



**PORTADA**  
**UNIVERSIDAD TÉCNICA DE AMBATO**

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E  
INDUSTRIAL**

**CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN**

**Tema:**

---

**APLICACIÓN MÓVIL UTILIZANDO APRENDIZAJE AUTOMÁTICO  
PARA MITIGAR PÉRDIDAS ECONÓMICAS AL ADQUIRIR VEHÍCULOS  
DE SEGUNDA MANO EN LA CIUDAD DE AMBATO.**

---

Trabajo de titulación modalidad Proyecto de Investigación, presentado previo a la obtención del título de Ingeniero en Tecnologías de la Información.

**ÁREA:** Gestión de tecnología de la información

**LÍNEA DE INVESTIGACIÓN:** Tecnología de la información y Sistema de control

**AUTOR:** Carlos Mauricio Vasconez Lozada

**TUTOR:** Ing. Rubén Eduardo Nogales Portero, Mg.

**Ambato - Ecuador**

**febrero – 2024**

## **APROBACIÓN DEL TUTOR**

En calidad de tutor del trabajo de titulación con el tema: APLICACIÓN MÓVIL UTILIZANDO APRENDIZAJE AUTOMÁTICO PARA MITIGAR PÉRDIDAS ECONÓMICAS AL ADQUIRIR VEHÍCULOS DE SEGUNDA MANO EN LA CIUDAD DE AMBATO, desarrollado bajo la modalidad Proyecto de Investigación por el señor Carlos Mauricio Vasquez Lozada, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 17 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.3 del instructivo del reglamento referido.

Ambato, febrero 2024.

-----  
Ing. Rubén Eduardo Nogales Portero, Mg.

TUTOR

## AUTORÍA

El presente trabajo de titulación con el tema: APLICACIÓN MÓVIL UTILIZANDO APRENDIZAJE AUTOMÁTICO PARA MITIGAR PÉRDIDAS ECONÓMICAS AL ADQUIRIR VEHÍCULOS DE SEGUNDA MANO EN LA CIUDAD DE AMBATO es absolutamente original, auténtico y personal y ha observado los preceptos establecidos en la Disposición General Quinta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, febrero 2024.



Carlos Mauricio Vasconez Lozada

C.C 1850270420

AUTOR

## DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato para que reproduzca total o parcialmente este trabajo de titulación dentro de las regulaciones legales e institucionales correspondientes. Además, cedo todos mis derechos de autor a favor de la institución con el propósito de su difusión pública, por lo tanto, autorizo su publicación en el repositorio virtual institucional como un documento disponible para la lectura y uso con fines académicos e investigativos de acuerdo con la Disposición General Cuarta del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato.

Ambato, febrero 2024.



---

Carlos Mauricio Vasconez Lozada

C.C. 1850270420

AUTOR

## **APROBACIÓN DEL TRIBUNAL DE GRADO**

En calidad de par calificador del informe final del trabajo de titulación presentado por el señor Carlos Mauricio Vasconez Lozada, estudiante de la Carrera de Tecnologías de la Información, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, bajo la Modalidad Proyecto de Investigación, titulado **APLICACIÓN MÓVIL UTILIZANDO APRENDIZAJE AUTOMÁTICO PARA MITIGAR PÉRDIDAS ECONÓMICAS AL ADQUIRIR VEHÍCULOS DE SEGUNDA MANO EN LA CIUDAD DE AMBATO**, nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 19 del Reglamento para la Titulación de Grado en la Universidad Técnica de Ambato y el numeral 6.4 del instructivo del reglamento referido. Para cuya constancia suscribimos, conjuntamente con la señora Presidente del Tribunal.

Ambato, febrero 2024.

-----  
Ing. Elsa Pilar Urrutia Urrutia, Mg.  
PRESIDENTE DEL TRIBUNAL

-----  
Ing. Mg. Hernán Naranjo Avalos

PROFESOR CALIFICADOR

-----  
PhD. Ing. Félix Fernández Peña

PROFESOR CALIFICADOR

## DEDICATORIA

*El presente trabajo de investigación está dedicado a Dios en primer lugar por haberme brindado la fortaleza, salud y sabiduría para poder culminar mis estudios universitarios.*

*A mi papá Carlos que desde el cielo ha sido mi sendero y mi recordatorio que nunca debo dejar de luchar por mis sueños.*

*A mi hermano Marcelo por haber sido como un segundo padre que ha sabido escucharme, guiarme y corregirme mis errores para convertirme en un hombre de bien.*

*A mi mamá Olga, mis hermanas Gladys, Fabiola y sobrina Cristina por haberme brindado su apoyo, cariño y atención en este proceso y a lo largo de mi vida.*

## AGRADECIMIENTO

*Agradezco principalmente a Dios por haberme dado licencia de vida, por ser mi refugio de paz. Gracias Dios mío por ser mi luz y mi aparo en mis momentos de angustia.*

*A mi familia infinitas gracias por apoyarme, ser mi fortaleza y mi motivación de cada día en especial a la Señora Elvia Lizano por preocuparse por mí.*

*A los docentes que han sido parte de mi proceso y formación académica muchas gracias por haber compartido sus enseñanzas que dejaron una huella indeleble en mi camino.*

*A mi tutor académico el Ingeniero Rubén Nogales por ser la persona que ha sabido guiarme y apoyarme en el desarrollo mi trabajo de investigación.*

## ÍNDICE GENERAL DE CONTENIDOS

<b>PORTADA</b> .....	<b>i</b>
<b>APROBACIÓN DEL TUTOR</b> .....	<b>ii</b>
<b>AUTORÍA</b> .....	<b>iii</b>
<b>DERECHOS DE AUTOR</b> .....	<b>iv</b>
<b>APROBACIÓN DEL TRIBUNAL DE GRADO</b> .....	<b>v</b>
<b>DEDICATORIA</b> .....	<b>vi</b>
<b>AGRADECIMIENTO</b> .....	<b>vii</b>
<b>ÍNDICE GENERAL DE CONTENIDOS</b> .....	<b>viii</b>
<b>ÍNDICE DE TABLAS</b> .....	<b>xii</b>
<b>ÍNDICE DE FIGURAS</b> .....	<b>xiv</b>
<b>ÍNDICE DE ANEXOS</b> .....	<b>xvii</b>
<b>RESUMEN EJECUTIVO</b> .....	<b>xix</b>
<b>ABSTRACT</b> .....	<b>xx</b>
<b>CAPÍTULO I. MARCO TEÓRICO</b> .....	<b>1</b>
1.1 Tema de investigación.....	1
1.1.1 Planteamiento del problema.....	1
1.2 Antecedentes investigativos .....	2
1.3 Fundamentación teórica .....	10



1.3.1 Matemáticas Aplicadas .....	10
1.3.2 Ciencias de la Computación.....	11
1.3.3 Inteligencia Artificial .....	12
1.3.4 Machine Learning .....	13
1.3.5 Visión Artificial .....	14
1.3.6 Redes Neuronales.....	14
1.3.7 Aplicaciones Móviles.....	15
1.3.8 Comportamiento Humano.....	15
1.3.9 Ciencias Sociales.....	15
1.3.10 Economía.....	16
1.3.11 Pérdidas Económicas .....	16
1.4 Objetivos .....	16
1.4.1 Objetivo general .....	16
1.4.2 Objetivos específicos .....	16
<b>CAPÍTULO II. METODOLOGÍA.....</b>	<b>17</b>
2.1 Materiales.....	17
2.1.1 Encuesta .....	17
2.2 Métodos.....	21

2.2.1 Modalidad de la investigación .....	21
2.2.2 Población y muestra .....	22
2.2.3 Recolección de información.....	23
2.2.4 Procesamiento y análisis de datos .....	39
<b>CAPÍTULO III. RESULTADOS Y DISCUSIÓN.....</b>	<b>41</b>
3.1 Análisis y discusión de resultados.....	41
3.1.1 Requerimientos funcionales y no funcionales .....	41
3.1.2 Análisis de bibliotecas de aprendizaje automático para dispositivos móviles..	42
3.1.3 Entornos de desarrollo para aprendizaje automático .....	46
3.1.4 Tecnologías para el desarrollo móvil .....	47
3.1.5 Metodología para el desarrollo móvil .....	48
3.2 Desarrollo de los modelos de aprendizaje automático .....	49
3.2.1 Proceso de segmentación, detección y clasificación de la placa vehicular.....	51
3.2.2 Proceso de clasificación de irregularidades del exterior del vehículo .....	68
3.3 Desarrollo de la aplicación móvil .....	71
3.3.1 Fase 1: Exploración.....	71
3.3.2 Fase 2: Inicialización.....	73
3.3.3 Fase 3: Producción .....	85
3.3.4 Fase 4: Estabilización.....	94
3.3.5 Fase 5: Pruebas.....	94
3.4 Muestra para aplicar en el testeo .....	102

3.4.1 Recolección de la información.....	102
<b>CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>108</b>
4.1 Conclusiones .....	108
4.2 Recomendaciones.....	109
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>110</b>
<b>ANEXOS.....</b>	<b>114</b>

## ÍNDICE DE TABLAS

Tabla 1. Cadenas de búsqueda .....	2
Tabla 2. Criterios de inclusión y exclusión .....	4
Tabla 3. Escala le Likert.....	17
Tabla 4. Interpretación de la escala de consistencia del alfa de Cronbach .....	24
Tabla 5. Resultados pregunta a .....	24
Tabla 6. Resultados pregunta b .....	25
Tabla 7. Resultados pregunta c .....	26
Tabla 8. Resultados pregunta d .....	27
Tabla 9. Resultados pregunta e .....	28
Tabla 10. Resultados pregunta f.....	29
Tabla 11. Resultados pregunta g .....	30
Tabla 12. Resultados pregunta h .....	31
Tabla 13. Resultados pregunta i .....	32
Tabla 14. Resultados pregunta j .....	33
Tabla 15. Resultados pregunta k .....	34
Tabla 16. Resultados pregunta l.....	35
Tabla 17. Resultados pregunta m .....	36
Tabla 18. Resultados pregunta n .....	37
Tabla 19. Resultados pregunta o .....	38
Tabla 20. Requerimientos funcionales .....	41

Tabla 21. Requerimientos no funcionales .....	42
Tabla 22. Características técnicas de las bibliotecas de aprendizaje automático para dispositivos móviles .....	44
Tabla 23. Comparación de entornos de desarrollo de aprendizaje automático.....	46
Tabla 24. Comparación entre tecnologías de desarrollo móvil.....	47
Tabla 25. Metodologías ágiles para desarrollo móvil .....	49
Tabla 26. Arquitecturas de segmentación .....	53
Tabla 27. comparativa de las arquitecturas YOLO y SSD-MobileNetV2.....	59
Tabla 28. Establecimiento de los interesados .....	72
Tabla 29. Planificación de fases .....	74
Tabla 30. Storycard - Pantalla principal.....	89
Tabla 31. Storycard – Inspección legal del vehículo .....	89
Tabla 32. Storycard – Guardar propietario .....	91
Tabla 33. Storycard – Pantalla home .....	92
Tabla 34. Storycard – Editar Propietario .....	92
Tabla 35. Storycard – Inspección externa del vehículo .....	93
Tabla 36. Storycard – Pantalla about .....	94
Tabla 37. Escala Likert para testear la aplicación.....	103

## ÍNDICE DE FIGURAS

Figura 1. Tipos de Aprendizaje [21] .....	13
Figura 2. Red Neuronal [22] .....	14
Figura 3. Resultados pregunta a .....	25
Figura 4. Resultados Pregunta b.....	26
Figura 5. Resultados pregunta c .....	27
Figura 6. Resultados pregunta d.....	28
Figura 7. Resultados pregunta e .....	29
Figura 8. Resultados pregunta f .....	30
Figura 9. Resultados pregunta g.....	31
Figura 10. Resultados pregunta h.....	32
Figura 11. Resultados pregunta i.....	33
Figura 12. Resultados pregunta j.....	34
Figura 13. Resultados pregunta k.....	35
Figura 14. Resultados pregunta l.....	36
Figura 15. Resultados pregunta m.....	37
Figura 16. Resultados pregunta n.....	38
Figura 17. Resultados pregunta o.....	39
Figura 18. Esquema de operatividad de los modelos .....	50
Figura 19. Segmentación mediante el software Labelme .....	53
Figura 20. Arquitectura ResUNet [41].....	54

Figura 21. Métricas de pérdida en entrenamiento y validación .....	56
Figura 22. Métricas de precisión en entrenamiento y validación .....	56
Figura 23. Métricas de pérdida en entrenamiento y validación con data augmentation .....	57
Figura 24. Métricas de precisión en entrenamiento y validación con data augmentation .....	58
Figura 25. Cuadros delimitadores con Labelme .....	59
Figura 26. Transfer Learning con SSD-MobileNetV2 para detección de caracteres. 60	
Figura 27. Bloque Inverted Residual de MobileNetV2 .....	61
Figura 28. Métrica de precisión en el entrenamiento.....	62
Figura 29. Métrica de pérdida en el entrenamiento.....	62
Figura 30. Arquitectura de la CNN .....	63
Figura 31. Métricas de precisión en entrenamiento y validación de la CNN .....	65
Figura 32. Métricas de pérdida en entrenamiento y validación de la CNN .....	65
Figura 33. Matriz de confusión .....	66
Figura 34. Flujo de trabajo de los modelos aprendizaje automático encascada.....	67
Figura 35. Data Augmentation.....	69
Figura 36. Arquitectura de Red Neuronal Convolutiva .....	69
Figura 37. Métricas de precisión en el entrenamiento .....	70
Figura 38. Métricas de pérdida en entrenamiento .....	71
Figura 39. Esquema de funcionamiento de la aplicación.....	75
Figura 40. Base de Datos Relacional .....	76

Figura 41. Esquema de navegabilidad .....	77
Figura 42. Diagrama de caso de uso .....	78
Figura 43. Pantalla principal .....	79
Figura 44. Pantalla home.....	80
Figura 45. Inspección legal del vehículo.....	81
Figura 46. Información del vehículo.....	82
Figura 47. Diálogo guardar propietario.....	83
Figura 48. Inspección externa .....	84
Figura 49. Información de la app .....	85
Figura 50. Arquitectura de la aplicación móvil con MVVM y clean Architecture....	87
Figura 51. Integración de los modelos tflite.....	88
Figura 52. Pantalla home.....	95
Figura 53. Inspección legal .....	96
Figura 54. Información del vehículo.....	97
Figura 55. Guardar Propietario .....	98
Figura 56. Actualizar propietario .....	99
Figura 57. Inspección estética del vehículo .....	100
Figura 58. Pantalla About .....	101
Figura 59. Grado de satisfacción de los usuarios .....	105
Figura 60. Media por pregunta.....	106



## ÍNDICE DE ANEXOS

Anexo A. Alfa de Cronbach.....	114
Anexo B. Script de conversión XML a Imagen binaria.....	115
Anexo C. Funcion de bloques residuales .....	116
Anexo D. Función de decodificación.....	117
Anexo E. Arquitectura ResUNet .....	118
Anexo F. Clase DataGenerator de imágenes dinamicas .....	119
Anexo G. Data Augmentation.....	121
Anexo H. Transfer Learning SDD-MobileNetV2.....	123
Anexo I. Preprocesamiento del dataset Chars74K.....	125
Anexo J. Arquitectura de la Red Neuronal Convolutacional.....	126
Anexo K. Conversión de Modelos a modelos TensorFlow Lite.....	127
Anexo L. Data Augmentation dataset .....	128
Anexo M. Arquitectura de Red Neuronal Convolutacional.....	130
Anexo N. Archivo de configuración Spring Boot.....	131
Anexo O. Dependencias Android .....	132
Anexo P. Archivo XML de la pantalla principal .....	133
Anexo Q. Configuracion de la cámara.....	134
Anexo R. Listener para escuchar eventos entre clases.....	135
Anexo S. Clase DetectPlateHelper.....	136
Anexo T. Preprocesamiento de la imagen.....	137

Anexo U. Método de conversión de a un bitmap.....	138
Anexo V. Operaciones con Opencv para recorta la placa.....	139
Anexo W. Método detectObjects .....	140
Anexo X. Recorte de los caracteres .....	141
Anexo Y. Clasificación de caracteres .....	142
Anexo Z. Retorno de la placa al hilo principal .....	143
Anexo 1A. Diálogo para guardar el propietario.....	144
Anexo 2B. Guardar propietario.....	145
Anexo 3C. Método getAllPropietarios.....	146
Anexo 4D. Editar propietario.....	147
Anexo 5E. Método DetectStetic.....	148
Anexo 6F. Clase DetectSteticHelper.....	149
Anexo 7G. Inferencia del modelo de detección de anomalías estéticas .....	150
Anexo 8H. Pantalla about .....	151
Anexo 9I. Alfa Cronbach validación cuestionario test de la app.....	152
Anexo 10J. Manual de Usuario.....	153

## RESUMEN EJECUTIVO

La venta de vehículos usados es uno de los pilares de la economía de muchas familias ambateñas y de la región centro del país, a lo largo de los años se han ingeniado diversos métodos de estafa y engaño en varias actividades económicas y el sector del comercio automotriz no ha sido la excepción. Se han observado diversos métodos de estafa y engaño que sufren los compradores, vendedores o personas en particular si no se toman las medidas necesarias para realizar una inspección minuciosa en los aspectos legales, estéticos y mecánicos del vehículo lo que resultaría un desgaste económico.

El presente trabajo de investigación tiene como objetivo desarrollar una aplicación móvil Android con inteligencia artificial para las personas que acuden a las ferias de vehículos usados. Su propósito es brindar información precisa y genuina sobre los aspectos legales y estéticos del vehículo para mitigar posibles pérdidas económicas.

La aplicación móvil se desarrolló utilizando el lenguaje programación nativo Kotlin. La aplicación móvil integra modelos de inteligencia artificial utilizando TensorFlow Lite para permitir inferencia en el mismo dispositivo móvil. Además, para el proceso de entrenamiento se utilizó conjuntos de datos propietarios. Por otra parte, el desarrollo de la aplicación se lo construyó utilizando el patrón arquitectónico Model-View-ViewModel conjuntamente con Clean Architecture para lograr separar claramente las responsabilidades y el diseño facilitando la evolución y el mantenimiento de la aplicación.

Como resultado de esta investigación, se logró obtener una aplicación móvil que asista a las personas participantes en el comercio automotriz a mitigar posibles pérdidas económicas al adquirir vehículos usados.

**Palabras clave:** Inteligencia artificial, Android, aplicación móvil, automotriz, TensorFlow Lite, pérdidas económicas.

## ABSTRACT

The sale of used vehicles is one of the pillars of the economy of many Ambateña families and the central region of the country. Over the years, various methods of scam and deception have been devised in various economic activities and the automotive trade sector. has not been the exception. Various methods of scam and deception have been observed that buyers, sellers or individuals in particular suffer if the necessary measures are not taken to carry out a thorough inspection of the legal, aesthetic and mechanical aspects of the vehicle, which would result in financial waste.

The objective of this research work is to develop an Android mobile application with artificial intelligence for people who attend used vehicle fairs. Its purpose is to provide accurate and genuine information on the legal and aesthetic aspects of the vehicle to mitigate possible financial losses.

The mobile application was developed using the native Kotlin programming language. The mobile application integrates artificial intelligence models using TensorFlow Lite to enable inference on the same mobile device. In addition, proprietary data sets were used for the training process. On the other hand, the development of the application was built using the Model-View-ViewModel architectural pattern together with Clean Architecture to clearly separate responsibilities and design, facilitating the evolution and maintenance of the application.

As a result of this research, it was possible to obtain a mobile application that assists participants in the automotive trade to mitigate possible economic losses when purchasing used vehicles.

**Keywords:** Artificial intelligence, Android, mobile application, automotive, TensorFlow Lite, economic losses

# CAPÍTULO I. MARCO TEÓRICO

## 1.1 Tema de investigación

APLICACIÓN MÓVIL UTILIZANDO APRENDIZAJE AUTOMÁTICO PARA MITIGAR PÉRDIDAS ECONÓMICAS AL ADQUIRIR VEHÍCULOS DE SEGUNDA MANO EN LA CIUDAD DE AMBATO.

### 1.1.1 Planteamiento del problema

Texto El sector automotriz es una de las industrias que se ha caracterizado por mantener procesos de reestructuración constante, transformándose en una de las industrias con mayor dinamismo a nivel mundial. El ámbito automotriz tiene un impacto considerable en la economía de los países en términos de productividad, competitividad y avance tecnológico. Asimismo, representa una de las principales fuentes de empleo, tanto directo como indirecto en los sectores de fabricación, comercialización y reventa de vehículos nuevos y usados.[1]

Según [2], la industria automotriz se localiza en los países industrializados, pero cabe señalar que en Latinoamérica el incremento del parque automotor, ha ampliado la problemática de tráfico vehicular, accidentes y conductores que infringen la ley. Como consecuencia, los vehículos que participan en los anteriores hechos mencionados se ven implicados en problemas legales y estéticos que influyen al momento de comercializar el vehículo. Al comercializar este tipo de vehículos que conllevan problemas legales y estéticos afectara el estado económico del cliente o persona que adquiriera el automotor y en casos excepcionales se puede perder por completo el vehículo.

En [1] menciona que en el Ecuador el sector automotriz es una de las fuentes fundamentales de la economía localizada mayoritariamente en el centro norte de la región sierra, siendo 198 millones de dólares que produce anual el Ecuador. En el mismo sentido en [3] menciona que el sector automotriz genera ingresos mediante las actividades directas e indirectas como la comercialización de vehículos de segunda

mano. La compra venta de vehículos usados es una actividad más que se desarrolla en el país para fundamentar la economía del país.

En el caso de la provincia del Tungurahua específicamente en la ciudad de Ambato, la compra venta de vehículos es una actividad económica de gran importancia para la ciudad. En consecuencia, se ha convertido en el centro de distribución de vehículos usados para todo el país, por lo tanto, la provincia del Tungurahua ocupa del cuarto lugar entre las provincias donde más se compra vehículos livianos según la Asociación de Empresas Automotrices del Ecuador (AEADE) [4]. Estas actividades de comercio se ven involucradas muy a menudo en actividades de estafa como la comercialización de vehículos implicados en robos, problemas legales, alteraciones estéticas o multas pendientes lo que deriva en una pérdida económica para la persona adquiriente.[5]

## 1.2 Antecedentes investigativos

En esta investigación, se llevó a cabo una revisión de la literatura en artículos publicados en las siguientes bases de datos científicas: Scopus, IEEE Xplore, ACM Digital Library, Willey y Springer. Las cadenas de búsqueda se definieron con las siguientes palabras claves: Machine Learning, vehicle license plate, shades colors, crashed cars, economic losses y mobile devices en el rango de los años 2019 – 2023. Por lo consiguiente, la Tabla 1. Muestra las cadenas de búsqueda utilizadas y el número de artículos encontrados.

Tabla 1. Cadenas de búsqueda

	Revista	Cadena	N Artículo
<b>Cadena 1</b>	Scopus	( TITLE-ABS-KEY ( "machine learning" ) AND TITLE-ABS-KEY ( " vehicle license plate" ) OR TITLE-ABS-KEY ( "crashed cars" ) OR TITLE-ABS-KEY ( "scraped cars" ) AND TITLE-ABS-KEY ( "used vehicle" ) OR TITLE-ABS-KEY ( "used cars" ) AND TITLE-ABS-KEY ( "economic losses" )) AND LIMIT-TO ( PUBYEAR , 2022 ) OR LIMIT-TO ( PUBYEAR , 2021 ) OR LIMIT-TO ( PUBYEAR , 2020 ) OR LIMIT-TO ( PUBYEAR , 2019 ) )	34

	Revista	Cadena	N Artículo
<b>Cadena 1</b>	IEEE Xplore	("All Metadata": "machine learning" AND ("All Metadata": vehicle license plate) ("All Metadata": "crashed cars") AND ("All Metadata": "scraped cars") AND ("All Metadata": used vehicles) OR ("All Metadata": used cars) OR ("All Metadata": economic losses)	8107
	ACM Digital Library	[All: "machine learning"] AND [All: "vehicle license plate"] AND [All: "crashed cars"] AND [All: scraped cars] OR [All: used vehicles] OR [All: used cars] AND [All: economic losses] AND [E-Publication Date: (01/01/2019 TO 12/31/2023)]	10552
	Wiley	"machine learning " AND "vehicle license plate" AND crashed cars OR scraped cars AND used vehicle AND used cars AND economic losses	6
	Science Direct	" machine learning " AND " vehicle license plate" AND crashed cars OR scraped cars AND used vehicle OR used cars AND economic losses	98
	Springer	" machine learning " AND " vehicle license plate" AND crashed cars OR scraped cars AND used vehicle OR used card AND economic losses	48
	Revista	Cadena	N Artículo
<b>Cadena 2</b>	Scopus	( TITLE-ABS-KEY ( "machine learning" ) AND TITLE-ABS-KEY ( "used cars" ) OR TITLE-ABS-KEY ( "used vehicles " ) AND TITLE-ABS-KEY ( "mobile devices" ) ) AND ( LIMIT-TO ( PUBYEAR , 2022 ) OR LIMIT-TO ( PUBYEAR , 2021 ) OR LIMIT-TO ( PUBYEAR , 2020 ) OR LIMIT-TO ( PUBYEAR , 2019 ) )	50
	IEEE Xplore	("All Metadata":machine learning) AND ("All Metadata":used cars) OR ("All Metadata":used vehicles) AND ("All Metadata":mobile devices )	11874
	ACM Digital Library	[Title: "machine learning"] AND [All: used cars] OR [Title: used vehicles] AND [Title: mobile devices] AND [E-Publication Date: (01/01/2019 TO 05/31/2023)]	551
	Wiley	"machine learning" in Title and "used cars" in Title and "used vehicles" in Title and "mobile devices" in Title	5
	Science Direct	"machine learning" AND used cars OR used vehicles AND "mobile devices"	228
	Springer	"machine learning" AND used cars OR used vehicles AND "mobile devices"	279
<b>Total</b>			31832

En función de las cadenas de búsqueda establecido en la Tabla 1 se encontraron 31.832 artículos para lo cual se establecieron criterios de inclusión y exclusión para seleccionar a los artículos más pertinentes para esta investigación como se puede observar en la Tabla 2.

Tabla 2. Criterios de inclusión y exclusión

<b>Inclusión</b>	- Artículos que usan algoritmos de aprendizaje automatico para detección de placas vehiculares.
	- Artículos que emplean modelos de aprendizaje automatico para estimar precio del vehículo basados en aspectos estéticos.
	- Artículos que se enfocan en la implantación de modelos aprendizaje automatico en dispositivos móviles.
<b>Exclusión</b>	- No describen sus técnicas ni resultados de validación.
	- Artículos que poseen limitaciones lingüísticas.
	- Artículos que se relacionan con los vehículos, pero en tienen otros objetivos.

El fin de la revisión de literatura es describir los modelos de aprendizaje automático para detectar placas y alteraciones estéticas del vehículo mediante dispositivos móviles. En este contexto, al revisar la literatura se encontraron un total 31.832 artículos, los cuales fueron investigados en búsqueda de las palabras clave establecidas. En base a la revisión literaria y la Tabla 2, se descartaron 16.395 debido a que poseen limitaciones lingüísticas dejando un total 15.437 artículos restantes. Posteriormente, se excluyeron aquellos que no estaban relacionados con la aplicación de algoritmos de aprendizaje automático en dispositivos móviles, resultando un total de 221 artículos. De estos, se descartaron 185 debido a que estaban enfocados en la aplicación de modelos de aprendizaje automático en otros campos relacionados con la automatización de vehículos y no describían las metodologías y validación de resultados. Finalmente, tras este análisis, quedaron 36 artículos. De los 36, se trabajó con 11 artículos debido a que, describen con certeza las metodologías y resultados de la investigación que son necesarios para el objeto de estudio de esta investigación.

El estudio realizado por N. Khaled, S. Mohsen, K. Emad, S. Akram, H. Metawie y A. Mohamed [6] proponen implementar una aplicación móvil usando Redes Neuronales Convolucionales (CNN) y Tensor Flow Lite. Los autores utilizan una técnica de aprendizaje profundo para la detección de objetos, el reconocimiento de texto a voz mediante una CNN previamente entrenada en un servidor en la nube (Google Colab) con un conjunto de datos Common Objects in Context (Coco), obteniendo un precisión



del 92% en el entrenamiento y en el testeo de 90%, una vez entrenado los modelos se emplea TensorFlow Lite que es una herramienta que convierte el modelo entrenado en un modelo más ligero y accesible para dispositivos móviles. En entorno al conjunto de datos es vital crear TFRecords para acortar el proceso de entrenamiento ya que almacena los datos de forma compacta y eficiente por lo tanto optimiza para su uso con redes neuronales. En conclusión, el software desarrollado devuelve una guía de navegación para llegar al objetivo específico mientras detecta obstáculos con el fin de respaldar el proceso localización de objetos específicos para personas con discapacidad visual. La propuesta por estos autores es muy interesante ya que aplican técnicas y metodologías para hacer el modelo más ligero y accesible, y por último aplican Tensor Flow Lite que es una librería que hace posible la implantación de modelos de aprendizaje de manera que se pueden ejecutar en dispositivos de bajos recursos de hardware.

A. Benhamida, A. Varkonyi-Koczy y M. Kolovszky [7] proponen una aplicación móvil para la detección de señales de tránsito mediante técnicas de transferencia de aprendizaje profundo. El modelo planteado en este estudio se basa en MobileNet v2 que aplica una única red profunda para entrenar el modelo utilizando imágenes de señales de tránsito obteniendo resultados del 92% de precisión en el entrenamiento y 93% en la validación, el modelo resultante del entrenamiento se lleva a cabo un proceso de conversión en un modelo cuantificado de TensorFlow Lite usando el convertidor de TensorFlow que proporciona compatibilidad con dispositivos móviles con baja capacidad computacional. Una vez finalizado el proceso de conversión es llevado una aplicación de Android Studio. En conclusión, la aplicación desarrollada para reconocimientos de señales de tránsito promete resultados muy favorables gracias a la aplicación de TensorFlow Lite conjuntamente con la aplicación del modelo MobileNet v2 que ha demostrado una compatibilidad alta en la implementación en dispositivos móviles.

El trabajo realizado por E. Dorter [8] propone una aplicación móvil para detección de comida turca aplicando técnicas de transferencia de aprendizaje profundo en plataformas móviles, el trabajo utiliza las bibliotecas Keras, TensorFlow y utilizando el lenguaje de programación Python en Google Colab. La CNN propuesta es Inceptionv3 ya que presento un 88% de precisión al momento de clasificar a diferencia

de otras arquitecturas planteadas como ResNet50, Xception e InceptionResnetV2, las funciones de activación es un factor importante, para ello se establece las siguientes funciones de activación (ReLU, Tanh y Softmax) y además proponen los algoritmos de optimización (Adam, Adadelta y SGD) donde el algoritmo Gradiente de Gradiente Estocástico (SGD) con una función de activación Softmax presentaron resultados más prometedores. El proceso de conversión se realizó con el uso de la biblioteca Coremltools Python para convertir el modelo entrenado por keras en un modelo. hdf5 al formulario CoreML que es la biblioteca de aprendizaje automatico de Apple. La importancia de este estudio se basa en exponer la importancia de las funciones de activación en cada capa, además exponen los 3 métodos de optimización de una CNN y por último explican cómo llevar un modelo de aprendizaje a un entorno iOS.

Según S. Kunjumon, R. Nandakumar y C. Sruthy [9] propone un modelo de clasificación de imágenes, reconocimiento de objetos y seguimiento de imágenes empleando varios modelos de aprendizaje profundo denominado Redes Neuronales Convolucionales (CNN) para la clasificación y detección de daños en vehículos principalmente se centra en clasificar 7 posibles opciones como alteraciones en parachoques, puertas, faros e incluso rayones entre otros aspectos. Para una mejor precisión, utiliza un método para la extracción de las partes dañadas del vehículo, esta información de daños se transfiere a una red clasificadora previamente entrenada. Los modelos de aprendizaje profundo que experimentaron en esta investigación son: InceptionV3, Xception, VGG16, VGG19, ResNet50 y MobileNet de la biblioteca de Keras. En conclusión, los autores determinan el algoritmo que satisface tanto en la precisión como en tiempo de procesamiento es el algoritmo MobileNet V2 dado que identifico y categorizo imágenes con un 97,28% de precisión con respecto a los demás algoritmos plateados. La importancia de esta investigación es que exponen varios modelos de aprendizaje dando a conocer la eficiencia para este caso de estudio mediante la aplicación de la biblioteca Keras.

Los autores D. Widjojo, E. Setyati y K. Yosi [10] plantea utilizar modelos de aprendizaje de transferencia profunda, de lo cual se basa en tres tareas principales como: segmentar el daño del vehículo, detectar partes dañadas y la clasificación de daños en el vehículo. El modelo específico para realizar la tarea de segmentación se utiliza el modelo Mask R-CNN mientras que para la clasificación se emplea 2 modelos

EfficientNet y MobileNet V2 para medir el rendimiento y eficiencia entre los dos modelos, cabe mencionar que se modificó los modelos EfficientNet y MobileNet V2 incorporando una Red Neuronal Convolutacionales (CNN) simple para aumentar el rendimiento del modelo obteniendo una precisión del 91%, mediante dicha incorporación la CNN simple recibe una máscara de área dañada del modelo de segmentación de daños como entrada adicional para EfficientNet y MobileNet V2. Recapitulando, el proceso combina la segmentación de daños, la segmentación de piezas del vehículo y la clasificación de daños. El resultado principal del proceso es averiguar qué parte del vehículo presenta alguna alteración. En conclusión, el modelo Mask R-CNN puede segmentar bien los daños y las piezas del vehículo, en cuanto al modelo de clasificación si se adhiere una CNN simple puede proporcionar un mejor rendimiento en comparación con aquellos que no incorporan una CNN simple. Además, se descubrió que el modelo MobileNet V2 con la adición de una CNN Simple se obtiene mejores resultados.

El estudio realizado por B. Balci, A. Elihos, M. Turan, B. Alkan e Y. Artan [11] proponen un método de reconocimiento de marca y modelo de vehículos basado en el aprendizaje profundo mediante Cámaras de Infrarrojo Cercano (NIR). Se plantea utilizar un clasificador basado en una Red Neuronal Convolutacional (CNN) y un conjunto de datos de diferentes sombras, luces, reflejos y otros efectos de iluminación para ser utilizados en las etapas de entrenamiento y prueba con una tasa de aprendizaje de 95%, se propone utilizar 3327 imágenes nocturnas para observar el rendimiento del modelo planteado. Los investigadores enfatizan que los modelos de CNN entrenados con imágenes diurnas no son directamente aplicables a las imágenes nocturnas para la deducción, debido a las condiciones de la imagen con poca luz durante la noche. El estudio realizado es llamativo debido a que plantean reducir el brillo de las imágenes de entrenamiento cambia la atención del modelo CNN a estructuras más distintivas que aparecen en condiciones de imágenes nocturnas.

Según C. Muñoz [12] plantea crear un sistema automático de reconocimiento de matrículas vehiculares (ALPR) que permita lectura de matrículas de los vehículos y automatizar la comprobación del vehículo dentro del sistema Policía XPO de las Islas Baleares. El investigado dividió en dos partes la solución: un modelo de You Only Look Once (YOLO) para detectar placas con una red neuronal y por el otro lado un

reconocedor de placas que preprocesa la imagen de la placa, segmenta los caracteres y transfiere la imagen al reconocedor óptico de caracteres para obtener la lectura de la placa. El entrenamiento del modelo se hace en tres fases: fase de entrenamiento, fase de validación y fase de test donde se obtuvo el 82% de precisión. La recopilación de imágenes se lo ha realizado mediante un dispositivo móvil lo que se deduce que se conoce la posición de la placa con respecto a la cámara para su posterior extracción de la región de la imagen que contiene la placa es decir se realizará un preprocesamiento y segmentación por cada carácter de la placa, cada uno servirá como entrada al algoritmo de Reconocimiento Óptico de Caracteres (OCR) la salida de este algoritmo corresponderá con los números y letras que definen la placa. En conclusión, la aplicando del algoritmo de You Only Look Once (YOLO) permite la detección de objetos en tiempo real con buenos resultados.

Los autores N. Shanti, H. Shakhshir, A. Assi y A. Salman [13] plantea aplicar técnicas de aprendizaje automático para generar una estimación del precio del vehículo mediante la aplicación de diversos algoritmos como: Redes Neuronales (ANN), Máquina de Vectores de Soporte (SVM), Vecinos más Cercanos K (KNN), Bosques Aleatorios (RF) y árboles de decisiones. El valor de esta investigación se centra en examinar varios modelos de aprendizaje. La recolección de datos se los realizo de un sitio web enfocada en anuncios de venta de vehículos usados (Web Scraping). El objetivo de esta investigación se basa en la integración de una aplicación móvil para ayudar a los usuarios inexpertos en la adquisición de vehículos. Al ejecutar los distintos modelos de aprendizaje automático mencionados anteriormente se eligió Random Forest donde obtuvo el 90% de precisión en pruebas ya que supero a los otros modelos siendo así Random Forest el modelo que integra en la aplicación móvil. La aplicación móvil utilizo el kit de desarrollo de software de interfaz de usuario multiplataforma de código abierto Dart & Flutter de Google y la comunicación de entre la aplicación y el servidor se lo hizo mediante una Interfaz de Programación de Aplicaciones (API).

Según los autores W. Chengcheng, L. Shicheng, H. Hoffmann, M. Maire y S. Lu [14] plantea diferentes opciones de implementación de Machine Learning mediante una Interfaz de Programación de Aplicaciones (API). Además, los autores enfatizan el desarrollo de sistemas de software de aprendizaje automático mediante la incorporación Machine Learning (ML) en la nube, que permite a los programadores

incorporar redes neuronales ya sea modelos propios o preestablecidos que facilitan la implementación de algoritmos de Machine Learning en la nube. Las principales API destacadas en este artículo son los servicios de Google Cloud AI y los de Amazon ML que incluyen visión, voz e idiomas entre otros servicios para ser consumidos a través de una API.

T. Chunling y S. Du [15] proponen un sistema de detección de múltiples vehículos y reconocimiento de matrículas basado en una Red Neuronal Convolutiva de Región Jerárquica (RCNN) en el cual los investigadores plantean utilizar tres niveles. En el primer nivel, se emplea un nivel más alto de un modelo RCNN para la extracción de vehículos de las imágenes o cuadros por video dado que en este nivel tiene mayor complejidad para detectar el vehículo desde el fondo obteniendo una precisión del 93,1%, la tarea se puede considerar como un problema de clasificación de dos clases el vehículo y fondo. En el Segundo nivel, se orienta en extraer regiones específicas de los vehículos es decir la extracción de la matrícula mediante la introducción de una RCNN de nivel inferior con una tasa éxito de 90,4% pero aquí también surge un problema de clasificación de dos clases que son la matrícula y carrocería del vehículo. Finalmente, en tercer lugar, el reconocimiento de caracteres alfanuméricos mediante una RCNN aún más pequeña con una tasa de éxito del 78%. En conclusión, el beneficio de establecer múltiples de RCNN en el mismo sistema reduce considerablemente la carga de cálculo de acuerdo al nivel que va descendiendo y va disminuyendo las perturbaciones del exterior del vehículo lo que mejora la precisión del modelo.

El autor M. Torres [16] sugiere la creación de un sistema cuyo propósito es reconocer placas de vehículos de Ecuador mediante el uso de técnicas de visión por computadora. Este sistema utiliza dos redes neuronales desarrolladas en Matlab y, como parte del preprocesamiento de las imágenes, se ha llevado a cabo el redimensionamiento y la binarización de las imágenes originales. La binarización implica convertir una imagen en escala de grises a una imagen en blanco y negro aplicando un umbral. Uno de los métodos más comunes para determinar automáticamente el umbral correcto es el método de Otsu, el cual calcula la varianza de la luminosidad de los píxeles de la imagen. La red neuronal establecida en este estudio tiene una capa oculta y tiene como función de activación una función Sigmoide tanto en las capas internas como en la de salida. En conclusión, la aplicación del método Otsu para el procesamiento de las

imágenes y las dos redes neuronales para la clasificación se obtuvo una precisión de 97% lo que se considera aceptable para clasificación de placas vehiculares.

### **1.3 Fundamentación teórica**

#### **1.3.1 Matemáticas Aplicadas**

Matemáticas aplicadas es una rama de la matemática que se enfoca en la aplicación de conceptos y técnicas matemáticas para resolver problemas prácticos en diversas áreas de la ciencia, la ingeniería, la economía, la informática y otros campos, a diferencia de la matemática pura tiene como objetivo utilizar esos conocimientos para abordar situaciones del mundo real. A continuación, se exponen algunas áreas donde las matemáticas aplicadas juegan un papel crucial.[17]

##### ***a. Algebra lineal***

Estudia todo lo relacionado con matrices, vectores o sistemas de ecuaciones lineales. Es muy utilizado, entre otros, en la mecánica o robótica. En el campo del aprendizaje automático, el algebra lineal es utilizado para representar y manipular datos en forma de matrices y vectores como el procesamiento de imágenes.

##### ***b. Calculo diferencial***

Permite conocer la variación que sufren funciones continuas cuando algunas de sus variables se modifican. Su aplicación es habitual, entre otras ciencias, en economía. El cálculo diferencial es fundamental para el entrenamiento de modelos, donde se ajustan los parámetros para minimizar funciones de pérdida.

##### ***c. Estadística inferencial***

Es un campo que tiene como objetivo generalizar los resultados de una prueba realizada en una muestra para aplicarlos a toda la población de la que se obtuvo. Este enfoque puede ser aplicado en áreas como la biología y la medicina. En problemas de aprendizaje automático, se utilizan técnicas estadísticas inferenciales para evaluar la importancia y significancia de las características en el modelo.

***d. Optimización***

El objetivo de las matemáticas aplicadas es encontrar los valores óptimos de una serie de variables representadas por funciones matemáticas. Se aplican en Inteligencia Artificial para minimizar la función de pérdida que mide la discrepancia entre las predicciones del modelo y las etiquetas reales.

**1.3.2 Ciencias de la Computación**

La informática es la disciplina encargada de investigar los procesos computacionales, llamados algoritmos. Su objetivo es desarrollar nuevos programas informáticos para transformar la informática, analizando como diseñar y utilizar herramientas computacionales para resolver problemas de manejo eficiente. Algunas áreas de estudio en ciencias de la computación incluyen: [18]

***a. Inteligencia artificial***

La Inteligencia Artificial es un campo de la ciencia de la computación que procura emular la inteligencia humana en una máquina a través de sistemas que operan con algoritmos, utilizando el aprendizaje profundo y el aprendizaje automático para exhibir comportamientos inteligentes.

***b. Arquitectura de computadora***

La Arquitectura de Computadoras se refiere al diseño conceptual y la estructura operativa fundamental de un sistema computacional, centrándose en como la unidad central de procesamiento lleva a cabo internamente el acceso a las direcciones de memoria.

***c. Análisis de rendimiento de computadoras***

El Análisis de Rendimiento de Computadoras implica estudiar el funcionamiento de los equipos con el fin de optimizar su rendimiento y controlar el tiempo de respuesta, utilizando los recursos de manera eficiente.

***d.      Cómputo científico***

El Computo Científico se dedica a la creación de modelos matemáticos con el fin de administrar la conectividad entre redes de área local (LAN) y redes de área amplia (WAN) de computadoras a nivel mundial.

***e.      Base de datos***

Las bases de datos son una disciplina de la informática que se encarga de organizar, almacenar y recuperar grandes cantidades de datos de forma sencilla, utilizando sistemas que gestionan, crean, mantienen y consultan los datos a través de modelos y lenguajes informáticos.

***f.      Ciencia de la información***

La Ciencia de la Información se encarga de analizar la estructura, algoritmos y comportamiento de los sistemas naturales y artificiales que almacenan, procesan, acceden y comunican información. Además, se dedica a desarrollar sus propios fundamentos conceptuales y teóricos.

***g.      Ingeniería de software***

La Ingeniería de Software se dedica al diseño, desarrollo y modificación del software para asegurar que sea de alta calidad, asequible, fácil de utilizar y rápido de construir.

**1.3.3 Inteligencia Artificial**

La Inteligencia Artificial (IA) se refiere a la simulación de las capacidades de inteligencia del cerebro humano. Se considera que la IA es una parte de las ciencias de la computación que se encarga de diseñar sistemas inteligentes. Estos sistemas exhiben características asociadas con la inteligencia en el comportamiento humano. La IA ofrece una variedad de métodos, técnicas y herramientas para modelar y resolver problemas, imitando la forma en que los individuos cognitivos actúan. [19]



### 1.3.4 Machine Learning

El Machine Learning (ML) es un término amplio que se usa para describir varias estrategias analíticas cuyo objetivo es desarrollar algoritmos para la extracción de información, ya sea para explicación, clasificación o predicción. Los ML pueden ser clasificados en: supervisados, no supervisados y parcialmente supervisados, y la principal diferencia entre estos tres tipos de algoritmos es la presencia o ausencia de una variable de resultados o dependiente, a la que se le denomina marca o etiqueta. [20]

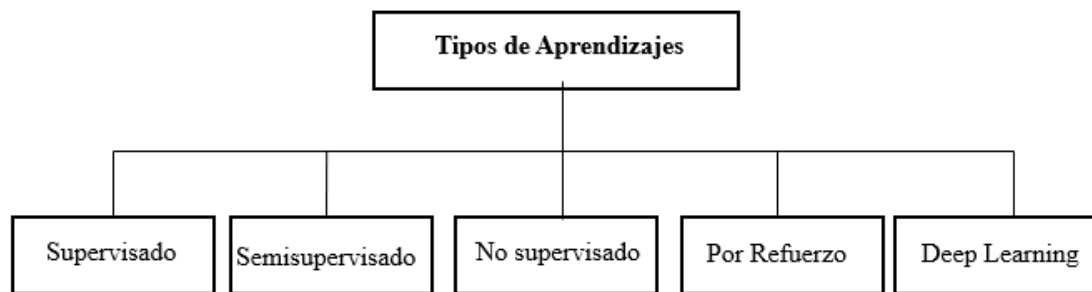


Figura 1. Tipos de Aprendizaje [21]

Existen varios tipos de aprendizaje, y a continuación se describe algunos de los aprendizajes más relevantes: [21]

#### *a. Algoritmos supervisados*

Los Algoritmos Supervisados tienen la finalidad de hacer predicciones. En este sentido, el algoritmo se entrena para relacionar un conjunto de variables con un resultado de interés.

#### *b. Deep Learning*

El Deep Learning es una rama del machine learning que incorpora redes neuronales multicapa lo que permite aprender representaciones complejas de datos. Esta tecnología ha demostrado ser muy efectiva en diversas aplicaciones debido a su capacidad para modelar relaciones no lineales y extraer características automáticamente.

### 1.3.5 Visión Artificial

Es un campo de estudio de la inteligencia artificial que se centra en permitir a las maquinas interpretar y comprender la información visual del mundo que las rodea. [21]

#### a. *Detección de Objetos*

Es una tarea dentro de campo de la visión artificial que implica identificar y localizar la presencia de objetos específicos en una imagen o video.

### 1.3.6 Redes Neuronales

Es un modelo matemático inspirado en el cerebro humano. Estos modelos son diseñados para tareas específicas de aprendizaje automático y procesamiento de información. Estas redes están compuestas por neuronas interconectadas y están organizadas por capas. [21]

#### a. *Redes neuronales Convolucionales (CNN)*

Una red Neuronal Convolutiva es un tipo de red neuronal especializada, diseñada específicamente para procesamiento eficiente de datos bidimensional como imágenes y videos. Las CNN son útiles en extraer información de forma automática sin ningún tipo de extracción manual debido que incorporan capas de convolucionales que realizan una operación matricial entre la imagen y una que pequeña matriz llamada kernel para obtener las características de la imagen.[21]

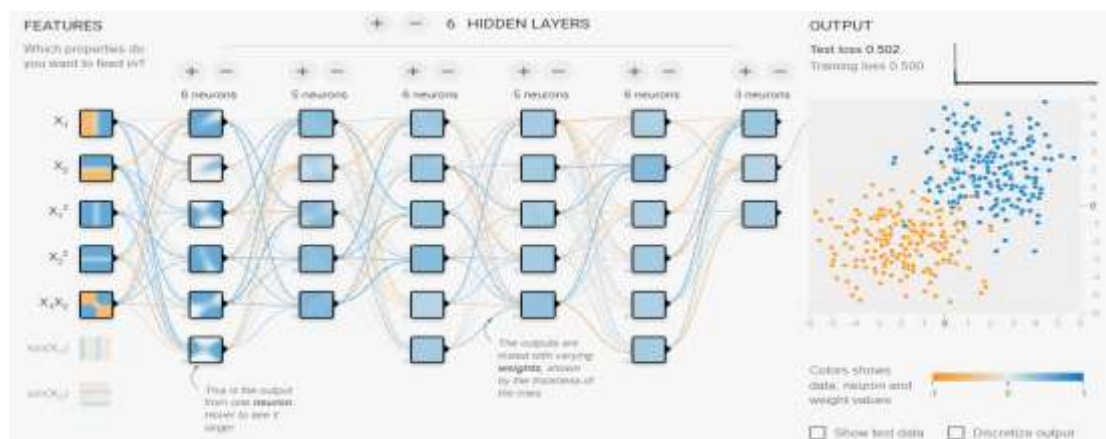


Figura 2. Red Neuronal [22]

### **1.3.7 Aplicaciones Móviles**

Una aplicación móvil es un programa de software diseñado para ejecutarse en dispositivos móviles como teléfonos inteligentes y tabletas. Estas aplicaciones son diseñadas para realizar una tarea específica, proporcionar servicios o entretenimiento a los usuarios en sus dispositivos móviles.[23]

#### **a. *Aplicaciones Nativas***

Están diseñadas para un único sistema operativo móvil, mediante el Development Kit (SDK) y se desarrollan con lenguajes de programación específicos como Java y Kotlin para Android y Swift para iOS lo cual trata de aprovechar los recursos de hardware al máximo y optimiza las funciones del dispositivo. [24]

#### **b. *Aplicaciones Híbridas***

Están diseñadas para la ejecución en los distintos sistemas operativos con un solo desarrollo.

### **1.3.8 Comportamiento Humano**

El comportamiento humano se refiere a las acciones, reacciones y conductas que las personas exhiben en respuestas a diversos estímulos, situaciones o circunstancias en su entorno. [25]

### **1.3.9 Ciencias Sociales**

Las Ciencias Sociales son un conjunto de disciplinas académicas que estudian aspectos diversos de la sociedad humana, sus instituciones, estructuras, comportamiento, interacciones y cambios a lo largo del tiempo. Estas disciplinas buscan comprender y explicar cómo las personas interactúan entre sí, como se organizan en comunidades y como se desarrollan las culturas. [26]

### **1.3.10 Economía**

La Economía es una disciplina que estudia como las sociedades gestionan sus recursos escasos para satisfacer sus necesidades y deseos.[27]

### **1.3.11 Pérdidas Económicas**

La pérdida económica se refiere a la disminución del valor o la utilidad de un recurso, bien, inversión o actividad económica. Puede manifestarse de diversas maneras y en diferentes contextos, y generalmente implica una reducción en el valor monetario de algún elemento. [28]

## **1.4 Objetivos**

### **1.4.1 Objetivo general**

Implantar una aplicación móvil utilizando aprendizaje automático para mitigar pérdidas económicas al adquirir vehículos de segunda mano basados en la detección de anomalías estéticas y consulta de información general del vehículo en la ciudad de Ambato.

### **1.4.2 Objetivos específicos**

- Identificar el nivel de pérdidas económicas de las personas involucradas en el comercio de vehículos usados.
- Investigar bibliotecas que optimicen la ejecución de algoritmos de aprendizaje automático en dispositivos móviles.
- Desarrollar dos modelos de clasificación de imágenes basados en aprendizaje automático para la detección de anomalías estéticas y consulta de información general del vehículo en una aplicación móvil.

## CAPÍTULO II. METODOLOGÍA

### 2.1 Materiales

El material empleado para la recolectar información en esta investigación se basó en la aplicación de encuestas, el cual consta de 15 preguntas establecidas con una escala de Likert 5. Este cuestionario fue aplicado a las personas que intervienen en la feria de vehículos usados de la explanada de Shuyurco de la ciudad de Ambato provincia del Tungurahua.

#### 2.1.1 Encuesta

##### **Encuesta para evaluar a las personas de la feria de vehículos usados de la ciudad de Ambato**

Esta encuesta tiene como finalidad satisfacer el objetivo 1 del presente trabajo de investigación recopilar.

Tabla 3. Escala le Likert

1	Extremadamente importante
2	Muy importante
3	Un poco importante
4	Ligeramente importante
5	Nada Importante

*a. Que tan importante es para Ud. considerar los aspectos legales y estéticos al momento de adquirir un vehículo usado.*

1	<input type="radio"/>
2	<input type="radio"/>
3	<input type="radio"/>
4	<input type="radio"/>
5	<input type="radio"/>

b. *El estado de la matrícula del vehículo que tan importante lo considera Ud. al adquirir un vehículo usado.*

1	<input type="radio"/>
2	<input type="radio"/>
3	<input type="radio"/>
4	<input type="radio"/>
5	<input type="radio"/>

c. *Es fundamental para Ud. conocer el historial del vehículo antes de concretar la compra de un vehículo usado.*

1	<input type="radio"/>
2	<input type="radio"/>
3	<input type="radio"/>
4	<input type="radio"/>
5	<input type="radio"/>

d. *Es crucial para Ud. conocer si el vehículo tiene muchas pendientes al momento de adquirir un vehículo usado.*

1	<input type="radio"/>
2	<input type="radio"/>
3	<input type="radio"/>
4	<input type="radio"/>
5	<input type="radio"/>

e. *Que tan importante es para Ud. saber si el vehículo tiene denuncias.*

	<input type="radio"/>
2	<input type="radio"/>
3	<input type="radio"/>
4	<input type="radio"/>
5	<input type="radio"/>

f. *Que importancia le da Ud. conocer si el vehículo es chocado.*

1	<input type="radio"/>
2	<input type="radio"/>
3	<input type="radio"/>
4	<input type="radio"/>
5	<input type="radio"/>

**g.** *Que tanta importancia le da Ud. conocer si el vehículo es alterado en las numeraciones.*

1	<input type="radio"/>
2	<input type="radio"/>
3	<input type="radio"/>
4	<input type="radio"/>
5	<input type="radio"/>

**h.** *Que tan importante es para Ud. que el número de chasis del vehículo sea el mismo registrado en la Policía Judicial (PJ).*

1	<input type="radio"/>
2	<input type="radio"/>
3	<input type="radio"/>
4	<input type="radio"/>
5	<input type="radio"/>

**i.** *Que tan importante es para Ud. que el número de motor del vehículo sea el mismo registrado en la (PJ).*

1	<input type="radio"/>
2	<input type="radio"/>
3	<input type="radio"/>
4	<input type="radio"/>
5	<input type="radio"/>

**j.** *Que tan fundamental es para Ud. conocer el origen de la placa del vehículo.*

1	<input type="radio"/>
2	<input type="radio"/>
3	<input type="radio"/>
4	<input type="radio"/>
5	<input type="radio"/>

**k.** *Que tan importante es para Ud. conocer el nombre del propietario actual de vehículo.*

1	<input type="radio"/>
2	<input type="radio"/>
3	<input type="radio"/>
4	<input type="radio"/>
5	<input type="radio"/>

*l. Que tan importante es para Ud. conocer el estado de la última revisión vehicular del vehículo.*

1	<input type="radio"/>
2	<input type="radio"/>
3	<input type="radio"/>
4	<input type="radio"/>
5	<input type="radio"/>

*m. Que importancia le da Ud. al estado de la pintura del vehículo.*

1	<input type="radio"/>
2	<input type="radio"/>
3	<input type="radio"/>
4	<input type="radio"/>
5	<input type="radio"/>

*n. Que tan fundamental es para Ud. conocer si el vehículo fue ex taxi.*

1	<input type="radio"/>
2	<input type="radio"/>
3	<input type="radio"/>
4	<input type="radio"/>
5	<input type="radio"/>

*o. Que le ha parecido la encuesta que acaba de responder.*

1	<input type="radio"/>
2	<input type="radio"/>
3	<input type="radio"/>
4	<input type="radio"/>
5	<input type="radio"/>

**¡Gracias por su colaboración!**

Utilizando la plataforma de Google Forms, se diseñó una encuesta que consta de 15 preguntas establecidas con una escala de Likert 5 y fue dirigida a las personas involucradas en el comercio de vehículos usados. Además, este cuestionario fue validado su consistencia interna mediante la aplicación del alfa de Cronbach descrito en la sección 2.2.3 de este trabajo de investigación.



## **2.2 Métodos**

### **2.2.1 Modalidad de la investigación**

La investigación actual se enmarca dentro de la modalidad de investigación de campo, documental bibliográfico y modalidades especiales.

#### ***a. Investigación de campo***

La investigación es de campo dado que el investigador obtuvo datos del lugar de los hechos por medio de encuestas que se planteó a las personas involucradas en el comercio de vehículos usados, y en base a las experiencias de las personas se obtuvo datos que contribuyan a determinar una solución factible para el problema objeto de estudio en este proyecto de investigación.

#### ***b. Investigación bibliográfica – documental***

La actual investigación es bibliográfica-documental dado que se fundamenta en la recolección de información a través de artículos, revistas, conferencias y tesis realizadas que están enfocadas al tema de estudio de este proyecto de investigación, la cual se utiliza para la construcción de la fundamentación teórica que estudia temas de: Aprendizaje automático, Machine Learning, Deep Learning, aplicaciones móviles entre otros.

#### ***c. Investigación de modalidades especiales***

Esta modalidad se involucra a la investigación por tratarse de un proyecto que se enfoca en realizar una aplicación móvil innovadora y útil que brinde una solución tecnológica mediante la implementación de la Inteligencia Artificial para ofrecer a las personas un soporte de toma de decisiones al momento de adquirir un vehículo usado.

### 2.2.2 Población y muestra

La población que se investigó está conformada por las personas que acuden a la feria de vehículos en la explanada de Shuyurco de la ciudad de Ambato provincia de Tungurahua. Debido a que la población es un numero desconocido de personas que acuden a la feria de vehículos es una población infinita.

#### a. *Muestra para aplicar en la encuesta*

Para el cálculo del tamaño de la muestra representativa para aplicar la encuesta se utilizó un nivel de confianza deseado del 95%, porque puede haber criterios diferentes acerca de los aspectos que influyen al adquirir un vehículo usado, un margen de error de 10% dado que se desea generalizar y una probabilidad de 0.5 debido a que no existen estudios previos, entonces se aplica la siguiente formula:

$$\text{Tamaño de la muestra (N)} = \frac{z^2 * p * (1 - p)}{e^2}$$

Donde:

N = Tamaño de la muestra.

NC = Nivel de confianza deseado 95% equivalente a 1.96.

Z = Puntuación de valor Z según NC.

P = Probabilidad de éxito.

E = Margen de error 10%.

$$n = \frac{1.96^2 * 0.5 * 0.5}{0.1^2} \quad n = 96$$

De la población infinita de las personas que acuden a las ferias de vehículos con un nivel de confianza de 95% y un margen de error 10%, la formula estadística arrojo un tamaño de muestra de 96 individuos, donde se recogió una muestra aleatoria simple que incluya tanto personas que acuden a la feria con la finalidad de comprar un vehículo usado para su uso personal y comerciantes de vehículos que ven como una actividad económica para ellos, entonces se aplicó una encuesta para poder definir las necesidades tanto en el campo legal y estético del vehículo que toman las personas al momento de adquirir un vehículo usado.

### 2.2.3 Recolección de información

Para el proceso de recolección de información se realizó a través de encuestas con la finalidad de conocer los aspectos a considerar al momento de adquirir un vehículo usado y así, deducir metodologías de estudio que servirán de apoyo en la ejecución del proyecto.

#### a. *Validación del instrumento*

Para evaluar la consistencia interna o fiabilidad del cuestionario se aplica la herramienta denominada alfa de Cronbach, esto determinara si las preguntas son coherentes y miden de manera confiable el concepto u objeto de estudio.

Para el cálculo del alfa de Cronbach se aplicó la siguiente fórmula:

$$\alpha = \left( \frac{k}{k-1} \right) \left( 1 - \frac{\sum v_i}{vt} \right)$$

Donde:

$\alpha$  = fiabilidad del cuestionario.

$k$  = Número de ítems.

$\sum v_i$  = Varianza de cada ítem

$vt$  = Varianza total.

Los cálculos de la fórmula del alfa de Cronbach fueron realizados en Excel a partir de la creación de una tabla, adjuntando las respuestas recolectadas de 15 encuestas realizadas aleatoriamente a comerciantes y compradores de vehículos usados en la explanada de Shuyurco, feria de vehículos usados de la ciudad de Ambato. Ver Anexo A.

$$\alpha = \left( \frac{15}{15-1} \right) \left( 1 - \frac{6,40}{20,51} \right) = 0,737$$

Tabla 4. Interpretación de la escala de consistencia del alfa de Cronbach

Rango	Confiabilidad
1	Confiabilidad perfecta
0,72 a 0,99	Excelente confiabilidad
0.66 a 0,71	Muy confiable
0,60 a 0,65	Confiable
0,54 a 0,59	Confiabilidad baja
< 0,53	Confiabilidad nula

Como resultado el alfa de Cronbach da un valor de 0,737, se interpreta según la tabla descrita anteriormente donde se determina que el cuestionario es de excelente confiabilidad dado las respuestas de las personas involucradas en comercio de vehículos usados. Una vez evaluado la consistencia interna del cuestionario se procede a aplicar la encuesta a 96 personas según el tamaño de la muestra establecido anteriormente.

**b. Resultados de la encuesta**

La encuesta fue realizada presencialmente a los comerciantes y compradores de vehículos usados de la explanada de Shuyurco mediante la herramienta de Google Forms donde se alojó el cuestionario.

- **Pregunta a: ¿Qué tan importante es para Ud. considerar los aspectos legales y estéticos al momento de adquirir un vehículo usado?**

Tabla 5. Resultados pregunta a

Indicador	Frecuencia	Porcentaje
Nada importante	0	0%
Ligeramente importante	0	0%
Un poco importante	2	2%
Muy importante	11	12%
Extremadamente importante	83	86%
<b>Total</b>	96	100%

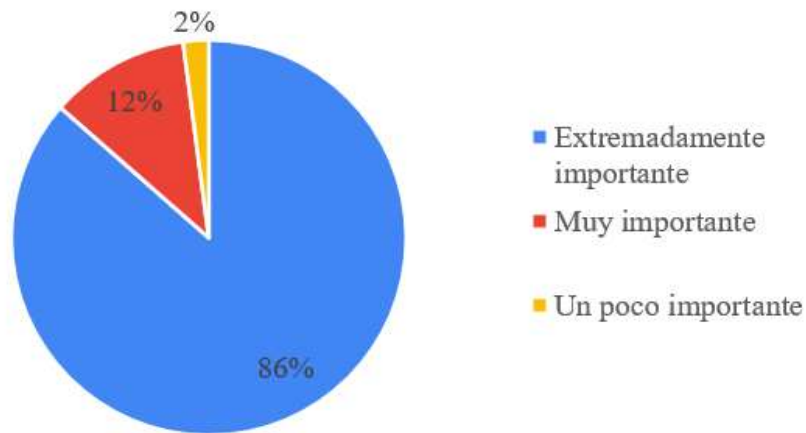


Figura 3. Resultados pregunta a

### **Análisis e interpretación de resultados**

Con los resultados obtenidos en la Figura. 3, se observa que mayoritariamente las personas vinculadas con el comercio de vehículos usados consideran extremadamente importante tomar en cuenta los aspectos legales y estéticos del vehículo a la hora de adquirir uno. Como resultado se evidencia que 86% de las personas tiene presente tomar en cuenta los aspectos anteriormente mencionados porque esto puede representar la pérdida o ganancia económica en este tipo de actividades.

- ***Pregunta b: ¿El estado de la matrícula del vehículo que tan importante lo considera Ud. al adquirir un vehículo usado?***

Tabla 6. Resultados pregunta b

<b>Indicador</b>	<b>Frecuencia</b>	<b>Porcentaje</b>
Nada importante	0	0%
Ligeramente importante	2	2%
Un poco importante	9	9%
Muy importante	24	25%
Extremadamente importante	61	64%
<b>Total</b>	<b>96</b>	<b>100%</b>

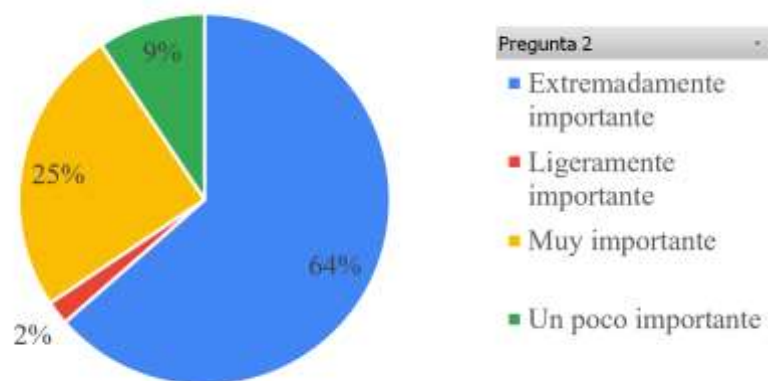


Figura 4. Resultados Pregunta b

### Análisis e interpretación de resultados

Con los resultados obtenidos en la Figura 4, se evidencia que más de la mitad de los comerciantes y personas lígales en este tipo de actividades considera extremadamente importante. En efecto se evidencia que más del 50% de las personas considera revisar el estado de la matricula ya que puede revelar información importante sobre el historial y la situación legal del vehículo para evitar un posible declive económico.

- ***Pregunta c: ¿Es fundamental para Ud. conocer el historial del vehículo antes de concretar la compra de un vehículo usado?***

Tabla 7. Resultados pregunta c

Indicador	Frecuencia	Porcentaje
Nada importante	0	0%
Ligeramente importante	0	0%
Un poco importante	5	5%
Muy importante	18	19%
Extremadamente importante	73	76%
<b>Total</b>	96	100%

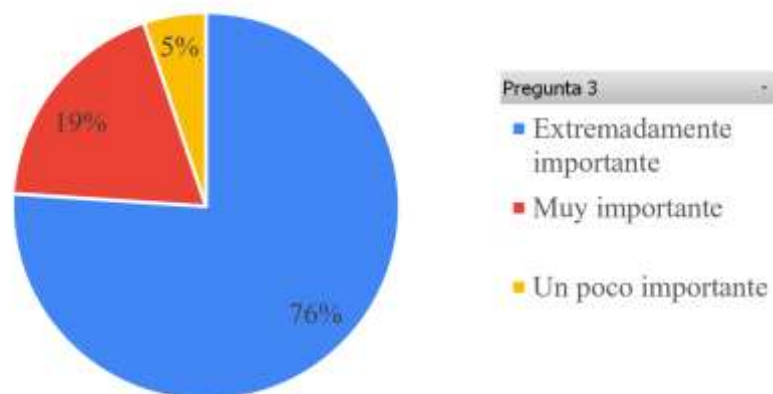


Figura 5. Resultados pregunta c

### Análisis e interpretación de resultados

Con los resultados obtenidos en la Figura 5, se observa que el 76% de las personas considera extremadamente importante. Como resultado se refleja que mayoritariamente las personas consideran revisar el historial del vehículo dado que este revela información detallada sobre la vida y el uso del vehículo a lo largo del tiempo para así poder estimar las condiciones actuales del vehículo.

- ***Pregunta d: ¿Es crucial para Ud. conocer si el vehículo tiene muchas pendientes al momento de adquirir un vehículo usado?***

Tabla 8. Resultados pregunta d

Indicador	Frecuencia	Porcentaje
Nada importante	0	0%
Ligeramente importante	0	0%
Un poco importante	1	1%
Muy importante	9	9%
Extremadamente importante	86	90%
<b>Total</b>	<b>96</b>	<b>100%</b>

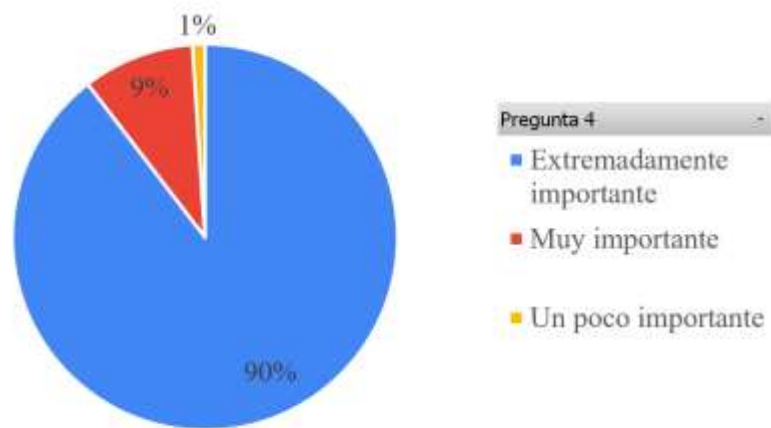


Figura 6. Resultados pregunta d

### Análisis e interpretación de resultados

Con los resultados obtenidos se observa en la Figura 6, que casi la mayoría de los individuos considera extremadamente importante chequear si el vehículo tiene multas pendientes. En efecto se evidencia que 90% de las personas lo consideran de suma importancia tomar en cuenta este aspecto ya que puede tener diversas consecuencias legales y financieras.

- **Pregunta e: ¿Qué tan importante es para Ud. saber si el vehículo tiene denuncias?**

Tabla 9. Resultados pregunta e

Indicador	Frecuencia	Porcentaje
Nada importante	0	0%
Ligeramente importante	0	0%
Un poco importante	0	0%
Muy importante	3	3%
Extremadamente importante	93	97%
<b>Total</b>	96	100%



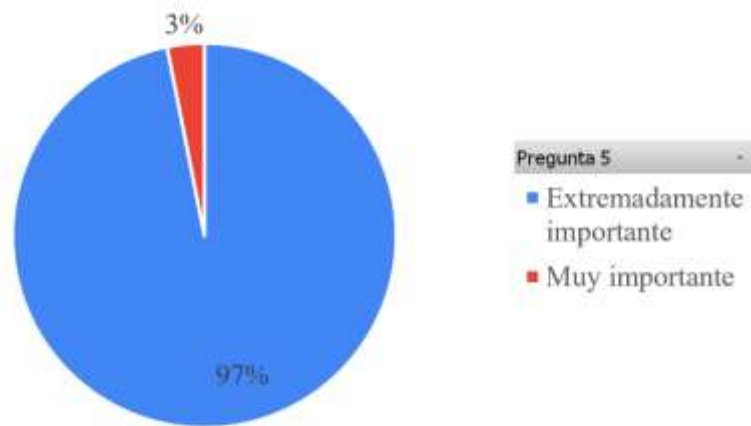


Figura 7. Resultados pregunta e

### Análisis e interpretación de resultados

Con los resultados obtenidos en la Figura 7, el 97% considera extremadamente importante tener en cuenta si el vehículo tiene denuncias, porque esto puede traer diversas complicaciones legales y financieras en el futuro.

- *Pregunta f: ¿Qué importancia le da Ud. conocer si el vehículo es chocado?*

Tabla 10. Resultados pregunta f

Indicador	Frecuencia	Porcentaje
Nada importante	0	0%
Ligeramente importante	4	4%
Un poco importante	10	11%
Muy importante	25	26%
Extremadamente importante	57	59%
<b>Total</b>	96	100%

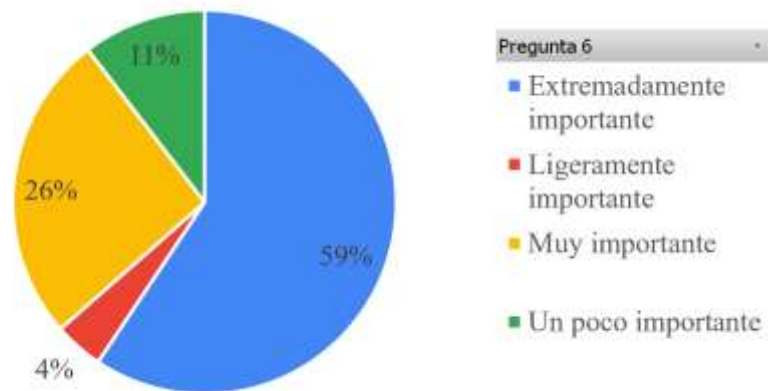


Figura 8. Resultados pregunta f

### Análisis e interpretación de resultados

Con los resultados obtenidos en la Figura 8, se puede visualizar que más de la mitad de las personas ligadas en este tipo de actividad consideran extremadamente importante verificar el estado de la carrocería, porque esto implica una desvalorización y dificultad para vender el vehículo a futuro.

- **Pregunta g: ¿Qué tanta importancia de da Ud. conocer si el vehículo es alterado en las numeraciones?**

Tabla 11. Resultados pregunta g

Indicador	Frecuencia	Porcentaje
Nada importante	0	0%
Ligeramente importante	14	15%
Un poco importante	19	20%
Muy importante	26	27%
Extremadamente importante	37	38%
<b>Total</b>	<b>96</b>	<b>100%</b>

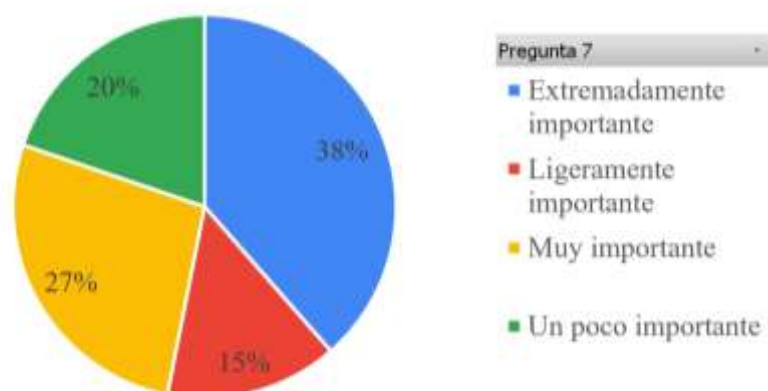


Figura 9. Resultados pregunta g

### Análisis e interpretación de resultados

Los resultados de pregunta formulada se han analizado en la Figura 9 donde el 38% responden extremadamente importante verificar las numeraciones de un vehículo. La verificación de la numeración resalta la significativa preocupación de las personas por la integridad y autenticidad de la información relacionada con la numeración del vehículo porque esto puede representar la pérdida del vehículo.

- **Pregunta h: ¿Qué tan importante es para Ud. que el número de chasis del vehículo sea el mismo registrado en la Policía Judicial (PJ)?**

Tabla 12. Resultados pregunta h

Indicador	Frecuencia	Porcentaje
Nada importante	0	0%
Ligeramente importante	0	0%
Un poco importante	1	1%
Muy importante	5	5%
Extremadamente importante	90	94%
<b>Total</b>	96	100%

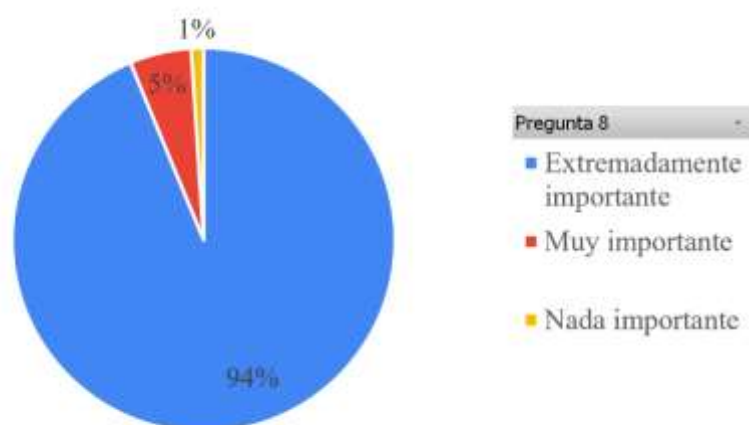


Figura 10. Resultados pregunta h

### Análisis e interpretación de resultados

Con los resultados obtenidos en la Figura 10, la mayoría de las personas considera extremadamente importante que el número de chasis coincida con el número de chasis registrado en la base de datos de la PJ. En conclusión, es evidente que casi el 100% de las personas lo considera extremadamente importante porque puede tener consecuencias graves como por ejemplo propiedad dudosa asociada a actividades ilegales como robo de vehículos lo cual implicaría la pérdida del automotor.

- **Pregunta i: ¿Qué tan importante es para Ud. que el número de motor del vehículo sea el mismo registrado en la Policía Judicial (PJ)?**

Tabla 13. Resultados pregunta i

Indicador	Frecuencia	Porcentaje
Nada importante	1	1%
Ligeramente importante	0	0%
Un poco importante	0	0%
Muy importante	6	6%
Extremadamente importante	89	93%
<b>Total</b>	<b>96</b>	<b>100%</b>

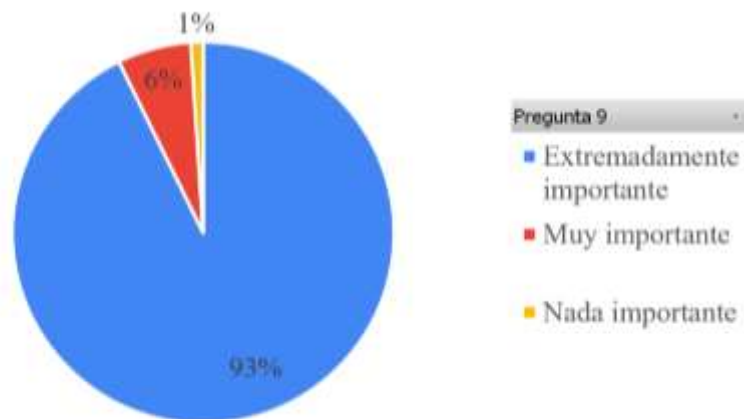


Figura 11. Resultados pregunta i

### Análisis e interpretación de resultados

Con los resultados obtenidos en la Figura 11, el 93% considera extremadamente importante verificar que la numeración sea la misma registrada en la PJ. En efecto, se evidencia que la mayoría lo considera extremadamente importante dado que esto puede estar involucrado a actividades ilícitas lo que conllevaría.

- **Pregunta j: ¿Qué tan fundamental es para Ud. conocer el origen de la placa del vehículo?**

Tabla 14. Resultados pregunta j

Indicador	Frecuencia	Porcentaje
Nada importante	1	1%
Ligeramente importante	1	1%
Un poco importante	11	11%
Muy importante	21	22%
Extremadamente importante	62	65%
<b>Total</b>	96	100%



Figura 12. Resultados pregunta j

### Análisis e interpretación de resultados

Con los resultados obtenidos en la Figura 12, se observa que el 65% de los individuos considera conocer el origen de la placa, porque revela donde fue matriculado por primera vez el vehículo y obtiene información útil para determinar si el vehículo ha estado en una región climática extrema que podría afectar su estado estético y consiga afectar la valorización del mismo.

- **Pregunta k: ¿Qué tan importante es para Ud. conocer el nombre del propietario actual de vehículo?**

Tabla 15. Resultados pregunta k

Indicador	Frecuencia	Porcentaje
Nada importante	1	1%
Ligeramente importante	18	19%
Un poco importante	12	13%
Muy importante	28	29%
Extremadamente importante	37	39%
<b>Total</b>	<b>96</b>	<b>100%</b>



Figura 13. Resultados pregunta k

### Análisis e interpretación de resultados

Con los resultados obtenidos en la Figura 13, el 39% de las personas considera extremadamente importante. Por lo tanto, los individuos reconocen la importancia de conocer el nombre del propietario actual del vehículo dado que, contar con los datos de contacto del propietario actual es útil en caso de que surjan problemas o preguntas posteriores a la compra.

- **Pregunta l: ¿Qué tan importante es para Ud. conocer el estado de la última revisión vehicular del vehículo?**

Tabla 16. Resultados pregunta l

Indicador	Frecuencia	Porcentaje
Nada importante	1	1%
Ligeramente importante	1	1%
Un poco importante	8	8%
Muy importante	33	35%
Extremadamente importante	53	55%
<b>Total</b>	96	100%



Figura 14. Resultados pregunta l

### Análisis e interpretación de resultados

Los resultados de la pregunta formulada se han analizado en la Figura 14, donde el 55% considera importante conocer el estado de la última revisión vehicular. En conclusión, las personas reconocen mayoritariamente extremadamente importante verificar la última revisión vehicular dado que garantiza el cumplimiento con las regulaciones de seguridad y emisiones periódicas.

- *Pregunta m: ¿Qué importancia le da Ud. al estado de la pintura del vehículo?*

Tabla 17. Resultados pregunta m

Indicador	Frecuencia	Porcentaje
Nada importante	1	1%
Ligeramente importante	0	0%
Un poco importante	4	4%
Muy importante	30	31%
Extremadamente importante	61	64%
<b>Total</b>	96	100%





Figura 15. Resultados pregunta m

### Análisis e interpretación de resultados

Una vez obtenido los resultados de la pregunta 13 se observa en la Figura 15 que el 64% de las personas considera extremadamente importante el estado de la pintura. Por lo tanto, es evidente que mayoritariamente las personas consideran este aspecto estético ya que no solo tiene un impacto estético, sino también puede proporcionar información sobre el estado general del vehículo y afecta su valor.

- **Pregunta n:** *¿Qué tan fundamental es para Ud. conocer si el vehículo fue ex taxi?*

Tabla 18. Resultados pregunta n

Indicador	Frecuencia	Porcentaje
Nada importante	2	2%
Ligeramente importante	1	1%
Un poco importante	3	3%
Muy importante	20	21%
Extremadamente importante	70	73%
<b>Total</b>	<b>96</b>	<b>100%</b>

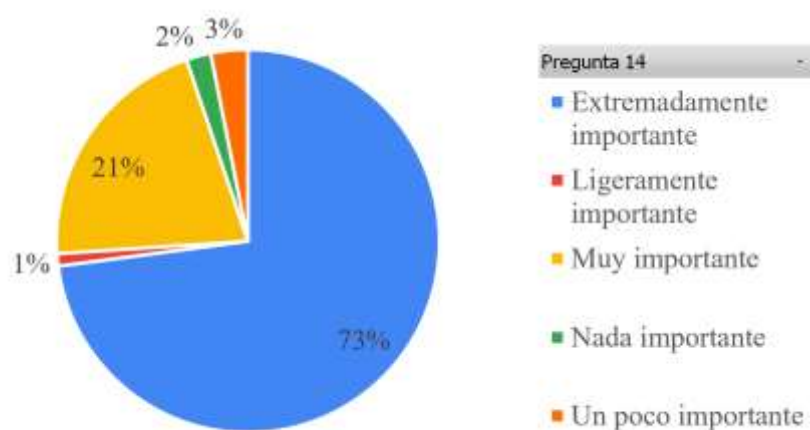


Figura 16. Resultados pregunta n

### Análisis e interpretación de resultados

Con referencia a la Figura 16, el 73% de los encuestados considera extremadamente importante conocer si el vehículo fue ex taxi. Dado que los encuestados reconocen que es importante conocer si el vehículo fue ex taxi porque representa un desgaste acelerado en componentes clave y puede presentar problemas a futuro.

- **Pregunta o: ¿Qué le ha parecido la encuesta que acaba de responder?**

Tabla 19. Resultados pregunta o

Indicador	Frecuencia	Porcentaje
Nada importante	1	1%
Ligeramente importante	1	1%
Un poco importante	5	5%
Muy importante	37	39%
Extremadamente importante	52	54%
<b>Total</b>	96	100%

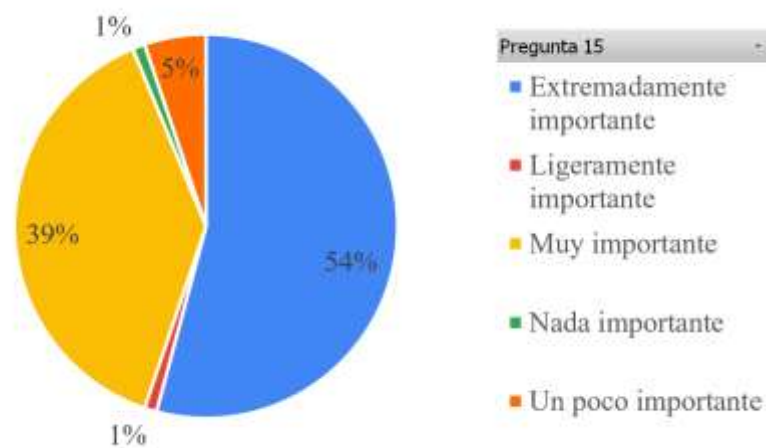


Figura 17. Resultados pregunta o

### **Análisis e interpretación de resultados**

Según la Figura 17, el 54% de los encuestados respondió que la encuesta aplicada a es extremadamente importante. En conclusión, los factores abordados en esta encuesta son considerados para la mayoría de suma importancia dado que si no se consideran estos factores implicaría una pérdida económica.

#### **2.2.4 Procesamiento y análisis de datos**

La información recolectada a través de la encuesta aplicada hacia las personas ligadas al comercio de vehículos usados demostró:

- Considerar los aspectos legales y estéticos al adquirir un vehículo usado es de extrema importancia para la gran mayoría de las personas. Esto refleja una alta conciencia de la importancia de verificar la documentación legal, el estado estético antes de la compra.
- Se identificó que los encuestados subraya la relevancia de tener en cuenta la legalidad y la documentación de un vehículo al realizar una compra, ya que puede tener un impacto significativo en la seguridad y la satisfacción del comprador.
- Queda claro que la mayoría de las personas considera que la revisión del historial del vehículo es un factor crucial al comprar un vehículo usado, esto

refleja la preocupación por aspectos como los accidentes pasados y cualquier problema legal relacionado con el vehículo.

- Se destaca la gran relevancia que se otorga a la verificación de multas pendientes antes de comprar, esto manifiesta la preocupación por posibles problemas legales y financieros que podría surgir si el vehículo tiene infracciones no resueltas.
- La mayoría de los encuestados resalta la importancia crítica que se atribuye a la verificación de denuncias asociadas al vehículo, esto manifiesta la profunda preocupación por conocer posibles problemas legales o de seguridad que podrían surgir de tales denuncias.
- Mayoritariamente las personas otorgan una alta importancia a conocer si el vehículo tiene desperfectos en la carrocería antes de adquirir un vehículo usado, por lo tanto, verificar los daños previos en el vehículo es un factor crítico al tomar una decisión de compra.
- Los encuestados valoran la importancia de conocer si el vehículo tiene detalles en la parte estética, demostrando preocupación por la posible existencia de reparaciones importantes en la carrocería del vehículo. Esto subraya la importancia de una inspección minuciosa de la carrocería antes de una compra para aquellos que valoran la apariencia estética y la integridad estructural del vehículo.

## CAPÍTULO III. RESULTADOS Y DISCUSIÓN

### 3.1 Análisis y discusión de resultados

En el capítulo II de este trabajo de investigación, se llevó a cabo la recopilación de información. A través de este proceso, se identificaron diversas cuestiones al adquirir vehículos usados. De este modo se procede a levantar requerimientos funcionales y no funcionales que debe cumplir la aplicación móvil propuesta.

#### 3.1.1 Requerimientos funcionales y no funcionales

##### a. *Requerimientos funcionales*

Mediante la recolección de la información se pudo identificar los requerimientos funcionales que debe contener la aplicación móvil propuesta en este trabajo de investigación.

Tabla 20. Requerimientos funcionales

<b>Función</b>	<b>Comportamiento</b>
Consulta del estado de la matricula	La aplicación consultará en los registros de la Agencia Nacional de Transito (ANT) el estado de la matricula.
Consulta del historial	La aplicación consultará el historial del vehículo en la ANT.
Verificación de multas	La aplicación verificará en los registros de la ANT si el vehículo tiene multas pendientes.
Verificación de denuncias	La aplicación consultará en la ANT si el vehículo tiene denuncias.
Analizar el exterior del vehículo	La aplicación examinará el exterior del vehículo para detectar posibles alteraciones estéticas (golpes, raspones, abolladuras y cambio de color).
Verificación de las numeraciones	La aplicación consultará las numeraciones (chasis y motor) en los registros de la Policía Judicial (PJ).
Consulta del propietario	La aplicación consultará el nombre del propietario actual del vehículo.

##### b. *Requerimientos no funcionales*

Los requerimientos no funcionales de la aplicación móvil se establecieron acuerdo a las expectativas de los usuarios, estos requerimientos aseguran que la app cumpla con los estándares de calidad y rendimiento necesario para satisfacer a los usuarios.

Tabla 21. Requerimientos no funcionales

<b>Función</b>	<b>Comportamiento</b>
Navegabilidad	La aplicación deberá poseer una interfaz sencilla e intuitiva para el usuario.
Seguridad y privacidad	La aplicación no compartirá datos que el usuario capture o guarde en un servidor en la nube.
Integridad de los datos	La aplicación deberá poseer una validación para el formulario de ingreso de información de contacto del vendedor.
Desempeño	La aplicación deberá ser rápida y eficiente, brindando al usuario respuestas breves.
Precisión de los modelos de aprendizaje automático	Los algoritmos de aprendizaje automático deberán tener una buena precisión de clasificación y detección de anomalías estéticas del vehículo.
Captura de video	La aplicación permitirá que el usuario acceda a la cámara para la detección de placa y características estéticas del vehículo.
Almacenar información	La aplicación permitirá al usuario guardar información del estado legal y estético del vehículo conjuntamente con la información de contacto del vendedor.
Eliminar la información	La aplicación permitirá al usuario eliminar información guardada.
Modificar la información	La aplicación deberá permitir la modificación de la información de contacto del vendedor previamente guardada.

### 3.1.2 Análisis de bibliotecas de aprendizaje automático para dispositivos móviles

En torno a dispositivos móviles y aprendizaje automático se puede encontrar una gran variedad de bibliotecas que cuentan con las funcionalidades adecuadas para la implementación de modelos de aprendizaje automático. Se selecciono PyTorch Mobile, TensorFlow Lite, CoreML y ML Kit que son bibliotecas muy usadas en el mundo del aprendizaje automático.

#### *a. PyTorch Mobile*

PyTorch Mobile es una biblioteca de aprendizaje automático bien recibida por una gran comunidad de investigadores debido a su flexibilidad al momento de entrenar los modelos y además su sintaxis es sencilla y fácil de aplicar.

#### *b. TensorFlow Lite*

TensorFlow Lite es amplio y tiene un gran soporte en la comunidad de desarrolladores expertos en la Inteligencia artificial, aparte tiene una fácil integración con TensorFlow, que es una biblioteca de aprendizaje automático más utilizada.

*c. CoreML*

CoreML es una biblioteca versátil en el ecosistema de Apple, debido a su integración nativa con el sistema operativo iOS, lo cual aprovecha las optimizaciones de hardware en dispositivos Apple.

*d. ML Kit*

ML Kit dispone de una amplia gama de funcionalidades en la nube entorno al aprendizaje automático, además cuenta con una gran facilidad de integración con otras herramientas de Firebase, por otra parte, no se requiere tener conocimientos profundos en el aprendizaje automático para usar sus funcionalidades.

Tabla 22. Características técnicas de las bibliotecas de aprendizaje automático para dispositivos móviles

Característica Técnica	PyTorch Mobile	TensorFlow Lite	CoreML	ML Kit
<b>Compatibilidad</b>	Modelos entrenados en PyTorch.	Modelos entrenados TensorFlow, TensorFlow SavedModel, TensorFlow Lite FlatButter.	Modelos de diferentes bibliotecas TensorFlow, Keras y scikit-learn.	Funciones preentrenadas de Google.
<b>Soporte</b>	Dispositivos iOS y Android y	Dispositivos iOS y Android.	Dispositivos iOS.	Dispositivos iOS y Android.
<b>Integración</b>	Biblioteca PyTorch Hub.	Repositorio TensorFlow Hub.	Nativa con el ecosistema Apple.	Servicios de Firebase.
<b>Herramientas</b>	Conversión de modelos PyTorch.	Conversión de modelos TensorFlow.	Core ML tools	No disponible
<b>Optimización</b>	Modelos entrenados.	Modelos entrenados, reducción de tamaño del modelo y acelerar la inferencia.	Modelos entrenados.	Modelos entrenados.
<b>Algoritmos</b>	Machine learning y Deep Learning.	Diversos formatos de modelo de aprendizaje automático, Machine Learning y Deep Learning.	Regresión, agrupación, bayesiano, árboles de decisión, redes neuronales, Support Vector Machine	Machine learning y Deep Learning.
<b>Inferencia</b>	En el mismo dispositivo móvil iOS y Android.	En el mismo dispositivo móvil iOS y Android.	En el mismo dispositivo móvil iOS.	En la nube.
<b>Framework</b>	No soportado	Flutter y React Native.	No soportado.	No soportado.
<b>Documentación</b>	Documentación actualizada.	Documentación actualizada y muy amplia con ejemplos de integración móvil en el repositorio de GitHub.	Documentación actualizada.	Documentación actualizada.
<b>Licencia</b>	Licencia BSD (Berkeley Software Distribution) de uso gratuito con ciertas restricciones	Licencia Apache 2.0 de uso gratuito, amplio uso y modificación de código fuente.	Parte de los términos y condiciones de Apple.	Parte de Firebase de la empresa Google.
<b>Comunidad</b>	Comunidad Activa y en crecimiento.	Comunidad masiva y diversa con un gran número de usuarios, desarrolladores y contribuyentes.	Respalda por la comunidad de desarrolladores de iOS y macOS.	Respalda por Google y una amplia comunidad de desarrolladores que utiliza tecnologías de Google.
<b>Rendimiento</b>	Eficiente.	Eficiente en dispositivos con recursos limitados.	Eficiente en dispositivos iOS.	Eficiente.
<b>Creador</b>	Facebook AI Research (FAIR)	Equipo de TensorFlow en Google.	Desarrollado por Apple.	Desarrollado por el equipo de Firebase de Google.



Tras realizar un análisis y comparación de las características técnicas de cada biblioteca descrita en la Tabla 22, se ha determinado como biblioteca de aprendizaje automático a TensorFlow Lite como la elección más adecuada para desarrollar parte este trabajo de investigación. Fundamentado por:

- ***Optimización de modelos***

Esto es especialmente relevante, ya que permite compactar el modelo ya entrenado reduciendo significativamente el tamaño de del modelo sin afectar su rendimiento y además hará que el modelo consuma menos memoria RAM en el dispositivo móvil.

- ***Inferencia en mismo dispositivo móvil***

Es muy importante la inferencia en el mismo dispositivo móvil dado que garantiza la integridad de la información del usuario para que no sea compartida en servidores remotos y además esto permite tener una preocupación menos con respecto a una conexión a internet.

- ***Framework:***

Disponer de un marco de desarrollo de código abierto multiplataforma es valioso para la creación de aplicaciones nativas como iOS y Android desde una sola base de código.

- ***Integración con TensorFlow Hub***

Es conveniente tener esta integración como aliado ya que al tratarse de aprendizaje automático se debe considerar un punto crucial el dataset para entrenar los modelos y al no contar con los suficientes datos se puede apoyar de esta integración para descargar modelos ya entrenados para aplicar técnicas de Transfer Learning.

- ***Calidad y profundidad de la documentación***

La información técnica detallada y claramente explicada facilita significativamente el proceso de integración de la biblioteca de aprendizaje automático en la aplicación móvil. Esto garantiza una implantación más eficiente.

- ***Rendimiento eficiente***

El buen rendimiento en los modelos de aprendizaje automático en dispositivos móviles o dispositivos con baja capacidad computacional es valioso si se desea realizar la inferencia en el lado del cliente.

- ***Implementación de algoritmos propios***

Tener la facilidad de implementar algoritmos propios basados en Inteligencia Artificial (IA) es un importante dato que se puede ajustar a la regla de negocio que se necesite implementar.

### **3.1.3 Entornos de desarrollo para aprendizaje automático**

Para definir el mejor entorno de desarrollo de modelos de aprendizaje automático importante considerar características como: soporte de bibliotecas, herramientas de visualización y recursos disponibles. Dichas características ayudaran a definir el entorno adecuado para el desarrollo del proyecto.

Tabla 23. Comparación de entornos de desarrollo de aprendizaje automático

<b>Característica</b>	<b>Jupyter Notebooks</b>	<b>Spyder</b>	<b>RStudio</b>
<b>Interfaz</b>	Interfaz basada en el navegador.	Entorno de desarrollo integrado (IDE).	Entorno de desarrollo integrado (IDE).
<b>Lenguaje</b>	Múltiples lenguajes de programación, destacado el lenguaje Python.	Específico para el lenguaje de programación Python.	Lenguaje de programación R.
<b>Herramientas de exploración</b>	Facil integración con paquetes de visualización y análisis exploratorio de datos	Exploración de variables	Integración con paquetes populares de R para estadística y visualización.
<b>Entornos virtuales</b>	Compatibilidad con entornos virtuales.	Compatibilidad con entornos virtuales.	Compatibilidad con entornos virtuales.

Característica	Jupyter Notebooks	Spyder	RStudio
<b>Soporte</b>	Bibliotecas de aprendizaje automatico como TensorFlow, PyTorch, scikit-learn, Keras entre otras.	Bibliotecas de aprendizaje automatico como TensorFlow, scikit-learn y PyTorch.	Bibliotecas de R para aprendizaje automatico.
<b>Ejecución de código por celdas</b>	Si.	Si.	Si.

Mediante el análisis realizado en la Tabla 23, detallando las características que posee cada entorno de desarrollo dedicado al aprendizaje automático, se decide optar por el entorno de desarrollo Jupyter Notebook ya que este permite el desarrollo y ejecución en el navegador web, proporcionando una preocupación menos de la instalación de un IDE de desarrollo.

Además, admite el uso de bibliotecas de aprendizaje automático como TensorFlow lo cual es importante para el desarrollo de este trabajo de investigación, por último, permite la visualización y análisis exploratorio de los datos, lo cual es importante dado que permite conocer mediante gráficos el comportamiento de los datos.

### 3.1.4 Tecnologías para el desarrollo móvil

Para definir la mejor tecnología para el desarrollo de aplicaciones móviles, es importante considerar características como el soporte para bibliotecas de aprendizaje automatico, el lenguaje de programación, las herramientas y recursos disponibles. Dichas características ayudaran a determina la mejor tecnología que se ajuste a las necesidades de este trabajo de investigación.

Tabla 24. Comparación entre tecnologías de desarrollo móvil

Característica	Android	Flutter	React Native
<b>Código</b>	Open Source.	Open source.	Open source.
<b>Lenguaje de programación</b>	Kotlin y Java.	Dart.	JavaScript.
<b>Tipo de aplicación</b>	Android.	Multiplataforma.	Multiplataforma.
<b>Rendimiento</b>	Compilación directa a código nativo, lo que puede proporcionar una ejecución más eficiente.	Compilación Ahead of Time (AOT), lo que mejora el rendimiento.	Compilación Just intime para un inicio rápido de la aplicación.
<b>Integración</b>	Acceso completo a las APIs y características nativas de Android.	Acceso a las APIs nativas.	Menos control de las APIs nativas.

Característica	Android	Flutter	React Native
<b>Soporte a bibliotecas de aprendizaje automático</b>	TensorFlow Lite, PyTorch Mobile y ML Kit.	TensorFlow Lite, PyTorch Mobile y ML kit.	TensorFlow Lite, PyTorch Mobile y ML Kit.
<b>Curva de aprendizaje</b>	Mas pronunciada para principiantes.	Pronunciada para principiantes.	Alta.

Mediante el análisis realizado en la Tabla 24, se realizó un análisis comparativo entre las distintas tecnologías de desarrollo móvil, se decidió optar por Android dado las siguientes características que nos facilita, el rendimiento al contar con una compilación directa con las APIs nativas lo que proporciona una ejecución más eficiente y considerando que este proyecto implica implementar modelos de aprendizaje automático, la eficiencia y rendimiento es un punto clave.

Además, la curva de aprendizaje es orientada para principiantes lo cual facilitara la ejecución y desarrollo de la aplicación móvil, por otra parte, la integración de biblioteca de aprendizaje automatico como TensorFlow Lite es un punto importante dado que se eligió en este trabajo de investigación a TensorFlow Lite como parte del desarrollo de los modelos de aprendizaje automatico.

### 3.1.5 Metodología para el desarrollo móvil

El siguiente cuadro comparativo se describe las características más relevantes de cada una de las metodologías ágiles preseleccionadas como Mobile D [29], Extreme Programming (XP) [30] y Scrumban [30] para su posterior selección de una metodología ágil para el desarrollo de la aplicación móvil.

Tabla 25. Metodologías ágiles para desarrollo móvil

Característica	Mobile D	XP	Scrumban
Tamaño de equipo	Menos de 10 personas	8 personas	Cualquier tamaño de equipo.
Tipos de proyectos	Proyectos cortos	Proyectos a corto plazo.	Proyectos cortos
Estilo de desarrollo	Iterativo e incremental.	Flexible a cambios.	Ágil e iterativo.
Consideraciones móviles	Limitación de memoria, variedad de plataformas y características particulares en términos móviles.	Variedad de plataformas.	No especificado.
Fases	<ul style="list-style-type: none"> <li>• Exploración</li> <li>• Inicialización</li> <li>• Producción</li> <li>• Estabilización</li> <li>• Pruebas</li> </ul>	<ul style="list-style-type: none"> <li>• Planificación</li> <li>• Diseño</li> <li>• Desarrollo</li> <li>• Pruebas</li> </ul>	<ul style="list-style-type: none"> <li>• Sprint planning</li> <li>• Flujo continuo</li> <li>• Límites de work in Progress (WIP)</li> <li>• Entrega continua</li> </ul>
Dificultad	Media.	Alta.	Media.
Tiempo de entrega	10 semanas.	4 semanas.	4 semanas.

Una vez expuesto las características de las diferentes metodologías ágiles se decide optar por Mobile D, ya que esta metodología no requiere un alto nivel de disciplina técnica como lo requiere XP, por otra parte Mobile D se ajusta al tipo de trabajo que se pretende desarrollar tales como: orientada a desarrollar proyectos cortos, grupos de trabajo pequeños, rapidez en el desarrollo siendo susceptible a cambios, desarrollo basado en pruebas y considera características particulares en dispositivos móviles para garantizar un producto de calidad.

### 3.2 Desarrollo de los modelos de aprendizaje automático

Para el desarrollo de la propuesta se planteó desarrollar una aplicación móvil que integre modelos de aprendizaje automático para determinar el estado legal y estado estético del vehículo. Para cumplir el objetivo general se procede a dividir en 2 procesos: Proceso de detección y extracción de la placa vehicular para determinar el estado legal del vehículo y finalmente el proceso de detección de irregularidades del exterior del vehículo para determinar el estado estético.

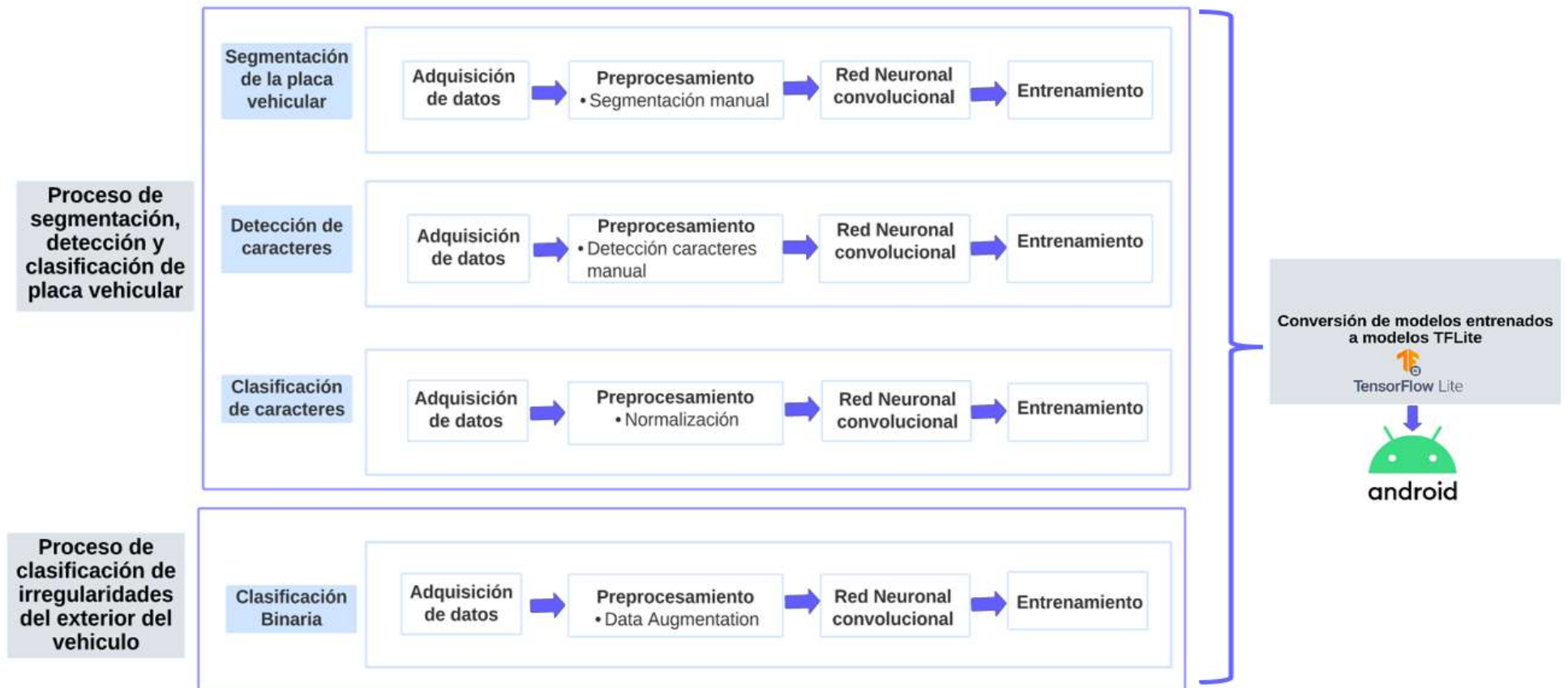


Figura 18. Esquema de operatividad de los modelos

### 3.2.1 Proceso de segmentación, detección y clasificación de la placa vehicular

Una vez establecido el esquema de desarrollo de los modelos de aprendizaje automático para el proceso de detección y extracción de la placa, se debe abordar cada punto descrito en la Figura 18, dicho proceso se basa en la obtención de la placa del vehículo mediante la aplicación de diferentes modelos de aprendizaje automático conectados encascada con el objetivo de extraer los caracteres de la placa para su posterior consulta en la base de datos de la Agencia Nacional de Transito (ANT).

#### a. *Segmentación de la placa vehicular*

- *Adquisición de datos*

Dado la naturaleza del problema se encuentra un dataset en la plataforma de Kaggle llamado Car License Plate Detection [31], donde el dataset cuenta con 433 imágenes de vehículos tomadas en distintas condiciones, ángulos, marcas y modelos lo cual es importante para el primer modelo delegado de segmentar la región de interés.

Uno de los factores crucial que impacta en el mundo del Machine Learning, Deep Learning y aprendizaje automático es la generalización del modelo, donde el modelo es útil en situaciones del mundo real, hay varios criterios para generalizar un modelo como, por ejemplo:

- Monitoreo del rendimiento.
- Optimización de los hiperparámetros del modelo.
- Aumento de datos.
- Regularización.
- División de datos.
- Transferencia de aprendizaje.

Entre los distintos criterios se optó por el aumento de datos porque es importante considerar que se plantea trabajar con una arquitectura profunda de Deep Learning y lo más conveniente es suministrar una gran cantidad de datos para generalizar y evitar fenómenos como el overfitting y underfitting. Se procede a capturar fotos de la parte frontal y posterior de distintos vehículos parqueados en los predios de la Universidad Técnica de Ambato, de las calles de la ciudad de Ambato y de la plataforma Facebook.

Como resultado se obtiene 540 imágenes sumadas al dataset de Kaggle se obtiene en total 973 imágenes.

- ***Preprocesamiento***

Una vez obtenido un dataset, es importante realizar un preprocesamiento de los datos, esto se lo hace con el fin de eliminar ruido, mejorar la calidad de los datos, compatibilidad con el modelo, normalización y estandarización, etiquetación.

Existen varios enfoques de modelos aprendizaje automático pero el primer modelo planteado en este trabajo de investigación se acordó realizar un modelo de segmentación, existen varios tipos de segmentación como, por ejemplo:

### **Segmentación semántica**

Estos modelos se basan en la utilización de Redes Neuronales Convolucionales para la clasificación de pixel por pixel, es decir el modelo realiza una clasificación a nivel de pixeles donde determina si el pixel pertenece o no pertenece a la región de interés que se desea detectar.

Dado por la naturaleza del problema es apropiado utilizar una segmentación semántica dado que la tarea es segmentar el área de interés en este caso la placa del vehículo. Una vez discutido que tipo de segmentación se aplicara para el primer modelo es momento de preparar los datos para ingresar al modelo aprendizaje automático, en cuestión de aprendizaje automático supervisado la tarea se concentra en enseñar lo suficientemente al modelo que tipo de tarea va a realizar, se debe tener un dataset ya etiquetado es decir segmentado con la región de interés para entrenar el modelo.

Al contar con un dataset, pero no segmentado se debe realizar la segmentación manual para cual se utiliza una herramienta que facilite el proceso de segmentación, Labelme [32] es un software que se utilizar para segmentación y otras tareas. El proceso de segmentación manual duro aproximadamente 2 días.

Labelme crea archivos externos con extensión XML para especificar atributos como el nombre la imagen, ancho, alto y sobre todo los puntos X e Y donde se seleccionó la segmentación, una vez realizado esta fase se necesita convertir esos archivos XML en imágenes binarias mediante un script en Python donde la región de interés se concentra



con píxeles de valor 255 y la región de no interés con píxeles de valor 0. Ver Anexo B.

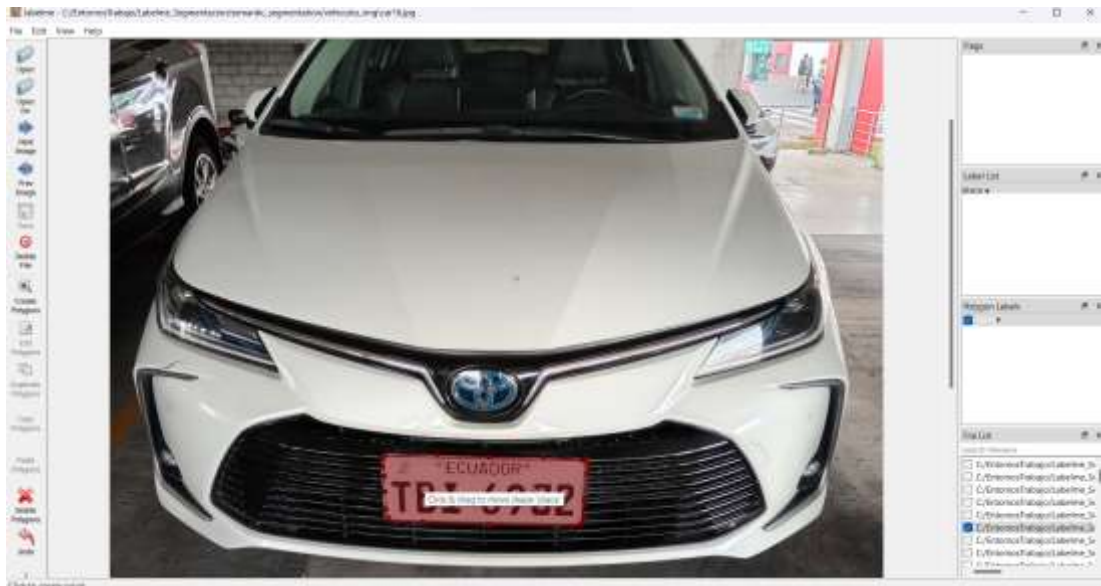


Figura 19. Segmentación mediante el software Labelme

- **Algoritmos de aprendizaje automático**

Existen diferentes arquitecturas de modelos de segmentación semántica basadas en Redes Neuronales Convolucionales, pero en esta investigación se abordará en concreto 2 arquitecturas U-Net [33] y ResUNet [34] para su posterior selección de una arquitectura.

Tabla 26. Arquitecturas de segmentación

Característica	U-Net	ResUNet
<b>Arquitectura</b>	Codificación y decodificación.	Codificación, decodificación e incorporación de bloque residuales.
<b>Conexiones laterales</b>	Capas de codificación directamente conectada a las capas de decodificación.	Capas de codificación directamente conectada a las capas de decodificación.
<b>Contenido de capas</b>	Normalización y activación ReLU.	Normalización y activación ReLU.
<b>Tamaño Dataset</b>	Datasets pequeños y no muy complejos.	Datasets grandes y complejos
<b>Profundidad de capas</b>	Profundo.	Muy Profundo.

Una vez descrito las características de las arquitecturas de se decide optar por la arquitectura ResUNet, por las siguientes características: codificadores, decodificadores y la incorporación de bloques residuales que ayudan a minimizar el desvanecimiento de gradiente en las capas primarias cuando se hace la retropropagación.

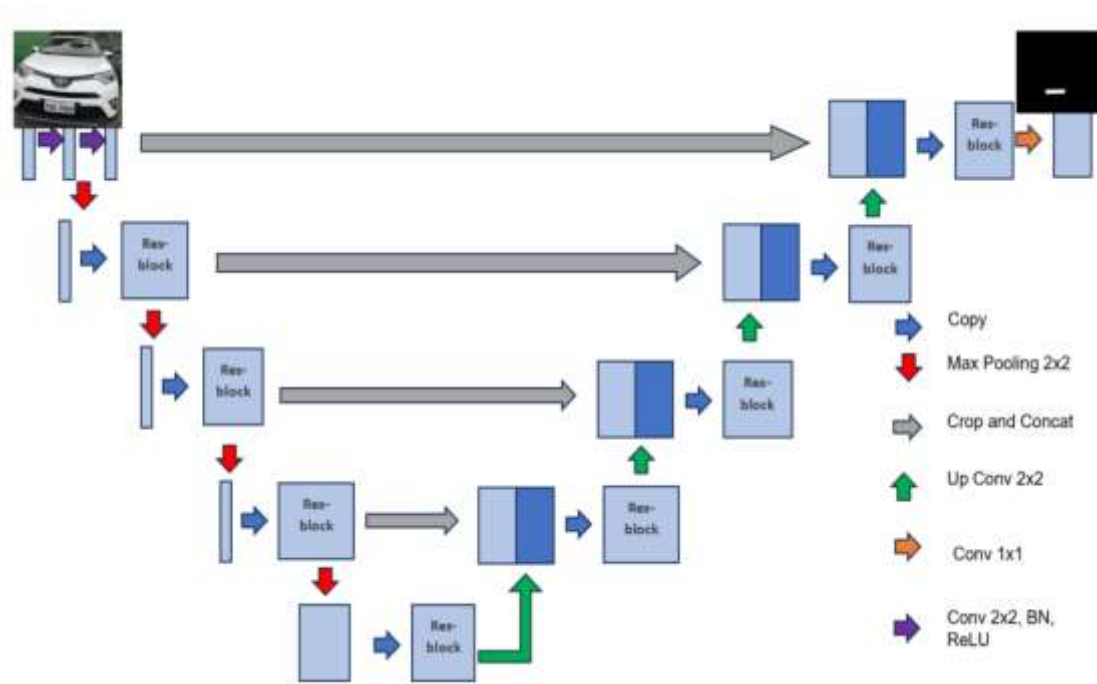


Figura 20. Arquitectura ResUNet [41]

Como se puede visualizar en la Figura 20, la arquitectura ResUNet es robusta dado que consta de bloques residuales, codificadores (MaxPool2D) y decodificados (UpSampling2D) pero además se realizan operaciones de convolución, normalización y activación (ReLU). Cuando la imagen de entrada ingresa a la primera capa se aplica las operaciones anteriormente mencionadas y a la vez se adhieren los codificadores o en ingles encoders. El objetivo de estas operaciones es reducir la resolución espacial de la imagen de entrada y a la vez se extraen características abstractas a medida que ingresa en la profundidad del modelo. Luego, cuando llega al cuello de botella se realiza un proceso inverso al anteriormente descrito, entra en acción los decodificadores o en ingles decoders. La funcionalidad de los decoders erradica en aumentar gradualmente la resolución espacial de imagen de entrada y se combinan con los bloques residuales que contienen características del proceso de codificación. Finalmente, en la última capa se agrega una función de activación Sigmoidal que

devuelve un valor en 0 y 1 que representa la probabilidad de que cada pixel pertenece o no a la clase de interés, en pocas palabras se segmenta el área de interés en este caso la placa.

Se utiliza el lenguaje de programación Python con las librerías de TensorFlow y Keras en el entorno de desarrollo Jupyter Notebook, para implementar la arquitectura anteriormente mencionada. Ver Anexo C, D y E.

- ***Entrenamiento***

La fase de entrenamiento se debe considerar aspectos como la memoria del servidor local, ya que cargará las imágenes para su posterior procesamiento, lo cual es importante cargar de manera dinámica las imágenes en el momento del entrenamiento para evitar colapsar la memoria del servidor local. Para ello se genera un archivo csv con los paths de las imágenes y los paths de las máscaras segmentadas manualmente, se crea una clase en Python que realice el proceso de cargar de imágenes dinámicas desde el archivo csv al momento del entrenar. Ver Anexo F.

Se procede al entrenamiento de la red se divide el dataset 85% en entrenamiento, 10% en validación y el 5% en test. Durante el proceso de entrenamiento es importante visualizar las métricas de precisión y de pérdida para una detección temprana de overfitting y underfitting para lo cual se apoya de la herramienta de TensorBoard para monitorizar el proceso de entrenamiento.

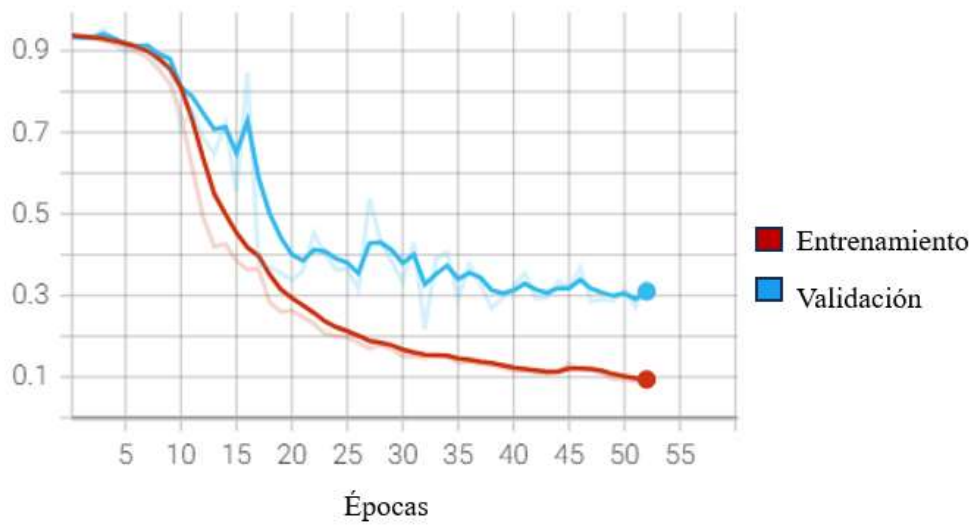


Figura 21. Métricas de pérdida en entrenamiento y validación

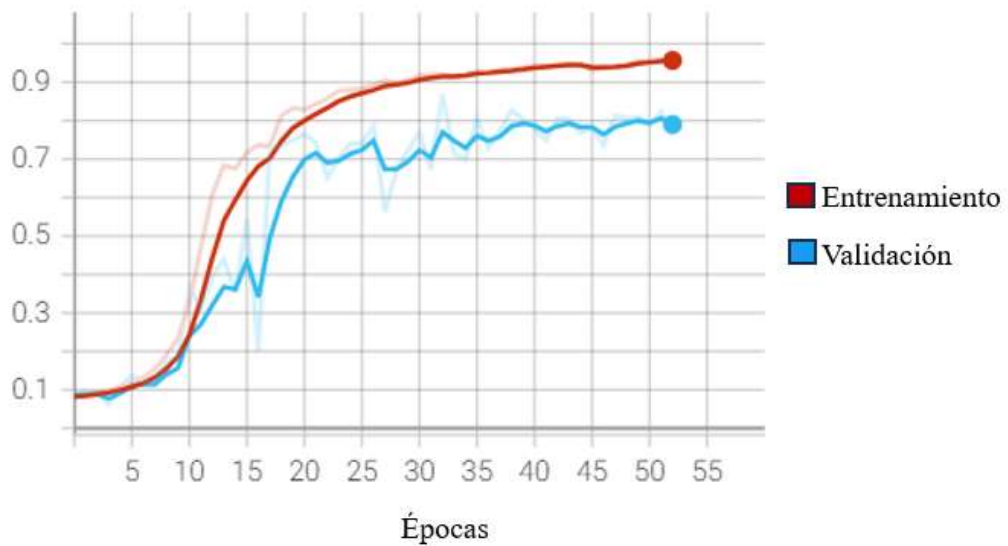


Figura 22. Métricas de precisión en entrenamiento y validación

El entrenamiento duro aproximadamente 2 hora, se estableció 52 épocas y batch de 21 imágenes por época. En la Figura 22, se puede observar la métrica de precisión en entrenamiento tiene un 95% y en validación tiene un 80% de precisión. En la Figura 21 Las métricas de pérdida en entrenamiento es de 0.09% y la métrica de pérdida en validación es de 0.30% lo que se deduce que existe overfitting.

Existen técnicas para mitigar el overfitting entre los más recomendados es el data augmentation o aumento de datos con librerías específicas de Python, la idea principal de esta técnica es generar nuevas imágenes a partir del dataset original, para obtener

nuevas muestras para el entrenamiento. Se aplica operaciones como: rotación, inversión horizontal y vertical, Zoom y desplazamiento para lo cual se creó un script en Python para realizar el data Augmentation. Ver Anexo G.

Aplicando la técnica de aumento de datos se obtiene un total de 5.838 imágenes resultantes donde se divide en 85% para entrenamiento, el 10% para validación y el 5% para test.

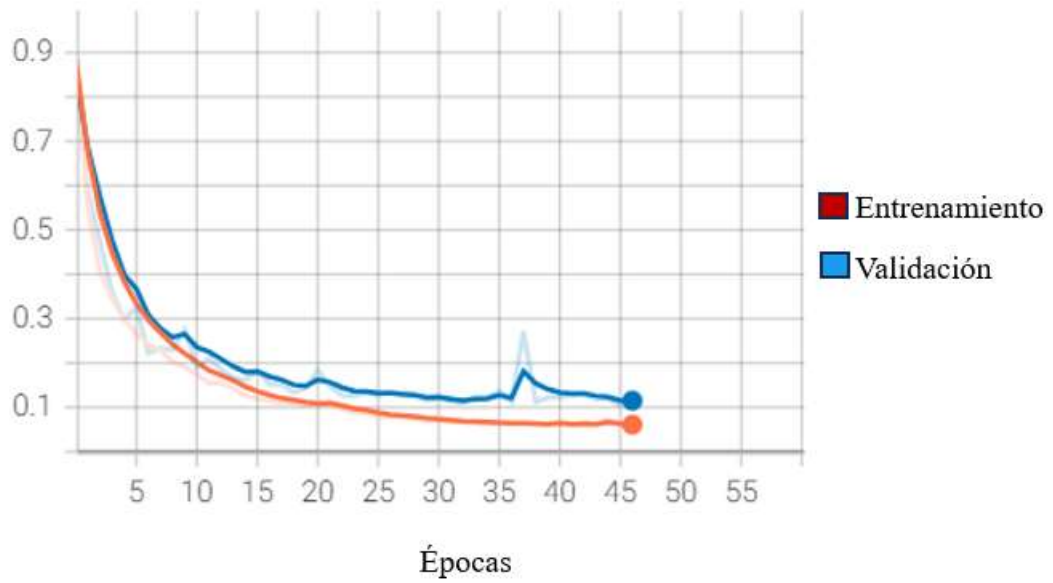


Figura 23. Métricas de pérdida en entrenamiento y validación con data augmentation

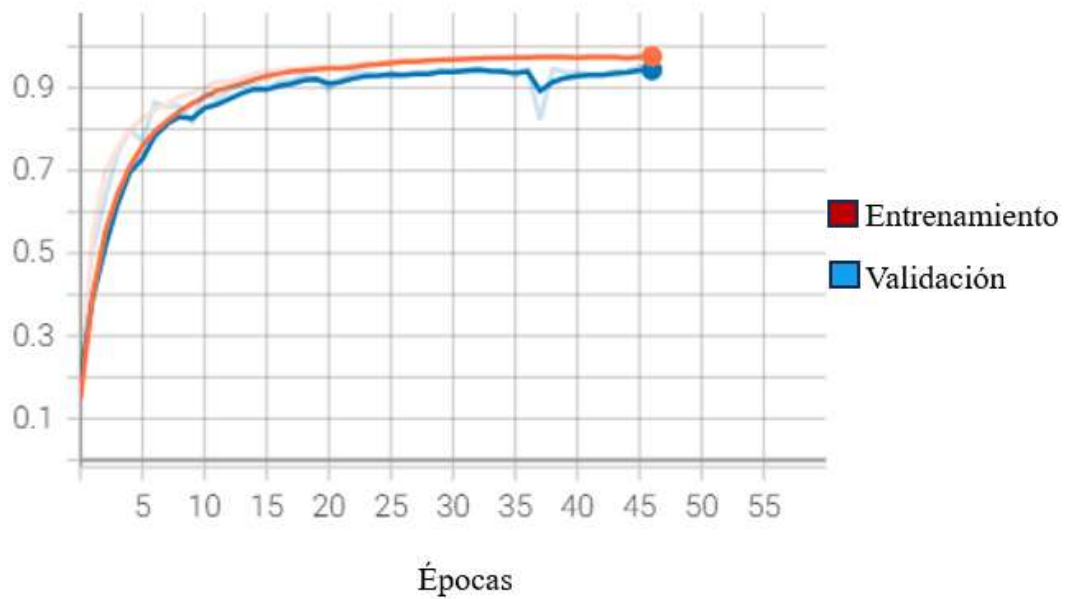


Figura 24. Métricas de precisión en entrenamiento y validación con data augmentation

El entrenamiento duro 15 horas aproximadamente, se estableció 48 épocas con batch de 155 imágenes por época, como se muestra en las Figuras 23 y 24 en las métricas de pérdida se establecen en valores cercanos entre 0.05% y 0.10% y las métricas de precisión se establecen en valores cercanos a 94% y 97%. Los resultados obtenidos se pueden ver un mejor rendimiento del modelo.

**b. Detección de caracteres**

- **Adquisición de datos**

Para la detección de caracteres se busca un dataset que facilite la detección de caracteres de la placa, se indago en las distintas plataformas de ciencia de datos para adquirir un dataset. Se determina que no existe un dataset que se ajuste a los requerimientos. Se procede a capturar fotografías de distintas placas vehiculares para construir el dataset conformado por 451 imágenes.

- **Preprocesamiento**

Los modelos de detección de objetos tienen como finalidad determinar la ubicación del objeto, para entrenar un modelo de detección de objetos es necesario contar con un dataset etiquetado. Labelme facilita dicha tarea para etiquetar los objetos de interés mediante cuadros delimitares o en inglés bounding box.

El proceso de etiquetado se tomó aproximadamente 2 días con el software Labelme como se puede observar en la Figura 25.



Figura 25. Cuadros delimitadores con Labelme

- *Algoritmos de detección de Objetos*

El dataset obtenido es pequeño para realizar un entrenamiento en una Red Neuronal Convolutiva fundamentada en detección de objetos. Una técnica que puede ajustarse a este tipo de situaciones es la transferencia de aprendizaje o Transfer Learning en ingles debido a las limitaciones entorno al dataset obtenido.

En detección de objetos, hay varios modelos, pero en concreto se tomó a You Only Look Once (YOLO) [35] y SSD-MobileNetV2[36] para su comparativa.

Tabla 27. comparativa de las arquitecturas YOLO y SSD-MobileNetV2

Característica	YOLO	SSD-MobileNetV2
<b>Arquitectura</b>	Única Red Neuronal.	Combina la arquitectura MobileNetV2 y Single-Short Detector (SSD)
<b>Precisión</b>	Alta.	Alta.
<b>Velocidad</b>	Eficiente.	Eficiente.
<b>Tamaño</b>	Ligero.	Extremadamente Ligero.
<b>Facilidad de Implementación</b>	TensorFlow y PyTorch.	TensorFlow y PyTorch.
<b>Transfer Learning</b>	Si.	Si.
<b>Dominio de Aplicación</b>	Hardware limitado.	Hardware limitado.

En la Tabla 27 se discute las arquitecturas YOLO y SSD-MobileNetV2 y se selecciona a la arquitectura SSD-MobileNetV2 por las siguientes razones: más ligero, eficiente en dispositivos móviles sin comprometer la precisión, fácil de implementar con TensorFlow y permite realizar Transfer Learning.

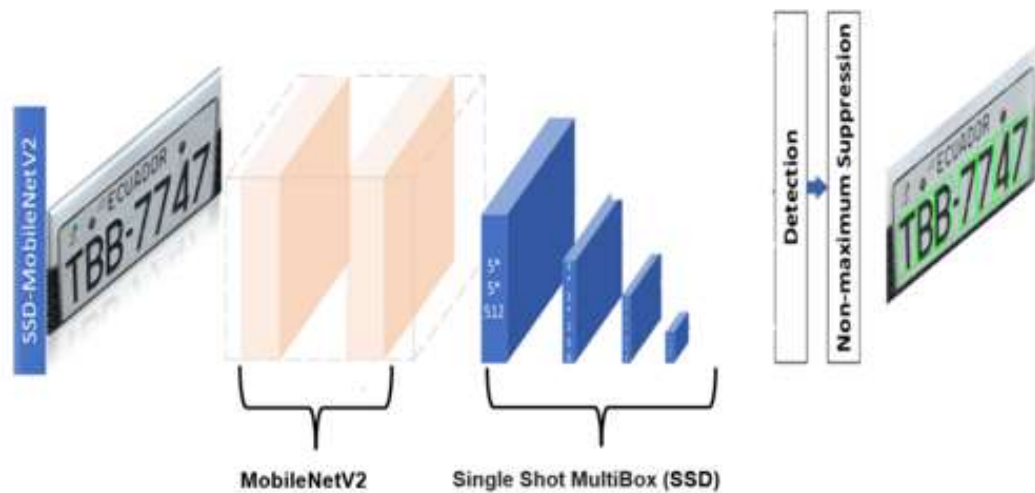


Figura 26. Transfer Learning con SSD-MobileNetV2 para detección de caracteres

La arquitectura SSD-MobileNetV2 [36] toma la arquitectura SSD para detección de objetos y la combina con la arquitectura MobileNetV2 que es una red convolucional para extraer características abstractas y además es capaz de extraer propiedades complejas gracias a la inclusión de bloques Inverted Residual que incluyen operaciones de expansión lineal, convolución profunda y conexión de atajo.

$$Salida = Entrada + ReLU(Conv(ReLU(ExpansiónLineal (Entrada))))$$



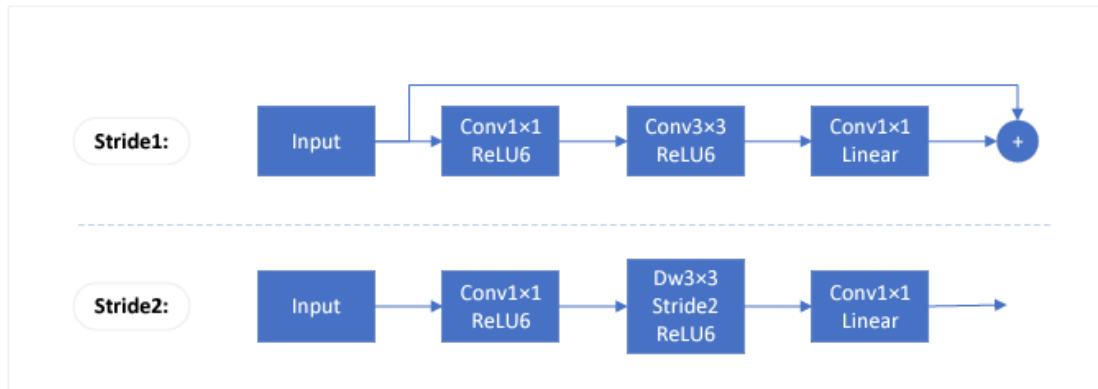


Figura 27. Bloque Inverted Residual de MobileNetV2

Transfer Learning es útil cuando se tiene un dataset limitado, parte de un modelo previamente entrenado con otro dataset para adaptar a una tarea de detección en específico. Se utiliza el lenguaje de programación Python y el entorno de desarrollo Jupyter Notebook para aplicar la técnica de Transfer Learning que proporciona la documentación de TensorFlow [37]. Ver Anexo H.

- **Entrenamiento**

En la fase de entrenamiento se especifica 20.000 pasos y la división del dataset se divide en 80% entrenamiento y 20% en test. La duración del entrenamiento duro aproximadamente 5 horas y el seguimiento del proceso de entrenamiento se realiza con TensorBoard.

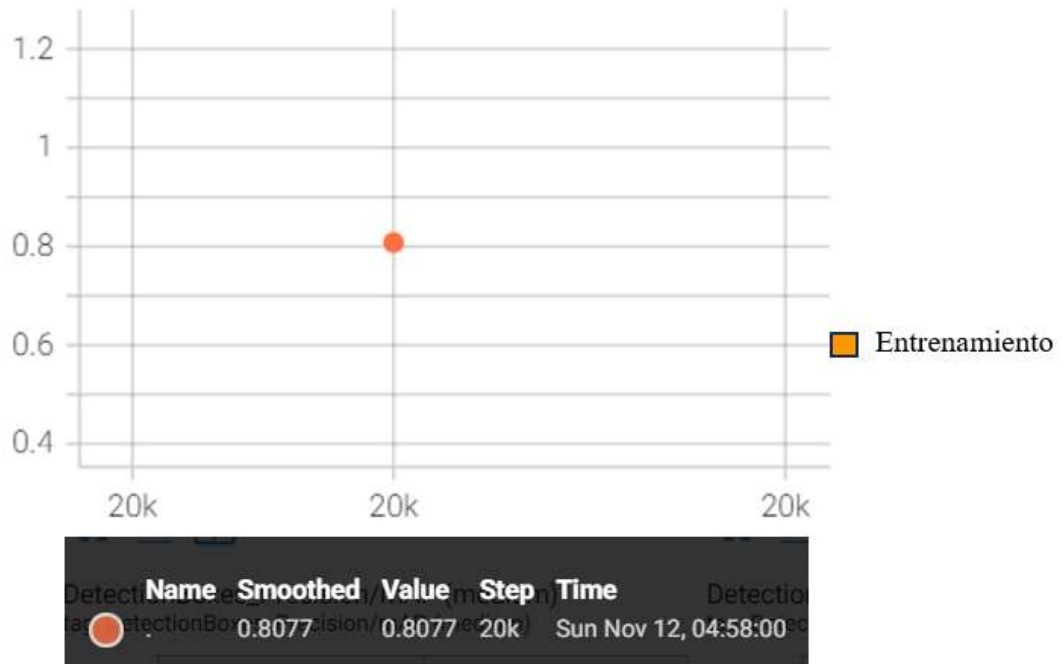


Figura 28. Métrica de precisión en el entrenamiento



Figura 29. Métrica de pérdida en el entrenamiento

c. *Clasificación de caracteres*

- *Adquisición de datos*

Para la clasificación de caracteres es importante poseer un dataset robusto, para cual se investiga un dataset que contenga caracteres alfanuméricos, se busca en las plataformas Kaggle y Data Science central, donde se encontró en la plataforma Kaggle un dataset cuantioso llamado Chars74K [38] con más de 36.000 imágenes, cada ejemplar cuenta con más de 1.000 imágenes en binario.

- *Preprocesamiento*

Para la clasificación de caracteres es necesario contar con dataset numero debido a que se realizara una clasificación entre 36 posibles clases, para obtener un mejor resultado se integra la técnica de Data Augmentation para aumentar el dataset original, se utiliza librerías específicas de TensorFlow para aplicar operaciones como: zoom, desplazamiento vertical y horizontal, cambio de ángulo de visión y escalado de las imágenes, obteniendo aproximadamente 180.000 imágenes. Ver Anexo I.

- *Algoritmos de aprendizaje automático*

Para este problema de clasificación se optó por una Red Neuronal Convolutacional ya que se cuenta con los datos suficientes para entrenar una red desde cero.

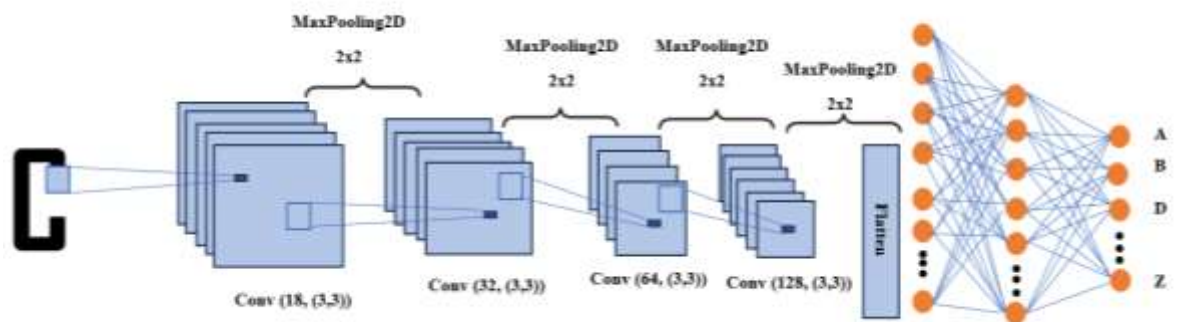


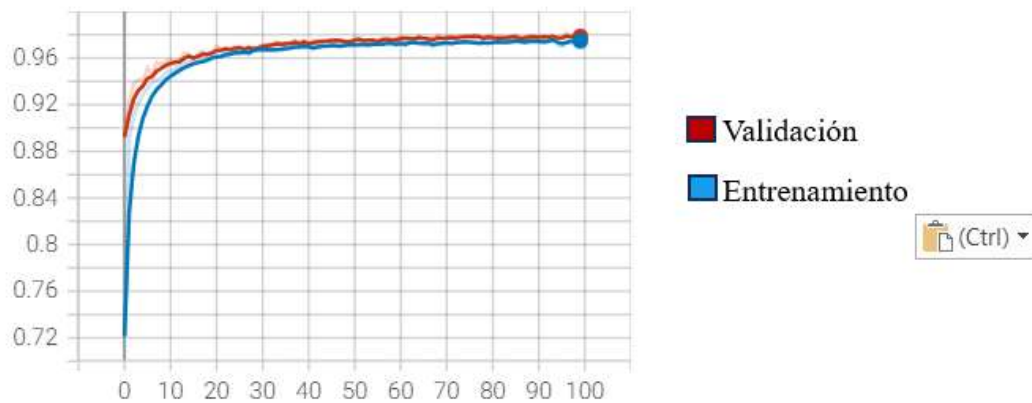
Figura 30.Arquitectura de la CNN

Como indica la Figura 30 esta arquitectura está basada en 4 operaciones de convolución la primera convolución obtiene 18 mapas de características, la segunda convolución obtiene 32 mapas de características, la tercera convolución obtiene 63 mapas de características y la cuarta convolución obtiene 128 mapas de características todas las convoluciones especifican un kernel de 3x3 y una función de activación ReLU. Luego de aplicar cada convolución se aplica un MaxPooling2D de 2x2 para reducir la resolución espacial de imagen para obtener características más abstractas a medida que se interna en la profundidad de la red. El resultado del último MaxPooling es aplanado e ingresa a la red neuronal simple que cuenta con 2 capas ocultas de 250 y 350 neuronas y finalmente en la capa de salida se agrega 36 neuronas dado por el número de clases a predecir con una función Softmax.

La implementación de la Red Neuronal Convolucional se realiza con el lenguaje de programación Python y las librerías de TensorFlow y keras. Ver Anexo J.

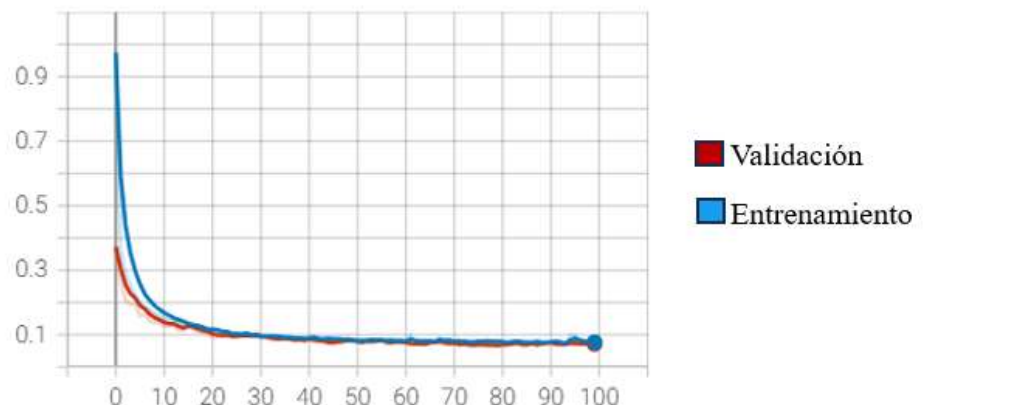
- ***Entrenamiento***

En el proceso de entrenamiento se dividió en dataset en 80% entrenamiento, 10 validación y test el 10% además se configuro con los siguientes parámetros de entrenamiento: 100 épocas y batch de 800 imágenes por época.



Name	Smoothed Value	Value	Step	Time	Relative
cnn2G\train	0.9742	0.9743	98	Wed Nov 22, 19:01:19	5h 46m 58s
cnn2G\validation	0.9783	0.9784	98	Wed Nov 22, 19:01:19	5h 46m 58s

Figura 31. Métricas de precisión en entrenamiento y validación de la CNN



Name	Smoothed Value	Value	Step	Time	Relative
cnn2G\train	0.07657	0.07171	99	Wed Nov 22, 19:03:47	5h 49m 26s
cnn2G\validation	0.07215	0.07179	99	Wed Nov 22, 19:03:47	5h 49m 26s

Figura 32. Métricas de pérdida en entrenamiento y validación de la CNN

El entrenamiento duro aproximadamente 6 horas donde las métricas de precisión tanto en entrenamiento y validación se acentúan en el 97% y las métricas de pérdida tanto en entrenamiento como en validación se acentúan en el 7%, lo cual es bueno ya que indica que el modelo generalizo y es capaz de realizar predicciones con datos nunca antes vistos. Para describir mejor la generalización del modelo se realiza la matriz de confusión con los datos de test.



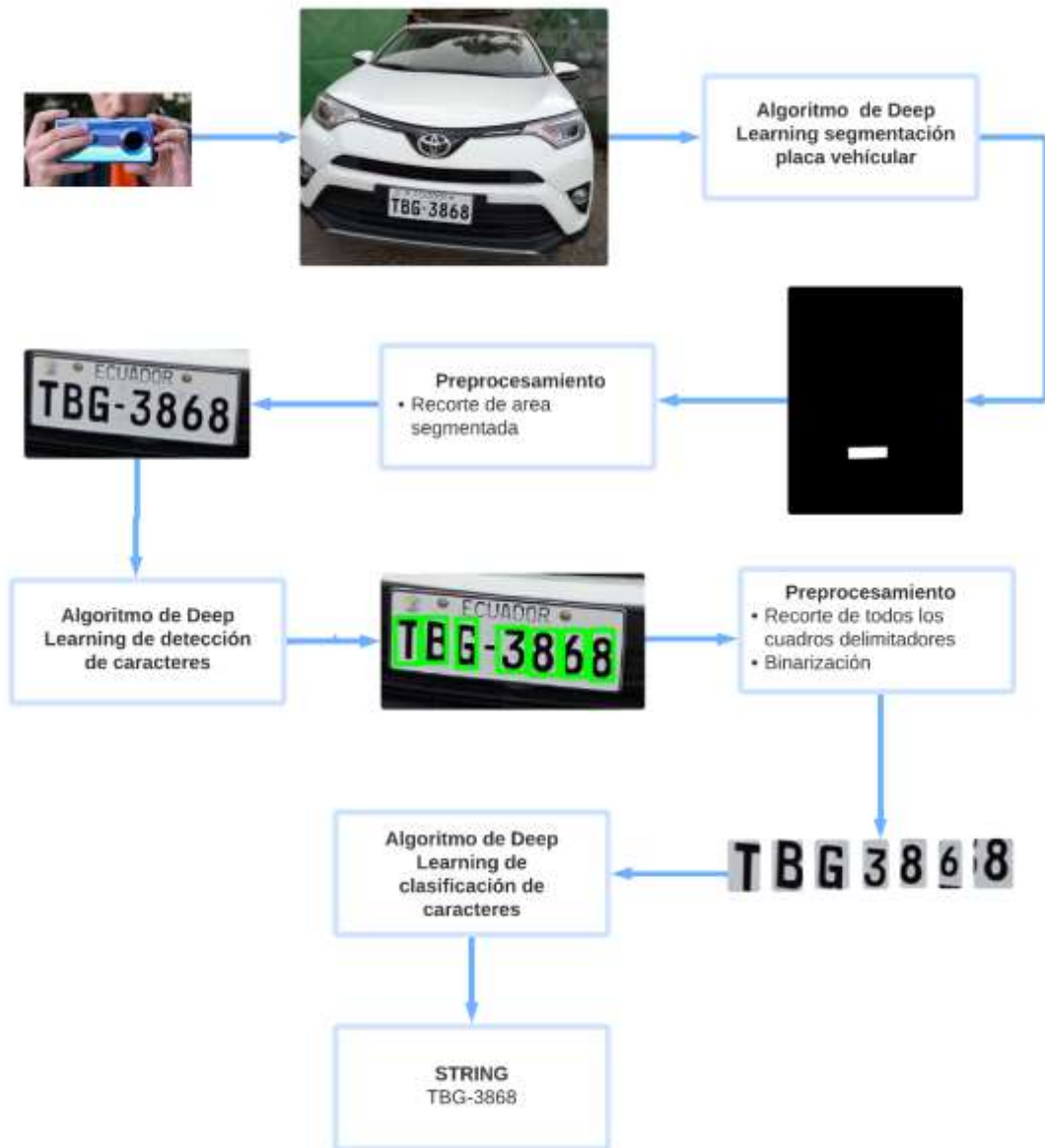


Figura 34. Flujo de trabajo de los modelos aprendizaje automático encascada

En la Figura 34 se puede observar el flujo de trabajo en conjunto de los modelos de aprendizaje automático para la obtención de la placa vehicular. La conversión de los modelos en modelos TensorFlow Lite se realiza con el mismo de TensorFlow. Ver Anexo K.

### 3.2.2 Proceso de clasificación de irregularidades del exterior del vehículo

El proceso de clasificación de irregularidades del exterior del vehículo se plantea establecer una Red Neuronal Convolutiva basada en una clasificación binaria para detectar irregularidades en la superficie del exterior del vehículo, para iniciar con esta tarea se debe empezar con la adquisición de datos.

#### *a. Clasificación binaria*

- *Adquisición de datos*

Dado la naturaleza del problema que se plantea solucionar se necesita adquirir un dataset que contenga imágenes de vehículos con irregularidades y sin irregularidades en la superficie exterior del vehículo. Se indaga en la web posibles dataset con estas características, se encuentra un dataset llamado CarDD [39] que contiene 700 imágenes de vehículos con irregularidades, pero no contiene imágenes de vehículos sin irregularidades para lo cual se procede a recopilar 700 imágenes de vehículos de la plataforma Facebook. Una vez recopilado las imágenes se procede a construir un dataset conformado por las dos clases flamantes y no flamantes cada clase conformada por 700 imágenes dado que es importante contar con un dataset balanceado para evitar un posible overfitting durante en entrenamiento.

- *Preprocesamiento de datos*

El dataset obtenido se cuenta con 1.400 imágenes, pero aun representa un dataset pequeño por lo tanto se debe hacer un Data Augmentation para obtener un dataset más robusto. Se crea un script en Python que utiliza la librería Albumentations para crear nuevas imágenes partir del dataset obtenido, esta librería realiza operaciones como recorte de centro, rotación aleatoria de 90 grados, distorsión de cuadrícula, volteo horizontal y volteo vertical de modo que se obtiene un dataset con 8.000. Ver Anexo L.





Figura 35. Data Augmentation

- *Algoritmo de aprendizaje automatico*

Para este caso se decide crear un Red Neuronal Convolutiva tradicional dado que para esta tarea se cuenta con un dataset robusto y además se trata una clasificación binaria por consecuencia es un dataset que encaja para este tipo de Red Neuronal Convolutiva.

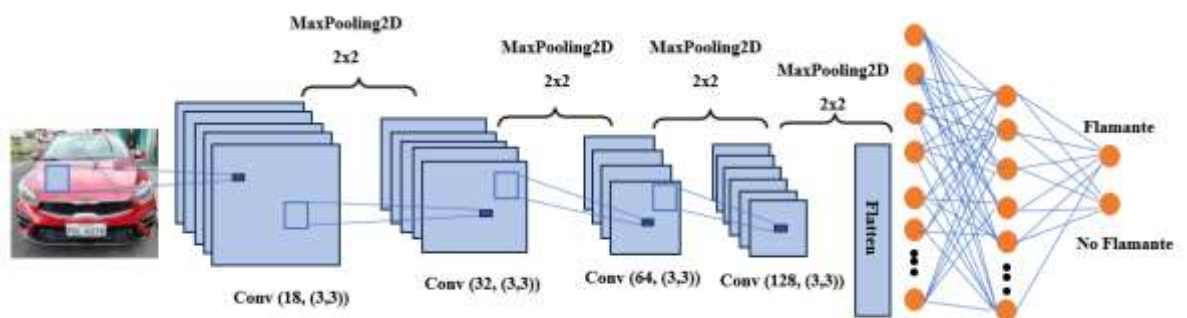


Figura 36. Arquitectura de Red Neuronal Convolutiva

La Red Neuronal Convolutiva está compuesta por 4 capas de convolución con filtros o kernels de 3x3 para apreciar detalles más finos en el entrenamiento y además luego de cada operación de convolución se aplican operaciones de MaxPooling para reducir la dimensión espacial de la imagen, las 4 capas convolutivas retornan 18, 32, 64, 128 mapas de características. Los mapas característicos de la última capa de convolución son aplanados en un vector unidimensional para finalmente ser introducidos en una Red Neuronal de 100, 50, 10 y 2 neuronas. Las capas ocultas de la red tienen una función de activación ReLU y la capa de salida que posee 2 neuronas tiene una función de activación Softmax porque se pretende tener una probabilidad dada en función de las 2 clases. Ver Anexo M.

- **Entrenamiento**

Para en proceso de entrenamiento se aplica una normalización de los pixeles para que Red Neuronal Convolucional converja más rápido durante el entrenamiento de manera que contribuya al descenso del gradiente ya que se establecen los valores entre 0 y 1 y hace que la optimización de función de perdida sea más rápida.

El entrenamiento se tomó aproximadamente 6 horas con una dataset dividido en 80% en entrenamiento, 10% en validación y 10% en test además se estableció los siguientes parámetros 60 épocas y batch de 32.

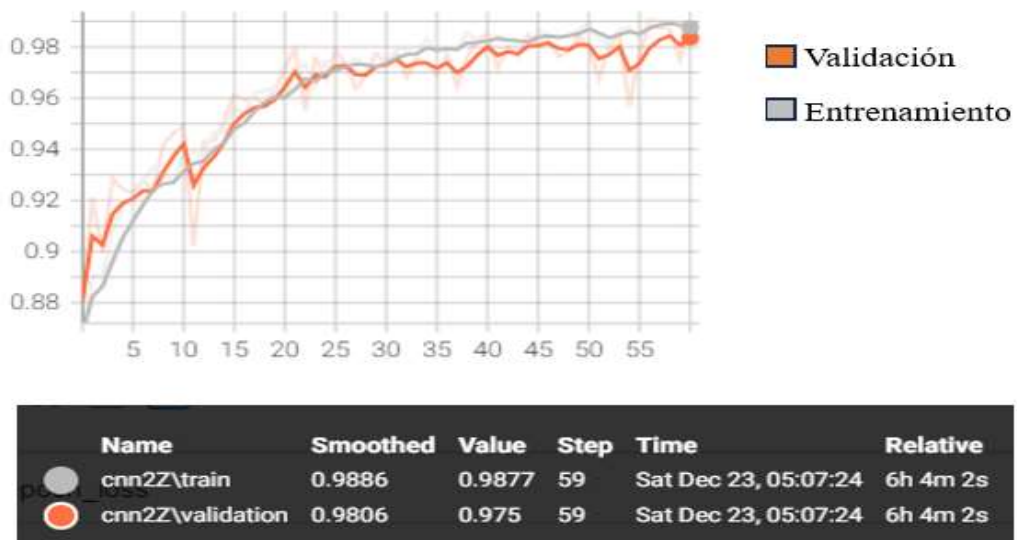


Figura 37. Métricas de precisión en el entrenamiento



Figura 38. Métricas de pérdida en entrenamiento

Como se puede observar en las Figura 37 y 38 las métricas de precisión y validación durante el entrenamiento son prometedoras debido a que en ambos gráficos las métricas tratan de ir simultáneamente en precisión y pérdida de tal forma que se garantiza que el modelo está aprendiendo y está generalizando. Una vez entrenado el modelo es convertido en un TensorFlow Lite para ser incorporado en una aplicación móvil.

### 3.3 Desarrollo de la aplicación móvil

Para el desarrollo de la aplicación móvil se propuso la metodología Mobile-D por lo tanto se debe seguir las 5 fases: exploración, iniciación, producción, estabilización y prueba de sistema para asegurar un desarrollo ágil y flexible entorno al desarrollo de aplicaciones móviles.

#### 3.3.1 Fase 1: Exploración

En esta fase se propone establecer un plan y establecer las características del proyecto. Esto se realiza en tres etapas: establecimiento de los interesados, definición del alcance y el establecimiento de proyectos.

*a. Establecimiento de los interesados (Stakeholders)*

Tabla 28. Establecimiento de los interesados

<b>Interesados</b>	<b>Descripción</b>
<b>Desarrollador</b>	Es un profesional que se dedica a la creación y programación de soluciones informáticas con la ayuda de varios lenguajes de programación y herramientas según el tipo de proyecto en el que este trabajando.
<b>Comprador Vendedor</b>	Persona que se dedica a la compra venta de vehículos usados, el cual puede ser un comprador o vendedor particular o una persona comerciante que hace una actividad económica.

*b. Definición de Alcance*

Las siguientes características definen el alcance del proyecto que se pretende cumplir.

- Las funcionalidades de la aplicación se establecen acorde a los requerimiento funcionales y no funcionales descritos en la sección 3.1.1 de este proyecto.
- La aplicación móvil funciona como un prototipo a implementarse en un entorno real.
- Para el correcto funcionamiento de la aplicación debe tener una conexión a internet para manejar peticiones HTTP por medio de una API.
- La aplicación móvil se ejecutará en dispositivos que ejecuten Android 7.0 (Nougat) o superior.

*c. Definición del proyecto*

Para esta etapa se utilizará las siguientes herramientas de desarrollo:

- **Android Studio**

Es un entorno de desarrollo integrado (IDE) creado por Google para el desarrollo de aplicaciones móviles, soporta lenguajes de programación como Java, Kotlin y Dark. Además, permite diseñar, codificar, depurar, probar y empaquetar aplicaciones móviles Android. Android Studio tiene una interfaz amigable y versátil.

- ***Spring Tool Suite4 (STS)***

Es un IDE basado en Eclipse y diseñado para trabajar con el Framework Spring para Java. El STS permite la integración de dependencias de Maven y Gradle.

- ***SQL Server***

Es un sistema de gestión de base de datos relacional creado por Microsoft, que permite el almacenamiento, recuperación y administración de datos.

- ***SQL Server Management Studio***

Herramienta grafica que administra las instancias del motor de base datos SQL Server.

- ***Kotlin***

Es un lenguaje de programación que ha tomado popularidad en el desarrollo de aplicación Android en los últimos años. Kotlin tiene una interoperabilidad con Java para hacer incorporaciones con librerías con java o viceversa. Además, integra la inferencia de tipos, Seguridad Nula y corrutinas para manejar tareas asincrónicas.

- ***Open-Source Computer Vision Library (Opencv)***

Es una biblioteca de código abierto especializado en visión por computadora y procesamiento de imágenes

### **3.3.2 Fase 2: Inicialización**

En esta fase se establece el diseño de la aplicación móvil.

#### ***a. Configuración del ambiente de desarrollo***

El ambiente de desarrollo se configuro con las herramientas necesarias para cumplir la propuesta de este proyecto de investigación.

- **Tipo de aplicación:** Aplicación nativa.
- **Desarrollo front-end:** Kotlin.
- **Procesamiento de imágenes:** Opencv

- **Desarrollo back-end:** Spring Boot y java.
  - **Sistema de base de datos:** SQL Server.
  - **Administrador de base de datos:** SQL Server Management Studio
- *Preparación del ambiente*

Se detallan las herramientas instaladas para el desarrollo de la aplicación móvil.

- Android Studio.
- Kotlin.
- Spring Tool Suite4.
- SQL Server.
- SQL Server Management Studio.
- Opencv

**b. Planificación de fases**

Tabla 29. Planificación de fases

Fase	Iteración	Descripción
Exploración	Iteración 0	<ul style="list-style-type: none"> <li>• Establecer interesados.</li> <li>• Definir alcances.</li> <li>• Definición del proyecto.</li> </ul>
Inicialización	Iteración 1	<ul style="list-style-type: none"> <li>• Configuración del ambiente de desarrollo.</li> </ul>
Producción	Iteración 2	<ul style="list-style-type: none"> <li>• Implementación de la pantalla principal.</li> <li>• Elaboración de las storycard.</li> <li>• Pruebas de funcionalidad.</li> </ul>
	Iteración 3	<ul style="list-style-type: none"> <li>• Implementación del módulo Inspección legal del vehículo.</li> <li>• Elaboración de las storycard.</li> <li>• Pruebas de funcionalidad.</li> </ul>
	Iteración 4	<ul style="list-style-type: none"> <li>• Implementación de la funcionalidad de guardar propietario.</li> <li>• Elaboración de las storycard.</li> <li>• Pruebas de funcionalidad.</li> </ul>
	Iteración 5	<ul style="list-style-type: none"> <li>• Implementación de la pantalla home.</li> <li>• Elaboración de las storycard.</li> <li>• Pruebas de funcionalidad.</li> </ul>
	Iteración 6	<ul style="list-style-type: none"> <li>• Implementación de la funcionalidad editar propietario.</li> <li>• Elaboración de las storycard.</li> <li>• Pruebas de funcionalidad.</li> </ul>
	Iteración 7	<ul style="list-style-type: none"> <li>• Implementación del módulo Inspección externa del vehículo.</li> <li>• Elaboración de las storycard.</li> <li>• Pruebas de funcionalidad.</li> </ul>

Fase	Iteración	Descripción
	Iteración 8	<ul style="list-style-type: none"> <li>• Implementación de la pantalla about.</li> <li>• Elaboración de las storycard.</li> <li>• Pruebas de funcionalidad.</li> </ul>
Estabilización	Iteración 10	<ul style="list-style-type: none"> <li>• Corrección de errores generales.</li> <li>• Ajustes al diseño.</li> </ul>
Pruebas del sistema	Iteración 11	<ul style="list-style-type: none"> <li>• Realización de pruebas de aceptación a la aplicación.</li> <li>• Pruebas de unidad a la aplicación.</li> </ul>

**c. Esquema de funcionamiento de la aplicación móvil.**

El esquema de funcionamiento de la aplicación móvil ha sido pensado para dos tipos de usuarios donde el usuario administrador es el Investigador que tiene acceso a la base de datos sin ningún tipo de interfaz gráfica dado que se trata de un prototipo, por otro lado, el usuario final proporcionara imágenes del vehículo para la extracción de la placa y análisis de estético del vehículo. Para consultar la información del vehículo se hace a través de una API con un Request Param pasado como argumento la placa por lo tanto se requiere una conexión a internet. En la Figura 39 se observa el proceso de funcional de la aplicación móvil.

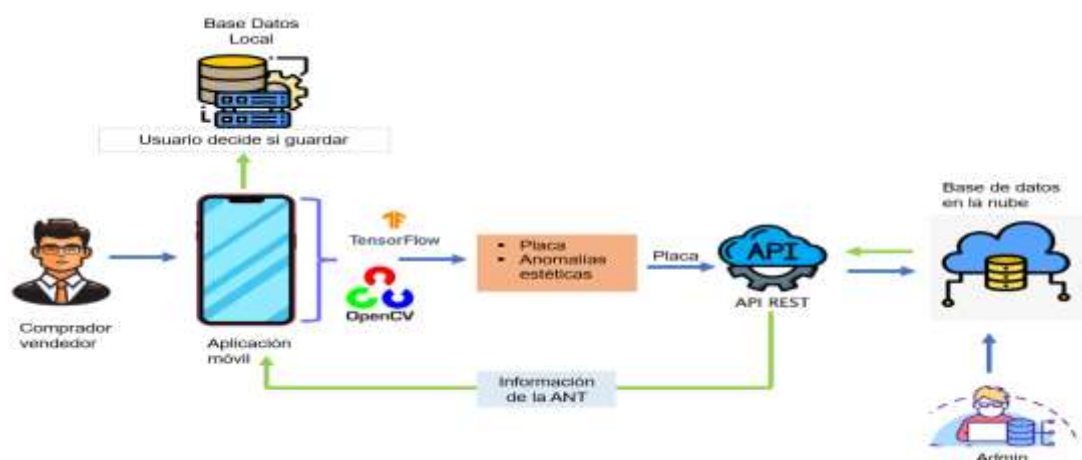


Figura 39. Esquema de funcionamiento de la aplicación

**d. Diseño de la base de datos**

Para el diseño de la base de datos se lo realizo con SQL Server como motor de base de datos y SQL Server Management Studio como herramienta administrativa grafica. El diagrama de base datos descrito en la Figura 41 es una simulación de la posible base datos que maneja la Agencia Nacional de Tránsito (ANT).

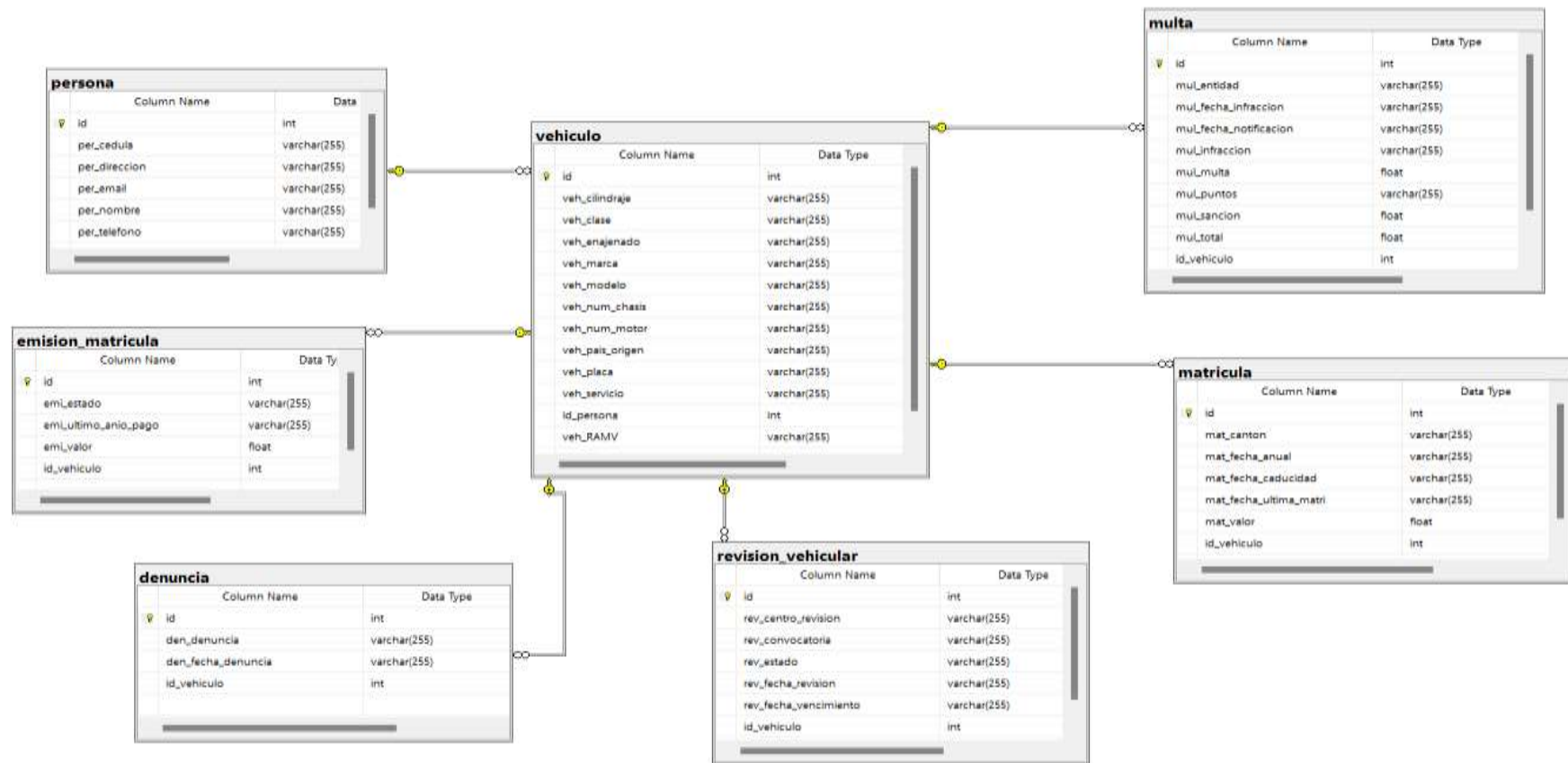


Figura 40. Base de Datos Relacional



e. *Esquema de navegabilidad*

En la Figura 41 se puede notar el flujo de navegación entre las distintas funcionalidades y posibles pantallas de aplicación.

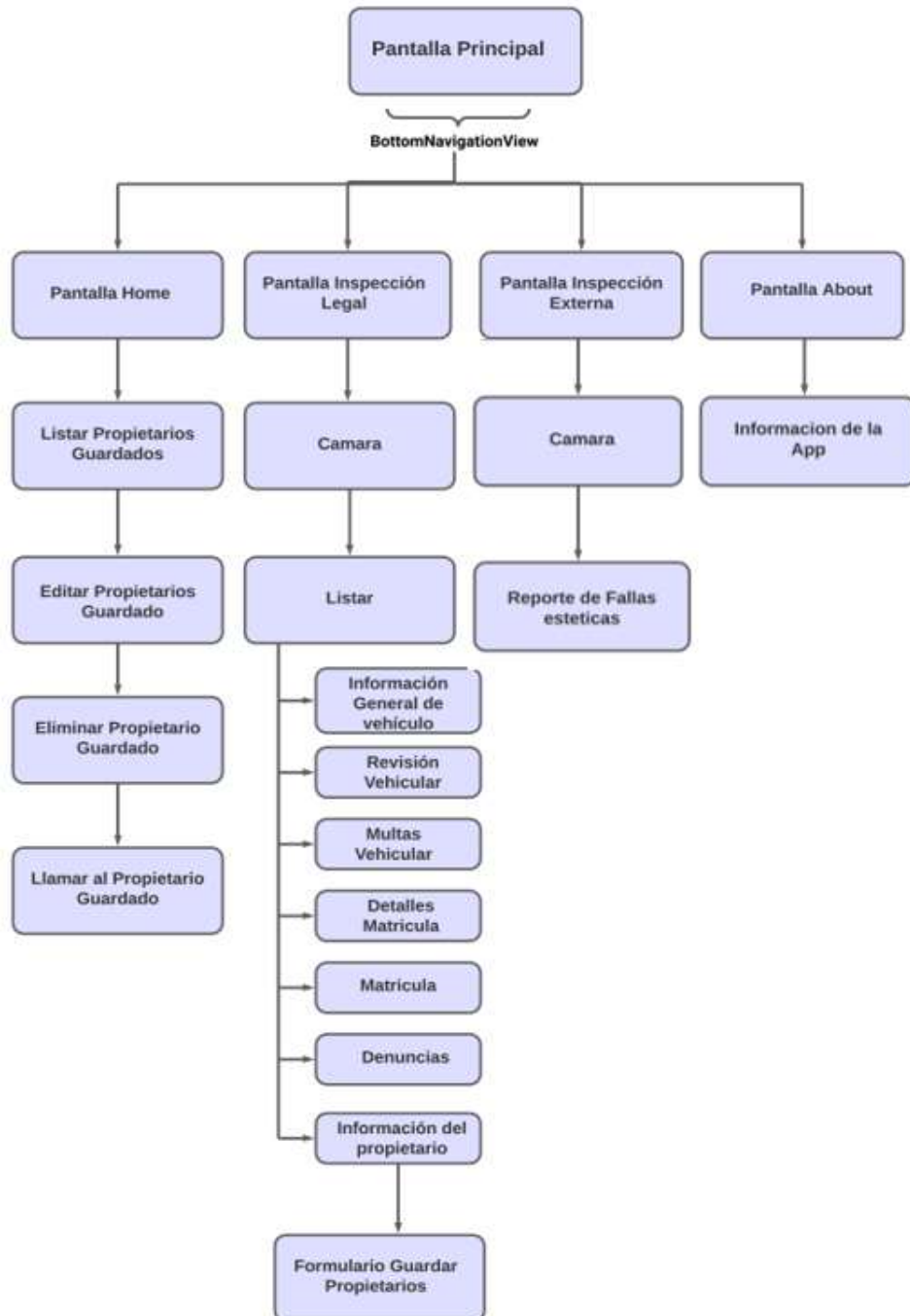


Figura 41. Esquema de navegabilidad

*f. Diagramas de casos de uso*

Los usuarios podrán realizar las siguientes acciones:

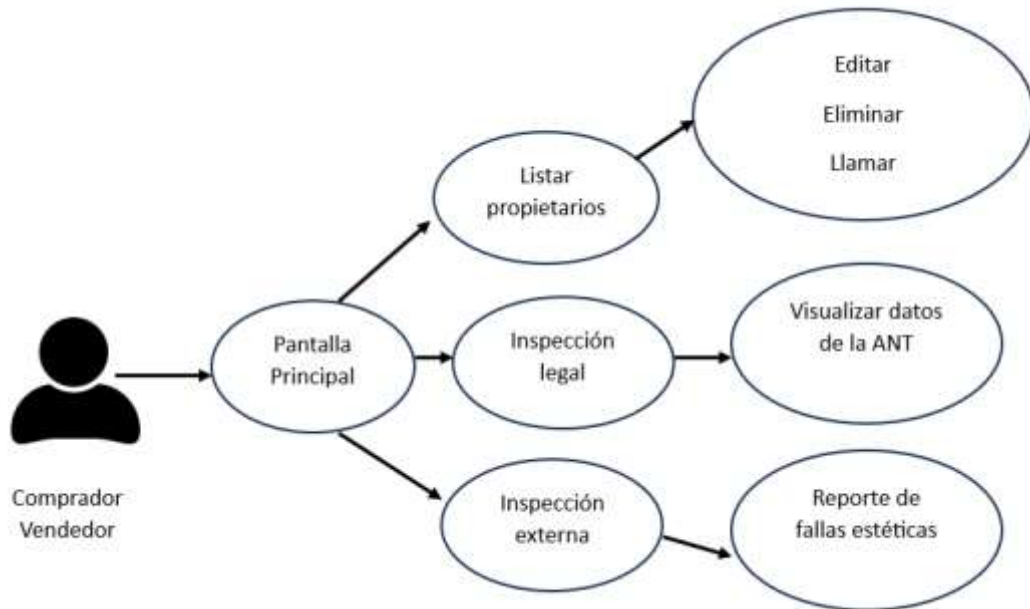


Figura 42. Diagrama de caso de uso

- ***Prototipo de la aplicación***

El diseño de una interfaz tiene varios factores significativos que impactan al usuario como por ejemplo un buen atractivo visual, ya que al contar con un diseño visualmente agradable puede influir en la percepción de la calidad de la aplicación y generar interés por parte de los usuarios finales por otro lado contar con una interfaz bien diseñada facilita la navegación haciendo que la aplicación sea fácil de usar. Para el diseño del prototipo de la aplicación se utiliza Mockitt [40] que es una herramienta en la web para diseñar, visualizar y comunicar ideas de diseño para realizar aplicaciones móviles.

- ***Pantalla Principal.*** Al iniciar la aplicación el usuario contara con una pantalla principal donde encontrara un menú en la parte inferior de la pantalla. Este menú contiene las 3 funciones principales: Listar propietarios almacenados, Inspección legal e Inspección externa de vehículo con se puede notar en la Figura 43.

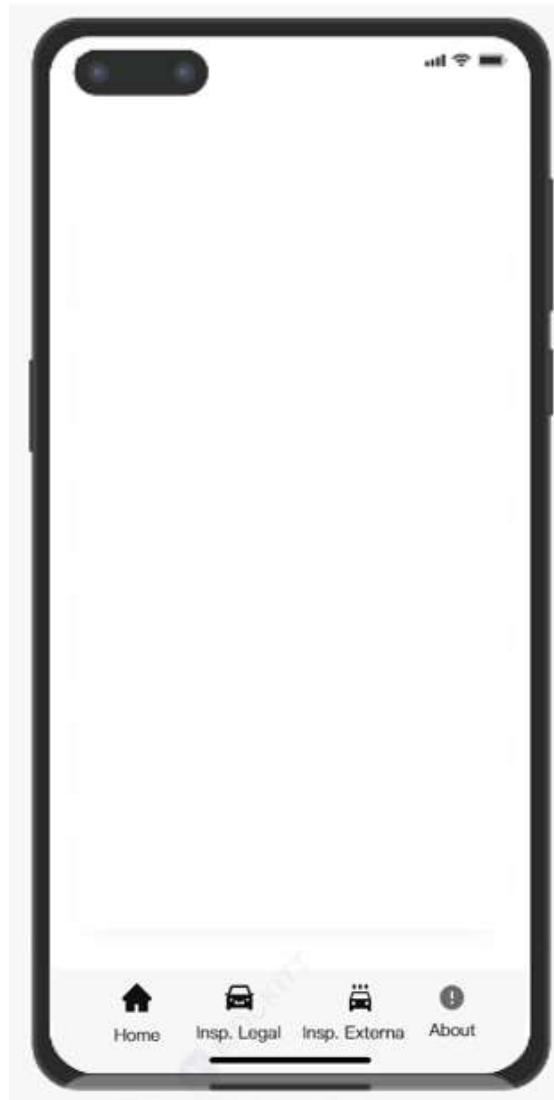


Figura 43. Pantalla principal

- **Home.** Esta pantalla será con la que inicie el usuario donde se lista todos propietarios de los vehículos que el usuario decida guardar. Cada ítem de la lista se podrá visualizar la marca y modelo del vehículo, con el nombre del propietario y un icono o imagen del vehículo además cuenta con un menú con 3 opciones editar, eliminar y llamar como se puede observar en la Figura 44.

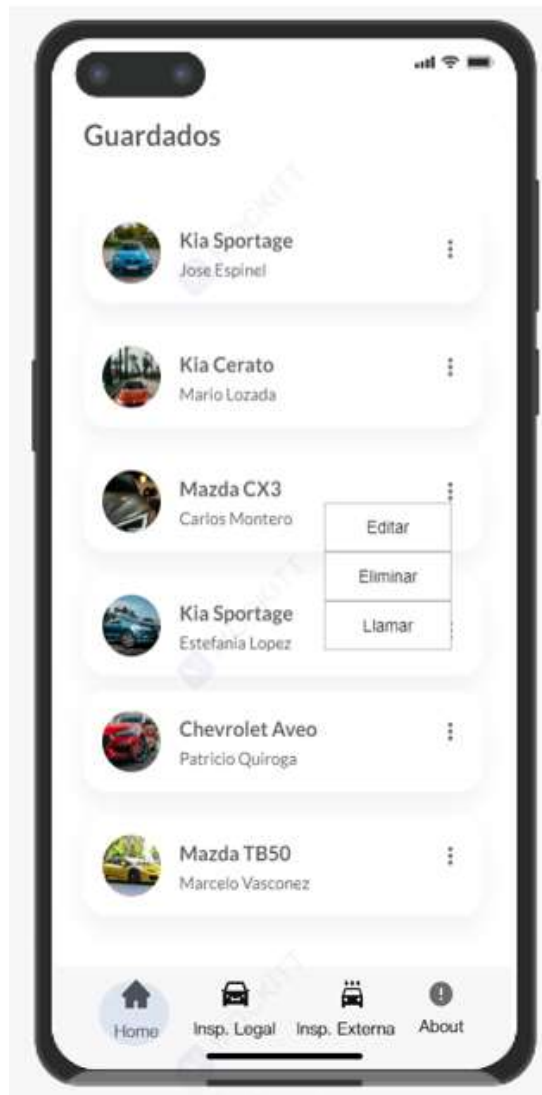


Figura 44. Pantalla home

- ***Insp. Legal.*** En la Figura 45 se puede observar la opción que permite acceder a la cámara del dispositivo para realizar la inspección legal del vehículo.



Figura 45. Inspección legal del vehículo

- **Detalles de la inspección legal.** Esta información se despliega una vez que se realiza la segmentación, detección, clasificación de caracteres y consulta en la base de datos de ANT. Esta despliega información como: datos de propietario, multas, revisiones, matricula, denuncias etc. Además, se dispone de un botón en la parte superior para guardar el propietario Ver Figura 46.

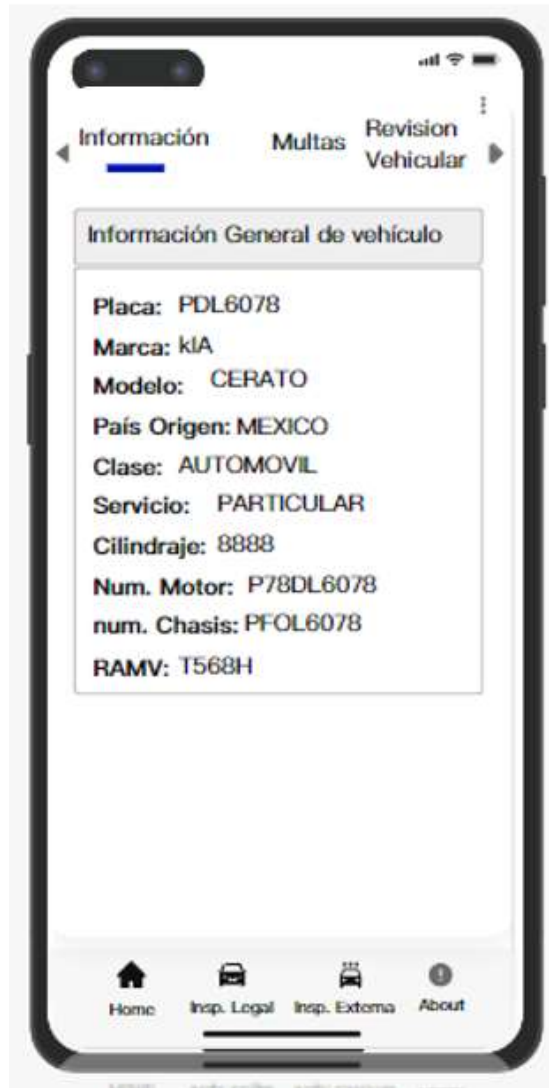


Figura 46. Información del vehículo

- **Diálogo guardar propietario.** Esta opción se muestra con los datos recuperados del propietario desde la base de datos para mostrar el dialogo y el usuario tiene la posibilidad de guardar el propietario como se puede ver en la Figura 47.



Figura 47. Diálogo guardar propietario

- ***Inspección Externa.*** Esta opción permite al usuario realizar una inspección estética al vehículo como se puede notar en la Figura 48.



Figura 48. Inspección externa



- **Información de la app.** Esta opción se muestra la información de aplicación como la versión y para qué sirve como se puede observar en la Figura 49.



Figura 49. Información de la app

### 3.3.3 Fase 3: Producción

En esta fase se desarrolla la aplicación móvil con sus funcionalidades acordadas en la fase 2.

### **a. Backend**

Para el desarrollo del backend se utilizó el framework Spring Boot conjuntamente con Java para desarrollar un API. Este desarrollo se lo realiza bajo las buenas prácticas de programación haciendo uso de la arquitectura Modelo Vista Controlador (MVC) y además se utilizó una arquitectura de capas para separar: persistencia, negocio y presentación. Spring Boot utiliza un archivo de configuración llamado `Application.properties` para establecer propiedades como el driver de conexión a la base de datos, usuario y contraseña de la base de datos, puerto y algunas funciones relacionadas con las sentencias SQL. Se establece el driver de conexión para SLQ Server, puerto y se le proporciona un usuario y contraseña de conexión. Ver Anexo N

- **Entidades.** La interacción con la base de datos en un proyecto realizado con Spring Boot puede ser flexible para este caso se trabaja con Java Persistence API (JPA) lo cual denota a las clases como entidades para mapear objetos relacionales de la base de datos.

Lista de entidades:

- Denuncias.java
  - Emisión Matricula.java
  - Multa.java
  - Matricula.java
  - Persona.java
  - RevisionVehicular.java
  - Vehiculo.java
- **Controladores**

Esta clase es la encargada de procesar las solicitudes HTTP de tipo GET.

Lista de controladores:

- VehiculoController.java

**b. Front-end**

- **Arquitectura de la aplicación móvil.** Para la arquitectura de la aplicación se aplica la arquitectura Model, View, ViewModel (MVVM) porque esta arquitectura facilita la gestión del estado y la lógica de presentación o UI sin mesclar con la lógica de la interfaz de usuario combinado esto con clean architecture para organizar el código de manera más eficiente que garantice la escalabilidad, mantenimiento y pruebas de la aplicación. En la Figura 50 se puede notar como la arquitectura organiza la estructura interna de la aplicación.

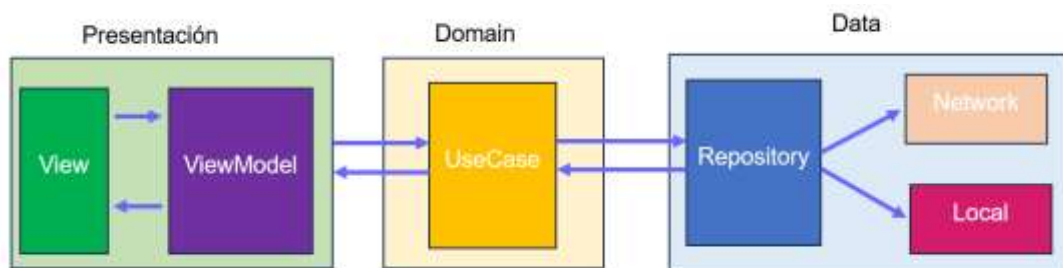


Figura 50. Arquitectura de la aplicación móvil con MVVM y clean Architecture

- **Dependencias.** El archivo build.gradle maneja las dependencias de un proyecto de Android Studio este archivo cuenta con las siguientes dependencias. Ver anexo O.
- **Integración de modelos Tflite.** La aplicación utiliza modelos de aprendizaje automático previamente entrenados descritos en la sección 3.2 de este trabajo de investigación. Se integra los modelos entrenados en el package llamado ml del proyecto de Android como se puede observar en la Figura 51.

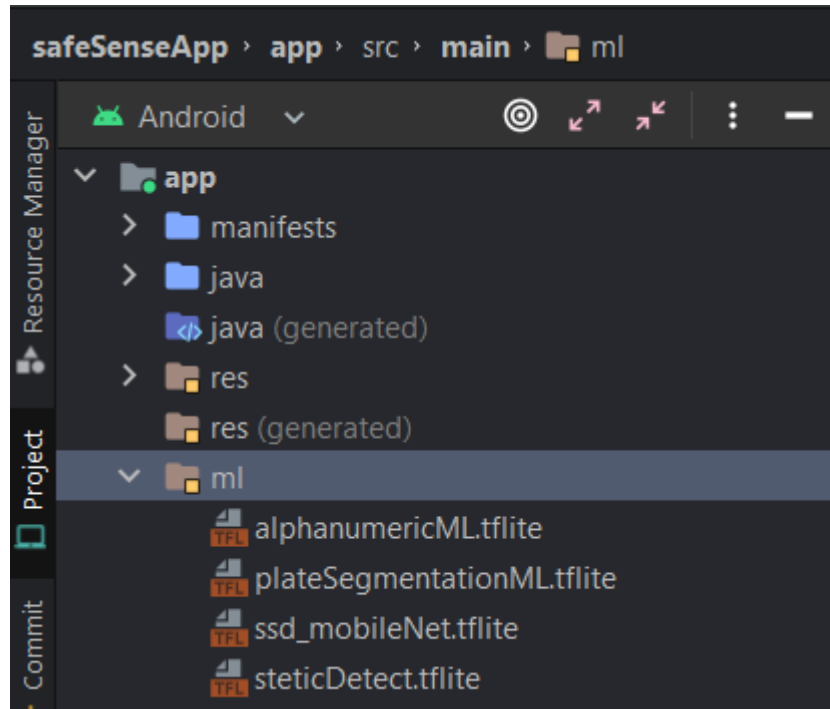


Figura 51. Integración de los modelos tflite

### c. *Storycards*

Las Storycards son útiles para el desarrollo de software, dado que es una manera de darle seguimientos a las funcionalidades o requerimientos de la aplicación. Dicho de otra forma, el objetivo de las Storycards se basa en: cumplir y satisfacer los requisitos de las historias de usuarios. En ellas se puede describir prioridades, tiempo de ejecución, describir dificultades, esfuerzo que implica cada actividad y añadir comentarios relevantes de las historias.

- *Iteración 2: Implementación de la pantalla principal*

### Storycard 1

Tabla 30. Storycard - Pantalla principal

N.º	Tipo	Dificultad	Esfuerzo	Prioridad
		Media	Media	Media
1	Nuevo			
<b>Descripción:</b> Pantalla principal				
Fecha		Acción	Comentario	
20/11/2023		Definido	Sin Comentario.	
22/11/2023		Implementado	Sin Comentario.	
23/11/2023		Ejecutado	Sin Comentario.	
24/11/2023		Verificado	Sin Comentario.	

La pantalla principal se estructura a través de Leguaje de Marcado Extensible (XML) que representa una interfaz de usuario. Ver anexo P.

- *Iteración 3: Implementación del módulo Inspección legal del vehículo*

### Storycard 2

Tabla 31. Storycard – Inspección legal del vehículo

N.º	Tipo	Dificultad	Esfuerzo	Prioridad
		Alta	Alta	Alta
2	Nuevo			
<b>Descripción:</b> Módulo Inspección legal del vehículo				
Fecha		Acción	Comentario	
20/11/2023		Definido	Sin Comentario.	
25/11/2023		Implementado	Sin Comentario.	
01/12/2023		Ejecutado	Sin Comentario.	
02/12/2023		Verificado	Sin Comentario.	

El método bindCameraUseCases que es parte de la clase InpsLegalCameraFragment establece las configuraciones necesarias para acceder la camera trasera del dispositivo y realiza el streaming. Una vez que empieza hacer el streaming las imágenes son pasadas como argumento al método detectPlaca. Ver Anexo Q.

El método detectPlaca recibe la imagen y la transforma en un bitmapBuffer y además obtiene la rotación de imagen para luego pasarlo al método detectPlaca de la clase detectPlateHelper que es un listener para los eventos de generados por la instancia de DetectPlacaHelper. Ver Anexo R.

Como se puede observar el código del Anexo S la clase DetectPlacaHelper a más de contener en método que es escucha en la clase InpsLegalCameraFragment se encarga de toda la lógica de cargar los modelos de Tflite, configura las entradas de los tensores de los modelos y el número de hilos que ocupara en segundo plano. Esta clase trabaja en paralelo al hilo principal de la aplicación para no bloquear el UI del usuario.

Retomando el método detectPlaca de la clase DetectPlacaHelper recibe 2 parámetros el bitmap que representa la imagen y la rotación de la imagen, se realiza una copia del bitmap para un post procesamiento y la imagen original conjuntamente con la rotación ingresa en el método processInputImage que realiza operaciones como rotar la imagen en una orientación correcta, realiza una normalización de pixeles retornando un TensorImagen que será la entrada para el primer modelo Tflite. Ver Anexo T.

Una vez obtenido un TensorImagen acorde a las especificaciones del modelo de segmentación para localizar la placa del vehículo es ingresado en un TensorBuffer para la entrada del modelo de segmentación, el modelo de segmentación retorna un TensorBuffer que representa la máscara de segmentación de la localización de la placa predicha. La máscara representa las probabilidades de cada pixel si pertenecen al área de interés es decir la placa, este TensorBuffer hay que convertir una imagen binaria donde la región de interés será de color blanco y todo lo demás será negro. Ver Anexo U.

En el código del Anexo T la imagen binaria resultante es pasada como argumento al método processOpencv, este método realiza operaciones de como: transformaciones, resize, findContours y recorte de la región de interés es decir recorta la placa.

La imagen resultante de este posprocesamiento es ahora la entrada para el modelo de detección de caracteres, esta imagen es ingresa en el método detectObjects que hace un preprocesamiento previo a ingresar en el modelo de detección. Se realiza un

preproceso de recorte acorde a la entrada del modelo y se hace una normalización de pixeles para ingresar al modelo. Ver Anexo V.

El resultado que retorna el modelo de detección son coordenadas de la ubicación de los caracteres y el score, el objetivo es tomar esos resultados y realizar el recorte de cada carácter en base una imagen que contiene la placa. Ver Anexo W.

Como se puede notar en el código del Anexo X el resultado es una lista con los caracteres recortado para ser ingresado en el modelo de clasificación de caracteres, el método `cnnClassify` recibe la lista y realiza las predicciones para determinar que carácter alfanumérico se trata.

El método `cnnClassify` retorna la placa en forma de string al hilo principal de la aplicación en la clase `InpsLegalCameraFragment`. Ver Anexo Y.

En el código del Anexo Z el string resultante se maneja con un `ViewModel` para ejecutar la función que consulta a través de la placa toda la información del vehículo, el resultado de la consulta es cargado en un `MutableLiveData` que está siendo observado desde la vista del usuario para indicar si hubo respuesta para alterar la UI del usuario.

- ***Iteración 4: Implementación de la función guardar propietario***

### **Storycard 3**

Tabla 32. Storycard – Guardar propietario

<b>N.º</b>	<b>Tipo</b>	<b>Dificultad</b>	<b>Esfuerzo</b>	<b>Prioridad</b>
		Alta	Alta	Alta
3	Nuevo			
<b>Descripción:</b> Guardar propietario				
<b>Fecha</b>	<b>Acción</b>	<b>Comentario</b>		
20/11/2023	Definido	Sin Comentario.		
03/12/2023	Implementado	Sin Comentario.		
06/12/2023	Ejecutado	Sin Comentario.		
08/12/2023	Verificado	Sin Comentario.		

El método showBottomDialog ejecuta la lógica de consulta en la API la información del propietario y lanza el formulario con los datos obtenidos. Ver Anexo 1A.

Para guardar el propietario llama al ViewModel del propietario y ejecuta la función de inserta propietario en la base de datos local, pero previo al guardado se realiza la respectiva validación del formulario Ver Anexo 2B.

- **Iteración 5: Implementación pantalla home**

#### Storycard 4

Tabla 33. Storycard – Pantalla home

N.º	Tipo	Dificultad	Esfuerzo	Prioridad
		Alta	Alta	Alta
4	Nuevo			
<b>Descripción:</b> Pantalla home				
Fecha		Acción	Comentario	
20/11/2023		Definido	Sin Comentario.	
09/12/2023		Implementado	Sin Comentario.	
12/12/2023		Ejecutado	Sin Comentario.	
13/12/2023		Verificado	Sin Comentario.	

Para implementar la pantalla home es necesario recuperar la información guardada en la base de datos local. El método onCreateView del fragment home llama al método getAllPropietarios encargado de extraer lo propietarios desde la base de datos local como se puede apreciar en el código del Anexo 3C.

- **Iteración 6: Implementación funcionalidad editar propietario**

#### Storycard 5

Tabla 34. Storycard – Editar Propietario

N.º	Tipo	Dificultad	Esfuerzo	Prioridad
		Alta	Alta	Alta
5	Nuevo			
<b>Descripción:</b> Editar Propietario				
Fecha		Acción	Comentario	
20/11/2023		Definido	Sin Comentario.	
14/12/2023		Implementado	Sin Comentario.	
20/12/2023		Ejecutado	Sin Comentario.	
21/12/2023		Verificado	Sin Comentario.	



En método showBottomDialogEdit muestra el dialogo y extrae la información obtenida a través del observable cuando el usuario presionó el botón editar como se puede notar en el Anexo 4D.

- **Iteración 7: Implementación módulo de inspección externa de vehículo**

### Storycard 6

Tabla 35. Storycard – Inspección externa del vehículo

N.º	Tipo	Dificultad	Esfuerzo	Prioridad
		Alta	Alta	Alta
6	Nuevo			
<b>Descripción:</b> Modulo inspección externa del vehículo				
Fecha		Acción	Comentario	
20/11/2023		Definido	Sin Comentario.	
22/12/2023		Implementado	Sin Comentario.	
27/12/2023		Ejecutado	Sin Comentario.	
28/12/2023		Verificado	Sin Comentario.	

El método detectStetic de la clase InspExternalCameraFragment configura la cámara e inicializa el streaming de la cámara, las imágenes son pasadas como parámetro al método detectStetic para su posterior análisis en un hilo distinto al principal. Ver Anexo 5E.

La clase DetectSteticHelper inicializa el modelo de aprendizaje automático para detectar anomalías estéticas del vehículo y además establece los parámetros de entrada del tensor. Ver Anexo 6F.

Como se puede notar en el código del Anexo 7G realiza la inferencia donde se establece los TensorBuffer de entrada y salida. Antes de realizar la inferencia se debe normalizar los pixeles. El resultado es una probabilidad dado por dos etiquetas y es retornado al hilo principal de la aplicación.

- **Iteración 8: Implementación pantalla about**

### Storycard 7

Tabla 36. Storycard – Pantalla about

N.º	Tipo	Dificultad	Esfuerzo	Prioridad
		Media	Media	Media
7	Nuevo			
<b>Descripción:</b> Pantalla about				
	<b>Fecha</b>	<b>Acción</b>	<b>Comentario</b>	
	20/11/2023	Definido	Sin Comentario.	
	29/12/2023	Implementado	Sin Comentario.	
	30/12/2023	Ejecutado	Sin Comentario.	
	30/12/2023	Verificado	Sin Comentario.	

Al tratarse de un diseño simple se lo estructuro en XML. Ver Anexo 8H.

#### 3.3.4 Fase 4: Estabilización

Esta fase esta dedica a corregir error, ajustes en el UI de la aplicación y garantizar que la aplicación esté libre de errores.

Se realizo los siguientes ajustes:

- Implementación de progress bar para dar retroalimentación al usuario.
- Ajustes de tamaño de letra.
- Ajustes de padding en la cardView información general del vehículo.
- Se implemento cuadros de dialogo en las acciones.

#### 3.3.5 Fase 5: Pruebas

Esta fase tiene como objetivo comprobar el funcionamiento de la aplicación, donde se constata el comportamiento esperado de la app, de ser el caso esta fase ayuda a identificar y corregir errores y mejorar su rendimiento.

a. *Pruebas de funcionalidad*

- **Pantalla home.** Esta pantalla se muestra al iniciar la aplicación donde se puede visualizar los propietarios almacenados localmente. Además, se muestra un menú con 3 opciones: eliminar, editar y llamar como se detalla en la Figura 52.

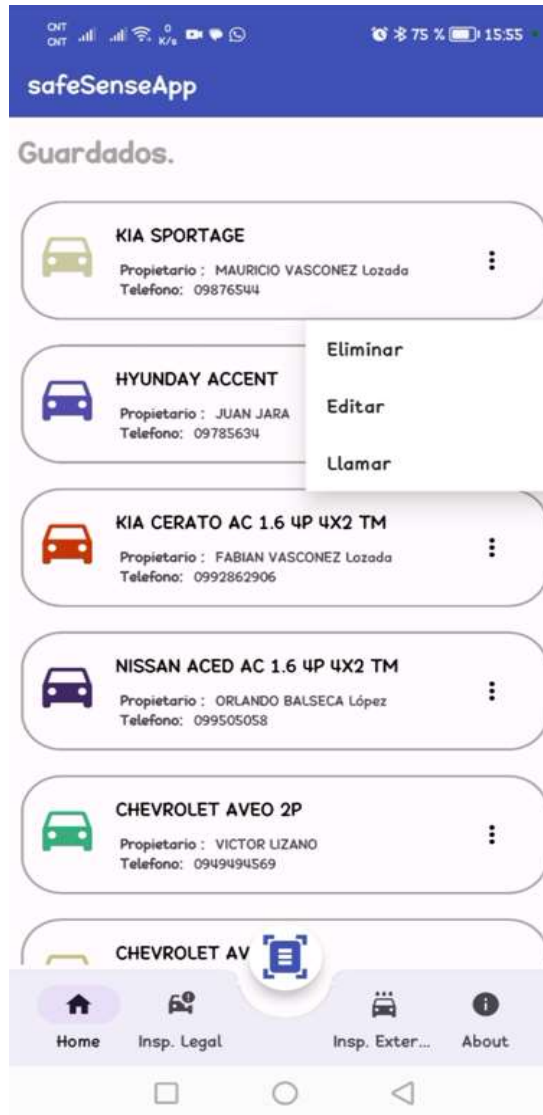


Figura 52. Pantalla home

- **Inspección Legal.** Como se puede observar en la Figura 53 esta funcionalidad permite al usuario abrir la cámara para realizar la segmentación, detección y clasificación de la placa del vehículo.



Figura 53. Inspección legal

- **Despliegue de toda la información del vehículo.** Una vez que los modelos de aprendizaje automático trabajan conjuntamente es realizada la petición a la API esa información es renderizada para mostrar al usuario la información general vehículo, revisión vehicular, multas, detalles de la matricula, matricula, denuncias e información del propietario del vehículo. Ver Figura 54.



Figura 54. Información del vehículo

- **Guardar propietario.** Esta opción se despliega cuando el usuario pulsa en el menú superior para guardar toda la información recuperada en la base datos local. El Dialogo muestra los datos del propietario en un formulario que este sujeto a validaciones de los campos. Ver Figura 55.

The screenshot shows the safeSenseApp interface. At the top, there's a navigation bar with 'Información', 'Revisión Vehicular', 'Multas', and 'Detalles Matrícula'. Below this is a section titled 'Información General del Vehículo' with the following details: Placa: PDL6078, Marca: KIA, Modelo: CERATO AC 1.6 4P 4X2 TM, País Origen: MEXICO, Clase: AUTOMOVIL, and Servicio: PARTICULAR. A modal dialog box titled 'Datos del Propietario' is overlaid on the screen. It contains four input fields: a name field with 'FABIAN VASCOHEZ', a location field with 'HUACHI', a phone field with '0992862903', and an email field with 'USER123@GMAIL.COM'. At the bottom of the dialog are two buttons: a red 'Cancelar' button and a blue 'Guardar' button.

Figura 55. Guardar Propietario

- **Editar propietario.** Como se puede notar en la Figura 56 esta opción le permite al usuario editar los datos del propietario, los campos del formulario también están sujetos a validaciones para evitar cualquier dato sucio.



Figura 56. Actualizar propietario

- **Inspección externa del vehículo.** Esta opción se puede observar en la Figura 57 que le permite acceder a la cámara y realizar una inspección estética del vehículo.



Figura 57. Inspección estética del vehículo



- **Pantalla About.** Esta pantalla le permite al usuario visualizar información acerca de la aplicación como se puede ver en la Figura 58.



Figura 58. Pantalla About

### 3.4 Muestra para aplicar en el testeo

Para el cálculo del tamaño de la muestra representativa para aplicar en el testeo de la aplicación se utilizará un nivel de confianza deseado del 80% y un margen de error de 11%, porque los criterios de las personas involucradas en el comercio de vehículos usados son variados en torno a los aspectos legales y estéticos y no es posible cubrir todas las necesidades, entonces se aplica la siguiente formula:

$$\text{Tamaño de la muestra (N)} = \frac{z^2 * p * (1 - p)}{e^2}$$

Donde:

N = Tamaño de la muestra.

NC = Nivel de confianza deseado 80% equivalente a 1.28.

Z = Puntuación de valor Z según NC.

P = Probabilidad de éxito.

E = Margen de error 11%.

$$n = \frac{1.28^2 * 0.5 * 0.5}{0.11^2} \quad n = 34$$

De la población infinita de las personas que acuden a las ferias de vehículos con un nivel de confianza de 80% y un margen de error 11% la formula estadística arrojo un tamaño de muestra de 34 personas para realizar el testeo de la aplicación móvil.

Se realiza un muestreo aleatorio simple donde solo serán personas comerciantes de vehículos usados dado que estas personas son más ocurrentes en la feria de vehículos.

#### 3.4.1 Recolección de la información

Para evaluar el proceso de validación de la aplicación desarrolla se lanza un cuestionario para validar su efectividad y utilidad para los usuarios. A través de este proceso, se busca confirmar que la aplicación móvil cumple con requisitos

especificados y, al mismo tiempo, proporciona una experiencia satisfactoria y sin inconvenientes para los usuarios.

*a. Cuestionario de test para aplicación móvil*

Tabla 37. Escala Likert para testear la aplicación

1	Muy en desacuerdo
2	En desacuerdo
3	Neutral
4	De acuerdo
5	Muy de acuerdo

¿La navegación dentro de la aplicación es clara y sencilla?

1	<input type="radio"/>
2	<input type="radio"/>
3	<input type="radio"/>
4	<input type="radio"/>
5	<input type="radio"/>

¿La función reporte legal del vehículo es útil?

1	<input type="radio"/>
2	<input type="radio"/>
3	<input type="radio"/>
4	<input type="radio"/>
5	<input type="radio"/>

¿La función inspección estética del vehículo es útil?

1	<input type="radio"/>
2	<input type="radio"/>
3	<input type="radio"/>
4	<input type="radio"/>
5	<input type="radio"/>

¿Me gusta la disposición de los elementos en la interfaz?

1	<input type="radio"/>
2	<input type="radio"/>
3	<input type="radio"/>
4	<input type="radio"/>
5	<input type="radio"/>

¿El rendimiento general de la aplicación es satisfactorio?

1	<input type="radio"/>
2	<input type="radio"/>
3	<input type="radio"/>
4	<input type="radio"/>
5	<input type="radio"/>

**b. Validación del instrumento**

Para evaluar la consistencia interna o fiabilidad del cuestionario se aplica la herramienta denominada alfa de Cronbach, esto determinara si las preguntas son coherentes y miden de manera confiable el concepto u objeto de estudio.

Para el cálculo del alfa de Cronbach se aplicó la siguiente fórmula:

$$\alpha = \left( \frac{k}{k-1} \right) \left( 1 - \frac{\sum v_i}{vt} \right)$$

Donde:

$\alpha$  = fiabilidad del cuestionario.

$k$  = Número de ítems.

$\sum v_i$  = Varianza de cada ítem

$vt$  = Varianza total.

Los cálculos de la fórmula del alfa de Cronbach fueron realizados en Excel a partir de la creación de una tabla, adjuntando las respuestas recolectadas de 7 encuestas realizadas aleatoriamente a comerciantes de vehículos usados en la explanada de Shuyurco, feria de vehículos usados de la ciudad de Ambato. Ver Anexo 9I.

$$\alpha = \left( \frac{7}{7-1} \right) \left( 1 - \frac{2,041}{5,714} \right) = 0,80$$

Como resultado el alfa de Cronbach da un valor de 0,80, se interpreta según la Tabla 4 donde se determina que el cuestionario es de excelente confiabilidad. Una vez evaluado la consistencia interna del cuestionario se procede a aplicar la encuesta a 34 personas según el tamaño de la muestra para testeo establecido anteriormente.

*c. Resultados de la encuesta*

La encuesta fue realizada presencialmente a los comerciantes y compradores de vehículos usados de la explanada de Shuyurco mediante la herramienta de Google Forms donde se alojó el cuestionario.

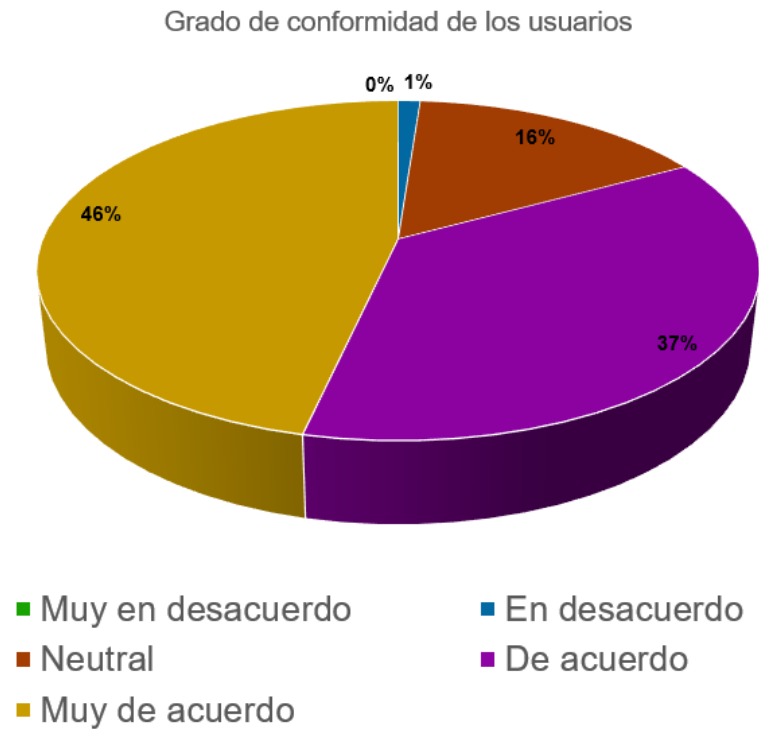


Figura 59. Grado de satisfacción de los usuarios

En la Figura 59 se puede observar el grado de conformidad de manera general de los usuarios al usar la aplicación móvil donde cerca de la mitad de los usuarios está muy de acuerdo o conforme con la funcionalidad que brinda la aplicación móvil. Las preguntas se basan en temas de la usabilidad y funcionalidad que se estableció mediante la encuesta descrita en la sección 2.2.3.

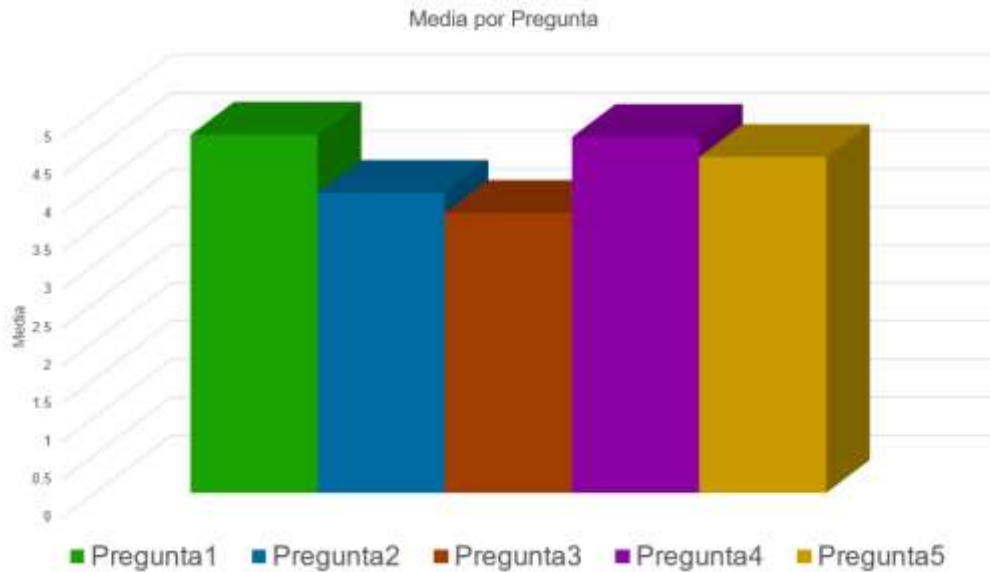


Figura 60. Media por pregunta

- La pregunta 1 es recibida con un grado de aceptación alta por parte de los usuarios dado que la interfaz es clara y sencilla. Esto es el resultado de realizar un estudio previo a la disposición de los elementos mediante un prototipo o mockup lo cual ayuda identificar y corregir problemas de diseño desde una etapa temprana.
- La pregunta 2 hace referencia a la funcionalidad de inspección legal del vehículo lo cual es recibida de buena manera por parte de los comerciantes de vehículos. Por lo tanto, los comerciantes ven útil esta función dado que contar con una aplicación móvil que centralice la información legal del vehículo en una sola app facilita y agiliza el proceso de verificación de datos. Esta función evita posibles errores de tipeado en las diferentes páginas web gubernamentales y evita contra tiempos.
- La inspección estética del vehículo que se planteó en la pregunta 3 es recibida de manera neutra por parte de los individuos. Esta función realiza una inspección estética en busca de posibles factores que revelan si el vehículo estas buenas condiciones con respecto a la carrocería lo cual, es útil en personas que no tiene mucha experiencia en el campo de la carrocería.

- En cuanto a la pregunta 4 y 5 se enfoca en la disposición de elementos en la interfaz y rendimiento lo cual es muy bien recibido por los comerciantes puesto que el rendimiento de la aplicación es satisfactorio gracias al estudio previo de la disposición de los elementos mediante un mockup.

## CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

### 4.1 Conclusiones

- Durante el proceso de recolección de la información las personas que intervienen en la feria de vehículos usados se pudo observar que la mayoría de personas ven rentable esta actividad siempre y cuando se aborde aspectos relacionados con el estado legal, estético y mecánico del automotor para evitar posibles inversiones que no generen retorno de capital económico.
- TensorFlow Lite resulto ser una biblioteca apropiada para el desarrollo y despliegue eficiente de modelos de IA en dispositivos con recursos computacionales limitados, gracias a su enfoque en la optimización y eficiencia sin afectar desmesuradamente la precisión del modelo.
- En virtud de lo estudiado, la combinación de los modelos de aprendizaje automático fundamentado en las arquitecturas: ResUNet, SDD Mobile V2 y una Red Neuronal Convolutiva tradicional resultan ser eficaces en este tipo de tareas ya que identificar patrones jerárquicos que van desde la obtención de características simples hasta obtener objetos más complejos para segmentar, detectar y clasificar la placa vehicular.
- Para la inspección estético del vehículo se desarrolló un modelo de aprendizaje automático, para lo cual se implementó una Red Neuronal Convolutiva y resulto ser eficiente en términos de detectar patrones superficiales, pero tiene sus limitaciones en cuanto a detectar distorsiones de color debidas a factores como la iluminación y la posición de cámara pueden representar ciertas limitaciones dado que la CNN es sensible a las variaciones en el color, y cambios en la iluminación pueden afectar significativamente la apariencia del vehículo haciendo que la CNN tenga dificultades para entrenar.



- Concluyendo, los aportes hechos por este estudio revelan resultados prometedores en cuanto a la integración de los modelos entrenados tflite en dispositivos móviles mediante el lenguaje de programación Kotlin resultan ser viables en el ámbito de reconocimiento de placas vehiculares.

## **4.2 Recomendaciones**

- Buscar espacios de retroalimentación con los comerciantes para promocionar el uso de la tecnología en el comercio de vehículos usados como estrategia clave para optimizar los procesos de adquisición, reducir el tiempo en tareas manuales y además ofrecer una experiencia de compra más transparente y atractiva.
- Se recomienda realizar pruebas exhaustivas previo a un despliegue en una aplicación móvil puesto que los modelos entrenados y convertidos en modelo Tflite pueden sufrir algún cambio, se debe asegurar que el modelo no haya introduciendo errores significativos al momento de la conversión y que el modelo siga siendo preciso.
- Se sugiere para trabajos futuros incrementar el dataset de segmentación obtenido en este trabajo de investigación para mejorar la precisión en el modelo de segmentación ResUNet puesto que el modelo ResUNet es la partida para detectar la placa y consigo los demás modelos de inteligencia artificial depende de dicho modelo de segmentación para completar la tarea.
- Se aconseja investigar técnicas más avanzadas de preprocesamiento y extracción de características en imágenes de vehículos usados con el propósito de mejorar la detección de anomalías más imperceptibles para consigo obtener mejores resultados en la Red Neuronal convolucional.
- Concluyendo, los aportes hechos por este estudio revelan resultados prometedores en cuanto a la integración de los modelos entrenados tflite en dispositivos móviles mediante el lenguaje de programación Kotlin resultan ser viables en el ámbito de reconocimiento de placas vehiculares.

## REFERENCIAS BIBLIOGRÁFICAS

- [1] Cujilema Paola Adriana, “República del Ecuador Universidad Tecnológica Empresarial de Guayaquil Trabajo de Titulación Para la Obtención del Título de: Licenciada en Comercio Exterior Mención Negocios Internacionales Tema: Importaciones Ecuatorianas del Sector Automotriz 2017-2021 y el Impacto de la Covid-19. Autora: Paola Adriana Cujilema Cujilema,” 2022.
- [2] M. Rivera, C. Renee, L. De, P. Albino, and W. Manuel, “FACULTAD DE INGENIERÍA Y ARQUITECTURA Sistema Web para el Proceso de Peritajes de Identificación ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS,” 2021.
- [3] R. Guamán *et al.*, “THE IMPOSSIBILITY OF RECONCILING THE CRIME OF SCAM IN ECUADOR, ANALYSIS OF ITS IMPROPER AND UNMOTIVATED PROHIBITION,” 2022. [Online]. Available: <http://doi.revistamqr.com/https://orcid.org/0000-0002-8695-5005http://mqrinvestigar.com/>
- [4] “Automóviles nuevos incrementan en Ambato – AEADE.” Accessed: May 09, 2023. [Online]. Available: <https://www.aeade.net/automoviles-nuevos-incrementan-en-ambato/>
- [5] J. Segundo *et al.*, “PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ ESCUELA DE POSGRADO,” 2018.
- [6] Gazi Üniversitesi, Aksaray Üniversitesi, University of Buner, P. International Islamic University (Islāmābād, Institute of Electrical and Electronics Engineers. Turkey Section, and Institute of Electrical and Electronics Engineers, *2nd International Conference on Electrical, Communication, and Computer Engineering (ICECCE 2020) : 12th-13th June 2020, Istanbul, Turkey*. 2020.
- [7] A. Benhamida, A. R. Varkonyi-Koczy, and M. Kozlowszky, “Traffic Signs Recognition in a mobile-based application using TensorFlow and Transfer Learning technics,” in *SOSE 2020 - IEEE 15th International Conference of System of Systems Engineering, Proceedings*, Institute of Electrical and Electronics Engineers Inc., Jun. 2020, pp. 537–541. doi: 10.1109/SoSE50414.2020.9130519.
- [8] E. Dorter and T. Bolu, “Classification of Turkish Cuisine With Deep Learning on Mobile Platform; Classification of Turkish Cuisine With Deep Learning on Mobile Platform,” 2019. [Online]. Available: [www.onlinedoctranslator.com](http://www.onlinedoctranslator.com)
- [9] C. M. Sruthy, S. Kunjumon, and R. Nandakumar, “Car Damage Identification and Categorization Using Various Transfer Learning Models,” in *Proceedings of the 5th International Conference on Trends in Electronics and Informatics*,

- ICOEI 2021*, Institute of Electrical and Electronics Engineers Inc., Jun. 2021, pp. 1097–1101. doi: 10.1109/ICOEI51242.2021.9452846.
- [10] D. Widjojo, E. Setyati, and Y. Kristian, “Integrated Deep Learning System for Car Damage Detection and Classification Using Deep Transfer Learning,” in *Proceeding - IEEE 8th Information Technology International Seminar, ITIS 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 21–26. doi: 10.1109/ITIS57155.2022.10010292.
- [11] IEEE Signal Processing Society and Institute of Electrical and Electronics Engineers, *Proceedings of AVSS 2018 : 2018 IEEE International Conference on Advanced Video and Signal-based Surveillance : Auckland, New Zealand, 27-30 November 2018*.
- [12] C. M. Ledesma, “SISTEMA AUTOMÁTICO DE RECONOCIMIENTO DE MATRÍCULAS DE COCHES,” 2021.
- [13] I. C. Draa, S. Niar, F. Bouquillon, and E. G. Le Strugeon, “Machine learning for improving mobile user satisfaction,” in *Proceedings of the ACM Symposium on Applied Computing*, Association for Computing Machinery, 2019, pp. 1200–1207. doi: 10.1145/3297280.3297398.
- [14] C. Wan, S. Liu, H. Hoffmann, M. Maire, and S. Lu, “Are Machine Learning Cloud APIs Used Correctly?,” *Institute of Electrical and Electronics Engineers (IEEE)*, May 2021, pp. 125–137. doi: 10.1109/icse43902.2021.00024.
- [15] C. Tu and S. Du, “A hierarchical RCNN for vehicle and vehicle license plate detection and recognition,” *International Journal of Electrical and Computer Engineering*, vol. 12, no. 1, pp. 731–737, Feb. 2022, doi: 10.11591/ijece.v12i1.pp731-737.
- [16] Torres Carrera Milton Daniel, “UNIVERSIDAD POLITÉCNICA SALESIANA SEDE QUITO,” 2020.
- [17] Arias Enrique, “Matemáticas aplicadas - Qué es, definición y concepto | 2023 | Economipedia.” Accessed: May 16, 2023. [Online]. Available: <https://economipedia.com/definiciones/matematicas-aplicadas.html>
- [18] Universidad de Ingeniería y Tecnología, “¿Qué es Ciencia de la Computación? | Carreras | UTEC.” Accessed: May 16, 2023. [Online]. Available: <https://utec.edu.pe/carreras/ciencia-de-la-computacion/que-es>
- [19] Y. Ocaña-Fernández, L. A. Valenzuela-Fernández, and L. L. Garro-Aburto, “Inteligencia artificial y sus implicaciones en la educación superior,” *Propósitos y Representaciones*, vol. 7, no. 2, Jan. 2019, doi: 10.20511/pyr2019.v7n2.274.
- [20] Pedrero Victor, Grandon Reynaldos, Achurra Joaquin, and Pinto Erick, “Generalidades del Machine Learning y su aplicación en la gestión sanitaria en Servicios de Urgencia,” 2021.

- [21] W. Ricardo Erazo Samboni and U. Antonio Nariño, “Desarrollo de aplicación Android móvil utilizando Visión Artificial y Deep Learning para la identificación de Aguacates Hass con la plaga Monalonia, en la finca ‘Las Palmas’, ubicada en el municipio de San Agustín, Huila,” 2022.
- [22] “A Neural Network Playground.” Accessed: Jul. 18, 2023. [Online]. Available: <https://playground.tensorflow.org/#activation=tanh&batchSize=10&dataset=circle&regDataset=reg-plane&learningRate=0.03&regularizationRate=0&noise=0&networkShape=4,2&seed=0.83838&showTestData=false&discretize=false&percTrainData=50&x=true&y=true&xTimesY=false&xSquared=false&ySquared=false&cosX=false&sinX=false&cosY=false&sinY=false&collectStats=false&problem=classification&initZero=false&hideText=false>
- [23] D. LA Información and I. Wellington Maliza Cruz, “EXAMEN COMPLEXIVO DE GRADO O DE FIN DE CARRERA PRUEBA PRÁCTICA PREVIO A LA OBTENCION DEL TITULO DE INGENIERA EN SISTEMAS EGRESADA: VALERO AVILES MILDRE VALERIA TUTOR: AÑO 2022 CONTENIDO.”
- [24] Bautista Salaza Paola Estefanía and Aldás Flores Clay Fernando, “1968ti,” 2022.
- [25] S. Boris Tristán Pérez, “La orientación del comportamiento humano en las instituciones de educación superior Human Behavior Orientation in Higher Education Institutions,” 2019.
- [26] “Ciencias Sociales - Concepto, historia, tipos y características.” Accessed: May 17, 2023. [Online]. Available: <https://concepto.de/ciencias-sociales/>
- [27] J. K. Galbraith, “What is economics? A policy discipline for the real world,” *Revista de Economía Institucional*, vol. 24, no. 46, pp. 3–25, Jan. 2022, doi: 10.18601/01245996.v24n46.02.
- [28] Westreicher Guillermo and Lopez Jose Francisco, “Pérdida - Qué es, definición y concepto | 2023 | Economipedia.” Accessed: May 17, 2023. [Online]. Available: <https://economipedia.com/definiciones/perdida.html>
- [29] J. R. Molina Ríos, J. A. Honores Tapia, N. Pedreira-Souto, and H. P. Pardo León, “Estado del arte: metodologías de desarrollo de aplicaciones móviles,” *3C Tecnología\_Glosas de innovación aplicadas a la pyme*, vol. 10, no. 2, pp. 17–45, Jun. 2021, doi: 10.17993/3ctecno/2021.v10n2e38.17-45.
- [30] A. Fernando and S. Oñate, “UNIVERSIDAD PERUANA UNIÓN FACULTAD DE INGENIERÍA Y ARQUITECTURA Escuela Profesional de Ingeniería de Sistemas.”
- [31] “Car License Plate Detection.” Accessed: Dec. 03, 2023. [Online]. Available: <https://www.kaggle.com/datasets/andrewmvd/car-plate-detection/>

- [32] “wkentaro/labelme: Image Polygonal Annotation with Python (polygon, rectangle, circle, line, point and image-level flag annotation).” Accessed: Dec. 03, 2023. [Online]. Available: <https://github.com/wkentaro/labelme>
- [33] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” May 2015, [Online]. Available: <http://arxiv.org/abs/1505.04597>
- [34] F. I. Diakogiannis, F. Waldner, P. Caccetta, and C. Wu, “ResUNet-a: a deep learning framework for semantic segmentation of remotely sensed data,” Apr. 2019, doi: 10.1016/j.isprsjprs.2020.01.013.
- [35] “YOLO: Real-Time Object Detection.” Accessed: Dec. 05, 2023. [Online]. Available: <https://pjreddie.com/darknet/yolo/>
- [36] C. Cheng, “Real-Time Mask Detection Based on SSD-MobileNetV2,” Aug. 2022, [Online]. Available: <http://arxiv.org/abs/2208.13333>
- [37] “Transferencia de aprendizaje y ajuste | TensorFlow Core.” Accessed: Dec. 05, 2023. [Online]. Available: [https://www.tensorflow.org/tutorials/images/transfer\\_learning?hl=es-419](https://www.tensorflow.org/tutorials/images/transfer_learning?hl=es-419)
- [38] “Chars74K-Digital-English-Font.” Accessed: Dec. 05, 2023. [Online]. Available: <https://www.kaggle.com/datasets/supreethrao/chars74kdigitalenglishfont>
- [39] X. Wang, W. Li, and Z. Wu, “CarDD: A New Dataset for Vision-Based Car Damage Detection,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 7, pp. 7202–7214, Jul. 2023, doi: 10.1109/TITS.2023.3258480.
- [40] “A Free Online Design and Prototyping Tool - Wondershare Mockitt-copy.” Accessed: Dec. 30, 2023. [Online]. Available: <https://mockitt.wondershare.com/home.html>
- [41] H. L. Elghazy and M. W. Fakhr, “Dual- and triple-stream RESUNET/UNET architectures for multi-modal liver segmentation,” *IET Image Process*, vol. 17, no. 4, pp. 1224–1235, Mar. 2023, doi: 10.1049/ipr2.12708.

## ANEXOS

### Anexo A. Alfa de Cronbach

1	Encu	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	P13	P14	P15	SUMA		
2	1	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	75		
3	2	5	3	4	5	5	4	4	4	4	5	3	5	3	5	4	63		
4	3	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	75		
5	4	5	5	5	5	5	5	4	5	5	5	5	4	5	5	5	73		
6	5	5	5	4	5	5	4	4	5	5	4	4	5	3	4	5	67		
7	6	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	75		
8	7	5	5	5	5	5	5	5	5	5	5	5	5	4	5	3	72	K = Numero de items del Instrumento	15
9	8	5	4	5	4	5	4	5	5	5	5	4	5	3	4	5	68	ΣV <sub>i</sub> Sumatoria de las varianzas de los Items	6,391111111
10	9	5	5	5	5	5	5	5	5	5	5	5	4	5	4	73	V <sub>T</sub> Varianza total del Instrumento	20,51555556	
11	10	5	3	4	5	5	3	5	5	5	5	2	2	5	5	64			
12	11	5	3	5	5	5	5	5	5	5	5	5	2	5	4	69			
13	12	5	3	5	5	5	4	4	5	5	5	4	4	3	4	5	66	α = Coeficiente de confiabilidad	0,737651646
14	13	5	5	5	3	5	4	5	5	5	3	1	4	5	3	63			
15	14	4	5	4	4	5	5	4	5	5	4	5	4	3	4	4	65		
16	15	5	5	5	5	5	5	5	5	5	5	5	5	5	5	75			

$$\alpha = \frac{K}{K - 1} \left[ 1 - \frac{\sum V_i^2}{V_T^2} \right]$$

## Anexo B. Script de conversión XML a Imagen binaria

```
import json, os
import numpy as np
import cv2

Label_Class = {'void': 0, 'placa':1 }

def convert(file, name):
    all_labels = json.load(open(file))
    contours = []
    width = all_labels['imageWidth']
    height = all_labels['imageHeight']

    for i_class_object in all_labels['shapes']:
        i_label = i_class_object.get('label')
        countOfPoints = len(i_class_object['points'])
        points = [None] * countOfPoints

        for i, point_dict in enumerate(i_class_object['points']):
            x = int(point_dict[0])
            y = int(point_dict[1])
            points[i] = [x, y]

        points_ = np.array([points], dtype=np.int32)
        contours.append(points_)

    maskImage = np.zeros((height, width), dtype=np.uint8)
    cv2.fillPoly(maskImage, contours, 255)

    savePath = path + "/mask/" + name
    cv2.imwrite(savePath, maskImage)
```

## Anexo C. Funcion de bloques residuales

```
def resblock(X, f):  
    # Hacemos la copia de la entrada  
    X_copy = X  
  
    X = Conv2D( f, kernel_size=(1, 1), strides=(1, 1), kernel_initializer="he_normal")(X)  
    X = BatchNormalization()(X)  
    X = Activation("relu")(X)  
  
    X = Conv2D( f, kernel_size=(3, 3), strides=(1, 1), padding="same", kernel_initializer="he_normal")(X)  
    X = BatchNormalization()(X)  
  
    X_copy = Conv2D( f, kernel_size=(1, 1), strides=(1, 1), kernel_initializer="he_normal")(X_copy)  
    X_copy = BatchNormalization()(X_copy)  
  
    X = Add()([X, X_copy])  
    X = Activation("relu")(X)  
  
    return X
```



## Anexo D. Función de decodificación

```
# Función para escalar y concatenar los valores pasados  
def upsample_concat(x, skip):  
    x = UpSampling2D((2, 2))(x)  
    merge = Concatenate()([x, skip])  
  
    return merge
```

## Anexo E. Arquitectura ResUNet

```
input_shape = (256, 256, 3)

# Forma del tensor de entrada
X_input = Input(input_shape)

# Fase 1
conv1_in = Conv2D(
    16, 3, activation="relu", padding="same", kernel_initializer="he_normal"
)(X_input)
conv1_in = BatchNormalization()(conv1_in)
conv1_in = Conv2D(
    16, 3, activation="relu", padding="same", kernel_initializer="he_normal"
)(conv1_in)
conv1_in = BatchNormalization()(conv1_in)
pool_1 = MaxPool2D(pool_size=(2, 2))(conv1_in)

# Fase 2
conv2_in = resblock(pool_1, 32)
pool_2 = MaxPool2D(pool_size=(2, 2))(conv2_in)

# Fase 3
conv3_in = resblock(pool_2, 64)
pool_3 = MaxPool2D(pool_size=(2, 2))(conv3_in)

# Fase 4
conv4_in = resblock(pool_3, 128)
pool_4 = MaxPool2D(pool_size=(2, 2))(conv4_in)

# Fase 5 (Cuello de Botella)
conv5_in = resblock(pool_4, 256)

# Fase de Escalada 1
up_1 = upsample_concat(conv5_in, conv4_in)
up_1 = resblock(up_1, 128)

# Fase de Escalada 2
up_2 = upsample_concat(up_1, conv3_in)
up_2 = resblock(up_2, 64)

# Fase de Escalada 3
up_3 = upsample_concat(up_2, conv2_in)
up_3 = resblock(up_3, 32)

# Fase de Escalada 4
up_4 = upsample_concat(up_3, conv1_in)
up_4 = resblock(up_4, 16)

# Salida Final
output = Conv2D(1, (1, 1), padding="same", activation="sigmoid")(up_4)

model_seg = Model(inputs=X_input, outputs=output)
```

## Anexo F. Clase DataGenerator de imágenes dinámicas

```
class DataGenerator(tf.keras.utils.Sequence):
    def __init__(
        self,
        ids,
        mask,
        image_dir=".",
        batch_size=16,
        img_h=256,
        img_w=256,
        shuffle=True,
    ):

        self.ids = ids
        self.mask = mask
        self.image_dir = image_dir
        self.batch_size = batch_size
        self.img_h = img_h
        self.img_w = img_w
        self.shuffle = shuffle
        self.on_epoch_end()

    def __len__(self):
        "Get the number of batches per epoch"

        return int(np.floor(len(self.ids)) / self.batch_size)

    def __getitem__(self, index):
        "Generate a batch of data"

        # generate index of batch_size length
        indexes = self.indexes[index * self.batch_size : (index + 1) * self.batch_size]

        # get the ImageId corresponding to the indexes created above based on batch size
        list_ids = [self.ids[i] for i in indexes]

        # get the MaskId corresponding to the indexes created above based on batch size
        list_mask = [self.mask[i] for i in indexes]

        # generate data for the X(features) and y(label)
        X, y = self.__data_generation(list_ids, list_mask)

        # returning the data
        return X, y

    def on_epoch_end(self):
        "Used for updating the indices after each epoch, once at the beginning as well as at the end of each epoch"

        # getting the array of indices based on the input dataframe
        self.indexes = np.arange(len(self.ids))

        # if shuffle is true, shuffle the indices
        if self.shuffle:
            np.random.shuffle(self.indexes)

    def __data_generation(self, list_ids, list_mask):
        "generate the data corresponding the indexes in a given batch of images"
```

```

# create empty arrays of shape (batch_size,height,width,depth)
# Depth is 3 for input and depth is taken as 1 for output because mask consist only of 1 channel.
X = np.empty((self.batch_size, self.img_h, self.img_w, 3))
y = np.empty((self.batch_size, self.img_h, self.img_w, 1))

# iterate through the dataframe rows, whose size is equal to the batch_size
for i in range(len(list_ids)):
    # path of the image
    img_path = "./" + str(list_ids[i])

    # mask path
    mask_path = "./" + str(list_mask[i])

    # reading the original image and the corresponding mask image
    img = io.imread(img_path)
    mask = io.imread(mask_path)

    # resizing and converting them to array of type float64
    img = cv2.resize(img, (self.img_h, self.img_w))
    img = np.array(img, dtype=np.float64)
    mask = cv2.cvtColor(mask, cv2.COLOR_BGR2GRAY)
    mask = cv2.resize(mask, (self.img_h, self.img_w))
    mask = np.array(mask, dtype=np.float64)

    # standardising
    img -= img.mean()
    img /= img.std()

    mask -= mask.mean()
    mask /= mask.std()

    # Adding image to the empty array
    X[i,:] = img

    # expanding the dimension of the image from (256,256) to (256,256,1)
    y[i,:] = np.expand_dims(mask, axis=2)

# normalizing y
y = (y > 0).astype(int)

return X, y

```

## Anexo G. Data Augmentation

```
import os
import cv2
from tqdm import tqdm
from glob import glob
from albumentations import CenterCrop, RandomRotate90, GridDistortion, HorizontalFlip, VerticalFlip

def load_data(path):
    images = sorted(glob(os.path.join(path, "images/*")))
    masks = sorted(glob(os.path.join(path, "masks/*")))
    return images, masks

def create_dir(path):
    if not os.path.exists(path):
        os.makedirs(path)

def augment_data(images, masks, save_path, augment=True):
    H = 256
    W = 256

    for x, y in tqdm(zip(images, masks), total=len(images)):
        name = x.split("/")[-1].split(".")
        """ Extracting the name and extension of the image and the mask. """
        image_name = name[0]
        image_extn = name[1]

        name = y.split("/")[-1].split(".")
        mask_name = name[0]
        mask_extn = name[1]

        """ Reading image and mask. """
        x = cv2.imread(x, cv2.IMREAD_COLOR)
        y = cv2.imread(y, cv2.IMREAD_COLOR)

        """ Augmentation """
        if augment == True:
            aug = CenterCrop(H, W, p=1.0)
            augmented = aug(image=x, mask=y)
            x1 = augmented['image']
            y1 = augmented['mask']

            aug = RandomRotate90(p=1.0)
            augmented = aug(image=x, mask=y)
            x2 = augmented['image']
            y2 = augmented['mask']

            aug = GridDistortion(p=1.0)
            augmented = aug(image=x, mask=y)
            x3 = augmented['image']
            y3 = augmented['mask']

            aug = HorizontalFlip(p=1.0)
            augmented = aug(image=x, mask=y)
            x4 = augmented['image']
            y4 = augmented['mask']

            aug = VerticalFlip(p=1.0)
            augmented = aug(image=x, mask=y)
            x5 = augmented['image']
            y5 = augmented['mask']

        save_images = [x, x1, x2, x3, x4, x5]
```

```

        save_images = [x, x1, x2, x3, x4, x5]
        save_masks = [y, y1, y2, y3, y4, y5]

    else:
        save_images = [x]
        save_masks = [y]

    """ Saving the image and mask. """
    idx = 0
    for i, m in zip(save_images, save_masks):
        i = cv2.resize(i, (W, H))
        m = cv2.resize(m, (W, H))

        if len(images) == 1:
            tmp_img_name = f"{image_name}.{image_extn}"
            tmp_mask_name = f"{mask_name}.{mask_extn}"
        else:
            tmp_img_name = f"{image_name}_{idx}.{image_extn}"
            tmp_mask_name = f"{mask_name}_{idx}.{mask_extn}"

        image_path = os.path.join(save_path, "images", tmp_img_name)
        mask_path = os.path.join(save_path, "masks", tmp_mask_name)

        cv2.imwrite(image_path, i)
        cv2.imwrite(mask_path, m)

        idx += 1

if __name__ == "__main__":
    """ Loading original images and masks. """
    if __name__ == "__main__":
        """ Loading original images and masks. """
        path = "CVC-612/"
        images, masks = load_data(path)
        print(f"Original Images: {len(images)} - Original Masks: {len(masks)}")

        """ Creating folders. """
        create_dir("new_data/images")
        create_dir("new_data/masks")

        """ Applying data augmentation. """
        augment_data(images, masks, "new_data", augment=True)

        """ Loading augmented images and masks. """
        images, masks = load_data("new_data/")
        print(f"Augmented Images: {len(images)} - Augmented Masks: {len(masks)}")

```

## Anexo H. Transfer Learning SDD-MobileNetV2

```
CUSTOM_MODEL_NAME = 'my_ssd_mobnet'
PRETRAINED_MODEL_NAME = 'ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8'
PRETRAINED_MODEL_URL =
'http://download.tensorflow.org/models/object_detection/tf2/20200711/ssd_mobilenet_v2_fpnlite_320x320_coco17_tpu-8.tar.gz'
TF_RECORD_SCRIPT_NAME = 'generate_tfrecord.py'
LABEL_MAP_NAME = 'label_map.pbtxt'

paths = {
    'WORKSPACE_PATH': os.path.join('Tensorflow', 'workspace'),
    'SCRIPTS_PATH': os.path.join('Tensorflow', 'scripts'),
    'APIMODEL_PATH': os.path.join('Tensorflow', 'models'),
    'ANNOTATION_PATH': os.path.join('Tensorflow', 'workspace', 'annotations'),
    'IMAGE_PATH': os.path.join('Tensorflow', 'workspace', 'images'),
    'MODEL_PATH': os.path.join('Tensorflow', 'workspace', 'models'),
    'PRETRAINED_MODEL_PATH': os.path.join('Tensorflow', 'workspace', 'pre-trained-models'),
    'CHECKPOINT_PATH': os.path.join('Tensorflow', 'workspace', 'models', CUSTOM_MODEL_NAME),
    'OUTPUT_PATH': os.path.join('Tensorflow', 'workspace', 'models', CUSTOM_MODEL_NAME, 'export'),
    'TF15_PATH': os.path.join('Tensorflow', 'workspace', 'models', CUSTOM_MODEL_NAME, 'tf15export'),
    'TF16_PATH': os.path.join('Tensorflow', 'workspace', 'models', CUSTOM_MODEL_NAME, 'tf16export'),
    'PROTOC_PATH': os.path.join('Tensorflow', 'protoc')
}

files = {
    'PIPELINE_CONFIG': os.path.join('Tensorflow', 'workspace', 'models', CUSTOM_MODEL_NAME, 'pipeline.config'),
    'TF_RECORD_SCRIPT': os.path.join(paths['SCRIPTS_PATH'], TF_RECORD_SCRIPT_NAME),
    'LABELMAP': os.path.join(paths['ANNOTATION_PATH'], LABEL_MAP_NAME)
}

for path in paths.values():
    if not os.path.exists(path):
        if os.name == 'posix':
            mkdir -p {path}
        if os.name == 'nt':
            mkdir {path}

if os.name == 'nt':
    pip install wget
    import wget

if not os.path.exists(os.path.join(paths['APIMODEL_PATH'], 'research', 'object_detection')):
    git clone https://github.com/tensorflow/models {paths['APIMODEL_PATH']}

if os.name == 'posix':
    apt-get install protobuf-compiler
    cd Tensorflow/models/research && protoc object_detection/protos/*.proto --python_out=. && cp object_detection/packages/tf2/setup.py .

if os.name == 'nt':
    url="https://github.com/protocolbuffers/protobuf/releases/download/v3.15.6/protoc-3.15.6-win64.zip"
    wget.download(url)
    move protoc-3.15.6-win64.zip {paths['PROTOC_PATH']}
    cd {paths['PROTOC_PATH']} && tar -xvf protoc-3.15.6-win64.zip
    os.environ['PATH'] += os.pathsep + os.path.abspath(os.path.join(paths['PROTOC_PATH'], 'bin'))
    cd Tensorflow/models/research && protoc object_detection/protos/*.proto --python_out=. && copy object_detection\packages\tf2\setup.py .
    cd Tensorflow/models/research/ && pip install -e .
```

```

VERIFICATION_SCRIPT = os.path.join(paths['APIMODEL_PATH'], 'research', 'object_detection', 'builders', 'model_builder_tf2_test.py')
# Verify Installation
!python (VERIFICATION_SCRIPT)

import object_detection

if os.name == 'posix':
    !wget (PRETRAINED_MODEL_URL)
    !mv (PRETRAINED_MODEL_NAME+'.tar.gz') (paths['PRETRAINED_MODEL_PATH'])
    !cd (paths['PRETRAINED_MODEL_PATH']) && tar -zxvf (PRETRAINED_MODEL_NAME+'.tar.gz')
if os.name == 'nt':
    !wget.download(PRETRAINED_MODEL_URL)
    !move (PRETRAINED_MODEL_NAME+'.tar.gz') (paths['PRETRAINED_MODEL_PATH'])
    !cd (paths['PRETRAINED_MODEL_PATH']) && tar -zxvf (PRETRAINED_MODEL_NAME+'.tar.gz')

labels = [{'name': 'character', 'id': 1}]

with open(files['LABELMAP'], 'w') as f:
    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')

if not os.path.exists(files['TF_RECORD_SCRIPT']):
    !git clone https://github.com/nicknochnack/GenerateTFRecord (paths['SCRIPTS_PATH'])

!python (files['TF_RECORD_SCRIPT']) -x (os.path.join(paths['IMAGE_PATH'], 'train')) -l (files['LABELMAP']) -o
(os.path.join(paths['ANNOTATION_PATH'], 'train.record'))
!python (files['TF_RECORD_SCRIPT']) -x (os.path.join(paths['IMAGE_PATH'], 'test')) -l (files['LABELMAP']) -o
(os.path.join(paths['ANNOTATION_PATH'], 'test.record'))

if os.name == 'posix':
    !cp (os.path.join(paths['PRETRAINED_MODEL_PATH'], PRETRAINED_MODEL_NAME, 'pipeline.config'))
(os.path.join(paths['CHECKPOINT_PATH']))
if os.name == 'nt':
    !copy (os.path.join(paths['PRETRAINED_MODEL_PATH'], PRETRAINED_MODEL_NAME, 'pipeline.config'))
(os.path.join(paths['CHECKPOINT_PATH']))

import tensorflow as tf
from object_detection.utils import config_util
from object_detection.protos import pipeline_pb2
from google.protobuf import text_format

config = config_util.get_configs_from_pipeline_file(files['PIPELINE_CONFIG'])

pipeline_config = pipeline_pb2.TrainEvalPipelineConfig()
with tf.io.gfile.GFile(files['PIPELINE_CONFIG'], "r") as f:
    proto_str = f.read()
    text_format.Merge(proto_str, pipeline_config)

pipeline_config.model.num_classes = len(labels)
pipeline_config.train_config.batch_size = 4
pipeline_config.train_config.fine_tune_checkpoint = os.path.join(paths['PRETRAINED_MODEL_PATH'],
PRETRAINED_MODEL_NAME, 'checkpoint', 'ckpt-0')
pipeline_config.train_config.fine_tune_checkpoint_type = "detection"
pipeline_config.train_input_reader.label_map_path = files['LABELMAP']
pipeline_config.train_input_reader.tf_record_input_reader.input_path[:] = [os.path.join(paths['ANNOTATION_PATH'], 'train.record')]
pipeline_config.eval_input_reader[0].label_map_path = files['LABELMAP']
pipeline_config.eval_input_reader[0].tf_record_input_reader.input_path[:] = [os.path.join(paths['ANNOTATION_PATH'], 'test.record')]

config_text = text_format.MessageToString(pipeline_config)
with tf.io.gfile.GFile(files['PIPELINE_CONFIG'], "wb") as f:
    f.write(config_text)

TRAINING_SCRIPT = os.path.join(paths['APIMODEL_PATH'], 'research', 'object_detection', 'model_main_tf2.py')
command = "python {} --model_dir={} --pipeline_config_path={} --num_train_steps=20000".format(TRAINING_SCRIPT,
paths['CHECKPOINT_PATH'],
files['PIPELINE_CONFIG'])

```



## Anexo I. Preprocesamiento del dataset Chars74K

```
# Utilizar ImageDataGenerator para cargar y preprocesar automáticamente los datos
datagen = ImageDataGenerator(
    rescale=1./255,
    zoom_range=0.2,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=15,
    horizontal_flip=False,
    vertical_flip=False
)

# Generadores de datos para train, test y validación
generador_train = datagen.flow_from_directory(
    ruta_train,
    target_size=(TAMANO_IMG, TAMANO_IMG),
    color_mode='grayscale',
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    shuffle=True
)

    ruta_test,
    target_size=(TAMANO_IMG, TAMANO_IMG),
    color_mode='grayscale',
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    shuffle=False
)

generador_validacion = datagen.flow_from_directory(
    ruta_validacion,
    target_size=(TAMANO_IMG, TAMANO_IMG),
    color_mode='grayscale',
    batch_size=BATCH_SIZE,
    class_mode='categorical',
    shuffle=False
)
```

## Anexo J. Arquitectura de la Red Neuronal Convolutacional

```
modeloCNN2 = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(18, (3,3), activation='relu', input_shape=(128, 128, 1)),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(32, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),

    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(250, activation='relu'),
    tf.keras.layers.Dense(350, activation='relu'),
    tf.keras.layers.Dense(36, activation='softmax')
])
```

## Anexo K. Conversión de Modelos a modelos TensorFlow Lite

```
# CONVERSIÓN DEL MODELO RESNET EN TFLITE
converter = tf.lite.TFLiteConverter.from_keras_model(model_seg)
tflite_float_model = converter.convert()

f = open('plateSegmetationML.tflite', "wb")
f.write(tflite_float_model)
f.close()

# CONVERSIÓN DEL MODELO SSD-MOBILENETV2 EN TFLITE
TFLITE_SCRIPT = os.path.join(paths['APIMODEL_PATH'], 'research', 'object_detection', 'export_tflite_graph_tf2.py ')
command = "python {} --pipeline_config_path={} --trained_checkpoint_dir={} --output_directory={}"
        .format(TFLITE_SCRIPT, files['PIPELINE_CONFIG'], paths['CHECKPOINT_PATH'], paths['TFLITE_PATH'])

#CONVERSIÓN DEL CNN EN TFLITE
converter = tf.lite.TFLiteConverter.from_keras_model(cnn)
tflite_float_model = converter.convert()

f = open('alphanumericML.tflite', "wb")
f.write(tflite_float_model)
f.close()
```

## Anexo L. Data Augmentation dataset

```
import os
import cv2
from tqdm import tqdm
from glob import glob
from albumentations import CenterCrop, RandomRotate90, GridDistortion, HorizontalFlip, VerticalFlip

def load_data(path):
    images = sorted(glob(os.path.join(path, "rgb/*")))
    return images

def create_dir(path):
    if not os.path.exists(path):
        os.makedirs(path)

def augment_data(images, save_path, augment=True):
    for x, in tqdm(zip(images), total=len(images)):
        image_name, image_extn = os.path.splitext(os.path.basename(x))

        """ Reading image """
        x = cv2.imread(x, cv2.IMREAD_COLOR)
        H, W = x.shape[:2]

        """ Augmentation """
        if augment == True:
            aug = CenterCrop(H, W, p=1.0)
            augmented = aug(image=x)
            x1 = augmented["image"]

            aug = RandomRotate90(p=1.0)
            augmented = aug(image=x)
            x2 = augmented['image']

            aug = GridDistortion(p=1.0)
            augmented = aug(image=x)
            x3 = augmented['image']

            aug = HorizontalFlip(p=1.0)
            augmented = aug(image=x)
            x4 = augmented['image']
```

```

        aug = VerticalFlip(p=1.0)
        augmented = aug(image=x)
        x5 = augmented['image']

        save_images = [x, x1, x2, x3, x4, x5]

    else:
        save_images = [x]

    """ Saving the image """
    idx = 0
    for i in save_images:
        i = cv2.resize(i, (W, H))

        if len(images) == 1:
            tmp_img_name = f"{image_name}{image_extn}"

        else:
            tmp_img_name = f"{image_name}_{idx}{image_extn}"

        image_path = os.path.join(save_path, "images", tmp_img_name)
        cv2.imwrite(f"Lableme_to_Mask_Fallas/NewData2/Flamantes/{tmp_img_name}", i)

        idx += 1
if __name__ == "__main__":
    """ Loading original images. """
    path = "Lableme_to_Mask_Fallas/"
    images = load_data(path)
    print(f"Original Images: {len(images)}")

    """ Applying data augmentation. """
    augment_data(images, "Lableme_to_Mask_Fallas/NewData2/NoFlamantes", augment=True)

    """ Loading augmented images """
    images = load_data("Lableme_to_Mask_Fallas/NewData2/NoFlamantes/")
    print(f"Augmented Images: {len(images)} ")

```

## Anexo M. Arquitectura de Red Neuronal Convolutacional

```
modeloCNN2 = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(18, (3,3), activation='relu', input_shape=(TAMANO_IMG, TAMANO_IMG, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(32, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),

    tf.keras.layers.Dropout(0.5),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(100, activation='relu'),
    tf.keras.layers.Dense(50, activation='relu'),
    tf.keras.layers.Dense(10, activation='relu'),
    tf.keras.layers.Dense(2, activation='softmax')
])
```

## Anexo N. Archivo de configuración Spring Boot

```
application.properties
1
2 spring.datasource.url=jdbc:sqlserver://SQL5097.site4now.net:1433;databaseName=db_aa2c12_maucar07001
3 spring.datasource.username=db_aa2c12_maucar07001_admin
4 spring.datasource.password=Mauricio1997@
5 spring.datasource.driver-class-name=com.microsoft.sqlserver.jdbc.SQLServerDriver
6 spring.jpa.hibernate.ddl-auto=update
7 spring.jpa.show-sql=true
8 spring.jpa.properties.hibernate.format_sql=true
9 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.SQLServer2012Dialect
10 spring.datasource.initialization-mode=always
11
12 server.port=8083
13
```

## Anexo O. Dependencias Android

```
build.gradle (app) X
55 dependencies {
56
57     implementation 'androidx.core:core-ktx:1.8.0'
58     implementation 'androidx.appcompat:appcompat:1.6.1'
59     implementation 'com.google.android.material:material:1.5.0'
60     implementation 'androidx.constraintlayout:constraintlayout:2.1.4'
61     //OpenCV
62     implementation project(path: ':opencv')
63     //Nav
64     def nav_version :String = "2.7.5"
65     implementation "androidx.navigation:navigation-fragment-ktx:$nav_version"
66     implementation "androidx.navigation:navigation-ui-ktx:$nav_version"
67     // CameraX core library
68     def camerax_version :String = "1.4.0-alpha02"
69     implementation "androidx.camera:camera-core:$camerax_version"
70     // CameraX Camera2 extensions
71     implementation "androidx.camera:camera-camera2:$camerax_version"
72     // CameraX lifecycle library
73     implementation "androidx.camera:camera-lifecycle:$camerax_version"
74     // CameraX View class
75     implementation "androidx.camera:camera-view:$camerax_version"
76     //TensorFlow
77     implementation 'org.tensorflow:tensorflow-lite-metadata:0.1.0'
78     implementation 'org.tensorflow:tensorflow-lite:2.9.0'
79     implementation 'org.tensorflow:tensorflow-lite-gpu:2.9.0'
80     implementation 'org.tensorflow:tensorflow-lite-support:0.4.2'
81     implementation 'org.tensorflow:tensorflow-lite-select-tf-ops:2.9.0'
82     //WindowManager
83     implementation 'androidx.window:window:1.1.0-alpha02'
84     testImplementation 'junit:junit:4.13.2'
85     androidTestImplementation 'androidx.test.ext:junit:1.1.5'
86     androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
87     //Retrofit
88     implementation 'com.squareup.retrofit2:retrofit:2.9.0'
89     implementation 'com.google.code.gson:gson:2.10.1'
90     implementation 'com.squareup.retrofit2:converter-gson:2.9.0'
91     //coroutines
92     implementation 'org.jetbrains.kotlinx:kotlinx-coroutines-android:1.5.0'
93     // ViewModel
94     implementation "androidx.lifecycle:lifecycle-viewmodel-ktx:2.2.0"
95     // LiveData
96     implementation "androidx.lifecycle:lifecycle-livedata-ktx:2.2.0"
97     //ViewPager2
98     implementation 'androidx.viewpager2:viewpager2:1.0.0'
99     //Bagger Hit
100    implementation("com.google.dagger:hilt-android:2.44")
101    kapt("com.google.dagger:hilt-android-compiler:2.44")
102    //ROOM
103    def room_version :String = "2.5.0"
104    implementation("androidx.room:room-runtime:$room_version")
105    annotationProcessor("androidx.room:room-compiler:$room_version")
106    kapt("androidx.room:room-compiler:$room_version")
107    implementation "androidx.room:room-ktx:$room_version"
108
```



## Anexo P. Archivo XML de la pantalla principal

```
<androidx.coordinatorlayout.widget.CoordinatorLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <com.google.android.material.bottomappbar.BottomAppBar
        android:id="@+id/bottomAppBar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom"
        android:background="@color/white"
        app:fabCradleMargin="10dp"
        app:fabCradleRoundedCornerRadius="18dp">

        <com.google.android.material.bottomnavigation.BottomNavigationView
            android:id="@+id/bottomNavigationView"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginEnd="20dp"
            android:background="@android:color/transparent"
            app:labelVisibilityMode="labeled"
            app:menu="@menu/bottom_menu" />

    </com.google.android.material.bottomappbar.BottomAppBar>

    <com.google.android.material.floatingactionbutton.FloatingActionButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:backgroundTint="@color/white"
        android:src="@drawable/ic_baseline_document_scanner_24"
        app:layout_anchor="@id/bottomAppBar"
        app:maxImageSize="48dp"
        android:id="@+id/fab"
        app:tint="@color/lavender"
        app:shapeAppearanceOverlay="@style/ShapeAppearanceOverlay.MyApp.ExtendedFloatingActionButton" />

</androidx.coordinatorlayout.widget.CoordinatorLayout>

</androidx.drawerlayout.widget.DrawerLayout>
```

## Anexo Q. Configuración de la cámara

```
private fun bindCameraUseCases() {

    // CameraProvider
    val cameraProvider =
        cameraProvider ?: throw IllegalStateException("Camera initialization failed.")
    val cameraSelector =
        CameraSelector.Builder().requireLensFacing(CameraSelector.LENS_FACING_BACK).build()
    preview =
        Preview.Builder()

        .setTargetRotation(fragmentCameraBinding.viewFinder.display.rotation)
        .setTargetResolution(Size(1500, 1800))
        .build()

    imageAnalyzer =
        ImageAnalysis.Builder()
        .setTargetRotation(fragmentCameraBinding.viewFinder.display.rotation)
        .setTargetResolution(Size(1500, 1800))
        .setBackpressureStrategy(ImageAnalysis.STRATEGY_KEEP_ONLY_LATEST)
        .setOutputImageFormat(ImageAnalysis.OUTPUT_IMAGE_FORMAT_RGBA_8888)
        .build()
    // The analyzer can then be assigned to the instance
    .also {
        it.setAnalyzer(cameraExecutor) { image ->
            if (::bitmapBuffer.isInitialized) {
                bitmapBuffer = Bitmap.createBitmap(
                    image.width,
                    image.height,
                    Bitmap.Config.ARGB_8888
                )
            }
            detectPlaca(image)
        }
    }

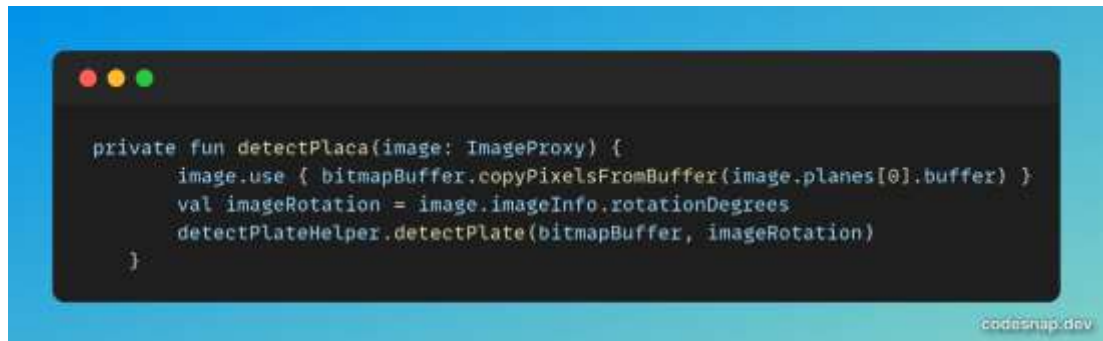
    cameraProvider.unbindAll()

    try {
        camera = cameraProvider.bindToLifecycle(this, cameraSelector, preview, imageAnalyzer)
        preview?.setSurfaceProvider(fragmentCameraBinding.viewFinder.surfaceProvider)
    } catch (exc: Exception) {
        Log.e(TAG, "Use case binding failed", exc)
    }
}
```

codesnap.net

## Anexo R. Listener para escuchar eventos entre clases

```
private fun detectPlaca(image: ImageProxy) {  
    image.use { bitmapBuffer.copyPixelsFromBuffer(image.planes[0].buffer) }  
    val imageRotation = image.imageInfo.rotationDegrees  
    detectPlateHelper.detectPlate(bitmapBuffer, imageRotation)  
}
```



## Anexo S. Clase DetectPlateHelper

```
class DetectPlacaHelper(
    var numThreads: Int = 4,
    val context: Context,
    val classifierListener: ClassifierListener?
) {

    private var interpreter_ReUNet: Interpreter? = null
    private var targetWidth_ReUNet: Int = 0
    private var targetHeight_ReUNet: Int = 0

    private var interpreter_ssdmobilenet: SsdMobilenet? = null
    private var targetWidth_ssdmobilenet : Int = 0
    private var targetHeight_ssdmobilenet : Int = 0

    private var interpreter_cnn: Interpreter? = null
    private var targetWidth_cnn : Int = 0
    private var targetHeight_cnn : Int = 0

    /**
     * Constructor de la clase
     */
    init {
        if (setupModelResUNet()) {

            targetWidth_ReUNet = interpreter_ReUNet!!.getInputTensor(0).shape()[2]
            targetHeight_ReUNet = interpreter_ReUNet!!.getInputTensor(0).shape()[1]

        } else {
            classifierListener?.onError("TFLite ResUNet failed to init. ✖")
        }

        if (setupModelSSDMobileNet()) {

            targetWidth_ssdmobilenet = 320
            targetHeight_ssdmobilenet = 320

        } else {
            classifierListener?.onError("TFLite ResUNet failed to init. ✖")
        }

        if (setupCnn()) {

            targetWidth_cnn = 128
            targetHeight_cnn = 128

        } else {
            classifierListener?.onError("TFLite CNN failed to init. ✖")
        }
    }
}
```

codesnap.dev

## Anexo T. Preprocesamiento de la imagen

```
private fun processInputImage(
    image: Bitmap,
    imageRotation: Int
): TensorImage? {
    val height = image.height
    val width = image.width
    val cropSize = min(height, width)
    val imageProcessor = ImageProcessor.Builder()
        .add(Rot90Op(-imageRotation / 90))
        .add(
            ResizeOp(
                targetHeight_ReUNet,
                targetWidth_ReUNet,
                ResizeOp.ResizeMethod.BILINEAR
            )
        )
        .add(NormalizeOp(127.5f, 127.5f))
        .build()
    val tensorImage = TensorImage(DataType.FLOAT32)
    tensorImage.load(image)
    return imageProcessor.process(tensorImage)
}
```

codesnap.dev

## Anexo U. Método de conversión de a un bitmap

```
private fun convertByteBufferToBitmap(
    byteBuffer: ByteBuffer,
    imgSizeX: Int,
    imgSizeY: Int
): Bitmap? {
    byteBuffer.rewind()
    byteBuffer.order(ByteOrder.nativeOrder())
    val bitmap = Bitmap.createBitmap(imgSizeX, imgSizeY, Bitmap.Config.ARGB_4444)
    val pixels = IntArray(imgSizeX * imgSizeY)
    for (i in 0 until imgSizeX * imgSizeY) if (byteBuffer.float == 1f) pixels[i] =
        Color.argb(255, 255, 255, 255) else pixels[i] = Color.argb(0, 0, 0, 0)
    bitmap.setPixels(pixels, 0, imgSizeX, 0, 0, imgSizeX, imgSizeY)
    return bitmap
}
```

codesnap.dev

## Anexo V. Operaciones con OpenCV para recorta la placa

```
private fun processOpencv(
    mask : Bitmap,
    img : Bitmap
): Bitmap? {
    //MASK
    val mat_mask = BitmapToMat(mask)
    val width = img.width
    val height = img.height
    val matImageOriginal = Mat(height, width, CvType.CV_8UC3)
    val imageDataOriginal = ByteArray(width * height * 3)
    img.getPixels(
        IntArray(width * height),
        0,
        width,
        0,
        0,
        width,
        height
    )

    for (i in 0 until width * height) {
        val pixelValue = img.getPixel(i % width, i / width)
        imageDataOriginal[i * 3] = (pixelValue shr 16).toByte() // Canal Rojo
        imageDataOriginal[i * 3 + 1] = (pixelValue shr 8).toByte() // Canal Verde
        imageDataOriginal[i * 3 + 2] = (pixelValue and 0xFF).toByte() // Canal Azul
    }

    matImageOriginal.put(0, 0, imageDataOriginal)

    val resizedMat = Mat(width, height, CvType.CV_8UC1)
    Imgproc.resize(mat_mask, resizedMat, Size(width.toDouble(), height.toDouble()))

    val resizedMatRGB = Mat(resizedMat.rows(), resizedMat.cols(), CvType.CV_8UC3)
    Core.merge(listOf(resizedMat), resizedMatRGB)
    val gray = Mat()
    Imgproc.cvtColor(resizedMatRGB, gray, Imgproc.COLOR_BGR2GRAY)

    val contours = ArrayList<Contour>()
    Imgproc.findContours(gray, contours, Mat(), Imgproc.RETR_EXTERNAL, Imgproc.CHAIN_APPROX_SIMPLE)

    if (contours.isNotEmpty()) {
        val contourSegmented = contours.toList().maxByOrNull { Imgproc.contourArea(it) }

        if (contourSegmented != null) {
            val boundingRect = Imgproc.boundingRect(contourSegmented)

            val regionRecortada = Mat(matImageOriginal, Rect(boundingRect.x, boundingRect.y, boundingRect.width, boundingRect.height))
            val bitmapReturn = Bitmap.createBitmap(regionRecortada.cols(), regionRecortada.rows(), Bitmap.Config.ARGB_8888)
            Utils.matToBitmap(regionRecortada, bitmapReturn)

            return bitmapReturn
        }
    }
    Log.e("Contornos ", " No encontrados ")
    return null
}
```

## Anexo W. Método detectObjects

```
private fun detectObjects(imgCrop : Bitmap): MutableList<Bitmap>(){
    try {
        var caracteres : MutableList<Bitmap>
        processInputImage_ssdmobilenet(imgCrop)?.let { image →
            if (interpreter_ssdmobilenet == null) {
                setupModelSSDMobileNet()
            }
            val inputFeature0 = TensorBuffer.createFixedSize(intArrayOf(1, 320, 320, 3), DataType.FLOAT32)
            inputFeature0.loadBuffer(image.buffer)
            val result = interpreter_ssdmobilenet.process(inputFeature0)
            val outputFeature0 = result?.outputFeature0AsTensorBuffer
            val outputFeature1 = result?.outputFeature1AsTensorBuffer
            caracteres = cropCharacters2(imgCrop, outputFeature1, outputFeature0, imgCrop.width, imgCrop.height, 0.25f)
            return caracteres
        }

        Inatch (e: Exception){
            Log.e(TAG, "Funcion detectObjects fallo ${e.message}")
        }
        return null
    }
}
```



## Anexo X. Recorte de los caracteres

```
private fun cropCaracteres2(
    img: Bitmap,
    boxesTensor: TensorBuffer?,
    scoresTensor: TensorBuffer?,
    width: Int,
    height: Int,
    minConfidence: Float
): MutableList<Bitmap> {
    val caracteres = mutableListOf<Bitmap>()

    val boxes: FloatArray? = boxesTensor?.floatArray
    val scores: FloatArray? = scoresTensor?.floatArray

    if (boxes != null && scores != null) {
        val numBoxes = scores.size
        val boxesOrdenadasIndices = (0 until numBoxes).sortedBy { boxes[it + 4 + 1] }
        for (i in boxesOrdenadasIndices) {
            val index = i + 4
            if (index + 3 < boxes.size && i < scores.size) {
                val score = scores[i]
                if (score > minConfidence && score <= 1.0) {
                    val ymin = boxes[index]
                    val xmin = boxes[index + 1]
                    val ymax = boxes[index + 2]
                    val xmax = boxes[index + 3]

                    val roi = floatArrayOf(
                        ymin + height,
                        xmin + width,
                        ymax + height,
                        xmax + width
                    )

                    val clippedROI = floatArrayOf(
                        roi[0].coerceIn(0f, height.toFloat()),
                        roi[1].coerceIn(0f, width.toFloat()),
                        roi[2].coerceIn(0f, height.toFloat()),
                        roi[3].coerceIn(0f, width.toFloat())
                    )

                    try {
                        val region = Bitmap.createBitmap(
                            img,
                            clippedROI[1].toInt(),
                            clippedROI[0].toInt(),
                            (clippedROI[3] - clippedROI[1]).toInt(),
                            (clippedROI[2] - clippedROI[0]).toInt()
                        )
                        val letra = escalarImagen(region)

                        caracteres.add(letra)
                    } catch (e: Exception) {
                        Log.e("cropCaracteres2", "Error creando Bitmap", e)
                    }
                } else {
                    Log.e("cropCaracteres2", "Indice fuera de rango: $i")
                }
            }
        }
    }
    return caracteres
}
```

codemap.org



## Anexo Z. Retorno de la placa al hilo principal

```
override fun onResults(results: String?, inferenceTime: Long) {

    activity?.runOnUiThread {
        delete("placa")
        if (results != null && !isProcessingResult) {
            vehiculoViewModel.obtenerVehiculo(results!!)
            vehiculoViewModel.vehiculo.observe(viewLifecycleOwner) { vehiculo ->
                if ( vehiculo?.idVehiculo != -1L && !isProcessingResult) {
                    isProcessingResult = true
                    vehiculo!!.placa?.let { savePlaca(it) }
                    replaceFragment( ReporteLegalFragment() )
                    vehiculoViewModel.vehiculo.removeObservers(viewLifecycleOwner)
                } else {
                    isProcessingResult = false
                }
            }
        }
    }
}
```

codesnap.dev

## Anexo 1A. Diálogo para guardar el propietario

```
private fun showBottomDialog(vehiculo: Vehiculo) {

    val dialog = Dialog(requireContext())
    dialog.requestWindowFeature(Window.FEATURE_NO_TITLE)
    dialog setContentView(R.layout.bottomsheetlayout_guardar)
    edittxtPropietario = dialog.findViewById(R.id.txt_guardar_propietario_nombre)
    edittxtDireccion = dialog.findViewById(R.id.txt_guardar_propietario_direccion)
    edittxtTelefono = dialog.findViewById(R.id.txt_guardar_propietario_telefono)
    edittxtEmail = dialog.findViewById(R.id.txt_guardar_propietario_email)
    cancelButton = dialog.findViewById(R.id.btn_guardar_propietario_cancelar)
    guardarButton = dialog.findViewById(R.id.btn_guardar_propietario_guardar)
    edittxtPropietario.setText(vehiculo.propietario.nombre)
    edittxtDireccion.setText(vehiculo.propietario.direccion)
    edittxtTelefono.setText(vehiculo.propietario.telefono)
    edittxtEmail.setText(vehiculo.propietario.email)

    cancelButton.setOnClickListener { dialog.dismiss() }
    guardarButton.setOnClickListener { guardarPropietario(vehiculo, dialog) }

    dialog.show()
    dialog.window!!.setLayout(
        ViewGroup.LayoutParams.MATCH_PARENT,
        ViewGroup.LayoutParams.WRAP_CONTENT
    )
    dialog.window!!.setBackgroundDrawable(ColorDrawable(Color.TRANSPARENT))
    dialog.window!!.attributes.windowAnimations = R.style.DialogAnimation
    dialog.window!!.setGravity(Gravity.BOTTOM)
}
```

codesnap.dev

## Anexo 2B. Guardar propietario

```
private fun guardarPropietario(vehiculo: Vehiculo, dialog: Dialog){
    var nombrePropietario = ""
    var direccionPropietario = ""
    var telefonoPropietario = ""
    var emailPropietario = ""
    var idPropietario : Long? = -1
    var cedulaPropietario :String? = ""

    nombrePropietario = edittxPropietario.text.toString()
    direccionPropietario = edittxtDireccion.text.toString()
    telefonoPropietario = edittxtTelefono.text.toString()
    emailPropietario = edittxtEmail.text.toString()
    idPropietario = vehiculo.propietario.id
    cedulaPropietario = vehiculo.propietario.cedula

    if(isValidForm(nombrePropietario,direccionPropietario, emailPropietario,telefonoPropietario)){
        var propietario = Propietario(
            idPropietario, nombrePropietario,
            cedulaPropietario,direccionPropietario,
            telefonoPropietario, emailPropietario
        )

        propietarioViewModel.insertPropietario(proprietario)
        vehiculoViewModel.insertVehiculo(vehiculo)
        revisionesVehicularViewModel.insertAllRevisionesVehicular(vehiculo.revisionVehicular)
        multasVehicularViewModel.insertAllMultasVehicular(vehiculo.multasVehiculo)
        emisionesMatriculaViewModel.insertAllEmisionesmatricula(vehiculo.emisionMatricula)
        matriculaViewModel.insertAllMatriculas(vehiculo.matricula)
        denunciaViewModel.insertAllDenuncias(vehiculo.denuncias)
        dialog.dismiss()
        replaceFragment( HomeFragment() )
        Toast.makeText(requireContext(), "Propietario Guardado!", Toast.LENGTH_SHORT).show()
    }
}
```

codecademy

### Anexo 3C. Método getAllPropietarios

```
private fun getAllPropietarios() {
    propietarioWithVehiculoQViewModel.obtenerAllPropietarioWithVehiculo()
    propietarioWithVehiculoQViewModel.propietarioWithVehiculoViewModel.observe(
        viewLifecycleOwner,
        Observer { vehiculosdb ->
            adapterInfoVehiculosDB = InfoVehiculosdbAdapter(
                requireActivity(),
                vehiculosdb as ArrayList<PropietarioWithVehiculosdb>
            )
            recyclerView?.adapter = adapterInfoVehiculosDB
            adapterInfoVehiculosDB?.onDeleteClickListener =
                object : InfoVehiculosdbAdapter.OnDeleteClickListener {
                    override fun onItemDeleteClick(placa: String) {
                        vehiculoViewModel.deleteRegistros(placa)
                    }

                    override fun onItemEditClick(placa: String) {
                        propietarioViewModel.getAllPropietarioFromDatabaseProgressBar(placa)
                    }
                }
        }
    )
}
```

codenap.dev

## Anexo 4D. Editar propietario

```
private fun showBottomDialogEdit(propietario: Propietario) {  
  
    val dialog = Dialog(requireContext())  
    dialog.requestWindowFeature(Window.FEATURE_NO_TITLE)  
    dialog setContentView(R.layout.bottomsheetlayout_guardar)  
    edittxPropietario = dialog.findViewById(R.id.txt_guardar_propietario_nombre)  
    edittxtDireccion = dialog.findViewById(R.id.txt_guardar_propietario_direccion)  
    edittxtTelefono = dialog.findViewById(R.id.txt_guardar_propietario_telefono)  
    edittxtEmail = dialog.findViewById(R.id.txt_guardar_propietario_email)  
    cancelButton = dialog.findViewById(R.id.btn_guardar_propietario_cancelar)  
    guardarButton = dialog.findViewById(R.id.btn_guardar_propietario_guardar)  
    val titulo = dialog.findViewById<TextView>(R.id.txt_titulo_dialog)  
    titulo.setText("Actualizar Propietario")  
    edittxPropietario.setText(propietario.nombre)  
    edittxtDireccion.setText(propietario.direccion)  
    edittxtTelefono.setText(propietario.telefono)  
    edittxtEmail.setText(propietario.email)  
  
    cancelButton.setOnClickListener { dialog.dismiss() }  
    guardarButton.setOnClickListener {  
        guardarPropietario(propietario, dialog)  
        getAllPropietarios()  
    }  
  
    dialog.show()  
    dialog.window!!.setLayout(  
        ViewGroup.LayoutParams.MATCH_PARENT,  
        ViewGroup.LayoutParams.WRAP_CONTENT  
    )  
    dialog.window!!.setBackgroundDrawable(ColorDrawable(Color.TRANSPARENT))  
    dialog.window!!.attributes.windowAnimations = R.style.DialogAnimation  
    dialog.window!!.setGravity(Gravity.BOTTOM)  
}
```

codesnap.dev

## Anexo 5E. Método DetectStetic

```
private fun detectStetic(image: ImageProxy) {  
    image.use { bitmapBuffer.copyPixelsFromBuffer(image.planes[0].buffer) }  
    val imageRotation = image.imageInfo.rotationDegrees  
    detectSteticHelper.detectStetic(bitmapBuffer, imageRotation)  
}
```

codesnap.dev



## Anexo 6F. Clase DetectSteticHelper

```
class DetectSteticHelper(  
    var numThreads: Int = 4,  
    val context: Context,  
    val classifierListener: ClassifierListener?  
) {  
  
    private var interpreter_cnn: Interpreter? = null  
    private var targetWidth_cnn: Int = 0  
    private var targetHeight_cnn: Int = 0  
  
    /**  
     * Constructor de la clase  
     */  
    init {  
  
        if (setupCnn()) {  
  
            targetWidth_cnn = 320  
            targetHeight_cnn = 320  
  
        } else {  
            classifierListener?.onError("TFLite CNN failed to init. ✖")  
        }  
    }  
}
```

codesnap.dev

## Anexo 7G. Inferencia del modelo de detección de anomalías estéticas

```
fun detectStetic(
    bitmap: Bitmap,
    rotation: Int
) {
    processInputImage(bitmap, rotation)?.let { image →
        if (interpreter_cnn == null) {
            setupCnn()
        }
        val inputFeature0 = TensorBuffer.createFixedSize(intArrayOf(1, 320, 320, 3), DataType.FLOAT32)
        val output = TensorBuffer.createFixedSize(intArrayOf(1, 2), DataType.FLOAT32)
        inputFeature0.loadBuffer(image.buffer)
        interpreter_cnn?.run(inputFeature0.buffer, output.buffer)
        val posiblesEtiquetas = listOf("Flamante", "No_Flamante")
        val outputArray: FloatArray = output.toFloatArray()
        val maxIndex = outputArray.indexOfLast { it == outputArray.maxOrNull() }
        val predictedClassLabel = posiblesEtiquetas[maxIndex]
        classifierListener?.onResults(predictedClassLabel)
    }
}
```

## Anexo 8H. Pantalla about

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <de.hdodenhof.circleimageview.CircleImageView
        android:id="@+id/circleimage_user"
        android:layout_width="140dp"
        android:layout_height="140dp"
        android:layout_gravity="center"
        android:layout_marginTop="60dp"
        android:src="@drawable/icon"
        app:civ_border_color="@color/white"
        app:civ_border_width="8dp"
        app:civ_circle_background_color="@color/white" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="28dp"
        android:layout_marginTop="20dp"
        android:layout_marginRight="20dp"
        android:gravity="center"
        android:orientation="horizontal">

        <TextView
            android:id="@+id/textview_nombre"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="18dp"
            android:layout_gravity="center"
            android:textSize="28sp"
            android:textStyle="bold"
            android:text="SafeSenseApp" />

    </LinearLayout>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="28dp"
        android:layout_marginTop="20dp"
        android:layout_marginRight="20dp"
        android:gravity="center_vertical"
        android:orientation="vertical">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="18dp"
            android:textStyle="bold"
            android:text="Acerca de"
            android:textSize="15sp"
            android:layout_margin="10dp" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginLeft="18dp"
            android:text="@string/acercade" />

    </LinearLayout>
</LinearLayout>
```

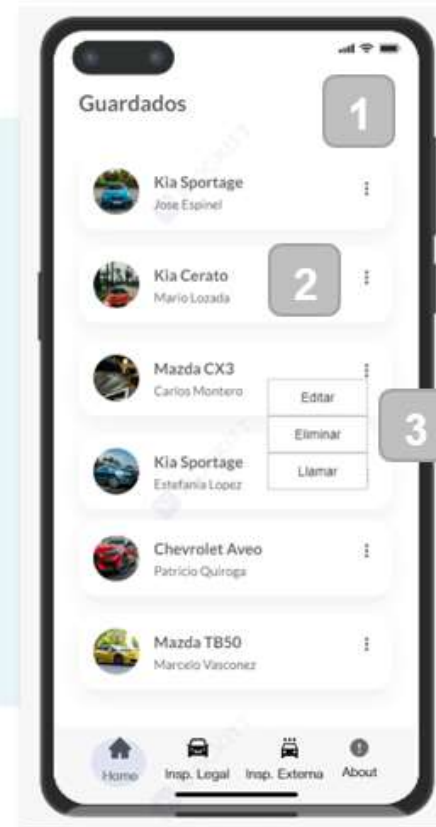
Anexo 9I. Alfa Cronbach validación cuestionario test de la app

Encuestados	1. La navegación dentro de la aplicación es clara y sencilla.	2. La función Reporte Legal del vehículo es útil.	3. La función Inspección Estética del vehículo es útil.	4. Me gusta la disposición de los elementos en la interfaz.	5. El rendimiento general de la aplicación es satisfactorio.	Suma
Persona 1	5	4	3	5	5	22
Persona 2	5	4	4	5	5	23
Persona 3	4	3	3	3	4	17
Persona 4	4	4	4	5	4	21
Persona 5	5	5	5	5	5	25
Persona 6	5	4	5	5	5	24
Persona 7	4	5	3	5	5	22
<b>Varianza</b>	0.245	0.408	0.694	0.490	0.204	
<b>Sumatoria de Varianzas</b>	2.041					
<b>Varianza de la suma de los Items</b>	5.714285714					
	$\alpha$ :	Coefficiente de confiabilidad	0.803571429			
	$k$ :	Num. de items	5			
	$\sum_{i=1}^k S_i^2$	Sumatoria de las varianzas	2.041			
	$S_t^2$	Varianza total	5.714285714			



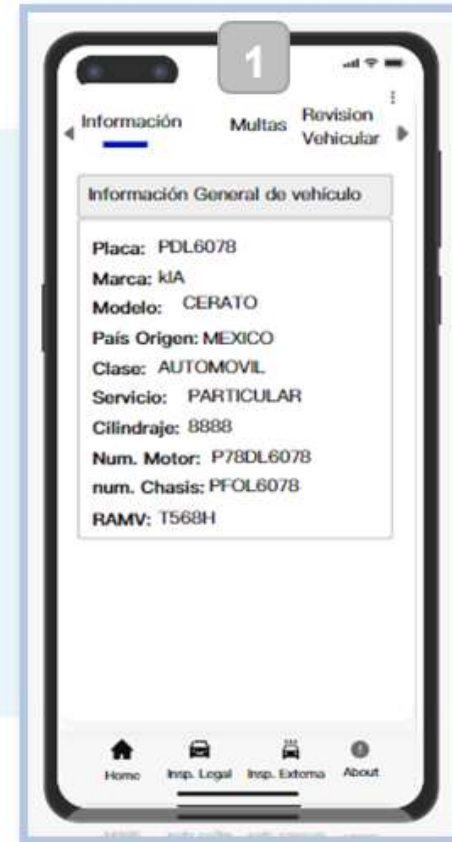
## PAGINA INICIAL

- 1 Se visualiza el listado de los propietarios guardados.
- 2 Al presionar el sobre un propietario, se visualizará toda la información de propietario y del vehículo.
- 3 Menú para editar, eliminar y llamar al propietario.



## DETALLES DEL VEHÍCULO Y PROPIETARIO

- 1 Al deslizar las distintas opciones se puede acceder a la respectiva información del vehículo.



## INSPECCIÓN LEGAL DEL VEHÍCULO

- 1 Al presionar en botón se podrá realizar la inspección legal del vehículo.





## GUARDAR PROPIETARIO

- 1** Al presionar en botón despliega el menú para poder guardar la información.
- 2** Diálogo para mostrar información del propietario.
- 3** Al presionar en botón se podrá guardar el propietario.
- 4** Al presionar en botón se contraerá el Diálogo.



## INSPECCIÓN ESTÉTICA DEL VEHÍCULO

- 1** Al presionar en botón se podrá realizar la inspección estética del vehículo.
- 2** La barra indicara el resultado de la inspección.



## PANTALLA ABOUT

- 1** Al presionar en botón se visualiza la información de la aplicación.
- 2** Indica una breve descripción de la aplicación.
- 3** Indica la versión de la aplicación.
- 4** Indica la información del desarrollador.

