

UNIVERSIDAD AMAZÓNICA DE PANDO

ÁREA DE CIENCIAS Y TECNOLOGÍA

CARRERA DE INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

“APLICACIÓN DE DETECCIÓN DE INFRACCIÓN DE TRÁNSITO, PARA EL RECONOCIMIENTO DE PLACAS DE AUTOMÓVILES, EN LOS PUNTOS DE SEMÁFOROS DE LA CIUDAD DE COBIJA”

PROYECTO DE GRADO PRESENTADO PARA OBTENER EL TÍTULO
ACADÉMICO DE LICENCIADO EN INGENIERÍA DE SISTEMAS

POSTULANTE: Ruddy Chao

TUTOR: Ing. Juan Carlos Gallardo Jiménez

ASESOR: Ing. Gabriel Lazcano Balanza

Cobija - Pando – Bolivia

2018

AGRADECIMIENTOS

Primeramente, agradecer a Dios, por darme la vida, salud, fuerza y entusiasmo, que brinda incondicionalmente, aun cuando le fallamos.

A mi familia que siempre está a mi lado dándome consejos, para que siga adelante.

A mi padre por sus amplias charlas que me da, que a la larga se convierten en catedra, de cómo debería ser una persona.

Un especial y grato agradecimiento a mi madre por ser la mentora, por apoyarme siempre, por comprenderme.

A mi tutor Ing. Juan Carlos Gallardo Jimenez, por sus constantes sugerencias, consejo, paciencia y confianza, decirle que le guardo un profundo respeto, cariño y gratitud.

A mi asesor Ing. Gabriel Lazcano Balanza, por la valiosa orientación, en el avance del proyecto, decirle que le guardo un profundo respeto.

A Rosmery Rosaycela Mamani Catari, por su amistad y apoyo constante.

Y a todas las personas que de alguna manera me apoyaron antes, durante el proyecto, me enseñaron cosas valiosas.

Para todos ellos y a quienes no nombro muchas muchas gracias.

DEDICATORIA

A mis padres por el apoyo que me dan día a día.

A mis hermanas, por estar siempre presentes
acompañando para poderme realizar.

A Rosmery Rosaycela Mamani Catari por su
apoyo incondicional.

A todos muchas gracias de todo corazón.

RESUMEN

El presente proyecto denominado” **APLICACIÓN DE DETECCIÓN DE INFRACCIÓN DE TRÁNSITO, PARA EL RECONOCIMIENTO DE PLACAS DE AUTOMÓVILES, EN LOS PUNTOS DE SEMÁFOROS DE LA CIUDAD DE COBIJA**”, se apoya en la teoría y práctica de la visión por computador, es una tecnología que permite emular de algún modo, el trabajo que realiza el ojo humano, en nuestro caso está aplicada para poder detectar infracción de tránsito de vehículos, de esa forma reconocer automáticamente las placas vehiculares.

De acuerdo al análisis realizado se establece: ”**Constantes hechos de tránsito que infringen normativa, en el cual las cámaras de seguridad no registran ni identifican de forma automática las placas de los automóviles, en los semáforos de la ciudad de Cobija**”. A tal efecto se planteó: “**Desarrollar una aplicación de detección de infracción de tránsito, para el reconocimiento de placas de automóviles, en los puntos de semáforos de la ciudad de cobija, en base a los fundamentos que nos da la metodología XP**”.

Se ha desarrollado el software denominada detección de infracción en java, para detectar infracción de cada uno de los automóviles en el paso de cebra, basándose en el reconocimiento de la placa, esto permite calcular automáticamente si el vehículo cometió la infracción , para su posterior análisis, extracción y guardado del número de la placa. Consta de la obtención del archivo a analizar (video), marcar el paso cebra, el reconocimiento de las placas de los vehículos, la detección de infracción (si el vehículo pasa el paso cebra en semáforo color rojo), y su posterior análisis y guardado de los caracteres de la placa.

Para el desarrollo del proyecto se ha trabajado utilizando la librería de código abierto OpenCV basada en el lenguaje de programación C++, la cual está diseñada para el tratamiento de imágenes por visión por computador, el lenguaje de programación java con el entorno de desarrollo integrado Net Beans, la librería de código abierto Teserract que está desarrollada para la obtención de caracteres a partir de imágenes preferentemente binarizadas.

Con el desarrollo sistema Informático se garantiza que la detección de la infracción del vehículo se realice a un 100 %, la detección de la placa se haga en un 80%, y la obtención de matrículas para su posterior registro se realice en un 73 %.

PALABRAS CLAVES: Visión por Computador, OpenCV, OCR, Viola-Jones.

SUMMARY

The present project called "APPLICATION OF DETECTION OF TRANSIT INFRACTION, FOR THE RECOGNITION OF MOTORCYCLE PLATES, IN THE POINTS OF TRAFFIC LIGHTS OF THE CITY OF COBIJA", was based on the theory and practice of the vision by computer, it is a technology that it allows us to emulate in some way, the work that the human eye performs, in our case, is related to the possibility of vehicle transit, in this way it automatically recognizes the vehicle plates.

According to the analysis carried out, it is established: "Constant traffic events that violate regulations, in which the security cameras do not automatically register or identify the license plates of the cars at the traffic lights of the city of Cobija." One such effect was raised: "Develop a traffic detection application for the recognition of automobile license plates at the traffic lights in the city of Cobija, based on the fundamentals given by the XP methodology".

The infraction detection software has been developed in java, to detect the infraction of each of the cars in the zebra crossing, based on the recognition of the license plate, this allows to calculate automatically if the vehicle committed the infraction, for its Subsequent analysis, extraction and storage of the plate number. It consists of obtaining the file to analyze (video), mark the zebra crossing, the recognition of the plates of the vehicles, detection of infraction in the zebra crossing in red traffic light, and its subsequent analysis and saving of the characters of the plate.

For the development of the project it has been used in the open source library. OpenCV based on the programming language. C ++, which is designed for the treatment of images by computer vision, the programming language. , the open source library. Teserract that is developed to obtain characters from preferably binarized images.

With the development of the computer system, it is guaranteed that the detection of the violation of the vehicle will be carried out 100%, the detection of the plate will be done in 80%, and the obtaining of license plates for its subsequent registration will be carried out in 73%.

KEYWORDS: Computer Vision, OpenCV, OCR, Viola-Jones.

ÍNDICE

1. MARCO INTRODUCTORIO	1
1.1. ANTECEDENTES	2
1.2. DESCRIPCIÓN DEL PROBLEMA.....	3
1.3. OBJETIVOS	4
1.3.1. Objetivo General.....	4
1.3.2. Objetivos Específicos	4
1.4. ALCANCES	5
1.5. METODOLOGÍA	5
2. MARCO REFERENCIAL	7
2.1. MARCO INSTITUCIONAL	8
2.2. MARCO LEGAL.....	10
2.3. MARCO TEÓRICO.....	11
2.3.1. Visión artificial	11
2.3.2. La Visión Humana.....	13
2.3.3. Procesamiento de imágenes:.....	14
2.3.4. Segmentación de imágenes.....	14
2.3.5. Imágenes Digitales	17
2.3.6. Reconocimiento automático de placas vehiculares(ANPR)	19
2.3.7. Reconocimiento óptico de caracteres	20
2.3.8. Clasificador de imágenes:.....	23
2.4. PRUEBAS DE CALIDAD DE SOFTWARE	24
2.4.1. Pruebas de caja negra	25
2.4.2. ISO/IEC 9126	25

2.5.	METODOLOGÍA Y HERRAMIENTAS.....	26
2.5.1.	Programación Extrema(XP):	26
2.5.2.	Objetivos y Características:	27
2.5.3.	Algunas prácticas de XP:.....	27
2.5.4.	Fases de la programación extrema (XP).....	28
2.5.5.	Herramientas de la metodología XP.....	32
2.5.6.	Roles de la metodología XP.	33
2.6.	HERRAMIENTAS UTILIZADAS	34
2.6.1.	Librerías de procesamiento de imagen:	34
2.6.2.	Java	38
2.6.3.	NetBeans 7.3.....	39
3.	MARCO APLICATIVO	40
3.1.	DESARROLLO DEL PROYECTO	41
3.2.	EXPLORACIÓN Y PLANIFICACIÓN DEL PROYECTO EN BASE A LA METODOLOGÍA XP.....	43
3.2.1.	Historias de usuario:	43
3.2.2.	Priorización de las historias de usuarios.....	44
3.2.3.	Spike Arquitectónico	49
3.2.4.	Metáfora de negocio	50
3.2.5.	Asignación de roles del proyecto.....	50
3.2.6.	Plan de entrega.....	50
3.2.7.	Representación general en diagrama de caso de uso de la aplicación	52
3.2.8.	Diagrama de secuencia de la aplicación	53
3.3.	FASE DE ITERACIONES Y PRODUCCIÓN:	54

3.3.1.	Primera iteración.....	54
3.3.2.	Segunda iteración	60
3.3.3.	Tercera iteración.....	70
3.3.4.	Cuarta Iteración	77
3.4.	PRUEBAS DE RECONOCIMIENTO	80
3.4.1.	Detección de la placa.....	80
3.4.2.	Detección de la infracción del automóvil.....	81
3.4.3.	Reconocimiento de caracteres de la placa:.....	82
3.5.	CALIDAD DE SOFTWARE.....	83
3.5.1.	Evaluación de la funcionalidad de la aplicación	84
3.5.2.	Evaluación de la confiabilidad	84
3.5.3.	Evaluación de la usabilidad	85
3.5.4.	Evaluación de la eficiencia.....	85
3.5.5.	Evaluación de Mantenimiento	86
3.5.6.	Evaluación de la portabilidad	86
3.5.7.	Resultado general	87
4.	CONCLUSIONES Y RECOMENDACIONES.....	88
4.1.	CONCLUSIONES. -.....	89
4.2.	RECOMENDACIONES.....	90
5.	BIBLIOGRAFÍA.....	91
6.	ANEXOS.....	93

ÍNDICE DE FIGURAS

FIGURA 1 <i>Proceso de la Metodología XP</i>	6
FIGURA 2. <i>Organigrama de la D.D.T.T.S.V</i>	9
FIGURA 3 <i>Etapas de la Visión Artificial</i>	11
FIGURA 4. <i>El ojo humano</i>	13
FIGURA 5. <i>Representación de una imagen</i>	18
FIGURA 6 <i>Representación gráfica del modelo HSV</i>	19
FIGURA 7 <i>Proceso general de un sistema ANPR</i>	20
FIGURA 8. <i>Proceso general de un sistema OCR</i>	21
FIGURA 9. <i>Pasos para crear clasificador</i>	24
FIGURA 10. <i>Componentes de Tesseract</i>	38
FIGURA 11 <i>Ejemplo de código java</i>	39
FIGURA 12 <i>Interfaz de desarrollo de NetBeans</i>	39
FIGURA 13 <i>Diagrama de flujo de la aplicación</i>	41
FIGURA 14. <i>Proceso general de la aplicación</i>	42
FIGURA 15. <i>Representación general de la aplicación</i>	52
FIGURA 16. <i>Diagrama de secuencia General</i>	53
FIGURA 17. <i>Imágenes positivas</i>	55
FIGURA 18. <i>Imágenes negativas</i>	56
FIGURA 19. <i>Marcación y obtención de datos</i>	57
FIGURA 20. <i>Verificación de la creación de imágenes positivas</i>	58
FIGURA 21. <i>Entrenamiento de clasificador</i>	58
FIGURA 22. <i>Screenshot de la detección de placa</i>	59
FIGURA 23. <i>Imágenes Positivas (placa Nacionales)</i>	61
FIGURA 24. <i>Reconocimiento de placas vehicular</i>	62
FIGURA 25. <i>Casos de Uso. - Marcador de paso Cebra</i>	63
FIGURA 26. <i>Marcador de paso cebra</i>	64
FIGURA 27. <i>Casos de Uso. – Segmentación De paso cebra</i>	66
FIGURA 28. <i>Pasos de la segmentación paso cebra mediante Visión por Computador</i>	68

FIGURA 29. <i>Prueba de la Segmentación del Paso cebra</i>	69
FIGURA 30. <i>Imagen 1 Detección de Infracciónl, Imagen 2 auto Infractor</i>	73
FIGURA 31. <i>Caso de uso segmentación e implementación de Teserract</i>	75
FIGURA 32. <i>Pasos del tratamiento e implementación de Teserract mediante Visión por Computador</i>	76
FIGURA 33 <i>Caso de uso Gestión de resultados</i>	78
FIGURA 34 <i>Visualización de resultados</i>	79

ÍNDICE DE TABLAS

TABLA 1. <i>Ejemplo de una mascara</i>	15
TABLA 2. <i>Mascaras para la segmentación por línea</i>	16
TABLA 3 <i>Etapas básica del OCR</i>	22
TABLA 4 <i>Fases de la metodología XP</i>	31
TABLA 5. <i>Historia de Usuario</i>	32
TABLA 6. <i>Tarjeta CRC</i>	32
TABLA 7. <i>Modelo de prueba de aceptación</i>	33
TABLA 8. <i>Historia de usuario N. 1</i>	44
TABLA 9. <i>Historia de usuario N. 2</i>	45
TABLA 10. <i>Historia de usuario N. 2</i>	45
TABLA 11. <i>Historia de usuario N. 4</i>	46
TABLA 12. <i>Historia de Usuario N. 5</i>	47
TABLA 13. <i>Historia de usuario N. 6</i>	48
TABLA 14. <i>Historia de usuario N. 7</i>	49
TABLA 15. <i>Iteraciones y posible entrega</i>	51
TABLA 16. <i>Primera iteración</i>	54
TABLA 17. <i>Tarjeta CRC detección de placa</i>	59
TABLA 18. <i>Plan de entrega da la 2ºda iteración</i>	60
TABLA 19. <i>Tarjeta CRC marcador de paso cebra</i>	63
TABLA 20 <i>Tarjeta CRC Segmentación de paso cebra</i>	65
TABLA 21. <i>Plan de entrega de 3ª iteración</i>	70
TABLA 22 <i>Detección de infracción (Segmentación paso cebra)</i>	71
TABLA 23. <i>Caso de uso detección de infracción</i>	72
TABLA 24. <i>Tarjeta CRC Segmentación e implementación de Teserract para el OCR</i>	74
TABLA 25 <i>Plan de entrega de 4ta iteración</i>	77
TABLA 26 <i>Tarjeta CRC Gestión de resultados</i>	78
TABLA 28 <i>Placas detectadas</i>	80
TABLA 29. <i>Vehículos infractores reconocidos</i>	81

TABLA 30. <i>Tabulación de resultados de placas reconocidas</i>	82
TABLA 31. <i>Descripción de medidas</i>	83
TABLA 32 <i>Resumen de evaluación a la funcionalidad de la aplicación</i>	84
TABLA 33 <i>Resumen de evaluación a la confiabilidad de la aplicación</i>	84
TABLA 34 <i>Resumen de evaluación a la usabilidad de la aplicación</i>	85
TABLA 35 <i>Resumen de evaluación a la eficiencia de la aplicación</i>	85
TABLA 36 <i>Resumen de evaluación al mantenimiento</i>	86
TABLA 37 <i>Resumen de evaluación a la portabilidad</i>	86
TABLA 38 <i>Resultados generales</i>	87

CAPITULO I

1.MARCO INTRODUCTORIO

1.1. ANTECEDENTES

La visión por computador o visión artificial es uno de los campos informáticos en constante avance en cuanto la extracción de características, detección de objetos y clasificación de imágenes, las cuales se pueden realizar por técnicas de visión por computador o redes neuronales, para su posterior interpretación.

En la ciudad de cobija no existen proyectos de visión por computador que se hayan realizado o implementado, es por eso que se pretende realizar una aplicación de detección de infracción de tránsito, mediante técnicas de visión por computador.

Un primer trabajo corresponde a (GÓMEZ, 2009) que realizo un proyecto de fin de carrera denominado “Mejora del Tráfico en un Cruce Regulado por Semáforos, Mediante un Sistema Basado en Visión Artificial”. La misma supone el desarrollo y pruebas experimentales de un sistema inteligente que, a partir de cámaras IP y antenas Bluetooth instaladas en los semáforos, ejecute en tiempo real, decisiones sobre los semáforos para la optimización del tráfico en una intersección.

En la Universidad Nacional Autónoma de México (RICARDO, 2012), realizo como tesis de grado, un proyecto denominado” Hacia un esquema de localización y reconocimiento de matrículas vehiculares”, se trata de un Sistema capaz de poder localizar y reconocer matrículas vehiculares tomando en cuenta diversas situaciones o condiciones en las que se obtuvo la imagen.

La Organización Mundial de la Salud en el año 2014, en su trabajo de investigación “Informe mundial sobre prevención de los traumatismos causados por el tránsito”, nos indica que la instalación de cámaras y el uso de reconocimiento automático de matrículas (ANPR) en los semáforos, redujo entre un 7% y un 32% los accidentes de tránsito en las intersecciones de diferentes avenidas de Australia.

En el instituto politécnico de México (Montiel, 2010), con la de tesis de grado “Reconocimiento de Placas Vehiculares”, tiene como objetivo desarrollar un algoritmo para la localización y reconocimiento automático de placas vehiculares.

Las investigaciones anteriormente descritas, muestran la importancia, que tienen los Sistemas de Reconocimiento automático de Matriculas (ANPR) a través del uso de OCR.

El contexto en el cual se realiza el presente proyecto, es en la Dirección Departamental de Transito Transporte y Seguridad Vial Pando, dependiente del comando Policial Pando y Dirección Nacional de Tránsito, Transporte y Seguridad Vial; siendo el ente encargado de regular, controlar y resolver los problemas de circulación de vehículos y peatones, prevenir e investigar accidentes de tránsito mediante actividades y técnicas especializadas. Además de proporcionar políticas que coadyuben a una buena circulación vial,

1.2. DESCRIPCIÓN DEL PROBLEMA

Se tiene 4.422 accidentes y hechos de tránsito registrados en Bolivia al primer bimestre de 2016, las colisiones representan 50,56% (2.236 casos), le siguen los atropellos a peatón con 16,32% (722 casos) y para los choques a objetos fijos un 8,20% (363 casos), entre los tres suman 75,08%, según la Dirección Nacional de Tránsito (La Razon, 2016).

Dentro de la ciudad de Cobija en los últimos años ocurren constantes hechos de tránsito en las intersecciones de las calles más específicamente en los semáforos, para el primer bimestre de 2016 hubo un total de 167 hechos de tránsito, de los cuales 30 fueron a peatones, 15 choque a objetos fijos, 82 colisiones (La Razon, 2016).

Esto se da por la falta de precaución, estado de ebriedad, exceso de velocidad, malas condiciones del vehículo, y falta de educación vial, expresan como resultado diversos hechos de tránsito, que dejan personas fallecidas, heridas, además de pérdidas materiales e infracciones de tránsito y el no registro de infractores que se pasan en rojo el paso de cebra.

Las cámaras de monitoreo dentro de la ciudad de Cobija, no cuentan con un software externo o interno que sea capaz de poder identificar los vehículos que cometieron la infracción.

Los Sistemas Reconocimiento Automático de Placas (ANPR por sus siglas en inglés), son necesarios y con ellos el control sobre los infractores es más exacto e instantáneo. Tales

sistemas están dirigidos principalmente a ayudar a los conductores en su tarea de conducir con menos errores y eliminar algunos comportamientos de inseguridad cuando se conduce.

Planteamiento del problema:

Por todo lo anterior descrito se propone como problema principal:

“Constantes hechos de tránsito que infringen normativa, en el cual las cámaras de seguridad no registran ni identifican de forma automática las placas de los automóviles, en los semáforos de la ciudad de Cobija.”

1.3. OBJETIVOS

1.3.1. Objetivo General

Desarrollar una aplicación de detección de infracción de tránsito, para el reconocimiento de placas de automóviles, en los puntos de semáforos de la ciudad de Cobija, en base a los fundamentos que nos da la metodología XP y técnicas de visión computador.

1.3.2. Objetivos Específicos

- Diseñar la arquitectura general para la detección de placas, a través de los fundamentos que proporciona el Lenguaje de Modelado Unificado (UML).
- Implementar la aplicación de detección de infracción y placa, a través la librería de visión artificial y código abierto OpenCV mediante el lenguaje de programación java.
- Validar la aplicación mediante captura de imágenes reales o grabaciones, a través de cámaras (teléfonos inteligentes, cámaras filmadoras, etc.), para procesar las detecciones de infracción de vehículos.

1.4. ALCANCES

El presente proyecto tiene como alcances y límites:

- Segmentar a los automóviles infractores, por medio del paso de cebra.
- Abarcar concretamente a la captura de las placas de automóviles, para procesar y guardar la infracción de los mismos.
- La muestra que se toma en cuenta para las diferentes pruebas, se basa en vehículos que tengan placa nacional, que estén debidamente reglamentadas y en un buen estado. Las pruebas y el funcionamiento de la aplicación se realizan con archivos de videos grabados durante el día y soleado.
- Proyecto de Grado realizado para la División de Educación Vial dependiente de la Dirección Departamental de Transito Transporte y Seguridad Vial de Cobija – Pando dependiente del Comando Policial Departamental.

1.5. METODOLOGÍA

La metodología de programación Extremas prescribe un conjunto de valores y prácticas que permiten a los desarrolladores dedicarse a lo que se hace mejor; programar, esta elimina la necesidad de dedicar tiempo a labores tediosas y burocráticas, prescritas por los no agiles.

A continuación, se muestra la figura que de todo el proceso a tomar en cuenta dentro de la metodología XP.

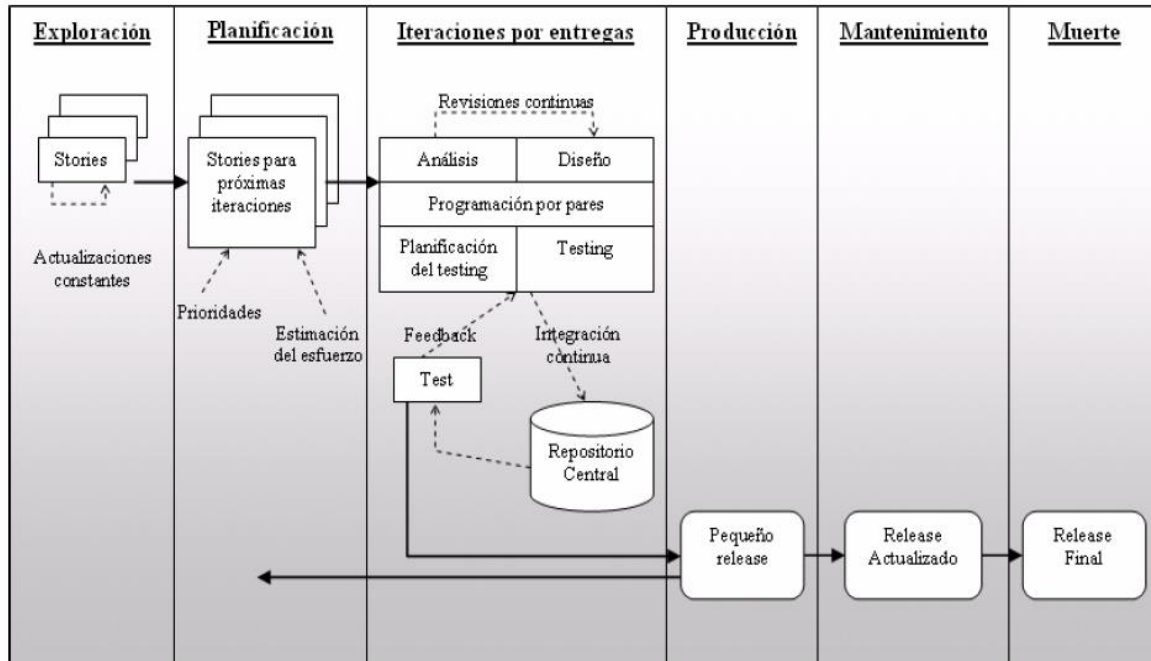


FIGURA 1 *Proceso de la Metodología XP*

Fuente: (Gonzales, 2017)

La figura 1 muestra el proceso de la Metodología XP, la misma se basa en el desarrollo de software ágil, se enfoca en 6 actividades estructurales: exploración, planificación, iteración por entregas, producción, mantenimiento, muerte. Cabe destacar cada una de ellas, la exploración se basa en la obtención de cada uno de los requerimientos basadas en historias de usuario, así como también, elegir cada una de las herramientas a utilizar y por ultimo realizar una metáfora de negocio para que el usuario entienda; en cuanto a la fase de planificación se prioriza el trabajo y se divide por iteraciones entregables para producción, se determina fechas, prioridades, de tal manera que al final de todas las iteraciones se tenga un producto terminado.

CAPITULO II

2. MARCO REFERENCIAL

2.1. MARCO INSTITUCIONAL

La Dirección Departamental de Transito Transporte y seguridad Vial Pando siendo un ente gubernamental dependiente del comando Policial Pando y Dirección Nacional de Tránsito, Transporte y Seguridad Vial; se encuentra ubicada en la ciudad de Cobija calle Miguel Becerra Toranzo frente al Banco Unión.

Formula acciones educativas y de seguridad vial, el procesamiento y resolución de infracciones, emisión de los certificados de antecedentes, controlar el Seguro Obligatorio de Accidentes de Tránsito (SOAT) la especialización de ingeniería de tránsito en la Facultad de Ciencias Policiales de la Universidad Policial(Unipol); hace cumplir leyes, códigos y normas que coadyuve a fortalecer el tránsito, Transito, Transporte y Seguridad Vial a favor de la sociedad boliviana.

Misión:

Regular. Controlar y resolver los problemas de circulación de vehículos y peatones, prevenir e investigar accidentes de tránsito mediante actividades técnicas especializadas

Visión:

Ser una de las mejores Direcciones logrando metas y objetivos Institucionales planteados en cada gestión demostrando transparencia y eficacia en el trabajo realizado y así ser una dirección respetadas y reconocida por la sociedad boliviana, que goce de confianza y el respeto de los funcionarios policiales.

La estructura organizacional de la Dirección Departamental de Transito está conformada en niveles jerárquicos y división de funciones, como se puede ver en la figura 2.

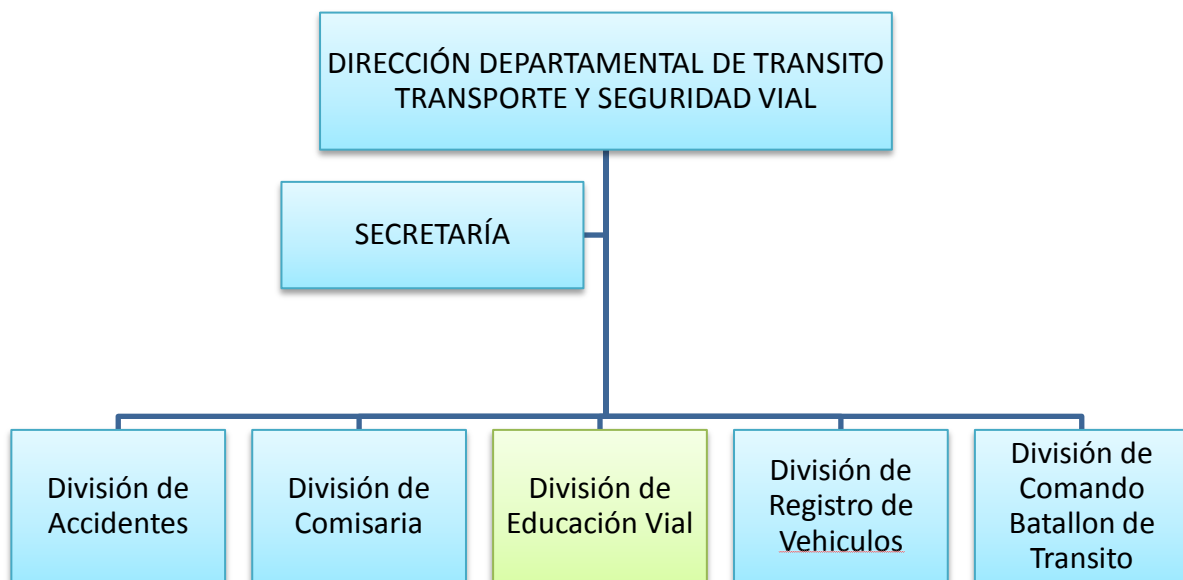


FIGURA 2. *Organigrama de la D.D.T.T.S.V*

Fuente: (Dirección Departamental de Transito Transporte y Seguridad Vial, 2018)

Según el manual de funciones a la cual se rige la Dirección Departamental de Transito, se tiene:

- **Director Departamental de Transito Transporte y Seguridad Vial:** Responsable de la dirección, control y ejecución de la administración y cumplimiento de las leyes, reglamentos y demás disposiciones vigentes que regulan el funcionamiento de la dirección.
- **Jefe de División de Accidentes de Tránsito:** Desarrollar medios de control para lograr la eficiencia y eficacia en las investigaciones de su división; controlar, supervisar y velar el buen desarrollo de la investigación de hechos de tránsito.
- **Jefe de División de Comisaria:** Conocer, resolver y sancionar infracciones tipificadas en el código nacional de tránsito y su reglamento
- **Jefe de División de Educación Vial:** Hacer cumplir normas de código de tránsito y otras relacionadas al tráfico vehicular, peatones y vías.
- **Jefe de División de Registros de Vehículos:** Controlar el trabajo de tramitación de vehículos en sus diferentes clases.

- **Jefe de División de Comando Batallón de Transito:** Responsable del mantenimiento de la disciplina, entrenamiento, instrucción del personal policial a su mando.

2.2.MARCO LEGAL

La Dirección Departamental de Transito está amparada y funciona entorno a las siguientes leyes y reglamentos:

1. Constitución Política del Estado
2. Ley orgánica de la Policía Boliviana
3. Ley N. 2028 código de tránsito y su reglamento
4. Ley de Administración y control gubernamental (ley N. 1178) de 23 de junio de 1990.
5. Normas básicas de sistema de organización administrativa, aprobó mediante la resolución suprema N. 217055 de 30 de mayo de 1997.

2.3. MARCO TEÓRICO

En esta sección se mostrará los fundamentos teóricos en las cuales se basa el proyecto, pasando de lo más general a lo más específico. Así también se expondrá de forma explícita los fundamentos de la metodología de Programación Extrema (XP) y las herramientas utilizadas para el presente proyecto.

2.3.1. Visión artificial

También llamado visión por computador es una disciplina científica que incluye métodos para adquirir, procesar, analizar y comprender las imágenes del mundo real con el fin de producir información numérica o simbólica que el computador pueda entender. Tal cual lo hacemos los humanos con los ojos, la visión por computador trata de producir el mismo efecto para que las computadoras puedan percibir y comprender una imagen o secuencia de imágenes. El objetivo de la visión por computador es extraer características de una imagen para su descripción e interpretación por la computadora.

Tiene como finalidad la extracción de información del mundo físico a partir de imágenes digitales, utilizando para ellos un computador, la misma se aplica en distintos campos, detección de rostros, reconocimientos de objetos que se encuentran en la vida real, así como también restauración de objetos. (Minardi, 2014)

Las aplicaciones que se les puede dar al sistema de visión artificial o visión computacional están: Robótica Móvil y vehículos autónomos, Manufactura, Interpretación de áreas y de satélites, Análisis de imágenes microscópicas, Análisis de imágenes para la Astronomía.

Las etapas de un sistema de visión artificial son 4:



FIGURA 3 *Etapas de la Visión Artificial*

Fuente: Elaboración Propia

- **Captura:**

Adquisición de imágenes digitales mediante algún tipo de dispositivo, depende mucho de la calidad de la imagen que se captura para su posterior análisis.

- **Pre procesado:**

Se hace con el fin de detectar y eliminar las fallas que puedan existir en la imagen. Es la técnica más utilizada para mejorar la calidad, como la corrección de iluminación, eliminación de ruidos, igualación y mejora del contraste; todo esto se lo tiene con la pérdida de información mínima.

Para la reducción de ruidos existen dos métodos que son: ruido gaussiano, ruido impulsivo.

- **Segmentación:**

Consiste en dividir una imagen digital en regiones homogéneas con respecto a una o más característica (puede ser el color o el brillo), esto facilita el análisis o el reconocimiento, la misma ayuda a localizar una cara o los límites de una palabra. Pasado el análisis se obtendrá otra imagen donde cada pixel estará etiquetada la misma corresponderá a un objeto en particular.

La segmentación de la imagen es el proceso de asignación de una etiqueta a cada píxel de la imagen de forma que los píxeles que compartan la misma etiqueta también tendrán ciertas características visuales similares.

Para la segmentación de imágenes existe distintas técnicas por las cuales pueden ser analizadas que son: por textura, contorno, basada en la umbralización, la detección de contornos, crecimiento de regiones y movimiento.

- **Reconocimiento:**

Distinguir los diferentes objetos de acuerdo a su característica. (Grupo de Inteligencia Computacional, 2015)

2.3.1.1. Aplicaciones de la visión artificial

Muchas veces las soluciones humanas tienden a ser inexactas, lentas, por lo que en esas ocasiones la visión artificial se convierte en una alternativa mejorada. Estas pueden constituirse en una herramienta por sí sola (la visión es el único sensor), o parte de otro sistema (referido a la navegación robótica donde la visión es parte del robot).

2.3.2. La Visión Humana

Es uno de los 5 sentidos sensoriales más importantes que tienen los seres vivos.

Tanto la visión por computador o la visión artificial tienen sus principios en los que es la visión humana.

La luz incide en nuestros ojos transformados en impulsos nerviosos por la retina. Para ello, la retina posee dos tipos de células especializadas en captar la luz (Foto-receptores); los conos y los bastones.

- **Conos** Son células sensibles a los colores. Esta sensibilidad es debida a unas moléculas presentes en los conos llamadas ospinas. Hay tres tipos de conos, cada uno de ellos sensibles a la longitud de onda (color) distintos: Rojo, verde y azul.
- **Bastones:** Son células sensibles a la intensidad de la luz (El brillo); la molécula presente en esta célula y que las dota de esta sensibilidad se llama rodopsina (Marin, 2006).

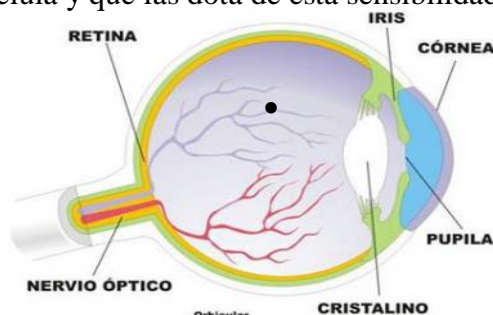


FIGURA 4. *El ojo humano*

Fuente: (Marin, 2006)

2.3.3. Procesamiento de imágenes:

Las imágenes digitalizadas que se llega a obtener, de algún tipo de cámara, no siempre llegan a tener una buena calidad, ello se puede deber a que no se toma bien la imagen o un mal procedimiento de obtención de la imagen.

Tiene fundamentos parecidos a la visión por computador o visión artificial, la diferencia radica en el objetivo final. El objetivo final es la mejora de la calidad de la imagen para su posterior utilización o interpretación.

- Por tanto, el procesamiento de imagen mejora y hace algunos retoques a la imagen original, existen varias técnicas ya implementadas que nos pueden facilitar a la hora de procesar una imagen.
- **RUIDO**: Es una variación aleatoria de la información de color o de claridad que no está presente en el objeto de la imagen, lo cual puede tener lugar de forma no deseada debido a un sensor y un sistema de circuitos de un escáner o cámara digital. (Baggio D. L., 2015)

Para la eliminación de ruidos existen técnicas como: La premeditación, el término medio de filtrado, Gaussiano, filtrando, y el filtrado bilateral. (Averaging, Gaussian, median filtering, and bilateral filtering) (Baggio D. L., 2015).

2.3.4. Segmentación de imágenes

Para el presente proyecto se toma muy en cuenta la segmentación de imágenes para un mayor y eficaz aislamiento de la información que se requiere bajo diversas técnicas de segmentación.

La segmentación es una parte fundamental cuando se trabaja con visión por computador, esta consiste en dividir una imagen digital, en partes o características iguales como la forma o color.

La segmentación de la imagen es el proceso de asignación de una etiqueta a cada píxel de la imagen de forma que los píxeles que compartan la misma etiqueta también tendrán ciertas características visuales similares. Los algoritmos de segmentación de imagen generalmente se basan en dos propiedades básicas de los niveles de gris de la imagen (Martin, 2004):

- **Discontinuidad:** Se intenta dividir la imagen basado en cambios bruscos en el nivel de gris.
- **Similitud:** Están basadas en las técnicas de umbrales, crecimiento de regiones, y técnicas de división y fusión

2.3.4.1. Segmentación basada en discontinuidades:

Se intenta dividir la imagen basándonos en los cambios bruscos en el nivel de gris. Las áreas de interés en esta categoría son la detección de puntos, de líneas y de bordes en la imagen. El método más común es realizar el producto de la máscara en relación a la imagen, resultados que se suma para obtener R, si este cumple con cierto umbral se habrá detectado cierta forma. La operación por discontinuidad viene dada por la siguiente formula:

$$R = \sum_{i=1}^9 W_i Z_i$$

Dónde: **W:** Es el coeficiente de la mascara

Z: Es el nivel de gris asociado al pixel de la imagen

TABLA 1. *Ejemplo de una mascara*

-1	-1	-1
-1	8	-1
-1	-1	-1

Fuente: (Martin, 2004)

- **Detección de puntos:** Está basada en la aplicación de una máscara centrada en el pixel a examinar. Si cumple cierto umbral T, el punto detectado.

$$|R| > T$$

- **Detección de puntos:** Se consideran las siguientes cuatro mascarar:

Cada respuesta R responderá a la dirección de los pesos positivos de cada mascarar.

TABLA 2. Mascarar para la segmentación por línea

-1	-1	-1	-1	-1	2	-1	2	-1	2	-1	-1
2	2	2	-1	2	-1	-1	2	-1	-1	2	-1
-1	-1	-1	2	-1	-1	-1	2	-1	-1	-1	2
Horizontal			45°			Vertical			-45°		

Fuente: (Martin, 2004)

2.3.4.2. Segmentación por similitud:

Agrupar los métodos que dividen una imagen comparando las propiedades de diferentes partes de la imagen que forman regiones, se muestran a continuación dichos métodos:

- **Umbralización:** Es una técnica de segmentación rápida que toma como partida el histograma de la imagen, en la que se trata de convertir una imagen a escala de grises o una imagen binaria (preferentemente binaria), de tal manera que los objetos de estudio se etiqueten con un valor distinto al del fondo.
- **Umbralización fija:** Se usa en imágenes en la que el contraste entre los diferentes objetos que se desean separar es grande, para esto se debe disponer de un valor fijo que sea el umbral, además de tener información, sobre los niveles de intensidad de los objetos a segmentar.

La imagen binaria resultante $B(i, j)$, se define a partir de la original $I(i, j)$, en función a un valor T, que es el umbral. La umbralización fija está dada por la siguiente ecuación:

$$B(I,J) = \begin{cases} 0, & I(i,j) < T \\ 1, & I(I,J) \geq T \end{cases}$$

- **Umbralización de banda:** Consiste en segmentar una imagen donde los objetos, contienen valores de gris dentro de un rango y el fondo tiene pixeles con valores en otro rango disjunto.

El histograma no toma en cuenta la información espacial si no solamente la distribución de grises, es por ello que algunas veces dos imágenes pueden tener el mismo histograma lo que hace que los métodos de segmentación basadas en umbrales sean limitados para problemas reales.

- **Segmentación basada en el crecimiento de Regiones:** Es un procedimiento que agrupa pixeles o sub regiones en regiones más grandes, comienza a partir de una semilla, esta crea regiones añadiendo pixeles de acuerdo a propiedades comunes entre los pixeles vecinos. (Martin, 2004)

2.3.5. Imágenes Digitales

La visión artificial o visión por computador, se fundamenta en el tratamiento de imágenes, que pueden fotos, o archivos de video, obtenidos a partir de cámaras web, cámaras de video digital, etc.

2.3.5.1. Formación y representación de la imagen

Una imagen es el resultado de la combinación de tres factores que intervienen en el proceso de captura de la imagen, color de luz, material de la superficie, sensibilidad de la cámara; Las cámaras tienen tres sensores que son sensibles a la longitud de onda de la luz, esta corresponde a los colores rojo, verde, azul.

La imagen es una materia de puntos los que conocemos como pixeles, cada valor del pixel es la combinación del canal verde, rojo y azul (RGB). Cada pixel tiene tres valores numéricos que generalmente viene en el rango de 0- 255, en donde 0 significa la ausencia de color y el 255 la máxima representación de ese color

La imagen de intensidad también se puede ver como una función bidimensional $f(x, y)$, en donde tendremos una coordenada en el eje de la X , y otra en el eje de la Y , y un valor de la función de intensidad.

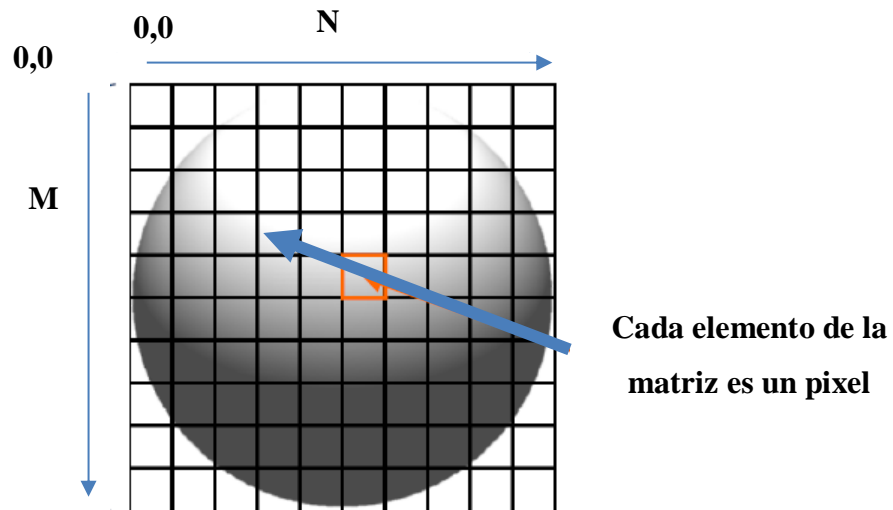


FIGURA 5. Representación de una imagen

Fuente: Elaboración propia

2.3.5.2. El color

Es un fenómeno perceptual relacionado con la respuesta humana a diferentes longitudes del espectro visible. Esto se debe a que existen tres sensores en el ojo que tiene una respuesta relativa diferente de acuerdo a la longitud de onda. Las combinaciones de los tres sensores captan señales que dan la sensación de toda la gama de colores que percibimos.

Existen distintas formas de codificar u ordenar los colores a partir de componentes básicos, lo que se conoce como espacios de color.

- **Modelo RGB:** Se basa en los tres sensores humanos, considerando que todos los colores son una combinación de los colores básicos o primarios: R-> rojo, G-> verde, B-> azul. Estos se normalizan obteniendo:

$$R = R/(R+G+B); \quad G = G/(R+G+B); \quad B = B/(R+G+B)$$

- **Modelo HSV:** Por sus siglas en inglés (Hue, Saturation, Value – Matiz, Saturación, Valor), este modelo se representa mediante un cono invertido. A continuación, se detallan cada una de las siglas:

H (Matiz): Es un intervalo de Angulo entre 0 y 360 grados, cada uno de ellos representa un color. OpenCV realiza la conversión de valores de 0 – 180 grados ya que se trata de imágenes de 8 bits.

S (Saturación): Se define como la distancia al eje de brillo negro y blanco (o la cantidad de blanco dentro del nuevo color), en la cual los valores se encuentran entre 0 y 100%. OpenCV realiza la conversión en los valores de 0 al 255 %.

V (Valor): Definido como la altura en el eje blanco y negro (grado de luminosidad de un color), el posible valor va de 0 al 100% 0 siempre es negro, OpenCV realiza la conversión en los valores de 0 al 255 grados.

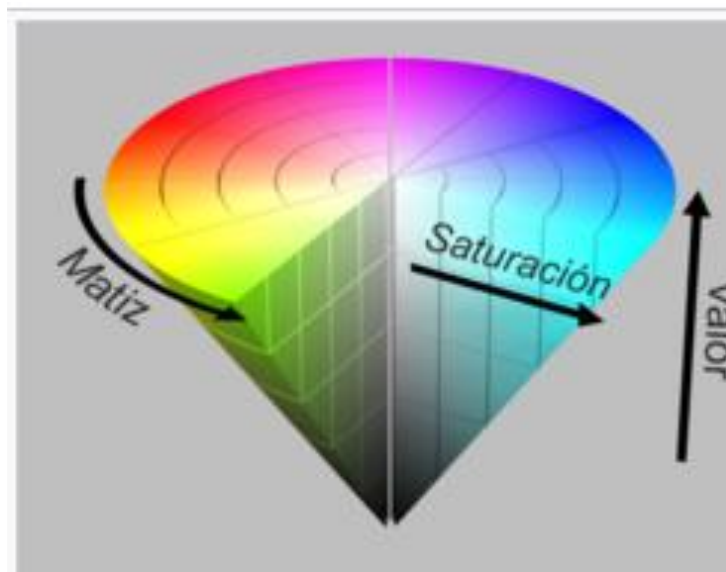


FIGURA 6 Representación gráfica del modelo HSV

Fuente: Wikipedia

2.3.6. Reconocimiento automático de placas vehiculares (ANPR)

Consiste en la obtención de imágenes clasificadas por un sistema o aplicación, para su posterior procesamiento, y detección automática de las placas vehiculares.

Este tipo de sistemas están dirigidos a ayudar a los conductores en su tarea de conducir con menos errores y eliminar algunos comportamientos de inseguridad cuando se conduce.

Durante los últimos años estos tipos de sistemas han sido muy útiles, para la obtención de placas de un vehículo en parqueos, semáforos, instituciones, etc. Además de poder asociarse con sensores de velocidad y de distancia.

Básicamente el proceso general de una ANPR se basa en los sistemas de reconocimiento de caracteres (OCR), con la única diferencia que acá tendrá que captar la imagen del vehículo en cuestión de segundos para luego procesar información. A continuación, se muestra el proceso general de un sistema ANPR:

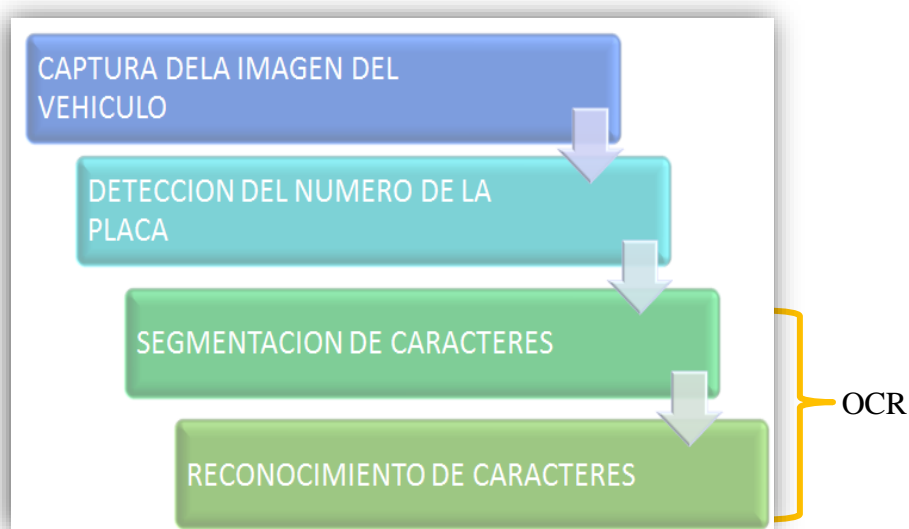


FIGURA 7 *Proceso general de un sistema ANPR*

Fuente: Elaboración Propia

2.3.7. Reconocimiento óptico de caracteres

Es un proceso dirigido a la digitalización de textos, los cuales identifican automáticamente a partir de imágenes símbolos o caracteres que pertenecen a un determinado alfabeto, para luego almacenarlo en forma de datos que las computadoras puedan reconocer, o de esa manera poder interactuar con algún editor de texto. De esta manera los OCR facilitan el

trabajo que tienen que realizar muchas personas, por lo tanto, brinda al ordenador la habilidad básica de poder leer.

Respecto de reconocimiento de caracteres y símbolos, existe una terminología de empleo extendido para referirse a cada una de las variantes de específicas de sistemas OCR, como se muestra a continuación.

- ICR.- (Inteligent carácter Recognition) se utiliza para el reconocimiento de caracteres de manuscritos.
- OCV.- (Optical Character Verification) hace referencia a la verificación de contenidos previamente conocidos.
- OMR.- (Optical Mark Recognition) designa funcionalidad de reconocimiento de marcas.



FIGURA 8. *Proceso general de un sistema OCR*

Fuente: Elaboración Propia

Los sistemas OCR tienen muchas aplicaciones y se han vuelto de vital importancia a la hora de poder digitalizar datos (caracteres), son muy usadas en documentos escaneados, reconocimiento de matrícula, etc.

2.3.7.1. Esquema básico de un algoritmo OCR. -

Los sistemas OCR se basan en cuatro etapas: binarización, segmentación, adelgazamiento y comparación de patrones.

TABLA 3 *Etapas básica del OCR*

ETAPA	CARACTERÍSTICA	TÉCNICA(S)
BINARIZACIÓN	Convertir una imagen de escalas de grises o color a una imagen en blanco y negro	<ul style="list-style-type: none"> • Histograma.
FRAGMENTACIÓN O SEGMENTACIÓN DE IMAGEN	Proceso más costoso, implica la detección mediante procedimientos de etiquetado de los contornos de imagen basándose en la intensidad. De tal forma que se eliminen componentes conexos de la imagen.	<ul style="list-style-type: none"> • Proyecciones lineales. • Modos o agrupamiento.
ADELGAZAMIENTO DE IMÁGENES	Consiste en ir borrando los puntos de contorno de forma que conserve su tipología	<ul style="list-style-type: none"> • Clasificación y reconocimiento.
COMPARACIÓN DE PATRONES	Comparan caracteres obtenidos con patrones almacenados en la base de datos.	<ul style="list-style-type: none"> • Métodos geométricos o estadísticos. • Métodos estructurales. • Métodos neuromiméticos. • Métodos markovianos (modelo oculto de Márkov). • Métodos de Zadeh.

Fuente: Elaboración propia

2.3.7.2. Inconvenientes con los OCR:

- Los dispositivos encargados de capturar la imagen, pueden introducir ruidos, afectando a los pixeles que serán procesados.
- La distancia que existen entre uno u otro carácter pueden no ser la misma, esto podría ocasionar errores.

2.3.8. Clasificador de imágenes:

Existen limitados clasificadores de imágenes en la actualidad que sean capaz de poder reconocer características puntuales dentro de una imagen. Para el presente proyecto nos apoyamos en el algoritmo de clasificación de imágenes centrado en el algoritmo de **Viola-Jones**.

2.3.8.1. Viola-Jones:

Es un algoritmo que permite la detección de rostros (inicialmente), en tiempo real. Propuesta por Paul Viola y Michael Jones en el año 2001.

Es un gran avance para la visión por computador, que supone la detección de objetos por su rapidez y además la clasificación se lo realiza por características y no así pixel por pixel. Dentro de las características más importantes de este algoritmo están que el porcentaje de detección positiva es muy alta, su objetivo principal es clasificar un objetivo en cuestión de imágenes que no son. Por lo tanto, para que la clasificación sea más óptima tendremos que darle un número significativo de imágenes positiva, así como también imágenes negativas.

La librería de OpenCV implementa este algoritmo mediante programas que nos proporciona (Create_sample.exe, opencv_traincascade.exe). (Baggio D. L., 2015)

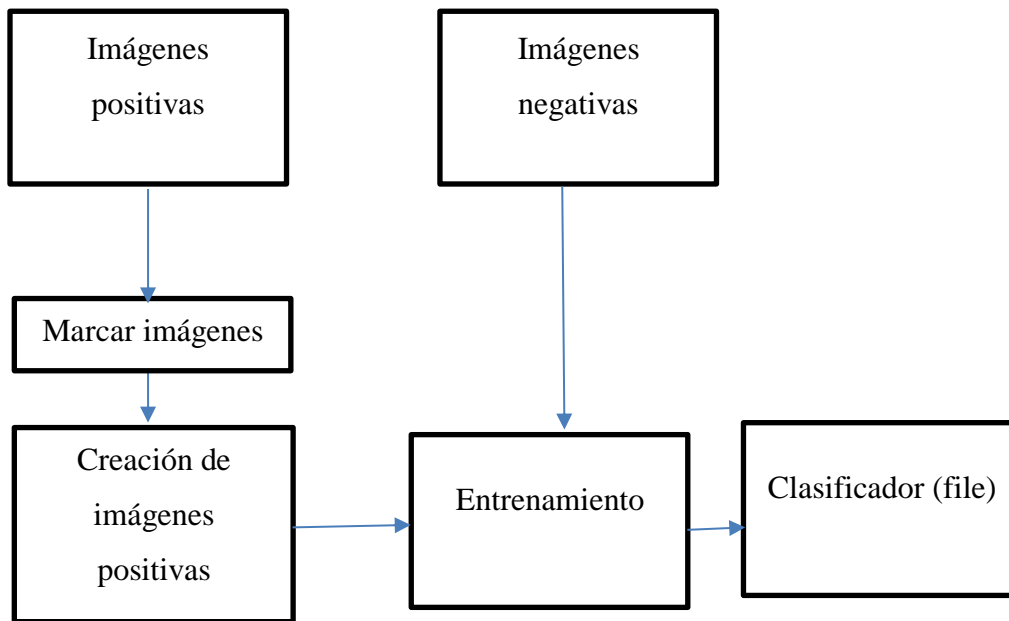


FIGURA 9. *Pasos para crear clasificador*

Fuente: Elaboración Propia

2.4. PRUEBAS DE CALIDAD DE SOFTWARE

Las pruebas son un elemento crítico para la calidad del software. La importancia de los costos asociados a los errores, promueve la definición y la aplicación de un proceso de pruebas minuciosas y bien planificadas. Estas permiten validar y verificar el software, entendiendo como validación al proceso externo del equipo de desarrollo que determina si el software satisface los requisitos, y verificación como el proceso interno que determina si los productos de una fase determinan las condiciones de dicha fase. (Pressman, 2010)

Para la presente aplicación se utilizará las pruebas de caja negra, también llamados prueba de comportamiento.

2.4.1. Pruebas de caja negra

Se enfocan en los requerimientos funcionales del software, es decir las técnicas de caja negra le permiten derivar conjuntos de condicionales de entrada que revisaran por completo todos los requerimientos funcionales de un programa.

2.4.2. ISO/IEC 9126

El estándar ISO 9126 se desarrolló con la intención de identificar los atributos clave del software. Este sistema identifica seis atributos clave de la calidad: (Pressman, 2010)

Funcionalidad. Grado en el que el software satisface las necesidades planteadas según las establecen los atributos siguientes: adaptabilidad, exactitud, interoperabilidad, cumplimiento y seguridad.

Confiabilidad. Cantidad de tiempo que el software se encuentra disponible para su uso, según lo indican los siguientes atributos: madurez, tolerancia a fallas y recuperación.

Usabilidad. De facilidad de usar, según lo indican los siguientes subatributos: entendible, fácil de aprender y operable.

Facilidad de recibir mantenimiento Facilidad con la que pueden efectuarse reparaciones al software, según lo indican los atributos que siguen: analizable, cambiable, estable y susceptible de someterse a pruebas.

Eficiencia. Grado en que el software emplea óptimamente los recursos del sistema, según lo indican los subatributos siguientes: comportamiento del tiempo y recursos.

Portabilidad. Facilidad con la que el software puede llevarse de un ambiente a otro según lo indican los siguientes atributos: adaptable, instalable, conformidad y sustituible.

2.5. METODOLOGÍA Y HERRAMIENTAS

En esta sección se mostrará en detalle todos los fundamentos teóricos que proporciona la metodología Xp, fases y sus herramientas (tarjetas CRC, tarjetas de usuario).

2.5.1. Programación Extrema(XP):

Para el desarrollo del presente proyecto se utilizó la metodología XP, que utiliza el proceso ágil como base fundamental para el desarrollo incremental de aplicaciones, que se basa en la simplicidad, la comunicación y la retroalimentación del código desarrollado. Es de gran utilidad en proyectos pequeños y con requerimientos altamente cambiantes. Creado por Kent Beck en 1996, cuando trabajaba Chrysler Corporation. Beck define cinco valores primordiales para la aplicación de la metodología:

- Se pone alto énfasis en la colaboración estrecha pero informal, entre el cliente y el desarrollador.
- Pone más énfasis en la adaptabilidad que en la previsibilidad, además de tener en cuenta la simplicidad de desarrollar solo el sistema que realmente se necesita en ese momento.
- Para poder alcanzar la simplicidad XP restringe a los desarrolladores para que diseñen solo las necesidades inmediatas, objetivo crear un diseño sencillo que implemente con facilidad en forma de código.
- La retroalimentación parte fundamental, esta se obtiene del software implementado, el usuario, y parte del equipo de desarrollo, el software por medio de resultados de pruebas da retroalimentación al equipo ágil. XP usa la prueba unitaria como su táctica principal de pruebas.
- Un equipo XP tiene que tener disciplina valentía, y tener en cuenta que en un futuro podría haber cambios bruscos en el sistema. Al apegarse a estos valores inculca respeto en cada uno de los desarrolladores e indirectamente en el software. (Pressman, 2010)

2.5.2. Objetivos y Características:

Entre los principales objetivos de la metodología XP se encuentran: la satisfacción del cliente, potenciar el trabajo en grupo, minimizar el riesgo actuando sobre variables del proyecto: costo, tiempo, calidad, alcance.

Se basa en prueba y error para obtener un software que funcione realmente, se fundamenta en principios, está orientada hacia quien produce, el cliente participa activamente.

2.5.3. Algunas prácticas de XP:

En XP existen prácticas que pueden llegar a dar mejores resultados, dentro de un proyecto le mostramos algunas de ella.

- **Diseño simple:** XP denota esta práctica para facilitar el mantenimiento y minimizar riesgos de modificaciones que se hagan sin entender el código. Define también que un diseño simple no tiene que entender código redundante, ni duplicado, supera todos los test de funcionalidad e integridad y aceptación, no utilizas sintaxis compleja, es decir que es clara la intención de cada línea de código, contiene el menor número de clases y métodos.
- **Refactorización:** Se refiere a la modificación del código para dejarlo en buen estado, escribirla de nuevo con el fin de que haya orden independientemente de los cambios que se haga. Así que el código final se debe ver claro y sencillo, describir comentarios.
- **Test:** Es una de las prácticas más importantes dentro de la metodología ya que le da flexibilidad, los test han de ser aplicados tras cada cambio y tiene que ser automatizado. El objetivo de los test no es detectar errores si no prevenirlos.

Los test siempre se escriben antes que el código a testear es por esto que estamos obligados a pensar los posibles problemas que podrían encontrarse cuando se empiece a usar el código. El usuario debe ayudar al programador a desarrollar de forma conjunta los test.

Existen tres tipos de test: **test de aceptación** creada juntamente con el usuario final y esta debe responder a las necesidades del cliente. **Test Unitario** es creado por el programador para ver qué métodos de la clase funciona correctamente. **Test de integridad** Es creado por el grupo de desarrollo para probar en conjunto la nueva modificación.

- **Estándares de codificación:** Todo grupo de trabajo debe tener normas de codificación, esto para entender con mayor facilidad el código desarrollado y además de facilitar las modificaciones y refactorización. Para el caso de que se trabaje con java se utiliza “Code Conventions for the Java Programming Language”.
- **Metáfora de negocio:** Antes y durante el trabajo, los programadores y el cliente deben tener un vocabulario común en donde los dos puedan entender los que se quiere para eso se debe crear una metáfora de negocio y esta le sirva al desarrollador para poder modelar clases y métodos. La metáfora es una historia común compartida por el usuario y el equipo que describe como debe comportarse el sistema.
- **Cliente in situ:** El cliente siempre tiene que formar parte del desarrollo para poder solventar las dudas en ese mismo momento.
- **Planificación incremental:** Esta debe estar condicionada por las necesidades del cliente, no puede ser estática, tiene que estar apta para cambios drásticos. (Pressman, 2010)

2.5.4. Fases de la programación extrema (XP)

La programación extrema usa como paradigma de programación un enfoque orientado a objetos. Es ciclo de vida de un proyecto XP está dividido en seis fases.

2.5.4.1. Exploración: La exploración se desarrolla en tres procesos:

1. Las historias de usuario
2. El spike arquitectónico
3. La metáfora del negocio

Las historias de usuario: como el mismo nombre lo son funcionalidades que se muestran a grandes rasgos por parte del cliente, son parecidos a los casos de usos, la diferencia radica que esta las escribe el usuario y los casos de usos los hacen los analistas, estas deben ser cortas, entendibles y sin terminología técnica.

Spike arquitectónico: El equipo empieza a familiarizarse con la metodología lenguaje y codificación que se usara dentro del proyecto.

Metáfora de Negocio: Es una historia común entre el desarrollador y el cliente, esta debe servir para que el cliente pueda entender los diferentes términos.

A medida que avanza el proyecto el cliente puede agregar historias, cambiar el valor existente, descomponerlos o eliminarlos, es ahí donde el equipo considera las entregas faltantes y modifica los planes.

2.5.4.2. Planificación:

La consecuencia de la primera fase ha de ser una planeación seguidos de los siguientes pasos:

- El cliente entrega al equipo de desarrollo las historias de usuarios, que el desarrollo sin términos técnicos priorizando de mayor a menor importancia.
- Una vez se tiene las historias priorizadas, se pasa a ver los planes de entregas.
- Estos planes de entregas se componen por iteraciones, en el que se especifica las funcionalidades. A cada iteración se le asigna un tiempo y se determina el alcance del proyecto.
- El equipo estudia y estima el coste, si se considera que es demasiado compleja, el usuario debe descomponerlas en partes más sencillas.
- Así también el equipo agrega las pruebas funcionales en cada una de las historias de usuarios.

2.5.4.3. Las iteraciones

Esta fase se repetirá tantas veces como iteraciones tengamos, se detallan las tareas a realizar por cada historia de usuario, estas tareas deben de ser de dos a tres días, se priorizan las tareas que más valor tengan, intentando que se finalicen las historias de usuarios.

El diseño del sistema tiene que ser sencillo como lo plantea Xp, este sigue rigurosamente los pasos y consideraciones que se tomó en cuenta en la planeación.

Xp aconseja el uso de tarjetas CRC, estas son el único producto que se hace en el proceso de diseño.

Así también se agrega los casos de uso para cada una de las historias de usuarios y así tener una mayor comprensión de la aplicación.

2.5.4.4. La fase de producción

En esta fase se llega a alcanzar la primera versión que el usuario final decida poner en producción, debe ser productiva si y solo sí, se alcancen las funcionalidades mínimas que aporten al sistema y además que esté estable.

Durante la fase de producción, el ritmo de desarrollo decae debido a que el equipo debe solventar las incidencias de los usuarios. Es por esto que a veces es necesario incorporar nuevo personal al equipo. Para el caso del proyecto cada vez que la prueba funcional sea positiva se denotará “**historia de usuario a producción**”.

2.5.4.5. La fase de mantenimiento

Una vez el alcance del proyecto se ha conseguido, y tenemos todas las funcionalidades en producción, se revisan con el usuario aquellas nuevas historias de usuario que se han producido tras la puesta en producción del proyecto. Estas nuevas funcionalidades se van incorporando según su valor de negocio y el presupuesto adicional del que se disponga.

2.5.4.6. La fase de muerte del proyecto

Cuando no existen más historias de usuario para introducir en nuestro sistema o cuando se reduce progresivamente valor de las historias de usuario implementadas en él, el proyecto entra en la fase de muerte. Se irá desinvirtiendo en él hasta abandonarlo totalmente cuando no aporte valor al negocio o cuando sus historias de usuario hayan sido absorbidas por otro sistema de información. (Gonzales, 2017)

TABLA 4 *Fases de la metodología XP*

ACTIVIDAD	CARACTERÍSTICA	HERRAMIENTAS
Exploración	El cliente realiza las historias de usuarios, spike arquitectónico, metáforas de negocio	Historias de usuario, búsqueda de definiciones.
Planificación	El cliente entrega las historias de usuarios, se priorizan, y se desarrolla un plan de entrega.	
Iteraciones	Diseño del software. Desarrollo y prueba del software, asignando características, duración, importancia, desarrollador, etc. a las iteraciones.	<ul style="list-style-type: none"> • Lenguaje de programación • Tarjetas CRC,
Producción	Lanzamiento de parte del sistema producido en la iteración	
Mantenimiento	Una vez terminado el proyecto en su totalidad se procede al mantenimiento respectivo.	
Muerte	Absorción total de las historias de usuario, entregados por el cliente.	

Fuente: Elaboración Propia

2.5.5. Herramientas de la metodología XP

- **HISTORIAS DE USUARIOS:**

Como herramienta de identificación de los requerimientos, que tiene el cliente entorno al desarrollo del software.

TABLA 5. *Historia de Usuario*

Numero: permite identificar a una historia de usuario	Usuario: persona que utilizara la funcionalidad del sistema.
Nombre de la Historia: Describe de manera general a la historia de usuario	
Prioridad: El grado de importancia que tiene el cliente	Riesgo en el desarrollo: Valor de la complejidad que una historia de usuario representa al equipo de desarrollo.
Puntos estimados: Las semanas que se tardara en hacer la historia de usuario	Iteración asignada: Numero de iteración en la que el cliente desea que se tome en cuenta la historia de usuario.
Programador responsable: Persona encargada de programar la historia.	
Descripción: Información detallada de la historia de usuario.	
Observaciones: Campo opcional	

Fuente: Elaboración Propia

- **TARJETAS CRC (CLASE-RESPONSABILIDAD- COLABORADORES):**

Herramienta de diseño donde se hace referencia, al nombre, sus atributos y colaboradores, de las clases que serán parte del software.

TABLA 6. *Tarjeta CRC*

NOMBRE DE LA CLASE: Nombre al cual hace referencia la tarjeta	
Responsabilidades: Atributo y operaciones de la clase	Colaboradores: Clases que se relacionan con la clase citada.

Fuente: Elaboración Propia

- **PRUEBAS DE ACEPTACIÓN:**

Herramienta de prueba de cada una de las historias de usuario donde se hace referencia, a los pasos de ejecución, condiciones de ejecución, resultados esperados, y su evaluación respectiva.

TABLA 7. *Modelo de prueba de aceptación*

Código: Numero único que permite identificar la prueba de aceptación	Número de historia: Numero único que identifica la historia de usuario.
Historia de usuario: Nombre de la historia de usuario	
Condiciones de ejecución: Condiciones que se deben tomar en cuenta para realizar la prueba de aceptación.	
Entrada/Pasos de ejecución: Pasos que siguen los usuarios para verificar la funcionalidad.	
Resultado esperado: Respuesta del sistema que el cliente espera.	
Evaluación de la prueba: Nivel de satisfacción que el cliente espera (Aprobada/no aprobada).	

Fuente: Elaboración Propia

2.5.6. Roles de la metodología XP.

Los roles son una parte importante dentro de la aplicación de la metodología XP, para el desarrollo de un proyecto; a continuación, se muestran:

- **Programador:** Produce el código del sistema y realiza las pruebas unitarias.
- **Cliente:** Escribe la historia de usuario y las pruebas funcionales para validar implementación.
- **Entrenador (Coach):** Es responsable de todo el proceso global, experto en la metodología (XP), además de que es el vigilante de que el proceso se llevó tal cual lo dice el manifiesto.

- **Encargado de pruebas (Tester):** Ayuda al cliente a escribir las pruebas funcionales. Ejecuta, difunde los resultados y es el responsable de las herramientas de soporte para pruebas.
- **Encargado de Seguimiento:** Proporciona realimentación al equipo en el proceso. Verifica el grado de acierto entre las estimaciones realizadas y el tiempo real dedicado, comunicando los resultados para mejorar futuras estimaciones. Determina cuándo es necesario realizar algún cambio para lograr los objetivos de cada iteración.
- **Consultor:** Guía al equipo dentro del proyecto para un fin específico (miembro externo).
- **Gestor:** Es la relación directa entre el equipo y el cliente, (kent beck Cinthia Andres, 2012)

2.6. HERRAMIENTAS UTILIZADAS

En esta sección explicaremos las herramientas seleccionadas por la simplicidad y accesibilidad que proporcionan a la hora de desarrollar el presente proyecto.

2.6.1. Librerías de procesamiento de imagen:

Existen una gran cantidad de librerías de código abierto para el procesamiento de imagen y extracción de caracteres u OCR, para el presente proyecto se utilizó OpenCV:

2.6.1.1. OpenCV

Es una librería software open source de visión artificial, tratamiento de imágenes, y reconocimiento de objetos, destinada principalmente a aplicaciones de visión por computador en tiempo real. (Raul Igual, 2008)

Tiene una licencia **BSD** (Distribución de Software de Berkeley) y por lo tanto está libre para ambos uso académico y comercial, lo que permite modificar y utilizar el código, tiene una comunidad de más de 47000 personas y más de 7 millones de descarga. La librería tiene más

de 5000 algoritmos de visión artificial y machine learning para utilizar, estos permiten identificar objetos, caras, clasificar acciones humanas, hacer tracking de movimiento de los objetos, extraer modelos 3D, encontrar imágenes similares, eliminar ojos rojos, seguir movimiento de los ojos y reconocer escenarios. OpenCV, está escrito en C++, y tiene versiones para Python, java, Matlab, funciona en Windows, Linux, Android y MacOS.

Fue diseñado para la eficiencia computacional y con un foco fuerte en aplicaciones de tiempo real. En el caso del proyecto usaremos la librería de OpenCV versión 3.0 donde existen distintas cascadas (archivos que ya reconocen determinados objetos figuras), para el lenguaje de programación Java. Las funcionalidades de OpenCV se clasifican por módulos las que se cita a continuación:

- a) **Core:** Estructura básica y algoritmos.
- b) **ImgProc:** Esta función para los algoritmos de procesamiento de la imagen (Filtrado, transformación geométrica, histogramas y binarización).
- c) **Video:** Análisis de los videos.
- d) **Imgcodecs:** Operaciones básicas para leer y guardar una imagen

OpenCV tiene su propia estructura de datos que se cita a continuación:

- a) **Mat:** Es un tipo de dato compatibles para asignarle una imagen, como una matriz en donde se resguardan los datos grafico (pixeles) de la imagen.
- b) **Point:** Tipo de dato en 2D, definido por las coordenadas (x,y), Point punto = new Point(2, 3);
- c) **Size:** Estructura de tamaño en 2 D, especifica el tamaño (ancho y alto) de una imagen o un rectángulo.
- d) **Rect:** Objeto rectángulo en 2D, especifica datos un determinado punto (x, y, ancho alto).

Para realizar dibujos sobre la imagen o video OpenCV tiene las siguientes métodos o funciones:

- a) **Imgproc.circle:** Dibuja un circulo con parámetros de entrada (centro, radio, imagen).

b) **Imgproc.Line:** Dibuja líneas, con dos puntos de coordenadas; entre otros dibujos que realiza la elipse, arco y rectángulo.

- **Representación digital de una imagen en OpenCV**

Para obtener una imagen, es necesario que algún tipo de sensor la proporcione, OpenCV la digitaliza la escena real a una imagen integrada por una matriz de $M \times N$ puntos, donde cada uno de ellos representan el nivel de gris de cada punto de la imagen, o los valores correspondientes al modelo de colores RGB. Al brillo de cada píxel se conoce como nivel de gris, los valores de estos varían entre 0 a 255, donde los extremos son el blanco y el negro.

En OpenCV, una imagen es una matriz de datos, es decir, un conjunto ordenado de elementos reales. Las imágenes se almacenan como matrices, donde cada elemento se corresponde con un píxel. Para que OpenCV, pueda desarrollar sus diferentes algoritmos es necesario utilizar la función Mat (transforma imagen a una matriz).

Existen tres tipos básicos de imágenes en OpenCV:

- **Imagen Binaria:** Existen dos tipos de valores que corresponde al blanco y negro. Este tipo se almacena en una matriz bidimensional de 0 (negro) y 1 (blanco), cada píxel corresponde a un píxel.
- **Imagen de Intensidad:** También llamada en escala de grises. Se refiere a una matriz cuyos valores representan intensidades dentro de un mismo rango. La matriz puede ser double, en cuyo caso contiene los valores en el intervalo [0, 1], o de tipo unit8, en cuyo caso el rango es [0, 255]. Los elementos de la matriz representan diferentes intensidades o niveles de gris, donde el 0 representa el negro y el 255 el blanco. Cada píxel se corresponde con 1 byte, de ahí los 255 niveles permitidos.
- **Imagen de Color:** Representada en formato RGB, cada color está representada por tres canales, misma denota intensidad del rojo, verde y azul, cada píxel se codifica en 3 bytes.

2.6.1.2. Teserract

Es un motor de Reconocimiento Óptico de Caracteres (OCR) open source, desarrollado por HP en los años 1984 y 1994. Nació como una tesis doctoral. En 2005 HP libero la librería como código abierto, por lo que hoy en día se constituye como una de las principales opciones, usadas para distintos servicio y aplicaciones. En el 2014 Google (Actual dueño), libera la versión **TESERRACT-OCR, 2014, para** utilizarlas con funciones nativas de java.

Se muestra a continuación código fuente de la instancia y proceso de extracción de caracteres por Teserract OCR, a partir de una imagen digital previamente analizada, procesada y mejorada:

```
File imageFile = new File("src/imagenes/formal.png");  
  
try { String result = instance.doOCR(imageFile);  
  
placaString = result;  
  
System.out.println("caracter " + result);  
  
    } catch (TesseractException e) {  
  
        System.err.println(e.getMessage()); } 
```

La figura que se muestra a continuación representa el funcionamiento o pasos que sigue la librería para lograr convertir una imagen a texto. En el primer paso se tiene como de dato de entrada una imagen preferentemente binarizadas, segmentado la imagen, en el siguiente paso la imagen es analizada delimitando líneas en torno al texto, luego el texto es delimitado se analiza y se rompe o divide en palabras de acuerdo al espaciado de las letras, luego las letras son analizadas por el clasificador de idiomas una a la vez si es reconocida, es clasificada y dada a entrenamiento.

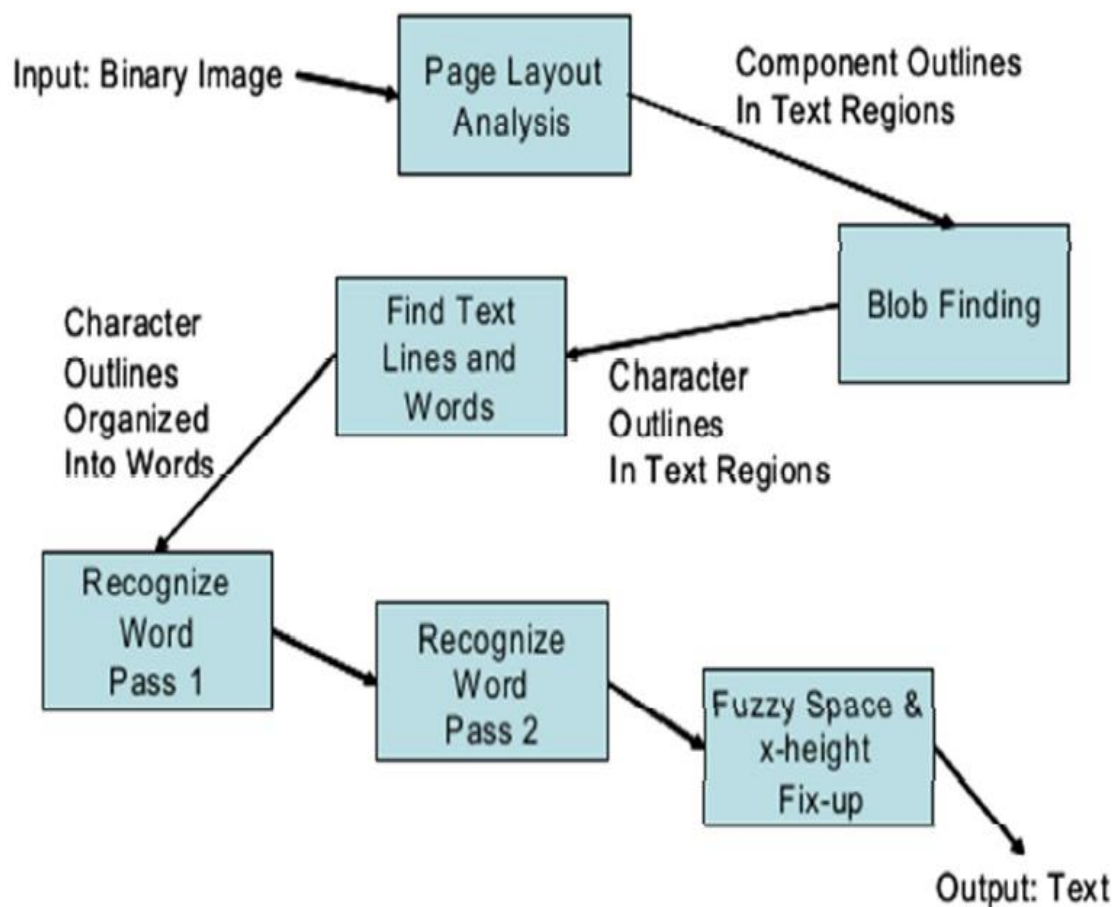


FIGURA 10. *Componentes de Tesseract*

Fuente: Tesseract-OCR, 2014

2.6.2. Java

Lenguaje de programación orientado a objetos desarrollado por MicroSystems que nos permite comunicar a la computadora a hacer tareas específicas. Entre las principales características de este lenguaje se encuentran: Encapsulación, herencia, polimorfismo, fuertemente tipado; acompañado de un conjunto de librerías (packajes, soporte para concurrencia, multihilo, conjunto de clases).

```

] private void modificarImagenAutoInfractor(int i) throws SQLException, ClassNotFoundException {

    Imgproc.resize(autoMat, autoMat, new Size(245, 170));
    deteccionPlaca dect = new deteccionPlaca();
    if (dect.procesoPlaca(autoMat).height() > 0) {
        Imgcodecs.imwrite("src/imagenes/temp.jpg", autoMat);
        reconocimientoOpticoCaracteres ocr1 = new reconocimientoOpticoCaracteres();
        ocr1.tratarImagen();

        System.err.println(" numero de palca " + num_placa);
    }
    if (num_placa == "") {
    }
    if (i == 0) {
        Mat carrolMat = autoMat.clone();
        Image carrolImg = imageProcessor.toBufferedImage(carrolMat);
        carrol.setIcon(new ImageIcon(carrolImg));
    }
}

```

FIGURA 11 Ejemplo de código java

Fuente: Elaboración Propia

2.6.3. NetBeans 7.3

Es un entorno de desarrollo - una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

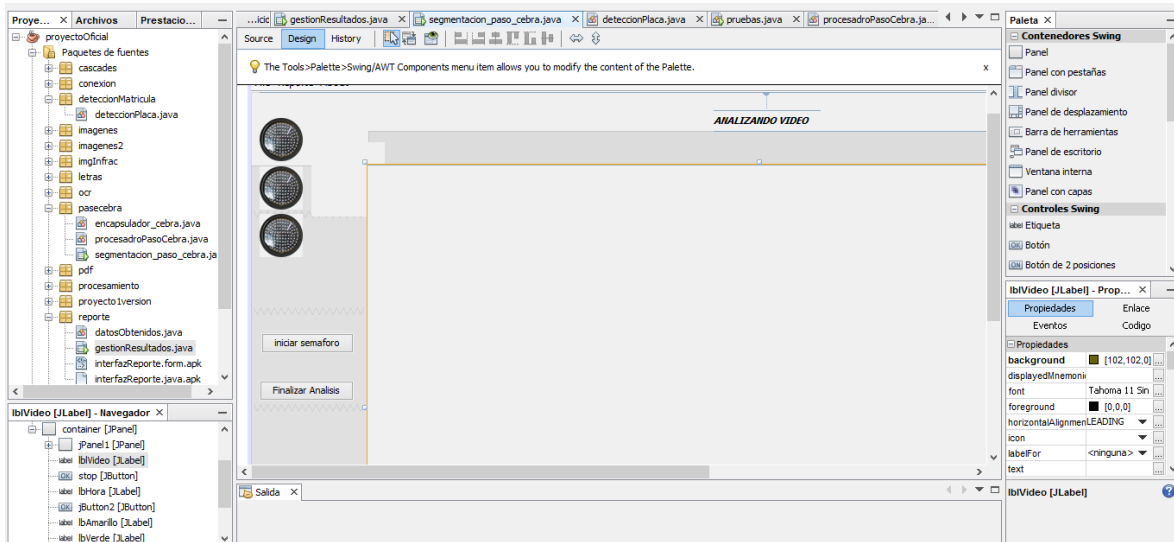


FIGURA 12 Interfaz de desarrollo de NetBeans

Fuente: Elaboración propia

CAPITULO III

3.MARCO APLICATIVO

3.1. DESARROLLO DEL PROYECTO

El presente capítulo describe el ciclo de vida del desarrollo de la aplicación de detección de infracción de tránsito y reconocimiento de placas vehiculares a través de la metodología XP, la misma se describe en detalle en el capítulo anterior, se emplean herramientas como el UML, con el objeto de poder desarrollar los diagramas correspondientes a la aplicación.

Se muestra a continuación el diagrama de flujo general de la aplicación, mostrando paso a paso las etapas para desarrollar sus objetivos.

El primer paso se basa en la obtención de una imagen digitalizada, para luego marcar la ubicación del paso de cebra, a continuación la aplicación procede a la detección de la infracción en el paso de cebra, si no reconoce la placa procede nuevamente a obtener una imagen, sin embargo sí reconoce una placa procede con el tratamiento de la imagen (ampliación, mejoramiento, etc), aplica Tesseract OCR, luego extrae y registra el número de la placa.

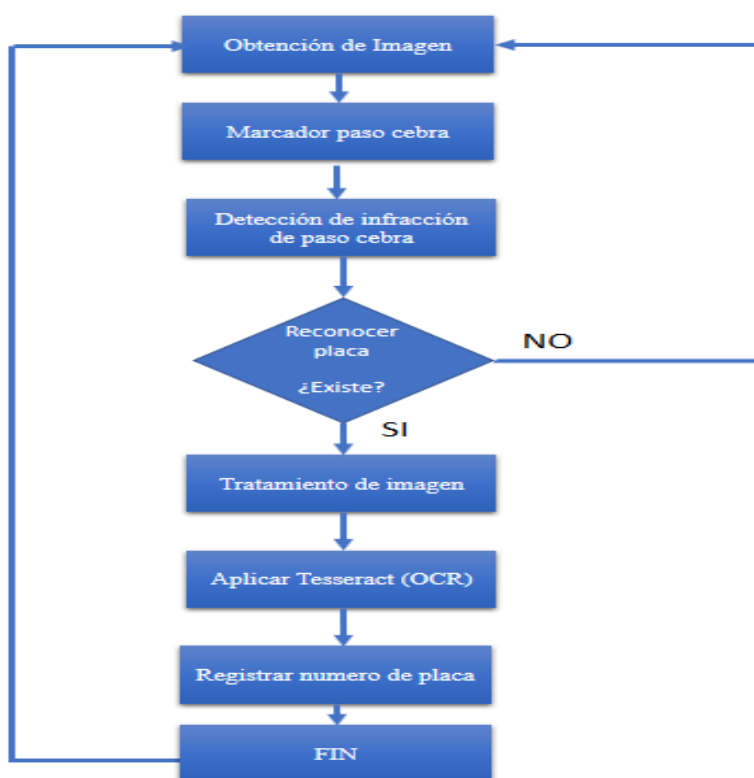


FIGURA 13 Diagrama de flujo de la aplicación

Fuente: Elaboración Propia

Proceso general de la aplicación

La aplicación cuenta con un proceso general parecido a una Detección Automática de Matriculas (ANPR) sistemas de detección de matrículas, pero menos sofisticados, ya que los mismos cuentan con un circuito cerrado, para el presente proyecto lo que se pretende es mostrar la aplicabilidad de la visión por computador en cuanto a:

1. La obtención de las imágenes mediante un dispositivo que capture las imágenes.
2. Pasar las imágenes obtenidas al ordenador mediante un flash o activar alguna cámara conectada al ordenador y procesar a tiempo real.
3. Una vez obtenida, el administrador marca los puntos del paso de cebra y continúa, la aplicación prosigue desarrollando su trabajo en cuanto a la segmentación del paso de cebra, reconocer la placa, obtener y registrar los caracteres de la placa obtenida.
4. El administrador podrá hacer consultas a la base de datos de los infractores.

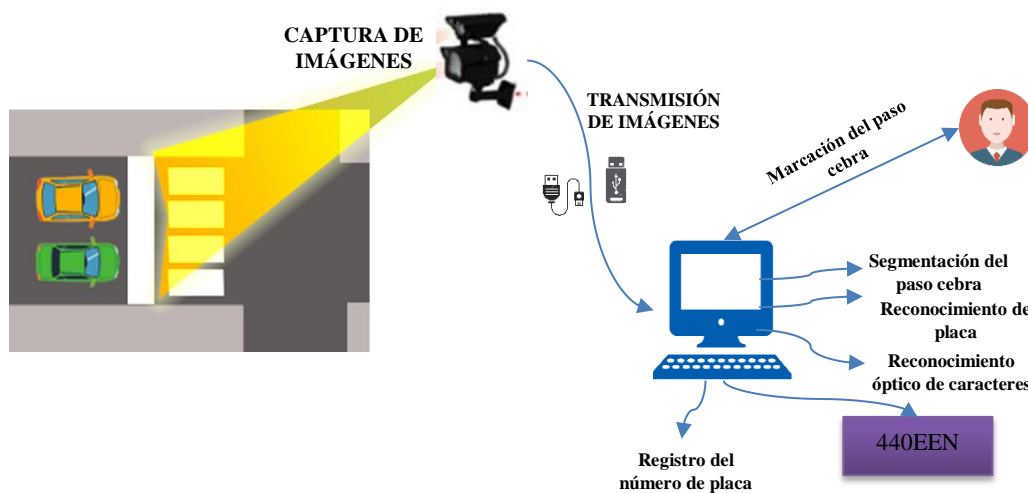


FIGURA 14. *Proceso general de la aplicación*

Fuente: Elaboración Propia

3.2. EXPLORACIÓN Y PLANIFICACIÓN DEL PROYECTO EN BASE A LA METODOLOGÍA XP

- Recabar información sobre los requisitos funcionales del software, de parte de los clientes al cual se les hace el trabajo, para esto el cliente nos brinda la siguiente información llamadas historias de usuarios, en donde se detalla lo que se quiere hacer sin términos técnicos, se identifican la complejidad de historias de usuario, de ser necesario se las convierten en dos.
- Realizar la exploración donde se identifica cada uno de las herramientas a utilizar en el desarrollo de la aplicación (Spike tecnológico), realizar una metáfora de negocio, además de poder identificar los usuarios. (Gonzales, 2017)
- Enfatizar en unos planes de entregas, de acuerdo a la priorización del cliente, definidos y guiados por la metodología XP.

3.2.1. Historias de usuario:

Para la presente aplicación se obtienen los siguientes requerimientos descritas por el cliente que se plasmaran en historias de usuarios (sin tecnicismo).

Historia de usuario 1: **Detectar la placa del vehículo infractor**

Nos gustaría que se pudiera detectar la placa del vehículo que se pasa en el semáforo rojo, o pase el paso peatonal.

Historia de usuario 2: **Respaldo de fotografía**

Que se pudiera tener la fotografía de la placa y del vehículo infractor, para que cuando haya algún reclamo de parte del conductor tengamos nuestro respaldo.

Historia de usuario 3: **Reporte de infractores**

Que se pudiera tener la lista de los infractores, dichas lista pudieran salir cada cierto intervalo de tiempo, algo así como una estadística.

Historia de usuario 4: **Copias de toda la información**

Por si algo llega a pasar con algunas de las maquinas (Quema de disco u otros), pueda haber respaldo de toda la información.

3.2.2. **Priorización de las historias de usuarios**

Debido a la complejidad que se vio en esta historia de usuario N 2, el equipo de desarrollo tomo la decisión de subdividirla, ya que es un tema de codificación más que de requerimiento, Y son las siguientes:

3.2.2.1. **Historia de usuario 1: Entrenamiento del clasificador**

A continuación, se muestra las observaciones correspondientes para describir iteraciones, su prioridad, su tiempo estimado y la prueba funcional de la historia de usuario.

TABLA 8. *Historia de usuario N. 1*

Numero: 1	Usuario: Administrador
Nombre de la Historia:	Entrenamiento del clasificador
Prioridad: Alta	Riesgo en el desarrollo: Alta
Puntos estimados: 4	Iteración asignada: 1
Programador responsable:	Ruddy Chao
Descripción:	Se procederá a la clasificación de imágenes negativas y positivas y la obtención de un archivo *.XML el mismo contendrá una serie de instrucciones que permitirán la detección de la placa.
Prueba funcional:	La simple obtención del archivo XML.
Observaciones:	Se procederá a la prueba del archivo cuando se desarrolle parte de la aplicación que permita obtener un archivo de video y probar la eficacia del clasificador. Para este caso se obtendrá un clasificador que detecte placas nacionales.

Fuente: Elaboración propia

3.2.2.2. Historia de usuario 2: Detector de placas

A continuación, se muestra las observaciones correspondientes para describir iteraciones, su prioridad, su tiempo estimado, la descripción y la prueba funcional de la historia de usuario.

TABLA 9. *Historia de usuario N. 2*

Numero: 2	Usuario: Administrador.
Nombre de la Historia:	Detector de placas
Prioridad: Alta	Riesgo en el desarrollo: Alta
Puntos estimados: 4	Iteración asignada: 1
Programador responsable:	Ruddy Chao
Descripción:	Tendrá la funcionalidad de poder clasificar e identificar las placas de los vehículos
Prueba funcional:	Los procedimientos son satisfactorios cuya detección de placa sea lo más preciso posible.
Observaciones:	Cabe hacer notar que por la complejidad que con lleva, hacer la presente funcionalidad, el clasificador no podrá identificar al 100%

Fuente: Elaboración propia

3.2.2.3. Historia de usuario 3: Marcador paso cebra

A continuación, se muestra las observaciones correspondientes para describir iteraciones, su prioridad, su tiempo estimado, la descripción y la prueba funcional de la historia de usuario.

TABLA 10. *Historia de usuario N. 2*

Numero: 3	Usuario: Administrador.
-----------	--------------------------------

Nombre de la Historia:	Marcador paso Cebra
Prioridad: Alta	Riesgo en el desarrollo: Alta
Puntos estimados: 4	Iteración asignada: 2
Programador responsable:	Ruddy Chao
Descripción:	El administrador tendrá la responsabilidad de obtener el archivo de video o activar cámara además de marcar 4 puntos del paso de cebra.
Prueba funcional:	Los procedimientos son satisfactorios cuyo marcado del paso de cebra se satisfactorio.

Fuente: Elaboración Propia

3.2.2.4. Historia de usuario 4: Segmentación paso cebra

A continuación, se muestra las observaciones correspondientes para describir iteraciones, su prioridad, su tiempo estimado, la descripción y la prueba funcional de la historia de usuario.

TABLA 11. *Historia de usuario N. 4*

Numero: 4	Usuario: Administrador.
Nombre de la Historia:	Segmentación paso Cebra
Prioridad: Alta	Riesgo en el desarrollo: Alta
Puntos estimados: 4	Iteración asignada: 2
Programador responsable:	Ruddy Chao

Descripción:	Tendrá la funcionalidad de poder reconocer cuando el vehículo se encuentre encima o haya por un momento borrado la visibilidad de una de las partes del paso de cebra para tener una información exacta de cuál podría ser la placa infractora.
Prueba funcional:	Los procedimientos son satisfactorios cuya detección de pase de autos y conteo de autos infractores sea lo más preciso posible.
Observaciones:	Se tiene inconvenientes con los autos de color blanco ya que el paso peatonal es del mismo color

Fuente: Elaboración propia

3.2.2.5. Historia de usuario 5: OCR

A continuación, se muestra las observaciones correspondientes para describir iteraciones, su prioridad, su tiempo estimado, la descripción y la prueba funcional de la historia de usuario.

TABLA 12. *Historia de Usuario N. 5*

Numero: 5	Usuario: Administrador.
Nombre de la Historia:	Reconocimiento óptico de caracteres
Prioridad: Alta	Riesgo en el desarrollo: Alta
Puntos estimados: 4	Iteración asignada: 3
Programador responsable:	Ruddy Chao
Descripción:	Tendrá la funcionalidad de poder transformar en letras entendibles para la computadora a partir de una imagen en dos canales (Blanco y negro)

Prueba Funcional	Los procedimientos son satisfactorios cuya implementación funcional del OCR basado en Tesseract sea satisfactorio
Observaciones:	Para el desarrollo de esta historia de usuario se utiliza la librería de reconocimiento de caracteres Tesseract.

Fuente: Elaboración propia

3.2.2.6. Historia de usuario 6: Respaldo de las infracciones (pdf)

A continuación, se muestra las observaciones correspondientes para describir iteraciones, su prioridad, su tiempo estimado, la descripción y la prueba funcional de la historia de usuario.

TABLA 13. *Historia de usuario N. 6*

Numero: 6

Usuario: Administrador.

Nombre de la Historia:	Respaldo de las infracciones (pdf)
Prioridad: Media	Riesgo en el desarrollo: Media
Puntos estimados: 1	Iteración asignada: 1
Programador responsable:	Ruddy Chao
Descripción:	Todo screenshot obtenido se tiene que guardar dentro de una archivo pdf para una mayor constancia de la infracción
Prueba funcional:	Los procedimientos son satisfactorios, cuyo guardado de archivo, sea haga con éxito.
Observaciones:	

Fuente: Elaboración propia

3.2.2.7. Historia de usuario 7: Gestión de resultados

A continuación, se muestra las observaciones correspondientes para describir iteraciones, su prioridad, su tiempo estimado, la descripción y la prueba funcional de la historia de usuario.

TABLA 14. *Historia de usuario N. 7*

Numero: 7	Usuario: Administrador.
Nombre de la Historia	Gestión de Resultados
Prioridad: Media	Iteración asignada: 5
Puntos estimados: 1	Riesgo en el desarrollo: Media
Programador responsable:	Ruddy Chao
Descripción:	Se generarán las imágenes y caracteres reconocidos de los infractores.
Prueba Funcional	Los procedimientos son satisfactorios, cuya generación de imágenes de infractores se haga con éxito.

Fuente: Elaboración propia

3.2.3. Spike Arquitectónico

Según Xp nos dice que tenemos que incluir y decidir qué tipo de tecnologías, librerías, frameworks que se utilizaran dentro del proyecto.

Se hizo pruebas de distintas tecnologías a usar en el presente proyecto y se eligieron las siguientes:

- **NetBeans:** Entorno de desarrolló integrado, de codificación y compilación de programas en java.
- **Apache:** Como gestor de base de datos, donde residirá los datos que la aplicación llegue a generar.
- **OpenCV 3.0:** Como librería de visión computador, la cual nos ayudara a la manipulación, filtrado de ruidos, transformaciones de imágenes videos que utilizaremos para las pruebas.

3.2.4. Metáfora de negocio

Se procederá a desarrollar un pequeño glosario en donde el cliente pueda recurrir y se sienta a gusto con los términos del desarrollador.

- **Historia de usuario:** Son las funcionalidades o requerimientos que desea que tenga la aplicación para poder desarrollarlas.
- **Visión por computador:** Parte del campo de inteligencia artificial, en la que se extrae información de imágenes, y dar soluciones en tiempo real a cosas de la vida real.
- **Clasificador de Imágenes:** Es un sistema, donde se puede clasificar las imágenes que uno desea a partir de un entrenamiento a dicho sistema.
- **OCR:** Son aplicaciones que extraen las letras de las imágenes para convertirlas a lenguaje que la computadora entienda y así se pueda procesar información para fines consiguientes.,
- **OpenCV:** Es una librería de código abierto la cual nos ayudara a desarrollar la aplicación, en cuanto al procesamiento de imágenes.

3.2.5. Asignación de roles del proyecto

Según los conceptos de XP todo proyecto tiene que tener a las personas asignadas en una tarea específica, en el caso particular del proyecto realizado se muestra a continuación:

ROLES	ASIGNADO A
PROGRAMADOR	RUDDY CHAO
CLIENTE	Dirección Departamental de Transito Transporte y Seguridad Vial

3.2.6. Plan de entrega

Tras evaluar cada una de las historias de usuarios de acuerdo a prioridad del cliente, se establece que cada iteración tendrá una duración de 1 a 4 semanas, terminado ese lapso se hace la entrega del incremento, el primer plan de entrega posible queda así.

TABLA 15. Iteraciones y posible entrega

MES (ITERACIÓN)	SEMANAS	ITERACIÓN	HISTORIA DE USUARIO
1	1	Clasificador de placas	Entrenamiento de clasificador.
	2		Detección de placa.
	3		
2	1	Superpuesto de vehículo en el paso cebra	Marcador de Paso cebra.
	2		Segmentación de paso cebra. (Detección de Infracción)
	3		
	4		
3	1	OCR	Desarrollo de Segmentación e implementación de Teserract para el OCR.
	2		
	3		
	4		
4	1	Resultados	Gestión de resultados
	2		

Fuente: Elaboración Propia

3.2.7. Representación general en diagrama de caso de uso de la aplicación

Se muestra continuación el diagrama de caso de uso de la aplicación, donde detalla los escenarios en los que la aplicación interactúa con los actores, así también los objetivos a lograr.

El actor denominado **administrador** es una clase de persona que interactúa con la aplicación, y está asociada con los casos de usos reportar infractores, además del marcador paso cebra, este a la vez se relación con el caso de uso segmentación paso cebra, que incluye casos de usos como reconoce la placa, analizar y registrara la placa.

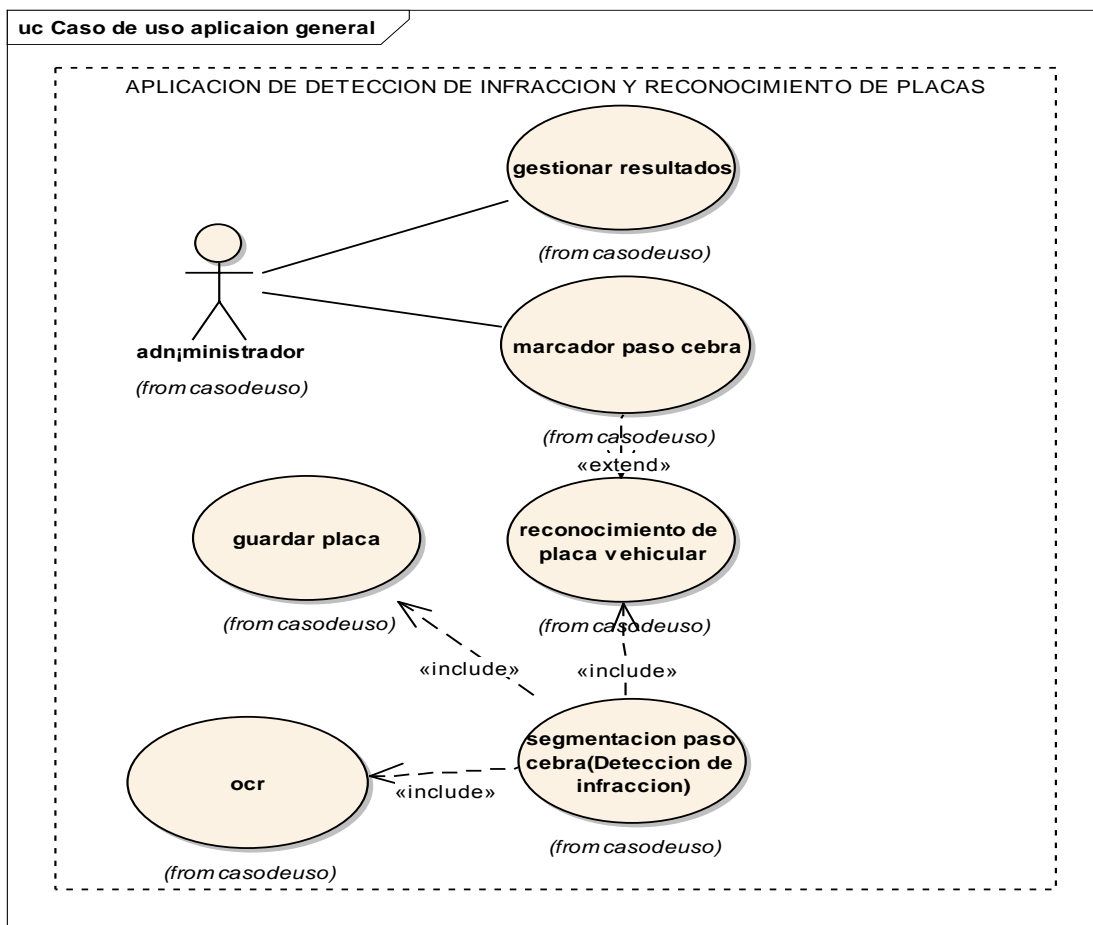


FIGURA 15. Representación general de la aplicación

Fuente: Elaboración Propia

3.2.8. Diagrama de secuencia de la aplicación

A continuación, se muestra el diagrama de secuencia general de la aplicación en cuanto a la detección de infracción, reconocimiento de placa, reconocimiento óptico de caracteres y la inserción del mismo en la base de datos.

El administrador busca un archivo de video o activa l cámara, ingresa los puntos del paso cebra, procesa y pasa a la siguiente interfaz de la segmentación del paso cebra que es el que se encarga de buscar infractores, reconocer la placa, y además de ello insertar los caracteres de la placa en la base de datos.

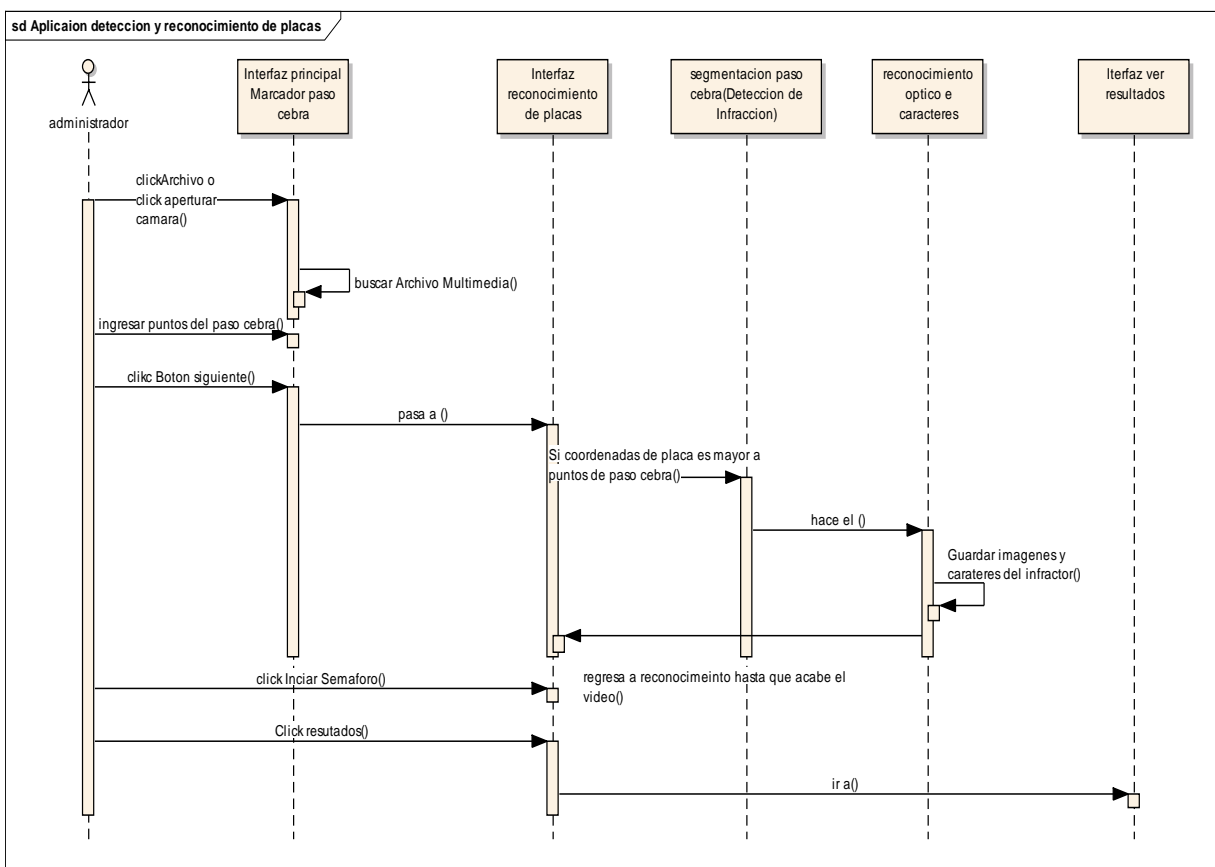


FIGURA 16. Diagrama de secuencia General

Fuente: Elaboración propia

3.3. FASE DE ITERACIONES Y PRODUCCIÓN:

Se desarrolla cada una de las historias de usuario, de acuerdo con plan establecido anteriormente, cabe recordar que el plan puede cambiar de acuerdo a requerimientos o problemas en desarrollo.

3.3.1. Primera iteración

Se procede a realizar las actividades, de acuerdo con el plan de la primera iteración.

TABLA 16. *Primera iteración*

MES	SEMANAS	ITERACIÓN	HISTORIA DE USUARIO
1	1	Clasificador de placas	Entrenamiento de clasificador
	2		Detección de la placa
	3		
Programador	RUDDY CHAO		

Fuente: Elaboración Propia

3.3.1.1. Historia de usuario Entrenamiento del clasificador

Se hace un trabajo de campo lo que consiste, en obtener el mayor número de posibles datos positivos (imágenes de placas) y negativos (imágenes donde no hay placas), para que el clasificador pueda, discriminar y así especificar cuáles imágenes es la que busca y cuáles no.

Para ello utilizamos las herramientas que nos brinda OpenCV, las cuales ya se las especificó, en el capítulo anterior.

- **Imágenes positivas y negativas**

Se muestra a continuación las imágenes positivas que el clasificador tendrá que discriminar, para este caso usaremos las placas que circulan a nivel nacional con un total de 1400 imágenes obtenidas.

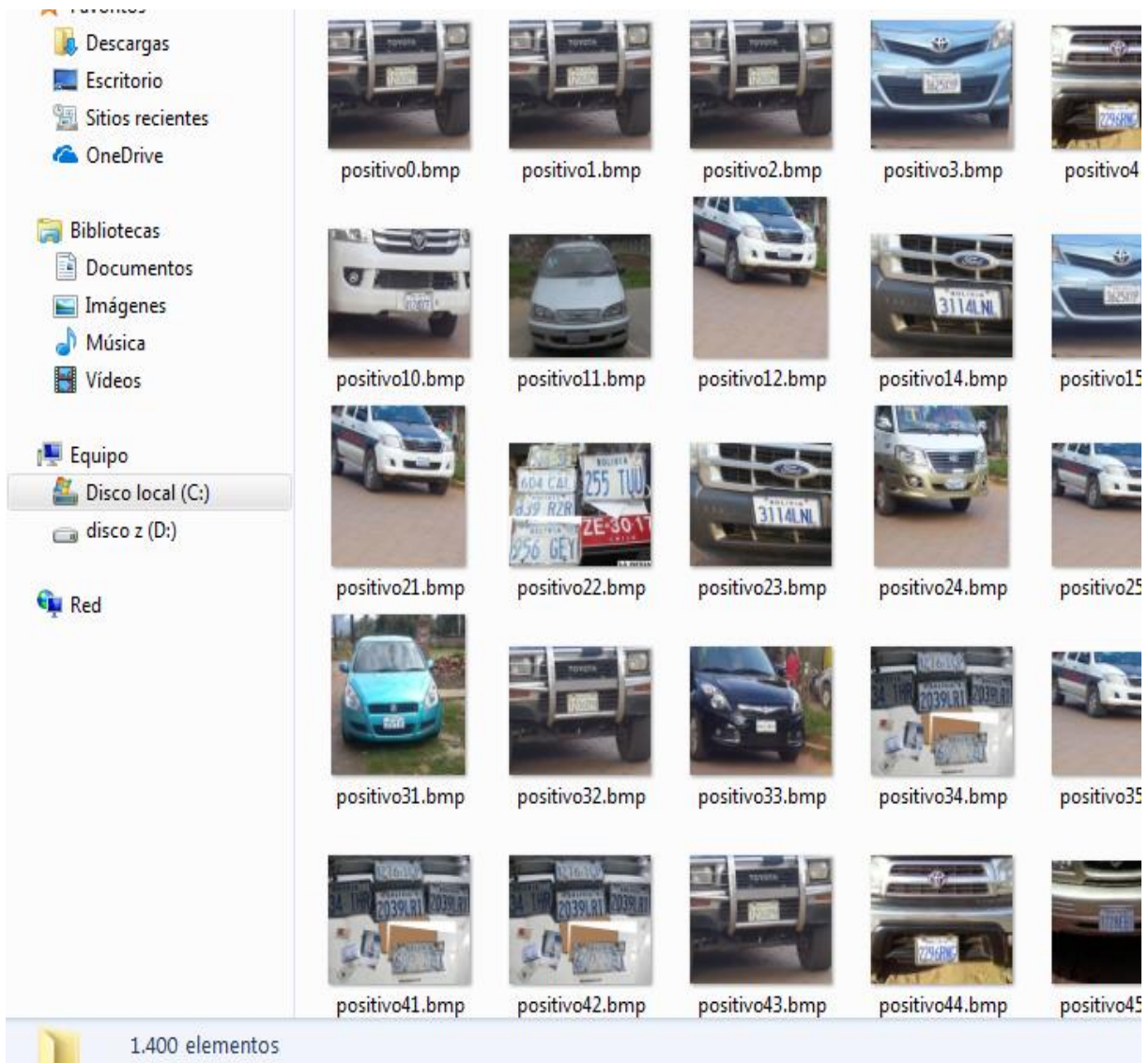


FIGURA 17. *Imágenes positivas*

Fuente: Elaboración propia

También es necesario tener un número determinado de imágenes negativas un total de 1086, las mismas se muestran a continuación. Las imágenes negativas son aquellas que no tiene o el objeto a reconocer durante el entrenamiento del calificador de imágenes.

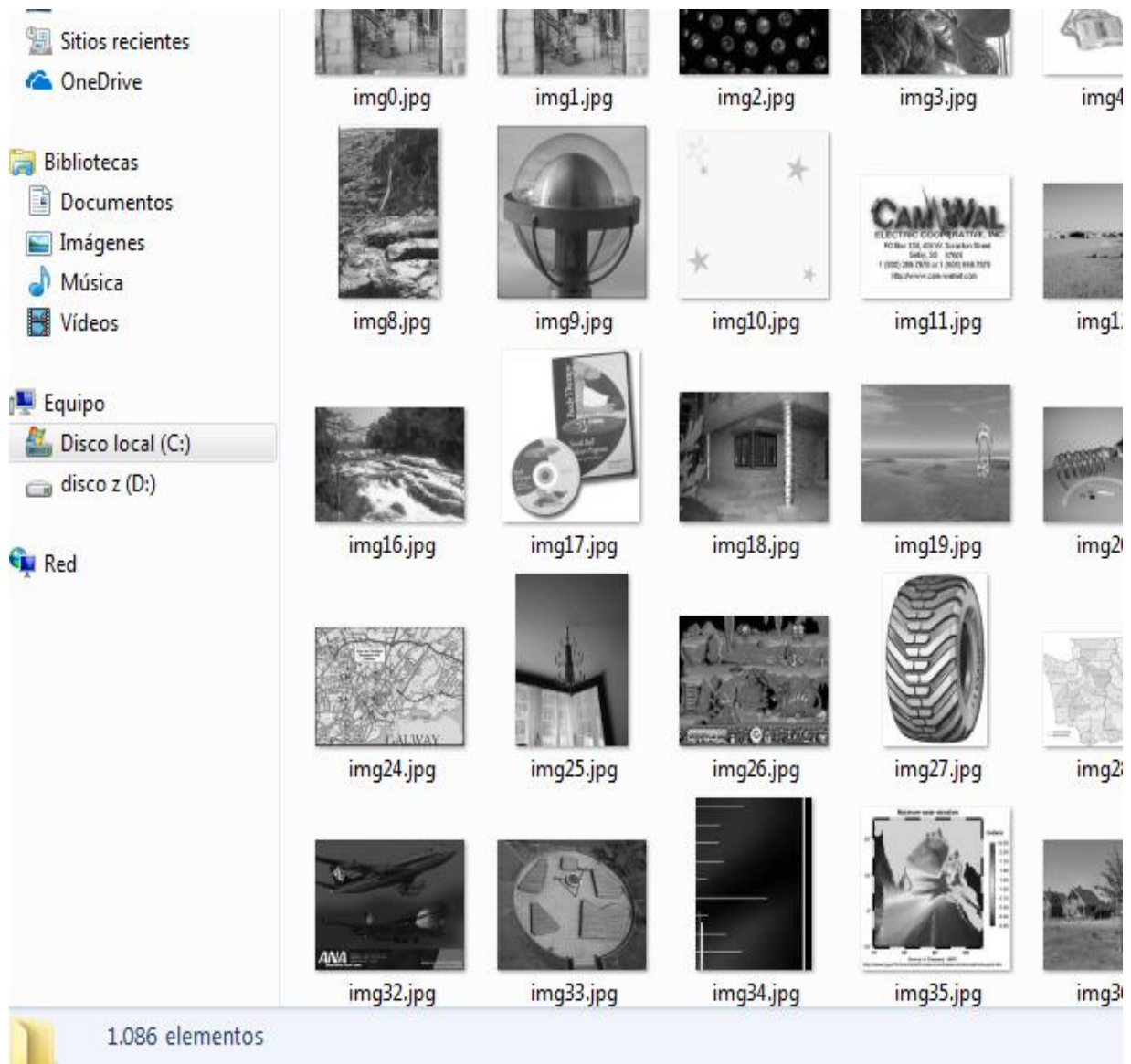


FIGURA 18. *Imágenes negativas*

Fuente: Elaboración propia

- **Marcador de imágenes**

Objectmarker es una herramienta de OpenCV, que nos permite encuadrar el objeto a reconocer dentro de las imágenes positivas.

El proceso es muy simple pero puede ser tedioso dependiendo de cuantas imágenes macaremos. Se marca cada una de las imágenes positiva, finalizando con la obtención de un archivo info.txt, en ella se estable los parámetros en el que se encuentra la placa (x, y, ancho alto), en pixeles.

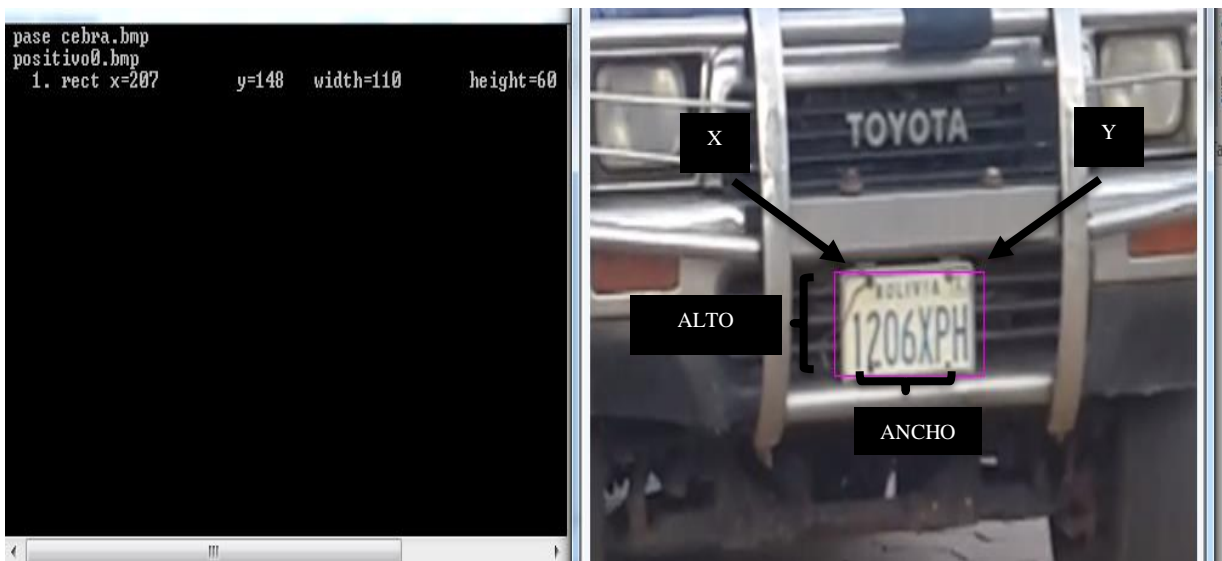


FIGURA 19. *Marcación y obtención de datos*

Fuente: Elaboración propia

- **Creación de imágenes positivas**

Para la creación de imágenes positivas se utiliza la herramienta de OpenCV `opencv_createsamples.exe`, la misma nos proporciona un archivo de vectores donde se encuentra las imágenes recortadas y en escala de grises de acuerdo a parámetros que le dimos. Como se muestra en la siguiente imagen:

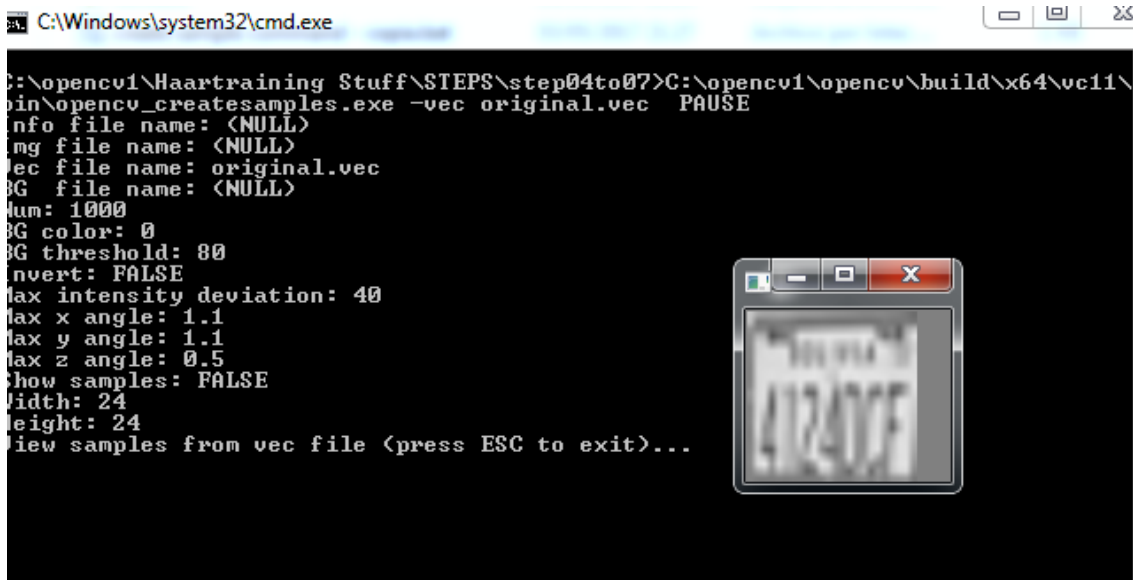


FIGURA 20. Verificación de la creación de imágenes positivas

Fuente: Elaboración Propia

- **Entrenamiento del clasificador:**

Para este último paso se procede al entrenamiento del clasificador, para lo cual usaremos, la herramienta de OpenCV `Opencv_traincascade.exe`, de acuerdo a documentación se le asigna una cantidad necesaria de etapas en la cual entrenará.

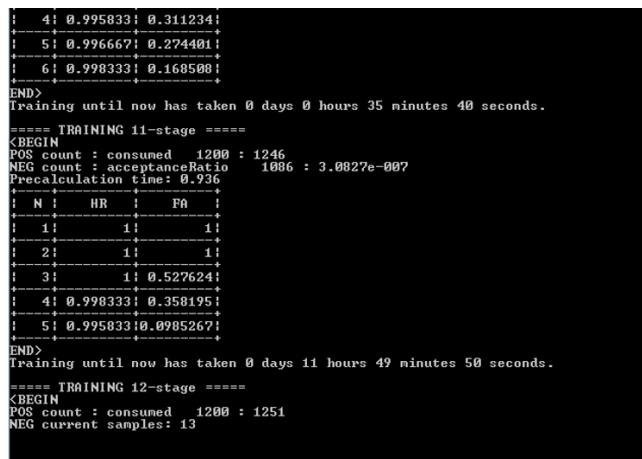


FIGURA 21. Entrenamiento de clasificador

Fuente: Elaboración Propia

Nos proporcionará un archivo cascade.xml, en el cual procedemos a utilizar para utilizar en la codificación y detección de las placas nacionales.

3.3.1.2. Historia de usuario detección de placa

- Para fines de diseño empezamos con el desarrollo de las tarjetas CRC (Clase-Responsabilidad- Colaboración), además de utilizar los casos de usos para una mejor comprensión de cada una de las historias de usuario.

TABLA 17. Tarjeta CRC detección de placa

Clase: Detección de placa	
Responsabilidad:	Colaboración:
<ul style="list-style-type: none"> - Obtener archivo multimedia a detectar o cámara - Detectar placa 	<ul style="list-style-type: none"> - CascadeClassifier. - Procesador de imagen.

Fuente: Elaboración propia

- **Funcionalidad de la aplicación:**

Esta esta interfaz que detecta la placa vehicular, según la historia de usuario.



FIGURA 22. Screenshot de la detección de placa

Fuente: Elaboración propia

PRUEBA DE ACEPTACIÓN:

La prueba realizada a cada una de las historias de usuario desarrolladas, se establecen a través de caja negra.

PRUEBA N.º 1: Detección de placa

Propósito: Verificar el correcto funcionamiento de la historia de usuario N.2. Detección de placa.

Pre Requisitos: Sin requisitos para ingreso, historia desarrollada para fines de prueba.

Resultados Esperados: No detecta placas vehiculares de circulación nacional.

Evaluación de la Prueba: No superado. Ya que no concuerda con la prueba funcional de la historia de usuario N2.

3.3.2. Segunda iteración

Durante la iteración 1, se dio el caso de que no se superó con éxito la historia de usuario N 1, que encomendó el usuario o la planificación que se estableció por lo tanto es necesario replantear nuestro cronograma de trabajo para la segunda Iteración, el nuevo plan de entrega queda así.

TABLA 18. *Plan de entrega da la 2ºda iteración*

MES	SEMANAS	ITERACIÓN	HISTORIA DE USUARIO
2	1	Clasificador de placas	Entrenamiento de clasificador
			detección de la placa
	2	Superpuesto de vehículo en el paso cebra	Marcador de Paso cebra
	3		
4	Segmentación de paso cebra		
Programador		RUDDY CHAO	

Fuente: Elaboración propia

3.3.2.1. Historia de usuario: Entrenamiento de clasificador

Nos dimos cuenta de que la falla que cometimos, en la primera iteración, se las cita a continuación:

- **Placas de distintas partes del país y extranjeras:** Ya el clasificador en cascada es muy específico en cuanto a la búsqueda de patrones, dado que el algoritmo de clasificación es por características y no así pixel por pixel.
- **Cantidad de mínima de imágenes positivas:** A mayor número de posibles imágenes positivas que le demos al clasificador mayor, más eficaz y preciso será la detección.
- **Etapas de entrenamiento:** Para una mayor precisión al clasificador se le tiene que dar una cierta cantidad de etapas, en la que va a entrenar.

Sabiendo las deficiencias que tuvimos en la primera iteración se procedió a obtener más posibles positivos (esta vez se tomó en cuenta solo placas nacionales), con un total de 1500 imágenes positivas.

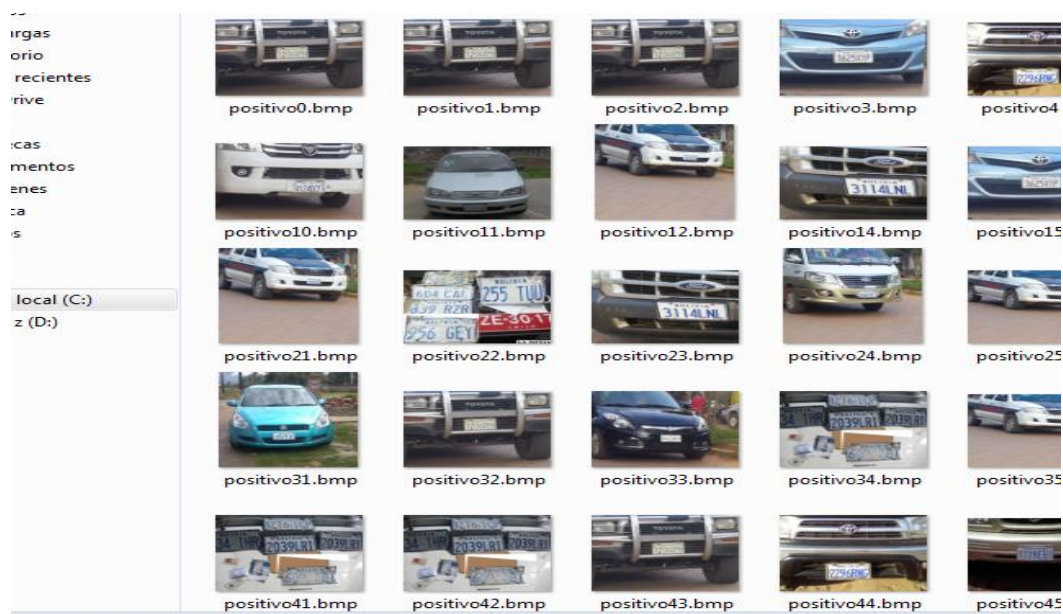


FIGURA 23. *Imágenes Positivas (placa Nacionales)*

Fuente: Elaboración propia

- **Funcionalidad de la aplicación:** Se muestra que a diferencia de la primera prueba el reconocimiento de placas vehiculares de circulación nacional, es mucho más eficaz, con un 60 % de acierto.



FIGURA 24. *Reconocimiento de placas vehicular*

Fuente: Elaboración Propia

3.3.2.2. Historia de usuario Marcador de Paso cebra

Viendo que una de las Historia de usuario de la iteración N1 ya se desarrolló con éxito, se procede automáticamente a desarrollar la siguiente tarjeta CRC y su correspondiente Historia de usuario.

TABLA 19. Tarjeta CRC marcador de paso cebra

Clase: marcador paso cebra	
Responsabilidad: <ul style="list-style-type: none"> • Buscar grabación o activar cámara. • Obtener el primer cuadro imagen de la cámara o del video a analizar. • Marcar los cuatro puntos respectivos del paso cebra. 	Colaboración: <ul style="list-style-type: none"> Procesador de imagen. Datos puntos Encapsulador de dimensiones

Fuente: Elaboración Propia

- **Diagrama de Casos de Uso Marcador paso cebra**

El Administrador es el que interactúa con el caso de uso **Marcador paso cebra**, la cual contiene opciones de buscar grabaciones o cámara web, así también la opción de marcar los puntos respectivos del paso cebra.

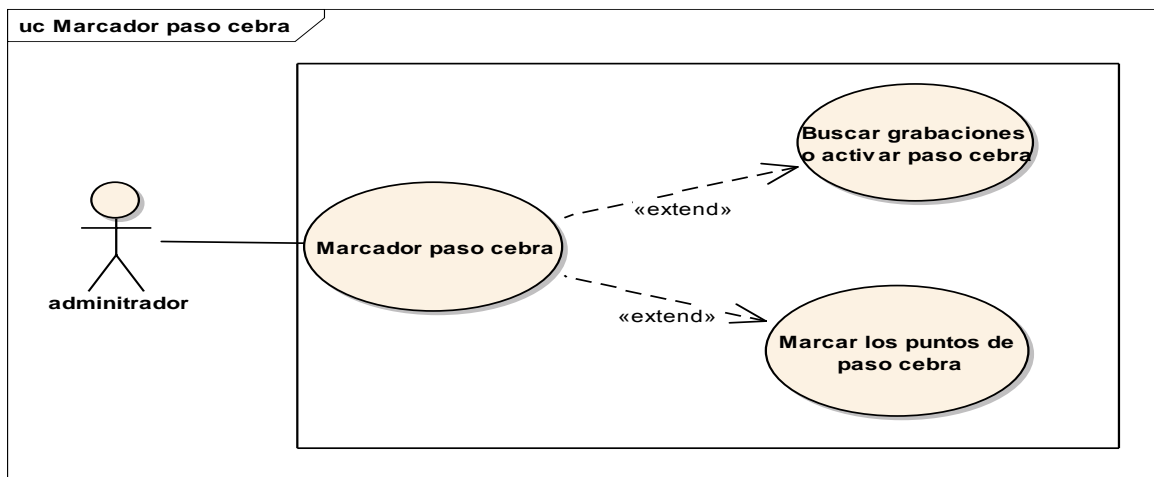


FIGURA 25. Casos de Uso. - Marcador de paso Cebra

Fuente: Elaboración Propia

- **Funcionalidad de la historia de usuario-.** Se muestra a continuación la imagen de la aplicación correspondiente al marcador de paso cebra. Donde se aprecia el marcado del paso cebra con los puntos 1,2,3,4.

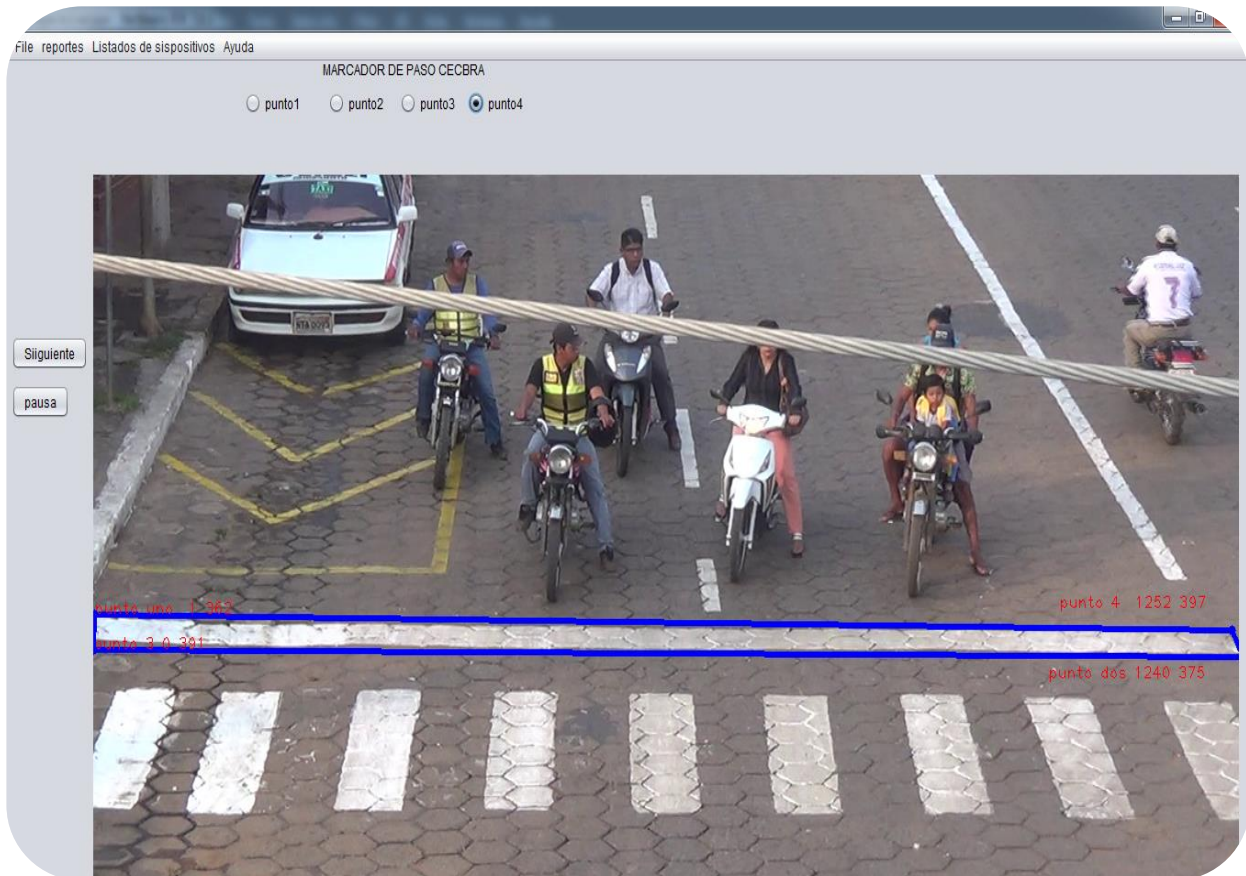


FIGURA 26. *Marcador de paso cebra*

Fuente: Elaboración Propia

- **PRUEBA DE ACEPTACIÓN:**

Se procede a realizar la prueba utilizando la técnica, de caja negra.

PRUEBA N.º 2: Marcador paso cebra

Propósito: Verificar el correcto funcionamiento de la historia de usuario N.3 marcador paso cebra.

Pre Requisitos: ninguno

Resultados Esperados: Búsqueda de archivo de video a analizar o activación de cámara, Marcado y obtención de coordenadas de la imagen con éxito.

Evaluación de la Prueba: Superada con éxito. Tal como se estableció en la prueba funcional respectiva.

3.3.2.3. Historia de usuario Segmentación de paso cebra

Se procede a desarrollar la siguiente la tarjeta CRC de la Historia de usuario planeada, con las responsabilidades dadas, así como también las clases que le colaboran.

- **Tarjeta CRC Segmentación de paso cebra**

TABLA 20 Tarjeta CRC Segmentación de paso cebra

Clase: Segmentación de paso cebra	
Responsabilidad: <ul style="list-style-type: none">• Reproducir automáticamente el video a analizar.• Segmentar y que sea visible en el video las partes donde está el infractor.• Identificar al infractor y mostrar en pantalla al vehículo infractor.	Colaboración: <ul style="list-style-type: none">• Encapsulador Cebra• Procesador de imagen.• CascadeClassifier• Datos

Fuente: Elaboración Propia

- **Diagrama de Casos de Uso Segmentación de paso Cebra (detección de infracción).**

El administrador tiene una participación implícita en la segmentación del paso ya que, del marcador de paso de cebra pasa parámetros de las posiciones de los puntos del paso cebra.

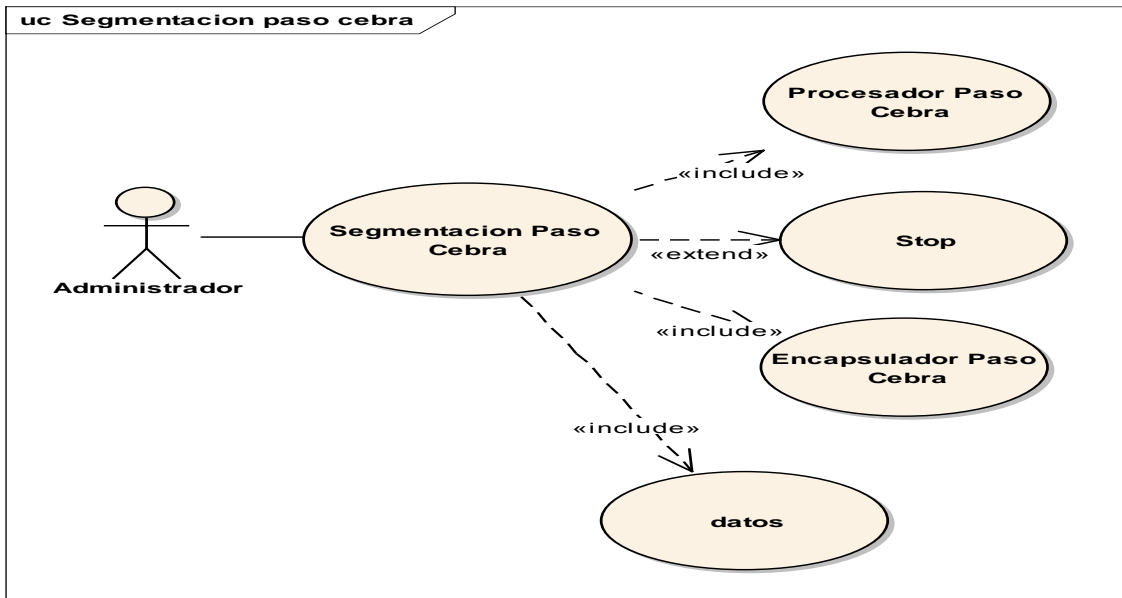


FIGURA 27. Casos de Uso. – Segmentación De paso cebra

Fuente: Elaboración Propia

- **Técnicas de Visión por computador par la Segmentación del paso Cebra-**. Para realizar los algoritmos de visión por computador se utilizó específicamente la librería OpenCV.
 En la segmentación del paso cebra se hizo los siguientes pasos:
 - **Paso1.** Redimensionar la imagen, dilatar la imagen (esto para que los pixeles se distribuyan en una determinada distancia).
 - **Paso 2.** Convertir la imagen de formato bgr a formato hsv, esto para que los parámetros de la imagen a tratar sean más accesibles y no complicado como los es con tres canales de colores (bgr).

- **Paso 3.** Umbralizar la imagen, esto permite binarizar la imagen (blanco y negro), extraer contornos de regiones que estén con color blanco.
- **Paso 4.** Separar los contornos que estén dentro de las dimensiones dadas anteriormente en la imagen (punto 1, punto 2, punto 3, punto 4), y almacenar los datos (x, y, ancho, alto de cada contorno), en un ArrayList dinámico y ordenarlos de mayor a menor.

ArrayList A

	x	y	ancho
lista	684	569	355
lista	684	569	339
lista	682	569	352
lista	702	569	335
lista	700	569	342

- **Paso 5:** Calcular la posición del infractor, con los datos dados anteriormente mediante el siguiente formula: Dado conjunto de datos **A** como entrada.

Donde:

X es la posición en x de la nueva imagen donde se encuentra el infractor

Ancho es la anchura en la imagen original del infractor.

$$\mathbf{Xi} = \mathbf{xi_{i+1}} + \mathbf{ancho_{i+1}}.$$

$$\mathbf{Ancho} = (\mathbf{xi_{i+1}} + \mathbf{ancho_{i+1}}) - \mathbf{xi}$$

La posición **y** se toma en cuenta del ArrayList original(**y**)

Con estos datos se procede a crear una nueva imagen para su procesamiento siguiente.

Como las distintas imágenes que mostramos a continuación:

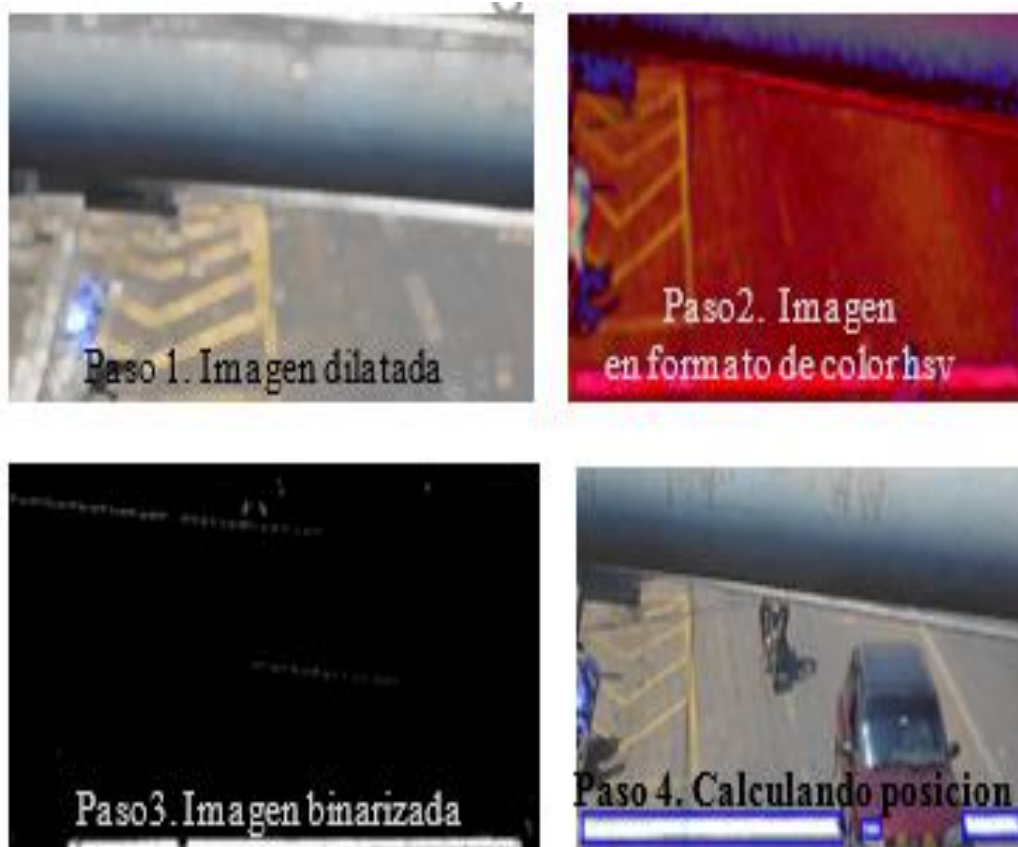


FIGURA 28. *Pasos de la segmentación paso cebra mediante Visión por Computador*

Fuente: Elaboración propia

- **Funcionalidad de la historia de usuario:** El algoritmo desarrollado para esta funcionalidad no discrimina entre autos y motocicletas por lo que toma como infractor todo aquel motorizado que se sobre ponga sobre la línea blanca y las marca con líneas azules (para fines de ilustración). Se muestra a continuación:



FIGURA 29. Prueba de la Segmentación del Paso cebra

Fuente: Elaboración Propia

- **PRUEBA DE ACEPTACIÓN:** Se procede a realizar la prueba de utilizando la técnica, de caja negra.

PRUEBA N. 3: Segmentación de Paso cebra.

Propósito: Verificar el correcto funcionamiento de la historia de usuario N 3. superpuesto de paso de cebra.

Pre Requisitos: Elección Previa de archivo de video y marcado de puntos correctamente.

Resultados Esperados: Segmentación correctas del paso cebra a excepción de objetos que tengan como tono de color blanco ya que el algoritmo se basa en umbral del color blanco, para para la detección de los infractores.

Evaluación de la Prueba: La prueba superada con ciertas dificultades. (el reconocimiento se hace lento, autos de color blanco en algunos casos no reconoce)

3.3.3. Tercera iteración

Siguiendo estrictamente el plan de entregas se procede a realizar la Historia de Usuarios enmarcadas a la tercera iteración correspondiente a la tarjeta CRC respectiva, terminando así con las tres iteraciones más importantes del desarrollo de la aplicación. Cabe mencionar que la historia de usuario Segmentación de paso cebra se procederá a replantear ya que no se logró el objetivo con éxito por lo que el cronograma queda de la siguiente manera.

TABLA 21. Plan de entrega de 3^{ra} iteración

MES	SEMANAS	ITERACIÓN	HISTORIA DE USUARIO
3	1	Superpuesto de vehículo en el paso cebra	Segmentación de paso cebra (Detección de infracción)
	2		
	3		
4	1	OCR	Desarrollo de Segmentación e implementación de Teserract
	2		
	3		
	4		
Programador		RUDDY CHAO	

Fuente: Elaboración propia

3.3.3.1. Historia de usuario Detección de infracción

Para fines de diseño empezamos con el desarrollo de las tarjetas CRC (Clase-Responsabilidad- Colaboración), además de utilizar los casos de usos para una mejor comprensión de cada una de las historias de usuario.

- **Tarjeta CRC Detección de Infracción**

TABLA 22 Detección de infracción (Segmentación paso cebra)

Clase: Detección de Infracción (Segmentación de paso cebra)	
Responsabilidad: <ul style="list-style-type: none">• Reproducir automáticamente el video a analizar.• Obtener imagen recortada del video analizar para más eficiencia de la aplicación.• Calcular la posición del infractor a partir de las coordenadas de la placa (si esta está más adelante de la placa).• Identificar al infractor y mostrar en pantalla al vehículo infractor.	Colaboración: <ul style="list-style-type: none">• Procesador de imagen.• CascadeClassifier• Proceso Video a Analizar• Encapsulador de datos de Infractor• Resultados

Fuente: Elaboración Propia

- **Diagrama de Casos de Uso Detección de Infracción.**

El administrador tiene una participación implícita en la segmentación del paso ya que, del marcador de paso de cebra pasa parámetros de las posiciones de los puntos del paso cebra.

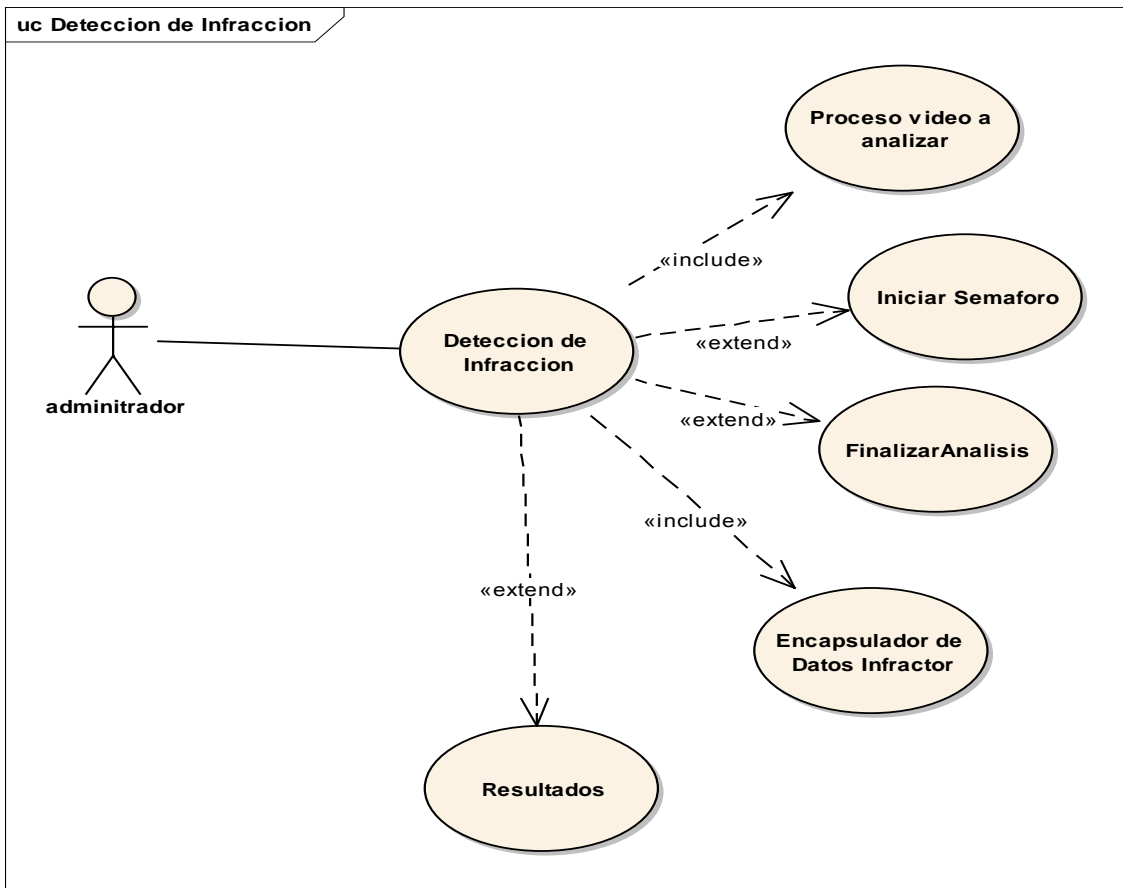


TABLA 23. Caso de uso detección de infracción

Fuente: Elaboración propia

Técnicas de Visión por computador para la Detección de infracción.- Para realizar los algoritmos de visión por computador se utilizó específicamente la librería OpenCV.

En la segmentación del paso cebra se hizo los siguientes pasos:

- **Paso1.-** A partir de la placa ya reconocida se procede a calcular si sus coordenadas **y**, son mayor a las coordenadas **y1** y **y3** del paso cebra previamente marcado.
- **Paso2.-** Si la condición es verídica se procede a guardar como imagen la placa, la imagen del auto del infractor, y la imagen general del video.
- **IMPLEMENTACIÓN DE LA HISTORIA DE USUARIO:** Se muestra a continuación imágenes de la infracción de detectada.



FIGURA 30. *Imagen 1 Detección de Infracciónl, Imagen 2 auto Infractor*

Fuente: Elaboración propia

- **PRUEBAS DE ACEPTACIÓN:**

Se procede a realizar la prueba utilizando la técnica de caja negra.

PRUEBA N. 4: Desarrollo de Detección de infracción

Propósito: Detectar la infracción de paso cebra, obtener imagen del auto infractor y obtención de imagen general.

Pre Requisitos: Obtener la imagen temporal de la placa detectada.

Resultados Esperados: Detección del infractor con éxito.

Evaluación de la Prueba: Superada con éxito. Tal como se estableció en la prueba funcional respectiva.

3.3.3.2. Historia de usuario Desarrollo de Segmentación e implementación de Teserract

Dentro de la iteración N3 OCR existe la Historia de Usuario Segmentación e implementación de Teserract para ello se desarrollará la tarjeta CRC respectiva.

Cabe mencionar que la historia de usuario OCR hace referencia a la conversión de imágenes de caracteres, en un formato que le sea entendible para la computadora y de esa forma poder registrar y controlar para fines de reportes en la aplicación.

Para Con el fin de obtener caracteres que se sean mayúsculas y números naturales

- **Tarjeta CRC Segmentación e implementación de Teserract para OCR**

TABLA 24. Tarjeta CRC Segmentación e implementación de Teserract para el OCR

Clase: Segmentación e implementación de Teserract para OCR	
Responsabilidad: <ul style="list-style-type: none"> • Obtener archivo de imagen temporal de la placa. • Segmentar el archivo (Procesamiento de la imagen). • Discriminar caracteres a obtener. • Obtener los caracteres respectivos. • Registrar los caracteres. 	Colaboración: <ul style="list-style-type: none"> • Procesador de imagen. • Teserract. • Guardar número de placa.

Fuente: Elaboración propia

- **Diagrama de Caso de Uso:** Segmentación e implementación de Teserract

Al igual que las historias de usuario de segmentación paso cebrá y detección de la placa, en el reconocimiento óptico de caracteres el actor no tiene una acción directa ya que es una consecuencia directa de la anterior historia de usuario.

Por lo tanto, se tiene el caso uso OCR, que incluye otros como La librería de OCR Teserract, caso de uso mejoramiento de imagen, y por último el caso de uso Registrar número de placa en la base de datos.

Dentro de los parámetros que se toma en cuenta, en el reconocimiento óptico de caracteres, se hace una discriminación a caracteres que no serían necesarios para la aplicación, tales como los puntos líneas caracteres especiales, letras minúsculas, etc. Como se denota a continuación:

$$C \in A \iff C = A$$

Donde $A = \{A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

C= Carácter que es reconocido por el OCR.

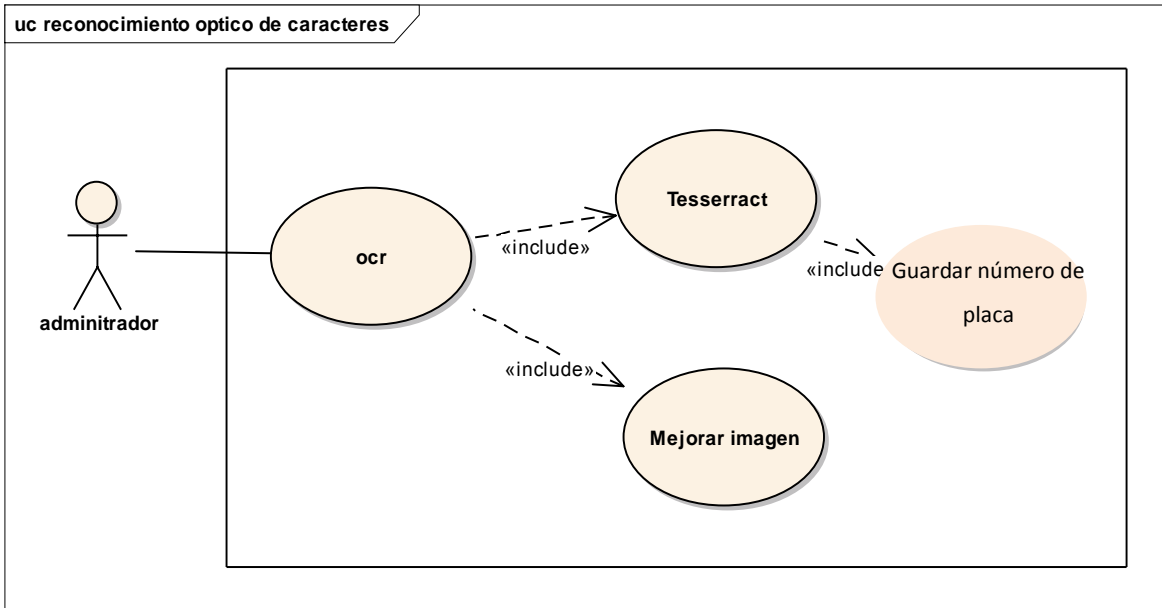


FIGURA 31. Caso de uso segmentación e implementación de Tesseract

Fuente: Elaboración propia

- **Técnicas de Visión por computador para el desarrollo del reconocimiento óptico de caracteres-**. Para realizar los algoritmos de visión por computador se utilizó la librería OpenCV, y la ayuda de la librería Tesseract (convierte una imagen de texto a caracteres).

Para la implementación de la librería se desarrolló previamente los siguientes pasos:

Paso 1. Teniendo en cuenta que tenemos la posición del auto, ya tenemos reconocida la placa en dicha región, procedemos a tratarla imagen, la convertimos en escala de grises, luego se hace una binarización adaptativa.

Paso 2. Se observa que las imágenes tienen ruidos, para ello se procede a pintar de color blanco aquellas zonas que sus píxeles sean negros además que sean menor a un número determinado.

Paso 3. Seguidamente se observa que los caracteres quedan con espacios blancos, para lo cual se procede aplicar el mismo paso anterior con la diferencia de que los espacios blancos se transformen en negro.

Paso 4. Se procede a cortar los contornos, para ello se hace una comparación de los píxeles de cada lado partiendo del centro, si el píxel analizado es igual a blanco se guarda como parámetro para ser cortado (x, y, ancho).

Paso 5. Se recorta la imagen recorta según parámetros y se guarda como archivo temporal.

Paso 6. Se procede a implementar la librería Teserract, a partir de la imagen temporal, para luego obtener los caracteres y posterior registro.



FIGURA 32. *Pasos del tratamiento e implementación de Teserract mediante Visión por Computador*

Fuente: Elaboración propia

- **IMPLEMENTACIÓN DE LA HISTORIA DE USUARIO:** Al implementar la historia de usuario el reconocimiento de los caracteres de la placa y el guardado se procesa de forma automática, por lo tanto su implementación será mostrada en la historia de usuario gestión de resultados.
- **PRUEBAS DE ACEPTACIÓN:**
Se procede a realizar la prueba utilizando la técnica de caja negra.
PRUEBA N. 5: Desarrollo de Segmentación e implementación de Teserract.
Propósito: Segmentar, obtener, registrar el número de la placa del vehículo infractor.
Pre Requisitos: Obtener la imagen guardada temporalmente.
Resultados Esperados: Segmentación, obtención y registro con éxito.
Evaluación de la Prueba: Superada con éxito. Tal como se estableció en la prueba funcional respectiva.

3.3.4. Cuarta Iteración

Siguiendo estrictamente el plan de entregas se procede a realizar la Historia de Usuarios Gestión de resultados, enmarcadas a la cuarta iteración correspondiente a la tarjeta CRC respectiva.

TABLA 25 Plan de entrega de 4ta iteración

MES	SEMANAS	ITERACIÓN	HISTORIA DE USUARIO
5	1	Resultados	Gestión de resultados
Programador		RUDDY CHAO	

Fuente: Elaboración propia

3.3.4.1. Historia de usuario Gestión de resultados

Para fines de diseño empezamos con el desarrollo de las tarjetas CRC (Clase-Responsabilidad- Colaboración), además de utilizar los casos de usos para una mejor comprensión de cada una de las historias de usuario.

- **Tarjeta CRC Gestión de resultados**

TABLA 26 *Tarjeta CRC Gestión de resultados*

Clase: Gestión de resultados	
Responsabilidad: <ul style="list-style-type: none"> • Mostrar las imágenes de los infractores (imagen general, imagen del infractor, imagen de la placa). • Mostrar numero de la placa reconocida. 	Colaboración: <ul style="list-style-type: none"> • Procesador de imagen. • Encapsulador de datos infractor

Fuente: Elaboración propia

- **Diagrama de Caso de uso Gestión de resultados**

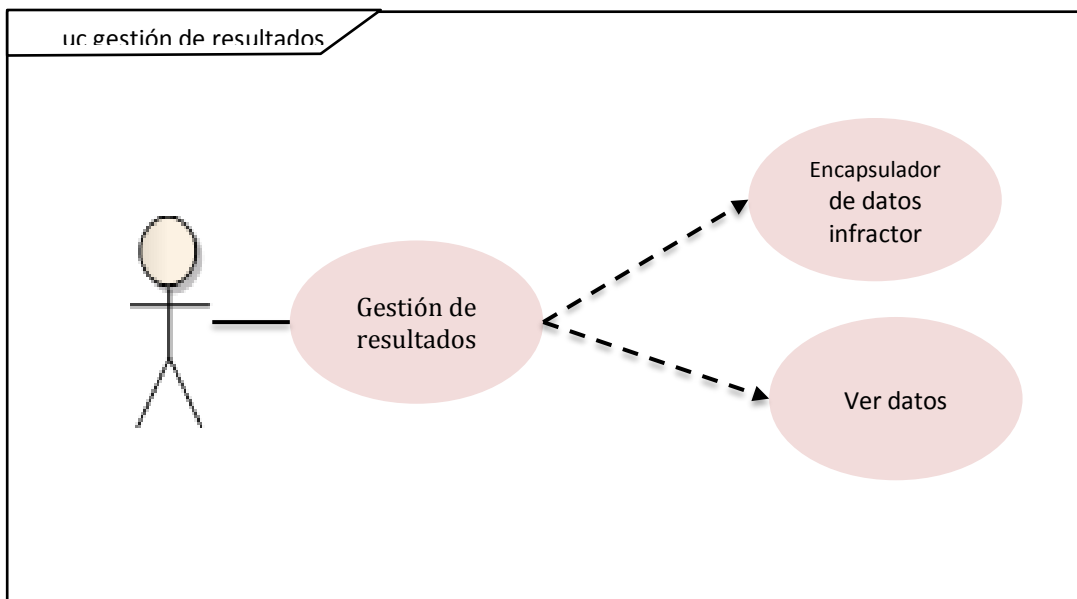


FIGURA 33 *Caso de uso Gestión de resultados*

Fuente: Elaboración propia

- **IMPLEMENTACIÓN DE LA HISTORIA DE USUARIO:** Al implementar la historia de usuario Gestión de resultados

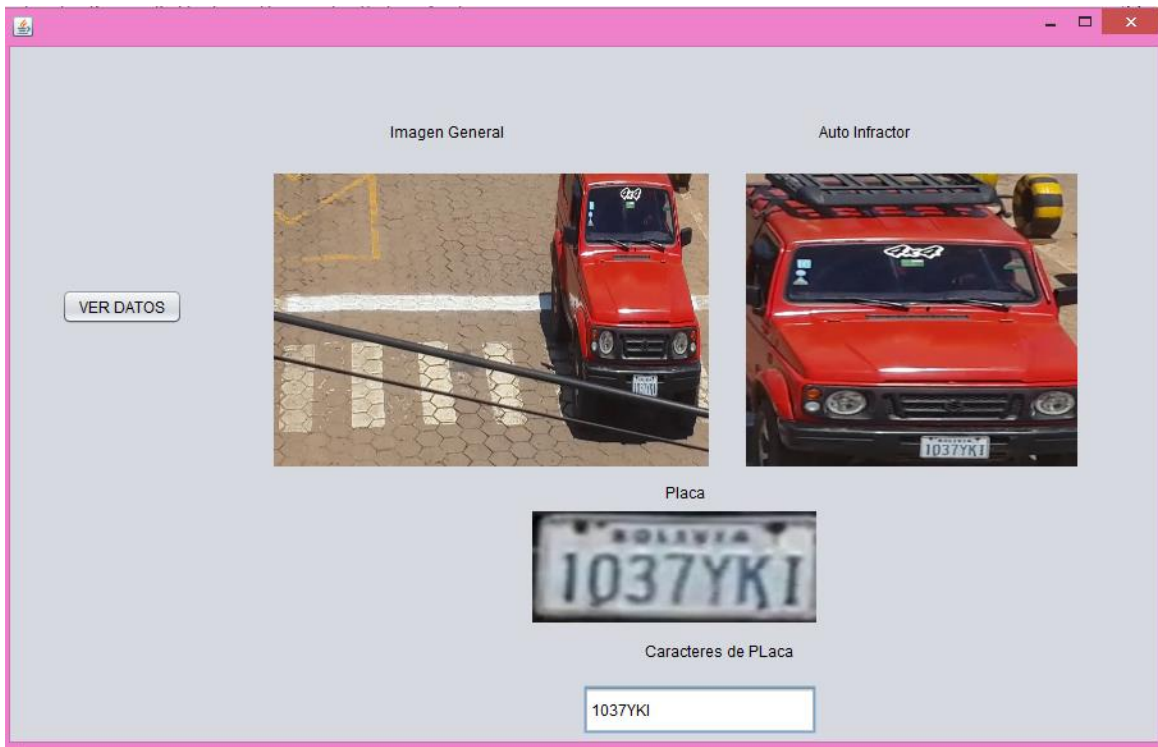


FIGURA 34 Visualización de resultados

Fuente: Elaboración propia


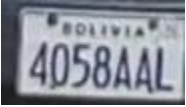







- **PRUEBAS DE ACEPTACIÓN:**
 Se procede a realizar la prueba utilizando la técnica de caja negra.
PRUEBA N. 6: Gestión de Resultados.
Propósito: Mostrar imagen de del infractor, imagen general del video, y los caracteres reconocidos.
Pre Requisitos: Obtener imagen de Placa.
Resultados Esperados: Muestra de imagen y obtención de caracteres con exito.
Evaluación de la Prueba: Superada con éxito. Tal como se estableció en la prueba funcional respectiva.

3.4. PRUEBAS DE RECONOCIMIENTO

En esta sección se mostrará mediante imágenes y tablas, los reconocimientos respectivos de la aplicación y se intentará promediar un porcentaje de éxito.

3.4.1. Detección de la placa

TABLA 27 Placas detectadas

Imagen	Placa Reconocida	Placa detectada si/no	Porcentaje de acierto
		Si	100%
		Si	100%
		Si	100 %
		Si	100%
	-	No	0
PORCENTAJE TOTAL			80%

Fuente: Elaboración propia





La detección de la placa es eficaz a un 90 %, cabe mencionar que para que haiga una detección más precisa se necesita un buen dispositivo que capte las imágenes a ser reconocidas.



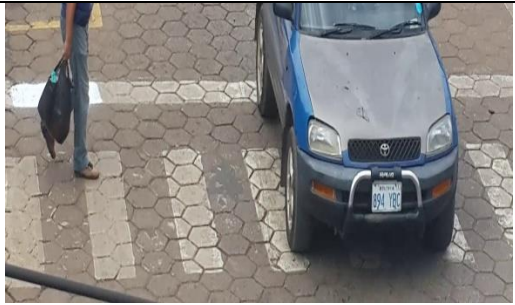

3.4.2. Detección de la infracción del automóvil

Como se explicó anteriormente la Detección de la infracción se refiere a la infracción del vehículo que se posa encima del paso cebra o pasa cuando el semáforo está en rojo.

En la tabla 23 se muestra resultados de reconocimiento para distintos vehículos, cabe mencionar que este modulo es gracias a la detección previa de la placa y al análisis de la posición donde se encuentra la misma (Si esta sobre o delante de la línea de paso cebra), se procede a recuperar la imágenes del infractor.

TABLA 28. Vehículos infractores reconocidos

Imagen	Vehículo reconocido	Cantidad de vehículo	Porcentaje de acierto
		1	100%
		1	100%


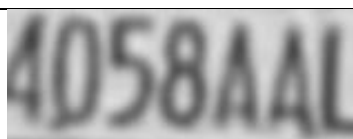

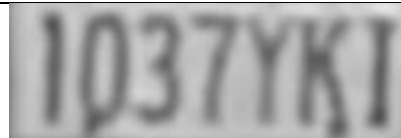
		1	100 %
		1	100%
PORCENTAJE TOTAL			100%


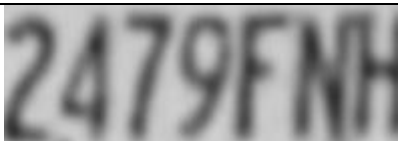


Fuente: Elaboración propia

La detección de la infracción del automóvil es eficaz en un 100%, cabe mencionar que esto es si solo si se reconoce la placa del automóvil una vez que sobre pase el paso cebra.

3.4.3. Reconocimiento de caracteres de la placa:

TABLA 29. Tabulación de resultados de placas reconocidas

Placa reconocida	Placa Mejorada	Cantidad de caracteres	Caracteres reconocidos	Porcentaje de acierto
		5	40586A1	71%
		7	1037YKI	100%

		4	24 79W	57 %
		5	894YH5	66%
PORCENTAJE TOTAL				73%

Fuente: Elaboración Propia

Según resultados obtenidos a partir del proceso de reconocimiento de caracteres de la placa, se evidencia que la capacidad de reconocimiento está a un 73%.

3.5. CALIDAD DE SOFTWARE

Basado en la norma ISO/IEC 9126 se ha elaborado un cuestionario de evaluación (Ver anexo 2), permite especificar y evaluar la calidad del software, que fue valorado a criterio de: un usuario identificado como Administrador

Respecto a la medida tipo nivel, se establecen 5 niveles según se muestra a continuación. (Universidad Politecnica de Madrid, 2008)

El usuario califica un puntaje desde 1 hasta 5, de acuerdo a la siguiente escala.

TABLA 30. Descripción de medidas

VALOR	ESCALA
1	Muy bajo
2	Bajo
3	Medio
4	Alto
5	Muy Alto

Fuente: (Universidad Politecnica de Madrid, 2008)

De acuerdo a la norma ISO/IEC 9126, esta se categoriza en seis elementos para realizar la evaluación del sistema. A continuación, se detalla estos elementos:

- Para la definición de métricas, se toma en cuenta los atributos que la componen bajo la siguiente expresión.

$$M = \sum_{n=1}^n (v * p) / \sum_{n=1}^n p$$

Donde **M**=a la evaluación del atributo, **v** = a la calificación dada, **p**= peso de cada atributo.

3.5.1. Evaluación de la funcionalidad de la aplicación

TABLA 31 *Resumen de evaluación a la funcionalidad de la aplicación*

REF	ATRIBUTO	PESO	RESULTADO
1	Adecuación	10	5
2	Exactitud	10	5
3	Seguridad	10	3

Fuente: Elaboración propia

Medida de Funcionalidad= $(10 * 5 + 10 * 5 + 10 * 3) / (10 + 10 + 10) = 4,33$

De acuerdo al resultado obtenido, se califica como valor aceptable, por lo que la aplicación si cumple con la métrica de funcionalidad de acuerdo a los requerimientos exigidos.

3.5.2. Evaluación de la confiabilidad

TABLA 32 *Resumen de evaluación a la confiabilidad de la aplicación*

REF	ATRIBUTO	PESO	RESULTADO
4	Madurez	10	5
5	Recuperabilidad	10	5

Fuente: Elaboración propia

Confiabilidad= $(10 * 5 + 10 * 5) / (10 + 10) = 5$

De acuerdo al resultado obtenido, se califica como valor aceptable de la evaluación de la aplicación.

3.5.3. Evaluación de la usabilidad

TABLA 33 *Resumen de evaluación a la usabilidad de la aplicación*

REF	ATRIBUTO	PESO	VALORACIÓN
6	Entendimiento	10	5
7	Aprendizaje	10	5
8	Operabilidad	10	5
9	Atracción	10	4

Fuente: Elaboración propia

$$\text{Usabilidad} = (10*5+10*5+10*5+10*4) / (10+10+10+10) = 4,75$$

De acuerdo al resultado obtenido en su métrica de Usabilidad, se establece que la aplicación tiene una gran aceptación por parte del usuario, en el manejo de las interfaces y la facilidad de aprendizaje. Por el cual se otorga una buena puntuación, en esta fase es importante estar entre la puntuación de cuatro a cinco.

3.5.4. Evaluación de la eficiencia

TABLA 34 *Resumen de evaluación a la eficiencia de la aplicación*

REF	ATRIBUTO	PESO	VALORACIÓN
10	Comportamiento de tiempo	10	3
11	Utilización de los recursos	10	5

Fuente: Elaboración propia

$$\text{Eficiencia} = (10*3+10*5) / (10+10) = 4$$

De acuerdo al resultado obtenido, se establece que la aplicación, cumple con los requerimientos exigidos en un determinado tiempo (hace lo que tiene que hacer, no demora mucho tiempo su ejecución), por lo que el software obtiene la puntuación aceptable de acuerdo a la métrica de eficiencia.

3.5.5. Evaluación de Mantenimiento

TABLA 35 *Resumen de evaluación al mantenimiento*

REF	ATRIBUTO	PESO	VALORACIÓN
12	Capacidad de ser analizado	10	3
13	Cambiabilidad	10	5
14	Facilidad de Prueba	10	5

Fuente: Elaboración propia

$$\text{Mantenimiento} = (10 \cdot 3 + 10 \cdot 5 + 10 \cdot 5) / (10 + 10 + 10) = 4,33$$

De acuerdo al resultado obtenido, se establece que la aplicación cumple con los requerimientos mínimos exigidos para su mantenimiento y aceptabilidad. Por lo que se obtuvo una puntuación de 4.33, lo cual significa que es aceptable.

3.5.6. Evaluación de la portabilidad

TABLA 36 *Resumen de evaluación a la portabilidad*

REF	ATRIBUTO	PESO	VALORACIÓN
15	Adaptabilidad	10	2
16	Facilidad de Instalación	10	4

Fuente: Elaboración propia

$$\text{Portabilidad} = (10 \cdot 0 + 4 \cdot 10) / (10 + 10) = 2$$

3.5.7. Resultado general

TABLA 37 *Resultados generales*

ATRIBUTO	PESO
Funcionalidad	4,3
Fiabilidad	5
Usabilidad	4,75
Eficiencia	4
Mantenimiento	4,33
Portabilidad	2
Promedio	4,06

Fuente: Elaboración propia

Como se observa en la (TABLA 39), indica que la aplicación cumple con las Métricas de Calidad de la Norma ISO/IEC 9126, por lo que SÍ es pertinente utilizarlo.

4.CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES. -

El presente proyecto de Grado “**APLICACIÓN DE DETECCIÓN DE INFRACCIÓN DE TRANSITO, PARA EL RECONOCIMIENTO DE PLACAS DE AUTOMÓVILES, EN LOS PUNTOS DE SEMÁFOROS DE LA CIUDAD DE COBIJA**”, va dirigido específicamente para la Dirección Departamental de Transito y Seguridad Vial de Cobija – Pando, dependiente del comando policial Departamental de Pando.

Los resultados conseguidos dentro del presente proyecto son deseables y ponderables en ciertos entornos, como se dio a conocer a los alcances.

- Se estableció el análisis de la detección de infracción mediante la historia de Usuarios.
- Se diseñó la aplicación a partir de las herramientas que nos proporciona XP, además de usar algunas herramientas del modelado unificado (UML).
- Se desarrolló y se hizo las pruebas funcionales respectivas de la aplicación, ya que el alcance, no es igual al de un sistema de información, ligeramente grande, la prueba de desarrollo se realizó con las pruebas de caja negra.

Teniendo en cuenta las pruebas de caja negra que se desarrolló en la aplicación, para la detección de la infracción, se tiene un 100 % de acierto por parte de la aplicación cuando se hace el vehículo comete la infracción, el acierto de la detección de la placa es de un 80%, por último, la extracción de caracteres (OCR), tiene un 73% de éxito.

Las dificultades importantes en el desarrollo del proyecto están en encontrar las herramientas ideales para la visión por computador, y que hasta la actualidad es muy complejo la manipulación y desarrollo de sistemas o aplicación que tengan que ver con visión por computador.

Las funcionalidades que presenta el presente proyecto se enmarcan en hechos que se dan en la vida real, obtener (imagen a partir de un video o alguna cámara conectada a la computadora), detectar, procesar y guardar información sobre los vehículos infractores que tengan placas nacionales.

Dentro del contexto internacional este tipo de sistemas existen, pero es necesario tomar el contexto local ya que existen clasificadores de imágenes que solo pueden reconocer un tipo de matrícula a la vez.

El presente proyecto de grado, se ha conseguido culminar con ciertas deficiencias en cuanto al desarrollo, teniendo en cuenta que la visión por computador es una rama de la inteligencia artificial que está en constante desarrollo y que aún no se llega o se logra, emular el ojo humano a la perfección.

4.2. RECOMENDACIONES

Se sugiere tomar en cuenta las siguientes recomendaciones:

- Debido a la complejidad de la implementación y desarrollo de algoritmos para visión por computador, se debe efectuar mejoras en el reconocimiento de placas vehiculares.
- Desarrollar los algoritmos de visión por computador para que la aplicación pueda trabajar en todo tipo de ambiente (noche, lluvia, bajo sombra etc.).
- Integrar sistemas alternos (placa Arduino, sensor de distancia, funcionalidad en red) a la aplicación para su implementación.
- Obtener dispositivos de entrada de imágenes (cámara) de alta gama, para un mejor procesamiento de la imagen.
- Debido a que la aplicación solo se enfoca a la detección de infracción, reconocimiento de placas y guardado de imágenes y placa reconocida, se recomienda implementar una base de datos completa, donde se pueda registrar personas vehículos, accesos, etc.

5.BIBLIOGRAFÍA

Baggio, D. L. (2015). *OpenCV 3.0 Computer Vision*. BIRMINGHAM - MUMBAI: Packt Publishing Ltd.

Baggio, D. L. (2015). *OpenCv 3.0 Computer Vision With Java*. Mumbai : Packt Publishing Ltd.

Direccion Departamental de Transito Transporte y Seguridad Vial. (2018). *Manual de Funciones*. Cobija.

Dra. Nora La Serna Palomino, L. U. (2007). *Técnicas de Segmentación en Procesamiento Digital de Imágenes*. Peru.

Estado Plurinacional de Bolivia. (1973). *CODIGO DEL TRANSITO*. La Paz: Gaceta Oficial.

ESTEVE, J. M. (2008). La educacion en la sociedad del conocimiento. *La Tercera revolucion educativa*.

GÓMEZ, E. D. (JUNIO de 2009). Mejora del tráfico en un cruce regulado por semáforos, mediante un sistema basado en visión artificial. MADRID, España.

Gomez, G. (s.f.). *Vision Computacional*. Puebla Mexico.

Gonzales, J. F. (2017). *Introduccion a la metodologias Agiles*. Cataluña.

Grupo de Inteligencia Computacional. (2015). *Vision por computador*.

kent beck Cinthia Andres. (2012). *Extreme Programming Explained*. Boston: Addison Wesley.

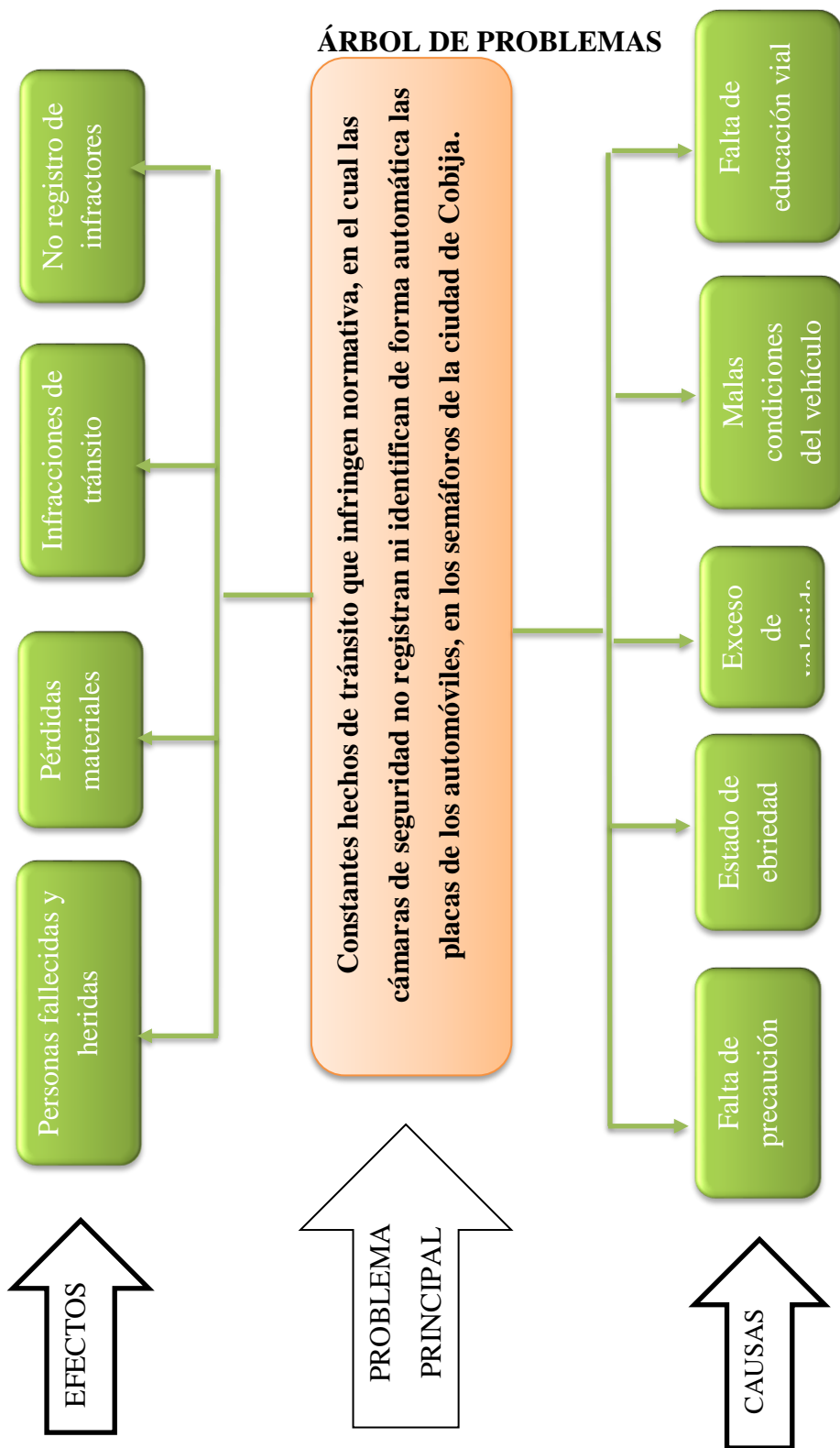
La Razon. (2015). *Colisiones son la mitad de los hechos de tránsito en Bolivia*. La Paz: La razon.

- La Razon. (2016). *Colisiones son la mitad de los hechos de tránsito en Bolivia*. La Paz: La razon.
- Marin, D. C. (2006). *El sistema optico del ojo y la vision binocular*. Madrid: Version Digital.
- Martin, M. (2004). *Tecnicas Clasicas de Segmentacion de Imagen*. Valladolid.
- Minardi, F. (2014). *Visi'on Artificial*. Asuncion.
- MINISTERIO DE EDUCACION. (s.f.). *EDUCACIÓN INCLUSIVA: ANATOMÍA Y FISIOLOGÍA OCULAR*.
- Montiel, J. L. (2010). *Reconocimiento de Placas Vehiculares*. Mexico.
- Nora La Serna Palomino, Ulises Román Concha¹. (2009). *Técnicas de Segmentación en Procesamiento Digital de*. Lima.
- Pressman, R. (2010). *Ingenieria de Software un enfoque practico*. Mexico: Mc Graw Hill.
- Prof. Dr Francisco Gomez Rodriguez, P. D. (2013). *Fundamentos de la Vision Artiicial*. Sevilla, España.
- Raul Igual, C. M. (2008). *Tutorial de OpenCV*. Mexico.
- RICARDO, R. R. (2012). *Hacia un esquema de localización y reconocimiento*. Mexico.
- Universidad Politecnica de Madrid. (2008). *Metricas Aplcadas a los Modelos de Calidad*. Valencia.

6.ANEXOS

ANEXO 1

ÁRBOL DE PROBLEMAS



ANEXO 2

EVALUACIÓN DEL SISTEMA DE ACUERDO A LA NORMA ISO/IEC 9126 POR PARTE DEL ADMINISTRADOR DE LA APLICACIÓN

VALOR	ESCALA
1	Muy bajo
2	Bajo
3	Medio
4	Alto
5	Muy Alto

REF.	ATRIBUTO	CALIFICACIÓN
FUNCIONALIDAD		
1	¿El software puede desempeñar las tareas requeridas en las historias de usuario?	5
2	¿Los resultados son los esperados?	5
3	¿La aplicación impide el acceso no autorizado?	3
CONFIABILIDAD		
4	¿Las entradas al sistema están debidamente validadas?	5
5	¿El software es capaz de manejar errores?	5
USABILIDAD		
6	¿La interfaz del usuario presenta buena estética para su utilización?	5
7	¿Puede el usuario aprender fácilmente la aplicación?	5
	¿Las entradas de datos están debidamente etiquetadas?	5
8	¿Es atractivo el diseño de la aplicación?	4
EFICIENCIA		

9	¿El sistema responde de manera eficiente en las peticiones que se realiza?	3
10	¿Toma menor tiempo para la búsqueda de información?	5
MANTENIMIENTO		
11	¿Las fallas Pueden ser fácilmente diagnosticadas?	3
12	¿Es fácil de realizar modificaciones en el sistema?	5
14	¿Son fáciles de validar las modificaciones?	5
PORTABILIDAD		
15	¿Es compatible con distintos sistemas operativos?	2
16	¿Es fácil la instalación de la aplicación?	4

ANEXO 3

CÓDIGO DE TRÁNSITO

Para desarrollar la presente aplicación se tomó en cuenta las disposiciones generales y normas que emana el código de tránsito vigente en Bolivia, a continuación, citaremos algunos artículos que nos ayudaran a entender las infracciones y otras disposiciones (Estado Plurinacional de Bolivia, 1973).

TÍTULO II DE LA CIRCULACIÓN

➤ **CAPÍTULO II DE LA CIRCULACIÓN DE LOS VEHÍCULOS**

- **Art. 57º.- CRUCE EN INTERSECCIONES.** Antes del cruce en intersecciones, los conductores disminuirán la velocidad de sus vehículos y en caso necesario los detendrán, dejando libre la franja de seguridad para el paso de peatones.

TÍTULO VI DE LAS FALTAS Y SANCIONES

➤ **CAPÍTULO I DE LAS INFRACCIONES**

- **Art. 139.- INFRACCIÓN:** Infracción llamada también transgresión o contravención, es el quebrantamiento de una o más reglas de tránsito.

TÍTULO II DE LA CIRCULACIÓN

➤ **CAPÍTULO II DE LA CIRCULACIÓN DE LOS VEHÍCULOS. –**

- **Artículo 20.- PROHIBICIÓN:** Se establecen las siguientes prohibiciones básicas para los vehículos en circulación:
 - Detener el vehículo en la franja de seguridad destinada al paso de los peatones.

➤ **Capítulo VII DE LAS SEÑALES**

- **Artículo 56.- FRANJAS DE PASO:** Nos dice que todos los lugares de paso destinados a los peatones, serán señalizados por medio de franjas demarcadas en la propia calzada (comúnmente llamado Paso de cebra).

TÍTULO V DE LOS DOCUMENTOS Y REGISTROS

CAPITULO V DE LAS PLACAS:

- **Artículo 123.- PLACAS:** La placa es una matrícula de identificación que debe tener obligatoriamente todo vehículo. El número de la placa debe coincidir con el del carnet de propiedad.
- **Artículo 124.- REQUISITO:** Ningún vehículo circulará sin placas.

ANEXO 4

CERTIFICACIÓN DEL PROYECTO DE GRADO DE LA DIRECCIÓN DEPARTAMENTAL DE TRANSITO TRANSPORTE Y SEGURIDAD VIAL



CERTIFICACIÓN

A quien corresponda:

EL SUSCRITO DIRECTOR DEPARTAMENTAL DE TRÁNSITO, TRANSPORTE Y SEGURIDAD VIAL DE PANDO, EN USO DE SUS ESPECÍFICAS FUNCIONES Y ATRIBUCIONES, EN CUANTO PUEDE Y EL DERECHO LE PERMITE:
CERTIFICA AL SEÑOR:


UNIV. RUDDY CHAO

C.I. 8407415 Exp. La Paz

RU. 11751

Que ha demostrado en oficinas de la Dirección De Tránsito que la aplicación de "DETECCIÓN DE INFRACCIÓN DE TRANSITO, PARA EL RECONOCIMIENTO DE PLACAS DE AUTOMÓVILES, EN LOS PUNTOS DE SEMÁFOROS DE LA CIUDAD DE COBIJA", cumple con los requisitos mínimos suscritos en los alcances del proyecto de grado. Detecta la placa, detecta la infracción, el reconocimiento de los caracteres depende mucho de la resolución y calidad de la imagen captada.

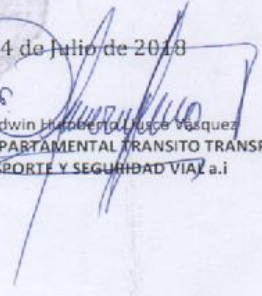
Es cuanto se CERTIFICO, para fines consiguientes.


Pol. Brandon Boris Aguilar Sandy
SECRETARIO DIR. DPTAL. DE TRANSITO, Y SEGURIDAD
VIAL PANDO TRANSPORTE

Vo Bo



Cobija, 04 de Julio de 2018


Srte. Edwin Humberto Lucero Vasquez
DIRECTOR DEPARTAMENTAL TRANSITO TRANSPORTE
Y SEGURIDAD VIAL a.i

ANEXO 5

SOLICITUDES A AUTORIDADES DE TRANSITO Y SEGURIDAD VIAL

COBIJA 12 DE JULIO DEL 2017

Señor:

Lic. Raunir Ribeiro Lozano

Responsable del proyecto de Seguridad Vial G.A.M.C

Presente. -

REF.: Solicitud de colaboración de pintado de paso peatonal

De mi consideración.

Mediante la presente primeramente saludar a su autoridad deseándole éxitos en las funciones que desempeña en favor de la comunidad Cobijeña.

El motivo de la presente, es poder solicitar a la Unidad de Seguridad Vial mediante su persona, pueda colaborarme con el pintado de dos pasos peatonales en punto estratégico de la ciudad , ya que mi persona está realizando un proyecto final “**APLICACIÓN DE DETECCIÓN DE INFRACCIÓN DE TRANSITO, PARA EL RECONOCIMIENTO DE PLACAS DE AUTOMÓVILES, EN LOS PUNTOS DE SEMÁFOROS DE LA CIUDAD DE COBIJA**” en la Universidad Amazónica de Pando, en tal sentido necesito grabar a los autos, para poder probar la aplicación que detecta la infracción de cruces de pasos peatonales y pase en semáforo rojo.

Sin otro particular y espera de su atención y colaboración me despido.

Atentamente.

Ruddy Chao
Estudiante

Señor:

Cnel. Alfredo Ramos

Director Departamental de Transito

Presente. -

REF.: Solicitud de colaboración y permiso de pintado de paso peatonal

De mi consideración.

Mediante la presente primeramente saludar a su autoridad deseándole éxitos en las funciones que desempeña en favor de la comunidad Pandina.

El motivo de la presente, es poder solicitar a la Dirección Departamental de Transito de mediante su persona, pueda concederme permiso si así fuese y la colaboración del pintado de dos pasos peatonales en punto estratégico de la ciudad , ya que mi persona está realizando un proyecto final “**APLICACIÓN DE DETECCIÓN DE INFRACCIÓN DE TRANSITO, PARA EL RECONOCIMIENTO DE PLACAS DE AUTOMÓVILES, EN LOS PUNTOS DE SEMÁFOROS DE LA CIUDAD DE COBIJA**” en la Universidad Amazónica de Pando, en tal sentido necesito grabar a los autos, para poder probar la aplicación que detecta la infracción de cruces de pasos peatonales y pase en semáforo rojo.

Sin otro particular y espera de su atención y colaboración me despido.

Adjunto fotocopia de matrícula.

Atentamente.

Ruddy Chao
Estudiante

Señor.

Coronel. Marco Antonio Cortez Castedo

Comandante Departamental Policial Pando

Presente.-

Ref. Solicitud de datos estructurales de la Dirección Departamental Tránsito

De mi consideración:

Mediante la presente primeramente saludar a su autoridad deseándole éxitos en las funciones.

El motivo de la presente, es para solicitar su persona, pueda facilitarme los siguientes datos de la Dirección Departamental de Tránsito.


1. Marco legal por el cual está funcionando la dirección (Resolución de estado, decreto supremo de creación).
2. Resolución Administrativa 305 2012.
3. Organigrama de la Dirección de Tránsito.
4. Manual de funciones.

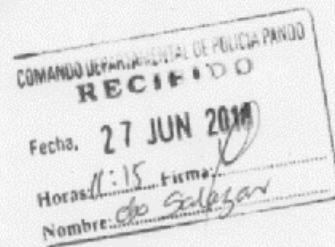
Es menester mencionar que ya se hizo una solicitud a La dirección Departamental de Tránsito, los mismo dieron información pero no completa, es así que me dijeron que me apersono a su persona para mayor información.

Ya que mi persona está realizando un proyecto de grado final denominado: **"APLICACIÓN DE DETECCIÓN DE INFRACCIÓN DE TRANSITO, PARA EL RECONOCIMIENTO DE PLACAS DE AUTOMÓVILES, EN LOS PUNTOS DE SEMÁFOROS DE LA CIUDAD DE COBIJA"**, en la Universidad Amazónica de Pando, en tal sentido, necesito la información requerida, para poder sustentar dicho proyecto.

Sin otro particular y en espera de su atención y colaboración me despido con las consideraciones respectivas.

ATENTAMENTE


Ruddy Chao
Estudiante



ANEXO 6

CERTIFICACIÓN DEL PROYECTO DE GRADO DE LA DIRECCIÓN DEPARTAMENTAL DE TENSITO TRANSPORTE



CERTIFICACIÓN

A quien corresponda:

EL SUSCRITO DIRECTOR DEPARTAMENTAL DE TRÁNSITO, TRANSPORTE Y SEGURIDAD VIAL DE PANDO, EN USO DE SUS ESPECÍFICAS FUNCIONES Y ATRIBUCIONES, EN CUANTO PUEDE Y EL DERECHO LE PERMITE:
CERTIFICA AL SEÑOR:


UNIV. RUDDY CHAO

C.I. 8407415 Exp. La Paz

RU. 11751

Que ha demostrado en oficinas de la Dirección De Tránsito que la aplicación de "DETECCIÓN DE INFRACCIÓN DE TRANSITO, PARA EL RECONOCIMIENTO DE PLACAS DE AUTOMÓVILES, EN LOS PUNTOS DE SEMÁFOROS DE LA CIUDAD DE COBIJA", cumple con los requisitos mínimos suscritos en los alcances del proyecto de grado. Detecta la placa, detecta la infracción, el reconocimiento de los caracteres depende mucho de la resolución y calidad de la imagen captada.

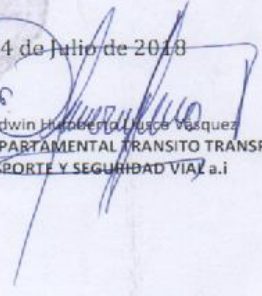
Es cuanto se CERTIFICO, para fines consiguientes.


Pol. Brandon Boris Aguilar Sandy
SECRETARIO DIR. DPTAL. DE TRANSITO, Y SEGURIDAD
VIAL PANDO TRANSPORTE

Vo Bo



Cobija, 04 de Julio de 2018


Srte. Edwin Humberto Lucio Vasquez
DIRECTOR DEPARTAMENTAL TRANSITO TRANSPORTE
TRANSPORTE Y SEGURIDAD VIAL a.i