

UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO
FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA, INFORMÁTICA Y MECÁNICA
ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA Y DE SISTEMAS



TESIS

**IMPLEMENTACIÓN DE UN SISTEMA WEB INTELIGENTE QUE
INTEGRA UNA CNN PARA LA DETECCIÓN Y EL REGISTRO
AUTOMÁTICO DE DETERIOROS EN VÍAS VEHICULARES
PAVIMENTADAS, A PARTIR DE IMÁGENES CAPTURADAS Y
ENVIADAS POR USUARIOS DESDE DISPOSITIVOS MÓVILES**

PRESENTADO POR:

Br. HERBERTH FITZROY GARCIA CHURA

**PARA OPTAR AL TÍTULO PROFESIONAL DE
INGENIERO INFORMÁTICO Y DE SISTEMAS**

ASESOR:

Dr. RONY VILLAFUERTE SERNA

CUSCO - PERÚ
2025

INFORME DE ORIGINALIDAD
(Aprobado por Resolución Nro.CU-303-2020-UNSAAC)

El que suscribe, asesor del trabajo de investigación titulado: “IMPLEMENTACIÓN DE UN SISTEMA WEB INTELIGENTE QUE INTEGRA UNA CNN PARA LA DETECCIÓN Y EL REGISTRO AUTOMÁTICO DE DETERIOROS EN VÍAS VEHICULARES PAVIMENTADAS, A PARTIR DE IMÁGENES CAPTURADAS Y ENVIADAS POR USUARIOS DESDE DISPOSITIVOS MÓVILES”. Presentado por el bachiller:

- HERBERTH FITZROY GARCIA CHURA. con DNI nro.: 70219990

Para optar al Título Profesional de Ingeniero Informático y de Sistemas, informo que el trabajo de investigación ha sido sometido a revisión por 3ra vez, mediante el software anti plagio, conforme al Artículo 6° del **Reglamento para Uso de Sistema Anti plagio de la UNSAAC** y de la evaluación de originalidad se tiene un porcentaje de 1% (uno por ciento).

**Evaluación y acciones del reporte de coincidencia para trabajos de investigación
conducentes a grado académico o título profesional, tesis**

PORCENTAJE	EVALUACIÓN Y ACCIONES	MARQUE CON UNA X
Del 1 al 10 %	No se considera plagio.	X
Del 11 al 30%	Devolver al usuario para las correcciones.	
Mayores a 31 %	El responsable de la revisión del documento emite un informe al inmediato jerárquico, quien a su vez eleva el informe a la autoridad académica para que tome las acciones correspondientes. Sin perjuicio de las sanciones administrativas que correspondan de acuerdo a ley.	

Por tanto, en mi condición de Asesor, firmo el presente informe en señal de conformidad y adjunto la primera hoja del reporte del software anti plagio.

Cusco, 05 de junio del 2024.



Dr. Rony Villafuerte Serna

DNI: 23957778

ORCID: 0000-0003-4607-522X

Se adjunta:

1. Reporte generado por el sistema anti plagio.
2. Enlace del reporte generado por el sistema anti plagio: [OID: 27259:464760675](https://doi.org/10.27259/5446075)

HERBERTH FITZROY GARCIA CHURA

IMPLEMENTACIÓN DE UN SISTEMA WEB INTELIGENTE QUE INTEGRA UNA CNN PARA LA DETECCIÓN Y EL REGISTRO AU...

 Universidad Nacional San Antonio Abad del Cusco

Detalles del documento

Identificador de la entrega

trn:oid:::27259:464760675

Fecha de entrega

5 jun 2025, 7:25 a.m. GMT-5

Fecha de descarga

5 jun 2025, 9:37 a.m. GMT-5

Nombre de archivo

deteccionFallasVisionArt_V3.pdf

Tamaño de archivo

13.4 MB

174 Páginas

34.190 Palabras

194.432 Caracteres

1% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- ▶ Bibliography
- ▶ Quoted Text
- ▶ Small Matches (less than 20 words)

Exclusions

- ▶ 12 Excluded Matches

Top Sources

- 1%  Internet sources
- 0%  Publications
- 1%  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Dedicatoria

Dedico este trabajo a toda mi familia, y en especial a mis abuelos Fitzroy Garcia Rojas y Remigio Chura Lianque. Aunque ya no están presentes, vivo siguiendo su sabia ideología y valores, manteniendo así su legado siempre presente en mi vida.

Agradecimientos

Quisiera expresar mi más sincera gratitud a Dios, quien constantemente me recuerda la importancia de practicar valores mientras me desarrollo en mi profesión. A mis padres, Herbeth García Salcedo y Hilda Chura Sercera, y a mi familia, quienes han hecho posible mi desarrollo y éxito en la sociedad. A la familia de la Universidad Nacional de San Antonio Abad del Cusco, por ofrecer la oportunidad de prepararnos y desarrollarnos como profesionales. A los docentes de la Escuela Profesional de Ingeniería Informática y de Sistemas, y especialmente a mi asesor de tesis, por su integridad y profesionalismo en guiarme a través del desafío de elaborar esta tesis.

Resumen

La identificación de imperfecciones en las vías de transporte para el mantenimiento de la infraestructura de vías vehiculares pavimentadas es un tema de investigación a nivel mundial, siendo una de las metodologías más recientemente utilizadas la visión computacional, haciendo uso de redes neuronales convolucionales. Esto permite identificar todo tipo de objetos sobre las infraestructuras viales y procesarlos para su evaluación y análisis. En específico, identificar los deterioros en la superficie de rodamiento de las autopistas es un factor importante, ya que permite realizar una evaluación constante de las infraestructuras viales antes de que se generen fallas en estas. Esto reduce gastos económicos, incentiva más el mantenimiento y previene la rehabilitación total de las autopistas, evita accidentes que generen lesiones en los ciudadanos y transmite una buena imagen regional para el turismo, al ofrecer calles seguras y en óptimas condiciones.

Este trabajo tiene como objetivo general aplicar visión computacional para la identificación y recolección de información sobre deterioros en las vías vehiculares pavimentadas mediante redes neuronales convolucionales. Para lograr este objetivo: en primer lugar, se recolectarán imágenes para construir y configurar un dataset; en segundo lugar, se adaptarán los datos a las entradas y salidas de la red neuronal MobileNetV2 para proceder a su evaluación, entrenamiento y optimización; en tercer lugar, se configurará un prototipo web que sirva como herramienta para recolectar información sobre los deterioros (geolocalización e imagen) que cualquier ciudadano encontraría en las vías vehiculares pavimentadas; en cuarto lugar, se realizarán pruebas del prototipo en rutas donde abundan los deterioros y fallas en el departamento del Cusco.

Palabras clave: Visión computacional, redes neuronales convolucionales, detección de imperfecciones en las vías públicas, visión artificial, deterioros, fallas, dataset, MobileNetV2.

Abstract

The identification of imperfections in transportation routes for the maintenance of terrestrial road infrastructure is a global research topic. One of the most recently adopted methodologies is computer vision, which makes use of convolutional neural networks. This approach enables the identification of all kinds of objects on road infrastructures and allows for their processing, evaluation, and analysis. Specifically, identifying deteriorations on highway pavement surfaces is a crucial factor, as it enables continuous evaluation of road infrastructure before failures occur. This reduces economic costs, encourages preventive maintenance, avoids the need for total road rehabilitation, helps prevent accidents that could harm citizens, and promotes a positive regional image for tourism by offering safe and well-maintained roads.

This work aims to apply computer vision for the identification and collection of information about deteriorations on terrestrial transportation routes using convolutional neural networks. To achieve this objective: first, images will be collected to build and configure a dataset; second, the data will be adapted to the input and output formats of the MobileNetV2 neural network for evaluation, training, and optimization; third, a web prototype will be developed to serve as a tool for collecting information about road deteriorations (including geolocation and image) that any citizen may encounter; and fourth, the prototype will be tested on routes with frequent road failures in the Cusco region.

Introducción

En los últimos años, el acelerado crecimiento urbano ha generado una expansión considerable de la infraestructura vial. Este incremento ha desafiado la capacidad operativa de entidades como el Ministerio de Transportes y Comunicaciones (MTC), las Instituciones de Vigilancia de Pavimentos (IVPs), OSITRAN y los gobiernos locales responsables del mantenimiento de los bienes públicos. En la región del Cusco, estas instituciones enfrentan limitaciones administrativas y burocráticas que dificultan una atención oportuna y eficiente a las demandas de la población.

Ante esta problemática, el presente proyecto plantea una solución basada en tecnologías avanzadas de inteligencia artificial y visión computacional. Se propone una herramienta confiable, imparcial y de rápida ejecución que permita a los ciudadanos reportar de forma efectiva los deterioros en las vías de tránsito. A su vez, esta herramienta proporciona a las entidades interesadas una vía eficiente para analizar y gestionar la información contenida en dichos reportes, facilitando la toma de decisiones orientadas al mantenimiento vial.

A lo largo de este proyecto, se explorará el proceso de creación de esta herramienta, un prototipo para la identificación de deterioros en las vías vehiculares pavimentadas, y se detallará en cuatro capítulos, conclusiones y recomendaciones:

En el primer Capítulo: ASPECTOS GENERALES, este capítulo se describe el planteamiento y formulación del problema, así como el alcance, justificación, delimitación y metodología para la posible solución propuesta en el proyecto.

El segundo capítulo: MARCO TEÓRICO, Se aborda la teoría necesaria para la construcción de la solución al problema, incluyendo proyectos anteriores como antecedentes, las entidades encargadas del mantenimiento vial terrestre, y conceptos clave en inteligencia artificial, visión computacional, sistemas de geoposicionamiento, entre otros.

En el tercer capítulo: IMPLEMENTACIÓN DE LA PLATAFORMA, detalla el desarrollo del prototipo propuesto, desde la construcción de un dataset de deterioros en las vías vehiculares pavimentadas del Cusco, el diseño de una red neuronal convolucional con visión artificial, el desarrollo de una base de datos, la creación de una aplicación web, hasta las pruebas del prototipo para identificar deterioros en pavimentos rígidos.

En el cuarto capítulo: ANÁLISIS Y DISCUSIÓN DE RESULTADOS, Se presenta un análisis detallado de los resultados obtenidos tras la implementación del prototipo, incluyendo una comparación con proyectos anteriores.

Finalmente, se presentarán las conclusiones, recomendaciones, bibliografía y anexos correspondientes.

Listado de Abreviaturas

- **MTC:** Ministerio de Transportes y Comunicaciones.
- **OSITRAN:** Organización Supervisora de la Inversión en Infraestructura de Transporte.
- **IVP:** Instituto Vial Provincial.
- **CV:** Computer Vision (Visión Computacional).
- **ANN:** Artificial Neural Network (Red Neuronal Artificial).
- **IA:** Artificial Intelligence (Inteligencia Artificial).
- **ML:** Machine Learning (Aprendizaje Automático).
- **DL:** Deep Learning (Aprendizaje Profundo).
- **NN:** Neural Network (Red Neuronal).
- **BD:** Base de datos.
- **CNN:** Convolutional Neural Network (Red Neuronal Convolutacional).
- **TF:** TensorFlow.
- **IoU:** Intersection over Union (Intersección sobre Unión).
- **AP:** Average Precision (Precisión Promedio).
- **YOLO:** You Only Look Once.
- **MobileNet V2:** Mobile Network versión 2 (una arquitectura de red neuronal).
- **SPA:** Single-page application o aplicacion de pagina unica.
- **SQL:** Lenguaje de dominio especifico utilizado para administra y recurrar información de gestión de bases de datos relacionales.
- **CRUD:** Funciones básicas en base de datos Crear, Leer, Actualizar y Borrar.
- **HTML:** HyperText Markup Language, hace referencia al lenguaje de marcado para la elaboración de páginas web.

- **CSS:** lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado.
- **JWT:** JSON Web Token es un estándar abierto basado en JSON propuesto por IETF para la creación de tokens de acceso que permiten la propagación de identidad y privilegios o claims en inglés.
- **HTTP:** El Protocolo de transferencia de hipertexto es el protocolo de comunicación que permite las transferencias de información a través de archivos en la World Wide Web.
- **HTTPS:** Versión segura de HTTP que cifra la información mediante SSL/TLS, protegiendo los datos transmitidos entre el cliente y el servidor.
- **SSL:** Protocolo criptográfico diseñado para asegurar la transmisión de datos en internet. Fue el precursor de TLS. Actualmente está obsoleto.
- **TLS:** Protocolo criptográfico que reemplaza a SSL. Se utiliza para cifrar y proteger los datos en comunicaciones seguras como HTTPS.
- **JSX:** es una extensión de sintaxis para JavaScript que se utiliza en React. Es similar a un template language, pero tiene todo el poder de JavaScript.
- **JS:** JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript.
- **JSON:** Es un formato de texto sencillo para el intercambio de datos.

Índice general

Dedicatoria	I
Agradecimientos	II
Resumen	III
Abstract	IV
Introducción	V
Listado de Abreviaturas	VII
Índice General	IX
Índice de figuras	XIII
Índice de tablas	XVII
1. Aspectos Generales	1
1.1. Planteamiento del Problema	1
1.1.1. Descripción del problema	1
1.1.2. Identificación del problema	2
1.2. Formulación del Problema	3
1.2.1. Problema General	3
1.2.2. Problemas Específicos	3

	X
1.3. Objetivos	4
1.3.1. Objetivo General	4
1.3.2. Objetivos Específicos	4
1.4. Justificación	5
1.4.1. Conveniencia	5
1.4.2. Relevancia	5
1.4.3. Implicancias Prácticas	5
1.4.4. Valor Teórico	5
1.4.5. Utilidad Metodológica	6
1.5. Delimitación de estudio	6
1.5.1. Delimitación Espacial	6
1.5.2. Delimitación Temporal	6
1.5.3. Delimitación Técnica	7
1.6. Método	7
1.6.1. Alcance	7
1.6.2. Diseño	8
1.6.3. Para el desarrollo de la parte informática	8
1.6.4. Cronograma de Actividades	9
1.6.5. Presupuesto	10
2. Marco Teórico	13
2.1. Antecedentes	13
2.1.1. Antecedentes Internacionales	13
2.1.2. Antecedentes Nacionales	14
2.2. Bases Teóricas	15
2.2.1. Entidades encargadas de la gestión vial	15
2.2.2. Clasificación de las Vías Terrestres en el Perú: Marco Legal y Gestión de Infraestructura Vial	16

	XI
2.2.3. Metodología de Prototipado Evolutivo	18
2.2.4. Deterioros en vías vehiculares pavimentadas	21
2.2.5. Inteligencia Artificial	26
2.2.6. Dataset	27
2.2.7. Deep Learning	33
2.2.8. Visión Computacional	39
2.2.9. Metadatos	43
2.2.10. Sistema de posicionamiento global (GPS)	44
2.2.11. Herramientas	48
3. Desarrollo e Implementación del Proyecto	52
3.1. Recolección de datos	52
3.1.1. Construcción del dataset	54
3.2. Construcción y entrenamiento de la CNN	58
3.2.1. Configuración de LabelMap	58
3.2.2. Configuración del pipeline	59
3.2.3. Entrenamiento de la CNN	60
3.2.4. Exportación de la CNN	61
3.2.5. Verificación de la CNN	61
3.2.6. Implementación del prototipo en Flask	62
3.3. Modelado del Sistema	64
3.3.1. Historias de Usuario	64
3.3.2. Casos de Uso	68
3.3.3. Fichas de Casos de Uso	71
3.3.4. Diagramas de Flujo	83
3.3.5. Diagramas de Secuencia	96
3.4. Diseño de Interfaz	106

	XII
3.5. Implementación del Prototipo Web	114
3.5.1. Arquitectura e infraestructura del Sistema	114
3.5.2. Desarrollo de la Base de Datos	116
3.5.3. Desarrollo del Backend	121
3.5.4. Desarrollo del Frontend	123
4. Análisis y Discusión de Resultados	146
4.1. Análisis de resultados respecto a los objetivos	146
4.1.1. Recolección de imágenes para la construcción y configuración de un Dataset	146
4.1.2. Desarrollo de una red neuronal para el reconocimiento de deterioros en imágenes	146
4.1.3. Desarrollo de un prototipo web para la recolección de evidencia del estado de las vías	147
4.1.4. Verificación del comportamiento del prototipo en una vía nacional	147
4.2. Discusión de resultados respecto a los antecedentes	147
Conclusiones	149
Recomendaciones	150
Bibliografía	152

Índice de figuras

1.1. Cronograma del proyecto	9
2.1. Entidades y proceso de mantenimiento de las vías de tránsito	18
2.2. Diagrama descriptivo de la ejecución de la metodología de prototipado evolutivo	21
2.3. Ejemplo grieta limacpampa - cusco	22
2.4. Ejemplo grieta avenida el sol - cusco	23
2.5. Ejemplo bache avenida grau - cusco	24
2.6. Ejemplo bache avenida circunvalacion - cusco	25
2.7. Ejemplo craquelado corao - cusco	26
2.8. Estructura de la Inteligencia Artificial	27
2.9. Dataset con deterioros	28
2.10. Dataset sin deterioros	29
2.11. Dataset	31
2.12. Imagen sometida a Data Augmentation.	32
2.13. Resultado de utilizar Data Augmentation.	33
2.14. Redes Neuronales Convolucionales	35
2.15. Overfitting	35
2.16. Underfitting	37
2.17. Detección de objetos	40
2.18. Intersection over Union	41

2.19. Non Maximum Suppression	42
2.20. Metadatos de una imagen	43
2.21. Reconocimiento de posición de un receptor	45
2.22. Líneas meridianas	45
2.23. Lineas meridianas	46
2.24. Latitud y longitud en el mapamundi	47
3.1. Mapa calles del Cusco - Mantenimiento	52
3.2. Ejemplares de datos capturados	53
3.3. Distribución de datos	55
3.4. Ejemplar 1 de bounding boxes	57
3.5. Ejemplar 2 de bounding boxes	57
3.6. Procesamiento de datos	58
3.7. Ultimo Step	61
3.8. Parámetro de perdida	62
3.9. Parámetro Tasa de aprendizaje	62
3.10. Tipos de usuario	68
3.11. Diagrama de Casos de Uso para Poblador y Usuario	69
3.12. Diagrama de Casos de Uso para Revisor y Administrador	70
3.13. Diagrama de flujo visualizar deterioros en mapa	85
3.14. Diagrama de flujo filtrar deterioros	86
3.15. Diagrama de Flujo Cargar Foto	87
3.16. Diagrama de Flujo Registrarse	88
3.17. Diagrama de Flujo Iniciar Sesión	89
3.18. Diagrama de Flujo Cerrar Sesión	90
3.19. Diagrama de Flujo Cargar Fotos por Lotes	92
3.20. Diagrama de Flujo Revisar Deterioros	93

3.21. Diagrama de Flujo Administrar Usuarios	94
3.22. Diagrama de Flujo Marcar Deterioros Reparados	95
3.23. Diagrama de secuencia visualización de deterioros en el mapa	97
3.24. Diagrama de secuencia cargar foto	98
3.25. Diagrama de secuencia sesiones y registro de usuarios	99
3.26. Diagrama de secuencia cargar foto por lotes	100
3.27. Diagrama de secuencia revisión de deterioros personales	102
3.28. Diagrama de secuencia revisión de deterioros comunitarios y por usuarios	103
3.29. Diagrama de secuencia marcar deterioros en mantenimiento o reparado .	104
3.30. Diagrama de secuencia administrar usuarios	105
3.31. Pagina de inicio	106
3.32. Pagina login	107
3.33. Pagina principal	108
3.34. Ver deterioros	108
3.35. Reportar deterioros paso 1	109
3.36. Reportar deterioros paso 2	110
3.37. Reportar deterioros paso 3	110
3.38. Reportar deterioros paso 4	111
3.39. Mis Deterioros reportados	112
3.40. Mis Deterioros reportados - ver detalle	112
3.41. Ver toda la lista de deterioros	113
3.42. Actualizar deterioro	113
3.43. Arquitectura e infraestructura del Sistema	114
3.44. Diagrama de Clases del sistema	117
3.45. Base de Datos del sistema	120
3.46. Estructura del Backend	122
3.47. Arquitectura del Backend	123

3.48. Frontend pagina de inicio	125
3.49. Frontend pagina de inicio ver deterioro	126
3.50. Frontend leyenda del mapa	127
3.51. Frontend Subir Imagen	128
3.52. Frontend Deterioros Identificados por la Red Neuronal Convolutcional . . .	128
3.53. Frontend Selección de Bounding Boxes	129
3.54. Frontend Metadatos Identificados	130
3.55. Frontend Ubicación de la Imagen	130
3.56. Frontend pagina para acceder al sistema	131
3.57. Frontend registro de usuario	132
3.58. Frontend menú del usuario poblador	133
3.59. Frontend modulo para salir del sistema	134
3.60. Frontend subir lote de imágenes	135
3.61. Frontend confirmación de subida de imágenes	135
3.62. Frontend lista deterioros personales	136
3.63. Frontend revisión deterioros personales	137
3.64. Frontend revisión deterioros personales bounding boxes	138
3.65. Frontend revisión deterioros personales metadatos	139
3.66. Frontend menú de usuario revisor	140
3.67. Frontend deterioros por usuarios	141
3.68. Frontend revisión deterioros globales	142
3.69. Frontend menú de usuario administrador	143
3.70. Frontend dibujar ruta	144
3.71. Frontend administrar usuarios	144
3.72. Frontend editar usuario	145

Índice de tablas

1.1. Tabla de Presupuesto	12
3.1. Registro de datos	54
3.2. Caso de uso visualización de deterioros en el mapa	71
3.3. Caso de uso filtrar deterioros	72
3.4. Caso de uso carga de fotos individuales	73
3.5. Caso de uso registrarse	74
3.6. Caso de uso iniciar sesión	75
3.7. Caso de uso cerrar sesión	76
3.8. Caso de uso cargar fotos por lotes	77
3.9. Caso de uso revisar deterioros personales	78
3.10. Caso de uso revisar comunitarios	79
3.11. Caso de uso filtrar deterioros por usuarios	80
3.12. Caso de uso marcar deterioros en mantenimiento	81
3.13. Caso de uso marcar deterioros reparados	82
3.14. Caso de uso administrar usuarios	83

Capítulo 1

Aspectos Generales

1.1. Planteamiento del Problema

1.1.1. Descripción del problema

El transporte terrestre es uno de los pilares fundamentales para el progreso del sistema económico de los países, ya que facilita la importación y exportación de productos hacia diferentes puntos poblacionales dentro de las naciones. Esto permite la creación de empleo, la generación de estabilidad y el crecimiento económico. Además, conecta a las personas con acceso a servicios especiales, como atención médica, educación y alimentación. Es por esta razón que la infraestructura vial es uno de los principales activos públicos de una nación. Por lo tanto, velar por su mantenimiento, reparación y mejora en cada una de las carreteras es responsabilidad y exigencia de los ciudadanos que hacen uso de ella y disfrutan de sus beneficios.

Los desperfectos en la infraestructura vial se deben a la disminución de la vida útil de las autopistas, debido al uso y a los efectos medioambientales. Es normal y común que existan con el paso del tiempo, comenzando como deterioros que son malformaciones en la superficie de rodadura. Al realizar un mantenimiento en este punto, se evita que empeoren y lleguen a convertirse en una falla. Una falla se describe como el empeoramiento de un deterioro que genera daños estructurales en las diferentes capas internas de la autopista, lo que puede llevar a la completa paralización operativa del elemento. Esto genera problemas en diferentes ámbitos:

En el ámbito económico, la identificación de deterioros en las autopistas permite que se les brinde mantenimiento lo antes posible y que no se agraven hasta convertirse en una falla. Esto reduce el presupuesto de inversión en mantenimiento y evita la paralización operativa completa de arterias en la infraestructura vial de un país, sin afectar el movimiento económico y los beneficios que se generan en estas.

En el ámbito sociocultural, permite crear una mejor imagen para el turismo, con calles bien pavimentadas, sin tráfico y con un mantenimiento preventivo que evita cualquier tipo de mala experiencia durante los viajes. Esto brinda igualdad de oportunidades

para todos, reduce la desigualdad e integra a todos los individuos para que disfruten de estos recursos en condiciones óptimas.

En el ámbito de la salud, según la OMS y la ONU, una de las principales causas de mortalidad en diversos países del mundo son las lesiones causadas por accidentes de tránsito terrestre OMS (2020). Los factores de riesgo incluyen la irresponsabilidad humana, la infraestructura vial, la falta de atención inmediata y la baja calidad en la revisión técnica de vehículos OMS (2022).

Los países más desarrollados del mundo, en busca de ciudades más seguras, eficientes y beneficiosas, se han centrado en mejorar sus autopistas. Han fomentado valores entre sus habitantes, implementado normativas de tráfico más eficaces Gerard Zwetsloot (2017) y construido carreteras e infraestructuras más confiables Johansson (2009). Cada uno de estos enfoques ha estado respaldado por tecnología, como sistemas de monitoreo de conductores CIPIA (2023), reconocimiento en tiempo real de vehículos Huansheng et al. (2019), detección del deterioro de señales de tráfico Kong et al. (2022) y detección de defectos en los pavimentos de las vías de tránsito Zhou et al. (2022). Estas implementaciones han arrojado resultados positivos en la reducción de accidentes y tasas de mortalidad, logrando que en el año 2019, en la ciudad de Oslo, capital de Noruega, no se registraran fallecimientos por accidentes de tránsito StreetsBlogUSA (2020). La Unión Europea mantiene su compromiso de continuar invirtiendo en sus autopistas y propone la implantación del proyecto denominado "Heron". Este proyecto implica un sistema automatizado y robusto para el monitoreo de autopistas, que hará uso de machine learning, redes neuronales, visión artificial en cámaras semafóricas, drones, robots y sensores automovilísticos Heron-Project (2021), con el objetivo de lograr un mejor control y una rápida identificación de problemas en sus vías de tránsito terrestre.

1.1.2. Identificación del problema

De acuerdo con la Ley N.º 27181 - Ley General de Transporte y Tránsito Terrestre Congreso de la República del Perú (1999), el Estado peruano es responsable del desarrollo, mantenimiento y conservación de la infraestructura vial, asegurando un transporte terrestre seguro, eficiente y accesible en todo el país. Este mandato cobra especial relevancia en el departamento del Cusco, donde la red vial conecta zonas agrícolas, centros turísticos de interés internacional, áreas de actividad minera y comunidades rurales que dependen del transporte terrestre para acceder a servicios básicos.

Cada año, el Perú destina recursos públicos a las entidades encargadas de velar por el estado de la infraestructura vial MTC (2022), como el Gobierno Regional del Cusco, los municipios provinciales y distritales, y el Ministerio de Transportes y Comunicaciones (MTC). Estas entidades, a través de los Institutos Viales Provinciales (IVP), las áreas municipales de mantenimiento y unidades ejecutoras como Pro-Vías Nacional y Pro-Vías Descentralizado, ejecutan proyectos de conservación vial conforme a sus planes operativos institucionales MTC (2023).

No obstante, persisten limitaciones críticas en la gestión vial. Por un lado, la identificación de deterioros aún se realiza de forma manual y visual, sin el respaldo de

tecnologías especializadas, lo que conlleva diagnósticos tardíos o imprecisos. Esto puede provocar que se prioricen tramos no estratégicos, en desmedro de aquellos que presentan mayor criticidad. Por otro lado, las entidades responsables operan bajo estructuras jerárquicas y burocráticas que ralentizan los procesos administrativos y limitan su capacidad de respuesta ante las solicitudes ciudadanas. Como consecuencia, los reportes de deterioros realizados por la población muchas veces no son atendidos con la debida oportunidad ni respaldados con una evaluación técnica adecuada.

Como resultado de ambos problemas (la deficiente identificación técnica y las limitaciones operativas en la atención), se produce una acumulación progresiva de daños en la infraestructura vial. La falta de intervenciones oportunas agrava el estado de las vías pavimentadas vehiculares, llegando a comprometer su transitabilidad e incluso forzando su cierre temporal. En estas condiciones, los trabajos de mantenimiento ya no son suficientes y se hace necesario ejecutar obras de reconstrucción, cuyo costo puede ser hasta ocho veces mayor que el del mantenimiento preventivo IPE (2008).

Esta situación contradice lo dispuesto en la Ley N.º 27293 - Ley del Sistema Nacional de Inversión Pública Congreso de la República del Perú (2000), que establece que toda inversión pública debe estar sustentada en información técnica confiable y oportuna. La deficiente identificación de los puntos críticos de la red vial incrementa el riesgo de asignar recursos a intervenciones de bajo impacto, disminuyendo la eficiencia del gasto público.

1.2. Formulación del Problema

1.2.1. Problema General

¿Cómo detectar y registrar deterioros en vías vehiculares pavimentadas a partir de imágenes capturadas y enviadas por usuarios desde sus dispositivos móviles?

1.2.2. Problemas Específicos

- No se dispone de un dataset público que contenga imágenes de deterioros viales (como grietas, baches o craquelados) en vías vehiculares pavimentadas de la ciudad Cusco, lo que representa una limitación para el entrenamiento adecuado de modelos de redes neuronales convolucionales orientados a la detección automática de dichos deterioros.
- La inspección de deterioros en vías pavimentadas suele depender de métodos visuales y manuales realizados por observadores humanos, lo que introduce subjetividad, variaciones en los resultados y limitaciones en el alcance. Esta situación dificulta la implementación de soluciones automatizadas y precisas para evaluar de manera eficiente el estado de la infraestructura vial en grandes extensiones territoriales.
- No se cuenta con una plataforma web accesible que permita a los usuarios registrar y compartir imágenes georreferenciadas de deterioros viales desde sus dispositivos,

dificultando la recolección colaborativa de información visual y espacial útil para el análisis vial.

- No se ha validado en condiciones reales la precisión de un modelo CNN ni la funcionalidad de un sistema web inteligente para detectar y georreferenciar deterioros viales, lo cual genera incertidumbre sobre su efectividad práctica en escenarios como la vía Pisac – Calca – Urubamba.

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar un sistema web inteligente que integre un modelo de red neuronal convolucional (CNN) para la detección y registro automático de deterioros en vías vehiculares pavimentadas, a partir de imágenes capturadas y enviadas por usuarios desde dispositivos móviles.

1.3.2. Objetivos Específicos

- Recopilar un dataset de imágenes que contenga diferentes tipos de deterioros viales (grietas, baches y craquelados) provenientes de vías pavimentadas de la ciudad del Cusco, asegurando variedad y calidad para el entrenamiento efectivo de una red neuronal convolucional (CNN).
- Diseñar, desarrollar y entrenar un modelo de red neuronal convolucional (CNN) para detectar, clasificar y localizar automáticamente los deterioros viales en imágenes, generando bounding boxes que señalen con precisión la ubicación de los daños.
- Implementar un sistema web inteligente que permita a los usuarios subir imágenes desde navegadores web, ya sea desde dispositivos móviles o computadoras, utilizando los metadatos para georreferenciar las imágenes y mostrar las detecciones sobre un mapa interactivo, facilitando el registro visual y geoespacial de los deterioros.
- Realizar pruebas y validación del prototipo web sobre la vía del departamento del Cusco (Pisac - Calca - Urubamba), evaluando la precisión de la detección de deterioros y la efectividad del sistema para asociar correctamente las imágenes con su ubicación geográfica, a través de los metadatos.

1.4. Justificación

1.4.1. Conveniencia

Las vías de tránsito terrestre son fundamentales para el desarrollo del Cusco, pero su constante deterioro afecta la seguridad, encarece el transporte y limita las oportunidades de muchas comunidades.

Es conveniente desarrollar este proyecto en el departamento del Cusco debido a la limitada implementación de soluciones tecnológicas por parte de las autoridades encargadas del mantenimiento vial. Esta carencia justifica la propuesta de un enfoque innovador y accesible, que se adapta a las condiciones locales y permite generar información útil desde la ciudadanía para mejorar la gestión de las vías. A diferencia de sistemas centralizados o costosos, este modelo puede ser implementado con recursos limitados y adaptado a distintos contextos, promoviendo así una solución sostenible y pertinente para la región.

1.4.2. Relevancia

Este proyecto es importante porque permite involucrar activamente a la ciudadanía en la mejora del estado de las vías. Su implementación no depende exclusivamente del Estado, sino que puede ser utilizada por organizaciones comunitarias, instituciones académicas o gobiernos locales comprometidos con el desarrollo vial. Además, promueve el control social y fortalece una cultura de transparencia y corresponsabilidad, facilitando intervenciones más oportunas, estratégicas y ajustadas a la realidad del Cusco. Con ello, se consolida como una alternativa viable para transformar la gestión de la infraestructura vial.

1.4.3. Implicancias Prácticas

Este proyecto puede aplicarse y adaptarse al uso de cualquier entidad interesada en la infraestructura vial del departamento del Cusco, ya que sus funcionalidades permiten registrar y clasificar daños viales con rapidez y bajo costo. Al generar información útil y organizada, el sistema contribuye a mejorar la administración del mantenimiento vial, optimizar la toma de decisiones y priorizar intervenciones. Además, el prototipo web brinda una herramienta práctica para ciudadanos y autoridades, permitiendo cargar imágenes, construir una base de datos histórica y fortalecer la gestión proactiva. Esta propuesta no solo mejora la eficiencia en el uso de recursos públicos, sino que también contribuye a elevar la calidad de vida en zonas urbanas y rurales.

1.4.4. Valor Teórico

Este proyecto adapta nuevas herramientas y procedimientos tecnológicos a una solución contextualizada en el departamento del Cusco. A partir de la construcción de

un dataset con información local, se desarrolla y entrena una red neuronal convolucional aplicada a un sistema funcional capaz de detectar deterioros en vías pavimentadas. Esta propuesta demuestra la aplicabilidad de la inteligencia artificial en contextos reales, donde la escasez de soluciones tecnológicas especializadas limita la eficiencia del mantenimiento vial. Además, integra visión por computadora y geolocalización en un entorno descentralizado, fortaleciendo la capacidad tecnológica local y aportando al campo académico mediante un enfoque práctico y replicable.

1.4.5. Utilidad Metodológica

La propuesta metodológica combina recolección ciudadana, análisis con redes neuronales y georreferenciación interactiva. Esta integración, basada en visión computacional, permite identificar fallas en la infraestructura vial y puede extenderse a otras áreas del mantenimiento urbano como señalización, semaforización, drenaje o fallas geográficas. El presente proyecto proporciona un ciclo completo para el desarrollo de un sistema inteligente, brindando una guía metodológica replicable en futuros trabajos. Esta puede ser aprovechada desde distintos puntos del proceso, ya sea construyendo una nueva red neuronal, generando un dataset, integrando un backend o reutilizando el modelo entrenado como una API. Su arquitectura escalable permite la incorporación de nuevos módulos, optimizando la gestión pública. Además, el modelo es adaptable a otras regiones con condiciones similares, demostrando el potencial de la inteligencia artificial para mejorar la administración de infraestructuras en contextos con limitaciones técnicas o presupuestarias.

1.5. Delimitación de estudio

1.5.1. Delimitación Espacial

La presente tesis se desarrollará en el departamento del Cusco, considerando la limitada implementación de soluciones tecnológicas por parte de las entidades responsables del mantenimiento de las vías de tránsito. Esta situación dificulta una atención adecuada a los problemas relacionados con el estado de las vías, lo que justifica la necesidad de proponer un enfoque innovador, adaptado a las condiciones locales. El sistema propuesto está diseñado para ser utilizado por cualquier entidad interesada en aprovechar los datos generados, contribuyendo así a una gestión más eficiente y sostenible de la infraestructura vial.

1.5.2. Delimitación Temporal

El proyecto se desarrollará con base en la información recopilada durante el año 2024, periodo en el que se confirmó la escasez de tecnología en las entidades previamente descritas. Esta situación se evidencia con mayor intensidad en determinadas zonas, las

cuales han sido identificadas en el Plan Anual del MTC 2023 y en la Memoria Descriptiva del Distrito Provincial del Cusco 2023, documentos que detallan específicamente los sectores con una alta concentración de deterioros en la infraestructura vial.

1.5.3. Delimitación Técnica

El presente trabajo se orienta al desarrollo de un prototipo funcional que permitirá identificar y recolectar información sobre deterioros en vías de transporte terrestre mediante redes neuronales convolucionales. El sistema se ejecutará como una aplicación web en un entorno local privado, con acceso limitado a dispositivos móviles dentro de la misma red interna. Para asegurar la comunicación cifrada, se utilizará un certificado HTTPS autogenerado, lo cual proporcionará un nivel adecuado de seguridad para la demostración del prototipo, aunque no se empleará un certificado oficial emitido por una autoridad certificadora.

Es importante señalar que, dado el carácter académico del trabajo, el sistema no será desplegado en un servidor público ni se integrará en un entorno de producción. La funcionalidad que se desarrollará se limitará a la validación técnica del enfoque propuesto, sin incluir módulos como la autenticación robusta, la gestión de usuarios ni mecanismos de escalabilidad. Cada uno de los objetivos específicos será abordado hasta la etapa de prototipado funcional, de manera que se cumplan los propósitos de demostración y validación dentro de un entorno controlado.

1.6. Método

1.6.1. Alcance

El presente trabajo es de tipo descriptivo, debido a que se enfoca en describir detalladamente las características del prototipo web propuesto y el proceso de su construcción. A través de la recolección de imágenes de deterioros viales, se documenta el desarrollo de un sistema basado en visión computacional que permite la detección, clasificación y georreferenciación de daños como grietas, baches y craquelados.

Asimismo, esta investigación busca representar de manera precisa el fenómeno del deterioro vial en el departamento del Cusco, sin intervenir directamente en el entorno, sino mediante la recopilación y análisis de información visual. Posteriormente, se analiza el comportamiento y la efectividad del prototipo en la fase de pruebas, contrastando los resultados con los lineamientos técnicos del Manual de Carreteras de Mantenimiento o Conservación Vial del Instructivo (2018).

De este modo, se sustenta como investigación descriptiva al centrarse en el estudio, análisis y representación de una realidad observable, tal como se establece en Hernández Sampieri (1997), permitiendo generar una visión clara del problema y de las herramientas tecnológicas utilizadas para abordarlo.

1.6.2. Diseño

El presente trabajo de tesis emplea un diseño no experimental, ya que no se realizará manipulación deliberada de variables independientes. En su lugar, se procederá a la observación y análisis de los resultados obtenidos a partir del desarrollo y prueba del prototipo, sin intervenir directamente en las condiciones del entorno o los datos recolectados.

Este diseño es coherente con el enfoque descriptivo del estudio, ya que se busca observar, registrar y analizar el comportamiento del sistema propuesto en condiciones reales, especialmente en la detección y georreferenciación de deterioros viales, sin alterar los factores que influyen en dicho fenómeno.

1.6.3. Para el desarrollo de la parte informática

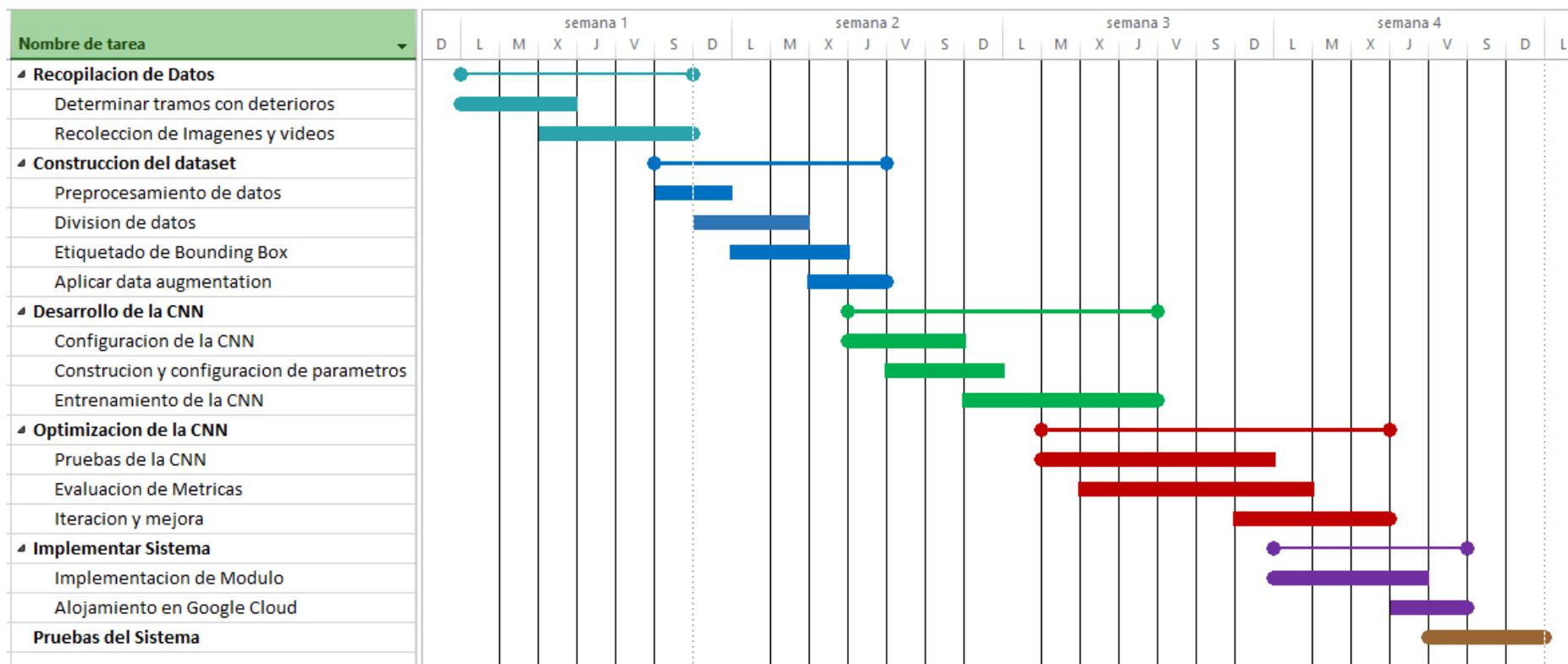
El desarrollo del proyecto se llevará a cabo en dos partes, utilizando dos metodologías complementarias. La primera parte consistirá en la construcción de la Red Neuronal Convolutiva (CNN) para identificar deterioros en imágenes de vías de tránsito terrestre. Esta parte del proyecto se realizará siguiendo la metodología estructurada en fases, basada en el enfoque general para el desarrollo de CNNs, que incluye la recopilación y preprocesamiento de datos, el diseño de la arquitectura de la red, el entrenamiento del modelo, su validación y optimización, y finalmente la integración con el prototipo web. Este enfoque se inspira en prácticas establecidas, como las descritas en el proyecto de Romero (2022), adaptadas específicamente a las necesidades del proyecto.

La segunda parte del proyecto se enfocará en el desarrollo del prototipo web, para lo cual se utilizará la metodología de prototipado evolutivo. Esta metodología permitió construir el sistema de forma incremental, partiendo de un conjunto inicial de requerimientos obtenidos mediante entrevistas con las entidades responsables del mantenimiento de vías de tránsito. A partir de dichas especificaciones, se elaboraron historias de usuario, casos de uso, fichas técnicas, diagramas de flujo, diagramas de secuencia y el diseño de las interfaces. Luego, se implementaron progresivamente los distintos componentes del sistema (backend, frontend y base de datos), ajustando y refinando el prototipo conforme se validaban los resultados parciales. Esta estrategia fue adecuada dada la limitada interacción continua con los usuarios finales, y facilitó el desarrollo de un sistema funcional, flexible y adaptado a las condiciones reales de la región del Cusco.

1.6.4. Cronograma de Actividades

En la Figura 1.1, se presenta el cronograma de actividades del proyecto. Este cronograma detalla las tareas principales junto con las subtareas que las desglosan, especifica sus respectivas duraciones y muestra la relación secuencial entre ellas. Además, permite visualizar la estructura jerárquica y el flujo de trabajo necesario para cumplir con los objetivos establecidos.

Figura 1.1: Cronograma del proyecto



Fuente: Elaboración propia.

1.6.5. Presupuesto

A continuación, se describen los recursos humanos, materiales y tecnológicos necesarios para el desarrollo e implementación del sistema propuesto. Estos recursos se han clasificado en categorías para facilitar su análisis y justificación.

Personal

- **Especialista en Base de Datos (BD):** Profesional encargado de colaborar en el trabajo de campo para la captura de imágenes. Su labor incluye la selección y curación de imágenes relevantes, con énfasis en la calidad visual y diversidad de deterioros. Posteriormente, se encarga del diseño y construcción de un dataset estructurado, así como de la implementación de una base de datos relacional optimizada (utilizando PostgreSQL) para el almacenamiento, consulta y vinculación eficiente de los datos recolectados con sus respectivos metadatos (georreferenciación, fecha, tipo de daño, etc.).
- **Programador en Redes Neuronales Artificiales:** Responsable del procesamiento de los datos de imagen, incluyendo su conversión al formato TFRecord necesario para su uso en TensorFlow. Lleva a cabo la configuración de los hiperparámetros, la arquitectura del modelo y el entrenamiento de la red neuronal convolucional (CNN) para la detección y clasificación automática de deterioros viales. Finalmente, implementa el modelo entrenado en un entorno funcional mediante una API REST utilizando Flask, facilitando su integración con otros componentes del sistema.
- **Programador Web:** Profesional especializado en el desarrollo de aplicaciones web. Su rol incluye el diseño e implementación tanto del frontend (interfaz de usuario amigable e intuitiva) como del backend (gestión de lógica, base de datos y servicios), evaluando librerías y frameworks adecuados. Además, se encarga de la integración de servicios como Leaflet para la visualización geoespacial, y asegura que la plataforma sea accesible desde navegadores web de distintos dispositivos (móviles y de escritorio).

Equipos

- **Cámara Fotográfica:** Dispositivo esencial para la captura de imágenes en campo, las cuales constituyen la base del entrenamiento del modelo de inteligencia artificial. Se requiere una cámara con buena resolución, preferentemente con capacidad de almacenar metadatos EXIF que incluyan coordenadas GPS para su posterior uso en la georreferenciación de las evidencias.
- **Computadora de Alto Rendimiento:** Equipo utilizado para tareas críticas como el procesamiento de imágenes, entrenamiento de modelos de aprendizaje profundo, construcción y administración de la base de datos, y programación de los módulos backend y frontend del sistema. Se requiere que cuente con características técnicas

adecuadas (procesador potente, memoria RAM suficiente, capacidad de almacenamiento y, de preferencia, una GPU dedicada).

- **Automóvil:** Medio de transporte utilizado para recorrer las rutas seleccionadas en la ciudad del Cusco durante la recolección de imágenes y para la fase de validación en campo del sistema. Se contempla el trayecto Pisac–Calca–Urubamba, con retorno incluido, por lo que el vehículo debe estar en condiciones óptimas de funcionamiento.

Herramientas de Software

- **Google Colab Pro:** Plataforma en la nube que permite acceder a máquinas virtuales con GPU, facilitando el entrenamiento de modelos complejos sin la necesidad de infraestructura local. Se contempla la suscripción al plan Pro para garantizar mayor disponibilidad, sesiones más largas y mayor capacidad de procesamiento, esenciales durante las fases intensivas del entrenamiento y validación del modelo.

Materiales

- **Combustible:** Insumo necesario para el funcionamiento del automóvil durante los recorridos de toma de datos y pruebas de campo. Se estima un consumo promedio en función de la distancia total y las condiciones del terreno.
- **Electricidad:** Recurso indispensable para el funcionamiento de los equipos utilizados en el desarrollo del proyecto, especialmente durante las jornadas de entrenamiento del modelo y pruebas del sistema. Se considera el consumo energético durante el periodo completo de ejecución.
- **Internet:** Servicio esencial para la descarga de librerías, documentación técnica, comunicación entre herramientas en la nube y acceso a servicios como Google Colab. Además, permite realizar pruebas remotas del sistema y garantizar la conectividad durante el desarrollo y despliegue.

Tabla 1.1: Tabla de Presupuesto

Categoría	Recurso	Descripción	Monto	Duración	Subtotal
Personal	Especialista en BD	Creación de dataset y BD	16.64 soles/hora	120 horas	S/. 1996.80
	Programador de ANN	Desarrollo de la ANN	24.53 soles/hora INDEED (2023)	112 Horas	S/. 2747.36
	Programador Web	Desarrollo Web	24.53 soles/hora INDEED (2023)	136 Horas	S/. 3336.08
Equipos	Cámara	Captura de Imágenes	4.1 soles/hora Visual (2023)	48 horas	S/. 200.16
	Computadora	Desarrollo del sistema	0.27 soles/hora LeaseIN (2023)	720 Horas	S/. 194.40
	Automóvil	Movilidad	282.75 soles/día KAYAK (2023)	4 días	S/. 1131.00
Herramientas de Software	Google Colab	Entrenamiento virtualizado	-	-	S/. 200.16
Materiales	Combustible	Movilidad	18.61/gal Peru (2023)	32 gal	S/. 595.00
	Electricidad	Energía eléctrica	-	-	S/. 500.00
	Internet	Programación remota	90 soles/mes	2 meses	S/. 180.00
				Total	S/. 11080.96

Capítulo 2

Marco Teórico

2.1. Antecedentes

2.1.1. Antecedentes Internacionales

Marcelo Fabrizio Letelier Palomares, (2021), *Detección de Grietas en la vía pública para prevenir caídas*, Universidad Andres Bello, Santiago de Chile.

Conclusiones:

- La Red Neuronal Pre-entrenada YOLO(You Only Look Once) - DarkNet53 favorece en gran medida la efectividad para el reconocimiento y detección de grietas en este proyecto.
- El proyecto llega a 2 alcances, uno que determina si en una imagen existen grietas o no, concluyendo únicamente a 2 respuestas, y la segunda que nos permite determinar las grietas dentro de un vídeo, especificando la existencia de las grietas y adicionalmente la posición de cada grieta.
- El proyecto se entreno en google Colab una herramienta que permite entrenar tu Red neuronal con una maquina virtual gratuita, lo que permite reducir tiempo de entrenamiento y pruebas.

Comentario: Para mejorar la precisión de la Red Neuronal es necesario utilizar una Red Neuronal Pre-entrenada o por lo menos una Híbrida. que nos permita facilitarnos la detección de objetos de manera óptima.

Marcelo Ignacio Orellana Espinoza, (2019), *Detección de Grietas mediante Deep Learning basado en imágenes en concreto*, Universidad de Chile, Chile.

Conclusiones:

- Al especificar las métricas de nuestra Red Neuronal es importante no generar Overfitting o sobre ajustarlo con muchas Epoch o épocas, debido a que estas pueden llegar

a tener un Accuracy muy bueno y óptimo pero solo eficiente para nuestros datos de entrenamiento. mas no para datos desconocidos que nunca vio nuestra Red Neuronal.

- Es importante diferenciar lo que hacen las Redes Neuronales Convoluciones que solo realizan un análisis predictivo y selectivo, y por otro lado lo que la visión computacional permite al ser predictivo, selectivo y localizador.
- Aplicar un mapa de calor a las imágenes permite determinar de mejor manera las zonas grietas en las imágenes.

Comentario: La Red Neuronal VGG es una opción viable en cuanto a redes Neuronales Pre-entrenadas, debido a que esta se entreno consumiendo la base de datos ImageNet, la cual ofrece una buena calidad para reutilizarla y obtener un optimización mas alta.

Julian Alberto Ortega Triana, (2021), *Aprendizaje profundo para la detección automática de fisuras de hormigón usando redes neuronales convolucionales*, Universidad Politécnica de Valencia, España.

Conclusiones:

- Los datos de entrenamiento tienen que ser diferentes a los de test, debido a que los de test deberían ser datos completamente desconocidos para la Red Neuronal entrenada.
- Es importante realizar el data Augmentation después de haber separado tu dataset en datos de entrenamiento y test.
- Transfer Learning permite acortar el ajuste de las métricas de evaluación como el learning rate, epochs, número de capas en la red neuronal, número de neuronas en cada capa, funciones que se utilizan dentro de estas, etc.

Comentario: El *Data Augmentation* es una técnica que permite masificar el *Dataset*, multiplicando su tamaño mediante el método de edición de imágenes que implica mas datos para entrenamiento, validación y test.

2.1.2. Antecedentes Nacionales

Ignacio Soto Percy Robustiano, (2022), *Identificación automática de las grietas en pistas de asfalto utilizando procesamiento digital de imágenes*, Universidad Señor De Sipán, Perú.

Conclusiones:

- El deterioro y mal estado de las vías de transporte se debe principalmente a condiciones medioambientales y las malas prácticas en construcción de pistas y asfalto.

- Las Redes Neuronales Convolucionales son el método mas óptimo y práctico para el procesamiento de imágenes e identificación de objetos.
- Es indispensable usar un Dataset limpio, ordenado y de buena calidad para obtener buenos resultados al momento de entrenar nuestra Red Neuronal.
- El aplicar el aumento de brillo y filtro de umbralizacion binario en las imágenes, permite ver mejor las grietas de asfalto. aportando a gran medida la identificación del modelo.

Comentario: El trabajo utiliza ciertas métricas al dividir su Dataset en 2 partes, una del 80 por ciento para entrenamiento y validación, otra del 20 por ciento para pruebas.

2.2. Bases Teóricas

2.2.1. Entidades encargadas de la gestión vial

Ministerio de transporte y comunicaciones (MTC)

El Ministerio de Transportes y Comunicaciones (MTC) es el principal responsable de la gestión y supervisión de las vías nacionales en el Perú. A través de este ministerio se planifican y ejecutan proyectos de infraestructura vial de gran importancia estratégica para el transporte interregional y la conexión entre las principales ciudades del país. El MTC también regula el transporte terrestre, garantizando la seguridad y eficiencia del sistema vial nacional. La Ley N.º 27181 - Ley General de Transporte y Tránsito Terrestre y el Decreto Supremo N.º 017-2009-MTC, Reglamento Nacional de Gestión de Infraestructura Vial, le otorgan las facultades necesarias para realizar estas funciones. Portal oficial del MTC (2023).

Provias Nacional Esta determinada como entidad adscrita al MTC, se encarga de la ejecución de proyectos de infraestructura vial en la Red Vial Nacional. Su función es asegurar el mantenimiento, rehabilitación y mejora de las carreteras nacionales, contribuyendo a la conectividad y desarrollo del transporte interregional. Esta entidad tiene su marco normativo en la Ley N.º 29089 - Ley de Infraestructura Vial, que regula la gestión de la infraestructura vial en el país. Portal oficial del MTC (2023)

Provias Descentralizado Esta adscrita bajo el MTC, tiene como misión apoyar a los gobiernos regionales en la ejecución de obras viales. Su trabajo se enfoca principalmente en las vías departamentales y provinciales, promoviendo la descentralización de la infraestructura vial y permitiendo que los gobiernos regionales asuman una mayor responsabilidad sobre las rutas que no forman parte de la Red Vial Nacional. Esta función se encuentra respaldada por la Ley N.º 29089 - Ley de Infraestructura Vial. Portal oficial del MTC (2023)

Municipios provinciales

Los municipios provinciales tienen competencias en el mantenimiento y gestión de las vías urbanas y rurales dentro de sus respectivas jurisdicciones. Estas entidades son responsables de asegurar que las vías urbanas, así como las de acceso a las zonas urbanas, se mantengan en condiciones adecuadas para la movilidad de las personas y el transporte de bienes. El marco legal que otorga estas competencias está establecido en la Ley N.º 27972 - Ley de Gobiernos Regionales y Locales, y en la Ley N.º 29089 - Ley de Infraestructura Vial.

Área de mantenimiento municipal La Sub-Gerencia de Mantenimiento de Infraestructura Pública es un órgano de línea de tercer nivel organizacional, responsable de programar y ejecutar las obras de mantenimiento, cautelando la correcta utilización de los presupuestos asignados y generando la información oportuna sobre el avance de los programas de mantenimiento. Municipalidad provincial del Cusco (2013)

Instituto Vial Provincial (IVP) El Instituto Vial Provincial Cusco - PROVIAS es un órgano descentralizado de la Municipalidad Provincial del Cusco y de las Municipalidades Distritales de la Provincia, con personería jurídica de Derecho Público Interno y con autonomía administrativa y económica, orientada a planificar y ejecutar la construcción, rehabilitación y mantenimiento de caminos rurales. Municipalidad provincial del Cusco (2013)

Empresas Privadas - Concesiones

Son otro aspecto fundamental en la gestión de las vías de tránsito terrestre. A través de contratos de concesión, empresas privadas se encargan de la construcción, operación y mantenimiento de ciertas carreteras a cambio de una contraprestación económica, generalmente en forma de peajes. Las concesiones permiten el desarrollo de infraestructura vial sin depender completamente del financiamiento estatal, lo que contribuye a la sostenibilidad de las vías. El marco legal para las concesiones viales está establecido en la Ley N.º 27181 - Ley General de Transporte y Tránsito Terrestre y la Ley N.º 29230 - Ley de Concesiones y Promoción de Infraestructura Vial. SUTRAN (2020)

2.2.2. Clasificación de las Vías Terrestres en el Perú: Marco Legal y Gestión de Infraestructura Vial

Vías Nacionales

Según la Ley N.º 27181 (Ley General de Transporte y Tránsito Terrestre), estas vías son gestionadas por el Ministerio de Transportes y Comunicaciones (MTC). Además, el Decreto Supremo N.º 017-2009-MTC (Reglamento Nacional de Gestión de Infraestructura Vial) establece que las vías nacionales son aquellas rutas que están incluidas en la

Red Vial Nacional. Su principal característica es que permiten la conexión entre regiones y sus principales ciudades, facilitando el tránsito de personas y mercancías a lo largo del país. Estas rutas son clave para el desarrollo económico y social, ya que contribuyen a la integración de diversas áreas geográficas y mejoran el acceso a servicios esenciales en diferentes regiones. Portal oficial del MTC (2023).

Vías Rurales

Las Leyes N.º 27181 (Ley General de Transporte y Tránsito Terrestre) y N.º 27783 (Ley de Bases de la Descentralización) establecen que las vías rurales están bajo las competencias de los gobiernos regionales y locales. Estas vías tienen como principal característica la interconexión de zonas rurales, facilitando el acceso a poblaciones alejadas que realizan actividades agrícolas. Su propósito es mejorar la infraestructura vial en áreas fuera de los centros urbanos, promoviendo el desarrollo económico y social de las comunidades rurales. A través de esta red de caminos, se busca mejorar el transporte de productos agrícolas, reducir la desigualdad en el acceso a servicios básicos y fomentar la integración de estas zonas con las principales ciudades. Este enfoque permite a los gobiernos regionales y locales tomar decisiones más adecuadas a las necesidades específicas de cada región, garantizando un mayor impacto en el desarrollo local. Municipalidad provincial del Cusco (2013).

Vías Urbanas

Según la Ley N.º 27972 (Ley Orgánica de Municipalidades), se determina que las municipalidades provinciales o distritales son las responsables de la jurisdicción de estas vías. Su principal característica se identifica por ser aquellas vías que se encuentran dentro de los límites urbanos, incluyendo calles, avenidas y pasajes, que permiten la circulación en ciudades y centros poblados. Estas vías son fundamentales para la movilidad cotidiana de los habitantes, facilitando el acceso a servicios, comercio y otras actividades urbanas. Además, las municipalidades tienen la responsabilidad de mantener y mejorar estas infraestructuras viales, velando por su conservación, señalización y adecuación a las necesidades del tránsito urbano. De esta manera, se busca garantizar la seguridad vial y un entorno adecuado para el desarrollo de las comunidades urbanas. Municipalidad provincial del Cusco (2013).

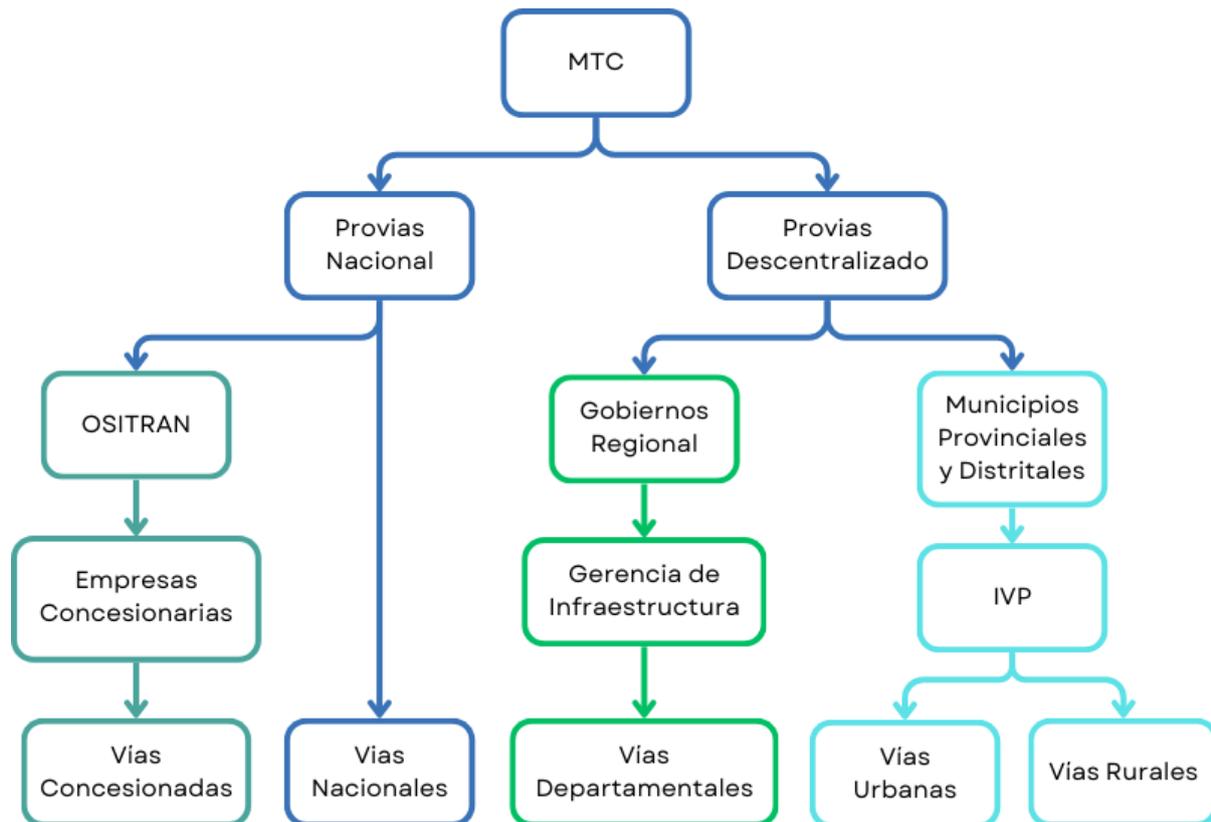
Vías Concesionadas

En el Perú, según las leyes Ley N.º 27181 (Ley General de Transporte y Tránsito Terrestre), Ley N.º 29338 (Ley del Sistema Nacional de Carreteras) y el Decreto Supremo N.º 059-96-PCM (Ley de Concesiones), se establece que algunas de las vías nacionales son entregadas a empresas privadas para su construcción, mantenimiento y explotación. Estas concesiones se realizan a cambio de peajes, con el objetivo de que las empresas privadas puedan recuperar su inversión a través de los ingresos generados por el cobro de estos peajes. Este modelo de concesión permite agilizar la infraestructura vial del país,

promoviendo la inversión privada en proyectos de gran escala, y contribuye a mejorar la calidad de las vías nacionales, garantizando su sostenibilidad a largo plazo. SUTRAN (2020).

Según la teoría especificada anteriormente, podemos visualizar en el siguiente gráfico Figura 2.1 cómo se organiza el manejo de las vías en el Perú. El gráfico detalla las entidades encargadas de su gestión y mantenimiento, especificando las responsabilidades de cada una según el tipo de vía.

Figura 2.1: Entidades y proceso de mantenimiento de las vías de tránsito



Fuente: Elaboración propia.

2.2.3. Metodología de Prototipado Evolutivo

El prototipado evolutivo es una metodología iterativa centrada en la construcción progresiva de un sistema mediante la entrega de versiones funcionales que se van refinando en función de observaciones técnicas o, cuando es posible, retroalimentación de los usuarios. Esta estrategia permite adaptar el desarrollo a contextos en los que los requisitos pueden cambiar o no estar completamente definidos desde el inicio, como ocurre con frecuencia en proyectos de innovación tecnológica vinculados a la gestión pública Pressman and Maxim (2014); Sommerville (2011).

A continuación, se describen las fases del prototipado evolutivo:

Recopilación inicial de requisitos

En esta fase se identifican y documentan las necesidades funcionales básicas del sistema desde la perspectiva del usuario final. Para ello, se emplea la técnica de elaboración de historias de usuario, que permite capturar los requerimientos en un lenguaje comprensible y orientado al uso real del sistema. Estas historias sirven como punto de partida para construir los casos de uso, los cuales describen de manera estructurada las interacciones entre los actores y el sistema.

A partir de estos casos de uso, se desarrollan fichas detalladas que incluyen los objetivos, actores involucrados, precondiciones, flujo de eventos y posibles excepciones. Asimismo, se elaboran los diagramas de flujo y de secuencia correspondientes, con el fin de representar gráficamente el comportamiento dinámico del sistema y facilitar una comprensión integral de los procesos clave. Este enfoque permite establecer una base sólida y coherente para las etapas posteriores del desarrollo Budde et al. (1992).

Diseño del primer prototipo funcional

Una vez definidos y validados los requisitos, se da inicio al diseño del primer prototipo funcional del sistema. Esta etapa comprende la elaboración de las interfaces gráficas preliminares, representadas mediante mockups y pantallas base, las cuales permiten visualizar la disposición de los elementos y la interacción esperada por parte del usuario.

Paralelamente, se diseña la arquitectura general del sistema, que incluye la estructura del frontend, el backend y el modelo de datos para la base de datos. Esta arquitectura garantiza la viabilidad técnica y la escalabilidad del sistema en futuras etapas de desarrollo.

El resultado de esta fase es una primera versión funcional del prototipo, que si bien no cuenta con todas las funcionalidades completas, integra de manera parcial y coherente los módulos clave del sistema. Esto permite validar tempranamente los aspectos esenciales de la solución, como la navegación, la usabilidad y la comunicación entre componentes.

Implementación iterativa y mejora continua

Durante esta etapa se adopta un enfoque de desarrollo incremental, en el cual los distintos módulos del sistema son construidos e integrados progresivamente a través de ciclos iterativos. Este enfoque permite validar avances parciales, recibir retroalimentación oportuna y realizar ajustes técnicos conforme surgen nuevas necesidades o se detectan posibles mejoras.

Los módulos desarrollados incluyen:

- **El backend:** encargado de la lógica del servidor, la gestión de peticiones y el procesamiento de datos, así como la conexión con la base de datos.

- **El frontend:** que proporciona la interfaz de usuario, permitiendo la interacción con el sistema de forma intuitiva y accesible desde distintos dispositivos.
- **El módulo de procesamiento de imágenes basado en redes neuronales convolucionales (CNN):** responsable de la identificación automática de tipos de deterioro vial a partir de las imágenes subidas por los usuarios. Este componente constituye el núcleo del sistema inteligente de detección.
- **El componente de georreferenciación y visualización en mapas interactivos:** permite ubicar espacialmente cada reporte de deterioro, facilitando el análisis territorial y la planificación de intervenciones por parte de las autoridades.

Cada iteración permite no solo añadir nuevas funcionalidades, sino también optimizar el rendimiento general del sistema, corregir errores detectados y ajustar la experiencia de usuario, garantizando así una evolución constante del prototipo hacia una solución más robusta y eficiente Sommerville (2011).

Evaluación y consolidación del sistema

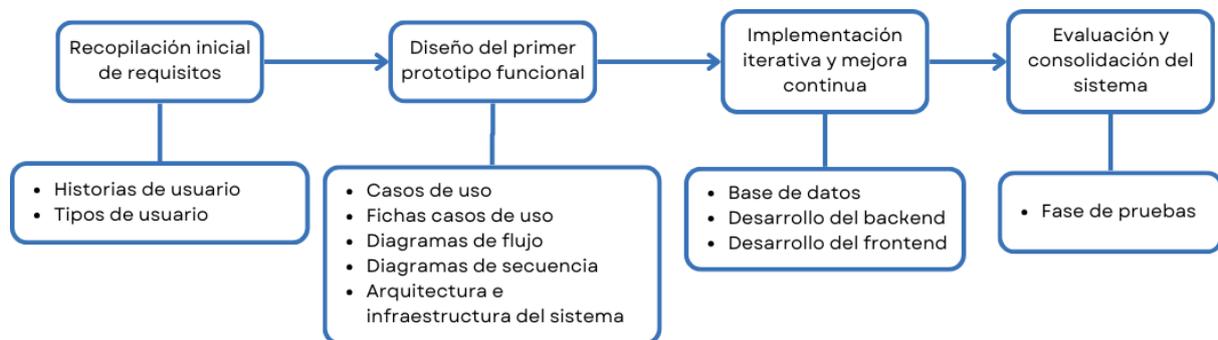
Tras varias iteraciones de mejora y ajustes funcionales, se alcanza una versión consolidada del sistema, la cual es sometida a un proceso de evaluación interna. Esta etapa tiene como objetivo validar la estabilidad, coherencia y desempeño general de la herramienta desarrollada.

Las pruebas internas permiten identificar posibles fallos remanentes, verificar la integración entre módulos (backend, frontend, procesamiento de imágenes y georreferenciación), y asegurar que el sistema cumpla con los requisitos previamente establecidos.

Si bien la participación directa de las entidades responsables del mantenimiento vial fue limitada en esta fase, se plantean escenarios de uso realista que simulan su intervención, con el fin de anticipar su experiencia y validar la utilidad práctica del prototipo. Esta evaluación sienta las bases para futuras pruebas piloto en campo y una eventual implementación oficial.

En la Figura 2.2 se muestra una representación simplificada del proceso correspondiente a la metodología de prototipado evolutivo, la cual fue aplicada en el desarrollo del sistema presentado en este proyecto, abarcando cada una de sus etapas hasta la obtención del producto final. Este enfoque permitió construir un sistema flexible y escalable, con potencial de adaptación a otros contextos de mantenimiento urbano y replicabilidad en diversas zonas del país que enfrenten desafíos similares en la gestión de infraestructuras públicas, como ocurre en el departamento del Cusco.

Figura 2.2: Diagrama descriptivo de la ejecución de la metodología de prototipado evolutivo



Fuente: Elaboración propia.

2.2.4. Deterioros en vías vehiculares pavimentadas

El término deterioro vial hace referencia al conjunto de alteraciones físicas que afectan la estructura o el funcionamiento de una vía de vías vehiculares pavimentadas, ya sea en el pavimento, la superficie de rodadura o en sus capas estructurales subyacentes. Estos deterioros suelen deberse a múltiples factores, entre los que se encuentran las condiciones climáticas, el tránsito vehicular intenso, deficiencias en el diseño o construcción, y la falta de mantenimiento adecuado Garber and Hoel (2009).

La identificación y clasificación de los deterioros en las vías resulta crucial para garantizar una gestión eficiente del mantenimiento vial. Existen diversas tipologías de deterioros según su forma, causa y evolución; sin embargo, en el presente estudio se consideran aquellos más frecuentes y representativos en vías asfaltadas, los cuales pueden ser detectados automáticamente mediante técnicas de visión por computador. A continuación, se describen los principales:

Grietas

Las grietas mostradas en las Figuras 2.3 y 2.4 son fisuras lineales que se manifiestan en la superficie del pavimento como resultado de diversos factores mecánicos y ambientales. Entre las principales causas se encuentran los esfuerzos de tracción generados por cargas vehiculares repetitivas, la fatiga acumulada en los materiales del pavimento, la contracción térmica debido a variaciones de temperatura, así como asentamientos diferenciales en la base o subrasante.

Estas grietas pueden presentarse en distintas formas, siendo las más comunes las grietas longitudinales (paralelas al eje de la vía) y las transversales (perpendiculares al eje). También pueden derivar en grietas en bloque o en forma de red si no se tratan oportunamente, lo cual compromete seriamente la integridad estructural del pavimento. Su presencia facilita la infiltración de agua y materiales contaminantes, lo que acelera el deterioro y reduce la vida útil de la vía.

Por ello, la detección temprana y el registro adecuado de las grietas constituyen una parte fundamental para la planificación de tareas de mantenimiento correctivo o preventivo, especialmente en zonas donde el tránsito es constante y los recursos para rehabilitación son limitados.

Figura 2.3: Ejemplo grieta limacpampa - cusco



Fuente: Elaboración propia.

Figura 2.4: Ejemplo grieta avenida el sol - cusco



Fuente: Elaboración propia.

Baches

Los baches mostrados en las Figuras 2.5 y 2.6 son pérdidas localizadas del material superficial del pavimento, que generan cavidades de diversas dimensiones perceptibles al tránsito vehicular y peatonal. Su presencia afecta significativamente la comodidad y seguridad de la circulación, además de incrementar los costos de operación de los vehículos y el riesgo de accidentes.

Generalmente, los baches se originan como consecuencia de otras fallas estructurales no tratadas a tiempo, especialmente las grietas. Estas permiten la infiltración de agua hacia las capas inferiores del pavimento, debilitando su estructura y provocando desprendimientos del material superficial debido a la acción combinada del tránsito, la humedad y los cambios de temperatura.

Una vez formado el bache, su deterioro tiende a acelerarse si no se interviene de manera oportuna, extendiéndose tanto en profundidad como en superficie. Por ello, su detección temprana y reparación adecuada es fundamental para evitar la propagación del daño y reducir los costos asociados a rehabilitaciones mayores. En contextos urbanos y regionales como el Cusco, donde las condiciones climáticas y topográficas pueden agravar estos efectos, la atención a los baches representa una prioridad en la gestión del

mantenimiento vial.

Figura 2.5: Ejemplo bache avenida grau - cusco



Fuente: Elaboración propia.

Figura 2.6: Ejemplo bache avenida circunvalacion - cusco



Fuente: Elaboración propia.

Craquelados

El craquelado mostrado en la Figura 2.7 es una forma de agrietamiento superficial caracterizada por la aparición de una red densa de pequeñas fisuras interconectadas, que forma un patrón visual similar al de la piel de un cocodrilo, de ahí que también sea conocido como agrietamiento tipo “piel de cocodrilo”.

Este tipo de deterioro es un indicativo de una falla estructural avanzada en el pavimento y suele estar asociado a la fatiga del material por cargas repetidas, especialmente en zonas de tráfico constante o de alto peso vehicular. Asimismo, puede deberse a una pérdida progresiva del soporte estructural de las capas inferiores del pavimento, ya sea por compactación deficiente, materiales inadecuados o filtraciones de agua que debilitan la base o subbase.

El craquelado representa una etapa crítica en el deterioro del pavimento, ya que tiende a evolucionar rápidamente hacia fallas más severas como desprendimientos del material o formación de baches si no se realiza un tratamiento adecuado. En regiones como el Cusco, donde factores climáticos, geográficos y de carga vehicular pueden agravar la condición estructural de las vías, el monitoreo temprano de este tipo de daño es esencial para priorizar intervenciones de mantenimiento y evitar su progresión.

Figura 2.7: Ejemplo craquelado corao - cusco



Fuente: Elaboración propia.

La detección automática de estos deterioros contribuye a establecer un diagnóstico preliminar del estado de la infraestructura vial, facilitando la planificación de intervenciones de mantenimiento preventivo y correctivo con mayor eficiencia Fernandes et al. (2022).

2.2.5. Inteligencia Artificial

Según Rouhiainen (2018), la inteligencia artificial se define como "la habilidad de los ordenadores para realizar actividades que normalmente requieren inteligencia humana". Esta definición describe el proceso de simular comportamientos inteligentes propios del ser humano, sin que sea necesario que cualquier máquina o algoritmo posea realmente una algún tipo de inteligencia.

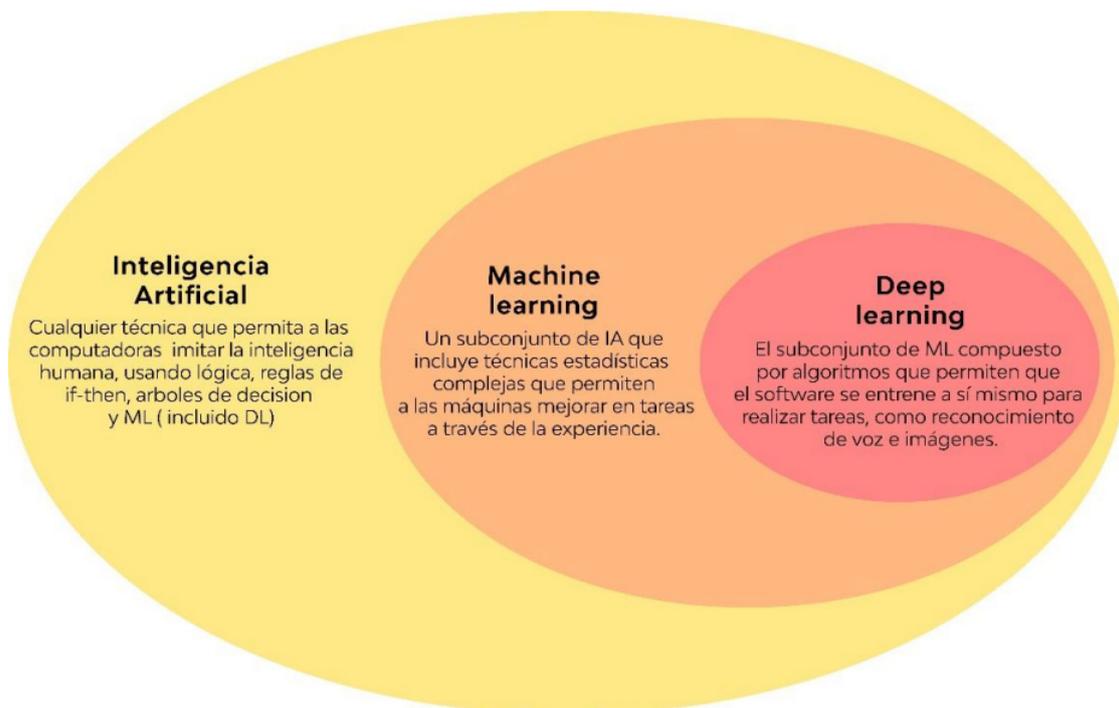
Machine Learning

También conocido como aprendizaje automático, Murphy (2012) lo describe como una rama de la inteligencia artificial con la capacidad de utilizar datos, identificar pa-

trones y estadísticas para aprender, sin ningún tipo de intervención humana. A partir de esta información, el sistema es capaz de tomar decisiones, mejorando continuamente su rendimiento y facilitando respuestas más precisas a medida que procesa más datos.

Esta jerarquía muestra en la Figura 2.8 cómo cada nivel es una especialización del anterior, con Deep Learning siendo una técnica avanzada dentro del ámbito de Machine Learning, y ambos, a su vez, formando parte de la mayor disciplina que es la Inteligencia Artificial.

Figura 2.8: Estructura de la Inteligencia Artificial



Fuente: Goodfellow et al. (2016)

2.2.6. Dataset

Es una recopilación de datos tabulados almacenados en cualquier sistema de almacenamiento de datos estructurados, que se utilizan para evaluar modelos de aprendizaje automático. Cada columna representa una variable y cada fila corresponde a un dato específico que estamos tratando, lo que facilita la organización y el análisis de grandes volúmenes de información. Estos datos pueden estar etiquetados o no etiquetados, dependiendo del objetivo o meta que se quiera alcanzar en el proceso de aprendizaje. Si los datos están etiquetados, se utilizan en el aprendizaje supervisado, donde el modelo aprende a partir de ejemplos con resultados conocidos, lo que permite hacer predicciones o clasificaciones más precisas. En cambio, si los datos no están etiquetados, se utilizan en el aprendizaje no supervisado, donde el modelo busca patrones o estructuras subyacentes en los datos sin la guía de etiquetas predefinidas, lo que resulta útil en tareas como la segmentación de clientes o la detección de anomalías. Dependiendo de la naturaleza de los datos y del problema a resolver, la elección entre aprendizaje supervisado o no su-

pervisado puede tener un impacto significativo en los resultados y en la capacidad del modelo para generalizar Goodfellow et al. (2016).

- **Imágenes sin deterioros** Figura 2.9, que representan superficies en buen estado o sin irregularidades significativas.
- **Imágenes con deterioros** Figura 2.10, que corresponden a aquellas que contienen defectos visibles en las superficies analizadas.

Figura 2.9: Dataset con deterioros



Fuente: Elaboración propia

Figura 2.10: Dataset sin deterioros



Fuente: Elaboración propia

Cada uno de estos conjuntos es organizado y etiquetado manualmente para garantizar la calidad de los datos utilizados en el modelo.

Training Dataset

Es el conjunto de datos que permitirá ajustar un modelo durante el proceso de entrenamiento de una red neuronal. Estos datos son utilizados para ajustar los parámetros internos del modelo, como los pesos y los sesgos, con el fin de que el modelo pueda aprender a partir de los ejemplos proporcionados. Durante el entrenamiento, el modelo toma como entrada los parámetros (mis datos) y genera como salida las respuestas (mis resultados esperados). A través de este proceso, el modelo ajusta continuamente sus parámetros internos para minimizar el error entre la salida predicha y la salida real. A medida que se realiza este ajuste, el modelo calcula su nivel de eficiencia o mejoramiento, evaluando su capacidad para hacer predicciones precisas al ser expuesto a nuevos datos durante el entrenamiento. Este proceso de ajuste y retroalimentación es clave para que el modelo aprenda de los datos y pueda generalizar su conocimiento a situaciones no vistas previamente Goodfellow et al. (2016).

Validation Dataset

Es el conjunto de datos utilizado para evaluar y monitorear el modelo en cada ciclo de su entrenamiento, hasta concluir en una época. Lo importante de estos datos es que no intervienen en el ajuste de los nodos del modelo, sino que se emplean exclusivamente para identificar si el modelo está mejorando o empeorando a lo largo del proceso de entrenamiento. Este conjunto de datos se usa como una herramienta para evaluar el rendimiento del modelo en tiempo real, permitiendo identificar tendencias de mejora o deterioro. Esto es crucial para determinar el momento adecuado para detener el entrenamiento, evitando el sobreajuste de los nodos, un problema que puede ocurrir cuando el modelo se ajusta demasiado a los datos de entrenamiento y pierde capacidad de generalización. Durante cada evaluación, estos datos analizan el rendimiento del modelo al comparar los parámetros de entrada y salida, calificando sus respuestas correctas e incorrectas. Este análisis proporciona información valiosa sobre qué tan bien está aprendiendo el modelo. De esta manera, se mide la eficiencia del modelo y se controla su capacidad de generalización, asegurando que sea capaz de hacer predicciones precisas sobre datos no vistos previamente, y no solo sobre los datos con los que fue entrenado Chollet (2017).

Test Dataset

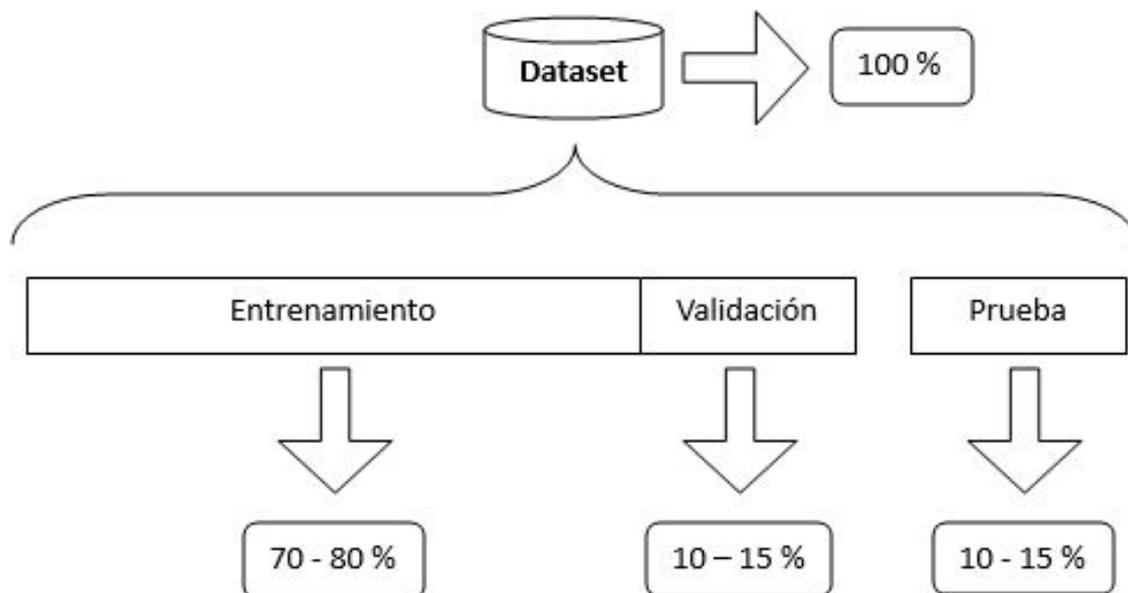
Es la prueba final del modelo o el estándar de oro para evaluarlo. Esta se realiza una sola vez, una vez completado todo el entrenamiento y las validaciones, con el objetivo de obtener las métricas de evaluación y determinar si el modelo es efectivo. Este conjunto de datos se utiliza para verificar cómo el modelo se desempeña en situaciones desconocidas, evaluando su capacidad de generalización a nuevos datos que no ha visto antes. Es importante resaltar que este conjunto de datos no se utiliza para ajustar el modelo, ya sea en sus parámetros o hiperparámetros, ya que sus valores son imparciales y desconocidos para el modelo al final del entrenamiento. Su propósito es garantizar que el modelo pueda hacer predicciones precisas fuera del conjunto de datos con el que fue entrenado, asegurando que no haya sobreajuste. Además, se emplea para comparar diferentes modelos entrenados y determinar cuál es el más eficiente, permitiendo seleccionar el mejor modelo entre los evaluados. Este proceso es fundamental para decidir qué modelo es el más adecuado para implementar en situaciones reales, donde los datos futuros aún no están disponibles. Towardsdatascience (2023)

En la Figura 2.11, se muestra la distribución típica de un dataset dividido en tres partes fundamentales para el entrenamiento de un modelo de machine learning: entrenamiento, validación y pruebas.

El 70-80 % del dataset se utiliza para el **entrenamiento**, que es la parte en la que el modelo ajusta sus parámetros internos a través de la exposición a los datos. Esta fase es clave para que el modelo aprenda las características y patrones de los datos.

El 10-15 % del dataset se reserva para la **validación**. Durante el proceso de entrenamiento, el conjunto de validación se utiliza para evaluar el rendimiento del modelo y ajustar los hiperparámetros, como el número de épocas o el tamaño del batch. Además, el conjunto de validación ayuda a identificar problemas como el sobreajuste (overfitting),

Figura 2.11: Dataset



Fuente: Elaboración propia

donde el modelo se ajusta demasiado a los datos de entrenamiento y pierde su capacidad de generalizar.

Finalmente, el 10-15 % restante del dataset se usa para las **pruebas**, con el objetivo de evaluar el modelo después del entrenamiento y la validación. Este conjunto nunca se utiliza durante el entrenamiento, por lo que proporciona una medida imparcial del rendimiento del modelo en datos no vistos previamente.

Esta división asegura que los modelos se entrenen, ajusten y evalúen de manera adecuada, permitiendo una evaluación robusta de su capacidad para generalizar a datos nuevos y no vistos Raschka and Mirjalili (2019).

Data Augmentation

Data augmentation es un estándar de regularización, una técnica utilizada para mejorar el rendimiento de los datos originales en un dataset, especialmente cuando se cuenta con una cantidad pequeña o limitada de datos. Consiste en generar más datos mediante perturbaciones en las imágenes del dataset, como rotaciones, traslaciones, cambios de escala, adición de ruido, recortes, entre otros. De esta manera, se aumenta la diversidad y cantidad de datos en el conjunto y se reduce el riesgo de sobreajuste al momento de entrenar el modelo. Guida (2023)

La Figura 2.12 muestra una única imagen que será sometida a técnicas de data augmentation. Finalmente, en la Figura 2.13 se presentan varias versiones de la imagen original, cada una modificada mediante diferentes operaciones de aumento de datos. Las

transformaciones ilustradas son las siguientes:

Figura 2.12: Imagen sometida a Data Augmentation.



Fuente: Elaboración propia basada en imagen Noticias RPP
(<https://andina.pe/agencia/noticia-cierran-carretera-cuscovalle-sagrado-danos-plataforma-de-via-746247.aspx>).

- Rotación: La imagen es rotada en distintos ángulos, lo que simula variaciones en la orientación de los objetos. Esta transformación permite al modelo aprender a reconocer los objetos desde diferentes perspectivas.
- Traslación: La imagen es desplazada en direcciones horizontales y verticales, mostrando cómo el objeto puede cambiar de posición dentro de la imagen. Esta transformación ayuda al modelo a generalizar situaciones donde el objeto no está centrado o tiene distintas posiciones.
- Escalado (Zoom): En algunos ejemplos, la imagen es escalada, acercando o alejando el objeto. Esto permite que el modelo sea capaz de identificar objetos a diferentes escalas o distancias.
- Recorte (Crop): Se recorta una porción de la imagen, dejando solo una parte del objeto. Esta transformación simula situaciones donde el objeto podría estar parcialmente visible o fuera del campo de visión completo.
- Adición de Ruido: Se introduce ruido aleatorio en las imágenes, lo que simula distorsiones que podrían ocurrir en condiciones reales de captura. Este proceso hace que el modelo sea más robusto a imperfecciones en los datos.

Figura 2.13: Resultado de utilizar Data Augmentation.



Fuente: Elaboración propia basada en imagen Noticias RPP
 (<https://andina.pe/agencia/noticia-cierran-carretera-cuscovalle-sagrado-danos-plataforma-de-via-746247.aspx>).

Estas transformaciones se realizan a partir de una sola imagen original, generando variaciones que aumentan la diversidad del conjunto de datos de entrenamiento sin la necesidad de recopilar más imágenes. Cada una de estas modificaciones tiene como objetivo mejorar la capacidad del modelo para generalizar y reducir el riesgo de sobreajuste, entrenando con datos más variados.

2.2.7. Deep Learning

También conocido como aprendizaje automático, el Deep Learning es un tipo de machine learning que se caracteriza por el uso de redes neuronales profundas. Según Goodfellow et al. (2016), estas redes emplean capas de aprendizaje, donde cada capa aprende una característica específica del dato para, finalmente, reconocer o identificar patrones más complejos. Gracias a esta estructura jerárquica, las redes neuronales profundas están diseñadas para reconocer patrones complejos en grandes volúmenes de datos, lo que las hace especialmente efectivas para tareas como la clasificación de imágenes, el procesamiento de lenguaje natural y la detección de anomalías.

El Deep Learning no solo es un tipo de machine learning, sino que también entrena a una computadora para realizar tareas que imitan las acciones humanas, tales como el reconocimiento del habla, la identificación de imágenes o la realización de predicciones. A diferencia de otros enfoques, en lugar de organizar los datos para que se ejecuten a través de ecuaciones predefinidas, el Deep Learning configura parámetros básicos sobre los datos y entrena a la computadora para que aprenda por cuenta propia, reconociendo patrones mediante el uso de muchas capas de procesamiento SAS (2023).

Redes Neuronales Artificiales (ANN)

Una red neuronal artificial (ANN) es un esquema de computación distribuida inspirada en la estructura del sistema nervioso de los seres humanos. La arquitectura de una red neuronal es formada conectando múltiples procesadores elementales, siendo éste un sistema adaptivo que posee un algoritmo para ajustar sus pesos (parámetros libres) para alcanzar los requerimientos de desempeño del problema basado en muestras representativas. Salas (2023)

Redes Neuronales Convolucionales (CNN)

Las redes neuronales convolucionales (CNN, por sus siglas en inglés) son una arquitectura avanzada de redes neuronales profundas, diseñadas para procesar y analizar datos en forma de cuadrícula, como los empleados en un mapa de bits. Estas redes están conformadas por diversas capas que extraen características simples en las primeras capas y, progresivamente, identifican patrones más complejos en capas posteriores. Esta estructura jerárquica las convierte en una herramienta poderosa para tareas de reconocimiento de objetos en imágenes y videos, ya que son capaces de identificar detalles específicos, como bordes, texturas y formas, para finalmente clasificar o localizar objetos dentro de una imagen.

Gracias a su capacidad para aprender representaciones jerárquicas de los datos, las CNN han demostrado ser fundamentales en aplicaciones como la visión por computadora, el procesamiento de imágenes, el reconocimiento facial, y la inteligencia artificial en general. Estos avances han permitido que las CNN se utilicen en una variedad de campos, desde la clasificación automática de imágenes hasta la detección de anomalías, haciendo posible la automatización de tareas que anteriormente requerían intervención humana LeCun et al. (2020).

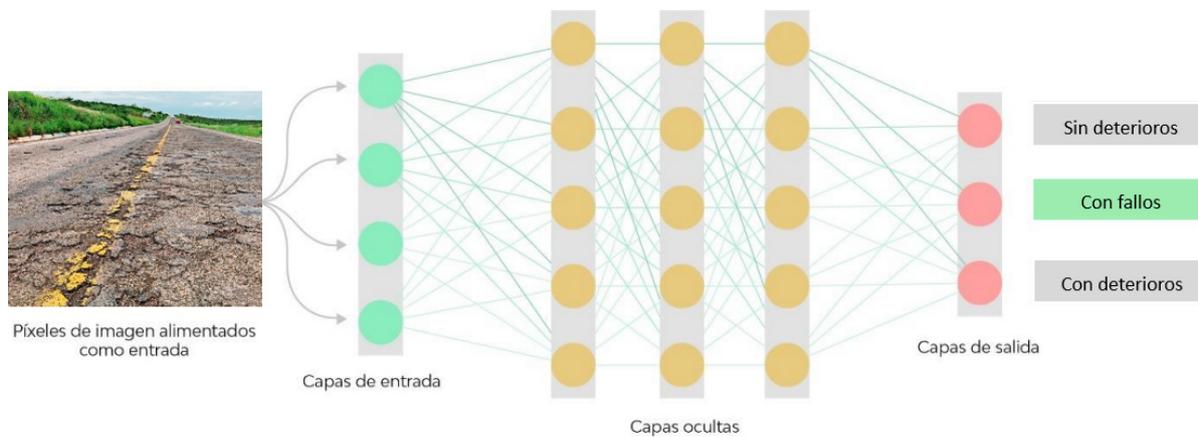
La Figura 2.14 muestra el funcionamiento de una red neuronal diseñada para detectar deterioros en una autopista. En el gráfico, se observa cómo una imagen de la autopista, que contiene grietas e imperfecciones, se ingresa como entrada a la red neuronal. Esta imagen se transforma en datos que se distribuyen a través de los nodos de la capa de entrada, donde cada nodo representa una característica relevante de la imagen, como los bordes, las texturas o las áreas con defectos visibles.

A medida que los datos pasan por la red, atraviesan varias capas ocultas, donde se realizan procesos de aprendizaje y extracción de características más complejas. Estas capas permiten que la red neuronal detecte patrones y relaciones entre los diferentes elementos de la imagen, como la ubicación y el tipo de deterioro presente en la autopista. Las capas ocultas permiten al modelo aprender a reconocer las características importantes que diferencian una zona deteriorada de una zona sin daños.

Finalmente, los datos llegan a las capas de salida, que en este caso están compuestas por tres nodos. Cada nodo corresponde a una categoría específica para la clasificación de la imagen: uno para "deterioros", otro para "fallos" y otro para "sin deterioros". El nodo con el valor más alto determinará el resultado final de la red neuronal, es decir,

si la imagen muestra zonas con deterioros, con fallos, o si no presenta ninguno de estos problemas.

Figura 2.14: Redes Neuronales Convolucionales



Overfitting

En español, el término "sobreajuste" se refiere a la situación en la que un modelo se ajusta excesivamente al conjunto de datos de entrenamiento, logrando un rendimiento muy eficiente al prácticamente memorizar estos datos. Esto sucede cuando el modelo captura tanto las tendencias generales como el ruido y las peculiaridades específicas del conjunto de entrenamiento, lo que no es representativo del comportamiento real de los datos.

Sin embargo, como consecuencia, el modelo no logra generalizar ni aplicar esta eficiencia a datos nuevos y desconocidos, lo que resulta en un desempeño deficiente. Este problema es especialmente común en modelos complejos, como las redes neuronales profundas, cuando no se aplican técnicas de regularización adecuadas.

Por esta razón, es fundamental emplear métodos de generalización, como el data augmentation y el entrenamiento mediante validación cruzada Géron (2019a).

Figura 2.15: Overfitting



En la Figura 2.15, se presenta una representación gráfica de un modelo de regresión que experimenta el fenómeno de overfitting (sobreajuste). En esta figura, los puntos representan los datos de entrenamiento, mientras que la curva o línea muestra el modelo ajustado a esos datos.

Como se puede observar, el modelo está excesivamente ajustado a los puntos de entrenamiento, lo que significa que la curva se adapta de manera forzada a cada uno de ellos. Este ajuste excesivo provoca que el modelo no capture correctamente la tendencia general de los datos. En lugar de reconocer patrones subyacentes, el modelo se ajusta a las fluctuaciones o el ruido presente en los datos, lo que genera una curva que pasa por todos los puntos. Como consecuencia, algunos puntos que deberían estar cerca de la curva, y que representan patrones reales, se perciben como alejados de ella.

Underfitting

Es definido como "infraajuste", una condición en la que un modelo no logra capturar las características, patrones o la estructura del conjunto de datos de entrenamiento, resultando en un bajo rendimiento tanto en las fases de validación como de prueba. Este fenómeno refleja que el modelo es incapaz de aprender lo necesario para representar correctamente la relación entre las variables de entrada y salida.

El infraajuste ocurre generalmente cuando el modelo es demasiado simple para la complejidad de los datos. Esto puede deberse al uso de un conjunto de datos reducido, lo que limita la capacidad del modelo para encontrar patrones relevantes, o porque los hiperparámetros no se ajustan de manera adecuada durante el entrenamiento.

Además, el infraajuste puede surgir al utilizar un modelo con una arquitectura insuficiente, donde el número de parámetros o capas no es capaz de captar la complejidad de los datos. En estos casos, el modelo no solo tiene un desempeño deficiente en los datos de prueba, sino que también falla en los datos de entrenamiento, lo que señala una necesidad de mejorar su diseño o configuración.

El infraajuste es una situación que debe evitarse, ya que compromete la capacidad del modelo para realizar predicciones precisas y confiables en contextos reales Murphy (2012). Medium (2023)

En la Figura 2.16, se muestra una representación gráfica de un modelo que experimenta el fenómeno de underfitting (infraajuste). En esta figura, los puntos representan los datos de entrenamiento, mientras que la línea recta muestra el modelo ajustado a esos datos.

Como se puede observar, la línea recta no sigue adecuadamente la distribución de los puntos de datos, lo que indica que el modelo es demasiado simple para capturar la complejidad de los patrones presentes en los datos. Los puntos que deberían estar cerca de la línea de ajuste se perciben como muy alejados, lo que sugiere que el modelo no está reconociendo correctamente la relación subyacente entre las variables.

Figura 2.16: Underfitting



Entrenamiento Cruzado

La validación cruzada es una técnica fundamental en el entrenamiento de modelos de machine learning, especialmente cuando se cuenta con un conjunto de datos limitado. En lugar de dividir el conjunto de datos de manera convencional en un solo conjunto de entrenamiento y otro de prueba, la validación cruzada permite maximizar el uso de todos los datos disponibles para tanto entrenar como evaluar el modelo, los pasos que sigue este entrenamiento son los siguientes:

- Dividir el conjunto de datos: El conjunto de datos D se divide en k subconjuntos o folds:

$$D = D_1, D_2, \dots, D_k$$

- Entrenamiento y validación: Para cada iteración i , el fold D_i se utiliza como conjunto de validación y los restantes $k - 1$ folds se combinan para formar el conjunto de entrenamiento. El modelo se entrena con el conjunto de entrenamiento y se evalúa utilizando el conjunto de validación.
- Repetir el proceso: Este proceso se repite para todos los folds, de manera que cada fold sea utilizado como conjunto de validación una vez.
- Promediar los resultados: Al final, se calcula el rendimiento del modelo promediando las métricas de evaluación obtenidas en cada iteración.

Error de Validación

El error de validación es la media de los errores obtenidos durante las iteraciones de validación cruzada. Si denotamos el error del modelo en la iteración i como E_i , el error promedio de validación E_{cv} se calcula como:

$$E_{cv} = \frac{1}{k} \sum_{i=1}^k E_i$$

Donde:

- E_i es el error de la iteración i (por ejemplo, el error cuadrático medio, la precisión, etc.).
- k es el número de folds (subconjuntos) en los que se divide el conjunto de datos.

Métrica de Evaluación Común: Error Cuadrático Medio (MSE)

Uno de los errores más comunes utilizados para medir el rendimiento en modelos de regresión es el Error Cuadrático Medio (MSE). Si y_i es la etiqueta verdadera y \hat{y}_i es la predicción del modelo, el MSE se calcula como:

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Donde:

- n es el número de muestras en el conjunto de validación.
- y_i es la etiqueta real y \hat{y}_i es la predicción del modelo para la muestra i .

Precisión (Accuracy)

La precisión es una métrica común para modelos de clasificación. Se calcula como el porcentaje de predicciones correctas respecto al total de predicciones:

$$\text{Precisión} = \frac{\text{Número de predicciones correctas}}{\text{Total de predicciones}}$$

Para calcular la precisión en el contexto de validación cruzada, se obtiene la precisión en cada iteración i y se promedian los resultados:

$$\text{Precisión}_{cv} = \frac{1}{k} \sum_{i=1}^k \text{Precisión}_i$$

Importancia en conjuntos de datos pequeños

La validación cruzada es particularmente útil cuando el número de datos disponibles es limitado, como en muchas aplicaciones prácticas donde los datos son costosos de recolectar o difíciles de obtener. Cuando se tienen pocos datos, dividir el conjunto en un único entrenamiento y conjunto de prueba puede resultar en una evaluación sesgada o inadecuada, ya que los datos de prueba no representan toda la variabilidad de los datos.

La validación cruzada, al utilizar múltiples particiones para la validación, ayuda a:

- Aprovechar al máximo los datos disponibles: Cada ejemplo en el conjunto de datos se utiliza para entrenamiento y evaluación, lo que permite entrenar modelos más robustos, incluso con conjuntos de datos pequeños.
- Reducir el riesgo de sobreajuste: Al entrenar el modelo en diferentes particiones de los datos, se reduce la probabilidad de que el modelo se ajuste demasiado a un subconjunto específico y no generalice bien a nuevos datos.
- Proporcionar una evaluación más precisa: Al promediar los resultados de las distintas iteraciones de validación, se obtiene una estimación más confiable del rendimiento del modelo, lo que puede mejorar la toma de decisiones sobre el ajuste de hiperparámetros.

2.2.8. Visión Computacional

Se define como una rama de la Inteligencia Artificial que se basa en la construcción de sistemas capaces de ver y entender el mundo visual. Este proceso involucra una serie de etapas complejas, que incluyen la obtención de datos visuales a través de cámaras o sensores, el procesamiento de imágenes para mejorar su calidad y la extracción de características relevantes que permitan comprender su contenido. A partir de esta información visual, los sistemas de visión por computadora analizan patrones y estructuras presentes en las imágenes y videos, lo que les permite interpretar contextos visuales. Además, permite que el sistema aprenda de esta información de manera automática, adaptándose y mejorando a lo largo del tiempo sin intervención directa. A medida que el modelo se entrena, es capaz de reconocer objetos, clasificar escenas y realizar tareas complejas en entornos reales. Este campo de estudio, conocido como Visión por Computadora, es crucial para el desarrollo de tecnologías avanzadas, como el reconocimiento facial, la conducción autónoma, el análisis de imágenes médicas, la vigilancia inteligente, y muchas otras aplicaciones que mejoran la interacción entre los seres humanos y las máquinas Forsyth and Ponce (2003).

En la Figura 2.17, se muestra una imagen procesada por un modelo de visión computacional que es capaz de identificar múltiples objetos en un entorno. En esta imagen, los diversos objetos presentes (por ejemplo, vehículos, señales de tránsito, peatones, etc.) están claramente etiquetados y rodeados por cuadros delimitadores (bounding boxes), cada uno acompañado de su correspondiente etiqueta. Esto ilustra cómo el modelo no solo detecta la presencia de diferentes objetos, sino que también puede predecir su posición exacta dentro de la imagen.

La posición de cada objeto está indicada por las coordenadas de los cuadros delimitadores, lo que permite al sistema conocer la localización precisa de cada elemento detectado. Además, el modelo es capaz de clasificar cada objeto según su categoría (por ejemplo, "carro", "persona", "bicicleta"), mostrando la capacidad de la visión computacional para reconocer y diferenciar entre objetos con alta precisión.

Figura 2.17: Detección de objetos



Fuente: Extraído de <https://cocodataset.org/#download>.

Intersection over Union (IoU)

La Intersección sobre Unión es una métrica clave utilizada para evaluar el rendimiento de los modelos de detección de objetos. Esta métrica compara el cuadro delimitador real (o de la verdad del suelo) con el cuadro delimitador predicho por el modelo, analizando cuánta superposición existe entre ambos.

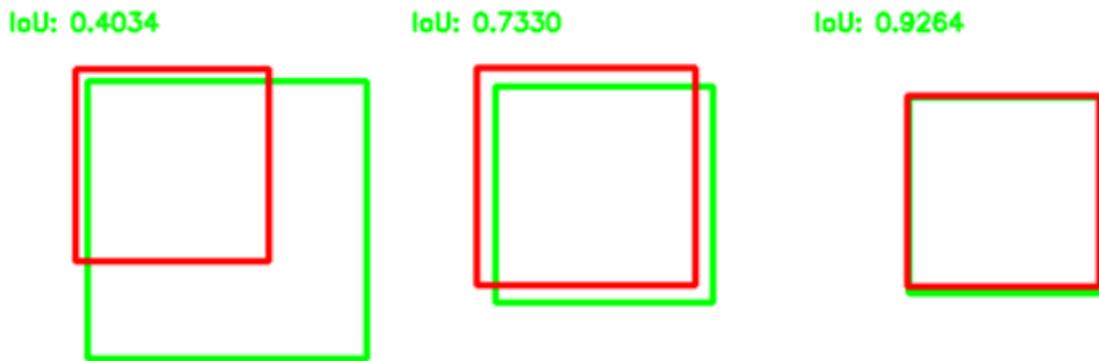
El cálculo del IoU se realiza en dos pasos:

- **Intersección:** Primero, se calcula el área de superposición entre el cuadro delimitador real y el cuadro delimitador predicho.
- **Unión:** Luego, se calcula el área total cubierta por ambos cuadros delimitadores, es decir, la unión de los dos cuadros.

El resultado del IoU se obtiene dividiendo el área de intersección entre el área de unión, generando un valor decimal en el rango de 0 a 1:

$$\text{IoU} = \frac{\text{Área}(A \cap B)}{\text{Área}(A \cup B)}$$

Figura 2.18: Intersection over Union



Como en el ejemplo en la figura 2.18 IoU cercano a 1 indica que la predicción es muy precisa, ya que los cuadros delimitadores están casi perfectamente alineados. Un IoU cercano a 0 significa que hay poca o ninguna superposición entre los cuadros, lo que indica una predicción incorrecta.

En términos prácticos, si el IoU es mayor a 0.5, generalmente se considera que la predicción es buena y el objeto ha sido correctamente detectado y localizado por el modelo.

Esta métrica es esencial para evaluar y mejorar la precisión de los modelos de detección de objetos, ya que proporciona una forma cuantitativa de medir la calidad de las predicciones en términos de la superposición espacial entre las predicciones del modelo y las etiquetas reales.

Non Maximum Suppression

El Non Maximum Suppression (NMS) es una técnica utilizada en tareas de detección de objetos para eliminar predicciones redundantes y seleccionar las más relevantes. Esta técnica se basa en la interacción de dos elementos clave:

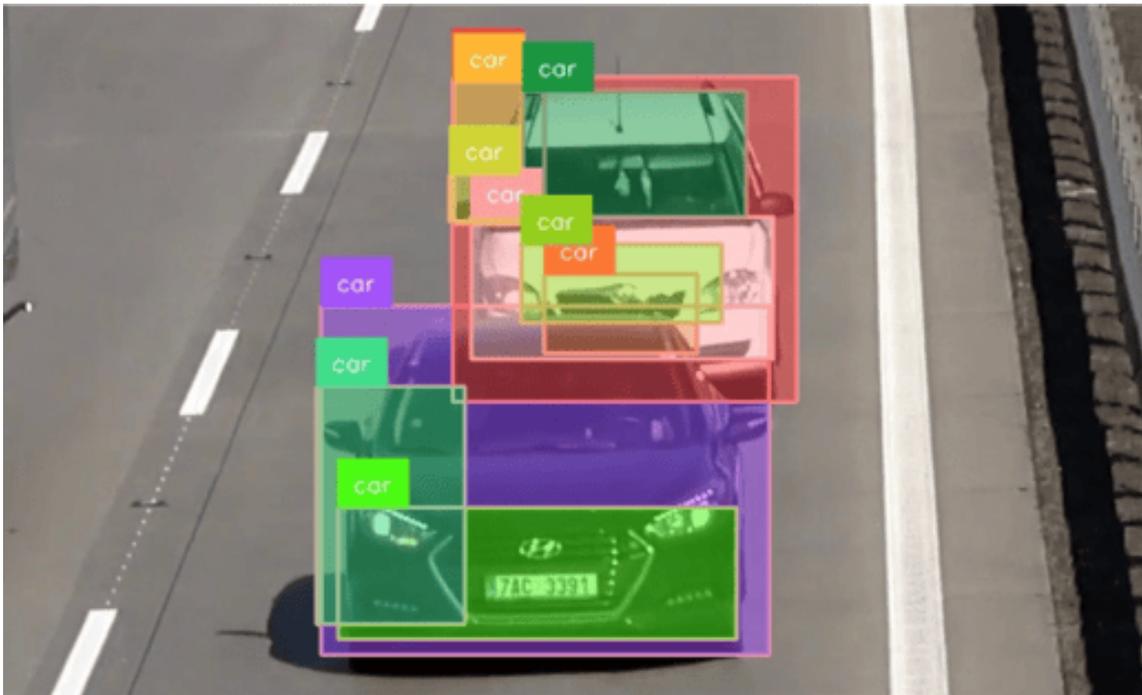
Puntaje de confianza: Cada predicción generada por el modelo tiene un puntaje que indica la probabilidad de que contenga un objeto relevante. **IoU (Intersection over Union):** Es una métrica que mide el grado de superposición entre dos cuadros delimitadores (bounding boxes). El procedimiento funciona de la siguiente manera:

Primero, se seleccionan las predicciones con las puntuaciones más altas dentro de una categoría específica. A continuación, se calcula el IoU entre el cuadro con el puntaje más alto y los demás cuadros. Si el IoU de un cuadro con respecto al cuadro seleccionado supera un umbral (generalmente 0.5), ese cuadro se elimina porque se considera redundante. Los cuadros que tienen un puntaje más bajo, pero no superan el umbral de IoU, se mantienen, ya que podrían corresponder a predicciones válidas de otros objetos. Este proceso asegura que el modelo retenga únicamente las predicciones más confiables y relevantes, reduciendo la redundancia sin descartar objetos válidos.

En la figura 2.19 se observan dos automóviles identificados mediante bounding

boxes (cuadros delimitadores). Estos cuadros representan las predicciones realizadas por un modelo de detección de objetos. Cada cuadro tiene asociado un puntaje de confianza (confidence score) que indica la probabilidad de que el objeto detectado sea un automóvil. Además, se aplica la técnica de Non Maximum Suppression (NMS) para determinar cuál bounding box es más relevante y eliminar los cuadros redundantes que representan la misma detección.

Figura 2.19: Non Maximum Suppression



Redes Neuronales Pre-entrenadas

Las redes neuronales preentrenadas son modelos de redes neuronales profundas que han sido entrenados previamente en grandes conjuntos de datos masivos. Posteriormente, estos modelos se reutilizan como puntos de partida para tareas específicas y personalizadas, particularmente en el ámbito de la visión por computadora.

Este enfoque permite mejorar el rendimiento de los modelos en nuevas tareas al compartir características aprendidas y reutilizar las capas de la red preentrenada de manera eficiente. Como resultado, se acelera el desarrollo de nuevas tecnologías y se evita la necesidad de construir y entrenar modelos desde cero, lo que ahorra tiempo y recursos computacionales.

Además, las primeras capas de una red neuronal suelen aprender características generales, como bordes o texturas, que son aplicables a múltiples tareas. Al aprovechar estas capas ya entrenadas, se puede reducir significativamente el esfuerzo requerido para ajustar el modelo a un problema específico, lo que mejora tanto la eficiencia como la efectividad del proceso de entrenamiento. Huang et al. (2017)

2.2.9. Metadatos

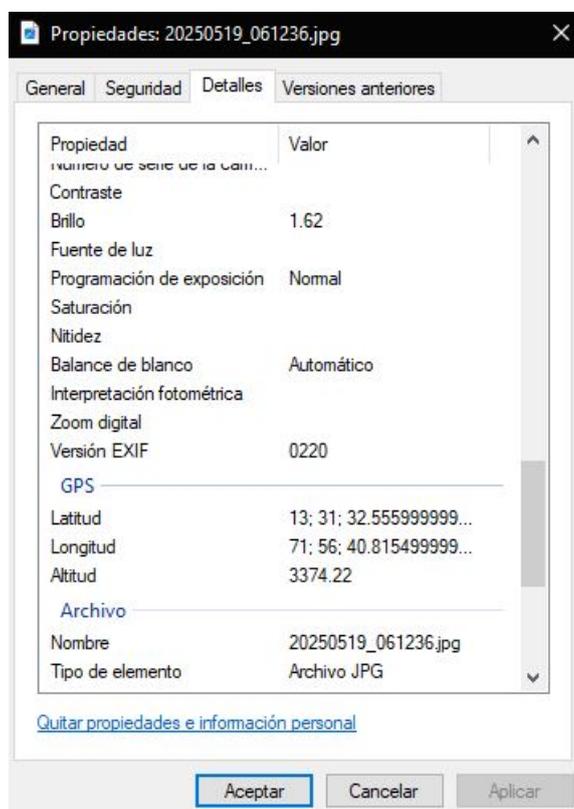
Los metadatos son información estructurada que describe, contextualiza y da soporte a la gestión, recuperación y preservación de recursos digitales. Se consideran esenciales para garantizar la organización, accesibilidad y sostenibilidad de los datos a lo largo del tiempo. Según Gilliland (2008), los metadatos “facilitan la identificación, el descubrimiento, la evaluación, la selección, la adquisición y el uso de los recursos de información”.

Esta información adicional puede incluir elementos técnicos, administrativos, descriptivos y de preservación. Entre ellos destacan el formato del archivo, la fecha de creación, la ubicación de almacenamiento, la autoría, y los derechos de uso. También puede incorporar datos del entorno técnico, como el nombre del servidor o del sistema de gestión donde el objeto digital está alojado.

Los metadatos permiten no solo una adecuada clasificación y recuperación de la información, sino también su verificación, autenticidad y trazabilidad. En el contexto de la preservación digital, resultan clave para asegurar que los archivos digitales permanezcan utilizables, interpretables y confiables a largo plazo, especialmente frente a la obsolescencia tecnológica, como podemos visualizar en la figura 2.20.

“Sin metadatos adecuados, incluso los archivos mejor almacenados podrían volverse inservibles o incomprensibles con el paso del tiempo” Gilliland (2008).

Figura 2.20: Metadatos de una imagen



Fuente: Elaboración propia.

2.2.10. Sistema de posicionamiento global (GPS)

El sistema de posicionamiento global (GPS) es un sistema de navegación por satélite que permite determinar la posición geográfica de un receptor en cualquier punto de la Tierra, en cualquier momento y bajo cualquier condición climática. Este sistema se basa en una red de satélites artificiales que orbitan la Tierra, conocidos como la constelación de satélites GPS. Estos satélites están distribuidos de manera que al menos cuatro de ellos estén siempre al alcance de un receptor, sin importar la ubicación geográfica o las condiciones del clima.

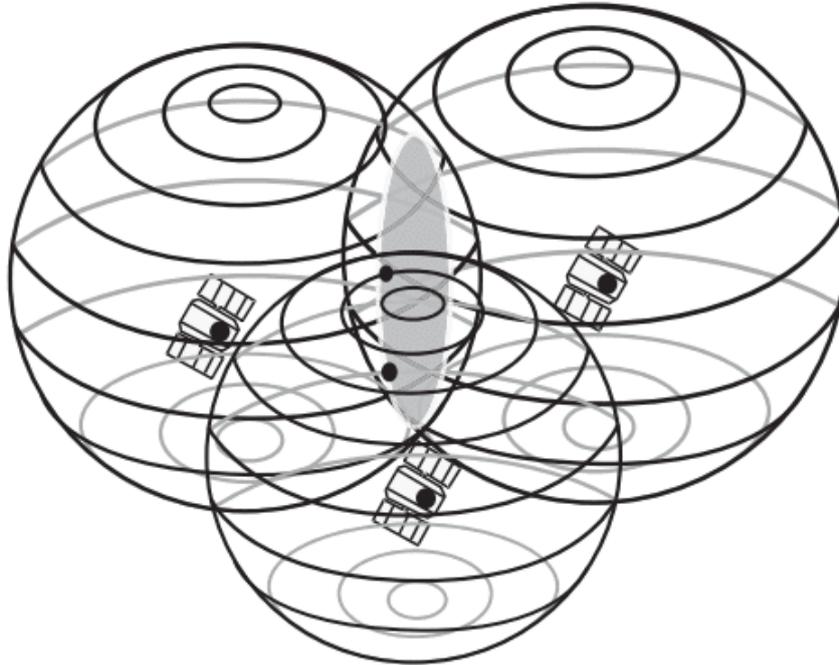
Cada satélite transmite una señal que contiene información sobre su posición actual y la hora exacta en que se envió la señal. El receptor en la Tierra utiliza esta información para calcular el tiempo que tarda la señal en llegar desde el satélite hasta él, lo que le permite determinar la distancia a cada uno de los satélites cercanos. Gracias a un proceso llamado trilateración, el receptor puede calcular su posición exacta, utilizando al menos tres satélites para obtener coordenadas de latitud y longitud y un cuarto satélite para determinar la altitud.

Este sistema se ha convertido en una herramienta clave para diversas aplicaciones, tales como la aviación, donde se utiliza para navegación precisa de aeronaves, minimizando el riesgo de errores en la ruta. En la navegación marítima, el GPS facilita la localización de embarcaciones, mejorando la seguridad en el mar. También es utilizado en el monitoreo de vehículos, gestión de flotas y rastreo de personas y objetos, permitiendo el seguimiento en tiempo real con alta precisión. Además, el GPS es esencial para muchas aplicaciones cotidianas, como en dispositivos móviles, sistemas de mapas en línea y aplicaciones de transporte. Kaplan and Hegarty (2006)

En esta Figura 2.21 se ilustra el funcionamiento básico del Sistema de Posicionamiento Global (GPS). Se observan tres satélites que orbitan alrededor de la Tierra, los cuales emiten señales hacia un receptor ubicado en la superficie terrestre. Cada uno de los satélites transmite información sobre su posición exacta en el espacio y la hora exacta en la que la señal fue enviada.

El receptor, que representa a la persona o dispositivo en la Tierra, recibe estas señales y, a través de un proceso de trilateración, utiliza la información de al menos tres satélites para calcular su posición exacta. A partir de las distancias conocidas hacia cada satélite, el receptor es capaz de determinar su latitud, longitud y altitud, ubicándose con precisión en la superficie terrestre.

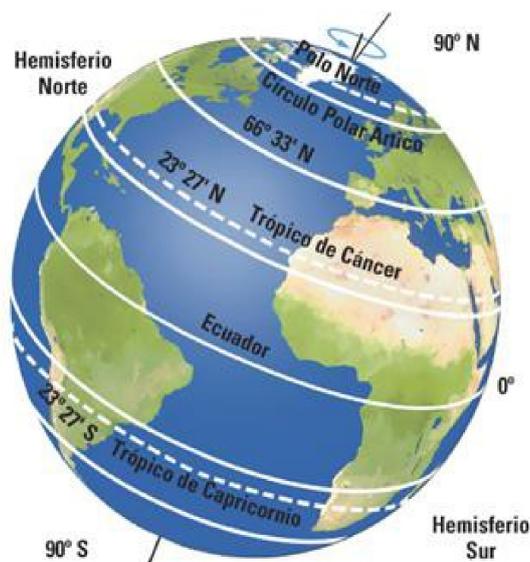
Figura 2.21: Reconocimiento de posición de un receptor



Fuente: Kaplan and Hegarty (2006)

- **Latitud:** Es el ángulo que determina el alejamiento entre la línea imaginaria del ecuador mostrada en la Figura 2.22 y cualquier otro punto de la tierra determinándose su fluctuación en valores de 0° en la línea ecuatorial y 90° como el punto mas alejado en los polos sur y norte, su valor se determina positivo si el punto esta por encima de la línea ecuatorial en el hemisferio norte y negativo si esta por debajo del ecuador en el hemisferio sur Huerta et al. (2005).

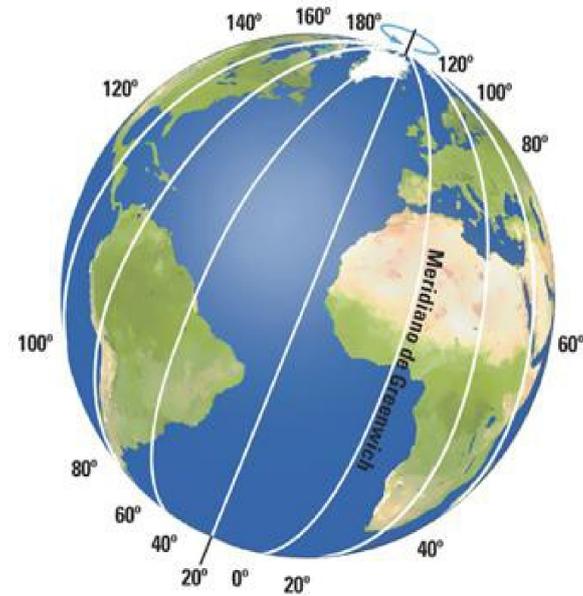
Figura 2.22: Líneas meridianas



Fuente: Elaboración propia.

- longitud: El el ángulo que determina la distancia entre la línea de greenwich y cualquier otro punto de la tierra fluctuando en valores de 0° a 180° empezando del meridiano o arco de greenwich mostrada en la Figura 2.23 y finalizando en la línea a su reverso conocida como el arco o meridiano internacional de fecha, positivo en el hemisferio oriental y negativo en el hemisferio occidental. Huerta et al. (2005)

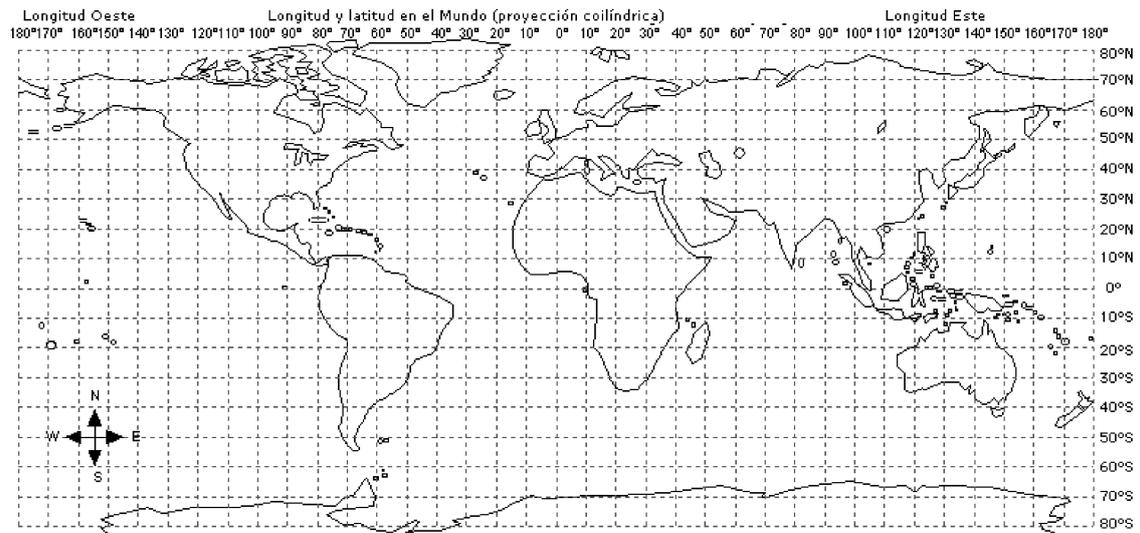
Figura 2.23: Líneas meridianas



Fuente: Kaplan and Hegarty (2006)

Teniendo en cuenta los valores de latitud y longitud, es posible identificar con precisión cualquier punto sobre la superficie terrestre. Estos dos valores conforman un sistema de coordenadas geográficas que permite ubicar lugares en el globo de manera universal, independientemente del país o la región. La latitud indica la distancia al norte o al sur del ecuador, mientras que la longitud señala la distancia al este o al oeste del meridiano de Greenwich. En la Figura 2.24 se muestra cómo se proyectan estos valores sobre el mapamundi, facilitando la localización exacta de cualquier punto geográfico mediante la intersección de estas coordenadas.

Figura 2.24: Latitud y longitud en el mapamundi



Fuente: Kaplan and Hegarty (2006)

2.2.11. Herramientas

Python

Python es un lenguaje de programación de alto nivel, interpretado y de propósito general, que se ha consolidado como una herramienta fundamental en diversas áreas como el desarrollo de aplicaciones web, la ingeniería de software, la ciencia de datos y el aprendizaje automático (machine learning). Su sintaxis clara y legible facilita la curva de aprendizaje, lo que lo hace accesible tanto para programadores novatos como para expertos. Además, Python es multiplataforma, permitiendo su ejecución en diferentes sistemas operativos sin modificaciones significativas en el código fuente.

Otra de las ventajas de Python es su extenso ecosistema de bibliotecas y frameworks especializados, que facilitan el desarrollo rápido y eficiente de soluciones complejas, especialmente en ámbitos relacionados con inteligencia artificial y análisis de datos. El lenguaje es de código abierto, lo que contribuye a su amplia adopción y constante actualización por parte de la comunidad global de desarrolladores Lutz (2013).

Colaboratory (Google Colab)

Colaboratory, conocido también como *Colab*, es una plataforma gratuita creada por Google Research que permite a los usuarios escribir, ejecutar y compartir código en Python directamente desde el navegador, sin necesidad de realizar configuraciones adicionales. Este entorno, basado en notebooks Jupyter, es ideal para actividades como aprendizaje automático, análisis de datos y enseñanza, ya que ofrece acceso a recursos computacionales avanzados como GPUs y TPUs, sin costo o a bajo precio. Su facilidad de uso y disponibilidad han contribuido a impulsar la colaboración y acelerar el desarrollo de proyectos en áreas de inteligencia artificial y ciencia de datos Bisong (2019).

TensorFlow

TensorFlow es una biblioteca de código abierto para el desarrollo y la implementación de modelos de aprendizaje automático (Machine Learning) y aprendizaje profundo (Deep Learning). Fue creada por el equipo de Google Brain y lanzada al público en 2015 con el objetivo de facilitar el diseño, entrenamiento y despliegue de redes neuronales artificiales de forma eficiente y escalable.

Esta biblioteca ofrece una arquitectura flexible que permite a los desarrolladores construir modelos computacionales complejos a través de gráficos de flujo de datos, donde las operaciones se representan como nodos y los datos como tensores que circulan entre ellos. Gracias a esta estructura, TensorFlow es capaz de aprovechar diferentes tipos de hardware para acelerar el cálculo, incluyendo procesadores centrales (CPUs), unidades de procesamiento gráfico (GPUs) y unidades de procesamiento tensorial (TPUs), estas últimas especialmente optimizadas para cargas de trabajo de inteligencia artificial.

TensorFlow es ampliamente utilizado en aplicaciones que requieren reconocimiento de patrones, análisis de imágenes, procesamiento de lenguaje natural, predicciones en tiempo real y otras tareas donde los algoritmos tradicionales presentan limitaciones. Su ecosistema incluye herramientas y APIs que cubren desde modelos de alto nivel para usuarios sin experiencia profunda en programación, hasta interfaces avanzadas para expertos en investigación.

Además, TensorFlow facilita la implementación de modelos en diferentes plataformas, desde dispositivos móviles hasta servidores en la nube, lo que contribuye a su popularidad en la comunidad científica y empresarial. Su naturaleza open source permite una continua actualización y ampliación por parte de la comunidad global, lo que asegura la incorporación constante de innovaciones y mejoras técnicas Géron (2019b).

En resumen, TensorFlow no solo es una herramienta poderosa para la construcción y entrenamiento de redes neuronales, sino también una plataforma integral que soporta todo el ciclo de vida del desarrollo de aplicaciones basadas en inteligencia artificial.

LinkedAI

LinkedAI es una plataforma tecnológica diseñada para facilitar la generación y gestión de los datos necesarios para el entrenamiento de modelos de inteligencia artificial (IA) orientados al análisis de imágenes. Estos modelos de IA dependen fundamentalmente de grandes volúmenes de datos de entrenamiento que les permitan identificar patrones complejos, como la detección de personas o el diagnóstico de enfermedades a partir de imágenes.

Sin embargo, uno de los desafíos más significativos en el desarrollo de modelos precisos y confiables es el proceso de etiquetado de datos. Este procedimiento, que consiste en asignar etiquetas o anotaciones precisas a los datos brutos para que el modelo pueda aprender de manera supervisada, suele ser laborioso, costoso y requiere una considerable inversión de tiempo y recursos. Como consecuencia, el etiquetado de datos representa uno de los principales cuellos de botella en la creación y despliegue de modelos de IA efectivos. Esta limitación frecuentemente constituye un obstáculo importante para las organizaciones que buscan implementar soluciones basadas en inteligencia artificial, y es una de las causas principales por las que muchas empresas fracasan en la adopción exitosa de estas tecnologías Smith and Doe (2021).

Por lo tanto, plataformas como LinkedAI resultan esenciales para optimizar y acelerar este proceso, reduciendo costos y mejorando la calidad de los datos etiquetados, lo que a su vez incrementa la precisión y efectividad de los modelos de IA entrenados con dichos datos.

React

ReactJS es una biblioteca de JavaScript ampliamente utilizada para el desarrollo de interfaces de usuario tanto en aplicaciones web como móviles. Desarrollada inicialmente

por Facebook, ReactJS se basa en el concepto de componentes reutilizables, que son fragmentos de código JavaScript que encapsulan la lógica, el diseño y el comportamiento de elementos específicos de la interfaz, facilitando así la creación y mantenimiento de aplicaciones escalables y dinámicas.

A diferencia de otros frameworks completos como Angular o Vue.js, ReactJS no proporciona una solución integral para el desarrollo de aplicaciones, sino que se centra exclusivamente en la capa de vista o presentación. Esto significa que ReactJS se encarga principalmente del renderizado eficiente y reactivo de los componentes de la interfaz, dejando a los desarrolladores la libertad de integrar otras bibliotecas o herramientas para gestionar estados, rutas o comunicación con servidores. Esta característica convierte a React en una alternativa flexible y modular para construir aplicaciones modernas que requieren una alta interacción y rendimiento .

El enfoque basado en componentes y el uso del DOM virtual (Virtual DOM) permiten que ReactJS actualice únicamente las partes necesarias de la interfaz, optimizando la experiencia del usuario y mejorando el desempeño general de las aplicaciones. React (2023).

ExpressJS

Express es un framework minimalista y flexible para aplicaciones web desarrollado sobre Node.js, que proporciona un conjunto robusto de características para construir aplicaciones tanto web como móviles. Su diseño ligero permite a los desarrolladores crear servidores y gestionar rutas de manera sencilla y eficiente, facilitando la integración con diversas bases de datos, middleware y servicios externos.

Gracias a su simplicidad y extensibilidad, Express se ha convertido en uno de los frameworks más populares dentro del ecosistema de Node.js, permitiendo el desarrollo rápido de aplicaciones escalables y con un manejo efectivo de solicitudes y respuestas HTTP. Además, su compatibilidad con middleware facilita la implementación de funcionalidades adicionales, como autenticación, validación y manejo de errores, contribuyendo a una arquitectura modular y mantenible.

Este framework es especialmente valorado por su capacidad para servir APIs RESTful y gestionar aplicaciones de una sola página (SPA), adaptándose a diferentes tipos de proyectos y necesidades JSExpress (2023).

PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos relacional orientado a objetos, de código abierto y con una trayectoria de más de 35 años de desarrollo continuo. Esta larga evolución le ha conferido una sólida reputación basada en su alta fiabilidad, robustez funcional y excelente rendimiento.

PostgreSQL es conocido por su conformidad con los estándares SQL, su capacidad

para manejar cargas de trabajo complejas y su extensibilidad, permitiendo la integración de funciones personalizadas, tipos de datos avanzados y procedimientos almacenados. Además, ofrece soporte para transacciones ACID, replicación, recuperación ante desastres y concurrencia mediante control multiversión (MVCC), lo que lo hace adecuado para aplicaciones críticas y de gran escala.

Gracias a su licencia abierta, PostgreSQL es ampliamente adoptado tanto en entornos académicos como empresariales, proporcionando una alternativa poderosa y flexible frente a sistemas propietarios PostgreSQL (2023).

Flask

Flask es un framework minimalista para el desarrollo de aplicaciones web en Python, diseñado para ofrecer una base ligera que facilite la construcción rápida y sencilla de servidores web. Proporciona funcionalidades básicas como el manejo de rutas, la gestión de solicitudes y respuestas, y el soporte para plantillas, sin imponer una estructura rígida ni incorporar componentes innecesarios.

Este enfoque flexible permite a los desarrolladores añadir únicamente las extensiones y herramientas que necesiten, lo que hace que Flask sea ideal tanto para proyectos pequeños como para aplicaciones escalables. Su diseño modular permite integrar fácilmente bases de datos, sistemas de autenticación, y otras funcionalidades a medida que el proyecto crece.

Debido a su simplicidad y adaptabilidad, Flask es una opción preferida para crear APIs RESTful, prototipos y aplicaciones web que requieren rapidez en el desarrollo y facilidad de mantenimiento Grinberg (2018).

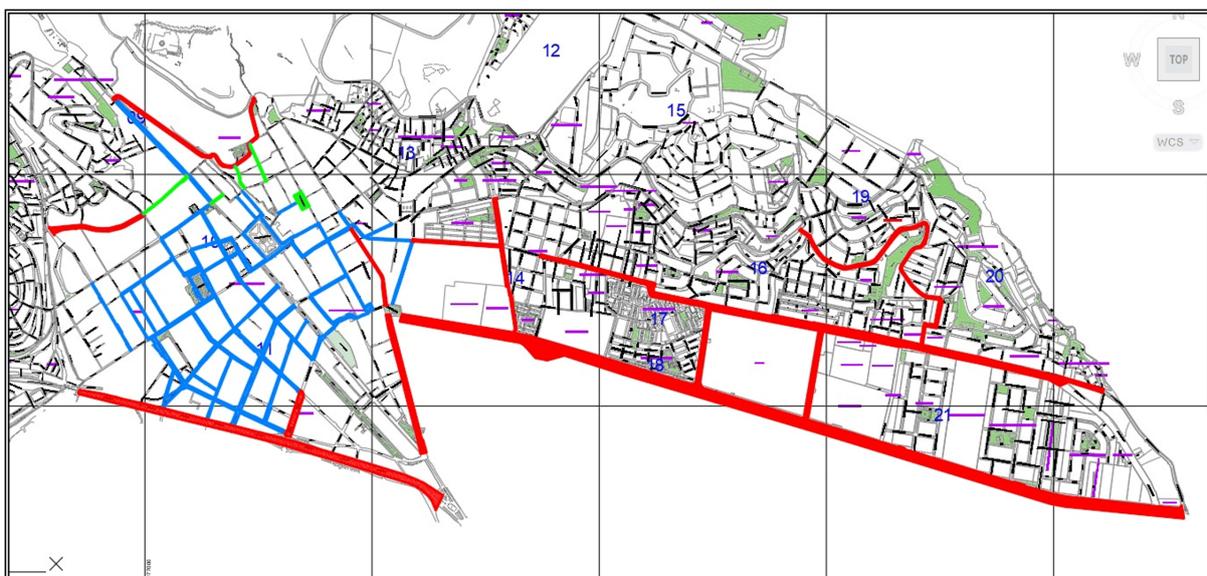
Capítulo 3

Desarrollo e Implementación del Proyecto

3.1. Recolección de datos

Para la extracción de imágenes, se tomó como referencia la memoria descriptiva elaborada por la Gerencia de Infraestructura del Municipio Provincial del Cusco. De este documento se obtuvo la Figura 3.1, que detalla las vías priorizadas para mantenimiento rutinario. Estas zonas presentan los mayores niveles de deterioro en la ciudad, lo que justifica su selección para el análisis y documentación del estado vial.

Figura 3.1: Mapa calles del Cusco - Mantenimiento



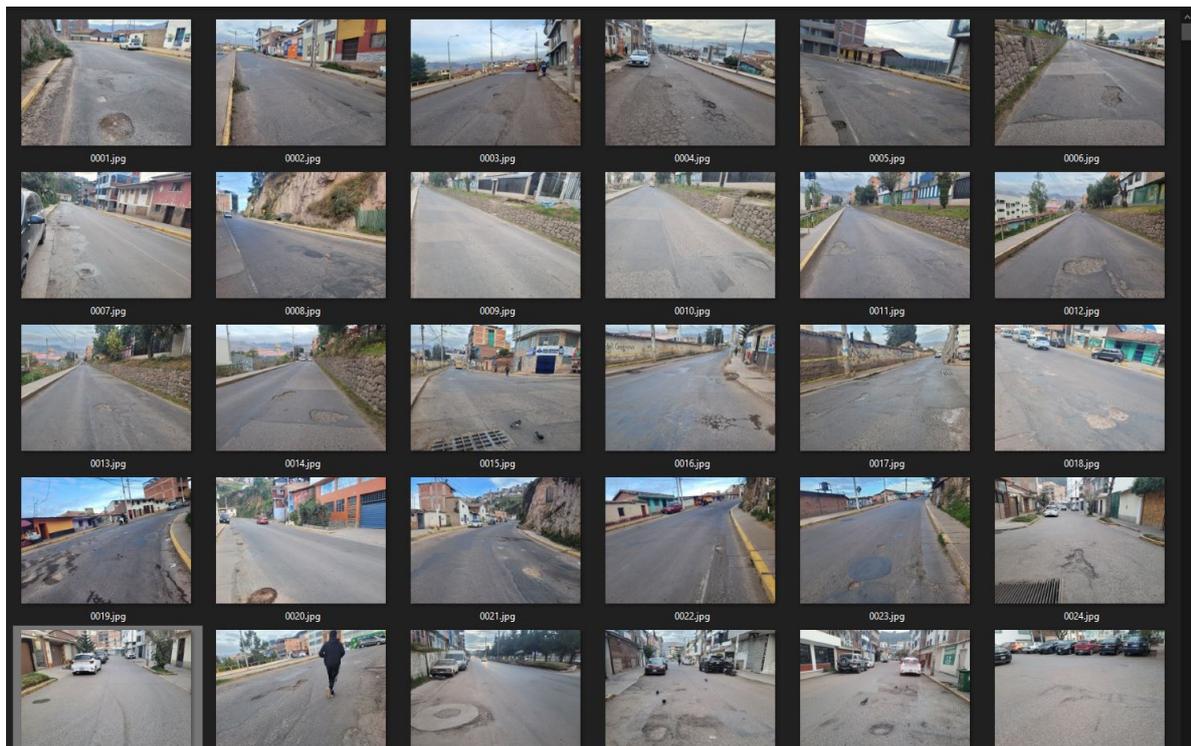
Fuente: Elaboración propia.

En el mapa adjunto se destacan las zonas programadas para mantenimiento, en las cuales se resaltan como áreas de pavimento flexible las siguientes avenidas:

- Avenida de la Cultura.
- Avenida Garcilaso.
- Avenida 28 de Julio.
- Avenida Qollasuyo.
- Avenida Del Ejercito.

La captura de fotos incluye un total de 600 imágenes en formato JPG mostrados en la Figura 3.2, cada una acompañada de metadatos y geocalización, lo que proporciona un registro detallado y preciso de su estado actual.

Figura 3.2: Ejemplares de datos capturados



Fuente: Elaboración propia.

Conjunto de datos

El registro de fotos en los distintos puntos especificados en la ciudad del Cusco se enumeró de la siguiente manera:

Tabla 3.1: Registro de datos

Avenida	Numero de Imágenes
Avenida la Cultura	145
Avenida Garcilaso	50
Avenida 28 de Julio.	90
Avenida Qollasuyo	70
Avenida Del Ejercito	125
Otros	120
Total	600

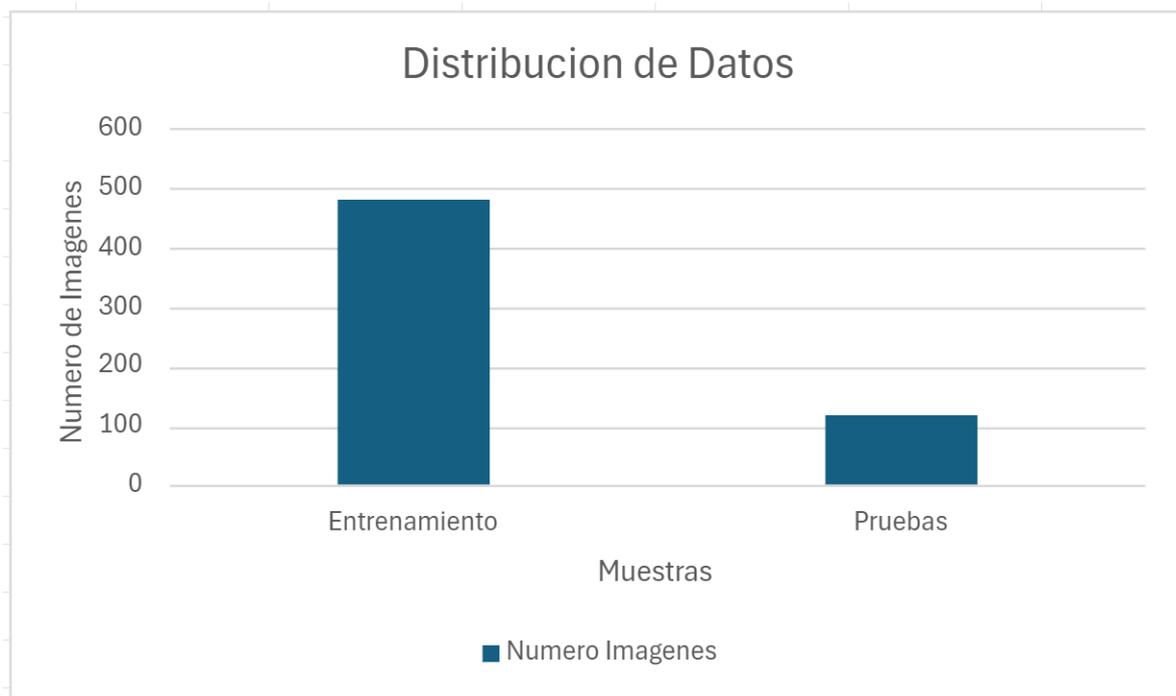
3.1.1. Construcción del dataset

Distribución de datos

La distribución en base al antecedente Ortega Triana (2021) donde explica que aleatoriamente el conjunto de datos se separó en 2 partes visualizada en la Figura 3.3, una para realizar el entrenamiento de la red neuronal y otra para realizar el testing o pruebas.

- Se utilizó un 80 % de las imágenes para el entrenamiento de la red neuronal.
- El 20 % de las imágenes para testing.

Figura 3.3: Distribución de datos



Fuente: Elaboración propia.

```

#Importar Librerias
import os
import zipfile
import random
import shutil

#Lectura de Datos(Imagenes)
local_zip = "/content/ImagenesGrietas.zip"
zip_ref = zipfile.ZipFile(local_zip, "r")
zip_ref.extractall("dataset")
zip_ref.close()

#Ruta de imagenes descomprimidas
path = "dataset/ImagenesGrietas"
content = os.listdir(path)

#Rutas para Test y Train
path_general = "DatasetFinal"
path_train = "{}/train".format(path_general)
path_test = "{}/test".format(path_general)
os.mkdir(path_general)
os.mkdir(path_train)
os.mkdir(path_test)

#Porcentaje para Train
train = 0.8

```

```

#Construccion de Train
for nCount in range(int(len(content)*train)):
    random_choice_img = random.choice(content)
    random_choice_img_abs = "{}/{}".format( path,
        random_choice_img)
    target_img = "{}/{}".format( path_train, random_choice_img)
    shutil.copyfile(random_choice_img_abs, target_img)
    content.remove(random_choice_img)

#Construccion de Test (Imagenes restante)
for img in content:
    random_choice_img_abs = "{}/{}".format(path, img)
    target_img = "{}/{}".format(path_test, img)
    shutil.copyfile( random_choice_img_abs, target_img)

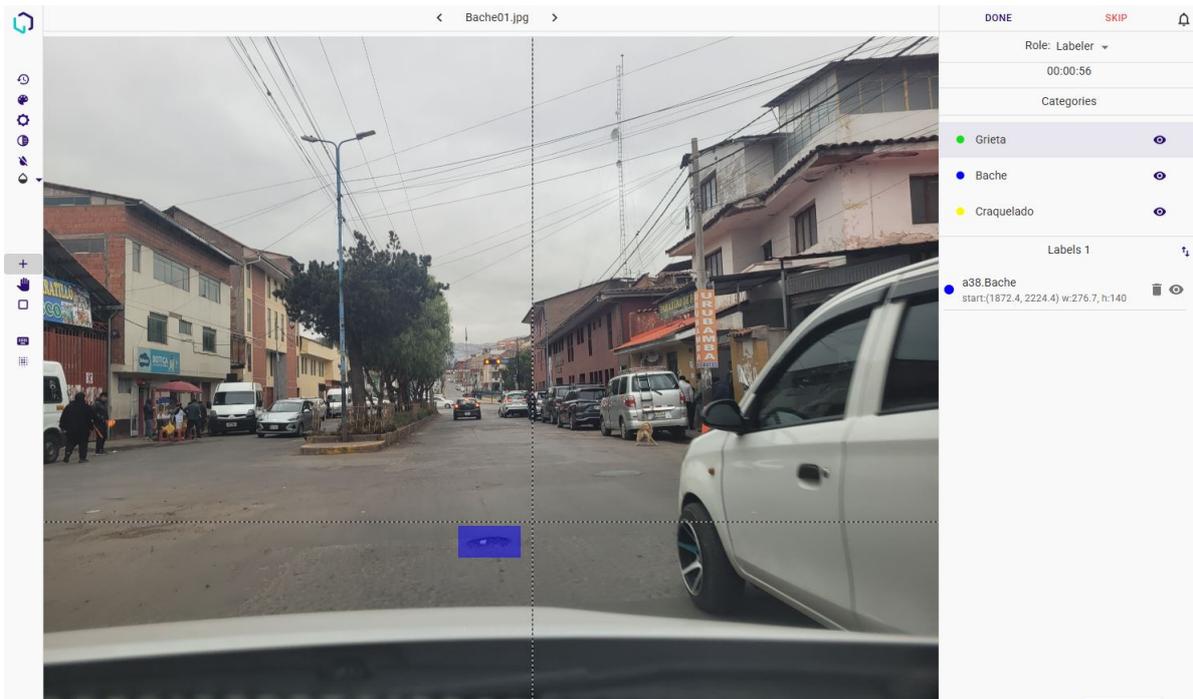
#Transformacion en formato Zip
!zip -r /content/DatasetFinal.zip /content/DatasetFinal

```

Etiquetado de datos

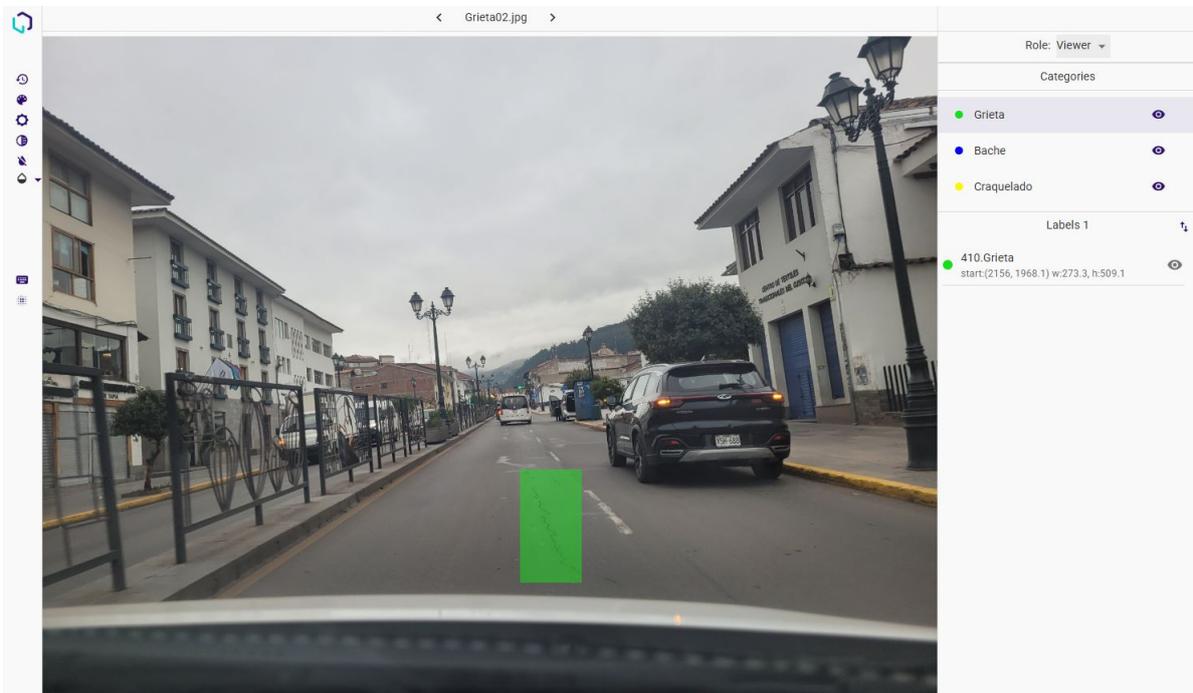
Haciendo uso de la herramienta LinkedAI, se subieron los dos conjuntos de datos para realizar el etiquetado manual de los bounding boxes en cada una de las imágenes, encerrando cada deterioro en pequeños recuadros, como se muestra en los ejemplares de las Figuras 3.4 y 3.5. De esta manera, se obtuvo un archivo JSON con cada bounding box y los datos específicos de cada imagen. Cabe recalcar que la herramienta solo exporta los datos generados, pero no las imágenes, las cuales deben guardarse por separado.

Figura 3.4: Ejemplar 1 de bounding boxes



Fuente: Elaboración propia.

Figura 3.5: Ejemplar 2 de bounding boxes

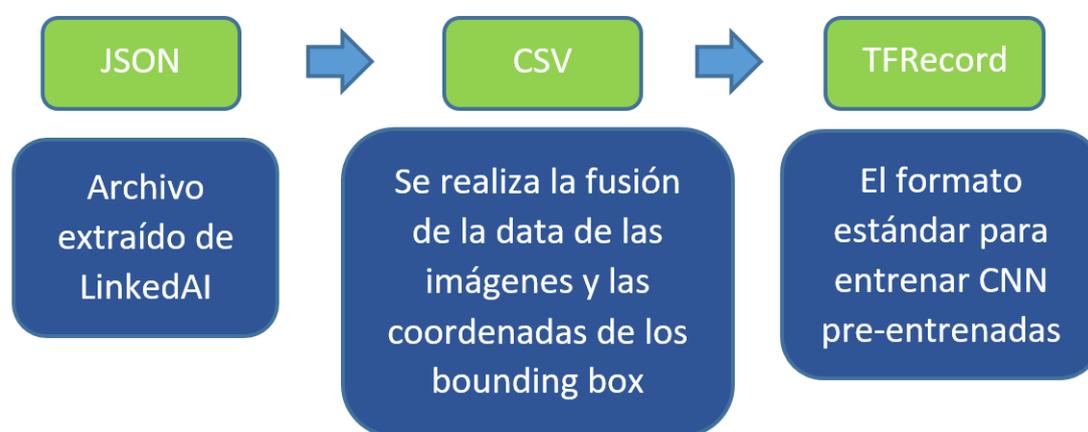


Fuente: Elaboración propia.

Procesamiento de datos

Antes de utilizar el dataset en la red neuronal preentrenada, es necesario procesar los datos y transformarlos a un formato especial de TensorFlow para el entrenamiento de modelos, llamado TFRecord Figura 3.6. El archivo JSON extraído de LinkedAI debe ser procesado para extraer las coordenadas de los bounding boxes que encierran los deterioros en cada imagen. Posteriormente, los nombres de las imágenes deben combinarse con las imágenes en formato binario dentro de un archivo CSV. Finalmente, este CSV se transforma en un TFRecord, que es el formato optimizado para el entrenamiento en TensorFlow.

Figura 3.6: Procesamiento de datos



Fuente: Elaboración propia.

Red neuronal pre-entrenada SSD MobileNetV2

Es necesario descargar la red neuronal SSD MobileNetV2 antes de realizar las configuraciones del pipeline. Este proceso es fundamental porque el checkpoint almacenado entre sus archivos será utilizado para reentrenar la CNN. Asegurándose de que la descarga y configuraciones se realicen correctamente para garantizar que el entrenamiento de la red neuronal se desarrolle sin problemas.

3.2. Construcción y entrenamiento de la CNN

3.2.1. Configuración de LabelMap

El archivo que define las etiquetas en un dataset es crucial para la consistencia del proyecto. Es fundamental que las etiquetas coincidan con las definidas en LinkedAI. En el caso del proyecto, debe utilizarse la etiqueta "deterioro" para identificar deterioros

únicos y "fisuras" para grupos incontables de deterioros en una sola área, como el conocido patrón de "piel de cocodrilo".^{en} pavimentos.

```

labels = [
    {'name': 'Bache', 'id': 1},
    {'name': 'Grieta', 'id': 2},
    {'name': 'Craquelado', 'id': 3}
]
#Creamos un archivo y lo abrimos en modo de escritura
with open("label_map.pbtxt", "w") as f:
    #Empezamos a crear nuestro archivo LabelMap
    for label in labels:
        f.write('item_{ }\n')
        f.write('\tname:\'{}\'\n'.format(label['name']))
        f.write('\tid:{}\n'.format(label['id']))
        f.write('}\n')

```

3.2.2. Configuración del pipeline

El pipeline es el archivo de configuración que da alcance TensorFlow, para definir la carga de datos, el pre-procesamiento y la división de datos, en nuestro caso especificamos lo siguiente:

- Cuantos tipos de etiquetas tenemos en nuestro dataset y en nuestro labelmap en nuestro caso 2 que son deterioro y fisura.
- Especificamos cuantas imágenes queremos operar en cada iteración, en este caso tendremos 4.
- Especificamos el checkpoint o el conjunto de pesos almacenados en cierto punto del entrenamiento de la red neuronal pre-entrenada.
- Se utilizará la configuración "detection" debido a que nuestro modelo necesita detectar objetos dentro de imágenes.
- Especifica la dirección del LabelMap para el entrenamiento, en este caso dentro de una variable.
- Especifica la ruta del archivo TFRecord de entrenamiento, en este caso la ruta en una variable.
- Especifica la dirección del LabelMap para la evaluación o test, en este caso dentro de una variable.
- Especifica la ruta del archivo TFRecord de evaluación o test, en este caso la ruta en una variable.

```

pipeline_config.model.ssd.num_classes = 3
pipeline_config.train_config.batch_size = 4
pipeline_config.train_config.fine_tune_checkpoint = "{}/
    checkpoint/ckpt-0".format(output_path)
pipeline_config.train_config.fine_tune_checkpoint_type = "
    detection"
pipeline_config.train_input_reader.label_map_path =
    label_map_pbtxt_fname
pipeline_config.train_input_reader.tf_record_input_reader.
    input_path[0] = train_record_fname
pipeline_config.eval_input_reader[0].label_map_path =
    label_map_pbtxt_fname
pipeline_config.eval_input_reader[0].tf_record_input_reader.
    input_path[0] = test_record_fname

```

finalmente almacenamos la configuración de nuestro pipeline con el siguiente fragmento de código:

```

config_text = text_format.MessageToString(pipeline_config)
with tf.io.gfile.GFile(target_config, "wb") as f:
    f.write(config_text)

```

3.2.3. Entrenamiento de la CNN

Se utilizo TensorFlow 2 para realizar el entrenamiento el modelo de detección de objetos, ya habiendo configurado previamente el labelmap, pipeline par ahora solo simplificar el trabajo facilitando la siguiente información:

- Especificamos la ruta del pipeline configurado.
- Especificamos la ruta donde guardaran los nuevos checkpoints generados por el entrenamiento.
- Especificaremos finalmente el numero de iteraciones que se realizaran en el entrenamiento, en este caso 5000.

```

!python /content/models/research/object_detection/
    model_main_tf2.py \
--pipeline_config_path={target_config} \
--model_dir={model_dir} \
--num_train_steps={num_steps}

```

3.2.4. Exportación de la CNN

Finalmente, se procede a exportar la CNN entrenada para construir una API y llevarla a producción alojándola en un servidor. En este caso, se utilizará localmente para la elaboración y prueba del prototipo, lo que permitirá validar su funcionamiento antes de un posible despliegue.

```
!python /content/models/research/object_detection/
  exporter_main_v2.py \
--input_type image_tensor \
--pipeline_config_path {target_config} \
--trained_checkpoint_dir {model_dir} \
--output_directory {output_directory}
```

3.2.5. Verificación de la CNN

En la última iteración del entrenamiento, la tasa de aprendizaje se estabiliza alrededor de 0.078 como se visualiza en la Figura 3.7, tras un aumento inicial rápido. Esto indica que el proceso de aprendizaje es estable y controlado.

Figura 3.7: Ultimo Step

```
INFO:tensorflow:Step 5000 per-step time 0.131s
I0224 21:33:16.901552 140365132986176 model_lib_v2.py:705] Step 5000 per-step time 0.131s
INFO:tensorflow: {'Loss/classification_loss': 2.9353962e-05,
'Loss/localization_loss': 0.0,
'Loss/regularization_loss': 0.114065886,
'Loss/total_loss': 0.11409524,
'learning_rate': 0.078691795}
I0224 21:33:16.902379 140365132986176 model_lib_v2.py:708] {'Loss/classification_loss': 2.9353962e-05,
'Loss/localization_loss': 0.0,
'Loss/regularization_loss': 0.114065886,
'Loss/total_loss': 0.11409524,
'learning_rate': 0.078691795}
```

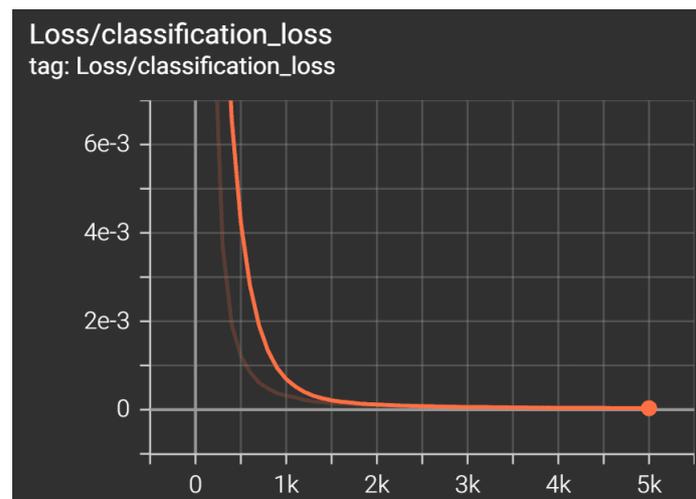
Fuente: Elaboración propia.

La pérdida de clasificación disminuye progresivamente hasta valores muy bajos (alrededor de 0.00003), lo que evidencia que el modelo ha aprendido a distinguir con alta precisión las clases de los objetos detectados.

La pérdida total mostrada en la Figura 3.8 se estabiliza en aproximadamente 0.11, con la mayor parte atribuida a la regularización, mientras que la pérdida de localización es prácticamente nula. Esto refleja un buen balance entre precisión y generalización, evitando sobreajuste.

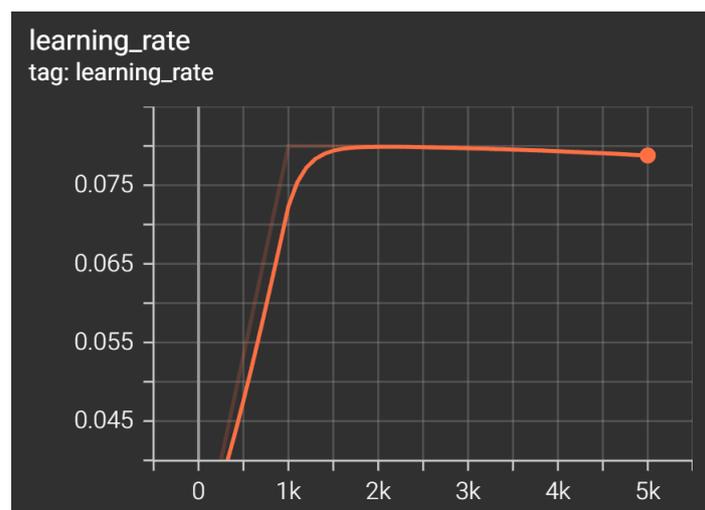
Con base en el comportamiento de estas métricas y el desempeño observado durante el entrenamiento mostrado en la Figura 3.9, se estima que el modelo alcanza una eficiencia aproximada del 82% en la tarea de clasificación, lo que indica un desempeño sólido aunque con margen para futuras mejoras.

Figura 3.8: Parámetro de pérdida



Fuente: Elaboración propia.

Figura 3.9: Parámetro Tasa de aprendizaje



Fuente: Elaboración propia.

3.2.6. Implementación del prototipo en Flask

Para la construcción del API REST que integra la Red Neuronal Convolutiva, se empleó el framework Flask. Este API cuenta con un único servicio de tipo POST, el cual recibe como entrada una imagen codificada en formato JSON (base64). A partir de esta imagen, la red neuronal realiza el análisis correspondiente, determinando la cantidad de detecciones, las clases a las que pertenece cada una de ellas, el nivel de probabilidad asociado a cada objeto identificado y, finalmente, los cuatro parámetros que definen la posición del bounding box que enmarca cada detección.

```
from flask import Flask, request, jsonify
from prediction import smartcities
import base64
```

```
app = Flask(__name__)
sc = smartcities()

@app.route('/', methods=['POST'])
def parse_request():
    request_data = request.get_json()
    imagenBase64 = request_data['imagen']
    response_64, detections = sc.predict(imagenBase64)
    result = {
        'output': response_64,
        'num_detections':detections['num_detections'],
        'detection_classes':detections['detection_classes'].
            tolist(),
        'detection_scores':detections['detection_scores'].
            tolist(),
        'detection_boxes':detections['detection_boxes'].
            tolist()
    }
    return(result)

if __name__ == "__main__":
    app.run(debug = True)
```

3.3. Modelado del Sistema

3.3.1. Historias de Usuario

En las entrevistas realizadas con el Ministerio de Transportes y Comunicaciones (MTC), Provias Nacional, Provias Descentralizado, el Instituto Vial Provincial (IVP) y la Gerencia de Infraestructura del Municipio Provincial y Regional del Cusco, se recabaron las siguientes historias de usuario que son fundamentales para el funcionamiento adecuado del prototipo:

Título: Como poblador, es imperativo que se carguen evidencias, tales como fotografías, con el fin de validar y certificar la existencia de los deterioros.

- **Descripción:**

- **Como:** Poblador.
- **Quiero:** Cargar imágenes como evidencia de lo informado.
- **Para:** Validar y certificar la existencia de los deterioros de manera efectiva y precisa.

- **Criterios de aceptación:**

1. El sistema debe proporcionar una opción para que los pobladores carguen fotografías al reportar un deterioro.
2. Las imágenes cargadas deben ser almacenadas al reporte de deterioro correspondiente.
3. El sistema debe mostrar una vista previa de las imágenes cargadas antes de confirmar el reporte.
4. El sistema debe guardar la imagen.

Título: Como poblador, es fundamental disponer de una herramienta que me permita subir en masa las fotografías, dado que, en ciertas zonas de nuestra región, no se cuenta con acceso a Internet. En tales casos, las imágenes serán recolectadas previamente para su posterior carga en el sistema.

- **Descripción:**

- **Como:** Poblador.
- **Quiero:** Tener una herramienta que permita subir en masa las fotografías.
- **Para:** Facilitar la carga de evidencias en áreas con acceso limitado a Internet, donde las imágenes serán recolectadas previamente para su posterior carga en el sistema.

- **Criterios de aceptación:**

1. La herramienta debe permitir la selección y carga de múltiples fotografías en un solo proceso.
2. La herramienta debe proporcionar un feedback sobre el estado de la carga de las imágenes, como progreso y confirmación de éxito o errores.
3. Las imágenes cargadas deben ser vinculadas correctamente a los reportes correspondientes y almacenadas de forma segura en el sistema.

Título: Como poblador, es fundamental contar con una herramienta de inteligencia artificial capaz de identificar deterioros en las fotografías presentadas como evidencia, a fin de garantizar un análisis imparcial, fidedigno e inmediato.

■ **Descripción:**

- **Como:** Poblador.
- **Quiero:** Una herramienta con inteligencia artificial que pueda identificar deterioros en las fotografías que presento como evidencias.
- **Para:** Obtener un análisis imparcial, fidedigno e instantáneo de los deterioros reportados.

■ **Criterios de aceptación:**

1. La herramienta debe analizar automáticamente las fotografías cargadas por los pobladores y detectar deterioros.
2. El análisis debe proporcionar un informe o resaltado de las áreas identificadas como deterioros en la imagen.
3. La herramienta debe funcionar de manera rápida para proporcionar resultados casi en tiempo real.
4. Los resultados del análisis deben ser almacenados junto con las evidencias presentadas y vinculados al reporte correspondiente.

Título: Como administrativo responsable del mantenimiento de las vías de tránsito, resulta fundamental disponer de una herramienta que facilite la visualización de las evidencias y datos proporcionados por los pobladores de una manera práctica y didáctica. Esta herramienta debe permitir un análisis eficiente y comprensible, optimizando así el proceso de evaluación y gestión de las condiciones de las vías.

■ **Descripción:**

- **Como:** Administrador.
- **Quiero:** Un mapa que especifique en que lugares existe deterioros.
- **Para:** Realizar un análisis eficiente de los datos y evidencias que presentan los pobladores.

■ **Criterios de aceptación:**

1. Mostrar un mapa de la región del Cusco.

2. Especificar las zonas en las cuales se encuentran los deterioros que presentaron los pobladores.
3. Actualizar constantemente el mapa con los nuevos datos proporcionados.
4. permitir editar el mapa y realizar configuraciones para mejorar la visibilidad de los datos.

Título: Como poblador, es esencial contar con las coordenadas exactas del lugar donde se han identificado los deterioros, idealmente obtenidas de manera automática. Esta funcionalidad permitirá una localización precisa y eficiente de las áreas afectadas, facilitando así la evaluación y el seguimiento de las condiciones en tiempo real.

■ **Descripción:**

- **Como:** Poblador.
- **Quiero:** Contar con coordenadas exactas del lugar donde se identifican los deterioros, obtenidas automáticamente a partir de los metadatos en las imágenes.
- **Para:** Localizar de manera precisa y eficiente las áreas afectadas, facilitando la evaluación y el seguimiento de las condiciones en tiempo real.

■ **Criterios de aceptación:**

1. La herramienta debe extraer automáticamente las coordenadas GPS de los metadatos de las imágenes cargadas por los pobladores.
2. Las coordenadas extraídas deben reflejarse en el mapa, mostrando la ubicación precisa del deterioro en el contexto geográfico.
3. Al cargar una imagen, el sistema debe proporcionar una confirmación de que las coordenadas se han extraído correctamente y se han asociado al reporte del deterioro.

Título: Como administrativo encargado del mantenimiento de las vías de tránsito, es imperativo que los datos acumulados en el mapa puedan ser filtrados y categorizados de acuerdo con su grado de veracidad, desde los más fiables hasta los menos confiables. Esta capacidad de clasificación permitirá una gestión más efectiva y precisa de la información, asegurando que se prioricen las áreas que requieren atención urgente.

■ **Descripción:**

- **Como:** Administrador.
- **Quiero:** Filtrar y categorizar los datos acumulados en el mapa de acuerdo con su grado de veracidad, desde los más fiables hasta los menos confiables.
- **Para:** Clasificar con mayor certeza la veracidad de la información recaudada de los pobladores..

■ **Criterios de aceptación:**

1. La herramienta debe permitir filtrar los datos en el mapa según su grado de veracidad, evaluado por los administradores, o confiabilidad por el número de imágenes en el mismo lugar.
2. La herramienta debe permitir resaltar o priorizar áreas en el mapa que requieren atención urgente basándose en la confiabilidad de los datos.
3. Debe ser posible actualizar la clasificación de los datos conforme se obtienen nuevas evidencias o se revisan los existentes, y estos cambios deben reflejarse automáticamente en el mapa.

Título: Como administrador, es fundamental contar con la capacidad de eliminar o editar el estado de los deterioros en las imágenes, ya que algunas zonas serán sometidas a mantenimiento y, por lo tanto, deberán ser actualizadas en el mapa. Esta funcionalidad permitirá mantener la información actualizada y precisa, reflejando correctamente el estado de las vías de tránsito después de las intervenciones realizadas.

■ **Descripción:**

- **Como:** Administrador.
- **Quiero:** Tener la capacidad de eliminar o editar el estado de los deterioros en las imágenes, ya que algunas zonas serán sometidas a mantenimiento.
- **Para:** Mantener la información actualizada y precisa, reflejando correctamente el estado de las vías de tránsito después de las intervenciones realizadas.

■ **Criterios de aceptación:**

1. La herramienta debe permitir editar el estado de los deterioros en el mapa, actualizando la información para reflejar los cambios realizados durante el mantenimiento.
2. Debe ser posible eliminar deterioros del mapa si han sido reparados o si la información ya no es relevante.
3. Los cambios realizados en el estado de los deterioros deben actualizarse automáticamente en el mapa y en los informes generados.
4. La herramienta debe mantener un historial de cambios para cada deterioro.
5. Debe haber una interfaz clara y accesible para editar y eliminar deterioros, con opciones para confirmar cambios y manejar posibles errores.

3.3.2. Casos de Uso

En base a las historias de usuario y los diferentes perfiles identificados durante las entrevistas, se definieron los casos de uso que se presentan en las Figuras 3.11 y 3.12. Estos casos de uso detallan las funcionalidades específicas contempladas en el prototipo del sistema, así como el proceso mediante el cual cada tipo de usuario puede interactuar de manera fluida con la plataforma.

Tipos de Usuarios

En el prototipo desarrollado se han definido cuatro tipos principales de usuarios mostrados en la Figura 3.10, cada uno con distintos niveles de acceso y responsabilidades, los cuales se describen a continuación:

Figura 3.10: Tipos de usuario

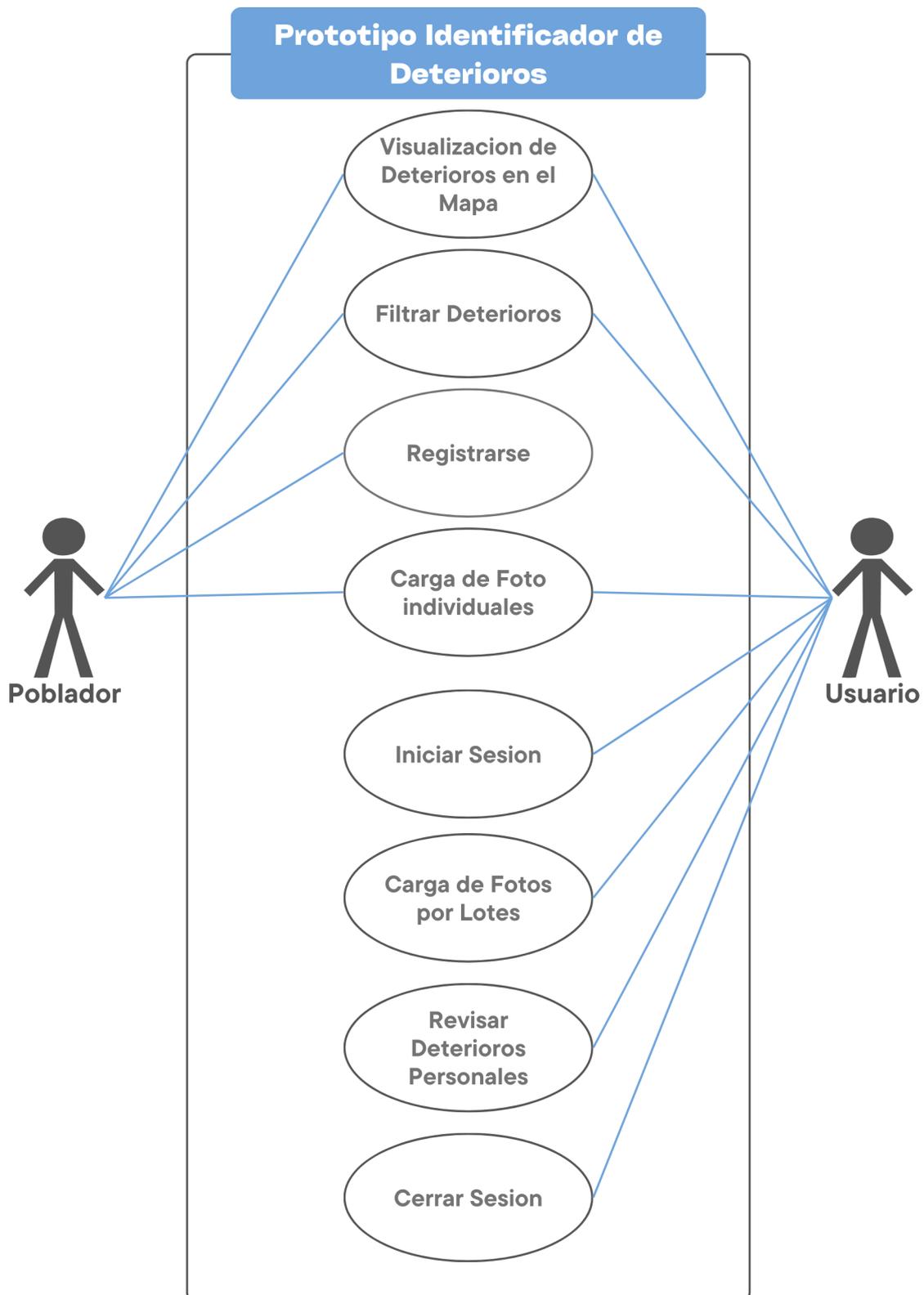


Fuente: Elaboración propia.

- **Poblador:** Cualquier ciudadano que accede libremente a la plataforma sin necesidad de registrarse. Este usuario puede interactuar con las funciones básicas del prototipo, como la carga de imágenes, permitiendo reportar posibles deterioros en las vías.
- **Usuario:** Es un poblador que ha decidido registrarse en la plataforma, obteniendo credenciales que le permiten autenticarse en el sistema. Este registro le otorga un mayor nivel de participación, seguimiento y trazabilidad dentro del prototipo.
- **Revisor:** Trabajador designado por alguna entidad interesada en el mantenimiento del sistema, cuya función principal es revisar y validar las imágenes enviadas por pobladores y usuarios. Su intervención aporta un mayor nivel de certeza a los reportes, actuando como un filtro técnico previo a su evaluación formal.
- **Administrador:** Responsable de la gestión general del sistema. Entre sus funciones se encuentra el mantenimiento de la plataforma y la administración del estado

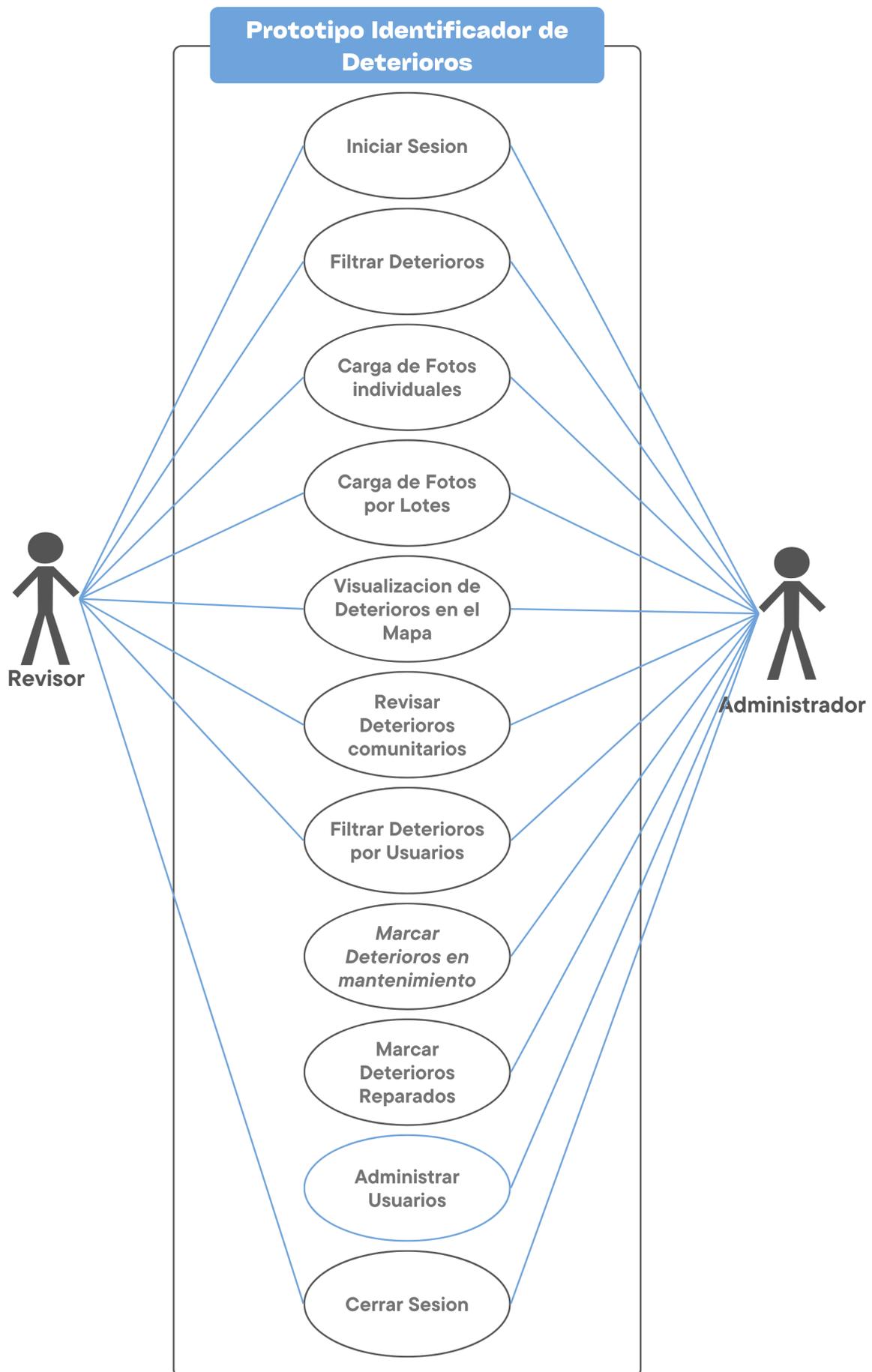
de las rutas geográficas, con la capacidad de categorizarlas como “en mantenimiento”, “reparadas” u otros estados relevantes que reflejen la situación actual de la infraestructura vial.

Figura 3.11: Diagrama de Casos de Uso para Poblador y Usuario



Fuente: Elaboración propia.

Figura 3.12: Diagrama de Casos de Uso para Revisor y Administrador



Fuente: Elaboración propia.

3.3.3. Fichas de Casos de Uso

Con el fin de detallar cada funcionalidad identificada en los diagramas de casos de uso, se han elaborado fichas que describen el comportamiento esperado del sistema ante distintas acciones de los usuarios, incluyendo el flujo normal, excepciones y una visión clara de las interacciones entre perfiles y prototipo.

Visualización de Deterioros en el Mapa

Tabla 3.2: Caso de uso visualización de deterioros en el mapa

Caso de uso	Mapa de la región del Cusco con sus respectivos deterioros.	
Visualizar Mapa		
Descripción	Esta funcionalidad permitirá visualizar los deterioros reportados por los pobladores en un mapa detallado de la región del Cusco, dividido por provincias. De esta manera, se facilitará la identificación y gestión de las áreas afectadas, proporcionando una visión clara y organizada del estado de las vías a nivel provincial.	
	Usuario	Sistema
Secuencia Normal	Poblador accede a la pagina principal del aplicativo web	El sistema carga filtros públicos y verifica el tipo perfil con el token del usuario Carga filtros especiales según el perfil del usuario, carga la leyenda del mapa y deterioros activos El sistema muestra el mapa con toda la región del Cusco y los marcadores de los deterioros
	Poblador selecciona un deterioro específico	El sistema extrae los detalles del deterioro y en un sidebar muestra el contenido.
	Poblador selecciona ver leyenda del Mapa	El sistema muestra y explica detalladamente el significado de cada tipo de Marcador dentro del mapa
	Poblador selecciona subir imagen	El sistema le redirige al modulo carga de fotos individuales
	Causa	Efecto
Excepciones	Poblador solicita subir lote de fotos	Sistema solicita crear cuenta o loguear al poblador para acceder a esta función

Filtrar Deterioros

Tabla 3.3: Caso de uso filtrar deterioros

Caso de uso	Mapa de la región marcada de los diferentes tipos de deterioros existentes en el mapa.	
Filtrar Deterioros		
Descripción	Esta funcionalidad brinda al usuario la posibilidad de crear una cuenta personal dentro de la plataforma. A través del proceso de registro, el usuario obtiene credenciales únicas que le permiten acceder a funciones avanzadas del sistema, como el historial de reportes realizados, la posibilidad de editar o eliminar sus propios reportes, y una interacción más personalizada con el entorno digital. De esta manera, se promueve una mayor participación ciudadana y se fortalece el control sobre la información que cada usuario aporta respecto al estado de las vías.	
	Usuario	Sistema
Secuencia Normal	Poblador accede a la pagina principal del aplicativo web	El sistema muestra el mapa con toda la región del Cusco
		El sistema extrae datos necesario para listar los Filtros del sistema
	Poblador selecciona los filtros que desea aplicar sobre el Mapa	El sistema utiliza el token de usuario para verificar si tiene los permisos necesarios para realizar ciertos filtros
		El sistema aplica los filtros solicitados por el usuario y enmarca solo los deterioros que están en el área de las provincias especificadas
	Poblador selecciona deshacer filtros	El sistema extrae todos los deterioros activos y limpia todos los filtros
	Causa	Efecto
Excepciones	Poblador solicita subir lote de fotos	Sistema solicita crear cuenta o iniciar sesion con sus credenciales al poblador para acceder a esta función

Carga de Fotos Individuales

Tabla 3.4: Caso de uso carga de fotos individuales

Caso de uso	Reportar imagen única con deterioros	
Carga de Fotos Individuales		
Descripción	Esta funcionalidad permitirá a los usuarios subir imágenes al sistema web, donde una red neuronal convolucional (CNN) identificará los deterioros presentes en ellas. Además, se extraerá automáticamente la geolocalización del lugar en el que fueron tomadas, facilitando así una documentación precisa y una gestión eficiente de las evidencias.	
	Usuario	Sistema
Secuencia Normal	El poblador carga o toma una foto y posteriormente solicita el análisis de esta al sistema	La imagen es analizada por la red neuronal convolucional
		Se extraen Metadatos de la imagen y se obtienen datos de su dirección
		El sistema devuelve la información extraída y los muestra en el sistema
	El poblador verifica los Bounding Box identificados por la CNN y agrega aquellos que cree pertinentes si los desea	El sistema selecciona los deterioros que el usuario considera correcto
	El usuario visualiza los metadatos del sistema.	El sistema guarda los cambios que genera el usuario.
	El usuario verifica la geoposición en la que se tomó la imagen	El sistema guarda los cambios que genera el usuario.
	El usuario completa sus ediciones y solicita Guardar los cambios.	El sistema guarda los datos en las tablas de la base de datos asegurándose adicionalmente si el usuario está logueado y muestra un mensaje que el proceso fue satisfactorio.
	Causa	Efecto
Excepciones	La imagen no es de un formato reconocible	Se rechaza la operación con un mensaje de error

Registro de usuarios

Tabla 3.5: Caso de uso registrarse

Caso de uso	Permite a un poblador crear una cuenta de usuario.	
Filtrar Deterioros		
Descripción	Esta funcionalidad permite a un poblador registrarse en el prototipo mediante la creación de una cuenta de usuario. A través de este proceso, se habilita un acceso autenticado que le otorga mayores privilegios dentro del sistema, como el seguimiento de sus reportes de deterioro, la posibilidad de gestionarlos y una interacción más completa con las funcionalidades disponibles.	
	Usuario	Sistema
Secuencia Normal	Poblador accede a la pagina de registro de usuario	El sistema muestra el formulario solicitando los datos del poblador
	Poblador completa sus datos y envía su información	El sistema valida los datos como el DNI y el nombre de usuario
		Después de la validación el sistema guarda los datos del usuario y crea la cuenta en la base de datos
		El sistema muestra un mensaje de registro exitoso.
		El sistema envía al usuario al login para que el usuario pueda acceder al sistema
	Causa	Efecto
Excepciones	EL usuario utiliza el mismo numero de documento en el registro	Sistema lanza error diciéndole que ya existe un usuario registrado con esos datos
	EL usuario utiliza el mismo nombre de usuario en sus credenciales	Sistema lanza error diciéndole que ya existe ese nombre de usuario solicitándole que modifique a uno nuevo y mas especifico.

Iniciar y Cerrar Sesión

Tabla 3.6: Caso de uso iniciar sesión

Caso de uso Iniciar Sesión	Permite acceder al sistema con sus credenciales.	
Descripción	Esta funcionalidad permite al usuario acceder al sistema mediante el uso de sus credenciales previamente registradas, es decir, su nombre de usuario y contraseña. El inicio de sesión autentica su identidad y le concede acceso a las funcionalidades específicas que corresponden a su perfil, permitiéndole interactuar con el sistema de manera segura y personalizada.	
	Usuario	Sistema
Secuencia Normal	Poblador accede a la pagina iniciar sesión	El sistema muestra el formulario de credenciales para su acceso.
	Poblador completa sus credenciales y envía su información	El sistema valida los datos.
		Crea un token de acceso al sistema y redirige al usuario al menú que le corresponda.
	Causa	Efecto
Excepciones	El usuario utiliza credenciales inválidas	Sistema lanza error mostrando un mensaje de credenciales incorrectas

Tabla 3.7: Caso de uso cerrar sesión

Caso de uso Cerrar Se- sión	Permite cerrar la sesión del sistema.	
Descripción	Permite al usuario finalizar su sesión actual en el sistema, cerrando de forma segura el acceso a su cuenta. Garantizando que ningún tercero pueda acceder a la información personal ni a las funciones asociadas al perfil del usuario.	
Secuencia Normal	Usuario	Sistema
	Usuario accede al sidebar que le permitirá cerrar la sesión	El valida si el token del usuario aun es valido.
	Usuario solicita cerrar sesión	El sistema invalida el token del usuario y lo redirige a la pagina principal.
Excepciones	Causa	Efecto
	El usuario utiliza credenciales inválidas	Sistema lanza error mostrando un mensaje de credenciales incorrectas

Cargar Fotos por Lotes

Tabla 3.8: Caso de uso cargar fotos por lotes

Caso de uso Cargar Fotos por Lotes	Permite subir un paquete de fotos al prototipo, en formato RAR.	
Descripción	Esta funcionalidad permite a los usuarios cargar múltiples fotografías comprimidas en un único archivo con extensión .rar. El sistema descomprime automáticamente el archivo al momento de la carga, para posteriormente analizar cada imagen contenida y extrae metadatos relevantes como la ubicación geográfica, la fecha de captura y el nombre del archivo. Esta opción está pensada para facilitar el envío masivo de evidencia visual, especialmente en contextos con conectividad limitada, donde resulta más práctico subir un único archivo comprimido que múltiples imágenes por separado. Asimismo, se mejora la organización y gestión del contenido subido, manteniendo la estructura de los archivos original.	
	Usuario	Sistema
Secuencia Normal	Usuario accede a la pagina de carga de fotos por lotes	El sistema verifica el token del usuario y muestra el componente que recepciona archivos de formato rar.
	El usuario selecciona su archivo y solicitar cargarlo al sistema	El sistema lleva el archivos alas carpetas del servidor. El sistema descomprime las imágenes y las guarda sin analizarlas.
		El sistema activa un servicio que analiza deterioros pendientes sin analizar.
	El usuario pasa a visualizar las imágenes que están siendo analizadas por CNN	El sistema Muestra un mensaje de que la operación fue exitosa. El sistema empieza a analizar imagen por imagen para que el usuario pueda revisar sus fotos en su lista de pendientes.
	Causa	Efecto
Excepciones	El usuario utiliza credenciales invalidas	Sistema lanza error mostrando un mensaje de credenciales incorrectas
	El archivo no tiene imágenes	Sistema muestra error explicando que el archivo no contiene fotos validas.

Revisar Deterioros Personales

Tabla 3.9: Caso de uso revisar deterioros personales

Caso de uso Revisar Deterioros Personales	Permite revisar los deterioros personales que fueron subidos por el mismo usuario de manera opcional.	
Descripción	Esta funcionalidad permite al usuario acceder a un listado detallado de los reportes que ha registrado previamente en la plataforma. Desde esta vista, puede consultar información relevante como ubicación, tipo de daño, fecha de carga y estado del reporte (pendiente, validado, etc.). Asimismo, el usuario puede revisar y validar sus propios registros, aportando mayor veracidad y confianza a la información. Esta herramienta facilita el seguimiento personalizado de sus aportes y fomenta la participación activa en la mejora del estado vial de su entorno.	
	Usuario	Sistema
Secuencia Normal	<p>Usuario accede a la pagina lsita de deterioros personales</p> <p>El usuario selecciona el deterioro que desea revisar</p> <p>El usuario selecciona la imagen con la que desea trabajar, ya sea la original, la que se reajusto o la que fue editada por la CNN.</p> <p>El usuario selecciona los Bounding Boxes que percibe como validos</p> <p>El usuario revisa y guarda los deterioros</p>	<p>El sistema verifica el token del usuario y su validez</p> <p>El sistema solicita y muestra los deterioros del usuario</p> <p>El sistema solicita los detalles.</p> <p>El sistema muestra el deterioro de manera detallada, lista para su evaluación.</p> <p>El sistema muestra los Bounding Boxes de la imagen.</p> <p>El sistema muestra los metadatos de la imagen.</p> <p>El sistema guarda los datos editados y sube el nivel de veracidad en el deterioro según el tipo de usuario.</p>
	Causa	Efecto
Excepciones	<p>El usuario utiliza credenciales invalidas</p> <p>El sistema no detecto deterioros en la imagen</p>	<p>Sistema lanza error mostrando un mensaje que la sesión expiro.</p> <p>Solicita al usuario seleccionar un bounding box como mínimo.</p>

Revisar Deterioros Comunitarios

Tabla 3.10: Caso de uso revisar comunitarios

Caso de uso Revisar Deterioros Comunitarios	Permite validar los deterioros de todos los usuarios.	
Descripción	Esta funcionalidad permite a los usuarios autorizados revisar los reportes registrados por la comunidad, evaluando su veracidad mediante la ubicación, descripción, imágenes y autor. El revisor puede validarlos o marcarlos como inválidos si presentan inconsistencias. Esta herramienta mejora la calidad de la información y fortalece el control comunitario para una gestión vial más transparente y eficaz.	
	Usuario	Sistema
Secuencia Normal	Usuario accede a la pagina de deterioros comunitarios	El sistema verifica el token del usuario y su validez de ser un perfil revisor o administrador
		El sistema solicita y muestra la lista de usuarios en el sistema
	El revisor selecciona el usuario	El sistema solicita todos los deterioros pendiente de evaluación de ese usuario.
	El revisor selecciona la imagen que desea evaluar, ya sea la original, la que se reajusto o la que fue editada por la CNN.	El sistema muestra los Bounding Boxes de la imagen.
	El revisor selecciona los Bounding Boxes que percibe como validos	El sistema muestra los metadatos de la imagen.
	El revisor revisa y guarda los deterioros	El sistema guarda los datos editados y sube el nivel de veracidad en el deterioro según el tipo de usuario.
	Causa	Efecto
Excepciones	El sistema no detecto deterioros en la imagen	Solicita al usuario seleccionar un bounding box como mínimo

Filtrar Deterioros por Usuarios

Tabla 3.11: Caso de uso filtrar deterioros por usuarios

Caso de uso Filtrar Deterioros por Usuarios	Permite filtrar los deterioros por usuario dentro del mapa.	
Descripción	Esta funcionalidad brinda la posibilidad de aplicar filtros personalizados sobre los deterioros visualizados en el mapa, permitiendo al usuario seleccionar específicamente aquellos reportes que hayan sido realizados por un determinado poblador. Al hacer uso de esta opción, el sistema actualiza la visualización del mapa para mostrar únicamente los deterioros asociados al usuario seleccionado, facilitando así el seguimiento, la revisión y la gestión individualizada de los reportes. Esta herramienta resulta útil tanto para fines administrativos como para fomentar la transparencia y la trazabilidad de las contribuciones realizadas por cada miembro de la comunidad.	
	Usuario	Sistema
Secuencia Normal	<p>Usuario accede al menú principal</p> <p>El revisor o administrador selecciona los usuarios que desea filtrar</p> <p>El revisor o administrador selecciona algún deterioro que desea revisar a detalle.</p>	<p>El sistema verifica el token del usuario y su validez de ser un perfil revisor o administrador</p> <p>El sistema solicita la lista de usuarios en el sistema</p> <p>El sistema alimenta el filtro de usuarios y los muestra en el sistema</p> <p>El sistema solicita los deterioros cargados por esos usuarios.</p> <p>El sistema actualiza los marcadores en el mapa.</p> <p>El sistema solicita detalles del usuario y los muestra sus detalles en un sidebar.</p>
	Causa	Efecto
Excepciones	<p>El usuario utiliza credenciales inválidas</p> <p>El sistema no detecto deterioros su- bido por un usuario</p>	<p>Sistema lanza error mostrando un mensaje que el usuario no tiene suficientes permisos.</p> <p>Lanza mensaje, el usuario seleccionado no tiene ningún deterioro cargado</p>

Marcar Deterioros en Mantenimiento

Tabla 3.12: Caso de uso marcar deterioros en mantenimiento

Caso de uso Marcar Deterioros En Mantenimiento	Permite dibujar una ruta en el Mapa para poner todos los deterioros cercanos en mantenimiento.	
Descripción	Esta funcionalidad permite al administrador del sistema trazar manualmente una ruta o un área sobre el mapa, con el fin de identificar y seleccionar de forma automática todos los deterioros geográficamente cercanos al recorrido marcado. Una vez definidos los límites, el sistema procede a actualizar el estado de dichos deterioros, clasificándolos como En Mantenimiento. Esta herramienta facilita una gestión masiva y eficiente de los reportes en zonas específicas, especialmente útil para intervenciones programadas por tramos viales o regiones afectadas, optimizando así el trabajo de seguimiento y control del avance de las labores de reparación.	
Secuencia Normal	Usuario	Sistema
	Usuario accede al menú principal	El sistema verifica el token del usuario y su validez de ser un perfil administrador
	El administrador realizar el trazo de una ruta alineado por puntos	El sistema analiza y verifica todos los deterioros cercanos a esa ruta para seleccionarlos y activa el botón para poder solicitar cambio de esta a En Mantenimiento
	El administrador solicita ponerlos en mantenimiento	El sistema envía toda la lista de deterioros que deben cambiar de estado a En Mantenimiento El sistema elimina el trazo, actualiza los deterioros y el Mapa El sistema muestra mensaje que el proceso fue satisfactorio
Excepciones	Causa	Efecto
	El usuario utiliza credenciales inválidas	Sistema lanza error mostrando un mensaje que el usuario no tiene suficientes permisos.
	El sistema no detecto deterioros en la ruta	Lanza mensaje, el sistema no encontró ningún deterioro cercano, limpia el trazo

Marcar Deterioros Reparados

Tabla 3.13: Caso de uso marcar deterioros reparados

Caso de uso Marcar Deterioros Reparados	Permite dibujar una ruta en el Mapa para poner todos los deterioros que están en mantenimiento a reparados.	
Descripción	Esta funcionalidad permite al administrador trazar una ruta o área dentro del mapa para identificar los deterioros que se encuentran actualmente en estado de En Mantenimiento. Una vez delimitada la zona, el sistema actualiza el estado de dichos deterioros a Reparado. Esta herramienta está diseñada para facilitar la actualización colectiva de múltiples registros, asegurando que la información sobre el estado de las vías se mantenga precisa y actualizada.	
	Usuario	Sistema
Secuencia Normal	Usuario accede al menú principal	El sistema verifica el token del usuario y su validez de ser un perfil administrador
	El administrador realizar el trazo de una ruta alineado por puntos	El sistema analiza y verifica todos los deterioros con estado En Mantenimiento, cercanos en esa ruta para seleccionarlos y activa el botón que permitirá solicitar el cambio de estos a Reparado
	El administrador solicita ponerlos como Reparados	El sistema envía toda la lista de deterioros que deben cambiar de estado a Reparado
		El sistema elimina el trazo, actualiza los deterioros y el Mapa
		El sistema muestra mensaje que el proceso fue satisfactorio
	Causa	Efecto
Excepciones	El usuario utiliza credenciales inválidas	Sistema lanza error mostrando un mensaje que el usuario no tiene suficientes permisos.
	El sistema no detecto deterioros En Mantenimiento en la ruta	Lanza mensaje, el sistema no encontró ningún deterioro que este siendo reparado cerca del trazo, limpia el trazo

Administrar Usuarios

Tabla 3.14: Caso de uso administrar usuarios

Caso de uso Administrar Usuarios	Permite dibujar administrar los usuarios, permitiendo actualizar y cambiar sus perfiles.	
Descripción	Esta funcionalidad permite gestionar a los usuarios registrados en el sistema. El administrador podrá visualizar la lista de usuarios, actualizar su información, modificar sus perfiles de acceso y realizar ajustes según los roles asignados, garantizando así un control adecuado sobre los permisos y funciones de cada usuario dentro de la plataforma.	
Secuencia Normal	Usuario	Sistema
	Usuario accede a la lista de Usuarios	El sistema verifica el token del usuario y su validez de ser un perfil administrador
		El sistema solicita la lista de Usuarios y los muestra en una tabla
	El administrador realiza una búsqueda y selecciona un Usuario	El sistema solicita los datos completos del usuario y los muestra al en un sidebar
	El administrador modifica o actualiza los datos del usuario	El sistema pregunta si esta seguro de realizar esos cambios
	El administrador confirma que se ejecute el cambio	El sistema actualiza la información
Excepciones	Causa	Efecto
	El usuario utiliza credenciales inválidas	Sistema lanza error mostrando un mensaje que el usuario no tiene suficientes permisos.

3.3.4. Diagramas de Flujo

Con el propósito de complementar la descripción funcional del sistema, se presentan los diagramas de flujo correspondientes a los casos de uso identificados. Estos diagramas permiten visualizar de forma secuencial y gráfica el comportamiento del sistema ante diferentes acciones realizadas por los actores, facilitando la comprensión de la lógica operativa involucrada en cada proceso.

Cada diagrama refleja el flujo de actividades, decisiones y condiciones que deben cumplirse para alcanzar los objetivos definidos por cada funcionalidad, desde la perspectiva del usuario y del sistema. Esta representación resulta útil tanto para el análisis como

para el desarrollo e implementación del sistema.

Visualizar deterioros en el mapa

El diagrama 3.13 muestra de manera gráfica la Tabla 3.2, donde los cuatro tipos de usuarios descritos previamente en la Figura 3.11 tienen acceso a este módulo de visualización de deterioros. Sin embargo, el alcance de sus funcionalidades varía según el perfil del usuario. Por ejemplo, el revisor puede aplicar filtros específicos por usuario, mientras que el administrador cuenta con herramientas adicionales, como la posibilidad de trazar rutas sobre el mapa para seleccionar y gestionar grupos de deterioros de manera masiva.

Filtrar deterioros

El diagrama mostrado en la Figura 3.14 representa gráficamente el flujo correspondiente al caso de uso descrito en la Tabla 3.3. En este flujo se establece que es indispensable que la página principal (HomePage) y el mapa del sistema se encuentre completamente cargada, ya que solo en dicha condición es posible aplicar los filtros directamente sobre el mapa.

Cargar Foto

El diagrama mostrado en la Figura 3.15 representa gráficamente el flujo correspondiente al caso de uso descrito en la Tabla 3.4. En él se detalla el proceso estándar que sigue una imagen con deterioros, desde la extracción de sus metadatos hasta la detección y delimitación de los deterioros mediante los cuadros delimitadores (bounding boxes) generados por la red neuronal convolucional (CNN).

Registrarse

El diagrama mostrado en la Figura 3.16 representa gráficamente el flujo correspondiente al caso de uso descrito en la Tabla 3.5. En este se detalla el proceso de registro de un nuevo usuario y la forma en que el sistema valida los datos proporcionados, asegurando la unicidad de las credenciales y evitando conflictos por información duplicada.

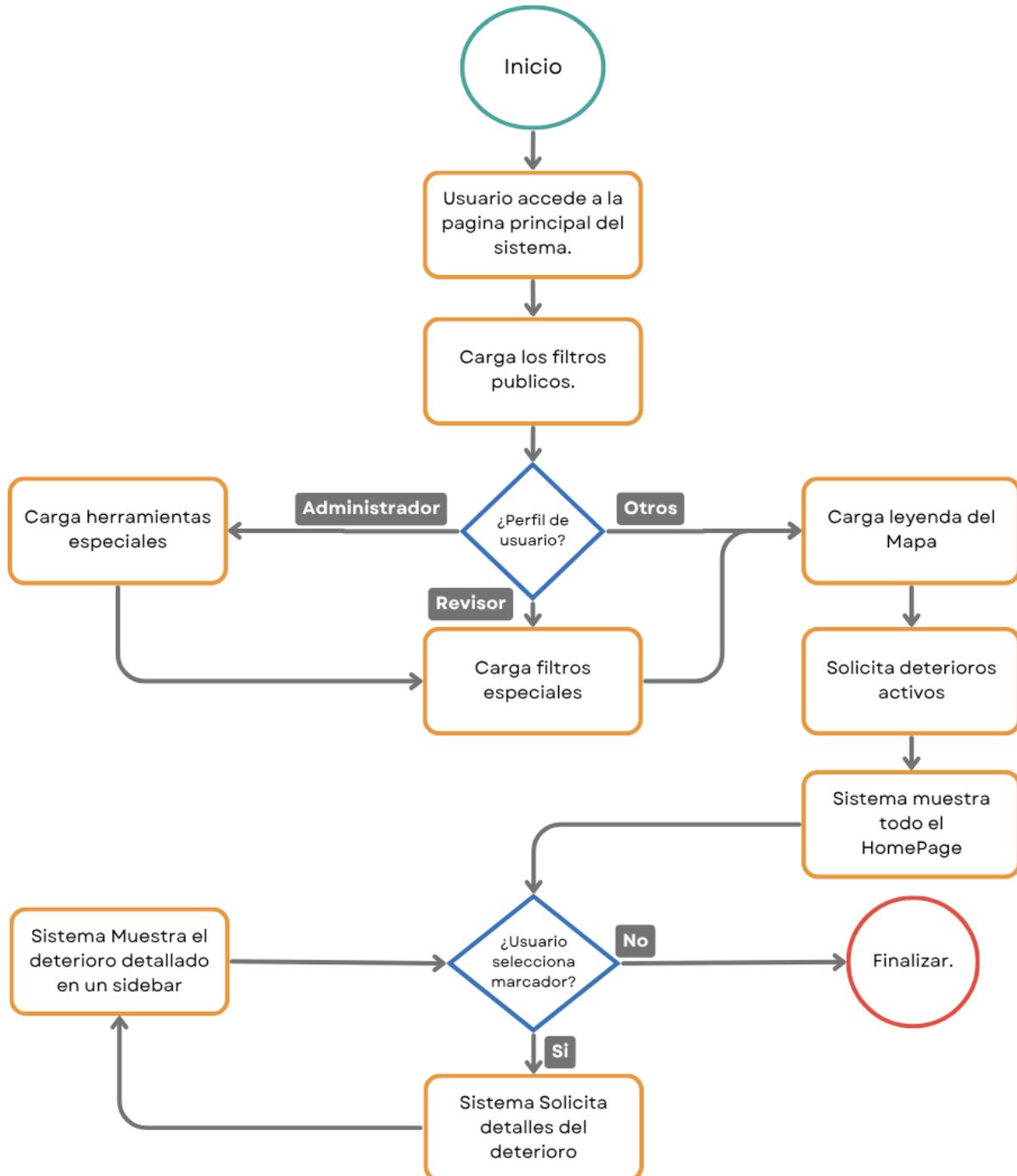
Iniciar Sesión

El diagrama mostrado en la Figura 3.17 representa gráficamente el flujo correspondiente al caso de uso descrito en la Tabla 3.6. En él se muestra el proceso que debe seguir un usuario para acceder al sistema, desde el ingreso de sus credenciales hasta la validación por parte del sistema. Se resalta el uso del token de autenticación, el cual es generado al iniciar sesión y permite respaldar de forma segura cada solicitud realizada durante la sesión activa del usuario.

Figura 3.13: Diagrama de flujo visualizar deterioros en mapa

VISUALIZACION DE DETERIOROS EN MAPA

Diagrama de flujo del caso de uso visualizar deterioro.

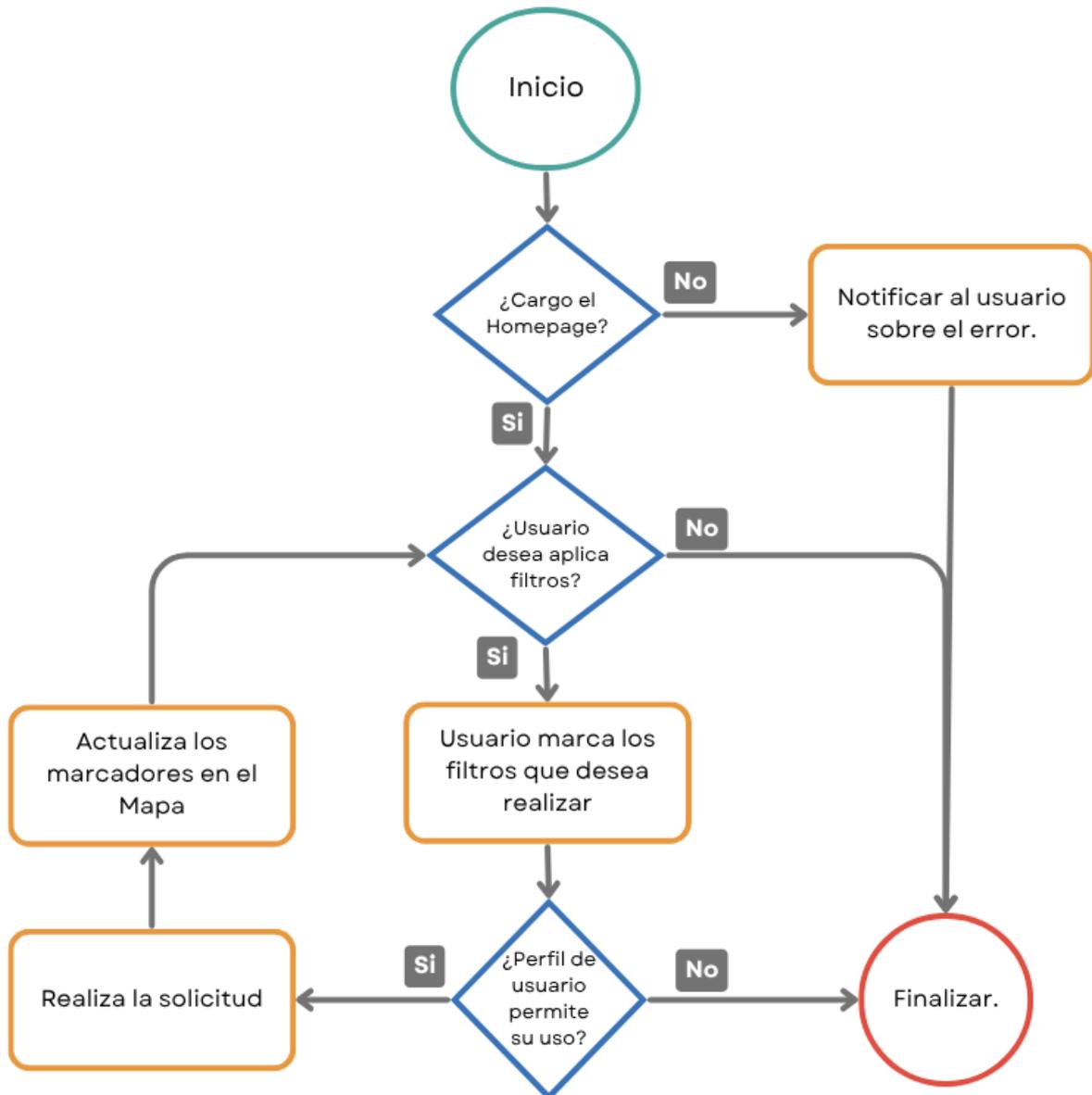


Fuente: Elaboración propia.

Figura 3.14: Diagrama de flujo filtrar deterioros

FILTRAR DETERIOROS

Diagrama de flujo del caso de uso filtrar deterioros.

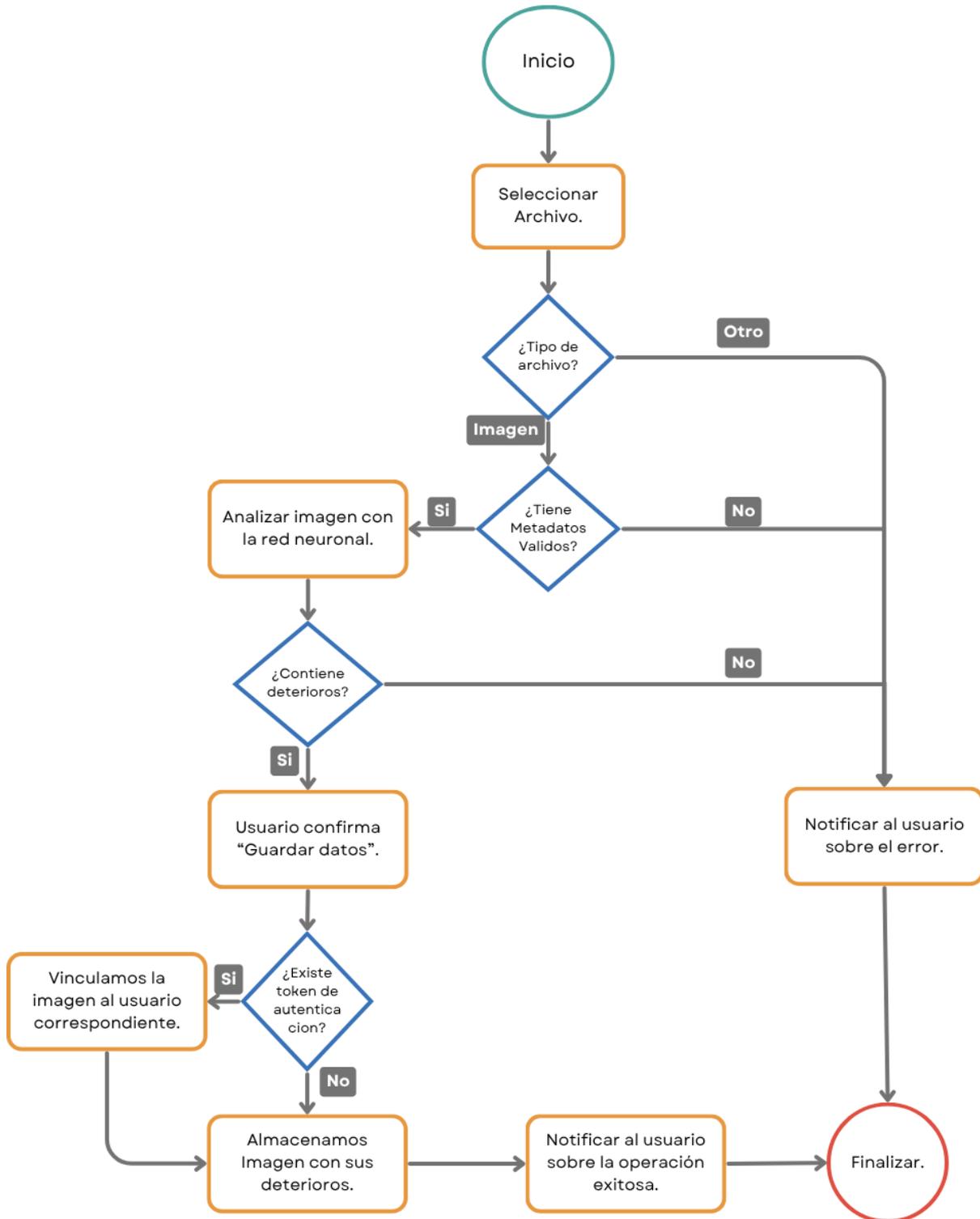


Fuente: Elaboración propia.

Figura 3.15: Diagrama de Flujo Cargar Foto

CARGAR FOTO

Diagrama de flujo del caso de uso cargar foto.

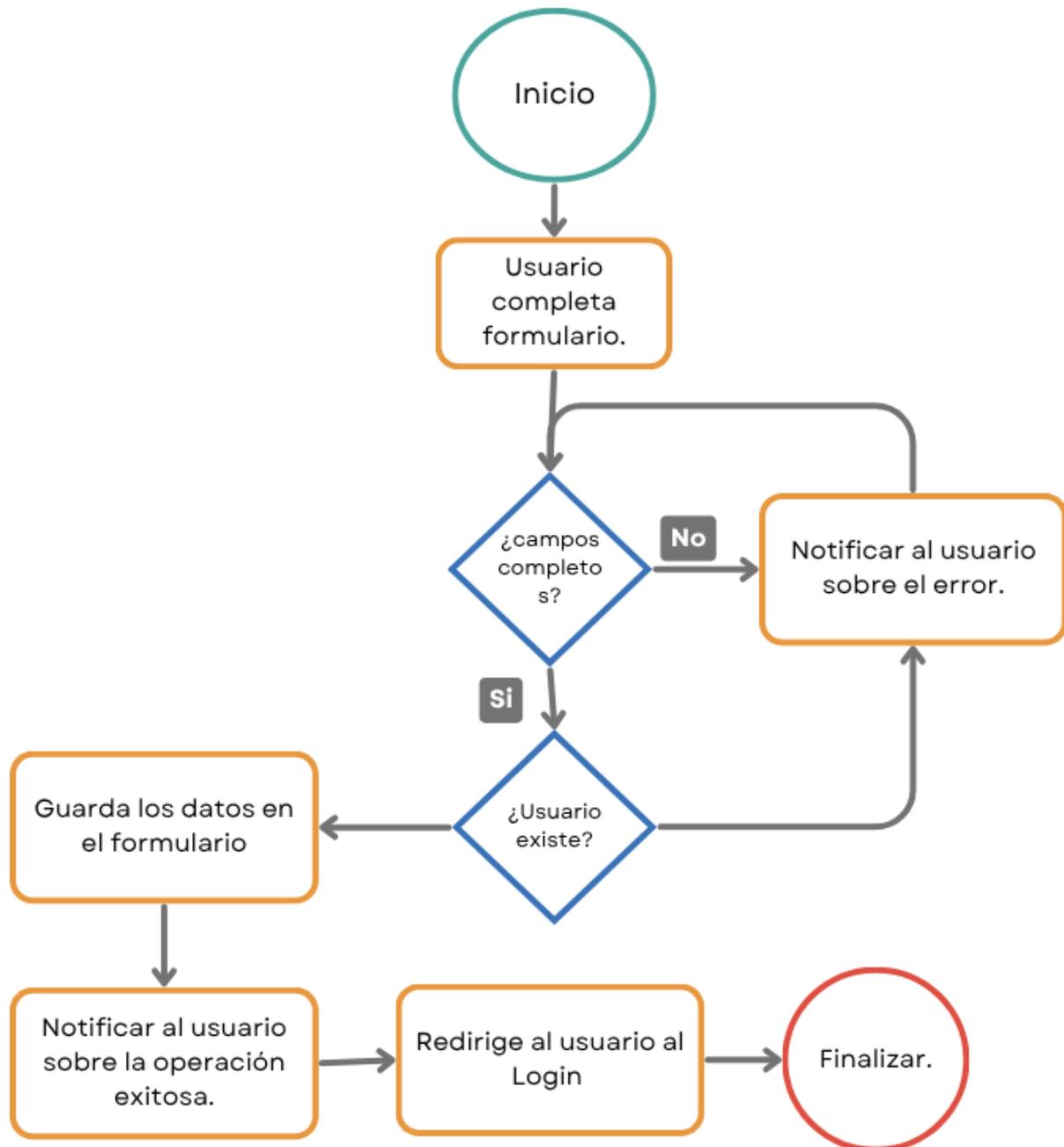


Fuente: Elaboración propia.

Figura 3.16: Diagrama de Flujo Registrarse

REGISTRARSE

Diagrama de flujo del caso de uso registrarse.

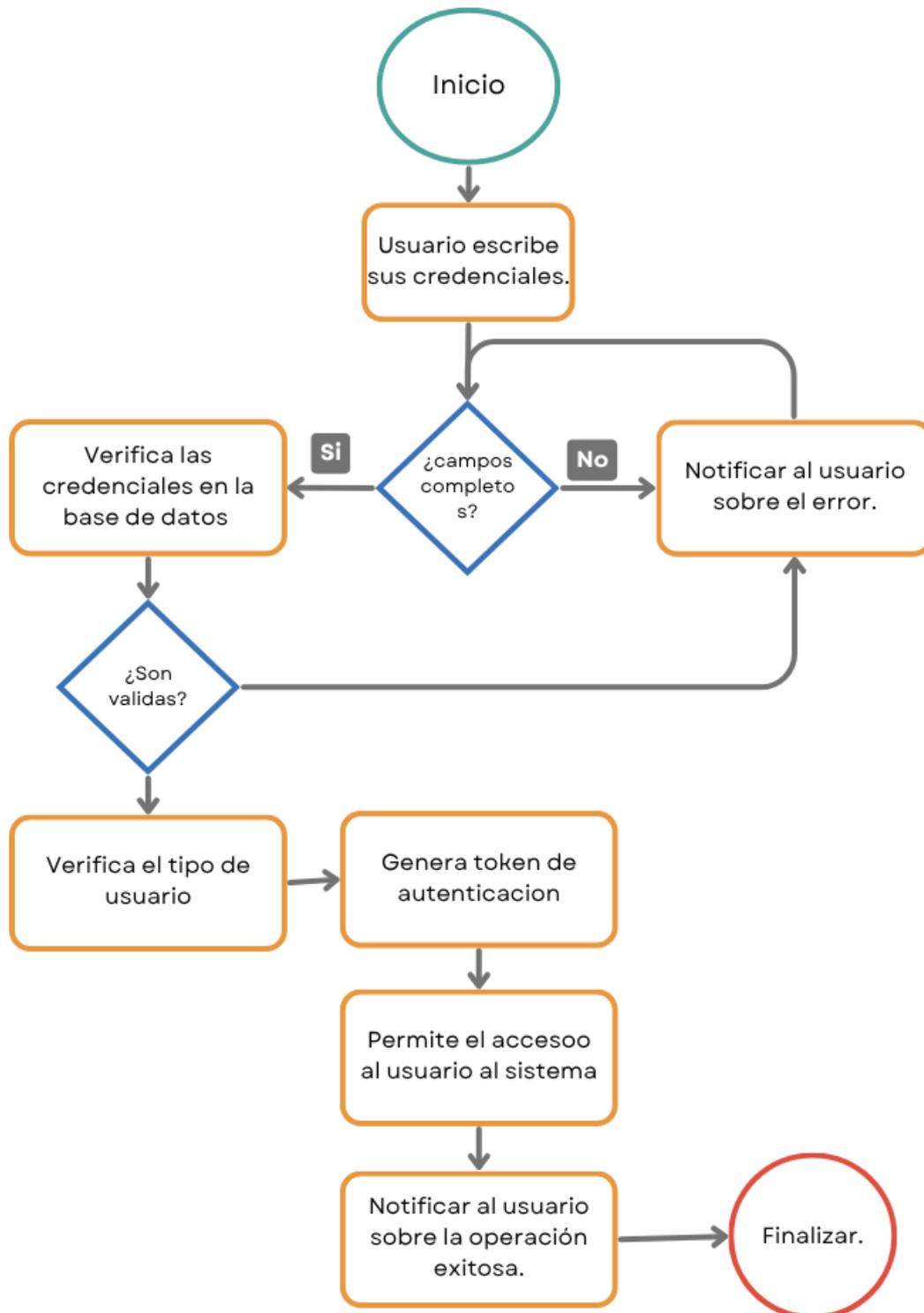


Fuente: Elaboración propia.

Figura 3.17: Diagrama de Flujo Iniciar Sesión

INICIAR SESION

Diagrama de flujo del caso de uso iniciar sesion.



Fuente: Elaboración propia.

Cerrar Sesión

El diagrama mostrado en la Figura 3.18 representa gráficamente el flujo correspondiente al caso de uso descrito en la Tabla 3.7. En él se evidencia que todo el respaldo de una sesión activa se sustenta en el token generado durante el inicio de sesión del usuario, el cual es invalidado al momento de cerrar sesión, garantizando así la seguridad del acceso al sistema.

Figura 3.18: Diagrama de Flujo Cerrar Sesión

CERRAR SESION

Diagrama de flujo del caso de uso cerrar sesion.



Fuente: Elaboración propia.

Cargar Fotos por Lotes

El diagrama mostrado en la Figura 3.19 representa el flujo correspondiente al caso de uso descrito en la Tabla 3.8. Este módulo se divide en dos fases: la primera permite cargar un archivo comprimido en formato RAR con múltiples imágenes, que el sistema extrae y almacena automáticamente.

La segunda fase activa un evento asincrónico encargado de procesar cada imagen de forma individual. Esta tarea, gestionada por el backend mediante una CNN, se realiza sin intervención del usuario, mejorando la experiencia al no requerir espera activa. Además, esta arquitectura permite analizar grandes volúmenes de imágenes sin afectar el rendimiento del sistema.

Revisar Deterioros

El diagrama mostrado en la Figura 3.20 representa gráficamente el flujo correspondiente a los casos de uso descritos en las Tablas 3.9 y 3.10. Ambos casos siguen un proceso similar, pero se diferencian por una condición determinada por el perfil de usuario logueado, lo que activa diferentes flujos y procesos adicionales según el tipo de usuario.

Administrar Usuarios

El diagrama mostrado en la Figura 3.21 representa gráficamente el flujo correspondiente al caso de uso descrito en la Tabla 3.14. Este módulo está diseñado para ser utilizado por los administradores del sistema, quienes cuentan con la autorización necesaria para gestionar las cuentas de los usuarios. A través de este flujo, el administrador tiene la capacidad de editar, actualizar y reconfigurar los perfiles de los usuarios dentro del sistema.

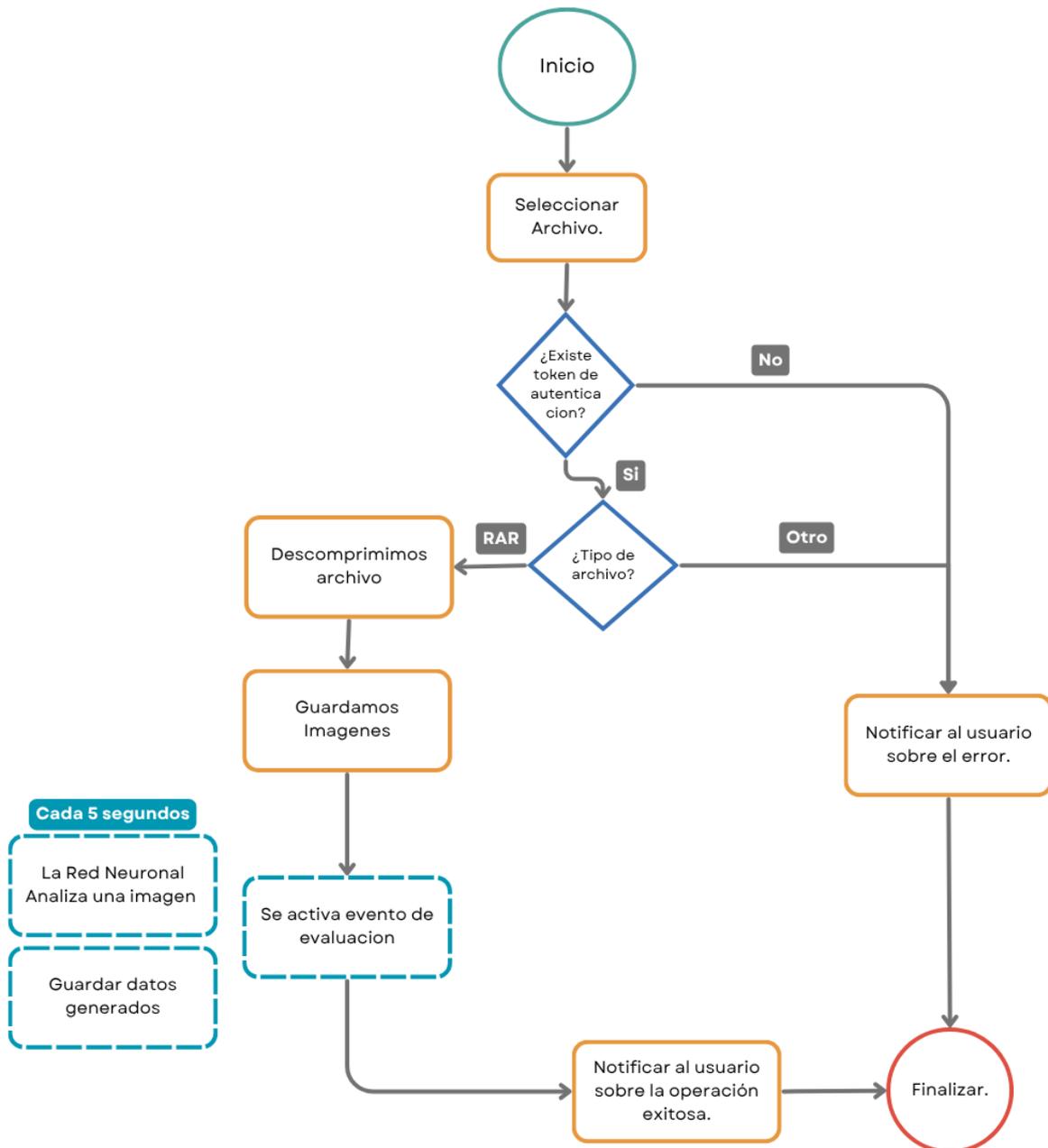
Marcar Deterioros En Mantenimiento y Reparados

El diagrama mostrado en la Figura 3.22 representa gráficamente el flujo correspondiente a dos casos de uso descritos en las Tablas 3.12 y 3.13. Esta funcionalidad está disponible únicamente para los administradores del sistema y describe dos procesos que, aunque similares, se encuentran relacionados jerárquicamente. La principal diferencia radica en que un deterioro solo puede ser marcado como “reparado” si previamente ha sido clasificado como “en mantenimiento”, estableciendo así una dependencia lógica entre ambas acciones para garantizar la coherencia del estado de los datos.

Figura 3.19: Diagrama de Flujo Cargar Fotos por Lotes

CARGAR FOTOS POR LOTES

Diagrama de flujo del caso de uso cargar fotos.

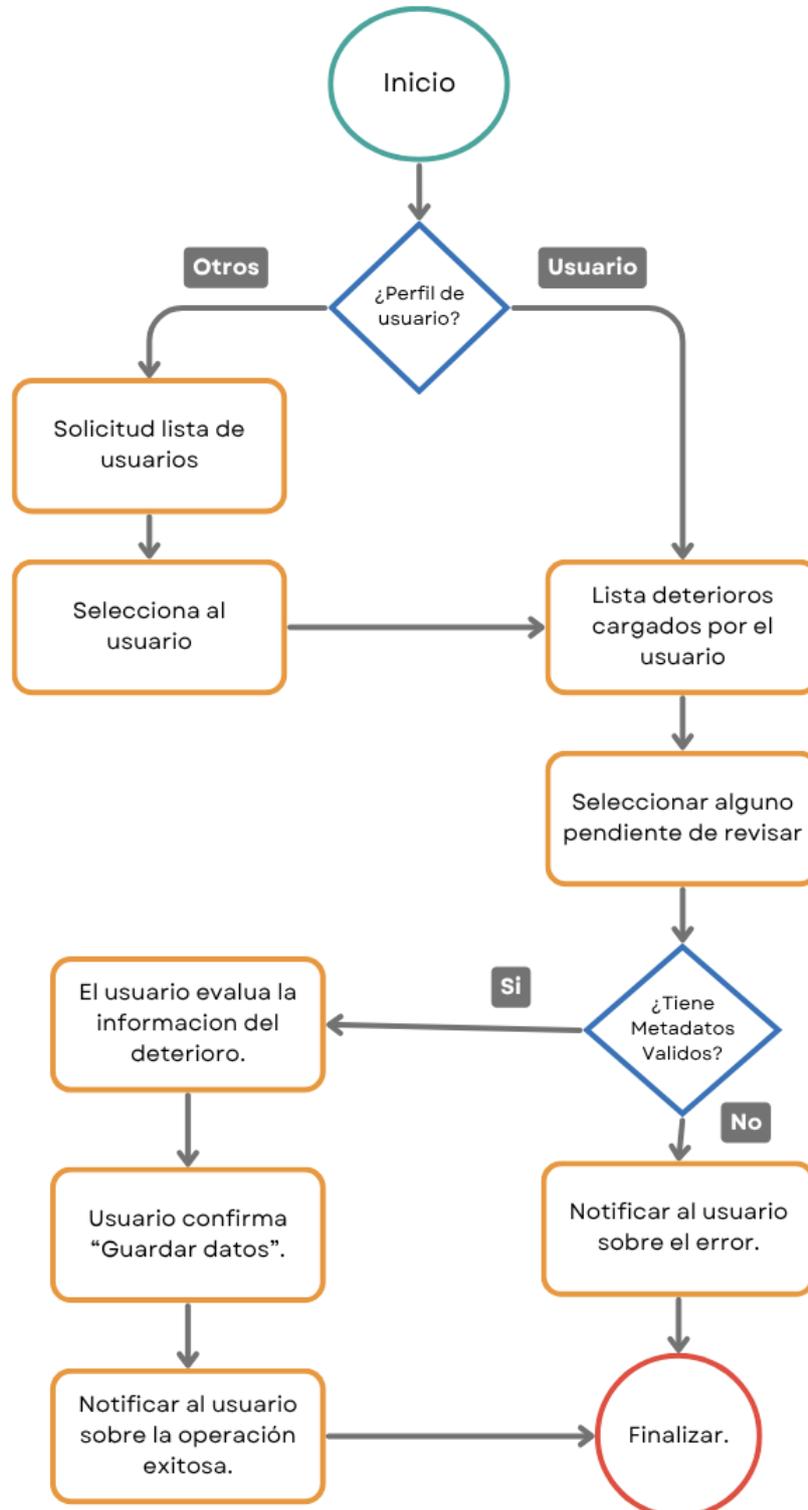


Fuente: Elaboración propia.

Figura 3.20: Diagrama de Flujo Revisar Deterioros

REVISAR DETERIOROS

Diagrama de flujo del caso de uso revisar deterioro.

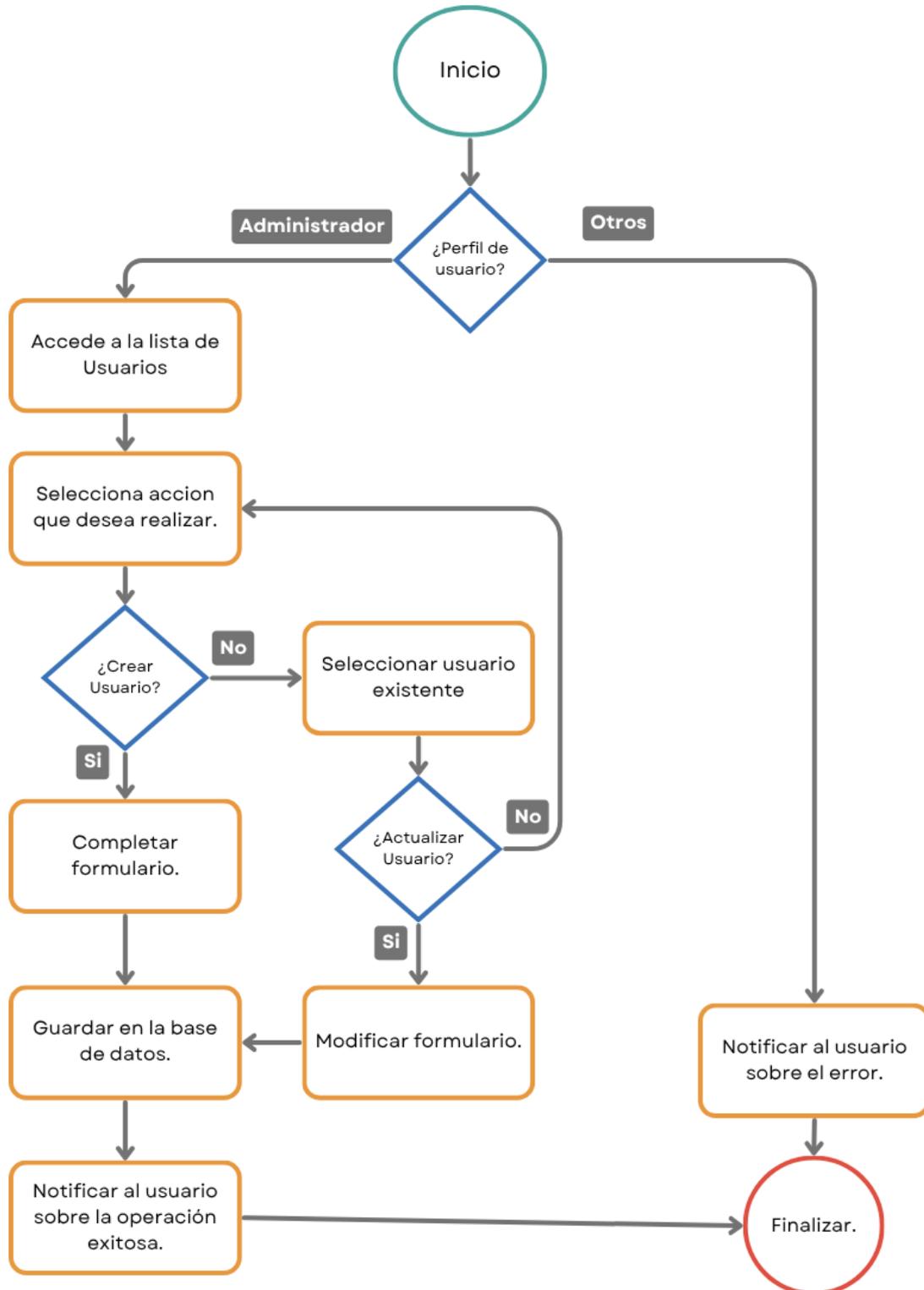


Fuente: Elaboración propia.

Figura 3.21: Diagrama de Flujo Administrar Usuarios

ADMINISTRAR USUARIOS

Diagrama de flujo del caso de uso administrar usuarios.

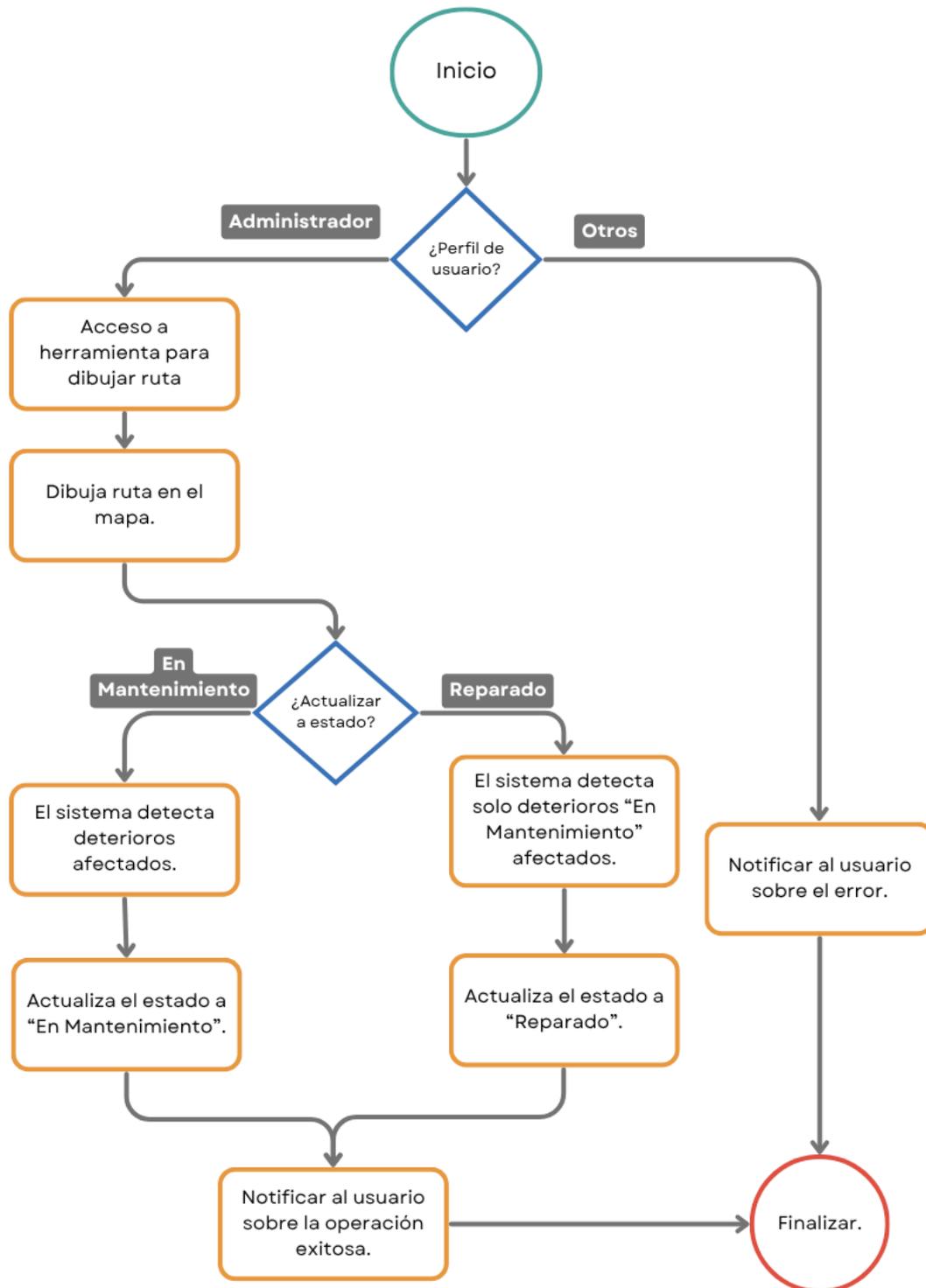


Fuente: Elaboración propia.

Figura 3.22: Diagrama de Flujo Marcar Deterioros Reparados

MARCAR DETERIOROS REPARADOS

Diagrama de flujo del caso de uso marcar deterioros.



Fuente: Elaboración propia.

3.3.5. Diagramas de Secuencia

En los siguientes diagramas se mostrara de manera grafica como se desarrollan las interacciones clave entre los actores del sistema (usuarios, administradores, revisores, etc.) y los distintos módulos funcionales involucrados en cada proceso. Estas ilustraciones permiten comprender no solo el orden en que ocurren las acciones, sino también la forma en que los distintos elementos del sistema colaboran para llevar a cabo tareas específicas, como el registro de usuarios, la autenticación, la carga de imágenes o la gestión de deterioros.

Cada representación refleja el flujo cronológico de mensajes, solicitudes y respuestas que se intercambian durante la ejecución de un caso de uso determinado. Este enfoque permite evidenciar la lógica de comunicación interna entre el cliente y el servidor, así como el comportamiento esperado del sistema ante ciertas condiciones. De esta manera, se proporciona una visión clara y estructurada del funcionamiento del sistema desde una perspectiva orientada a la interacción.

Mostrar deterioros en el mapa

La Figura 3.23 representa la secuencia de interacciones que permite a los distintos tipos de usuarios poblador, usuario registrado, revisor y administrador acceder al módulo de visualización de deterioros. Todos los actores pueden consultar el mapa con los marcadores correspondientes y explorar los detalles individuales de cada deterioro detectado. No obstante, ciertas funcionalidades específicas, como la aplicación de filtros avanzados por usuario o por estado, están restringidas a perfiles con mayores permisos, como los revisores y administradores. Esta segmentación garantiza tanto la accesibilidad general del módulo como el control de operaciones más sensibles dentro del sistema.

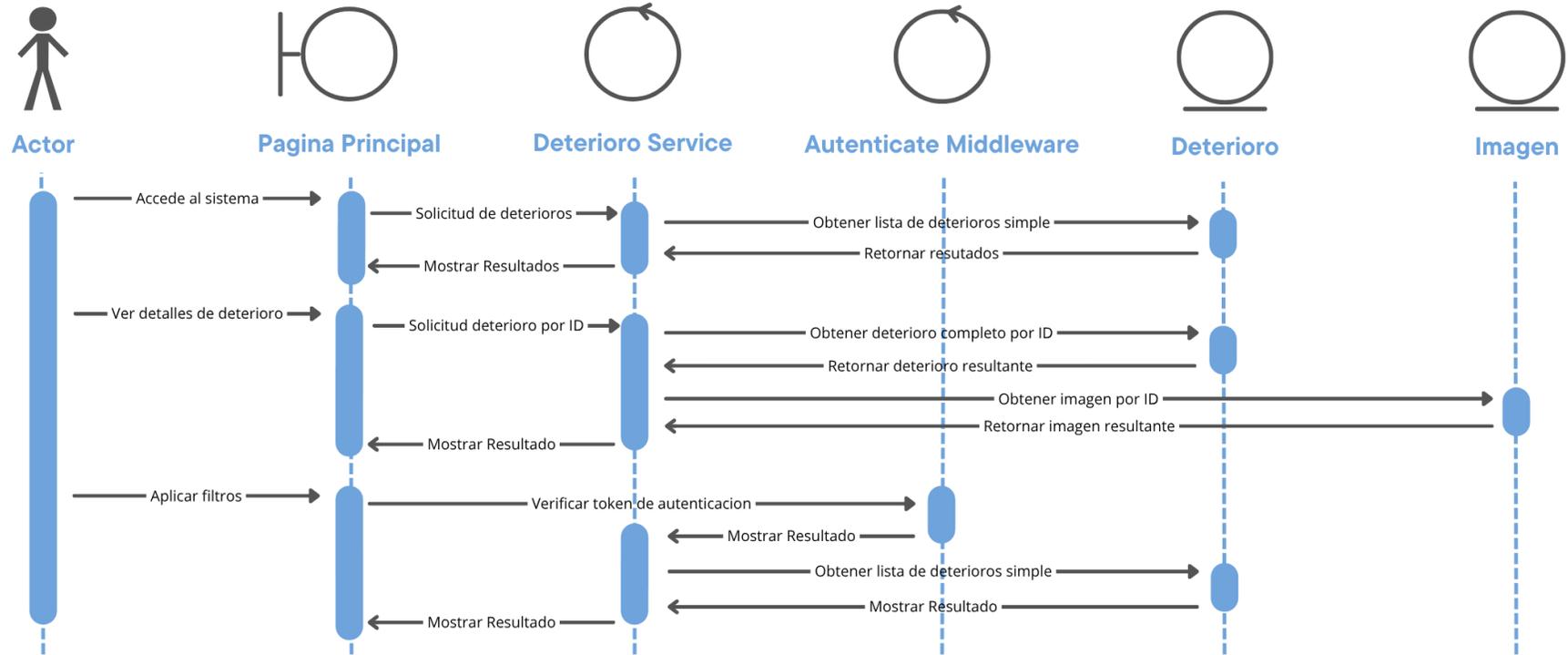
Cargar foto

La Figura 3.24 ilustra el proceso mediante el cual los distintos tipos de usuario pueden cargar imágenes con deterioros al sistema. Esta funcionalidad está disponible para todos los perfiles, desde pobladores hasta administradores. Sin embargo, existe una diferencia fundamental: en el caso de los usuarios autenticados, las imágenes quedan asociadas directamente a sus cuentas, lo que permite un seguimiento personalizado. Por el contrario, cuando la carga es realizada por un poblador sin credenciales, el deterioro registrado no se vincula a ningún usuario específico, manteniéndose como un reporte anónimo dentro del sistema.

Figura 3.23: Diagrama de secuencia visualización de deterioros en el mapa

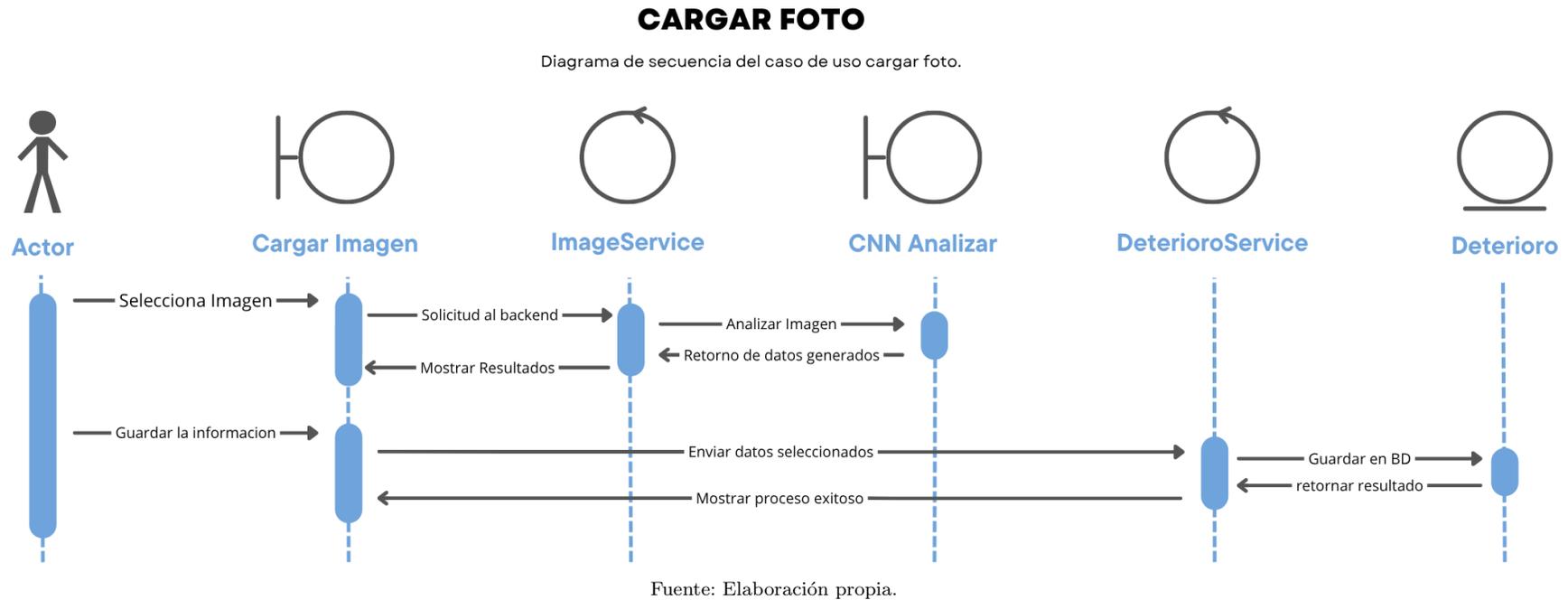
VISUALIZACION DE DETERIOROS EN EL MAPA

Diagrama de secuencia del caso de uso visualizar deterioro.



Fuente: Elaboración propia.

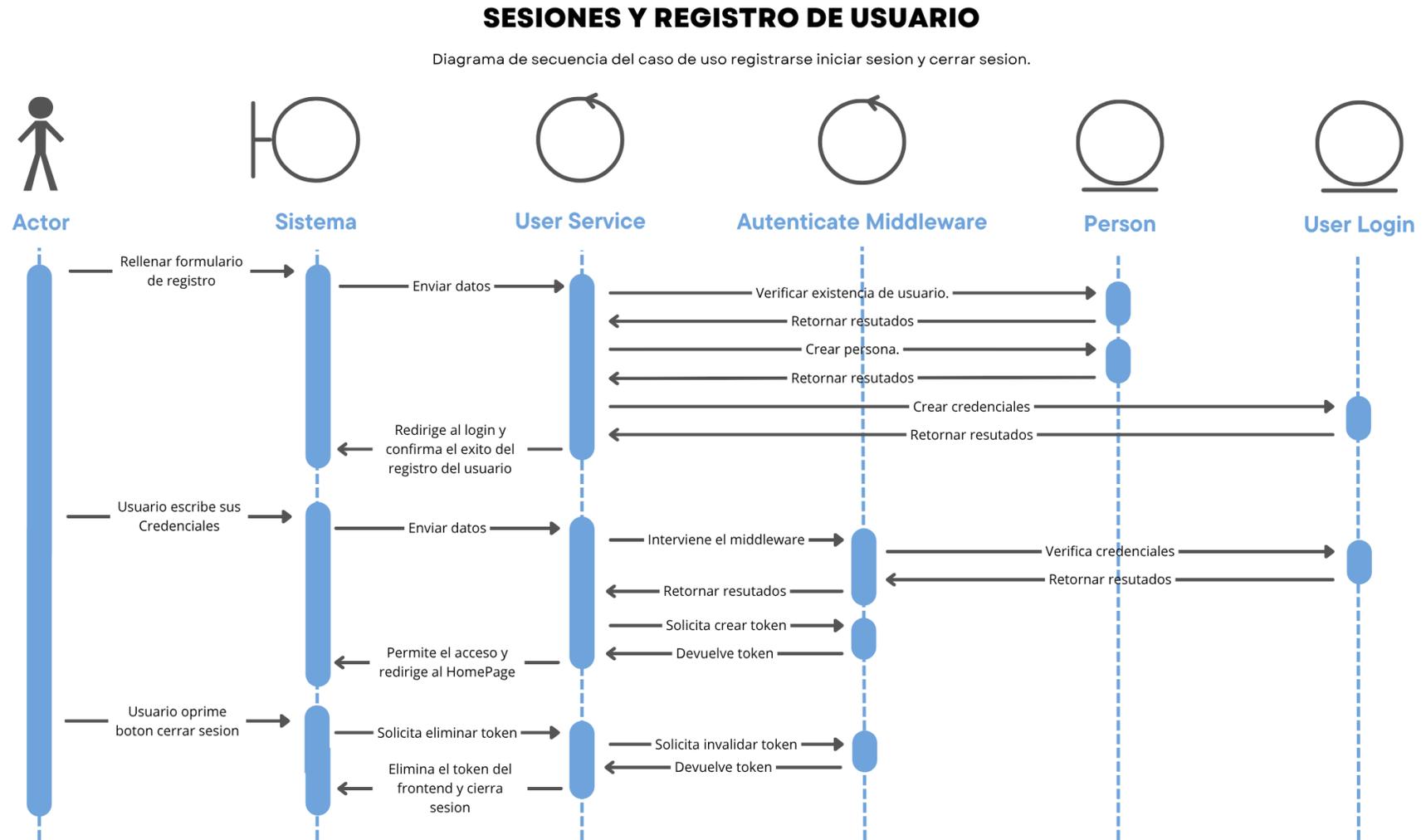
Figura 3.24: Diagrama de secuencia cargar foto



Sesiones y registro de usuarios

La Figura 3.25 muestra el proceso de ingreso al sistema mediante un token encryptado, el cual respalda cada acción del usuario como mecanismo de seguridad. También se ilustra cómo este token es eliminado al cerrar sesión, garantizando así un acceso seguro y controlado.

Figura 3.25: Diagrama de secuencia sesiones y registro de usuarios

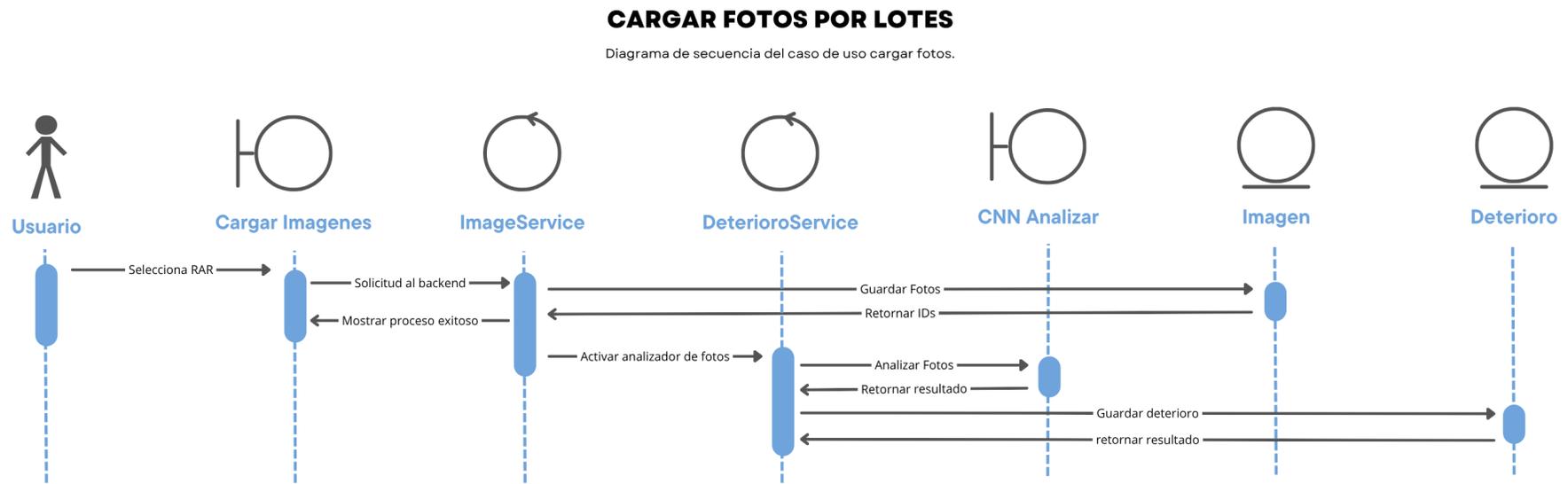


Fuente: Elaboración propia.

Cargar foto por lotes

La Figura 3.26 muestra cómo las imágenes pueden subirse de forma sencilla al sistema mediante un archivo comprimido, y cómo estas son analizadas automáticamente por la red neuronal convolucional sin necesidad de que el usuario espere a que finalice todo el procesamiento. Esto permite una navegación fluida y continua por el sistema mientras las imágenes se procesan en segundo plano.

Figura 3.26: Diagrama de secuencia cargar foto por lotes



Fuente: Elaboración propia.

Revisión de deterioros personales

La Figura 3.27 ilustra dos funcionalidades principales para la revisión de deterioros. La primera permite visualizar toda la lista de deterioros asociados al usuario, facilitando la búsqueda y selección de aquellos que necesitan ser revisados individualmente. La segunda funcionalidad ofrece una herramienta que muestra directamente los detalles de los deterioros pendientes, utilizando un índice que recorre la base de datos para identificarlos rápidamente, optimizando el proceso de revisión.

Diagrama de secuencia revisión de deterioros comunitarios y por usuarios

La Figura 3.28 muestra el módulo que está restringido a ser utilizado únicamente por revisores y administradores. De manera similar a la revisión individual de deterioros, este módulo permite la visualización de los deterioros pendientes, pero con la diferencia de que ofrece un acceso más amplio, permitiendo gestionar y revisar los deterioros reportados por todos los usuarios del sistema, no solo los asociados a una cuenta específica.

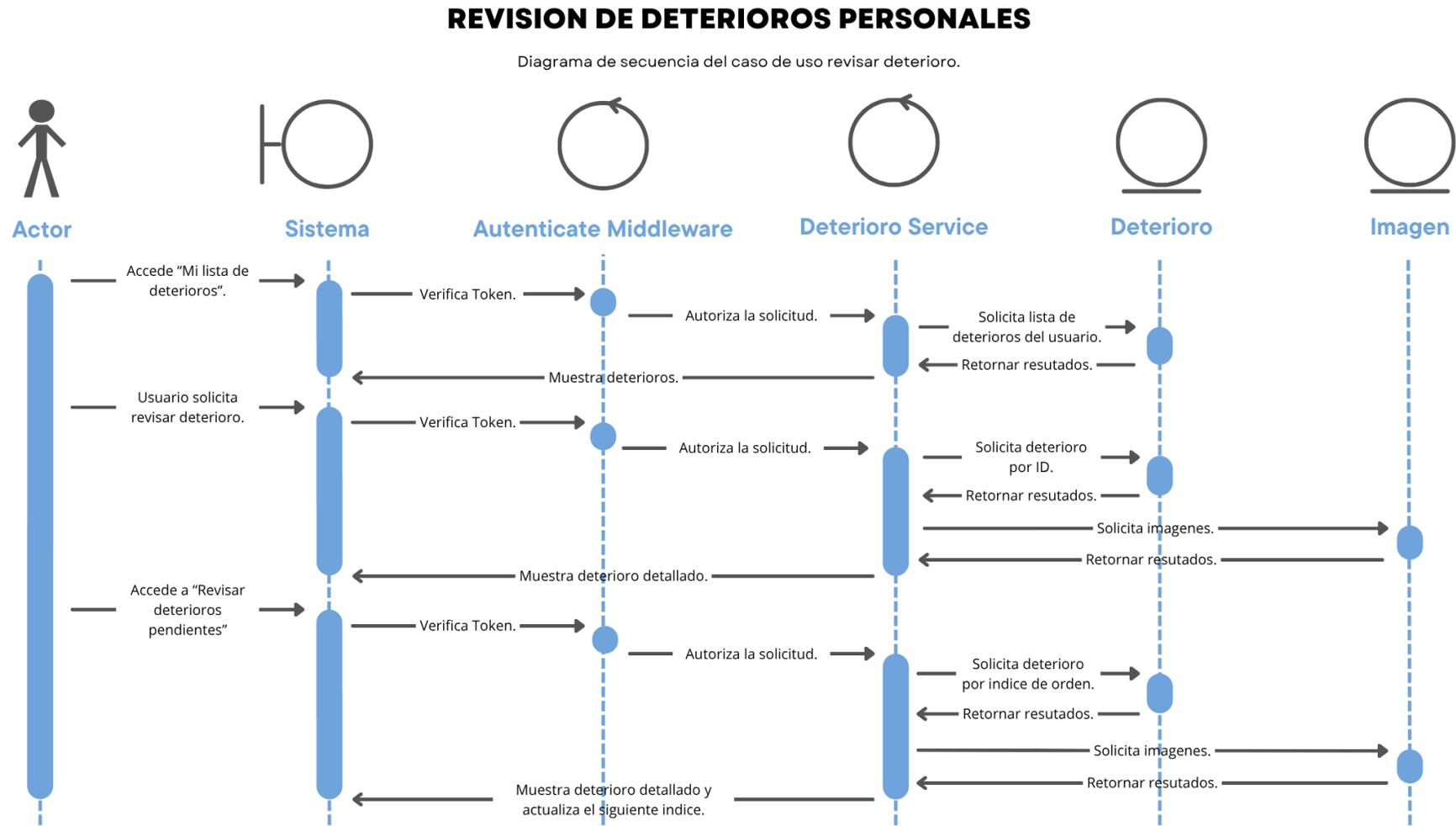
Marcar deterioros en mantenimiento o reparado

La Figura 3.29 muestra la secuencia correspondiente a la herramienta exclusiva para los administradores, mediante la cual se pueden marcar rutas en el mapa y seleccionar los deterioros más cercanos a esta. A través de esta funcionalidad, es posible cambiar el estado de dichos deterioros primero a en mantenimiento y, posteriormente, a reparados, con el fin de identificar y registrar qué rutas ya se encuentran en condiciones óptimas para su tránsito.

Administrar usuarios

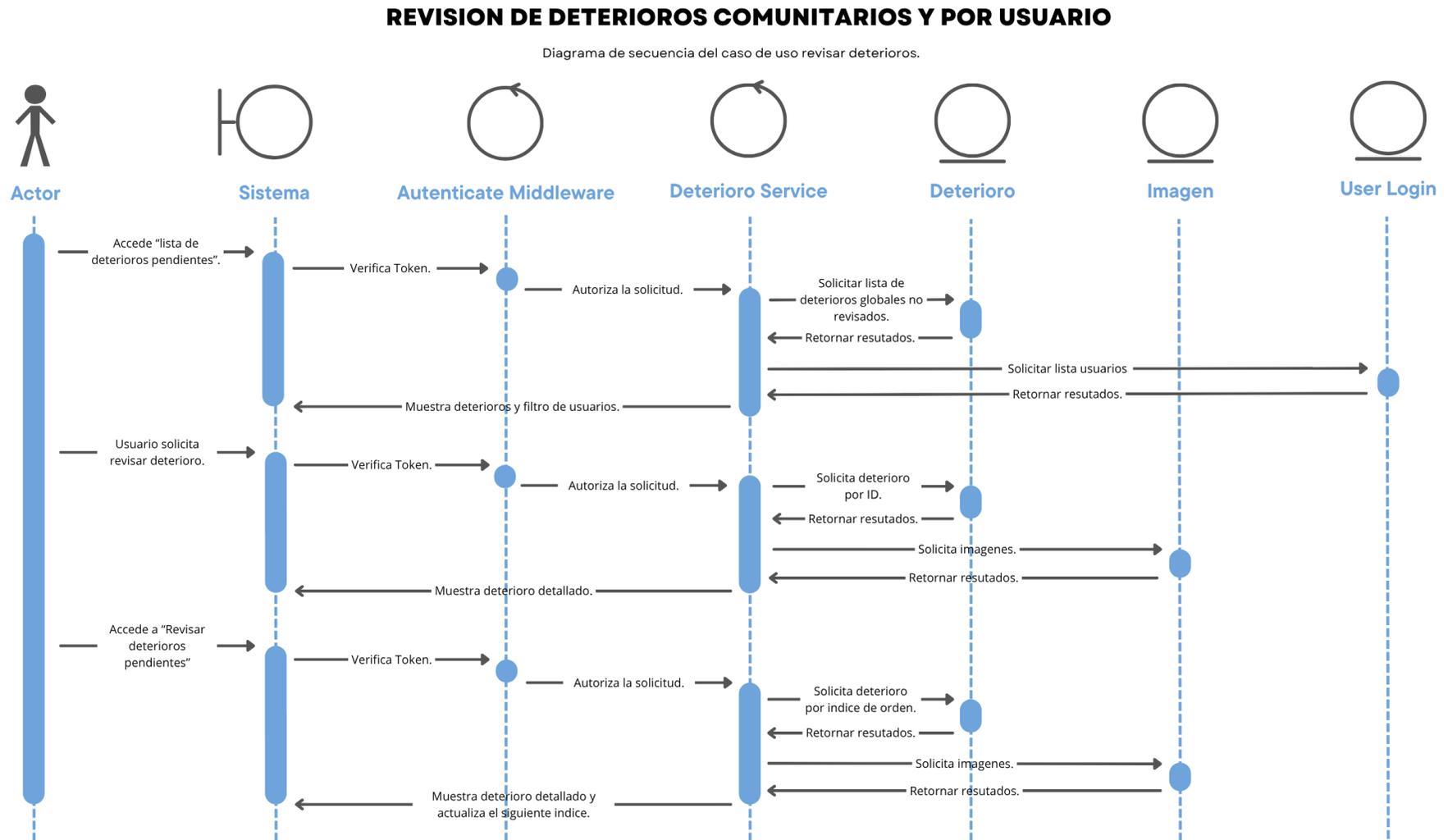
La Figura 3.30 muestra la secuencia correspondiente al módulo de administración de usuarios, una funcionalidad exclusiva de los perfiles con rol de administrador. A través de esta herramienta, es posible realizar tareas de mantenimiento sobre las cuentas registradas en el sistema, como la modificación de datos personales, la actualización de credenciales, la activación o desactivación de accesos, así como la verificación del estado general de cada cuenta. Este módulo garantiza un control adecuado sobre los usuarios activos.

Figura 3.27: Diagrama de secuencia revisión de deterioros personales



Fuente: Elaboración propia.

Figura 3.28: Diagrama de secuencia revisión de deterioros comunitarios y por usuarios

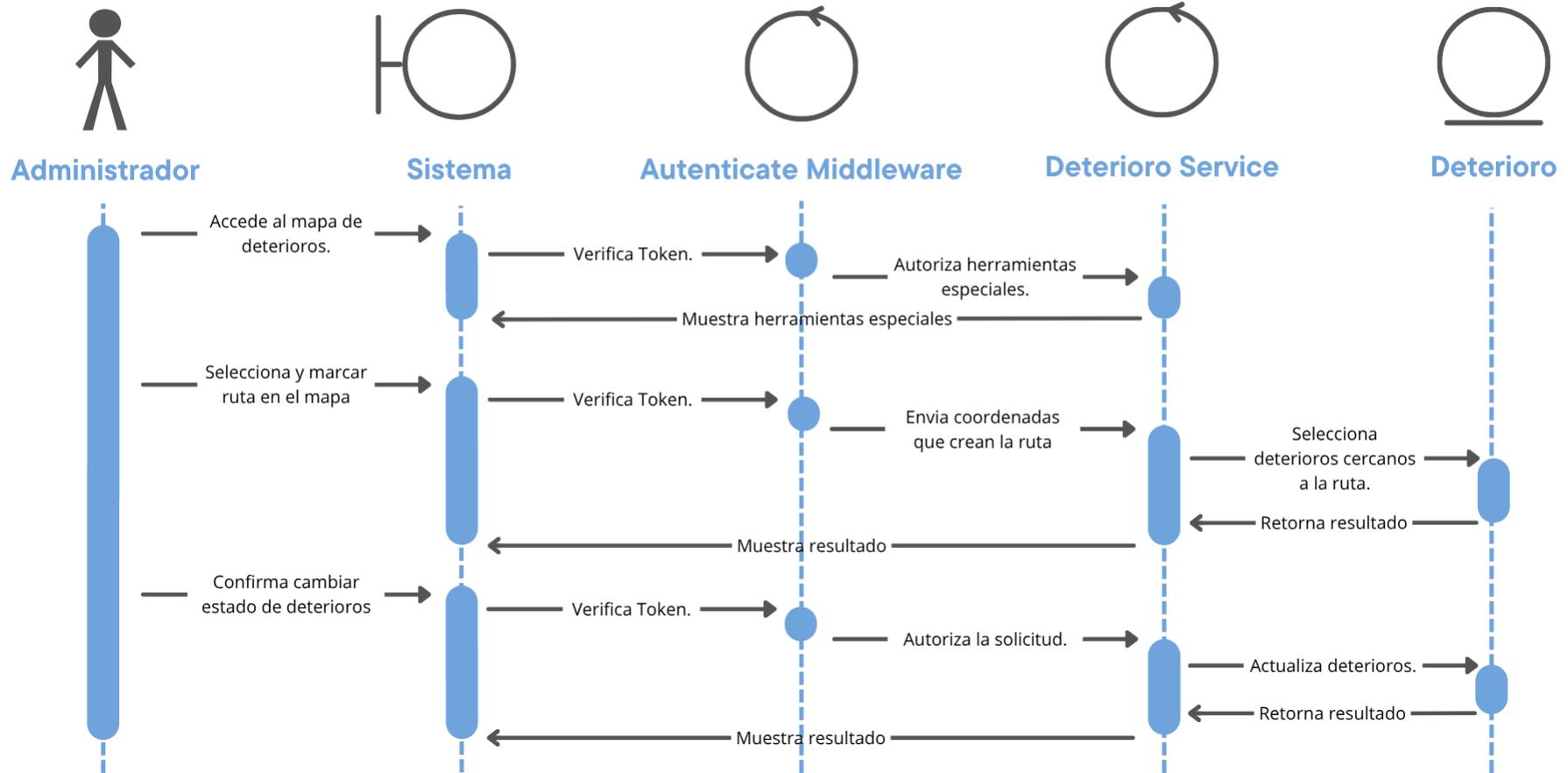


Fuente: Elaboración propia.

Figura 3.29: Diagrama de secuencia marcar deterioros en mantenimiento o reparado

MARCAR DETERIOROS EN MANTENIMIENTO O REPARADOS

Diagrama de secuencia del caso de uso marcar deterioro en mantenimiento o reparado.

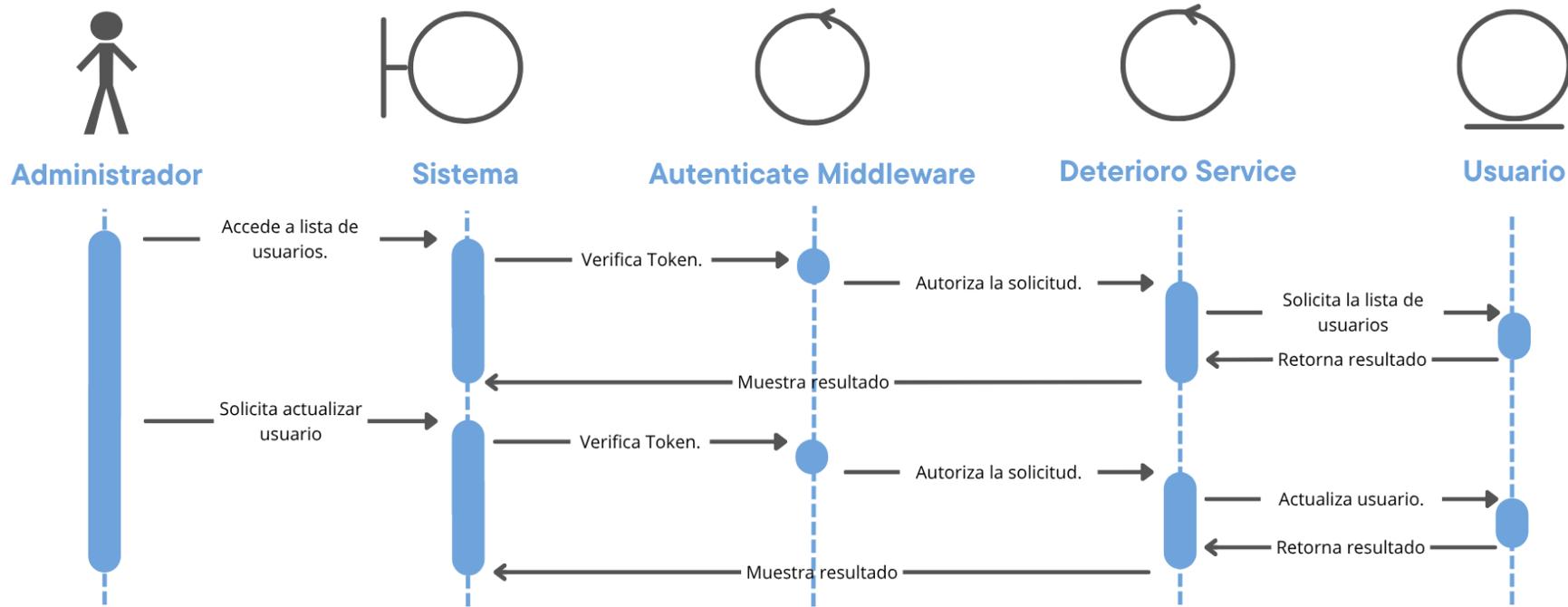


Fuente: Elaboración propia.

Figura 3.30: Diagrama de secuencia administrar usuarios

ADMINISTRAR USUARIOS

Diagrama de secuencia del caso de uso administrar usuarios.



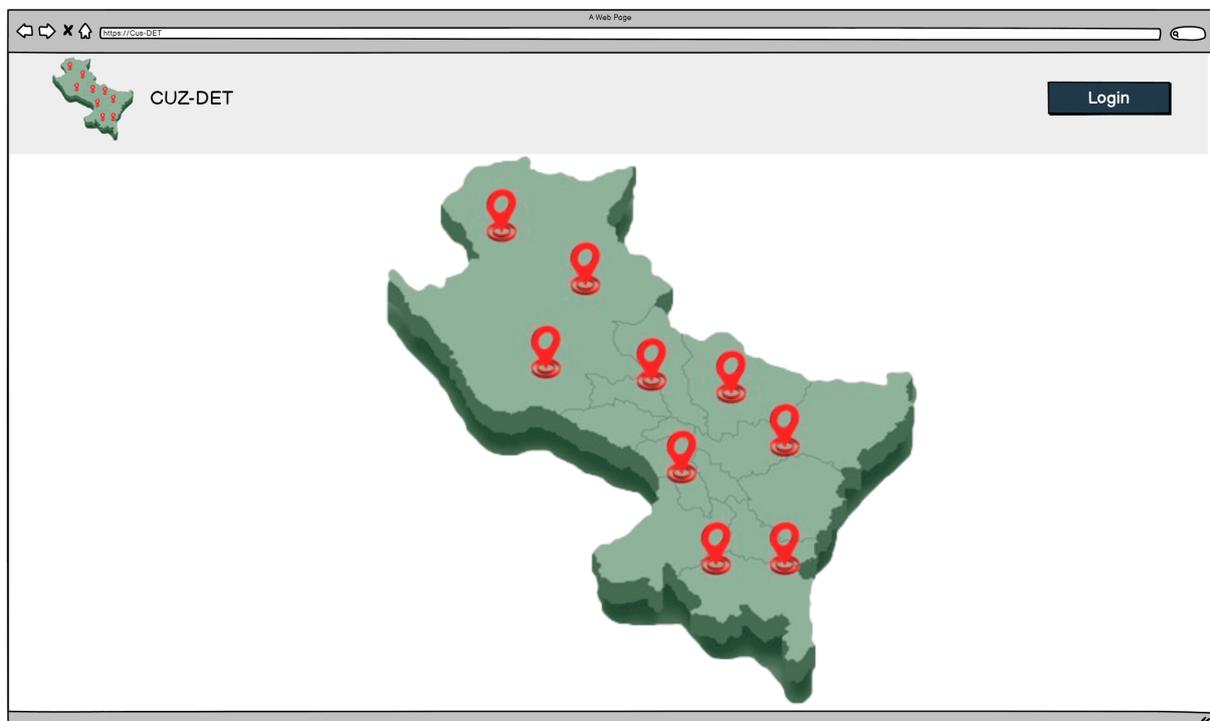
Fuente: Elaboración propia.

3.4. Diseño de Interfaz

Se diseñaron los prototipos para validar los requisitos necesarios del sistema, determinar el espacio de trabajo para cada uno de los usuarios del sistema y también identificar los componentes para React que se debe construir en la interfaz de usuario.

Página de Inicio: El mockup de la página principal mostrada en la Figura 3.31 a la que todo usuario tendrá acceso. Podrá visualizar libremente el mapa del departamento del Cusco y los lugares donde existe afluencia de deterioros en el body del sistema, representados por pequeños marcadores, ya si ellos desearan utilizar filtros avanzados, este se lo impedirá y solicitar crear una cuenta.

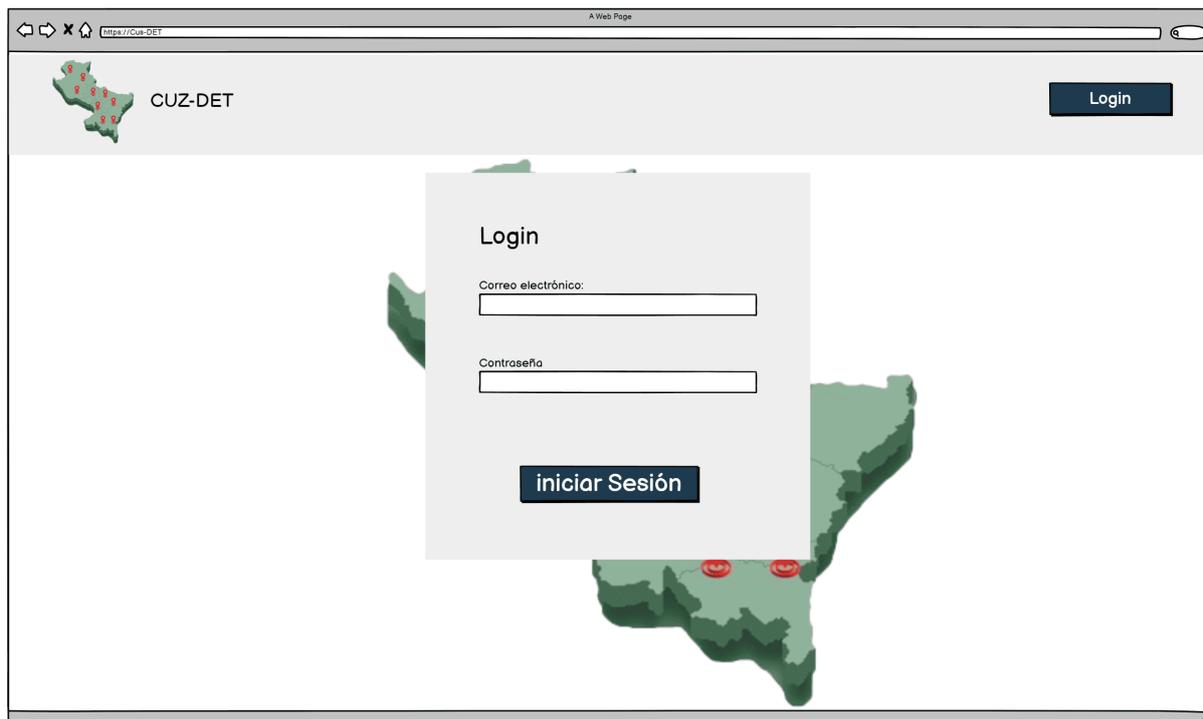
Figura 3.31: Pagina de inicio



Fuente: Elaboración propia.

Página Login: El sistema permitirá que el usuario utilice un login con su nombre de usuario y contraseña, y se le habilitará un token de acceso, junto con los permisos necesarios para acceder a los servicios, como se muestra en el Mockup de la Figura 3.32.

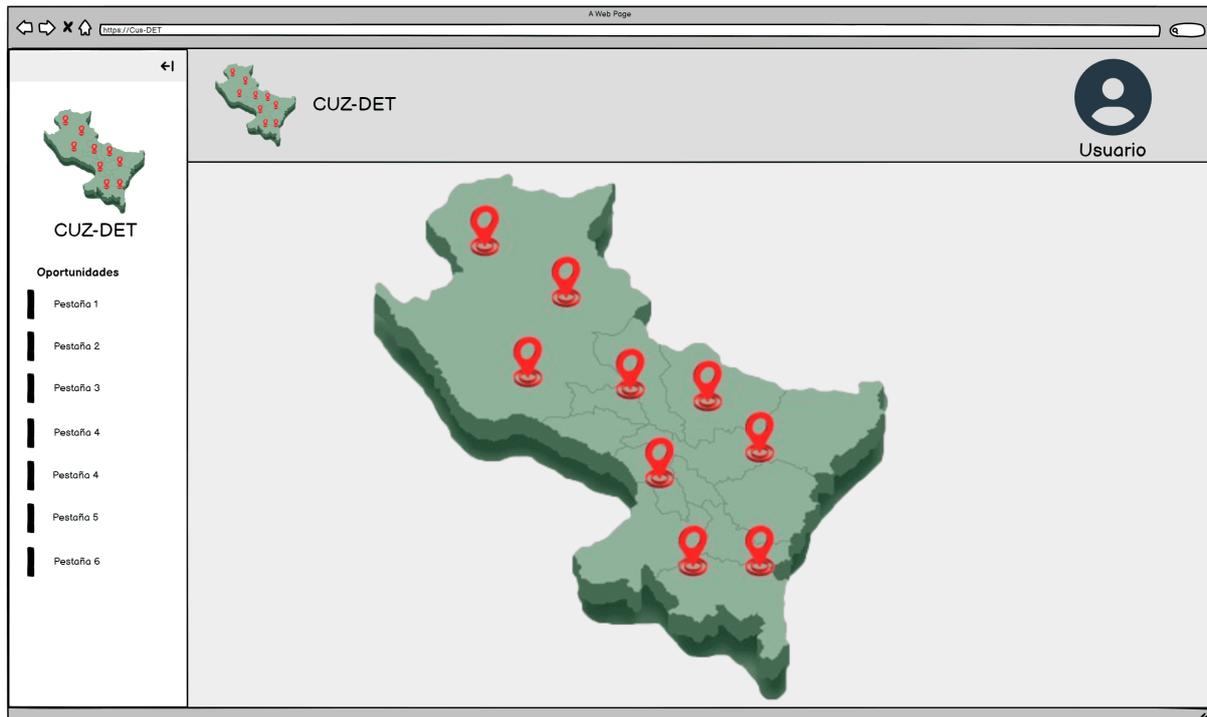
Figura 3.32: Pagina login



Fuente: Elaboración propia.

Página Principal: El usuario podrá visualizar el mapa con la posibilidad de aplicar filtros y observar los detalles de cada marcador donde se han reportado deterioros. Además, se incluye una barra lateral (sidebar) en el lado izquierdo de la pantalla que ofrece más opciones y funcionalidades, como se muestra en la Figura 3.33.

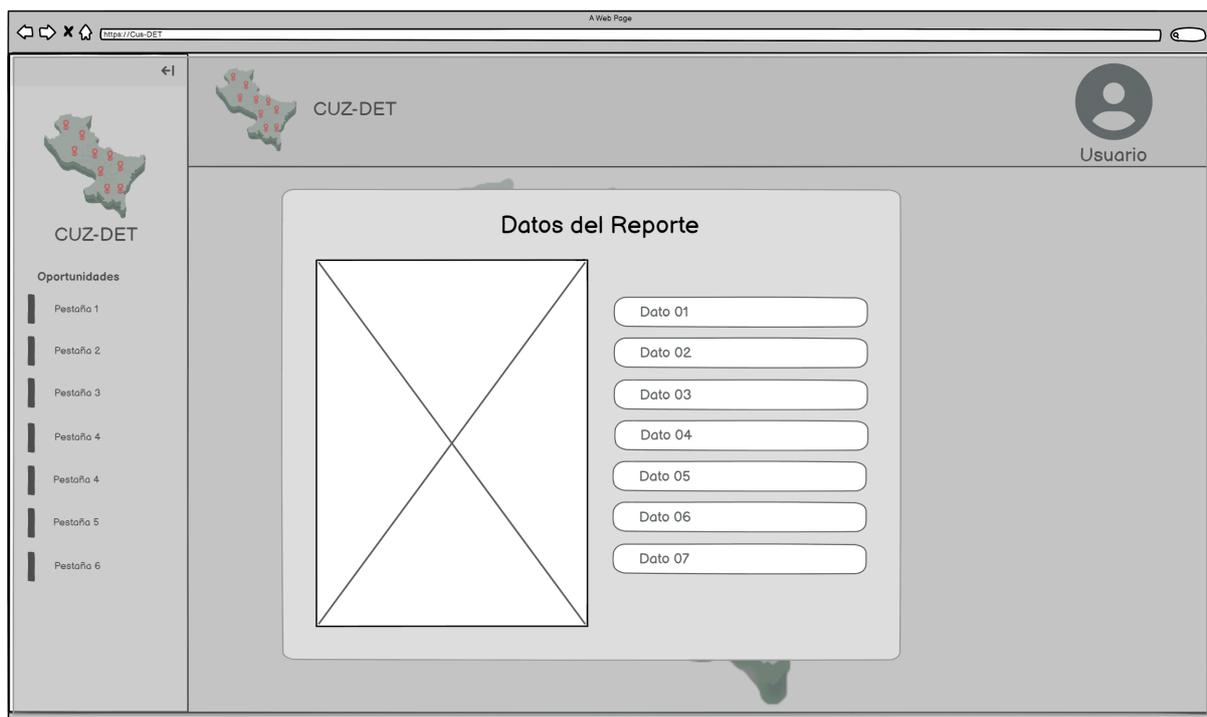
Figura 3.33: Pagina principal



Fuente: Elaboración propia.

Detalles de deterioro: El sistema mostrará los detalles del deterioro una vez que se seleccione un marcador en el mapa, como se muestra en la Figura 3.34.

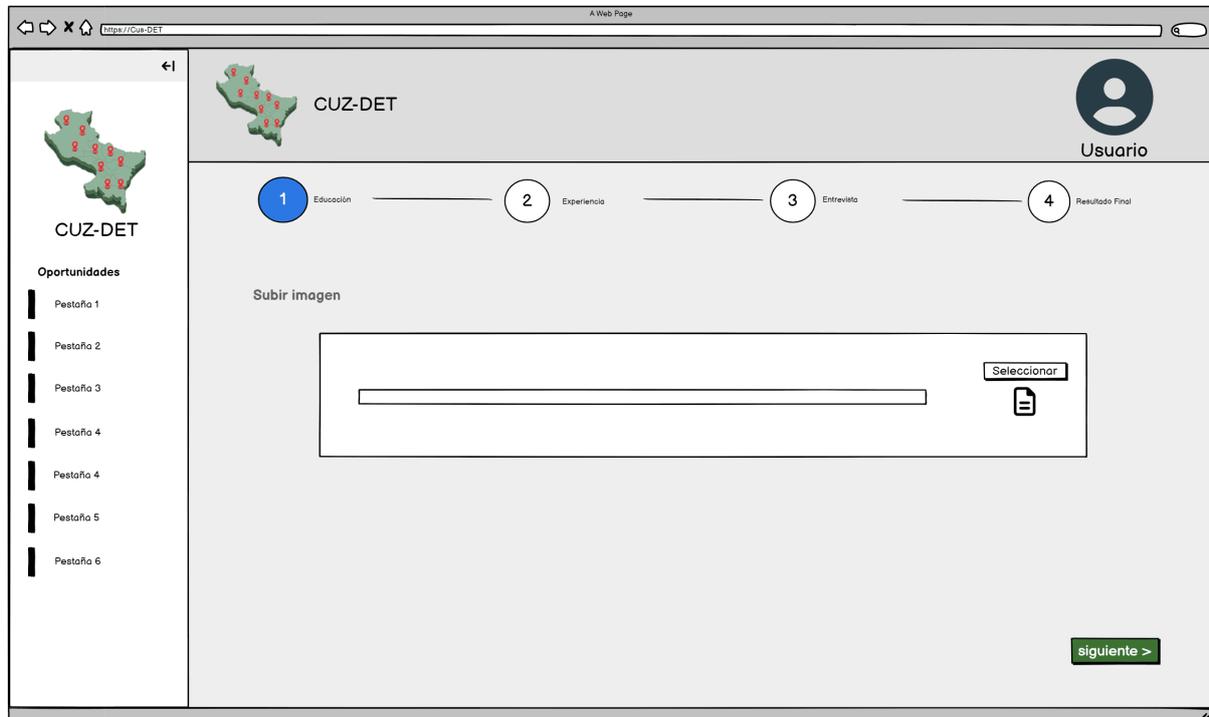
Figura 3.34: Ver deterioros



Fuente: Elaboración propia.

Página Reportar Deterioros: Los pobladores podrán reportar los lugares donde hayan observado deterioros. Cabe destacar que la foto deberá ser tomada con la geolocalización activada. Aquí, en la Figura 3.35 podrán subir su foto.

Figura 3.35: Reportar deterioros paso 1



Fuente: Elaboración propia.

Aquí puedes visualizar tres fotos Figura 3.36: la imagen original, la imagen reducida y, finalmente, aquella que fue analizada por la red neuronal.

Figura 3.36: Reportar deterioros paso 2

Fuente: Elaboración propia.

Posteriormente, en la Figura 3.37, se podrá especificar cuáles *bounding boxes* se consideran relevantes y cuáles deben ser descartadas.

Figura 3.37: Reportar deterioros paso 3

Fuente: Elaboración propia.

Finalmente, los pobladores podrán visualizar los metadatos extraídos de la imagen, como se muestra en la Figura 3.38, para posteriormente confirmar y guardar la fotografía junto con los deterioros detectados.

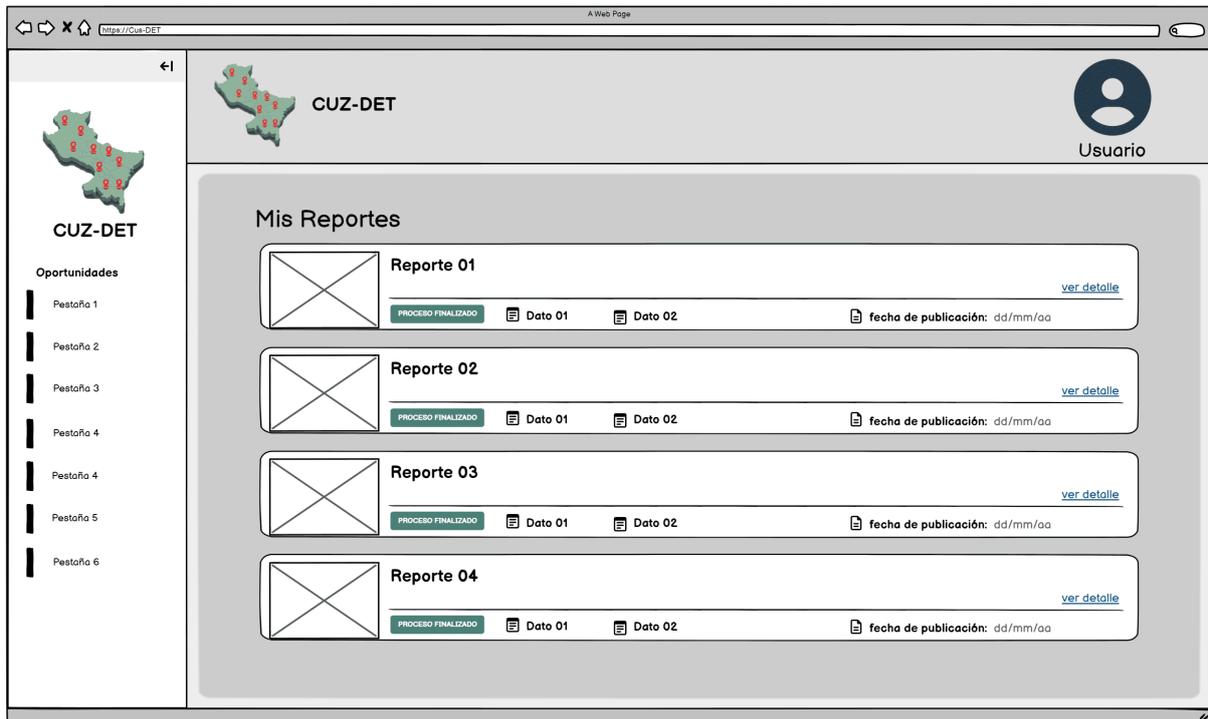
Figura 3.38: Reportar deterioros paso 4

The screenshot displays the 'Reportar deterioros' step 4 in the CUZ-DET application. The interface features a navigation menu on the left with 'Oportunidades' and 'Pestaña 1' through 'Pestaña 6'. The main content area shows a progress bar with four steps: 1 Educación, 2 Experiencia, 3 Entrevista, and 4 Resultado Final (highlighted). Below the progress bar is a 'Metadatos' section with a form containing eight input fields labeled 'Dato 1' through 'Dato 4' on the left and 'Dato 1' through 'dato 4' on the right. A 'siguiente >' button is located at the bottom right of the form.

Fuente: Elaboración propia.

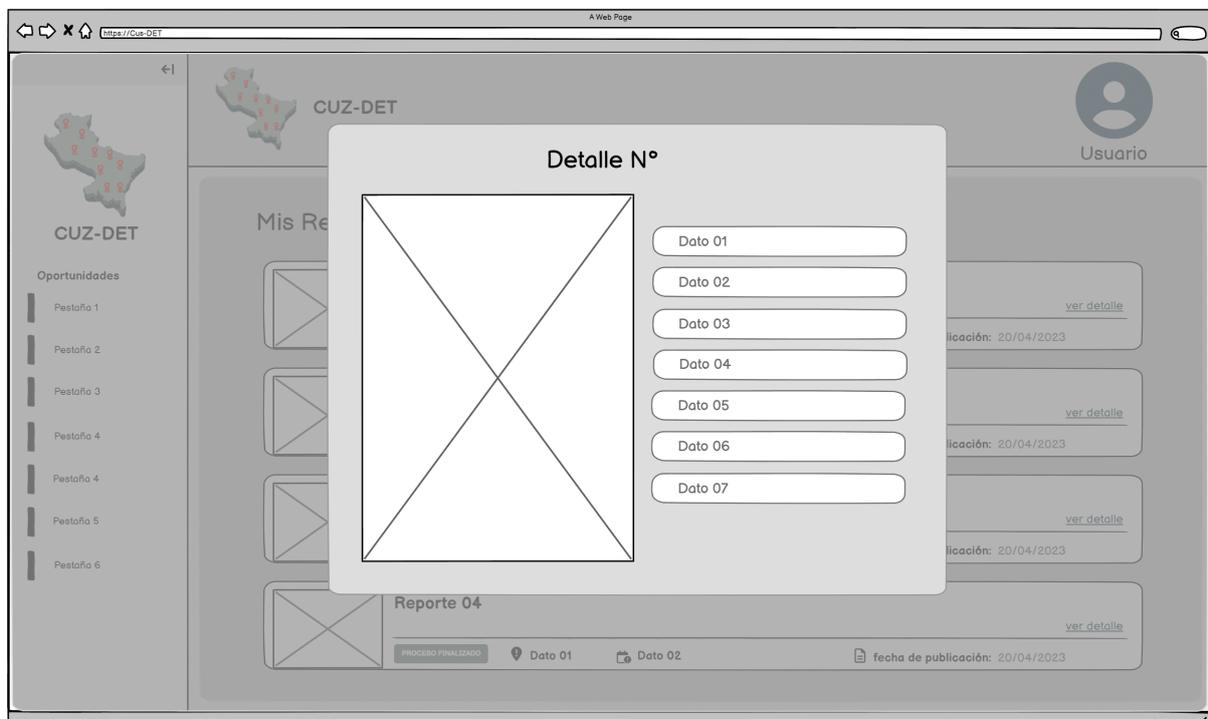
Mi Lista de Reportes de Deterioros: En este apartado, mostrado en la Figura 3.39, el usuario podrá visualizar la lista de los reportes de deterioros que ha realizado, así como acceder a los detalles de cada uno Mostrado en la Figura 3.40.

Figura 3.39: Mis Deterioros reportados



Fuente: Elaboración propia.

Figura 3.40: Mis Deterioros reportados - ver detalle

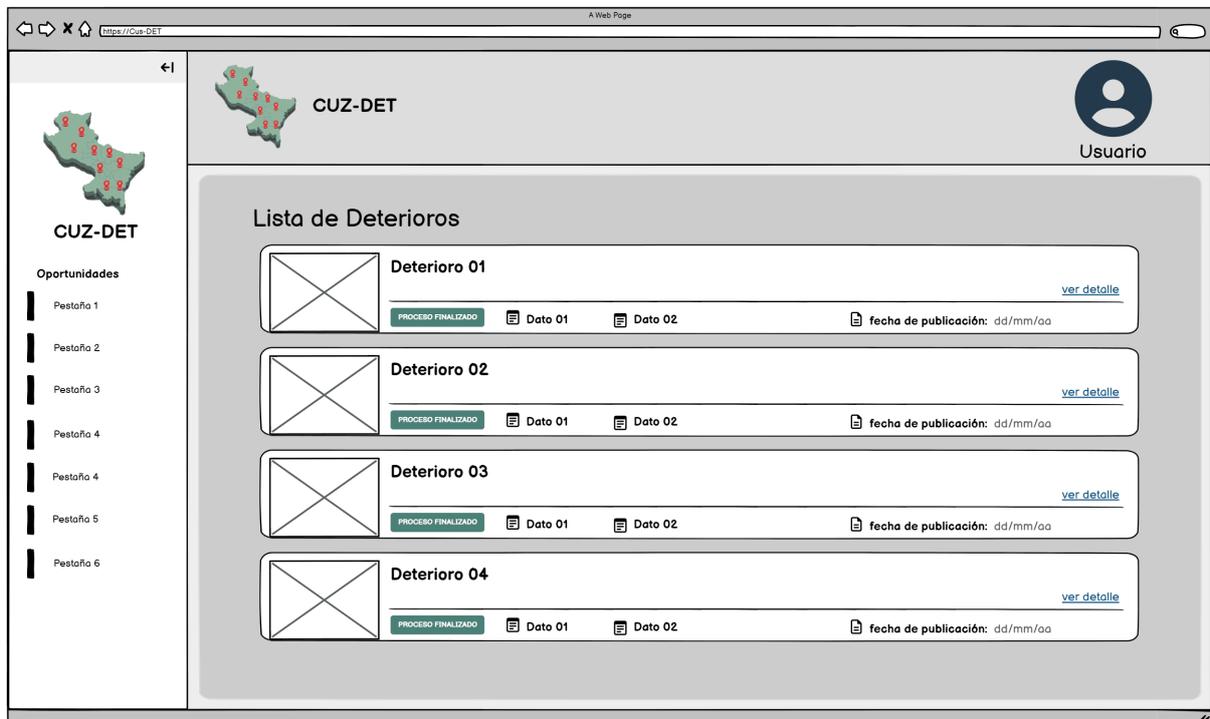


Fuente: Elaboración propia.

Lista completa de deterioros: En este apartado Figura 3.41 el administrador listara todos los deterioros activos, para que pueda editarlos o especificar que algunos ya fueron

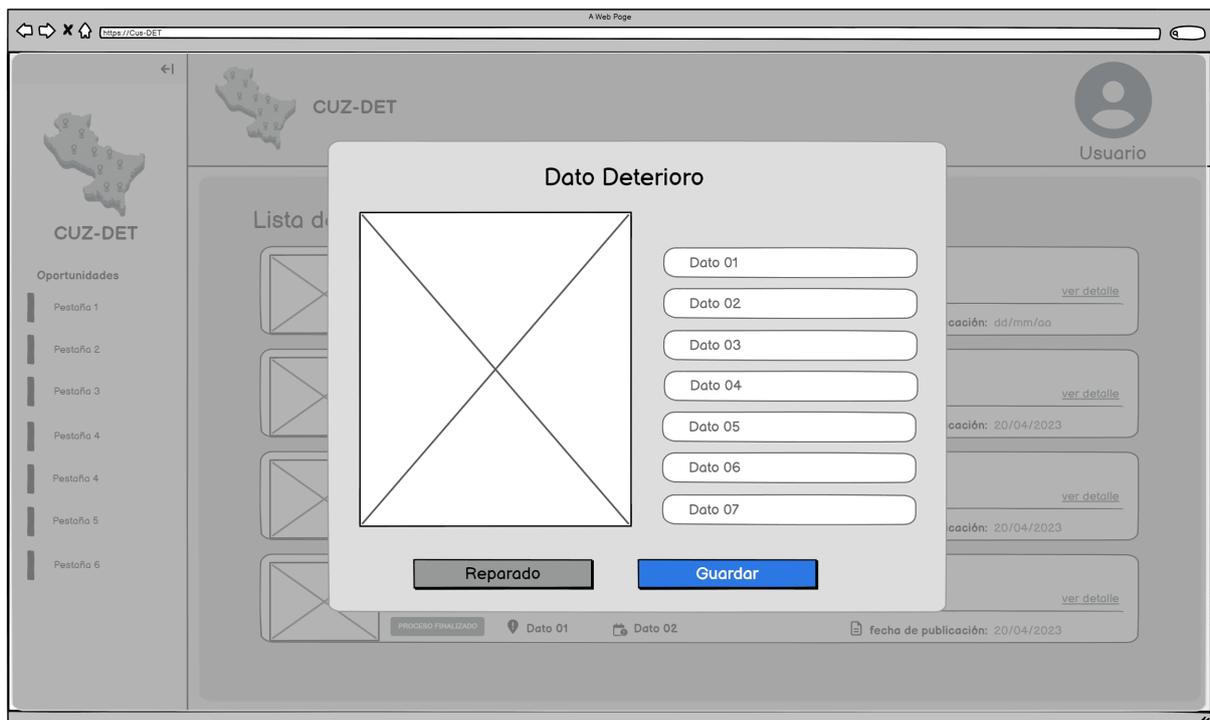
reparados y pasaron por un mantenimiento mostrado en la Figura 3.42.

Figura 3.41: Ver toda la lista de deterioros



Fuente: Elaboración propia.

Figura 3.42: Actualizar deterioro



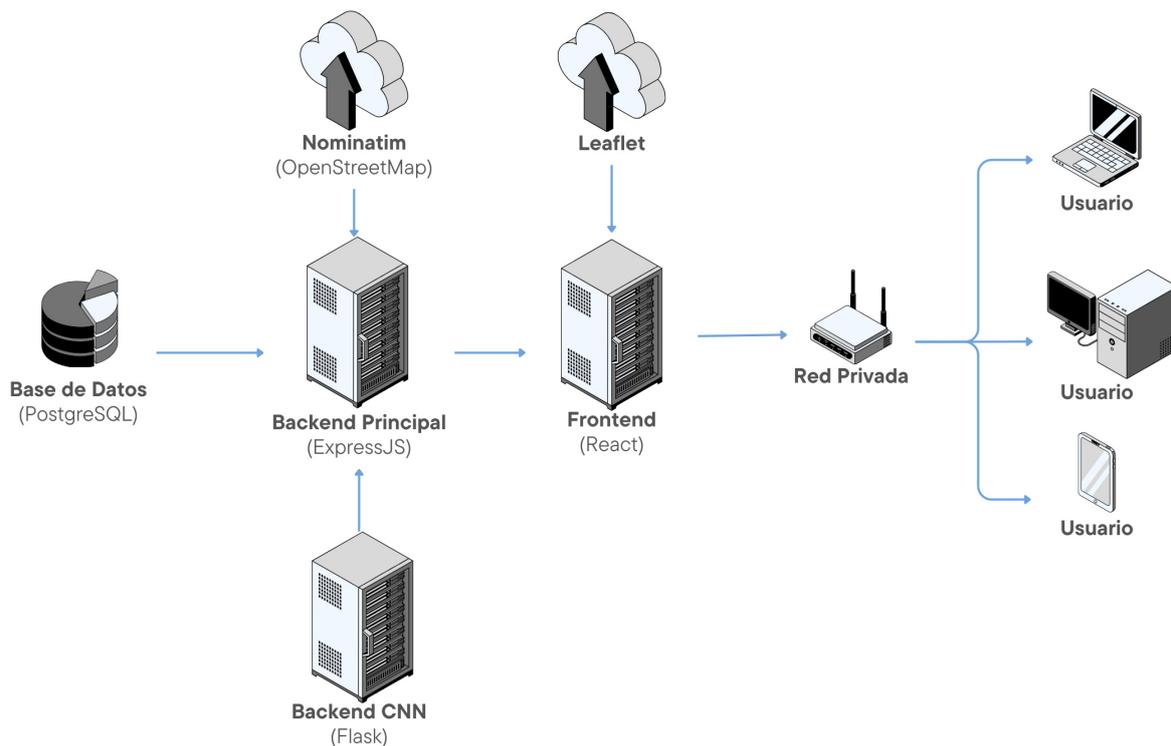
Fuente: Elaboración propia.

3.5. Implementación del Prototipo Web

3.5.1. Arquitectura e infraestructura del Sistema

El sistema fue desarrollado bajo la arquitectura mostrada en la Figura 3.43, donde se evidencia la separación entre la base de datos, los servicios backend, el frontend, las principales APIs utilizadas y su alcance hasta los usuarios finales dentro de una red privada.

Figura 3.43: Arquitectura e infraestructura del Sistema



Fuente: Elaboración propia.

Base de Datos: El sistema utiliza una base de datos relacional en PostgreSQL para almacenar la información de usuarios, deterioros, fotos, ubicaciones, entre otros datos. Esta base de datos se comunica de forma continua con el backend principal, asegurando la persistencia de los datos y la integridad de la información.

Backend Principal (ExpressJS): El backend principal está construido en ExpressJS, que es responsable de manejar las peticiones que provienen del frontend y gestionar la comunicación con la base de datos. Este backend se encarga de procesar las solicitudes del usuario, realizar la lógica de negocio, y coordinar el flujo de información entre el frontend, la base de datos y otros servicios. Es la parte central que conecta todas las piezas del sistema.

Backend Secundario (Flask y Red Neuronal Convolutiva - CNN): El sistema también cuenta con un backend secundario construido en Flask, que aloja una Red Neuronal Convolutiva (CNN). Este servicio se dedica a procesar las imágenes que se cargan al sistema. Cuando una imagen es cargada, se envía a este backend, que analiza la imagen para detectar deterioros y devuelve los bounding boxes, la imagen editada con los marcadores de deterioros, y el número de deterioros encontrados. Este backend es independiente y especializado en el procesamiento de imágenes, lo que permite que el backend principal se enfoque en otras tareas.

Integración con Nominatim API: El backend principal también consume la API Nominatim, la cual es responsable de extraer la dirección, provincia, distrito y otros detalles geográficos a partir de la latitud y longitud extraída de los metadatos de las imágenes cargadas al sistema. De esta manera, se puede asociar la información geográfica precisa a cada foto, lo que facilita la visualización de los deterioros en el mapa y mejora la precisión de los datos.

Frontend (React): El frontend del sistema está construido en React. Esta parte se encarga de cargar todas las interfaces y funcionalidades del sistema, proporcionando una experiencia interactiva y dinámica para el usuario. Los componentes de React gestionan las distintas vistas de la aplicación, permitiendo a los usuarios interactuar con el sistema de manera intuitiva y eficiente. A través de React, el frontend se conecta con el backend y muestra la información relevante de manera visual.

Leaflet (Mapa de Deterioros): El mapa de la aplicación se visualiza mediante la biblioteca Leaflet, que se integra en el frontend. Leaflet es una herramienta poderosa para mostrar mapas interactivos en la web, y en este caso, se utiliza para mostrar la ubicación de los deterioros en el mapa. Los deterioros se representan con chinchetas o marcadores en el mapa, que permiten a los usuarios ver la distribución geográfica de los deterioros en el área cubierta. Leaflet también permite realizar interacciones como el zoom, desplazamiento y visualización de detalles al hacer clic en los marcadores.

Seguridad en la Comunicación (HTTPS y Certificados): La comunicación entre el frontend y los usuarios se realiza a través de HTTPS, lo que garantiza la seguridad de los datos transmitidos. Esta conexión segura se utiliza tanto para el acceso de los usuarios a la aplicación como para la visualización de los deterioros en el mapa y la gestión de los datos. Además, se utiliza un certificado SSL que facilita el acceso a los usuarios que se encuentran dentro de una red privada, asegurando que las solicitudes y respuestas estén cifradas y protegidas.

De igual manera, la comunicación entre el backend principal y el frontend también se realiza utilizando HTTPS, garantizando que todos los datos, como la información del usuario, los deterioros y las imágenes, se transmitan de forma segura y sin comprometer la privacidad.

3.5.2. Desarrollo de la Base de Datos

La base de datos del sistema fue desarrollada utilizando PostgreSQL, un sistema de gestión de bases de datos relacional (RDBMS) de código abierto ampliamente reconocido por su robustez, escalabilidad y cumplimiento de estándares SQL. Su diseño se basó en los atributos identificados a partir del análisis de las historias de usuario, casos de uso, diagramas de flujo y diagramas de secuencia, lo cual permitió construir el diagrama de clases en la Figura 3.44.

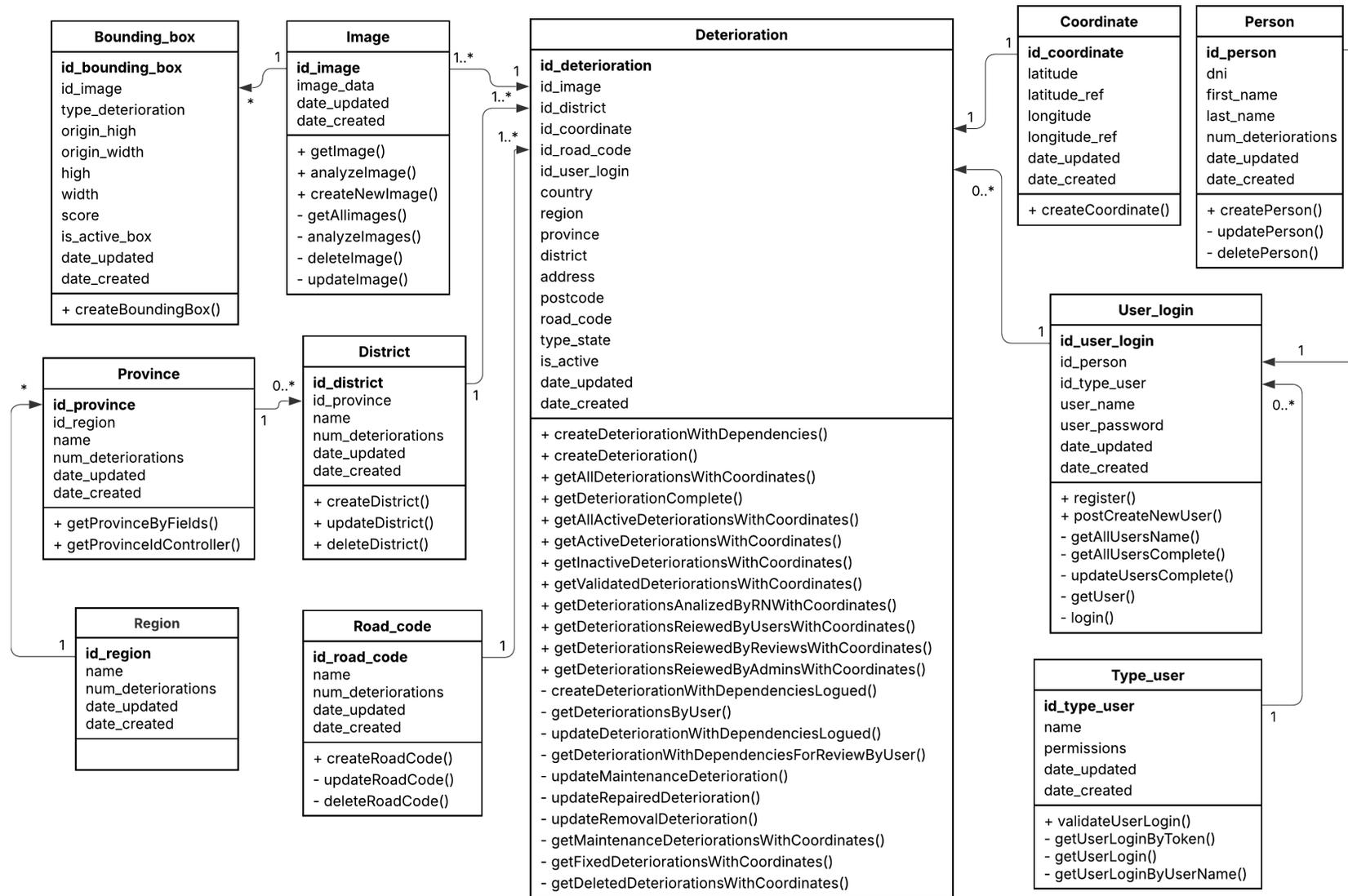
Si bien no se realizó una comparación exhaustiva con otros motores como MySQL o SQLite, se optó por PostgreSQL debido a su alta fiabilidad, su activa comunidad de desarrollo y su compatibilidad con herramientas modernas empleadas en aplicaciones web y sistemas de análisis de datos. Esta elección facilitó una integración eficiente con los componentes del backend y permitió una gestión óptima del modelo relacional diseñado.

El diseño de la base de datos se estructuró bajo un enfoque relacional, organizando los datos en tablas interconectadas mediante claves primarias y foráneas. Esta estructura garantiza la integridad referencial y facilita la ejecución de consultas complejas mediante uniones (JOINS) entre entidades relacionadas. Cada tabla fue diseñada para representar una entidad específica dentro del dominio del sistema, como usuarios, deterioros, imágenes, coordenadas, entre otros, permitiendo una representación fiel del modelo de información y asegurando su escalabilidad ante futuras ampliaciones.

Cabe resaltar que el sistema no almacena las imágenes directamente como datos binarios en la base de datos, sino que estas se guardan de forma física en el servidor, organizadas en carpetas estructuradas por usuario y en directorios temporales para facilitar su procesamiento. La base de datos únicamente conserva la ruta de acceso a dichas imágenes a través de la tabla `image`, lo cual reduce significativamente el tamaño del almacenamiento requerido y mejora el rendimiento general del sistema, especialmente en operaciones de carga y respaldo.

En conjunto, la base de datos funciona como el núcleo del sistema, permitiendo almacenar, consultar y relacionar eficientemente toda la información relevante para la gestión de usuarios, deterioros, procesos de revisión, geolocalización y evidencia visual.

Figura 3.44: Diagrama de Clases del sistema



Fuente: Elaboración propia.

Modelo de Localización Geográfica: La base de datos implementa una estructura jerárquica basada en entidades geográficas: región, provincia y distrito. Esta jerarquía permite asociar cada deterioro a una ubicación administrativa específica, facilitando la elaboración de reportes estadísticos por nivel territorial, así como la implementación de filtros geográficos dentro del sistema. Esta modelación no solo optimiza el análisis espacial de la información, sino que también sienta las bases para futuras integraciones con sistemas geográficos externos o módulos de planificación territorial.

Gestión de Usuarios: El modelo distingue entre los datos personales (almacenados en la tabla person) y las credenciales de acceso (tabla user_login), lo cual permite una mejor separación de responsabilidades, mayor seguridad en el manejo de contraseñas, y flexibilidad para asociar múltiples tipos de usuarios con distintos niveles de permisos definidos en la tabla type_user. Esta arquitectura también facilita la auditoría de acciones y el seguimiento de contribuciones por parte de los usuarios.

Procesamiento de Imágenes: Las imágenes capturadas o cargadas por los usuarios no se almacenan como objetos binarios en la base de datos. En su lugar, se guardan como rutas de archivo asociadas mediante la tabla image. A cada imagen se le pueden detectar múltiples áreas de interés denominadas cajas delimitadoras (bounding boxes), registradas en la tabla bounding_box, las cuales almacenan información sobre el tipo de deterioro detectado, su puntuación de confianza, y las dimensiones relativas dentro de la imagen. Esto permite una representación precisa y reutilizable de los datos detectados mediante modelos de visión por computadora.

Registro de Deterioros: La tabla deterioration representa el componente central del sistema, ya que contiene los registros individuales de cada deterioro identificado o reportado. Cada entrada está vinculada con una imagen, un usuario, una ubicación geográfica (coordenadas GPS y distrito), así como metadatos obtenidos por geocodificación como la dirección y el código vial. Además, permite almacenar el estado actual del deterioro (activo o no) y el tipo de estado, facilitando procesos de seguimiento, evaluación y priorización.

Información Geoespacial: La tabla coordinate registra los valores de latitud y longitud extraídos automáticamente de los metadatos EXIF de las imágenes, cuando están disponibles. Esta información, junto con los nombres de región, provincia y distrito, permite visualizar los deterioros sobre un mapa interactivo en el frontend, brindando una herramienta visual poderosa para la supervisión territorial y la toma de decisiones.

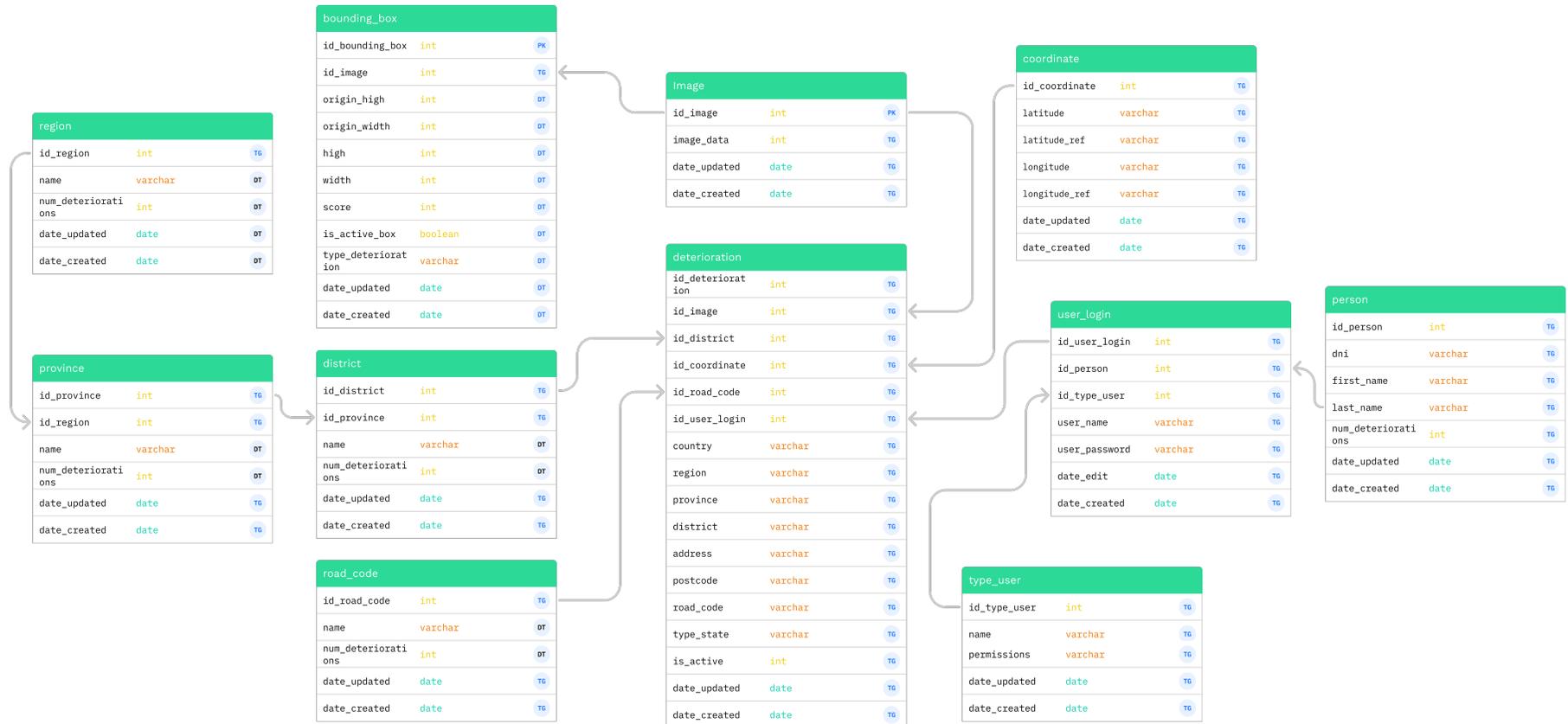
Datos Precargados: Se incluyeron registros iniciales como la región Cusco y sus provincias, lo cual facilita el despliegue inicial del sistema y permite operar con datos reales desde el primer uso.

El diseño de clases sirvió como base para la posterior implementación de la base de datos relacional. A partir de este modelo orientado a objetos, se construyó la estructura

definitiva del esquema relacional, adaptando los atributos a tipos de datos compatibles con PostgreSQL y estableciendo las relaciones mediante claves primarias y foráneas. Esta transición permitió mantener la correspondencia semántica entre el modelo lógico del sistema y su representación física en la base de datos.

La Figura 3.45 presenta el diagrama relacional implementado, en el cual se visualizan las tablas principales del sistema, sus campos y las relaciones entre ellas. Este diagrama fue esencial para guiar la codificación de las migraciones y sentar las bases del modelo de persistencia utilizado por la aplicación.

Figura 3.45: Base de Datos del sistema



Fuente: Elaboración propia.

3.5.3. Desarrollo del Backend

El backend del prototipo fue implementado utilizando el framework Express.js, una solución basada en Node.js que permite estructurar aplicaciones web de manera modular, eficiente y escalable. Esta elección respondió a la necesidad de construir una arquitectura ligera y flexible, adecuada para las funciones críticas del sistema propuesto, como el procesamiento de imágenes, la gestión de usuarios y la interoperabilidad con el frontend.

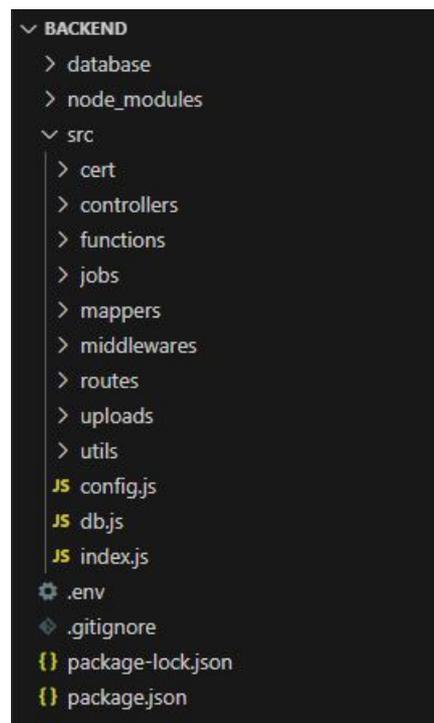
Uno de los aspectos fundamentales de este desarrollo fue la implementación de una conexión segura entre el backend, el frontend y los clientes mediante el protocolo HTTPS, lo que garantiza la integridad y confidencialidad de los datos transmitidos. Esta medida es especialmente relevante en el contexto del sistema, donde se gestionan imágenes, credenciales de acceso y datos sensibles derivados del análisis de deterioros viales. El uso de HTTPS refuerza la protección contra ataques de interceptación y asegura una comunicación confiable entre los componentes de la plataforma.

- `bcrypt`: Librería que permite encriptar las contraseñas que los usuarios desean establecer en sus cuentas, asegurando que los datos sensibles estén protegidos.
- `exif`: Permite acceder a los metadatos de las fotografías subidas por los usuarios, como la latitud, longitud, fecha de creación, entre otros, lo que facilita la georreferenciación de los deterioros en las vías.
- `jsonwebtoken`: Utilizada para crear tokens que gestionan el tiempo de vida de las sesiones de los usuarios, garantizando un acceso seguro y limitado a los recursos del sistema.
- `multer`: Middleware que permite la carga de archivos al servidor, facilitando la recepción y almacenamiento de las imágenes de las vías para su posterior procesamiento.
- `pg`: Cliente de Node.js para interactuar con bases de datos PostgreSQL, utilizado para almacenar y consultar la información relacionada con los deterioros, usuarios y demás datos del sistema.
- `sharp`: Librería que permite redimensionar el tamaño de las imágenes, optimizando su procesamiento y acelerando el uso de la API de la red neuronal para la detección de deterioros.
- `cors`: Middleware que permite habilitar las solicitudes de origen cruzado (CORS), asegurando la interoperabilidad entre el frontend y el backend en diferentes dominios o puertos.
- `unrar`: Librería para manejar archivos RAR, útil en caso de que sea necesario procesar archivos comprimidos.

Estructura del backend

En la Figura 3.46 se presenta la estructura principal del backend del prototipo desarrollado. Esta estructura está organizada en diversas carpetas que contienen los distintos componentes y archivos necesarios para el correcto funcionamiento del sistema. Dentro de estas carpetas, el archivo `index` destaca como el punto de entrada principal para la ejecución del backend. Este archivo es responsable de inicializar y levantar el servidor, así como de exponer públicamente las rutas mediante las cuales se reciben y atienden las solicitudes de los usuarios o clientes.

Figura 3.46: Estructura del Backend



Fuente: Elaboración propia.

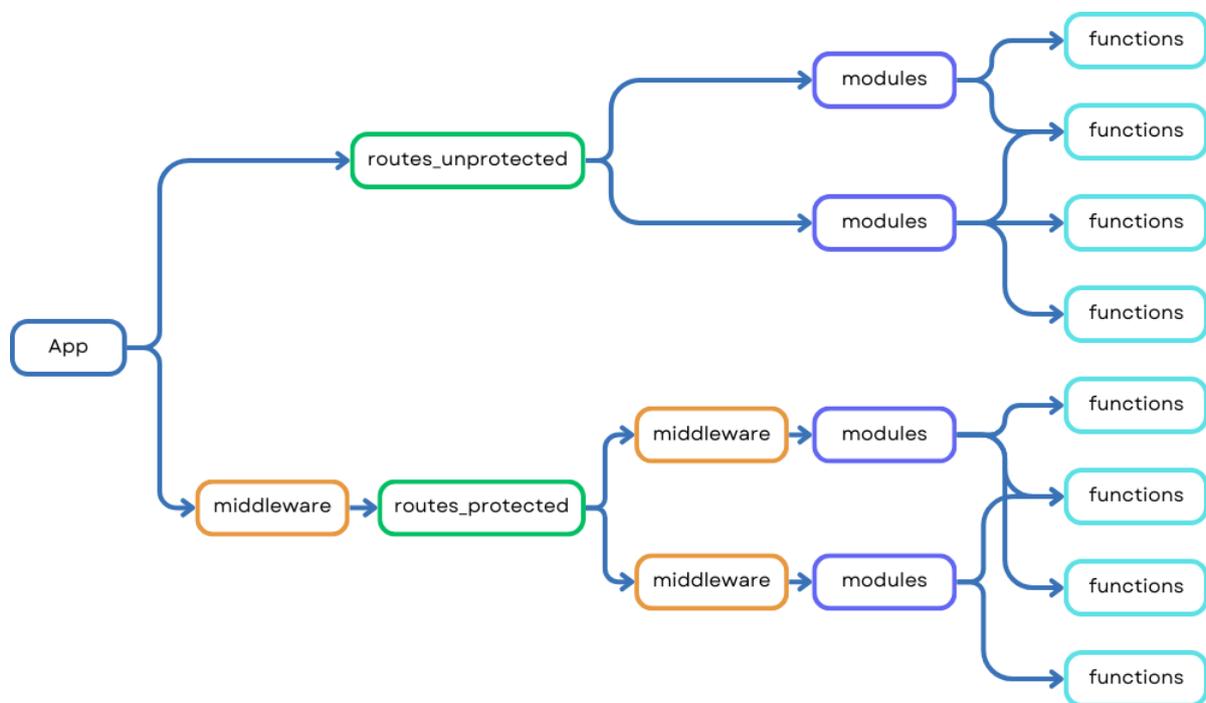
La Figura 3.47 muestra en detalle la organización de las rutas que conforman la API del backend. Dichas rutas se clasifican en dos categorías principales: rutas protegidas y rutas no protegidas. Las rutas no protegidas permiten el acceso libre a cualquier usuario, dado que no requieren la presentación de un token de autenticación, lo que facilita el acceso a funcionalidades básicas o públicas del sistema. En cambio, las rutas protegidas exigen que el usuario presente un token válido emitido tras un proceso de autenticación, además de que su perfil mínimo debe ser de tipo usuario registrado en el sistema. Esta diferenciación garantiza un control de acceso adecuado, protegiendo información sensible y funciones críticas del sistema frente a usuarios no autorizados.

Los módulos representan las funcionalidades principales encargadas de procesar las solicitudes recibidas a través de las rutas. Cada módulo contiene la lógica necesaria para ejecutar las operaciones específicas solicitadas por el cliente. Estos módulos están directamente vinculados a las rutas correspondientes, de modo que cada ruta invoca las funciones del módulo asociado para responder a las peticiones. En particular, para

las rutas protegidas, se implementan middlewares que actúan como filtros o capas de validación adicionales. Estos middlewares verifican los permisos y roles asignados a los usuarios, asegurando que sólo aquellos con el nivel adecuado puedan acceder a ciertas rutas y funcionalidades, estableciendo así una política de seguridad y control de acceso efectiva.

Finalmente, dentro de los módulos se encuentran funciones más pequeñas y especializadas que realizan tareas repetitivas y específicas. Estas funciones incluyen la ejecución de consultas (queries) a la base de datos, que permiten obtener, insertar o modificar información almacenada, así como operaciones auxiliares de conversión, validación o manipulación de datos que optimizan el procesamiento y la integridad de la información manejada en el backend. Esta arquitectura modular y jerárquica facilita el mantenimiento, escalabilidad y la evolución del sistema, al mismo tiempo que asegura un flujo de trabajo organizado y eficiente en el manejo de las solicitudes y respuestas del prototipo.

Figura 3.47: Arquitectura del Backend



Fuente: Elaboración propia.

3.5.4. Desarrollo del Frontend

El frontend del prototipo fue desarrollado utilizando el framework React, con soporte de TypeScript, lo que permite un desarrollo tipado, estructurado y más robusto frente a errores. La aplicación cliente actúa como intermediario visual e interactivo entre los usuarios finales y los servicios ofrecidos por el backend, haciendo uso del protocolo HTTPS para asegurar la transmisión de datos.

Cabe destacar que, en su estado actual, el prototipo está diseñado para funcionar exclusivamente dentro de una red privada local, lo cual garantiza un entorno controlado

para pruebas, evita accesos externos no autorizados.

Para la construcción de la interfaz y la gestión de la experiencia de usuario, se integraron las siguientes herramientas y librerías:

- **react-router-dom**: Librería encargada de gestionar la navegación entre las distintas páginas de la aplicación mediante rutas declarativas. Facilita la creación de interfaces de una sola página (SPA), mejorando el rendimiento y la fluidez de la navegación.
- **PrimeReact**: Conjunto de componentes UI modernos, reutilizables y responsivos, que permiten acelerar el desarrollo de interfaces visualmente consistentes, accesibles y funcionales, como formularios, tablas, botones, diálogos, entre otros.
- **Leaflet**: Librería especializada en la manipulación de mapas interactivos. En el contexto de este proyecto, permite representar los deterioros georreferenciados mediante marcadores, integrando coordenadas obtenidas a partir de las imágenes cargadas al sistema.

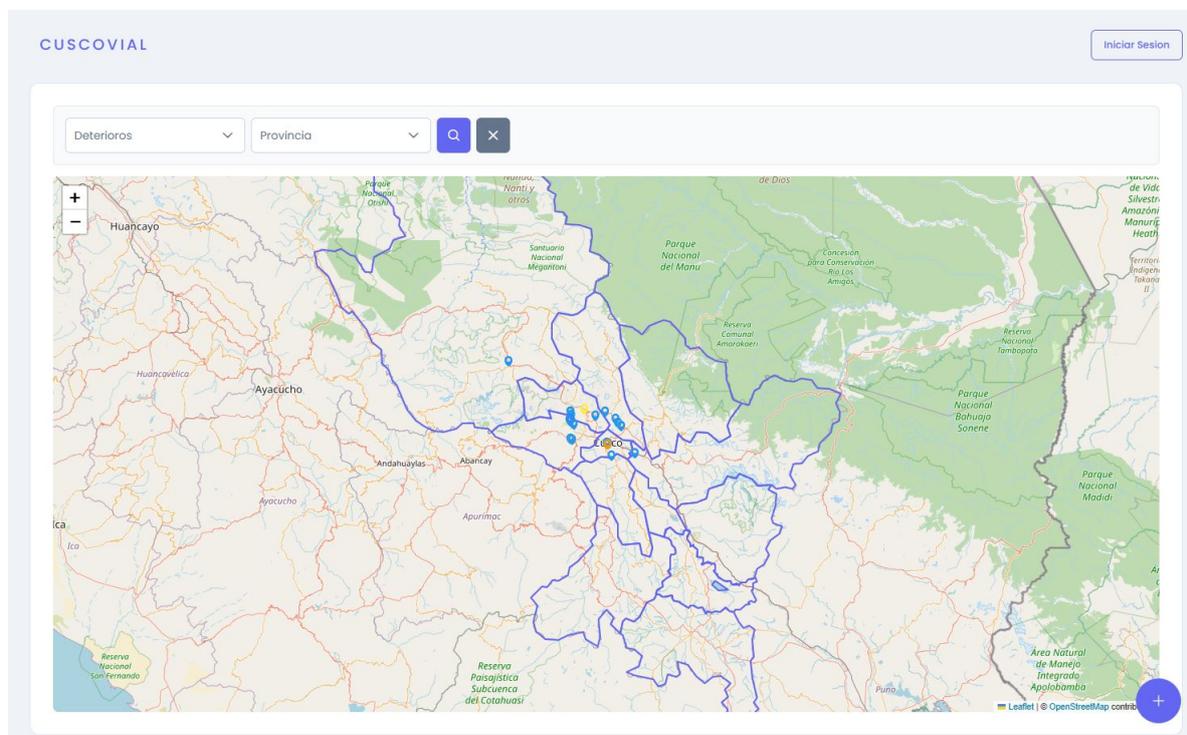
Página principal

La Figura 3.48 presenta la vista principal del frontend del sistema, en la cual se identifican cuatro componentes fundamentales que conforman la interfaz de usuario. Cada uno de estos elementos cumple funciones específicas dentro del flujo de interacción del prototipo:

1. **Barra superior (Topbar)**: Se ubica en la parte superior de la interfaz y contiene el logotipo del sistema, así como un botón destinado al inicio de sesión. Este componente permanece fijo durante la navegación, proporcionando acceso constante a funcionalidades clave relacionadas con la autenticación del usuario.
2. **Caja de herramientas**: Este componente varía según el perfil del usuario. En el caso mostrado, correspondiente a un usuario con perfil de poblador, se limita a permitir el uso de filtros por tipo de deterioro y por provincias. Ambos filtros pueden ser combinados para refinar la visualización de los datos en el mapa, mejorando así la experiencia de búsqueda y exploración del usuario.
3. **Mapa interactivo**: Ocupa el área central de la interfaz y está construido mediante la librería Leaflet, la cual permite la visualización y manipulación de mapas interactivos. Sobre el mapa se representan marcadores correspondientes a los deterioros previamente identificados por el sistema. Una funcionalidad adicional es que, al hacer clic sobre un marcador, se despliega un sidebar lateral con información detallada del deterioro correspondiente, como su ubicación, características y fecha de detección.
4. **Botón flotante de accesos rápidos**: Este elemento se ubica en la esquina inferior de la interfaz y permite desplegar un conjunto de botones secundarios que

brindan acceso rápido a otras funcionalidades del sistema. Entre ellas se incluyen: visualización de la leyenda del mapa, carga individual de imágenes, y carga masiva de imágenes mediante archivos comprimidos.

Figura 3.48: Frontend pagina de inicio



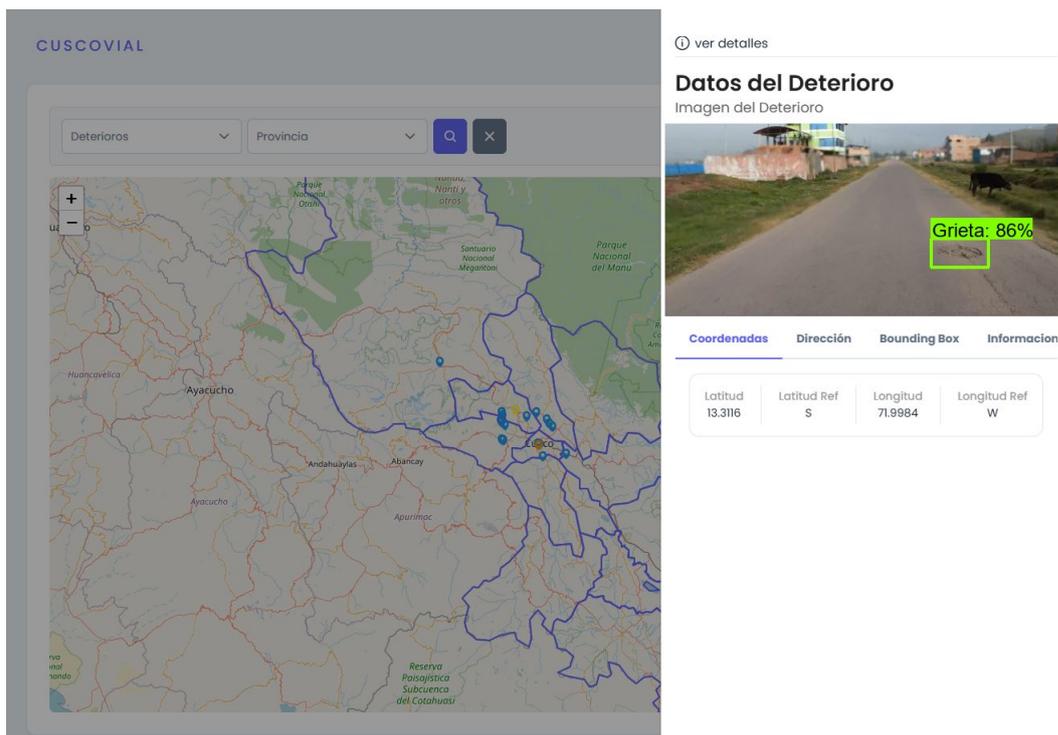
Fuente: Elaboración propia.

Página principal: visualización del deterioro

Al seleccionar un marcador en el mapa interactivo, el sistema despliega una vista detallada del deterioro correspondiente, como se muestra en la Figura 3.49. Esta vista incluye información geográfica precisa, como la latitud y longitud del punto registrado, así como la dirección aproximada del lugar donde se capturó la imagen.

Además, se visualizan los bounding boxes generados por la red neuronal convolucional, los cuales delimitan las zonas específicas donde se ha detectado deterioro. Finalmente, se incluye información complementaria como el nombre del usuario que realizó la carga de la imagen y la fecha en que el sistema registró la identificación del deterioro. Esta funcionalidad permite una trazabilidad clara y ordenada de cada evidencia registrada en la plataforma.

Figura 3.49: Frontend pagina de inicio ver deterioro



Elaboración propia.

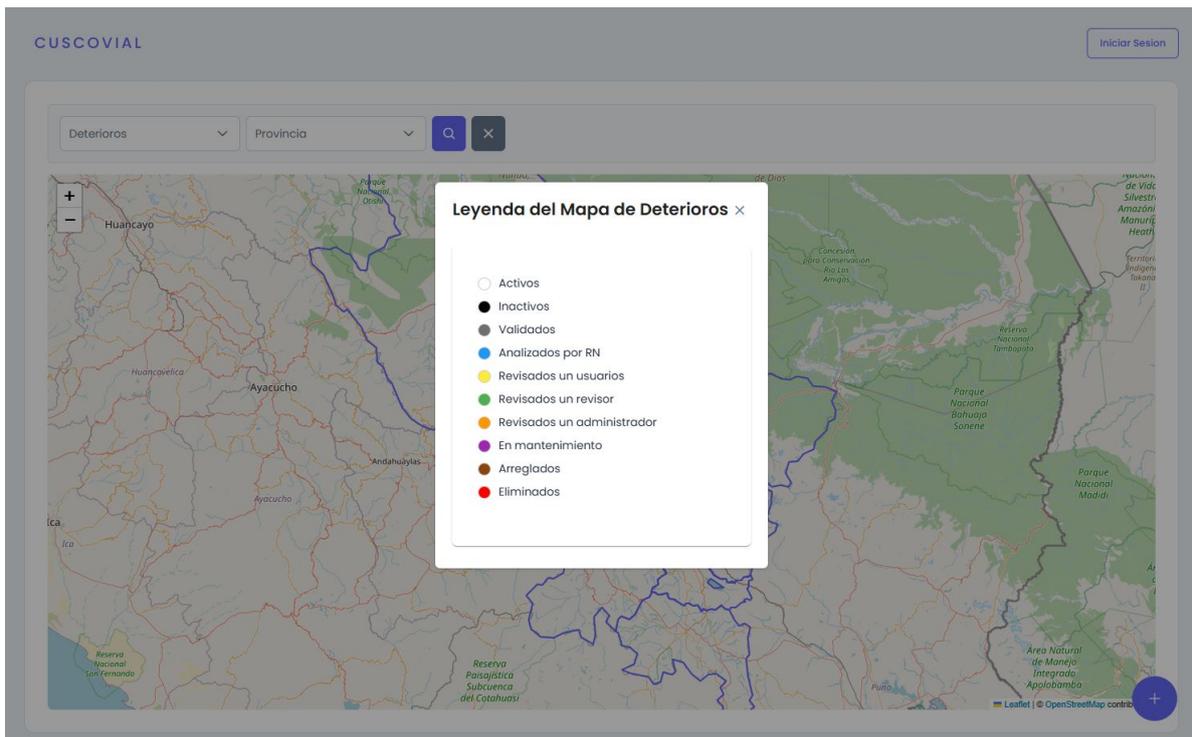
Fuente:

Página principal: leyenda del mapa

Al seleccionar el botón flotante ubicado en la parte inferior de la interfaz, se despliega un componente tipo SpeedDial que proporciona acceso a distintas funcionalidades adicionales. Una de ellas es la visualización de la leyenda del mapa, la cual permite interpretar adecuadamente la simbología utilizada.

La leyenda muestra la correspondencia entre los colores de los marcadores y los distintos tipos de deterioro identificados, facilitando así la comprensión de la información geoespacial representada. Esta funcionalidad tiene como objetivo mantener al usuario informado y orientado durante la exploración del mapa. La Figura 3.50 ilustra esta funcionalidad en la interfaz del sistema.

Figura 3.50: Frontend leyenda del mapa



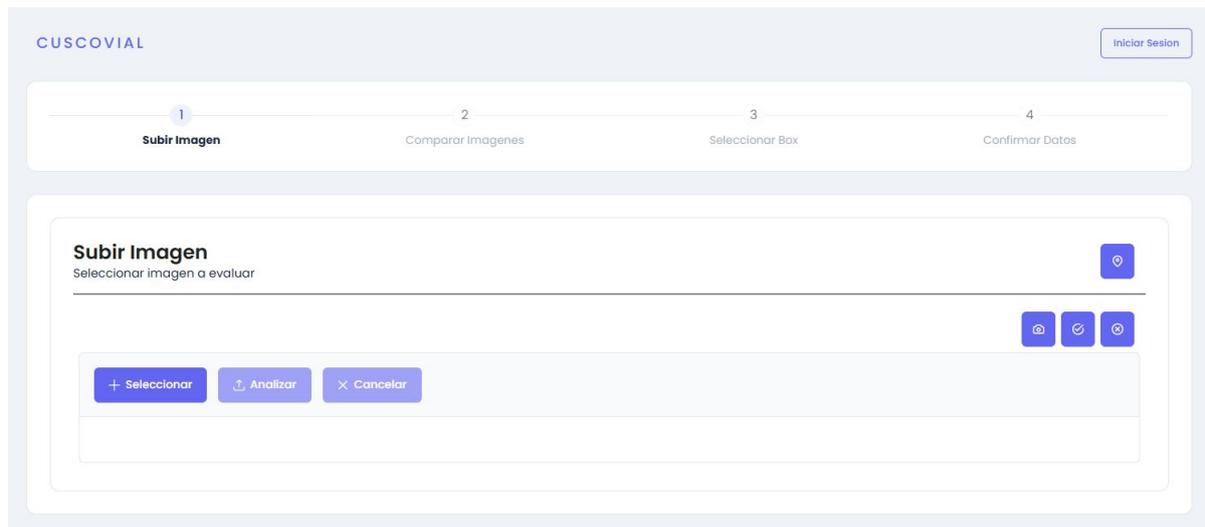
Fuente: Elaboración propia.

Página cargar imagen

Una de las funcionalidades disponibles mediante el botón flotante (SpeedDial) en la página principal es el acceso al módulo de carga de imágenes. Esta sección guía al usuario a través de un proceso dividido en cuatro fases, diseñado para facilitar el análisis automatizado y la georreferenciación de deterioros viales a partir de evidencia fotográfica.

- Selección o captura de imagen:** En esta primera fase, ilustrada en la Figura 3.51, el usuario puede seleccionar desde su dispositivo una imagen previamente almacenada o capturar una nueva fotografía del deterioro vial mediante la cámara del equipo cliente.

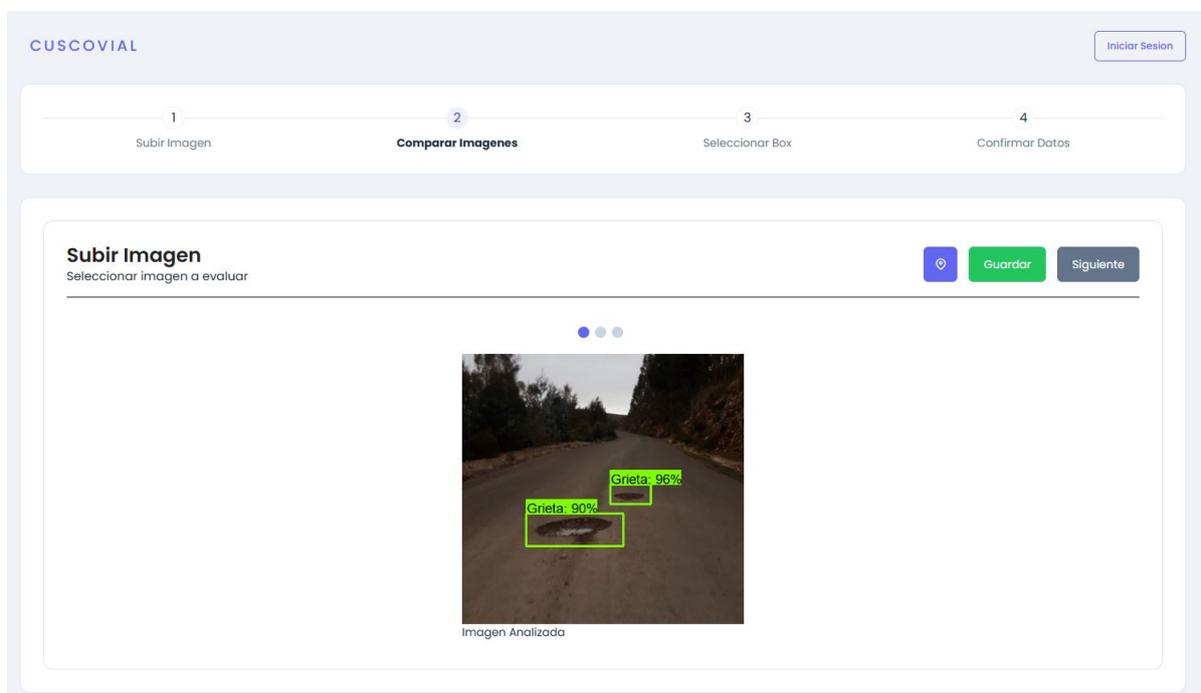
Figura 3.51: Frontend Subir Imagen



Fuente: Elaboración propia.

- Análisis automático con red neuronal convolucional:** Una vez seleccionada la imagen, el sistema ofrece la opción de solicitar el análisis automatizado mediante una red neuronal convolucional. Este análisis tiene como objetivo identificar los deterioros visibles presentes en la imagen. La Figura 3.52 muestra un ejemplo de esta funcionalidad.

Figura 3.52: Frontend Deterioros Identificados por la Red Neuronal Convolucional



Fuente: Elaboración propia.

- Revisión y edición de *bounding boxes*:** En esta etapa, se presentan los posibles contornos (*bounding boxes*) detectados automáticamente por la red neuronal.

El usuario tiene la posibilidad de confirmar, eliminar o añadir manualmente los deterioros detectados, ajustando así el resultado final. Esta funcionalidad se muestra en la Figura 3.53.

Figura 3.53: Frontend Selección de Bounding Boxes

Bounding Box Identificados
Selecciona los bounding box correctos

Nº	Puntaje	Coordenadas	
1	0.961	0.485 - 0.524 0.554 - 0.671	<input checked="" type="checkbox"/>
2	0.909	0.593 - 0.227 0.710 - 0.574	<input checked="" type="checkbox"/>
3	0.219	0.324 - 0.380 0.345 - 0.465	<input type="checkbox"/>
4	0.188	0.334 - 0.301 0.353 - 0.387	<input type="checkbox"/>
5	0.181	0.344 - 0.248 0.362 - 0.321	<input type="checkbox"/>
6	0.156	0.351 - 0.241 0.373 - 0.326	<input type="checkbox"/>

Fuente: Elaboración propia.

- Visualización de metadatos:** Finalmente, el sistema extrae y muestra los metadatos embebidos en la imagen, como la latitud, longitud, fecha de captura y otros atributos relevantes. Esta información contribuye a la georreferenciación precisa del deterioro. La Figura 3.54 ejemplifica esta funcionalidad.

Figura 3.54: Frontend Metadatos Identificados

CUSCOVIAL Iniciar Sesión

1 Subir Imagen 2 Comparar Imágenes 3 Seleccionar Box 4 **Confirmar Datos**

Metadatos
Información de la imagen 📍 Guardar Siguiente

Dirección
Asociación Pro Vivienda Praderas del Inca, San Sebastián, Cusco, 08006, Perú

País: Perú Departamento: Cusco Provincia: Cusco

Distrito: San Sebastián Código Postal: 08006 Código de vía: Código de vía

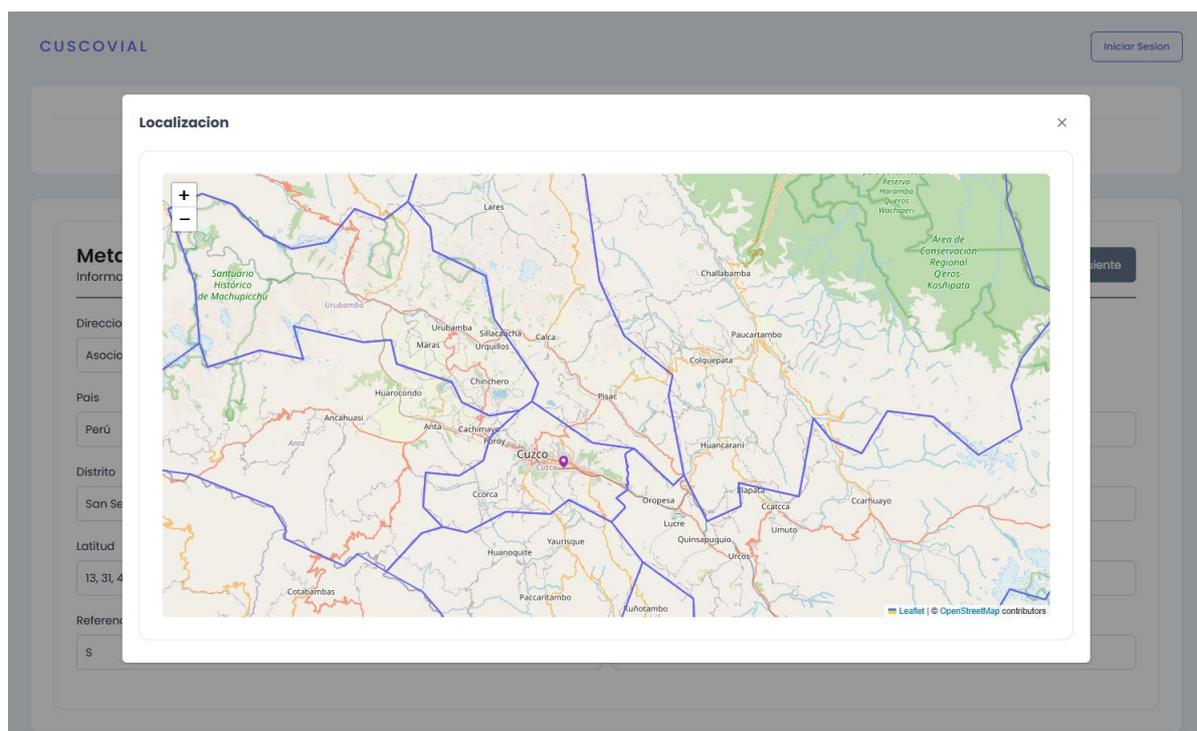
Latitud: 13, 31, 47.9893599 Longitud: 71, 55, 29.4440746

Referencia Latitud: S Referencia Longitud: W

Fuente: Elaboración propia.

Adicionalmente, el sistema permite visualizar de forma preliminar la ubicación geográfica aproximada del deterioro sobre el mapa, mediante un componente tipo dialog emergente, como se muestra en la Figura 3.55.

Figura 3.55: Frontend Ubicación de la Imagen



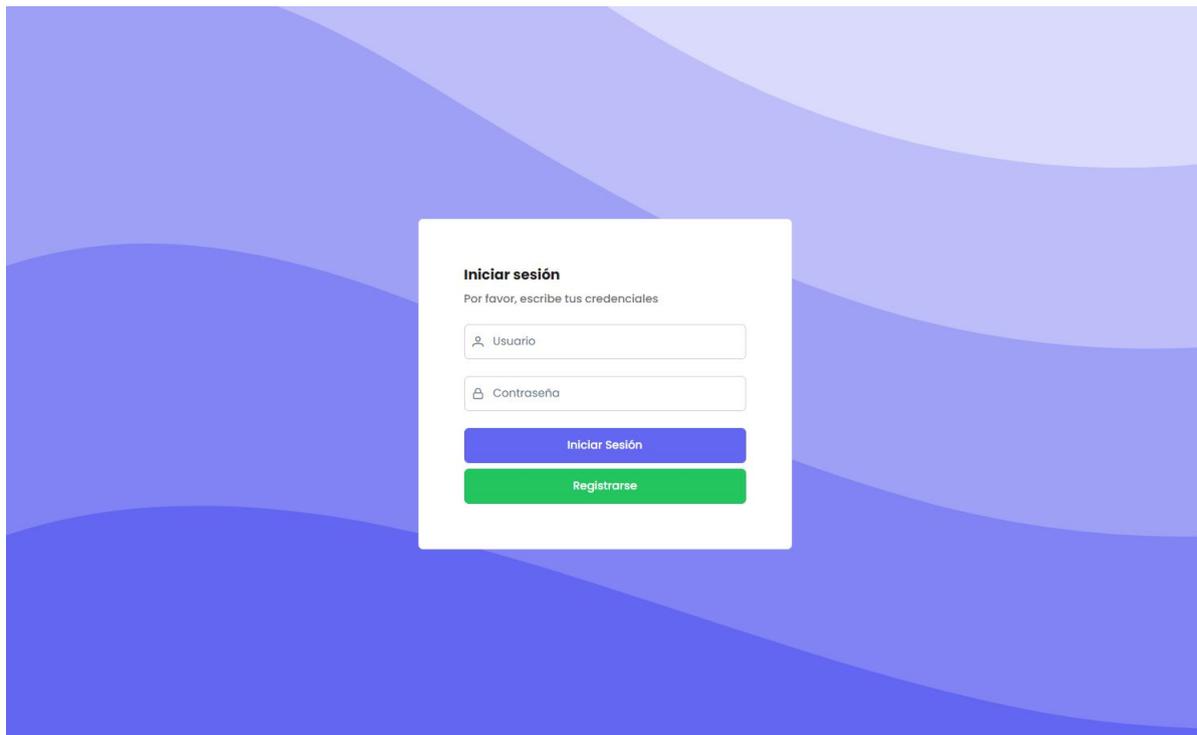
Fuente: Elaboración propia.

Página iniciar sesión

El sistema contempla un mecanismo de autenticación que permite a ciertos usuarios acceder mediante credenciales a cuentas con privilegios diferenciados. Estas cuentas privilegiadas habilitan herramientas adicionales y funcionalidades de gestión no disponibles para usuarios estándar (Poblador), tales como la administración de los deterioros reportados, la validación de datos y el acceso a estadísticas del sistema.

La interfaz que permite el acceso a estas cuentas se presenta en la Figura 3.56.

Figura 3.56: Frontend página para acceder al sistema



Iniciar sesión
Por favor, escribe tus credenciales

🔍 Usuario

🔑 Contraseña

Iniciar Sesión

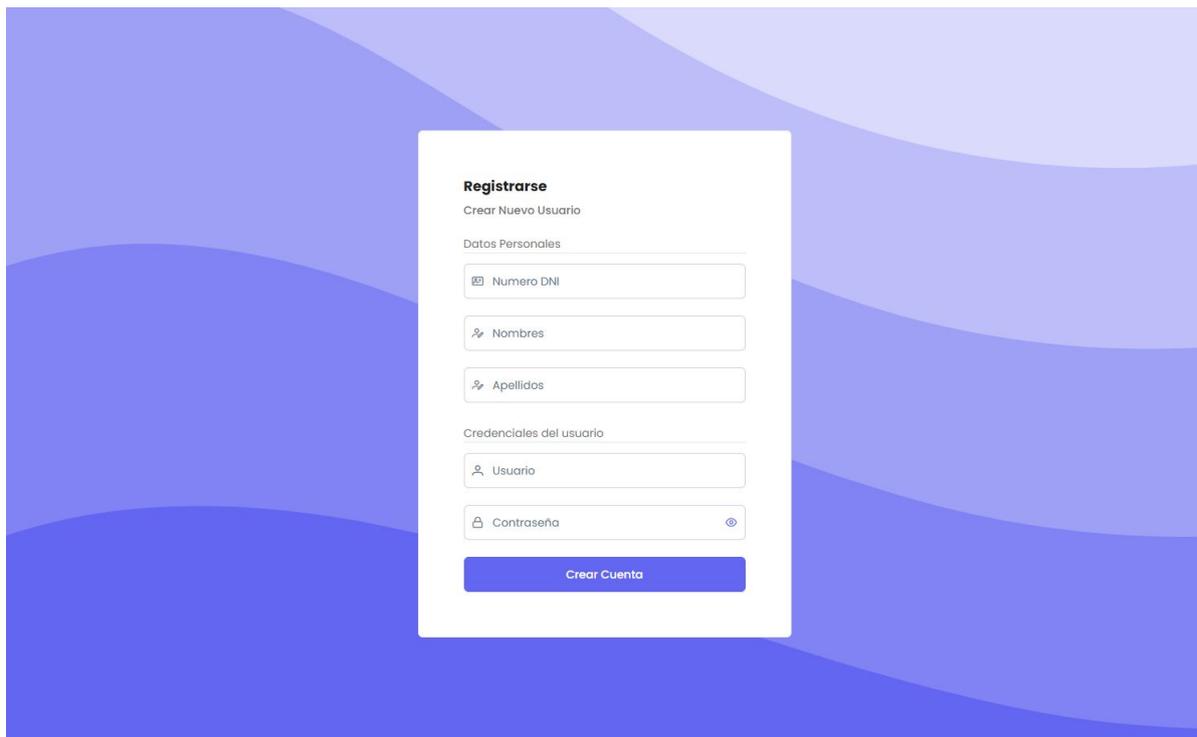
Registrarse

Fuente: Elaboración propia.

Página registrar usuario

Esta interfaz permite a los usuarios, particularmente a los pobladores, registrarse en el sistema mediante la creación de una cuenta personal. El formulario de registro solicita información básica, así como la definición de credenciales de acceso seguras, incluyendo una contraseña, donde su interfaz correspondiente está señalada en 3.57.

Figura 3.57: Frontend registro de usuario



The image shows a user registration form titled "Registrarse" (Register) with the subtitle "Crear Nuevo Usuario" (Create New User). The form is set against a blue background with abstract wave patterns. It is divided into two main sections: "Datos Personales" (Personal Data) and "Credenciales del usuario" (User Credentials). The "Datos Personales" section includes three input fields: "Numero DNI" (DNI Number) with a document icon, "Nombres" (First Name) with a person icon, and "Apellidos" (Last Name) with a person icon. The "Credenciales del usuario" section includes two input fields: "Usuario" (Username) with a magnifying glass icon and "Contraseña" (Password) with a lock icon and a toggle for visibility. A blue "Crear Cuenta" (Create Account) button is positioned at the bottom of the form.

Fuente: Elaboración propia.

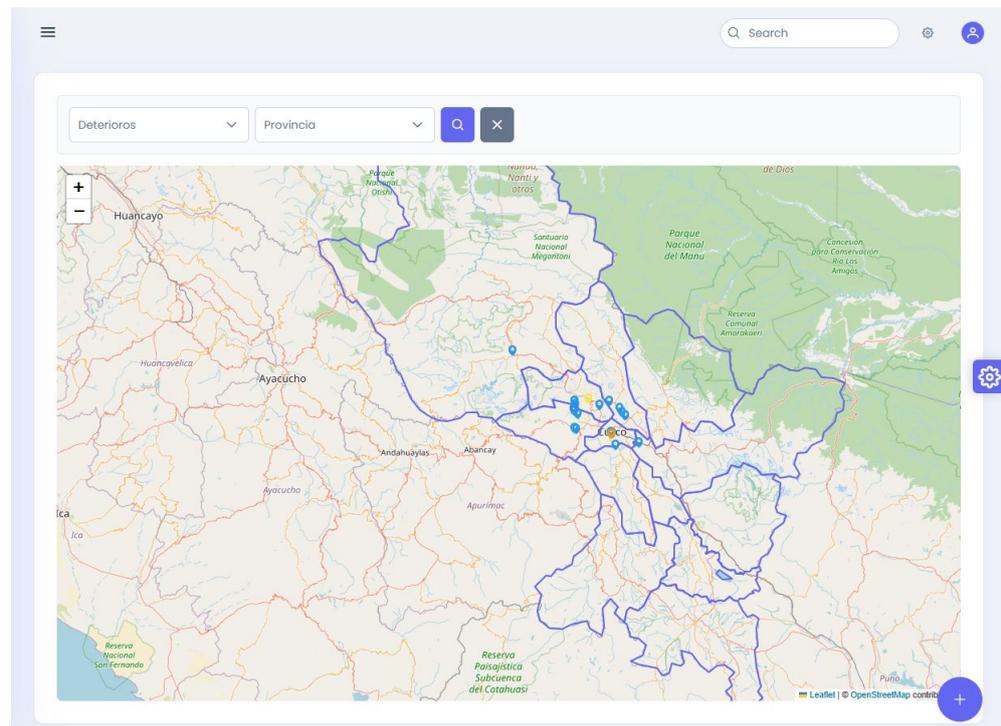
Página principal de usuario registrado

La interfaz destinada a los usuarios registrados habilita nuevas funcionalidades orientadas a la gestión personalizada de los deterioros registrados. Mediante un sidebar lateral, el usuario puede acceder a un conjunto de herramientas adicionales, entre las cuales se encuentra la sección que permite visualizar los deterioros previamente reportados por él mismo.

También se incluye el acceso a un módulo de revisión, el cual permite al usuario verificar y validar la información asociada a los deterioros que ha registrado. Esta funcionalidad contribuye a aumentar la certeza y precisión de los datos entregados al sistema, reforzando la calidad y fiabilidad del repositorio de evidencias georreferenciadas.

La Figura 3.58 presenta la interfaz correspondiente a esta vista.

Figura 3.58: Frontend menú del usuario poblador

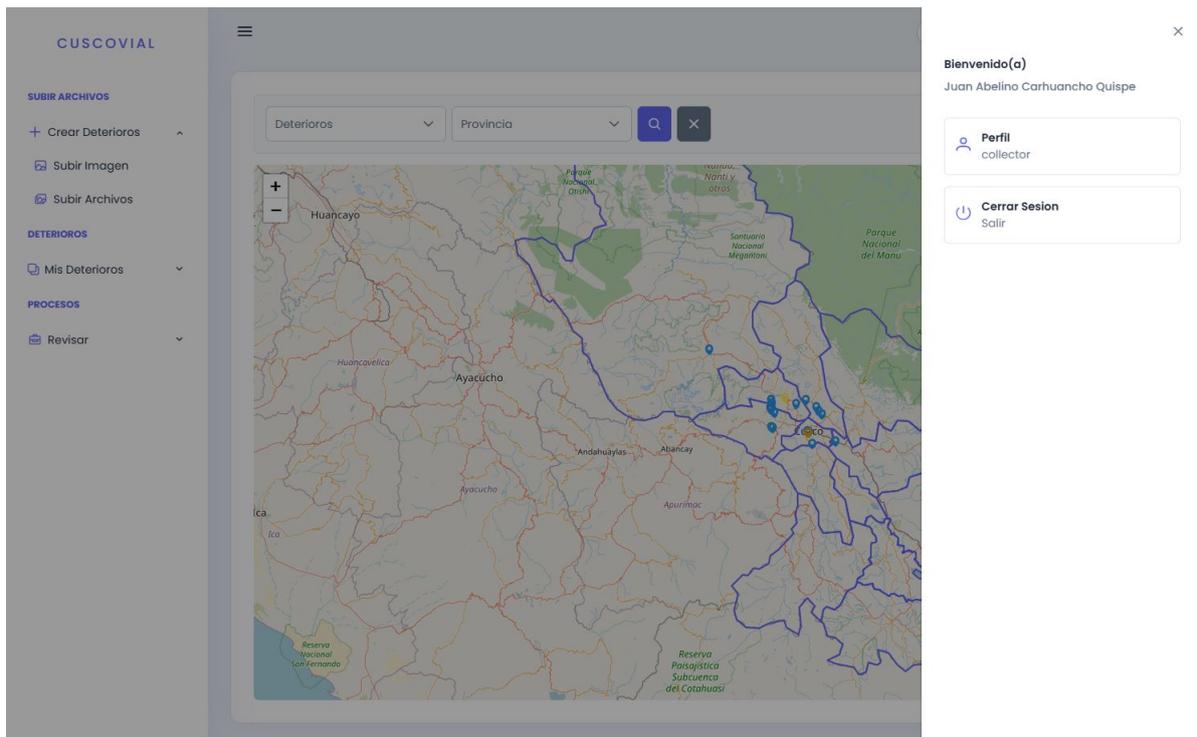


Fuente: Elaboración propia.

Página principal del usuario registrado

Además de las funcionalidades orientadas a la gestión de deterioros, la interfaz principal del usuario registrado incorpora un segundo sidebar, accesible desde el menú lateral. Este componente, ilustrado en la Figura 3.59, permite al usuario cerrar su sesión activa de manera segura, garantizando así el resguardo de su información personal y evitando accesos no autorizados.

Figura 3.59: Frontend modulo para salir del sistema



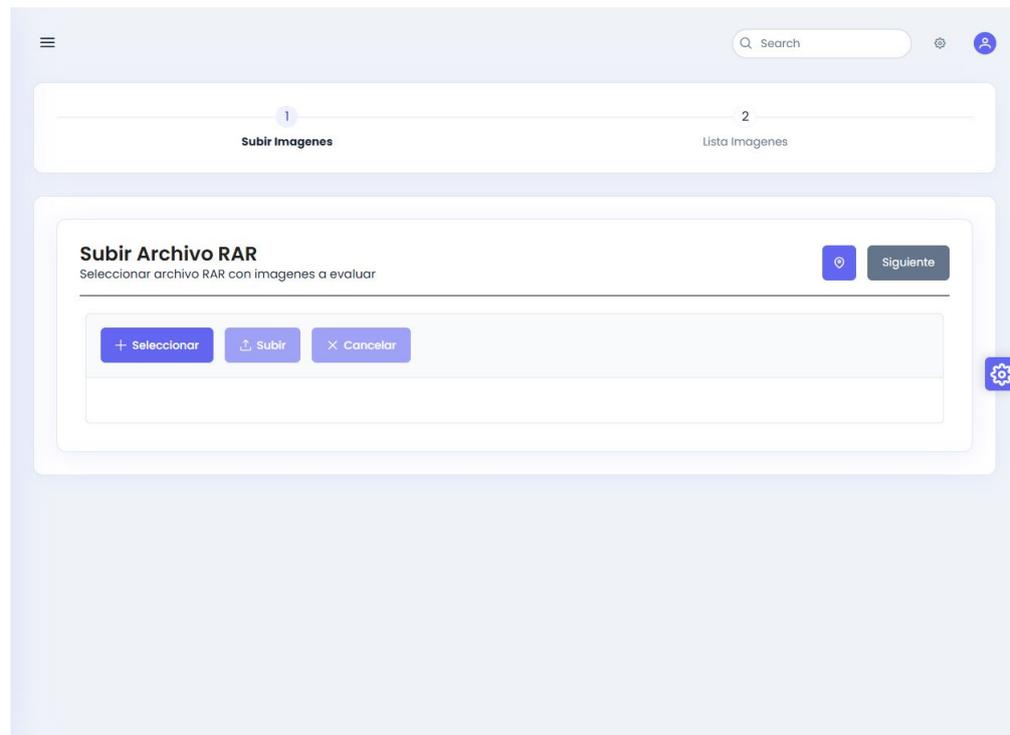
Fuente: Elaboración propia.

Página subir lote de imágenes

Los usuarios autenticados también cuentan con la funcionalidad de cargar un conjunto de imágenes de manera simultánea, permitiendo así una gestión más eficiente en contextos donde se dispone de múltiples evidencias fotográficas. Esta opción resulta especialmente útil para usuarios que requieren registrar deterioros de forma masiva, como técnicos municipales o gestores viales.

La interfaz correspondiente, representada en la Figura 3.60, permite seleccionar múltiples archivos desde el dispositivo del usuario para su posterior análisis por parte del sistema.

Figura 3.60: Frontend subir lote de imágenes



Fuente: Elaboración propia.

La siguiente interfaz, ilustrada en la Figura 3.61, cumple la función de notificar al usuario que el conjunto de imágenes seleccionadas ha sido cargado exitosamente al sistema. Sin embargo, se debe tener en cuenta que en este punto aún no se ha iniciado el análisis de las imágenes por parte de la red neuronal convolucional.

Esta interfaz representa una etapa intermedia en el flujo de trabajo, en la cual el sistema ha recibido los archivos y queda a la espera de que se ejecute el proceso de análisis automatizado. Dicha acción puede ser activada posteriormente por el usuario o por procedimientos automáticos internos del sistema.

Figura 3.61: Frontend confirmación de subida de imágenes



Fuente: Elaboración propia.

Página lista de deterioros personales

Esta interfaz, representada en la Figura 3.62, permite al usuario visualizar un listado de los deterioros que él mismo ha registrado previamente en el sistema. A través de esta vista, el usuario puede consultar el estado actual de cada deterioro dentro del flujo de procesamiento, lo cual incluye si ya ha sido analizado por la red neuronal convolucional.

Además, se proporciona la posibilidad de realizar un seguimiento de los resultados generados por el análisis automático, permitiendo al usuario revisar y, en caso de considerarlo necesario, ajustar o validar los datos asociados. Esto contribuye a mejorar la certeza y veracidad de la información almacenada en el sistema.

Figura 3.62: Frontend lista deterioros personales

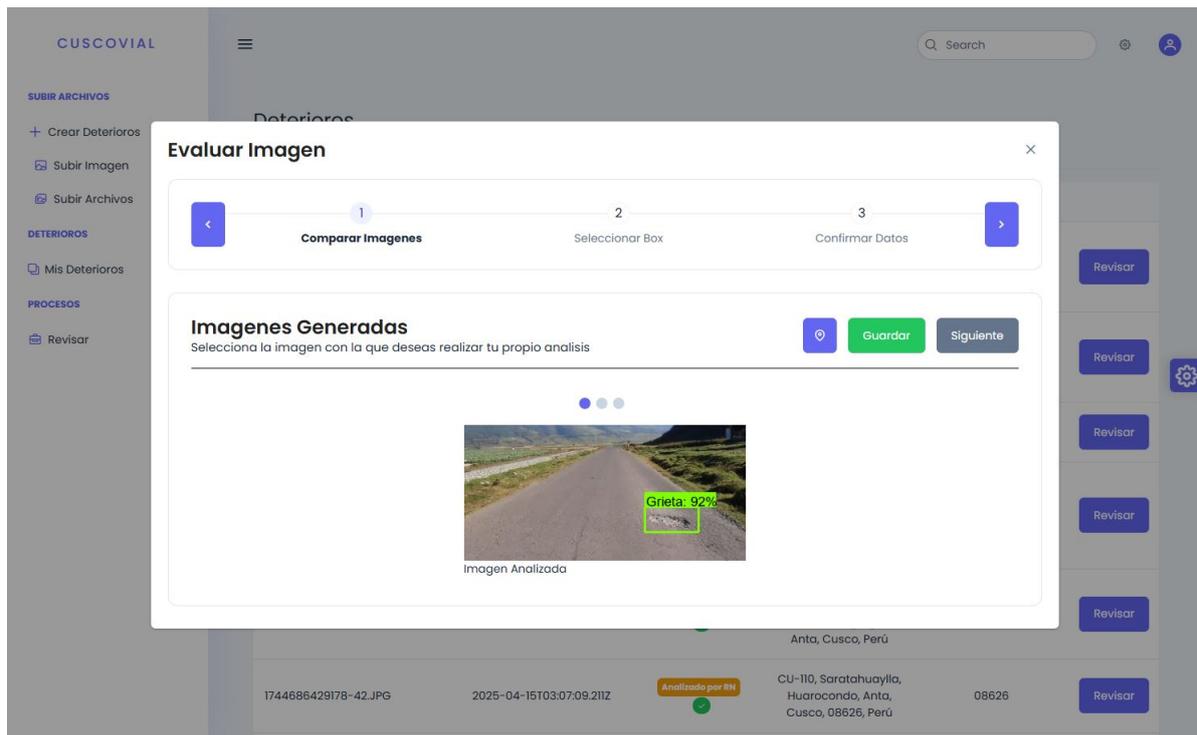
Nombre	Fecha Creacion	Estado	Direccion	Coordenadas
1744686429250-44.JPG	2025-04-15T03:07:09.274Z	Analizado por RN	CU-110, Utcubamba, Pumatales, Hurocondo, Anta, Cusco, Perú	
1744686429276-45.JPG	2025-04-15T03:07:09.304Z	Analizado por RN	CU-110, Ollantaytambo, Urubamba, Cusco, Perú	
1744686429306-46.JPG	2025-04-15T03:07:09.329Z	Analizado por RN	PE-288, Pisac, Calca, Cusco, 08106, Perú	08106
1744686429331-47.JPG	2025-04-15T03:07:09.360Z	Analizado por RN	Carretera Longitudinal de la Sierra Sur, Yanama, Zurite, Anta, Cusco, Perú	
1744686429361-48.JPG	2025-04-15T03:07:09.388Z	Analizado por RN	Carretera Longitudinal de la Sierra Sur, Inquilpata, Anta, Cusco, Perú	
1744686429178-42.JPG	2025-04-15T03:07:09.211Z	Analizado por RN	CU-110, Saratahuaylla, Hurocondo, Anta, Cusco, 08626, Perú	08626

Fuente: Elaboración propia.

Página revisar Deterioro

La interfaz ilustrada en la Figura 3.63 se despliega mediante un dialog emergiendo cuando el usuario selecciona uno de los deterioros previamente registrados desde la lista de deterioros personales.

Figura 3.63: Frontend revisión deterioros personales

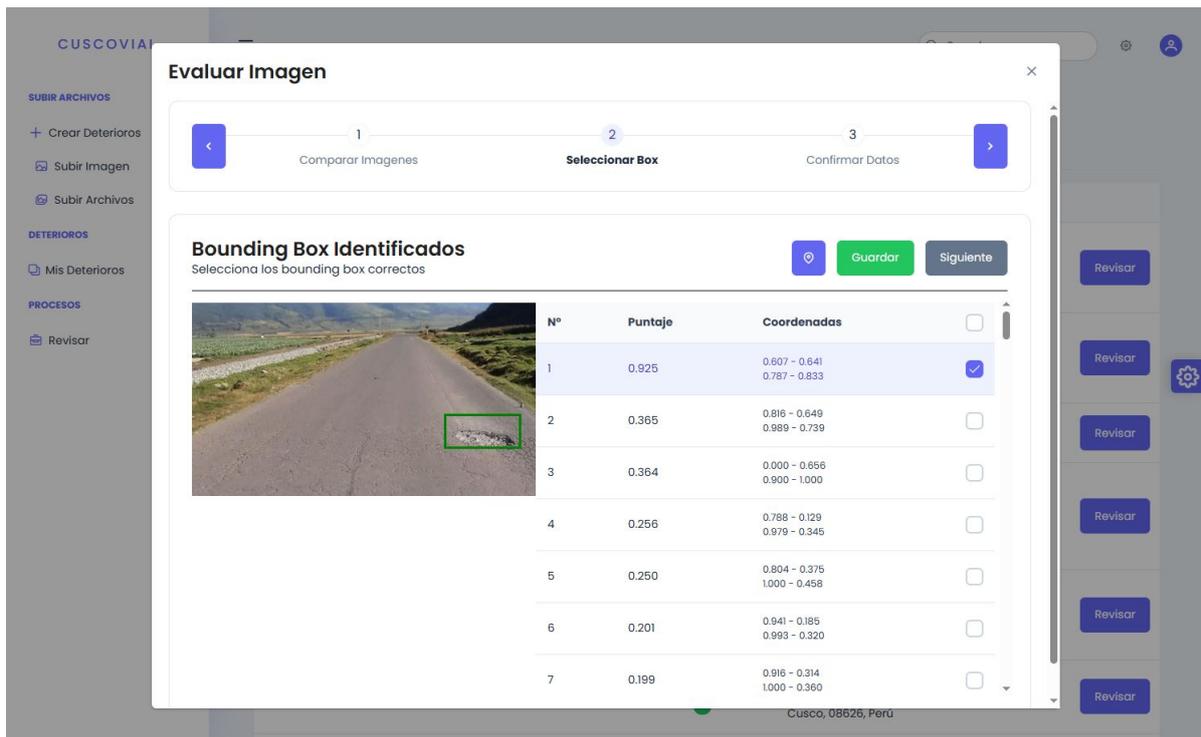


Fuente: Elaboración propia.

Esta funcionalidad permite al usuario revisar con mayor detalle los resultados del análisis automático realizado por la red neuronal. Específicamente, el usuario puede observar los bounding boxes generados automáticamente y decidir, de forma manual, si desea mantenerlos, eliminarlos o añadir otros nuevos en función de su propio criterio o conocimiento contextual.

Este proceso busca incrementar la certeza de la información almacenada, permitiendo una validación colaborativa entre la inteligencia artificial y la intervención humana, mostrada en la Figura 3.64.

Figura 3.64: Frontend revisión deterioros personales bounding boxes



The screenshot shows a web application interface for evaluating image damage. The main window is titled "Evaluar Imagen" and has a progress bar with three steps: "1 Comparar Imagenes", "2 Seleccionar Box" (current step), and "3 Confirmar Datos". Below the progress bar, there is a section titled "Bounding Box Identificados" with the instruction "Selecciona los bounding box correctos". This section includes a small image of a road with a green bounding box around a pothole. To the right of the image is a table with the following data:

Nº	Puntaje	Coordenadas	
1	0.925	0.607 - 0.641 0.787 - 0.833	<input checked="" type="checkbox"/>
2	0.365	0.816 - 0.649 0.989 - 0.739	<input type="checkbox"/>
3	0.364	0.000 - 0.656 0.900 - 1.000	<input type="checkbox"/>
4	0.256	0.788 - 0.129 0.979 - 0.345	<input type="checkbox"/>
5	0.250	0.804 - 0.375 1.000 - 0.458	<input type="checkbox"/>
6	0.201	0.941 - 0.185 0.993 - 0.320	<input type="checkbox"/>
7	0.199	0.916 - 0.314 1.000 - 0.360	<input type="checkbox"/>

At the bottom of the table, the location "Cusco, 08526, Peru" is displayed. On the right side of the interface, there is a vertical stack of "Revisar" buttons.

Fuente: Elaboración propia.

Finalmente, esta interfaz la Figura 3.65 también permite visualizar los metadatos extraídos automáticamente a partir de la imagen analizada. Entre ellos se incluyen la latitud y longitud de la fotografía, Esta información complementaria fortalece el valor del registro y facilita su posterior trazabilidad y análisis dentro del sistema.

Figura 3.65: Frontend revisión deterioros personales metadatos

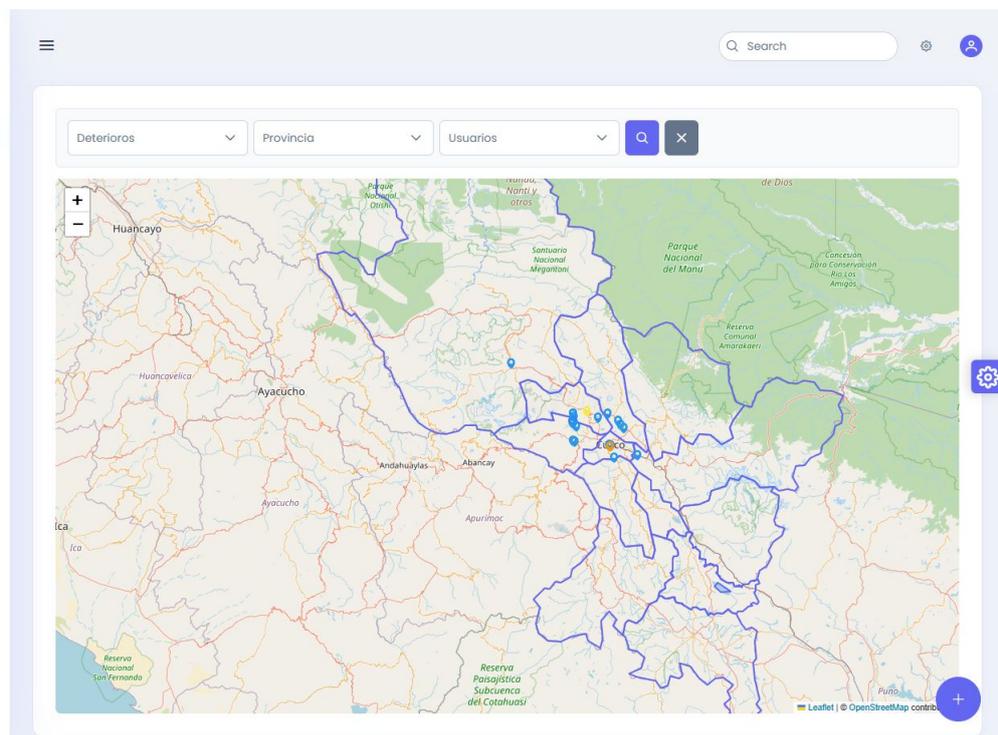
Fuente: Elaboración propia.

Página principal del perfil revisor

La interfaz ilustrada en la Figura 3.66 corresponde al panel principal al que acceden los usuarios con perfil revisor. Este perfil cuenta con privilegios adicionales orientados a la validación y supervisión del contenido registrado en el sistema.

Desde este panel, el revisor puede realizar un filtrado avanzado de los marcadores georreferenciados en el mapa, permitiendo clasificar la información en función a una lista de usuarios.

Figura 3.66: Frontend menú de usuario revisor



Fuente: Elaboración propia.

Página principal del perfil revisor

La interfaz mostrada en la Figura 3.67 habilita a los usuarios con perfil revisor el acceso a una vista consolidada de los deterioros registrados por cada uno de los usuarios del sistema. Esta funcionalidad permite identificar y seleccionar rápidamente a los usuarios que tienen deterioros pendientes de revisión.

A través de esta vista, el revisor puede filtrar, acceder y validar los deterioros reportados, lo que fortalece el control de calidad de la información ingresada al sistema. Esta herramienta es fundamental para garantizar que los datos georreferenciados sean precisos, pertinentes y estén debidamente verificados antes de formar parte del repositorio oficial.

Figura 3.67: Frontend deterioros por usuarios

CUSCOVIAL

SUBIR ARCHIVOS

+ Crear Deterioros

DETERIOROS

Mis Deterioros

Mis deterioros

Deterioros por Usuario

PROCESOS

Revisar

Q Search

Deterioros Por Usuario

Lista de imagenes con deterioros por usuario

jcarhuancho

Nombre ↑↓	Fecha Creacion ↑↓	Estado ↑↓	Direccion ↑↓	Coordenadas ↑↓
1744686429250-44.JPG	2025-04-15T03:07:09.274Z	Analizado por RH ✓	CU-110, Utcubamba, Pumatales, Huaracundo, Anta, Cusco, Perú	
1744686429276-45.JPG	2025-04-15T03:07:09.304Z	Analizado por RH ✓	CU-110, Ollantaytambo, Urubamba, Cusco, Perú	
1744686429306-46.JPG	2025-04-15T03:07:09.329Z	Analizado por RH ✓	PE-28B, Pisac, Calca, Cusco, 08106, Perú	08106
1744686429331-47.JPG	2025-04-15T03:07:09.360Z	Analizado por RH ✓	Carretera Longitudinal de la Sierra Sur, Yanama, Zurite, Anta, Cusco, Perú	
1744686429361-48.JPG	2025-04-15T03:07:09.388Z	Analizado por RH ✓	Carretera Longitudinal de la Sierra Sur, Inquillpata, Anta, Cusco, Perú	
1744686429378-49.JPG	2025-04-15T03:07:09.211Z	Analizado por RH ✓	CU-110, Saratahuaylla, Huaracundo, Anta	08626

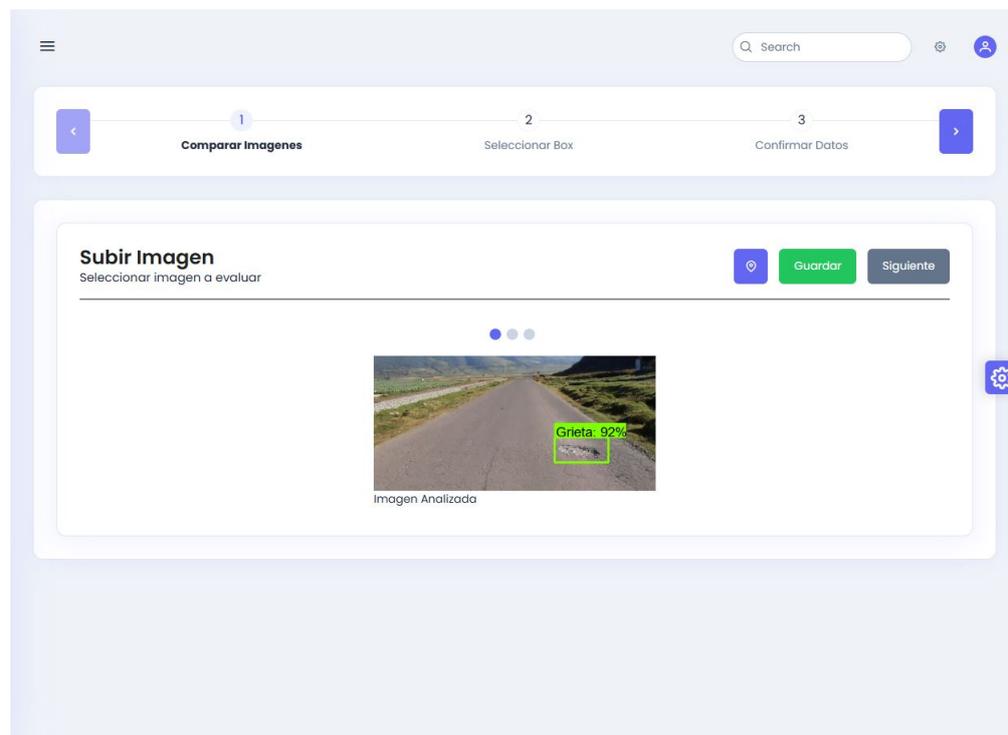
Fuente: Elaboración propia.

Página de revisión de deterioros globales

La interfaz mostrada en la Figura 3.68 permite a los usuarios con privilegios adecuados acceder al módulo de revisión de deterioros globales. Esta funcionalidad proporciona una vista consolidada de los deterioros registrados por todos los usuarios del sistema que aún no han sido validados completamente.

Su propósito principal es garantizar la integridad y calidad de los datos almacenados, permitiendo que otros usuarios contribuyan a la validación cruzada de la información. De este modo, se busca mantener la base de datos del sistema actualizada, confiable y con un adecuado grado de certeza en la identificación y caracterización de los deterioros.

Figura 3.68: Frontend revisión deterioros globales



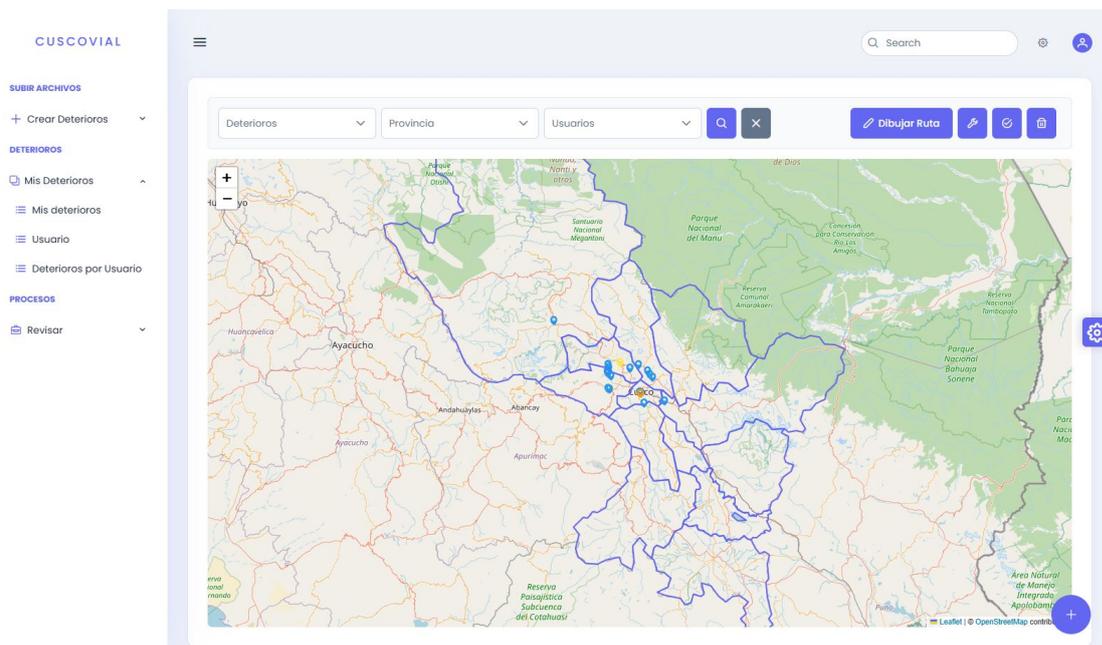
Fuente: Elaboración propia.

Página principal del perfil administrador

La interfaz representada en la Figura 3.69 corresponde al entorno principal al que acceden los usuarios con perfil de administrador. Esta vista, además de presentar el mapa interactivo con los marcadores de deterioros geolocalizados, incorpora un conjunto de herramientas exclusivas orientadas a la gestión integral del sistema.

En particular, el administrador cuenta con funcionalidades que permiten modificar el estado de los deterioros registrados, ya sea para corregir inconsistencias, actualizar su progreso en el flujo de validación, o realizar ajustes necesarios conforme a criterios técnicos o administrativos. De esta manera, el perfil administrador cumple un rol clave en el mantenimiento de la integridad y actualización permanente de la información dentro de la plataforma.

Figura 3.69: Frontend menú de usuario administrador



Elaboración propia.

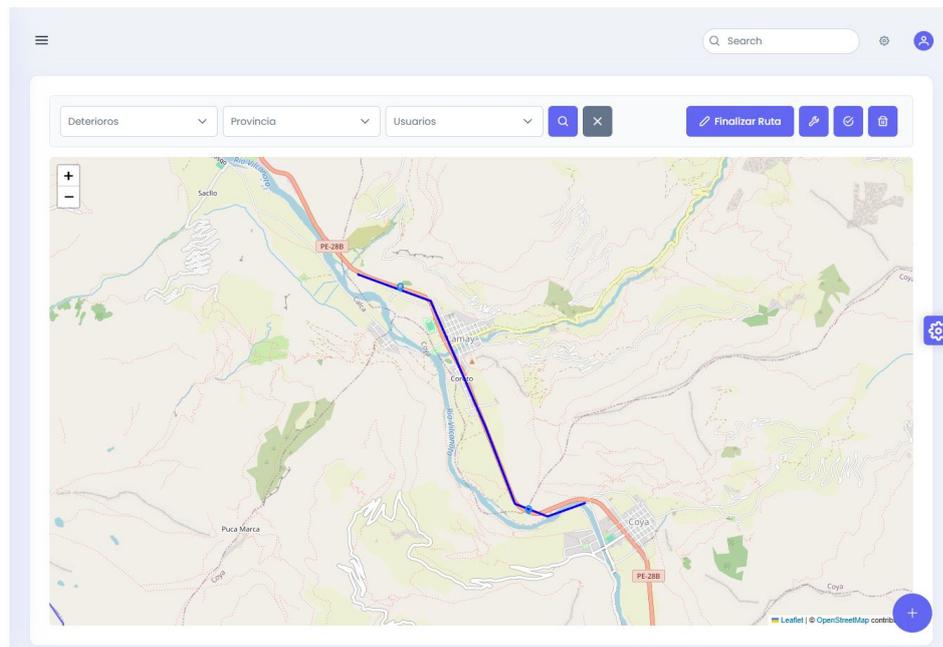
Fuente:

Página principal del perfil administrador: Dibujar Ruta

La interfaz representada en la Figura 3.70 permite a los usuarios con perfil de administrador trazar una ruta directamente sobre el mapa interactivo. Esta funcionalidad permite al administrador seleccionar y visualizar los deterioros geolocalizados cercanos a la línea o ruta dibujada.

Una vez que la ruta ha sido definida, el sistema identifica automáticamente todos los deterioros ubicados en su proximidad. Posteriormente, el administrador podrá modificar el estado de estos deterioros seleccionados, actualizando su clasificación, progreso en la validación o cualquier otra información relevante. Esta herramienta resulta fundamental para la gestión eficiente de las áreas de deterioro y para la optimización de los recursos del sistema, al facilitar una rápida intervención sobre los deterioros más cercanos a rutas específicas.

Figura 3.70: Frontend dibujar ruta



Fuente:

Elaboración propia.

Página Administrar Usuarios

La interfaz representada en la Figura 3.71 permite al administrador visualizar una lista completa de todos los usuarios registrados en el sistema. A través de esta página, el administrador tiene la capacidad de gestionar las cuentas de usuario, visualizando información relevante de cada uno.

Figura 3.71: Frontend administrar usuarios

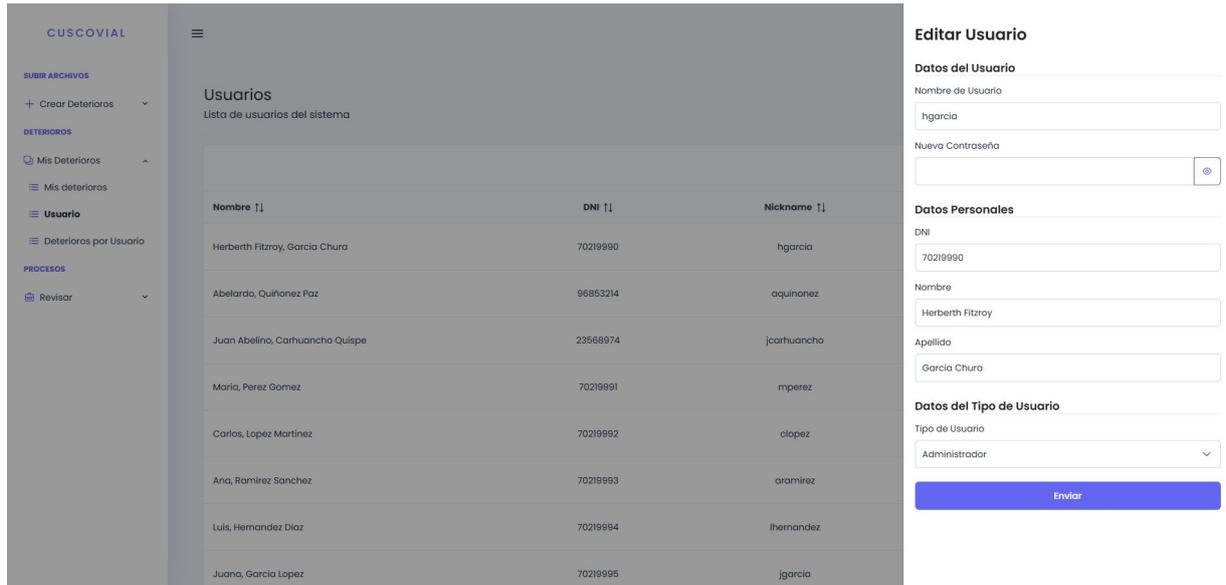
Nombre [↑]	DNI [↑]	Nickname [↑]	Tipo de Usuario [↑]	
Herberth Fitzroy, Garcia Chura	70218990	hgarcia	Administrador	Revisar
Abelardo, Quiñonez Paz	96853214	aquinonez	Publicador	Revisar
Juan Abelino, Carhuancha Quispe	23568974	jcarhuancha	Recolector	Revisar
Maria, Perez Gomez	70219991	mperez	Administrador	Revisar
Carlos, Lopez Martinez	70219992	clopez	Publicador	Revisar
Ana, Ramirez Sanchez	70219993	aramirez	Recolector	Revisar
Luis, Hernandez Diaz	70219994	lhernandez	Administrador	Revisar
Juana, Garcia Lopez	70219995	jgarcia	Publicador	Revisar

Fuente: Elaboración propia.

Página Administrar Usuarios: Editar Usuario

Al seleccionar un usuario específico de la lista, la interfaz representada en la Figura 3.72 desplegará un sidebar con la información detallada del usuario seleccionado. En este sidebar, el administrador podrá visualizar y modificar los datos personales del usuario. Además, se podrá actualizar el estado de la cuenta, ya sea para cambiar de perfil o darle nuevos permisos.

Figura 3.72: Frontend editar usuario



The screenshot shows the 'Editar Usuario' form in the CUSCOVIAL system. The form is divided into several sections:

- Datos del Usuario:**
 - Nombre de Usuario:
 - Nueva Contraseña:
- Datos Personales:**
 - DNI:
 - Nombre:
 - Apellido:
- Datos del Tipo de Usuario:**
 - Tipo de Usuario:

At the bottom of the form is a blue button labeled 'Enviar'.

Fuente: Elaboración propia.

Capítulo 4

Análisis y Discusión de Resultados

4.1. Análisis de resultados respecto a los objetivos

El proceso de desarrollo del prototipo identificador de deterioros en vías vehiculares pavimentadas se realizó tomando en cuenta en todo momento los objetivos propuestos en el plan de tesis, alcanzando los siguientes resultados en cada uno de ellos.

4.1.1. Recolección de imágenes para la construcción y configuración de un Dataset

La construcción del dataset se realizó de manera manual, dirigiéndose a la toma de fotografías en las zonas del Cusco con más deterioros y en distintas condiciones, según la información proporcionada por el municipio del Cusco y el MTC. Estas imágenes fueron sometidas a ciertas evaluaciones, identificando manualmente todos los deterioros en cada fotografía a través de LinkedAI, para obtener los bounding boxes de las imágenes en formato JSON. Esto permitió realizar la división de los datos para testing y entrenamiento en un 20% y 80%, respectivamente. Estas bases fueron fundamentales para obtener los archivos TFRecord y concluir la construcción de manera satisfactoria un dataset de deterioros para el entrenamiento de la CNN y posteriormente para posibles entrenamientos en futuros proyectos.

4.1.2. Desarrollo de una red neuronal para el reconocimiento de deterioros en imágenes

El reconocimiento de deterioros en imágenes se llevó a cabo mediante una red neuronal convolucional entrenada con el 70% de los datos del dataset y testeada con el 30% restante. Se determinó que la red alcanza una precisión del 80% al identificar deterioros en imágenes de vías vehiculares pavimentadas, generando 100 bounding boxes donde se detectaron estos deterioros y proporcionando sus respectivos scores o puntua-

ciones de probabilidad. Se consideraron verídicos aquellos scores que alcanzaron al menos un 70 %. Sin embargo, la red presenta cierta debilidad en el reconocimiento debido a factores ambientales, como la luz del sol, que genera sombras de los árboles y postes sobre el pavimento, lo que puede dificultar la identificación precisa en algunos casos. A pesar de estas limitaciones, la mayor relevancia de este desarrollo radica en la existencia de una herramienta práctica, fidedigna, imparcial y rápida para la identificación de deterioros en imágenes.

4.1.3. Desarrollo de un prototipo web para la recolección de evidencia del estado de las vías

La aplicación web, construida utilizando PostgreSQL para la base de datos y los frameworks de backend y frontend Flask, ExpressJS y React, ofrece una solución práctica y poco convencional. Facilita el muestreo de datos mediante un mapa de la región del Cusco, donde los datos están representados por marcadores y diferenciados según cada una de las provincias. La aplicación permite a cualquier poblador subir imágenes, ya sea de forma individual o grupal, para que posteriormente sean analizadas por una red neuronal convolucional, identificando los deterioros presentes en cada imagen y permitiendo así hacer llegar su reclamo a las entidades encargadas del mantenimiento de las vías de tránsito. Además, permite que el contenido pueda ser supervisado y actualizado constantemente por un administrador.

4.1.4. Verificación del comportamiento del prototipo en una vía nacional

El análisis del sistema se realizó basándonos en una de las vías nacionales más antiguas de nuestra región, la ruta Cusco - Pisac - Urubamba. Se capturó un total de 250 imágenes, en las cuales la red neuronal convolucional identificó 600 deterioros. De estos, 300 bounding boxes obtuvieron una puntuación entre el 60 % y el 70 %, 250 entre el 70 % y el 80 %, y 50 superaron el 80 %. Esto permite concluir que si un humano identifica un elemento agrietado o algún deterioro en una imagen, la red neuronal también lo detectará en la mayoría de los casos. Por lo tanto, la precisión del 80 % de la red es suficiente para asegurar que no habrá problemas al proponer el proyecto para su implementación en producción.

4.2. Discusión de resultados respecto a los antecedentes

Aquí tienes el texto corregido y mejorado:

Marcelo Fabrizio Letelier Palomares (2021) empleó la red neuronal YOLO-DarkNet53 para la detección de grietas, demostrando la eficacia de una red pre-entrenada en tareas

de reconocimiento y localización en imágenes y videos. En nuestro proyecto, aunque no utilizamos YOLO, la implementación de una red neuronal con MobileNetV2 mostró altos niveles de precisión, alcanzando un 80%. Esto confirma que las redes neuronales pre-entrenadas son herramientas poderosas que pueden reutilizarse en diferentes tipos de tareas y especialidades. Además, la herramienta Google Colab permite el entrenamiento de redes neuronales sin problemas y sin la necesidad de contar con equipos de alto rendimiento, ya que estos entornos de desarrollo ejecutan todo en la nube con servidores de gran capacidad de procesamiento.

Marcelo Ignacio Orellana Espinoza (2019) destacó la importancia de evitar el sobreajuste (overfitting) en la red neuronal y de no someterla a demasiadas épocas de entrenamiento. El sobreentrenamiento puede hacer que la red neuronal se vuelva demasiado estricta y rígida, buscando patrones muy específicos en las imágenes, lo que dificultaría la identificación de deterioros debido a la variabilidad en los patrones de las imágenes.

Julián Alberto Ortega Triana (2021) especificó que los datos de prueba y entrenamiento deben ser completamente distintos y separados. Además, sugirió que la data augmentation se aplique solo después de haber separado los datos y no durante el entrenamiento. Esto se debe a que la red neuronal, al ser entrenada, puede mostrar altos niveles de precisión en el conjunto de prueba, pero estos resultados no siempre se reflejan al aplicarla en otros entornos. Al seguir este procedimiento, el resultado fue positivo, ya que la red neuronal construida no tuvo problemas para adaptarse a las nuevas imágenes obtenidas en la ruta nacional Cusco - Pisac - Urubamba.

Ignacio Soto Percy Robustiano (2022), en su trabajo sobre la identificación de grietas en asfalto en Perú, concluyó que las redes neuronales convolucionales son eficaces para la detección de objetos en imágenes. En nuestro caso, se comprobó que por cada deterioro observado por un humano, la red neuronal identificó al menos uno, con una precisión del 80%. Además, se observó que las condiciones medioambientales no solo afectan el deterioro de las vías de tránsito, sino que también pueden influir severamente en la identificación de estos deterioros mediante tecnologías.

Conclusiones

- La adquisición y preparación de datos fue uno de los mayores desafíos del proyecto, debido al tiempo y esfuerzo requeridos para organizar y etiquetar imágenes. Aunque reutilizar datasets existentes es útil para acelerar el entrenamiento de redes neuronales, también es necesario fomentar la generación de nuevos datasets especializados desde instituciones públicas, privadas y académicas.
- La red neuronal desarrollada, con una precisión del 82 %, demostró una capacidad efectiva para identificar deterioros de forma comparable a un evaluador humano. Su ventaja radica en la consistencia del análisis y la ausencia de sesgos, lo que la convierte en una herramienta confiable y accesible para usuarios sin conocimientos técnicos.
- El prototipo desarrollado no solo valida el uso de la red neuronal, sino que demuestra la importancia de contar con una plataforma funcional que facilite su uso. Sin una interfaz accesible y orientada al usuario, el potencial del modelo se vería limitado, reduciendo su impacto en la mejora de la gestión vial.
- La implementación del sistema en el tramo Pisac - Calca – Urubamba confirmó su viabilidad en un entorno real. Sin embargo, se identificó que el proceso de reporte aún puede resultar complejo para algunos usuarios. Es necesario simplificar esta tarea para fomentar una mayor participación ciudadana y mejorar la adopción del sistema.

Recomendaciones

- Es fundamental que los conjuntos de entrenamiento (Train) y prueba (Test) sean completamente independientes, evitando que imágenes duplicadas o muy similares, como diferentes ángulos de la misma fotografía, se repitan en ambos grupos. Esto garantiza una evaluación más precisa y realista del modelo. Además, las técnicas de aumento de datos (data augmentation) deben aplicarse exclusivamente al conjunto de entrenamiento para no sesgar los resultados.
- Durante el entrenamiento de la red neuronal, es clave evitar el sobreentrenamiento (overfitting), ya que compromete la capacidad del modelo para generalizar. Es recomendable controlar el número de épocas y evitar una alimentación excesiva de datos para mantener un equilibrio entre precisión y flexibilidad.
- Para desarrollar un sistema en un dominio específico, es esencial trabajar en conjunto con especialistas del área. El uso de metodologías como historias de usuario y diagramas de flujo facilita la identificación de requerimientos técnicos y permite construir una base de datos coherente, agilizando así la implementación de un sistema alineado con las necesidades reales.
- La ampliación del sistema mediante una aplicación móvil y el uso de redes neuronales representa una mejora clave. Una app desarrollada para dispositivos Android e iOS permitiría procesar imágenes directamente desde el dispositivo, mejorando la autonomía operativa en contextos con conectividad limitada. Además, se recomienda fomentar la participación ciudadana mediante campañas informativas y mecanismos de incentivo, asegurando así la continuidad del sistema y una gestión vial más eficiente.
- Se recomienda adaptar el modelo de red neuronal convolucional para operar sobre secuencias de video capturadas por cámaras o teléfonos celulares en movimiento, lo que permitiría una detección más rápida y continua de deterioros viales. Esta mejora facilitaría la recolección eficiente de datos en tiempo real, optimizando el monitoreo de grandes tramos y fortaleciendo la capacidad del sistema para apoyar la toma de decisiones en mantenimiento vial.
- Recomendaciones Se recomienda explorar el uso del sistema en otros enfoques de aprendizaje profundo aplicados a la detección de diversas fallas en la infraestructura vial y el entorno, como problemas en señalización, semaforización, accidentes, fallas geológicas, etc. Esto requeriría adaptar el modelo con nuevas arquitecturas de redes neuronales, construir datasets específicos y definir criterios técnicos adecuados según el tipo de anomalía a identificar.

- El sistema presenta un diseño escalable y flexible que permite su adaptación a distintas entidades del Estado o empresas responsables del mantenimiento de vías vehiculares pavimentadas. Esta capacidad de integración facilita su implementación a mayor escala, promoviendo una coordinación más efectiva entre la ciudadanía y las autoridades, y fortaleciendo el trabajo colaborativo en la gestión y conservación de la infraestructura vial.

Bibliografía

- Bisong, E. (2019). Google colab. Disponible en: <https://www.oreilly.com/library/view/google-colaboratory/9781484244752/>.
- Budde, R., Kautz, K., Kuhlenkamp, K., and Züllighoven, H. (1992). *Prototyping: An Approach to Evolutionary System Development*. Springer-Verlag.
- Chollet, F. (2017). *Deep Learning with Python*. Manning Publications, Shelter Island, NY.
- CIPIA (2023). Driver sense. <https://cipia.com/driver-sense/>.
- Congreso de la República del Perú (1999). Ley n.º 27181 - ley general de transporte y tránsito terrestre. \LaTeX : Ley General de Transporte y Tránsito Terrestre. Disponible en: <https://www.gob.pe/institucion/mtc/normas-legales/294819-27181>.
- Congreso de la República del Perú (2000). Ley n.º 27293 - ley del sistema nacional de inversión pública. \LaTeX : Ley del Sistema Nacional de Inversión Pública. Disponible en: <https://www.mef.gob.pe/es/snip/ley-del-sistema-nacional-de-inversion-publica>.
- Fernandes, L. C., Silva, M. L., and Lima, P. M. (2022). Pavement distress detection using deep convolutional neural networks: A review. *Automation in Construction*, 137:104224.
- Forsyth, D. A. and Ponce, J. (2003). *Computer Vision: A Modern Approach*. Prentice Hall, Upper Saddle River, NJ.
- Garber, N. J. and Hoel, L. A. (2009). *Principles of Highway Engineering and Traffic Analysis*. Cengage Learning, 4 edition.
- Gerard Zwetsloot, S. L. y. P. K. (2017). *Vision zero: from accident prevention to the promotion of health, safety and well-being at work*. Institution of Occupational Safety and Health, United Kingdom.
- Gilliland, A. J. (2008). *Setting the Stage: An Introduction to Metadata*. Getty Research Institute, Los Angeles.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press, Cambridge, MA.
- Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media, Inc.

- Guida, F. P. (2023). Generación de datos artificiales (data augmentation). <https://franspg.dev/2020/01/27/generacion-de-datos>.
- Géron, A. (2019a). *Hands-On Machine Learning with Scikit-Learn, Keras, and Tensor-Flow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Sebastopol, CA, 2nd edition.
- Géron, A. (2019b). *Hands-On Machine Learning with Scikit-Learn, Keras, and Tensor-Flow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Sebastopol, CA, 2nd edition.
- Hernández Sampieri, C. R. (1997). *Metodología de la Investigación*. Panamericana Formas e Impresos S.A., Colombia.
- Heron-Project (2021). Heron project. https://www.heron-h2020.eu/?page_id=15.
- Huang, G., Liu, Z., van der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.
- Huansheng, S., Haoxiang, L., Huaiyu, L., Zhe, D., and Xu, Y. (2019). *Vision-based vehicle detection and counting system using deep learning in highway scenes*. European Transport Research Review.
- Huerta, E., Mangiaterra, A., and Noguera, G. (2005). *GPS - Posicionamiento Satelital*. Universidad nacional de rosario - Argentina.
- INDEED (2023). Salario desarrollador de software. <https://pe.indeed.com/career/desarrollador-de-software/salaries>.
- Instructivo, M. (2018). *Manual de carreteras de mantenimiento o conservacion vial*. Ministerio de transporte y comunicaciones, peru.
- IPE (2008). A lo que lleva la falta de mantenimiento de carreteras. <https://ipe.org.pe/new/comentario-diario/23-7-2008>.
- Johansson, R. (2009). *Vision Zero – Implementing a policy for traffic safety*. Road Safety Division - El Servier, Sweden.
- JSExpress (2023). Express - infraestructura web rápida, minimalista y flexible para node.js. <https://expressjs.com/es/>.
- Kaplan, E. D. and Hegarty, C. (2006). *Understanding GPS: Principles and Applications*. Artech House.
- KAYAK (2023). Alquileres de autos cusco. https://www.kayak.com.pe/cars/Cusco,Peru-c12159/2023-07-06/2023-07-10;map?sort=rank_a.
- Kong, W., Zhong, T., Mai, X., Zhang, S., Chen, M., and Lu, G. (2022). *Automatic Detection and Assessment of Pavement Marking Defects with Street View Imagery at the City Scales*. Multidisciplinary Digital Publishing Institute - MDPI.
- LeaseIN (2023). Alquiler laptop. <https://leasein.pe/>.

- LeCun, Y., Bengio, Y., and Hinton, G. (2020). Deep learning. *Nature*, 521(7553):436–444.
- Lutz, M. (2013). *Learning Python*. O’Reilly Media.
- Medium, R. A. (2023). Qué es underfitting y overfitting). <https://rubialesalberto.medium.com/qu%C3%A9-es-underfitting-y-overfitting-c73d51ffd3f9>.
- MTC (2022). Asignacion de presupuesto mtc-2023. <https://www.gob.pe/institucion/mtc/noticias/675279-ministro-richard-tineo-sustento-presupuesto>.
- MTC (2023). *Plan Operativo Institucional Anual 2023*. Ministerio de transportes y comunicaciones, Perú.
- Municipalidad provincial del Cusco (2013). *ROF-Reglamento de Organizacion y Funciones*. Municipalidad provincial del Cusco, Cusco.
- Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press, Cambridge, MA.
- OMS (2020). Principales causas de defunción. <https://www.who.int/es/news-room/fact-sheets/detail/the-top-10-causes-of-death#:~:text=La%20mayor%20causa%20de%20defunci%C3%B3n,8%2C9%20millones%20en%202019>.
- OMS (2022). Traumatismos causados por el tránsito. <https://www.who.int/es/news-room/fact-sheets/detail/road-traffic-injuries>.
- Ortega Triana, J. A. (2021). *Aprendizaje profundo para la detección automática de fisuras de hormigón usando redes neuronales convolucionales*. Universidad Politécnica de Valencia, España.
- Peru, C. (2023). Precio del combustible en cusco. <https://www.preciocombustible.com/cusco/>.
- Portal oficial del MTC (2023). Ministerio de transportes y comunicaciones. <https://portal.mtc.gob.pe/proyectos.html>.
- PostgreSQL (2023). PostgreSQL: la base de datos relacional de código abierto más avanzada del mundo. <https://www.postgresql.org/>.
- Pressman, R. S. and Maxim, B. R. (2014). *Ingeniería del software: un enfoque práctico*. McGraw-Hill Education, 7 edition.
- Raschka, S. and Mirjalili, V. (2019). *Python Machine Learning*. Packt Publishing, Birmingham, UK, 3rd edition.
- React (2023). React-una biblioteca de javascript para construir interfaces de usuario. <https://es.legacy.reactjs.org/>.
- Romero, M. M. (2022). *Reconocimiento de objetos mediante tecnicas de vision por computador y aprendizaje automatico*. Universidad Publica de Sevilla - España.
- Rouhiainen, L. (2018). *Inteligencia Artificial*. Alienta Editorial, España.

- Salas, R. (2023). *Redes Neuronales Artificiales*. Departamento de Computación, Universidad de Valparaíso.
- SAS (2023). Red neuronal artificial. http://https://www.sas.com/es_pe/insights/analytics/deep-learning.html.
- Smith, J. and Doe, J. (2021). Challenges in data labeling for artificial intelligence applications. *Journal of AI Research*, 45(3):123–135.
- Sommerville, I. (2011). *Ingeniería del software*. Pearson Educación, 9 edition.
- StreetsBlogUSA (2020). Vision zero! norwegian capital completely quashes road deaths. <https://usa.streetsblog.org/2020/01/03/vision-zero-norwegian-capital-completely-quashes-road-deaths/>.
- SUTRAN (2020). *Reglamento Nacional de Administracion de transporte - Decreto Supremo N°017-2009-MTC*. Ministerio de Transportes y Comunicaciones, Peru.
- Towardsdatascience (2023). Datasets. <https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7>.
- Visual (2023). Alquiler de equipos de video. <https://tallerdefotografia.edu.pe/estudio/alquiler-equipos.html>.
- Zhou, Y., Guo, X., Hou, F., and Wu, J. (2022). *Review of Intelligent Road Defects Detection Technology*. Multidisciplinary Digital Publishing Institute - MDPI.