



**UNIVERSIDAD
NACIONAL
DE LOJA**



Facultad de la Energía, las Industrias y los Recursos Naturales No Renovables

CARRERA DE INGENIERÍA EN SISTEMAS

“SISTEMA PARA DETECCIÓN Y CONTEO VEHICULAR APLICANDO TÉCNICAS DE VISIÓN ARTIFICIAL”

TESIS PREVIA LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN
SISTEMAS.

Autor:

Fabricio Roberto Sisalima Ortega

Director:

Ing. Luis Roberto Jácome Galarza, Mg. Sc

**LOJA-ECUADOR
2018**

CERTIFICACIÓN DEL DIRECTOR

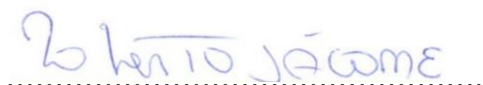
Loja, 31 mayo del 2018

Ing. Luis Roberto Jácome Galarza, Mg. Sc.

DOCENTE DE LA CARRERA DE INGENIERÍA EN SISTEMAS, DE LA FACULTAD DE LA ENERGÍA, LAS INDUSTRIAS Y LOS RECURSOS NATURALES NO RENOVABLES DE LA UNIVERSIDAD NACIONAL DE LOJA.

CERTIFICA:

Que el Sr. Fabricio Roberto Sisalima Ortega, egresado de la carrera de Ingeniería en Sistemas ha trabajado bajo mi tutoría el presente trabajo de titulación, previo a la obtención del título de Ingeniero en Sistemas, cuyo tema versa sobre “**SISTEMA PARA DETECCIÓN Y CONTEO VEHICULAR APLICANDO TÉCNICAS DE VISIÓN ARTIFICIAL**”, el mismo que ha sido dirigido, orientado y discutido bajo mi asesoramiento y cumple con la reglamentación pertinente, así como lo programado en el plan del proyecto, razones por las cuales reúne la suficiente validez técnica y práctica, por consiguiente autorizo su presentación y sustentación.



Ing. Luis Roberto Jácome Galarza, Mg. Sc.

DIRECTOR DEL TRABAJO DE TITULACIÓN

AUTORÍA

Yo, **FABRICIO ROBERTO SISALIMA ORTEGA**, declaro ser autor del presente trabajo de titulación y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales, por el contenido de la misma.

Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mi trabajo de titulación en el Repositorio Institucional-Biblioteca Virtual.

Firma:


.....

Cédula:

1105012866

Fecha:

30-07-2018

CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.

Yo, **FABRICIO ROBERTO SISALIMA ORTEGA**, declaro ser autor del trabajo de titulación que versa: **“SISTEMA PARA DETECCIÓN Y CONTEO VEHICULAR APLICANDO TÉCNICAS DE VISIÓN ARTIFICIAL”**, como requisito para optar al grado de: **INGENIERO EN SISTEMAS**; autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con los cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia de la tesis que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, a los treinta días del mes de julio del dos mil dieciocho.

Firma:


Autor: Fabricio Roberto Sisalima Ortega

Cedula: 1105012866

Dirección: Loja (José de San Martín y José Martí)

Correo Electrónico: frsisalimao@unl.edu.ec

Celular: 0986647089

DATOS COMPLEMENTARIOS

Director de Tesis: Ing. Luis Roberto Jácome Galarza, Mg. Sc

Tribunal de Grado: Ing. Edison Leonardo Coronel Romero, Msc.

Ing. Oscar Miguel Cumbicus Pineda, Msc.

Ing. José Luis Granda Sivisapa, Msc.

AGRADECIMIENTOS

El presente trabajo de tesis primeramente me gustaría agradecer a Dios por haberme permitido cumplir esta importante meta.

A la universidad Nacional de Loja por darme la oportunidad de prepararme en sus aulas y permitirme ser un profesional.

A mi director de tesis Luis Roberto Jácome por sus esfuerzos y dedicación, quien con su conocimiento y experiencia me supo guiar por el camino correcto en el desarrollo del proyecto.

Son muchas las personas a las que me gustaría agradecer por su apoyo y consejos, a todos ellos muchas gracias.

El Autor

DEDICATORIA

Este proyecto está dedicado en primer lugar a mi padre Alcisar Sisalima y mi madre María Ortega por el apoyo incondicional para que pueda culminar mis estudios universitarios, a mis hermanas Lady Sisalima y Madeleine Sisalima por ser una motivación para que siga preparándome cada día más, además este proyecto está dedicado a todos los docentes que en algún momento compartieron sus conocimientos conmigo y sea eso una parte fundamental para la realización de este proyecto.

El Autor

Índice de Contenidos

CERTIFICACIÓN DEL DIRECTOR.....	II
AUTORÍA.....	III
CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DEL AUTOR.	IV
AGRADECIMIENTOS.....	V
DEDICATORIA	VI
Índice de Contenidos	VII
Índice de Figuras	X
Índice de Figuras	XIII
A. TÍTULO	17
B. RESUMEN	18
Summary	19
C. INTRODUCCIÓN	20
D. REVISIÓN DE LITERATURA.....	22
1. Ingeniería de Tránsito	22
1.1. Estudios de Volumen	22
2. Inteligencia Artificial.....	26
2.1. Ramas que componen la Inteligencia Artificial	28
2.2. Visión por Computador.....	32
3. Arquitectura y Diseño de Aplicaciones Web	35
3.1. Sistemas web.....	35
3.2. Aplicaciones web	35
4. Tecnologías empleadas para el desarrollo del software	39
4.1. Backend de la Aplicación web.....	39
5. Librerías para el procesamiento de Imágenes.....	42
5.1. OpenCV(Open Source Computer Vision).....	42
5.2. Torch3vision.....	43

5.3.	LTI-Lib.....	44
5.4.	Comparativa de Librerías de Procesamiento de Imágenes	45
6.	Algoritmos de Detección de objetos	47
6.1.	Análisis de movimiento.....	47
6.2.	Descriptor de características	49
7.	Metodología de desarrollo de software.....	51
7.1.	Metodología XP.....	51
7.2.	Comparativa de Metodologías de Desarrollo.....	53
E.	MATERIALES Y MÉTODOS	58
1.	Métodos	58
2.	Técnicas.....	58
3.	Metodología de desarrollo de software.....	59
F.	RESULTADOS.....	60
1.	PRIMERA FASE: PLANIFICACIÓN	60
1.1.	Especificación de requerimientos	60
1.2.	Módulos de las aplicaciones.....	62
1.3.	Especificación de historias de usuario.....	63
1.4.	Estimación de historias de usuario	72
2.	SEGUNDA FASE: DISEÑO.....	73
2.1.	Arquitectura del sistema.....	74
2.2.	Diagrama de clases	75
2.3.	Modelo entidad relación	76
2.4.	Tarjetas CRC	76
2.5.	Prototipos de las aplicaciones	79
3.	TERCERA FASE: CODIFICACIÓN	85
3.1.	Estándares de programación	85
3.2.	Descripción de herramientas utilizadas para la implementación del software	86

3.3.	Codificación de los módulos del sistema	87
4.	FASE CUARTA: PRUEBAS	96
4.1.	Pruebas de Aceptación	96
4.2.	Pruebas de Rendimiento.....	107
G.	DISCUSIÓN	116
1.	Desarrollo de la Propuesta Alternativa	116
2.	Valoración Técnica - Económica - Ambiental	118
2.1.	Recursos humanos	118
2.2.	Recursos materiales	119
2.3.	Recursos Técnicos y Tecnológicos	119
2.4.	Resumen de Presupuesto.....	120
H.	CONCLUSIONES.....	121
I.	RECOMENDACIONES	122
J.	REFERENCIAS.....	123
K.	ANEXO.....	126

Índice de Figuras

Figura 1. Ecuación para medir el flujo vehicular	24
Figura 2. Ecuación para calcular la velocidad en un punto	25
Figura 3. Ecuación para calcular la velocidad media en el espacio.....	25
Figura 4. Imagen original	48
Figura 5. Algoritmo BackgroundSubtractorMOG	49
b Figura 6. Algoritmo BackgroundSubtractorMOG2	49
Figura 7. Algoritmo BackgroundSubtractorGMG	49
Figura 8. Arquitectura del sistema.....	74
Figura 9. Diagrama de clases	75
Figura 10. Modelo entidad relación	76
Figura 11. Pantalla principal del software de conteo vehicular	79
Figura 12. Visualización de video enviado por cámara ip.....	80
Figura 13. Pantalla de configuraciones de ajuste de línea	80
Figura 14. Pantalla para ajustes de clasificación por tamaño de vehículos.	81
Figura 15. Pantalla principal de inicio de sistema web.	82
Figura 16. Login del sistema web.....	82
Figura 17. Tabla de contadores registrados.....	83
Figura 18. Formulario de registro de contador.	84
Figura 19. Reporte con presentación de datos en una tabla	84
Figura 20. Pantalla de cálculo de volumen de tránsito.	85

Figura 21. Código de la interfaz de configuración	87
Figura 22. Código de la visualización de imagen	88
Figura 23. Código de la configuración de posición de líneas de conteo	89
Figura 24. Código de la configuración de la fuente de video	90
Figura 25. Código de la detección de vehículos.....	90
Figura 26. Código del rastreo de vehículo en la imagen	91
Figura 27. Código del conteo de vehículos según línea de conteo.....	92
Figura 28. Código de registrar contador.....	92
Figura 29. Código de editar contador.....	93
Figura 30. Código de eliminar contador	93
Figura 31. Código de visualizar contador	93
Figura 32. Código de generar reporte	94
Figura 33. Código de exportar reporte	95
Figura 34. Opción en menú para módulo empresas	145
Figura 35. Pantalla de cálculo de volumen de tránsito	145
Figura 36. Formulario de registro de empresa	146
Figura 37. Pantalla con detalle de empresa	146
Figura 38. Formulario de edición de empresa.....	147
Figura 39. Menú de opción de asignación de contadores a empresas.....	147
Figura 40. Lista de contadores asignados a empresas	148
Figura 41. Formulario de asignación de contador a empresa.....	148

Figura 42. Vista con detalle de contador asignado a empresa.	149
Figura 43. Formulario de asignación de contador	149
Figura 44. Vista de menú de usuarios.....	150
Figura 45. Lista de usuarios registrados	150
Figura 46. Formulario de registro de usuarios.....	151
Figura 47. Formulario de edición de usuarios	151
Figura 48. Vista con detalle de usuario registrado	152
Figura 49. Vista con menú de usuarios empresa	152
Figura 50. Lista de usuarios asignados a empresas	153
Figura 51. Formulario de asignación de usuarios a empresas	153
Figura 52. Formulario de edición de usuarios a empresas.....	154
Figura 53. Vista con detalle de contador asignado a empresa.	154

Índice de Tablas

TABLA I. DISPOSITIVOS PARA REALIZAR ESTUDIOS DE VOLUMEN UTILIZANDO EL MÉTODO AUTOMÁTICO	23
TABLA III. COMPARATIVA DE LIBRERÍAS DE PROCESAMIENTO DE IMÁGENES	45
TABLA IV. COMPARATIVA DE METODOLOGÍAS DE DESARROLLO	54
TABLA V. LISTA DE REQUERIMIENTOS FUNCIONALES	60
TABLA VI. LISTA DE REQUERIMIENTOS NO FUNCIONALES	61
TABLA VII. TIPOS DE USUARIOS DE LAS APLICACIONES	62
TABLA VIII. HISTORIA DE USUARIO: REGISTRAR USUARIO	64
TABLA IX. HISTORIA DE USUARIO: EDITAR USUARIO	64
TABLA X. HISTORIA DE USUARIO: ELIMINAR USUARIO	65
TABLA XI. HISTORIA DE USUARIO: VISUALIZAR USUARIO	65
TABLA XII. HISTORIA DE USUARIO: REGISTRAR CONTADOR	66
TABLA XIII. HISTORIA DE USUARIO: EDITAR CONTADOR	66
TABLA XIV. HISTORIA DE USUARIO: ELIMINAR CONTADOR	67
TABLA XV. HISTORIA DE USUARIO: VISUALIZAR CONTADOR	67
TABLA XVI. HISTORIA DE USUARIO: REPORTE DE CONTEO	68
TABLA XVII. HISTORIA DE USUARIO: EXPORTAR REPORTE	68
TABLA XVIII. HISTORIA DE USUARIO: VISUALIZACIÓN DE IMAGEN	69
TABLA XIX. HISTORIA DE USUARIO: CONFIGURACIÓN DE LINEAS DE CONTEO	69

TABLA XX. HISTORIA DE USUARIO: CONFIGURACIÓN DE LA FUENTE DE VIDEO	70
TABLA XXI. HISTORIA DE USUARIO: DETECCIÓN DE VEHÍCULOS	70
TABLA XXII. HISTORIA DE USUARIO: RASTREO DE VEHÍCULOS EN LA IMAGEN	71
TABLA XXIII. HISTORIA DE USUARIO: CONTEO DE VEHÍCULOS SEGUN LINEA DE CONTEO	71
TABLA XXIV. HISTORIA DE USUARIO: INICIO DEL SOFTWARE CON EL CONTEO DEL SISTEMA	72
TABLA XXV. ESTIMACIÓN DE HISTORIAS DE USUARIO	72
TABLA XXVI. TARJETA CRC: PROCESADOR	77
TABLA XXVII. TARJETA CRC: CAMARA	77
TABLA XXVIII. TARJETA CRC: TRACKER	77
TABLA XXIX. TARJETA CRC: CONEXION SERVER	77
TABLA XXX. TARJETA CRC: USUARIO	78
TABLA XXXI. TARJETA CRC: EMPRESA	78
TABLA XXXII. TARJETA CRC: CONTADOR	78
TABLA XXXIII. TARJETA CRC: CONTEO	78
TABLA XXXIV. PRUEBA DE ACEPTACIÓN: REGISTRAR USUARIO	96
TABLA XXXV. PRUEBA DE ACEPTACIÓN: EDITAR USUARIO	97
TABLA XXXVI. PRUEBA DE ACEPTACIÓN: ELIMINAR USUARIO	98
TABLA XXXVII. PRUEBA DE ACEPTACIÓN: VISUALIZAR USUARIO	98
TABLA XXXVIII. PRUEBA DE ACEPTACIÓN: REGISTRAR CONTRADOR	99

TABLA XXXIX. PRUEBA DE ACEPTACIÓN: EDITAR CONTADOR	99
TABLA XL. PRUEBA DE ACEPTACIÓN: ELIMINAR CONTADOR.....	100
TABLA XLI. PRUEBA DE ACEPTACIÓN: VISUALIZAR CONTADOR.....	101
TABLA XLII. PRUEBA DE ACEPTACIÓN: GENERAR REPORTE DE CONTEO	101
TABLA XLIII. PRUEBA DE ACEPTACIÓN: EXPORTAR REPORTE DE CONTEO ..	102
TABLA XLIV. PRUEBA DE ACEPTACIÓN: VISUALIZACIÓN DE IMAGEN.....	103
TABLA XLV. PRUEBA DE ACEPTACIÓN: CONFIGURAR LINEA DE CONTEO	103
TABLA XLVI. PRUEBA DE ACEPTACIÓN: CONFIGURAR FUENTE DE VIDEO.....	104
TABLA XLVII. PRUEBA DE ACEPTACIÓN: DETECTAR VEHÍCULOS.....	105
TABLA XLVIII. PRUEBA DE ACEPTACIÓN: RASTREO DE VEHÍCULOS EN LA IMAGEN	105
TABLA XLIX. PRUEBA DE ACEPTACIÓN: CONTEO DE VEHÍCULOS	106
TABLA L. VERIFICACIÓN DEL CUMPLIMIENTO DE REQUERIMIENTOS FUNCIONALES	106
TABLA LI. INFORMACIÓN DEL VIDEO DE PRUEBA 1 PARA LA DETECCIÓN DE VEHÍCULOS.....	107
TABLA LII. MATRIZ DE CONFUSIÓN PARA EL PRIMER VIDEO DE DETECCIÓN DE VEHÍCULOS.....	108
TABLA LIII. INFORMACIÓN DEL VIDEO DE PRUEBA 2 PARA LA DETECCIÓN DE VEHÍCULOS.....	108
TABLA LIV. MATRIZ DE CONFUSIÓN PARA EL PRIMER VIDEO DE DETECCIÓN DE VEHÍCULOS.....	109
TABLA LVI. MATRIZ DE CONFUSIÓN PARA EL PRIMER VIDEO DE DETECCIÓN DE VEHÍCULOS.....	110

TABLA LVII. INFORMACIÓN DEL VIDEO DE PRUEBA 4 PARA CONTEO DE VEHÍCULOS.....	110
TABLA LVIII. MATRIZ DE CONFUSIÓN PARA EL PRIMER VIDEO DE DETECCIÓN DE VEHÍCULOS.....	111
TABLA LX. MATRIZ DE CONFUSIÓN PARA EL PRIMER VIDEO DE DETECCIÓN DE VEHÍCULOS.....	112
TABLA LXI. INFORMACIÓN DEL VIDEO DE PRUEBA 6 PARA CONTEO DE VEHÍCULOS.....	112
TABLA LXII. MATRIZ DE CONFUSIÓN PARA EL PRIMER VIDEO DE DETECCIÓN DE VEHÍCULOS.....	113
TABLA LXIII. INFORMACIÓN DEL VIDEO DE PRUEBA 7 PARA CONTEO DE VEHÍCULOS.....	113
TABLA LXIV. MATRIZ DE CONFUSIÓN PARA EL PRIMER VIDEO DE DETECCIÓN DE VEHÍCULOS.....	114
TABLA LXV. INFORMACIÓN DEL VIDEO DE PRUEBA 8 PARA CONTEO DE VEHÍCULOS.....	114
TABLA LXVI. MATRIZ DE CONFUSIÓN PARA EL PRIMER VIDEO DE DETECCIÓN DE VEHÍCULOS.....	115
TABLA LXVII. RECURSOS HUMANOS.....	118
TABLA LXVIII. RECURSOS MATERIALES	119
TABLA LXIX. RECURSOS TÉCNICOS Y TECNOLÓGICOS	119
TABLA LXX. RESUMEN DE PRESUPUESTO	120

A.TÍTULO

“SISTEMA PARA DETECCIÓN Y CONTEO VEHICULAR
APLICANDO TÉCNICAS DE VISIÓN ARTIFICIAL”

B.RESUMEN

El presente trabajo de titulación consiste en el desarrollo de un software de conteo vehicular aplicando técnicas de visión por computador. Esta aplicación beneficia en gran medida al área de tránsito para la toma de decisiones en base a los datos recolectados por el software, demostrando con esto la factibilidad hoy en día del desarrollo de software de procesamiento de imágenes para aplicaciones en entornos reales. Se presenta los algoritmos posibles a utilizar y la elección de la mejor alternativa para la solución de este problema en particular.

Para el desarrollo del trabajo de titulación se utilizaron métodos de investigación que permitieron identificar el problema del proyecto y brindar una solución óptima al mismo, se aplicaron técnicas como la entrevista para la recolección de información con el cliente y la bibliográfica para sustentar la base teórica del proyecto y finalmente la metodología de desarrollo XP que permitió llevar un orden durante el proceso de desarrollo del software.

El software de conteo fue desarrollado con el lenguaje C++ utilizando el framework Qt, la librería de visión por computador seleccionada fue Opencv misma que cuenta con las implementaciones de los algoritmos a emplear en el proceso de detección y conteo de vehículos.

El uso de los métodos, técnicas y herramientas expuestas anteriormente permitieron dar cumplimiento a los objetivos planteados en el proyecto, dando como resultado la construcción de un software que cumple con las necesidades solicitadas por el cliente.

Summary

The present work of certification consists in the development of a software of counting vehicular applying techniques of vision by computer. This application greatly benefits the transit area for decision making based on the data collected by the software, demonstrating with this the feasibility today of the development of image processing software for applications in real environments. It presents the possible algorithms to be used and the choice of the best alternative for the solution of this particular problem.

For the development of the titling work, research methods were used to identify the problem of the project and provide an optimal solution to it, techniques were applied such as the interview for the collection of information with the client and the bibliography to support the theoretical basis of the project. project and finally the XP development methodology that allowed to keep an order during the software development process.

The counting software was developed with the C ++ language using the Qt framework, the computer vision library selected was Opencv itself, which has the implementations of the algorithms to be used in the vehicle detection and counting process.

The use of the methods, techniques and tools previously exposed allowed to fulfill the objectives set out in the project, resulting in the construction of a software that meets the needs requested by the client.

C.INTRODUCCIÓN

Con el avance tecnológico en los últimos años las aplicaciones de visión por computador son cada día más viables para su desarrollo como productos a implementarse en entornos reales, algunas áreas para estas aplicaciones son la vigilancia, control de calidad de productos y el control de tráfico, entre otras.

Antes que estas aplicaciones fueran implementadas las tareas que hoy se encuentran automatizadas o se están automatizando eran realizadas por humanos o que simplemente no era factible su realización.

El conteo vehicular es una tarea muy utilizada en el área de control de tránsito, ya que es en base a estos datos que las personas encargadas toman decisiones para favorecer la movilidad en el entorno que estos se encuentran, un ejemplo puede ser la distribución de semáforos en la ciudad, al automatizar esta tarea con software se consigue una gran cantidad de beneficios como pueden ser ahorro de recursos económicos, calidad en los datos recolectados y uno de los más importantes beneficios que es la toma de decisiones efectiva. Es por ello que se ha propuesto el desarrollo del presente trabajo de titulación cuyo objetivo principal es “Desarrollar un sistema que permita la detección y conteo de vehículos aplicando técnicas de visión artificial”.

Para dar cumplimiento al objetivo principal se establecieron tres objetivos específicos, los cuales son: Analizar los diferentes algoritmos de detección y reconocimiento de objetos, diseñar e implementar el sistema para la detección y conteo de vehículos, y finalmente realizar y ejecutar un plan de pruebas que permitan evaluar al sistema en cada uno de sus módulos.

De acuerdo al reglamento establecido por la Universidad Nacional de Loja el Trabajo de Titulación se encuentra estructurado de la siguiente manera:

RESUMEN explica una breve síntesis de lo que se trata el proyecto;

INTRODUCCIÓN indica la importancia del tema, el aporte y la estructura del trabajo de tesis,

REVISIÓN LITERARIA contiene todos los fundamentos teóricos para el desarrollo de todos los componentes del proyecto;

MATERIALES Y MÉTODOS se detalla las técnicas de recolección de información, los

métodos de investigación y la metodología de software empleados para la elaboración del trabajo de titulación;

RESULTADOS se incluyen todos los documentos obtenidos y realizados en cada una de las fases establecidas por la metodología de desarrollo XP.;

DISCUSIÓN se detalla la evaluación de los resultados obtenidos y se justifica el contexto en el que fue desarrollado el trabajo;

CONCLUSIONES Y RECOMENDACIONES se las obtiene del desarrollo del trabajo.

D.REVISIÓN DE LITERATURA

1. Ingeniería de Tránsito

La ingeniería de tránsito es un área de la ingeniería civil encargada de realizar el estudio de volúmenes, velocidad y capacidades de una vía y todo aquello concerniente al movimiento de los vehículos y peatones. [1] [2]

Estos estudios de tránsito pueden agruparse en 3 categorías:

- **Inventarios:** Corresponde a toda la información existente de una vía tales como anchos de carriles, señalizaciones, espacios para parqueo. Esta información debe ser actualizada de manera periódica.
- **Estudios Administrativos:** son todos aquellos registros que tienen las oficinas de gobierno, correspondiente a encuestas, mediciones de campo, fotografías, entre otros. Estos datos permiten alimentar los inventarios.
- **Estudios Dinámicos:** estos estudios son los encargados de recoger toda la información de las condiciones de operación de una vía. Incluye los estudios de velocidad, volumen de tránsito, tiempos de viaje, parqueaderos y choques. [1]

1.1. Estudios de Volumen

Los estudios de volumen de tránsito se realizan con el objetivo de recolectar información sobre el número de vehículos o peatones que pasa por un punto durante un periodo de tiempo. Estos intervalos varían desde 15 minutos hasta un año. Adicionalmente a la cantidad de datos recolectados, estos también pueden ser clasificados en subcategorías como: movimiento direccional, ocupación, clasificación de los vehículos o edad de los peatones. [1] [3] [4] [5]

Los estudios de volumen se clasifican en:

- **Tránsito Promedio Diario (TPD):** es el número total de vehículos que pasan durante un día.
- **Tránsito Promedio Diario Semanal (TPDS):** es el número de vehículos que pasan durante una semana.

- **Tránsito Promedio Diario Mensual (TPDM):** es el número de vehículos que pasan durante un mes.
- **Tránsito Promedio Diario Anual (TPDA):** es el número de vehículos que pasan durante un año.
- Volumen de la hora pico (VHP)
- Clasificación de vehículos (CV) [1] [3] [4] [2]

1.1.1. Métodos para realizar estudios de volumen

Para realizar estudios de volumen se pueden emplear métodos manuales o automáticos.

- **Métodos manuales:** se realiza el conteo de vehículos mediante un contador o el uso de una planilla, por una o varias personas ubicadas en una vía o intersección específica.
- **Métodos automáticos:** para realizar este tipo de conteo se utilizan los siguientes dispositivos con los cuales también se realiza estudios de velocidad. [1] [3] [5]

TABLA I. DISPOSITIVOS PARA REALIZAR ESTUDIOS DE VOLUMEN UTILIZANDO EL MÉTODO AUTOMÁTICO

Método	Conteo	Velocidad	Clasificación	Múltiples carriles y zonas de detección
Circuitos inductores	X	X	X	
Magnetómetros	X	X		
Radar de microondas	X	X	X	X
Infrarrojo activo	X	X	X	X
Infrarrojo pasivo	X	X		
Ultrasonido	X	X		
Acústicos	X	X		X

Procesamiento de Imágenes	X	X	X	X
---------------------------	---	---	---	---

1.2. Principios del Flujo de Tránsito

La teoría del flujo del tránsito busca establecer todas las relaciones matemáticas entre los principales elementos del flujo vehicular como son el flujo, la densidad y la velocidad. Estas relaciones son utilizadas por los ingenieros de tránsito para el diseño, planeación y operación de las vías. También es utilizada para la simulación en las cuales se implementan algoritmos con el objetivo de determinar los comportamientos de las vías las cuales se ven afectadas por factores como accidentes, tiempo de viajes, vehículos estacionados por fallas. Además de analizar otros factores como la contaminación ambiental y consumo de gasolina.

1.2.1. Principales elementos del flujo de tránsito

Elementos que componen los principios fundamentales del flujo de tránsito:

- **Flujo (q):** se define como la relación entre la cantidad de vehículos que pasan por una sección durante un intervalo de tiempo (menor a 1 hora). Está representada mediante la siguiente ecuación:

$$q = \frac{n \times 3600}{T} [\text{veh/h}]$$

Figura 1. Ecuación para medir el flujo vehicular

Donde:

q: flujo horario equivalente.

n: número de vehículos

T: tiempo en segundos

- **Densidad (k):** se define como el número de vehículos sobre una longitud determinada de una vía en un instante de tiempo. La longitud es generalmente expresada como 1 milla o 1 kilómetro. Las unidades en que están

representadas corresponden a vehículos por kilómetro (veh/km). [1] [4] [6]

- **Velocidad (u):** se define como la distancia recorrida por un vehículo en un intervalo de tiempo. Se representan generalmente en millas por hora (mi/h) o kilómetros por hora (km/h).
- **Velocidad media en el tiempo (ut):** es la media aritmética de las velocidades de un punto. Se calculada mediante la siguiente ecuación:

$$\bar{u}_t = \frac{1}{n} \sum_{i=1}^n u_i$$

Figura 2. Ecuación para calcular la velocidad en un punto

Donde:

n: número total de vehículos observados

ui : velocidad del iésimo vehículo.

- **Velocidad media en el espacio (Us):** es la media armónica de las velocidades de un punto. Es calculada mediante la ecuación:

$$\bar{u}_s = \frac{n}{\sum_{i=1}^n (1/u_i)}$$

Figura 3. Ecuación para calcular la velocidad media en el espacio

Donde:

n: número de vehículos

ui: Velocidad del iésimo vehículo. [1]

2. Internet de las Cosas

El Internet de las cosas es un tema emergente de importancia técnica, social y económica. En este momento se están combinando productos de consumo, bienes duraderos, automóviles, componentes industriales y de servicios públicos, sensores y otros objetos de uso cotidiano con conectividad a Internet y potentes capacidades de análisis de datos que prometen transformar el modo en que vivimos.

Las proyecciones del impacto de la IoT sobre Internet y la economía son impresionantes: hay quienes anticipan que en el año 2025 habrá hasta cien mil millones de dispositivos conectados a la IoT y que su impacto será de US\$ 11.000.000.000.000. Sin embargo, la Internet de las Cosas también plantea importantes desafíos que podrían dificultar la realización de sus potenciales beneficios. Noticias sobre ataques a dispositivos conectados a Internet, el temor a la vigilancia y las preocupaciones relacionadas con la privacidad ya han captado la atención del público. Los desafíos técnicos siguen allí, pero además están surgiendo nuevos desafíos de políticas, jurídicos y de desarrollo [7].

2.1. Historia

El término “Internet de las Cosas” (IoT) fue empleado por primera vez en 1999 por el pionero británico Kevin Ashton para describir un sistema en el cual los objetos del mundo físico se podían conectar a Internet por medio de sensores. Ashton acuñó este término para ilustrar el poder de conectar a Internet las etiquetas de identificación por radiofrecuencia (RFID) que se utilizaban en las cadenas de suministro corporativas para contar y realizar un seguimiento de las mercancías sin necesidad de intervención humana.

Hoy en día, el término Internet de las Cosas se ha popularizado para describir escenarios en los que la conectividad a Internet y la capacidad de cómputo se extienden a una variedad de objetos, dispositivos, sensores y artículos de uso diario [7].

2.2. Concepto

El «Internet de las Cosas» (IoT) hace referencia, como se ha adelantado, a una tecnología basada en la conexión de objetos cotidianos a Internet que intercambian,

agregan y procesan información sobre su entorno físico para proporcionar servicios de valor añadido a los usuarios finales. También reconoce eventos o cambios, y tales sistemas pueden reaccionar de forma autónoma y adecuada. Su finalidad es, por tanto, brindar una infraestructura que supere la barrera entre los objetos en el mundo físico y su representación en los sistemas de información [8].

2.3. Tendencias tecnológicas y de mercado que están impulsando la IoT

Conectividad ubicua: La conectividad generalizada, de bajo costo y alta velocidad, sobre todo a través de servicios y tecnología inalámbricos con y sin licencia, hace que casi todo sea “conectable “.

Miniaturización: Los avances logrados en la fabricación permiten incorporar tecnología de cómputo y comunicaciones de vanguardia en objetos muy pequeños. Junto con una mayor economía en la capacidad de cómputo, esto ha impulsado el desarrollo de sensores pequeños y de bajo costo que a su vez impulsan muchas aplicaciones de la IoT.

Adopción generalizada de redes basadas en el protocolo IP: El protocolo IP se ha convertido en el estándar dominante para la creación de redes y ofrece una plataforma bien definida y ampliamente implementada en software y herramientas que se pueden incorporar en una variedad de dispositivos de forma fácil y económica.

Avances en el análisis de datos: La existencia de nuevos algoritmos y el rápido aumento de la potencia de cálculo, el almacenamiento de datos y los servicios en la nube permiten agregar, correlacionar y analizar grandes cantidades de datos. Estos conjuntos de datos grandes y dinámicos ofrecen nuevas oportunidades para extraer información y conocimiento.

Avances en el análisis de datos: Impulsada por las inversiones de la industria en las áreas de investigación, desarrollo y fabricación, la Ley de Moore continúa ofreciendo mayor potencia de cálculo a precios más bajos y con menor consumo de energía.

Surgimiento de la computación en la nube: La computación en la nube aprovecha recursos informáticos conectados en red para procesar, gestionar y almacenar datos. Este paradigma permite que dispositivos pequeños y distribuidos interactúen con potentes sistemas de soporte que brindan capacidades analíticas y de control [8].

3. Inteligencia Artificial

La Inteligencia Artificial (IA) es la parte de la Ciencia que se ocupa del diseño de sistemas de computación inteligentes, es decir, sistemas que exhiben las características que asociamos a la inteligencia en el comportamiento humano que se refiere a la comprensión del lenguaje, el aprendizaje, el razonamiento, la resolución de problemas, etc [9].

La IA es una rama de las ciencias computacionales que se ocupa de los símbolos y métodos no algorítmicos para la resolución de problemas [9].

3.1. Ramas que componen la Inteligencia Artificial

Existen varios elementos que componen la ciencia de la IA, dentro de los cuales se pueden encontrar tres grandes ramas:

3.1.1. Lógica difusa

La lógica difusa es una rama de la IA que le permite a una computadora analizar información del mundo real en una escala entre lo falso y verdadero. Los matemáticos dedicados a la lógica en la década de 1920 definieron un concepto clave: todo es cuestión de grado. La lógica difusa manipula conceptos vagos como “caliente” o “húmedo” y permite a los ingenieros construir televisores, acondicionadores de aire, lavadoras y otros dispositivos que juzgan información difícil de definir. Los sistemas difusos son una alternativa a las nociones de pertenencia y lógica que se iniciaron en la Grecia antigua [9].

Características de la Lógica Difusa

- El razonamiento exacto puede verse como un caso particular del razonamiento aproximado. Cualquier sistema lógico puede ser fuzzificado. Mediante lógica difusa se puede formular el conocimiento humano de una forma sistemática, y puede ser fácilmente incluido en sistemas de ingeniería.
- El conocimiento se interpreta como una colección de restricciones difusas sobre una colección de variables. Los sistemas difusos son especialmente interesantes para la definición de sistemas cuyo modelo exacto es difícil de obtener (es necesario introducir una aproximación).

- La inferencia puede verse como un proceso de propagación de estas restricciones difusas.
- Se utiliza ampliamente en sistemas de ayuda a la decisión. La lógica difusa permite obtener decisiones con valores incompletos o información incierta.

Los sistemas difusos son muy recomendables en aquellos problemas muy complejos donde no existe un modelo matemático simple asociado. Igualmente, en procesos que obedecen a un comportamiento no lineal, la solución difusa plantea grandes ventajas. La solución difusa requiere que el conocimiento experto sea expresado lingüísticamente, requisito que es normalmente fácil de obtener [9].

3.1.2. Redes neuronales artificiales

La tecnología neural trata de reproducir el proceso de solución de problemas del cerebro. Así como los humanos aplican el conocimiento ganado con la experiencia a nuevos problemas o situaciones, una red neural toma como ejemplos problemas resueltos para construir un sistema que toma decisiones y realiza clasificaciones. Los problemas adecuados para la solución neural son aquellos que no tienen solución computacional precisa o que requieren algoritmos muy extensos como en el caso del reconocimiento de imágenes. [10]

Algunas ventajas de las redes neuronales artificiales (RNA) frente a otros sistemas de procesamiento de información son:

- Las RNA pueden sintetizar algoritmos a través de un proceso de aprendizaje.
- Para utilizar la tecnología neural no es necesario conocer los detalles matemáticos. Sólo se requiere estar familiarizado con los datos del trabajo.
- La solución de problemas no lineales es uno de los fuertes de las RNA.
- Las RNA son robustas, pueden fallar algunos elementos de procesamiento, pero la red continúa trabajando; esto es contrario a lo que sucede en programación tradicional. [10]

Las desventajas de las RNA son:

- Las RNA se deben entrenar para cada problema. Además, es necesario

realizar múltiples pruebas para determinar la arquitectura adecuada. El entrenamiento es largo y puede consumir varias horas de la computadora (CPU).

- Debido a que las redes se entrenan en lugar de programarlas, éstas necesitan muchos datos.
- Las RNA representan un aspecto complejo para un observador externo que desee realizar cambios. Para añadir nuevo conocimiento es necesario cambiar las iteraciones entre muchas unidades para que su efecto unificado sintetice este conocimiento. Para un problema de tamaño considerable es imposible hacer esto manualmente, por lo tanto, una red con representación distribuida debe emplear algún esquema de aprendizaje. [9] [10] [11]

3.1.3. Redes Bayesianas

Las redes Bayesianas son modelos gráficos probabilísticos utilizados en la toma de decisiones. Esta red está compuesta por nodos y arcos, donde los nodos representan variables aleatorias que pueden ser continuas o discretas y los arcos interpretan influencias causales. [12]

La utilización de las redes bayesianas ha sido muy difundida en diferentes campos como en la manipulación de sistemas expertos en la medicina, en la depuración de programas de inteligencia artificial, en la genética, en procesos de producción; todos estos enfocados a la solución de problemas y en la identificación de probabilidades que disminuyan riesgo, como es el uso primordial que se le da en la rama financiera. El uso que se le da a las redes bayesianas en el estudio y tratamiento de datos, consiste básicamente con mejorar el esquema de prevención de riesgo operacional; tal que sirva como sistema de ayuda a la toma de decisiones en una situación de emergencia y realizar mejoras en las estrategias de operacionales, etc. [13]

Las redes Bayesianas tienen un número de características que hacen que sean apropiadas para la ingeniería del software:

- **Representación gráfica.** Por su naturaleza, las redes Bayesianas proveen una representación gráfica de las relaciones explícitas de dependencia del dominio. Generalmente las variables en la ingeniería del software como por ejemplo esfuerzo o coste, están influenciados por muchos factores. Las redes

Bayesianas nos permiten modelar sistemas complejos permitiéndonos entender las relaciones causales visualizándolas por medio del grafo.

- **Modelado cualitativo y cuantitativo.** Las redes Bayesianas están formadas por un componente cualitativo, el grafo, y una parte cuantitativa, las tablas de probabilidades, que permiten utilizar criterios objetivos (por ejemplo, utilizando proyectos finalizados) y subjetivos (por ejemplo, utilizando creencias de expertos del dominio).
- **Inferencia bidireccional.** Las redes Bayesianas pueden hacer inferencia en ambos sentidos, es decir, las variables de entrada pueden ser usadas para predecir las variables de salida y viceversa. Fijando las variables de salida con los valores deseados, es posible predecir qué valores de las variables de entrada permiten dicha salida. Por ejemplo, usando inferencia hacia adelante, se puede predecir el número final de defectos basándose en variables como tamaño del proyecto, complejidad, esfuerzo en diseño, esfuerzo en testeo, etc. Por el contrario, podríamos fijar un número de defectos y predecir qué esfuerzo necesario satisface dicha salida.
- **Análisis de sensibilidad.** Dado un conjunto de evidencias, las redes Bayesianas permiten fácilmente calcular la sensibilidad de ciertas variables, simplemente modificando las evidencias.
- **Incertidumbre.** Las redes bayesianas pueden modelar grados de certidumbre, en vez de valores exactos. Por tanto, permiten modelar la incertidumbre de manera efectiva y explícitamente, por lo que pueden realizar buenas predicciones con información incompleta. De hecho, Kitchenham y Linkman (1997) afirman que las estimaciones en la ingeniería del software son una evaluación probabilística de un suceso futuro, y es por tanto la razón de que los gestores de proyecto no obtengan buenos resultados.
- **Valores de confianza.** La salida de una red Bayesiana es una probabilidad de distribución en vez de valores únicos. Este tipo de información se puede usar para medir la confianza que podemos depositar en la salida de la red Bayesiana, lo cual es esencial si el modelo va a ser usado en la toma de decisiones. Por ejemplo, en una variable con estados bajo, medio, y alto, la red Bayesiana estima la probabilidad de cada uno de los estados. [13]

3.1.4. Aprendizaje automático

El machine learning o aprendizaje automático es una disciplina científica del entorno de la ingeniería y la ciencia de la computación, cuyo objetivo reside en que los sistemas aprendan automáticamente (sin intervención humana). Aprender en este sentido, no se refiere a lo que comúnmente se conoce como aprendizaje humano, basado en la experiencia y en la razón, sino a la identificación de patrones complejos dentro de una gran cantidad de datos obtenidos mediante ejemplos, la experiencia o las instrucciones predefinidas. Tras el concepto de aprendizaje se encuentra un algoritmo que revisa los datos y es capaz de predecir comportamientos futuros, adaptándose a la incorporación de información adicional y recalibrando los resultados. Esto quiere decir que las personas no tengan que desempeñar ningún papel. Los analistas son necesarios para participar en tareas como la revisión y confirmación de decisiones, y en casos especiales en la toma de decisiones efectivas.

Además, incluso los sistemas más automatizados, confían en expertos para crear y mantener reglas y monitorizar los resultados.

Como punto de partida, una de las definiciones formales de machine learning más citada y ampliamente aceptada en el campo es la de Tom M. Mitchell:

“Se dice de un programa informático que aprende de la experiencia E con respecto algún conjunto de tareas T y la medida de rendimiento P si su rendimiento en las tareas en T , medido por P , mejora con la experiencia E ”. [14]

3.2. Visión por Computador

En la actualidad el gran avance que se ha tenido en las ciencias de la computación, ha permitido que estas incursionen en diferentes áreas de la ingeniería, medicina, ciencias de la tierra, entre otras. En el campo de la Visión por Computador (o sea el análisis digital de imágenes por medio de computador), los grandes avances en informática en la década de los 80, permitió que se realizarán las primeras aplicaciones industriales utilizando visión por computador. [1]

La imagen es el insumo principal para una aplicación de visión por computador. El término hace referencia a una función $f(x,y)$ la cual representa el nivel de intensidad de luz para un coordenada (x,y) , esta función es también conocida como pixel y una

imagen es representada como una matriz de píxeles.

Con frecuencia, para el análisis de una imagen es preciso considerar más concretamente un área dentro de la propia imagen que recibe el nombre de Región de Interés (en terminología inglesa Region Of Interest - ROI). Para ello, es necesario disponer de operaciones geométricas que modifiquen coordenadas espaciales de la imagen. Una operación geométrica tiene como objetivo principal transformar los valores de una imagen tal y como podrían observarse desde otro punto de vista. De esta manera, operaciones como magnificar o reducir una imagen no son más que aproximar o alejar el punto de vista, realizar una rotación equivale a girar el punto de vista, y una traslación equivale a hacer lo propio con dicho punto.

El procesamiento digital de imágenes está compuesto por una serie de etapas en las cuales se busca mejorar la información contenida en una imagen para la interpretación y el tratamiento de la información contenida en ella. Las etapas fundamentales para el procesamiento digital de imágenes son:

3.2.1. Adquisición de una imagen

El proceso de adquisición consiste básicamente en el uso de un sensor el cual se encarga de capturar la luz emitida por una escena y almacenarlo en un archivo con formato de imagen válido. Estos sensores se encuentran instalados en cámaras de video, cámaras de fotografía digital o cámaras convencionales.

3.2.2. Realce de una imagen

El proceso de realce consiste en mejorar la apariencia visual de una imagen a través del uso de técnicas que permiten la manipulación, mejorar contraste, brillo, eliminar distorsiones, falta de nitidez, etc.

3.2.3. Procesamiento del color

El procesamiento del color consiste en la manipulación de la imagen a color la cual puede contener más información que la imagen en niveles de gris. Este permite cambiar las imágenes entre los diferentes espacios por ejemplo de RGB a HSV, CMY o niveles de gris.

3.2.4. Procesamiento morfológico

El procesamiento morfológico consiste en la aplicación de una serie de operadores, que permiten extraer los componentes útiles de una imagen, entre ellos bordes, esqueletos, operaciones de llenado y operadores lógicos, filtros, entre otros.

3.2.5. Segmentación de una imagen

El proceso de segmentación consiste en extraer de una imagen un objeto u objetos que son importantes o se requieren para el proceso de representación y descripción. Se describe generalmente como extraer los elementos de una escena (también conocida como fondo). Este proceso también es conocido como binarización de una imagen. Existen diferentes técnicas básicas o robustas para el proceso de segmentación entre ellas una de las más utilizadas es la umbralización, la cual seleccionar los píxeles de una imagen que corresponden a un valor o rango de niveles de gris.

3.2.6. Representación y descripción

El proceso de representación y descripción permite extraer las características de los objetos que fueron seleccionados en el proceso de segmentación. Entre las más importantes se destacan la representación de bordes, cálculo de áreas, longitudes, direcciones, formas, etc.

3.2.7. Reconocimiento

El proceso de reconocimiento permite identificar teniendo en cuenta un patrón la información contenida en la imagen. El reconocimiento está compuesto por una serie de etapas, para ello pasa por un proceso de condicionamiento el cual permite filtrar la información de la imagen (ejemplo eliminar ruido), etiquetado, donde un píxel correspondiente a un patrón se marcan como pertenecientes a una clase u otra, agrupamiento, se identifican los píxeles correspondientes a una misma clase, extracción, esta determina las propiedades o características de los píxeles agrupados, comparación, donde finalmente un grupo de píxeles se asocian a un objeto ya conocido. [1] [13]

4. Arquitectura y Diseño de Aplicaciones Web

4.1. Sistemas web

Los sistemas web en pocos años se han convertido en complejos sistemas con interfaces de usuario cada vez más parecidas a las aplicaciones de escritorio, dando servicio a procesos de negocio de considerable envergadura y estableciéndose sobre ellos requisitos estrictos de accesibilidad y respuesta. [15]

En los últimos años, el avance de Internet y del uso de intranets corporativas ha supuesto una transformación en las necesidades de información de las organizaciones. En particular esto afecta a dos necesidades relacionadas con:

- Accesibilidad de la información desde cualquier lugar dentro de la organización e incluso desde el exterior.
- La forma en que se comparte la información entre todas las partes interesadas, de manera que todas tengan acceso a la información completa (o a aquella parte que les corresponda según su función) en cada momento.

4.2. Aplicaciones web

Las aplicaciones web, son software diseñados para ser accedidos desde un navegador web, sin necesidad de distribuir, instalar o actualizar software adicional a los usuarios. Una de las características principales, es el acceso inmediato y desde cualquier lugar. Las aplicaciones basadas en tecnologías web no necesitan ser descargadas, instaladas y configuradas.

Características de las aplicaciones web

Las Aplicaciones web tienen una serie de atributos comunes que la diferencia de otro tipo de aplicaciones de software, y son las siguientes:

- **Accesibles:** El usuario puede acceder fácilmente a estas aplicaciones empleando un navegador web.
- **Disponibles:** El usuario puede acceder a estas en cualquier momento y desde cualquier lugar del mundo donde disponga de una conexión a internet.

- **Mantenible:** Se puede actualizar y mantener una única aplicación y todos sus miles de usuarios verán los resultados inmediatamente. Los usuarios no tienen que actualizarla.
- **Portables:** Emplean tecnologías como Java, JavaFX, JavaScript, DHTML, Flash, Ajax entre otras, que dan gran potencia a la interfaz de usuario.
- **Multiplataforma:** Emplean tecnologías que permiten una gran portabilidad entre diferentes plataformas. Se reducen los problemas de compatibilidad.
- **Livianas:** Requieren poco o nada de espacio en disco.
- **Interactivas:** Las aplicaciones web pueden ejecutar audio, vídeo, mostrar animaciones. [19]

4.2.1. Arquitectura para aplicaciones web

A continuación, se presenta un análisis de las arquitecturas seleccionadas que cumplen con las expectativas para el desarrollo de sistema de conteo vehicular.

a. Arquitectura multicapa

En la arquitectura multicapa la aplicación se descompone en varias capas. El objetivo principal es separar los componentes de acuerdo a su función, es decir separar los componentes encargados de la presentación, otros de la lógica de negocio y otro de la persistencia de los datos. Esto significa que la arquitectura está generalmente compartida por:

- Un cliente, es decir, el equipo que solicita los recursos, equipado con una interfaz de usuario (generalmente un navegador web) para la presentación.
- El servidor de aplicaciones (también denominado software intermedio), cuya tarea es proporcionar los recursos solicitados, pero que requiere de otro servidor para hacerlo.
- El servidor de datos, que proporciona al servidor de aplicaciones los datos que éste le solicitó. [15]

b. Arquitectura Modelo Vista Controlador

Este tipo de arquitectura separa los componentes relacionados con los datos de la aplicación de los componentes de la interfaz de usuario. La separación de estas capas, permite tener, a nivel de desarrollo un código más claro, flexible y reutilizable.

La arquitectura MVC descompone la aplicación en capas, permitiendo tener una separación entre la lógica de negocio de la aplicación, la representación y la persistencia. Así como identifica tres capas que son importantes para cualquier aplicación las cuales son:

- **Modelo:** Encapsula los datos de la aplicación y la lógica para interactuar con ellos.
- **Vista:** Maneja la interacción con el usuario y la representación del modelo.
- **Controlador:** Es el intermediario entre el modelo y la vista ante las peticiones generadas por el cliente en la vista. El controlador se encarga de seleccionar el modelo solicitado por el usuario y la vista adecuada para representarlo. [18]

c. Arquitectura Cliente/Servidor

La arquitectura cliente-servidor es un modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa y el servidor es el que remite una respuesta. La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un solo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma. [15]

Características

En la arquitectura C/S el remitente de una solicitud es conocido como cliente. Sus características son:

- Es el que inicia solicitudes o peticiones. Tiene, por tanto, un papel activo en la comunicación (dispositivo maestro o amo).

- Espera y recibe las respuestas del servidor.
- Por lo general, puede conectarse a varios servidores a la vez.
- Normalmente, interactúa directamente con los usuarios finales mediante una interfaz gráfica de usuario.
- Al contratar un servicio de red, se debe tener en cuenta la velocidad de conexión que se le otorga al cliente y el tipo de cable que utiliza.

Al receptor de la solicitud enviada por el cliente se conoce como servidor. Sus características son:

- Al iniciarse espera a que le lleguen las solicitudes de los clientes. Desempeñan entonces un papel pasivo en la comunicación (dispositivo esclavo).
- Tras la recepción de una solicitud, la procesan y luego envían la respuesta al cliente.
- Por lo general, aceptan conexiones desde un gran número de clientes (en ciertos casos el número máximo de peticiones puede estar limitado).
- No es frecuente que interactúen directamente con los usuarios finales.

Ventajas

- Centralización del control: Los accesos, recursos y la integridad de los datos son controlados por el servidor, de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema. Esta centralización también facilita la tarea de poner al día datos u otros recursos (mejor que en las redes P2P).
- Escalabilidad: Se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- Fácil mantenimiento: Al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio (o se afectarán mínimamente). Esta independencia de los cambios también se conoce como encapsulación.
- Tecnologías: Existen algunas suficientemente desarrolladas, diseñadas para el paradigma de C/S, que aseguran la seguridad en las transacciones, la

amigabilidad de la interfaz y la facilidad de empleo. [16]

En base al análisis realizado de todas las arquitecturas mostradas, se seleccionó la arquitectura cliente servidor, ya que es la que se más se acopla a las necesidades que se pretende resolver con el software de conteo vehicular.

5. Tecnologías empleadas para el desarrollo del software

5.1. Backend de la Aplicación web

En desarrollo web, se utiliza los términos backend y frontend, los cuales permiten realizar un tipo de abstracción, que sirve para mantener separadas de cierta forma las partes de un sistema. Backend hace referencia a todos los procesos del lado del servidor, donde hay trabajo de ingeniería que compone el acceso a bases de datos y generación de scripts del lado del servidor. [16]

5.1.1. Lenguaje de programación JAVA

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos diez millones de usuarios reportados [18]

5.1.2. Lenguaje de programación C++

El lenguaje de programación C++ es uno de los más empleados en la actualidad. Se puede decir que C++ es un lenguaje híbrido, ya que permite programar tanto en estilo procedimental (como si fuese C), como en estilo orientado a objetos, como en ambos a la vez. Además, también se puede emplear mediante programación basada en eventos para crear programas que usen interfaz gráfica de usuario. [17]

5.1.3. Sistema Operativo Linux

Para poder hacer uso de cualquier programa dentro de un ordenador, es necesario

que todo equipo tenga un sistema operativo. Todos los programas utilizados para el desarrollo del sistema de conteo vehicular fueron bajo el sistema operativo Ubuntu 16.04 [18].

5.1.4. Servidor web

Un servidor web es un programa informático que se ejecuta continuamente en un computador, manteniéndose a la espera de peticiones de ejecución que le hará un cliente o un usuario de internet. El servidor web se encarga de contestar a estas peticiones de forma adecuada, entregando como resultado una página web o información de todo tipo de acuerdo a los comandos solicitados. Su tarea es alojar sitios y/o aplicaciones, las cuales son accedidas por los clientes utilizando un navegador que se comunica con el servidor utilizando el protocolo HTTP. [18]

Un servidor web consta de un intérprete http el cual se mantiene a la espera de peticiones de clientes y le responde con el contenido según sea solicitado. El cliente, una vez recibido el código, lo interpreta y lo exhibe al usuario.

5.1.5. Gestor de Base de datos

Un gestor de base de datos o sistema de gestión de base de datos (SGBD o DBMS) es un software que permite introducir, organizar y recuperar la información de las bases de datos; en definitiva, administrarlas. Existen distintos tipos de gestores de bases de datos: relacional, jerárquico, red, entre otros. El modelo relacional es el utilizado por casi todos los gestores de bases de datos para ordenadores.

La solución a los ficheros fueron bases de datos que aseguran:

- Evitar la redundancia “gratuita”.
- Flexibilidad de búsqueda.
- Independencia de los programas.
- Seguridad y confidencialidad integral [18].

A continuación, se explica el gestor de base de datos elegido para el sistema de conteo vehicular.

5.1.6. Frontend de la Aplicación web

En desarrollo web el frontend es la parte del software que interactúa con el usuario. Es decir, hace referencia a todos los procesos del lado del cliente. Para el desarrollo del frontend de la aplicación web se hizo uso de las estructuras Html5, Bootstrap, y para agregar interacciones JavaScript [16].

5.1.6.1. HTML

Para la creación de la estructura y contenido de la aplicación web se hizo uso del lenguaje Html5, el cual básicamente es el lenguaje de marcado predominante para la construcción de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo, JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML [16].

5.1.6.2. Bootstrap

Bootstrap, es un framework originalmente creado por Twitter, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de un pc, una tablet u otro dispositivo. El uso de este framework fue implementado para darle un diseño adaptativo a la aplicación web. Este framework ayuda al desarrollador a despreocuparse de las compatibilidades entre navegadores web, además contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño, también tiene extensiones de JavaScript opcionales [16].

5.1.6.3. JavaScript

JavaScript es un lenguaje de programación, al igual que PHP, si bien tiene diferencias importantes con éste. JavaScript se utiliza principalmente del lado del cliente, es decir, se ejecuta en nuestro ordenador, no en el servidor. Se hizo uso de este lenguaje para crear efectos atractivos y dinámicos en la aplicación web [19].

6. Librerías para el procesamiento de Imágenes

En la actualidad existe una variedad de herramientas para visión por computador a continuación se va realizar un análisis de las más utilizadas:

6.1. OpenCV(Open Source Computer Vision)

OpenCV (Open Source Computer Vision Library) es una biblioteca de software de visión abierta y software de aprendizaje automático. OpenCV fue construido para proporcionar una infraestructura común para aplicaciones de visión por computadora y para acelerar el uso de la percepción de la máquina en los productos comerciales. Al ser un producto con licencia de BSD, OpenCV facilita a las empresas utilizar y modificar el código.

La biblioteca cuenta con más de 2500 algoritmos optimizados, que incluyen un conjunto completo de algoritmos de visión artificial y de aprendizaje automático tanto clásico como avanzado.

Los algoritmos se pueden utilizar para detectar y reconocer rostros, identificar objetos, clasificar acciones humanas en videos, rastrear movimientos de la cámara, rastrear objetos en movimiento, extraer modelos 3D de objetos, producir nubes de puntos 3D desde cámaras estéreo, unir imágenes para producir una alta resolución imagen de una escena completa, encuentre imágenes similares de una base de datos de imágenes, elimine los ojos rojos de las imágenes tomadas con flash, siga los movimientos oculares, reconozca el escenario y establezca marcadores para superponerlo con realidad aumentada, etc. La biblioteca se usa ampliamente en compañías, grupos de investigación y por organismos gubernamentales.

Tiene interfaces C ++, Python, Java y MATLAB y es compatible con Windows, Linux, Android y Mac OS. OpenCV se inclina principalmente hacia las aplicaciones de visión en tiempo real y aprovecha las instrucciones de MMX y SSE cuando están disponibles. Hay más de 500 algoritmos y aproximadamente 10 veces más funciones que componen o admiten esos algoritmos. OpenCV está escrito nativamente en C ++ y tiene una interfaz de plantilla que funciona a la perfección con contenedores STL [20].

OpenCV tiene una estructura modular, lo que significa que el paquete incluye varias

bibliotecas compartidas o estáticas. Los siguientes módulos están disponibles [21]:

- **Core:** un módulo compacto que define las estructuras de datos básicos, incluida la matriz multidimensional densa Mat y las funciones básicas utilizadas por todos los demás módulos.
- **Imgproc:** módulo de procesamiento de imágenes que incluye filtrado de imágenes lineales y no lineales, transformaciones geométricas de imágenes (cambio de tamaño, afinación y deformación en perspectiva, remapeo genérico basado en tablas), conversión del espacio de color, histogramas, etc.
- **Video:** un módulo de análisis de video que incluye estimación de movimiento, sustracción de fondo y algoritmos de seguimiento de objetos.
- **Calib3d:** algoritmos básicos de geometría de vista múltiple, calibración de cámara única y estéreo, estimación de pose de objeto, algoritmos de correspondencia estéreo y elementos de reconstrucción en 3D.
- **Features2d:** detectores de características sobresalientes, descriptores y emparejadores de descriptores.
- **Objdetect:** detección de objetos e instancias de las clases predefinidas (por ejemplo, caras, ojos, tazas, personas, autos, etc.).
- **Highgui:** una interfaz fácil de usar para la captura de video, códecs de imágenes y video, así como capacidades simples de interfaz de usuario.

6.2. Torch3vision

Es una biblioteca de visión artificial, escrita en C ++ simple y basada en la biblioteca de aprendizaje automático Torch, es gratuito y distribuido bajo una licencia de BSD [22].

Esta librería consta de las siguientes características:

- Leer / escribir formatos de imagen básicos (pgm, ppm, gif, tif, jpeg)
- Algoritmos básicos de procesamiento de imágenes y extracción de características como rotación, voltereta, normalizaciones fotométricas

(Ecuación Histograma, Retinex Multiescala, Imagen de Auto-Cociente o Gross-BrajoVIC), detección de bordes, DCT 2D, FFT 2D, Gabor 2D, PCA para hacer Eigen-Faces, LDA para hacer Fisher-Faces

- Detección de rostros utilizando MLP, cascada de clasificadores similares a Haar
- Manipulación y dibujo de objetos 2D
- Decodificación / codificación de archivos de video, captura de imágenes desde un capturador de fotogramas o desde USB
- Control de giro / inclinación / zoom de la cámara Sony utilizando la interfaz VISCA

6.3. LTI-Lib

El LTI-Lib es una biblioteca orientada a objetos con algoritmos y estructuras de datos utilizados frecuentemente en el procesamiento de imágenes y la visión por computadora. Ha sido desarrollado en la Cátedra de Informática Técnica (Lehrstuhl fuer Technische Informatik) LTI en la Universidad Tecnológica de Aquisgrán, como parte de muchos proyectos de investigación en visión artificial relacionados con robótica, reconocimiento de objetos y lenguaje de cantar y reconocimiento de gestos.

El objetivo principal de LTI-Lib es proporcionar una biblioteca orientada a objetos en C ++, lo que simplifica el uso compartido y el mantenimiento de códigos, pero sigue proporcionando algoritmos rápidos que se pueden usar en aplicaciones reales.

Se ha desarrollado utilizando GCC en Linux y Visual C ++ en Windows NT [23].

Cuenta con más de 500 clases que tratan principalmente los siguientes campos:

- **Álgebra lineal:** Se proporcionan matrices, vectores, tensores y funtores para extraer valores propios, vectores propios, soluciones de ecuaciones lineales, estadísticas, etc.
- **Clasificación y Agrupación:** Los clasificadores de funciones de base radial, máquinas de vectores de soporte, medios k-medias, medios difusos C-medias, estadísticas de clasificación son solo algunos ejemplos de lo que puede hacer

con LTI-Lib.

- **Procesamiento de imágenes:** La mayoría de las clases se ocupan de los problemas de procesamiento de imágenes. Diferentes enfoques de segmentación, filtros lineales, wavelets, filtros orientables y mucho más ya están disponibles.
- **Visualización y herramientas de dibujo:** La parte más difícil al desarrollar algoritmos de procesamiento de imágenes en C ++ es mostrar imágenes temporales durante la depuración. Debido a la arquitectura orientada a objetos de LTI-Lib, solo necesita crear un objeto de visor y darle la imagen que necesita mostrar.

6.4. Comparativa de Librerías de Procesamiento de Imágenes

TABLA II. COMPARATIVA DE LIBRERÍAS DE PROCESAMIENTO DE IMÁGENES

Nombre	Ventajas	Desventajas
Opencv	<ul style="list-style-type: none"> • Dirigido hacia aplicaciones de visión por ordenador en tiempo real • Esta bajo la Licencia de BSD. • Cuenta con más de 2500 algoritmos optimizados para visión artificial y aprendizaje automático tanto clásico como avanzado. • Tiene interfaces C++, Python, Java y MATLAB. • Es compatible con Windows, Linux, Android y Mac OS. • Está escrito nativamente en C ++. 	

	<ul style="list-style-type: none"> • Tiene estructura modular. 	
Torch3vision	<ul style="list-style-type: none"> • Contiene algoritmos de procesamiento de imágenes y extracción de características. • Esta distribuido bajo la licencia BSD. • Es modular. 	<ul style="list-style-type: none"> • Se limita a trabajar únicamente con el lenguaje c++. • Sus algoritmos son muy básicos.
LTI-Lib	<ul style="list-style-type: none"> • Consta de algoritmos y estructuras de datos utilizados en el procesamiento de imágenes y la visión por computadora • Orientada a objetos. • Tiene módulos para despeñar diferentes funciones. • Esta bajo los términos de la Licencia Pública GNU. 	<ul style="list-style-type: none"> • Se limita a trabajar únicamente con el lenguaje c++. • Funciona solo en las plataformas GCC en Linux y Visual C ++ en Windows NT.

Luego de realizar la comparativa entre las ventajas y desventajas de las diferentes librerías para el procesamiento de imágenes se pudo llegar a la conclusión que Opencv ofrece las mejores las mejores prestaciones frente a Torch3vision y LTI-Lib ya que este se adaptan mejor a los requerimientos del sistema a desarrollar.

7. Algoritmos de Detección de objetos

Para realizar la detección de objetos podemos realizarla aplicando técnicas de análisis de movimiento, descriptores de características, y mediante clasificadores en cascada, cada uno se adapta mejor según el problema a resolver.

7.1. Análisis de movimiento

En este apartado se hace referencia a la sustracción de fondo que proporciona una implementación óptima para la detección de movimiento.

La sustracción de fondo es un paso importante de preproceso en muchas aplicaciones basadas en visión, técnicamente consiste en extraer el primer plano en movimiento del fondo estático.

Si tiene una imagen con el mismo fondo, como una imagen de la habitación sin visitantes, una imagen de la carretera sin vehículos, etc., el trabajo consiste en restar la nueva imagen del fondo, se dispone de 3 algoritmos basados en sustracción de fondo [24].

7.1.1. BackgroundsubtractorMog

Es un algoritmo de segmentación de fondo / primer plano basado en la combinación gaussiana. Se introdujo en el documento "Un modelo de mezcla de fondo adaptable mejorado para el seguimiento en tiempo real con detección de sombras" por P. KadewTraKuPong y R. Bowden en 2001.

Utiliza un método para modelar cada píxel de fondo mediante una mezcla de distribuciones K Gaussianas ($K = 3$ a 5). Los pesos de la mezcla representan las proporciones de tiempo que esos colores permanecen en la escena. Los colores de fondo probables son los que permanecen más largos y más estáticos [24].

7.1.2. BackgroundsubtractorMog2

También es un algoritmo de segmentación de fondo / primer plano basado en la combinación gaussiana. Se basa en dos artículos de Z. Zivkovic, "Modelo de mezcla gaussiana adaptable mejorado para la sustracción de fondo" en 2004 y "Estimación de densidad adaptativa eficiente por píxel de imagen para la tarea de resta de fondo" en 2006.

Una característica importante de este algoritmo es que selecciona el número apropiado de distribución gaussiana para cada píxel. Proporciona una mejor

adaptabilidad a escenas variables debido a cambios de iluminación, etc [24].

6.1.2. BackgroundsubtractorGMG

Este algoritmo combina la estimación de la imagen de fondo estadística y la segmentación bayesiana por píxel. Fue presentado por Andrew B. Godbehere, Akihiro Matsukawa, Ken Goldberg en su artículo "Seguimiento visual de visitantes humanos en condiciones de iluminación variable para una instalación de audio receptiva" en 2012.

Utiliza los primeros pocos (120 por defecto) marcos para modelar en segundo plano. Emplea un algoritmo probabilístico de segmentación en primer plano que identifica posibles objetos en primer plano utilizando la inferencia bayesiana. Las estimaciones son adaptativas; las observaciones más recientes tienen más peso que las observaciones antiguas para adaptarse a la iluminación variable. Se realizan varias operaciones de filtrado morfológico, como el cierre y la apertura, para eliminar el ruido no deseado. Obtendrá una ventana negra durante los primeros pocos fotogramas.

Sería mejor aplicar la apertura morfológica al resultado para eliminar los ruidos [24].

Ejemplo aplicando los algoritmos de análisis de movimiento

Marco original



Figura 4. Imagen original

1. Resultado de BackgroundSubtractorMOG

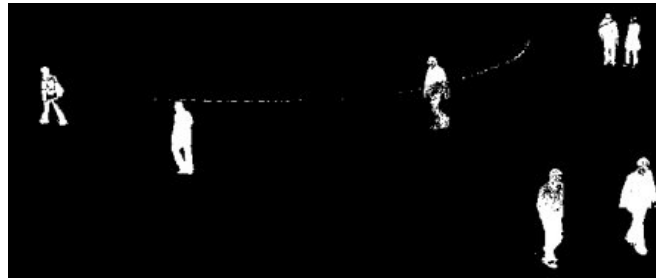


Figura 5. Algoritmo BackgroundSubtractorMOG

2. Resultado de BackgroundSubtractorMOG2

La región de color gris muestra la región sombreada.



Figura 6. Algoritmo BackgroundSubtractorMOG2

3. Resultado de BackgroundSubtractorGMG

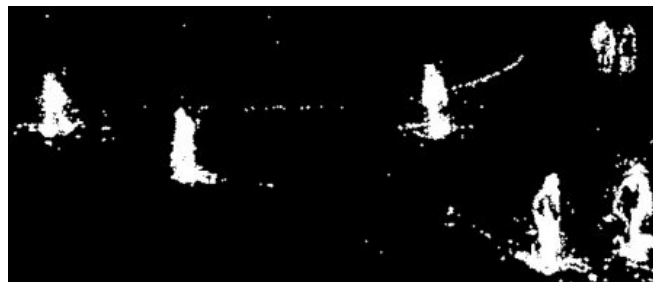


Figura 7. Algoritmo BackgroundSubtractorGMG

6.2. Descriptor de características

Consiste en la búsqueda de patrones específicos o características específicas que son únicas, se pueden rastrear fácilmente y se pueden comparar fácilmente. Si buscamos

una definición de tal característica, podemos encontrar que es difícil expresarla en palabras, pero sabemos cuáles son.

Una vez que tenga las características y su descripción, puede encontrar las mismas características en todas las imágenes y alinearlas, unir las o hacer lo que desee [25].

6.2.1. Algoritmo SIFT

El algoritmo SIFT es un algoritmo de visión artificial publicado originalmente por David Lowe en 1999 que se encarga de extraer características distintivas de las imágenes en escala de grises. Mediante estas, es posible luego reconocer dicha imagen dentro de una base de datos o incluso dentro de otra imagen mayor con otra cantidad de elementos en desorden [28].

Estas características son invariantes a factores de escala, traslación, rotación y parcialmente invariantes a cambios de iluminación y afinidades

- **Descriptores de puntos claves SIFT**

Las primeras etapas del algoritmo SIFT encuentran las coordenadas de los puntos claves en determinada escala y a cada punto le asignan una orientación. Los resultados de estas etapas garantizan invariabilidad a localización en la imagen, escala y rotación. Luego se calcula un descriptor para cada punto clave. Este descriptor debe ser altamente distintivo y parcialmente robusto a otro tipo de variaciones como iluminación y perspectiva 3D [26].

6.3. Clasificadores en cascada

Un clasificador es un sistema capaz de proporcionar una predicción de pertenencia a una clase como salida a partir de un conjunto de características tomadas como entradas. En la rama de aprendizaje supervisado, construir un clasificador se basa en definir una regla que pueda asignar una etiqueta a cualquier otro dato de entrada a partir de un conjunto de ejemplos etiquetados [27].

La palabra "cascada" en el nombre del clasificador significa que el clasificador resultante consiste en varios clasificadores (etapas) más simples que se aplican posteriormente a una región de interés hasta que en algún momento el candidato es rechazado o se pasan todas las etapas [28].

Clasificadores en cascada basados en características de Haar es un método eficaz de detección de objetos propuesto por Paul Viola y Michael Jones. Es un enfoque de aprendizaje automático donde la función de cascada está entrenada a partir de muchas imágenes positivas y negativas. Luego se usa para detectar objetos en otras imágenes [29].

7. Metodología de desarrollo de software

Una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Una metodología está formada por fases, cada una de las cuales se puede dividir en sub-fases, que guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo [30].

7.1. Metodología XP

XP es una metodología ágil para el desarrollo de software y consiste básicamente en ajustarse estrictamente a una serie de reglas que se centran en las necesidades del cliente para lograr un producto de buena calidad en poco tiempo, centrada en potenciar las relaciones interpersonales como clave para el éxito del desarrollo de software [33].

7.1.1. Características de XP

- Se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad.
- Se aplica de manera dinámica durante el ciclo de vida del software.
- Es capaz de adaptarse a los cambios de requisitos.
- Los individuos e interacciones son más importantes que los procesos y herramientas.
- Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas [30].

7.1.2. Fases de la metodología XP

La metodología XP consta de 4 fases las cuales se detallan a continuación:

1. Planificación

El primer paso de cualquier proyecto que siga la metodología XP es definir las historias de usuario con el cliente. Las historias de usuario tienen la misma finalidad que los casos de uso, pero con algunas diferencias: Constan de 3 ó 4 líneas escritas por el cliente en un lenguaje no técnico sin hacer mucho hincapié en los detalles; no se debe hablar ni de posibles algoritmos para su implementación ni de diseños de base de datos adecuados, etc. Son usadas para estimar tiempos de desarrollo de la parte de la aplicación que describen. También se utilizan en la fase de pruebas, para verificar si el programa cumple con lo que especifica la historia de usuario. Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia. El tiempo de desarrollo ideal para una historia de usuario es entre 1 y 3 semanas [30].

2. Diseño

La metodología XP sugiere que hay que conseguir diseños simples y sencillos. Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible e implementable que a la larga costará menos tiempo y esfuerzo desarrollar [30].

3. Codificación

Como ya se dijo en la introducción, el cliente es una parte más del equipo de desarrollo; su presencia es indispensable en las distintas fases de XP. A la hora de codificar una historia de usuario su presencia es aún más necesaria. No olvidemos que los clientes son los que crean las historias de usuario y negocian los tiempos en los que serán implementadas. Antes del desarrollo de cada historia de usuario el cliente debe especificar detalladamente lo que ésta hará y también tendrá que estar presente cuando se realicen los test que verifiquen que la historia implementada cumple la funcionalidad especificada. La codificación debe hacerse ateniendo a estándares de codificación ya creados. Programar bajo estándares mantiene el código consistente y facilita su comprensión y escalabilidad [30].

4. Pruebas

Uno de los pilares de la metodología XP es el uso de test para comprobar el funcionamiento de los códigos que vayamos implementando. El uso de los test en XP es el siguiente:

Se deben crear las aplicaciones que realizarán los test con un entorno de desarrollo específico para test.

Hay que someter a test las distintas clases del sistema omitiendo los métodos más triviales.

Se deben crear los test que pasarán los códigos antes de implementarlos; en el apartado anterior se explicó la importancia de crear antes los test que el código.

Un punto importante es crear test que no tengan ninguna dependencia del código que en un futuro evaluará.

Como se comentó anteriormente los distintos test se deben subir al repositorio de código acompañados del código que verifican.

Test de aceptación. Los test mencionados anteriormente sirven para evaluar las distintas tareas en las que ha sido dividida una historia de usuario.

Al ser las distintas funcionalidades de nuestra aplicación no demasiado extensas, no se harán test que analicen partes de las mismas, sino que las pruebas se realizarán para las funcionalidades generales que debe cumplir el programa especificado en la descripción de requisitos [30].

7.2. Comparativa de Metodologías de Desarrollo

Para la comparativa que se muestra en la TABLA IV, se seleccionó las siguientes metodologías orientadas al desarrollo de software Web: EORM (Metodología de relación entre Objetos), WSDM (Método de diseño para sitios web), XP (Programación extrema). El contenido que se ha tomado está elaborado de acuerdo a [30].

TABLA III. COMPARATIVA DE METODOLOGÍAS DE DESARROLLO

	Fases	Ventajas	Desventajas
EORM (Metodología de relación entre Objetos)	<p>Análisis: Se trata de orientar a objetos al sistema, sin considerar los aspectos hipermediales.</p> <p>Diseño: Modifica el modelo de objetos obtenidas añadiendo la semántica apropiada a las relaciones.</p> <p>Construcción: Se transforman los esquemas en código y se guardan en una base de datos orientadas a objetos [36].</p>	<p>Flexibilidad entre relaciones de los nodos.</p> <p>La reutilización de código y librerías.</p> <p>Encajamiento de relaciones semánticas</p>	<p>Problemas al funcionamiento del sistema o aspectos de interfaz.</p> <p>La captura de requisitos es muy pobre. Falta de comentarios o documentación.</p>
WSDM (Método de diseño para sitios web)	<p>Modelo de usuario: Se identifican y clasifican los usuarios que van a hacer uso de la aplicación Web.</p> <p>Diseño conceptual: Durante el modelado conceptual se realizan dos tareas a la vez: el modelado de objetos, que se llama modelo conceptual y el diseño de la navegación.</p> <p>Diseño de implementación: Se modela la interfaz para</p>	<p>Es adaptable a nuevos requisitos.</p>	<p>Al utilizar más tiempo en la obtención de requerimientos utiliza menos tiempo para su desarrollo y la entrega del proyecto suele ser mucho después a la fecha de entrega estimada.</p>

	<p>cada rol de usuario, la creación de alta funcionalidad, el código de los programas técnicos y la creación y diseño de la página principal.</p> <p>Realización de implementación: Se codifican todos estos los aspectos en el lenguaje seleccionado [37].</p>		
XP (Programación extrema)	<p>Planificación: El primer paso de cualquier proyecto que siga la metodología XP es definir las historias de usuario con el cliente. Las historias de usuario tienen la misma finalidad que los casos de uso pero con algunas diferencias: Constan de 3 ó 4 líneas escritas por el cliente en un lenguaje no técnico sin hacer mucho hincapié en los detalles</p> <p>Diseño: La metodología XP sugiere que hay que conseguir diseños simples y sencillos.</p>	<p>Programación organizada.</p> <p>Es capaz de adaptarse a los cambios de requisitos.</p> <p>Menor tasa de errores.</p> <p>Satisfacción del programador.</p>	<p>Es recomendable emplearla solo en proyectos a corto plazo.</p>

	<p>Hay que procurar hacerlo todo lo menos complicado posible para conseguir un diseño fácilmente entendible e impleméntable que a la larga costará menos tiempo y esfuerzo desarrollar.</p> <p>Codificación: La codificación debe hacerse atendiendo a estándares de codificación ya creados. Programar bajo estándares mantiene el código consistente y facilita su comprensión y escalabilidad.</p> <p>Pruebas: Uno de los pilares de la metodología XP es el uso de test para comprobar el funcionamiento de los códigos que vayamos implementando.</p>		
--	--	--	--

Del cuadro comparativo se excluye a las siguientes metodologías:

- EORM, ya que está diseñada para proyectos con gran cantidad de gráficos y multimedia; solo trata de manera específica los aspectos de almacenamiento y navegación, dejando a un lado temas como la funcionalidad del sistema o los

aspectos de interfaz; no hay una buena documentación y la captura de requisitos es muy pobre.

- WSDM, ya que está dirigida para desarrollar kioscos web, es decir, aplicaciones que muestran una determinada información al usuario y les permiten navegar hacia ella. Además, WSDM está orientado a aplicaciones web muy robustas, ofreciéndoles un estándar para administrar y supervisar el estado de otros servicios web o dispositivos de hardware habilitado para la red. Por ejemplo, un sistema habilitado para WSDM puede proporcionar información acerca del uso de CPU, uso de memoria y espacio de disco. Un consumidor de gestión es una entidad que recupera de forma activa información de estado desde el recurso, o de forma pasiva recibiendo notificaciones del mismo.
- XP, se basada en una serie de valores y de prácticas que permiten aumentar la productividad a la hora de desarrollar programas, se orienta en realimentación continua entre el cliente y el equipo de desarrollo, simplicidad en las soluciones implementadas y menor tasa de errores.

XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Es capaz de adaptarse a los cambios de requisitos.

Por lo cual, el análisis de elección se reduce a la metodología web XP, que cumple con los requerimientos para el desarrollo de un producto software con múltiples componentes como lo es el sistema de conteo vehicular.

E. MATERIALES Y MÉTODOS

Para el desarrollo del trabajo de titulación se utilizaron diferentes técnicas y métodos que permitieron realizar el análisis y valoración del trabajo y así brindar una solución óptima al problema.

Además, se eligió algunos materiales que mediante comparativas realizadas con soluciones similares se acordó que serían las mejores alternativas para nuestro propósito, herramientas como Gantt Project para desarrollo del cronograma, QtCreator como framework para desarrollo de contador en C++, NodeJs aplicado en el desarrollo del servidor, Justinmind utilizado para prototipos, etc.

1. Métodos

Los métodos que se utilizaron son:

- **Método Deductivo:** Se aplicó este método para obtener la información necesaria sobre los procesos que se realizan para el conteo normal de vehículos y los problemas que traen cada uno de ellos.
- **Método Inductivo:** A través de este método se planteó las principales dificultades al realizar el conteo vehicular para determinar si la solución planteada permite resolverlos de forma satisfactoria.

2. Técnicas

Las técnicas que se utilizaron para recopilar información son las siguientes:

- **Entrevista:** Esta técnica se aplicó con la finalidad de recopilar información verídica y relevante sobre el problema que existe y así brindar una solución óptima al mismo. La entrevista fue aplicada al Ing. Luis Roberto Jácome Galarza (Véase ANEXO 1).
- **Investigación Bibliográfica:** Con esta técnica se sustentó la base teórica de la investigación, a través de la recolección de información en fuentes bibliográficas confiables como libros, revistas indexadas, artículos científicos, base de datos científicas entre otras.

3. Metodología de desarrollo de software

Para el desarrollo de los módulos establecidos se aplicó la Metodología de desarrollo Programación Extrema (XP). La utilización de esta metodología que se ve reflejada al largo del desarrollo:

En la fase de **planificación** del proyecto se realizó recopilación de todos los requerimientos del proyecto mediante el uso de historias de usuario, se estableció el plan de entregas y el registro de iteraciones.

En la fase de **diseño** se realizó los diagramas que contienen la lógica del sistema, se desarrolló las tarjetas CRC y los prototipos del sistema.

En la fase de **desarrollo** se procedió a programar los módulos del sistema tomando en cuenta estándares de desarrollo con el fin de garantizar la calidad de las mismas

En la fase de **pruebas** se validaron cada una de las pruebas de aceptación establecidas, con el fin de determinar que el sistema cumpla con los objetivos para los que fue creado.

F. RESULTADOS

A continuación, se presenta los resultados obtenidos por cada fase del proyecto que se establecieron previamente.

1. PRIMERA FASE: PLANIFICACIÓN

De acuerdo con la metodología de desarrollo utilizada (Programación Extrema XP), en esta fase se va a realizar las historias de usuario mediante el uso de los requerimientos encontrados y se establecen los módulos de las aplicaciones.

1.1. Especificación de requerimientos

Para llevar a cabo el levantamiento de requisitos, se recolecto información relevante sobre las necesidades de la institución a través de una entrevista desarrollada al Ing. Luis Roberto Jácome Galarza (Véase ANEXO 1), una vez obtenida la información se la analizó y clasificó estableciendo de esta manera la lista de requisitos del software.

Después se realizó el Documento de Especificación de Requisitos de software (ERS) basado en el estándar IEEE830-1998, en donde se realizó una descripción completa del comportamiento de la aplicación en base a los requerimientos obtenidos anteriormente (Véase ANEXO 2).

A continuación, se presentan los requisitos funcionales y no funcionales de la aplicación:

1.1.1. Requerimientos Funcionales

En la siguiente TABLA V se muestran los requisitos funcionales del sistema, los cuales son la base para su desarrollo.

TABLA IV. LISTA DE REQUERIMIENTOS FUNCIONALES

Requerimiento	Descripción
RF01	Detección de vehículos.
RF02	Conteo de vehículos según el sentido en que transiten.
RF03	Envío datos de conteo a un servidor.
RF04	Almacenamiento de datos local.
RF05	Inicio de software con el arranque del sistema.

RF06	Configuración de la fuente de video.
RF07	Interfaz de configuración de software de conteo.
RF08	Gestión de usuarios en aplicación web.
RF09	Gestión de contadores en aplicación web.
RF10	Reportes de conteo en aplicación web
RF11	Georreferenciación de contadores en aplicación web.

1.1.2. Requerimientos No Funcionales

En la TABLA VI se muestran los requisitos no funcionales del sistema.

TABLA V. LISTA DE REQUERIMIENTOS NO FUNCIONALES

Requerimiento	Aspecto	Descripción
RNF01	Usabilidad	La aplicación web contará con una interfaz amigable e intuitiva
RNF02	Portabilidad	El software de conteo funcionará en plataformas con sistema operativo Linux con Debian derivado.
RNF03	Accesibilidad	El acceso al sistema web de conteo vehicular podrá realizarse desde los navegadores Firefox o Google Chrome, independientemente del sistema operativo en el que este se ejecute.
RNF04	Mantenibilidad	Todo el sistema debe contar con documentación entendible y fiable para que facilite la actualización de las funciones y la corrección de errores.
RNF05	Extendibilidad	El software de conteo vehicular permite incrementar nuevas funcionalidades o características sin ningún esfuerzo.
RNF06	Rendimiento	La aplicación permitirá el acceso a uno o más usuarios al mismo

		tiempo sin ningún tipo de inconveniente.
--	--	--

Luego de haber especificado los requerimientos de las aplicaciones, se identificó los siguientes tipos de usuarios, los mismos que tendrán determinados privilegios.

TABLA VI. TIPOS DE USUARIOS DE LAS APLICACIONES

Tipo de usuario	Descripción
Usuario Normal	Este usuario tiene acceso a las funcionalidades del sistema de administración web, donde se gestiona la información de los contadores y generación de reportes de conteo.
Administrador	Este usuario tendrá acceso a las mismas funcionalidades que el usuario normal y además la gestión de usuarios.

1.2. Módulos de las aplicaciones

Se desarrolló una aplicación web para gestión de contadores y software de conteo vehicular. Los módulos y las funcionalidades de cada una de las aplicaciones, se especifican a continuación:

1.2.1. Módulos de la Aplicación web

Módulo No. 1: Gestión de usuarios

- Registrar usuario
- Editar Usuario
- Eliminar usuario
- Visualizar usuarios

Módulo No. 2: Gestión de contadores

- Registrar contador
- Editar contador

- Eliminar contador
- Visualizar contador

Módulo No. 3: Reportes de conteo

- Generar reporte de conteo vehicular
- Exportar reporte de conteo vehicular

1.2.1. Módulos del Sistema de Conteo Vehicular

Módulo No. 1: Interfaz de configuración

- Visualización de imagen
- Configuración de posición de líneas de conteo.
- Configuración de la fuente de video

Módulo No.2: Contador vehicular

- Detección de vehículos
- Rastreo de vehículos en la imagen
- Conteo de vehículos según línea de conteo

1.3. Especificación de historias de usuario

A continuación, se especifican las historias de usuario por cada una de las aplicaciones y sus módulos respectivos:

1.3.1. Historias de usuario - Aplicación web

Módulo No. 1: Gestión de usuarios

TABLA VII. HISTORIA DE USUARIO: REGISTRAR USUARIO

Historia de Usuario	
Numero: 1	Nombre: Registrar usuario
Usuario: Cliente	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta / Medio / Bajo)	Puntos Estimados: 1
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Requerimiento: RF08
Descripción En la aplicación web el administrador puede registrar un usuario, para ello debe ingresar datos del usuario (nombre, apellido, correo, contraseña) e información del administrador para crear la cuenta de usuario.	
Observaciones	El administrador debe estar autenticado en la aplicación a través de su usuario y contraseña.

TABLA VIII. HISTORIA DE USUARIO: EDITAR USUARIO

Historia de Usuario	
Numero: 2	Nombre: Editar usuario
Usuario: Cliente	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta / Medio / Bajo)	Puntos Estimados: 1
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Requerimiento: RF08
Descripción En la aplicación web el administrador puede editar un usuario, para ello puede modificar datos del usuario (nombre, apellido, correo, contraseña).	
Observaciones	El administrador debe estar autenticado en la aplicación a través de su usuario y contraseña.

TABLA IX. HISTORIA DE USUARIO: ELIMINAR USUARIO

Historia de Usuario	
Numero: 3	Nombre: Eliminar usuario
Usuario: Cliente	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta / Medio / Bajo)	Puntos Estimados: 1
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Requerimiento: RF08
Descripción En la aplicación web el administrador puede eliminar un usuario, para ello debe seleccionar el usuario a eliminar.	
Observaciones	El administrador debe estar autenticado en la aplicación a través de su usuario y contraseña.

TABLA X. HISTORIA DE USUARIO: VISUALIZAR USUARIO

Historia de Usuario	
Numero: 4	Nombre: Visualizar usuario
Usuario: Cliente	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta / Medio / Bajo)	Puntos Estimados: 1
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Requerimiento: RF08
Descripción En la aplicación web el administrador puede visualizar los datos únicamente del usuario seleccionado.	
Observaciones	El administrador debe estar autenticado en la aplicación a través de su usuario y contraseña.

Módulo No. 2: Gestión de contadores

TABLA XI. HISTORIA DE USUARIO: REGISTRAR CONTADOR

Historia de Usuario	
Numero: 5	Nombre: Registrar contador
Usuario: Cliente	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta / Medio / Bajo)	Puntos Estimados: 1
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Requerimiento: RF09
Descripción En la aplicación web el administrador puede registrar un contador vehicular, para ello debe ingresar datos del contador (etiqueta, dirección, coordenadas).	
Observaciones	El administrador debe estar autenticado en la aplicación a través de su usuario y contraseña.

TABLA XII. HISTORIA DE USUARIO: EDITAR CONTADOR

Historia de Usuario	
Numero: 6	Nombre: Editar contador
Usuario: Cliente	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta / Medio / Bajo)	Puntos Estimados: 1
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Requerimiento: RF09
Descripción En la aplicación web el administrador puede editar un contador vehicular, para ello puede modificar datos del contador (etiqueta, dirección, coordenadas).	
Observaciones	El administrador debe estar autenticado en la aplicación a través de su usuario y contraseña.

TABLA XIII. HISTORIA DE USUARIO: ELIMINAR CONTADOR

Historia de Usuario	
Numero: 7	Nombre: Eliminar contador
Usuario: Cliente	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta / Medio / Bajo)	Puntos Estimados: 1
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Requerimiento: RF09
Descripción En la aplicación web el administrador puede eliminar un contador vehicular, para ello el administrador debe seleccionar el contador a eliminar.	
Observaciones	El administrador debe estar autenticado en la aplicación a través de su usuario y contraseña.

TABLA XIV. HISTORIA DE USUARIO: VISUALIZAR CONTADOR

Historia de Usuario	
Numero: 8	Nombre: Visualizar contador
Usuario: Cliente	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta / Medio / Bajo)	Puntos Estimados: 1
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Requerimiento: RF09, RF011
Descripción En la aplicación web el administrador puede visualizar la información de un contador en específico pudiendo ver en un mapa su ubicación.	
Observaciones	El administrador debe estar autenticado en la aplicación a través de su usuario y contraseña.

Módulo No. 3: Reportes de conteo

TABLA XV. HISTORIA DE USUARIO: REPORTES DE CONTEO

Historia de Usuario	
Numero: 9	Nombre: Generar reporte de conteo vehicular
Usuario: Cliente	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta / Medio / Bajo)	Puntos Estimados: 1
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Requerimiento: RF10
Descripción En la aplicación web el usuario puede generar reportes de los datos recolectados en rangos de fecha y hora.	
Observaciones	El administrador debe estar autenticado en la aplicación a través de su usuario y contraseña.

TABLA XVI. HISTORIA DE USUARIO: EXPORTAR REPORTE

Historia de Usuario	
Numero: 10	Nombre: Exportar reporte de conteo vehicular
Usuario: Cliente	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta / Medio / Bajo)	Puntos Estimados: 1
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Requerimiento: RF10
Descripción En la aplicación web el usuario puede generar reportes de los datos recolectados en rangos de fecha y hora, y exportar en los formatos, pdf, Excel, csv.	
Observaciones	El administrador debe estar autenticado en la aplicación a través de su usuario y contraseña

1.3.2. Historia de Usuario del Contador Vehicular

Módulo No. 1: Interfaz de configuración

TABLA XVII. HISTORIA DE USUARIO: VISUALIZACION DE IMAGEN

Historia de Usuario	
Numero: 11	Nombre: Visualización de imagen
Usuario: Cliente	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta / Medio / Bajo)	Puntos Estimados: 1
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Requerimiento: RF06 , RF07
Descripción El usuario encargado de la instalación del software de conteo vehicular puede visualizar la imagen que está transmitiendo la cámara configurada en el software.	
Observaciones	Ninguna

TABLA XVIII. HISTORIA DE USUARIO: CONFIGURACION DE LINEAS DE CONTEO

Historia de Usuario	
Numero: 12	Nombre: Configuración de líneas de conteo
Usuario: Cliente	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta / Medio / Bajo)	Puntos Estimados: 1
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Requerimiento: RF07
Descripción El usuario encargado de la instalación del software de conteo vehicular puede modificar la posición de la línea de conteo, esta línea se visualizará en la interfaz de transmisión de imagen.	
Observaciones	Ninguna

TABLA XIX. HISTORIA DE USUARIO: CONFIGURACIÓN DE LA FUENTE DE VIDEO

Historia de Usuario	
Numero: 13	Nombre: Configuración de la fuente de video
Usuario: Cliente	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta / Medio / Bajo)	Puntos Estimados: 1
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Requerimiento: RF07
Descripción El usuario encargado de la instalación del software de conteo vehicular puede modificar la fuente de video para el procesamiento de las imágenes, pudiendo ser cámaras IP, cámaras USB o video.	
Observaciones	Ninguna

Módulo No. 2: Contador vehicular

TABLA XX. HISTORIA DE USUARIO: DETECCIÓN DE VEHÍCULOS

Historia de Usuario	
Numero: 14	Nombre: Detección de vehículos
Usuario: Cliente	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta / Medio / Bajo)	Puntos Estimados: 1
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Requerimiento: RF01
Descripción El software de conteo vehicular debe detectar los vehículos en la fuente de video configurada.	
Observaciones	Ninguna

TABLA XXI. HISTORIA DE USUARIO: RASTREO DE VEHÍCULOS EN LA IMAGEN

Historia de Usuario	
Numero: 15	Nombre: Rastreo de vehículos en la imagen.
Usuario: Cliente	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta / Medio / Bajo)	Puntos Estimados: 1
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Requerimiento: RF02
Descripción El software de conteo vehicular debe poder realizar el seguimiento de cada vehículos en la imagen una vez los vehículos hayan sido detectados.	
Observaciones	Ninguna

TABLA XXII. HISTORIA DE USUARIO: CONTEO DE VEHÍCULOS SEGUN LINEA DE CONTEO

Historia de Usuario	
Numero: 16	Nombre: Conteo de vehículos según línea de conteo.
Usuario: Cliente	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta / Medio / Bajo)	Puntos Estimados: 1
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Requerimiento: RF02, RF03, RF04
Descripción El software de conteo vehicular incrementará el contador en uno, cada vez que un vehículo interseque la línea de conteo, este conteo se realiza según el sentido configurado en la línea de conteo, además se enviará una trama de datos al servidor y se almacenará localmente.	
Observaciones	Ninguna

TABLA XXIII. HISTORIA DE USUARIO: INICIO DEL SOFTWARE CON EL CONTEO DEL SISTEMA

Historia de Usuario	
Numero: 17	Nombre: Inicio del software de conteo con el sistema.
Usuario: Cliente	Iteración Asignada: 1
Prioridad en Negocio: Medio (Alta / Medio / Bajo)	Puntos Estimados: 1
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Requerimiento: RF05
Descripción El software de conteo vehicular iniciará con el sistema operativo como un servicio en segundo plano.	
Observaciones	Ninguna

1.4. Estimación de historias de usuario

En la TABLA X, se muestra la estimación del tiempo de cada una de las historias de usuario establecidas.

TABLA XXIV. ESTIMACIÓN DE HISTORIAS DE USUARIO

MODULO	Nro.	NOMBRE DE HISTORIA DE USUARIO	TIEMPO ESTIMADO		
			Semanas Estimadas	Días Estimados	Horas estimadas
Gestión de usuarios	1	Registrar usuario	1	4	20
	2	Editar Usuario	1	3	16
	3	Eliminar usuario	1	2	5
	4	Visualizar usuarios	1	3	10
Gestión de contadores	5	Registrar contador	1	4	20
	6	Editar contador	1	3	16
	7	Eliminar contador	1	2	5
	8	Visualizar contador	1	3	10
Reportes de conteo	9	Generar reporte de conteo vehicular	2	5	40

	10	Exportar reporte de conteo vehicular	2	4	35
Interfaz de configuración	12	Visualización de imagen	1	5	30
	13	Configuración de posición de líneas de conteo.	1	5	35
	14	Configuración de la fuente de video	1	4	25
Contador vehicular	15	Detección de vehículos	3	5	80
	16	Rastreo de vehículos en la imagen	2	5	60
	17	Conteo de vehículos según línea de conteo	2	5	40

2. SEGUNDA FASE: DISEÑO

Se elabora el modelo de la base datos relacionales, se estableció la arquitectura de las aplicaciones, y de acuerdo con la metodología de desarrollo utilizada (Programación Extrema XP), en la fase de diseño se van a realizar las tarjetas CRC.

2.1. Arquitectura del sistema

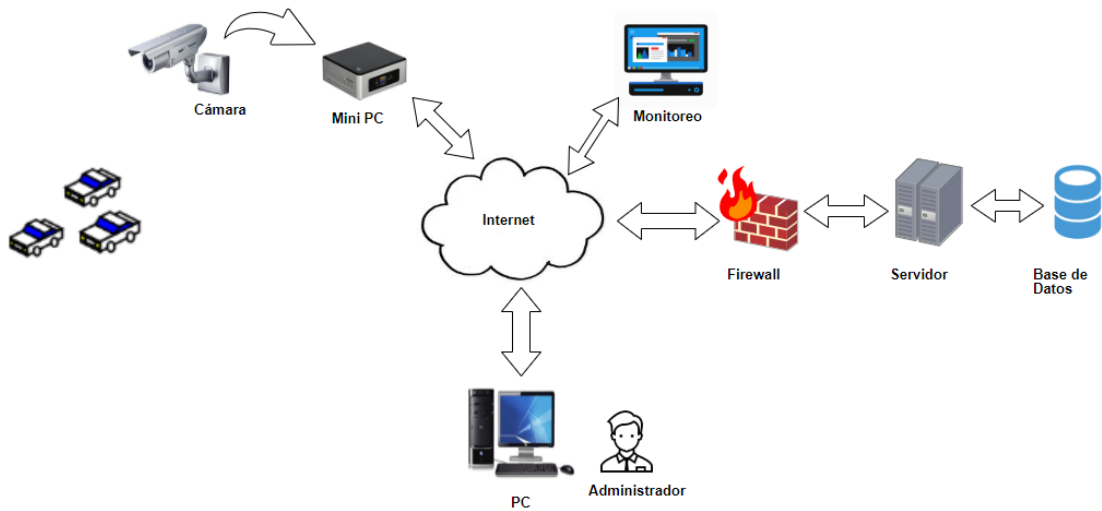


Figura 8. Arquitectura del sistema

- **Cámara:** Este dispositivo es utilizado para obtener las imágenes que luego serán procesadas por el software de conteo vehicular, la cámara puede ser tipo ip o usb, además se puede configurar como fuente de video un archivo grabado con anterioridad.
- **Equipo de Contador:** El computador en el cual será instalado el software de conteo vehicular basado en visión artificial, el sistema operativo debe ser Linux Debian derivado.
- **Servidor:** El servidor será encargado de persistir los datos enviados por el software de conteo, y alojar la aplicación del administrador del sistema.

La aplicación del servidor estará programada sobre NodeJs y la base de datos a utilizar es MySQL.

Características:

- Centos 7.4
- 1GB de Ram
- **Computador:** Equipo mediante el cual el administrador accede al sistema para administrar los contadores y visualizar los datos de conteo.

El requisito necesario para acceder a la página de administración es tener instalado un navegador en el computador.

2.2. Diagrama de clases

El diagrama de clases sirve para identificar las relaciones entre las clases que están involucradas en el sistema. Las clases pueden tener información como los atributos y funciones, mientras las relaciones pueden ser asociativas, herencia, agregación o asociación.

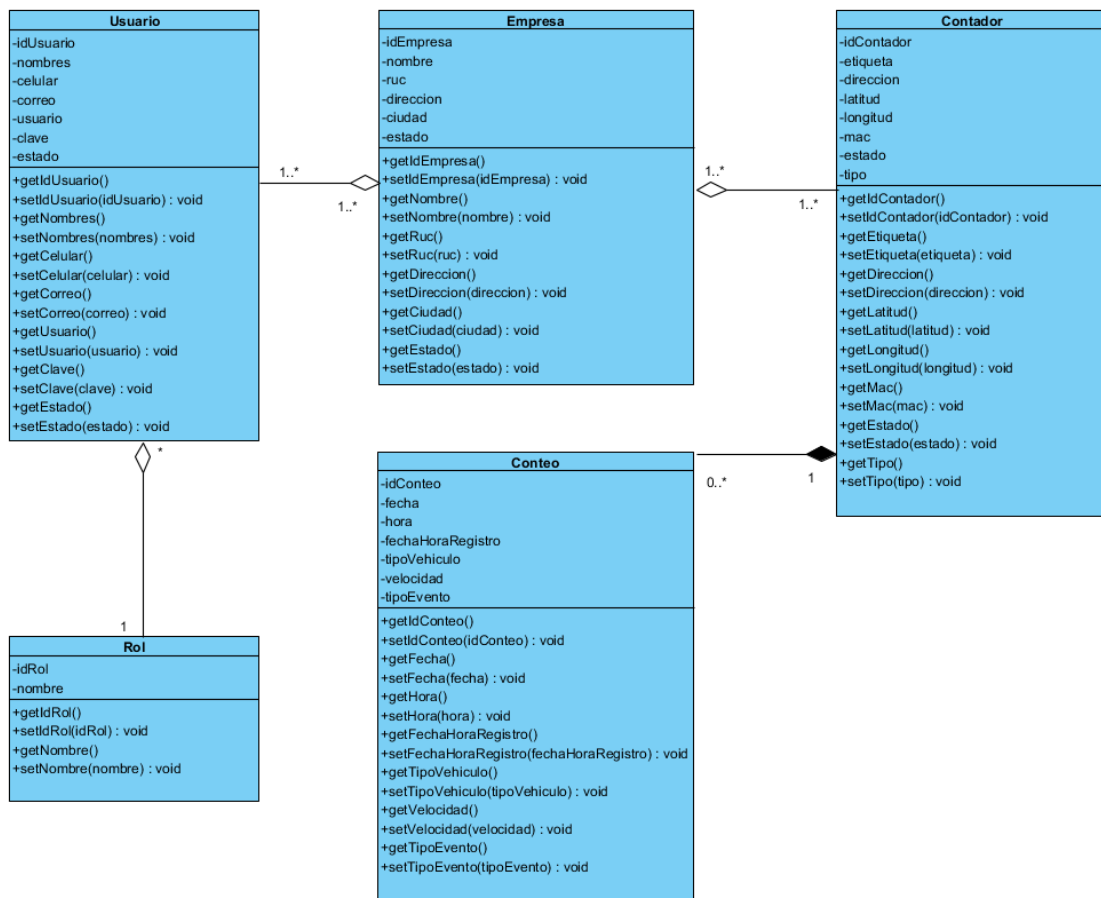


Figura 9. Diagrama de clases

Este diagrama se compone principalmente de cinco clases en las que se divide la lógica del software de conteo.

2.3. Modelo entidad relación

El modelo entidad relación consiste en describir la información del sistema mediante la definición de entidades con sus respectivos atributos e interrelaciones.

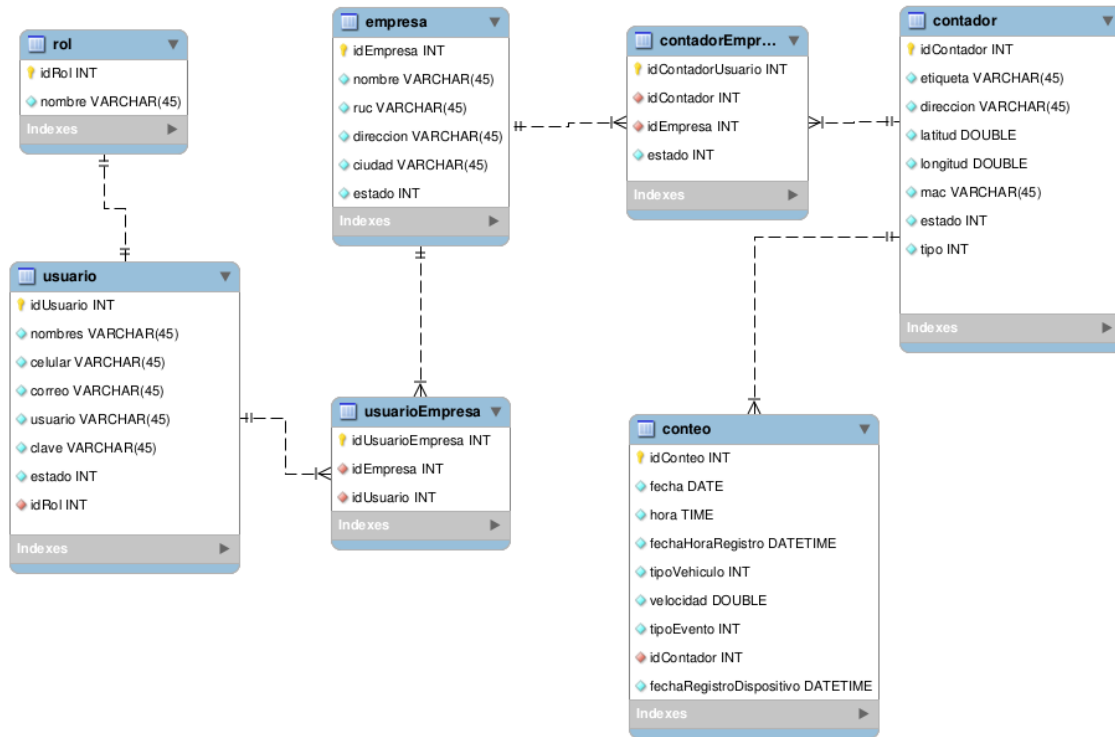


Figura 10. Modelo entidad relación

2.4. Tarjetas CRC

La metodología de desarrollo XP promueve el uso de tarjetas CRC (Clase-Responsabilidad-Colaboración) con el fin de que se establezcan las clases, atributos y métodos que se estén utilizando dentro de las aplicaciones. Las tarjetas CRC elaboradas se muestran a continuación:

2.4.1. Tarjetas CRC de aplicación de conteo

TABLA XXV: TARJETA CRC: PROCESADOR

Procesador	
Responsabilidad	Colaboración
<ul style="list-style-type: none">- Verificar rastreo de objetos.- Contar los objetos.- Generar trama de conteo.- Guardar trama en almacenamiento local.	<ul style="list-style-type: none">- Cámara- Tracker- ConexionServer

TABLA XXVI. TARJETA CRC: CAMARA

Camara	
Responsabilidad	Colaboración
<ul style="list-style-type: none">- Adquirir imagen- Detectar objetos.- Rastrear objetos.	<ul style="list-style-type: none">- Tracker- Procesador

TABLA XXVII. TARJETA CRC: TRACKER

Tracker	
Responsabilidad	Colaboración
<ul style="list-style-type: none">- Guardar posición del objeto en la imagen.- Guardar posición anterior del objeto en la imagen.- Guardar estado de conteo del objeto.	<ul style="list-style-type: none">- Camara- Procesador

TABLA XXVIII. TARJETA CRC: CONEXION SERVER

Conexión Server	
Responsabilidad	Colaboración
<ul style="list-style-type: none">- Mantener conexión TCP con servidor de datos.- Enviar tramas de conteo.	<ul style="list-style-type: none">- Procesador

2.4.2. Tarjetas CRC de aplicación web (Administración)

TABLA XXIX. TARJETA CRC: USUARIO

Usuario	
Responsabilidad	Colaboración
<ul style="list-style-type: none">- Datos de usuario para el registro nuevo.- Datos de usuario para la edición.- Contiene datos para el ingreso al sistema.	<ul style="list-style-type: none">- Empresa

TABLA XXX. TARJETA CRC: EMPRESA

Empresa	
Responsabilidad	Colaboración
<ul style="list-style-type: none">- Datos de empresa para el registro nuevo.- Datos de empresa para la edición.	<ul style="list-style-type: none">- Usuario- Contador

TABLA XXXI. TARJETA CRC: CONTADOR

Contador	
Responsabilidad	Colaboración
<ul style="list-style-type: none">- Datos de contador para el registro nuevo.- Datos de contador para la edición.- Referencia para graficar en el mapa.	<ul style="list-style-type: none">- Empresa- Conteo

TABLA XXXII. TARJETA CRC: CONTEO

Conteo	
Responsabilidad	Colaboración
<ul style="list-style-type: none">- Contener la información de cada evento de conteo registrado en el sistema.	<ul style="list-style-type: none">- Contador

2.5. Prototipos de las aplicaciones

En esta sección se presentan los prototipos principales tanto del contador vehicular como de la aplicación web.

2.5.1. Contador vehicular

- **Pantalla inicial**

La primera pantalla en presentarse cuando se ejecuta la aplicación es la siguiente:



Figura 11. Pantalla principal del software de conteo vehicular

En la parte izquierda se presenta por defecto una imagen que se muestra únicamente cuando aún no se tiene conexión con software de conteo, en la parte derecha se tiene las opciones de configuración para el funcionamiento del software, además de la opción conectar en la que ingresamos el host en el cual se ejecuta el software.

- **Transmisión de video**

Cuando se establece la conexión con el software de conteo se habilitan las opciones de configuración y en la parte inferior se muestra la información del conteo realizado.

En la sección izquierda se puede visualizar la transmisión de video y la detección de los vehículos.

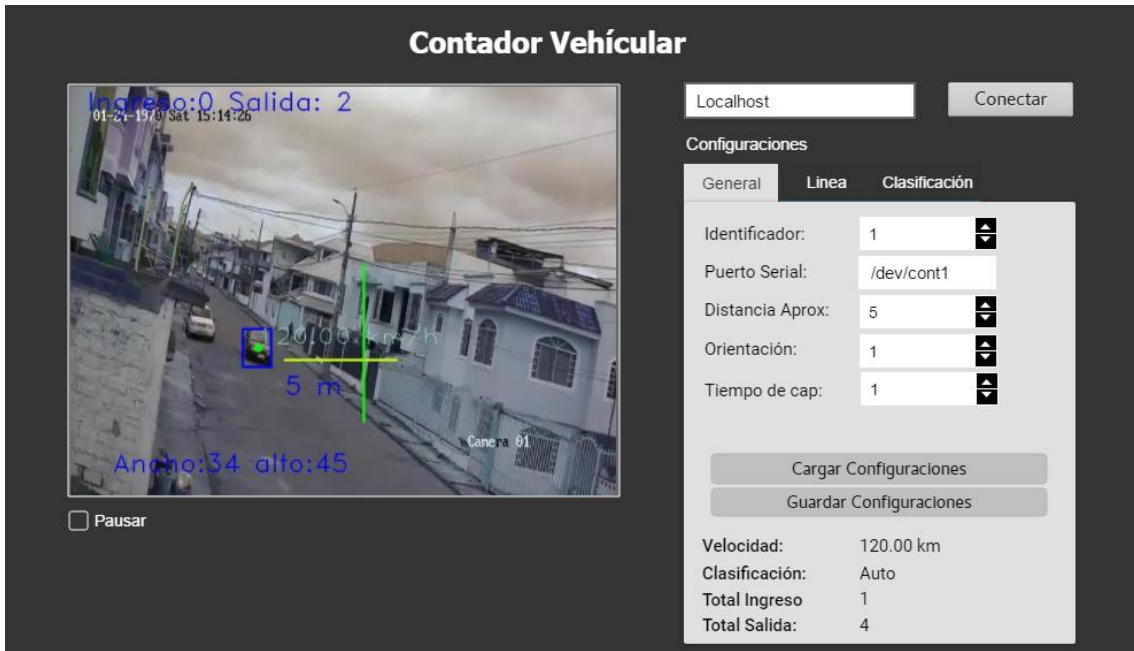


Figura 12. Visualización de video enviado por cámara Ip.

- **Configuración de línea de conteo**

En la sección “Línea” se puede configurar la línea de conteo, entre las opciones que se dispone es mover el punto de inicio y fin, mover en el sentido horizontal, adicionalmente en este apartado se puede configurar el giro de la imagen para ajustar correctamente la línea de conteo.

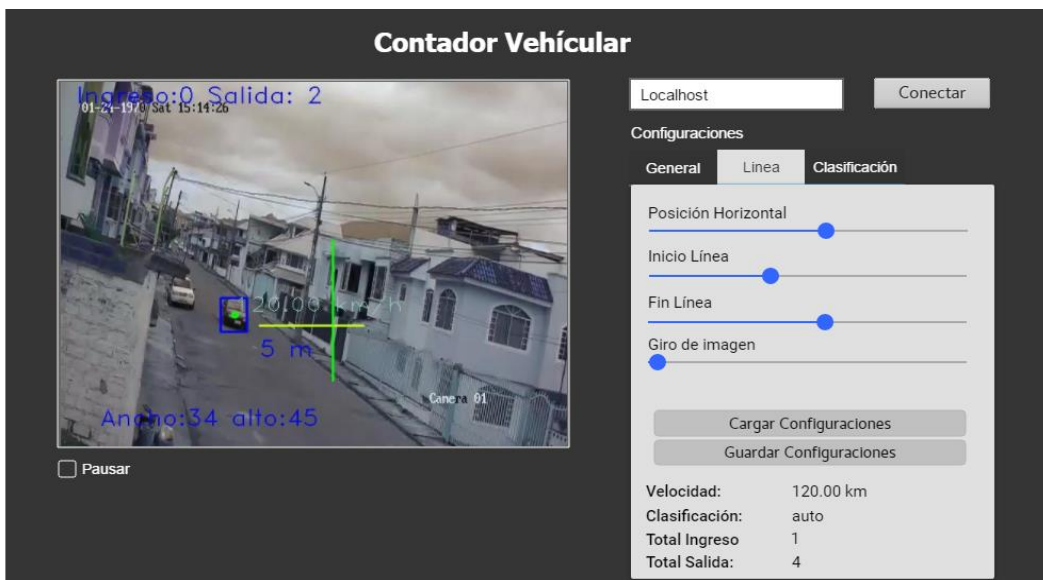


Figura 13. Pantalla de configuraciones de ajuste de línea

- **Configuraciones de clasificación**

En este apartado se configura los rangos de los tamaños para la clasificación de los vehículos detectados, esta clasificación se hace únicamente en base al tamaño.

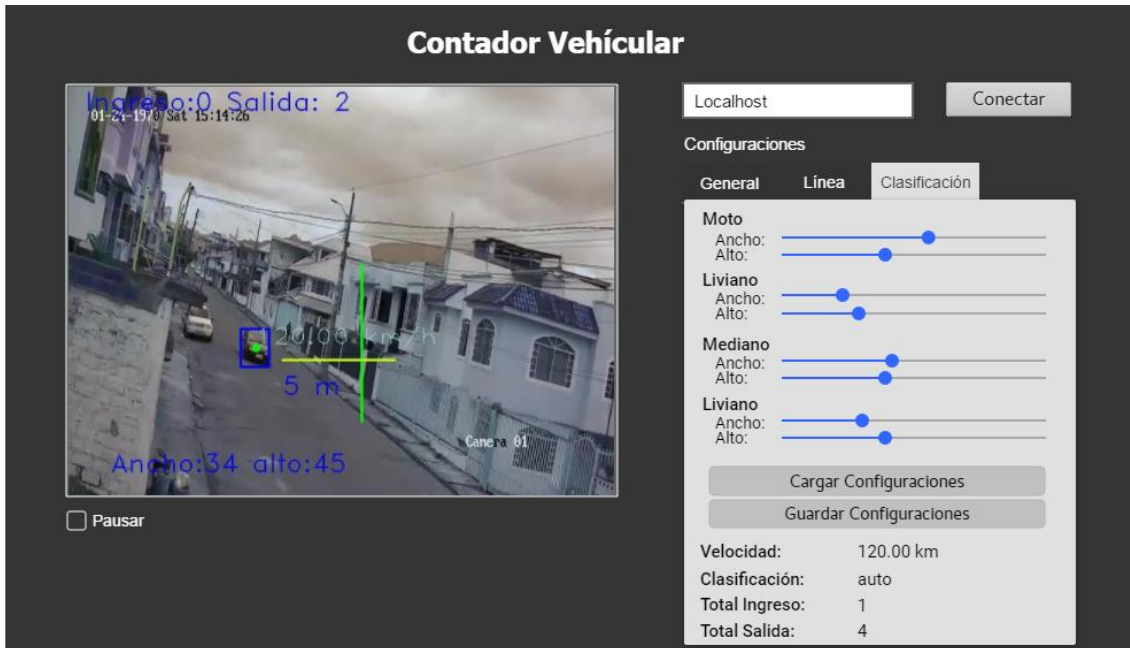


Figura 14. Pantalla para ajustes de clasificación por tamaño de vehículos.

2.5.2. Aplicación web de administración

En este apartado se mostrará los prototipos de la pantalla inicio, el login y la opción contador, los demás prototipos se los puede visualizar en el ANEXO X.

- **Pantalla inicio**

Por defecto siempre será dirigido a la vista principal de la página en la que se muestra una barra superior en la que se tiene la opción de inicio de sesión, como contenido muestra información general del sistema.



Figura 15. Pantalla principal de inicio de sistema web.

- **Inicio de sesión**

Cuando presiona sobre “Iniciar sesión” se muestra un formulario de login en el cual se solicita el ingreso de un usuario y contraseña que deberán ser proporcionados por el usuario administrador del sistema.

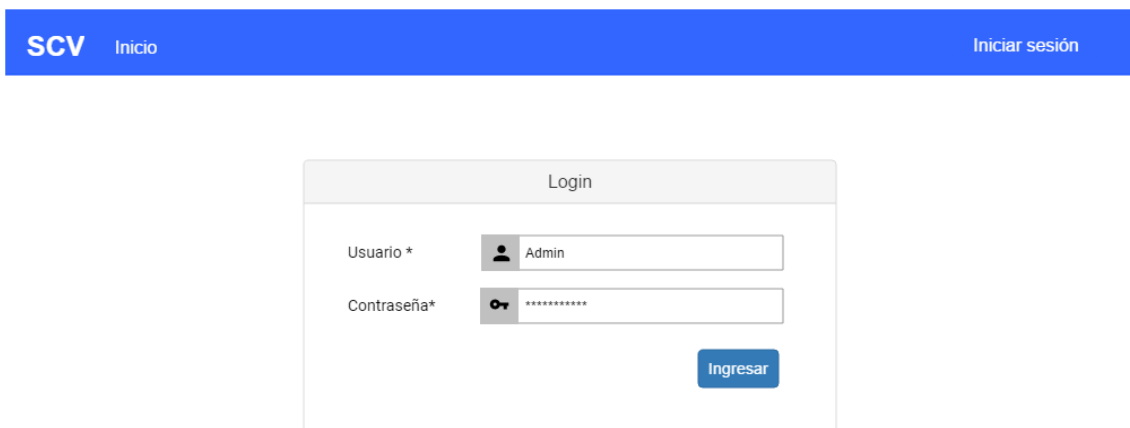
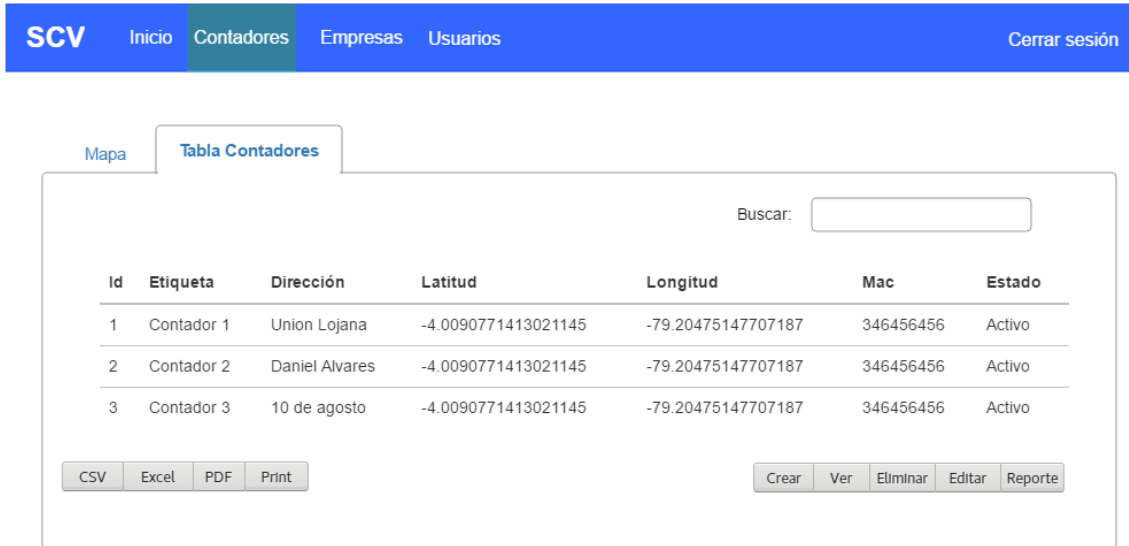


Figura 16. Login del sistema web.

- **Tabla de contadores**

En la opción “Tabla Contadores” nos muestra todos los contadores registrados, además las opciones de crear, modificar, eliminar y obtener reportes.



Mapa **Tabla Contadores**

Buscar:

Id	Etiqueta	Dirección	Latitud	Longitud	Mac	Estado
1	Contador 1	Union Lojana	-4.0090771413021145	-79.20475147707187	346456456	Activo
2	Contador 2	Daniel Alvares	-4.0090771413021145	-79.20475147707187	346456456	Activo
3	Contador 3	10 de agosto	-4.0090771413021145	-79.20475147707187	346456456	Activo

CSV Excel PDF Print Crear Ver Eliminar Editar Reporte

Figura 17. Tabla de contadores registrados

- **Formulación registro contador**

Cuando se presiona en el botón “Crear” se nos presenta un formulario para el ingreso de los datos del contador a registrar, además en la parte derecha un mapa en el cual podemos georreferenciar rápidamente.

Crear Contador

Etiqueta * :

Latitud * :

Longitud * :

Dirección * :

Mac * :

Estado * :

Tipo * :

Orientación * :

Figura 18. Formulario de registro de contador.

- **Reporte con presentación de datos en una tabla**

En la sección “Tabla de conteo” se presenta una tabla con los datos obtenidos en el rango de fecha seleccionado, con opciones para exportar el reporte.

Desde: Hasta:

Periodo: Full 5 min 10 min 15 min

Grafica **Tabla conteo** Volumen de tránsito Visualización

Buscar:

Fecha	Hora	Velocidad	Tipo	Evento
02/18/2018	19:11:40	61,05 km/h	auto	Salida
02/20/2018	19:11:41	102,25 km/h	auto	Salida

Figura 19. Reporte con presentación de datos en una tabla

- **Pantalla de reporte de volumen de tránsito**

En la sección volumen de tránsito se presenta una tabla que muestra los resultados de la aplicación de las fórmulas para el cálculo de volumen de tránsito semanal, mensual y anual.

Mes	Semana Nro	Tránsito semanal	Tránsito mensual
Febrero	6	Ingreso: 0 Salida 2	Ingreso: 0 Salida 203
	7	Ingreso: 0 Salida 201	

Volumen de tránsito anual

VTA Vehículos de ingreso: 0
VTA Vehículos de salida: 203

Figura 20. Pantalla de cálculo de volumen de tránsito.

3. TERCERA FASE: CODIFICACIÓN

Una vez que se realizó el diseño de las aplicaciones, se procede a la codificación, estableciendo previamente estándares que se va a utilizar.

Para realizar la implementación se utilizó Sistema Operativo Linux, Lenguaje de Programación C++, el IDE QT Creator, MySql para la Base de datos.

3.1. Estándares de programación

Según la metodología Programación Extrema (XP), recomienda seguir un proceso de desarrollo definiendo estándares para mantener el código consistente y facilitar su comprensión y escalabilidad.

Los estándares que se describen a continuación constituyen un conjunto de buenas

prácticas para el desarrollo de software, las mismas que tiene como objetivo facilitar la comprensión del código.

A continuación, se describen algunos estándares y patrones que se utilizarán para codificar:

- Todos los paquetes del directorio se encuentran en minúsculas.
- Todos los archivos del directorio se encuentran en minúsculas
- El código se encuentra comentado en ciertas partes, que se creyeron importantes para una mejor comprensión del mismo. Los comentarios siguen los siguientes formatos:
- Para comentar una línea de código:

```
// Comentario.
```

Para comentar un conjunto de líneas de código

```
/* Comentario.....
```

```
.....
```

```
*/
```

- Los nombres de las variables se encuentran basados en el estilo de escritura CamelCase.

3.2. Descripción de herramientas utilizadas para la implementación del software

- **Sistema Operativo Linux:** Sistema operativo de Software Libre que cuenta con distintas distribuciones como Ubuntu, Fedora, Open Suse, entre otros.
- **Lenguaje de Programación C++:** Potente Lenguaje de Programación que permite la fácil integración con las funciones de la Librería OpenCV.
- **IDE QT Creator:** Es un entorno de desarrollo libre, dispuesto para trabajar con C++ e interfaz gráfica.

- **Base de Datos MySQL:** Gestor de Base de Datos de la aplicación.
- **OpenCV:** Biblioteca de Visión por Computadora, proporciona los métodos necesarios para la detección de objetos como vehículos, personas, etc.

3.3. Codificación de los módulos del sistema

En esta sección se muestra la codificación de cada actividad establecida en cada módulo.

3.3.1. Codificación del Contador vehicular

3.3.1.1. Desarrollo de módulo Interfaz de configuración

En esta sección se presenta los datos obtenidos de la última configuración almacenada, estos parámetros son configurados en los elementos visuales para que el usuario pueda editarlos nuevamente como se crea conveniente y tenga el mejor ajuste para el conteo de vehículos.

```

QStringList parametros=trama.split("-")[1].split(",");
ui->spIdentificador->setValue(parametros.at(0).toInt());
ui->txtPuertoSerial->setText(parametros.at(1));
ui->spDistanciaAproximada->setValue(parametros.at(2).toInt());
ui->spOrientacion->setValue(parametros.at(3).toInt());
ui->spTiempoCaptura->setValue(parametros.at(4).toInt());
ui->spAngulo->setValue(parametros.at(5).toInt());
ui->spInicioX1->setValue(parametros.at(6).toInt());
ui->spInicioY1->setValue(parametros.at(7).toInt());
ui->spFinX1->setValue(parametros.at(8).toInt());
ui->spFinY1->setValue(parametros.at(9).toInt());
ui->chkMostrarClasificacion->setChecked(parametros.at(10).toInt()==1?true:false);
ui->spAreaAnchoMotocicleta->setValue(parametros.at(11).toInt());
ui->spAreaAltoMotocicleta->setValue(parametros.at(12).toInt());
ui->spAreaAnchoLiviano->setValue(parametros.at(13).toInt());
ui->spAreaAltoLiviano->setValue(parametros.at(14).toInt());
ui->spAreaAnchoMediano->setValue(parametros.at(15).toInt());
ui->spAreaAltoMediano->setValue(parametros.at(16).toInt());
ui->spAreaAnchoCamion->setValue(parametros.at(17).toInt());
ui->spAreaAltoCamion->setValue(parametros.at(18).toInt());
ui->txtUrlCamara->setText(parametros.at(19));
ui->spTipoConteo->setValue(parametros.at(20).toInt());
ui->spTipoCamara->setValue(parametros.at(21).toInt());
ui->spInicioX1_2->setValue(parametros.at(22).toInt());
ui->spInicioY1_2->setValue(parametros.at(23).toInt());
ui->spFinX1_2->setValue(parametros.at(24).toInt());
ui->spFinY1_2->setValue(parametros.at(25).toInt());

```

Figura 21. Código de la interfaz de configuración

- **Visualización de imagen**

Cuando la imagen es receptada como una array de bytes esta es transformada a un objeto QImage para luego ser seteada en el elemento visual y así el usuario pueda verificar la imagen receptada desde la cámara.

```
void MainWindow::onNewData(QByteArray data){  
    if(!ui->chkPausar->isChecked()){  
        QImage img = QImage::fromData(data,"JPG");  
        ui->lblImagen-> setPixmap(QPixmap::fromImage(img, Qt::ColorOnly));  
        ui->lblImagen->resize(ui->lblImagen->pixmap()->size());  
    }  
}
```

Figura 22. Código de la visualización de imagen

- **Configuración de posición de líneas de conteo**

Para configurar las líneas de conteo el usuario debe modificar las sliders según crea conveniente, en cada evento generado de cambio de posición se captura las posiciones y se envían para ser guardados en un archivo de configuración y ser cargado para realizar el conteo según estas variables.


```

socket->sendData("$$1,"+
    QString::number(ui->spIdentificador->value())+", "+
    ui->txtPuertoSerial->text()+"", "+
    QString::number(ui->spDistanciaAproximada->value())+", "+
    QString::number(ui->spOrientacion->value())+", "+
    QString::number(ui->spTiempoCaptura->value())+", "+
    QString::number(ui->spAngulo->value())+", "+
    QString::number(ui->spInicioX1->value())+", "+
    QString::number(ui->spInicioY1->value())+", "+
    QString::number(ui->spFinX1->value())+", "+
    QString::number(ui->spFinY1->value())+", "+
    QString::number(mostrarClasificacion)+"", "+
    QString::number(ui->spAreaAnchoMotocicleta->value())+", "+
    QString::number(ui->spAreaAltoMotocicleta->value())+", "+
    QString::number(ui->spAreaAnchoLiviano->value())+", "+
    QString::number(ui->spAreaAltoLiviano->value())+", "+
    QString::number(ui->spAreaAnchoMediano->value())+", "+
    QString::number(ui->spAreaAltoMediano->value())+", "+
    QString::number(ui->spAreaAnchoCamion->value())+", "+
    QString::number(ui->spAreaAltoCamion->value())+", "+
    ui->txtUrlCamara->text()+"", "+
    QString::number(ui->spTipoConteo->value())+", "+
    QString::number(ui->spTipoCamara->value())+", "+
    QString::number(ui->spInicioX1_2->value())+", "+
    QString::number(ui->spInicioY1_2->value())+", "+
    QString::number(ui->spFinX1_2->value())+", "+
    QString::number(ui->spFinY1_2->value())
);

```

Figura 23. Código de la configuración de posición de líneas de conteo

- **Configuración de la fuente de video**

Se puede configurar para que el software tomé la fuente de video que deseamos, esto lo hacemos utilizando la librería vlc para el protocolo rtsp, la implementación de Opencv para adquirir imagen de un archivo de video.

```

void MainWindow::abrirCamara(){
    if(conf->value("datos/tipocamara").toInt()==1){
        conf->setValue("datos/delaycamara",100);
        vlcV = new VlcOpenCV(conf->value("datos/urlcamara").toString().toStdString().c_str());
        QObject::connect(vlcV, SIGNAL(sendEvent(int)), this, SLOT(onVlcEvent(int)));
        vlcV->Start();
    }else{
        if(conf->value("datos/tipocamara").toInt()==2){
            conf->setValue("datos/delaycamara",40);
            string camara=conf->value("datos/urlcamara").toString().toStdString();
            cap.open(camara);
        }else{
            conf->setValue("datos/delaycamara",100);
            int camara=conf->value("datos/urlcamara").toInt();
            cap.open(camara);
        }
    }

    if(conf->value("datos/streaming").toInt()==1){
        cap.set(cv::CAP_PROP_FRAME_WIDTH,680);
        cap.set(cv::CAP_PROP_FRAME_HEIGHT,480);
        cap.set(cv::CAP_PROP_FPS,conf->value("datos/fps").toInt());
    }
}

```

Figura 24. Código de la configuración de la fuente de video

3.3.1.2. Desarrollo de módulo Contador vehicular

- Detección de vehículos

Para la detección de los vehículos se realiza aplicando el algoritmo de sustracción de fondo, luego se encuentra los contornos del resultado de la sustracción y de obtiene los objetos en la lista rects.

```

bg->setVarThreshold(50);
bg->apply(img,bg2);
findContours(bg2, contours, CV_RETR_EXTERNAL,CV_CHAIN_APPROX_NONE);
vector<vector<Point>> contours_poly( contours.size() );
vector<Rect> boundRect( contours.size() );
vector<Point2f>center( contours.size() );
vector<float>radius( contours.size() );

for( int i = 0; i < contours.size(); i++ ){
    approxPolyDP( Mat(contours[i]), contours_poly[i], 3.0, true );
    boundRect[i] = boundingRect( Mat(contours_poly[i]) );
    minEnclosingCircle( (Mat)contours_poly[i], center[i], radius[i] );
    if(boundRect[i].area()>1000){
        circle(img,center[i],3,Scalar(0,255,0),2,CV_AA);
        rectangle( img, boundRect[i], color, 2, 8, 0 );
        pts.push_back(center[i]);
        rects.push_back(boundRect[i]);
    }
}

```

Figura 25. Código de la detección de vehículos

- **Rastreo de vehículos en la imagen**

Para realizar el rastreo de los vehículos en la imagen, una vez hayan sido detectados actualiza la lista de los mismo en el objeto tracker que se encarga de almacenar la información de cada objeto detectado.

```
for( int i = 0; i < contours.size(); i++ ){
    approxPolyDP( Mat(contours[i]), contours_poly[i], 3.0, true );
    boundRect[i] = boundingRect( Mat(contours_poly[i]) );
    minEnclosingCircle( (Mat)contours_poly[i], center[i], radius[i] );
    if(boundRect[i].area()>1000){
        circle(img,center[i],3,Scalar(0,255,0),2,CV_AA);
        rectangle( img, boundRect[i], color, 2, 8, 0 );
        pts.push_back(center[i]);
        rects.push_back(boundRect[i]);
    }
}

if(pts.size()>0){
    tracker.Update(pts,rects);
}
```

Figura 26. Código del rastreo de vehículo en la imagen

- **Conteo de vehículos según línea de conteo**

Para realizar el conteo de vehículos tenemos una línea de referencia la misma que es previamente configurada desde la interfaz de configuración, luego para cada objeto detectado realizamos la comparación de sus posición actual y anterior respecto a la línea de conteo, si cumple la condición que el objeto está atravesando la línea ya sea de ingreso como de salida, se procede a generar el evento de conteo y envío de la trama hacia el servidor de datos.


```

public void update() {
    persist(PersistAction.UPDATE, ResourceBundle.getBundle("/Bundle").getString("ContadorUpdated"));
}

```

Figura 29. Código de editar contador

- **Eliminar contador**

Para la eliminación de un contador se realiza su borrado de la base de datos, luego se muestra un mensaje de eliminación correcta y se actualiza la información.

```

public void destroy() {
    persist(PersistAction.DELETE, ResourceBundle.getBundle("/Bundle").getString("ContadorDeleted"));
    if (!JsfUtil.isValidatationFailed()) {
        selected = null;
        items = null;
    }
}

```

Figura 30. Código de eliminar contador

- **Visualizar contador**

Para visualizar un contador se obtiene el objeto con la información y se presenta los datos en la vista mostrando su ubicación en el mapa.

```

<script>
    var contador =#{contadorController.getSelectedJson()};
    var mapView = new mapboxgl.Map(
        {
            container: 'mapMapboxView',
            style: 'mapbox://styles/mapbox/streets-v9',
            center: [contador.longitud, contador.latitud],
            zoom: 15
        });

    var marker = new mapboxgl.Marker()
        .setLngLat([contador.longitud, contador.latitud])
        .setPopup(new mapboxgl.Popup({offset: 25})
            .setHTML('<h5>' + contador.etiqueta + '</h5><p>' + contador.direccion + '</p>'))
        .addTo(mapView);
</script>

```

Figura 31. Código de visualizar contador

3.3.2.2. Desarrollo de módulo Reportes de conteo

- **Generar reporte de conteo vehicular**

Para generar el reporte se realiza la consulta de los datos de conteo que se encuentran en la base de datos, tomando como parámetros la fecha y hora de inicio y

la fecha hora fin, además del idContador.

```
public void obtenerConteo() {  
    mostrarTablaConteo = false;  
    items = getFacade().findByDate(dateDesde, dateHasta, idContador);  
  
    addMessage("Datos obtenidos correctamente" + items.toString());  
    createBarModel();  
    generateModel();  
  
    mostrarTablaConteo = true;  
}
```

Figura 32. Código de generar reporte

- **Exportar reporte de conteo vehicular**

Para la generación de los reportes primero se renderiza toda la información en una tabla para su vista previa y luego el usuario selecciona que tipo de reporte desea entre ellas excel, pdf o csv.

```

<b:dataTable style="width:100%" id="datalist"
value="#{reporteConteoController.items}"
var="item"
selectionMode="single"
excel="true"
csv="true"
pdf="true"
print="true"
>
<b:column>
<f:facet name="header">
<h:outputText value="#{bundle.ListConteoTitle_fecha}"/>
</f:facet>
<h:outputText value="#{item.fecha}">
<f:convertDateTime pattern="MM/dd/yyyy" />
</h:outputText>
</b:column>
<b:column>
<f:facet name="header">
<h:outputText value="#{bundle.ListConteoTitle_hora}"/>
</f:facet>
<h:outputText value="#{item.hora}">
<f:convertDateTime type="date" pattern="HH:mm:ss" timeZone
</h:outputText>
</b:column>
<b:column>
<f:facet name="header">
<h:outputText value="Velocidad"/>
</f:facet>
<h:outputText value="#{item.getVelocidadString()}" />
</b:column>
<b:column>
<f:facet name="header">
<h:outputText value="Tipo"/>
</f:facet>
<h:outputText value="#{item.getTipoVehiculoString()}" />
</b:column>
<b:column>
<f:facet name="header">
<h:outputText value="Evento"/>
</f:facet>
<h:outputText value="#{item.getEventoString()}" />
</b:column>

```

Figura 33. Código de exportar reporte

4. FASE CUARTA: PRUEBAS

Una vez construido el software de conteo vehicular se procede a realizar las pruebas correspondientes para comprobar su efectividad.

4.1. Pruebas de Aceptación

Las pruebas de aceptación se realizaron para cada aplicación con la finalidad de verificar y validar los requerimientos y funcionalidad de las mismas por parte del cliente, para esto se tomó como base las historias de usuario elaboradas en la fase de planificación.

Estas pruebas se llevaron a cabo por el Ingeniero Ing. Luis Roberto Jácome Galarza (Véase ANEXO 4), en donde se obtuvo un resultado exitoso ya que no se encontraron errores en ninguno de los casos de prueba realizados, concluyendo de esta manera que las aplicaciones cumplen con las expectativas del cliente, por lo que se establece esta prueba como aceptada (Véase ANEXO 5).

4.1.1. Ejecución de pruebas

A continuación, se describe cada una de las pruebas de aceptación correspondientes a las historias de usuario de cada módulo según la aplicación.

Módulos de la Aplicación web

Módulo No. 1: Gestión de usuarios

- Historia de Usuario 1

TABLA XXXIII. PRUEBA DE ACEPTACIÓN: REGISTRAR USUARIO

PRUEBAS DE ACEPTACION	
Caso de prueba: Registrar Usuario	
Numero de caso de prueba: 1	Numero de Historia de usuario: 1
Descripción: Verificar el registro de un nuevo usuario para el ingreso al sistema.	
Condición de ejecución: Haber ingresado al sistema de administración como administrador.	
Pasos:	

<ul style="list-style-type: none"> • El administrador hace click en el menú usuarios. • El administrador presiona en el botón “Crear” • El administrador ingresa los datos. • El administrador presiona en el botón de confirmación de registro.
Resultado esperado: Registro exitoso de un nuevo usuario del sistema de administración.
Resultado: Exitoso

- **Historia de usuario 2**

TABLA XXXIV. PRUEBA DE ACEPTACIÓN: EDITAR USUARIO

PRUEBAS DE ACEPTACION	
Caso de prueba: Editar Usuario	
Numero de caso de prueba: 2	Numero de Historia de usuario: 2
Descripción: Verificar la edición de un usuario registrado en el sistema.	
Condición de ejecución: Haber ingresado al sistema de administración como administrador.	
Pasos: <ul style="list-style-type: none"> • El administrador hace click en el menú usuarios. • El administrador selecciona el usuario a editar. • El administrador presiona en el botón “Editar” • El administrador modifica los datos. • El administrador presiona en el botón de confirmación de edición. 	
Resultado esperado: Edición exitosa de un usuario del sistema de administración.	
Resultado: Exitoso	

- **Historia de usuario 3**

TABLA XXXV. PRUEBA DE ACEPTACIÓN: ELIMINAR USUARIO

PRUEBAS DE ACEPTACION	
Caso de prueba: Eliminar Usuario	
Numero de caso de prueba: 3	Numero de Historia de usuario: 3
Descripción: Verificar la eliminación de un usuario registrado en el sistema.	
Condición de ejecución: Haber ingresado al sistema de administración como administrador.	
Pasos: <ul style="list-style-type: none"> • El administrador hace click en el menú usuarios. • El administrador selecciona el usuario a eliminar. • El administrador presiona en el botón “Eliminar” 	
Resultado esperado: Eliminación exitosa de un usuario del sistema de administración.	
Resultado: Exitoso	

- **Historia de usuario 4**

TABLA XXXVI. PRUEBA DE ACEPTACIÓN: VISUALIZAR USUARIO

PRUEBAS DE ACEPTACION	
Caso de prueba: Visualizar Usuario	
Numero de caso de prueba: 4	Numero de Historia de usuario: 4
Descripción: Verificar la presentación de los datos de un usuario registrado en el sistema.	
Condición de ejecución: Haber ingresado al sistema de administración como administrador.	
Pasos: <ul style="list-style-type: none"> • El administrador hace click en el menú usuarios. • El administrador selecciona el usuario a visualizar. • El administrador presiona en el botón “Ver”. 	
Resultado esperado:	

Presentación de los datos exitosa de un usuario del sistema de administración.

Resultado:

Exitoso

Módulo No. 2: Gestión de contadores

- **Historia de usuario 5**

TABLA XXXVII. PRUEBA DE ACEPTACIÓN: REGISTRAR CONTRADOR

PRUEBAS DE ACEPTACION	
Caso de prueba: Registrar Contador	
Numero de caso de prueba: 5	Numero de Historia de usuario: 5
Descripción: Verificar el registro de un nuevo contador vehicular en el sistema.	
Condición de ejecución: Haber ingresado al sistema de administración web como administrador.	
Pasos: <ul style="list-style-type: none">• El administrador hace click en el menú contadores.• El administrador presiona en el botón "Crear".• El administrador registra los datos del contador vehicular.• El administrador presiona en el botón de guardar.	
Resultado esperado: Registro exitoso de un nuevo contador en el sistema.	
Resultado: Exitoso	

- **Historia de usuario 6**

TABLA XXXVIII. PRUEBA DE ACEPTACIÓN: EDITAR CONTADOR

PRUEBAS DE ACEPTACIÓN	
Caso de prueba: Editar Contador	
Numero de caso de prueba: 6	Numero de Historia de usuario: 6
Descripción: Verificar la edición de un contador vehicular en el sistema.	

<p>Condición de ejecución:</p> <p>Haber ingresado al sistema de administración web como administrador.</p>
<p>Pasos:</p> <ul style="list-style-type: none"> • El administrador hace click en el menú contadores. • El administrador selecciona el contador a editar. • El administrador presiona en el botón “Editar”. • El administrador modifica los datos del contador vehicular. • El administrador presiona en el botón de guardar.
<p>Resultado esperado:</p> <p>Edición exitosa de un nuevo contador en el sistema.</p>
<p>Resultado:</p> <p>Exitoso</p>

- **Historia de usuario 7**

TABLA XXXIX. PRUEBA DE ACEPTACIÓN: ELIMINAR CONTADOR

PRUEBAS DE ACEPTACIÓN	
Caso de prueba: Eliminar Contador	
Numero de caso de prueba: 7	Numero de Historia de usuario: 7
Descripción:	
Verificar la eliminación de un contador vehicular registrado en el sistema.	
Condición de ejecución:	
Haber ingresado al sistema de administración web como administrador.	
Pasos:	
<ul style="list-style-type: none"> • El administrador hace click en el menú contadores. • El administrador selecciona el contador a eliminar. • El administrador presiona en el botón “Eliminar”. 	
Resultado esperado:	
Eliminación exitosa de un nuevo contador en el sistema.	
Resultado:	
Exitoso	

- **Historia de usuario 8**

TABLA XL. PRUEBA DE ACEPTACIÓN: VISUALIZAR CONTADOR

PRUEBAS DE ACEPTACIÓN	
Caso de prueba: Visualizar Contador	
Numero de caso de prueba: 8	Numero de Historia de usuario: 8
Descripción: Verificar la presentación de la información de un contador vehicular registrado en el sistema.	
Condición de ejecución: Haber ingresado al sistema de administración web como administrador.	
Pasos: <ul style="list-style-type: none"> • El administrador hace click en el menú contadores. • El administrador selecciona el contador a visualizar. • El administrador presiona en el botón “Ver”. 	
Resultado esperado: Presentación de la información de contador vehicular exitosa de un nuevo contador en el sistema.	
Resultado: Exitoso	

Módulo No. 3: Reportes de conteo

- **Historia de usuario 9**

TABLA XLI. PRUEBA DE ACEPTACIÓN: GENERAR REPORTE DE CONTEO

PRUEBAS DE ACEPTACIÓN	
Caso de prueba: Generar reporte de conteo	
Numero de caso de prueba: 9	Numero de Historia de usuario: 9
Descripción: Generar un reporte de los datos de conteo realizado por un contador vehicular.	
Condición de ejecución: Haber ingresado al sistema de administración.	
Pasos: <ul style="list-style-type: none"> • El usuario hace click en el menú contadores. 	

<ul style="list-style-type: none"> • El usuario selecciona el contador. • El usuario presiona en el botón “Reporte”. • Seleccionar fecha para consulta de datos. • Presionar en el botón obtener.
<p>Resultado esperado:</p> <p>Presentación del reporte de conteo realizado por el software según el rango de fecha seleccionado.</p>
<p>Resultado:</p> <p>Exitoso</p>

- **Historia de usuario 10**

TABLA XLII. PRUEBA DE ACEPTACIÓN: EXPORTAR REPORTE DE CONTEO

PRUEBAS DE ACEPTACIÓN	
Caso de prueba: Exportar reporte de conteo	
Numero de caso de prueba: 10	Numero de Historia de usuario: 10
Descripción: Exportar datos de un reporte de conteo realizado.	
Condición de ejecución: Haber ingresado al sistema de administración, y haber generado un reporte.	
Pasos: <ul style="list-style-type: none"> • Presionar en el botón obtener, para generar el reporte. • Presionar en el botón para exportar los datos del reporte ya sea pdf, excel o csv. 	
Resultado esperado: Exportación de los datos exitosa.	
Resultado: Exitoso	

1.2.2. Módulos del Sistema de Conteo Vehicular

Módulo No. 1: Interfaz de configuración

- Historia de usuario 11

TABLA XLIII. PRUEBA DE ACEPTACIÓN: VISUALIZACION DE IMAGEN

PRUEBAS DE ACEPTACIÓN	
Caso de prueba: Visualización de imagen	
Numero de caso de prueba: 11	Numero de Historia de usuario: 11
Descripción: Visualizar la imagen de la cámara para comprobar la detección de los vehículos.	
Condición de ejecución: Haber ejecutado la aplicación de escritorio.	
Pasos: <ul style="list-style-type: none">• Ingresar el host del software de conteo.• Presionar en el botón conectar.	
Resultado esperado: Visualización de la imagen exitosa.	
Resultado: Exitoso	

- Historia de usuario 12

TABLA XLIV. PRUEBA DE ACEPTACIÓN: CONFIGURAR LINEA DE CONTEO

PRUEBAS DE ACEPTACIÓN	
Caso de prueba: Configurar línea de conteo	
Numero de caso de prueba: 12	Numero de Historia de usuario: 12
Descripción: Verificar la configuración de la línea de conteo de vehículos.	
Condición de ejecución: Haber ejecutado la aplicación de escritorio.	
Pasos: <ul style="list-style-type: none">• Ingresar el host del software de conteo.• Presionar en el botón conectar.	

<ul style="list-style-type: none"> • Configurar la línea de conteo con los sliders.
Resultado esperado: Configuración de línea de conteo la imagen exitosa.
Resultado: Exitoso

- **Historia de usuario 13**

TABLA XLV. PRUEBA DE ACEPTACIÓN: CONFIGURAR FUENTE DE VIDEO

PRUEBAS DE ACEPTACIÓN	
Caso de prueba: Configurar fuente video	
Numero de caso de prueba: 13	Numero de Historia de usuario: 13
Descripción: Verificar la configuración de la fuente de video para procesamiento de imagen.	
Condición de ejecución: Haber ejecutado la aplicación de escritorio.	
Pasos: <ul style="list-style-type: none"> • Ingresar el host del software de conteo. • Presionar en el botón conectar. • Ingresar la fuente de video pudiendo ser esta una url de la cámara ip. 	
Resultado esperado: Configuración de la fuente de video exitosa.	
Resultado: Exitoso	

Módulo No.2: Contador vehicular

- Historia de usuario 14

TABLA XLVI. PRUEBA DE ACEPTACIÓN: DETECTAR VEHICULOS

PRUEBAS DE ACEPTACIÓN	
Caso de prueba: Detectar vehículos	
Numero de caso de prueba: 14	Numero de Historia de usuario: 14
Descripción: Verificar la detección de los vehículos según la fuente de video configurada.	
Condición de ejecución: Haber ejecutado la aplicación de escritorio.	
Pasos: <ul style="list-style-type: none">• Ingresar el host del software de conteo.• Presionar en el botón conectar.	
Resultado esperado: Visualización de la detección de vehículos en el video exitosa.	
Resultado: Exitoso	

- Historia de usuario 15

TABLA XLVII. PRUEBA DE ACEPTACIÓN: RASTREO DE VEHICULOS EN LA IMAGEN

PRUEBAS DE ACEPTACIÓN	
Caso de prueba: Rastreo de vehículos en la imagen	
Numero de caso de prueba: 15	Numero de Historia de usuario: 15
Descripción: Verificar la detección y seguimiento de los vehículos según la fuente de video configurada.	
Condición de ejecución: Haber ejecutado la aplicación de escritorio.	
Pasos: <ul style="list-style-type: none">• Ingresar el host del software de conteo.• Presionar en el botón conectar.	
Resultado esperado:	

Visualización de la detección y seguimiento de vehículos en el video exitosa.
Resultado: Exitoso

- **Historia de usuario 16**

TABLA XLVIII. PRUEBA DE ACEPTACIÓN: CONTEO DE VEHICULOS

PRUEBAS DE ACEPTACIÓN	
Caso de prueba: Conteo de vehículos	
Numero de caso de prueba: 16	Numero de Historia de usuario: 16
Descripción: Verificar el conteo de vehículos según la configuración de la línea de conteo.	
Condición de ejecución: Haber ejecutado la aplicación de escritorio.	
Pasos: <ul style="list-style-type: none"> • Ingresar el host del software de conteo. • Presionar en el botón conectar. • Configurar línea de conteo. 	
Resultado esperado: Visualización del conteo de vehículos en el video exitosa.	
Resultado: Exitoso	

Verificación del cumplimiento de requerimientos funcionales

Para verificar el cumplimiento de los requerimientos funcionales, se utilizaron cada una de las pruebas de aceptación de cada historia de usuario. La Tabla L muestra el cumplimiento de los requerimientos

TABLA XLIX. VERIFICACIÓN DEL CUMPLIMIENTO DE REQUERIMIENTOS FUNCIONALES

REQUERIMIENTOS FUNCIONALES		
Requerimiento	Descripción	Cumplimiento
RF01	Detección de vehículos.	✓
RF02	Conteo de vehículos según el sentido	✓

	en que transiten.	
RF03	Envío datos de conteo a un servidor.	✓
RF04	Almacenamiento de datos local.	✓
RF05	Inicio de software con el arranque del sistema.	✓
RF06	Configuración de la fuente de video.	✓
RF07	Interfaz de configuración de software de conteo.	✓
RF08	Gestión de usuarios en aplicación web.	✓
RF09	Gestión de contadores en aplicación web.	✓
RF10	Reportes de conteo en aplicación web	✓
RF11	Georreferenciación de contadores en aplicación web.	✓

4.2. Pruebas de Rendimiento

Esta prueba se realizó al sistema de conteo vehicular.

Para desarrollarla se utilizó 8 videos grabados en distintos lugares de la ciudad de Loja en donde existe más flujo vehicular. La evaluación de los resultados se realizó de la siguiente manera:

4.2.1.1. Prueba 1

TABLA L. INFORMACIÓN DEL VIDEO DE PRUEBA 1 PARA LA DETECCIÓN DE VEHÍCULOS.

Información del Video	
Video	Video1.mp4
Ubicación	Puente peatonal Av. Universitaria y Mercadillo.
Duración	4.07 minutos
Sentido	Ingreso
Contador Software	81 vehículos
Contador Manual	81 vehículos

TABLA LII. 51 MATRIZ DE CONFUSIÓN PARA EL PRIMER VIDEO DE DETECCIÓN DE VEHÍCULOS.

Matriz de Confusión			
		Resultado de Clasificación	
		Vehículos	No Vehículos
Instancias Reales	Vehículos	81	0
	No Vehículos	0	0
Cálculo de Valores			
		Exactitud	1
		Precisión	1
		Sensibilidad	1
		Especificidad	1
		F1 Score	1

Para la primera prueba se realizó con un video capturado desde el puente peatonal ubicado en la avenida Universitaria y calle Mercadillo de la ciudad de Loja, la cámara fue ubicada con una vista frontal al tránsito de los vehículos, los resultados obtenidos de la prueba demuestran un conteo exacto con referencia al conteo realizado manualmente.

4.2.2. Prueba 2

TABLA LIII. INFORMACIÓN DEL VIDEO DE PRUEBA 2 PARA LA DETECCIÓN DE VEHÍCULOS

Información del Video	
Video	Video2.mp4
Ubicación	Puente peatonal Av. Universitaria y Mercadillo.
Duración	3.17 minutos
Sentido	Salida
Contador Software	46 vehículos
Contador Manual	46 vehículos

TABLA LIV. 53 MATRIZ DE CONFUSIÓN PARA EL PRIMER VIDEO DE DETECCIÓN DE VEHÍCULOS.

Matriz de Confusión			
		Resultado de Clasificación	
		Vehículos	No Vehículos
Instancias Reales	Vehículos	46	0
	No Vehículos	0	0
Cálculo de Valores			
		Exactitud	1
		Precisión	1
		Sensibilidad	1
		Especificidad	1
		F1 Score	1

Para la segunda prueba se realizó con un video capturado desde el puente peatonal ubicado en la avenida Universitaria y calle Mercadillo de la ciudad de Loja, la cámara fue ubicada con una vista frontal al tránsito de los vehículos, los resultados obtenidos de la prueba demuestran un conteo exacto con referencia al conteo realizado manualmente, la duración del video de prueba es de 3.17 minutos.

4.2.3. Prueba 3.

TABLA LV. INFORMACIÓN DEL VIDEO DE PRUEBA 3 PARA CONTEO DE VEHÍCULOS

Información del Video	
Video	Video3.mp4
Ubicación	Puente peatonal Av. Universitaria y Mercadillo.
Duración	3.24 minutos
Sentido	Salida
Contador Software	59 vehículos
Contador Manual	60 vehículos

TABLA LVI. 54 MATRIZ DE CONFUSIÓN PARA EL PRIMER VIDEO DE DETECCIÓN DE VEHÍCULOS.

Matriz de Confusión			
		Resultado de Clasificación	
		Vehículos	No Vehículos
Instancias Reales	Vehículos	59	1
	No Vehículos	0	0
Cálculo de Valores			
		Exactitud	0.98
		Precisión	1
		Sensibilidad	0.98
		Especificidad	1
		F1 Score	0.99

Para la tercera prueba se realizó con un video capturado desde el puente peatonal ubicado en la avenida Universitaria y calle Mercadillo de la ciudad de Loja, la cámara fue ubicada con una vista frontal al tránsito de los vehículos la duración del video de prueba es de 3.24 minutos, los resultados obtenidos de la prueba demuestran que el software no pudo realizar la detección de un vehículo.

4.2.4. Prueba 4

TABLA LVII. INFORMACIÓN DEL VIDEO DE PRUEBA 4 PARA CONTEO DE VEHÍCULOS

Información del Video	
Video	Video4.mp4
Ubicación	Puente peatonal Av. Manual Agustín Aguirre y Mercadillo.
Duración	4.23 minutos
Sentido	Ingreso
Contador Software	98 vehículos
Contador Manual	96 vehículos

TABLA LVIII.56 MATRIZ DE CONFUSIÓN PARA EL PRIMER VIDEO DE DETECCIÓN DE VEHÍCULOS.

Matriz de Confusión			
		Resultado de Clasificación	
		Vehículos	No Vehículos
Instancias Reales	Vehículos	96	0
	No Vehículos	2	0
Cálculo de Valores			
		Exactitud	0.97
		Precisión	0.97
		Sensibilidad	1
		Especificidad	1
		F1 Score	0.98

Para la cuarta prueba se realizó con un video capturado desde el puente peatonal ubicado en la avenida Manuel Agustín Aguirre y calle Mercadillo de la ciudad de Loja, la cámara fue ubicada con una vista frontal al tránsito de los vehículos la duración del video de prueba es de 4.23 minutos, los resultados obtenidos de la prueba demuestran que el software realizo el conteo de 2 vehículos más al conteo manual.

4.2.5. Prueba 5

TABLA LIX. INFORMACIÓN DEL VIDEO DE PRUEBA 5 PARA CONTEO DE VEHÍCULOS

Información del Video	
Video	Video5.mp4
Ubicación	Puente peatonal Av. Manuel Agustín Aguirre y Mercadillo.
Duración	3.37 minutos
Sentido	Ingreso
Contador Software	29 vehículos
Contador Manual	30 vehículos

TABLA LX. 57 MATRIZ DE CONFUSIÓN PARA EL PRIMER VIDEO DE DETECCIÓN DE VEHÍCULOS.

Matriz de Confusión			
		Resultado de Clasificación	
		Vehículos	No Vehículos
Instancias Reales	Vehículos	29	1
	No Vehículos	0	0
Cálculo de Valores			
		Exactitud	0.96
		Precisión	1
		Sensibilidad	0.96
		Especificidad	1
		F1 Score	0.98

Para la quinta prueba se realizó con un video capturado desde el puente peatonal ubicado en la avenida Manuel Agustín Aguirre y calle Mercadillo de la ciudad de Loja, la cámara fue ubicada con una vista frontal al tránsito de los vehículos la duración del video de prueba es de 3.37 minutos, los resultados obtenidos de la prueba demuestran que el software no realizó el conteo de 1 vehículo.

4.2.6. Prueba 6

TABLA LVIII. INFORMACIÓN DEL VIDEO DE PRUEBA 6 PARA CONTEO DE VEHÍCULOS

Información del Video	
Video	Video6.mp4
Ubicación	Puente peatonal Av. 8 de diciembre, Terminal Terrestre.
Duración	5.11 minutos
Sentido	Salida
Contador Software	146 vehículos
Contador Manual	149 vehículos

TABLA LXII.59 MATRIZ DE CONFUSIÓN PARA EL PRIMER VIDEO DE DETECCIÓN DE VEHÍCULOS.

Matriz de Confusión			
		Resultado de Clasificación	
		Vehículos	No Vehículos
Instancias Reales	Vehículos	146	3
	No Vehículos	0	0
Cálculo de Valores			
		Exactitud	0.98
		Precisión	1
		Sensibilidad	0.98
		Especificidad	1
		F1 Score	0.99

La sexta prueba se realizó con un video capturado desde el puente peatonal ubicado en la avenida 8 de diciembre, Terminal Terrestre de la ciudad de Loja, la cámara fue ubicada con una vista frontal al tránsito de los vehículos la duración del video de prueba es de 5.11 minutos, los resultados obtenidos de la prueba demuestran que el software no realizó el conteo de 3 vehículos.

Prueba 7

TABLA LX. INFORMACIÓN DEL VIDEO DE PRUEBA 7 PARA CONTEO DE VEHÍCULOS

Información del Video	
Video	Video7.mp4
Ubicación	Puente peatonal Av. 8 de diciembre, Terminal Terrestre.
Duración	3.26 minutos
Sentido	Ingreso
Contador Software	87 vehículos
Contador Manual	85 vehículos

TABLA LXIV.61 MATRIZ DE CONFUSIÓN PARA EL PRIMER VIDEO DE DETECCIÓN DE VEHÍCULOS.

Matriz de Confusión			
		Resultado de Clasificación	
		Vehículos	No Vehículos
Instancias Reales	Vehículos	85	0
	No Vehículos	2	0
Cálculo de Valores			
		Exactitud	0.97
		Precisión	0.97
		Sensibilidad	1
		Especificidad	1
		F1 Score	0.98

La séptima prueba se realizó con un video capturado desde el puente peatonal ubicado en la avenida 8 de diciembre, Terminal Terrestre de la ciudad de Loja, la cámara fue ubicada con una vista frontal al tránsito de los vehículos la duración del video de prueba es de 5.11 minutos, los resultados obtenidos de la prueba demuestran que el software realizó el conteo de 2 vehículos demás en referencia al conteo manual.

4.2.7. Prueba 8

TABLA LXII. INFORMACIÓN DEL VIDEO DE PRUEBA 8 PARA CONTEO DE VEHÍCULOS

Información del Video	
Video	Video8.mp4
Ubicación	Puente peatonal Av. 8 de diciembre, Terminal Terrestre.
Duración	3.30 minutos
Sentido	Salida
Contador Software	77 vehículos
Contador Manual	74 vehículos

TABLA LXVI.63 MATRIZ DE CONFUSIÓN PARA EL PRIMER VIDEO DE DETECCIÓN DE VEHÍCULOS.

Matriz de Confusión			
		Resultado de Clasificación	
		Vehículos	No Vehículos
Instancias Reales	Vehículos	74	0
	No Vehículos	3	0
Cálculo de Valores			
		Exactitud	0.96
		Precisión	0.96
		Sensibilidad	1
		Especificidad	1
		F1 Score	0.98

La octava prueba se realizó con un video capturado desde el puente peatonal ubicado en la avenida 8 de diciembre, Terminal Terrestre de la ciudad de Loja, la cámara fue ubicada con una vista frontal al tránsito de los vehículos la duración del video de prueba es de 3.30 minutos, los resultados obtenidos de la prueba demuestran que el software realizó el conteo de 3 vehículos adicionales.

G. DISCUSIÓN

1. Desarrollo de la Propuesta Alternativa

El presente Trabajo de Titulación “Sistema para detección y conteo vehicular aplicando técnicas de visión artificial”, se lo desarrollo en cuatro fases con el propósito de cumplir cada uno de los objetivos planteados. La primera fase se realizó el análisis, la segunda el diseño, la tercera la codificación y la cuarta fase las pruebas.

A continuación, se describe las actividades realizadas en cada una de las fases, así como los resultados obtenidos en las mismas.

OBJETIVO ESPECÍFICO 1: Analizar los diferentes algoritmos de detección y reconocimiento de objetos.

Para obtener los resultados planteados en este objetivo se realizó una investigación de las técnicas y algoritmos existentes para realizar la detección y reconocimiento de objetos, se describe la técnica de los clasificadores, así como también las técnicas de sustracción de fondo para la detección de objetos en movimiento.

Además, se realiza una descripción del software disponible para realizar aplicaciones con visión por computador con información detallada sobre las mismas lo cual nos ayudó para realizar la elección de las más idóneas para el desarrollo del software.

OBJETIVO ESPECÍFICO 2: Diseñar e implementar el sistema para la detección y conteo de vehículos.

Para el cumplimiento de este objetivo se desarrollaron tres fases: La fase de Análisis, Diseño y la Fase de Codificación.

En la Fase de Análisis se recolecto la información para determinar los requisitos de las aplicaciones utilizando la técnica de la entrevista que se realizó al Ing. Luis Roberto Jácome Galarza (Véase ANEXO 1), posteriormente se clasifíco la información en donde se estableció la lista de requerimientos tanto funcionales como no funcionales. Seguidamente se elaboró el Documento de Especificación de Requisitos (ERS) basado en las normativas IEEE830-98 en el cual se detallan las funcionalidades y restricciones de las aplicaciones, convirtiéndose este documento en la base fundamental para el proceso de desarrollo del producto final (Véase ANEXO 2), luego

se realizó las historias de usuario y se estableció los módulos de las aplicaciones que componen el sistema. Todos estos resultados se los obtuvo en base a la metodología XP (Véase sección 1: PRIMERA FASE: Análisis).

En la Fase del Diseño se utilizó la información obtenida en la Fase de Análisis. Como primera actividad se construyeron los diagramas que sirvieron de base para la codificación del software, entre ellos el diagrama de clases y el modelo entidad-relación, se elaboraron las tarjetas CRC, se realizó la arquitectura del sistema y finalmente los prototipos de las aplicaciones (Véase sección 2: SEGUNDA FASE: Diseño).

En la Fase de codificación se realizó la implementación del sistema de conteo vehicular, primeramente, se establecieron los estándares de programación, después con la ayuda del Framework Jsf se desarrolló el software de administración web en el cual se registran los contadores y se realiza reportes de los datos de conteo recolectados por los contadores y con el Framework Qt y librería Opencv se desarrolló el software de conteo vehicular (Véase sección 3: TERCERA FASE: Codificación).

OBJETIVO ESPECÍFICO 3: Realizar y ejecutar un plan de pruebas que permita evaluar al sistema en cada uno de sus módulos.

Para cumplir con este objetivo se realizaron las pruebas al sistema de conteo vehicular, en las cuales se encuentran las pruebas de aceptación que permitieron verificar la funcionalidad de las aplicaciones y a su vez comprobar si se cumple los requisitos establecidos en Fase de Análisis, dando como resultado la aprobación del sistema de conteo vehicular por parte del cliente (Véase ANEXO 5).

También se realizaron pruebas de rendimiento con 8 videos grabados en distintos puntos de la ciudad de Loja, con lo cual se verificó la detección de los vehículos de forma correcta (Véase sección 4: CUARTA FASE: Pruebas, Apartado 7.2).

OBJETIVO ESPECÍFICO 4: Divulgar la información obtenida en los procesos de desarrollo e implementación del proyecto a la comunidad científica.

Para cumplir con este objetivo se realizó un artículo científico en el cual se describe los pasos más importantes para el desarrollo del software de conteo aplicando técnicas de visión por computador, además se describe las diferentes técnicas de visión artificial y la elección de la más adecuada para el desarrollo del sistema de

conteo vehicular.

2. Valoración Técnica - Económica - Ambiental

Para el desarrollo del presente Trabajo de Titulación (TT) se lo realizó con diferentes recursos técnicos, económicos y ambientales.

Los recursos técnicos fueron de gran importancia debido a que permitieron el desarrollo del proyecto logrando de esta manera cumplir con un desarrollo eficiente del sistema de conteo vehicular.

En el aspecto económico los gastos que presentaron son mínimos ya que se utilizó herramientas de código libre para el desarrollo del sistema; además permitió ahorrar tiempo y dinero ya que no necesito adquirir ninguna licencia del software privativo para el desarrollo de este proyecto.

En el aspecto ambiental contribuye positivamente ya que el sistema permitirá una toma de decisiones efectiva para reducir la congestión vehicular en la ciudad. Este software se ejecuta en cualquier computador con sistema operativo Linux incluyendo mini pcs y con cualquier cámara evitando así la utilización de elementos contaminantes.

A continuación, se detallan los recursos empleados y el coste respectivo:

2.1. Recursos humanos

La TABLA LIX, muestra los recursos humanos empleados en el desarrollo del proyecto, como son el investigador y el director del Trabajo de titulación que brindó asesoramiento continuo para su correcta culminación.

TABLA LXIVVII. RECURSOS HUMANOS

Recurso	Cantidad	Número de Horas	Valor Unitario	Valor Total
Fabricio Sisalima	1	500	\$ 5.00	\$2500.00
Director del Trabajo de Titulación	1	200	\$ 0.00	\$ 0.00
			Subtotal:	\$2500.00

El costo de las horas del Director del Trabajo de Titulación se tomó en cero ya que son horas de tutorías y son cubiertas por de la Universidad Nacional de Loja.

2.2. Recursos materiales

La TABLA LX, muestra los recursos materiales utilizados durante el desarrollo del proyecto.

TABLA LXVIII. RECURSOS MATERIALES

Recursos	Cantidad	Valor Unitario	Valor Total
Materiales de Oficina(lápiz, borradores, perfiles, cds entre otros)	1	\$20.00	\$20.00
Impresiones	3	\$10.00	\$30.00
Anillado	3	\$1.25	\$3.75
Subtotal:			\$53.75

2.3. Recursos Técnicos y Tecnológicos

En la TABLA LXI, se detalla los recursos técnicos y tecnológicos empleados en el desarrollo del proyecto:

TABLA LXIX. RECURSOS TÉCNICOS Y TECNOLÓGICOS

Recursos	Cantidad	Valor Unitario	Número Horas	Valor total
Recursos de Hardware				
Portátil	1	\$1.00	500.00	\$500.00
Flash Memory	1	\$0.20	100.00	\$20.00
Recursos de Software				
Netbeans	1	\$0.00	\$0.00	\$0.00
Qt Framework	1	\$0.00	\$0.00	\$0.00
NodeJs	1	\$0.00	\$0.00	\$0.00
Recursos de Telecomunicaciones				
Internet		\$0.40	800.00	\$320.00
Subtotal:				\$840.00

2.4. Resumen de Presupuesto

En la TABLA LXII, se presenta el resumen del coste total del proyecto de titulación, para esto se realizó la suma de todos recursos detallados anteriormente.

TABLA LXVIII. RESUMEN DE PRESUPUESTO

Descripción	Total
Talento Humano	\$2500.00
Recursos Materiales	\$53.75
Recursos Técnicos	\$840.00
Total:	\$3393.75

H. CONCLUSIONES

Una vez finalizado el Trabajo de Titulación (TT) se puede expresar las siguientes conclusiones

- Existen una gran variedad de algoritmos para lograr detectar objetos como SIFT, SURF, ORB o aplicando clasificadores en cascada, sin embargo, no todos se adaptan al problema de conteo de vehículos en movimiento, o requieren gran capacidad de procesamiento, los algoritmos que conviene utilizar para este problema son los de análisis de movimiento.
- La aplicación del algoritmo backgroundSubtractorMog2 permitió la detección correcta de los objetos en movimiento sin la necesidad alto procesamiento del computador y adaptándose cambios en el ambiente.
- La aplicación del algoritmo de rastreo Kalman permitió de manera eficiente el rastreo en la imagen de todos los objetos detectados y la persistencia de sus estados mientras estos se desplazan en la imagen.
- La utilización de algoritmos que no requieren gran capacidad de procesamiento permitió ejecutar el software de conteo vehicular en equipos de gama baja, inclusive en miniordenadores como Raspberry Pi o similares.
- La utilización de la librería Opencv facilitó en gran medida el desarrollo de la aplicación de visión artificial ya que cuenta con la implementación de múltiples algoritmos para procesamiento de imágenes, además de ser una librería open source cuenta con una gran comunidad que apoya diariamente el desarrollo de la misma.

I. RECOMENDACIONES

Una vez finalizado el Trabajo de Titulación (TT) se puede expresar las siguientes recomendaciones.

- OpenCV es una gran ayuda para realizar el procesamiento de imágenes, sin embargo, para la adquisición de las mismas se recomienda utilizar la librería Vlc que nos proporciona una mejor calidad en la decodificación de las imágenes utilizando cámaras de tipo Ip.
- Se recomienda el desarrollo de aplicaciones de procesamiento de imágenes en lenguaje C++, ya que las librerías se encuentran desarrolladas en este lenguaje, aunque existen implementaciones para otros lenguajes de programación por lo general son envoltorios de la librería nativa, como consecuencia tenemos una reducción en la eficiencia de procesamiento que siendo mínima a la hora de ejecutar el software en plataformas de pocos recursos la diferencia en el rendimiento es muy notoria.
- La utilización de algoritmos livianos es recomendable para la distribución del software como un producto final sea económicamente viable y no requiera de equipos de alto costo para su ejecución, y además facilita la reutilización para otras plataformas como pueden ser las móviles.
- La iluminación en sistemas de visión artificial debe ser controlada en un rango de aceptación ya que es un factor importante para el éxito del sistema.

J. REFERENCIAS

- [1] C. Alberto and Q. Guti, "METODOLOGÍA PARA LA DETECCIÓN DE INCIDENTES DE TRÁFICO A PARTIR DE IMÁGENES DIGITALES EMPLEANDO TÉCNICAS DE INTELIGENCIA ARTIFICIAL," *Univ. Nac. Colomb. Sede Medellín*, p. 124, 2012.
- [2] V. Gabriel, "Ingeniería De Tránsito," *Esc. Ing. Civ.*, pp. 1–176, 2010.
- [3] A. L. J. Alejandro, "Algoritmo automático de detección y aforo vehicular en tiempo real en horario diurno," p. 132, 2014.
- [4] Guisselle Montoya, "Ingeniería De Transito," 2005.
- [5] D. G. D. O. DEL TERRITORIO, "Manual de Estudios de Ingeniería de Tránsito," *Subsecr. Desarro. Urbano y Ordenación del Territ.*, vol. 1, p. 231, 2009.
- [6] Departamento de Ingeniería Mecánica, "Ingeniería del tráfico, parámetros fundamentales," pp. 1–26, 2000.
- [7] E. Castillo, M. Guti, and E. Castillo, *Sistemas Expertos y Modelos de Redes Probabilísticas*. .
- [8] P. Ponce Cruz, *Inteligencia Artificial Con Aplicaciones a La Ingeniería*. 2010.
- [9] C. González, "Lógica Difusa," *Técnicas de Softcomputing*, p. 29.
- [10] F. Izaurieta, "Redes Neuronales Artificiales," pp. 1–15.
- [11] A. Serrano, E. Soria, and J. Martín, "Redes Neuronales Artificiales," *Open Course Ware*, p. 145.
- [12] J. Ramiro and M. De Dios, "Visión Por Computador."
- [13] S. D. C. Zientzia-astea, "Visión por computador ¿ Qué es la visión por computador ?"
- [14] D. Trujillo Fernández, "Aplicación de Metodologías Machine Learning en la Gestión de Riesgo de Crédito," p. 87, 2017.

- [15] J. Tahuiton Mora, "Arquitectura de software para aplicaciones Web," Centro De Investigación y De Estudios Avanzados Del Instituto Politécnico Nacional.México, DF, 2011.
- [16] W. Antensteiner, Development of a Front-End-Module for New Web Based Training Software. Citeseer, 1999.
- [17] S. Luj, "C++ paso a paso," p. 44, 2012.
- [18] D. J. Chamba Saca, "Creación e implantación de una herramienta web para la supervisión, control y monitoreo de presupuestos en los proyectos impulsados por el Ministerio de Inclusión Económica y Social (MIES) de Zamora Chinchipe." 2016.
- [19] J. Garrett, "Ajax: A new approach to web applications," 2005.
- [20] Opencv dev Team, "OPEN CV," 2018. [Online]. Available: <https://opencv.org/about.html>.
- [21] Opencv dev Team, "Introducción," 2015. [Online]. Available: <https://docs.opencv.org/2.4.11/modules/core/doc/intro.html>
- [22] D. Pulgarín-velásquez, "Development and Implementation of an Artificial Vision System Based on Free Use Languages for the Coach System Products on the Integrated," no. Cim, pp. 43–50, 2016
- [23] Itilib, "Introduction," 2010, 2010. [Online]. Available: <http://itilib.sourceforge.net/doc/homepage/index.shtml>.
- [24] Intel, "Background Subtraction," 2011. [Online]. Available: https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_video/py_bg_subtraction/py_bg_subtraction.html.
- [25] Intel, "Características de imagenes," 2015. [Online]. Available: https://docs.opencv.org/3.2.0/df/d54/tutorial_py_features_meaning.html.
- [26] M. Miriam, "Clasificación de objetos en imagenes usando SIFT," p. 14, 2014.
- [27] O. T. Gómez, P. P. R. López and J. S. Bacalla, "Criterios de selección de metodologías de desarrollo de software," Industrial Data, vol. 13, pp. 070-074, 2010.

- [28] Jiménez. J “Cuadro comparativo entre metodologías de desarrollo de Aplicaciones Web”. [Online]. Disponible en: <http://goo.gl/sHRjYF>
- [29] C. E. Gallardo y H. C. Ahumada, «Modelado de un Prototipo de Aplicación Web para la Gestión de Guardias Médicas del Gobierno de Catamarca», pp. 47-66, 2015.
- [30] C. Daniel, L. Gámez, S. Gustavo, P. Camarena, y U. J. Martínez, «Propuesta de artefactos basados en una notación con grafos y conjuntos para el modelado conceptual de aplicaciones Web», vol. 107, n.o 2015, pp. 41-50.

K. ANEXO

ANEXO 1: Entrevista aplicada al cliente



**UNIVERSIDAD
NACIONAL
DE LOJA**



Facultad de Energía las Industrias y los Recursos Naturales No Renovables

CARRERA DE INGENIERÍA EN SISTEMAS

Institución: Universidad Nacional de Loja

Departamento: Área de la Energía las Industrias y Recursos naturales No Renovables.

Entrevistado: Ing. Luis Roberto Jácome Galarza,

Entrevistador: Sr. Fabricio Sisalima Ortega

Fecha: 17/06/2017

Trabajo de titulación el cual versa sobre **“SISTEMA DE DETECCIÓN Y CONTEO VEHICULAR APLICANDO TÉCNICAS DE VISIÓN ARTIFICIAL”**.

Preguntas:

¿Tiene alguna preferencia por una tecnología específica para el desarrollo del entorno Web?

No ninguna tecnología en específico, debe ser una que sea conocida y soportada actualmente.

¿Usted cree necesario el desarrollo de un sistema para detección y conteo de vehículos que permita automatizar el proceso de conteo de vehículos?

Si ya que en base es estos datos se puede tomar muchas decisiones con las que se puede optimizar recursos de todo tipo.

¿Cree conveniente que el sistema debe realizar el conteo de vehículos en calles de una vía o doble vía?

Debe permitir el conteo en ambos sentidos y en un sentido ya que hay vías de ambos casos.

¿Considera usted que el almacenamiento de los datos capturados por el sistema debe ser almacenados localmente y externamente en un servidor de base de datos?

Se deben almacenar externamente en un servidor para que sea accesible en cualquier momento.

¿Con qué frecuencia cree usted que deberían ser enviados los datos capturados por el sistema de conteo al servidor de Base de datos?

Se debería enviar según como se genere el evento de conteo.

¿Usted como usuario qué espera que haga la Aplicación Web, qué espera visualizar, qué información le gustaría que esté disponible?

Visualizar donde se ubican los contadores como algo informativo, realizar consultas de los datos de conteo generados y exportar los datos consultados.

¿De qué datos se deberá componer las credenciales de acceso a la Aplicación Web?

Usuario y contraseña se acceso que puedan ser modificadas en cualquier momento por el administrador.

¿Cuántas personas deberían tener acceso a los datos registrados en la Aplicación Web?

No se conoce un número exacto ya que los datos pueden se necesitados por cualquier persona que requiera este tipo de datos.

¿Qué datos cree necesario que se registren en el alta de un nuevo Usuario?

Los nombres, correo, celular y los que sean necesarios.

¿Cree conveniente que se realice la gestión de contadores (crear, eliminar, editar, visualizar) desde el entorno Web?

Si ya que la información de los contadores poder modificarse ejemplo se puede cambiar el lugar de su ubicación.

¿Considera necesario implementar la geolocalización de contadores registrados en el entorno Web, para poder conocer el lugar donde ha sido instalado?

Si poder visualizar en un mapa en el sistema web la ubicación es necesario para conocer su ubicación rápidamente en el mapa.

¿Está de acuerdo en que los contadores sean configurables (nombre, posición, sentido de conteo etc.) desde el entorno Web?

Si ya que será de gran utilidad para conocer donde están ubicados y diferenciarlos fácilmente.

¿Cree conveniente la incorporación de reportes en formato PDF en la Aplicación Web?

Si ya que puede servir para generar algún tipo de informe digital o impreso.

¿Qué tipo de información cree que sea necesaria presentar en un reporte?

Información como por ejemplo si el vehículo contado es de ingreso o de salida, tipo de vehículo de ser posible.

¿Considera necesario la generación de reportes de conteo de vehículos en rangos específicos de fecha y hora?

Si es necesarios ya que hay casos en que se necesita analizar los datos en intervalos de tiempo específicos.

Luis Roberto Jácome

Ing. Luis Roberto Jácome Galarza

ANEXO 2: Documento de Especificación de Requerimientos (ERS)

ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE SRS 830

Proyecto: Sistema para detección y conteo vehicular aplicando técnicas de visión artificial.

Revisión 0.3

Fabricio R. Sisalima O.

El presente documento es aprobado por el cliente referenciado a continuación:

Firma Cliente

Roberto Jácome

Ing. Luis Roberto Jácome Galarza

DOCUMENTO DE ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE IEEE

830

1. Introducción

1.1. Propósito

El presente documento tiene como propósito presentar los requerimientos funcionales y no funcionales del sistema de conteo vehicular que permitirá la toma de decisiones en base a los datos obtenidos en el sistema.

1.2. Ámbito del Sistema

El software de conteo vehicular se identificará con el nombre de "SCV", será desarrollado para realizar el conteo de vehículos en perímetro urbano ciudad de Loja, el software de conteo se ejecutará en plataformas Linux, y el sistema de administración y reporte podrá accederse desde cualquier navegador web.

El software tiene como objetivo realizar el conteo de vehículos con visión por computador y enviar los datos a un servidor, y luego realizar reportes de los datos recolectados.

Es por esto que el software SCV realizará las siguientes funcionalidades:

- Conteo de vehículos en una vía de ingreso.
- Conteo de vehículos en una vía de salida.
- Conteo de vehículos en una vía de doble sentido.
- Reportes en rangos de fechas.
- Cálculos de volumen de tránsito.
- Georreferenciación de contadores en el sistema web.

1.3. Definiciones, Acrónimos y Abreviaturas

La siguiente Tabla describe las definiciones, acrónimos y abreviaturas claves que se utilizarán en el documento.

TABLA I. DEFINICIONES, ACRÓNIMOS Y ABREVIATURAS

Término	Significado
Contador vehicular	Software de conteo de vehículos basado en visión por computador.
Usuario	Persona que utilizara la aplicación móvil.
Software	Soporte lógico de un sistema informático, que comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas.
Sistemas operativo	Conjunto de programas que permite gestionar los recursos de un computador.
Georreferenciación	Posicionamiento que define la localización de un objeto en un sistema de coordenadas.
QT Framework	Framework para desarrollar aplicaciones en lenguaje C++.
Framework	Es un entorno de trabajo para desarrollo de aplicaciones que dependiendo del lenguaje integran componentes como bibliotecas, plantillas, entre otros, que facilitan la creación de aplicaciones.
RF	Requisito Funcional.
RNF	Requisito No Funcional.
SCV	Sistema de Conteo Vehicular.

1.4. Referencias

Estándar IEEE 830 – 1998

1.5. Visión General del Documento

Este documento tiene como objetivo dar a conocer una descripción completa del funcionamiento del sistema a desarrollar. Se encuentra dividido en tres secciones. En la primera sección se realiza una introducción al documento en donde se detallan el propósito del documento, el ámbito del sistema, referencias y la visión general del documento. En la segunda sección se presenta una descripción general del sistema teniendo como puntos importantes la perspectiva del Producto, Funciones del

Producto, Características de los Usuarios, Restricciones, Suposiciones y Dependencias. Finalmente, en la última sección se muestra una descripción de los requerimientos específicos del sistema, donde se revisarán los siguientes temas Interfaces Externas y Requisitos Funcionales.

2. Descripción General

2.1. Perspectiva del Producto

El software de conteo vehicular denominado SCV, funcionará siempre que se disponga de una conexión a una cámara ya sea tipo ip o usb, para el envío de los datos el servidor es necesario se disponga de una conexión estable a internet. El software de conteo vehicular se ejecutará en pc con sistema operativo Linux.

2.2. Funciones del Producto

El sistema desarrollado permitirá realizar las siguientes funcionalidades:

- Realizar conteo de vehículos mediante visión por computador.
- Generar reportes de los datos recolectados.

2.3. Características de los Usuarios

En el sistema SCV se identifican dos tipos de usuarios: Súper usuario y usuario Normal, a continuación, se presenta la interacción de cada uno de ellos con la aplicación web.

TABLA II. CARACTERÍSTICA DEL SÚPER USUARIO

Tipo de usuario	Súper Usuario
Formación	Ingeniero o técnico en sistemas
Habilidades	Debe tener conocimientos en el manejo de HTML, CSS, JavaScript, C++.
Actividades	Dar mantenimiento y mejorar el sistema.

TABLA III. CARACTERÍSTICA DEL USUARIO NORMAL

Tipo de usuario	Usuario Normal
Formación	Ninguna formación específica
Habilidades	Conocimiento en el manejo de aplicaciones web.
Actividades	Utilizar las funciones del sistema de conteo vehicular.

2.4. Restricciones

- Para el desarrollo se tomará como referencia la metodología XP que es la que más se adapta a este tipo de desarrollo.
- Se hará uso del JSF el cual nos permite el desarrollo de aplicaciones web en lenguaje de programa Java.
- La aplicación web será compatible con los navegadores Chrome y Firefox.
- Funcionará siempre que exista conexión a internet.

2.5. Suposiciones y Dependencias

2.5.1. Suposiciones

Se asume que los requisitos planteados en este documento son correctos y estables, lo cual generará un sistema que cumplirá con las necesidades solicitadas por la UMT y las características de un software de calidad.

2.5.2. Dependencias

El funcionamiento del sistema depende de una de internet estable tanto en el software de conteo como en el sistema web para obtener reportes.

3. Requisitos Específicos

3.1. Interfaces Externas

3.1.1. Interfaces de usuario

La interfaz del sistema web estará basada en el prototipo presentado, siguiendo los mismos criterios ahí señalados.

3.1.2. Interfaces de hardware

En cuanto al hardware es necesario disponer de un dispositivo con las siguientes

características como mínimo:

- Procesador 1GHz
- 1GB Memoria interna

3.1.3. Interfaces de software

El software necesario para la ejecución del aplicativo de conteo vehicular es:

- Sistema operativo Linux, Debian derivado.

3.2. Requisitos Funcionales

A continuación, se realizará una descripción de los requisitos funcionales obtenidos anteriormente.

3.2.1. Requisito Funcional 1

Número de requisito	RF01
Nombre de requisito	Detección de vehículos.
Tipo	Requisito
Prioridad del requisito	Alta
Descripción	El sistema debe de permitir detectar vehículos

3.2.2. Requisito Funcional 2

Número de requisito	RF02
Nombre de requisito	Conteo de vehículos según sentido de vía .
Tipo	Requisito
Prioridad del requisito	Alta
Descripción	El sistema debe de permitir contar vehículos en vías de solo ingreso, salida o ambos sentidos.

3.2.3. Requisito Funcional 3

Número de requisito	RF03
Nombre de requisito	Envío de datos a servidor.
Tipo	Requisito
Prioridad del requisito	Alta
Descripción	El sistema debe de permitir enviar los datos de conteo a un servidor.

3.2.4. Requisito Funcional 4

Número de requisito	RF04
Nombre de requisito	Almacenamiento de datos offline.
Tipo	Requisito
Prioridad del requisito	Media
Descripción	El sistema de permitir almacenar localmente los datos de conteo en caso de fallas en la red

3.2.5. Requisito Funcional 5

Número de requisito	RF05
Nombre de requisito	Inicio de software de conteo con el sistema.
Tipo	Requisito
Prioridad del requisito	Media
Descripción	En caso de reinicio del pc el sistema deberá iniciarse automáticamente para seguir con sus funciones para las que fue programado.

3.2.6. Requisito Funcional 6

Número de requisito	RF06
Nombre de requisito	Configuración de fuente de video.
Tipo	Requisito
Prioridad del requisito	Alta
Descripción	El sistema deberá permitir la configuración de la fuente de video.

3.2.7. Requisito Funcional 7

Número de requisito	RF07
Nombre de requisito	Configuración de sentido de conteo.
Tipo	Requisito
Prioridad del requisito	Alta
Descripción	El software de conteo debe permitir la configuración para conteo de vehículos según el sentido de vía.

3.2.8. Requisito Funcional 8

Número de requisito	RF08
Nombre de requisito	Gestión de usuarios.
Tipo	Requisito
Prioridad del requisito	Alta
Descripción	La Aplicación deberá permitir realizar la gestión de usuarios (alta, baja, edición y visualización de usuarios)

3.2.9. Requisito Funcional 9

Número de requisito	RF09
Nombre de requisito	Registro de usuarios.
Tipo	Requisito
Prioridad del requisito	Alta
Descripción	Los campos para el registro de un usuario en la Aplicación Web deberán ser: nombre, apellido, celular, cédula, correo y contraseña.

3.2.10. Requisito Funcional 10

Número de requisito	RF010
Nombre de requisito	Gestión de contadores.
Tipo	Requisito
Prioridad del requisito	Alta
Descripción	La Aplicación deberá permitir realizar la gestión de contadores (alta, baja, edición y visualización de contadores)

3.2.11. Requisito Funcional 11

Número de requisito	RF011
Nombre de requisito	Reporte de conteo.
Tipo	Requisito
Prioridad del requisito	Alta
Descripción	La Aplicación deberá permitir realizar reportes de: contadores registrados y datos almacenados por cada contador.

3.2.12. Requisito Funcional 12

Número de requisito	RF012
Nombre de requisito	Geolocalización de contadores.
Tipo	Requisito
Prioridad del requisito	Alta
Descripción	La Aplicación deberá permitir geolocalizar los contadores registrados e instalados.

3.3. Requisitos No funcionales

3.3.1. Disponibilidad

La disponibilidad del sistema será continua las 24 horas al día y los 7 días de la semana, aunque si existe algún fallo en cualquiera de los componentes del sistema se tomaran las medidas necesarias para solucionarlo.

3.3.2. Usabilidad

La aplicación web contará con una interfaz amigable e intuitiva para facilitar los reportes de datos de conteo, así mismo el software de conteo en el pc contará con una interfaz para modificar todas las configuraciones del contador fácilmente.

3.3.3. Portabilidad

El software de conteo funcionará en plataformas con sistema operativo Linux con Debian derivado.

3.3.4. Accesibilidad

El sistema web permitirá el acceso a los usuarios únicamente registrados en el sistema.

3.3.5. Mantenibilidad

Todo el sistema debe contar con documentación entendible y fiable para que facilite la actualización de las funciones y la corrección de errores.

3.3.6. Extendibilidad

El software de conteo vehicular permite incrementar nuevas funcionalidades o características sin ningún esfuerzo.

3.3.7. Rendimiento

La aplicación permitirá el acceso a uno o más usuarios al mismo tiempo sin ningún tipo de inconveniente.

3.4. Apéndices

Apéndice 1: Elicitación

1. Recolección de información

Es la base para la obtención, descubrimiento y adquisición de los requerimientos de un sistema mediante la comunicación directa con el cliente, para esto existe diferentes técnicas que se pueden aplicar entre estas tenemos entrevista, casos de uso, lluvia de ideas, encuestas, etc.

Las técnicas que se aplicaron para la captura de requerimientos son las siguientes:

- **Entrevista:** Esta técnica se la utilizó para obtener información de manera rápida, fácil de analizar y obtener resultados cuantificables. Fue realizada al Ingeniero Luis Roberto Jácome Galarza, la cual consto de una serie de preguntas concretas que facilitaron la extracción de los requisitos y a su vez conocer el alcance o límites del software.
- **Lluvia de Ideas o tormenta de ideas:** Esta técnica se la utilizó para tener más claro el ámbito del problema, generar nuevas ideas y complementar las ideas establecidas por el cliente.

2. Lista preliminar de requerimientos

De las técnicas de Entrevista y lluvia de ideas, se obtuvo una lista preliminar de requerimientos, los cuales se presenta en la siguiente tabla:

TABLA IV. LISTA PRELIMINAR DE REQUERIMIENTOS

Numero	Requisitos	Solicitado/ Inferido	Referencia	
			Técnica	Pregunta
R01	El sistema debe de permitir la detección de vehículos.	Solicitado	Entrevista	E.P2
R02	El sistema debe de permitir contar vehículos según el sentido en que transiten.	Solicitado	Entrevista	E.P2, E.P6
R03	El sistema debe de permitir enviar los datos de conteo a un servidor.	Solicitado	Entrevista	E.P7
R04	El sistema de permitir almacenar localmente los datos de conteo en caso de fallas en la red.	Inferido	Lluvia de ideas	
R05	El software de conteo deberá iniciar conjuntamente con el sistema.	Inferido	Lluvia de ideas	
R06	El sistema deberá permitir la configuración de la fuente de video.	Inferido	Lluvia de ideas	
R07	El sistema debe permitir configurar el sentido de conteo (entrada o de salida) de vehículos.	Solicitado	Entrevista	E.P5
R08	La Aplicación web deberá permitir realizar la gestión de usuarios.	Inferido	Lluvia de ideas	
R09	Los campos para el registro de un usuario en la Aplicación Web deberán ser: nombre, apellido, celular, cédula, correo y contraseña.	Inferido	Lluvia de ideas	

R10	La aplicación web debe permitir la gestión de contadores.	Inferido	Lluvia de ideas	
R11	La aplicación web debe permitir generar reportes de los datos de conteo.	Solicitado	Entrevista	E.P3
R12	La aplicación web debe permitir la georreferenciación de los contadores registrados	Inferido	Lluvia de ideas	

Apéndice 2: Análisis de los Requerimientos

El análisis de los requerimientos permite establecer las necesidades que debe cumplir un determinado sistema, después de haber obtenido la lista preliminar de requerimientos es necesario descartar la existencia de requerimientos contradictorios, repetidos y poco entendibles, para esto se utilizara una matriz de iteración.

Matriz de Iteración

Esta consiste en una matriz de doble entrada, en donde cada entrada contiene todos los requisitos logrando que se crucen entre sí, donde se debe comprobar lo siguiente:

- Si r_1 se solapa con r_2 , significa que r_1 trata aspectos del sistema también tratados en r_2 . Esto daría lugar a problemas de redundancia. Esto se lo representara con la letra S.
- Si r_1 está en conflicto con r_2 , significa, r_1 y r_2 son contradictorios. Esto da lugar a problemas de consistencia interna. Esto se lo representara con la letra C.

S: Solapamiento

C: Conflicto

La siguiente matriz representa los solapamientos y conflictos de los requisitos anteriormente obtenidos.

TABLA V. MATRIZ DE INTERACCIÓN

	R01	R02	R03	R04	R05	R06	R07	R08	R09	R10	R11	R12
R01												
R02												
R03												
R04												
R05												
R06												
R07												
R08												
R09								S				
R10												
R11												
R12												

Después de realizar la matriz de iteración es necesario describir cada uno de los solapamientos y conflictos encontrados, para brindarles una solución mediante la construcción de un nuevo requisito. En la TABLA VI se describe los solapamientos y conflictos.

TABLA VI. SOLAPAMIENTOS Y CONFLICTOS

Nº	Solapamiento/ Conflicto	Requisito	Descripción	Requisito Final
1	S	(8,9)	Requisito 8 Requisito 9	La Aplicación web deberá permitir realizar la gestión de usuarios.

Una vez solucionados todos los inconvenientes encontrados en la lista preliminar de requerimientos se puede determinar la lista final de estos.

Lista final de requerimientos

En la siguiente TABLA VII se establecen los requerimientos finales de la aplicación móvil, los cuales serán utilizados durante el desarrollo de esta.

TABLA VII. LISTA DE REQUERIMIENTOS FINAL

Código	Requerimiento
RF01	Detección de vehículos.
RF02	Conteo de vehículos según el sentido en que transiten.
RF03	Envío datos de conteo a un servidor.
RF04	Almacenamiento de datos local.
RF05	Inicio de software con el arranque del sistema.
RF06	Configuración de la fuente de video.
RF07	Interfaz de configuración de software de conteo.
RF08	Gestión de usuarios en aplicación web.
RF09	Gestión de contadores en aplicación web.
RF10	Reportes de conteo en aplicación web
RF11	Georreferenciación de contadores en aplicación web.

ANEXO 3. Prototipos de la Aplicación Web

En esta sección se muestra la continuación de los prototipos de la aplicación web, correspondiente a los módulos restantes.

1. Menú Empresas

En esta sección se puede realizar gestión de empresas que requieren el servicio de conteo vehicular y la asociación de contadores a las empresas registradas.



Figura 34. Opción en menú para módulo empresas

1.1. Módulo Empresas

Se muestran las empresas registradas en el sistema, a estas empresas será a quienes se les asigne un contador y usuarios.

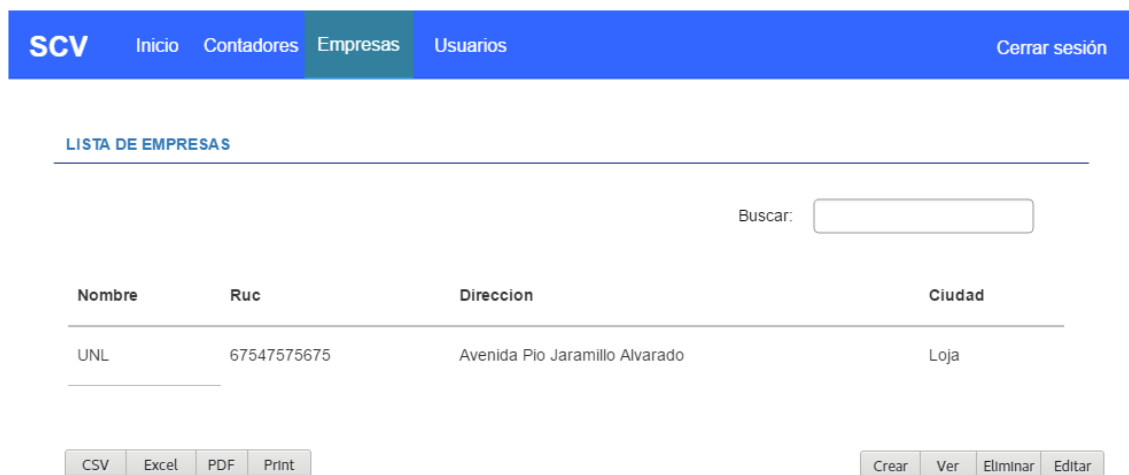


Figura 35. Pantalla de cálculo de volumen de tránsito

1.1.1. Formulario registro empresa

Al presionar en el botón “Crear” se muestra el formulario para el ingreso de los datos correspondientes a la empresa a registrar.

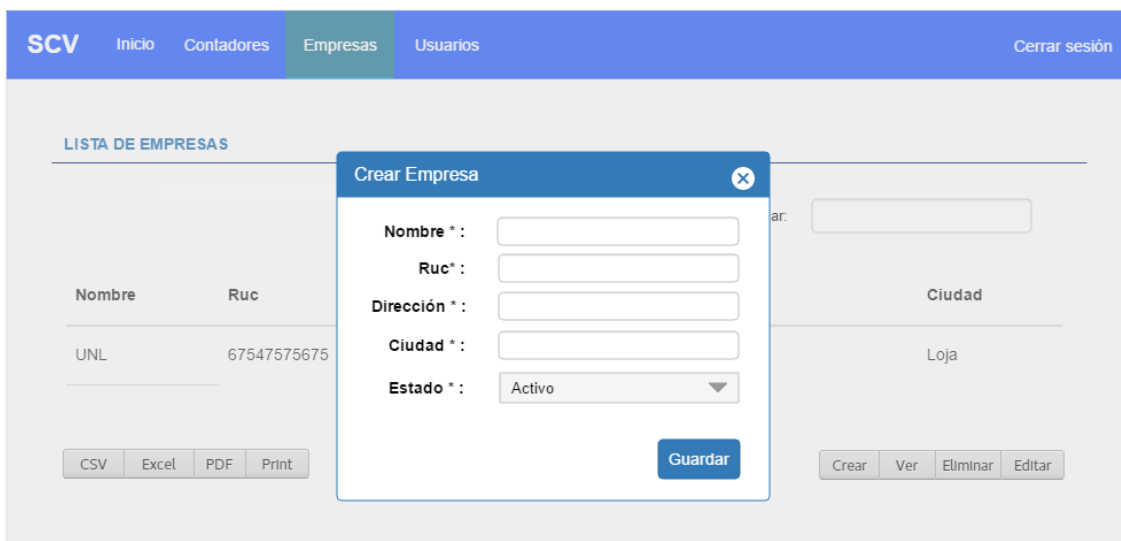


Figura 36. Formulario de registro de empresa

1.1.2. Detalle de empresa

Al presionar en el botón “Ver” se presenta una ventana con la información de la empresa seleccionada.

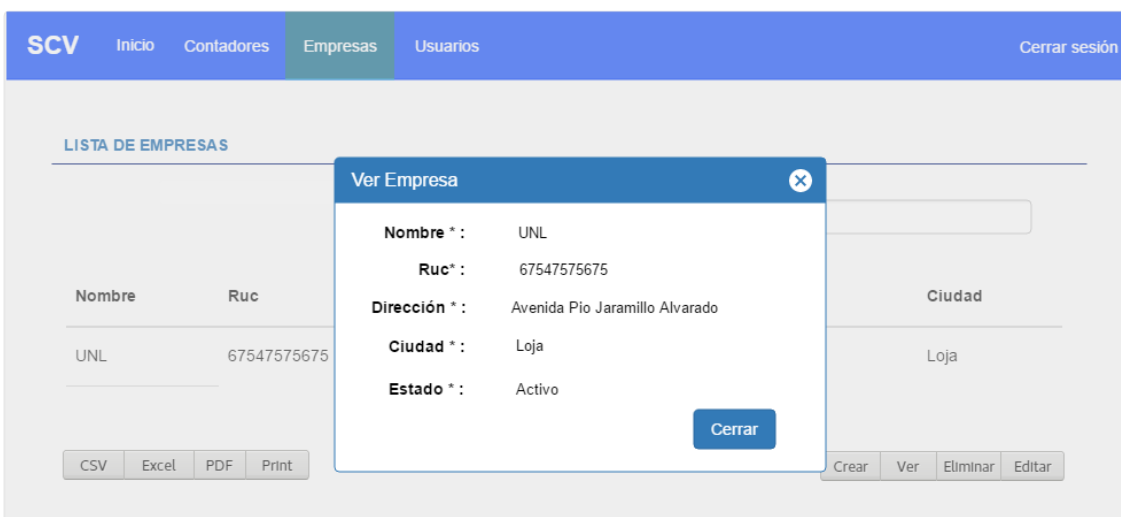


Figura 37. Pantalla con detalle de empresa

1.1.3. Formulario de edición de empresa

Al presionar sobre el botón “Editar” se muestra el formulario de edición con los datos antes registrados.

Nombre	Ruc
UNL	67547575675

CSV Excel PDF Print

Crear Ver Eliminar Editar

Figura 38. Formulario de edición de empresa

1.1.4. Menú Contadores de empresas

En este módulo se puede asignar los contadores registrados a una empresa registrada.

Empresas

Contadores de empresas

Conteo Vehicular

Sistema de administración de dispositivos contadores de vehiculos, utilizamos visión por computador para realizar el proceso de conteo, los datos son enviados a la nube inmediatamente para se analizados.

- Geolocalización**
Disponemos de las ubicaciones de cada contador registrado en el sistema, así conoceras en cualquier momento su ubicación en el mapa.
- Volúmen de tránsito**
Opciones para generar volúmen de tránsito, anual, mensual semanal. Todos estos cálculos el sistema los realiza automáticamente.
- Offline**
Los equipos contadores pueden funcionar de forma offline para luego cargar los datos en el sistema y así realizar el procesamiento de los datos.
- Reportes**
Disponemos de opciones para exportar sus datos a formatos pdf, csv, excel o para imprimir directamente, con esto podrá utilizar sus datos para analisis manual o con software de terceros.

Figura 39. Menú de opción de asignación de contadores a empresas.

1.1.5. Tabla de contadores de empresas

Inicialmente se presenta una tabla con la información de los contadores asignados a las empresas registradas.

The screenshot shows the 'Empresas' menu in the SCV system. The header includes 'Inicio', 'Contadores', 'Empresas', and 'Usuarios', with 'Cerrar sesión' on the right. Below the header, the title 'LISTA DE CONTADORES ASIGNADOS A EMPRESAS' is displayed. A search bar labeled 'Buscar:' is present. The main content is a table with three columns: 'Estado', 'Empresa', and 'Contador'. The table contains one row with the values 'Activo', 'UNL', and 'Contador 1'. Below the table, there are two sets of buttons: 'CSV', 'Excel', 'PDF', 'Print' on the left, and 'Crear', 'Ver', 'Eliminar', 'Editar' on the right.

Estado	Empresa	Contador
Activo	UNL	Contador 1

Figura 40. Lista de contadores asignados a empresas

1.1.6. Formulario de asignación de contadores a empresa

Cuando presiona sobre el botón "Crear" se muestra una vista para la selección de la empresa y contador que se desea asociar.

The screenshot shows the 'Empresas' menu with a modal form titled 'Editar Asignación de Contador' overlaid. The modal contains three dropdown menus: 'Estado *' with 'Activo' selected, 'Empresa*' with 'UNL' selected, and 'Contador *' with 'Contador 1' selected. A 'Guardar' button is at the bottom of the modal. The background shows the same table as in Figure 40, but it is dimmed.

Figura 41. Formulario de asignación de contador a empresa

1.1.7. Detalle de contador asignado

Al presionar en el botón “Ver” se presenta una ventana con la información del registro y además de su estado.

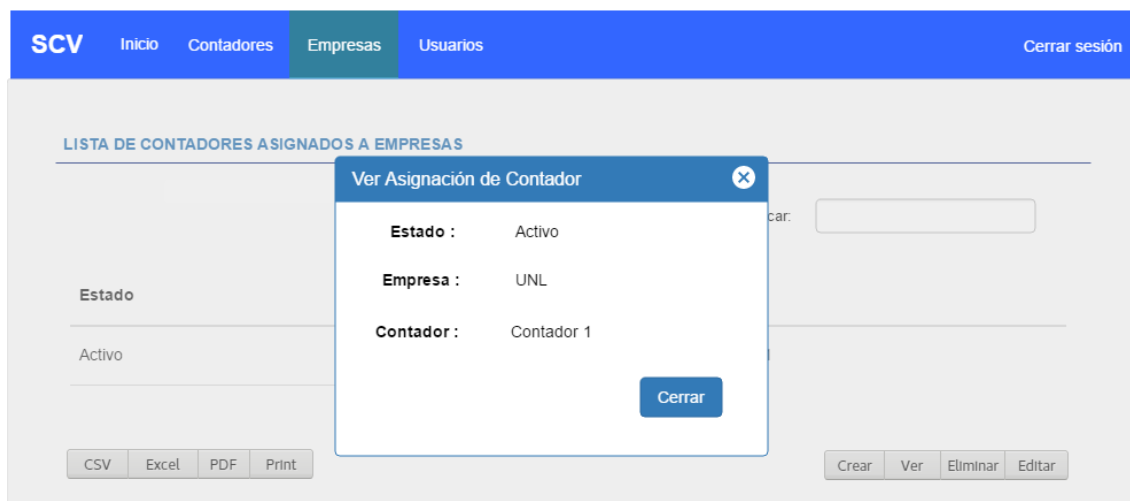


Figura 42. Vista con detalle de contador asignado a empresa.

1.1.8. Formulario para editar asignación de contador

Cuando se presiona el botón “Editar” se presenta la ventana de edición con los datos cargados para su edición.

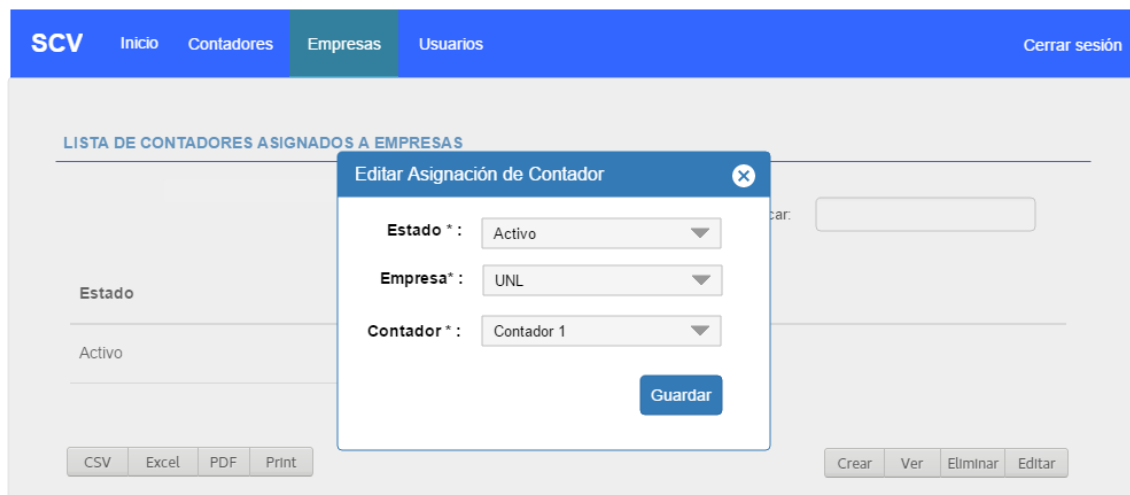


Figura 43. Formulario de asignación de contador

1.2. Menú Usuarios

Al seleccionar el ítem “Usuarios” se muestra una vista en la cual se puede crear, editar, eliminar y ver registros de usuarios en el sistema.



Figura 44. Vista de menú de usuarios

1.2.1. Lista de usuarios registrados

La primera vista que se presenta es la tabla con todos los registros de usuarios realizados hasta el momento, además las opciones para exportar estos registros.

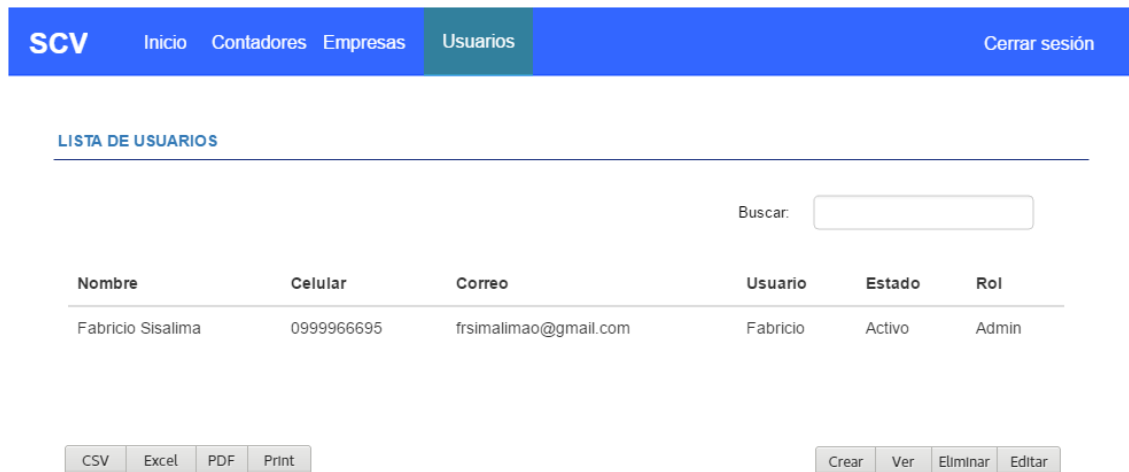


Figura 45. Lista de usuarios registrados

1.2.2. Formulario de registro de usuario

Al presionar el botón “Crear” se presenta el formulario para el ingreso de los datos del nuevo usuario.

The screenshot shows the 'Crear Usuario' (Create User) modal form. The form contains the following fields and options:

- Nombre *:
- Celular *:
- Correo *:
- Usuario *:
- Estado *:
- Rol *:

At the bottom right of the modal is a 'Guardar' (Save) button. The background shows a table with columns for 'Nombre', 'Celular', 'Estado', and 'Rol', and a 'Crear' button.

Figura 46. Formulario de registro de usuarios

1.2.3. Formulario de edición de usuario

Si presiona el botón “Editar” se muestra un formulario con los datos del registro seleccionado para que se pueda realizar la edición de los mismos.

The screenshot shows the 'Editar Usuario' (Edit User) modal form. The form contains the following fields and options, with pre-filled data:

- Nombre *:
- Celular *:
- Correo *:
- Usuario *:
- Estado *:
- Rol *:

At the bottom right of the modal is a 'Guardar' (Save) button. The background shows a table with columns for 'Nombre', 'Celular', 'Estado', and 'Rol', and an 'Editar' button.

Figura 47. Formulario de edición de usuarios

1.2.4. Detalle de usuario registrado

En caso de presionar en el botón “Ver” se muestra la vista con el resumen del registro seleccionado.

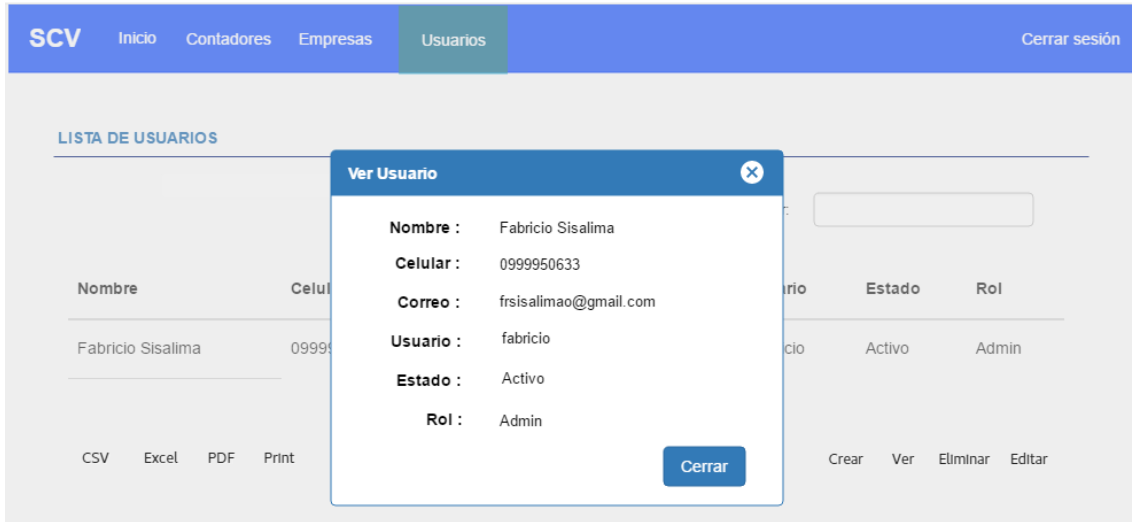


Figura 48. Vista con detalle de usuario registrado

1.2.5. Menú Usuarios empresas

En esta sección se presenta el módulo de usuarios empresa que es la asociación de un usuario a una empresa en particular para que este pueda acceder a los datos de sus contadores.



Figura 49. Vista con menú de usuarios empresa

1.2.6. Lista de usuarios asignados a empresas

Muestra la lista de empresas y usuario que están asignados, además se muestra las opciones para la generación de reportes.

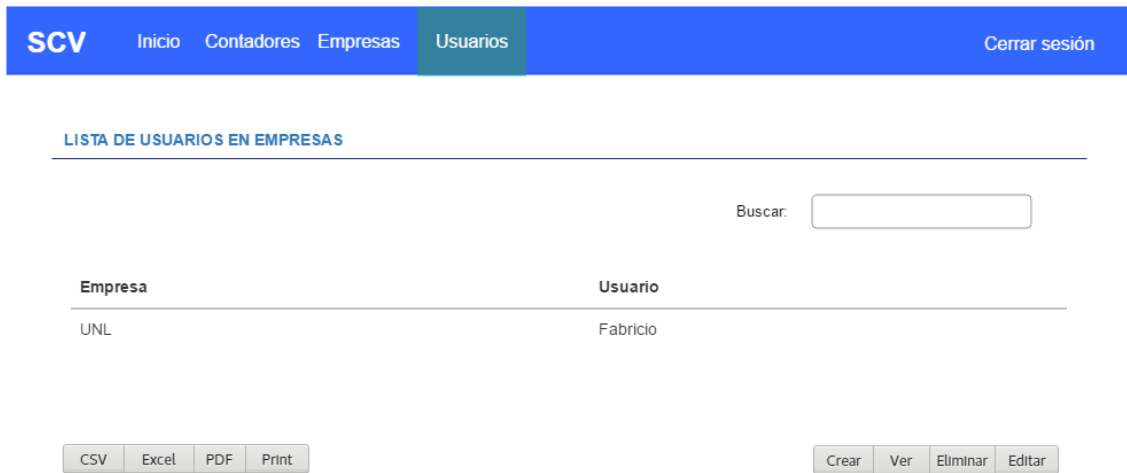


Figura 50. Lista de usuarios asignados a empresas

1.2.7. Formulario de asignación de usuario a empresa

Permite asignar a una empresa un usuario para que pueda acceder a los contadores pertenecientes a la misma.

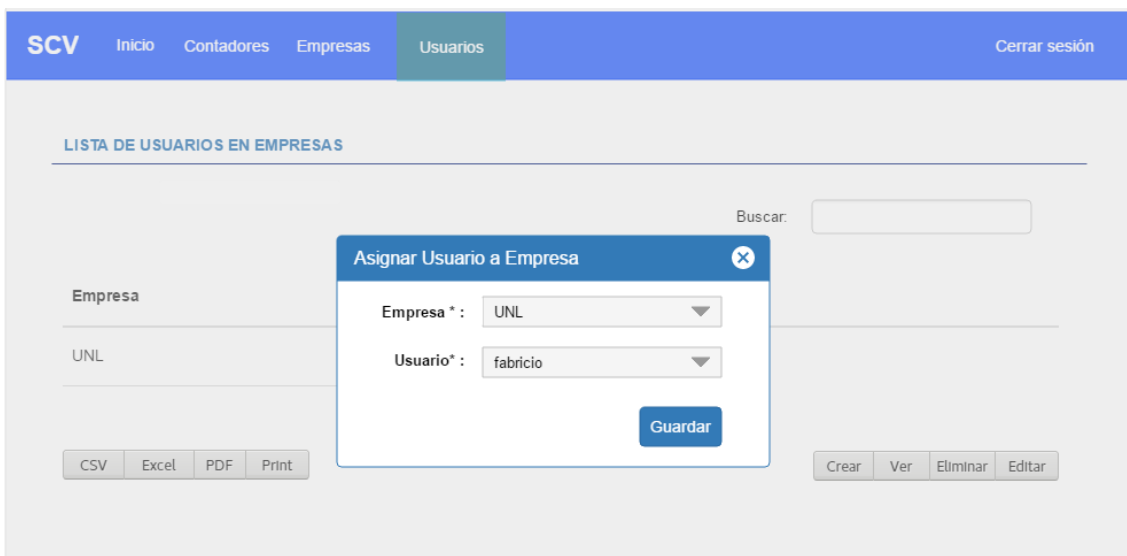


Figura 51. Formulario de asignación de usuarios a empresas

1.2.8. Formulario de edición de usuario asignado a empresa

Cuando se presiona en el botón “Editar”, se presenta el formulario de asignación de empresa y usuario.

The screenshot shows the SCV web application interface. The top navigation bar includes 'Inicio', 'Contadores', 'Empresas', 'Usuarios', and 'Cerrar sesión'. The main content area is titled 'LISTA DE USUARIOS EN EMPRESAS'. A search bar labeled 'Buscar:' is present. A modal window titled 'Editar Asignación' is open, displaying two dropdown menus: 'Empresa *' with 'UNL' selected and 'Usuario*' with 'fabricio' selected. A 'Guardar' button is at the bottom of the modal. In the background, there are buttons for 'CSV', 'Excel', 'PDF', 'Print', 'Crear', 'Ver', 'Eliminar', and 'Editar'.

Figura 52. Formulario de edición de usuarios a empresas

1.2.9. Detalle de usuario asignado a empresa

Al presionar en el botón “Ver” se muestra la relación entre empresa y usuario que se ha registrado.

The screenshot shows the SCV web application interface. The top navigation bar includes 'Inicio', 'Contadores', 'Empresas', 'Usuarios', and 'Cerrar sesión'. The main content area is titled 'LISTA DE USUARIOS EN EMPRESAS'. A search bar labeled 'Buscar:' is present. A modal window titled 'Ver Asignación' is open, displaying the assigned values: 'Empresa : UNL' and 'Usuario : Fabricio'. A 'Cerrar' button is at the bottom of the modal. In the background, there are buttons for 'CSV', 'Excel', 'PDF', 'Print', 'Crear', 'Ver', 'Eliminar', and 'Editar'.

Figura 53. Vista con detalle de contador asignado a empresa.

ANEXO 4. Pruebas de Aceptación



UNIVERSIDAD NACIONAL DE LOJA
ÁREA DE LA ENERGÍA, LAS INDUSTRIAS Y RECURSOS NATURALES NO
RENOVABLES

PRUEBA DE ACEPTACIÓN

La presente prueba de aceptación se la realizó con la finalidad de comprobar las funcionalidades del sistema denominado “**SCV**”. Los resultados generados por esta prueba serán de vital importancia, así como de justificación para la culminación del proyecto de titulación.

CASOS DE PRUEBA

Marque su respuesta en la casilla correspondiente, teniendo en cuenta los siguientes criterios.

- **Exitoso:** El resultado de la prueba ha sido el esperado.
- **Detenido:** Presenta dificultades para la correcta ejecución de la prueba.
- **Fallido:** El resultado no ha sido el esperado.
- **No ejecutado:** La prueba no se ha llevado acabo.

Caso de prueba	Valor			
	Exitoso	Detenido	Fallido	No ejecutado
Obtener ubicación del usuario				
Observaciones:				

Caso de prueba	Parámetros			
	Exitoso	Detenido	Fallido	No ejecutado
Registrar Usuario				
Observaciones:				

Caso de prueba	Parámetros			
	Exitoso	Detenido	Fallido	No ejecutado
Editar Usuario				
Observaciones:				

Caso de prueba	Parámetros			
	Exitoso	Detenido	Fallido	No ejecutado
Eliminar Usuario				
Observaciones:				

Caso de prueba	Parámetros			
	Exitoso	Detenido	Fallido	No ejecutado
Visualizar Usuario				
Observaciones:				

Caso de prueba	Parámetros			
	Exitoso	Detenido	Fallido	No ejecutado
Registrar Contador				
Observaciones:				

Caso de prueba	Parámetros			
	Exitoso	Detenido	Fallido	No ejecutado
Editar Contador				
Observaciones:				

Caso de prueba	Parámetros			
	Exitoso	Detenido	Fallido	No ejecutado
Eliminar Contador				
Observaciones:				

Caso de prueba	Parámetros			
	Exitoso	Detenido	Fallido	No ejecutado
Visualizar Contador				
Observaciones:				

Caso de prueba	Parámetros			
	Exitoso	Detenido	Fallido	No ejecutado
Generar reporte de conteo				
Observaciones:				

Caso de prueba	Parámetros			
	Exitoso	Detenido	Fallido	No ejecutado
Exportar reporte de conteo				
Observaciones:				

Caso de prueba	Parámetros			
	Exitoso	Detenido	Fallido	No ejecutado
Visualización de imagen				
Observaciones:				

Caso de prueba	Parámetros			
	Exitoso	Detenido	Fallido	No ejecutado
Configurar fuente video				
Observaciones:				

Caso de prueba	Parámetros			
	Exitoso	Detenido	Fallido	No ejecutado
Detectar vehículos				
Observaciones:				

Caso de prueba	Parámetros			
	Exitoso	Detenido	Fallido	No ejecutado
Rastreo de vehículos en la imagen				
Observaciones:				

Caso de prueba	Parámetros			
	Exitoso	Detenido	Fallido	No ejecutado
Conteo de vehículos				
Observaciones:				

Luis Roberto Jácome

Ing. Luis Roberto Jácome Galarza

ANEXO 5: Certificado de las Pruebas de Aceptación

Loja, 16 de mayo de 2018

Ing. Luis Roberto Jácome Galarza

Certifica:

Que las Aplicaciones Web y Contador Vehicular presentadas por el Sr. Fabricio Roberto Sisalima Ortega con cedula: 1105012866, estudiante de la Carrera de Ingeniera en Sistemas de la Universidad Nacional de Loja, ha sido evaluada en los aspectos relacionados a la funcionalidad, accesibilidad, navegabilidad y usabilidad, por lo que se da fe de la realización de las pruebas y aceptación satisfactoria de las funcionalidades de las aplicaciones por parte de mi persona.

Es todo cuanto puedo certificar en honor a la verdad.

Muy atentamente.



Ing. Luis Roberto Jácome Galarza

ANEXO 6: Anteproyecto



**UNIVERSIDAD
NACIONAL
DE LOJA**



Facultad de Energía, las Industrias y los Recursos Naturales No Renovables

CARRERA DE INGENIERÍA EN SISTEMAS

“Sistema para detección y conteo vehicular aplicando técnicas de visión artificial”

PERFIL DE PROYECTO DE TRABAJO DE TITULACIÓN

Autor:

[\[ECINF5206\]](#) Sisalima Ortega Fabricio Roberto

Revisor:

[\[ECINF4836\]](#) Ing. Jácome Galarza Luis Roberto, Mg.Sc

16 de Mayo de 2017

LOJA – ECUADOR

Tabla de contenido

A. TEMA	1641
B. PROBLEMÁTICA	1
1. Situación Problemática	1
2. Problema de Investigación	3
C. JUSTIFICACIÓN	4
D. OBJETIVOS	5
1. OBJETIVO GENERAL	5
2. OBJETIVOS ESPECÍFICOS	5
E. ALCANCE	6
F. MARCO TEÓRICO	8
G. METODOLOGÍA	10
1. Metodología para el desarrollo de software	10
2. Métodos	10
3. Técnicas	11
H. CRONOGRAMA	12
I. PRESUPUESTO Y FINANCIAMIENTO	15
1. Presupuesto	15
2. FINANCIAMIENTO	17
J. BIBLIOGRAFÍA	18
K. ANEXOS	
19	

Índice de Figuras

Figura 1 FASE 1 del Cronograma de Actividades	13
Figura 2 FASE 2 del Cronograma de Actividades	13
Figura 3 FASE 3 del Cronograma de Actividades	14
Figura 4 FASE 4 del Cronograma de Actividades	14

Índice de Tablas

Tabla 1 Descripción del Cronograma	12
Tabla 2 Talento Humano para el proyecto	15
Tabla 3 Recursos Hardware y Software	16
Tabla 4 Prestación de Servicios a adquirir	16
Tabla 5 Presupuesto total del Proyecto	17

A. TEMA

“Sistema para detección y conteo vehicular aplicando técnicas de visión artificial”

B. PROBLEMÁTICA

1. Situación Problemática

La congestión vehicular en los núcleos urbanos a nivel mundial, es una grave crisis que de a poco se ha constituido en una problemática para el desarrollo de las ciudades. El uso del suelo, la expansión urbana y la migración han causado serios impactos en la gestión de movilidad, evidenciándose un déficit en el servicio de transporte público, así como un aumento en el parque automotor, lo que ha derivado en una problemática al momento de circular por los anillos céntricos de las ciudades. [1] El tráfico vehicular es la consecuencia de múltiples factores sociales, culturales, económicos y políticos que se presentan en las principales ciudades del mundo. [2] Las ciudades de gran tamaño se asocian con congestiones de tráfico vehicular, ya sea porque la infraestructura no creció o porque no se desarrolló de manera adecuada con respecto al aumento en el número de vehículos. El problema de tráfico tiene diferentes componentes que pueden ser analizados: los vehículos, los peatones y la interacción entre ellos, la forma de cómo se recolecta la información es muy dispendiosa ya que normalmente se requiere un personal que haga esta labor manualmente durante tiempos prolongados, por lo cual se podrían cometer errores involuntarios que afectarían la precisión de la medición. [3]

En los últimos años el aumento de la demanda de transporte y del tránsito vial han traído como consecuencia, particularmente en las ciudades grandes, incrementos en la congestión, demoras, accidentes y problemas ambientales, bastante mayores que los considerados aceptables por los ciudadanos. El aumento explosivo de esos fenómenos se debe a la facilidad creciente de acceso a automóviles [4], fruto del aumento del poder adquisitivo de las clases socioeconómicas de medianos ingresos, el mayor acceso al crédito, la reducción relativa de los precios de venta, una mayor oferta por recirculación de autos usados, el crecimiento de la población, la reducción del número de habitantes por hogar y la escasa aplicación de políticas estructuradas en el área del transporte urbano [1]. Por otra parte, la capacidad viaria no es suficiente para sostener la cantidad de vehículos que circulan en las denominadas horas pico, por lo que es necesario buscar las causas y las

posibles soluciones al congestionamiento vehicular utilizando la infraestructura vial existente.

En la actualidad las soluciones al tema de tráfico vehicular terrestre sigue estancadas ya que si bien es cierto se han socializado trabajos relacionados a la rama de Visión Artificial tales como: identificación de placas de vehículos, semáforos inteligentes, conteo automático vehicular entre otros no han sido la solución definitiva el problema de congestión vehicular.[4]

En la ciudad de Loja, a partir del último censo poblacional 2001, se ha observado un acelerado crecimiento demográfico, el cual se manifiesta con la creciente demanda de vehículos particulares, unidades de transporte público y taxis. [5] Esta situación ha puesto en alerta a la Unidad Municipal de Tránsito que es la institución encargada en la ciudad de Loja del control, señalización y organización de tránsito para que tome cartas en el asunto y de soluciones a los problemas que enfrenta por el acelerado crecimiento del parque automotor en la ciudad.

Enfocados en la situación problemática actual que se atraviesa en el tema de congestionamiento vehicular en las principales ciudades se ha podido identificar los siguientes problemas:

- Debido al crecimiento poblacional y la cantidad de automóviles por habitante descontrolado, la ciudad de Loja se encuentran en una situación crítica en cuestión de congestionamiento vehicular principalmente en las horas pico.
- Los datos en cuanto a zonas de mayor congestión y tránsito vehicular con los que cuenta la Unidad Municipal de Tránsito son desactualizados ya que los estudios relacionados al tráfico vehicular no se realizan de forma constante por sus altos costos y hasta el momento el método de conteo de vehículos en la ciudad se realizan de forma manual por lo que sus resultados son susceptibles al error humano.

- El congestionamiento vehicular conduce a que se origine otras serias consecuencias que afectan severamente las condiciones de la vida urbana tales como: la incrementada contaminación del aire provocada por el consumo de combustibles en vehículos que circulan en un tránsito convulsionado a baja velocidad, los mayores niveles de ruido en el entorno de las vías principales, la irritabilidad causada por la pérdida de tiempo y el aumento del estrés por conducir inmerso en una masa vehicular excesiva.

2. Problema de Investigación

Concluida la identificación y análisis de los problemas existentes relacionados al congestionamiento vehicular, se plantea el siguiente problema central el cual resume toda la problemática anteriormente planteada

Las carreteras y calles se están convirtiendo en centros de aglomeración vehicular, especialmente en la ciudad de Loja, por el aumento del parque automotriz y la carencia de soluciones viables al problema de tráfico vehicular.

Para constatar el problema central, se recurrió a investigar en fuentes confiables todos los estudios realizados acerca del tema de congestionamiento vehicular. En dichos estudios se afirma y se contractan los diferentes problemas citados con anterioridad y plasman futuras soluciones que se podrían dar para solventar los problemas encontrados.

En base a la investigación realizada e identificación de los problemas citados previamente se valida la existencia del problema central identificado y se plantea la siguiente interrogante:

¿El desarrollo de un sistema de detección y conteo vehicular ayudará en la toma de decisiones en el ámbito de tránsito urbano?

C. JUSTIFICACIÓN

El presente trabajo se lo realiza como respuesta al compromiso que tenemos los estudiantes de la Universidad Nacional de Loja con la colectividad, brindando soluciones eficientes y adecuadas mediante la implementación de los conocimientos adquiridos en las aulas, capaces de cubrir con las necesidades que surgen de problemas en la colectividad, así como justifica el trabajo final de carrera que como estudiantes debemos realizar para optar por el título profesional como ingenieros en Sistemas.

El conteo de vehículos es una tarea fundamental en los aforos de tráfico, tanto para el diseño de nuevas vías como para el control y optimización del tráfico urbano. Por lo cual la implementación del sistema de detección y contador de vehículos basado en técnicas de visión artificial, permitirá reducir la aglomeración vehicular existe, mejorar la movilidad y por ende decrementar los problemas asociados al tráfico vehicular en las zonas en las que se haga el conteo mediante el sistema propuesto.

Además, el desarrollo de este proyecto demanda de gastos económicos los cuales serán cubiertos en su totalidad por quien realiza el proyecto, así como será necesario el uso de equipos informáticos y herramientas tecnológicas, que podrán ser adquiridas sin dificultad, por lo cual se asegura que no existirán inconvenientes económicos ni tecnológicos en el transcurso de la realización del mismo. En cuanto a la parte ambiental estamos en la obligación como futuros profesionales a dar soluciones que permitan automatizar procesos y disminuir el uso de materiales y recursos no amigables con el medio ambiente, así con la implementación de este proyecto, se logrará reducir en gran porcentaje la contaminación auditiva y atmosférica que hasta la actualidad existe.

D. OBJETIVOS

1. OBJETIVO GENERAL

Desarrollar un sistema que permita la detección y conteo de vehículos aplicando técnicas de visión artificial.

2. OBJETIVOS ESPECÍFICOS

- Analizar los diferentes algoritmos de detección y reconocimiento de objetos.
- Diseñar e implementar el sistema para la detección y conteo de vehículos
- Realizar y ejecutar un plan de pruebas que permitan evaluar al sistema en cada uno de sus módulos.
- Divulgar la información obtenida en los procesos de desarrollo e implementación del proyecto a la comunidad científica

E. ALCANCE

La presente propuesta de proyecto de trabajo de titulación abarca el desarrollo de un sistema que permita la detección y conteo vehicular basado en técnicas de visión artificial, el cual enviará los datos del conteo vehicular que realice a un servidor en la nube para ser almacenados y posteriormente administrados bajo un entorno web.

Se estima que el tiempo necesario para dar cumplimiento a los objetivos propuestos anteriormente y culminar de forma total el proyecto presentado es de cuatrocientas horas, las cuales estarán distribuidas en las siguientes cuatro fases que se mencionan a continuación cada una con sus respectivas actividades y tareas.

FASE 1: Análisis de algoritmos de detección y reconocimiento de objetos.

- **Búsqueda de Algoritmos de detección y reconocimiento:** Se procederá a la recolección de información acerca de los algoritmos de detección y reconocimiento de objetos basados en visión artificial mediante el uso de diversas técnicas de recolección de información.
- **Comparativa de Algoritmos de detección y reconocimiento:** Una vez obtenida la información necesaria acerca de los algoritmos de detección y reconocimiento de objetos basados en visión artificial, se procederá a realizar una comparativa de algoritmos para conocer cuál de ellos se adapta mejor a las necesidades del presente proyecto
- **Selección de Algoritmos de detección y reconocimiento:** En base a la comparativa que se realice acerca de los algoritmos de detección y reconocimiento de objetos basados en visión artificial, se procederá a seleccionar un algoritmo para su posterior uso en el desarrollo del sistema de conteo y detección vehicular.

FASE 2: Diseño e implementación del sistema

- **Ingeniería de Software:** Se procederá a realizar las cuatro fases que conforman la ingeniería de software mismas que son: Elicitación, Análisis, Especificación y Validación de requerimientos. Las cuáles serán plasmadas en un documento que cumpla con los lineamientos establecidos en el estándar IEEE-830.
- **Diseño del Sistema:** En este apartado se procederá a elaborar el diseño correspondiente al sistema a desarrollar, el mismo que estará respaldado por una metodología de desarrollo de software que deberá ser seleccionada previamente al diseño del sistema.
- **Implementación del Sistema:** En este apartado se cumplirá con el proceso de codificación bajo el lenguaje de programación C++ para software de conteo y Java para la aplicación web, tomando en consideración los requerimientos y el diseño desarrollado en las fases anteriores.

FASE 3: Plan de Pruebas

- En este apartado se procederá a establecer un plan de pruebas el cual permitirá garantizar la calidad del software, así como también evaluar el sistema a desarrollar en todos sus módulos y funcionalidades.

FASE 4: Divulgación de información

- Una vez finalizado el presente trabajo se procederá a realizar la divulgación de la información obtenida en los procesos de desarrollo e implementación del proyecto a la comunidad científica.

F. MARCO TEÓRICO

- **Sistema**

Conjunto ordenado de normas y procedimientos que regulan el funcionamiento de un grupo o colectividad

- **Sistema Informático**

Un sistema informático es un conjunto de partes o recursos formados por el hardware, software y las personas que lo emplean, que se relacionan entre sí para almacenar y procesar información con un objetivo en común [5]

- **Visión Artificial**

La visión por computador es la transformación de datos de una imagen o una videocámara en una decisión o una nueva representación. Todas estas transformaciones se realizan para lograr un objetivo particular [4]

- **Tráfico**

Es la circulación de vehículos por una vía pública o una carretera. [6]

- **Tránsito**

Es la acción de ir de un lugar a otro, por vías o parajes públicos, el cual suele utilizarse para denominar al movimiento o actividad de los vehículos y las personas que pasan por una calle, carretera u otro tipo de camino. [6]

- **Tránsito vehicular**

El tránsito vehicular o automovilístico (también llamado tráfico vehicular, o simplemente tráfico) es el fenómeno causado por el flujo de vehículos en una vía, calle o autopista. Se presenta también con muchas similitudes en otros fenómenos como el flujo de partículas (líquidos, gases o sólidos) y el de peatones. [7]

- **Contaminación Auditiva**

La contaminación auditiva es el resultado de la producción de ruido que viaja por el aire lo cual afecta a las personas que transitan cada día por las avenidas más concurrentes de la ciudad lo cual puede verse reflejado en

problemas serios de salud mental y problemas fisiológicos. [8]

- **Semáforo**

Un semáforo es un dispositivo de regulación del tráfico que asigna de forma secuencial el derecho de paso a cada movimiento o grupo de movimientos que confluyen en una intersección. Para regular una intersección se coloca al menos un semáforo en cada uno de sus accesos, en cuya cabeza aparecen tres luces (roja, ámbar y verde) que se encienden sucesivamente. [6]

- **Flujo Vehicular**

La teoría de flujo del tránsito se basa en el desarrollo de las relaciones matemáticas entre los elementos primarios del flujo vehicular tales como flujo, densidad y velocidad. Mediante el análisis del flujo vehicular se pueden entender las características y el comportamiento del tránsito y se describe la forma como circulan los vehículos en cualquier sistema vial. [4]

- **Algoritmo**

Conjunto ordenado de operaciones sistemáticas que permite hacer un cálculo y hallar la solución de un tipo de problemas. [8]

- **Contador Vehicular**

Es un software que tiene por objetivo realizar el conteo de vehículos que se trasladan de un lugar a otro en diferentes sentidos.

G. METODOLOGÍA

Para el desarrollo del presente proyecto se hará uso de diferentes métodos y técnicas de recolección de información bibliográfica que ayuden a lograr cumplir los objetivos planteados.

1. Metodología para el desarrollo de software

Se plantea la utilización de una metodología que permita hacer un desarrollo ágil dado que da flexibilidad e intervalos de tiempos aceptables para el desarrollo de cualquier tipo de software, además el uso de la misma permite la generación de una documentación clara útil para los Stakeholders de un proyecto.

2. Métodos

2.1. Método Deductivo: Se hará uso de este método para plantear la problemática del proyecto y conocer acerca de problemas generales similares que han sido resueltos mediante software, lo que nos dará la pauta necesaria para enfocarnos en nuestro caso particular y desarrollar nuestro proyecto.

2.2. Método Inductivo: A través de este método se plantea conocer las necesidades y problemas que actualmente existen en el tema de tráfico vehicular que enfrentan la mayoría de ciudades, partiendo de conocimientos particulares y generalizarlos para todos los usuarios.

2.3. Enfoque de Sistemas: El método de enfoque se sistemas nos permitirá enfocarnos en profundidad en la resolución de nuestro problema encontrado en el presente proyecto, ya que con el uso de este método se logra tomar el problema como un sistema entero, y por ende descomponer el problema en varios módulos independientes para facilitar su resolución.

3. Técnicas

- 3.1. **Encuesta:** Con esta técnica se podrá obtener información relevante y necesaria referente a las necesidades que actualmente existen en cuanto al tema de tráfico vehicular, así como permitirá realizar parte de la elicitación de requerimientos necesarios para el desarrollo sistema.
- 3.2. **Técnica de la investigación Bibliográfica:** Con esta técnica será usada para recopilar información relevante del problema de resolver y sustentar la base teórica de la investigación, mediante consultas a: fuentes bibliográficas confiables, libros, revistas indexadas, artículos científicos, base de datos científicas entre otras.
- 3.3. **Técnica de la Observación:** Se usará esta técnica para visualizar aspectos que ayuden a recolectar información suficiente para lograr entender de mejor manera la realidad del problema y darnos una idea clara y concisa de los requerimientos que el sistema demanda.
- 3.4. **Técnica de la Entrevista:** Se hará uso de esta técnica ya que permitirá obtener información útil de una manera rápida, fácil de analizar y resultados cuantificables.

H. CRONOGRAMA

La tabla del cronograma de actividades que se presenta a continuación, muestra cómo están distribuidas y organizadas todas las tareas y actividades necesarias para llevar a cabo el desarrollo del presente proyecto durante las 400 horas destinadas para la elaboración del mismo, las columnas que contiene son: Nombre que es donde se especifica el nombre de la actividad, Inicia que indica el inicio de ejecución de dicha actividad, Termina es la fecha estimada de culminación de la actividad, Duración es el tiempo estimado en horas que tomará realizar la actividad o tarea y Recurso asignado es el encargado de realizar la actividad o tarea correspondiente

	Nombre	Inicia	Termina	Duración en horas	Recurso Asignado
1	“Sistema para detección y conteo vehicular aplicando técnicas de visión artificial”.	22/05/2017	20/10/2017	400	Fabricio Sisalima
2	FASE 1: Análisis de Algoritmos de Reconocimiento y detección vehicular	22/05/2017	02/06/2017	50	Fabricio Sisalima
3	Búsqueda de Algoritmos de detección y reconocimiento	22/05/2017	26/05/2017	30	Fabricio Sisalima
4	Comparativa de Algoritmos de detección y reconocimiento:	29/05/2017	01/06/2017	15	Fabricio Sisalima
5	Selección de Algoritmos de detección y reconocimiento	02/06/2017	02/06/2017	5	Fabricio Sisalima
6	FASE 2: Diseño del Sistema	05/06/2017	08/09/2017	250	Fabricio Sisalima
7	Ingeniería de Software	05/06/2017	07/07/2017	100	Fabricio Sisalima

8	Diseño del Sistema	10/07/2017	28/07/2017	50	Fabricio Sisalima
9	Implementación del Sistema	31/07/2017	08/09/2017	100	Fabricio Sisalima
10	FASE 3: Plan de Pruebas	11/09/2017	29/09/2017	50	Fabricio Sisalima
11	Ejecución del Plan de pruebas	11/09/2017	29/09/2017	50	Fabricio Sisalima
12	FASE 4: Divulgación de Información	02/10/2017	20/10/2017	50	Fabricio Sisalima
13	Divulgación de información obtenida en comunidades científicas	02/10/2017	20/10/2017	50	Fabricio Sisalima

Tabla 1 Descripción del Cronograma

Diagrama de Gantt

El Diagrama de Gantt que se presenta a continuación está desglosado en cuatro figuras, donde cada una de ellas hace referencia a una fase del proyecto a desarrollar y además se muestra en cada una de ellas la fecha de inicio, culminación y la duración que tomará realizar cada una de ellas.

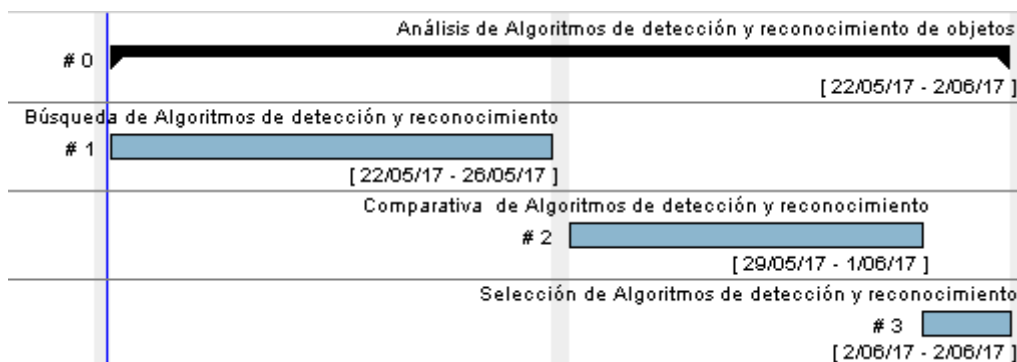


Figura 1 FASE 1 del Cronograma de Actividades

Fecha-Inicio: 22 de Mayo de 2017

Fecha-Fin: 2 de Junio de 2017

Duración: 50 Horas

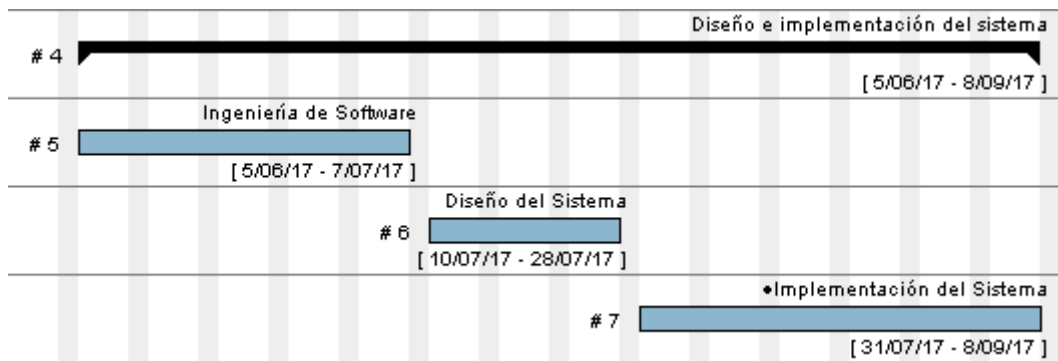


Figura 2 FASE 2 del Cronograma de Actividades

Fecha-Inicio: 5 de Junio de 2017 **Fecha-Fin:** 8 de Septiembre de 2017
Duración: 250 Horas

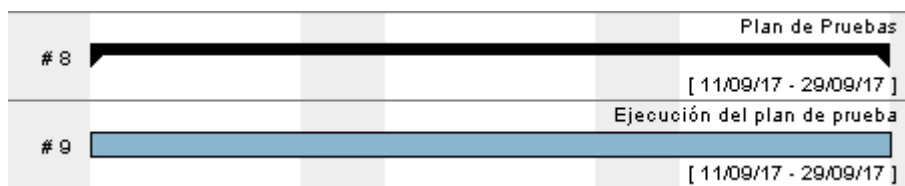


Figura 3 FASE 3 del Cronograma de Actividades

Fecha-Inicio: 11 de Septiembre de 2017 **Fecha-Fin:** 29 de Septiembre de 2017
Duración: 50 Horas

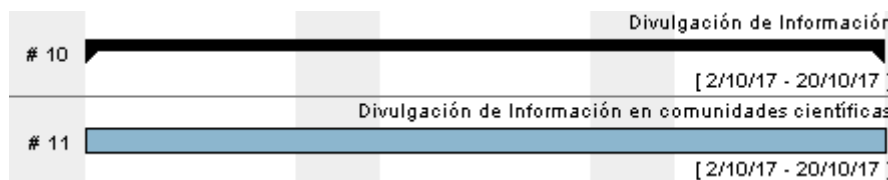


Figura 4 FASE 4 del Cronograma de Actividades

Fecha-Inicio: 2 de Octubre de 2017 **Fecha-Fin:** 20 de Octubre de 2017
Duración: 50 Horas

I. PRESUPUESTO Y FINANCIAMIENTO

1. Presupuesto

Para la elaboración del presupuesto se han tomado en cuenta los bienes, servicios, imprevistos y talento humano necesario para lograr llevar a cabo los objetivos que demanda este proyecto, los cuales son detallados a continuación.

1.1. Talento Humano

Para el desarrollo del presente proyecto se necesitará contar con los siguientes recursos que se describen en la Tabla 2, los mismos que cubrirán las actividades necesarias del trabajo en su totalidad.

ROL	Número de Horas	Valor por hora (\$)	Valor Total
Coordinador	50.00	15,00	750,00
Analista	100.00	20,00	2.000,00
Desarrollador	150.00	20,00	3.000,00
Evaluator	50.00	15,00	750,00
Tutor	50.00	25,00	1.250,00
Total (\$):			6.500,00

Tabla 2 Talento Humano para el proyecto

Para cubrir con los gastos pertenecientes a tutorías por parte del docente encargado del presente trabajo, correrán a responsabilidad netamente de la Universidad, por lo que el rubro especificado anteriormente por tutorías no se lo tomará como valor para el presupuesto

1.2. Bienes

Los siguientes recursos hardware y software que se presentan en la Tabla 3, representan todos bienes que serán necesarios adquirir para poder realizar sin inconvenientes el desarrollo del presente proyecto.

BIEN	Cantidad	Valor unitario	Valor Total
HARDWARE			
Depreciación Laptop	1	1200,00	1200,00
Depreciación Impresora	1	50,00	50,00
Memoria Flash	1	10,00	10,00
SubTotal (\$):			1260,00
SOFTWARE			
Gantt Project	1	00,00	00,00
IDE	1	00,00	00,00
Racional Rose	1	00,00	00,00
SubTotal (\$):			00,00
Total (\$):			1260,00

Tabla 3 Recursos Hardware y Software

1.3. Servicios

En el transcurso del desarrollo del presente proyecto será necesario adquirir ciertos servicios que servirán de complemento para culminar con éxito las tareas que demanda el presente trabajo.

SERVICIO	Cantidad	Valor unitario	Valor Total
Transporte	200	0,30	60,00
Copias	250	0,02	5,00
Resma de Papel	2	4,00	8,00
Anillados	6	1,00	6,00
Cartuchos	4	25,00	100,00
Internet	200 h	0,75	150,00
Total (\$):			329,00

Tabla 4 Prestación de Servicios a adquirir

1.4. Imprevistos

Para imprevistos se cree conveniente tomar el 10 % del valor total del presupuesto, los cuales serán agregados al valor total del proyecto.

RECURSO	SUBTOTAL
T. Humano	6500,00
Bienes	1260,00
Servicios	329,00
SubTotal (\$):	8.089,00
Imprevistos 10%	808,90
Total (\$):	8.897,90

Tabla 5 Presupuesto total del Proyecto

2. FINANCIAMIENTO

Al ser un proyecto de carácter académico todos los gastos que incurran en el presente proyecto en su totalidad serán cubiertos netamente por el autor de este trabajo, a excepción de los gastos por tutorías que será un valor cubierto por la Universidad Nacional de Loja, como se especifica en la tabla 1 referente a talento humano.

J. BIBLIOGRAFÍA

[1] QUINTANA ANDRÉS, MANZANO DAVID, MÚNERA LUIS, “Sistema de visión artificial para conteo de objetos en movimiento” [online]. Disponible en: <http://ingenieria.uao.edu.co/hombreymaquina/revistas/40%202012-3/art%2009%20Sistema%20de%20vision%20artificial.pdf>

[2] Thomson Ian, Bull Alberto, “La congestión del tránsito urbano: causas y consecuencias económicas y sociales” [online]. Disponible en: http://repositorio.cepal.org/bitstream/handle/11362/6381/S01060513_es.pdf?sequence=1&isAllowed=y

[3] [online]. Disponible en: <https://repository.javeriana.edu.co:8443/bitstream/handle/10554/7034/tesis480.pdf?sequence=1&isAllowed=y>

[4] Guamán Juan, “ETRANLOJA: Estudio y análisis de soluciones al congestionamiento vehicular en el centro histórico de la ciudad de Loja” Loja, 2012 [online]. Disponible en: <http://dspace.utpl.edu.ec/bitstream/123456789/4021/1/JUAN%20GABRIEL%20GUAMAN%20MOROCHO-ETRANLOJA.pdf>

[5] Alvarado José, Hernández Gustavo, “EFECTOS DEL MONÓXIDO DE CARBONO EN LA SALUD DE LOS COMERCIANTES DE LA CIUDAD DE LOJA, ABRIL – AGOSTO 2008” [online]. Disponible en: http://dspace.utpl.edu.ec/bitstream/123456789/1641/3/UTPL_Alvarado_Zumba_Jose_Daniel_360X1079.pdf

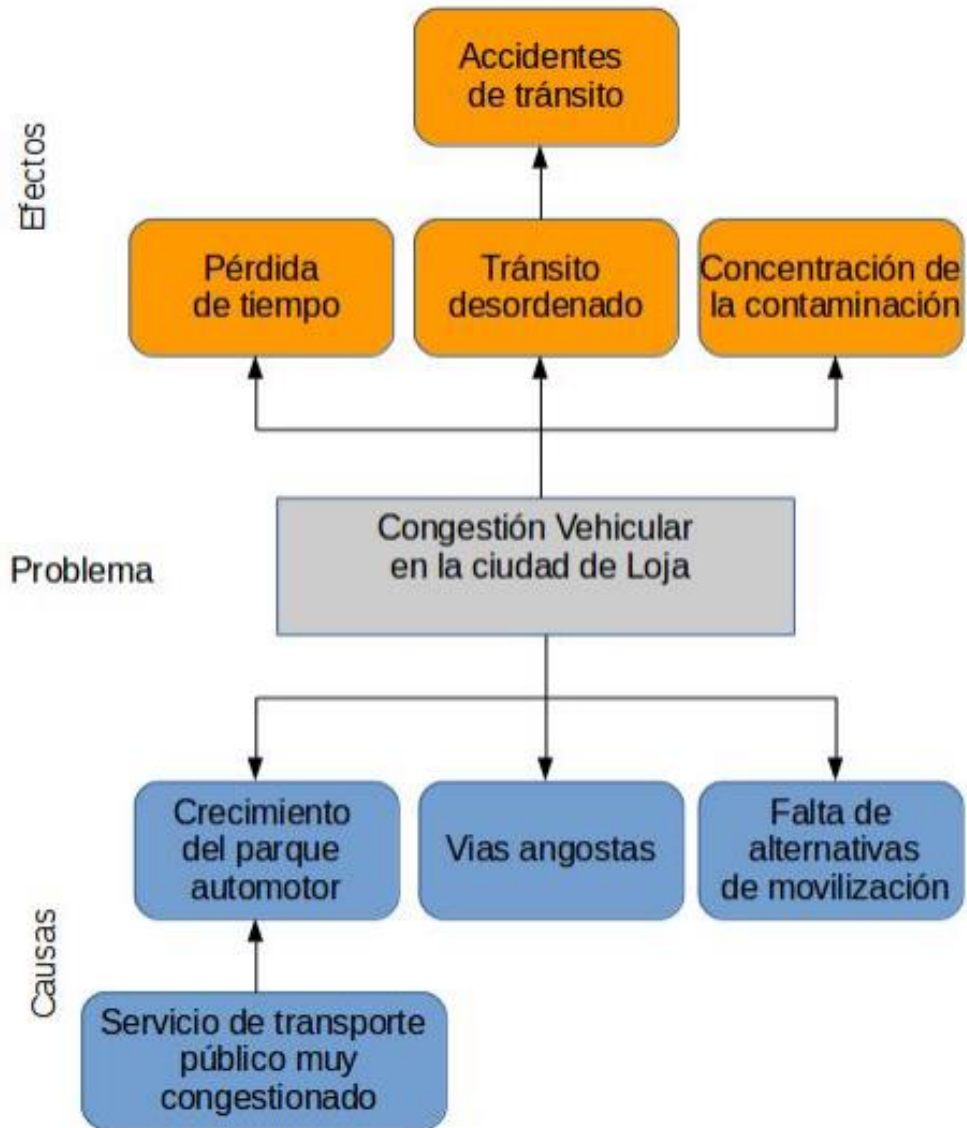
[6] Urrego Germán, “Adquisición de variables de tráfico vehicular usando visión por computador” [online]. Disponible en: <http://www.scielo.org.co/pdf/ring/n30/n30a2.pdf>

[7] González Aida, Hernández Melissa, García Rodrigo, Castilla Jonathan, “SISTEMA DE VISIÓN COMPUTACIONAL PARA LA DETERMINACIÓN DE LOS NIVELES DE SERVICIOS EN VÍAS PEATONALES DE LA ZONA CÉNTRICA DE LA CIUDAD DE MONTERÍA”, Universidad del Sinú Montería, Colombia [online]. Disponible en: <https://www.acofipapers.org/index.php/eiei2015/2015/paper/viewFile/1368/475>

[8] Baranyi Peter “7th International Symposium on Applied Machine Intelligence and Informatics” 2009 [online]. Disponible en: <http://ieeexplore.ieee.org/abstract/document/4956624/>

K. ANEXOS

ANEXO 1: Árbol de Problemas



ANEXO 2: Licencia Creative Commons



Sistema para detección y conteo vehicular aplicando técnicas de visión artificial by Sisalima Ortega Fabricio Roberto is licensed under a [Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional License](#).

You are free to:

Share — copy and redistribute the material in any medium or format

Adapt — remix, transform, and build upon the material

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:



Attribution — You must give [appropriate credit](#), provide a link to the license, and [indicate if changes were made](#). You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.



NonCommercial — You may not use the material for [commercial purposes](#).



ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the [same license](#) as the original.