de un reentrenamiento que se será posible gracias a la agilización del proceso de etiquetado de imágenes en las que interviene un profesional veterinario experto en el dominio.
 - 2- Permitir que conocimiento experto en el dominio y encapsulado en una red de aprendizaje profundo pueda ser utilizado para automatizar la clasificación de ganado empleando dispositivos de usuario como plataforma de ejecución.
 - 3- Permitir la validación de resultados de estrategias de planificación de tareas en grupos de dispositivos móviles estudiados con anterioridad en el grupo de trabajo y empleando simulación.
- De este objetivo general se desprenden los siguientes objetivos específicos:

- Diseño y desarrollo de una app móvil Android Evaluador que oficia de cliente y permite al profesional veterinario experto en el dominio etiquetar imágenes de ganado vacuno

tomadas con una cámara de profundidad en tiempo real en el contexto de labores realizados en una manga, sin requerir de conexión a Internet.

- Diseño y desarrollo de una app móvil Android Visualizador que permite a un usuario no experto en el dominio conocer la condición corporal del ganado vacuno a medida que se obtienen imágenes de este en tiempo real y en el contexto de labores realizadas en una manga, sin requerir de conexión a Internet.
- Diseño y desarrollo de un componente servidor encargado de la gestión de tareas que son el etiquetado y clasificación automática diferenciando la app cliente de la cual provenga una petición.

Para los objetivos que se relacionan con el desarrollo móvil se adoptará como tecnología el framework Flutter por varias razones entre las que se encuentran la amplia disponibilidad de librerías y documentación, la posibilidad de generar aplicaciones multiplataforma -aunque el desarrollo quedará instanciado para la plataforma Android- y la experiencia previa en el uso del framework por parte de los directores del proyecto.

1.4.2 Contexto

El proyecto se enmarca en la aplicación de Tecnologías de la Información y la Comunicación (TICs) en el sector agropecuario, abordando la necesidad de evaluar con precisión la nutrición del rodeo lechero a través del BCS. Este proceso se realiza tradicionalmente en un lugar llamado "manga", donde cada vaca es inspeccionada visualmente por un profesional veterinario, denominado "Calificador", quien asigna un score basado en su experiencia.

1.4.3 Desafíos actuales

Uno de los principales desafíos es la escasez de calificadores expertos, lo que a menudo lleva a que las evaluaciones no se realicen o, cuando se llevan a cabo, se tornen propensas a errores debido a la alta carga de trabajo, especialmente en rodeos grandes. Además, la precisión de la evaluación puede variar según el estado del calificador, lo que introduce un margen de error adicional.

1.4.4 Solución propuesta

La aplicación móvil abordará estos desafíos mediante las siguientes funcionalidades:

- Predicción del BCS mediante IA: Utilizando un modelo de red neuronal entrenado con imágenes de vacas, la aplicación podrá clasificar automáticamente el BCS de cada animal

con un score similar al que otorgaría un calificador humano. Este modelo ya ha sido desarrollado y entrenado en trabajos previos para vacas de , pero requiere un proceso continuo de reentrenamiento para mejorar su precisión y adaptabilidad.

- **Reentrenamiento automático del modelo:** La aplicación permitirá almacenar las calificaciones manuales realizadas por expertos, creando una base de datos de referencia que se utilizará para reentrenar el modelo de red neuronal. De este modo, cada nueva evaluación contribuirá a mejorar la precisión del algoritmo, permitiendo su adaptación a distintas condiciones y características del ganado.
- **Facilitación de uso por veterinarios:** La herramienta será accesible y fácil de usar por cualquier veterinario, incluso aquellos que no sean calificadores expertos. Esto no solo democratizará el uso de la tecnología, sino que también garantizará que las evaluaciones del BCS puedan realizarse con mayor frecuencia y precisión, sin depender exclusivamente de la disponibilidad de un calificador.

Resultados esperados

- **Automatización y precisión:** Al automatizar la predicción del BCS, se reducirá la dependencia de calificadores humanos y se minimizarán los errores derivados del cansancio o la subjetividad.
- **Mejora continua:** La posibilidad de reentrenar el modelo con nuevas evaluaciones permitirá una mejora continua del algoritmo, adaptándose mejor a las variaciones individuales del ganado y las condiciones específicas de cada rodeo.
- **Accesibilidad y eficiencia:** Veterinarios no especializados podrán realizar evaluaciones precisas del BCS, optimizando el tiempo y los recursos invertidos en el manejo nutricional del rodeo. Este prototipo no sólo permitirá optimizar la gestión nutricional en el sector agropecuario, sino que también contribuirá a la evolución de las prácticas tradicionales hacia un enfoque más tecnológico y eficiente.

Trabajos relacionados

2.1 Importancia y avances de la IA para calcular BCS

El procesamiento de imágenes es un paso clave para extraer características relevantes para la evaluación del BCS. Tradicionalmente, la extracción de características visuales se ha realizado mediante métodos manuales, en combinación con técnicas estadísticas como el Análisis de Componentes Principales (PCA) Kurita [2019]. El PCA es un método que transforma un conjunto de características originales en un nuevo conjunto de variables no correlacionadas llamadas componentes principales, reduciendo así la dimensionalidad de los datos. Sin embargo, el uso del PCA en este contexto, al igual que otros enfoques lineales, tiene limitaciones importantes. Al depender de un número limitado de puntos de características y muestras, el PCA puede perder información crucial en escenarios más complejos, lo que resulta en modelos predictivos menos eficientes y precisos Fischer et al. [2015]. Para abordar estos desafíos, el aprendizaje profundo basado en redes neuronales artificiales ha demostrado ser una herramienta poderosa para la modelización predictiva; las técnicas populares utilizadas para analizar imágenes incluyen las redes neuronales convolucionales (CNN). La estructura de CNN imita una red neuronal biológica y se aplica fácilmente a la identificación de imágenes visuales Singh et al. [2016]. En la agricultura ganadera, las CNN se han aplicado ampliamente a la clasificación e identificación de imágenes en áreas como el reconocimiento de comportamientos animales, la monitorización del estado de salud y la identificación de individuos Kamilaris and Prenafeta-Boldú [2018]. En este sentido, las CNN también han sido empleadas para la estimación del BCS de vacas a partir de imágenes de profundidad, como se demuestra en Rodríguez Alvarez et al. [2018]. Este trabajo destaca el uso de imágenes tridimensionales y técnicas de aprendizaje profundo para lograr evaluaciones precisas del BCS, mejorando así el proceso de monitoreo del bienestar animal.

Además de los avances en inteligencia artificial para la evaluación del BCS, es crucial considerar el papel de las tecnologías de soporte que facilitan el procesamiento de datos en entornos

distribuidos y con recursos limitados, como es el caso de Dew Computing, un paradigma que se presenta como una alternativa sustentable para la ejecución de aplicaciones IoT en áreas rurales y de conectividad limitada.

2.2 Importancia de dew computing

Dew Computing Skala et al. [2015] Rindos and Wang [2016] nace como un paradigma de computación posterior a Edge/Fog Computing que se presenta como una alternativa sustentable para soportar aplicaciones IoT. Puede operar acoplada o desacoplada de los servicios de ejecución que ofrece el Cloud Gusev and Wang [2018]. En el segundo caso, asociado a lo que puede ser una falta de conectividad, se explotan recursos presentes en el contexto local al usuario, también denominados “on premises resources” dentro de los cuales se encuentran dispositivos de usuario como smartphones o single board computers (SBC). Las capacidades de cómputo no despreciables que ofrecen hoy en día estos dispositivos posibilitan la ejecución de software recurso-intensivo. Existen ejemplos donde estos dispositivos consituyen el principal soporte de ejecución de software complejo en ámbitos profesionales Prego et al. [2018] Singh et al. [2017] Aquino et al. [2018]. Incluso cuando un único de estos dispositivo carezca de los recursos necesarios para cumplir con la ejecución de una tarea en el tiempos que se espera, aparecen trabajos en la literatura que plantean enfoques de ejecución colaborativos, donde una tarea se paraleliza y ejecuta en grupos de dispositivos Hirsch et al. [2018]. La eficiente utilización de un grupo de dispositivos en un contexto local plantea desafíos a nivel plataforma, tal como el estudio de componente de planificación de tareas, que es necesario seguir estudiando dada la heterogeneidad en términos de eficiencia energética y prestaciones que puede encontrarse de los dispositivos de usuario Hirsch et al. [2017] Hong and Varghese [2019]. Es una práctica común y aceptada en el estudio de sistemas distribuidos la experimentación mediante simulación Hirsch et al. [2018] Apolónia et al. [2017] Hirsch et al. [2020]. Esto es dada la inherente complejidad de configuración, costos monetarios y tiempo que conlleva estudiar estrategias de planificación en sistemas distribuidos montados sobre ambientes de ejecución reales. En relación a esto, DewSim Hirsch et al. [2020] permite modelar clusters de dispositivos móviles y evaluar estrategias de planificación de tareas teniendo en cuenta aspectos que tienen que ver con el consumo de energía de las baterías y uso de CPU de los dispositivos. Si bien la simulación resulta de suma utilidad para estudiar de forma exploratoria nuevas estrategias de planificación de tareas en ambientes Dew, es relevante contar con datos extraídos de la ejecución de aplicaciones Dew reales, ya se sea para validar los resultados obtenidos mediante simulación como para retroalimentar el modelo de entidades que forman parte de DewSim.

2.3 Comparaciones

En el campo de la gestión ganadera, varios estudios recientes han abordado la automatización y optimización de procesos clave en áreas como la alimentación, bienestar animal, salud y reproducción del ganado.

- En Çevik [2020] se usa BCS para agrupar y mejorar las tareas de cuidado y alimentación de los animales en determinados momentos. El objetivo es proporcionar una herramienta de fácil acceso a los productores de lácteos con la ayuda del software móvil diseñado. Se desarrolló un software para estimar los valores de BCS utilizando imágenes en tiempo real en dispositivos móviles Android llevándose a cabo con la ayuda de arquitecturas CNN que fueron entrenadas con imágenes tomadas por un software y evaluadas por un experto de manera offline. La red entrenada se transfirió al software móvil para utilizar el modelo entrenado y realizar predicciones en tiempo real. El software posee el modelo entrenado para realizar la toma de la foto y predecir el valor de BCS. De esta manera, no necesita acceso a cloud.
- Alonso et al. [2020]: Desarrollo de una plataforma diseñada para monitorear, rastrear y optimizar las tareas de gestión de explotaciones mixtas de cultivo y ganadería. La tecnología integra visualización de datos, un sistema de alertas, un mapa interactivo de dispositivos IoT y una interfaz para configurar sensores. Además, registra y almacena datos biométricos del ganado lechero, mostrando su actividad y un historial de fiebre, combinando temperatura, ritmo respiratorio y frecuencia cardíaca.
- En Piedade et al. se propone una red de sensores inalámbricos (WSN) con nodos LoRa, MongoDB y comunicaciones de largo alcance para monitorear la salud del ganado en hatos lecheros. Los nodos LoRa recopilan datos y los transmiten a través de un LoRa Gateway, validando la viabilidad del sistema. Este enfoque permite generar alertas sobre la salud del ganado, detectar estro y prevenir robos. El sistema convierte los datos recibidos por el gateway a JSON y los almacena en MongoDB. Luego, los datos procesados se publican a través de un broker MQTT, permitiendo que aplicaciones móviles accedan y presenten los resultados en un dashboard externo.
- En Yang et al. [2017] se desarrolló un sistema de monitoreo usando cámaras infrarrojas para registrar imágenes nocturnas de vacas lecheras. El procesamiento de imágenes identifica comportamientos de celo, facilitando que los propietarios tomen decisiones sobre la inseminación artificial. Utilizando análisis de energía dinámica de imágenes y procesamiento de imágenes binarias, pueden identificar cambios en la longitud de las vacas durante el comportamiento de montaje específico del estro. Los eventos que se ajusten a nuestra regla se grabarán como videos y se guardarán en la computadora. El prototipo se realizó con una sola cámara pero sugiere utilizar una cámara nocturna de

Trabajo	Objetivos de monitoreo	Rol de la app móvil	Computación en el Edge	Usa deep Learning	Dispositivos/ ámbito de sensado de datos
Çevik [2020]	Alimentación, Bienestar animal	Toma de imágenes y soporte para inferencia de BCS	Completamente	Si	Cámara del móvil, Cualquier lugar
Piedade et al.	Salud Animal, Reproducción	Visualización de resultados	Completamente	no	Red de Sensores (Collar con nodo ESP32), Campo abierto
Alonso et al. [2020]	Bienestar animal, Alimentación	Visualización de resultados	Parcial	Si	Red de Sensores (Collar, Sensores de ambiente (termómetro, higrómetro, gas), Estaciones meterológicas), Campo abierto/Corral
Yang et al. [2017]	Reproducción	N/A	Completamente	Si	Cámara infrarroja, Corral

Table 2.1: Uso de dispositivos móviles e IA en tareas de ganadería

ojo de pez a futuro para registrar actividades en un área más amplia de la granja lechera, superando la limitación de cobertura de una sola cámara. Esto evitaría la necesidad de combinar imágenes de múltiples cámaras y supera los problemas de los sensores de contacto directo utilizados en otras propuestas, que pueden ser perjudiciales para las vacas. Los eventos que se ajusten a la regla establecida se grabarán como videos y se guardarán en la computadora. Estos eventos serán evaluados más adelante por los propietarios. Por lo tanto, no es necesario acceso a internet.

Estos trabajos han explorado diversas tecnologías para mejorar la captura de datos, el análisis y la toma de decisiones, utilizando aplicaciones móviles, inteligencia artificial y diferentes contextos y modos de ejecución. A continuación, se presenta una comparación que sintetiza estos aspectos en los principales trabajos relacionados, destacando no solo el uso de inteligencia artificial para la evaluación de la Condición Corporal (BCS), sino también su aplicación en otras áreas críticas de la ganadería, así como el uso de distintos dispositivos, técnicas de sensado y contextos de ejecución.

2.3.1 Limitaciones halladas en trabajos relacionados

Si bien los trabajos revisados ofrecen soluciones tecnológicas interesantes para el sector agro-ganadero, presentan algunas limitaciones importantes. En primer lugar, no se observa un sistema que proporcione asistencia en tiempo real para el apartamiento o agrupamiento de vacas utilizando el cómputo del BCS y cámaras estratégicamente colocadas. La dependencia de la captura manual de imágenes en Çevik [2020] por parte del usuario introduce márgenes de error que pueden afectar la precisión de las predicciones, además de aumentar la carga de trabajo.

En segundo lugar, aunque algunos trabajos como Çevik [2020] calculan el BCS en tiempo real sin depender de la nube, otros, como Alonso et al. [2020], utilizan técnicas de inteligencia artificial que requieren procesamiento en la nube para obtener resultados avanzados. Aunque Alonso et al. [2020] almacena estadísticas y datos brutos en el edge, el procesamiento avanzado de IA y la visualización de resultados dependen de la nube. Esto puede ser una limitación en entornos rurales con conectividad inestable, ya que las funcionalidades más complejas del sistema no estarán disponibles sin conexión a internet.

En cuanto al historial del BCS, ninguno de los trabajos revisados ofrece un sistema que almacene y mantenga registros continuos de las predicciones de BCS, ni en el edge ni en la nube. Aunque Alonso et al. [2020] mantiene algunas estadísticas en el edge, los cálculos y resultados avanzados de IA solo están disponibles a través de la nube, lo que restringe el análisis profundo en el edge, dónde solo se puede acceder a datos no procesados.

Además, tanto en el trabajo de Piedade et al. como en el de Yang et al. [2017], la interpretación de los datos y el análisis de las imágenes capturadas dependen totalmente del usuario, ya que no se emplean técnicas de inteligencia artificial para automatizar estos procesos. Esto introduce una mayor carga de trabajo y puede ralentizar la toma de decisiones, aumentando el riesgo de interpretaciones subjetivas y potencialmente erróneas. Asimismo, en el caso de Piedade et al., el uso de collares en las vacas añade costos adicionales y posibles inconvenientes para los animales.

Por último, no se ha abordado en los trabajos revisados la posibilidad de mejorar de forma continua los modelos predictivos del BCS. La falta de un mecanismo que permita la retroalimentación del modelo mediante el etiquetado de imágenes y la actualización del sistema de predicción limita la personalización y ajuste del sistema a las características particulares de cada rodeo, algo fundamental para mejorar la precisión y eficiencia de las soluciones en el largo plazo.

A pesar de que existen propuestas que utilizan computación en el edge, aplicaciones móviles, cálculo del BCS, visualización de datos o automatización, no se observa ninguna que reúna todas estas características de forma integrada. Esta falta de integración limita la adaptabilidad de las soluciones a diferentes entornos y necesidades, lo que resalta la relevancia a la propuesta de este trabajo, que busca ofrecer una solución completa y robusta para abordar estas limitaciones en el sector agro-ganadero.

2.3.2 Mejoras de mi propuesta

El objetivo general de este proyecto es el de desarrollar un prototipo de ambiente de computación Dew que aproveche capacidad de cómputo presente en la cercanía de un usuario para ejecutar tareas de clasificación y etiquetado. Desde el punto de vista práctico, la capacidad de cómputo estará representada por dispositivos de bajo costo tipo SBC (Single Board Computers) y/o dispositivos móviles (smartphones) que podrán asumirse ya presentes en el contexto de utilización (tambo lechero) o en su defecto factibles de ser desplegados con mínimos requerimientos para llevar a cabo tareas de clasificación de ganado en tiempo real. Las tareas de clasificación propiamente dichas, emplearán un modelo tensorflow previamente desarrollado en el marco de un trabajo publicado en una revista internacional de alto impacto en el área de tecnologías aplicadas al agro Alvarez et al. [2018]. El prototipo a desarrollar tiene múltiples propósitos que pueden resumirse en:

1. Contribuir a mejorar el modelo tensorflow a partir de un reentrenamiento que se será posible gracias a la agilización del proceso de etiquetado de imágenes en las que interviene un profesional veterinario experto en el dominio.
2. Permitir que conocimiento experto en el dominio y encapsulado en una red de aprendizaje profundo pueda ser utilizado para automatizar la clasificación de ganado empleando dispositivos de usuario como plataforma de ejecución.
3. Permitir la validación de resultados de estrategias de planificación de tareas en grupos de dispositivos móviles estudiados con anterioridad en el grupo de trabajo y empleando simulación.

De este objetivo general se desprenden los siguientes objetivos específicos:

- Diseño y desarrollo de una app móvil Android Evaluador que oficia de cliente y permite al profesional veterinario experto en el dominio etiquetar imágenes de ganado vacuno tomadas con una cámara de profundidad en tiempo real en el contexto de labores realizados en una manga, sin requerir de conexión a Internet.
- Diseño y desarrollo de una app móvil Android Visualizador que permite a un usuario no experto en el dominio conocer la condición corporal del ganado vacuno a medida que se obtienen imágenes de este en tiempo real y en el contexto de labores realizadas en una manga, sin requerir de conexión a Internet.
- Diseño y desarrollo de un componente servidor encargado de la gestión de tareas que pueden ser de etiquetado o clasificación automática diferenciando la app cliente de la cual provenga una petición.

- Diseño y desarrollo de un módulo de generación de reportes que permitirá tanto a observadores como a calificadores visualizar reportes gráficos del BCS calculada por el algoritmo de predicción.

Requerimientos

El sistema debe ofrecer una solución para automatizar el cálculo del índice BCS del ganado, a través de la aplicación observador y mejorar dicho índice con la ayuda de la aplicación calificador. . El sistema integra tecnologías de IoT, deep learning y una infraestructura de computación Dew. A continuación se detallan los requerimientos claves.

3.1 Requerimientos funcionales

- **Calidad de Imágenes y Datos de Entrenamiento:** La aplicación debe ser capaz de capturar y procesar imágenes de alta calidad de vacas de diferentes razas lecheras, más allá de la raza Holstein. Estas imágenes deben ser etiquetadas con valores de BSC proporcionados por expertos en el lugar, quienes puntuarán de manera independiente para minimizar la subjetividad. Estos datos constituyen el ground truth necesario para entrenar, rediseñar y validar los modelos de predicción del BSC.
- **Capacidad Predictiva y Almacenamiento de Datos:** Una vez que los modelos de predicción del BSC estén operativos, la aplicación debe alojar estos algoritmos en dispositivos computacionales que se integren al aparejo móvil. Estos dispositivos deben ser capaces de ejecutar los algoritmos en tiempo real, utilizando computación Dew para reducir la dependencia de la conectividad a internet. El servidor central debe almacenar datos de los usuarios, sesiones de calificación, BSC asignadas por expertos, y resultados de los algoritmos predictivos, asegurando la disponibilidad de esta información para la generación de reportes.
- **Aplicación Móvil y Perfiles de Usuario:** La aplicación móvil debe soportar dos perfiles de usuario: calificadores y observadores. Los calificadores, generalmente expertos, deben poder realizar la calificación del BSC de cada vaca y visualizar reportes detallados. Los observadores, en cambio, tendrán acceso al cálculo automatizado de BCS -resultado de

una inferencia- y a la visualización de reportes, los cuales se actualizan en tiempo real según los últimos datos procesados. Además, los dispositivos utilizados por los observadores podrán participar de la ejecución del algoritmo predictivo, informando tanto a otros observadores como al servidor central para el almacenamiento y uso de estos datos en reportes.

- **Interfaz de Comunicación y Reportes:** La aplicación debe incluir un módulo para la generación de reportes en formato gráfico, como diagramas de torta, que permita a los usuarios visualizar la distribución del BSC en diferentes rangos. Estos reportes deben ser accesibles para diferentes sesiones de calificación (actual, última finalizada, y anteriores) y se actualizarán en tiempo real. La interfaz de usuario debe ser intuitiva, permitiendo la fácil navegación entre las diferentes opciones de visualización.
- **Interfaz de captura de datos:** El prototipo incluirá una interfaz de entrada diseñada para recibir las imágenes capturadas de las vacas y su correspondiente identificación. Las imágenes serán tomadas automáticamente por la cámara ubicada en la manga mientras las vacas pasan debajo de esta. Simultáneamente, la identificación de cada vaca se obtendrá mediante un sistema RFID, asegurando que cada imagen esté correctamente asociada al animal correspondiente. Esta interfaz permitirá un flujo de datos continuo y preciso, facilitando la posterior clasificación y análisis del BSC en tiempo real.
- **Autorización:** definirá los permisos de cada usuario según su rol dentro del sistema, diferenciando entre calificadores y observadores.
 - **Calificadores:** Pueden iniciar nuevas sesiones de calificación, etiquetar imágenes, y acceder a los reportes generados por las sesiones que han realizado.
 - **Observadores:** Pueden visualizar los reportes generados, consultar el historial de sesiones, y acceder tanto a la predicción en tiempo real del BCS como al historial asociado de cada vaca. Sin embargo, no tienen permisos para realizar calificaciones.

3.2 Casos de uso

3.2.1 Login

La interfaz de usuario permite ingresar el nombre de usuario para iniciar sesión sin necesidad de una contraseña. Durante el proceso de registro, los usuarios tienen la opción de seleccionar su rol, ya sea como calificador u observador.

El flujo de creación de usuario comienza desde la página de usuario en la aplicación móvil. Los usuarios crean una nueva cuenta ingresando su nombre y seleccionando su rol. Este proceso garantiza que el nombre de usuario sea único y que el usuario sea identificado correctamente

como calificador u observador. Tras la creación de la cuenta, los usuarios inician sesión ingresando simplemente su nombre, lo que les permite acceder a las funcionalidades específicas de su rol.

El sistema clasifica a los usuarios según su rol, lo que determina las funciones y vistas a las que tienen acceso. Los calificadores pueden realizar nuevas calificaciones y etiquetar imágenes, mientras que los observadores tienen acceso a la visualización de reportes y predicciones en tiempo real, sin la posibilidad de modificar datos. Esta diferenciación asegura que cada usuario interactúe con el sistema de acuerdo con sus funciones específicas, proporcionando una experiencia personalizada.

La sesión de usuario es fundamental para asociar las calificaciones realizadas a la cuenta del usuario. Esto permite al sistema rastrear y organizar las calificaciones, así como gestionar el historial de reportes y predicciones, mostrando sólo la información relevante para cada usuario.

3.2.2 Sesión de calificación

Las sesiones de calificación son esenciales para gestionar y registrar datos que forman parte del ground truth para el entrenamiento de un modelo de aprendizaje profundo. Durante una sesión de calificación, un experto, conocido como calificador, evalúa de a una por vez, un grupo de vacas usando la aplicación móvil. En estas sesiones, uno o más calificadores participan en la evaluación de cada vaca, y sus calificaciones se almacenan para su uso posterior, como en el entrenamiento de modelos de predicción.

La gestión de las sesiones de calificación es clave para garantizar que las evaluaciones se registren y organicen adecuadamente. Un calificador inicia una sesión en una locación (campo) específica, y se le asocia automáticamente a esa sesión. No se permite la existencia de más de una sesión activa en la misma locación, aunque es posible que varios calificadores participen en una misma sesión.

Las sesiones de calificación se almacenan con un identificador único y registran las fechas de inicio y finalización, lo que es crucial para generar reportes detallados sobre las evaluaciones realizadas.

Los casos de uso incluyen:

- **Inicio de Sesión de Calificación:** Un calificador inicia una sesión en una locación específica y se asocia automáticamente a esta sesión.
- **Finalización de Sesión de Calificación:** Un calificador finaliza una sesión previamente iniciada, marcando el cierre de la sesión en el sistema.
- **Unirse a una Sesión de Calificación:** Un calificador puede unirse a una sesión activa en su locación si no ha iniciado su propia sesión. Aunque la aplicación móvil actual no permite unirse a sesiones activas, el sistema está diseñado para manejar esta funcionalidad, permitiendo su extensión futura.

- **Locación de la Sesión de Calificación:** La administración de locaciones no está implementada en la aplicación móvil, aunque el sistema soporta estas opciones para facilitar futuras mejoras.

3.2.3 Sistema de notificación de imágenes

El sistema recibe imágenes de vacas y procesa la información asociada para la calificación y visualización del BCS. El flujo comienza cuando las imágenes de cada vaca, junto con su metadata son transmitidas al servidor. El servidor, que cuenta con una interfaz de recepción de datos, es responsable de recibir y procesar esta información.

- **Sistema de Envío:** Este componente es responsable de transmitir las imágenes de las vacas al servidor, junto con la identificación de la vaca, posición en la caravana y conjunto de imágenes. El sistema de envío incluye un sistema RFID que captura automáticamente la identificación de la vaca cuando pasa por la manga, sincronizando esta información con la captura de la imagen. De esta manera, se asegura que cada imagen esté correctamente asociada a la vaca correspondiente. Cabe destacar que el sistema de captura de imágenes, queda fuera del alcance de este proyecto.
- **Recepción y Almacenamiento de Imágenes:** Las imágenes y su meta-data asociada se reciben en el servidor donde son almacenadas. Esta interfaz permitirá un flujo de datos continuo y preciso, facilitando la posterior clasificación y análisis del BSC en tiempo real.
- **Gestor de mensajes de imágenes:** Una vez que las imágenes y su meta-data han sido almacenadas, se genera una notificación interna entre el servidor y la aplicación móvil que permite un flujo continuo de imágenes hacia los calificadores y observadores. Los calificadores pueden acceder a las imágenes solicitando la siguiente disponible dentro de su sesión activa, mientras que los observadores reciben automáticamente las imágenes en tiempo real, acompañadas de los metadatos correspondientes.

3.2.4 Calificadores

Los calificadores son expertos especializados en la evaluación del BCS, ya sea a partir de imágenes recibidas o mediante la observación directa de las vacas en la manga. Con base en su experiencia y conocimiento, asignan una puntuación de BCS a cada vaca evaluada, la cual es almacenada. Al ingresar el puntaje, este se asocia tanto a la vaca como a la sesión de calificación correspondiente, lo que garantiza una gestión de datos organizada y trazable. Esta información resulta fundamental para entrenar y mejorar continuamente los modelos predictivos de BCS en futuras iteraciones del sistema. Además de asignar las puntuaciones, los calificadores tienen la capacidad de visualizar reportes.

3.2.5 Observadores

Los usuarios con rol observador pueden visualizar la predicción del BCS y también tendrán acceso a diversos reportes, por ejemplo, evolución en el tiempo del BCS de un animal e historial de BCS de un rodeo. Los dispositivos de usuarios observadores por defecto participan de la ejecución del algoritmo de predicción del BCS aportando recursos computacionales, acción que no afecta la experiencia de usuario con la app y cuyo resultado se distribuirá automáticamente por el sistema entre otros observadores presentes.

3.2.6 Reportes

El sistema de reportes proporciona a calificadores y observadores acceso a la visualización de datos sobre las evaluaciones del BCS predichas por el algoritmo de predicción. Estos reportes se presentan en diagramas de torta que agrupan las vacas dentro de los siguientes rangos de puntaje:

- 0 a 0.99
- 1 a 1.99
- 2 a 2.99
- 3 a 3.99
- 4 a 5
- Sin predicción

Los casos de uso incluyen:

- Visualización de Reportes en Sesiones Actuales:
 - Los observadores pueden acceder a reportes que se actualizan automáticamente durante la sesión de calificación en curso. Los valores de BCS generados por el algoritmo se reflejan en los diagramas de torta en tiempo real, lo que permite un monitoreo continuo del estado de las vacas.
 - Los calificadores tienen la opción de visualizar los reportes cuando lo consideren necesario. Sin embargo, dado que su función principal es la evaluación directa de las vacas, los diagramas se actualizan solo cuando el calificador decide consultar esta información, asegurando que accedan a los datos más recientes en ese momento.
- Acceso a Reportes de Sesiones Anteriores: Los usuarios pueden acceder a los reportes de sesiones de calificación terminadas, donde pueden consultar los puntajes BCS registrados durante esas sesiones.

- **Historial BCS para Observadores:** Los observadores, además de ver los reportes actuales, pueden acceder al historial de predicciones BCS de vacas en sesiones de calificación anteriores, lo que les permite tener una visión más amplia del progreso o cambios en la condición corporal de las vacas a lo largo del tiempo.

Diseño e implementación

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 ubicación 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.

4.1 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

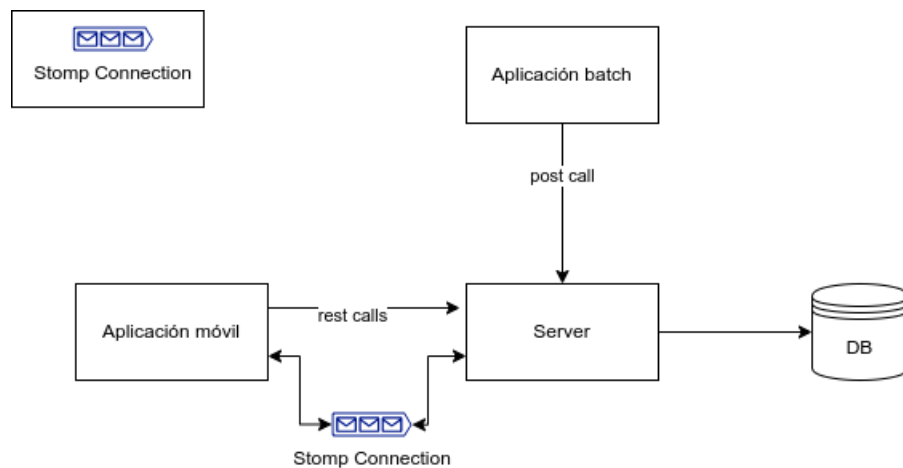


Figure 4.1: Componentes del sistema

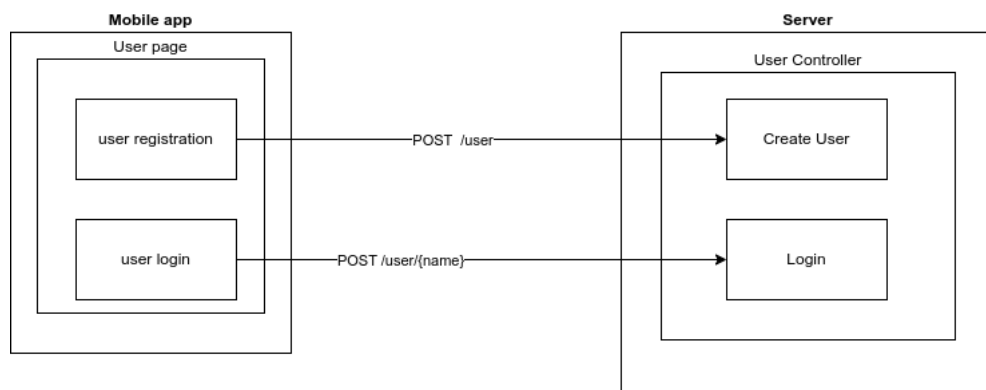


Figure 4.2: Diagrama de componentes - login

la identidad del usuario únicamente a través de su nombre, sin necesidad de una contraseña. Una vez identificado, el 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.

4.2 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

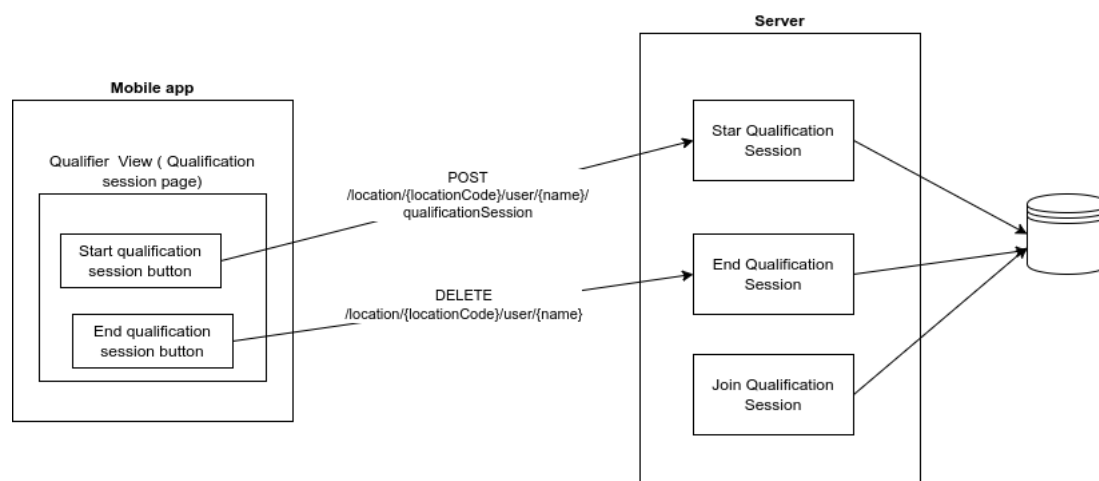


Figure 4.3: Diagrama de componentes - sesión de calificación

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 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.

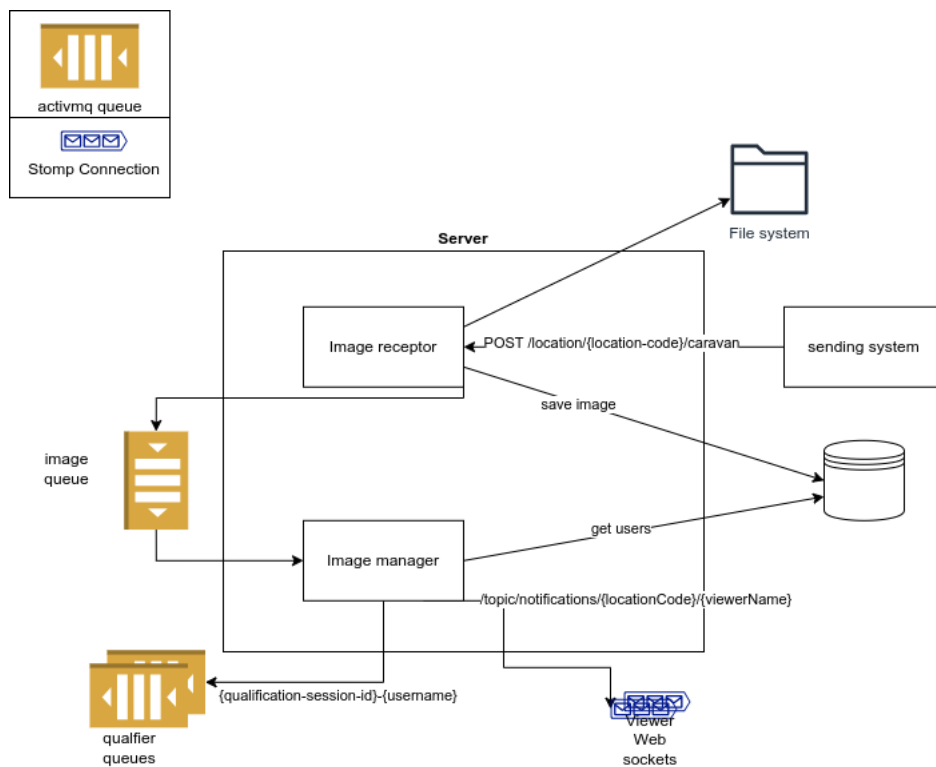


Figure 4.4: Diagrama de componentes - imágenes

- 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.

4.3 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.

4.3.1 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 los animales se registran al pasar por la manga. Este mecanismo asegura que las imágenes estén asociadas a la vaca correcta y 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 tomadas 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. 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.

4.3.2 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.

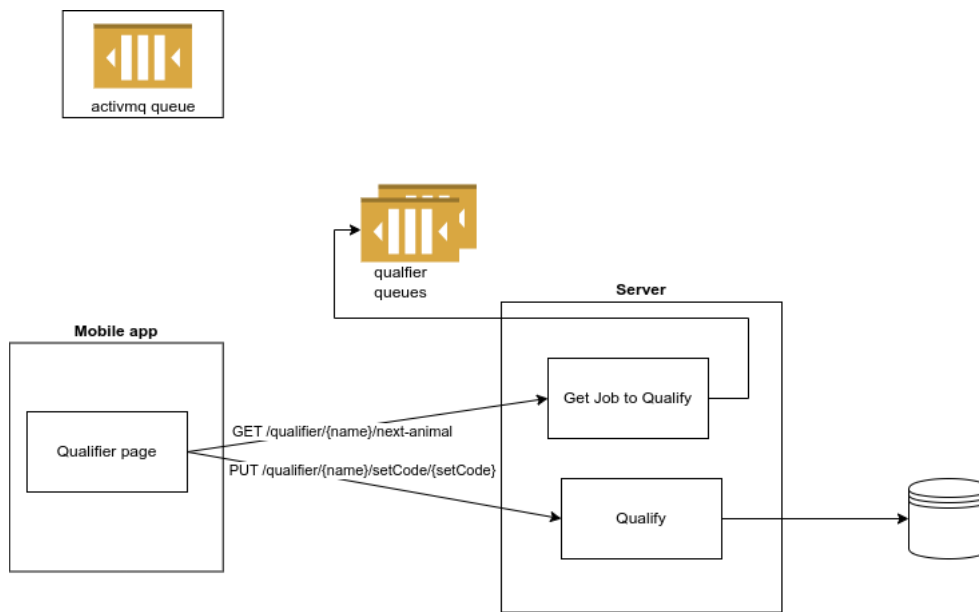


Figure 4.5: Diagrama de componentes: calificador

4.3.3 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.

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.

4.3.4 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

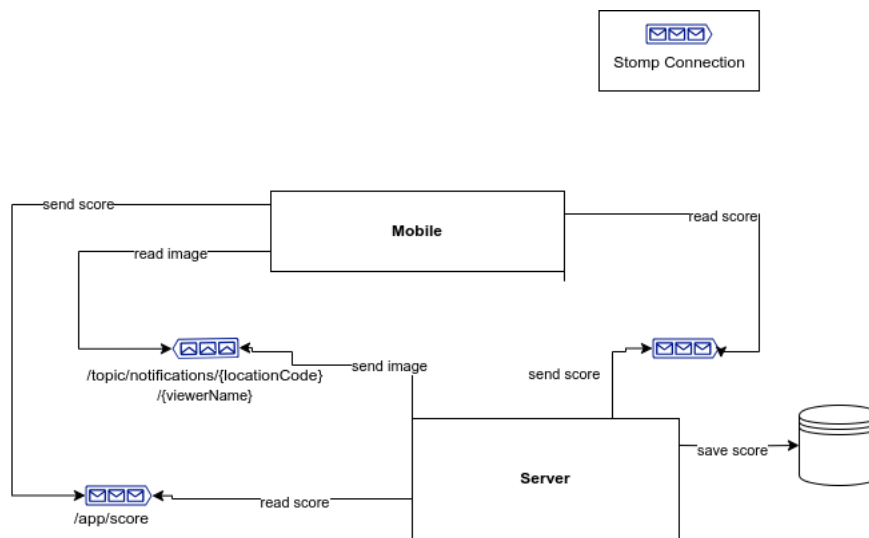


Figure 4.6: Diagrama de componentes - Observadores

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 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.

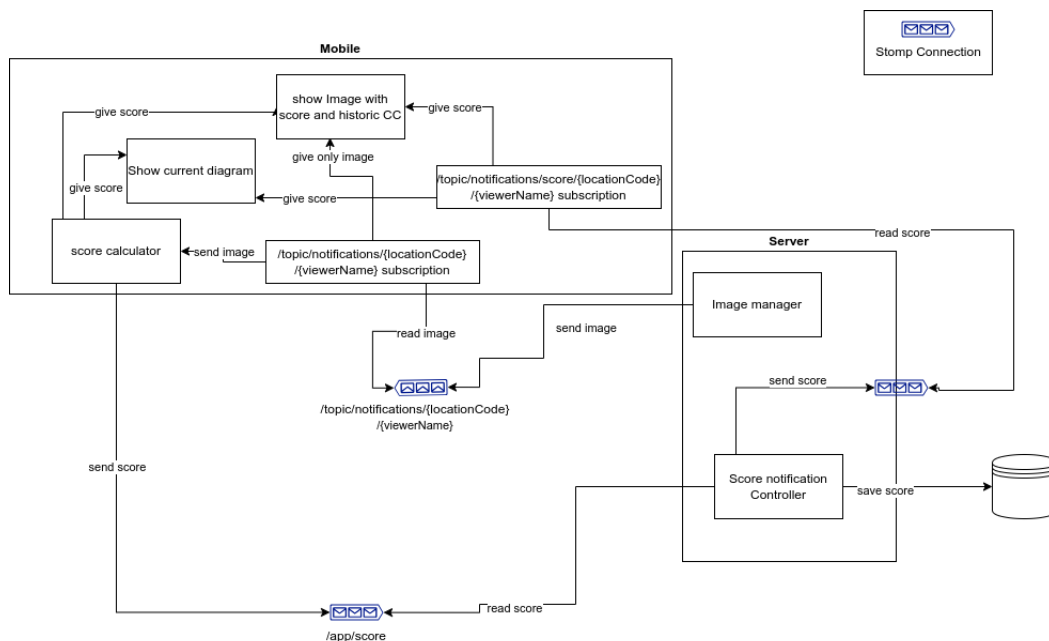


Figure 4.7: Diagrama de componentes - Observadores en detalle

- Los demás observadores conectados: Para que también puedan ver el puntaje obtenido.

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.

4.4 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 alcarar que los reportes 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

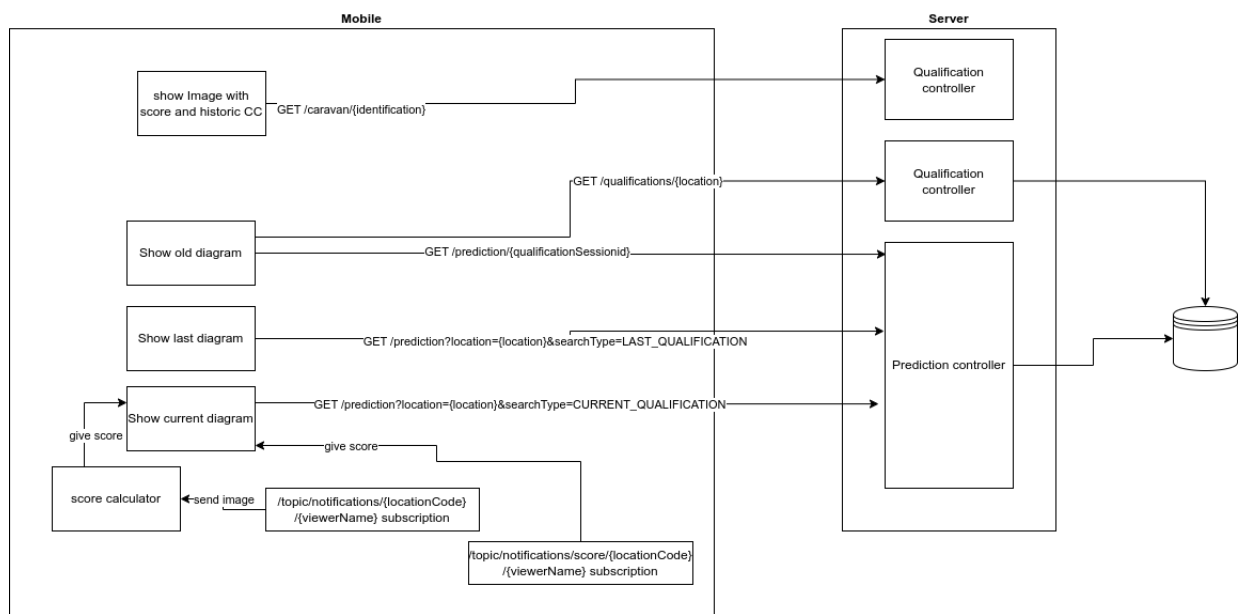


Figure 4.8: Diagrama de componentes - reporte de observadores

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.

4.4.1 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

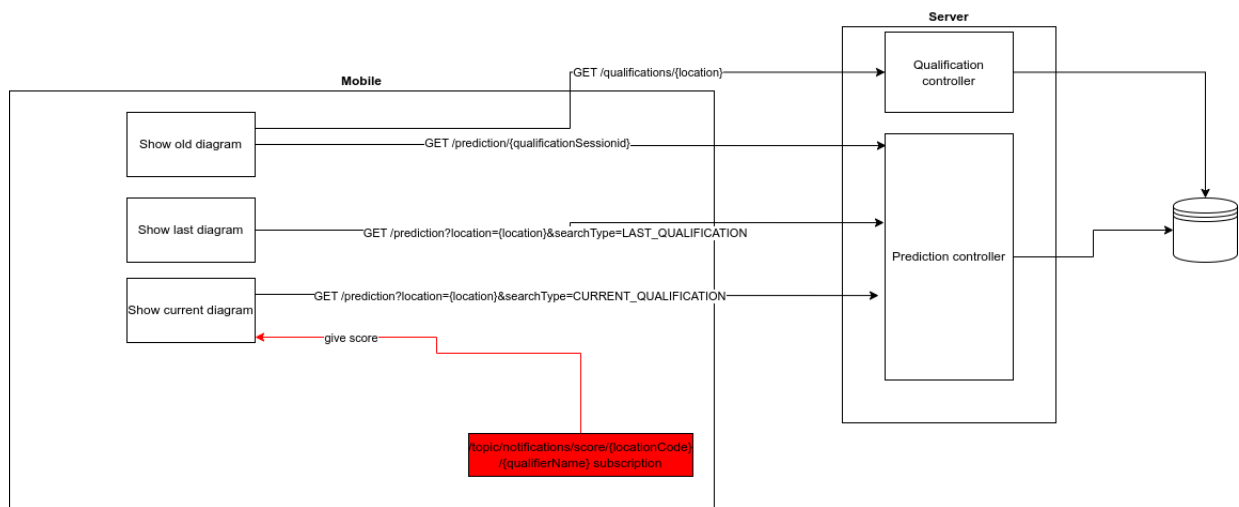


Figure 4.9: Diagrama de componentes - reporte de calificadores

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.

4.4.2 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.

4.5 Implementación

Esta sección describe las tecnologías, bibliotecas y configuraciones empleadas para desarrollar el prototipo. A continuación, se detallan los componentes principales, la configuración del sistema y el modelo de base de datos.

4.5.1 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.

4.5.2 Pasos para la configuración del sistema

Este apartado explica cómo se configuran los diversos componentes del sistema, incluidos el servidor, base de datos y las dependencias necesarias para que el sistema pueda ser desplegado correctamente.

4.5.2.1 Configuración del componente Servidor

1. Antes de proceder es necesario de tener instalado:
 - (a) Java 11: Es mandatorio para ejecutar la aplicación. La versión de Java instalada puede ser verificada ejecutando el siguiente comando: `java -version`
Parea instalarlo se puede hacer desde el sitio oficial de Oracle o mediante un gestor de paquetes como apt, brew, o sdkman.
 - (b) Maven: Sólo en caso de necesitar compilar usando el código fuente del proyecto. En ubuntu/Debian: `sudo apt install maven`. En macOS con Homebrew: `brew install maven`
2. Configuración de la base de datos MySQL
 - (a) Descargar e instalar MySQL 8.0 siguiendo las instrucciones del asistente.
 - (b) Iniciar el servidor MySQL: Desde consola en macOS o Linux ingresando el comando `mysql.server start`. En windows utilizando `net start mysql`
 - (c) Acceder a MySQL:
 - i. Abre el cliente de MySQL ingresando por consola
`mysql -u root -p`
 - ii. Ingresar la contraseña de root
 - (d) Crear la base de datos: Desde consola ejecutar el siguiente comando para crear el esquema necesario `CREATE SCHEMA server`
3. Instalación y configuración de mensajería mediante ActiveMQ Artemis
 - (a) Descargar e instalar desde el sitio oficial de ActiveMQ Artemis la última versión del broker

- (b) Descomprimir el archivo descargado en un directorio cualquiera.
- (c) Desde consola posicionado en el subdirectorio bin/ dentro de la carpeta de ActiveMQ Artemis ejecutar el siguiente comando para crear una nueva instancia del broker:

```
./artemis create mybroker --user admin --password admin
```

Esto creará una instancia de ActiveMQ con el nombre mybroker

- (d) Iniciar el broker desde el directorio mybroker/bin, ejecutando

```
./artemis run
```

El broker correrá por defecto en el puerto 6161

4.5.2.2 Configuración de la aplicación móvil.

En esta sección se detallará cómo configurar la aplicación móvil desarrollada con Flutter para interactuar con el sistema.

1. Configuración del entorno Python

La aplicación requiere Python 3.8 para generar correctamente el APK. Asegúrate de configurar el entorno Python antes de proceder.

- (a) Definir la ruta de Python 3.8 en el sistema:

- i. Para sistemas Linux:

```
export PYTHON_3_8_HOME='/home/linuxbrew/.linuxbrew/opt/  
python@3.8/bin '  
export PATH=$PATH:$PYTHON_3_8_HOME
```

- (b) Verificar la configuración:

- i. Asegúrate de que Python 3.8 esté configurado correctamente ejecutando:

```
python3.8 --version
```

- ii. Esto debe devolver algo como:

```
Python 3.8.x
```

2. Construcción del APK

- (a) Una vez configurado Python, ejecuta el siguiente comando desde la raíz del proyecto Flutter:

```
flutter build apk
```

Esto generará el archivo APK de la aplicación.

3. El APK se encontrará en:

`build/app/outputs/flutter-apk/`

4. Host y Puerto del Servidor: La configuración del host y puerto de servidor se ajusta en las variables `serverHost` y `serverPort`. Por defecto:

(a) `localhost` en iOS y `10.0.2.2` en Android para entornos de emulación.

(b) Puerto `8080` si no se especifica otro.

5. Configuración de escritura:

Para que la aplicación pueda acceder y escribir en el sistema de archivos, es necesario seguir estos pasos.

(a) Copiar archivo de imagen utilizado por el detector automático: `assets/empty_image.png` al directorio `sdcard/Download/` del emulador o dispositivo Android.

(b) Asegurarse de que la aplicación tenga permisos de escritura en el sistema de archivos del dispositivo. Esto se puede configurar desde los ajustes del emulador o del dispositivo Android en la opción de permisos de la aplicación calificador

4.5.2.3 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 que configura el broker de mensajería en el puerto 61616:

```
docker run --name=activemq -p 61616:61616 -e AMQ_USER=admin -e  
AMQ_PASSWORD=admin quay.io/artemiscloud/activemq-artemis-broker
```

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 ingresando:

```
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;
```

4.5.3 Modelo de base de datos

La base de datos utiliza el siguiente modelo relacional, implementado en MySQL para gestionar los datos de las sesiones de calificación y sus componentes.

Define las ubicaciones físicas en las que se realizan las calificaciones.

```
CREATE TABLE server.LOCATION ( ID INT AUTO_INCREMENT PRIMARY KEY,  
    CODE VARCHAR(20) , NAME VARCHAR(100) );
```

Algorithm 4.1: Script tabla LOCATION

Registra las sesiones de calificación realizadas en una ubicación específica, asociando cada sesión con su locación.

```
CREATE TABLE server.QUALIFICATION_SESSION (   
    ID BIGINT AUTO_INCREMENT PRIMARY KEY, LOCATION_ID INT NOT NULL,  
    START_DATE TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP, END_DATE  
    TIMESTAMP NULL, CONSTRAINT fk_QUALIFICATION_SESSION_LOCATION_ID  
    FOREIGN KEY (LOCATION_ID) REFERENCES LOCATION (ID) );
```

Algorithm 4.2: Script tabla QUALIFICATION_SESSION

Sesiones activas en cada locación.

```
CREATE TABLE server.LOCATION_QUALIFICATION_SESSION(  
    QUALIFICATION_SESSION_ID BIGINT, LOCATION_ID INT, CONSTRAINT  
    pk_LOCATION_QUALIFICATION_SESSION PRIMARY KEY(LOCATION_ID) ,  
    CONSTRAINT 'fk_LOCATION_QUALIFICATION_SESSION_LOCATION_ID' FOREIGN  
    KEY('LOCATION_ID') REFERENCES 'LOCATION'('ID') , CONSTRAINT '  
    fk_LOCATION_QUALIFICATION_SESSION_QUALIFICATION_SESSION_ID'  
    FOREIGN KEY('QUALIFICATION_SESSION_ID') REFERENCES '  
    QUALIFICATION_SESSION'('ID') );
```

Algorithm 4.3: Script tabla LOCATION_QUALIFICATION_SESSION

Almacena la información de los usuarios registrados en el sistema y asegura nombres de usuario únicos.

```
CREATE TABLE server.USER (
  ID INT AUTO_INCREMENT PRIMARY KEY, USER_NAME VARCHAR(100), TYPE
  VARCHAR(20), CONSTRAINT uk_USER_USER_NAME UNIQUE (USER_NAME) );
```

Algorithm 4.4: Script tabla USER

Usuarios asociados a la sesión de calificación.

```
CREATE TABLE server.USER_QUALIFICATION_SESSION( USER_ID INT,
  QUALIFICATION_SESSION_ID BIGINT, PRIMARY KEY(USER_ID,
  QUALIFICATION_SESSION_ID), CONSTRAINT '
  fk_USER_QUALIFICATION_SESSION_QUALIFICATION_SESSION_ID' FOREIGN
  KEY('QUALIFICATION_SESSION_ID') REFERENCES 'QUALIFICATION_SESSION
  '('ID')', CONSTRAINT 'fk_USER_QUALIFICATION_SESSION_USER_ID'
  FOREIGN KEY('USER_ID') REFERENCES 'USER'('ID')
);
```

Algorithm 4.5: Script tabla USER_QUALIFICATION_SESSION

Usuarios registrados en la sesión de calificación actual.

```
CREATE TABLE server.USER_LOCATION_QUALIFICATION_SESSION(
  USER_ID INT PRIMARY KEY, QUALIFICATION_SESSION_ID BIGINT, CONSTRAINT
  UNIQUE (USER_ID), CONSTRAINT '
  fk_USER_LOCATION_QUALIF_SESSION_USER_QUALIF_SESSION' FOREIGN KEY('
  USER_ID', 'QUALIFICATION_SESSION_ID') REFERENCES '
  USER_QUALIFICATION_SESSION'('USER_ID', 'QUALIFICATION_SESSION_ID'),
  CONSTRAINT '
  fk_USER_LOCATION_QUALIFICATION_SESSION_QUALIFICATION_SESSION_ID'
  FOREIGN KEY('QUALIFICATION_SESSION_ID') REFERENCES '
  LOCATION_QUALIFICATION_SESSION'('QUALIFICATION_SESSION_ID'),
  CONSTRAINT 'fk_USER_LOCATION_QUALIFICATION_SESSION_USER_ID'
  FOREIGN KEY('USER_ID') REFERENCES 'USER'('ID') );
```

Algorithm 4.6: Script tabla USER_LOCATION_QUALIFICATION_SESSION

Representa conjuntos de imágenes relacionados a una sesión de calificación en una locación. Esta tabla contiene un identificador único (SET_CODE) que agrupa las imágenes realizadas para una caravana específica, permitiendo su agrupamiento y trazabilidad.

```
CREATE TABLE IMAGE_SET (
  ID BIGINT AUTO_INCREMENT PRIMARY KEY, SET_CODE VARCHAR(36) UNIQUE,);
```

Algorithm 4.7: Script tabla IMAGE_SET

Conjuntos de imágenes actuales de una caravana en una locación.

```
CREATE TABLE server.IMAGE_SET_LOCATION(  
IMAGE_SET_ID BIGINT PRIMARY KEY, LOCATION_ID INT, CREATION_DATE  
TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP, 'POSITION' SMALLINT  
NOT NULL, CONSTRAINT uk_IMAGE_SET_LOCATION_POSITION UNIQUE( '  
LOCATION_ID', 'POSITION' ), CONSTRAINT  
fk_IMAGE_SET_LOCATION_LOCATION_ID FOREIGN KEY( 'LOCATION_ID' )  
REFERENCES 'LOCATION'( 'ID' ), CONSTRAINT  
fk_IMAGE_SET_LOCATION_SET_ID FOREIGN KEY( 'IMAGE_SET_ID' )  
REFERENCES 'IMAGE_SET'( 'ID' ), CONSTRAINT  
nn_IMAGE_SET_LOCATION_LOCATION_ID CHECK( 'LOCATION_ID' is NOT NULL  
) );
```

Algorithm 4.8: Script tabla IMAGE_SET_LOCATION

Conjuntos de imágenes de una sesión de calificación.

```
CREATE TABLE server.IMAGE_SET_QUALIFICATION_SESSION(  
IMAGE_SET_ID BIGINT PRIMARY KEY, QUALIFICATION_SESSION_ID BIGINT NOT  
NULL, CONSTRAINT fk_IMAGE_SET_QUALIFICATION_SESSION_SET_ID  
FOREIGN KEY( 'IMAGE_SET_ID' ) REFERENCES 'IMAGE_SET'( 'ID' ),  
CONSTRAINT 'fk_IMAGE_SET_QUALIFICATION_SESSION_ID' FOREIGN KEY( '  
QUALIFICATION_SESSION_ID' ) REFERENCES 'QUALIFICATION_SESSION'( 'ID'  
) );
```

Algorithm 4.9: Script tabla IMAGE_SET_QUALIFICATION_SESSION

Contiene una columna llamada prioridad que será enviada al calificador y al observador y el path se construye con el Directorio mas nombre del archivo.

```
CREATE TABLE server.IMAGE(  
ID BIGINT NOT NULL AUTO_INCREMENT PRIMARY KEY, IMAGE_SET_ID BIGINT  
NOT NULL, PRIORITY INT NULL COMMENT , 'PATH' VARCHAR(500) NULL  
UNIQUE COMMENT, CONSTRAINT 'fk_IMAGE_SET_ID' FOREIGN KEY ( '  
IMAGE_SET_ID' ) REFERENCES IMAGE_SET ( 'ID' ) );
```

Algorithm 4.10: Script tabla IMAGE

Puntaje para una caravana en una sesión dada definido por el algoritmo de predicción.

```
CREATE TABLE server.PREDICTED_SCORE(  
  IMAGE_SET_ID BIGINT NOT NULL PRIMARY KEY , SCORE DECIMAL(5, 4) NOT  
  NULL, QUALIFICATION_SESSION_ID BIGINT NULL, CONSTRAINT '  
  fk_PREDICTED_SCORE_SET_ID' FOREIGN KEY ( 'IMAGE_SET_ID') REFERENCES  
  'IMAGE_SET' ( 'ID' ) , CONSTRAINT '  
  fk_PREDICTED_SCORE_QUALIFICATION_SESSION_ID' FOREIGN KEY( '  
  QUALIFICATION_SESSION_ID') REFERENCES 'QUALIFICATION_SESSION'( 'ID  
  '));
```

Algorithm 4.11: Script tabla PREDICTED_SCORES

Guarda las puntuaciones otorgadas a los conjuntos de imágenes en cada sesión de calificación. La tabla QUALIFIED_SCORE almacena el puntaje definido por el calificador para una caravana en una sesión específica. La puntuación va de 1 a 5. Los valores de puntaje están restringidos para asegurar que se mantengan dentro de un rango de calificación válido.

```
CREATE TABLE server.QUALIFIED_SCORE (  
  USER_ID INT, QUALIFICATION_SESSION_ID BIGINT, IMAGE_SET_ID BIGINT  
  NOT NULL, SCORE SMALLINT NOT NULL, PRIMARY KEY (USER_ID,  
  QUALIFICATION_SESSION_ID, IMAGE_SET_ID), CONSTRAINT  
  fk_QUALIFIED_SCORE_USER_QUALIFICATION_SESSION FOREIGN KEY (  
  USER_ID, QUALIFICATION_SESSION_ID) REFERENCES  
  USER_QUALIFICATION_SESSION (USER_ID, QUALIFICATION_SESSION_ID),  
  CONSTRAINT fk_QUALIFIED_SCORE_SET_ID FOREIGN KEY (IMAGE_SET_ID)  
  REFERENCES IMAGE_SET (ID), CONSTRAINT  
  CHK_QUALIFIED_SCORE_SCORE_MIN CHECK (SCORE > 0), CONSTRAINT  
  CHK_QUALIFIED_SCORE_SCORE_MAX CHECK (SCORE <= 5) );
```

Algorithm 4.12: Script tabla QUALIFIED_SCORE

Resultados

En este capítulo, se presentan las capturas de pantalla y el análisis de las principales interfaces de usuario de cada caso de uso, proporcionando una evaluación del funcionamiento y la usabilidad de cada módulo en la aplicación. Adicionalmente, se proporciona información sobre el rendimiento del sistema, incluyendo métricas específicas relacionadas con el consumo de memoria y su impacto en la eficiencia operativa.

5.1 Pantalla de Inicio de Sesión (Login)

El módulo de inicio de sesión permite a los usuarios acceder al sistema con distintos roles, con diferentes permisos y funciones dentro de la aplicación. La pantalla principal de acceso ofrece dos opciones: "Iniciar sesión" y "Registrar usuario", de modo que el usuario puede elegir entre ingresar al sistema o registrarse como nuevo usuario, según sea necesario.

En Figura 5.1a, si el usuario selecciona "Iniciar sesión", es dirigido a una pantalla donde puede introducir su nombre de usuario (Figura 5.1b). En caso de un ingreso correcto, el sistema autentica el perfil y redirige al usuario a su módulo correspondiente, ya sea como calificador o como observador.

En cambio, si el usuario selecciona "Registrar usuario", el sistema muestra la pantalla que está en la Figura 5.1c donde el usuario puede introducir su nombre y seleccionar su rol (calificador u observador). Esta pantalla permite que los nuevos usuarios definan adecuadamente su perfil dentro de la aplicación. Una vez completado el registro, el sistema redirige automáticamente al usuario a la pantalla de inicio de sesión para facilitar el acceso.

La interfaz de inicio de sesión presenta mensajes de error en caso de datos incorrectos o problemas de conexión con el servidor. El sistema está configurado para gestionar el acceso según el rol del usuario, asegurando que cada perfil (calificador u observador) acceda exclusivamente a las funciones asignadas.

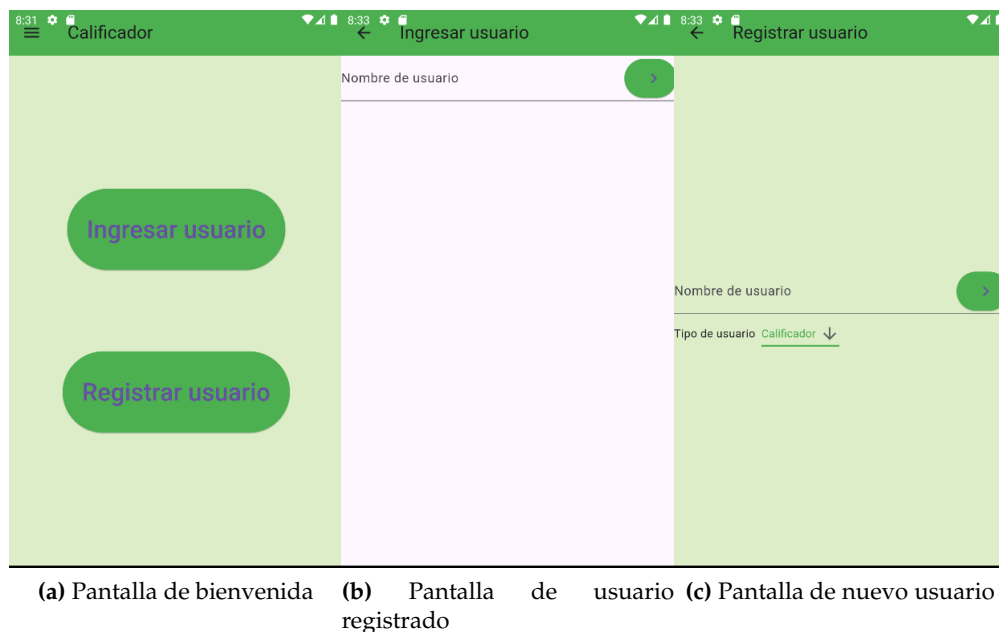


Figure 5.1: App móvil: caso de uso de login

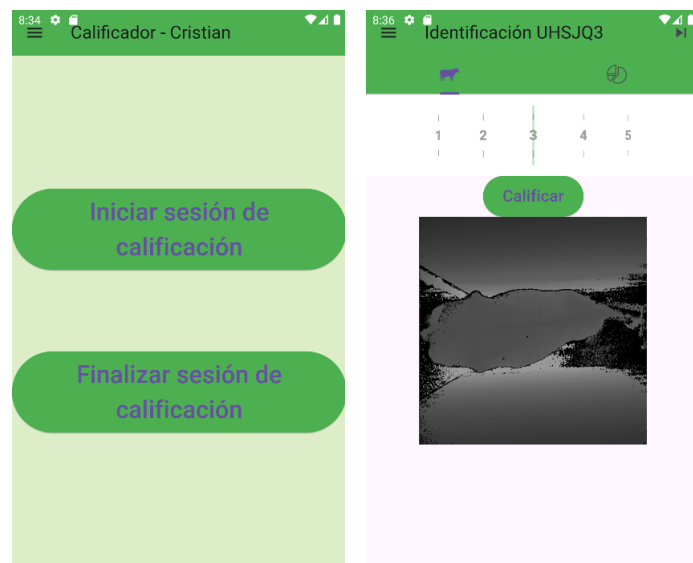
5.2 Calificador

El módulo de calificación permite al usuario con rol de calificador gestionar sus sesiones de calificación de manera eficiente como muestra 5.2a. Desde esta pantalla, el calificador puede iniciar una nueva sesión de calificación o finalizar una sesión previamente activa, asegurando que solo una sesión esté abierta a la vez.

Si el calificador intenta iniciar una nueva sesión mientras otra está activa, el sistema muestra mensajes de error claros e informativos, indicando la necesidad de finalizar la sesión actual antes de comenzar una nueva. Esta funcionalidad garantiza el control adecuado del flujo de trabajo del calificador, evitando conflictos o superposición de datos.

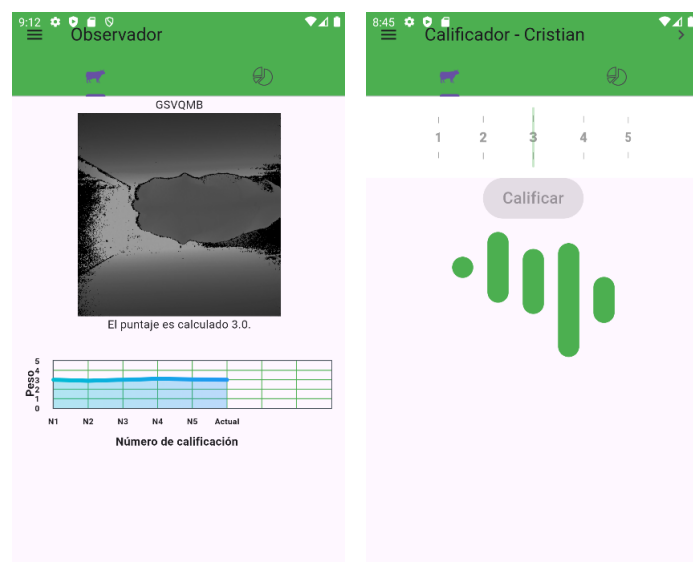
En el caso de iniciar sesión, la aplicación los redirige a la pantalla principal del calificador como se muestra en 5.2b.

Durante la sesión de calificación, los usuarios calificadores tienen la capacidad de asignar puntajes BCS a las vacas, una funcionalidad clave para garantizar la recopilación y el almacenamiento de datos de evaluación. Estos datos no solo respaldan el proceso de calificación actual, sino que también son esenciales para el entrenamiento y la mejora continua de futuros modelos predictivos.



(a) Iniciar/Finalizar sesión de calificación
(b) App móvil: pantalla calificador

Figure 5.2: App móvil: caso de uso calificación



(a) App móvil: Notificación de observador
(b) App móvil: arribo de imagen de calificador

Figure 5.3: App móvil: Arribo de imágenes

5.3 Sistema de Notificación de Imágenes

El módulo del observador, tras iniciar sesión, redirige automáticamente al usuario a la pantalla principal, diseñada para recibir y mostrar las imágenes a medida que son enviadas al sistema. Este módulo permite a los usuarios con el rol de observador visualizar las imágenes capturadas de las caravanas junto con su identificación asociada y un historial detallado de los puntajes de condición corporal (BCS) asignados en sesiones anteriores para cada caravana.

Además, una vez que se calcula el puntaje de condición corporal (BCS) predicho, ya sea por el dispositivo móvil de este observador u otro, el resultado se muestra en la misma pantalla. Luego de este cálculo se actualiza en tiempo real el gráfico del historial como se muestra en 5.3a.

La interfaz del observador es completamente de solo lectura, sin opciones para interactuar o realizar modificaciones. Las imágenes se actualizan de manera automática conforme las vacas pasan por debajo de la cámara, ofreciendo una experiencia fluida y enfocada en la visualización y monitoreo en tiempo real.

Por otro lado, el módulo del calificador redirige automáticamente al usuario, tras iniciar sesión de calificación, a su pantalla principal. Esta interfaz está diseñada para permitir que el calificador avance manualmente a la siguiente imagen de una vaca haciendo clic en el botón con el símbolo de > (Siguiendo) a la espera de la llegada de una imagen como se ve en 5.3b. Esto le da al calificador el control de cuándo visualizar la próxima imagen para asignar el valor BCS correspondiente. Es importante destacar que el calificador debe interactuar activamente con la aplicación para avanzar o esperar la llegada de una nueva imagen.

En la pantalla principal, ambos usuarios tienen acceso a un menú que les permite cerrar la sesión como se ve en 5.4. Además, para los calificadores, este menú incluye la opción adicional de finalizar una sesión de calificación en curso.

5.4 Reportes Generados

Tanto los calificadores como los observadores cuentan con herramientas visuales que facilitan el análisis de los datos generados durante las sesiones de calificación. Entre estas herramientas destaca un gráfico de tipo torta, que proporciona una representación clara y organizada de las sesiones en curso y las completadas. Este gráfico agrupa los valores predichos del puntaje de condición corporal (BCS) en categorías enteras del 1 al 5, basándose en la parte entera del valor incluso cuando la predicción incluye decimales. Esta metodología asegura una visualización uniforme, intuitiva y de fácil interpretación para todos los usuarios.

Adicionalmente, se incluye un diagrama de dispersión que muestra el promedio del puntaje de condición corporal (BCS) de las sesiones anteriores, ofreciendo una perspectiva comparativa y útil para evaluar tendencias a lo largo del tiempo.

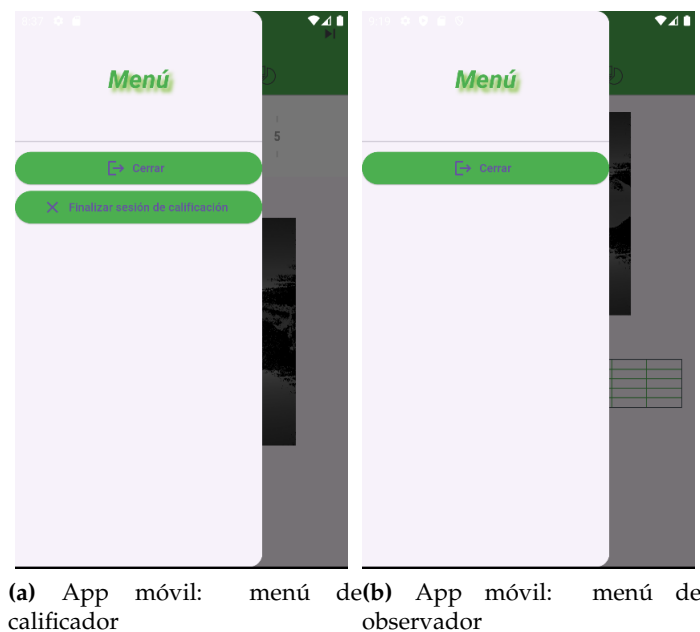


Figure 5.4: App móvil: menú

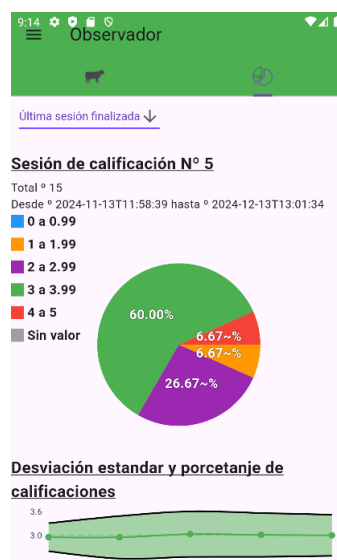


Figure 5.5: App móvil: reportes

5.5 Rendimientos en dispositivos móviles

Para evaluar la eficiencia del sistema, se midieron los tiempos de predicción del BCS en dispositivos móviles, un aspecto crítico para asegurar una experiencia fluida al obtener los resultados de la predicción. Antes de la implementación del prototipo, el algoritmo fue probado mediante una versión preliminar diseñada para Android. Estas pruebas se llevaron a cabo en diversos dispositivos móviles, variando tanto la cantidad de dispositivos como el tamaño de los lotes de procesamiento ("jobs").

En la versión preliminar, los datos se transmitían de forma continua desde un servidor a los dispositivos móviles usando round-robin, operando dentro de la misma red local. En el prototipo final, el flujo de datos comenzó con una máquina que simulaba el sistema de envíos hacia el servidor. Posteriormente, el servidor redistribuía los trabajos a los dispositivos móviles usando round robin, permitiendo evaluar el rendimiento en condiciones más representativas del entorno real en una red local.

En 5.1 se puede ver una comparación de la versión preliminar y el prototipo con respecto a la transferencia de datos. La tabla muestra que, a medida que aumenta la cantidad de imágenes por job, el tiempo de transferencia de datos disminuye en la versión preliminar. Esto se debe a que el cliente obtenía cada imagen a procesar mediante una llamada HTTP, lo que generaba mayores tiempos de transferencia. Sin embargo, al cambiar el mecanismo de transferencia a una conexión WebSocket, se logra una reducción significativa en el tiempo de transferencia. Además, la menor variabilidad en la versión del prototipo sugiere una gestión más eficiente de la transferencia de imágenes.

En 5.2 y 5.6 se puede ver un análisis comparativo entre las diferentes pruebas en términos del tiempo de detección por imagen. En la comparación se puede ver que Note 7 y el Moto G9 Play obtuvieron un mejor tiempo promedio de detección de imágenes y una menor variabilidad en la versión preliminar, en comparación con la versión prototipo, que incorpora dispositivos más modernos. Esta diferencia se debe a que, en la versión preliminar, el dispositivo se centraba exclusivamente en la ejecución del algoritmo, mientras que en el prototipo se realizaron tareas adicionales, como la visualización de todas las imágenes procesadas (no solo las utilizadas para la predicción) y el envío de los resultados al servidor. A pesar de estas diferencias en la carga de procesamiento, es un aspecto clave a evaluar con el objetivo de optimizar tanto el tiempo de detección como la desviación estándar en futuras iteraciones.

A continuación, se presenta la comparación del tiempo total de procesamiento.

Se observa que la versión preliminar logra un procesamiento más rápido a medida que los jobs contienen un mayor número de imágenes, lo cual se debe a que, como se analizó anteriormente, la transferencia de datos representaba el mayor costo en términos de tiempo. En la versión prototipo, este aspecto se optimizó mediante el uso de una conexión WebSocket, reduciendo significativamente los tiempos de transferencia. Sin embargo, dado que cada job en el prototipo

Job por imágenes	versión	Dispositivo	tiempo de tranferencia por imagen (ms)
2	Preliminar	Note 7	697 ± 610
		Moto g9 play	746 ± 447
2	Preliminar	Note 7	983 ± 910
		Moto g6 play	807 ± 265
		Moto g9 play	1065 ± 3092
15	Preliminar	Note 7	57 ± 13
		Moto g6 play	58 ± 11
		Moto g9 play	64 ± 14
1	Prototipo	g9 plus	34 ± 92
		moto g32	41 ± 99

Table 5.1: Transferencia de datos

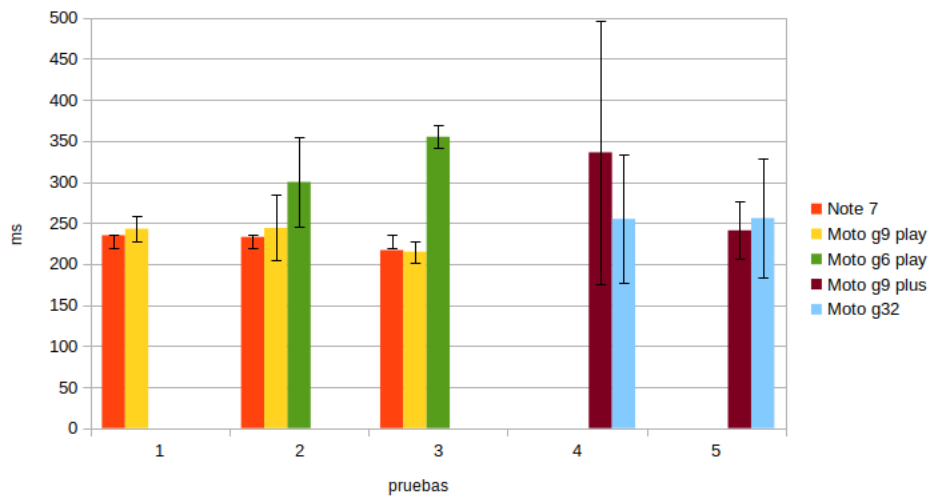


Figure 5.6: Tiempo de detección

Imágenes por job	Versión	Stream	Cluster de smartphones marca-modelo	Tiempo de detección por imagen (ms)
Stream de dos imágenes por job	preliminar	10 segundos entre ráfagas de 12 job	Note7	235 ± 24
			moto g9 play	243 ± 15
Stream de dos imágenes por job	preliminar	10 segundos entre ráfagas de 12 job	Note 7	233 ± 23
			moto g6	300 ± 54
			moto g9 play	244 ± 30
Stream de 15 imágenes por job	preliminar	10 segundos entre ráfagas de 12 job	Note 7	217 ± 3
			moto g6	355 ± 14
			moto g9 play	215 ± 13
Stream de 1 imagen por job	prototipo	10 segundos entre ráfagas de 12 job	moto g9 plus	336 ± 161
			Motorola moto g32	255 ± 78
Stream de 1 imagen por job	prototipo	10 segundos entre ráfagas de 12 job	moto g9 plus	249 ± 48
			moto g32	298 ± 47
Stream de 1 imagen por job	prototipo	5 segundos entre ráfagas de 12 job	Motorala moto g9 plus	241 ± 35
			Motorola moto g32	256 ± 73

Table 5.2: Tiempos de ejecución móvil

Cantidad de dispositivos	Imágenes por job	Versión	Imágenes procesadas	Job Recibidos	Tiempo total
2	2	Preliminar	1752	876	26:43
3	2	Preliminar	1752	876	26:44
3	3	Preliminar	1740	116	3:47
2	1	Prototipo	876	876	876
2	1	Prototipo	432	432	432
2	2	Prototipo	432	432	432

Table 5.3: Tiempo de procesamiento total

contiene una única imagen y es necesario visualizarla en el dispositivo del observador, el tiempo de procesamiento por imagen es mayor.

A lo largo de las pruebas, se redujo progresivamente el tiempo de espera entre ráfagas, como se observa en la prueba 3 del prototipo, sin afectar la eficiencia del sistema. En futuras evaluaciones, sería recomendable continuar disminuyendo este tiempo de espera y mejorar el prototipo para permitir el procesamiento de múltiples imágenes por job. Asimismo, sería útil incorporar la opción de deshabilitar la visualización de las imágenes, mostrando únicamente los resultados de la predicción, con el objetivo de realizar comparaciones más precisas y optimizar el rendimiento.

Adicionalmente, las pruebas realizadas con el prototipo permitieron evaluar el consumo de memoria en los dispositivos utilizados. Los resultados obtenidos fueron los siguientes:

- En la primera prueba, el Moto G32 alcanzó un consumo máximo de 663.2 MB, mientras que el G9 Plus registró un pico de 445.49 MB.
- En la segunda prueba, el consumo máximo del Moto G32 fue de 558.29 MB, mientras que el G9 Plus redujo su uso de memoria a 425.64 MB.

5.6 Rendimientos en el servidor

Como se mencionó anteriormente el sistema fue probado en un entorno con dos observadores, sin calificadores para verificar el consumo de recursos durante la ejecución del algoritmo de predicción, y demostró los siguientes consumos de memoria:

- ActiveMQ: fue incrementando desde 277 MB de memoria hasta 490 MB de memoria.
- MySQL: fue incrementando desde 395 MB de memoria hasta 403 MB de memoria.
- Servidor (API principal): no superaron los 500 MB de memoria las pruebas realizadas.

Conclusiones, Limitaciones y Trabajos Futuros

6.1 Conclusión

En este proyecto se desarrolló un sistema funcional y extensible diseñado para su implementación en entornos ganaderos, logrando mejorar la precisión y eficiencia en la predicción del BCS. Este desarrollo supuso enfrentar diferentes niveles de desafío según el componente en cuestión. En el caso del servidor, mi experiencia de más de 10 años como desarrollador backend fue determinante para diseñar e implementar este componente de manera ágil, utilizando tecnologías como Java, Spring Boot, JMS Template y STOMP. Además, esta experiencia facilitó la comprensión del dominio y los requerimientos funcionales, permitiendo una interacción efectiva con el modelo de negocio y sus particularidades.

Por otro lado, el desarrollo de la aplicación móvil representó un desafío mayor, debido a mi falta de experiencia con Flutter, Dart y el desarrollo de interfaces visuales. Esta situación implicó una curva de aprendizaje significativa, que demandó más tiempo y esfuerzo en comparación con el desarrollo del servidor. Sin embargo, esta experiencia se convirtió en una oportunidad valiosa para ampliar mis habilidades en áreas complementarias y explorar nuevos enfoques en el desarrollo de software.

En total, el proyecto involucró el desarrollo de 113 clases java y 4748 líneas de código en el servidor y 52 archivos dart y 37 clases java y 6006 líneas de código en la aplicación móvil, alcanzando un total de 10754 líneas de código, reflejando su complejidad técnica y su alcance.

La modularidad y escalabilidad del sistema abren puertas para futuras ampliaciones, como la integración con dispositivos IoT, la optimización de modelos predictivos según distintas razas bovinas y el desarrollo de estrategias avanzadas de tolerancia a fallos. En conjunto, este proyecto no solo cumplió con los objetivos iniciales, sino que también sienta las bases para

nuevas innovaciones en la automatización y gestión del sector ganadero. Además, representa un avance significativo tanto en términos técnicos como en el aprendizaje y adaptación a tecnologías y dominios desconocidos.

6.2 Limitaciones

6.2.1 Seguridad

El sistema de inicio de sesión presenta una limitación en términos de seguridad. Actualmente, este componente únicamente permite identificar el tipo de usuario y asociar calificaciones a un experto específico, pero no incluye mecanismos de autenticación robustos ni validaciones avanzadas para garantizar la protección del acceso.

Asimismo, no se han implementado restricciones para controlar quiénes pueden registrarse como calificadores. Esto permite que cualquier usuario pueda asumir este rol, lo que podría comprometer la calidad de las evaluaciones realizadas. La falta de validación previa de las credenciales o experiencia de los calificadores representa una limitación para la confiabilidad del sistema en entornos reales.

6.2.2 Tolerancia a fallos

El sistema actual presenta limitaciones en su capacidad de tolerancia a fallos. Ante interrupciones en la conexión WebSocket o errores relacionados, no existen mecanismos implementados para realizar reintentos automáticos o redirigir los trabajos ("jobs"), lo que puede resultar en predicciones no realizadas por el modelo.

Asimismo, el proceso de calificación carece de estrategias que manejen fallos en llamadas REST, lo que puede ocasionar la pérdida de evaluaciones realizadas por los expertos y comprometer la integridad de los datos almacenados.

6.2.3 Mejora en la Gestión de Jobs para Calificadores

Una limitación en la implementación actual es que los calificadores no pueden procesar los trabajos ("jobs") generados antes de su ingreso a una sesión de calificación. Esto significa que los trabajos existentes antes del inicio de la sesión quedan sin calificar, lo que afecta la completitud del proceso de calificación.

Además, no se contempla la carga automática de trabajos previos desde la base de datos hacia la cola JMS del usuario al momento de iniciar sesión, lo que limita la capacidad de los calificadores para abordar tareas pendientes acumuladas antes de su participación activa en la sesión.

6.2.4 Observadores y Algoritmo de Predicción

En la implementación actual, la predicción del puntaje de condición corporal (BCS) está limitada a ser ejecutada exclusivamente en la aplicación móvil por usuarios con logueados con el rol de observador. Esto implica que, en ausencia de observadores, el sistema no genera predicciones. En etapas de prueba preliminares del prototipo en entornos reales en los que el apartamiento de animales según condición corporal se realice de forma manual, sin mecanización automática, es decir, por humanos, esto puede no verse como una limitación ya que se puede asumir que el sistema contará con el poder de cómputo de los dispositivos de quienes realizan dicha labor. Sin embargo, en un entorno que cuenta con apartamiento automático, es deseable que el sistema cuente con una forma alternativa de generar la predicción. Para esto, se prevee que en el futuro el componente servidor actúe como un observador por defecto.

6.3 Trabajos futuros

6.3.1 Múltiples calificadores

En futuras versiones del servidor, se planea implementar mejoras que permitan a los calificadores compartir una única cola JMS, facilitando la distribución equitativa de las tareas de calificación. Esto incluiría la opción de dividir un grupo de vacas (rodeo) entre calificadores o mantener la funcionalidad actual, donde todos los calificadores evalúan todas las vacas. Además, esta mejora permitiría que los calificadores puedan abandonar una sesión y unirse a otra según sea necesario, ofreciendo mayor flexibilidad en la gestión de las sesiones.

Actualmente, aunque la aplicación móvil no permite que dos calificadores trabajen simultáneamente, el servidor ha sido diseñado para admitir múltiples calificadores en una misma sesión. Cada calificador dispone de su propia cola JMS y tiene la capacidad de calificar todas las vacas, lo que sienta las bases para la implementación de las mejoras mencionadas.

6.3.2 Sistema generador de Jobs

En las pruebas de rendimiento realizadas, fue necesario la creación de dicho componente para simular un escenario de ejecución. Sin embargo, la implementación real de un sistema de generación de jobs quedó fuera del alcance de este proyecto. Este sistema debiera configurarse para notificar al servidor sobre la existencia de una secuencia de imágenes capturadas por una cámara y asociadas unívocamente a una vaca, lo que da lugar a la creación de un "job" que deberá ser tratado por el componente servidor desarrollado en este proyecto. El generador de jobs deberá permitir, además, configurar la cantidad de imágenes por vaca a ser capturadas, teniendo en cuenta calidad de las mismas, por ejemplo, mediante la aplicación de filtros para

determinar borrosidad de la imagen, presencia total o parcial de la parte del cuerpo que se emplea en el cálculo del BCS, etc..

6.3.3 Procesamiento de imágenes

Actualmente, el sistema está diseñado para recibir y almacenar múltiples imágenes asociadas a cada vaca; sin embargo, sólo la primera imagen es enviada a los usuarios calificadores y observadores. Una posible mejora futura sería habilitar la capacidad de enviar todas las imágenes relacionadas con cada vaca, permitiendo un análisis más exhaustivo por parte de los calificadores y observadores. Además, esto abriría la posibilidad de optimizar el algoritmo de predicción para procesar múltiples imágenes dentro de un mismo job, lo que podría mejorar la precisión de los resultados generados.

6.3.4 Integración con IoT

El sistema tiene el potencial de integrarse en un entorno IoT, permitiendo la conexión con dispositivos y elementos del entorno ganadero, como mangas, corrales y comederos. Esto requerirá desarrollar una capa de comunicación que facilite la captura automatizada de datos y acciones de contingencia, estructurando el sistema como un Sistema Integral de Gestión de la Información (SIGI) para apoyar el monitoreo, la planificación y la toma de decisiones.

6.3.5 Múltiples modelos

Actualmente, la aplicación móvil está diseñada de manera modular, lo que permite delegar la ejecución del algoritmo de predicción a un plugin específico de Android. Esta estructura facilita la integración de nuevos modelos de predicción mediante ajustes en el código. Sin embargo, para aprovechar esta flexibilidad, es necesario desarrollar una funcionalidad adicional en la aplicación que permita seleccionar el modelo de predicción deseado, lo que representa una posible mejora a futuro.

6.3.6 Múltiples plataformas

Al desarrollar la aplicación móvil en Flutter, la portabilidad a iOS se facilita significativamente debido a la naturaleza multiplataforma del framework. Sin embargo, para lograr esta transición, es necesario analizar y ajustar el modelo predictivo, ya que actualmente está diseñado y probado específicamente para dispositivos Android.

6.3.7 Configuración de la participación de observadores en el cálculo de BCS

En futuras implementaciones, se planea incorporar a la aplicación móvil un interruptor (switch) junto con un campo de configuración que permita a un usuario observador decidir si aportar o no recursos de cómputo para el cálculo del BCS, o incluso que dicha opción pueda ser habilitada dependiendo del nivel de batería con el que cuenta el dispositivo. Este mecanismo permitirá al dispositivo móvil notificar al servidor si puede ofrecer recursos de procesamiento para el cálculo del BCS como así también fijar cuotas de uso de batería.

Al activar este interruptor, se deberá iniciar un servicio en segundo plano encargado de gestionar un protocolo de intercambio de mensajes entre la aplicación y el servidor, asegurando una coordinación eficiente para el uso compartido de recursos. Aunque esta funcionalidad estaba contemplada desde el inicio, no se logró implementar debido a las dificultades encontradas durante el desarrollo de la aplicación móvil.

Bibliography

- R. S. Alonso, I. Sittón-Candanedo, Ó. García, J. Prieto, and S. Rodríguez-González. An intelligent edge-iot platform for monitoring livestock and crops in a dairy farming scenario. *Ad Hoc Networks*, 98:102047, 2020.
- J. R. Alvarez, M. Arroqui, P. Mangudo, J. Toloza, D. Jatip, J. M. Rodríguez, A. Teyseyre, C. Sanz, A. Zunino, C. Machado, et al. Body condition estimation on cows from depth images using convolutional neural networks. *Computers and electronics in agriculture*, 155:12–22, 2018.
- N. Apolónia, F. Freitag, and L. Navarro. Leveraging deployment models on low-resource devices for cloud services in community networks. *Simulation Modelling Practice and Theory*, 77:390–406, 2017.
- A. Aquino, I. Barrio, M.-P. Diago, B. Millan, and J. Tardaguila. vitisberry: An android-smartphone application to early evaluate the number of grapevine berries by means of image analysis. *Computers and Electronics in Agriculture*, 148:19–28, 2018.
- J. Bao and Q. Xie. Artificial intelligence in animal farming: A systematic literature review. *Journal of Cleaner Production*, 331:129956, 2022.
- K. K. Çevik. Deep learning based real-time body condition score classification system. *IEEE Access*, 8:213950–213957, 2020.
- J. De Hagen, R. Chayer, C. Machado, O. Melluci, and R. Catalano. Estudio de la asociación entre la respuesta reproductiva y el estado corporal evaluado al momento del diagnóstico de gestación en bovinos. 2010.
- A. Fischer, T. Luginbühl, L. Delattre, J. Delouard, and P. Faverdin. Rear shape in 3 dimensions summarized by principal component analysis is a good predictor of body condition score in holstein dairy cows. *Journal of dairy science*, 98(7):4465–4476, 2015.
- M. Gusev and Y. Wang. Formal description of dew computing. In *Proceedings of The 3rd International Workshop on Dew Computing*, pages 8–13, 2018.

- M. Hirsch, J. M. Rodriguez, C. Mateos, and A. Zunino. A two-phase energy-aware scheduling approach for cpu-intensive jobs in mobile grids. *Journal of Grid Computing*, 15:55–80, 2017.
- M. Hirsch, C. Mateos, and A. Zunino. Augmenting computing capabilities at the edge by jointly exploiting mobile devices: A survey. *Future Generation Computer Systems*, 88:644–662, 2018.
- M. Hirsch, C. Mateos, J. M. Rodriguez, and A. Zunino. Dewsim: A trace-driven toolkit for simulating mobile device clusters in dew computing environments. *Software: Practice and Experience*, 50(5):688–718, 2020.
- C.-H. Hong and B. Varghese. Resource management in fog/edge computing: a survey on architectures, infrastructure, and algorithms. *ACM Computing Surveys (CSUR)*, 52(5):1–37, 2019.
- A. Kamilaris and F. X. Prenafeta-Boldú. Deep learning in agriculture: A survey. *Computers and electronics in agriculture*, 147:70–90, 2018.
- S. Kumar, A. Pandey, K. S. R. Satwik, S. Kumar, S. K. Singh, A. K. Singh, and A. Mohan. Deep learning framework for recognition of cattle using muzzle point image pattern. *Measurement*, 116:1–17, 2018.
- T. Kurita. Principal component analysis (pca). *Computer vision: a reference guide*, pages 1–4, 2019.
- J. Li, J. Fang, Y. Fan, and C. Zhang. Design on the monitoring system of physical characteristics of dairy cattle based on zigbee technology. In *2010 World Automation Congress*, pages 63–66. IEEE, 2010.
- M. S. Mahmud, A. Zahid, A. K. Das, M. Muzammil, and M. U. Khan. A systematic literature review on deep learning applications for precision cattle farming. *Computers and Electronics in Agriculture*, 187:106313, 2021.
- S. Maresca, J. Quiroz García, G. Melani, J. Burges, G. Brusca, and F. Plorutti. El estado corporal y su efecto en la eficiencia reproductiva en rodeos de cría de la cuenca del salado. *Publicación Técnica INTA*, (3), 2008.
- G. Masferrer, R. Carreras, M. Font-i Furnols, M. Gispert, P. Marti-Puig, and M. Serra. On-line ham grading using pattern recognition models based on available data in commercial pig slaughterhouses. *Meat science*, 143:39–45, 2018.
- P.-A. Morin, Y. Chorfi, J. Dubuc, J.-P. Roy, D. Santschi, and S. Dufour. An observational study investigating inter-observer agreement for variation over time of body condition score in dairy cows. *Journal of Dairy Science*, 100(4):3086–3090, 2017.

- A. Nasirahmadi, B. Sturm, A.-C. Olsson, K.-H. Jeppsson, S. Müller, S. Edwards, and O. Hensel. Automatic scoring of lateral and sternal lying posture in grouped pigs using image processing and support vector machine. *Computers and electronics in agriculture*, 156:475–481, 2019.
- G. P. Piedade, S. B. Mafra, D. M. de Oliveira, and M. R. da Cruz. Design, deployment, and validation of a low-cost iot platform based on lora for precision dairy farming.
- F. J. Prego, L. Nieto, M. Solla, and I. Puente. A mobile android tool for simplified gpr data processing in construction applications. *Automation in Construction*, 89:170–182, 2018.
- M. Richards, J. Spitzer, and M. Warner. Effect of varying levels of postpartum nutrition and body condition at calving on subsequent reproductive performance in beef cattle. *Journal of Animal Science*, 62(2):300–306, 1986.
- M. Riekert, A. Klein, F. Adrion, C. Hoffmann, and E. Gallmann. Automatically detecting pig position and posture by 2d camera imaging and deep learning. *Computers and Electronics in Agriculture*, 174:105391, 2020.
- A. Rindos and Y. Wang. Dew computing: The complementary piece of cloud computing. In *2016 IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom)(BDCloud-SocialCom-SustainCom)*, pages 15–20. IEEE, 2016.
- J. R. Roche, N. C. Friggens, J. K. Kay, M. W. Fisher, K. J. Stafford, and D. P. Berry. Invited review: Body condition score and its association with dairy cow productivity, health, and welfare. *Journal of dairy science*, 92(12):5769–5801, 2009.
- J. M. Rodríguez Álvarez, M. Arroqui, P. Mangudo, J. M. Toloza, D. E. Jatip, J. M. Rodriguez, A. O. Zunino Suarez, C. Mateos, and C. Machado. Revisión y análisis de técnicas y métodos computacionales para la evaluación de la condición corporal en vacas. 2018.
- G. Selk, R. Wettemann, K. Lusby, J. Oltjen, S. Mobley, R. Rasby, and J. Garmendia. Relationships among weight change, body condition and reproductive performance of range beef cows. *Journal of animal science*, 66(12):3153–3159, 1988.
- A. Singh, B. Ganapathysubramanian, A. K. Singh, and S. Sarkar. Machine learning for high-throughput stress phenotyping in plants. *Trends in plant science*, 21(2):110–124, 2016.
- G. Singh, D. Bansal, S. Sofat, and N. Aggarwal. Smart patrolling: An efficient road surface monitoring using smartphone sensors and crowdsourcing. *Pervasive and Mobile Computing*, 40:71–88, 2017.
- K. Skala, D. Davidovic, E. Afgan, I. Sovic, and Z. Sojat. Scalable distributed computing hierarchy: Cloud, fog and dew computing. *Open Journal of Cloud Computing (OJCC)*, 2(1): 16–24, 2015.

- E. Wildman, G. Jones, P. Wagner, R. Boman, H. Troutt Jr, and T. Lesch. A dairy cow body condition scoring system and its relationship to selected production characteristics. *Journal of dairy science*, 65(3):495–501, 1982.
- C.-J. Yang, Y.-H. Lin, and S.-Y. Peng. Develop a video monitoring system for dairy estrus detection at night. In *2017 International Conference on Applied System Innovation (ICASI)*, pages 1900–1903. IEEE, 2017.
- S. Yukun, H. Pengju, W. Yujie, C. Ziqi, L. Yang, D. Baisheng, L. Runze, and Z. Yonggen. Automatic monitoring system for individual dairy cows based on a deep learning framework that provides identification via body parts and estimation of body condition score. *Journal of dairy science*, 102(11):10140–10151, 2019.