



**UNIVERSIDAD
NACIONAL
DE LOJA**

TT-CIS



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

CARRERA DE INGENIERÍA EN SISTEMAS

**“Desarrollo de un Sistema de Visión Artificial para la
detección de transeúntes con discapacidad y vehículos
pequeños utilizando Software Libre”**

*Tesis previa a la obtención del
título de Ingeniero en Sistemas*

Autor:

- Fernando Adrián, Carrión Chamba

Tutor:

- Ing. Luis Roberto, Jácome Galarza, Mg. Sc

LOJA-ECUADOR

2017

Certificación del Director

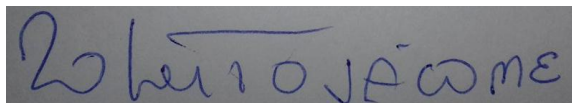
Ing. Luis Roberto Jácome Galarza, Mg. Sc.

DOCENTE DE LA CARRERA DE INGENIERÍA EN SISTEMAS DE LA UNIVERSIDAD NACIONAL DE LOJA, DIRECTOR DE TESIS

CERTIFICA:

Que el egresado **Fernando Adrián Carrión Chamba**, realizó el trabajo de investigación titulado **“Desarrollo de un sistema de Visión Artificial para la detección de transeúntes con discapacidad y vehículos pequeños utilizando Software Libre”** bajo mi dirección y asesoramiento, mismo que fue revisado, enmendado y corregido minuciosamente. En virtud que la Tesis reúne, a satisfacción las cualidades de fondo y forma exigidas para un trabajo de este nivel, autorizo su presentación, sustentación y defensa ante el tribunal respectivo.

Loja, 15 de Diciembre del 2016

A rectangular box containing a handwritten signature in blue ink that reads "Luis Roberto Jácome".

.....
Ing. Luis Roberto Jácome Galarza, Mg. Sc

DIRECTOR DE TESIS

Autoría

FERNANDO ADRIAN CARRIÓN CHAMBA declaro ser autor del presente trabajo de tesis 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 la tesis en el Repositorio Institucional – Biblioteca Virtual.

A rectangular image showing a handwritten signature in blue ink on a light-colored background. The signature is stylized and appears to be the name of the author.

Firma:

Cédula: 1104952328

Fecha: 29 de Mayo del 2017

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

Yo, **FERNANDO ADRIAN CARRIÓN CHAMBA**, declaro ser autor de la tesis titulada: **“DESARROLLO DE UN SISTEMA DE VISIÓN ARTIFICIAL PARA LA DETECCIÓN DE TRANSEÚNTES CON DISCAPACIDAD Y VEHÍCULOS PEQUEÑOS UTILIZANDO SOFTWARE LIBRE”**, 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 las 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 veintinueve días del mes de mayo del dos mil diecisiete.

Firma:



Autor: Fernando Adrián Carrión Chamba

Cédula: 1104952328

Dirección: Malacatos (Vilcabamba)

Correo Electrónico: facarrionc@unl.edu.ec/fernando20carrion@gmail.com

Teléfono: s/n

Celular: 0990398328

DATOS COMPLEMENTARIOS

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

Tribunal de Grado: Ing. Alex Vinicio Padilla Encalada, Mg. Sc.

Ing. Manuel Alberto Córdova Neira, Mg. Sc.

Ing. Roberto Carlos Pineda López, Mg. Sc.

Dedicatoria

Este trabajo se lo dedico a la memoria de mis padres Luz María y Víctor Enrique y a mis hermanos Adela, Rober y Margarita por ser mi principal inspiración y mi motor de vida para seguir siempre adelante cumpliendo cada una de las metas propuestas.

El Autor

Agradecimiento

Agradezco a Dios por haberme permitido cumplir con las metas que he planteado para mi vida, a mis padres Luz María y Víctor Enrique por haberme inculcado los valores necesarios para alcanzar mis objetivos.

También quiero agradecer a mis hermanos Adela, Rober y Margarita por haberme brindado siempre su apoyo incondicional en el transcurso de mi vida que fue fundamental para cumplir mis metas estudiantiles.

Además quiero expresar mi agradecimiento a la Universidad Nacional de Loja, al Área de la Energía las Industrias y los Recursos naturales no Renovables y a la Carrera de Ingeniería en Sistemas, por haberme abierto las puertas para mi formación académica y agradezco a todos los docentes que me brindaron su conocimiento, sus consejos y apoyo durante los cinco años de periodo universitario y de manera especial quiero agradecer al Ing. Luis Roberto Jácome Galarza, director de tesis, por el conocimiento profesional y la ayuda brindada que fue pilar fundamental para cumplir con los objetivos planteados en el presente trabajo.

El Autor

Índice de contenidos

Certificación del Director	ii
Autoría.....	iii
Carta de Autorización.....	iv
Dedicatoria.....	v
Agradecimiento	vi
Índice de contenidos	vii
Índice de figuras	xiii
Índice tablas	xv
1. Título	1
2. Resumen.....	2
2.1. Summary	3
4. Revisión de Literatura	5
4.1. Introducción a la Visión Artificial.....	5
4.1.1. Antecedentes.....	5
4.1.2. Visión Artificial.....	6
4.1.2.1. Concepto de Visión.....	6
4.1.2.2. Concepto de Visión Artificial.....	7
4.1.2.3. Etapas de una Aplicación de Visión Artificial.....	7
4.1.3. Adquisición y representación de imágenes digitales	8
4.1.3.1. Captura y digitalización de imágenes.....	8
4.1.3.3. Formatos de imágenes	9
4.2. Técnicas de Visión Artificial.....	10
4.2.1. Descriptores	10
4.2.1.1. LBP(Local Binary Patterns)	10
4.2.1.2. Histograma de Bloques	11

4.2.1.3.	HOG(Histogram Oriented Gradient)	11
4.2.1.4.	HAAR	12
4.2.2.	Clasificadores	12
4.2.2.1.	SVM(Support Vector Machines)	13
4.2.2.2.	AdaBoost	13
4.2.2.3.	Cascadas	14
4.2.3.	Búsqueda y Refinación.....	14
4.2.3.1.	Ventana Deslizante	14
4.2.3.2.	Pirámides.....	15
4.3.	Evaluación de Rendimiento.....	16
4.3.1.	Matriz de confusión	16
4.4.	Semáforos Inteligentes.....	18
4.4.1.	Tecnologías	18
4.4.1.1.	Semáforo Inteligente con RFID.....	19
4.4.1.2.	Semáforo inteligente usando Redes de Sensores Inalámbricos	19
4.4.1.3.	Semáforos Inteligentes mediante Procesamiento de Imágenes	19
4.4.1.4.	Semáforos Inteligentes basados en Inteligencia Artificial.....	20
➤	Lógica Difusa.....	20
➤	Algoritmos Genéticos	20
➤	Aprendizaje por Refuerzo	21
4.5.	Hardware y Software Libre	21
4.5.1.	Hardware Libre.....	21
4.5.1.1.	Arduino.....	21
4.5.1.2.	Raspberry Pi	22
4.5.1.3.	Banana Pi.....	23
4.5.1.4.	BeagleBoard	23
4.5.1.5.	Cuadro comparativo de Hardware Libre	24
4.5.2.	Software Libre	27
4.5.2.1.	Software para el procesamiento de Imágenes Digitales.....	27
➤	Java.....	27
➤	C.....	28

➤ C++	29
➤ Python	29
➤ MATLAB.....	30
➤ Comparativo de Software para el Procesamiento de Imágenes.....	31
4.5.2.2. Librerías para el procesamiento de Imágenes Digitales.....	32
➤ OpenCV(Open Source Computer Vision).....	32
➤ JavaVis	34
➤ SciPy.....	34
➤ Comparativo de Librerías de Visión Artificial.....	35
4.5.3. Internet de las Cosas.....	36
5. Materiales y Métodos	38
5.1. Materiales	38
5.1.1. Talento Humano.....	38
5.1.2. Servicios.....	38
5.1.3. Recursos Hardware y Software.....	39
5.1.4. Materiales de Oficina.....	39
5.2. Métodos	40
5.3. Metodología de Desarrollo.....	41
5.3.1. Metodología ICONIX	41
5.3.1.1. Fases de ICONIX.....	41
➤ Análisis de requisitos.....	41
➤ Análisis y Diseño preliminar	42
➤ Diseño	42
➤ Implementación	42
5.4. Técnicas.....	42
6. Resultados	44
6.1. Análisis de requerimientos.....	44
6.1.1. Requerimientos Funcionales	44
6.1.2. Requerimientos no Funcionales	44

6.1.3.	Modelo del Dominio.....	45
6.1.4.	Prototipado	46
6.1.5.	Modelo de Casos de Uso.....	49
6.2.	Análisis y Diseño Preliminar	49
6.2.1.	Descripción de Casos de Uso.....	49
6.2.1.1.	Acceso al Sistema.....	50
6.2.1.2.	Gestionar Usuario	51
➤	Crear Usuario	51
➤	Modificar Usuario	54
➤	Eliminar Usuario	56
6.2.1.3.	Consultar Estadísticas de Conteo de Objetos.....	58
6.2.1.4.	Realizar Conteo de Objetos	59
6.2.2.	Diagrama de Robustez.....	61
6.2.2.1.	Caso de Uso: Acceso al Sistema	61
6.2.2.2.	Caso de Uso: Gestionar Usuarios	62
➤	Crear Usuario	62
➤	Modificar Usuario	63
➤	Eliminar Usuario	64
6.2.2.3.	Consultar Estadísticas de Conteo de Objetos.....	65
6.2.2.4.	Realizar Conteo de Objetos	66
6.3.	Diseño	66
6.3.1.	Diagrama de Secuencia	66
6.3.1.1.	Acceso al Sistema.....	66
6.3.1.2.	Caso de Uso: Gestionar Usuario.....	67
➤	Crear Usuario	67
➤	Modificar Usuario	67
➤	Eliminar Usuario	68
6.3.1.3.	Consultar Estadísticas de Conteo de Objetos.....	68
6.3.1.4.	Realizar Conteo de Objetos	69
6.3.2.	Diagrama de Clases.....	70
6.3.3.	Modelo de Base de Datos	71

6.3.4.	Diagrama de Despliegue	72
6.3.5.	Arquitectura del Sistema.....	72
6.4.	Implementación	73
6.4.1.	Descripción de herramientas utilizadas para las implementación del software	73
6.4.1.1.	Sistema Operativo Linux	73
6.4.1.2.	Lenguaje de Programación C++	73
6.4.1.3.	IDE QT Creator	73
6.4.1.4.	Base de Datos MySQL	73
6.4.1.5.	OpenCV	73
6.4.1.6.	Construcción del Clasificador	73
6.4.2.	Implementación del Software.....	73
6.4.2.1.	Librerías utilizadas.....	74
6.4.2.2.	Método para cargar archivos	74
6.4.2.3.	Método para detectar objetos	74
6.4.2.4.	Método para dibujar rectángulos	74
6.4.2.5.	Método para realizar la conexión con la Base de Datos.....	75
6.4.2.6.	Método para insertar los datos en la Base de Datos	75
6.5.	Pruebas de Rendimiento de los Clasificadores	76
6.5.1.	Detección de Vehículos	76
6.5.1.1.	Prueba 1.....	76
6.5.1.2.	Prueba 2.....	77
6.5.1.3.	Prueba 3.....	78
6.5.1.4.	Prueba 4.....	79
6.5.1.5.	Prueba 5.....	80
6.5.1.6.	Diagrama de análisis de Detección de Vehículos	81
6.5.2.	Detección de Personas.....	81
6.5.2.1.	Prueba 1.....	82
6.5.2.2.	Prueba 2.....	83
6.5.2.3.	Prueba 3.....	84
6.5.2.4.	Prueba 4.....	85

6.5.2.5.	Prueba 5.....	86
6.5.2.6.	Diagrama de análisis de Detección de Personas	87
6.5.3.	Detección de Bicicletas	87
6.5.3.1.	Prueba 1.....	88
6.5.3.2.	Prueba 2.....	89
6.5.3.3.	Prueba 3.....	90
6.5.3.4.	Prueba 4.....	91
6.5.3.5.	Diagrama de análisis de Detección de Bicicletas.....	92
6.5.4.	Detección de Motos.....	92
6.5.4.1.	Prueba 1.....	93
6.5.4.2.	Prueba 2.....	94
6.5.4.3.	Diagrama de análisis de Detección de Motos.....	95
6.5.5.	Detección de Sillas de Ruedas	95
6.5.5.1.	Prueba 1.....	96
6.5.5.2.	Diagrama de análisis de Detección de Sillas de Ruedas.....	97
6.6.	Entrenamiento del detector de objetos	97
6.7.	Pruebas de acceso a la página web	101
7.	Discusión	103
8.	Conclusiones	105
9.	Recomendaciones	106
10.	Bibliografía	107
11.	Anexos	112
	Anexo 1: Especificación de Requerimientos.....	112
	Anexo 2: Creación de un Clasificador en Cascada	118
	Anexo 3: Instalación de OpenCV en Ubuntu	123

Índice de figuras

Figura 1. Etapas de una aplicación de visión artificial	8
Figura 2. Algoritmo de patrones binarios locales.....	10
Figura 3. Histograma por bloques	11
Figura 4. Recorrido piramidal de ventana deslizante	16
Figura 5. Muestra para la evaluación de la clase Persona	17
Figura 6. Matriz de confusión de la muestra de la Figura 5	17
Figura 7. Placa de Arduino	22
Figura 8. Raspberry Pi	22
Figura 9. Banana Pi	23
Figura 10. BeagleBoard	24
Figura 11. Java	27
Figura 12. Modelo de Dominio.....	45
Figura 13. Interfaz principal de la aplicación.....	46
Figura 14. Interfaz para login de Usuario	46
Figura 15. Pantalla de Inicio del Usuario.....	47
Figura 16. Pantalla de Inicio del Administrador.....	47
Figura 17. Interfaz para consulta de Estadísticas de conteo de Objetos.....	48
Figura 18. Pantalla para Gestionar Usuario.....	48
Figura 19. Modelo de Casos de Uso.....	49
Figura 20. Diagrama de Robustez para Acceso al Sistema.....	61
Figura 21. Diagrama de Robustez para Crear Usuario.....	62
Figura 22. Diagrama de Robustez para Modificar Usuario.....	63
Figura 23. Diagrama de Robustez para Eliminar Usuario.....	64
Figura 24. Diagrama de Robustez para Consultar Estadísticas de Conteo de Objetos.....	65
Figura 25. Diagrama de Robustez para Realizar Conteo de Objetos.....	66
Figura 26. Diagrama de Secuencia para Acceso al Sistema.....	66
Figura 27. Diagrama de Secuencia para Crear Usuario.....	67
Figura 28. Diagrama de Secuencia para Modificar Usuario.....	67
Figura 29. Diagrama de Secuencia para Eliminar Usuario.....	68
Figura 30. Diagrama de Secuencia para Consultar estadísticas de Conteo de Objetos....	68

Figura 31. Diagrama de Secuencia para Realizar Conteo de Objetos.....	69
Figura 32. Diagrama de clases.	70
Figura 33. Modelo de la Base de Datos.	71
Figura 34. Diagrama de Despliegue.....	72
Figura 35. Arquitectura del Sistema de Detección de Objetos.	72
Figura 36. Diagrama de análisis de detección de Vehículos.	81
Figura 37. Diagrama de análisis de detección de Personas.	87
Figura 38. Diagrama de análisis de detección de Bicicletas.	92
Figura 39. Diagrama de análisis de detección de Motos.	95
Figura 40. Diagrama de análisis de detección de sillas de ruedas.	97
Figura 41. filtros de Haar.....	97
Figura 42. Ejemplo de filtros de Haar.....	98
Figura 43. Primera etapa de entrenamiento.	100
Figura 44 Clasificador de la primera etapa	100
Figura 45 Segunda etapa de entrenamiento.....	100
Figura 46 Nodo del clasificador.....	101
Figura 47 Herramienta Load Impact.....	101
Figura 48 Inicio de prueba	102
Figura 49 Ejecución de la prueba	102
Figura 50 Prueba de acceso de usuarios a la página web.....	102

Índice tablas

Tabla 1. Tabla comparativa de Hardware Libre.	24
Tabla 2. Tabla comparativa del Software utilizado en el procesamiento de imágenes.	31
Tabla 3. Tabla comparativa de las librerías utilizadas en Visión Artificial.	35
Tabla 4. Descripción del Talento Humano.	38
Tabla 5. Descripción de Servicios.	38
Tabla 6. Descripción de Hardware y Software	39
Tabla 7. Descripción de Recursos Software.	39
Tabla 8. Descripción de Materiales de Oficina.	40
Tabla 9. Presupuesto Total.	40
Tabla 10. Requisitos Funcionales.	44
Tabla 11. Requisitos no Funcionales.	44
Tabla 12. Caso de Uso de Acceso al Sistema.	50
Tabla 13. Caso de Uso para Crear Usuarios.	51
Tabla 14. Caso de Uso para Modificar Usuario.	54
Tabla 15. Caso de Uso para Eliminar Usuario.	56
Tabla 16. Caso de Uso para Consultar Estadísticas de Conteo de Objetos.	58
Tabla 17. Caso de Uso para Realizar Conteo de Objetos.	59
Tabla 18. Método para cargar archivos	74
Tabla 19. Algoritmo para detectar objetos.	74
Tabla 20. Algoritmo para dibujar rectángulos	75
Tabla 21. Algoritmo para realizar la conexión con la base de datos.	75
Tabla 22. Algoritmo para insertar los datos en la base de datos.	75
Tabla 23. Información del video de Prueba 1 para detección de Vehículos.	76
Tabla 24. Matriz de confusión para primer video de detección de Vehículos.	76
Tabla 25. Información del video de Prueba 2 para detección de Vehículos.	77
Tabla 26. Matriz de confusión para segundo video de detección de Vehículos.	77
Tabla 27. Información del video de prueba 3 para detección de Vehículos.	78
Tabla 28. Matriz de confusión para tercer video de detección de Vehículos.	78
Tabla 29. Información del video de Prueba 4 para la detección de Vehículos.	79
Tabla 30. Matriz de confusión para cuarto video de detección de Vehículos.	79

Tabla 31. Información del video de Prueba 5 para la detección de Vehículos.....	80
Tabla 32. Matriz de confusión para quinto video de detección de Vehículos.....	80
Tabla 33. Información del video de Prueba 1 para la detección de Personas.	82
Tabla 34. Matriz de confusión para primer video de detección de Personas.	82
Tabla 35. Información del video de Prueba 2 para la detección de Personas.	83
Tabla 36. Matriz de confusión para segundo video de detección de Personas.....	83
Tabla 37. Información del video de Prueba 3 para la detección de Personas.	84
Tabla 38. Matriz de confusión para tercer video de detección de Personas.	84
Tabla 39. Información del video de Prueba 4 para la detección de Personas.	85
Tabla 40. Matriz de confusión para cuarto video de detección de Personas.	85
Tabla 41. Información del video de Prueba 5 para la detección de Personas.	86
Tabla 42. Matriz de confusión para quinto video de detección de Personas.	86
Tabla 43. Información del video de Prueba 1 para la detección de Bicicletas.	88
Tabla 44. Matriz de confusión para el primer video de detección de Bicicletas.....	88
Tabla 45. Información del video de Prueba 2 para la detección de Bicicletas.	89
Tabla 46. Matriz de confusión para el segundo video de detección de Bicicletas.....	89
Tabla 47. Información del video de Prueba 3 para la detección de Bicicletas.	90
Tabla 48. Matriz de confusión para el tercer video de detección de Bicicletas.....	90
Tabla 49. Información del video de Prueba 4 para la detección de Bicicletas.	91
Tabla 50. Matriz de confusión para el cuarto video de detección de Bicicletas.	91
Tabla 51. Información de video de Prueba 1 para la detección de Motos.....	93
Tabla 52. Matriz de confusión para el primer video de detección de Motos.....	93
Tabla 53. Información de video de Prueba 2 para la detección de Motos.....	94
Tabla 54. Matriz de confusión para el segundo video de detección de Motos.....	94
Tabla 55. Información de video de Prueba 1 para la detección de Sillas de Ruedas.....	96
Tabla 56. Matriz de confusión para el segundo video de detección de Sillas de Ruedas..	96
Tabla 57. Algoritmo Adaboost.....	98

1. Título

“Desarrollo de un Sistema de Visión Artificial para la detección de transeúntes con discapacidad y vehículos pequeños utilizando Software Libre”

2. Resumen

El presente Trabajo de Titulación consiste en el desarrollo de un sistema para la detección y conteo de personas, vehículos pequeños, bicicletas, motos y sillas de ruedas; el cual consta de dos partes. La primera parte corresponde al agente inteligente, el cual es el encargado de la captura de video en tiempo real, detección y conteo de objetos (personas, vehículos pequeños, bicicletas, motos y sillas de ruedas) y de realizar el envío de datos cada determinado tiempo a una base de datos remota; la segunda parte es una aplicación web la cual permite generar reportes de conteo de objetos y la administración de usuarios del sistema (agregar, eliminar, modificar).

En el diseño del sistema se utilizó la metodología ICONIX, como resultado de la aplicación de esta metodología se obtuvo el análisis de requerimientos a partir de los cuales se procedió a realizar la elaboración del modelo de dominio, casos de uso y prototipado; seguidamente se realizó el análisis y diseño preliminar en el que constan la descripción de casos de uso y diagrama de robustez; en la siguiente etapa de la metodología se especifica el comportamiento del sistema a través de los diagramas de secuencia y posteriormente se escribió el código de todo el diseño antes realizado.

Para la construcción de los detectores de objetos se utilizó las características de Haar como descriptores de la imagen y Adaboost como método de clasificación. Estas técnicas para la detección de objetos están basadas en el método propuesto por Viola y Jones enfocado principalmente a la detección de caras, pero se lo puede utilizar para la detección de cualquier tipo de objetos; dicho método proporciona una gran robustez y alta eficiencia respecto al tiempo de cálculo lo que implica que se puede implementar en tiempo real incluso en dispositivos con bajas características de procesamiento.

Finalmente, se procedió a realizar las pruebas de funcionalidad y rendimiento, para lo cual se evaluó a cada uno de los clasificadores en la detección de objetos, en este punto se tomó en cuenta algunos parámetros de evaluación tales como: Exactitud, Precisión, Especificidad, Sensibilidad y F1 Score; parámetros que se consideraron suficientes para saber si los resultados finales están acorde a los requisitos planteados.

2.1. Summary

The present work of degree consists of the development of a system for the detection and counting of people, small vehicles, bicycles, motorcycles and wheelchairs; Which consists of two parts. The first part corresponds to the intelligent agent, who is responsible for the capture of real-time video, detection and counting of objects (people, small vehicles, bicycles, motorcycles and wheelchairs) and send data to a remote database each time; The second part is a web application which allows to generate counting objects reports the users administration system (add, delete, modify).

In the design of the system ICONIX methodology was used, as a result of the application of this methodology was obtained the analysis of requirements from which proceeded to make the domain model, cases use and prototyping; Next, the analysis and preliminary design were carried out, which includes the description cases use and robustness diagram; In the next stage of the methodology the performance of the system is specified through the sequence diagrams and later the code of all the previous design was written.

For the construction of the object detectors the previous characteristics were used as descriptors of the picture and Adaboost as a classification method. These techniques for objects detection are based on the method proposed by Viola and Jones focused mainly on faces detection, but can be used for the detection of any type of objects; This method provides a high robustness and high efficiency with respect to the calculation time which implies that it can be implemented in real time even in devices with low processing characteristics.

Finally, we performed the functional and performance tests, for which we evaluated each of the classifiers in the objects detection, at this point we took into account some evaluation parameters such as: Accuracy, Specificity, Sensitivity and F1 Score; Parameters that were considered sufficient to know if the final results are related to the requirements.

3. Introducción

Durante las últimas décadas la industria y la tecnología han crecido aceleradamente a gran escala. Dentro del campo de Inteligencia Artificial se encuentra la Visión Artificial la cuál mediante el uso de técnicas adecuadas permite la obtención, procesamiento y análisis de cualquier tipo de información adquirida a través de imágenes digitales.

El Desarrollo de un sistema de Visión Artificial para la detección de transeúntes con discapacidad y vehículos pequeños utilizando Software Libre tiene la finalidad de brindar un aporte a la sociedad, que eleve el nivel de seguridad de los peatones en las intersecciones de las calles donde se encuentran ubicados los semáforos, además pretende servir como un módulo en el proyecto de semáforos inteligentes.

Las técnicas de Visión Artificial utilizadas para el desarrollo de este proyecto son los descriptores Haar, método de clasificación Adaboost y clasificador en Cascadas. El desarrollo de este proyecto se basó en el método para detección de objetos propuesto por Viola y Jones enfocado principalmente a la detección de caras, pero se lo puede utilizar para la detección de cualquier tipo de objetos; dicho método proporciona una gran robustez y alta eficiencia respecto al tiempo de cálculo lo que implica que se puede implementar en tiempo real incluso en dispositivos con bajas características de procesamiento.

El presente trabajo de titulación está conformado por varias secciones; la primera sección corresponde a: Tema, Resumen e Introducción del trabajo.

La siguiente sección corresponde a la revisión literaria, en la cual se describe las técnicas de Visión Artificial y las herramientas utilizadas con el desarrollo del proyecto, seguidamente se muestra los resultados que de acuerdo a la metodología ICONIX se debe cumplir con varias etapas como son: análisis de requisitos, análisis y diseño preliminar, diseño e implementación donde se escribe todo el código de acuerdo al diseño antes realizado y por último tenemos las pruebas donde se evalúa cada uno de los clasificadores para verificar su correcto funcionamiento

Finalmente, la sección última corresponde a las conclusiones y recomendaciones del trabajo de titulación.

4. Revisión de Literatura

4.1. Introducción a la Visión Artificial

4.1.1. Antecedentes

A lo largo de la historia la humanidad ha evolucionado, tanto social como tecnológicamente, para poder satisfacer las necesidades que se presentan día a día, gracias a esta habilidad y con la ayuda de la tecnología han descubierto varias soluciones, entre las cuales está la Inteligencia Artificial que posee infinidad de usos [1].

Entre las soluciones que se han dado gracias a la Inteligencia Artificial, se puede mencionar las siguientes:

- Un calentador de agua puede usar una máquina de estados finitos para su control. La lavadora automática de ropa puede utilizar lógica difusa para seleccionar el ciclo de lavado, la temperatura del agua y la cantidad de detergente [2].
- Las cámaras de video y de fotografía también pueden hacer uso de la Lógica Difusa para enfocar la imagen y otro tipo de tecnología para seleccionar las características necesarias para tomar la mejor foto [2].
- Los automóviles modernos usan sistemas inteligentes para proporcionar un mejor frenado, y evitar los derrapes en curvas y caminos los días de lluvia. Los videojuegos actuales hacen uso de la Inteligencia Artificial de muchas maneras diferentes: la selección de la estrategia por utilizar, la búsqueda del jugador dentro de un ambiente complejo, el evitar la colisión con el jugador, el control de los personajes, etc. [2].
- Sistemas de Inteligencia Artificial se usan en los mercados financieros o como apoyo en consultas médicas para la evaluación de los síntomas [3].
- Un sistema de Inteligencia Artificial puede ser usado para controlar el riego y la distribución de fertilizantes en las plantaciones. Incluso, se usa en las máquinas de búsqueda en Internet para dar mejores resultados y encontrar páginas significativas [3].

La Inteligencia Artificial es considerada una rama de la computación relacionada a un fenómeno natural con una analogía artificial a través de programas de computador, como muestra de ello tenemos el Test de Turing desarrollado 1950 por Alan Turing, se trata de un método para saber si una máquina es o no inteligente, en el cual un operador tiene que

mantener una conversación en dos sentidos con otra entidad, a través de un teclado, e intentar que la otra parte le diga si se trata de una máquina o de otro ser humano [4].

4.1.2. Visión Artificial

La ciencia Informática camina actualmente hacia la creación de computadoras cada vez más rápidas, más expertas y más autónomas. Proporcionar capacidad sensorial a un ordenador es una tarea difícil ya que entran en juego elementos accesorios al microprocesador, como son los sensores, las tarjetas adaptadoras de señal entre otros elementos; a pesar de todos estos inconvenientes, existe en la actualidad un interés especial por dotar a los ordenadores de uno de los cinco sentidos del hombre: la habilidad de ver, que en el mundo tecnológico toma el nombre de Visión Artificial [5].

En el año 1826 el químico francés Niepce llevó a cabo la primera fotografía, colocando una superficie fotosensible dentro de una cámara oscura para fijar la imagen, posteriormente, en 1838 el químico francés Daguerre hizo el primer proceso fotográfico práctico, Daguerre utilizó una placa fotográfica que era revelada con vapor de mercurio y fijada con trisulfato de sodio [5].

4.1.2.1. Concepto de Visión

La visión es el sentido más importante que tiene el ser humano. Así, mientras, para el oído se tiene alrededor de treinta mil terminaciones nerviosas, en la vista hay más de dos millones. La radiación exterior recibida por el ojo debe ser transformada en señales que sean procesadas por el cerebro. El ojo es el elemento transductor mientras que el cerebro es el que procesa dicha información. El ojo tiene una forma, aproximadamente, esférica de unos 2.5 cm de diámetro, está formado por una óptica y una zona sensorial, la óptica está constituida por la córnea, el iris o pupila y el cristalino. La córnea es un material transparente y funciona como un lente fijo. La pupila regula la cantidad de luz que entra en el interior y el cristalino hace las veces de lente variable, permitiendo el enfoque dependiendo de la distancia de los objetos. Los músculos que sujetan el cristalino provocan su deformación, cuya consecuencia es el enfoque del campo visual. La misión de la óptica del ojo es conducir la radiación electromagnética, del espectro visible, hacia la retina [6].

4.1.2.2. Concepto de Visión Artificial

La Visión Artificial tiene como finalidad la extracción de información del mundo físico a partir de imágenes, utilizando para ello un computador. La Visión Artificial es la utilización de las técnicas adecuadas, que permite la obtención, procesamiento y análisis de cualquier tipo de información especial obtenida a través de imágenes digitales [7].

Un sistema de Visión Artificial actúa sobre una representación de una realidad que le proporciona información sobre brillo, colores, formas, etcétera. Estas representaciones suelen estar en forma de imágenes estáticas, escenas tridimensionales o imágenes en movimiento. Una imagen bidimensional es una función que a cada par de coordenadas (x, y) asocia un valor relativo a alguna propiedad del punto que representa [5].

4.1.2.3. Etapas de una Aplicación de Visión Artificial

Cada aplicación de Visión Artificial tiene sus especificidades, se puede decir que existe un tronco común de etapas entre ellas, aunque la exposición muestra un encadenamiento temporal de una etapa sobre otra, no es real esta simplificación; para facilitar la comprensión y en la puesta en práctica siempre se encuentra realimentación entre las distintas fases [1].

La primera etapa es la construcción del sistema de formación de las imágenes. Su objetivo es realzar, mediante técnicas fotográficas, las características visuales de los objetos.

Una vez adquirida la imagen se pasará a la etapa de procesado para mejorar la calidad informativa de la imagen, su contraste y las características de la imagen como bordes o áreas.

Segmentación es la fase donde se particiona la imagen en áreas con significado; existen varias técnicas: umbralización, discontinuidad, crecimiento de regiones, uso del color o del movimiento, etc.

Una vez dividida la imagen en zonas con características de más alto nivel se extrae las características como: área, perímetro, textura, color, etc.

Con las características analizadas de cada región se debe de clasificar e interpretar. Por tanto, se diseñarán clasificadores que le dé a cada área segmentada una etiqueta de alto nivel, como por ejemplo, en una imagen de áreas que zonas son tierras de cultivo y que zonas no.

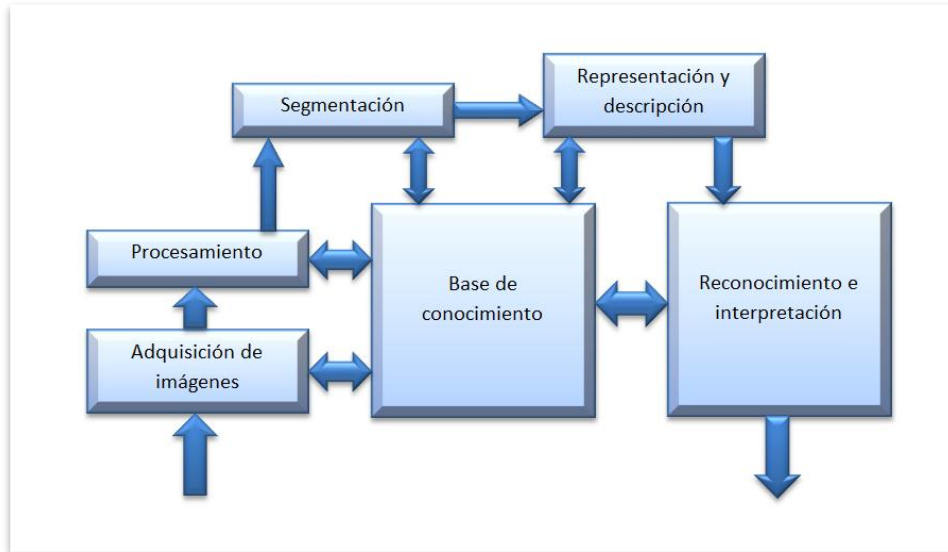


Figura 1. Etapas de una aplicación de visión artificial [8].

4.1.3. Adquisición y representación de imágenes digitales

Una vez digitalizada, una imagen bidimensional está constituida por un conjunto de elementos llamados píxeles. Cada píxel ofrece información sobre una región elemental de la imagen. En imágenes en niveles de gris esta información es el brillo. En imágenes de color, la información corresponde a la intensidad de cada una de las componentes de una base de color [9].

4.1.3.1. Captura y digitalización de imágenes

Las imágenes digitales son señales discretas, que suelen tener origen en una señal continua. En el proceso de obtención de imágenes digitales se distinguen dos etapas: la primera, conocida como captura utiliza un dispositivo generalmente óptico con la que obtiene información relativa de la escena; la segunda etapa, que se conoce como digitalización, en la cual se transforma esa información que es una señal con una o varios componentes continuas en la imagen digital con todas sus componentes discretas [10].

La digitalización es el proceso de paso del mundo continuo al mundo discreto, en la digitalización normalmente se distinguen dos procesos: el muestreo y la cuantización. El muestreo de una señal continua consiste en la medición a intervalos, siendo su parámetro fundamental la frecuencia de muestreo, que representa el número de veces que se mide un valor analógico por unidad de cambio y la cuantización es la discretización de los posibles valores de cada píxel [10].

4.1.3.3. Formatos de imágenes

De manera general existen dos tipos de imágenes digitales; Imágenes vectoriales y las imágenes de mapas de bits [20].

Las imágenes vectoriales son imágenes constituidas por objetos geométricos autónomos (líneas, curvas, polígonos), definidos por ciertas funciones matemáticas (vectores) que determinan sus características (forma, color, posición), se crean con programas de diseño o dibujo vectorial y suelen usarse en dibujos, rótulos, logotipos. Su principal ventaja es que una imagen puede ampliarse sin sufrir el efecto de pixelado que tienen las imágenes de mapa de bits al aumentarse [20].

Las imágenes de mapa de bits dependen de la resolución, es decir, contienen un número fijo de píxeles. Cada uno de estos píxeles posee una situación y un valor de color concreto. Cuando trabajamos sobre un mapa de bits, lo que hacemos en realidad es trabajar sobre cada uno de estos puntos. El píxel es por tanto, dentro de la imagen, la unidad de información básica. Los píxeles están colocados de tal manera que juntos forman una rejilla, cada celda de la rejilla es un píxel y todos juntos forman la imagen [20].

Para poder reproducirse o utilizarse en un ordenador u otros dispositivos las imágenes vectoriales y de mapa de bits se guardan en archivos o ficheros (conjunto de datos que se almacenan en algún medio, como un disco duro, DVD, flash memory) Cada archivo gráfico, se identifica además de por su nombre, por su extensión, que indica el tipo o formato de que se trata. A continuación se presentan algunos formatos de imágenes [21]:

- **BMP.** Formato introducido por Microsoft y usado originariamente por el sistema operativo Windows para guardar sus imágenes.
- **GIF.** Formato bastante antiguo desarrollado por CompuServe con el fin de conseguir archivos de tamaño muy pequeños. Admite solo 256 colores por lo que no es adecuado para imágenes fotográficas pero si es muy apropiado para logotipos, dibujos, etc. Permite crear animaciones (gif animado) y transparencias.
- **JPEG.** Es uno de los formatos más conocido y usado para fotografías digitales ya que admite millones de colores. Lo admiten la mayor parte de las cámaras fotográficas y escáneres y es muy utilizado en páginas web, envío de fotografías por correo electrónico, presentaciones multimedia y elaboración de vídeos de fotografías.
- **PNG.** Formato creado con el fin de sustituir a GIF. Utiliza sistemas de compresión gratuitos, y admite muchos más colores que GIF. También admite transparencias pero

no animaciones. Al admitir más colores es posible crear imágenes transparentes con mayor detalle.

- **RAW.** Formato “en bruto”. Esto quiere decir que contiene todos los píxeles de la imagen captada, tal y como se han tomado. Es el formato que ofrece la mayor calidad fotográfica y suele ser admitido por cámaras de gama media y alta (réflex, y compactas) indicadas para fotógrafos aficionados avanzados y profesionales.

4.2. Técnicas de Visión Artificial

4.2.1. Descriptores

4.2.1.1. LBP(Local Binary Patterns)

El algoritmo de Patrones Binarios Locales tiene sus raíces en el análisis de la textura 2D. La idea básica es resumir la estructura local de una imagen mediante la comparación de cada píxel con su vecindario. Toma un píxel como centro y el umbral de sus vecinos en contra. Si la intensidad del píxel central es mayor o igual a la de su vecino se denotará con 1 a su vecino y con 0 en caso contrario [22].

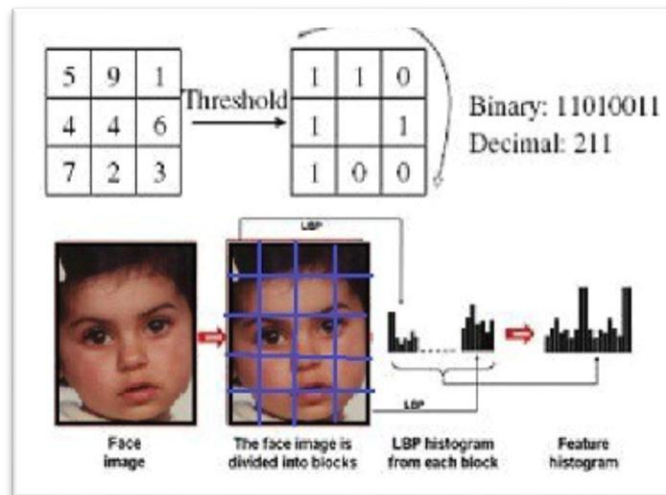


Figura 2. Algoritmo de patrones binarios locales [22].

LBP es un muy eficiente operador de textura que trabaja con cada uno de los píxeles y sus vecinos. Debido a su poder de discriminación y la simplicidad computacional LBP se ha convertido en un método popular en diversas aplicaciones y puede verse como un enfoque unificador para los modelos estadísticos y estructurales tradicionalmente divergentes de análisis de textura [23].

Una de la propiedad más importante del LBP en aplicaciones del mundo real es su robustez a los cambios monotónicos en escala de grises causadas por la variación de iluminación. Otra propiedad importante es su simplicidad computacional, lo que hace posible el análisis de imágenes en los ajustes en tiempo real desafiantes [23].

4.2.1.2. Histograma de Bloques

Los histogramas se usan como descriptores eficientes de la imagen, especialmente cuando son transformados a su código ULBP (Uniform Local Binary Patterns) pero en ocasiones determinadas, su invariante global ante la traslación produce efectos indeseados en el comportamiento del identificador [24].

Si en una aplicación se requiere detectar una persona, sin importar si la misma se encuentra en varias imágenes desordenadas, este método funcionará sin problema ya que las imágenes generarán el mismo histograma. Si por alguna razón no se aplican los histogramas por bloques sobre una misma imagen puede presentar errores y detectar más de un objeto como se muestra a continuación [24]:

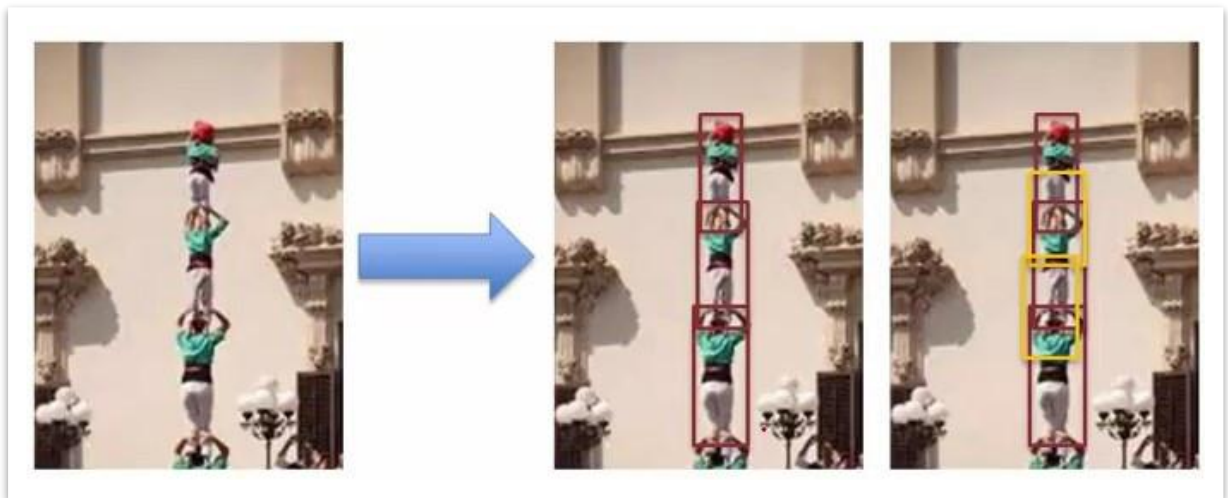


Figura 3. Histograma por bloques [24].

4.2.1.3. HOG(Histogram Oriented Gradient)

La técnica descriptor HOG cuenta las ocurrencias de la orientación del gradiente en porciones localizadas de una imagen o región de interés [25].

Aplicación del algoritmo de descriptor HOG es el siguiente [25]:

- Dividir la imagen en pequeñas regiones, llamadas celdas conectadas, y para cada celda calcular un histograma de gradiente direcciones u orientaciones de borde para los píxeles dentro de la célula.
- Discretizar cada célula en los contenedores angulares de acuerdo con la orientación del gradiente.
- Los grupos de células adyacentes se consideran como regiones espaciales denominan bloques. La agrupación de células en un bloque es la base para la normalización de histogramas.
- Grupo normalizado de histogramas representa el histograma bloque. El conjunto de estos histogramas de bloque representa el descriptor.

4.2.1.4. HAAR

Este algoritmo permite obtener información de una zona concreta mediante una operación aritmética simple: Esto nos lleva a una eficiencia de cálculo.

En concreto se usan tres características de HAAR [26]:

- Característica de dos rectángulos: es la diferencia entre la suma de los píxeles de ambas regiones rectangulares. Las regiones tienen el mismo tamaño y forma y están horizontalmente o verticalmente adyacentes.
- Característica de tres rectángulos: es la suma de los píxeles en ambos rectángulos exteriores sustraídos por la suma en el rectángulo central.
- Característica de cuatro rectángulos: es la diferencia entre los pares diagonales de los rectángulos.

De manera simplificada, las características pueden ser vistas como evaluaciones de la intensidad de conjuntos de píxeles. La suma de la luminancia de los píxeles en la región blanca se resta de la suma de los píxeles en la región oscura. El valor obtenido mediante la diferencia es el valor de la características las cuales pueden estar en cualquier posición y escala de la imagen original y puede combinarse con otros. Cada tipo de característica puede indicar la existencia o no de un tipo de características en la imagen [26].

4.2.2. Clasificadores

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. Un ejemplo de un clasificador es aquel que acepta datos de sueldos de una persona, edad,

estado civil, dirección e historial de crédito y clasifica a la persona como aceptables o inaceptables para recibir una nueva tarjeta de crédito o préstamo [27].

Una de las posibles combinaciones de un conjunto de clasificadores es la cascada y ésta dará un mejor resultado dependiendo de la cantidad de datos que se introduzcan, el número de clasificadores, etc. La motivación para hacer uso de cascadas se debe a que éstas se usan para aprender conjuntos desbalanceados de datos. Por ejemplo, en el caso de querer aprender “cabeza” contra cualquier otro elemento del mundo visual, incluimos en la cascada más imágenes de lo que no es “cabeza” [27].

Una vez encontradas las características dentro de una imagen, el objetivo del sistema es encontrar aquellas características que mejor definan una cara o parte superior del cuerpo y nos ayuden a localizarla en la imagen o video. La hipótesis planteada en las publicaciones de **Viola y Jones** establece que un pequeño número de esas características pueden combinarse para formar un clasificador. El algoritmo se utiliza para mejorar el rendimiento de un algoritmo de aprendizaje simple [28].

4.2.2.1. SVM(Support Vector Machines)

Estos métodos están propiamente relacionados con problemas de clasificación y regresión. Dado un conjunto de ejemplos de entrenamiento podemos etiquetar las clases y entrenar una SVM para construir un modelo que representa a los puntos de muestra en el espacio, separando las clases por un espacio lo más amplio posible [29].

Las máquinas de vectores de soporte son efectivas en espacios de alta dimensión, siendo eficaz en los casos en número de dimensiones es mayor que el número de muestras, utilizan un subconjunto de puntos de entrenamiento en la función de decisión, si el número de características es mucho mayor que el número de muestras, es probable que el método pueda dar malos resultados [30].

4.2.2.2. AdaBoost

El algoritmo de aprendizaje AdaBoost, que significa “Adaptative Boosting”, se utiliza para elegir los mejores clasificadores simples y formar la función de clasificación. Es uno de los algoritmos más utilizados en aprendizaje automático. La ventaja principal de AdaBoost es su velocidad de aprendizaje [26].

Teniendo en cuenta que existe un clasificador simple por cada característica hay que evaluar un total de $K*N$ clasificadores siendo K el número de características por imagen y N el número de imágenes que contenga el conjunto de entrenamiento.

En este algoritmo los pesos se asignan de manera que favorezcan la clasificación de los objetos, consiguiendo que estos tengan mayores pesos o importancia. Las mejores características se eligen basándose en el error ponderado que se produce, este error ponderado es una función que utiliza los errores pertenecientes a los ejemplos de entrenamiento. El peso de un objeto clasificado correctamente se modifica, aumenta, mientras que el peso de un objeto mal clasificado se mantiene constante. Con esto se consigue que sea más difícil que la segunda característica clasifique erróneamente un objeto que haya sido clasificado erróneamente por la primera característica frente a un objeto clasificado correctamente por esa primera característica [31] [26].

4.2.2.3. Cascadas

El principio básico del algoritmo de detección **Viola y Jones** consiste en escanear el detector muchas veces a través de una misma imagen, en diferentes posiciones y a distintas escalas. Incluso si una imagen contiene muchos objetos está claro que la mayoría de las sub-ventanas que se escaneen no contendrán ningún objeto [27].

Esto lleva a una nueva manera de ver el problema: Por ejemplo si se quisiera detectar caras, en vez de encontrar caras, el algoritmo debería descartar no-caras. La idea subyacente se basa en que es más fácil descartar una imagen que no contenga una cara que encontrar una cara en una imagen. Parece ineficiente construir un detector que contenga un solo clasificador fuerte, puesto que el tiempo de clasificación será constante sin importarnos la entrada del clasificador, a consecuencia de que el clasificador tiene que evaluar todas las características que lo forman. Aumentar la velocidad de clasificación generalmente implica que el error de clasificación aumentará inevitablemente, dado que para disminuir el tiempo de clasificación se debería disminuir el número de clasificadores simples que se utilizan [31] [28].

4.2.3. Búsqueda y Refinación

4.2.3.1. Ventana Deslizante

Teniendo una ventana con dimensiones determinadas la cual contiene la muestra clasificada previamente durante el aprendizaje, se pretende buscar equivalencias dentro de una nueva imagen.

Se determina una posición inicial, la cual usualmente es la esquina superior izquierda, y a partir de esta se dibuja una ventana de análisis, se toma el descriptor de nueva ventana y se la envía al módulo clasificador donde se determina si se trata de una detección válida. Posteriormente esta ventana se desplaza un intervalo “delta” en X y un “delta” en Y para luego repetir el procedimiento descrito. Un concepto importante durante estos procedimientos es la de evitar la redundancia de cálculos, dado que se trabaja con histogramas de código ULBP. Ante esto se proponen dos soluciones [32]:

La primera consiste en convertir toda la imagen a su equivalente ULBP antes de realizar el método de la ventana deslizante, de esta forma sin importar el delta en el paso de la traslación de la ventana, se evita la redundancia de este cálculo pues el sistema únicamente deberá obtener el histograma de la ventana enviada.

La segunda propuesta consiste en calcular los histogramas de sub-bloques de dimensiones $\Delta x * \Delta y$ de la imagen ULBP completa de manera que la ventana deslizante únicamente debe concatenar los histogramas encerrados, reduciendo al mínimo los cálculos necesarios y eliminando las redundancias, este proceso está esquematizado en la imagen anterior.

4.2.3.2. Pirámides

Existe un problema con la ventana de dimensiones definidas que permite únicamente localizar objetos dentro de la imagen que mantengan exactamente dichas dimensiones. Existen varias técnicas a fin de solventar esta dificultad, una de ellas es conocida como filtros Haar, que consiste, a grandes rasgos, en re-escalar la ventana deslizante entre pasadas. La técnica que se puede utilizar consiste en mantener el tamaño de la ventana y escalar con cada pasada la imagen sujeta al análisis, con esto, la primera imagen detectará los objetos que aparezcan con menor tamaño, y la última localizará los objetos que aparezcan con mayor tamaño dentro de la imagen. Se lo conoce como Pyramidal Sliding Window [31].

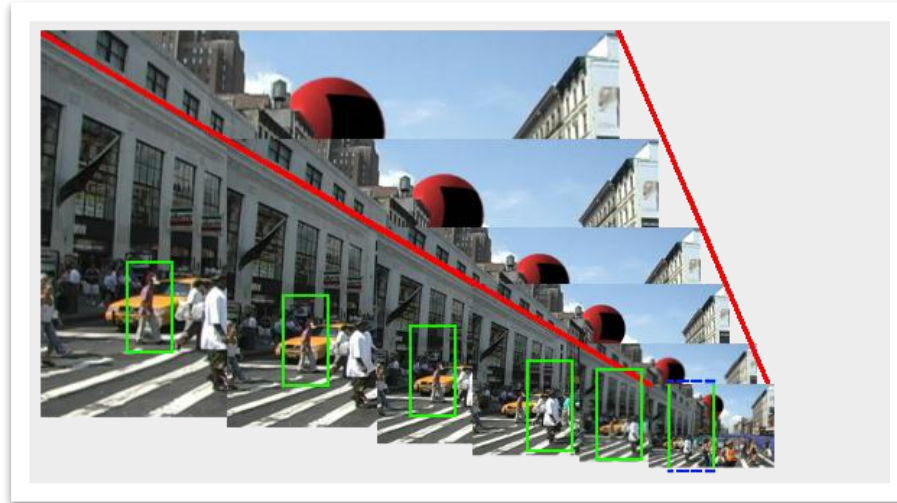


Figura 4. Recorrido piramidal de ventana deslizante [31].

Es adecuado definir un nuevo parámetro que indique el nivel de escalado entre cada imagen, para cuanto debe dividirse cada imagen superior para convertirse en la imagen inmediatamente posterior. La punta de la pirámide debe ser el tamaño más pequeño que permita al contener dentro de la misma la ventana deslizante, como se ve en la Figura 4, la última imagen es demasiado pequeña y por lo tanto no debería incluirse en este algoritmo. Al realizar el escalado de la imagen es importante usar algún método que no introduzca ruido dentro de la imagen y pueda provocar fallos del algoritmo [33].

4.3. Evaluación de Rendimiento

4.3.1. Matriz de confusión

La matriz de confusión es una herramienta básica que permite evaluar el nivel de confusión de un clasificador [34], por ejemplo si se estuviera evaluando la clase persona y se tuviera la siguiente muestra (ver Figura 5), la matriz de confusión nos quedaría de la siguiente manera (ver Figura 6):

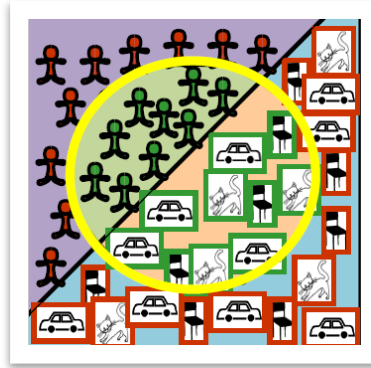


Figura 5. Muestra para la evaluación de la clase Persona [34].

		Resultado Clasificación	
		PERSONA	NO-PERSONA
Instancias Reales	PERSONA	Reales Positivos	Falsos Negativos
	NO-PERSONA	Falsos Positivos	Reales Negativos

Figura 6. Matriz de confusión de la muestra de la Figura 5 [34].

Las medidas que se pueden sacar sobre la matriz de confusión son las siguientes:

Exactitud: Es la proximidad que existe entre el resultado y la clasificación exacta, su fórmula es la siguiente [34]:

$$Exactitud = \frac{Reales\ Positivos + Reales\ Negativos}{Predicciones\ Totales}$$

Precisión: Es la calidad de respuesta de un clasificador, su fórmula es la siguiente [34]:

$$Precisión = \frac{Reales\ Positivos}{Reales\ Positivos + Falsos\ Positivos}$$

Sensibilidad: Es la eficiencia en la clasificación de todos los elementos que son de la clase, su fórmula correspondiente es la siguiente [26]:

$$Sensibilidad = \frac{Reales\ Positivos}{Reales\ Positivos + Falsos\ Negativos}$$

Especificidad: Es la eficiencia en la clasificación de todos los elementos que no son de la clase, su fórmula es la siguiente [26]:

$$Especificidad = \frac{Reales\ Negativos}{Reales\ Negativos + Falsos\ Positivos}$$

F1 Score: Es la medida de exactitud de una prueba, es la medida ponderada que tiene en cuenta a los falsos positivos y falsos negativos [35].

$$F1\ Score = 2 * \frac{Precisión * Sensibilidad}{Precisión + Sensibilidad}$$

4.4. Semáforos Inteligentes

Los semáforos son dispositivos utilizados para gestionar el tráfico, permitiendo o impidiendo el paso de conductores y peatones en cada vía [36].

Un Semáforo Inteligente es todo aquel capaz de tomar decisiones dependiendo de una serie de parámetros de entrada (flujo de vehículos, velocidad media, etc.), el comportamiento del mismo, es de forma dinámica y se ajusta en base a varios parámetros [36].

Estos semáforos se presentan para solucionar varios problemas de tránsito comunes en todo el mundo, por citar algunos:

- Congestión vehicular.
- Tiempos excesivos de viaje.
- Esperas innecesarias.
- Mayor gasto de combustible.
- Mayor contaminación en el medio ambiente.

4.4.1. Tecnologías

Existen diferentes formas de implementar semáforos con inteligencia, esto puede abarcar desde una simple decisión del estado en que debe permanecer de acuerdo al tránsito hasta una decisión de mayor jerarquía, como tomar una decisión acerca de un accidente, cambiar el flujo de tráfico, etc.

4.4.1.1. Semáforo Inteligente con RFID

La idea principal es tener un control inteligente del tráfico, haciendo variar los tiempos en los puntos de intersección de acuerdo a la congestión vehicular. El sistema consta de 4 partes principales: la tarjeta RFID, punto de acceso, ubicación de servidor de redes, red WAN, además cuenta con una base centralizada que almacena datos y a partir de ellos realiza una acción [37].

El proceso de decisión para el control de tráfico inteligente depende de la información en tiempo real proporcionado por el sistema RFID, se captan los datos y son almacenados en una base de datos centralizada. En cada cruce, la espera de los Semáforos depende de la congestión, la longitud de la cola y otros parámetros de entrada, la comunicación entre el semáforo y el algoritmo de decisión se hace a través de Internet [38].

4.4.1.2. Semáforo inteligente usando Redes de Sensores Inalámbricos

La red de sensores inalámbricos, consiste en un grupo de sensores diseñados para proporcionar la infraestructura de comunicación de tráfico y facilitar el flujo del tránsito. Cada sensor se encarga de generar los datos de tráfico como el número de vehículos, los procesos de salida, velocidad de cada vehículo, su longitud, los datos se envían en tiempo real a la base [38].

El sistema de detección de vehículos requiere cuatro componentes: un sensor para detectar las señales generadas por los vehículos, un procesador para los datos detectados, una unidad de comunicación para transferir los datos a la base, y una fuente de energía continua.

4.4.1.3. Semáforos Inteligentes mediante Procesamiento de Imágenes

El control de tráfico utilizando procesamiento de imágenes, significa, medir la densidad del tráfico y de acuerdo a los datos obtenidos modificar los tiempos del semáforo. De la siguiente manera [38]:

- Adquisición de Imágenes, se captura la carretera vacía y la imagen con el tráfico actual, la imagen de la carretera se guarda como una imagen de referencia.
- Se convierte ambas imágenes RGB a escala de grises.
- Mejora la imagen.
- Verificar coincidencia entre la imagen de referencia y la imagen con el tráfico actual.

Después de un procedimiento de detección de bordes de ambas imágenes, los semáforos se pueden controlar basándose en el porcentaje de coincidencia.

4.4.1.4. Semáforos Inteligentes basados en Inteligencia Artificial

Existen diferentes enfoques en cuanto a este tema, entre ellos están los basados en lógica difusa, algoritmos genéticos y refuerzo de aprendizaje.

➤ Lógica Difusa

La tecnología de Lógica Difusa permite la aplicación de las reglas de similar manera en la que los seres humanos podrían pensar. Por ejemplo, los seres humanos podrían pensar de la siguiente forma para controlar el tráfico en un cruce seguro: “si el tráfico es más pesado en el norte o el sur de los carriles y el tráfico en el carril este u oeste es menor entonces el semáforo debería permanecer verde más tiempo en los carriles norte y sur” [37].

Lo que se desea lograr es:

- Determinar la presencia y ausencia de vehículos con imágenes de las rutas.
- Mantener la luz verde por más tiempo en caso de mayor flujo de vehículos.
- Si un carril está vacío, buscar un carril con vehículos y colocar dicho carril en verde.

➤ Algoritmos Genéticos

Los Algoritmos Genéticos (AGs) son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Están basados en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza acorde con los principios de la selección natural y la supervivencia de los más fuertes, postulados por Darwin. Por imitación de este proceso, los Algoritmos Genéticos son capaces de ir creando soluciones para problemas del mundo real. La evolución de dichas soluciones hacia valores óptimos del problema depende en buena medida de una adecuada codificación de las mismas [39].

Un Algoritmo Genético consiste en una función matemática o una rutina de software que toma como entradas a los ejemplares y retorna como salidas cuáles de ellos deben generar descendencia para la nueva generación [39].

➤ **Aprendizaje por Refuerzo**

Es un tipo de máquina de aprendizaje, permite a las máquinas y agente de software determinar automáticamente el comportamiento dentro de un contexto con el fin de maximizar su rendimiento. El modelo básico de Reinforcement Learning consiste en [40]:

- Un conjunto de estados de entorno.
- Un conjunto de acciones.
- Reglas de transición entre los estados.
- Reglas que determinan una transición.
- Reglas que describen lo que observa un agente.

Un agente de Reinforcement Learning interactúa con su entorno en pasos de tiempo discretos, el objetivo es recoger tanta información como sea posible, el agente puede elegir una acción en función de la información [40].

4.5. Hardware y Software Libre

4.5.1. Hardware Libre

Hardware abierto o libre es una propuesta casi tan antigua como la del Software Libre, sin embargo su empleo no es tan directo. Compartir diseños hardware es más complicado. Hardware Libre es un Hardware cuyo diseño se pone a disposición del público de modo que cualquiera pueda estudiar, modificar, distribuir, hacer y vender el diseño o el hardware que se sustente en dicho diseño [41].

El Hardware Libre utiliza elementos y materiales inmediatamente disponibles, procesos estandarizados, infraestructura abierta, contenido no restringido y herramientas de diseño libres para aumentar al máximo la capacidad de los individuos de hacer y utilizar el hardware. El Hardware Libre brinda a los individuos la libertad de controlar su tecnología mientras comparten el conocimiento [42] [41].

4.5.1.1. Arduino

Arduino es una plataforma de creación de prototipos de código abierto. Las placas Arduino son capaces de leer entradas tales como: la luz en un sensor, un dedo sobre un botón y lo convierten en una salida que puede ser: la activación de un motor, encender un LED. Para ello se utiliza el lenguaje de programación de Arduino (basado en el cableado), y el software de Arduino (IDE), sobre la base de procesamiento [43].

El software de Arduino es fácil de usar para los principiantes, pero lo suficientemente flexible para los usuarios avanzados. Se ejecuta en Mac, Windows y Linux. Los diseñadores y arquitectos pueden construir prototipos interactivos, músicos y artistas lo utilizan para instalaciones y experimentar con nuevos instrumentos musicales [43].



Figura 7. Placa de Arduino [43].

4.5.1.2. Raspberry Pi

Raspberry Pi es un pequeño ordenador de bajo costo, tan pequeño como una tarjeta de crédito con un mínimo de consumo eléctrico. Es una placa que soporta varios componentes necesarios en un ordenador común. Es un pequeño ordenador, que puede ser utilizado por muchas de las cosas que un PC de escritorio hace, como hojas de cálculo, procesadores de texto y juegos. También reproduce vídeo de alta definición [44].

El Raspberry Pi usa mayoritariamente sistemas operativos basados en el núcleo Linux. Raspbian, una distribución derivada de Debian que está optimizada para el hardware de Raspberry Pi, en función del modelo, se disponen algunas opciones como la de puerto de conexión a Internet, pero siempre tendremos al menos un puerto de salida de vídeo HDMI y otro de tipo RCA, mini Jack de audio y un puerto USB 2.0 al que conectar un teclado y ratón [44].



Figura 8. Raspberry Pi [44].

4.5.1.3. Banana Pi

Banana Pi es un mini ordenador dotado de un hardware bastante potente, su funcionamiento es muy similar al de otros miniordenadores. Su sistema operativo se instala en una tarjeta SD, se puede instalar varios sistemas de manera que en una única tarjeta podemos tener Debian y Android convirtiendo a Banana Pi en un completo dispositivo para Smart TV. El conector Sata nos ayudará a conectar un disco duro fácilmente, por ejemplo, para utilizar Banana Pi como un servidor de almacenamiento [45].

Banana Pi posee algunas características respecto a la conectividad, pues en esta pequeña placa encontramos 2 USB 2.0, 2 conectores Micro USB. En cuanto a la salida de video, nos da un montón de posibilidades pues dispone de salida HDMI, RCA compuesto y conector LVDS. En cuanto al audio disponemos de salida de Jack 3.5mm stereo y entrada de micrófono [46].

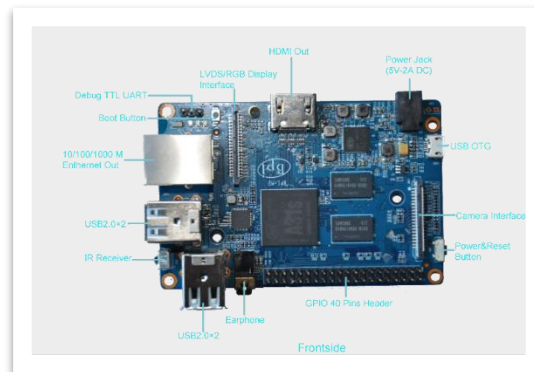


Figura 9. Banana Pi [45].

4.5.1.4. BeagleBoard

BeagleBoard es una computadora de placa única, sin ventilador y de bajo costo basada en los procesadores de bajo consumo de Texas Instruments que incluye un núcleo de la serie ARM Cortex-A con todas las capacidades de expansión de máquinas de escritorio, fue diseñado con el desarrollo de software de código abierto como una placa educativa que podría ser usada en colegios alrededor del mundo para enseñar las capacidades del Software y Hardware Libre [47] [48].

La BeagleBoard mide aproximadamente 75 por 75 mm y cuenta con todas las funciones de una computadora básica, La salida de video está provista de dos conexiones separadas de S-Video y HDMI. Una entrada para tarjetas SD/MMC con soporte SDIO, un

puerto USB, un conector serial RS-232, una conexión JTAG, y dos Jack de 3.5 mm para entrada y salida de audio, la placa usa arriba de 2 W de alimentación y puede ser alimentada por el conector USB o por una fuente externa de 5V. A causa del bajo consumo no son necesarios disipadores de calor [48].

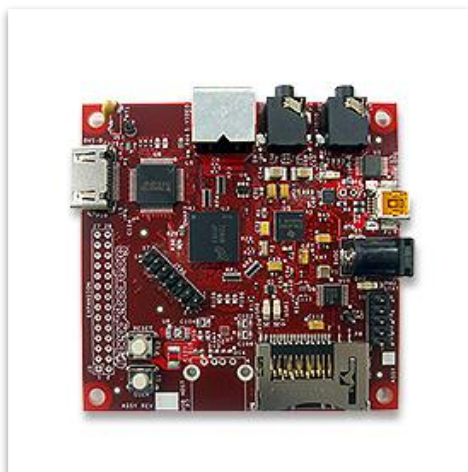


Figura 10. BeagleBoard [48].

4.5.1.5. Cuadro comparativo de Hardware Libre

De acuerdo a las características y especificaciones revisadas anteriormente se puede sacar las siguientes ventajas y desventajas:

Tabla 1. Tabla comparativa de Hardware Libre.

Nombre	Ventajas	Desventajas
Arduino	<ul style="list-style-type: none"> • Es un micro controlador • Documentación y Tutoriales • Open Source • Fácil de programar • Se puede hacer múltiples aplicaciones • Capaz de leer múltiples entradas • Se puede utilizar en diferentes plataformas • Fácil de conectarse con el mundo debido a sus entradas analógicas 	<ul style="list-style-type: none"> • Lento en ejecución de Instrucciones. • Poca memoria • Poca potencia del procesador • Necesita de un ordenador para poder cargar instrucciones

	<p>y digitales</p> <ul style="list-style-type: none"> • Fácil activación de entradas y salidas gracias a sus software 	
Raspberry Pi	<ul style="list-style-type: none"> • Ordenador de bajo costo • Se puede utilizar como una PC de escritorio. • Usa un sistema operativo basado en Linux • Documentación y Tutoriales • Gran potencia de cálculo • Permite reproducción de video • Puertos de conexión como USB, HDMI, Ethernet y otros. • Puede trabajar como un dispositivo autónomo, es un mini computador • Tiene miles de usuarios que dan soporte o publican software, guías de uso o instalación. • Raspberry Pi 3 cuenta con las siguientes características: <ul style="list-style-type: none"> ○ Procesador de 1,2 GHz con cuatro núcleos ARM Cortex-A53 ○ LAN inalámbrica 802.11 BGN y Bluetooth 4.1 (Classic Bluetooth y LE) ○ RAM de 1GB ○ CPU de 64 bits ○ 4 x puertos USB ○ 4 polos de salida estéreo y un puerto de vídeo 	<ul style="list-style-type: none"> • Complejo a la hora de hacer proyectos sencillos

	<p>compuesto</p> <ul style="list-style-type: none"> ○ Puerto HDMI ○ 10/100 Base T Ethernet socketbr ○ Puerto de la cámara de CSI para la conexión de la cámara Raspberry Pi ○ Puerto de pantalla DSI para la conexión de la pantalla táctil Raspberry Pi ○ Puerto Micro SD para cargar el sistema operativo y el almacenamiento de datos ○ Micro fuente de alimentación USB 	
Banana Pi	<ul style="list-style-type: none"> ● Puertos de conexión como USB, SATA, HDMI, Ethernet y otros. ● Se le puede instalar Sistemas Operativos basados en Linux ● Es un mini ordenador 	<ul style="list-style-type: none"> ● Por ser un dispositivo nuevo no existe mucho soporte ni documentación. ●
BeagleBoard	<ul style="list-style-type: none"> ● Se pueden instalar Sistemas operativos como Debian, Android, Ubuntu, entre otros. ● Es un minicomputador. ● Posee puertos de conexión como USB, SATA, HDMI, Ethernet y otros. ● Ofrece gran cantidad de documentación. 	<ul style="list-style-type: none"> ● Precio elevado

Una vez revisada cada una de las ventajas y desventajas de Hardware Libre se puede concluir que el Hardware con mejores características es Raspberry Pi, debido a su autonomía ya que puede ser utilizada como si fuera un PC de escritorio.

4.5.2. Software Libre

Software Libre son los programas informáticos, en los que prevalece la libertad para los usuarios de ejecutar, copiar, distribuir, estudiar, cambiar y modificar el software. De modo más preciso, un programa es software libre si los usuarios tienen las siguientes libertades [49]:

- Libertad de usar el programa, con cualquiera que sea su propósito.
- Libertad de estudiar cómo funciona el programa, y adaptarlo a lo que se requiera, se deberá tener acceso a código fuente, dado que sin él la tarea de incorporar cambios en un programa es extremadamente difícil.
- Libertad de redistribuir copias, ya sea de forma gratuita o pagada.
- Libertad de distribuir versiones modificadas del programa, de tal forma que la comunidad pueda aprovechar las mejoras introducidas.

Software libre no significa no comercial. Un programa libre debe estar disponible para uso comercial, desarrollo comercial y distribución comercial. El desarrollo comercial del software libre ha dejado de ser inusual; el software comercial libre es muy importante. [50]

4.5.2.1. Software para el procesamiento de Imágenes Digitales

➤ Java

Java es la base para prácticamente todos los tipos de aplicaciones, permite desarrollar, implementar y utilizar de forma eficaz interesantes aplicaciones y servicios. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes [51].



Figura 11. Java [51].

Java está diseñado para permitir el desarrollo de aplicaciones portátiles de elevado rendimiento para el más amplio rango de plataformas informáticas posible, java permite a los desarrolladores escribir software en una plataforma y ejecutar virtualmente en otra, crear programas para ejecutar en distintos exploradores web, desarrollar aplicaciones de servidor, combinar aplicaciones que utilizan lenguaje Java para crear aplicaciones o servicios con un gran nivel de personalización, escribir aplicaciones potentes y eficaces para teléfonos móviles y cualquier dispositivo electrónico [51].

Algunas de las características que presenta java frente a otro lenguaje de programación son las siguientes:

- Es un lenguaje simple con características muy útiles.
- Es orientado a objetos, trabajo con sus datos como objetos y con interfaces de esos objetos.
- Es distribuido, proporciona las librerías y herramientas para que lo programas puedan ser distribuidos.
- Es robusto, proporciona comprobación de punteros, límites de arrays, excepciones y verificación de byte-codes.
- Es seguro, con características como el casting implícito que previene el acceso ilegal a memoria.
- Es portable, posee una arquitectura independiente.
- Es interpretado, puede ejecutar directamente el código objeto.
- Puede implementar múltiples librerías entre ellas librerías de Visión Artificial.

➤ **C**

Es un lenguaje de medio nivel, controla directamente la circuitería del ordenador, pudiendo obtenerse con ellos la eficiencia máxima [52]. Es un lenguaje de propósito general, se ha utilizado para el desarrollo de diversas aplicaciones: sistemas operativos, hojas de cálculo, gestores de bases de datos, etc.

El lenguaje C posee un núcleo muy simple, una serie de muy pocas palabras reservadas, y pocos tipos nativos de datos, pero resulta un lenguaje extremadamente extensible a nuevas funcionalidades mediante definiciones de nuevos tipos de datos de usuario, incluso independiente de la arquitectura, y la suma de una gran cantidad de funciones de biblioteca estándar y bibliotecas especializadas en diferentes tareas [52].

Es un lenguaje portable, es decir, es independiente del hardware. Los programas escritos en C son fácilmente transportables a otros sistemas. Algunas características de C se describen a continuación:

- Es un lenguaje potente y eficiente, permitiendo obtener programas rápidos y compactos.
- Requiere una cierta experiencia para poder sacarle el máximo rendimiento.
- Sin disciplina es difícil mantener el control del programa.
- Proporciona un completo control de cuanto sucede en el interior del ordenador.
- Permite una amplia libertad de organización del trabajo.

➤ **C++**

C++ es un lenguaje imperativo orientado a objetos derivado del C, a C++ se han añadido nuevos tipos de datos, clases, plantillas, mecanismo de excepciones, sistema de espacios de nombres, sobrecarga de operadores, referencias, operadores para manejo de memoria persistente, y algunas utilidades adicionales de librería [52].

C++ es un lenguaje de propósito general, todo puede programarse con ellos, desde sistemas operativos y compiladores hasta aplicaciones de bases de datos y procesadores de texto, pasando por juegos, aplicaciones a medida, etc. Características de C++ [52]:

- Lenguaje de programación orientado a objetos.
- Lenguaje muy didáctico, con este lenguaje se puede aprender muchos otros lenguajes con gran facilidad, como C#, Java, Visual Basic, Javascript, PHP, entre otros.
- Es mucho más difícil de usar que otros lenguajes de programación.
- Es muy potente en lo que se refiere a creación de sistemas complejos, un lenguaje muy robusto.
- Actualmente, puede compilar y ejecutar código de C, ya viene con librerías para realizar esta labor.
- Es difícil programar bases de datos
- Tiene un conjunto completo de instrucciones de control.

➤ **Python**

Python es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a

servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad. Algunas de las principales características de Python son las siguientes [53]:

- Es multiplataforma, hay versiones disponibles para múltiples sistemas.
- Es interpretado, los programas se pueden ejecutar directamente.
- Es interactivo, dispone de un intérprete por línea de comandos en el que se pueden introducir sentencias.
- Es orientado a objetos, permite crear programas con componentes reutilizables.
- Permite trabajar con múltiples funciones y librerías.

➤ **MATLAB**

MATLAB es un lenguaje de programación desarrollado por The Mathworks, MATLAB nace como el resultado de tratar de satisfacer las necesidades computacionales de científicos, ingenieros y matemáticos, por lo que el desarrollo de MATLAB comprende un potente lenguaje de alto nivel y la incorporación de funciones gráficas más allá de los lenguajes de programación C y Fortran [54].

El nombre MATLAB proviene de “MATrix LABoratory” (Laboratorio de Matrices). MATLAB fue escrito originalmente para proporcionar un acceso sencillo al software matricial desarrollado por los proyectos LINPACK y EISPACK, que juntos representan lo más avanzado en programas de cálculo matricial. MATLAB es un sistema interactivo cuyo elemento básico de datos es una matriz que no requiere dimensionamiento [54].

En la industria, MATLAB se utiliza para investigación y para resolver problemas prácticos de ingeniería y matemáticas, con un gran énfasis en aplicaciones de control y procesamiento de señales. MATLAB también proporciona una serie de soluciones específicas denominadas TOOLBOXES. Estas son muy importantes para la mayoría de los usuarios de MATLAB y son conjuntos de funciones MATLAB que extienden el entorno MATLAB para resolver clases particulares de problemas como [54]:

- Procesamiento de señales
- Diseño de sistemas de control
- Simulación de sistemas dinámicos
- Identificación de sistemas

- Redes neuronales y otros.

➤ **Comparativo de Software para el Procesamiento de Imágenes**

Tabla 2. Tabla comparativa del Software utilizado en el procesamiento de imágenes.

Nombre	Ventajas	Desventajas
Java	<ul style="list-style-type: none"> • El JDK es libre. • Es independiente de la plataforma de desarrollo. • Existen dentro de su librería clase graficas como AWT y Swing. • Permite la programación orientada a objetos. • Permite acceder a base de datos con facilidad. • Es portable. • Puede implementar múltiples librerías. 	<ul style="list-style-type: none"> • La máquina virtual hace que la ejecución de aplicaciones sea lenta.
C	<ul style="list-style-type: none"> • Es un lenguaje potente y eficiente. • Acceso a memoria de bajo nivel mediante el uso de punteros. • Conjunto reducido de palabras reservadas. • Incorpora varias librerías. 	<ul style="list-style-type: none"> • No permite programación orientada a objetos. • En manejo de memoria puede resultar difícil si no se cuenta con la suficiente experiencia.
C++	<ul style="list-style-type: none"> • Es orientado a objetos. • Lenguaje potente y robusto. • Puede compilar y ejecutar código de C. • Permite la utilización de punteros, por consiguiente 	<ul style="list-style-type: none"> • Sin la suficiente experiencia es difícil el manejo de punteros. • Mayor control del Hardware por ende más complicado.

	<p>manejo de memoria.</p> <ul style="list-style-type: none"> • Posee todo el soporte de librerías de C. • Eficiente con respecto al hardware, se puede acceder directamente. • Estable. 	
Python	<ul style="list-style-type: none"> • Multiplataforma. • Código Abierto. • Interpretado. • Interactivo. • Orientado a objetos. • Permite el trabajo con múltiples librerías. • Código fácil de entender. 	<ul style="list-style-type: none"> • Configuración difícil. • Lentitud por ser interpretado.
Matlab	<ul style="list-style-type: none"> • Amplio soporte de funciones ya desarrolladas. • Integración con dispositivos Hardware. • Es fácil de aprender. • Soporte Matemático. • Alta precisión. 	<ul style="list-style-type: none"> • Se requiere una licencia. • El ordenador donde se ejecute necesita MCR para que los archivos Matlab funcionen correctamente. •

Dado que en este proyecto una de las principales premisas es la rapidez de ejecución, la elección es el lenguaje C++; dado que otros lenguajes por ejemplos Java necesita una máquina virtual C++ no la necesita, permite trabajar tanto a alto nivel como a bajo nivel de una manera óptima, a pesar de ser complicado la gran cantidad de información y la mayor experiencia en este lenguaje hacen que sea la opción elegida.

4.5.2.2. Librerías para el procesamiento de Imágenes Digitales

➤ OpenCV(Open Source Computer Vision)

OpenCV es una librería de funciones de programación dirigida principalmente a la visión por ordenador en tiempo real. OpenCV es liberado bajo una licencia BSD (Berkeley

Software Distribution) y por lo tanto está libre para uso académico y comercial. Tiene interfaces de C ++, C, Python y Java siendo compatible con Windows, Linux, Mac OS, iOS y Android. OpenCV fue diseñado para la eficiencia computacional y con un fuerte enfoque en aplicaciones en tiempo real. Escrito en C / C ++, la biblioteca puede tomar ventaja de procesamiento multi-núcleo. Se activa con OpenCL, se puede aprovechar la aceleración de hardware de la plataforma de computación heterogénea subyacente [55].

Entre sus principales aplicaciones tenemos:

- Seguimiento del movimiento.
- Detección de características 2D y 3D.
- Segmentación y reconocimiento.
- Identificación de objetos.
- Visión estéreo y calibración.
- Estimación de movimiento.
- Reconocimiento facial y de gestos.
- Estructura del movimiento.

OpenCV tiene una estructura modular, lo que significa que el paquete incluye varias bibliotecas compartidas o estáticas. Los siguientes módulos están disponibles:

- Funcionalidad básica. Un módulo compacto que define las estructuras de datos básicas, incluyendo funciones básicas utilizadas por todos los demás módulos.
- El procesamiento de imágenes. Un módulo de procesamiento de imágenes que incluye el filtrado de imágenes no lineal y lineal, transformaciones geométricas, la conversión de espacio de color, histogramas, etc.
- Video. Un módulo de análisis de vídeo que incluye la estimación de movimiento, la sustracción del fondo, y los algoritmos de seguimiento de objetos.
- Calib3d. Consta de múltiples algoritmos de visión geometría, la calibración individual de la cámara, y los elementos de la reconstrucción 3D.
- Features2d. Tiene algoritmos detectores de características sobresalientes, descriptores y comparadores de descriptores.
- Objdetect. Detección de objetos e instancias de las clases predefinidas (por ejemplo, caras, ojos, tazas, personas, vehículos, etc.).
- Highgui. Tiene una interfaz fácil de usar para cubrir las necesidades de interfaz de usuario sencilla.

- Videoio. Tiene una interfaz fácil de usar para la captura de vídeo.

➤ **JavaVis**

JavaVis es una Librería desarrollada en la Universidad de Alicante, consta de varios algoritmos clásicos de Visión Artificial, como el algoritmo Canny de detección de aristas. En esta librería se integran clases y métodos para dar soporte a los algoritmos que se deseen implementar, con el uso de clases de soporte básicas para: definir el tipo de datos de imagen, manejar lectura y escritura de ficheros, usar una interfaz gráfica que nos permita ver el resultado de aplicar determinados algoritmos. Cuenta con algunas funciones las cuales se detallan a continuación [56]:

- Conversión: Convierten los formatos de las imágenes.
- Ajustes: Realizan ajustes de brillo, contraste e intensidad en la imagen.
- Suavizado: Realizan operaciones de suavizado para reducir ruido y otros efectos que pueden estar presentes en la imagen.
- Realce: Efectúan operaciones de realce sobre la imagen para compensar efectos tales como sombras y falsos reflejos.
- Geometría: Añaden puntos, segmentos a la imagen y convierten una imagen de tipo geométrico a escala de grises.
- Contorno: Realizan detección de contornos.
- Gradientes: Calculan el gradiente de la imagen.

➤ **SciPy**

Scipy es una librería de herramientas numéricas para Python que se distribuye libremente, confiere al lenguaje general Python capacidades de cálculo numérico de gran capacidad y contiene módulos para optimización, álgebra lineal, integración, interpolación, funciones especiales, procesamiento de señales y de imagen [57].

Su organización se estructura en subpaquetes, que se pueden considerar especializados en dominios científicos determinados. Entre los principales tenemos los siguientes [57]:

- Clustering package
- Constants
- Special functions
- Statistical functions

- Image Array Manipulation and Convolutio
- Input and output
- Linear algebra
- Maximum entropy models

➤ **Comparativo de Librerías de Visión Artificial**

Tabla 3. Tabla comparativa de las librerías utilizadas en Visión Artificial.

Nombre	Ventajas	Desventajas
OpenCV	<ul style="list-style-type: none"> • Librería dirigida a la Visión por Ordenador en tiempo real • Es liberado bajo licencia BSD • Tiene interfaces de C++, C, Python y Java • Es compatible con múltiples plataformas • Escrito en C/C++ • Tiene estructura modular • Contiene más de 500 funciones para el proceso de Visión por Computador 	
Java VIS	<ul style="list-style-type: none"> • Escrito en java • Cuenta con más de 60 algoritmos de Visión por Computador. • Integra varias clases y métodos • Cuenta con varias funciones para el tratamiento de imágenes • 	<ul style="list-style-type: none"> • Se limita a trabajar únicamente con Java
SciPy	<ul style="list-style-type: none"> • Es una librería de herramientas numéricas • Se distribuye libremente • Tiene módulos para despeñar diferentes trabajos 	<ul style="list-style-type: none"> • Se limita a trabajar únicamente con Python

	<ul style="list-style-type: none">• Es modular• Orientado a objetos	
--	--	--

No solo es importante elegir un buen lenguaje de programación sino también las librerías de visión por computador, anteriormente se eligió el lenguaje C++ para la programación, ahora, la librería que mejor se acopla a este lenguaje y la que mejor características presenta es OpenCV, esta librería provee un marco de desarrollo fácil de utilizar, altamente eficiente y además es escrita en C/C++.

4.5.3. Internet de las Cosas

El Internet de las cosas potencia objetos que antiguamente se conectaban mediante circuito cerrado, como comunicadores, cámaras, sensores, y demás, y les permite comunicarse globalmente mediante el uso de la red de redes, se utiliza hardware especializado que le permite no solo la conectividad a Internet, sino que además programa eventos específicos en función de las tareas que le sean dictadas remotamente [58].

Se trata de chips que cuentan con todas las herramientas necesarias para cumplir labores específicas. No hay un tipo específico de objetos conectados a Internet de las cosas, en lugar de eso se les puede clasificar como objetos que funcionan como sensores y objetos que realizan acciones [59].

A continuación se describen las diferentes áreas en las que el Internet de las Cosas está cogiendo mayor realce [58]:

- La industria de producción en masa: la maquinaria que se encarga de controlar los procesos de fabricación, robots ensambladores, sensores de temperatura, control de producción.
- Control de infraestructura urbana: control de semáforos, puentes, vías de tren, cámaras urbanas.
- Control ambiental: una de las áreas en las que está teniendo más éxito el Internet de las cosas, pues permite acceder desde prácticamente cualquier parte a información de sensores atmosféricos, meteorológicos, y sísmicos.

- Sector salud: cada vez más clínicas y hospitales alrededor del mundo confían en sistemas que les permiten al personal de salud monitorear activamente a los pacientes.

5. Materiales y Métodos

Los materiales y métodos empleados en el desarrollo del presente proyecto de fin de carrera se detallan a continuación:

5.1. Materiales

Para que el desarrollo del presente Trabajo de Titulación sea exitoso es necesario realizar una planificación y control en términos económicos de un conjunto de materiales, los cuales están ligados con las actividades que se desarrollaron en el proyecto.

5.1.1. Talento Humano

En la siguiente tabla se detallan el total de horas que se emplearon en el desarrollo del proyecto con sus respectivos costos:

Tabla 4. Descripción del Talento Humano.

TALENTO HUMANO			
Talento Humano	Tiempo(Horas)	(\$)Precio/Horas	(\$)Valor Total
Tutor	150	10.00	1500.00
Tesista	400 h	5.00	2000.00
Subtotal			3500.00

5.1.2. Servicios

Además de contar con el talento humano, también se utilizaron servicios básicos que fueron necesarios para el desarrollo del proyecto.

Tabla 5. Descripción de Servicios.

Servicio	Descripción	(\$) P. Unitario	(\$) Total
Internet	6 meses	20.00	120.00
Transporte	150 días	1.00	150.00
SUBTOTAL			270.00

5.1.3. Recursos Hardware y Software

Durante el desarrollo del Sistema se utilizaron recursos Hardware y Software.

Tabla 6. Descripción de Hardware y Software

RECURSOS HARDWARE			
Recursos Técnicos	Tiempo de utilización(horas)	(\$)Valor Unitario	(\$)Valor Total
Computadores	400	1.00	400.00
Cámara ip	—	200.00	200.00
Impresora	—	85.00	85.00
Raspberry Pi	—	90.00	90.00
Pi Camera	—	50.00	50.00
Subtotal			825.00

Tabla 7. Descripción de Recursos Software.

RECURSOS SOFTWARE			
Recursos Tecnológicos	Valor Unitario	Desarrollador	Costo Total
CodeIgniter	Gratuito	--	—
Enterprise Architect 3.6	Demo 30 días	SpaxSystems	—
Open Office V.3.0	Gratuito	Sun Microsystems	—
Grand Project	Gratuito	—	—
C++	Gratuito	—	—
Eclipse	Gratuito	—	—
Linux	Gratuito	—	—
Raspbian	Gratuito	—	—
OpenCV	Gratuito	—	—
Subtotal			\$ 00.00

5.1.4. Materiales de Oficina

Para el desarrollo del proyecto también se emplearon algunos materiales de oficina, en los que se destacan insumos de papelería, anillados, empastados, etc.

Tabla 8. Descripción de Materiales de Oficina.

Materiales de Oficina			
	Cantidad	(\$ P. Unitario)	(\$ Total)
Insumos de papelería	1000	0.005	5.00
Anillados	7	2.60	18.20
Empastados	4	40.00	160.00
SUBTOTAL			183.20

Tabla 9. Presupuesto Total.

	(\$ Total)
Talento Humano	3500.00
Servicios	270.00
Recursos Hardware y Software	825.00
Materiales de Oficina	183.20
SUBTOTAL	3228.20
Imprevistos (10%)	322.82
Total PFC	5101.02

5.2. Métodos

Los métodos utilizados para la presente investigación son los siguientes:

Método Deductivo: Este método permite conocer de manera específica cada uno de los problemas más comunes, partiendo del problema general. Es un método de razonamiento que consiste en tomar conclusiones generales para explicaciones particulares. El método se inicia con el análisis de los postulados, teoremas, leyes y principios de la aplicación universal y de comprobada validez, para aplicarlos a soluciones o hechos particulares.

Método Inductivo: Este método nos da la posibilidad de que a partir del análisis de cada uno de los problemas identificados, ejecutando una revisión del resultado obtenido con el método deductivo comprobando la funcionalidad de la solución dada a dichos problemas. Con este método se utiliza el razonamiento para obtener conclusiones que parten de hechos particulares aceptados como válidos, para llegar a conclusiones, cuya aplicación

sea de carácter general. El método se inicia con un estudio individual de los hechos y se formulan conclusiones universales que se postulan como leyes, principios o fundamentos de una teoría

5.3. Metodología de Desarrollo

5.3.1. Metodología ICONIX

Es una metodología de desarrollo de software, basada en la complejidad de análisis de la metodología RUP (Rational Unified Processes) y la practicidad para desarrollar de la metodología XP (Extreme Programming), unifica un conjunto de métodos de orientación a objetos, con el objetivo de abarcar todo el ciclo de vida de un proyecto. Se considera un enfoque minimalista, ya que comprende el conjunto mínimo de medidas que son necesarias para el proyecto de desarrollo [60].

ICONIX cuenta con tres características fundamentales [61]:

Iterativo e Incremental: Durante el desarrollo del modelo del dominio y la definición de los casos de uso se producen varias iteraciones. El ciclo de vida incremental consiste en desarrollar por partes el producto de manera que puedas integrarlas funcionalmente.

Trazabilidad: Cada paso que se realiza está definido por un requisito, se define la trazabilidad como la capacidad de seguir una relación entre los diferentes artefactos de software producidos.

Dinámica del UML: Ofrece un uso dinámico del UML porque utiliza algunos diagramas UML, sin exigir la utilización de todos, como en el caso de RUP (Rational Unified Process).

5.3.1.1. Fases de ICONIX

➤ Análisis de requisitos

Se deben analizar todos los requisitos que formaran parte del sistema y con estos construir el diagrama de clases, que representa las agrupaciones funcionales que estructuran el sistema en desarrollo, para esta fase se utilizan tres herramientas [58]:

- **Modelo de Dominio:** Esto se refiere a identificar objetos y cosas del mundo real que intervienen en el sistema.
- **Modelo de Casos de Uso:** Describe las acciones o el comportamiento que un usuario realiza dentro del sistema.

- Prototipo de Interfaz de Usuario: Implica la creación de modelos operativos del trabajo de un sistema, en el que analistas y clientes deben estar de acuerdo.

➤ **Análisis y Diseño preliminar**

En esta fase a partir de cada caso de uso se obtendrá su respectiva ficha, esta ficha estará formada por un nombre, una descripción, una precondición que debe cumplir antes de iniciarse, una pos condición que debe cumplir al terminar. Se deben describir los casos de uso, como un flujo principal de acciones, pudiendo contener los flujos alternativos y los flujos de excepción [59].

Debería usarse un estilo consistente que sea adecuado al contexto del proyecto. Realizar Diagrama de Robustez: Es un híbrido entre un Diagrama de Clases y un Diagrama de Actividades

El diagrama de Robustez se divide en:

- Objetos fronterizos: Usado por los actores para comunicarse con el sistema.
- Objetos entidad: Son objetos del modelo del dominio.
- Objetos de Control: Es la unión entre la interfaz y los objetos de entidad.
- Diagrama de Clases: Describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos

➤ **Diseño**

El diseño consiste en especificar el comportamiento del sistema a través del diagrama de secuencia, para cada caso de uso identificar los mensajes entre los diferentes objetos. Se debe terminar el modelo estático, añadiendo los detalles del diseño en el diagrama de clases [58] [59].

➤ **Implementación**

En esta fase es donde se escribe el código tal y como fue especificado en las fases anteriores y se planean las pruebas basándonos en los requisitos iniciales.

5.4. Técnicas

La recolección de datos es el procedimiento más usual en la investigación, y tiene que ver con las técnicas y herramientas utilizadas de las que se seleccionó la más conveniente a criterio tal como: la observación directa y bibliográfica.

Observación Directa: Esta técnica permitirá conocer más a fondo el tráfico peatonal y vehicular en las diferentes intersecciones viales de la ciudad de Loja.

Bibliográfica: Esta técnica permitirá acceder e identificar todos aquellos documentos que contienen la información pertinente para desarrollar el presente proyecto de fin de carrera.

6. Resultados

6.1. Análisis de requerimientos

6.1.1. Requerimientos Funcionales

El sistema permitirá:

Tabla 10. Requisitos Funcionales.

REF.	Descripción
REF001	Tener video en tiempo real a través de una cámara
REF002	Identificar y contar vehículos pequeños (automóviles)
REF003	Identificar y contar personas
REF004	Identificar y contar bicicletas
REF005	Identificar y contar motos
REF006	Identificar y contar sillas de rueda
REF007	Tener un clasificador de objetos que identifique y establezca la diferencia entre clases de objetos
REF008	Crear un Backup de la información a través de un servidor
REF009	Al usuario consultar estadísticas del clasificador de acuerdo a la fecha, hora y contadores a través de una interfaz gráfica.
REF010	Al usuario observar el incremento de los contadores de objetos en tiempo real.

6.1.2. Requerimientos no Funcionales

De la misma forma que los requerimientos funcionales, también existen los requerimientos no funcionales, en donde se dice que el sistema deberá:

Tabla 11. Requisitos no Funcionales.

RENF.	Descripción
RENF001	Poseer una interfaz donde se pueda visualizar los datos
RENF002	Ser de fácil uso para realizar las pruebas
RENF003	Grabar información en una Base de Datos "MySQL"
RENF004	Desplegar resultados en un Display
RENF005	Disminuir los falsos negativos y falsos positivos

RENF006	El sistema se va a ejecutar en un dispositivo de Hardware Libre (Raspberry Pi) con sistema operativo Raspbian.
----------------	--

6.1.3. Modelo del Dominio

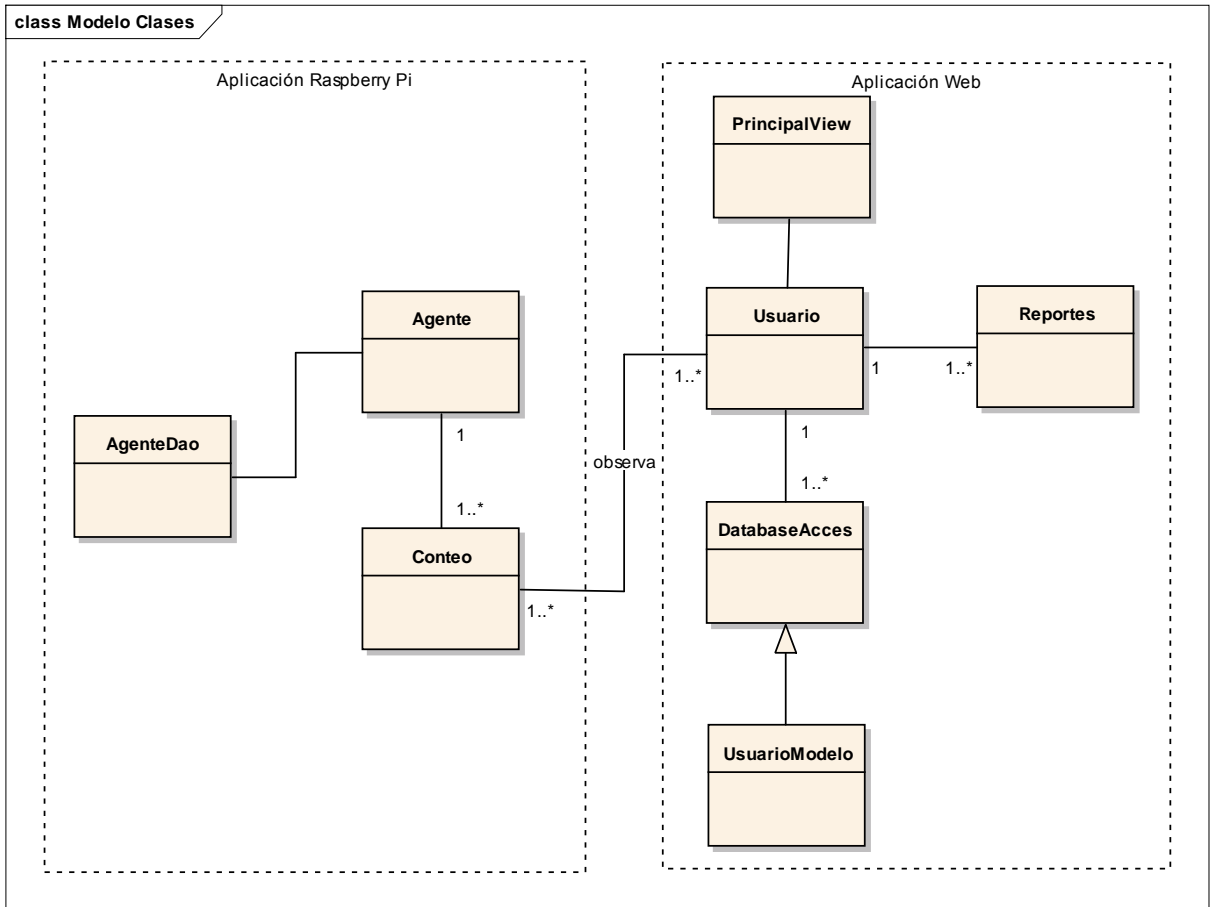


Figura 12. Modelo de Dominio.

6.1.4. Prototipado



Figura 13. Interfaz principal de la aplicación.

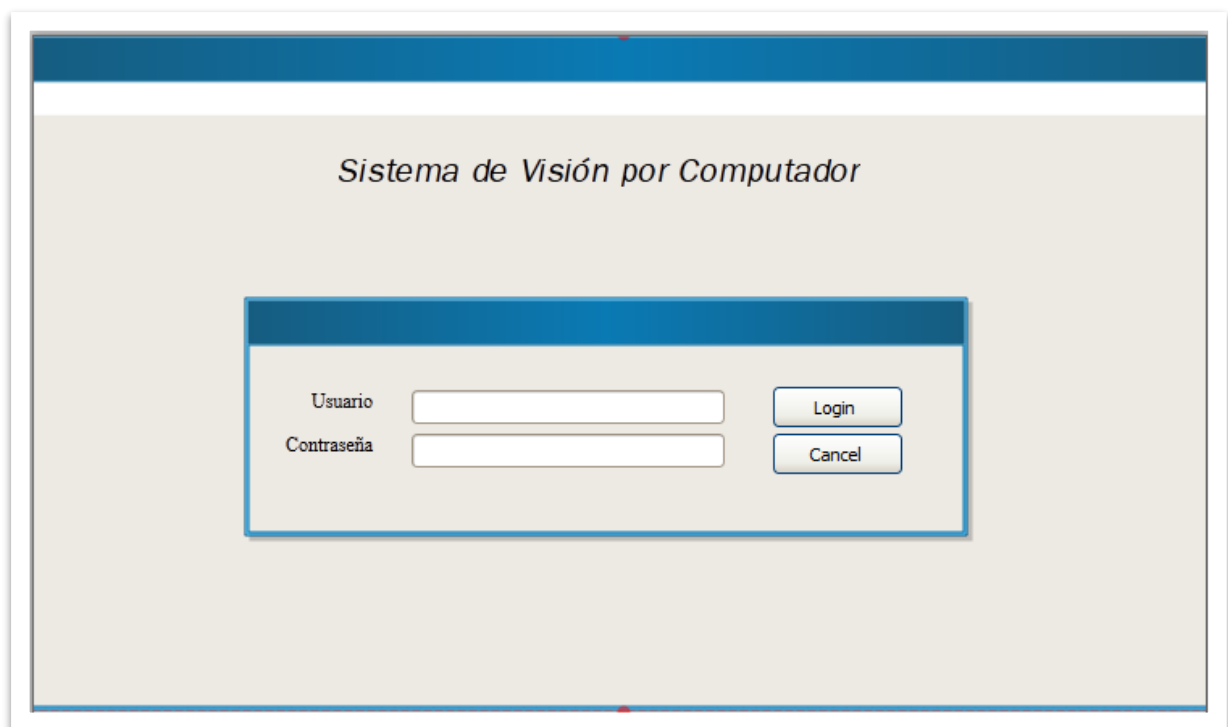


Figura 14. Interfaz para login de Usuario

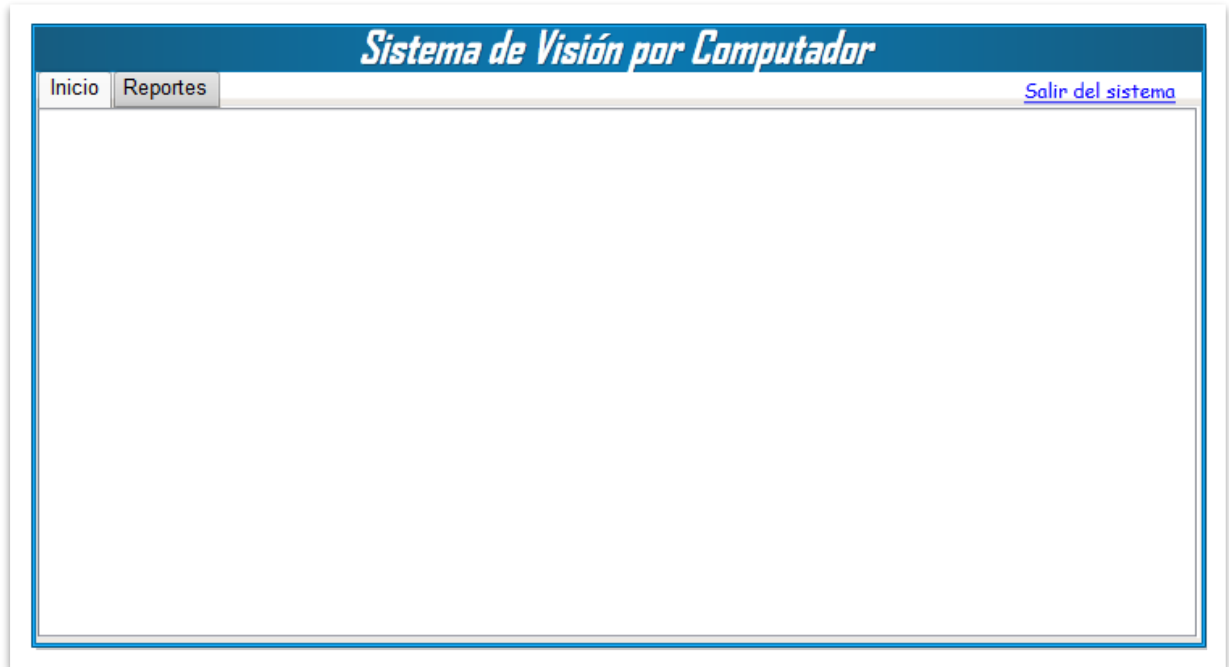


Figura 15. Pantalla de Inicio del Usuario

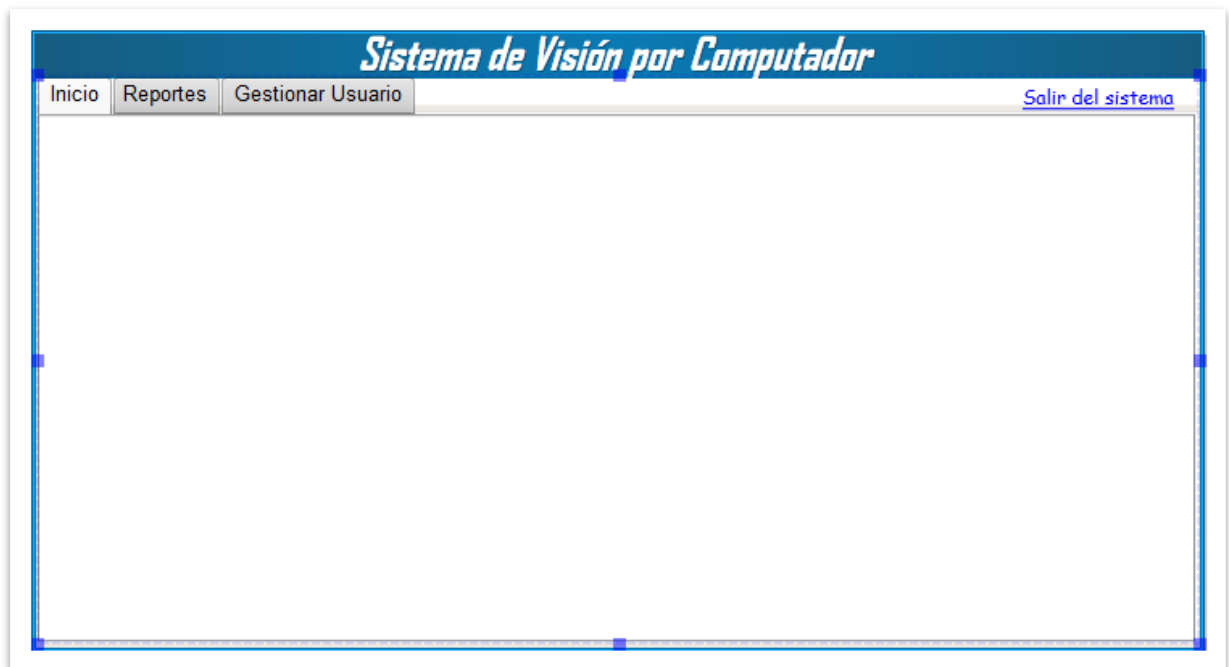


Figura 16. Pantalla de Inicio del Administrador.

Sistema de Visión por Computador

Inicio Reportes Estadísticas [Salir del sistema](#)

Consulta Base de Datos

Generación de reporte

Fecha: Inicio Fin
Hora: Inicio Fin

Reporte

ID	TIPO	HORA	FECHA	CONTADOR	UBICACIÓN
1	auto	12:30	12/12/2016	2	
2	auto	12:31	12/12/2016	3	
3	persona	12:31	12/12/2016	1	
4	bicicleta	12:30	12/12/2016	0	

Figura 17. Interfaz para consulta de Estadísticas de conteo de Objetos.

Sistema de Visión por Computador

Inicio Reportes Estadísticas **Gestionar Usuario** [Salir del sistema](#)

AGREGAR

Nombre	Apellido	Cedula	Us...	...	Acción
Usuario	Usuario	Usuario	Usu...	...	Edit/Eli...

Figura 18. Pantalla para Gestionar Usuario.

6.1.5. Modelo de Casos de Uso

Aquí se describen las diferentes actividades que debe realizar el Sistema.

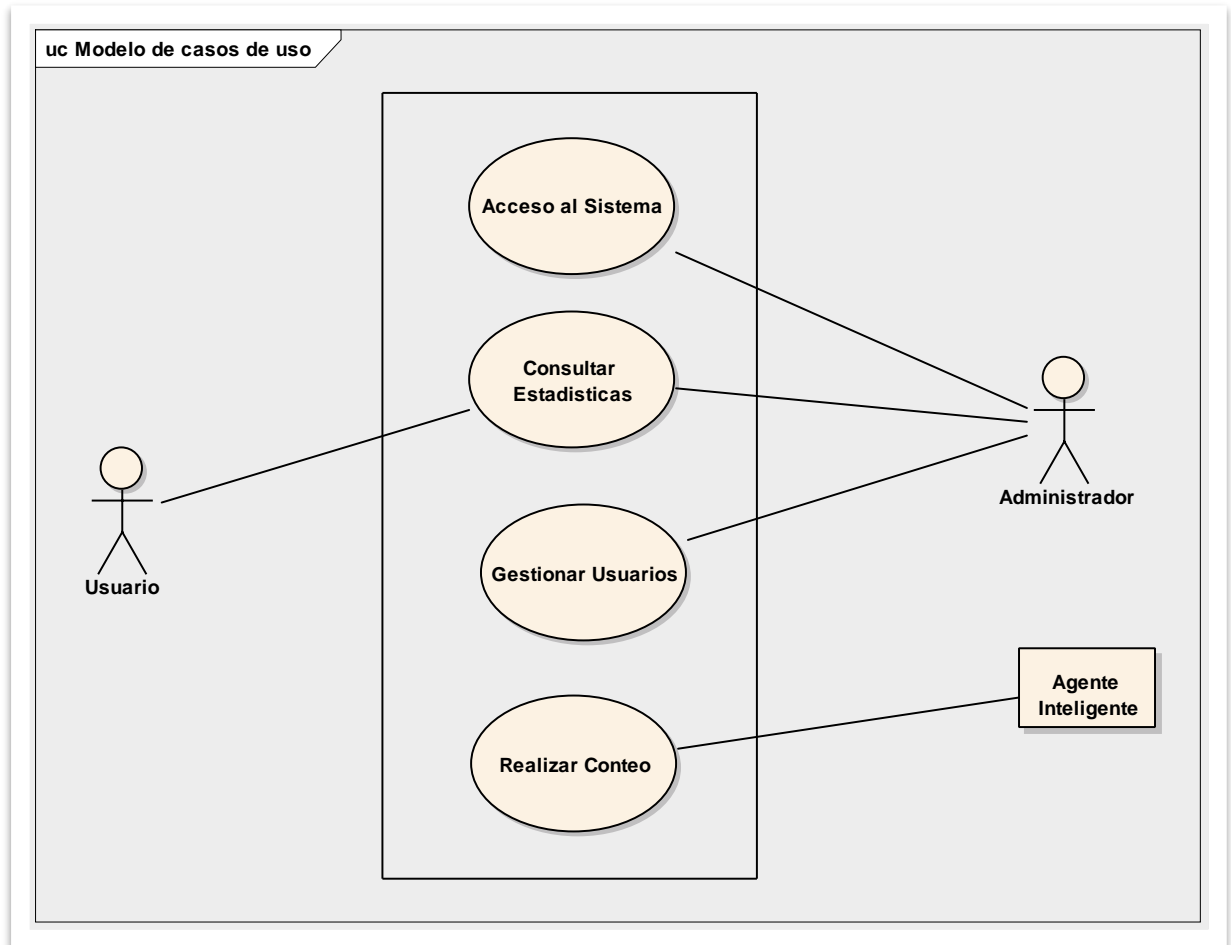


Figura 19. Modelo de Casos de Uso.

6.2. Análisis y Diseño Preliminar

6.2.1. Descripción de Casos de Uso

A partir de cada caso de uso se obtienen sus correspondientes fichas de caso de uso las cuales se describen a continuación.

6.2.1.1. Acceso al Sistema

Tabla 12. Caso de Uso de Acceso al Sistema.

Nombre: Acceso al Sistema			
Autor:		Fernando Carrión	
Fecha:		15 de septiembre del 2016	
Descripción: El sistema permitirá autenticarse al administrador cuando este solicite su ingreso.			
Actores: Administrador, Usuario			
Precondiciones: Contar con un nombre de usuario y contraseña respectiva.			
Flujo normal:	Paso	Acción	Respuesta
	1	El usuario ingresa los campos Usuario, Clave y elige el botón Acceder.	
	2		El sistema verifica que el usuario exista e inicia sesión del usuario autenticado habilitando las opciones propias del rol de usuario (usuario administrador o usuario normal) con el que haya ingresado.
	3	El caso de uso finaliza.	
Flujo Alternativo:	Paso	Acción	
	2		En caso de que existan campos vacíos, el sistema muestra un mensaje de error "Existen campos obligatorios vacíos", el caso de uso continúa en el paso 2 de la secuencia normal de eventos.

	3		En caso de ser incorrectos los datos ingresados el sistema muestra un mensaje de error “Datos Incorrectos”, el caso de uso continúa en el paso 2 de la secuencia normal de eventos.
Rendimiento:	Paso	Cota de tiempo	
	2	3 segundos	
Postcondición: El usuario ha ingresado correctamente al sistema.			
Importancia: Vital			

6.2.1.2. Gestionar Usuario

➤ Crear Usuario

Tabla 13. Caso de Uso para Crear Usuarios.

Nombre:	Crear Usuario		
Autor:	Fernando Carrión		
Fecha:	15 de septiembre del 2016		
Descripción:	El sistema permitirá crear diferentes usuarios, además en caso de ser necesario se podrá modificar la información de este.		
Actores:	Administrador		
Precondiciones:	El administrador debe haber ingresado correctamente al sistema.		
Flujo normal:	Paso	Acción	Respuesta
	1	El administrador elige la opción Administrar Usuarios, ubicada en el panel desplegable de la pantalla Principal.	

	2		El sistema muestra la pantalla Administrar Usuarios.
	3	El administrador elige la opción Agregar.	
	4		El sistema muestra la pantalla para agregar un nuevo usuario
	5	El administrador ingresa el nombre de usuario, contraseña, confirmación de contraseña, nombre, apellido, cedula, tipo de usuario (usuario normal o usuario administrador) y presiona el botón Guardar.	
	6		El sistema guarda la información del nuevo Usuario.
	7	El caso de uso finaliza.	
Flujo Alternativo:	Paso	Acción	Respuesta
	4		En caso de que existan campos vacíos, el sistema muestra un mensaje de error "Existen campos obligatorios vacíos", el caso de uso continúa en el paso 5 de la secuencia normal de eventos.
	5		Si el nombre de usuario o contraseña tiene menos de 8 caracteres, el sistema muestra un mensaje de error "El

			Nombre de Usuario y contraseña debe tener al menos 8 caracteres”, el caso de uso continúa en el paso 5 de la secuencia normal de eventos.
	6		En caso de que la contraseña y la confirmación de contraseña ingresadas no coincidan el sistema muestra un mensaje de error “Confirmación de contraseña incorrecta”, el caso de uso continúa en el paso 5 de la secuencia normal de eventos
	7		En caso de que el nombre de usuario ingresado ya exista, el sistema muestra un mensaje de error “Nombre de usuario ya existe”, el caso de uso continúa en el paso 5 de la secuencia normal de eventos.
Rendimiento:	Paso	Cota de tiempo	
	2	1 segundo	
	4	1 segundos	
	6	2 segundos	
Postcondición: La cuenta de usuario se ha creado correctamente.			
Importancia: Vital			

➤ **Modificar Usuario**

Tabla 14. Caso de Uso para Modificar Usuario.

Nombre: Modificar Usuario			
Autor:		Fernando Carrión	
Fecha:		15 de septiembre del 2016	
Descripción: El sistema permitirá modificar una cuenta de usuario.			
Actores: Administrador			
Precondiciones: Haber creado una cuenta de usuario del sistema de tipo administrador.			
Flujo normal:	Paso	Acción	Respuesta
	1	El administrador elige la opción Administrar Usuarios, ubicada en el panel desplegable de la pantalla Principal.	
	2		El sistema muestra la pantalla Administrar Usuarios.
	3	El Administrador elige el usuario a modificar y da clic en el botón Editar.	
	4		El Sistema presenta la pantalla de Modificar Usuario con los datos del Usuario seleccionado.
	5	El Administrador modifica los campos deseados (nombre de usuario, contraseña, nombre, apellido, cedula, tipo) y presiona el botón Modificar Usuario.	

	6		El sistema guarda la información modificada del Usuario.
	7	El caso de uso finaliza.	
Flujo Alternativo:	Paso	Acción	Respuesta
	6		En caso de que el usuario haya dejado algún campo vacío, el sistema mostrará un mensaje de error “Existen campos obligatorios vacíos”, el caso de uso continúa en el paso 5 de la secuencia normal.
	7		Si el nombre de usuario o contraseña tiene menos de 5 caracteres, el sistema muestra un mensaje de error “El Nombre de Usuario y contraseña debe tener al menos 8 caracteres”, el caso de uso continúa en el paso 5 de la secuencia normal de eventos.
	8		El sistema verifica que la contraseña y verificación de contraseña coincidan, en caso de que la contraseña y la confirmación de contraseña ingresadas no coincidan el sistema muestra un mensaje de error “Confirmación de contraseña incorrecta”, el caso de uso continúa en el paso 5 de la secuencia normal de

			eventos.
	9		En caso de que el nombre de usuario ingresado ya exista, el sistema muestra un mensaje de error "Nombre de usuario ya creado", el caso de uso continúa en el paso 5 de la secuencia normal de eventos.
Rendimiento:	Paso	Cota de tiempo	
	2	1 segundo	
	4	2 segundos	
Postcondición: La cuenta de usuario se ha modificado correctamente.			
Importancia: Vital			

➤ Eliminar Usuario

Tabla 15. Caso de Uso para Eliminar Usuario.

Nombre:	Eliminar Usuario		
Autor:	Fernando Carrión		
Fecha:	15 de septiembre del 2016		
Descripción:	El sistema permitirá Eliminar una cuenta de usuario.		
Actores:	Administrador		
Precondiciones:	Que el usuario esté registrado y que la información del usuario esté disponible.		
Flujo normal:	Paso	Acción	Respuesta
	1	El administrador elige la opción Administrar Usuarios, ubicada en el panel desplegable de la pantalla	

		Principal.	
	2		El sistema muestra la pantalla Administrar Usuarios.
	3	El Administrador elige el Usuario a Eliminar y presiona el botón Eliminar.	
	4		El sistema presenta un mensaje de confirmación "El Usuario será eliminado" para eliminar Usuario
	5	El Administrador confirma la eliminación del Usuario	
	6		El sistema pone al Usuario en estado Inactivo
	7	El caso de uso finaliza.	
Flujo Alternativo:	Paso	Acción	Respuesta
	5	El Administrador Cancela la eliminación del Usuario y el proceso continúa en el paso 7 del flujo normal.	.
Rendimiento:	Paso	Cota de tiempo	
	4	1 segundo	
	6	1 segundos	
Postcondición: La cuenta de usuario queda desactivada.			
Importancia: Vital			

6.2.1.3. Consultar Estadísticas de Conteo de Objetos

Tabla 16. Caso de Uso para Consultar Estadísticas de Conteo de Objetos.

Nombre: Consultar estadísticas de Conteo de Objetos			
Autor:			
Fecha:			
Descripción: El Sistema permite consultar estadísticas de conteo de objetos.			
Actores: Usuario y Administrador			
Precondiciones: El Usuario o Administrador debe haber ingresado al sistema y la información exista en la Base de Datos			
Flujo normal:	Paso	Acción	Respuesta
	1	El Usuario elige la opción Reportes, ubicada en el panel de la pantalla principal.	
	2		El sistema muestra la pantalla para Generar Reportes.
	3	El usuario ingresa los parámetros correspondientes (hora y fecha inicial, hora y fecha final, sitio donde está ubicada la cámara) y presiona el botón Generar Reporte.	
	4		El sistema muestra los resultados de conteo de objetos: personas, bicicletas, motos, sillas de ruedas y vehículos pequeños.
	5	El caso de uso finaliza.	

Flujo Alternativo:	Paso	Acción	Respuesta
	4		En caso de estar vacíos el sistema presentará el mensaje “campos obligatorios vacíos”, el caso de uso continúa en el paso 3 de la secuencia normal.
	5		En caso de que los datos sean incorrectos el sistema pedirá que vuelva a ingresar los datos. “Los datos ingresados son incorrectos” el caso de uso continúa en el paso 3 de la secuencia normal.
Rendimiento:	Paso	Cota de tiempo	
	4	2 segundos	
Postcondición: El reporte estadístico de conteo de objetos se ha generado correctamente.			
Importancia: Vital			

6.2.1.4. Realizar Conteo de Objetos

Tabla 17. Caso de Uso para Realizar Conteo de Objetos.

Nombre:	Realizar Conteo de objetos (personas, bicicletas, motos, sillas de ruedas y vehículos pequeños)
Autor:	Fernando Carrión
Fecha:	15 de septiembre del 2016
Descripción:	El sistema deberá detectar y contar personas, bicicletas, motos, sillas de ruedas y vehículos pequeños.
Actores:	Agente Inteligente

Precondiciones:		
Que la cámara esté disponible, que haya red de comunicación.		
Flujo normal:	Paso	Acción
	1	El Agente Inteligente capta vídeo en tiempo real, reconoce personas, bicicletas, motos, sillas de ruedas y vehículos pequeños, realiza el respectivo conteo de cada tipo de objeto detectado y envía los datos de los objetos detectados a una Base de Datos remota.
	2	El caso de uso finaliza.
Flujo Alternativo:	Paso	Acción
	2	El Agente Inteligente comprueba que la red de comunicación esté disponible, en caso de no estar disponible presenta un mensaje de error "Red no disponible", el caso de uso continúa en el paso 2 del flujo normal.
	3	El Agente Inteligente comprueba que la cámara esté disponible, en caso de no estar disponible presenta un mensaje de error "Cámara no disponible", el caso de uso continúa en el paso 2 del flujo normal.
Rendimiento:	Paso	Cota de tiempo
Postcondición: Los datos de los objetos detectados y contados quedan guardados en una Base de Datos.		
Importancia: Vital		

6.2.2. Diagrama de Robustez

6.2.2.1. Caso de Uso: Acceso al Sistema

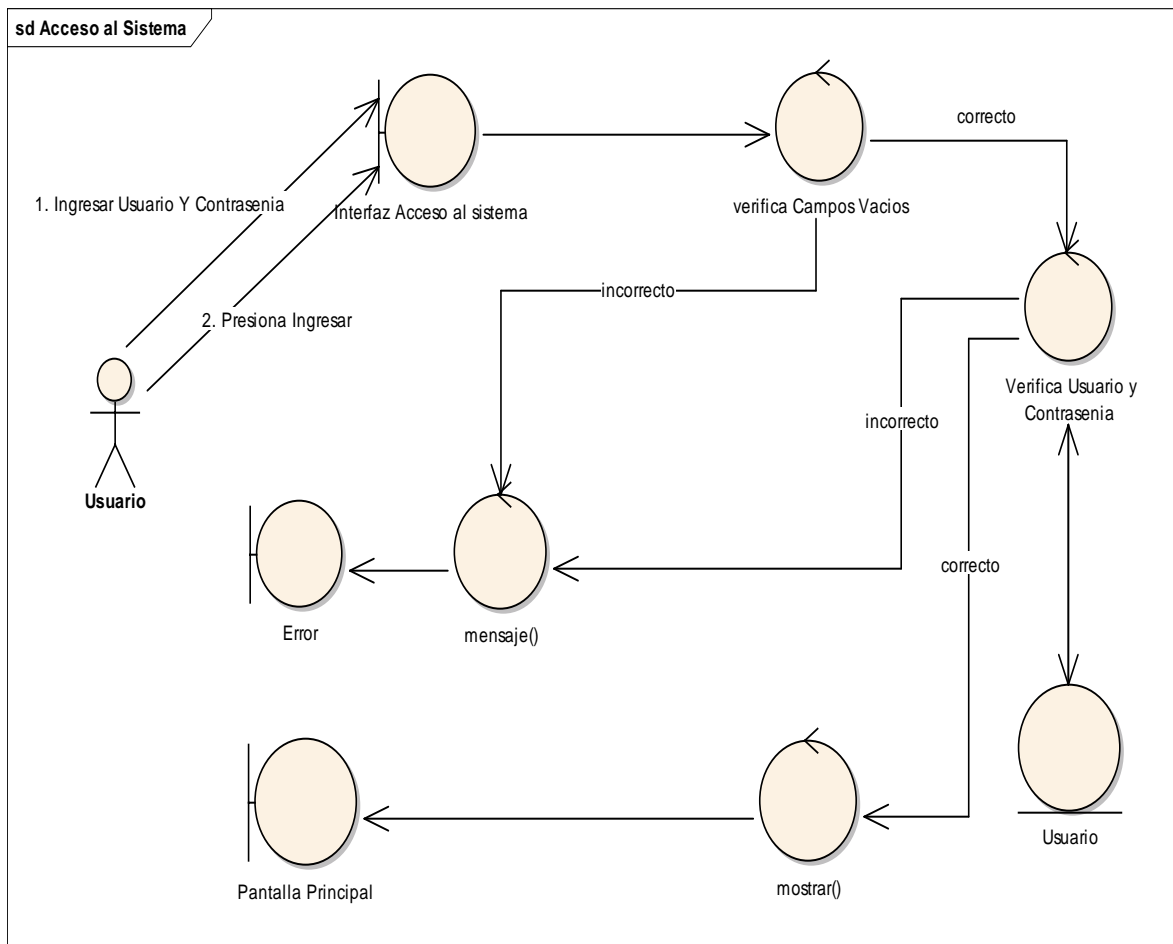


Figura 20. Diagrama de Robustez para Acceso al Sistema.

6.2.2.2. Caso de Uso: Gestionar Usuarios

➤ Crear Usuario

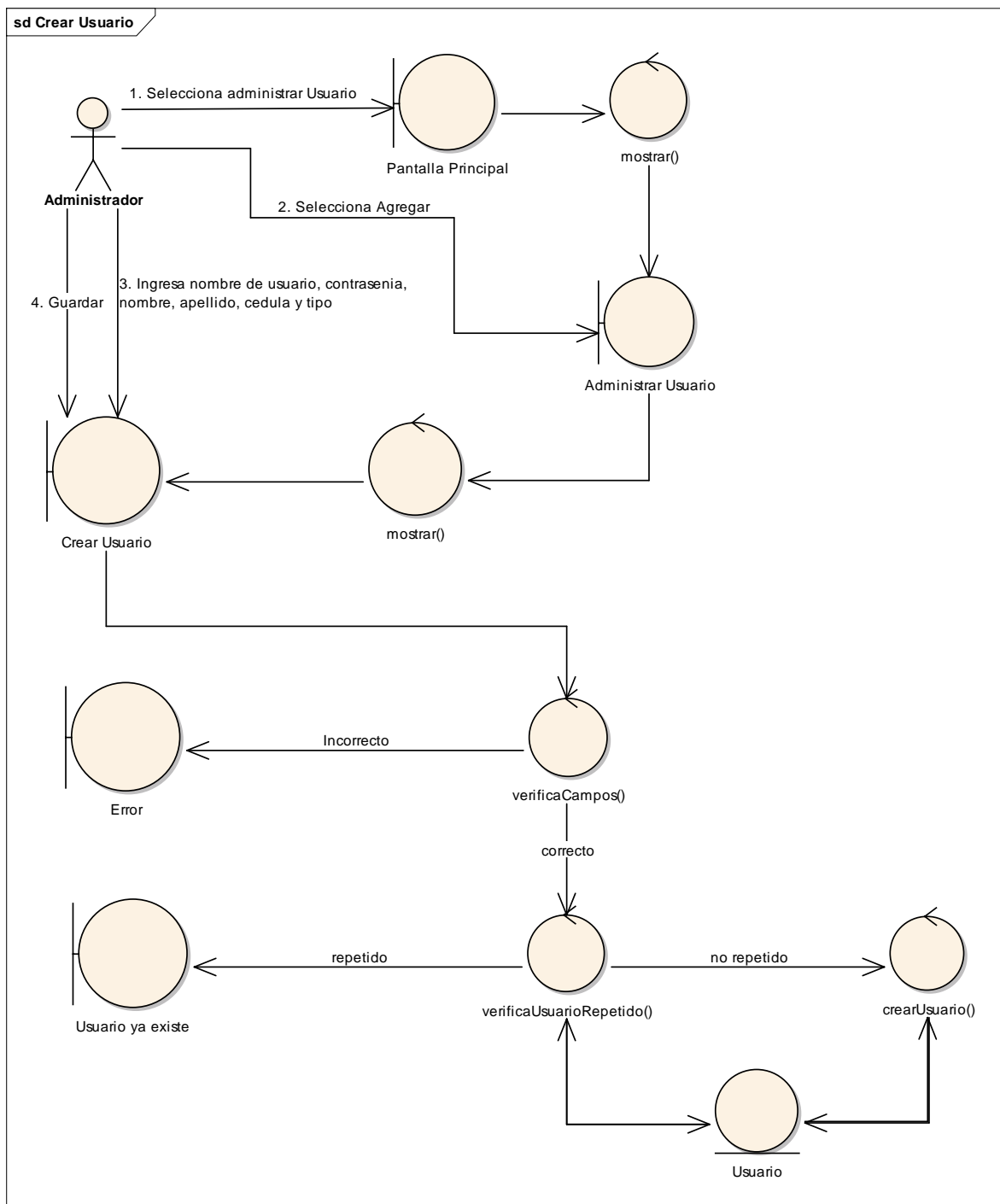


Figura 21. Diagrama de Robustez para Crear Usuario.

➤ **Modificar Usuario**

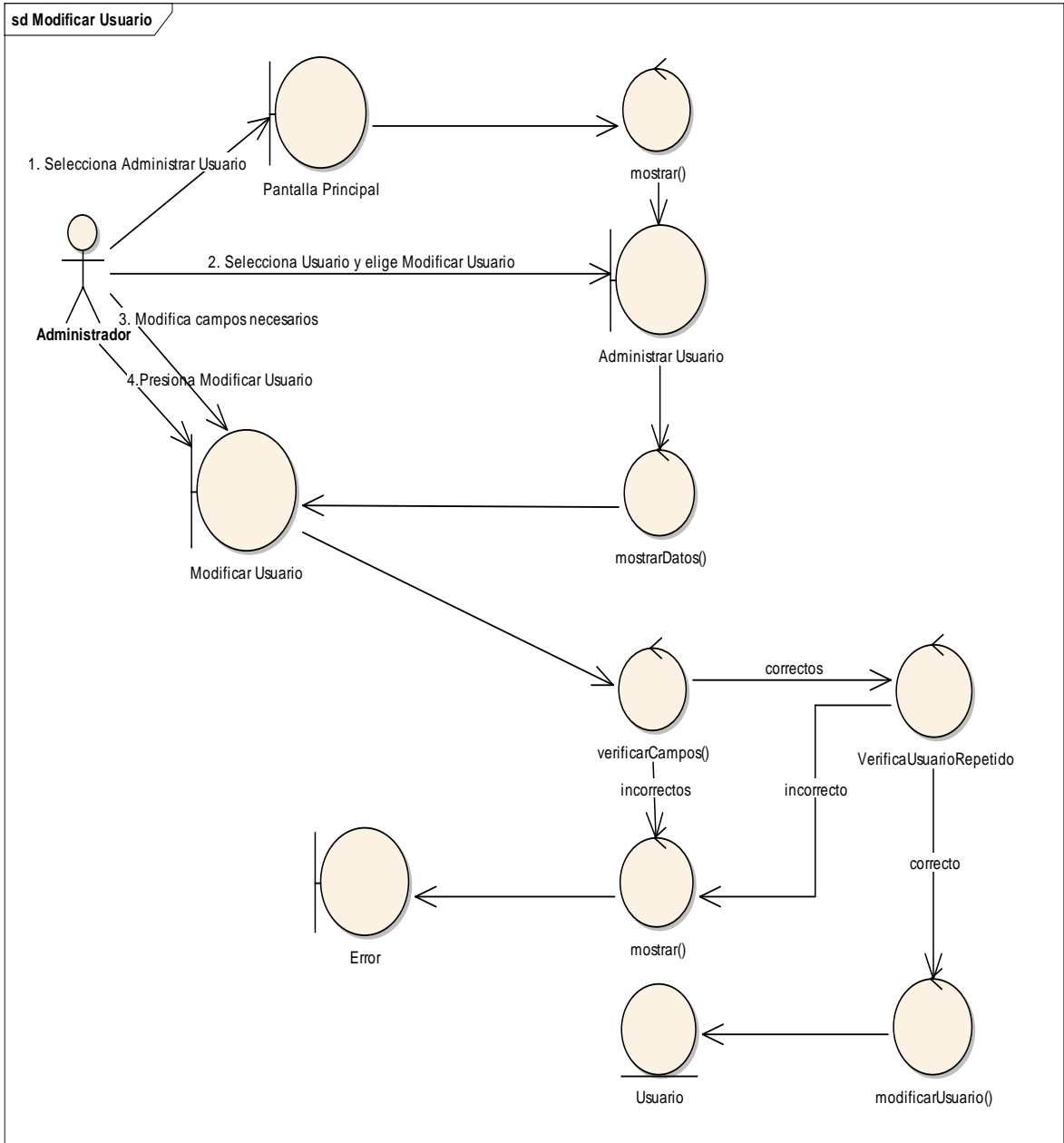


Figura 22. Diagrama de Robustez para Modificar Usuario.

➤ Eliminar Usuario

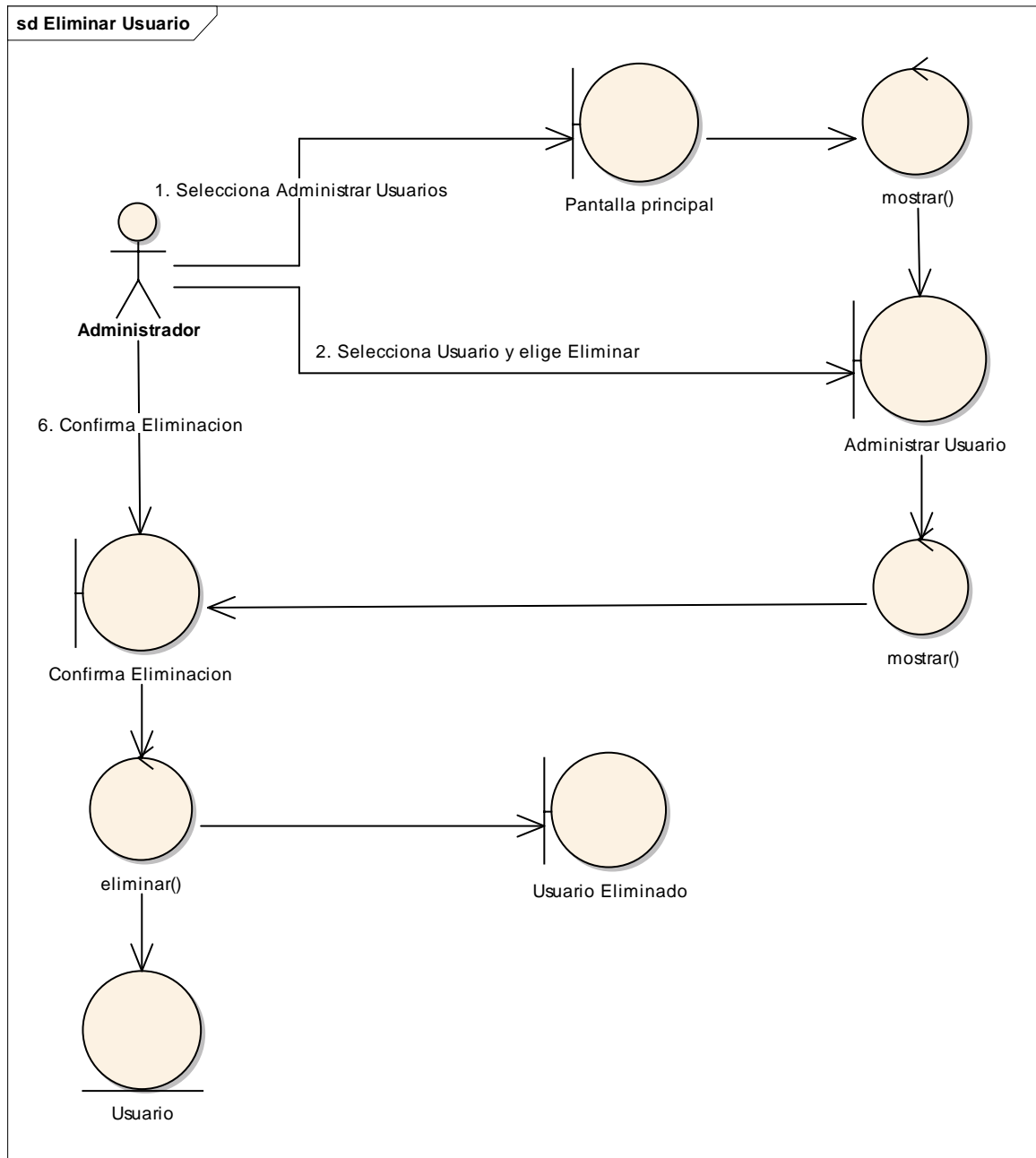


Figura 23. Diagrama de Robustez para Eliminar Usuario.

6.2.2.3. Consultar Estadísticas de Conteo de Objetos

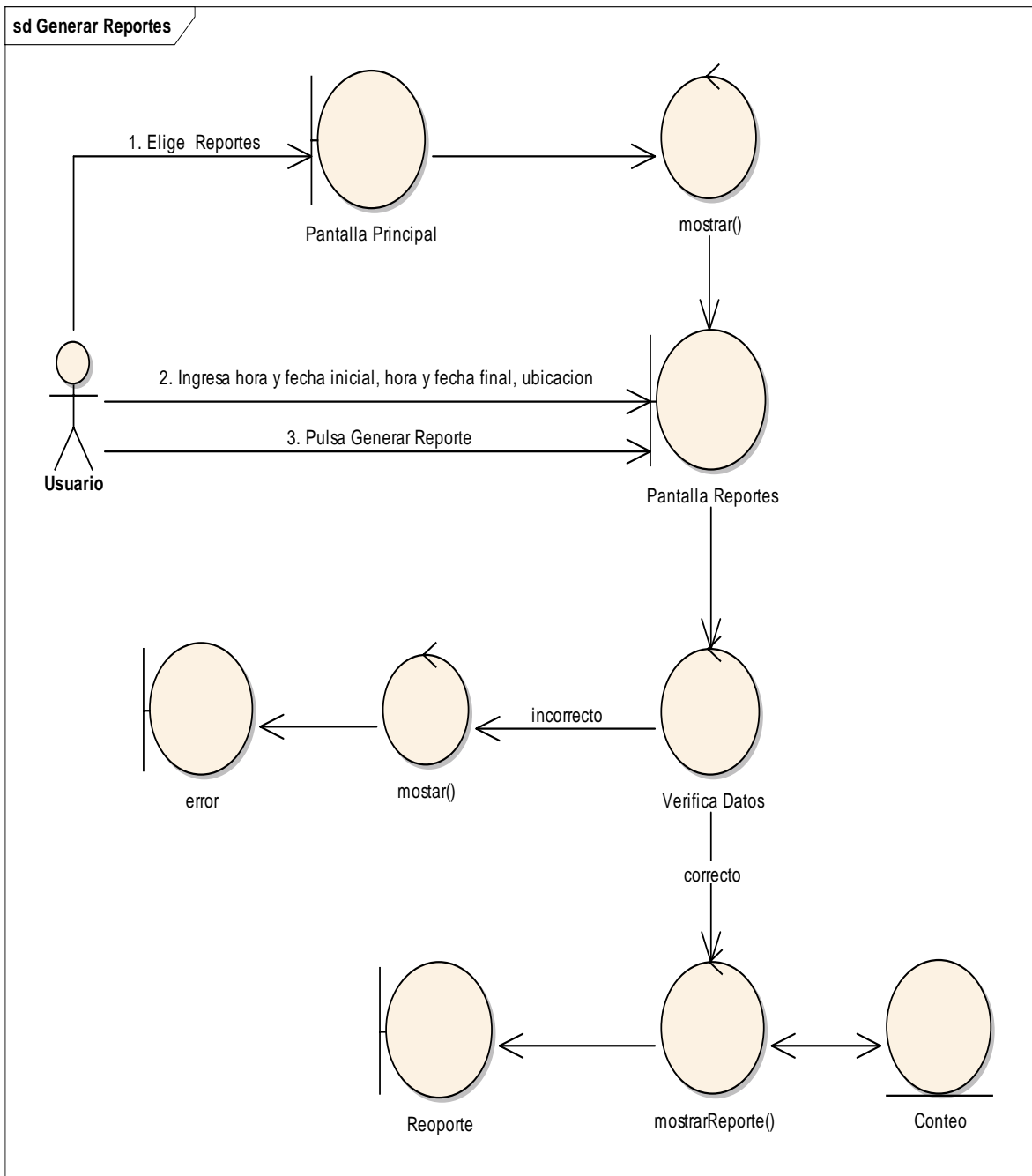


Figura 24. Diagrama de Robustez para Consultar Estadísticas de Conteo de Objetos.

6.2.2.4. Realizar Conteo de Objetos

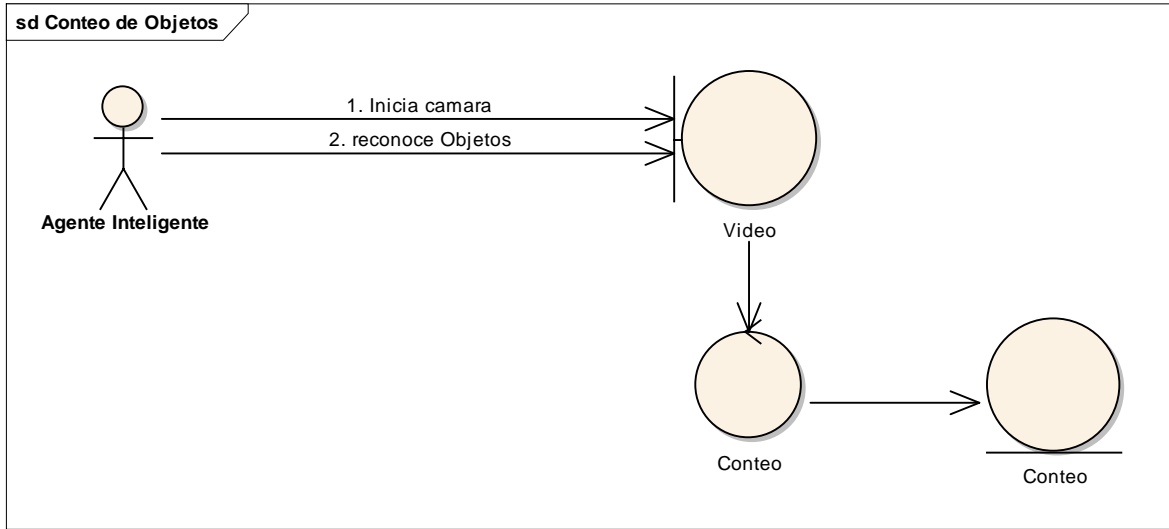


Figura 25. Diagrama de Robustez para Realizar Conteo de Objetos.

6.3. Diseño

6.3.1. Diagrama de Secuencia

6.3.1.1. Acceso al Sistema

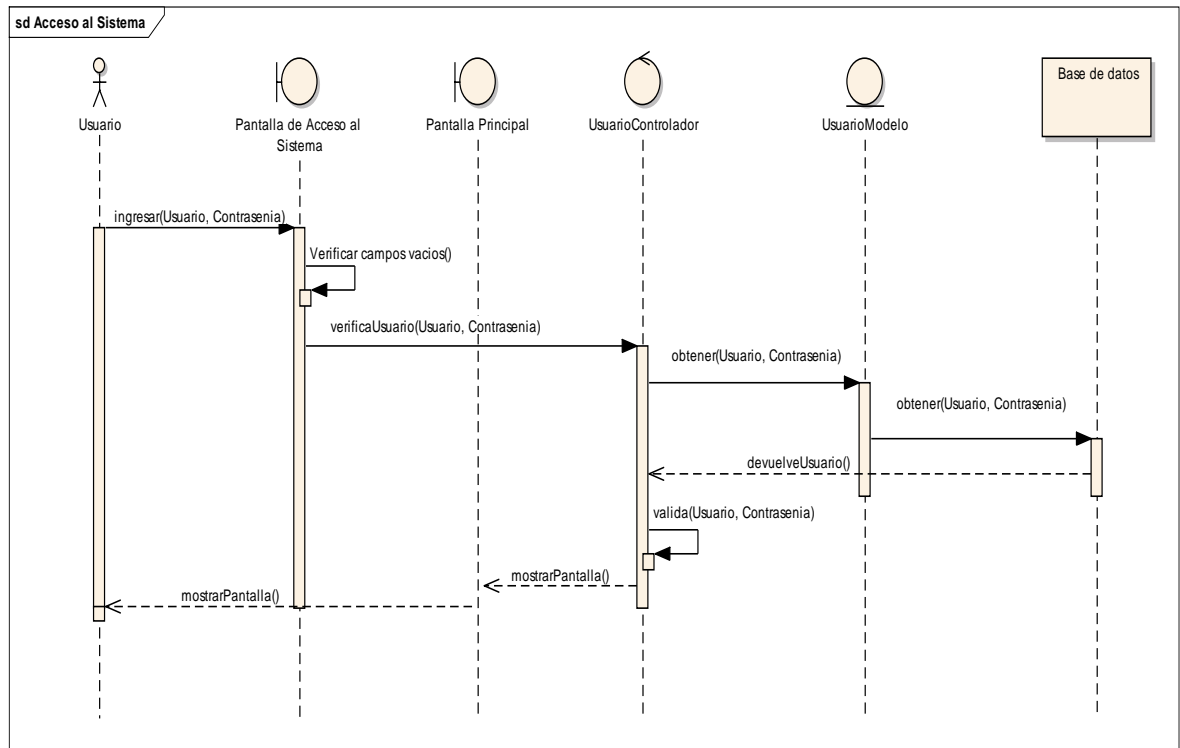


Figura 26. Diagrama de Secuencia para Acceso al Sistema.

6.3.1.2. Caso de Uso: Gestionar Usuario

➤ Crear Usuario

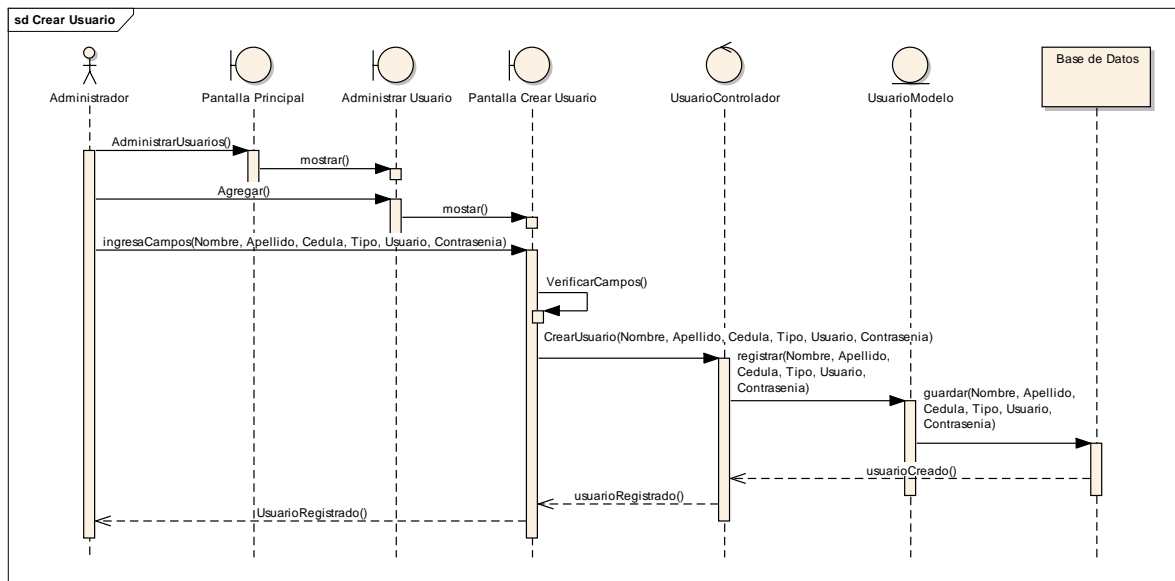


Figura 27. Diagrama de Secuencia para Crear Usuario.

➤ Modificar Usuario

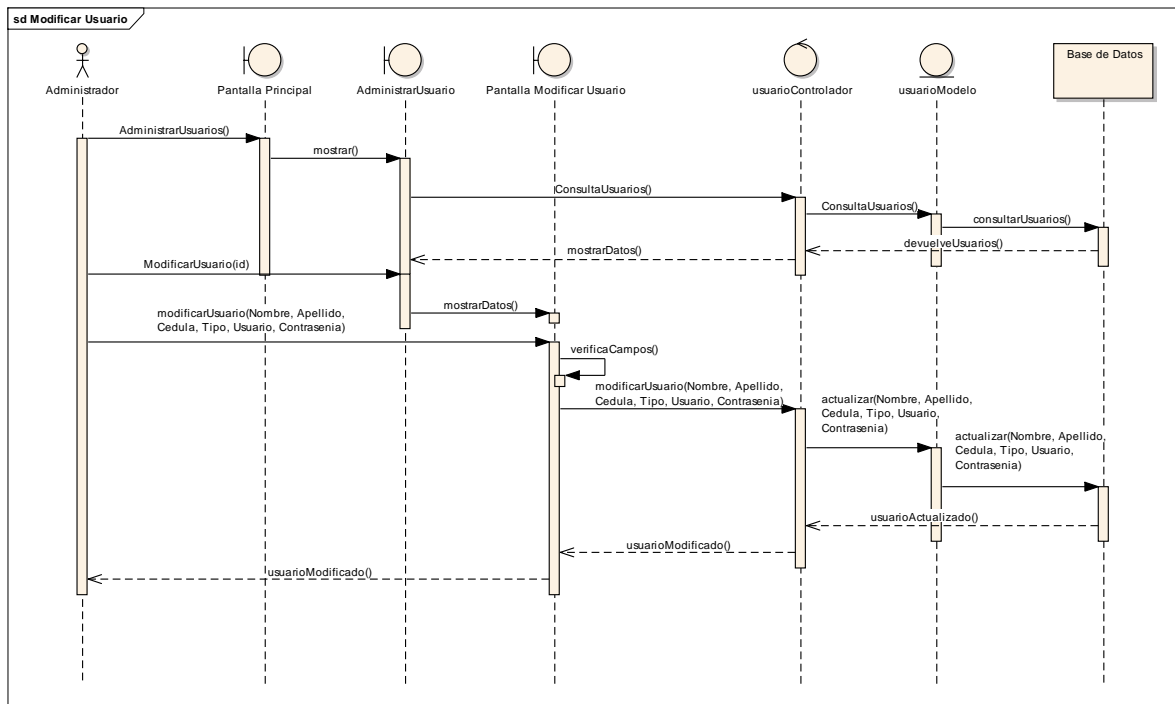


Figura 28. Diagrama de Secuencia para Modificar Usuario.

➤ Eliminar Usuario

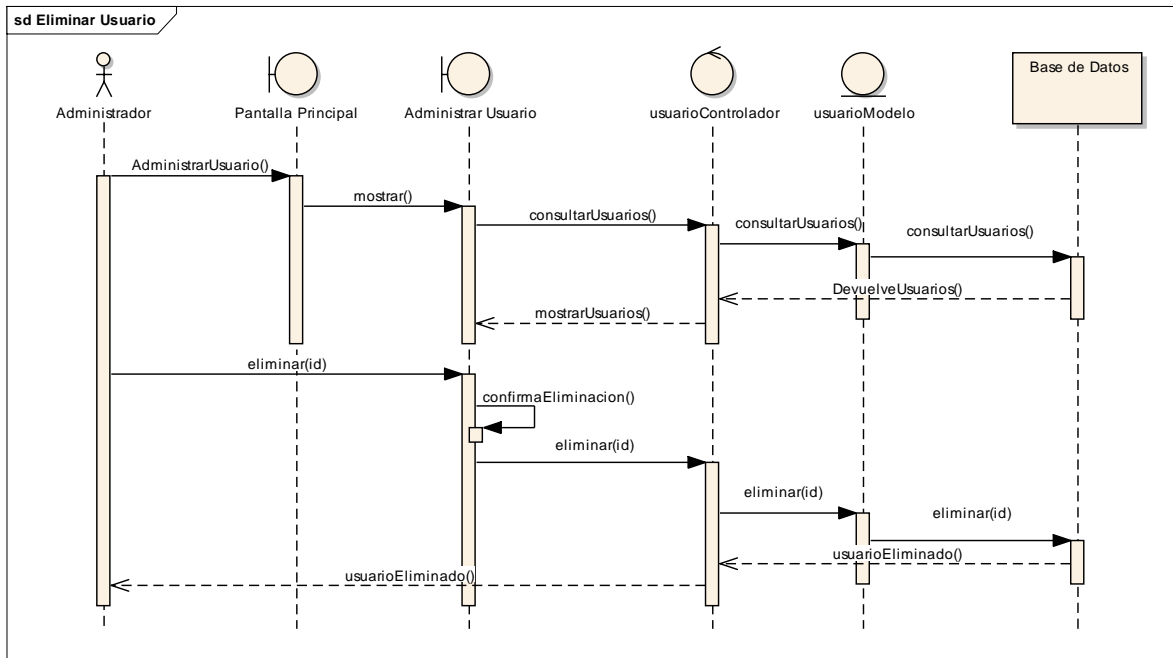


Figura 29. Diagrama de Secuencia para Eliminar Usuario.

6.3.1.3. Consultar Estadísticas de Conteo de Objetos

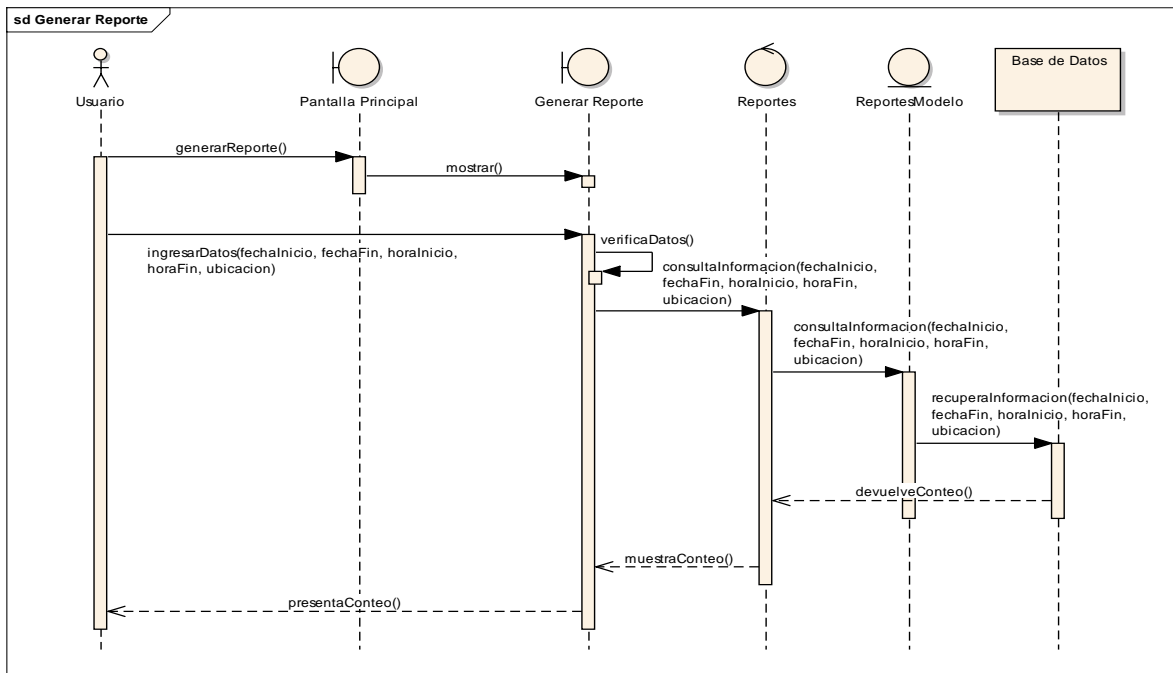


Figura 30. Diagrama de Secuencia para Consultar estadísticas de Conteo de Objetos.

6.3.1.4. Realizar Conteo de Objetos

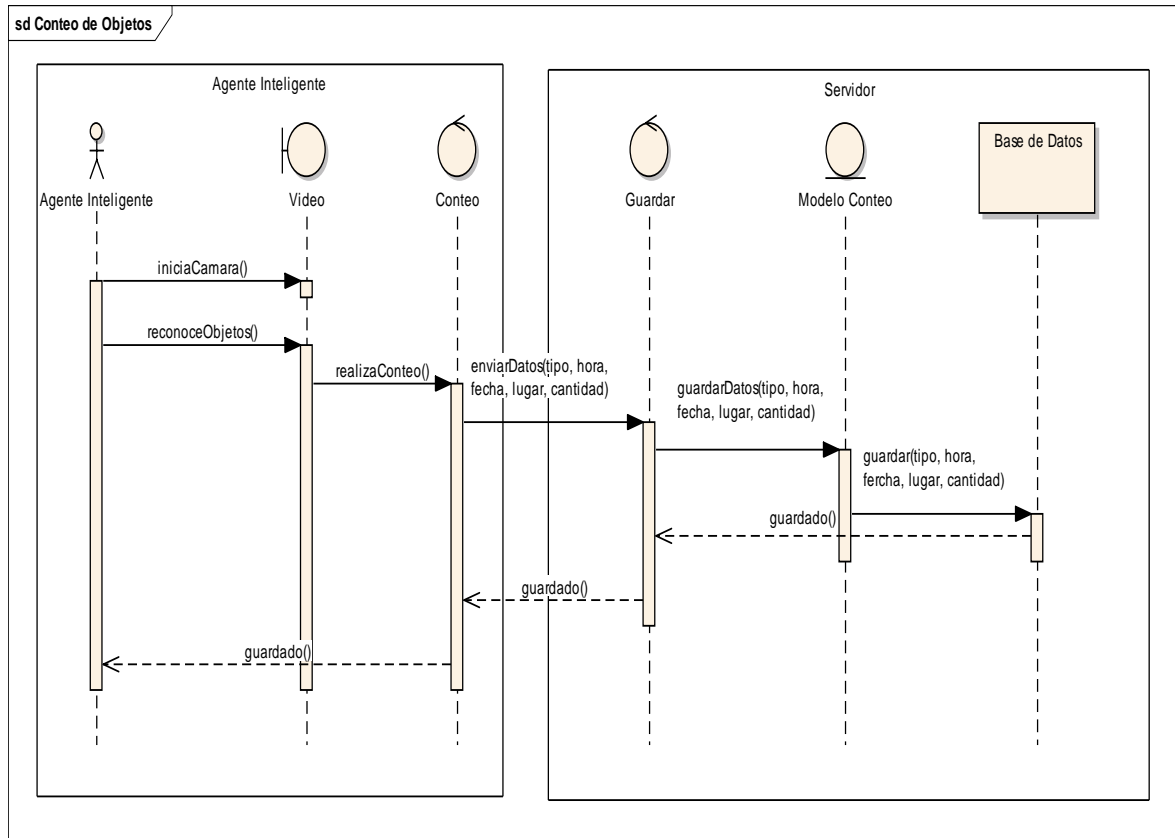


Figura 31. Diagrama de Secuencia para Realizar Conteo de Objetos.

6.3.2. Diagrama de Clases

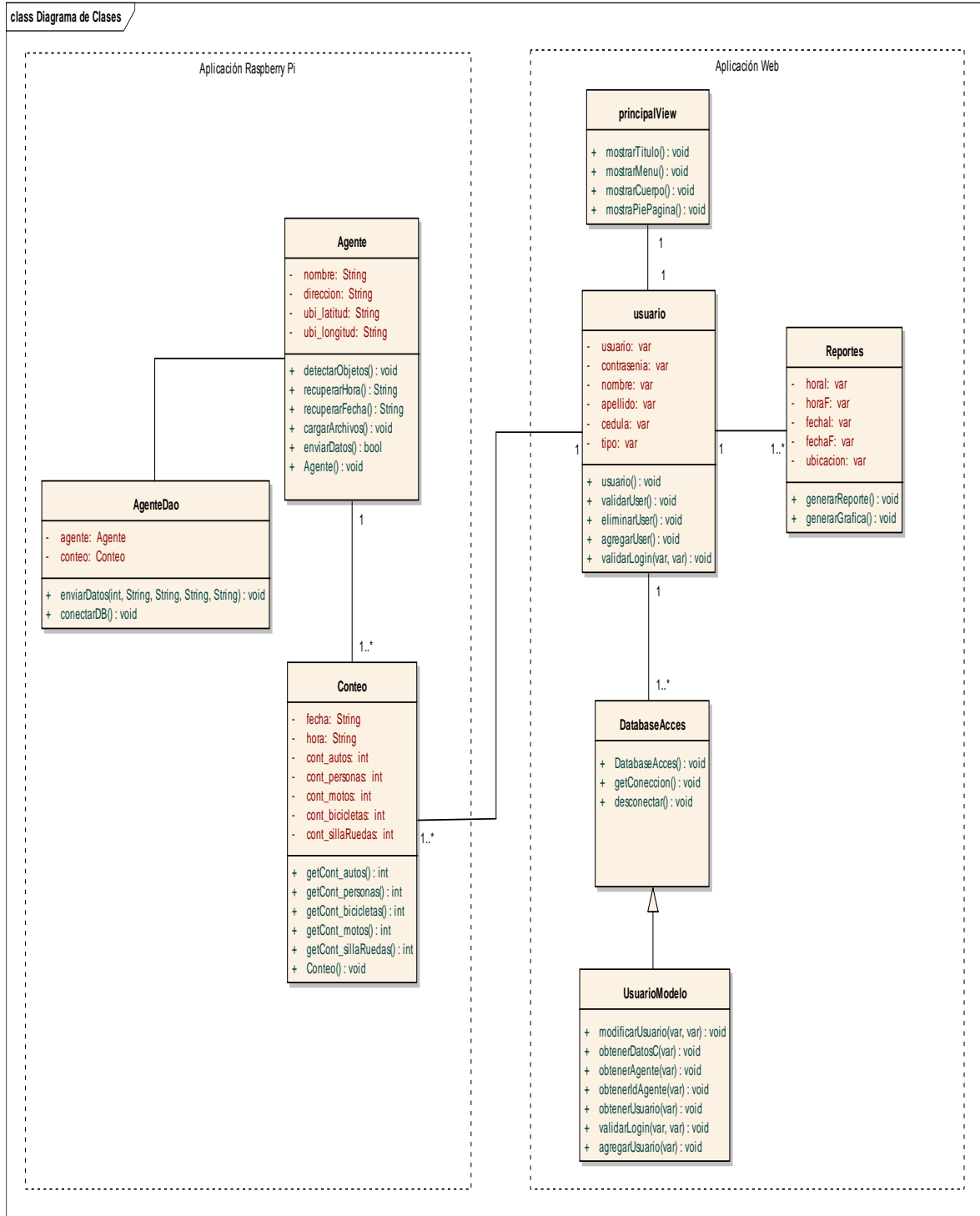


Figura 32. Diagrama de clases.

6.3.3. Modelo de Base de Datos

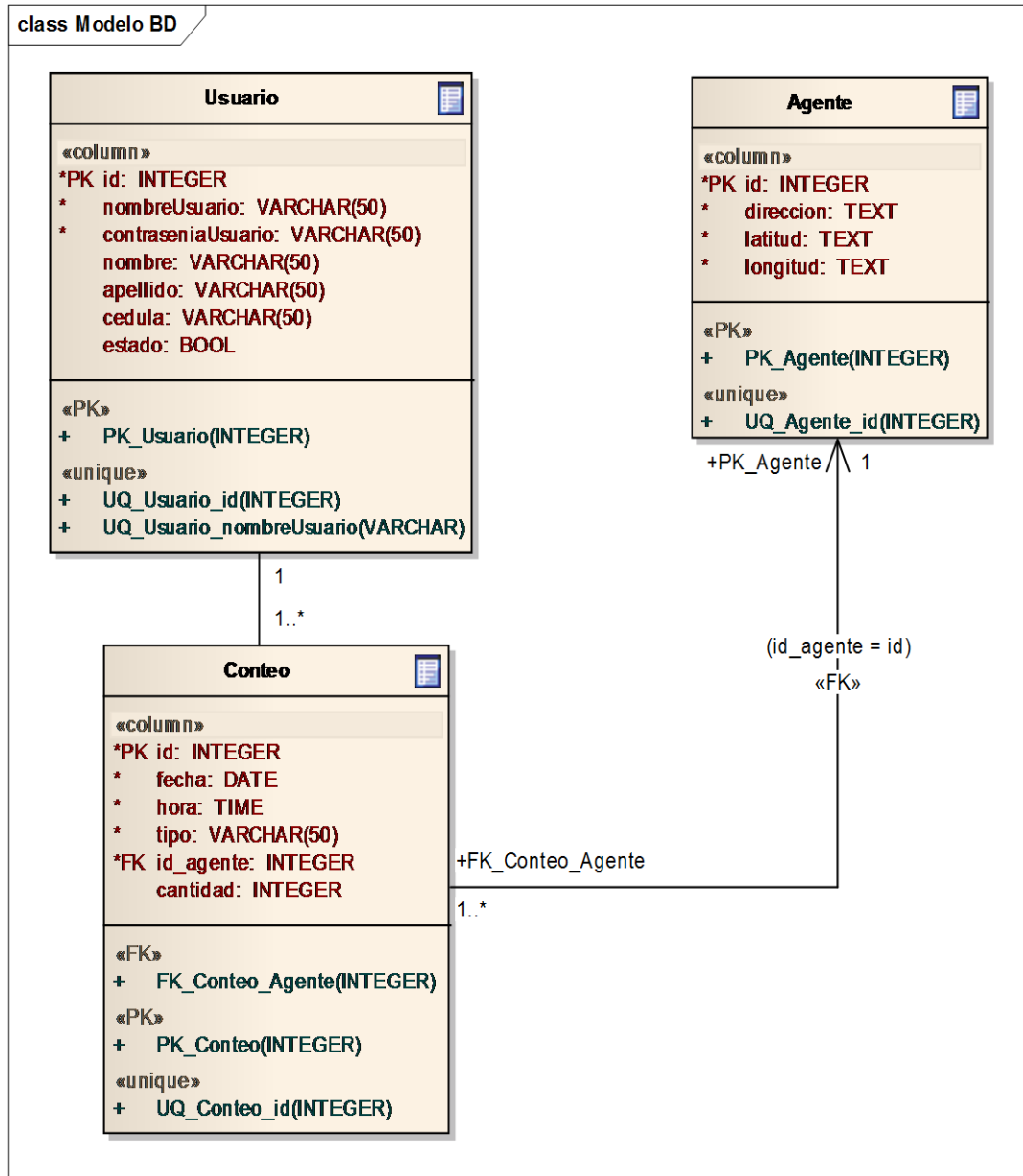


Figura 33. Modelo de la Base de Datos.

6.3.4. Diagrama de Despliegue

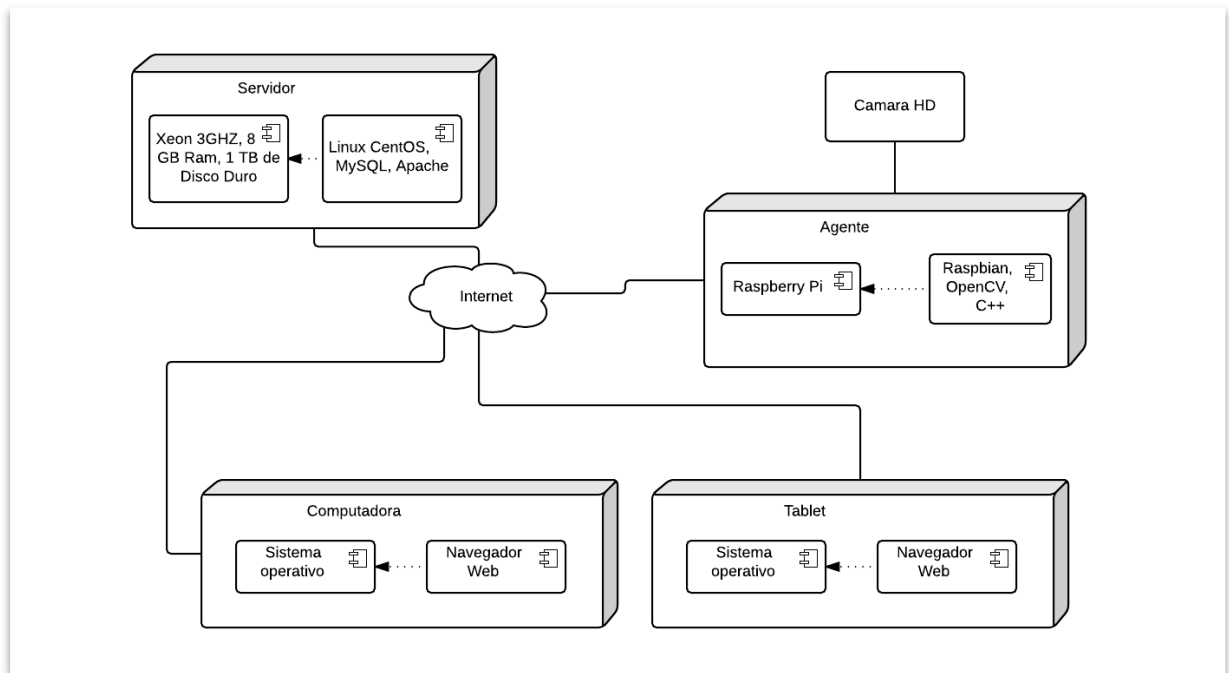


Figura 34. Diagrama de Despliegue.

6.3.5. Arquitectura del Sistema

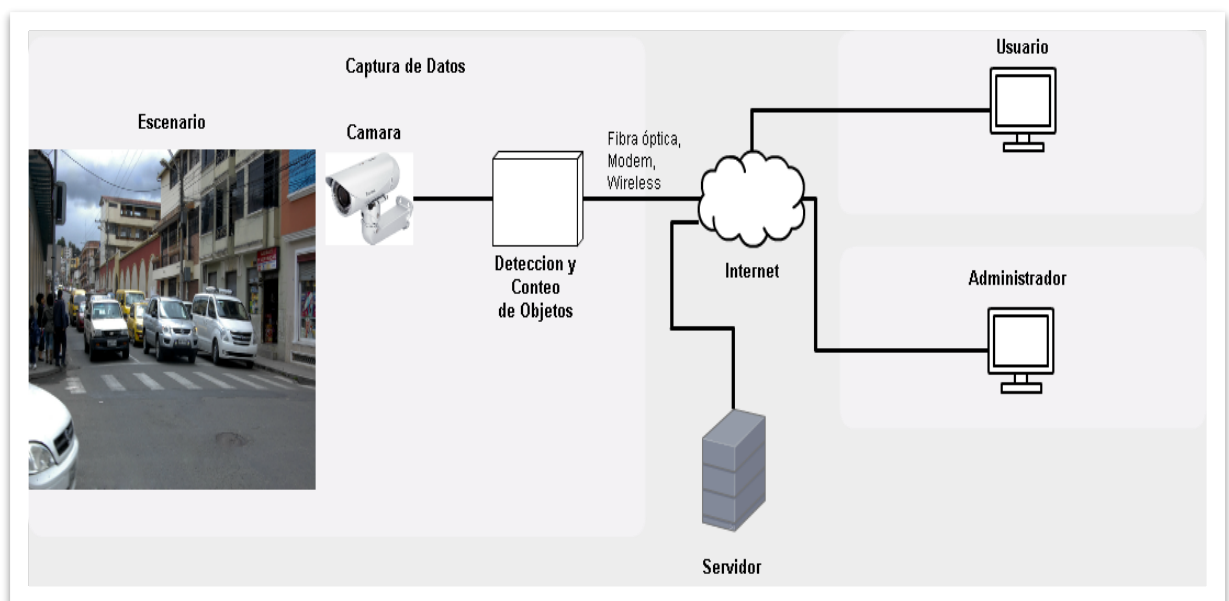


Figura 35. Arquitectura del Sistema de Detección de Objetos.

6.4. Implementación

Para realizar la implementación se utilizó Sistema Operativo Linux, Lenguaje de Programación C++, el IDE QT Creator, MySQL para la Base de Datos y fue necesaria la construcción de clasificadores para el reconocimiento de Personas, Bicicletas, Motos, Autos y Sillas de Ruedas.

6.4.1. Descripción de herramientas utilizadas para las implementación del software

6.4.1.1. Sistema Operativo Linux

Sistema operativo de Software Libre que cuenta con distintas distribuciones como Ubuntu, Fedora, Open Suse, entre otros.

6.4.1.2. Lenguaje de Programación C++

Potente Lenguaje de Programación que permite la fácil integración con la funciones de la Librería OpenCV.

6.4.1.3. IDE QT Creator

Es un entorno de desarrollo libre, dispuesto para trabajar con C++ e interfaz gráfica.

6.4.1.4. Base de Datos MySQL

Gestor de Base de Datos de la aplicación.

6.4.1.5. OpenCV

Biblioteca de Visión por Computadora, proporciona los métodos necesarios para la detección de Personas, Bicicletas, Motos, Autos y Sillas de Ruedas.

6.4.1.6. Construcción del Clasificador

La construcción del clasificador se describe en el Anexo 2.

6.4.2. Implementación del Software

Para la implementación del código se utilizó el entorno de desarrollo QT Creator y el Lenguaje de Programación C++, a continuación se presentan algunos de los segmentos de código utilizados en la implementación del Software.

6.4.2.1. Librerías utilizadas

Las librerías utilizadas en la aplicación, entre otras son las librerías de Opencv, MySQL y librerías para tipos de datos.

6.4.2.2. Método para cargar archivos

Este método carga los clasificadores necesarios para el reconocimiento de objetos y llama al método para detectar objetos, temporizador para enviar los datos a la base de datos y para mostrar los contadores.

Tabla 18. Método para cargar archivos

INICIO
ABRIR cámara
SI existe error de la cámara ENTONCES
MOSTRAR “error de cámara”
CARGAR clasificador de autos, motos, sillas de ruedas, bicicletas y personas
FIN

6.4.2.3. Método para detectar objetos

Utilizando los clasificadores anteriormente cargados se procede a realizar la detección de los diferentes objetos, se detecta los objetos y los guarda en un vector de rectángulos.

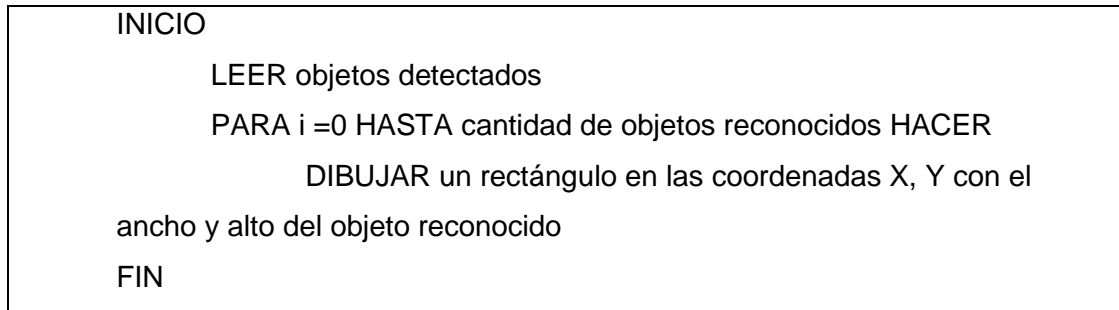
Tabla 19. Algoritmo para detectar objetos

INICIO
CAPTURAR nueva imagen
TRANSFORMAR imagen a escala de grises
DETECTAR objetos en la imagen en escala de grises
GUARDAR objetos detectados
MOSTAR objetos detectados
FIN

6.4.2.4. Método para dibujar rectángulos

Una vez detectado los objetos y guardado en un vector, se dibuja un rectángulo que encierra a cada uno de los objetos detectados, luego al momento que cada objeto pasa por la región de interés se procede a realizar el conteo de dicho objeto.

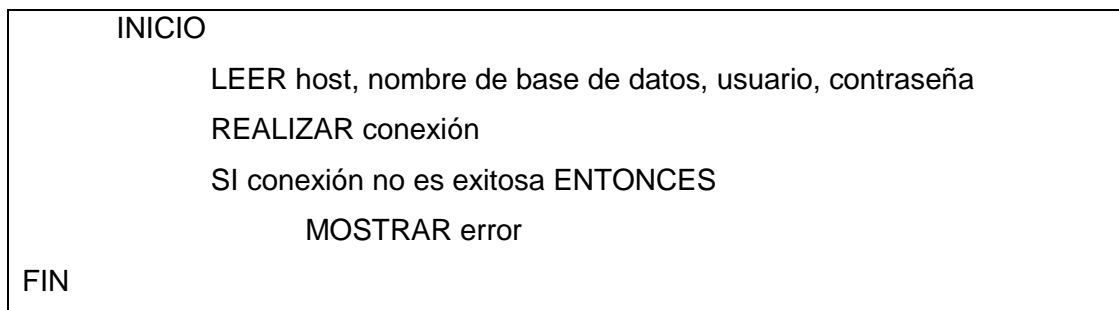
Tabla 20. Algoritmo para dibujar rectángulos



6.4.2.5. Método para realizar la conexión con la Base de Datos

Este método permite realizar la conexión con la base de datos y verificar que no existan errores para luego proceder a realizar el envío de datos.

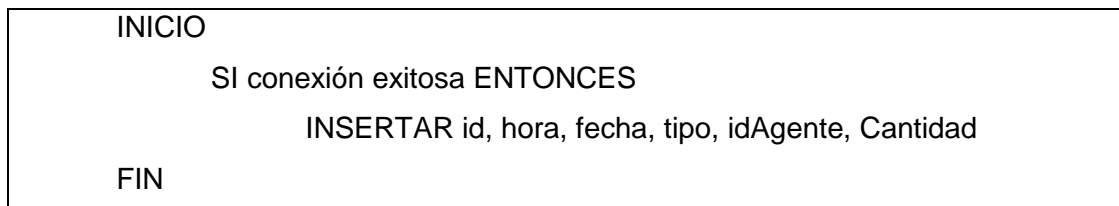
Tabla 21. Algoritmo para realizar la conexión con la base de datos



6.4.2.6. Método para insertar los datos en la Base de Datos

Este método inserta los contadores de objetos a la base de datos cada determinado tiempo, una vez insertados los contadores se los vuelve a poner en 0.

Tabla 22. Algoritmo para insertar los datos en la base de datos



6.5. Pruebas de Rendimiento de los Clasificadores

Una vez construida la aplicación se procede a realizar las pruebas correspondientes para comprobar el rendimiento de los distintos clasificadores en la detección de objetos.

6.5.1. Detección de Vehículos

Para la construcción del clasificador de Vehículos se utilizó 2500 imágenes positivas y 3000 imágenes negativas, siendo una cantidad de imágenes suficiente para obtener un resultado óptimo. La evaluación de los resultados del Clasificador de Vehículos se realizará de la siguiente manera:

6.5.1.1. Prueba 1

Tabla 23. Información del video de Prueba 1 para detección de Vehículos.

Información del Video	
Video	VidVeh1.mp4
Ubicación	Puente peatonal terminal terrestre
Duración	3.03 minutos
Contador Agente	68 vehículos
Contador Manual	69 vehículos
Total de Objetos(cualquier tipo)	78 objetos

Tabla 24. Matriz de confusión para primer video de detección de Vehículos.

Matriz de Confusión			
		Resultado de Clasificación	
		Vehículo	No Vehículo
Instancias Reales	Vehículo	68	2
	No Vehículo	1	8
Cálculo de Valores			
		Exactitud	0.96
		Precisión	0.98
		Sensibilidad	0.97
		Especificidad	0.88
		F1 Score	0.97

Para la primera prueba se utilizó un video tomado desde el puente peatonal del terminal terrestre de la ciudad de Loja, con la cámara ubicada a una altura de 4 metros, vista frontal con respecto a los objetivos y poniendo el punto de referencia a una distancia de 10 metros, los resultados obtenidos en la evaluación del clasificador de vehículos son bastante aceptables que en su mayoría superan el 95%.

6.5.1.2. Prueba 2

Tabla 25. Información del video de Prueba 2 para detección de Vehículos.

Información del Video	
Video	VidVeh2.mp4
Ubicación	Puente peatonal terminal terrestre
Duración	3.17 minutos
Contador Agente	62 vehículos
Contador Manual	61 vehículos
Total de Objetos(cualquier tipo)	73 objetos

Tabla 26. Matriz de confusión para segundo video de detección de Vehículos.

Matriz de Confusión			
		Resultado de Clasificación	
		Vehículo	No Vehículo
Instancias Reales	Vehículo	57	4
	No Vehículo	5	7
Cálculo de Valores			
		Exactitud	0.87
		Precisión	0.91
		Sensibilidad	0.93
		Especificidad	0.58
		F1 Score	0.91

Para la segunda prueba se utilizó un video tomado desde el puente peatonal del terminal terrestre de la ciudad de Loja, con la cámara ubicada a una altura de 3 metros, vista frontal con respecto a los objetivos y poniendo el punto de referencia a una distancia de

10 metros, los resultados obtenidos en la evaluación del clasificador de vehículos con respecto al primer video varían ya que el punto de ubicación de la cámara cambia, por lo tanto se puede deducir que dicha ubicación de la cámara no es la más adecuada para realizar la detección de vehículos.

6.5.1.3. Prueba 3

Tabla 27. Información del video de prueba 3 para detección de Vehículos.

Información del Video	
Video	VidVeh3.mp4
Ubicación	Puente peatonal Mercadillo
Duración	5.08 minutos
Contador Agente	52 vehículos
Contador Manual	80 vehículos
Total de Objetos(cualquier tipo)	93 objetos

Tabla 28. Matriz de confusión para tercer video de detección de Vehículos.

Matriz de Confusión			
		Resultado de Clasificación	
		Vehículo	No Vehículo
Instancias Reales	Vehículo	47	33
	No Vehículo	5	8
Cálculo de Valores			
		Exactitud	0.61
		Precisión	0.90
		Sensibilidad	0.67
		Especificidad	0.61
		F1 Score	0.76

Para la tercera prueba se utilizó un video tomado desde el puente peatonal ubicado en la calle Alonso de Mercadillo y Av. 18 de Noviembre de la ciudad de Loja, con la cámara ubicada a una altura de 4 metros, vista frontal con respecto a los objetivos y poniendo el punto de referencia a una distancia de 5 metros, por la cercanía del punto de referencia no se puede apreciar en su totalidad la parte frontal de los vehículos.

6.5.1.4. Prueba 4

Tabla 29. Información del video de Prueba 4 para la detección de Vehículos.

Información del Video	
Video	VidVeh4.mp4
Ubicación	Puente peatonal Mercadillo
Duración	0.54 minutos
Contador Agente	25 vehículos
Contador Manual	15 vehículos
Total de Objetos(cualquier tipo)	33 objetos

Tabla 30. Matriz de confusión para cuarto video de detección de Vehículos.

Matriz de Confusión			
		Resultado de Clasificación	
		Vehículo	No Vehículo
Instancias Reales	Vehículo	15	0
	No Vehículo	10	8
Cálculo de Valores			
		Exactitud	0.69
		Precisión	0.60
		Sensibilidad	1
		Especificidad	0.44
		F1 Score	0.75

Para la cuarta prueba se utilizó un video tomado desde el puente peatonal ubicado en la calle Alonso de Mercadillo y Av. 18 de Noviembre de la ciudad de Loja, con la cámara ubicada a una altura de 4 metros, vista frontal con respecto a los objetivos y poniendo el punto de referencia a una distancia de 5 metros, como se puede observar en la tabla anterior la mayoría de los resultados obtenidos en la evaluación del clasificador de vehículos no supera el 70%, esto se debe principalmente a la cercanía del punto de referencia por lo que la captura de video no se pudo apreciar de manera clara los objetivos.

6.5.1.5. Prueba 5

Tabla 31. Información del video de Prueba 5 para la detección de Vehículos.

Información del Video	
Video	VidVeh5.mp4
Ubicación	Puente peatonal Terminal Terrestre
Duración	4.13 minutos
Contador Agente	50 vehículos
Contador Manual	40 vehículos
Total de Objetos(cualquier tipo)	63 objetos

Tabla 32. Matriz de confusión para quinto video de detección de Vehículos.

Matriz de Confusión			
		Resultado de Clasificación	
		Vehículo	No Vehículo
Instancias Reales	Vehículo	40	0
	No Vehículo	10	13
Cálculo de Valores			
		Exactitud	0.84
		Precisión	0.8
		Sensibilidad	1
		Especificidad	0.56
		F1 Score	0.89

Para la quinta y última prueba se utilizó un video tomado desde el puente peatonal ubicado en el terminal terrestre de la ciudad de Loja, con la cámara ubicada a una altura de 4 metros, poniendo el punto de referencia a una distancia de 5 metros, los resultados obtenidos en la evaluación del clasificador de vehículos varían entre el 56% y 100% esto se debe principalmente a la cercanía del punto de referencia por lo que en la captura de video no se pudo apreciar en su totalidad la parte frontal de los vehículos.

6.5.1.6. Diagrama de análisis de Detección de Vehículos

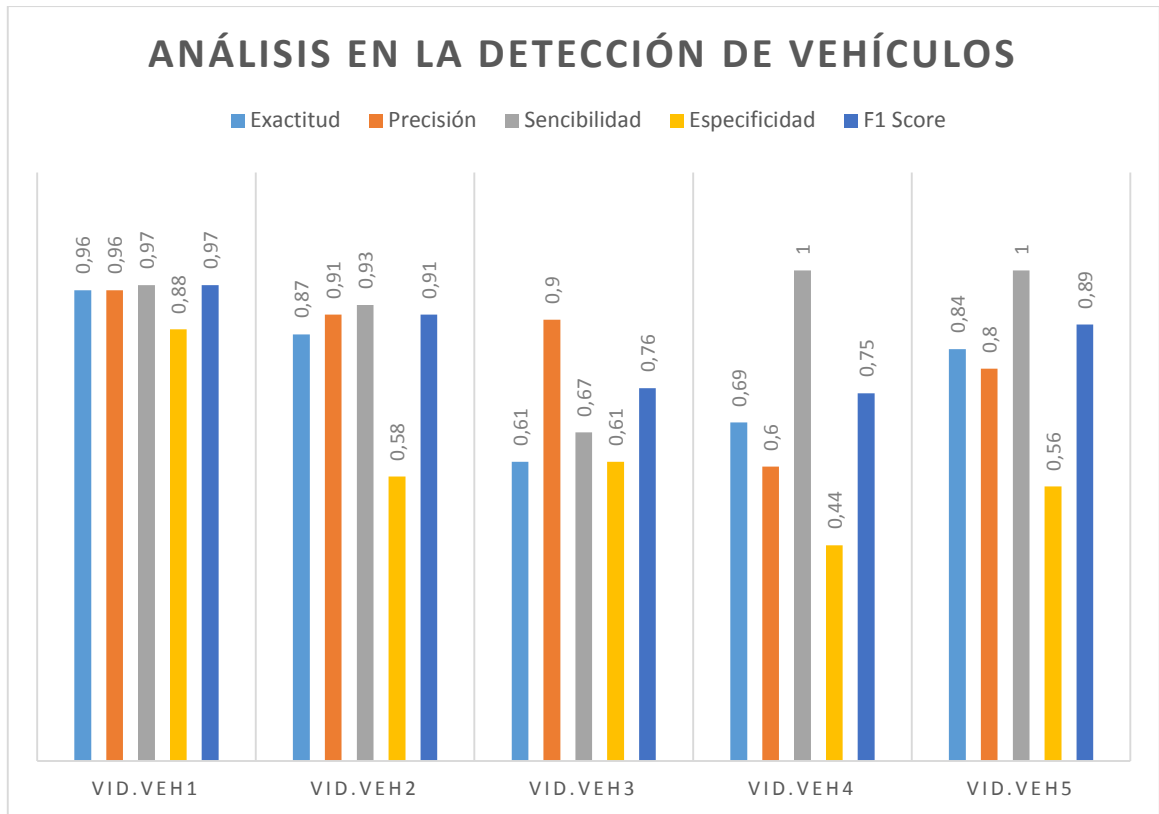


Figura 36. Diagrama de análisis de detección de Vehículos.

En la figura 36 se denota los resultados obtenidos luego de realizar las pruebas con el clasificador de vehículos, se puede deducir que se obtienen mejores resultados en el primer video, en estos resultados influye mucho las condiciones en las que se ha tomado el video, como por ejemplo la ubicación de la cámara y la distancia del punto de referencia.

6.5.2. Detección de Personas

Para la construcción del clasificador de Personas se utilizó 1900 imágenes positivas y 2500 imágenes negativas, siendo una cantidad de imágenes suficiente para obtener un resultado óptimo. La evaluación de los resultados del Clasificador de Personas se realiza de la siguiente manera:

6.5.2.1. Prueba 1

Tabla 33. Información del video de Prueba 1 para la detección de Personas.

Información del Video	
Video	VidPers1.mp4
Ubicación	Terminal Terrestre
Duración	1.07 minutos
Contador Agente	16 personas
Contador Manual	15 personas
Total de Objetos(cualquier tipo)	23 objetos

Tabla 34. Matriz de confusión para primer video de detección de Personas.

Matriz de Confusión			
		Resultado de Clasificación	
		Persona	No Persona
Instancias Reales	Persona	14	1
	No Persona	2	6
Cálculo de Valores			
		Exactitud	0.87
		Precisión	0.87
		Sensibilidad	0.93
		Especificidad	0.75
		F1 Score	0.89

Para la primera prueba se utilizó un video tomado en el terminal terrestre de la ciudad de Loja, con la cámara ubicada a una altura de 3 metros, vista frontal con respecto a los objetivos y poniendo el punto de referencia a una distancia de 5 metros, los resultados obtenidos en la evaluación del clasificador de personas son bastante aceptables que en su mayoría superan el 85%, además cabe destacar que existen varias personas sentadas lo cual provoca confusión en el clasificador ya fue entrenado únicamente con personas de pie.

6.5.2.2. Prueba 2

Tabla 35. Información del video de Prueba 2 para la detección de Personas.

Información del Video	
Video	VidPers2.mp4
Ubicación	Puente peatonal Terminal Terrestre
Duración	4.39 minutos
Contador Agente	15 personas
Contador Manual	13 personas
Total de Objetos(cualquier tipo)	20 objetos

Tabla 36. Matriz de confusión para segundo video de detección de Personas.

Matriz de Confusión			
		Resultado de Clasificación	
		Persona	No Persona
Instancias Reales	Persona	12	1
	No Persona	3	4
Cálculo de Valores			
		Exactitud	0.8
		Precisión	0.8
		Sensibilidad	0.92
		Especificidad	0.57
		F1 Score	0.86

Para la segunda prueba se utilizó un video tomado desde el puente peatonal del terminal terrestre de la ciudad de Loja, con la cámara ubicada a una altura de 4 metros, poniendo el punto de referencia a una distancia de 10 metros, los resultados obtenidos en la evaluación del clasificador de personas son bastante aceptables que en su mayoría superan el 80%, en este caso el video no se tomó de una manera frontal lo cual provoca que los objetos no se aprecien de manera correcta.

6.5.2.3. Prueba 3

Tabla 37. Información del video de Prueba 3 para la detección de Personas.

Información del Video	
Video	VidPers3.mp4
Ubicación	Puente peatonal Terminal Terrestre
Duración	4.37 minutos
Contador Agente	22 personas
Contador Manual	14 personas
Total de Objetos(cualquier tipo)	33 objetos

Tabla 38. Matriz de confusión para tercer video de detección de Personas.

Matriz de Confusión			
		Resultado de Clasificación	
		Persona	No Persona
Instancias Reales	Persona	14	0
	No Persona	8	11
Cálculo de Valores			
		Exactitud	0.75
		Precisión	0.64
		Sensibilidad	1
		Especificidad	0.57
		F1 Score	0.78

Para la segunda prueba se utilizó un video tomado desde el puente peatonal del terminal terrestre de la ciudad de Loja, con la cámara ubicada a una altura de 4 metros, poniendo el punto de referencia a una distancia de 15 metros, los resultados obtenidos en la evaluación del clasificador de personas varían entre el 64% y 100%, en este caso el video no se tomó de una manera frontal lo cual provoca que los objetos no se aprecien de manera correcta, además se debe tomar en cuenta que el punto de referencia está demasiado distante.

6.5.2.4. Prueba 4

Tabla 39. Información del video de Prueba 4 para la detección de Personas.

Información del Video	
Video	VidPers4.mp4
Ubicación	Esquina calle 18 de Noviembre y 10 de Agosto
Duración	1.11 minutos
Contador Agente	17 personas
Contador Manual	13 personas
Total de Objetos(cualquier tipo)	25 objetos

Tabla 40. Matriz de confusión para cuarto video de detección de Personas.

Matriz de Confusión			
		Resultado de Clasificación	
		Persona	No Persona
Instancias Reales	Persona	13	0
	No Persona	4	8
Cálculo de Valores			
		Exactitud	0.84
		Precisión	0.76
		Sensibilidad	1
		Especificidad	0.66
		F1 Score	0.86

Para la cuarta prueba se utilizó un video tomado en la intersección de la calle 18 de Noviembre y 10 de Agosto de la ciudad de Loja, con la cámara ubicada a una altura de 2 metros, poniendo el punto de referencia a una distancia de 3 metros, los resultados obtenidos en la evaluación del clasificador de personas son variados, al estar la cámara ubicada a una baja altura en algunos momentos los objetos se ubican uno detrás de otro lo cual provocaba ineficiencia en la detección.

6.5.2.5. Prueba 5

Tabla 41. Información del video de Prueba 5 para la detección de Personas.

Información del Video	
Video	VidPers5.mp4
Ubicación	Esquina calle 10 de Agosto y 18 de Noviembre
Duración	0.56 minutos
Contador Agente	21 personas
Contador Manual	22 personas
Total de Objetos(cualquier tipo)	26 objetos

Tabla 42. Matriz de confusión para quinto video de detección de Personas.

Matriz de Confusión			
		Resultado de Clasificación	
		Persona	No Persona
Instancias Reales	Persona	21	1
	No Persona	0	4
Cálculo de Valores			
		Exactitud	0.96
		Precisión	1
		Sensibilidad	0.95
		Especificidad	1
		F1 Score	0.97

Para la quinta prueba se utilizó un video tomado en la intersección de la calle 18 de Noviembre y 10 de Agosto de la ciudad de Loja, con la cámara ubicada a una altura de 3.5 metros, poniendo el punto de referencia a una distancia de 4 metros, los resultados obtenidos en la evaluación del clasificador de personas son bastante aceptables superando todos el 95%.

6.5.2.6. Diagrama de análisis de Detección de Personas

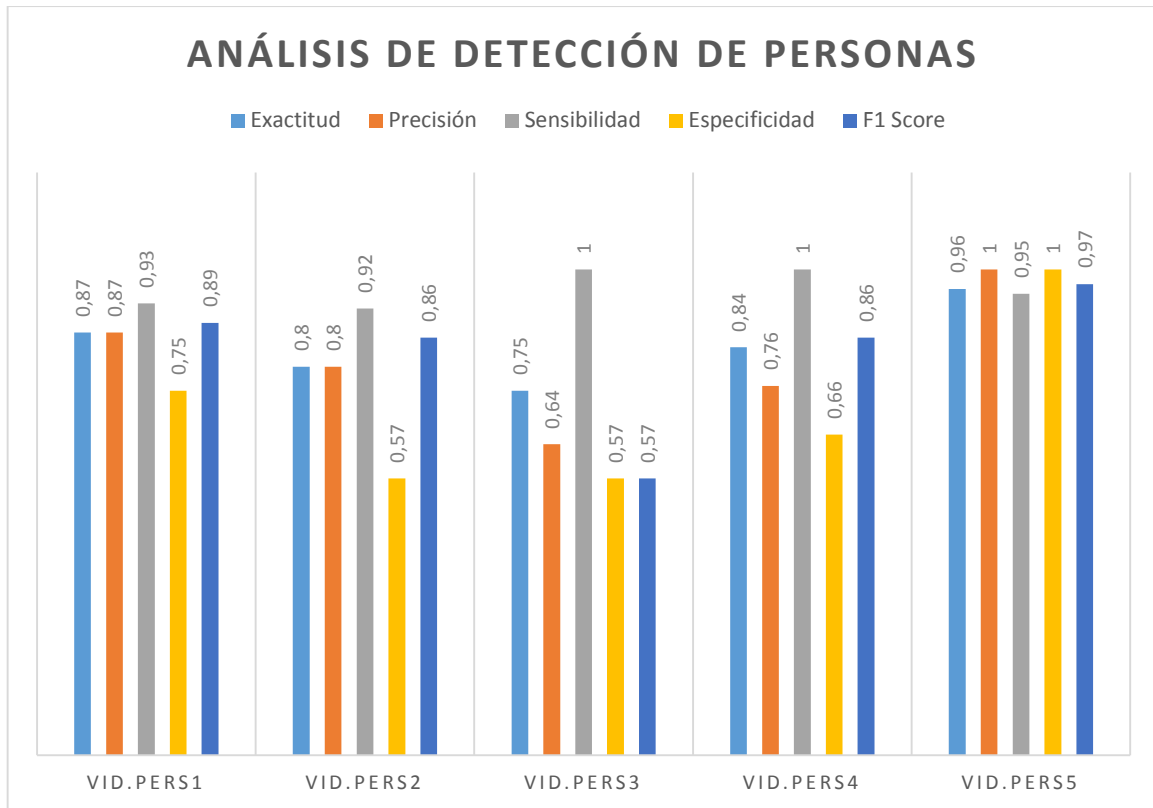


Figura 37. Diagrama de análisis de detección de Personas.

En la figura 37 se muestra los resultados obtenidos luego de realizar la evaluación del clasificador de personas con videos tomados desde diferentes alturas y con puntos de referencia a diferentes distancias, se puede apreciar que los mejores resultados se han obtenido en el quinto video.

6.5.3. Detección de Bicicletas

Para la construcción del clasificador de Bicicletas se utilizó 1940 imágenes positivas y 2530 imágenes negativas, siendo una cantidad de imágenes suficiente para obtener un resultado óptimo. La evaluación de los resultados del Clasificador de Bicicletas se realiza de la siguiente manera:

6.5.3.1. Prueba 1

Tabla 43. Información del video de Prueba 1 para la detección de Bicicletas.

Información del Video	
Video	VidBici1.mp4
Ubicación	Descargado de la web
Duración	0.47 minutos
Contador Agente	49 bicicletas
Contador Manual	50 bicicletas
Total de Objetos(cualquier tipo)	55 objetos

Tabla 44. Matriz de confusión para el primer video de detección de Bicicletas.

Matriz de Confusión			
		Resultado de Clasificación	
		Bicicleta	No Bicicleta
Instancias Reales	Bicicleta	46	4
	No Bicicleta	3	2
Cálculo de Valores			
		Exactitud	0.87
		Precisión	0.93
		Sensibilidad	0.92
		Especificidad	0.4
		F1 Score	0.92

Para la primera prueba se utilizó un video descargado de internet, con la cámara ubicada a una altura de 2 metros, poniendo el punto de referencia a una distancia de 3 metros, los resultados obtenidos en la evaluación del clasificador de bicicletas son variados, al no estar la cámara ubicada frontalmente los objetos no se pueden apreciar correctamente.

6.5.3.2. Prueba 2

Tabla 45. Información del video de Prueba 2 para la detección de Bicicletas.

Información del Video	
Video	VidBici2.mp4
Ubicación	Descargado de la web
Duración	0.15 minutos
Contador Agente	11 bicicletas
Contador Manual	10 bicicletas
Total de Objetos(cualquier tipo)	14 objetos

Tabla 46. Matriz de confusión para el segundo video de detección de Bicicletas.

Matriz de Confusión			
		Resultado de Clasificación	
		Bicicleta	No Bicicleta
Instancias Reales	Bicicleta	10	0
	No Bicicleta	1	3
Cálculo de Valores			
		Exactitud	0.92
		Precisión	0.90
		Sensibilidad	1
		Especificidad	0.75
		F1 Score	0.94

Para la segunda prueba se utilizó un video descargado de internet, con la cámara ubicada a una altura de 2 metros, poniendo el punto de referencia a una distancia de 4 metros, los resultados obtenidos en la evaluación del clasificador de bicicletas son variados pero en su mayoría aceptables, al no estar la cámara ubicada frontalmente los objetos no se pueden apreciar correctamente.

6.5.3.3. Prueba 3

Tabla 47. Información del video de Prueba 3 para la detección de Bicicletas.

Información del Video	
Video	VidBici4.mp4
Ubicación	Descargado de la web
Duración	0.33 minutos
Contador Agente	19 bicicletas
Contador Manual	22 bicicletas
Total de Objetos(cualquier tipo)	23 objetos

Tabla 48. Matriz de confusión para el tercer video de detección de Bicicletas.

Matriz de Confusión			
		Resultado de Clasificación	
		Bicicleta	No Bicicleta
Instancias Reales	Bicicleta	19	3
	No Bicicletas	0	1
Cálculo de Valores			
		Exactitud	0.87
		Precisión	1
		Sensibilidad	0.86
		Especificidad	1
		F1 Score	0.92

Para la tercera prueba se utilizó un video descargado de internet, con la cámara ubicada a una altura de 1.5 metros, poniendo el punto de referencia a una distancia de 2 metros, los resultados obtenidos en la evaluación del clasificador de bicicletas superan el 85%, al estar la cámara demasiado baja y el punto de referencia muy cerca crea cierto grado de dificultad en la detección ya que al momento de acercarse los objetos al lugar donde se encuentra ubicada la cámara el tamaño de los mismo va aumentando.

6.5.3.4. Prueba 4

Tabla 49. Información del video de Prueba 4 para la detección de Bicicletas.

Información del Video	
Video	VidBici7.mp4
Ubicación	Descargado de la web
Duración	0.33 minutos
Contador Agente	6 bicicletas
Contador Manual	7 bicicletas
Total de Objetos(cualquier tipo)	11 objetos

Tabla 50. Matriz de confusión para el cuarto video de detección de Bicicletas.

Matriz de Confusión			
		Resultado de Clasificación	
		Bicicletas	No Bicicletas
Instancias Reales	Bicicletas	6	1
	No Bicicletas	0	4
Cálculo de Valores			
		Exactitud	0.90
		Precisión	1
		Sensibilidad	0.85
		Especificidad	1
		F1 Score	0.92

Para la primera prueba se utilizó un video descargado de internet, con la cámara ubicada a una altura de 3 metros, poniendo el punto de referencia a una distancia de 5 metros, los resultados obtenidos en la evaluación del clasificador de bicicletas en su mayoría superan el 90%, por consiguiente se puede deducir este como un resultado aceptable.

6.5.3.5. Diagrama de análisis de Detección de Bicicletas

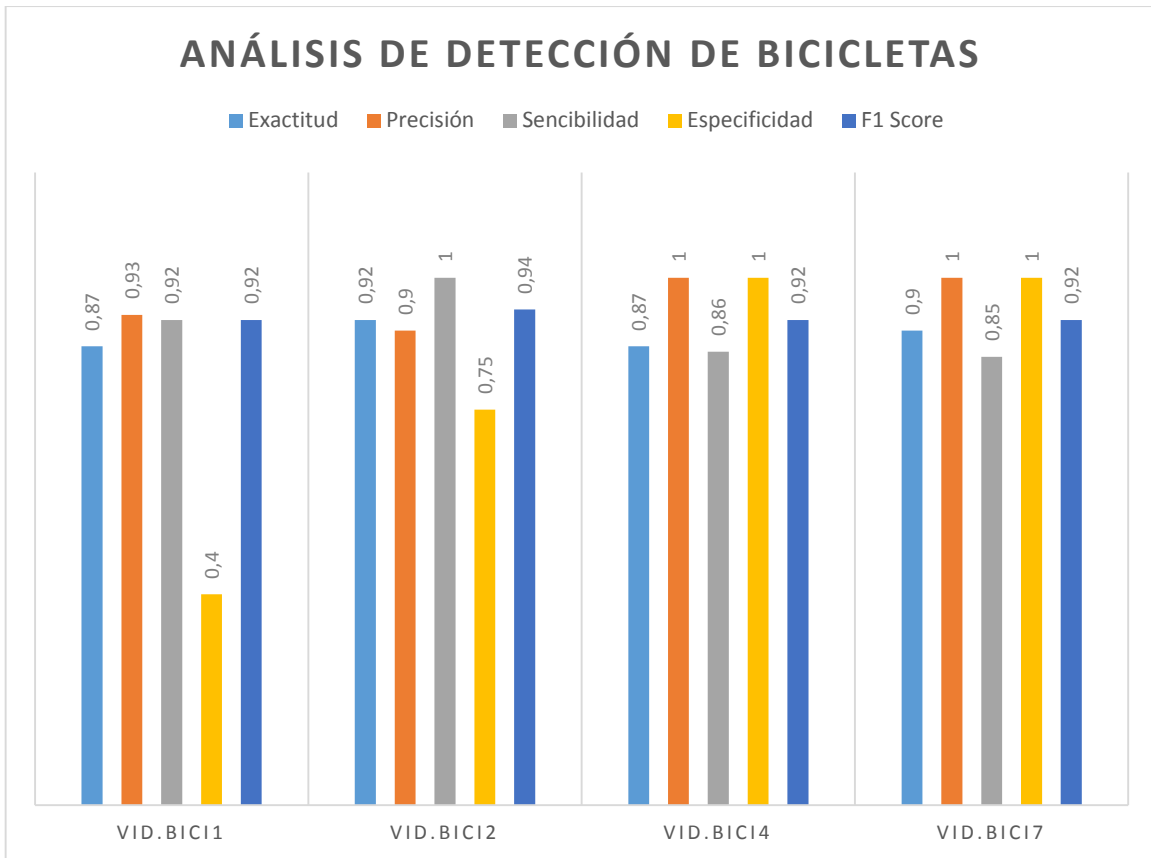


Figura 38. Diagrama de análisis de detección de Bicicletas.

Una vez terminada la evaluación del clasificador de bicicletas, en la figura 38 se muestran los resultados obtenidos de las pruebas realizadas con diferentes videos y con la ubicación de la cámara en diferentes puntos; se puede deducir que las condiciones ideales para la detección de bicicletas son las que presenta el video utilizado en la cuarta prueba

6.5.4. Detección de Motos

Para la construcción del clasificador de Motos se utilizó 1500 imágenes positivas y 2200 imágenes negativas, siendo una cantidad de imágenes suficiente para obtener un resultado óptimo. La evaluación de los resultados del Clasificador de Motos se realizará de la siguiente manera:

6.5.4.1. Prueba 1

Tabla 51. Información de video de Prueba 1 para la detección de Motos.

Información del Video	
Video	VidMt7.mp4
Ubicación	Descargado de la web
Duración	0.20 minutos
Contador Agente	5 motos
Contador Manual	7 motos
Total de Objetos(cualquier tipo)	9 objetos

Tabla 52. Matriz de confusión para el primer video de detección de Motos.

Matriz de Confusión			
		Resultado de Clasificación	
		Motos	No Motos
Instancias Reales	Motos	5	2
	No Motos	0	2
Cálculo de Valores			
		Exactitud	0.77
		Precisión	1
		Sensibilidad	0.71
		Especificidad	1
		F1 Score	0.83

Para la primera prueba se utilizó un video descargado de internet, con la cámara ubicada a una altura de 2 metros, poniendo el punto de referencia a una distancia variada, los resultados obtenidos en la evaluación del clasificador de motos son variados, la cámara se encuentra en movimiento y al no estar ubicada frontalmente los objetos no se pueden apreciar correctamente.

6.5.4.2. Prueba 2

Tabla 53. Información de video de Prueba 2 para la detección de Motos.

Información del Video	
Video	VidMt8.mp4
Ubicación	Descargado de la web
Duración	0.13 minutos
Contador Agente	5 motos
Contador Manual	4 motos
Total de Objetos(cualquier tipo)	6 objetos

Tabla 54. Matriz de confusión para el segundo video de detección de Motos.

Matriz de Confusión			
		Resultado de Clasificación	
		Motos	No Motos
Instancias Reales	Motos	4	0
	No Motos	1	1
Cálculo de Valores			
		Exactitud	0.83
		Precisión	0.8
		Sensibilidad	1
		Especificidad	0.5
		F1 Score	0.89

Para la primera prueba se utilizó un video descargado de internet, con la cámara ubicada a una altura de 2 metros, poniendo el punto de referencia a una distancia variada, los resultados obtenidos en la evaluación del clasificador de motos son variados, la cámara se encuentra en movimiento y al no estar ubicada frontalmente los objetos no se pueden apreciar correctamente.

6.5.4.3. Diagrama de análisis de Detección de Motos

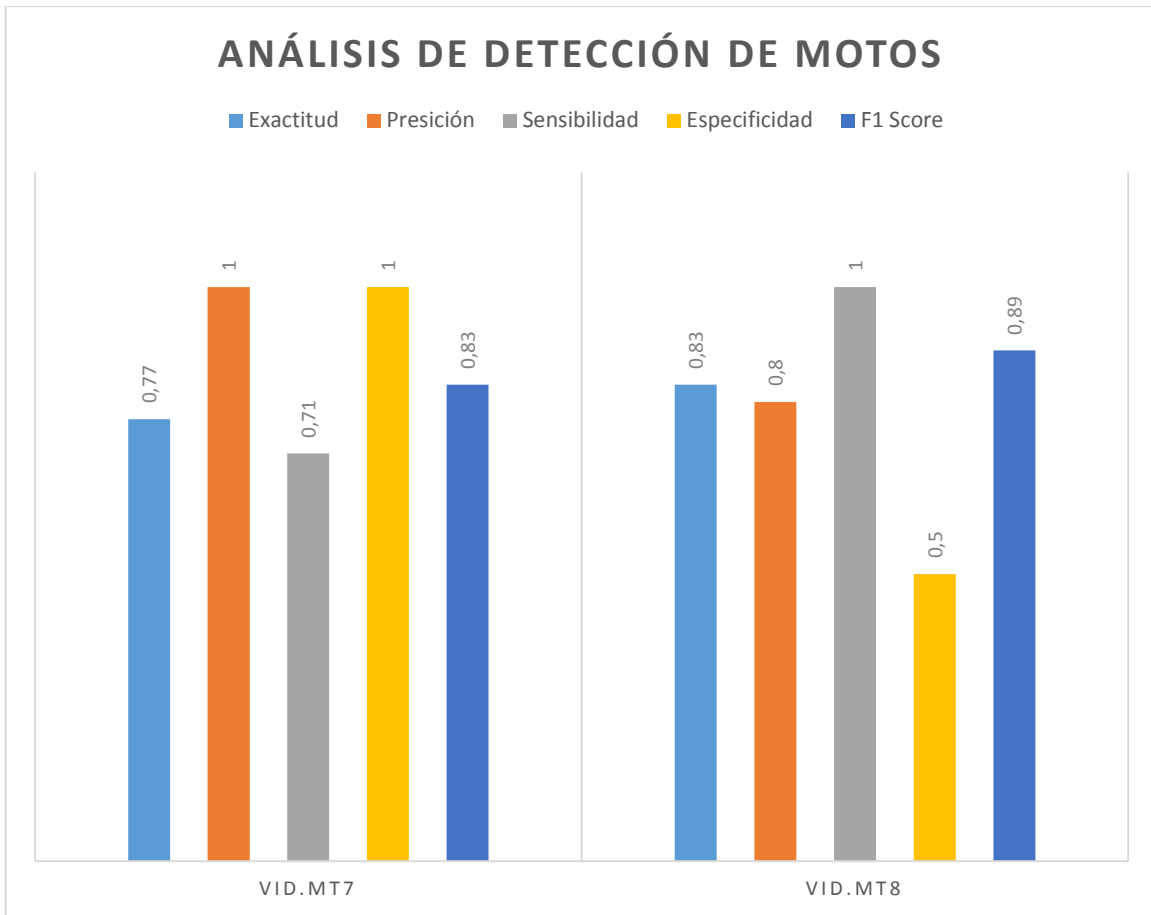


Figura 39. Diagrama de análisis de detección de Motos.

Por la dificultad para encontrar videos de motos grabados en condiciones ideales para que puedan ser detectados los objetos solo se realizó las pruebas con estos dos videos tal como se indica en la figura 39, a pesar de que ninguno de los videos son tomados desde un puntos aceptable los valores obtenidos en los dos videos en su mayoría superan el 75%.

6.5.5. Detección de Sillas de Ruedas

Para la construcción del clasificador de Sillas de Ruedas se utilizó 800 imágenes positivas y 1500 imágenes negativas, siendo una cantidad de imágenes suficiente para obtener un resultado óptimo. La evaluación de los resultados del Clasificador de Sillas de Ruedas se realiza de la siguiente manera:

6.5.5.1. Prueba 1

Tabla 55. Información de video de Prueba 1 para la detección de Sillas de Ruedas.

Información del Video	
Video	VidSillas.mp4
Ubicación	Descargado de la web
Duración	0.54 minutos
Contador Agente	8 sillas de ruedas
Contador Manual	9 sillas de ruedas
Total de Objetos(cualquier tipo)	10 objetos

Tabla 56. Matriz de confusión para el segundo video de detección de Sillas de Ruedas.

Matriz de Confusión			
		Resultado de Clasificación	
		Sillas	No Sillas
Instancias Reales	Sillas	8	1
	No Sillas	0	1
Cálculo de Valores			
		Exactitud	0.9
		Precisión	1
		Sensibilidad	0.88
		Especificidad	1
		F1 Score	0.93

Para esta prueba se utilizó un video descargado de internet, con la cámara ubicada a una altura variada, poniendo el punto de referencia a una distancia variada, los resultados obtenidos en la evaluación del clasificador de sillas de ruedas superan el 85%, la cámara se encuentra en movimiento y al no estar ubicada frontalmente los objetos no se pueden apreciar correctamente.

6.5.5.2. Diagrama de análisis de Detección de Sillas de Ruedas

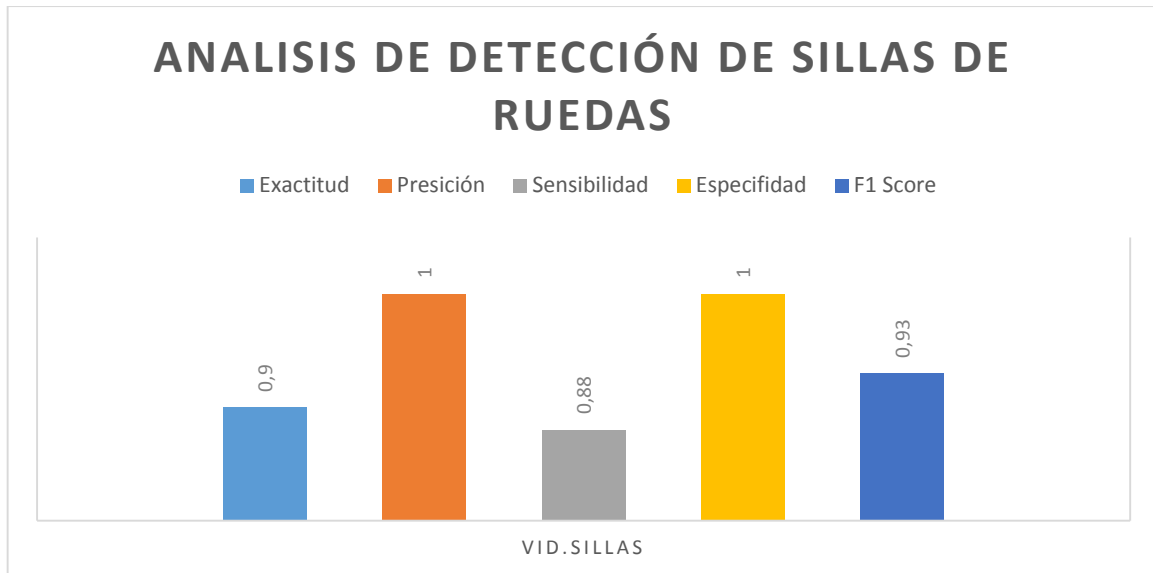


Figura 40. Diagrama de análisis de detección de sillas de ruedas.

Al igual que el caso anterior fue difícil encontrar videos de sillas de ruedas grabados en condiciones ideales para detectar los objetos, a pesar de estos inconvenientes se puede apreciar en la figura 40 que los valores obtenidos son bastante aceptables.

6.6. Entrenamiento del detector de objetos

Para realizar el entrenamiento de los detectores de objetos OpenCV cuenta con aplicaciones específicas para realizar esta tarea, una de ellas es **opencv_haartraining.exe**. Esta aplicación hace uso de los filtros de Haar, estos filtros son una combinación de rectángulos del mismo tamaño adyacentes horizontal y verticalmente.

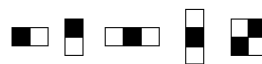


Figura 41. filtros de Haar

Los rectángulos negros de cada filtro representan una contribución positiva al filtro y los blancos representan una contribución negativa, los resultados del filtro es la diferencia entre la suma de los valores de los pixeles entre zonas en negro y zonas en blanco. Los filtros recorren toda la imagen a diferentes escalas recogiendo de esta manera un gran número de características dentro de la imagen.

200	200	100	100	200	200	100	100
250	250	50	50	250	250	50	50
255	255	255	255	100	100	100	100
255	255	255	255	100	100	100	100
200	200	100	100	200	200	100	100
250	250	50	50	250	250	50	50
255	255	255	255	100	100	200	200
255	255	255	255	100	100	250	250

Figura 42. Ejemplo de filtros de Haar

Además dicha aplicación está basada en el método de clasificación Adaboost, este algoritmo se describe a continuación:

Tabla 57. Algoritmo Adaboost

INICIO

- Teniendo un set de ejemplos $(e_1, l_1), \dots, (e_N, l_N)$ y siendo a el número de muestras positivas ($l_n = +1$) y b el número de negativas ($l_n = -1$), inicializa los pesos tal que:

- Para muestras positivas

$$D_1(n) = \frac{1}{2a}$$

- Para muestras negativas

$$D_1(n) = \frac{1}{2b}$$

- Desde $t = 1$ hasta T , siendo T el número de clasificadores débiles que se quieren usar:

- Se normaliza los pesos:

$$D_t(n) = \frac{D_t(n)}{\sum_{i=1}^N D_t(i)}$$

- Para cada característica se entrena un clasificador débil h_j usando los pesos D_t

- Para cada h_j se calcula:

$$r_j = \sum_{n=1}^N D_t(n) l_n h_j(e_n)$$

Donde $h_j(e_n) \in \{+1, -1\}$

- Elegir h_j que maximice $|r_j|$ y guardarlo como $(h_t, r_t) = (h_j, r_j)$
- Actualizar los pesos:

$$D_{t+1}(n) = D_t(n)\exp(-\alpha_t l_n h_t(e_n))$$

Donde

$$\alpha = \frac{1}{2} \log\left(\frac{1+r_t}{1-r_t}\right)$$

- Finalmente los datos son guardados para poder ser utilizados

FIN

Los requisitos previos antes de iniciar con el entrenamiento son las imágenes positivas y negativas, las imágenes positivas representadas en un vector el cual almacena la muestra de cada una de ellas y las negativas representadas en un archivo de texto plano que guarda el nombre y la extensión de cada imagen.

Para ejecutar el programa de entrenamiento es necesario proporcionar la siguiente información:

haartraining.exe -data cascades -vec vector/sillasvector.vec -bg negative/bg.txt -npos 800 -nneg 1300 -nstages 20 -mem 1024 -mode ALL -w 50 -h 50 rem -nonsym

Donde:

- data cascades = Nombre que le damos al clasificador
- vec sillasvector.vec = Vector
- bg bg.txt = Archivo de las imágenes negativas
- npos 800 = Número de imágenes positivas.
- nneg 1300 = Número de imágenes negativas.
- w 50 = Ancho
- h 24 = Alto
- nstages 20 = Número de fases.
- mem 1024 = Memoria dedicada.
- mode ALL = El modo ALL es el recomendado para el entrenamiento (utiliza el conjunto completo de características).

Una vez especificados todos estos parámetros inicia el entrenamiento, la primera etapa es una exploración rápida de la imagen, nos proporciona nuestro primer clasificador:

```

*** 1 cluster ***
POS: 75 75 1.000000
NEG: 121 1
BACKGROUND PROCESSING TIME: 0.09
Precalculation time: 3.70
+-----+-----+-----+-----+-----+
| N | %SMP | F | ST.THR | HR | FA | EXP. ERR |
+-----+-----+-----+-----+-----+
| 1 | 100% | - | -0.845960 | 1.000000 | 1.000000 | 0.056122 |
+-----+-----+-----+-----+-----+
| 2 | 100% | + | -0.003078 | 1.000000 | 0.140496 | 0.056122 |
+-----+-----+-----+-----+-----+
Stage training time: 236.15
Number of used features: 2

```

Figura 43. Primera etapa de entrenamiento.

```

|2
1
2
0 9 12 28 0 -1
6 9 6 28 0 2
haar_x2
-1.273783e-002 0 -1
9.140300e-001 -8.459601e-001
1
2
1 30 48 3 0 -1
13 30 24 3 0 2
haar_x4
-1.006251e-002 0 -1
8.420818e-001 -7.784180e-001
-3.878295e-003

-1
-1

```

Figura 44 Clasificador de la primera etapa

Si se realizara pruebas de rendimiento, este clasificador no sería todo lo preciso que nos gustaría, para ello es necesario más etapas.

En la siguiente etapa se pretende obtener, basándonos en las imágenes usadas para las pruebas en la etapa anterior, una versión mejorada de dicho clasificador, para ello el número de características aumenta y por tanto se producen más iteraciones.

```

*** 1 cluster ***
POS: 75 75 1.000000
NEG: 121 0.25
BACKGROUND PROCESSING TIME: 0.08
Precalculation time: 3.68
+-----+-----+-----+-----+-----+
| N | %SMP | F | ST.THR | HR | FA | EXP. ERR |
+-----+-----+-----+-----+-----+
| 1 | 100% | - | -0.672223 | 1.000000 | 1.000000 | 0.137755 |
+-----+-----+-----+-----+-----+
| 2 | 100% | + | -1.118875 | 1.000000 | 1.000000 | 0.137755 |
+-----+-----+-----+-----+-----+
| 3 | 100% | - | -1.692874 | 1.000000 | 1.000000 | 0.086735 |
+-----+-----+-----+-----+-----+
| 4 | 100% | + | -1.085270 | 1.000000 | 0.553719 | 0.086735 |
+-----+-----+-----+-----+-----+
| 5 | 82% | - | -0.489517 | 1.000000 | 0.173554 | 0.035714 |
+-----+-----+-----+-----+-----+
Stage training time: 573.74
Number of used features: 5

```

Figura 45 Segunda etapa de entrenamiento

A medida que avanza en las etapas las características necesarias van aumentando y por tanto el tiempo para calcular va a ser mayor.

Una vez hayan transcurrido todas las etapas y se haya obtenido el clasificador correspondiente de cada una de ellas se procede a formar un solo clasificador, en el cual cada nodo pertenece a una característica.

```
<?xml version="1.0"?>
<opencv_storage>
<sillarueda type_id="opencv-haar-classifier">
  <size>
    50 50</size>
  <stages>
    <_>
      <trees>
        <_>
          <_>
            <feature>
              <rects>
                <_>
                  0 9 12 28 -1.</_>
                <_>
                  6 9 6 28 2.</_></rects>
              <tilted>0</tilted></feature>
              <threshold>-0.0127378301694989</threshold>
              <left_val>0.9140300154685974</left_val>
              <right_val>-0.8459600806236267</right_val></_></_>
            <_>
          </trees>
        </_>
      </stages>
    </_>
  </sillarueda>
</opencv_storage>
```

Figura 46 Nodo del clasificador

6.7. Pruebas de acceso a la página web

Con el fin de verificar el número de usuario que puede acceder a la página web en un determinado tiempo se utilizó la herramienta Load Impact.

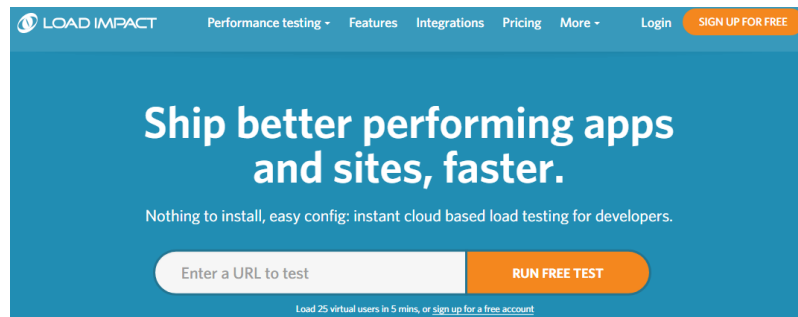


Figura 47 Herramienta Load Impact

Load Impact es una herramienta gratuita, permite ingresar la dirección de la página web a la que se desea realizar la prueba y se inicia el proceso.

Figura 48 Inicio de prueba

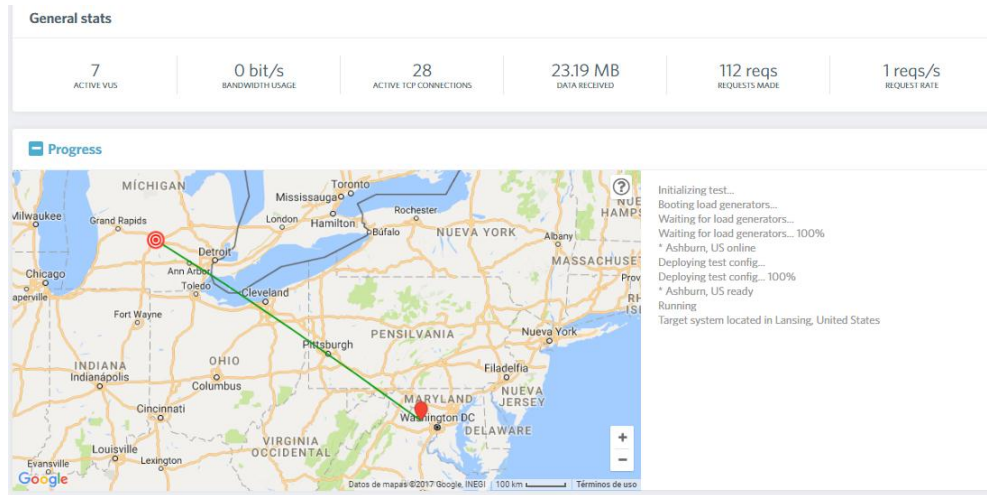


Figura 49 Ejecución de la prueba

La herramienta simula hasta 25 usuarios en tiempo determinado de 5 minutos, los resultados se muestran a continuación:

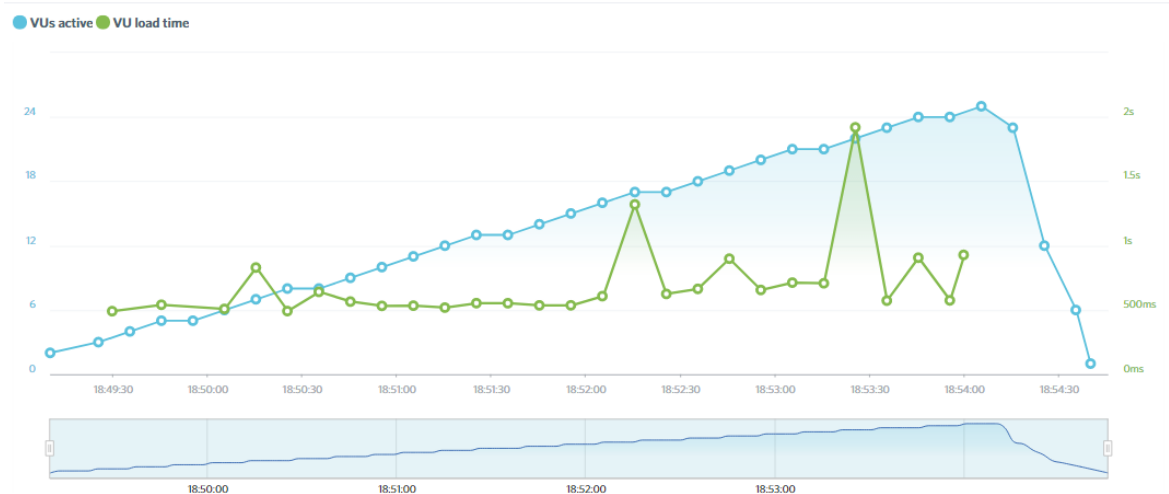


Figura 50 Prueba de acceso de usuarios a la página web

Como se muestra en la imagen los puntos de color azul representan el número de usuarios activos, y los puntos de color verde representan el tiempo (segundos) de respuesta del sistema.

7. Discusión

A continuación se detallan todas las actividades realizadas para dar cumplimiento a los objetivos planteados en el presente Trabajo de Titulación:

- Analizar y valorar las técnicas y herramientas de Visión Artificial usadas para el reconocimiento de personas, bicicletas y motos.

Para obtener los resultados planeados en este objetivo se realizó una investigación de conceptos, técnicas y algoritmos de reconocimientos de objetos, se describen algunas Técnicas de Visión Artificial como la de Descriptores en la cual se hace mención algunos algoritmos como por ejemplo: El algoritmo de patrones binarios locales, histogramas de bloques, entre otros. Se describe también la técnica de los Clasificadores en la cual se hace mención a algoritmos como: La Máquina de Vectores de Soporte, AdaBoost y Cascadas.

Finalmente, se hace una descripción de las herramientas Hardware y Software Libre utilizados para realizar aplicaciones de Visión Artificial, en la cual se hace una descripción detallada de cada una de ellas, lo cual nos sirvió para luego decidir cuáles son las más idóneas para el desarrollo de la aplicación.

- Analizar y diseñar la solución del sistema para el reconocimiento de personas con discapacidad, bicicletas y motos.

Para cumplir con este objetivo se utilizó la metodología ICONIX, la cual cuenta con las siguientes fases: Análisis de Requisitos, Análisis y Diseño Preliminar, Diseño e Implementación.

Para dar cumplimiento a la primera fase de la metodología se realizó el análisis de requerimientos en el que se describen los requerimientos funcionales y no funcionales, seguidamente se realizó un prototipado, el cual ayudó a tener una visión más clara del sistema, para posteriormente realizar un modelo de Casos de Uso.

Luego se realizó la descripción de cada Caso de Uso con su respectivo Diagrama de Robustez dando así por terminada la segunda fase de la metodología.

Para la tercera fase de la metodología se realizaron los Diagramas de Secuencia para cada Caso de Uso, a continuación se realizó el diagrama de clases en el cual se describe la estructura del sistema.

- Codificar el diseño elaborado para el reconocimiento de personas con discapacidad, bicicletas y motos.

Este objetivo se da cumplimiento en la cuarta fase de la metodología ICONIX que corresponde a escribir el código tal y como fue especificado en las fases anteriores, para el desarrollo de la aplicación se utilizó el IDE de programación QT Creator, el lenguaje de programación C++ y se utilizó la librería de Visión Artificial OpenCV.

- Realizar pruebas de funcionamiento del software en el reconocimiento de personas con discapacidad, bicicletas y motos.

Una vez realizada la aplicación se procedió a realizar las pruebas con el fin de verificar su correcto funcionamiento y que los resultados cumplan con todos los requerimientos antes planteados.

Se evaluó cada uno de los clasificadores (personas, bicicletas, motos, autos) con videos tomados en escenarios reales grabados en las diferentes calles de la ciudad de Loja.

8. Conclusiones

Una vez concluido el presente Trabajo de Titulación se puede establecer las siguientes conclusiones:

- Para obtener un clasificador potente se debe realizar el entrenamiento con un gran número de imágenes positivas y negativas, ya que a mayor número de imágenes se obtiene mayor precisión en la detección de objetos.
- La utilización de la librería OpenCV fue un gran acierto en este proyecto de Visión Artificial, la integración con C++ es muy fácil y posee un gran número de algoritmos para el procesamiento de imágenes lo cual facilitó la detección de objetos.
- Raspberry Pi es más eficiente trabajando con sus propios componentes, al momento de ejecutar la aplicación utilizando una cámara USB básica la cantidad de procesamiento es mayor que al utilizar el módulo de cámara propio de Raspberry Pi.
- La detección de objetos se realiza durante el día o en lugares con presencia de luz, en lugares oscuros o sin presencia de luz no se puede detectar correctamente los objetos, lo cual ocasiona ineficiencia en el sistema.
- Al ser un dispositivo con bajas características Raspberry Pi 3 no permite ejecutar la aplicación con más de 3 clasificadores al mismo tiempo, al utilizar 4 clasificadores conjuntamente la velocidad de procesamiento baja y por ende la captura de video en tiempo real es ineficiente.
- Para mayor precisión en el conteo de objetos, dentro de la región de interés se utiliza tres subregiones de interés ubicadas secuencialmente cada cierta distancia, el objetivo principal de estas subregiones es que ningún objeto pase sin ser contado, si un objeto no fue contado en la subregión 1 será contado en la subregión 2 o en la 3 y finalmente se saca un promedio con los contadores de cada subregión.

9. Recomendaciones

A continuación se detallan algunas recomendaciones a tener en cuenta al momento de desarrollar sistemas de Visión Artificial:

- Asegúrese que las versiones de las librerías utilizadas sean las mismas que se utilizaron en la etapa de desarrollo para evitar errores de compatibilidad.
- Para realizar el entrenamiento del clasificador lo más aconsejable es mantener una relación de dos imágenes positivas para una negativa y se debe tomar en cuenta el tamaño de los objetos que se van a detectar, por ejemplo, si se quiere detectar autos el tamaño de la imagen debe ser cuadrado y por otra parte si se requiere detectar personas el tamaño de la imagen debería ser rectangular ya que siempre el alto va a ser mayor al ancho.
- La ubicación de la cámara es importante para la captura de vídeo, de ello depende la certeza con que puede detectar y contar los objetos, evitando de esta manera la unión de los objetos por perspectiva y obteniendo además un tamaño relativo entre objetos.
- Para mejorar el rendimiento de Raspberry Pi 3 y crear un computador con mejores prestaciones es recomendable realizar un Cluster uniendo dos o más Raspberry Pi 3, lo cual además nos brindará beneficios como la tolerancia a fallos y ejecutar sin complicación aplicaciones que necesiten gran capacidad de proceso como si de un supercomputador se tratase.
- El proceso de entrenamiento del clasificador puede demorar varios días dependiendo la cantidad de imágenes y el número de etapas de entrenamiento por lo que es recomendable utilizar un computador con buenas prestaciones (procesador Core i5 o superior y memoria RAM de 4GB o más), para evitar inconvenientes de procesamiento o que se demora más tiempo del esperado.

10. Bibliografía

- [1] M. Alfonso, M. Cazorla y O. Colomina, Inteligencia artificial: modelos, técnicas y áreas de aplicación, Paraninfo, 2003.
- [2] Galeon, «Inteligencia Artificial,» [En línea]. Available: <http://laroboticadelmundo.galeon.com/>. [Último acceso: 12 Mayo 2016].
- [3] A. Ruiz, Sistemas de Percepción y Visión por Computador, Murcia, 2015.
- [4] A. Garcia Serrano, Inteligencia artificial : fundamentos, práctica y aplicacione, Rc Libros, 2012.
- [5] R. Pino, A. Gómez y N. d. A. Martínez, Introducción a la inteligencia artificial, Universidad de Oviedo, 2001.
- [6] Dpto. Electrónica, Automática e Informática Industrial , [En línea]. Available: <http://www.elai.upm.es/webantigua/spain/Asignaturas/Robotica/ApuntesVA/cap1IntroVA.pdf>.
- [7] «Vision Artificial e Interaccion sin mandos,» [En línea]. Available: <http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/3D/VisionArtificial/index.html>.
- [8] Vision Online SL, «Vision Online,» [En línea]. Available: <http://www.visiononline.es/es/actualidad-en-vision-artificial/la-vision-artificial-y-su-influencia-en-espana>.
- [9] Definición, «Definición Imagen Digital,» [En línea]. Available: <http://www.definicionabc.com/tecnologia/imagen-digital.php>. [Último acceso: 28 Abril 2016].
- [10] L. Ledda I, B. Enrique M y C. Carlos, «ALGORITMO PARA DETECCIÓN DE BORDES Y ULTERIOR,» [En línea]. Available: <http://www.cimec.org.ar/ojs/index.php/mc/article/viewFile/3955/3872>. [Último acceso: 28 Abril 2016].
- [11] H. Sánchez y a. Morales, «Docplayer,» 2009. [En línea]. Available: SISTEMA DE VISIÓN ARTIFICIAL PARA LA INSPECCIÓN, SELECCIÓN Y CONTROL DE CALIDAD DE FRESAS.. [Último acceso: 12 Mayo 2016].
- [12] Mycrosoft, «Cámara de vídeo analógica,» [En línea]. Available: <http://windows.microsoft.com/es-419/windows-vista/import-analog-video-to-your-computer-by-using-a-dv-camera>. [Último acceso: 12 Mayo 2016].

- [13] Sony, «Cámaras,» [En línea]. Available: <http://www.sony.es/electronics/camaras>. [Último acceso: 12 Mayo 2016].
- [14] Seguridad via IP, «Camaras Digitales,» [En línea]. Available: http://www.seguridadviaip.com/instalacion_de_camaras_de_seguridad_cctv_ip_wifi_dvr_diferencias.php. [Último acceso: 15 Mayo 2016].
- [15] IT Negocios, «Tecnología para ser mas productivos,» [En línea]. Available: <http://itnegocios.com/>. [Último acceso: 15 Mayo 2016].
- [16] Tecnología fácil, «Tecnología fácil,» [En línea]. Available: <http://tecnologia-facil.com/que-es/que-es-un-scanner/>. [Último acceso: 9 Mayo 2016].
- [17] Ordenadores y Portátiles, « Ordenadores y Portátiles,» 2014. [En línea]. Available: <http://www.ordenadores-y-portatiles.com/scanner.html>. [Último acceso: 9 Mayo 2016].
- [18] MedLine Plus, «MedLine Plus,» [En línea]. Available: <https://www.nlm.nih.gov/medlineplus/spanish/xrays.html>. [Último acceso: 9 Mayo 2016].
- [19] Radiological Society of North America, «Radiology Info,» 2016. [En línea]. Available: <http://www.radiologyinfo.org/sp/info.cfm?pg=bonerad>. [Último acceso: 9 Mayo 2016].
- [20] Ministerio de Educación, Cultura y Deporte, «Imágenes vectoriales y mapas de bits,» [En línea]. Available: <http://recursostic.educacion.es/observatorio/web/ca/software/software-general/293-lorena-arranz>. [Último acceso: 12 Mayo 2016].
- [21] A. Gonzales, «Aprende TIC,» [En línea]. Available: <https://sites.google.com/site/ticvalcarcel/optimizacion-de-imagenes-para-internet/tipos-de-imagenes-y-formatos>. [Último acceso: 12 Mayo 2016].
- [22] P. Wagner, «Local Binary Patterns,» 8 Noviembre 2011. [En línea]. Available: http://bytefish.de/local_binary_patterns/. [Último acceso: 27 Abril 2016].
- [23] J. Ramon, «Transformaciones basadas en el algoritmo,» [En línea]. Available: http://acceda.ulpgc.es/bitstream/10553/6728/5/Memoria_TFM.pdf. [Último acceso: 28 Abril 2016].
- [24] A. M. López, «Coursera,» [En línea]. Available: <https://es.coursera.org/learn/deteccion-objetos/lecture/8ahDw/l2-2-c-histograma-lbp-por-bloques>. [Último acceso: 28 Abril 2016].
- [25] Intel, «Histograma de Gradientes Orientados (HOG) descriptor,» [En línea]. Available: <https://software.intel.com/en-us/node/529070>. [Último acceso: 27 Abril 2016].
- [26] E. Valveny, «Coursera,» [En línea]. Available: <https://es.coursera.org/learn/deteccion-objetos/lecture/eczp1/l5-1-detector-de-caras-basado-en-filtros-de-haar-adaboost>. [Último acceso: 28 Abril 2016].

- [27] A. Armijos y P. Aponte, «Dspace Universidad Nacional de Loja,» 2015. [En línea]. Available: <http://dspace.unl.edu.ec/jspui/bitstream/123456789/11588/1/Aponte%20Rueda,%20Pedro%20Fernando,%20Armijos%20Armijos,%20Andr%C3%A9s%20Armanado.pdf>. [Último acceso: 12 Mayo 2016].
- [28] R. Paredes y J. Planells, «Repositorio UPV,» 2010. [En línea]. Available: <https://riunet.upv.es/handle/10251/8774?show=full>. [Último acceso: 12 Mayo 2016].
- [29] L. de la Cruz, «Repositorio Digital upv,» 15 Septiembre 2014. [En línea]. Available: <https://riunet.upv.es/bitstream/handle/10251/50406/TFMJoseluisdelaCruzGarrido.pdf?sequence=1>. [Último acceso: 12 Mayo 2016].
- [30] Scikit learn, «Scikit Learn,» [En línea]. Available: <http://scikit-learn.org/stable/modules/svm.html>. [Último acceso: 27 Abril 2016].
- [31] J. Perez, «Multi-clasificación Discriminativa de Partes Corporales basada en Códigos Correctores de errores,» [En línea]. Available: <http://www.maia.ub.es/~sergio/linked/pfctomas.pdf>. [Último acceso: 12 Mayo 2016].
- [32] A. M. López, « Generación de Candidatos – Ventana Deslizante,» [En línea]. Available: <https://es.coursera.org/learn/deteccion-objetos/lecture/kwUBz/l3-2-a-generacion-de-candidatos-ventana-deslizante>. [Último acceso: 28 Abril 2016].
- [33] M. García, «Diseño e implementacion de una herramienta de deteccion facial,» Junio 2009. [En línea]. Available: <http://tesis.ipn.mx/bitstream/handle/123456789/6111/DISENOIMPLEMFACIAL.pdf?sequence=1>. [Último acceso: 12 Mayo 2016].
- [34] E. Valveny, «Coursera,» 2016. [En línea]. Available: <https://es.coursera.org/learn/clasificacion-imagenes/lecture/NqKmX/evaluacion-del-rendimiento>. [Último acceso: 10 Octubre 2016].
- [35] C. Goutte y E. Gaussier, «Semanticscholar,» [En línea]. Available: https://pdfs.semanticscholar.org/e399/9a46cb8aaf71131a77670da5c5c113aad01d.pdf?_ga=1.219161168.874410388.1479265517. [Último acceso: 15 Octubre 2016].
- [36] Tecnocarreteras, «Tecnocarreteras,» 7 Noviembre 2011. [En línea]. Available: <https://www.tecnocarreteras.es/2011/11/07/los-semaforos-un-poco-de-historia-y-funcionamiento-13/>. [Último acceso: 12 Mayo 2016].
- [37] M. Martinez, «Semaforos Inteligentes,» [En línea]. Available: http://jeuazarru.com/wp-content/uploads/2014/10/semaforos_inteligentes.pdf. [Último acceso: 12 Mayo 2016].
- [38] L. M. E. Mantilla, «Sistema de Control Difuso de Semáforos para mejorar el trafico vehicular en el Centro Histórico de Trujillo,» 2014. [En línea]. Available: <http://www.inf.unitru.edu.pe/revistas/2014/8.pdf>. [Último acceso: 12 Mayo 2016].

- [39] Hipanavista, «Galeon,» [En línea]. Available: <http://eddyalfaro.galeon.com/geneticos.html>. [Último acceso: 8 Mayo 2016].
- [40] A. Champandard, «Reinforcement Learning,» 2002. [En línea]. Available: <http://reinforcementlearning.ai-depot.com/>. [Último acceso: 13 Mayo 2016].
- [41] I. Gonzales, J. Gonzales y F. Gomez, «Hardware Libre,» 6 Septiembre 2003. [En línea]. Available: <http://ftp1.nluug.nl/ftp/pub/ftp/os/Linux/doc/LuCaS/Presentaciones/200309hispalinux/8/8.pdf>. [Último acceso: 8 Mayo 2016].
- [42] A. Lazalde, J. Torres y D. Vila, «Floksocty,» 28 Enero 2015. [En línea]. Available: <http://floksocty.org/docs/Espanol/4/4.1.pdf>. [Último acceso: 8 Mayo 2016].
- [43] Arduino, «Arduino,» [En línea]. Available: <https://www.arduino.cc/en/Guide/Introduction>. [Último acceso: 8 Mayo 2016].
- [44] Raspberry Pi, «Raspberry Pi,» [En línea]. Available: <https://www.raspberrypi.org/>. [Último acceso: 8 Mayo 2016].
- [45] Banana Pi, «Banana Pi,» 2014. [En línea]. Available: <http://www.bananapi.org>. [Último acceso: 24 Mayo 2016].
- [46] BPI, «Banana Pi,» 2016. [En línea]. Available: <http://www.banana-pi.org/>. [Último acceso: 24 Mayo 2016].
- [47] Digi-Key, «BeagleBoard,» 2016. [En línea]. Available: <http://www.digikey.com/es/product-highlight/t/texas-instruments/beagleboard>. [Último acceso: 24 Mayo 2016].
- [48] BeagleBoard, «BeagleBoard,» 2016. [En línea]. Available: <http://beagleboard.org/beagleboard>. [Último acceso: 24 Mayo 2016].
- [49] R. Stallman, «Meta Biblioteca,» Diciembre 2004. [En línea]. Available: <http://bibliotecadigital.org/handle/001/144>. [Último acceso: 9 Mayo 2016].
- [50] Hispa Linux, «Hispa Linux,» [En línea]. Available: <http://hispalinux.es/SoftwareLibre>. [Último acceso: 9 Mayo 2016].
- [51] Oracle, «Java,» [En línea]. Available: <https://www.oracle.com/es/java/index.html>. [Último acceso: 10 Mayo 2016].
- [52] Deitel y J. Paul, *Cómo programar en C/C++ y Java*, Pearson Educación, 2004.
- [53] A. Marzal y I. Gracia, *Introducción a la programación con Python*, 2003.
- [54] Matlab, «Matlab,» [En línea]. Available: <http://www.mathworks.com>. [Último acceso: 24 Mayo 2016].

- [55] OpenCV, «OpenCV,» [En línea]. Available: <http://opencv.org/>. [Último acceso: 9 Mayo 2016].
- [56] M. y. o. Cazorla, «JavaVis: Una librería para visión artificial en Java,» *Rua*, p. 4, 2001.
- [57] MMGeo, «PyDocs,» 2010. [En línea]. Available: <http://pydocs.dim.uchile.cl/scipy/scipy.html>. [Último acceso: 13 Mayo 2016].
- [58] Hipertextual, «Internet de las Cosas,» 20 Octubre 2014. [En línea]. Available: <http://hipertextual.com/archivo/2014/10/internet-cosas/>. [Último acceso: 13 Mayo 2016].
- [59] T y N Media Group , «Cuando el Internet de lasCosas nos Alcance,» 12 Noviembre 2015. [En línea]. Available: <http://www.tyniot.com/cuando-el-internet-de-las-cosas-nos-alcance-estaras-listo/>. [Último acceso: 13 Mayo 2016].
- [60] A. Laura, G. Abraham y y. otros, «Aplicación de la metodología semi-ágil ICONIX para el desarrollo de software,» Julio 22 2014. [En línea]. Available: <http://www.laccei.org/LACCEI2014-Guayaquil/RefereedPapers/RP246.pdf>. [Último acceso: 2016 Junio 2].
- [61] F. Manuel y S. Maria, «Iconix,» 2004. [En línea]. Available: <http://www.uv.mx/personal/asumano/files/2010/07/iconix2.pdf>. [Último acceso: 2016 Junio 2].
- [62] W. Xiaoyu, H. Tony y Y. Shuicheng, «An HOG-LBP human detector with partial occlusion handling,» *Ieee Xplore Digital Library*, p. 8.
- [63] J. Velez, A. Moreno, A. Calle y L. Sanchez, *Visión por Computador*, 2002 y 2003.
- [64] D. Mery, «Visión por Computador,» 17 Agosto 2004. [En línea]. Available: <http://web.ing.puc.cl/~dmery/Prints/Books/2004-ApuntesVision.pdf>. [Último acceso: 12 Mayo 2016].
- [65] J. BERENGUEL, UF1844 - Desarrollo de aplicaciones web en el entorno servidor, Ediciones Paraninfo, S.A..
- [66] J. E. Kendall, *Análisis y diseño de sistemas*, Pearson Educación, 2005.
- [67] Mycrosoft, «Evolución de la programación visual.Herramientas RAD.,» 2016. [En línea]. Available: <https://support.microsoft.com/es-es/kb/550153>. [Último acceso: 17 Mayo 2016].
- [68] R. Noriega, J. Laínez, A. Durango y D. Ramos, *Curso de Ingeniería de Software*, IT Campus Academy, 2015.
- [69] K. Beck y C. Andres, *Extreme Programming Explained*, Addison-Wesley Professional, 2004.

11. Anexos

Anexo 1: Especificación de Requerimientos

1. Introducción

El presente documento de Especificación de Requerimientos describe detalladamente los requisitos que debe cumplir el sistema de reconocimiento de personas, bicicletas, motos y vehículos pequeños, el mismo que será utilizado en el proyecto de semáforos inteligentes.

1.1. Propósito

El objetivo de esta especificación es definir de una manera clara y precisa las funcionalidades y restricciones que tendrá el sistema de reconocimiento de personas, bicicletas, motos y vehículos pequeños.

1.2. Ámbito del Sistema

El sistema a desarrollar permitirá el reconocimiento y conteo de personas, bicicletas, motos, sillas de ruedas y vehículos pequeños, además, guardará datos como la hora, fecha y lugar de los objetos detectados en una Base de Datos para luego consultar mediante un navegador web los datos estadísticos guardados.

1.3. Definiciones, Acrónimos y Abreviaturas

1.3.1. Definiciones

Internet: Interconexión de redes informáticas que permite a los ordenadores o computadoras comunicarse directamente, es decir, cada ordenador de la red puede conectarse a cualquier otro ordenador de la red [1].

Navegador: Aplicación que opera a través de Internet, interpretando la información de archivos y sitios web para que estos puedan ser leídos [2].

1.3.2. Acrónimos

ERS: Acrónimo de Software Requirements Specifications(Especificación de Requerimientos de Software).

URL: Cadena de caracteres que identifica el tipo de documento, la computadora, el directorio y los subdirectorios en donde se encuentra el documento y su nombre.

WWW: World Wide Web, red informática mundial.

RF: Requerimiento Funcional.

RNF: Requerimiento no Funcional

1.4. Referencias

[1] Angelfire, «Internet,» 2016. [En línea]. Available: <http://www.angelfire.com/amiga2/k-baylon/Trabajo.htm>. [Último acceso: 8 Junio 2016].

[2] Mundo Digital, «Navegadores Web,» 2016. [En línea]. Available: <http://elidiomadelaweb.com/navegadores-web-2/>. [Último acceso: 8 Junio 2016].

1.5. Visión General del Documento

Este documento consta de tres secciones. Esta primera sección es la introducción y proporciona una visión general de la ERS. En la sección dos se da una descripción general del sistema, con el fin de conocer las principales funciones que debe realizar, los supuestos, restricciones y dependencias que afectan al desarrollo, sin entrar en excesivo detalle. Finalmente en la sección tres se definen de una manera detallada los requisitos que debe satisfacer el sistema.

2. Descripción General

2.1. Perspectiva del Producto

El sistema permitirá el reconocimiento y conteo de personas, bicicletas, motos y vehículos pequeños, el sistema constará de dos partes, la primera es la aplicación para el reconocimiento y conteo de personas, bicicletas, motos, sillas de ruedas y vehículos pequeños que funcionará en sistema operativo Linux y la segunda es la interfaz web que permitirá al usuario visualizar y generar reportes estadísticos, un requerimiento adicional es que el sistema deberá trabajar como un módulo del proyecto de semáforos inteligentes

2.2. Funciones del Producto

Las funciones que el sistema debe realizar son las siguientes:

Gestionar Usuarios: El sistema debe permitir crear usuarios del sistema y modificar los datos de usuario.

Consultar Estadísticas: El sistema permitirá visualizar reportes de los datos estadísticos guardados en la Base de Datos

Realizar Cuento: El sistema deberá detectar y contar personas, bicicletas, motos, sillas de ruedas y vehículos pequeños.

2.3. Características de los Usuarios

La función del sistema es detectar y contar objetos, enviar datos a una base de datos y generar reportes estadísticos. Debe existir un usuario el cual podrá generar reportes únicamente y un administrador el cual podrá generar reportes y agregar nuevos usuarios del sistema.

Características del Usuario y Administrador	
Tipo de Usuario	Usuario – Administrador
Formación	Bachiller
Habilidades	Conocimientos básicos en computación.
Actividades	Manejo del sistema.

2.4. Restricciones

El sistema deberá tener una interfaz que muestre video en tiempo real en un display y una interfaz para ser usada con internet, mismo que para el desarrollo del sistema se utilizará HTML5, CSS, Javascript y C++, se desarrollará en Linux con una base de datos MySQL y deberá tener un diseño e interfaz sencilla.

2.5. Suposiciones y Dependencias

- Se asume que los requisitos son estables.
- El sistema operativo necesario para el funcionamiento del sistema estará disponible en el o los equipos donde se instalará el sistema.
- Para la puesta en marcha y uso del sistema es necesario un puesto con conexión a Internet.

3. Requisitos específicos

3.1. Interfaces Externas

Los requisitos de interfaces externas son aquellos que ayudan al funcionamiento del sistema, los cuales se detallan a continuación:

3.1.1. Interfaces de Usuario

Al hablar de interfaz de usuario no referimos al entorno con el que trabajará el usuario. El entorno constará de botones, menús desplegables, mensajes informativos y diversas pantallas que permitirán al usuario interactuar con el sistema.

3.1.2. Interfaces de Hardware

El usuario deberá disponer de una máquina con acceso a internet para poder realizar las diferentes consultas, además se tomará en cuenta otras características que se enfocan en el proceso como se va a llevar a cabo la detección de objetos para lo cual se utilizará una cámara IP para la captura de video en tiempo real.

3.1.3. Interfaces de Software

El acceso al sistema para la generación de reporte será independiente del sistema operativo con el que trabaje, el usuario podrá acceder a través de un navegador de internet, la aplicación de reconocimiento y conteo de objetos se ejecutará en Sistema Operativo Linux el cual debe tener instalado C++ y las librerías de Visión Artificial OpenCV para el correcto funcionamiento de la aplicación.

3.1.4. Interfaces de Comunicación

El servidor, usuario y aplicación se comunicarán entre sí mediante protocolos estándares en internet, siempre que sea posible.

3.2. Requisitos Funcionales

Dentro de los requisitos funcionales tenemos los siguientes:

3.2.1. Requisito Funcional 1 “Gestionar Usuario” (RF-001)

- El usuario permitirá al administrador y usuario que estén creados en el sistema el acceso al mismo, para la utilización de las diferentes funcionalidades.
- El sistema permitirá al administrador crear diferentes usuarios del sistema.
- El sistema permitirá al administrador modificar la información de registro de usuarios.

3.2.2. Requisito Funcional 2 “Consultar Estadísticas” (RF-002)

- El sistema permitirá al usuario o administrador visualizar estadísticas o generar reportes estadísticos de los datos guardados anteriormente en la Base de Datos.

3.2.3. Requisito Funcional 3 “Realizar Conteo” (RF-003)

- El sistema debe detectar y contar personas, bicicletas, motos, sillas de ruedas y vehículos pequeños y enviar los datos a una Base de Datos.

3.3. Requisitos no Funcionales

3.3.1. Requisitos de seguridad

En cuanto a seguridad del sistema, el administrador será el encargado garantizar la seguridad de los datos utilizados por ésta, debido a que se utilizará claves de acceso al sistema.

3.3.2. Funcionalidad

El sistema deberá ejecutar las funcionalidades con las que se lo desarrolló y se deberá garantizar su funcionamiento durante todo el período de su ejecución.

3.3.3. Disponibilidad

Los datos deben estar disponibles para la explotación del usuario.

3.3.4. Entorno de Explotación

Para la ejecución de la aplicación se utilizará los siguientes elementos:

- Procesador de 2.2 GHz de velocidad como mínimo.
- Memoria RAM 4 GB como mínimo
- Disco Duro de 50 GB como mínimo
- C++
- OpenCV
- Una Cámara(Pi Camera)

3.4. Requisitos de Rendimiento

La rapidez de detección y conteo de objetos dependerá del procesador del equipo en el que se ejecute la aplicación, el número de usuarios simultáneos dependerá de la rapidez que deseemos, si la página es muy lenta debemos limitar el número de usuarios concurrentes.

3.5. Atributos del Sistema

Fiabilidad: El sistema debe ser confiable, la única persona que tendrá acceso a la modificación de datos será el administrador.

Seguridad: El sistema debe ser seguro, para el acceso modificación de datos se necesita autenticación.

Anexo 2: Creación de un Clasificador en Cascada

A continuación se describe el procedimiento paso a paso para la creación de un clasificador en cascada, los pasos a seguir son los siguientes:

1. Recolección de imágenes positivas y negativas para el entrenamiento.
2. Organización de imágenes negativas
3. Selección de imágenes positivas, en este caso se utilizó `objectmarker.exe`
4. Creación de un vector `.vec` basado en las imágenes positivas utilizando `createsamples.exe`.
5. La formación del clasificador utilizando `haartraining.exe`
6. Creación del archivo xml

Las herramientas necesarias para la construcción del clasificador se pueden descargar en el siguiente link: <https://www.cs.auckland.ac.nz/~m.Rezaei/tutoriales/Haar-Training.zip>.

Paso 1: Recolección de imágenes positivas y negativas para el entrenamiento.

Se debe seleccionar el número de imágenes positivas y negativas necesarias para realizar el entrenamiento, se debe tener en cuenta que mientras más imágenes se utilicen el clasificador será más potente y preciso.

Las imágenes positivas son aquellas que contienen el objeto que queremos detectar (por ejemplo si queremos detectar autos las imágenes deben contener autos), las imágenes negativas son aquellas que no contienen el objeto.

Paso 2: Organización de imágenes negativas

Las imágenes negativas se deben ubicar en el directorio `.....\training\negative` del archivo descargado y ejecutar el archivo `create_list.bat` ubicado en el mismo directorio.

El contenido del archivo `create_list.bat` es el siguiente:

```
dir /b *.jpg >bg.txt
```

El cual arroja el siguiente resultado:

```
neg-0001.jpg
```

```
neg-0003.jpg
```

```
neg-0004.jpg
```

neg-0005.jpg

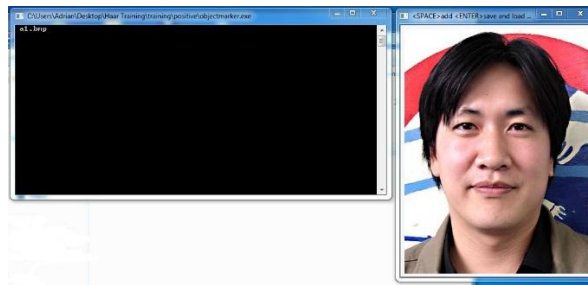
.....

Toda esta información de las imágenes negativas se utilizara al momento de realizar el entrenamiento del clasificador.

Paso 3: Selección de las imágenes positivas

En este paso se debe recortar las imágenes positivas para lo cual se utilizará *objectmarker*.

En la carpeta *...\training\positivo* hay un archivo llamado *objectmarker.exe* el cual utilizaremos para marcar los objetos en las imágenes positivas, al ejecutar dicho archivo se verá dos ventanas, en la una se muestra la imagen cargada y en la otra los datos de la imagen tal como se muestra a continuación.

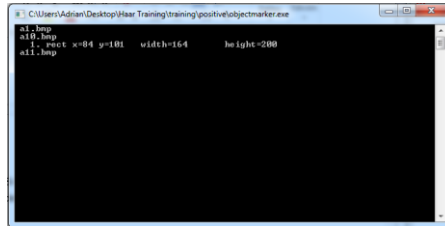


El procedimiento que debe seguir para seleccionar los objetos en cada imagen es el siguiente:

- a) Debe hacer clic en la parte superior izquierda del área donde se encuentra el objeto (por ejemplo una cara) y mantener el clic pulsado.
- b) Manteniendo el clic pulsado arrastre hacia debajo de tal manera que toda el área donde se encuentra el objeto quede seleccionada, tal como se muestra en la imagen siguiente:



- c) Si la selección ha sido correcta presione la tecla de ESPACIO, luego todos los datos del área seleccionada aparecerá en la ventana de datos como se muestra en la figura siguiente:



- d) Repita los pasos anteriores si hay varios objetos en la misma imagen.
 e) Cuando haya terminado de seleccionar el objeto o los objetos de la imagen pulse la tecla ENTER para cargar la siguiente imagen.
 f) Repetir todos los pasos anteriores hasta terminar con todas las imágenes.
 g) Cuando haya terminado con todas las imágenes debe presionar la tecla de ESCAPE, automáticamente se creará un archivo *info.txt*.

El archivo *info.txt* contiene la información de cada una de las imágenes como son: la dirección en la que se encuentra, el número de objetos que hay en cada imagen con sus respectivos datos como las coordenadas x e y en el que se encuentra y el alto y ancho de cada objeto.

Paso 4: Creación del vector de imágenes

En la carpeta *...\training* hay un archivo por lotes llamado *samples_creation.bat*, el cual tiene el siguiente contenido:

```
createsamples.exe -info positive/info.txt -vec vector/vector.vec -num 204 -w 24 -h 24
```

Los parámetros se describen a continuación:

- | | |
|--------------------------------|--|
| <i>-info positive/info.txt</i> | Ruta para el archivo de información de las imágenes positivas. |
| <i>-vec vector/vector.vec</i> | Ruta del archivo de vector de salida. |
| <i>-num 204</i> | Numero de imágenes positivas con las que se está trabajando. |
| <i>-w 24</i> | Ancho de los objetos. |

Se debe cargar el archivo por lotes *convert.bat* ubicado en ...*\cascade2xm* y el proceso finaliza.

Anexo 3: Instalación de OpenCV en Ubuntu

A continuación se describen los pasos necesarios para instalar OpenCV en Ubuntu 14.04.

1. Actualizar el sistema para luego instalar las dependencias.

```
sudo apt-get -y update
```

```
sudo apt-get -y upgrade
```

```
sudo apt-get -y dist-upgrade
```

```
sudo apt-get -y autoremove
```

2. Construir herramientas

```
sudo apt-get install -y build-essential cmake
```

3. Instalar librerías para interfaz gráfica

```
sudo apt-get install -y qt5-default libvtk6-dev
```

4. Librerías para medios de E/S

```
sudo apt-get install -y zlib1g-dev libjpeg-dev libwebp-dev libpng-dev libtiff5-dev  
libjasper-dev libopenexr-dev libgdal-dev
```

5. Librerías para video

```
sudo apt-get install -y libdc1394-22-dev libavcodec-dev libavformat-dev libswscale-dev  
libtheora-dev libvorbis-dev libxvidcore-dev libx264-dev yasm libopencore-amrnb-dev  
libopencore-amrwb-dev libv4l-dev libxine2-dev
```

6. Librerías para algebra

```
sudo apt-get install -y libtbb-dev libeigen3-dev
```

7. Instalamos Python

```
sudo apt-get install -y python-dev python-tk python-numpy python3-dev python3-tk  
python3-numpy
```

8. Instalamos Java

```
sudo apt-get install -y ant default-jdk
```

9. Documentación

```
sudo apt-get install -y doxygen
```

10. Instalamos la biblioteca OpenCV 3.1.0

```
sudo apt-get install -y unzip wget
```

```
wget https://github.com/Itseez/opencv/archive/3.1.0.zip
```

```
unzip 3.1.0.zip
```

```
rm 3.1.0.zip
```

```
mv opencv-3.1.0 OpenCV
```

```
cd OpenCV
```

```
mkdir build
```

```
cd build
```

```
cmake -DWITH_QT=ON -DWITH_OPENGL=ON -DFORCE_VTK=ON -  
DWITH_TBB=ON -DWITH_GDAL=ON -DWITH_XINE=ON -DBUILD_EXAMPLES=ON
```

```
..
```

```
make -j4
```

```
sudo make install
```

```
sudo ldconfig
```