



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

ESCOM

Trabajo Terminal

***“Cálculo del flujo vehicular mediante segmentación
de imágenes”***

Que para cumplir con la opción de titulación curricular en la carrera de:

**“Ingeniero en Sistemas Computacionales con especialidad en
Sistemas”**

Presentan:

**Blanco Álvarez Brenda
Dorantes Reyes Cristina
Gómez González David Jorge
Morales Báez Jorge**

Directores:

Dr. Benjamín Luna Benoso

Dr. Rolando Flores Carapia



México D.F. a 21 de Mayo de 201



INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO



No. registro: TT 2011-0011

Serie: Amarilla

Mayo de 2012

Documento Técnico:

“Cálculo del flujo vehicular mediante segmentación de imágenes”

Presentan:

Blanco Álvarez Brenda¹
Dorantes Reyes Cristina²
Gómez González David Jorge³
Morales Báez Jorge⁴

Directores:

Dr. Benjamín Luna Benoso
Dr. Rolando Flores Carapia

RESUMEN:

En la actualidad la contaminación ambiental se ha convertido en un problema de alto impacto, no solo en nuestro país, si no en el mundo. Con el fin de contribuir a una mejor distribución vial para reducir los índices de contaminación, se presenta una propuesta para el desarrollo de un sistema que permita determinar la cantidad de flujo vehicular y velocidad promedio existente en una avenida determinada a partir del análisis de las imágenes obtenidas de un sistema de videocámaras.

Palabras clave: Segmentación de imágenes, tratamiento digital de imágenes, flujo vehicular, consumo energético, contaminación ambiental.

México D.F. a 21 de mayo de 2012

¹ brenda.blanco.isc@gmail.com

² cristina.dorantes.isc@gmail.com

³ david.isc.escom@gmail.com

⁴ jorge.morales.isc@gmail.co



ESCUELA SUPERIOR DE CÓMPUTO
SUBDIRECCIÓN ACADÉMICA
DEPARTAMENTO DE FORMACIÓN INTEGRAL E INSTITUCIONAL



COMISIÓN ACADÉMICA DE TRABAJO TERMINAL

ING. APOLINAR FCO. CRUZ LÁZARO
PRESIDENTE DE LA COMISIÓN ACADÉMICA
DE TRABAJO TERMINAL
P R E S E N T E

Por medio del presente, informo que los alumnos que integran el **TRABAJO TERMINAL 2011-0011** titulado “**Cálculo del flujo vehicular mediante segmentación de imágenes**”, concluyeron satisfactoriamente su trabajo.

El empastado del Reporte Técnico Final y el Disco Compacto (CD) fueron revisados ampliamente por un servidor y corregidos, cubriendo el alcance y el objetivo planteados en el protocolo original y de acuerdo a los requisitos establecidos por la Comisión que Usted preside.

ATENTAMENTE

Dr. Benjamín Luna Benoso

Advertencia

“Este informe contiene información desarrollada por la Escuela Superior de Cómputo del Instituto Politécnico Nacional a partir de datos y documentos con derecho de propiedad y por lo tanto su uso quedar restringido a las aplicaciones que explícitamente se convengan.”

La aplicación no convenida exime a la escuela su responsabilidad técnica y da lugar a las consecuencias legales que para tal efecto se determinen.

Información adicional sobre este reporte técnico podrá obtenerse en:

En La Subdirección Académica de la Escuela Superior de Cómputo del Instituto Politécnico Nacional, situada en Av. Juan de Dios Bátiz s/n

Teléfono: 57296000 Extensión: 52000

Agradecimientos

Agradezco principalmente a Dios por permitirme llegar hasta este día, por guiar mi camino y llenarme de tantas bendiciones y amor. A mis padres Fernando Blanco y Rosalba Rosalía Álvarez porque además de ser excelentes personas, son los mejores padres que Dios me pudo dar, son mi soporte, mi fuerza y mi ejemplo a seguir. Gracias por su amor, su comprensión, su preocupación y aliento en todo momento. Mamá gracias por enseñarme lo que es la dedicación, la paciencia y la entrega con amor. Papá gracias por enseñarme el verdadero significado de la responsabilidad, la constancia y siempre pensar a futuro. A mis hermanas Alejandra, Fernanda y a mi hermano Moisés que aunque menores que yo, siempre me han demostrado que la edad no es un factor para aprender de alguien más, me enseñan el lado simple de la vida y sus detalles son los más grandes regalos que he podido tener. Gracias familia por ser la parte más importante de mi vida, los amo.

A Omar Vargas, por ser mi motivación para caminar hacia esa nueva vida que nos espera. Gracias por tu comprensión y amor.

A mis amigos: Blanca, David, Fernando, Abraham, Cristina, Nicolás, Rogelio y Jorge que en este trayecto estuvieron junto a mí en momentos felices, más aún en los difíciles, gracias por haber formado parte de esta historia y brindarme su cariño.

Y finalmente a todos mis profesores, en especial a mi profesora Jacqueline y al profesor Euler, gracias por compartir sus conocimientos y ser mucho más que solo profesores.

Brenda Blanco Álvarez

Agradezco a Dios por todas las bendiciones que me ha permitido vivir, a mis padres Cristina Reyes Galván, José Juan Dorantes Ríos y hermana Claudia Dorantes Reyes que son el motor que le da sentido a cada paso que doy, gracias por su inmenso amor y su entrega, por darlo todo por mí, por caminar a mi lado y darme la seguridad de que puedo llegar muy lejos, por enseñarme a seguir adelante y a superarme profesional y personalmente. Agradezco mi nina Ma. De la Luz Reyes Galván por su amor y apoyo incondicional, por estar siempre junto a mí en los momentos más felices y frente a las adversidades. Agradezco a mi abuelita Ma. De la Luz Galván Castro porque ha sido como una madre para mí y siempre me ha cobijado con cariño y ternura y me ha apoyado durante toda mi vida.

Agradezco a mis grandes amigos por permitirme recorrer este camino como hermanos, por el apoyo, el cariño, por creer en mí, por compartir las desveladas y superar juntos las dificultades.

Gracias a toda mi familia y amigos por ser parte de mis victorias y éxitos y sobre todo por apoyarme día a día para hacer posibles mis sueños y ver satisfechas mis metas.

Cristina Dorantes Reyes

A mis padres: A los cuales amo con todo mi corazón, porque gracias a ellos son quien soy en esta vida. Me han enseñado el verdadero valor de una familia, y que la vida en pareja no es fácil, pero trae muchas recompensas y felicidad si sabes escuchar, ceder, apoyar y amar como ustedes lo hacen cada día.

A mi mamá: Quien me enseña el valor de amar intensamente, de ser compasivo con las demás personas, a ser empático, y a nunca darle la espalda a una persona que necesita de mí, gracias madre por todo tu cariño y amor.

A mi papá: Quien me enseña a trabajar arduamente por conseguir lo que uno anhela, a ser fuerte y sano como un roble, a ser perseverante y siempre levantarse ante las caídas de la vida, a proteger a las personas que más quiero, a ser siempre honrado y decir la verdad, así como ser agradecido por lo que tengo, gracias padre por tu fuerza.

A mis hermanos Joaquín y Juan: Que definitivamente sin ellos mi vida no sería tan genial y maravillosa como lo es, enseñándome día a día grandes lecciones, y lo más importante que he aprendido de ellos, el verdadero amor de hermanos, ese amor tan fuerte que nunca tendrá final y que me da fuerza y valor para seguir adelante, que a pesar de cualquier cosa, siempre estaremos ahí para apoyarnos incondicionalmente.

A mis primos: Que son como hermanos para mí por que han estado para mí en los momentos precisos de mi vida, para ayudarme y también para compartir mis alegrías y tristezas.

Al compadre: Porque siempre estuvo ahí dándome palabras de aliento cuando más las necesitaba y haciéndome reír en los momentos más tristes de mi vida, eres un gran ejemplo y un gran amigo.

A mis amigos Jorge, Kristina, Rogelio, Oliver, Nico, Oscar y todos mis demás amigos de ESCOM: Quienes han hecho de ésta experiencia y de ésta parte de mi vida, grandiosa, inesperada y divertida, sepan que nunca habría querido compartir con otras personas ésta parte de mi vida.

A mi hermano Jake: Que en verdad ha sido como un hermano para mí, él me enseñó que aunque dos personas no tengan lazos de sangre, pueden ser verdaderos hermanos, siempre me has brindado tu amor de hermano incondicional, gracias por siempre estar ahí.

A mi ami: Que aunque no compartió conmigo ésta parte de mi vida, sé que siempre estará ahí para ayudarme y apoyarme, ami....gracias por ser mi mejor amiga.
Por último pero no menos importante

A Brenda: La persona más especial que he conocido a lo largo de ESCOM y de mi vida hasta ahora, tú fuiste mi ángel a lo largo de ésta parte de mi vida, siempre estuviste ahí no solo apoyándome, también regañándome y preocupándote por mí, todos los momentos que paso contigo día a día, los considero un regalo de Dios, gracias por entrar en mi vida, espero que siempre estés en ella.

Siempre los llevaré a todos ustedes en mi corazón...

David Jorge Gómez González

A Dios, por cada día de vida que me ha dado y por haberme dado una familia que ha sabido ser un apoyo y una guía durante toda mi vida.

A mis padres y mi hermano, Irma Báez Velasco, Gabriel Morales Cárdelas y James Morales Báez, por todo el apoyo incondicional y el esfuerzo que han compartido conmigo durante todos estos años, por enseñarme la importancia de la responsabilidad y el esfuerzo, por ser parte de mi vida y haberme llevado a ser lo que soy hoy en día.

A mis tíos Mario Báez Velasco, Armando Báez Velasco y Alejandra Guzmán Rangel, por siempre haber estado a mi lado, por haberme apoyado en todas mis decisiones y por todas las enseñanzas que han compartido conmigo.

Jorge Morales Báez

RESUMEN

En la actualidad la contaminación ambiental se ha convertido en un problema de alto impacto, no solo en nuestro país, si no en el mundo. Con el fin de contribuir a una mejor distribución vial para reducir los índices de contaminación, se presenta una propuesta para el desarrollo de un sistema que permita determinar la cantidad de flujo vehicular y velocidad promedio existente en una avenida determinada a partir del análisis de las imágenes obtenidas de un sistema de videocámaras.

Palabras clave

Segmentación de imágenes, tratamiento digital de imágenes, flujo vehicular, consumo energético, contaminación ambiental.

ABSTRACT

Nowadays, environmental pollution has become a problem of high impact, not only in our country, in the world too. To contribute getting better vial distribution for low levels of pollution, a proposal is develop a system to determine the magnitude of average speed and traffic flow existing in a particular avenue from the images obtained from a video system.

Key words

Segmentation of images, digital image processing, traffic flow, energy consumption, environmental pollution.

Índice de Contenido

CAPÍTULO 1 INTRODUCCIÓN.....	1
1.1 ANTECEDENTES.....	2
1.2 JUSTIFICACIÓN	3
1.3 OBJETIVO	4
1.3.1 <i>Objetivo General</i>	4
1.3.2 <i>Objetivos Particulares</i>	4
1.4 ORGANIZACIÓN DEL DOCUMENTO	4
CAPÍTULO 2 ESTADO DEL ARTE	6
2.1 PRINCIPALES TENDENCIAS.....	6
2.2 APLICACIONES FUNCIONALES	11
CAPÍTULO 3 CONCEPTOS BÁSICOS	13
3.1 INGENIERÍA DE TRÁFICO	13
3.1.1 PRINCIPIOS FUNDAMENTALES DEL FLUJO DEL TRÁNSITO	13
3.1.2 <i>Elementos del Flujo de Tránsito</i>	13
3.2 PROCESAMIENTO DIGITAL DE IMÁGENES.....	15
3.2.1 <i>Representación de una imagen digital</i>	15
3.2.2 <i>Elementos de percepción visual</i>	16
3.3 COMPONENTES DE UN SISTEMA DE VISIÓN.....	18
3.3.1 <i>Pre-procesamiento de la Imagen</i>	18
3.3.2 <i>Segmentación</i>	18
3.3.3 <i>Extracción de características (representación y descripción)</i>	19
3.3.4 <i>Reconocimiento e interpretación</i>	19
3.3.5 <i>Base de Conocimientos</i>	19
CAPÍTULO 4 ANÁLISIS	20
4.1 ESTUDIO DE FACTIBILIDAD	20
4.2 METODOLOGÍA DE DESARROLLO	22
4.2.1 <i>Paradigma de construcción de prototipos evolutivo</i>	22
4.3 DOCUMENTO DE VISIÓN	23
4.3.1 <i>Introducción</i>	23
4.3.2 <i>Posición del producto</i>	23
4.4 CASOS DE USO.....	26
4.4.1 <i>Diagrama de Casos de Uso</i>	26
4.4.2 <i>Descripción de Casos de Uso</i>	27
4.5 ANÁLISIS DE RIESGOS	28
4.5.1 <i>Descripción de los Riesgos</i>	28
4.5.2 <i>Plan de Prevención y Plan de Contingencia</i>	29
4.6 ANÁLISIS DE LENGUAJES DE PROGRAMACIÓN	31
4.6.1 <i>Descripción de los benchmarks considerados</i>	32
4.6.2 <i>Conclusión</i>	33

4.7 ANÁLISIS DE HERRAMIENTAS DE DESARROLLO	34
4.7.1 <i>Conclusión</i>	36
4.8 PRESUPUESTO DE COSTOS	37
4.8.1 <i>Costos de software</i>	37
4.8.2 <i>Costo de recursos y materiales</i>	37
4.8.3 <i>Gastos</i>	37
4.8.4 <i>Costo de esfuerzo</i>	38
4.8.5 <i>Coste total</i>	38
4.9 DEFINICIÓN DE PROTOTIPOS	39
CAPÍTULO 5 DISEÑO	40
5.1 CONTEXTO DEL SISTEMA FLUVER SYSTEM	40
5.2 DIAGRAMA DE BLOQUES	41
5.3 VISIÓN GENERAL	42
5.4 DIAGRAMA DE FLUJO DE DATOS	43
5.4.1 <i>DFD nivel 0</i>	43
5.4.2 <i>DFD nivel 1</i>	43
5.4.3 <i>DFD nivel 2</i>	44
5.5 DIAGRAMA DE PAQUETES	45
CAPÍTULO 6 PRIMER PROTOTIPO	46
6.1 ANÁLISIS	46
6.1.1 <i>Restricciones</i>	46
6.1.2 <i>Algoritmos, métodos y técnicas</i>	47
6.2 DISEÑO	51
6.2.1 <i>Diagrama de clases</i>	52
6.2.2 <i>Diagrama de secuencia</i>	53
6.3 DESARROLLO	55
6.3.1 <i>Adquisición del video</i>	55
6.3.2 <i>Método análisis del movimiento por diferencia de imágenes</i>	55
6.3.3 <i>Problemas comunes del algoritmo</i>	58
6.3.4 <i>Supresión del ruido residual</i>	62
6.4 IMPLEMENTACIÓN DE PRUEBAS	63
CAPÍTULO 7 SEGUNDO PROTOTIPO	68
7.1 ANÁLISIS	68
7.1.1 <i>Algoritmos</i>	68
7.2 DISEÑO	73
7.2.1 <i>Diagrama de clases</i>	73
7.2.2 <i>Diagrama de secuencia</i>	74
CAPÍTULO 8 TERCER PROTOTIPO	75
8.1 ANÁLISIS	75
8.1.1 <i>Gestor de Base de Datos</i>	75
8.1.2 <i>Comparación de MySql y SQL Sever</i>	76
8.1.3 <i>Comparación de SQL Server con Oracle</i>	77

8.1.4 Comparación de SQL Server con IBM DB2	77
8.1.5 Benchmarks de SQL Server 2008 R2.....	78
8.1.5 Elección del Sistema Gestor de Base de Datos.....	78
8.2 DISEÑO	79
8.2.1 Requerimientos de hardware y software para instalar SQL Server 2008 R2	79
8.2.2 Datos a almacenar y Modelo Relacional	80
8.2.3 Diccionario de Datos.....	81
8.2.4 Diagrama de Clases	83
8.2.4 Diagramas de Secuencia.....	84
8.3 DESARROLLO	87
8.3.1 Creación de la Base de Datos.....	87
8.3.2 Comunicación Qt Creator y SQL Server 2008 R2	90
8.3.3 Creación de un Origen de Datos	91
8.3.4 Desarrollo e implementación de la Interfaz.....	93
8.4 IMPLEMENTACIÓN DE PRUEBAS	96
8.5 INTEGRACIÓN.....	103
8.5.1 Diagrama de Clases	103
CONCLUSIONES	104
TRABAJO FUTURO	104
MEJORAS:	104
POSIBLES PROYECTOS:	104
• SISTEMA DE MONITOREO DE TRÁFICO MÓVIL.....	104
• SISTEMA DE OBTENCIÓN DE RUTAS ALTERNATIVAS (SORA)	105
• RECONOCIMIENTO DE AUTOMÓVILES:	105
REFERENCIAS	106

Índice de Tablas

Tabla 1. Resumen de productos similares.....	2
Tabla 2. Cantidad de gases generados en la Zona Metropolitana	3
Tabla 3. Factibilidad del proyecto	21
Tabla 4. Definición del problema	23
Tabla 5. Declaración de la posición del producto	24
Tabla 6. Requerimientos	24
Tabla 7. Prioridad de requerimientos	25
Tabla 8. Descripción de los casos de uso	27
Tabla 9. Descripción de Riesgos	28
Tabla 10. Tiempo de ejecución en segundos de cada benchmark.....	33
Tabla 11. Memoria en KB requerida para la ejecución de cada benchmark.	33
Tabla 12. Comparación de herramientas.....	36
Tabla 13. Costo de software.....	37
Tabla 14. Costo de recursos y materiales.....	37
Tabla 15. Gastos del Proyecto.....	38
Tabla 16. Costo de esfuerzo	38
Tabla 17. Costo total	38
Tabla 18. Prototipos	39
Tabla 19 Comparación de Algoritmos de Detección de Movimiento	51
Tabla 20. Prueba de adquisición de video.....	64
Tabla 21. Prueba de adquisición de máscara de movimiento	65
Tabla 22. Prueba de detección de automóviles por su sombra	66
Tabla 23. Prueba de segmentación de automóviles	67
Tabla 24. Comparación SQL Server 2008 R2 vs MySQL.....	77
Tabla 25. Comparación SQL Server 2008 R2 vs Oracle	77
Tabla 26. Comparación SQL Server 2008 R2 vs DB2	78
Tabla 27. Requerimientos de hardware y software para instalar SQL Server 2008 R2	80
Tabla 28. Diccionario de Datos Entidad Cámara	81
Tabla 29. Diccionario de Datos Entidad Estadística	82
Tabla 30. Controladores incluidos en Qt.....	90
Tabla 31. Prueba de conexión entre Qt y SQL Server 2008 R2	97
Tabla 32. Prueba de inserción de registros a la Base de Datos.....	98
Tabla 33. Prueba de actualización de registros en la Base de Datos	100
Tabla 34. Prueba para eliminar uno o más registros en la Base de Datos	102

Índice de Ilustraciones

Ilustración 1. Corte Horizontal del ojo humano	16
Ilustración 2. Componentes de un sistema de visión.....	18
Ilustración 3. Metodología Prototipado Evolutivo	22
Ilustración 4. Diagrama de Casos de Uso	26
Ilustración 5. Diagrama de contexto	40
Ilustración 6. Diagrama de bloques.....	41
Ilustración 7. Esquema general	42
Ilustración 8. DFD nivel 0.....	43
Ilustración 9. DFD nivel 1.....	43
Ilustración 10. Extracción de fotogramas. Nivel 2.....	44
Ilustración 11. Segmentación de la imagen. Nivel 2	44
Ilustración 12. Cálculo de estadísticas. Nivel 2	44
Ilustración 13. Diagrama de Paquetes	45
Ilustración 14 Pruebas del Algoritmo de Flujo Óptico	50
Ilustración 15. Diagrama de Clases: ProcesamientoImagen	52
Ilustración 16. Diagrama de secuencia.....	54
Ilustración 17: Código de ejemplo para la captura de un video y la adquisición de sus fotogramas.	55
Ilustración 18: Secuencia de imágenes donde el objeto A se desplaza a la derecha, mientras que el objeto B se mantiene estático.	56
Ilustración 19: (A) Imagen anterior; (B) Imagen actual, con el objeto A en diferente posición; (C) Comparación de las imágenes, las zonas con cambio son marcadas en negro.	57
Ilustración 20: Implementación del algoritmo básico, la detección del movimiento es mostrada en verde.	57
Ilustración 21: (A) Imagen anterior. (B) Imagen actual con cambios en el brillo de los objetos. (C) Comparación de las imágenes las zonas con cambio son marcadas en negro.	58
Ilustración 22: Imagen original a la izquierda e imagen contaminada con ruido a la derecha.....	59
Ilustración 23: (A) Imagen anterior. (B) Imagen actual contaminada con ruido. (C) Resultado de la comparación entre imágenes.....	59
Ilustración 24: (A) Video original; (B) Detección de movimiento básica; (C), (D), (E) y (F), Detección de movimiento considerando una vecindad de 3x3 y con un valor de tolerancia de 8, 16, 24 y 32 respectivamente.	61
Ilustración 25: (A) Detección de movimiento considerando una vecindad de 3x3 y un valor de tolerancia de 32; (B) Resultado de aplicar la operación de erosión a la imagen (A).	62
Ilustración 26. Diagrama de Clases: Cálculo-Estadísticas.....	73
Ilustración 27. Diagrama de secuencia.....	74
Ilustración 28. Correspondencia de Cardinalidad entre Entidades 1:N	80
Ilustración 29. Modelo Relacional.....	81

Ilustración 30 Diagrama de Clases de 3er Prototipo.....	83
Ilustración 31 Diagrama de secuencia Conexión a la base de datos.....	84
Ilustración 32 Diagrama de secuencia Registrar Cámara.....	84
Ilustración 34 Diagrama de secuencia Buscar Registro.....	85
Ilustración 33 Diagrama de secuencia Modificar datos	85
Ilustración 35 Diagrama de Secuencia Eliminar registro.....	86
Ilustración 36 Diagrama de Secuencia Obtener estadísticas	86
Ilustración 37. Scripts para crear una Nueva Base de Datos.....	87
Ilustración 38. Pantalla de inicio SQL Server 2008 R2.....	87
Ilustración 39. Pantalla de conexión SQL Server 2008 R2.....	88
Ilustración 40. Explorador de Objetos SQL Server	88
Ilustración 41. Abrir Script SQL Server	89
Ilustración 42. Administrador de orígenes de datos ODBC.....	91
Ilustración 43. Controlador para establecer Origen de Datos	92
Ilustración 44. Configuración del Origen de Datos parte 1	92
Ilustración 45. Configuración del Origen de Datos parte 2	93
Ilustración 46. Ajustes de Configuración Adicionales	93
Ilustración 47. Conexión a Base de Datos	94
Ilustración 48. Operaciones para insertar y actualizar registros de la tabla Cámara	94
Ilustración 49. Consultar Base de Datos.....	95
Ilustración 50 Diagra de clases Integración.....	103

Índice de Ecuaciones

Ecuación 1. Flujo vehicular	14
Ecuación 2. Velocidad media en el tiempo	14
Ecuación 3. Aproximación del fondo.....	47
Ecuación 4. Distancia Euclidiana del fondo	47
Ecuación 5. Calculo del Centro del Vehículo	48
Ecuación 6. Distancia de vehículos entre fotogramas	48
Ecuación 7. Gradientes en dirección 'X' y 'Y'	48
Ecuación 8. Algoritmo Laplacian of Gaussian.....	49
Ecuación 9: Análisis de movimiento por diferencia de imágenes.....	56
Ecuación 10. Metros Avanzados por Segundo	69
Ecuación 11. Relaciones pixel/metro	69
Ecuación 12. Pixeles avanzados por segundo	69
Ecuación 13. Pixeles avanzados por frame	69
Ecuación 14. Pendiente de la recta	70
Ecuación 15. Cantidad de Carriles	70
Ecuación 16. Velocidad	71
Ecuación 17. Número de Operaciones	72
Ecuación 18. Conversión de velocidad de px/s a km/hr	72

CAPÍTULO 1

INTRODUCCIÓN

El transporte en la ciudad de México se ha convertido en uno de los principales contaminantes y demandante de combustible, concentrando el 53% del consumo energético nacional [1].

Esta situación no solo afecta a la Ciudad de México, si no que se ha convertido en un problema a nivel mundial. Es por este motivo que países como Estados Unidos, Japón, Brasil, Chile, Canadá entre otros, han implementado sistemas de transporte inteligente (ITS¹, por sus siglas en inglés) para de esta manera mejorar el flujo vehicular, los tiempos de viaje, congestionamientos, etc. Lo que a su vez minimiza gasto de combustible y sus efectos ambientales.

La mayoría de los conductores al carecer de la información acerca del flujo vehicular en una vialidad de la ciudad, toman rutas con tránsito denso, lo que implica que el vehículo gaste más combustible del necesario para llegar a su destino y produzca un impacto negativo al medio ambiente.

Al conducir un vehículo se recomienda mantener la velocidad de circulación lo más uniforme posible. Si se evita el frenar y acelerar de manera continua, así como los constantes cambios de marcha, se ahorra hasta un 15% de combustible, dicho en otras palabras el tránsito denso aumenta hasta en un 15% el consumo de combustible. El hecho de conducir a muy bajas velocidades es igual de contraproducente que el hacerlo a una alta velocidad, ya que la mayoría de los vehículos ofrecen un rendimiento óptimo de combustible entre 60 y 80 km/h [2].

Debido al aumento en el problema de la congestión del tráfico urbano, existe una necesidad urgente para la introducción de tecnologías avanzadas y equipos para mejorar el estado de la técnica de control de tráfico.

Algunos de los sistemas de transporte inteligente que se han desarrollado hacen uso de una red de sensores para poder determinar la velocidad de los automóviles, otros como el proyecto de videocámaras no calibradas WSDOT² en Washington, EUA, hacen uso de tratamiento de imágenes para calcular la velocidad.

¹ Intelligent Transport System

² Sistema del Departamento de Transporte del Estado de Washington.

1.1 Antecedentes

La siguiente tabla muestra algunos de los sistemas que se han realizado en otros países enfocados a contribuir a la disminución de contaminantes.

Título	Año	País	Herramientas de desarrollo
PROMETHEUS (Program for a European Traffic Highest Efficiency and Unprecedented Safety)	1986-1987	Europa	Sistemas, basados en microelectrónica e inteligencia artificial, que facilitan el manejo autónomo del vehículo, y ayudan a tener un flujo de tráfico más seguro, eficiente y con menos contaminación [3].
ADVANCE (Advance Driver and Vehicle Advisory Navigation Concept)	1992	Estados Unidos en la Ciudad de Chicago, Illinois	Los vehículos son equipados con Asistentes de navegación móvil (MNA, por sus siglas en inglés) que calculan rutas convenientes, basados en los tiempos de viaje previstos que se obtienen a través de radio desde una computadora central [4].
AMTICS (Advanced Mobile Traffic Information and Communication System) RACS (Road Automobile Communication System)	1996	Japón	Sistemas de transporte inteligente con 974 videocámaras para vigilancia en carreteras, 74 centros de control, 57,537 detectores de vehículos ultrasónicos para medir la velocidad media, tiempo de ocupación y volúmenes de tráfico; 1,478 tableros de información de tráfico electrónicos [3,5].
WSDOT (Departamento de Transporte del Estado de Washington)		Estados Unidos Washington	Videocámaras no calibradas para detectar la velocidad del tráfico con un algoritmo capaz de calcular la velocidad de los vehículos que transitan las carreteras por medio de la detección de bordes de la autopista y el desvanecimiento y estiramiento de la imagen [6].
Detección de flujo vehicular basado en visión artificial	2007	Pereira, Colombia	Sistema encargado de detectar y clasificar vehículos y peatones a partir de secuencias de imágenes con el propósito de determinar el flujo vehicular y peatonal en las intersecciones de calles y avenidas utilizando descriptores de Fourier, filtrado homomórfico y procesamiento de imágenes [7].

Tabla 1. Resumen de productos similares.

1.2 Justificación

En situaciones de tránsito pesado los vehículos pueden estar parados pero con el motor encendido, avanzando tramos cortos y deteniéndose, cuando esto sucede el motor se encuentra funcionando en ralentí, lo que representa un consumo promedio de 0.4 a 0.7 litros/hora. Estas cifras no son altas, pero representan un consumo considerable si se acumulan a lo largo de la vida de la unidad.

La siguiente tabla muestra las cantidades de gases que se generan en la Zona Metropolitana del Valle de México, como consecuencia de las diferentes actividades que conlleva la quema de combustibles fósiles.

Compuesto	Cantidad
Monóxido de carbono	3041 Ton / día
Bióxido de carbono	64923 Ton / día
Compuestos orgánicos volátiles	370 Ton / día
NOx (Óxido de nitrógeno)	330 Ton / día

Tabla 2. Cantidad de gases generados en la Zona Metropolitana

El motor de combustión interna de un vehículo se constituye como un convertidor de energía química a mecánica, y por su escape se emiten gases tóxicos, tales como el monóxido (CO) y bióxido de carbono (CO₂); compuestos orgánicos volátiles; óxidos de nitrógeno (NO_x) o también conocido como óxido nítrico (NO) y dióxido de nitrógeno (NO₂), los cuales se generan en importantes cantidades como consecuencia del proceso de combustión que se lleva a cabo en el interior del motor [8].

En adición al problema de la contaminación, en México no se cuenta con un desarrollo completo de algún sistema automatizado que ofrezca información acerca del flujo vehicular a los automovilistas, tampoco se cuenta con redes de sensores y no se sabe mucho de sistemas basados en visión artificial que puedan facilitar la medición de la velocidad de los automóviles y por tanto la operación del tráfico en las calles y autopistas.

Se dice que una de las grandes problemáticas en la que la visión artificial puede ser una solución, es el monitoreo y vigilancia de tráfico [9,10]. Por lo cual existe la necesidad de tener sistemas de monitoreo y vigilancia de tráfico que superen las limitaciones del sistema de percepción humano y permitan que la cantidad de información que se procese e interprete no sea limitada como lo es usualmente. Por este motivo, se pretende dar una solución al problema de reconocer y medir el flujo vehicular utilizando las imágenes que proporcionan las cámaras de un sistema de video vigilancia y de esta manera dar pauta a la creación de la base de un sistema que pueda determinar el flujo de las calles y

autopistas e informe a los usuarios las mejores rutas que puede utilizar en caso de que ciertas avenidas estén congestionadas, agilizando su viaje, disminuyendo contaminantes, combustibles y costos.

1.3 Objetivo

1.3.1 Objetivo General

Implementar un sistema capaz de determinar la cantidad del flujo vehicular y velocidad promedio existente en una avenida en función de las imágenes obtenidas en un tiempo determinado.

1.3.2 Objetivos Particulares

- Investigar y analizar la teoría de análisis de imágenes.
- Recopilar y analizar algoritmos de detección de movimiento.
- Recopilar imágenes en video de vehículos en movimiento en una avenida.
- Determinar la velocidad promedio de los automóviles en una avenida mediante el tratamiento digital de imágenes.

1.4 Organización del Documento

- **Capítulo 1 Introducción** Se muestra una breve introducción de la problemática a la que se pretende dar solución y se despliegan algunos sistemas que ya han sido desarrollados. Se detalla la razón por la cual se desea desarrollar el sistema y se delimitan los objetivos del mismo.
- **Capítulo 2 Estado del Arte** Se detalla el desarrollo de Sistemas de Tráfico Inteligentes alrededor del mundo, así como el tipo de algoritmos que se han utilizado, comentarios y resultados de algunos de ellos. Muestra un panorama general de los resultados de investigación y estado actual de los conocimientos sobre técnicas y algoritmos para la detección de vehículos por medio de visión artificial.
- **Capítulo 3 Conceptos Básicos** Se explican los conceptos que se van a utilizar a lo largo del documento que son propios del procesamiento digital de imágenes.

FLUVER SYSTEM

- **Capítulo 4 Análisis** Se muestran los estudios de factibilidad, costo, herramientas y lenguajes de programación para realizar el sistema. Además se detalla la metodología utilizada y las ventajas que ofrece. Este capítulo cuenta con un documento de visión, el cual, debido a la metodología empleada, ayuda a brindar un panorama general de a dónde se quiere llegar con el desarrollo del sistema. Los resultados que se obtengan en cada prototipo ayudarán a saber cuál será el alcance final y real del sistema. Al final se desglosa una tabla con los alcances de cada prototipo.
- **Capítulo 5 Diseño** Se muestran diagramas que permiten a describir el funcionamiento del sistema de manera general, uno de los más importantes es el Diagrama de Flujo de Datos. No se muestran diagramas a detalle, ya que serán mostrados en el diseño de cada prototipo.
- **Capítulo 6 Primer Prototipo** Se detalla el Análisis, Diseño e Implementación de Pruebas del primer prototipo. El cual tiene como finalidad especificar los algoritmos de detección de movimiento, probarlos y elegir el que mejor se ajuste al sistema.
- **Capítulo 7 Segundo Prototipo** Se detalla el Análisis, Diseño e Implementación de Pruebas del segundo prototipo. El cual a partir del tratamiento y la segmentación del video, tiene la finalidad de calcular el flujo vehicular y la velocidad promedio de una avenida determinada.
- **Capítulo 8 Tercer Prototipo** Se detalla el Análisis, Diseño, Desarrollo e Implementación de Pruebas del tercer prototipo. En el cual se implementa una base de datos, con las estadísticas de velocidad y flujo de tránsito correspondientes a cada cámara existente en el sistema, que permite la administración de la información de cada registro.

CAPÍTULO 2 ESTADO DEL ARTE

Esta sección tiene la finalidad de ser un trabajo de revisión preliminar que resume y organiza temporalmente los resultados de investigación y estado actual de los conocimientos sobre técnicas y algoritmos para la detección de vehículos, o algunas de sus características, por medio de la visión artificial para el monitoreo en los Sistemas de Tráfico Inteligentes.

2.1 Principales tendencias

La detección de vehículos por medio de cámaras de vídeo es una de las más prometedoras tecnologías para colecciones de datos inalámbricas a gran escala y la implementación avanzada del control de tráfico y sistemas de gestión tales como la conducción/navegación de vehículos.

Debido al aumento en el problema de la congestión del tráfico urbano, existe una necesidad urgente para la introducción de tecnologías avanzadas y equipos para mejorar el estado de la técnica de control de tráfico. Sin embargo, la detección de vehículos es el eslabón más débil en la aplicación de los conceptos más sofisticados de control de tráfico que han surgido en los últimos años. La detección a través del procesamiento de imágenes de video es una de las más atractivas alternativas de las nuevas tecnologías, ya que implica la detección de vehículos y extracción de parámetros de tráfico en tiempo real de imágenes generadas por cámaras de vídeo con vistas a un escenario de tráfico.

El mayor inconveniente de los detectores es su limitación para medir algunos parámetros importantes y evaluar con precisión las condiciones del tráfico. Esto se debe a que en la tecnología utilizada sólo la presencia o ausencia de vehículos en los detectores se puede evaluar con suficiente precisión.

Parámetros de tráfico, como la velocidad, la composición, longitud de la cola, entre otros deben ser derivados de la presencia o el paso y requieren detección múltiple, lo que ocasiona aumento en los costos y problemas de fiabilidad. Además, los detectores comunes no tienen capacidades de vigilancia visual y su colocación no es flexible ya que detectan el tráfico sólo en puntos fijos.

FLUVER SYSTEM

Un sistema de visión artificial para detección de vehículos se compone de una cámara electrónica con vistas a un largo tramo de la carretera, a partir de las imágenes recibidas por la cámara, un microprocesador determina la presencia o el paso de vehículos y deriva otros parámetros en tiempo real.

La investigación en el área de visión artificial para detección de vehículos se inició a mediados de la década de 1970 en los Estados Unidos y el extranjero, sobre todo en Japón, Francia, Australia, Inglaterra y Bélgica.

A pesar de los esfuerzos más importantes del mundo para desarrollar un sistema de visión artificial para la vigilancia y control del tráfico, un dispositivo fiable en tiempo real con las capacidades y el rendimiento necesarios para las aplicaciones prácticas ha sido difícil de alcanzar. Debido a esto, actualmente no existen instalaciones prácticas conocidas, las pocas que existen son experimentales [11].

La medición en tiempo real de los parámetros de la cola de tráfico es necesaria en muchas situaciones de tránsito, como accidentes, monitoreo de la congestión y ajuste de los tiempos de semáforos.

El algoritmo de detección de colas propuesto por Siyal & Fathy en 1995 consiste en una detección de movimiento y una operación de detección de vehículos, ambas basadas en la extracción de bordes de la escena. El algoritmo es capaz de medir diferentes parámetros como longitud, período de aparición y pendiente de la cola.

El principal problema con respecto a cualquier método de procesamiento de imágenes aplicado a sistemas de detección de tráfico se debe al hecho de que las imágenes del mundo real se van a procesar en tiempo real. En una escena del mundo real, la variación de las condiciones de iluminación, forma o tamaño de los vehículos, la geometría de la escena, la oclusión e incontrolables movimientos, son serias dificultades para medir los diferentes parámetros.

El objetivo principal del algoritmo consiste en medir con precisión los parámetros de la cola, los cuales pueden dar información más valiosa, que el estado de la cola, a los ingenieros y controladores de tráfico en muchas situaciones de tránsito. El algoritmo propuesto consta de dos operaciones, una detección de movimiento y una técnica de detección de vehículos.

Algoritmo de detección de movimiento

Un método simple para la detección de movimiento se basa en restar dos imágenes consecutivas y aplicar operadores de eliminación de ruido. En este método, el histograma de las partes claves de la región de los marcos se analiza mediante la comparación con un valor umbral para detectar el movimiento. Para reducir la cantidad de datos y eliminar los

FLUVER SYSTEM

efectos del movimiento de la cámara, la región clave tiene que tener por lo menos un perfil de 3 píxeles de la imagen a lo largo de la carretera, una operación de filtrado de mediana se aplica en primer lugar a la región clave (perfil) de cada fotograma y un perfil de un píxel de ancho se extrae. Entonces, la diferencia de dos perfiles se compara para detectar el movimiento.

Algoritmo de detección de vehículos

Una operación de detección de vehículos se aplica en el perfil de la imagen sin procesar, este método es sensible a las variaciones de la iluminación ambiental y por lo tanto no es adecuado para aplicaciones del mundo real. Se basa en la aplicación de operadores de detección de bordes, ya que son menos sensibles a la variación de la iluminación ambiental y se han utilizado para la detección de objetos en movimiento en las aplicaciones de pantalla completa [12].

Un problema importante asociado con la visión en tiempo real, aplicada a la recopilación de datos del tráfico, es la exigencia de alta potencia de cálculo [13]. En aplicaciones de tráfico, la variación de la iluminación ambiental, sombras, oclusión, y el movimiento de vehículos complica aún más la tarea. Una revisión de la literatura indica que una gran variedad de algoritmos han sido desarrollados para el tráfico de recolección de datos y control.

En los Estados Unidos, el sistema de detección de área amplia (WADS por sus siglas en inglés *Wide-Area Detection System*) fue desarrollado sobre la base de la diferenciación de fondo y un método de actualización de fondo exponencial [14]. El sistema WADS es capaz de medir el volumen de tráfico y la velocidad de los vehículos.

En Inglaterra, el grupo de investigación de tráfico mediante el procesamiento de imágenes (TRIP por sus siglas en inglés *Traffic Research using Image Processing*) desarrolló un sistema de basado en tuberías para la medición en off-line del número y velocidad de los vehículos, con un fotogramas completo y el enfoque de diferenciación de fondo [15].

Una versión actualizada de TRIP se desarrolló basada en un ordenador 80 386 por Dickinson y Wan (1989). El sistema parece tener un mejor rendimiento que la versión anterior. Otro grupo de Inglaterra desarrolló un sistema de análisis de tráfico con un bajo costo de procesamiento de la imagen (TULIP por sus siglas en inglés *Traffic analysis Using a Low cost Image Processing*), el cual es capaz de medir diferentes parámetros de tráfico [16]. Este sistema utiliza dos métodos para la detección de vehículos dentro de una ventana definida por el usuario. El primer método aplica un umbral en el nivel de escala de grises y un valor predefinido. El segundo método consiste en utilizar varios umbrales para poner de relieve la mayor parte del vehículo, incluyendo sombras, justo debajo del borde delantero de los vehículos.

FLUVER SYSTEM

En Bélgica se ha desarrollado un sistema de sensores de tráfico asistidos por computadora (CATS por sus siglas en inglés *Computer-Aided Traffic Sensor*). CATS utiliza la técnica de diferencia de fondos y procesa un subconjunto muy pequeño de un carril, a unos 20 píxeles. El sistema es capaz de contar los vehículos y medir su velocidad y longitud [14].

En Japón, algunos sistemas han sido desarrollados, incluyendo el controlador y escáner de datos de una imagen para detectar los vehículos (IDSC por sus siglas en inglés *Image Data Scanner and Controller*) [17], [18]. En este método, los píxeles de la imagen actual se restan de los puntos de muestreo (fondo), y la diferencia se compara con un umbral de detección de objetos. Una técnica exponencial se utiliza para actualizar periódicamente los puntos de muestra.

Técnicas de detección de bordes

La detección de vehículos basada en detección de bordes es más eficaz que las técnicas de diferenciación de fondo, ya que la información del borde sigue siendo importante a pesar de la variación de la iluminación ambiental. El análisis de imágenes de tráfico ha demostrado que las varias superficies y diferentes partes y colores de un vehículo crean bordes significativos, que pueden ser detectados cuando se está pasando a través de una ventana.

Muchos algoritmos de detección de bordes se han desarrollado basados en operadores de gradiente o en métodos estadísticos. Los operadores de gradiente generalmente son muy sensibles al ruido y se utilizan junto con operadores de filtrado para reducir el efecto del ruido.

Un enfoque estadístico es incapaz de detectar los bordes delgados. Los detectores de bordes morfológicos han demostrado ser más eficaces que las técnicas convencionales basadas en gradiente [19]

Dailey, Cathey & Pumrin (2000) afirman que la calibración exacta no es necesaria para estimar la velocidad. En cambio, se utilizan 1) las relaciones geométricas inherentemente disponibles en la imagen, 2) algunos supuestos de sentido común que reducen el problema a una dimensión geométrica (1D), 3) la diferencia entre fotograma para aislar los bordes en movimiento y dar seguimiento a los vehículos, y 4) los parámetros de la distribución de las longitudes de los vehículos.

Las técnicas del procesamiento de imágenes se han aplicado a las escenas de tráfico para una gran variedad de propósitos, incluyendo: detección del tamaño de la cola, de incidentes, clasificación de vehículos, y el conteo de vehículos.

Tareas necesarias para obtener la velocidad desde imágenes no calibradas: 1) obtener imágenes secuenciales, 2) identificar los vehículos en movimiento en las imágenes secuenciales, 3) realizar un seguimiento de los vehículos entre las imágenes, 4) estimar de

FLUVER SYSTEM

forma dinámica el factor de escala en metros por pixel, y 5) cálculo de la velocidad por medio de la distancia recorrida y la demora entre fotogramas [20].

La detección de vehículos por medio de la visión artificial es una tarea crítica para el monitoreo en los modernos Sistemas de Tráfico Inteligentes (ITS por sus siglas en inglés *Intelligent Traffic Systems*). Debido a la naturaleza del ángulo de las cámaras de vigilancia de tráfico instaladas en la realidad, la detección de vehículos tiene que lidiar con un problema fundamental - la oclusión. La oclusión se vuelve mucho más grave en las intersecciones cuando los vehículos están haciendo cola esperando la luz verde. La abundante y constante oclusión en las escenas de tráfico ha hecho que la mayoría de los métodos tradicionales de detección de vehículos no sean eficaces.

En lugar de detectar el vehículo en su conjunto, se propone un algoritmo de detección de vehículos basados en la coincidencia de parabrisas. Si existen obstrucciones en una vista frontal de bajo ángulo de la cámara, el parabrisas es probablemente la parte menos afectada en cada vehículo, por lo que se puede utilizar como una característica confiable. Mediante la detección directa del parabrisas, el algoritmo logra solidez a la oclusión. Junto con la calibración de la cámara, el seguimiento de vehículos y coincidencias de forma y borde, el sistema es capaz de proporcionar una estimación fiable del estado del tráfico.

Para facilitar el seguimiento y control del tráfico, los ITS por lo general requieren datos de entrada, tales como densidad del tráfico, longitud de la cola, número de vehículos, posición del vehículo, velocidad del vehículo, tipo de vehículo, etc., provenientes de un componente de análisis de algunas escenas del tráfico (TSA por sus siglas en inglés *Traffic Scene Analysis*). Cuanto más un componente de la TSA pueda ofrecer, se logrará una mejor estimación del estado del tráfico.

En las últimas dos décadas, muchos métodos basados en visión artificial para la detección de vehículos en el contexto de la vigilancia del tráfico han sido propuestos. Con base en sus planteamientos técnicos pueden dividirse en seis categorías: la sustracción de fondo, la segmentación del plano, el detector bajo la supervisión de aprendizaje, la detección de partes sobresalientes del vehículo, el modelo de bordes coincidentes, puntos de seguimiento y agrupación [21].

Gu, Yendo, Tehrani, Fujii, y Tanimoto (2011) afirman que los sistemas de reconocimiento de señales de tránsito pueden ser aplicados para ayudar a los conductores y mejorar el nivel de seguridad, y proponen un sistema activo con cámara de doble focal para obtener una imagen de alta resolución de una señal de tránsito. El sistema está equipado con una cámara de gran ángulo. Para que el sistema propuesto sea capaz de capturar correctamente una imagen de alta resolución de la señal de tránsito, la forma, color y la relación entre los marcos se utilizan conjuntamente en la detección. Los resultados del experimento demuestran que el método propuesto hace que el sistema de cámara activa

FLUVER SYSTEM

con doble focal, instalado en un automóvil, trabaje de forma efectiva aún cuando se mueve en el camino.

Con el aumento de los automóviles, el estado del tráfico se agrava y hace más difícil la conducción. Para aligerar la carga de los conductores se propone implementar un Sistema de asistencia al conductor (DAS por sus siglas en inglés *Driver Assistance System*) el cual activamente recopila la información del medio ambiente alrededor de los automóviles, y ofrece condiciones potencialmente riesgosas para la atención del conductor tan pronto como sea posible.

Las señales de tráfico son la guía principal de transporte por carretera. Ofrecen información sobre las carreteras a los participantes en el transporte y las advertencias sobre situaciones potencialmente peligrosas [22].

2.2 Aplicaciones Funcionales

Autoscope, utilizado para la detección de incidentes en una sección de la carretera interestatal 394 EU [6].

Características de Autoscope:

- ✓ Adaptación automática a una amplia variedad de escenarios de carretera sin marcas de referencia.
- ✓ Operación aún en presencia de sombras, cambios de iluminación y reflejos.
- ✓ Funcionamiento en condiciones de congestión de tráfico y vehículos detenidos.
- ✓ Colocación arbitraria de cualquier tipo de detector en cualquier configuración en cualquier lugar dentro del campo de visión de la cámara.
- ✓ Rentabilidad, operación en tiempo real, y capacidad de programación.

Algoritmo para calcular la velocidad utilizando una secuencia de imágenes de vídeo desde una cámara sin calibrar. Las cámaras en las carreteras no suelen instalarse de manera que fácilmente puedan ser calibradas, y por lo general son utilizadas por los operadores quienes pueden inclinar y dar aumento con un joystick para cambiar la calibración de la cámara. La combinación de cámaras móviles y la falta de calibración hacen que la estimación de la velocidad sea un desafío.

El algoritmo está diseñado para crear una velocidad promedio en el tiempo. Hipótesis que simplifican el problema:

1) La velocidad de los vehículos es una magnitud finita. La velocidad de un vehículo tiene sus límites físicos y legales.

FLUVER SYSTEM

- 2) El movimiento del vehículo es suave. No hay cambios bruscos de dirección en el intervalo de tiempo (200 ms) entre los fotogramas de la secuencia de imágenes.
- 3) El movimiento es limitado al plano de la carretera. El seguimiento de vehículos en la secuencia de imágenes es un problema de 1-D.
- 4) El factor de escala (metros por píxel) varía linealmente a lo largo de la dirección de desplazamiento del vehículo.
- 5) Las longitudes de los vehículos en las imágenes son medidas de una distribución de vehículos de longitud conocida.

Con dichas hipótesis, los vehículos son tratados como si viajaran en una dimensión a lo largo de una línea recta en la imagen. Se realiza un seguimiento a los vehículos a través de las imágenes para obtener factores de escala que estimen la distancia del mundo real representada por píxeles en diversos lugares de la imagen. Utilizando una función lineal para adaptarse a los factores de escala empírica, es posible calcular la distancia recorrida en el mundo real. La combinación de la distancia recorrida con la velocidad de fotogramas conocidos nos permite estimar la velocidad [20].

CAPÍTULO 3

CONCEPTOS BÁSICOS

3.1 Ingeniería de Tráfico

La ingeniería de tráfico es una rama de la ingeniería civil que se encarga de planear, diseñar y organizar la operación del tráfico en calles y autopistas, con el fin de obtener una movilidad segura y eficiente [23].

3.1.1 Principios fundamentales del flujo del tránsito

La teoría del flujo de tránsito consiste en el desarrollo de relaciones matemáticas entre los elementos primarios de un flujo vehicular tales como flujo, densidad y velocidad. Estas relaciones ayudan al ingeniero de tránsito para la planificación, diseño y evaluación de la efectividad de la implementación de las medidas de ingeniería de tránsito en un sistema de carreteras.

La teoría del flujo de tránsito se usa en el diseño, para determinar las longitudes adecuadas de carril, para almacenar a los vehículos que dan vuelta a la izquierda en carriles separados para realizar esta maniobra, la demora promedio de las intersecciones o cruces y las áreas de incorporación al tránsito mediante rampas en los viaductos, así como cambios en el nivel de comportamiento del viaducto, debido a la instalación de dispositivos de control vehicular mejorados en las rampas. Otra aplicación importante de la teoría del flujo de tránsito es la simulación, en la cual se utilizan algoritmos matemáticos para estudiar las complejas interrelaciones, que existen entre los elementos de un flujo vehicular o red vehicular y para estimar el efecto de los cambios en el flujo del tránsito, sobre factores tales como accidentes, tiempo de viaje, contaminación del aire y consumo de gasolina [24].

3.1.2 Elementos del Flujo de Tránsito

Los elementos primarios del flujo del tránsito son el flujo, densidad y velocidad [24].

El flujo (q) es la tasa horaria equivalente a la cual transitan los vehículos por un punto, en una carretera durante un periodo menor a 1 hora. Puede determinarse mediante:

FLUVER SYSTEM

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

Ecuación 1. Flujo vehicular

donde:

n= el número de vehículos que transitan por un punto en el camino en T segundos,
q= el flujo horario equivalente.

La densidad (k), se define como el número de vehículos que ocupan un segmento de longitud específica en una vía de tal manera que esta densidad está dada en vehículos por kilómetro, este valor depende del segmento considerado y del tiempo en el que se realizó la medición [23].

La velocidad (u), es la distancia recorrida por un vehículo durante una unidad de tiempo. Puede expresarse en millas por hora (mi/h), kilómetros por hora (km/h) o pies por segundo (pies/segundo).

La velocidad media en el tiempo (\bar{u}_t), la velocidad media de recorrido se calcula tomando una porción de la vía, y dividiéndola entre el tiempo medio que tardan los vehículos en recorrerla, en este cálculo están incluidas las demoras de tráfico por detenciones completas o congestión del mismo. La velocidad media en el tiempo se calcula mediante:

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

Ecuación 2. Velocidad media en el tiempo

donde:

n= número de vehículos que transitan por un punto de la carretera.
 u_i = velocidad del vehículo i ésimo (pies/segundo)

3.2 Procesamiento Digital de Imágenes

El desarrollo de los métodos de procesamiento digital de imágenes tiene su origen en dos áreas principales de aplicación: el mejoramiento de la información pictórica para la interpretación humana, y el procesamiento de datos de la imagen para la percepción de máquina autónoma en el que se incluyen etapas de transmisión y/o almacenamiento de estos datos.

La herramienta usada en el tratamiento digital de las imágenes son las matemáticas; los conceptos que se verán son básicos. La computadora y los algoritmos que se implementan sobre éstas también tienen un papel muy importante en la manipulación de las imágenes.

3.2.1 Representación de una imagen digital

El término "imagen monocromática" o imagen simplemente, se refiere a una función de intensidad de luz bidimensional $f(x,y)$, donde x e y indican las coordenadas espaciales y el valor de f en cualquier punto (x,y) es proporcional a la luminosidad (o nivel de gris) de la imagen en dicho punto.

Una imagen digital es una imagen (función) $f(x,y)$ que ha sido discretizada tanto en coordenadas espaciales como en luminosidad. Una imagen digital puede ser considerada como una matriz cuyos índices de renglón y columna identifican un punto (un lugar en el espacio bidimensional) en la imagen y el correspondiente valor de elemento de matriz identifica el nivel de gris en aquel punto. Los elementos de estos arreglos digitales son llamados elementos de imagen o píxeles.

En el tratamiento de imágenes se pueden distinguir tres etapas principales:

1. Adquisición de la imagen
2. Procesamiento de la imagen
3. Presentación al observador

La adquisición de la imagen está a cargo de algún transductor o conjunto de transductores que mediante la manipulación de la luz o de alguna otra forma de radiación que es emitida o reflejada por los cuerpos, se logra formar una representación del objeto dando lugar a la imagen. Ejemplos: el ojo humano, sensores de una cámara fotográfica o de vídeo, tomógrafos.

Es importante saber que durante la etapa de adquisición, los transductores agregan ruido a la imagen. Además del ruido, los transductores poseen una resolución limitada, lo cual repercute en la apreciación de dicha imagen. El procesamiento digital de la imagen consiste en eliminar la mayor cantidad de ruido que se le agrega durante la adquisición así como también mejorar las características de dicha imagen como: definición de contornos, color, brillo, etc., valiéndose de procedimientos y herramientas matemáticas. En esta etapa se encuentran también técnicas de codificación para el almacenamiento o bien para la transmisión.

FLUVER SYSTEM

La presentación al observador consiste en el método empleado para exponer la imagen la cual puede ser impresa o por medios electrónicos como la televisión, el monitor de una computadora, o algún otro medio. Para la presentación de la imagen se deben considerar ciertos aspectos de percepción humana así como las velocidades de despliegue del dispositivo utilizado [25].

3.2.2 Elementos de percepción visual

Estructura del ojo humano

El ojo humano puede considerarse como una disposición de doble lente positiva que forma una imagen real en una superficie fotosensible. La *ilustración 1* muestra un corte horizontal simplificado del ojo humano. El ojo humano es una masa gelatinosa casi esférica contenida dentro de una estructura dura y flexible, la esclerótica. El ojo es parecido a una esfera, con un diámetro promedio de 20 mm. Excepto por la porción frontal o córnea que es transparente, la esclerótica es blanca y opaca. Sobresaliendo del cuerpo de la esfera está la córnea (que está ligeramente reduciendo así la aberración esférica) sirve como primer y más fuerte elemento convexo del sistema de lentes [26].

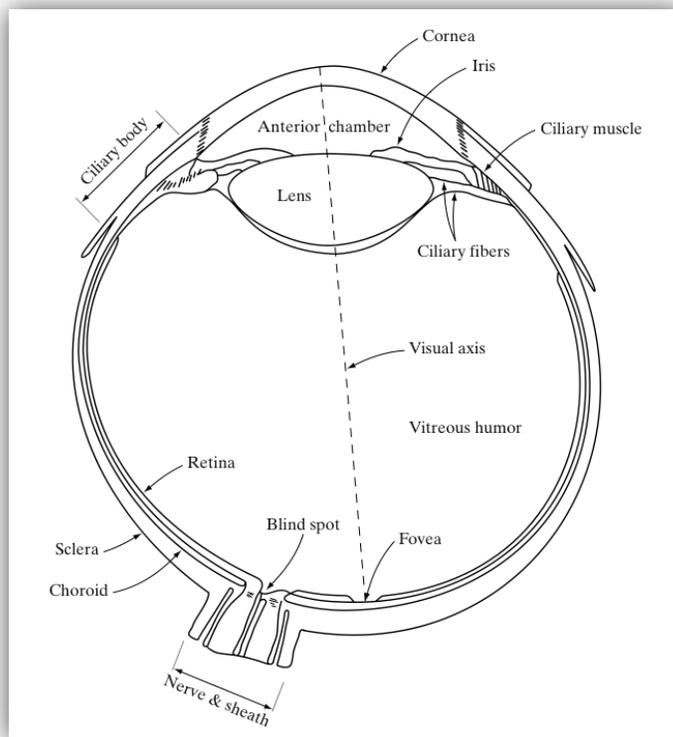


Ilustración 1. Corte Horizontal del ojo humano

Al interior de la envoltura esclerótica se halla una envoltura interna, la coroides, que es, una capa oscura provista de una red de vasos sanguíneos que funcionan como la principal fuente de nutrición del ojo. La coroides es muy pigmentada y por tanto ayuda a reducir la cantidad de luz innecesaria que entra al ojo. En su extremo anterior, la coroides está dividida en el cuerpo ciliar y el iris. El segundo se contrae o expande para controlar la cantidad de luz que entra al ojo. La apertura central del iris (la pupila) varía en diámetro, desde 2 a 8 milímetros aproximadamente.

El frente del iris contiene el pigmento visible del ojo, mientras que la parte posterior contiene un pigmento negro.

FLUVER SYSTEM

El lente está hecho de capas concéntricas de células fibrosas y está suspendido por fibras asociadas al cuerpo ciliar. Éste tiene una composición de 60 a 70 por ciento de agua, cerca de 6 por ciento de grasa y más proteína que cualquier otro tejido en el ojo. El lente es coloreado por una ligera pigmentación amarilla que se incrementa por el paso de los años. En casos extremos, el nublamiento excesivo del lente, causado por la aflicción comúnmente llamada como cataratas, puede provocar incapacidad para diferenciar los colores y la pérdida de una visión clara.

El lente absorbe un 8 por ciento del espectro de luz visible y una relativamente más alta absorción en longitudes de onda más cortas. Tanto la luz infrarroja como la ultravioleta son absorbidas considerablemente por las proteínas contenidas en la estructura del lente, pero en cantidades excesivas pueden dañar al ojo. La más interior de las membranas es la retina, la cual bordea el interior de la porción posterior de la pared. La visión patrón es posible por la distribución de receptores de luz discretos sobre la superficie de la retina. Existen dos clases de receptores: conos y bastones. El número de conos en cada uno de los ojos es de 6 a 7 millones.

Se localizan ante todo en la porción central de la retina, llamada la fovea, y son altamente sensibles al color. Los humanos pueden resolver detalles finos en gran parte con los conos porque cada uno de éstos se encuentra conectado a su propia terminal nerviosa. Los músculos controlan de la rotación del ojo hasta que la imagen de un objeto de interés cae en la fovea. La visión de cono es llamada fotópica o visión de luz brillante. El número de bastones es mucho más grande y de 75 a 150 millones de éstos se hallan distribuidos sobre la superficie de la retina. El área de distribución más amplia y el hecho de qué cantidad de bastones están conectados a un solo extremo de nervio reduce la cantidad de detalles discernibles por estos receptores. Los bastones sirven para dar una imagen general, en conjunto, del campo de visión. Ellos no están involucrados en la visión a color y son sensibles a bajos niveles de iluminación [26].

Formación de la imagen en el ojo

La principal diferencia entre el lente del ojo y un lente óptico ordinario es que el primero es flexible. Como se ilustró en la *ilustración 1*, el radio de curvatura de la superficie anterior del lente es más grande que el radio de su superficie posterior. La forma del lente es controlada por la tensión en las fibras del cuerpo ciliar. Para enfocar objetos distantes, los músculos controladores causan que el lente sea relativamente allanado. Asimismo, estos músculos permiten al lente volverse espeso de acuerdo al enfoque que se haga sobre objetos cercanos al ojo. La distancia entre el centro del lente y la retina (llamada longitud focal) varía de 17 a 14 mm aproximadamente, ya que la potencia refractiva de los lentes se incrementa a partir de su mínimo hasta su máximo. Cuando el ojo enfoca sobre un objeto más lejano que 3 metros, el lente muestra su potencia refractiva más baja. Cuando el ojo enfoca sobre un objeto cercano, el lente es altamente refractivo. Con estos datos, resulta sencillo encontrar la medida de la retina [27].

3.3 Componentes de un sistema de visión

Un sistema de visión por computadora está compuesto de las siguientes etapas [28]:

- Adquisición de la imagen
- Pre-procesamiento
- Segmentación
- Extracción de características (representación y descripción)
- Reconocimiento e interpretación (Identificación de objetos)

Enseguida se ilustra cómo se relacionan los componentes de un sistema de visión por computadora.

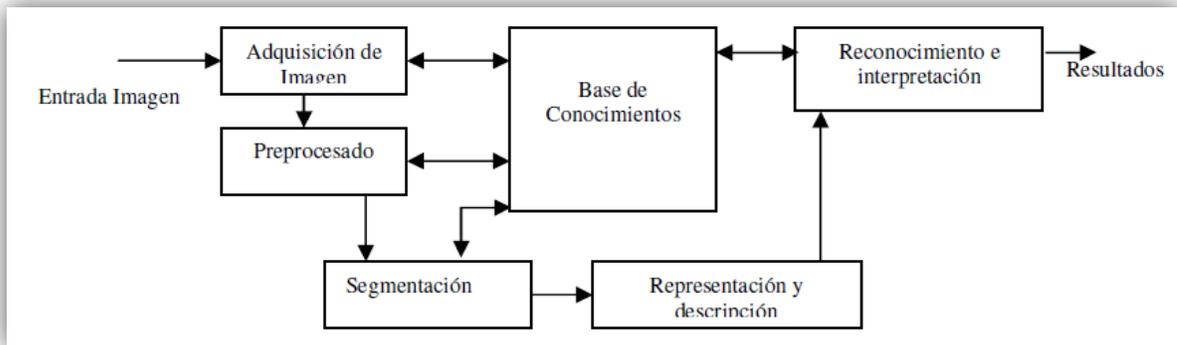


Ilustración 2. Componentes de un sistema de visión

3.3.1 Pre-procesamiento de la Imagen

El pre-procesamiento de la imagen consiste en reducir la información en la misma, de modo que pueda ser interpretada por medio de una computadora.

3.3.2 Segmentación

Definida en forma amplia, la segmentación particiona una imagen de entrada en sus partes constituyentes u objetos. Generalmente, la segmentación automática es una de las tareas más difíciles en el procesamiento de imágenes. En términos del reconocimiento de caracteres, el rol de la segmentación es extraer caracteres individuales y palabras del fondo de la imagen.

Métodos y algoritmos

A continuación se presentan los algoritmos utilizados en la segmentación de la imagen:

- Enmascaramiento Lane

FLUVER SYSTEM

- Eliminación del fondo
- Filtración de manchas y ruido
- Rastreo del vehículo
- Extracción del contorno

3.3.3 Extracción de características (representación y descripción)

La descripción llamada también selección de características, trata con extracción de los rasgos que resulta en alguna información cuantitativa de interés o características que son básicas para diferenciar una clase de objetos con otra.

3.3.4 Reconocimiento e interpretación

El reconocimiento es el proceso que etiqueta, o asigna un nombre, a un objeto basándose en la información que proveen sus descriptores. La interpretación involucra la asignación de significado a un conjunto de objetos reconocido.

3.3.5 Base de Conocimientos

La base de conocimientos mantiene todo el conocimiento que se tiene acerca del problema tratado. Este conocimiento puede ser simple, teniendo solo las regiones de la imagen donde la información de interés se conoce que está localizada, con lo que se limita la búsqueda que conduce a esa información.

La base de conocimientos también puede ser compleja, tal como una lista interrelacionada de todos los defectos mayores posibles en un problema de inspección de materiales o una base de datos de imágenes conteniendo imágenes de satélite en alta resolución de una región en conexión con aplicaciones de detección de cambios. Además de guiar la operación de cada módulo del proceso, la base de conocimientos también controla la interacción entre módulos.

CAPÍTULO 4 ANÁLISIS

4.1 Estudio de Factibilidad

En este estudio

Una vez definido el problema y las causas que conllevan el desarrollo de un sistema, es factible analizar la tecnología correspondiente y la capacidad técnica que implica la creación de este proyecto, así como los costos, beneficios y grado de aceptación que generará la aplicación en los usuarios finales.

A quien está dirigido este proyecto

El presente trabajo está dirigido a sectores de seguridad pública y aquellas dependencias de gobierno o empresas privadas que cuenten con un sistema de monitoreo de las calles y avenidas mediante cámaras de video.

Importancia del proyecto

- **¿Cómo se las arreglaría la organización si no se lleva a cabo este sistema?**
Los operadores de los sistemas de video-vigilancia deberían de revisar constantemente y cada cierto lapso de tiempo la carga de flujo vehicular.
- **¿Cuáles son los problemas con los procesos actuales y cómo ayudaría el nuevo sistema a resolverlos?**
Llevar a cabo una revisión manual siempre es más tardado que una automatizada, además de distraer a los operadores de otros eventos.
- **¿La información se puede obtener y transferir a otros sistemas de la organización?**
Si no se conoce la infraestructura de dichos sistemas o si el mantenimiento de estos es por parte de empresas privadas no es probable.
- **¿El sistema requiere de tecnología que no se ha utilizado anteriormente en la organización?**
No, la única tecnología necesaria son las cámaras de video-vigilancia, con las que la organización trabaja actualmente.
- **¿A qué debe ayudar el sistema y a qué no necesita ayudar?**
Debe ayudar a la pronta detección de congestionamiento vehicular, dejando las medidas ya sean preventivas o correctivas a los operadores.

Factibilidad del proyecto

FACTIBILIDAD DEL PROYECTO		
FACTIBILIDAD ECONÓMICA	Económicamente es rentable llevar a cabo la realización del proyecto ya que se trata de un trabajo terminal que se elaborará completamente en computadoras personales y aunque se requiere de una infraestructura para el monitoreo de vialidades del Distrito Federal se ocupará la de otras empresas o dependencias del gobierno.	
FACTIBILIDAD COMERCIAL	El trabajo terminal es comercialmente factible, dado que aunque existen ya sistemas que pueden detectar la presencia de automóviles en alguna vialidad no hay sistemas implementados que detecten y calculen la velocidad de los automóviles.	
FACTIBILIDAD TÉCNICA	El trabajo terminal es factible técnicamente hablando, ya que aunque no contamos en este momento con todos los conocimientos necesarios, si tenemos los suficientes para el inicio del desarrollo, además de tiempo para investigar a cerca de estos temas.	
FACTIBILIDAD LEGAL	El trabajo terminal es factible legalmente, ya que en la parte del desarrollo se utilizaran herramientas de software libre además de que se cuenta con el apoyo de instalaciones del gobierno que pueden proveer los videos con los que se harán las pruebas del sistema.	

Tabla 3. Factibilidad del proyecto

4.2 Metodología de Desarrollo

4.2.1 Paradigma de construcción de prototipos evolutivo

En algunas ocasiones se tiene claro un conjunto de requisitos del producto o sistema esencial, pero todavía se deben definir los detalles de las extensiones del producto o sistema, para este tipo de situaciones se tienen los modelos de proceso evolutivos, donde se desarrollan versiones cada vez más completas del software. [29]

Para realizar nuestro Trabajo Terminal necesitamos tener conocimientos sobre tratamiento digital de imágenes, de lo cual, conocemos lo básico. Por lo tanto, no sabemos la eficacia que pueda tener algún algoritmo para el tratamiento de las imágenes. El paradigma que hemos encontrado adecuado es el paradigma de construcción de prototipos, ya que nos puede ofrecer un mejor enfoque.

¿Cómo funciona este paradigma?

- Inicia con la comunicación con el cliente y se definen los objetivos globales para el software
- Se identifican los requisitos conocidos y las áreas del esquema donde es necesaria más definición
- Se plantea con rapidez una iteración de construcción de prototipos
- Se presenta un modelado

El prototipo nos servirá como un mecanismo para identificar los requisitos del software. La ventaja de un proceso del software que se basa en un enfoque evolutivo es que la especificación se puede desarrollar de forma creciente.

Sus desventajas:

- Los sistemas se desarrollan rápidamente, no es rentable producir documentos que reflejen cada versión del sistema.
- Los cambios continuos tienden a corromper la estructura del software [30].

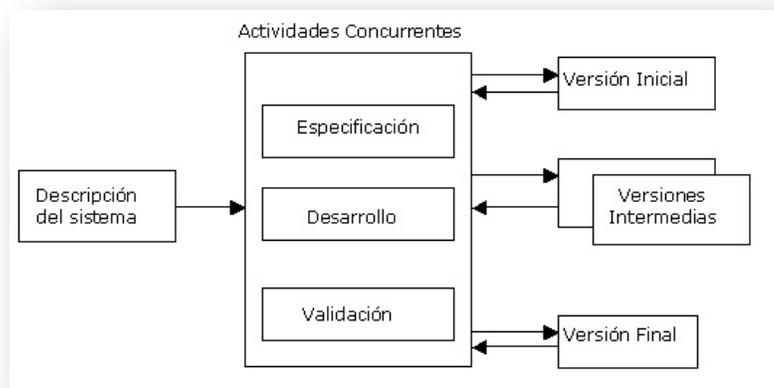


Ilustración 3. Metodología Prototipado Evolutivo

4.3 Documento de Visión

Nombre del Proyecto: Cálculo del flujo vehicular mediante segmentación de imágenes. Fluver System. **Versión:** <2.0>

Visión: Generar un sistema que sea capaz de medir el flujo vehicular por medio de un video. **Fecha:** 20/09/2011

Identificador del documento: Trabajo Terminal No. 2011-0011

4.3.1 Introducción

Fluver System es un desarrollo de software que tiene como propósito determinar la cantidad de flujo vehicular y la velocidad promedio existentes en una avenida en un tiempo determinado a partir del análisis de un video. El sistema se encargará de reconocer a los vehículos y su velocidad además de llevar un control estadístico de dichos datos.

4.3.2 Posición del producto

a) Definición del Problema

El problema es:	En el sistema de video vigilancia de la ciudad de México no se ha implementado un sistema que mida el flujo vehicular a través de los videos proporcionados por las cámaras instaladas en las avenidas, por lo tanto no se tiene un sistema automatizado que ayude a conocer el tránsito existente en alguna avenida y proporcione rutas alternativas.
Afecta a:	A los conductores que circulan por avenidas con tránsito pesado y al medio ambiente.
Impacto:	Los conductores reciben un informe subjetivo del tránsito, incluso desconocen el flujo vehicular de una determinada avenida, lo que impide que tomen una decisión de la ruta que pueden elegir y muchas veces toman la que tiene un tránsito denso. Esta situación conlleva a una mayor emisión de contaminantes al ambiente.
Buena Solución	Contar con un sistema de información que pueda dar a conocer la cantidad de flujo vehicular que hay en una avenida y lleve una estadística, para así generar un dato objetivo y que los conductores tomen rutas más fluidas, lo que aminoraría la emisión de contaminación.

Tabla 4. Definición del problema

FLUVER SYSTEM

b) Declaración de la posición del producto

Para:	El sistema Fluver System no tendrá un usuario final como tal, pero asignaremos usuario operativo que se hará cargo de verificar el funcionamiento del sistema.
El cual:	Se encargará solo del reconocimiento de la velocidad de los vehículos.
Fluver System:	Es el subsistema que formará parte de un sistema final, el cual utiliza segmentación de imágenes para identificar vehículos en las avenidas a través de un video proporcionado.
El cual:	Mide la velocidad de dichos vehículos
Diferencias:	Aún no se cuenta con un sistema automatizado que realice tales cálculos en la ciudad de México, lo que lo hace único.
Nuestro Producto:	Formará parte de un sistema completo, en el cual, se encargará de analizar y realizar los cálculos correspondientes a los videos que se proporcionen.

Tabla 5. Declaración de la posición del producto

c) Requerimientos de los stakeholders

Identificador	Descripción del Requerimiento	Origen
REQ01	El sistema requiere como entrada un video en tiempo real para su posterior tratamiento.	Operador
REQ02	Se requiere que el sistema identifique los vehículos en movimiento en cada imagen.	Operador
REQ03	El sistema debe calcular la velocidad promedio de los vehículos en determinada avenida.	Operador
REQ04	Se requiere que el sistema guarde en un archivo un registro de la velocidad promedio, en principio, cada 10 minutos.	Operador
REQ05	El sistema debe graficar la velocidad promedio de los vehículos de acuerdo a los registros almacenados.	Operador
REQ06	Procesamiento rápido de los datos, que el sistema pueda procesar una imagen por segundo como máximo.	Operador
REQ07	Se debe realizar la extracción de fotogramas del video para generar las imágenes necesarias.	Sistema
REQ08	Se requiere que el sistema realice el pre-procesamiento o mejora de las imágenes por medio de un filtro.	Sistema
REQ09	Se requiere que el sistema realice la segmentación de las imágenes para quitar objetos que no son útiles.	Sistema
REQ10	El sistema considerará solo los ángulos del video en vista aérea.	Sistema
REQ11	El formato de video tendrá que ser .avi, tamaño de 640 x 480.	Sistema
REQ12	Se requiere que la medición de la velocidad de los vehículos sea a la luz del día.	Sistema
REQ13	El sistema no tomará en cuenta las características propias de los vehículos.	Sistema

Tabla 6. Requerimientos

d) Priorizando los requerimientos

Prioridad	Requerimiento
1	REQ01 El sistema requiere como entrada un video en tiempo real para su posterior tratamiento.
8	REQ02 Se requiere que el sistema identifique los vehículos en movimiento en cada imagen.
10	REQ03 El sistema debe calcular la velocidad promedio de los vehículos en determinada avenida.
11	REQ04 Se requiere que el sistema guarde en un archivo un registro de la velocidad promedio, en principio, cada 10 minutos.
12	REQ05 El sistema debe graficar la velocidad promedio de los vehículos de acuerdo a los registros almacenados.
13	REQ06 Procesamiento rápido de los datos, que el sistema pueda procesar una imagen por segundo como máximo.
5	REQ07 Se debe realizar la extracción de fotogramas del video para generar las imágenes necesarias.
6	REQ08 Se requiere que el sistema realice el pre-procesamiento o mejora de las imágenes por medio de un filtro.
7	REQ09 Se requiere que el sistema realice la segmentación de las imágenes para quitar objetos que no son útiles.
3	REQ10 El sistema considerará solo los ángulos del video en vista aérea.
2	REQ11 El formato de video tendrá que ser .avi, con un tamaño de 640 x 480.
4	REQ12 Se requiere que la medición de la velocidad de los vehículos sea a la luz del día.
9	REQ13 El sistema no tomará en cuenta las características propias de los vehículos.

Tabla 7. Prioridad de requerimientos

4.4 Casos de Uso

Un caso de uso representa una unidad discreta de interacción entre un usuario (humano o máquina) y el sistema. Cada caso de uso describe la funcionalidad que construirá el sistema propuesto, el cual puede incluir la funcionalidad de otro caso de uso [31].

El sistema *Fluver System* cuenta con los siguientes casos de uso:

1. Gestión del procesamiento
2. Cálculo del flujo vehicular y velocidad
3. Gestión de la Base de datos

4.4.1 Diagrama de Casos de Uso

El siguiente diagrama describe la funcionalidad de nuestro sistema

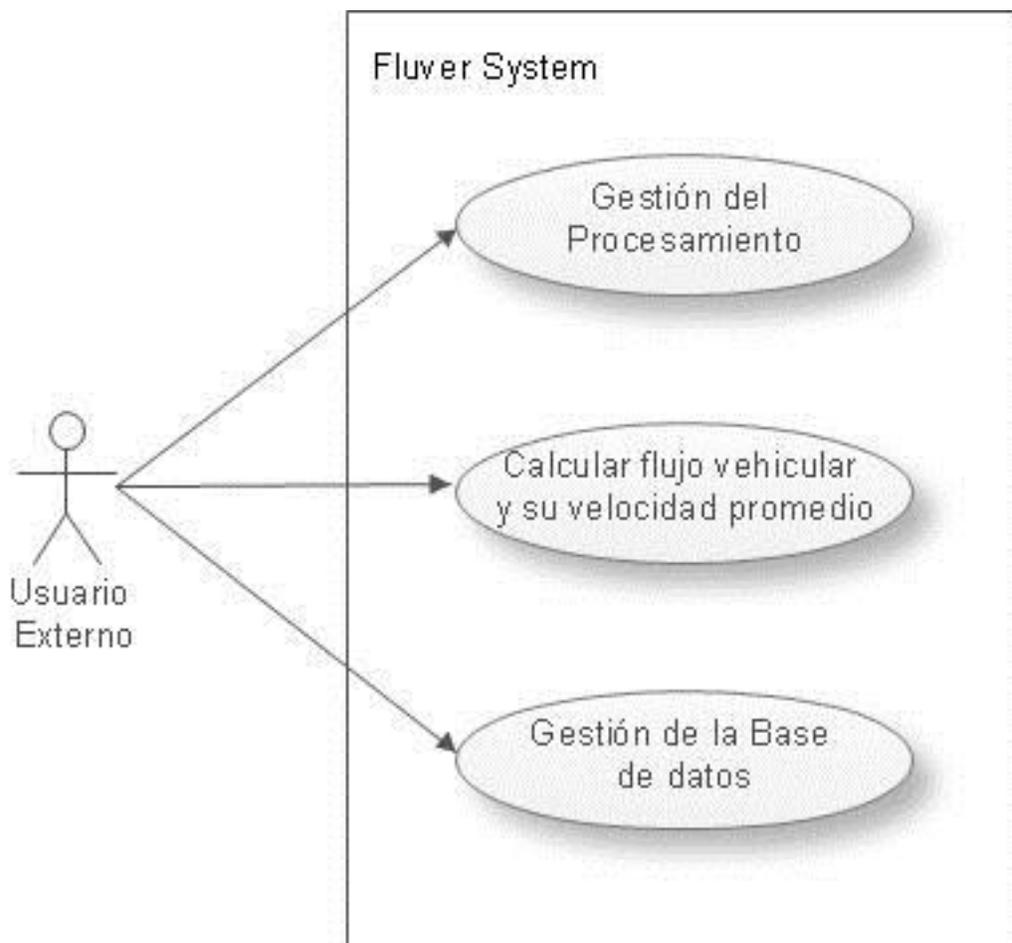


Ilustración 4. Diagrama de Casos de Uso

4.4.2 Descripción de Casos de Uso

Debido a que el modelo que estamos utilizando es evolutivo, se realizará una descripción de los casos de uso a trazo grueso.

Número	Nombre	Actores	Objetivo	Descripción	Tipo ³
01	Gestión del procesamiento	Usuario Externo	El usuario es consciente de los pasos que se generan para el tratamiento digital del video y puede gestionar cada paso.	<ul style="list-style-type: none"> El usuario permite una entrada de video El sistema ofrece opciones para reproducir el video. El usuario puede manipular el video. 	Base
02	Cálculo del flujo vehicular y su velocidad	Usuario Externo	Obtener las estadísticas con los resultados de la velocidad y flujo vehicular.	<ul style="list-style-type: none"> El usuario solicita calcular la velocidad en una avenida determinada. El sistema procesa el video y calcula la velocidad. El sistema guarda los datos en la Base de Datos. El sistema despliega el resultado y algunas estadísticas del flujo en esa avenida 	Adicional
03	Gestión de la base de datos	Usuario Externo	El usuario podrá obtener resultados de la base de datos, además de poder gestionarla.	<ul style="list-style-type: none"> El usuario solicita la velocidad en una determinada avenida El sistema le ofrece una interfaz de usuario para que interactúe con la base de datos. 	Adicional

Tabla 8. Descripción de los casos de uso

³ En la columna "Tipo" se indicará como base aquellos casos de uso que el usuario utilice en primer lugar, iniciando el paquete. El resto es indicado como adicional.

4.5 Análisis de Riesgos

Esta sección tiene la finalidad de advertir los posibles riesgos del proyecto, describiendo cada uno, analizando la probabilidad de que sucedan, sus efectos en el proyecto, así como sus planes de prevención y contingencia.

4.5.1 Descripción de los Riesgos

Id	Riesgo	Descripción
RI01	Fallo en el equipo y pérdida de información	Que algún equipo de los integrantes sea afectado por un accidente, virus, fallo en el disco, sea robado, etc. Esto implicaría que la información almacenada en dicho equipo se pierda totalmente.
RI02	Problemas de salud	Que algún integrante del equipo de trabajo tenga problemas de salud por alguna enfermedad o discapacidad física temporal, lo cual afectaría el rendimiento del mismo, su trabajo y el tiempo en la planeación.
RI03	Baja de algún integrante del equipo de trabajo	Algún integrante por razones escolares o personales, se dé de baja del equipo.
RI04	Bajo rendimiento del equipo de trabajo	Podría haber un bajo rendimiento en el equipo por pérdida de tiempo, ya que el llegar a un acuerdo entre cuatro integrantes muchas veces es complicado. También se corre el riesgo de que la falta de motivación en el proyecto sea un factor para retrasar las entregas.
RI05	Subestimación del proyecto y falta de experiencia	Tener una idea errónea del tamaño real del proyecto y que alguna de las fases planificadas requiera más tiempo que el asignado en un principio. Además de nuestra inexperiencia en el área.
RI06	No obtención del video	La falla en la obtención del video de las cámaras de seguridad ya sea por falta de permisos, contratiempos, etc.,
RI07	Pérdida de tiempo en la documentación	En los sistemas que utilizan una metodología de prototipo evolutivo, no es rentable producir documentos que reflejen cada versión del sistema.
RI08	Corromper la estructura del software	Cuando más de dos personas intervienen en el código, muy probablemente hagan cambios en la estructura del software que no sean del todo benéficos, lo hacen poco entendible o no funcional.
RI09	Mezcla de versiones	Múltiples respaldos o varias versiones del sistema o documento, puede provocar confusión al momento de actualizarlos.
RI10	El video tenga características diferentes a las esperadas	No se tiene la certeza de cuales sean las características del video que vamos a utilizar y puede afectar los resultados del sistema

Tabla 9. Descripción de Riesgos

4.5.2 Plan de Prevención y Plan de Contingencia

RI01: Fallo en un equipo

Impacto:	Catastrófico
Probabilidad:	Moderada
Plan de prevención:	Contamos con un servicio de alojamiento de archivos multiplataforma en la nube llamado Dropbox, en el cual se tienen que guardar todos los archivos generados. Por otro lado cada uno de los integrantes debe hacer un respaldo en su equipo, en memorias y en caso del video en CD, el cual lo estará actualizando en cada mejora o nuevo archivo.
Plan de contingencia:	Es difícil que perdamos la información por completo y menos si se sigue el plan de prevención, ya que la información estará respaldada en cada equipo de los cuatro integrantes. Si alguno de ellos llegara a perder información, sería mínima, por lo tanto trabajaría un día más fuera del calendario para ponerse al corriente

RI02: Problemas de salud

Impacto:	Moderado
Probabilidad:	Baja
Plan de prevención:	Determinar un tiempo de holgura en las entregas. Estar al tanto de lo que hace cada integrante, para que en algún momento dado, si la persona llega a faltar se sepa cuáles son sus entregables y que el equipo en el tiempo de holgura trabaje la parte del integrante faltante.
Plan de contingencia:	Que todo el equipo trabaje en el tiempo de holgura, en caso de existir, de no ser así, trabajar horas extras.

RI03: Baja de algún integrante del equipo de trabajo

Impacto:	Moderado
Probabilidad:	Baja
Plan de prevención:	Estar al tanto de las actividades que desempeña cada integrante
Plan de contingencia:	Este plan depende del tiempo de desarrollo que se haya realizado en el proyecto, si es en un principio, no hay mucho problema ya que podemos seguir los integrantes restantes sin problema pero con tiempos más largos de trabajo. Si es a medio proyecto, tal vez podría ser un impacto alto, porque aunque el equipo esté al tanto de sus actividades, el cubrir las propias y las del integrante faltante podría ser muy difícil de realizar. Si es al final, podría no haber problema, ya que el mayor porcentaje del proyecto ha sido efectuado.

RI04: Bajo rendimiento del equipo de trabajo

FLUVER SYSTEM

Impacto:	Alto
Probabilidad:	Moderado
Plan de prevención	Asignar a uno de los integrantes como moderador cuando se tengan que llegar a acuerdos, para que todos puedan expresar su idea y así bajo la guía del moderador, tomando en cuenta los argumentos de cada uno, construir la mejor solución. Además de tratar de motivar a nuestros compañeros de trabajo.
Plan de contingencia:	La persona que se percate de este asunto, deberá convocar una junta para platicar sobre las posibles razones del bajo rendimiento de los integrantes, para así llegar a una solución y tal vez un ajuste en la calendarización; en caso extremo, horas extra.

RI05: Subestimación del proyecto y falta de experiencia

Impacto:	Alto
Probabilidad:	Alta
Plan de prevención	Platicar con nuestro director, seguramente él nos podrá dar una idea de la manera en que podemos planificar nuestro tiempo debidamente. Además se debe hacer un marco teórico y un estado del arte que nos ayudarán a saber qué es lo que se ha hecho en el área y el uso de las tecnologías para desarrollarlo. Otra medida es asignar tiempos de holgura entre entregas de prototipos.
Plan de contingencia:	Muchas veces aunque se tenga una estimación del tiempo, no podemos pasar por alto que somos inexpertos en el área y que tal vez necesitemos más tiempo en una fase determinada. En caso de que tengamos problemas de tiempo, lo recomendable sería plantearle el problema al director y buscar una solución, de no haber dicha solución reajustar tiempos y horas extra.

RI06: No obtención del video

Impacto:	Catastrófico
Probabilidad:	Moderada
Plan de prevención:	Hacer varias peticiones en diferentes lugares para la obtención del video. El limitarnos a una sola fuente podría ser catastrófico para el sistema, ya que el video es la parte fundamental.
Plan de contingencia:	En caso de no obtener ningún video a tiempo, trabajar con un video que se encuentre en Internet y nos pueda servir para reconocer los automóviles y su velocidad.

RI07: Pérdida de tiempo en la documentación

Impacto:	Moderado
-----------------	-----------------

FLUVER SYSTEM

Probabilidad:	Moderada
Plan de prevención:	Un prototipo no es la versión final del sistema, así que de nada sirve documentar exhaustivamente un prototipo. Se debe llevar un control de lo que se está haciendo, pero solo se deben documentar las partes importantes del prototipo que se está claro estarán en la versión final.
Plan de contingencia:	Enfocarse a la próxima entrega de algún prototipo y seguir el plan de prevención

RI08: Corromper la estructura del software

Impacto:	Alto
Probabilidad:	Alta
Plan de prevención:	Siempre que se realice un cambio, comentarlo dentro del mismo código y si es posible, comentarlo en las reuniones que el equipo tiene programadas, siempre teniendo un argumento del por qué hizo tal cambio. Tener un respaldo del último incremento funcional.
Plan de contingencia:	Determinar que tanto se puede corregir el problema y evaluar la posibilidad de volver a programar dicho módulo o cambiarlo.

RI09: Mezcla de versiones

Impacto:	Alto
Probabilidad:	Moderada
Plan de prevención:	El uso de una herramienta de software que controle las versiones, además de que cada integrante debe llevar un control de nombres.
Plan de contingencia:	Hacer una junta en el equipo para establecer los archivos más actuales y llegar uno final.

RI10: El video tenga características diferentes a las esperadas

Impacto:	Alto
Probabilidad:	Alta
Plan de prevención:	Contar con varios videos.
Plan de contingencia:	Grabar un video nosotros mismos.

4.6 Análisis de Lenguajes de Programación

El procesamiento de imágenes suele ser un proceso con un costo computacional elevado, teniendo eso en cuenta es importante elegir un lenguaje de programación que nos permita un buen rendimiento en el procesamiento de datos.

Nuestra comparación está basada en resultados publicados en internet [32] en donde se aplican varios *benchmarks* a estos lenguajes, estos pueden ser problemas con cálculos complejos, problemas con varios cálculos simples, problemas que requieren muchas iteraciones o una combinación de estos. Las pruebas se realizan en computadoras de un solo núcleo y de cuatro núcleos y con sistemas operativos de 32 bits y 64 bits.

Para nuestra comparación decidimos tomar en cuenta 3 lenguajes de programación diferentes, estos son:

- **C++:** Ya que es un lenguaje ampliamente usado para sistemas que requieren un mayor rendimiento y suele ser considerado como uno de los mejores en ese aspecto.
- **Java:** Ya que ofrece una gran variedad de herramientas para el manejo de imágenes y video, además de tener la ventaja de ser multiplataforma.
- **Python.**

4.6.1 Descripción de los benchmarks considerados

N-Cuerpos

Motivo: Se prueba el rendimiento del lenguaje para ejecutar cálculos complejos que pueden ser necesarios en algunas etapas del procesamiento de las imágenes.

Descripción: Este problema consiste en calcular la interacción de n cuerpos celestiales que interactúan gravitacionalmente entre sí, la solución a este problema cambia según la cantidad de cuerpos considerados, pero involucra el cálculo de varias integrales o series.

Anillo de hilos

Motivo: Se prueba la eficiencia con que el lenguaje usa diversos hilos de ejecución que seguramente serán importantes en este proyecto.

Descripción: Consiste en crear 503 hilos de ejecución y pasar un token entre los hilos. Se deben mantener vivos los hilos durante todo el proceso, estos hilos estarán unidos explícita o implícitamente en un anillo y cada hilo debe pasar el token al siguiente hilo 1000 veces o en otras palabras el token debe recorrer todo el anillo 1000 veces.

Mandelbrot

Motivo: Se prueba la eficiencia del lenguaje al realizar varios cálculos simples.

Descripción: Consiste en el cálculo de un conjunto de Mandelbrot (fractal) con un nivel de recursión de 200 que después es guardado como un bmp de 200x200.

Reducción Fannkuch

Motivo: Se prueba la eficiencia de los lenguajes al acceder al enteros pequeños, lo que será similar a cuando se acceda a la información de las imágenes en nuestro proyecto.

Descripción: Consiste en todas posibles permutaciones de números del 1 al N, se lee el primer número de la permutación y se invierten los primeros n números, donde n es el primer número de la permutación. Por ejemplo en la permutación {4, 6, 2, 3, 7, 5, 1} se invierten los primeros 4 números quedando de la siguiente forma: {3, 2, 6, 4, 7, 5, 1}. Este proceso se repite hasta que el primer número es 1, en este *benchmark* se toma N = 7, lo que da 5,040 permutaciones en total.

La siguiente tabla muestra el tiempo en segundos que se requirió para completar el *benchmark* con cada uno de los lenguajes, se considera una maquina con 4 núcleos y sistema operativo Ubuntu de 64 bits.

	C++	Java	Python
N-Cuerpos	19.62	22.37	105.58
Anillo de hilos	281.48	420	420
Mandelbrot	24.47	27.67	117
Reducción Fannkuch	49.93	54.17	189.16

Tabla 10. Tiempo de ejecución en segundos de cada benchmark.

Finalmente vemos la comparación de la memoria en KB utilizada por el programa durante su ejecución.

	C++	Java	Python
N-Cuerpos	744	17,692	6,676
Anillo de hilos	5,024	289,056	10,760
Mandelbrot	32,220	67,876	9,464
Reducción Fannkuch	1,128	16,724	6,584

Tabla 11. Memoria en KB requerida para la ejecución de cada benchmark.

4.6.2 Conclusión

De los resultados podemos ver que C++ fue superior en los puntos analizados, exceptuando la memoria utilizada para el fractal Mandelbrot. Con esto en cuenta se decide que el lenguaje que se utilizara será C++.

4.7 Análisis de Herramientas de Desarrollo

Esta sección tiene como objetivo presentar una serie de herramientas con las que podemos desarrollar el sistema. Se realiza una comparación entre ellas y posteriormente se elige la que mejor se adapta a las necesidades que se requieren cubrir.

Nombre	Características	Ventajas	Desventajas
<p>Marvin (Image Processing)</p>	 <p>Framework de código abierto desarrollado en Java que permite:</p> <ul style="list-style-type: none"> • Manipular imágenes • Manipular fotogramas de video capturados • Procesamiento de imágenes multihilos • Integra plug-ins con Interfaces de Usuario Gráficas • Análisis del rendimiento del plug-in • Extensión de características por medio de plug-ins • Pruebas de automatización <p>Los algoritmos de procesamiento de imágenes son implementados como plug-in y pueden ser adheridos como partes de otros plug-ins. Actualmente cuenta con 44 plug-ins disponibles.</p>	<p>Los plug-ins son cargados en tiempo real y proporcionan una manera fácil de extensibilidad e integración.</p> <p>Esto facilita el tratamiento de imágenes en cuanto a código.</p> <p>Cuenta con multihilos para procesar varias imágenes o una imagen procesada por varios hilos.</p>	<p>Los plug-ins ya están programados y solo se pueden utilizar los que ya están definidos.</p> <p>Uno de los objetivos del desarrollo de este sistema es crear algoritmos para el tratamiento de imágenes, y al ya estar hechos, no nos dan mucha flexibilidad para poder optimizarlos [33].</p>

FLUVER SYSTEM

<p>Cinder</p> 	<ul style="list-style-type: none"> • Cinder es una biblioteca de código abierto. • Multiplataforma (Mac OS X, iOS y Windows). • Manejo de OpenCV • Captura de video por webcam • Extracción de fotogramas • Procesamiento de imágenes 2D <p>Utiliza el lenguaje de programación C++ Open Source</p> <p>Se puede utilizar esta biblioteca solo para Windows o MAC Su documentación no está explicada a detalle [34].</p>
<p>Ultimate++</p> 	<ul style="list-style-type: none"> • Es framework para desarrollar aplicaciones en C++. • Incluye una serie de bibliotecas (GUI, SQL, etc.) y un ambiente integrado de desarrollo <p>Se incluyen comparaciones con Qt y con Java/Swing donde se muestra que el código se optimiza utilizando Ultimate++.</p> <p>La documentación es deficiente para utilizar la nueva herramienta [35].</p>
<p>Qt SDK</p> 	<p>Qt SDK combina el framework de Qt con las herramientas diseñadas para la creación de aplicaciones para Symbian y Maemo, en adición a plataformas de escritorio como Microsoft Windows, Mac OS X y Linux.</p> <p>Utiliza el lenguaje de programación C++. Cuenta con un IDE para desarrollar interfaces de usuario de manera más rápida.</p> <ul style="list-style-type: none"> • Qt framework. APIs para C++ y CSS/JavaScript • Qt Creator IDE. Ambiente integrado de programación que incluye herramientas <p>Cuenta con una API para tratar con videos e imágenes.</p>

FLUVER SYSTEM

	para diseñar una interfaz de usuario. Con Qt se puede reusar código eficientemente para utilizarlo en diferentes plataformas con un solo código base.	Su documentación está bien detallada y tiene varios ejemplos [36].
--	--	--

Tabla 12. Comparación de herramientas

Una herramienta necesaria para el desarrollo del proyecto es OpenCV. OpenCV es un conjunto de bibliotecas de código abierto u *Open Source* desarrolladas en un principio por Intel. Están desarrolladas en el lenguaje C y C++ y se pueden ejecutar desde Windows, Linux y Mac OS X.

Estas bibliotecas ofrecen código diseñado eficientemente, orientado a aplicaciones capaces de ejecutarse en tiempo real en procesadores modernos. Estas bibliotecas tienen como objetivo proveer las funciones más usadas en Visión por computadora y tiene más de 500 funciones implementadas. Se usan ampliamente en vigilancia, imágenes médicas, inspección industrial, interfaces de usuario, calibración de cámaras, etc. Por este motivo se a decidido utilizar esta herramienta, para la obtención del video y las codificaciones de los algoritmos de detección de movimiento.

4.7.1 Conclusión

De acuerdo a la tabla anterior, se ha concluido que el framework que ofrece mayores ventajas es QT SDK. Por manejar un lenguaje de programación C++, se puede introducir código de OpenCV, su documentación está muy detallada y se pueden encontrar varios ejemplos en internet, a comparación de otras herramientas. Un factor importante es el tiempo, no existe el suficiente tiempo para aprender a utilizar otro tipo de herramienta y peor aún, si la documentación no es tan completa. Otra ventaja que ofrece son las API's para manejar video e imágenes, lo que facilitará el trabajo.



4.8 Presupuesto de Costos

En las primeras etapas del proyecto se requieren algunas estimaciones de costos, antes de que se haga la planificación detallada. Estas estimaciones son necesarias para establecer un presupuesto para el proyecto o para asignar un precio al software para un cliente.

4.8.1 Costos de software

En esta sección se toman en cuenta los costos de las licencias necesarias para poder desarrollar y vender el sistema.

Nombre	Costo
Licencia de Office 2010 Home & Business	\$2,900
Licencia de Windows 7 Profesional	\$3,200
Licencia de ConceptDraw Pro	\$3,486
Licencia de Visual Paradigm Modeler	\$1,386
Licencia de Qt	\$3,000
Licencia de SqlServer 2008 R2	\$7,000
TOTAL	\$20,972

Tabla 13. Costo de software

4.8.2 Costo de recursos y materiales

En esta sección se toman en cuenta los costos materiales y recursos que se deben tomar en cuenta en el desarrollo del proyecto.

Nombre	Costo mensual	Costo anual
Luz	\$800	\$9,600
Internet	\$1200	\$14,400
Laptop ⁴	-----	\$42,800
TOTAL	\$2,000	\$66,800

Tabla 14. Costo de recursos y materiales

4.8.3 Gastos

En esta sección se identifican los gastos que se derivan del desarrollo del proyecto.

Nombre	Costo mensual	Costo anual
Gastos de comunicación	\$400	\$4,800
Gastos de transporte	\$1080	\$12,960
Comida	\$1440	\$17,280
Memorias de almacenamiento	-----	\$1,000
Papelería	\$50	\$600
Renta de un espacio amueblado	\$2000	\$24,000
Fotocopias	\$30	\$360

⁴ Costo de cuatro laptops.

FLUVER SYSTEM

Encuadernado	-----	\$200
Tintas para impresora	\$400	\$4,800
Impresora	-----	\$500
Otros (mantenimiento de equipo)	-----	\$5,000
TOTAL	\$5,400	\$71,500

Tabla 15. Gastos del Proyecto

4.8.4 Costo de esfuerzo

En esta sección se toma en cuenta el esfuerzo y conocimiento de cada uno de los programadores en el sistema tomando en cuenta que el número de programadores es cuatro:

Nombre	Costo mensual	Costo anual
Salario por programador	\$15,000	\$180,000
TOTAL	\$60,000	\$720,000

Tabla 16. Costo de esfuerzo

4.8.5 Coste total

Nombre	Costo
Costo de software	\$20,972
Costo de materiales y recursos	\$66,800
Gastos	\$71,500
Costo de esfuerzo	\$720,000
Total	\$879,272

Tabla 17. Costo total

La estimación del costo del proyecto es de \$879,272.00 pesos M.N (Ochocientos setenta y nueve mil doscientos setenta y dos pesos moneda nacional) con una duración de 12 meses, con la participación de cuatro integrantes.

4.9 Definición de Prototipos

Para el desarrollo del sistema se han definido tres prototipos a entregar. En la siguiente tabla se muestra tanto el objetivo como la descripción del sistema, lo que nos da un panorama más amplio de los alcances del mismo.

NOMBRE	OBJETIVO	DESCRIPCIÓN
Primer Prototipo	Desarrollar el módulo del sistema que se encargue de proporcionar una imagen segmentada.	<ul style="list-style-type: none"> ✓ Cargar un video al sistema ✓ Extracción de fotogramas del video ✓ Aplicación de filtros (pre-procesamiento) <ul style="list-style-type: none"> ❖ Nivelar contraste y brillo ❖ Escala de grises ✓ Aplicación de los algoritmos de detección de movimiento ✓ Segmentación de las imágenes
Segundo Prototipo	Implementar la entrada de video directamente desde las cámaras y el conteo de los vehículos.	<ul style="list-style-type: none"> ✓ Implementar entrada de cámaras ✓ Implementar el reconocimiento de los vehículos ✓ Implementar interfaz que muestre el reconocimiento de los vehículos ✓ Cálculo del flujo vehicular y velocidad
Tercer Prototipo	Almacenar la información de cada avenida, la velocidad promedio, registros anteriores, etc.	<ul style="list-style-type: none"> ✓ Implementar base de datos con la información de las avenidas

Tabla 18. Prototipos

CAPÍTULO 5 DISEÑO

5.1 Contexto del sistema Fluver System

El siguiente diagrama muestra el contexto en el cual el sistema va a interactuar. Se puede observar la dependencia con el sistema de videocámaras y la relación que existe con las diferentes entidades.

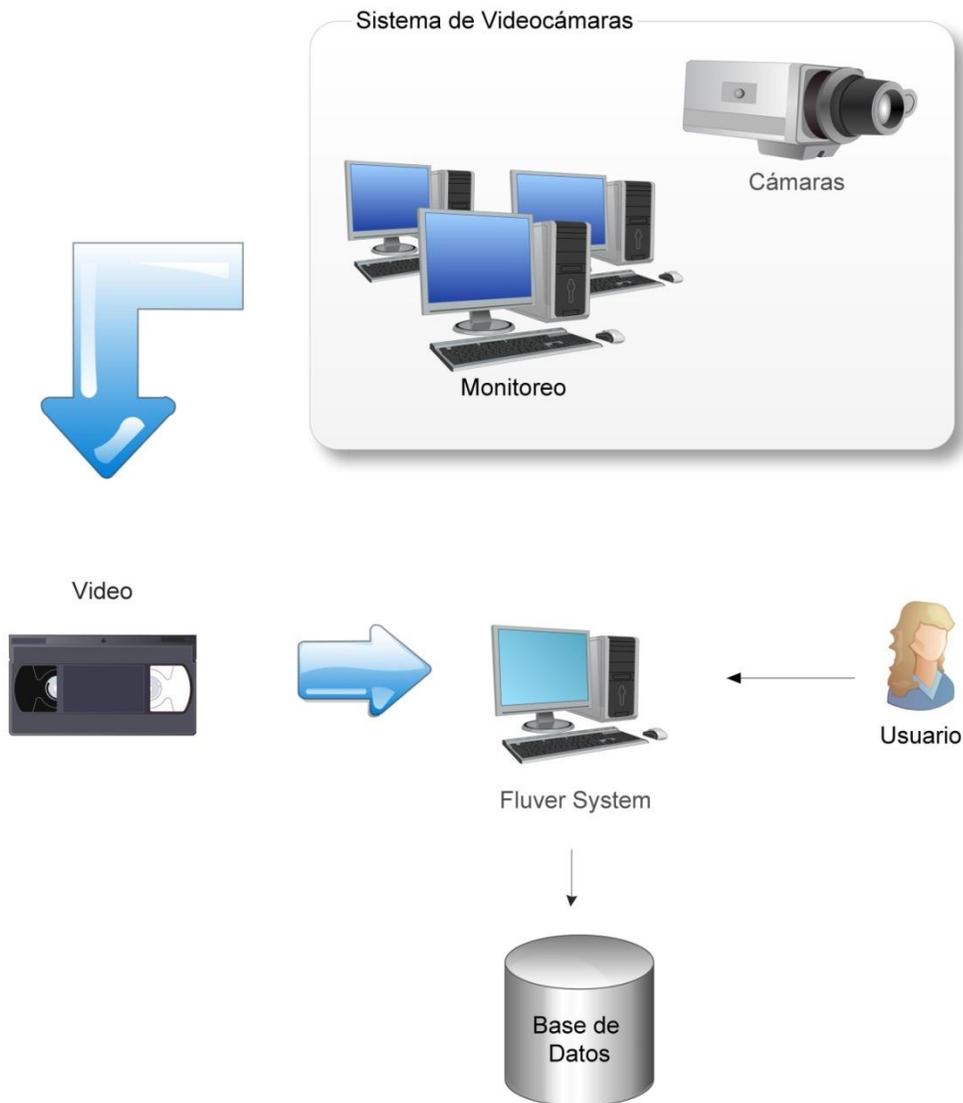


Ilustración 5. Diagrama de contexto

5.2 Diagrama de Bloques

En el siguiente diagrama de bloques se observa en qué consisten las etapas que se tienen en el sistema. Se ha escogido este diagrama para representar los procesos que se realizarán, ya que el flujo de los datos es lineal y sufre transformaciones a lo largo del proceso en general.

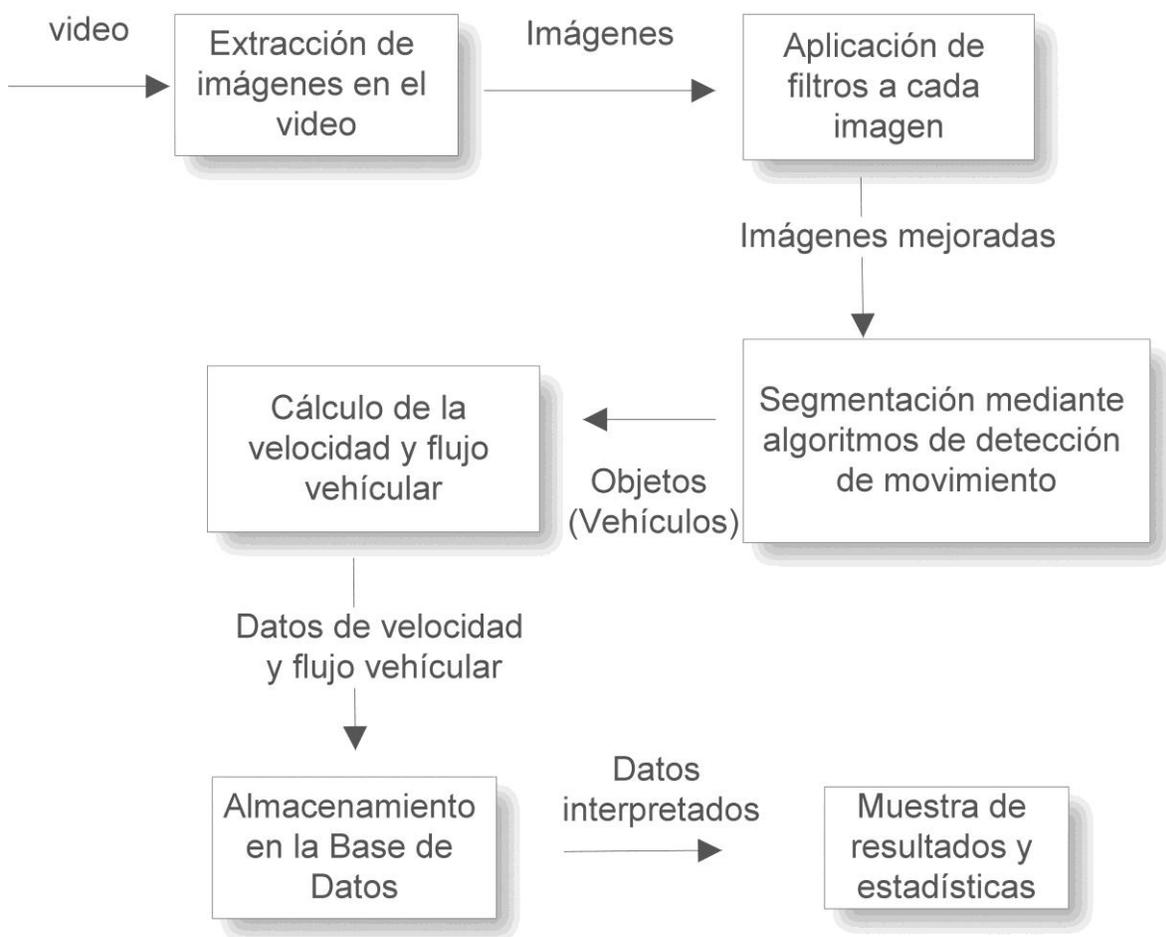


Ilustración 6. Diagrama de bloques

5.3 Visión general

El siguiente esquema pretende demostrar cómo es que el sistema interactúa con sus agentes externos y cuáles son sus procesos internos.

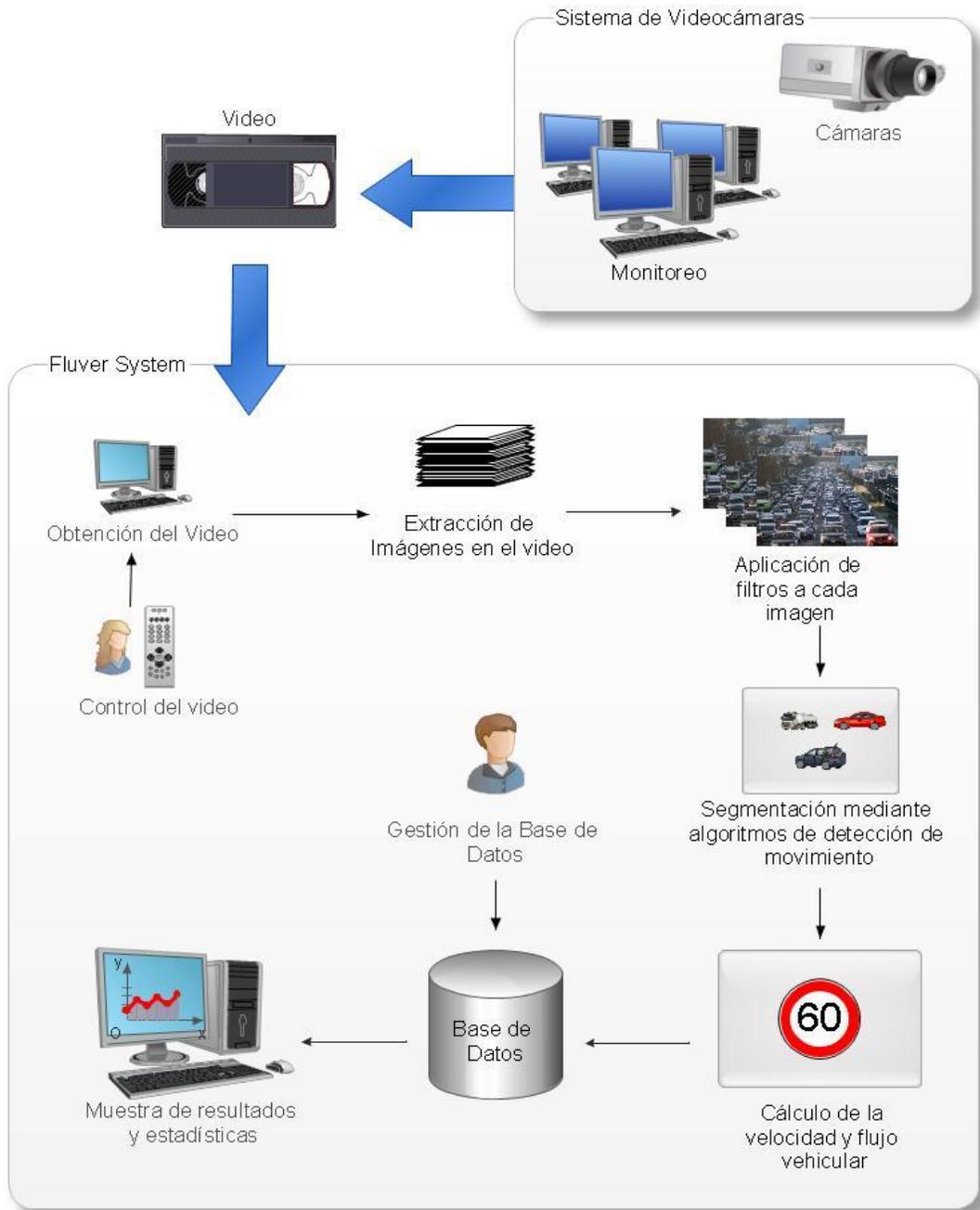


Ilustración 7. Esquema general

5.4 Diagrama de Flujo de Datos

El diagrama de flujo de datos especifica los procesos del sistema y como se van transformando los datos a lo largo de cada uno.

5.4.1 DFD nivel 0

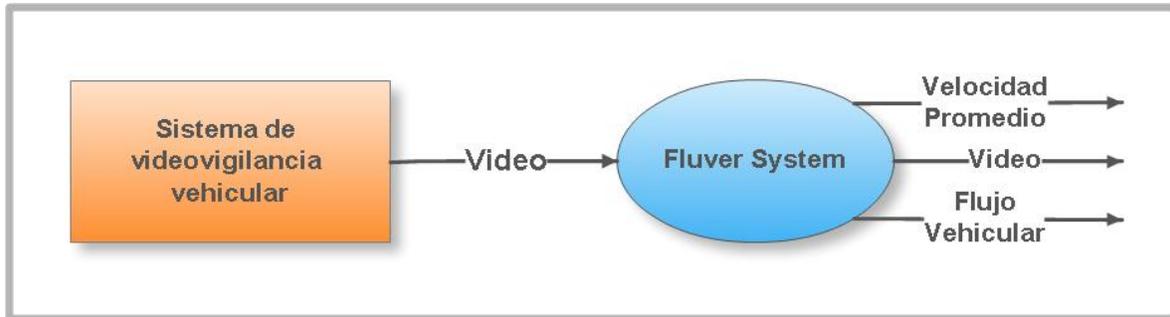


Ilustración 8. DFD nivel 0

5.4.2 DFD nivel 1

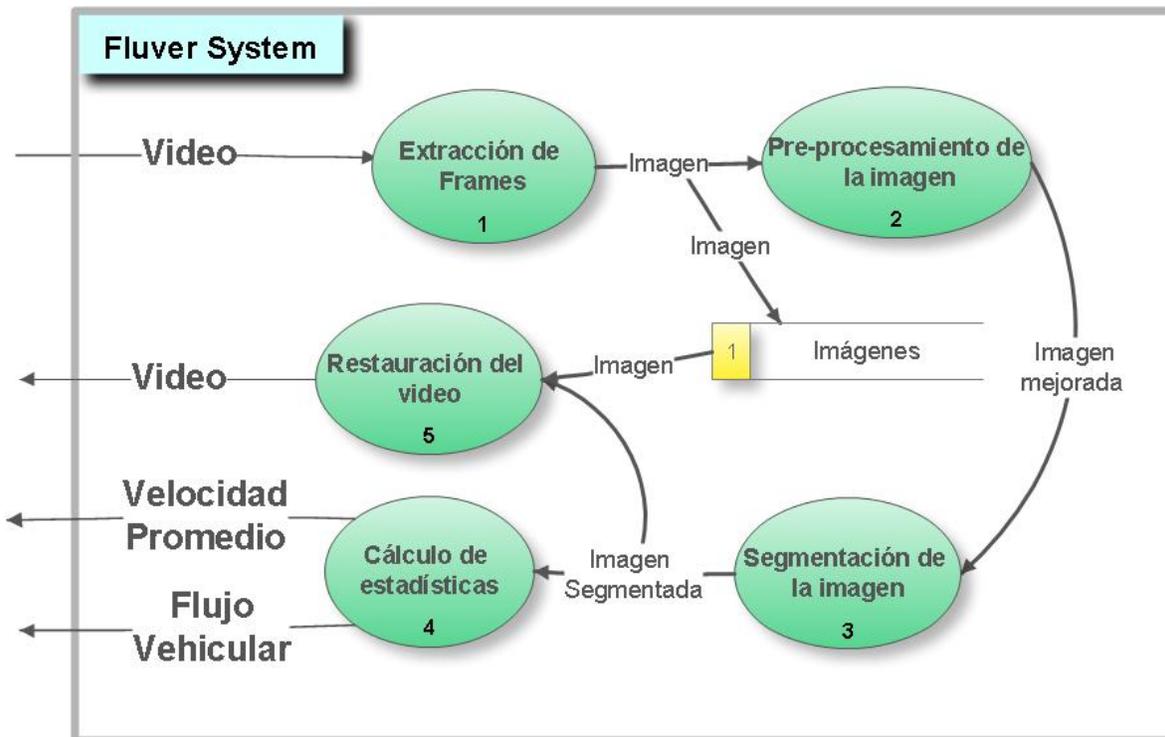


Ilustración 9. DFD nivel 1

5.4.3 DFD nivel 2

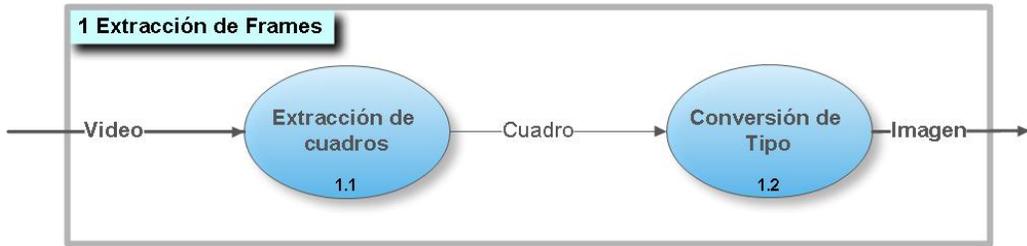


Ilustración 10. Extracción de fotogramas. Nivel 2

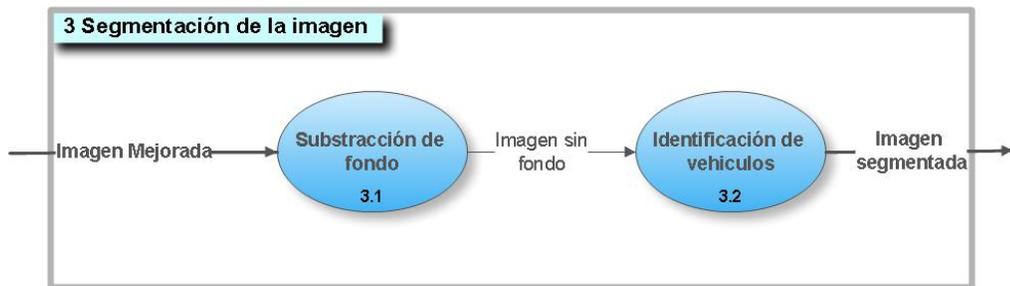


Ilustración 11. Segmentación de la imagen. Nivel 2

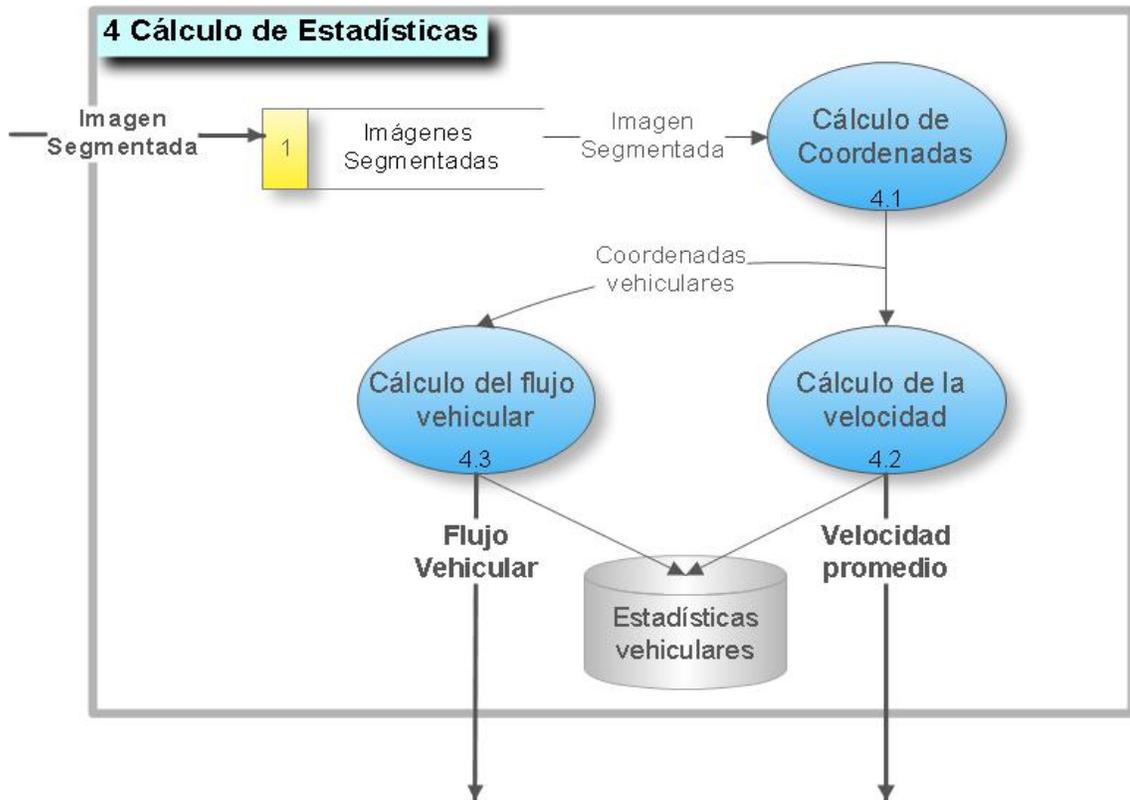


Ilustración 12. Cálculo de estadísticas. Nivel 2

5.5 Diagrama de Paquetes

El siguiente diagrama de paquetes muestra las relaciones lógicas que existen entre ellos y tiene la finalidad de expresar como es que están organizados entre sí.

Como se puede observar, existen cuatro paquetes:

- **Gui:** Paquete de manejo de la Interfaz de Usuario
- **BD:** Paquete de manejo de Base de Datos
- **ProcesamientoVideo:** Paquete que se encarga del procesamiento del video
- **ProcesamientoImagen:** Paquete que se encarga del procesamiento de cada imagen.

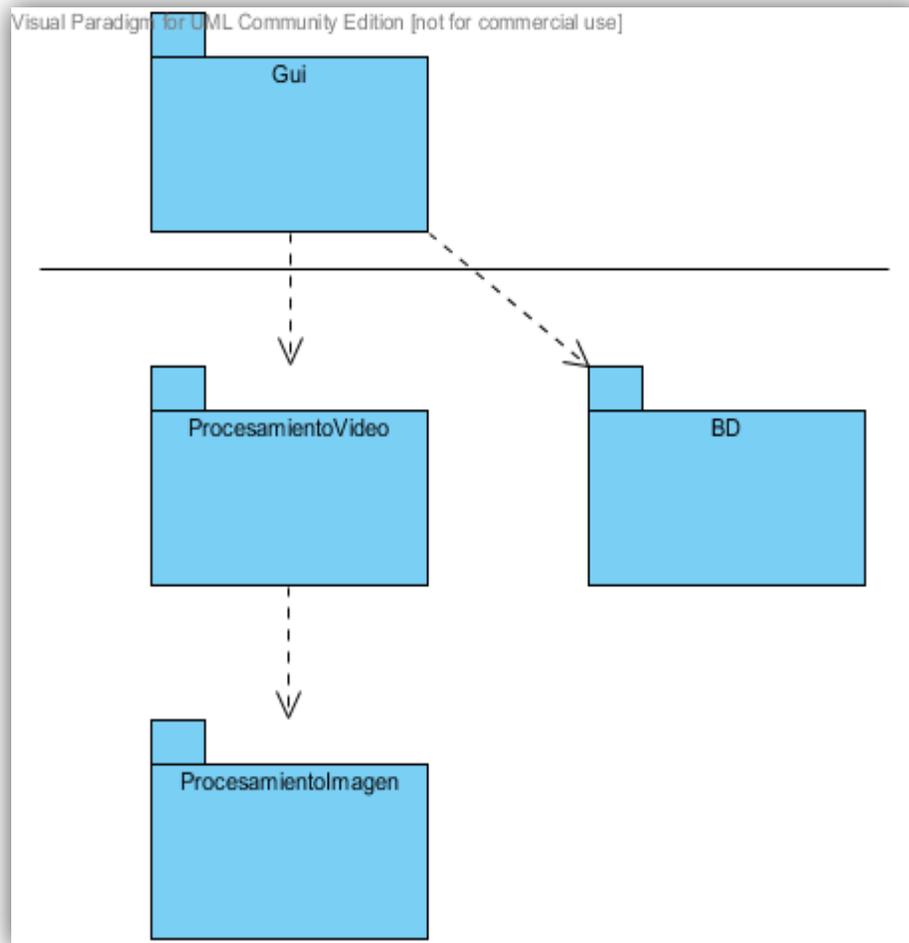


Ilustración 13. Diagrama de Paquetes

CAPÍTULO 6 Primer Prototipo

6.1 Análisis

Como se puede observar en la Tabla 18, este primer prototipo debe cubrir los siguientes puntos:

- Cargar un video al sistema
- Extracción de fotogramas del video
- Aplicación de filtros (pre-procesamiento)
- Nivelar contraste y brillo
- Pasar la imagen a escala de grises
- Aplicación de los algoritmos de detección de movimiento
- Segmentación de las imágenes

Objetivo: Desarrollar el módulo del sistema que se encargue de proporcionar una imagen segmentada.

El proceso principal del sistema a desarrollar es la segmentación de las imágenes para identificar los vehículos. Las estrategias más comunes para la detección pueden dividirse en dos; las basadas en movimiento y las basadas en la sustracción del fondo; en muchos casos, las dos estrategias son utilizadas de forma conjunta para obtener un sistema más robusto.

A continuación se presentan algunos algoritmos y métodos para segmentar las imágenes dinámicas y extraer los objetos de interés.

6.1.1 Restricciones

Una parte fundamental del sistema es la obtención del video por medio de las cámaras de video vigilancia, es por esta razón que es importante que la región de la cual se realizará la detección de movimiento sea la adecuada. Se necesita que la cámara no se encuentre en intersecciones entre calles o avenidas.

Las cámaras de video vigilancia están colocadas a alturas diferentes, por lo tanto se recomienda que tengan un ángulo respecto al plano horizontal de $-45^\circ \pm 30^\circ$.

Una limitación del sistema es que los videos tendrán que ser tomados durante el día, para contar con buenas condiciones de iluminación.

Los videos que se carguen en el sistema, deberán tener un formato AVI, debido a que es el formato que mejor compatibilidad presenta con las bibliotecas de OpenCV.

6.1.2 Algoritmos, métodos y técnicas

6.1.2.1 Algoritmo propuesto por Blake y Kazimianec

Este algoritmo remueve todos los objetos estáticos de la imagen dejando solo vehículos y algunos detalles, que van cambiando de fotograma a fotograma.

El fondo $B(p)$ es calculado como un promedio de cada valor RGB para el mismo punto p en la imagen en el fondo seleccionado de los fotogramas:

$$B(p) = \sum_k \frac{I_B(k, p)}{n}$$

Ecuación 3. Aproximación del fondo

Donde $I_B(k, p)$ es un valor de un pixel en color para un punto p en el fotograma k .

El fondo removido del tráfico de la escena de la imagen $I(k, p)$ genera una imagen en color RGB

$$D(k, p) = \{ I(k, p) - B(p) \}$$

Ecuación 4. Distancia Euclidiana del fondo

Como una distancia Euclidiana entre $I(k, p)$ y $B(p)$.

Filtración de manchas y ruido

Después de la eliminación del fondo la imagen presenta muchos puntos causados por el ruido, que podría ser removido solo por una filtración. Un método efectivo es filtro por umbral. Hay dos principales variaciones de éste método:

- Umbral fijo – es determinado por el conocimiento empírico de una secuencia de pruebas
- Histograma derivado del umbral – este último es seleccionado de un histograma del brillo en la región de segmentación

El primer algoritmo se adapta mejor y éste genera adicionalmente una imagen:

$$M(k, p) = D(k, p) \text{ si } D(k, p) > \text{umbral y } M(k, p) = 0 \text{ para el resto}$$

Rastreo del vehículo

Para rastrear el vehículo en la secuencia del video, se debe marcar la imagen de alguna forma. Una de las formas es marcar el centro del objeto que es calculado como:

$$x_c = \frac{\sum x_j}{n}, \quad y_c = \frac{\sum y_j}{n}$$

Ecuación 5. Calculo del Centro del Vehículo

Donde: x_c y y_c son las coordenadas del centro del vehículo y x_j, y_j son coordenadas de una de los n puntos de imágenes de la zona limitada por el contorno externo del vehículo. Para rastrear el vehículo se deben calcular todas las distancias d_k entre el vehículo de la imagen en el fotograma n y el vehículo en todas las imágenes en el fotograma n+1 usando coordenadas (x_k, y_k) de sus centros:

$$d_k = [(k_k - x_c)^2 + (y_k - y_c)^2]^{\frac{1}{2}}$$

Ecuación 6. Distancia de vehículos entre fotogramas

Calculando $d = \min(d_k)$ rastreamos el vehículo en el fotograma n+1. Aplicando este método a todos los fotogramas, se monitoreara el vehículo en la secuencia del video.

Extracción del contorno

Para poder localizar el vehículo en la zona de observación del fotograma, utilizaremos el algoritmo de detección de bordes basado en la calculación de las imágenes derivadas.

El cálculo de la primera derivación intenta detectar la velocidad en el cambio de las intensidades en los fotogramas de la imagen. La segunda derivación es usada para determinar la ubicación de la máxima velocidad de cambio. Para reducir saltos derivados dentro y cerca de la zona del contorno inicial, la imagen es suavizada.

Para poder utilizar este método, necesitaremos del gradiente en la dirección X y el gradiente en la dirección Y, que están dados por:

$$S_x = (a_2 + ca_3 + a_4) - (a_0 + ca_7 + a_6),$$

$$S_y = (a_0 + ca_1 + a_2) - (a_6 + ca_5 + a_4),$$

Ecuación 7. Gradientes en dirección 'X' y 'Y'

Donde:

$M = (S_x^2 + S_y^2)^{\frac{1}{2}}$, $c = 2$ y a_i = mascara cubierta por un área de 3x3 pixeles. La asignación de a_1, a_2, \dots, a_7 se hace en sentido a las manecillas del reloj, empezando por la esquina superior izquierda de la máscara. Si $M > \text{umbral}$ entonces $M(k,p) = 1$ de lo contrario $M(k,p) = 0$.

El "kernel" es aplicado a casi todos los pixeles de la imagen excepto a la zona del contorno; la imagen resultante es reducida en 2 pixeles en cada dirección, debido a los efectos del contorno.

Otro algoritmo para extraer el contorno es el de “LoG (Laplacian of Gaussian)”, el cual está dado por:

$$I(i, j) = \sum_{l, k} S(l, k) \times G(i - l, j - k), \text{ donde } G(i, j) = \frac{\pi A}{2} \times \exp\left(-\frac{i^2 + j^2}{2A^2}\right)$$

Ecuación 8. Algoritmo Laplacian of Gaussian

Donde A es un ancho de la función de Gauss G.

6.1.2.2 Flujo Óptico

El flujo óptico mide el movimiento de cada pixel a partir de la comparación entre el fotograma de video actual y el fotograma de video siguiente, ubicando la variación espacial y temporal del patrón de brillo de la imagen.

Los algoritmos de interpretación de secuencia de imágenes que requieren la extracción del flujo óptico, están basados en dos métodos, el método de los gradientes y el método de Segmentación.

La técnica de Lukas y Kanade para la estimación de flujo óptico consiste en utilizar un método de multiresolución o pirámides que compara los pixeles del fotograma actual con el siguiente mediante una ventana que aumenta su área de comparación cada vez que realiza una iteración hasta que la ventana llega a tener la misma área que el fotogramas de imagen.

Los datos arrojados por la función de flujo óptico de Lukas y Kanade de la librería OpenCV, son analizados y procesados mediante un algoritmo que determina el desplazamiento del pixel encontrado mediante la relación de la posición escalar del pixel en el fotograma actual con la posición escalar del mismo pixel en el fotograma siguiente, arrojan dos tipos de información, magnitud y ángulo del flujo óptico en un plano coordenado.

Una de las ventajas de este algoritmo es que puede realizar un estimado de la velocidad de modo directo ya que se puede interpolar la trayectoria de cada uno de esos puntos de interés lo que podría facilitar la agrupación de puntos a un vehículo determinado, pero dos o más objetos que posean velocidades similares serán tomados como uno solo.

La desventaja es que al depender la información del movimiento, este algoritmo es susceptible a fallos ambientales de poca velocidad o con objetos detenidos.



Ilustración 14 Pruebas del Algoritmo de Flujo Óptico

6.1.2.3 Fondo por media aumentada (FMA)

En la detección de vehículos, se tiene como constante la vía por la que transitan, además de elementos como postes, alcantarillas, manchas, etc. Y un dato importante es el color del pavimento, que por lo general se encuentra en tonos grises (entre 30 y 70 en los canales RGB).

Existe un algoritmo de fondo llamado FGDStatmodel, el cuál se encuentra disponible en la biblioteca de OpenCV. La intención del algoritmo es realizar un promedio fotograma por fotograma que cumpla con la característica de que sea similar al fondo esperado, lo que hace que se le asigne un mayor peso en el promedio a realizar y tienda rápidamente a estos valores.

La estimación de fondo definitiva que se toma, realiza la media aritmética de cada uno de los fotogramas. Con este método se llega a la convergencia rápida del fondo.

Se realiza un cambio al algoritmo que consiste en modificar la manera de escoger los fotogramas para actualizar el modelo estadístico, actualizándolos solo cuando detectan movimiento a partir de una resta de fotogramas y aplicando un umbral, se obtiene una gran mejora en la estimación de fondo.

El algoritmo FGDStatmodel corresponde a un conjunto de algoritmos de Detección estadística de primer plano, consiste en una parte de estimación estadística por ocurrencias de color y otra parte de estimación de umbrales de movimiento.

FGDStatmodel tiene el inconveniente de no detectar objetos que tengan un color parecido al fondo, los vidrios en general y los vehículos de un color similar al pavimento o concreto confunden a este algoritmo.

Posteriormente, se aplica un algoritmo de bordes (Máscara de Bordes de Fondo), capaz de detectar cambios en la imagen lo suficientemente sutiles para poder detectar la presencia de un automóvil, incluso si es de un color parecido al del fondo, esto debido a las sobras y a la iluminación. Con el fondo ya calculado, periódicamente se sacan los bordes de este fondo, se aplican diferentes operaciones morfológicas, y se invierte con el objetivo de tener una máscara de bordes del fondo, para poder substraer de los bordes del fotograma los bordes del fondo y tener solamente los vehículos.

De esta manera se hace la estimación de fondo con el algoritmo FGDStatmodel, para posteriormente hallar los bordes propios del fondo y de esta manera no tenerlos en cuenta en el procesamiento de los bordes del fotograma.

Los resultados de la unión de estos algoritmos es la obtención de una imagen con el fondo y una imagen binarizada con el reconocimiento de los vehículos.

Algoritmo de detección de Movimiento	Ventajas	Desventajas
Algoritmo propuesto por Blake y Kazimianec	Reconocimiento rápido de los objetos si se tiene una imagen previa del fondo.	Ocupa una imagen previa del fondo.
Flujo Óptico	Puede realizar un estimado de la velocidad de modo directo ya que se puede interpolar la trayectoria de cada uno de esos puntos de interés lo que podría facilitar la agrupación de puntos a un vehículo determinado.	Dos o más objetos que posean velocidades similares serán tomados como uno solo. Al depender la información del movimiento, este algoritmo es susceptible a fallos ambientales de poca velocidad o con objetos detenidos.
Fondo por media aumentada	Corresponde a un conjunto de algoritmos de Detección estadística de primer plano, consiste en una parte de estimación estadística por ocurrencias de color y otra parte de estimación de umbrales de movimiento.	Tiene el inconveniente de no detectar objetos que tengan un color parecido al fondo, los vidrios en general y los vehículos de un color similar al pavimento o concreto confunden a este algoritmo.

Tabla 19 Comparación de Algoritmos de Detección de Movimiento

6.2 Diseño

Después de haber definido los algoritmos de detección de movimiento, se procede al diseño del primer prototipo, donde se hace énfasis en la adquisición del video, la separación de fotogramas y la aplicación de dichos algoritmos.

FLUVER SYSTEM

6.2.1 Diagrama de clases

En el diseño general del sistema se puede observar que existen dos paquetes importantes: ProcesamientoVideo y ProcesamientoImagen. A continuación se muestran tales paquetes con sus clases.

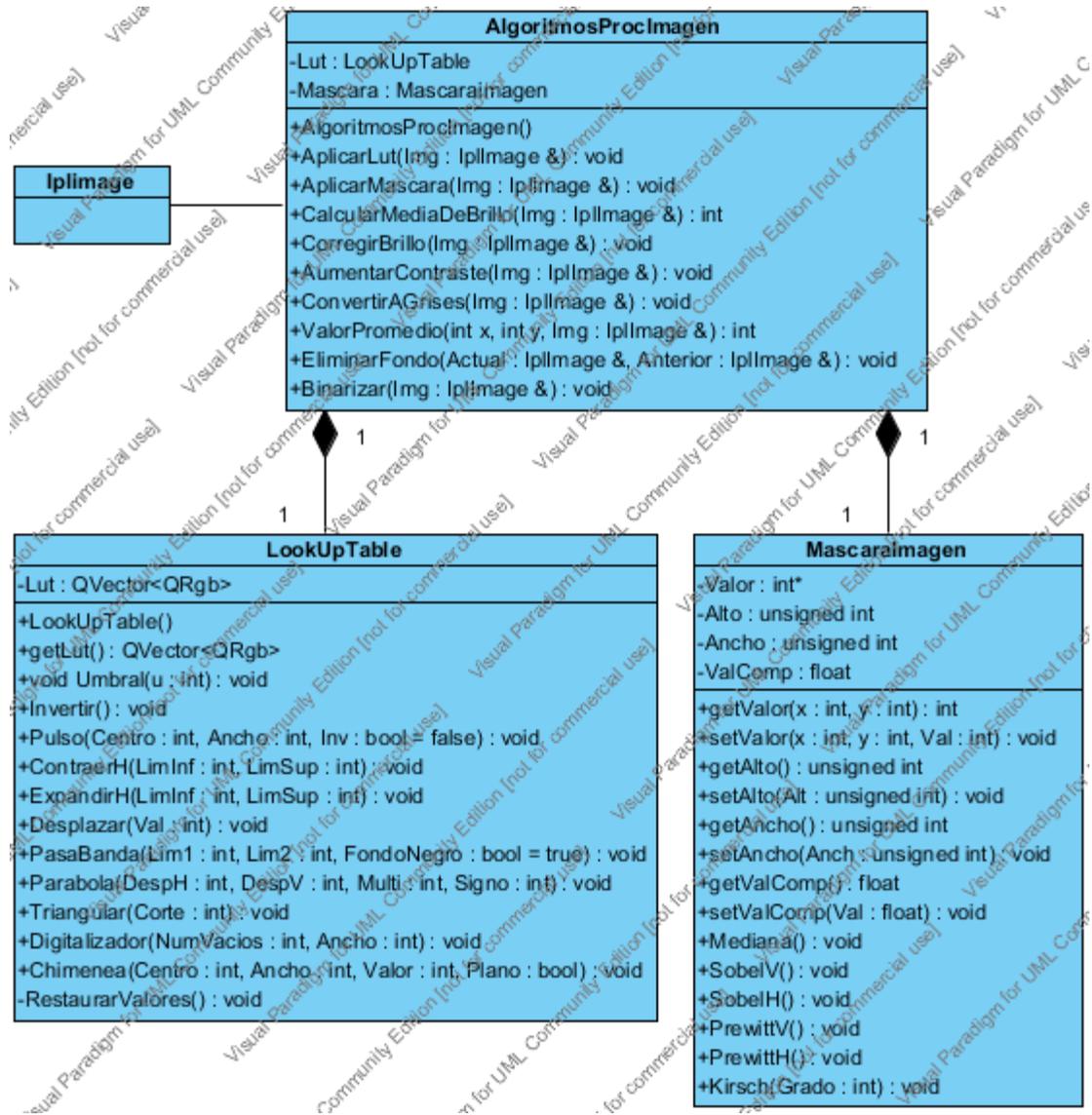
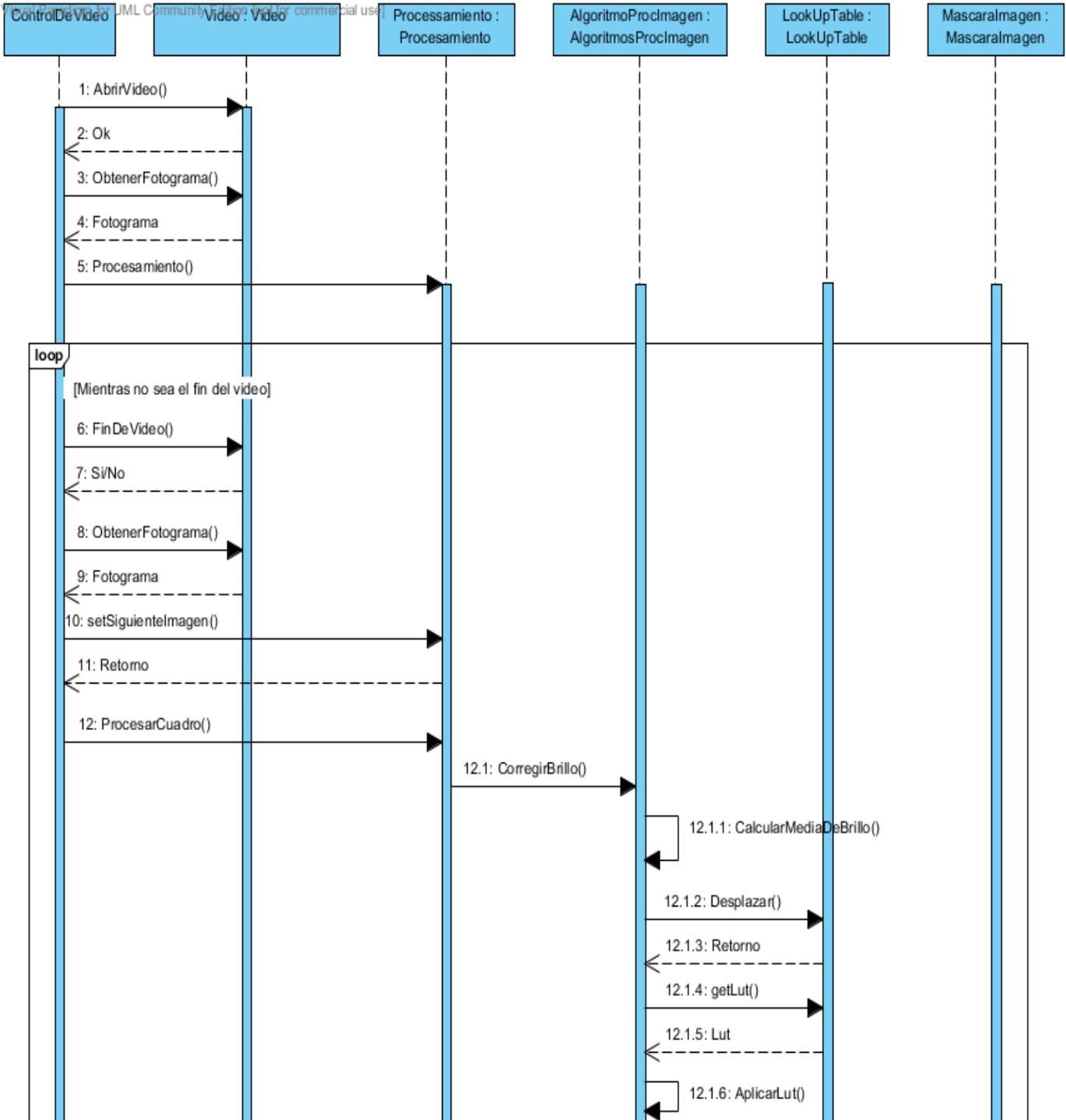


Ilustración 15. Diagrama de Clases: ProcesamientoImagen

En la documentación técnica se hace referencia a cada una de las clases y se explica de forma breve su funcionamiento.

6.2.2 Diagrama de secuencia

En el siguiente diagrama se detalla que mensajes y que objetos son los que interactúan en el proceso desde la carga del video al sistema hasta la obtención de la imagen segmentada.



FLUVER SYSTEM

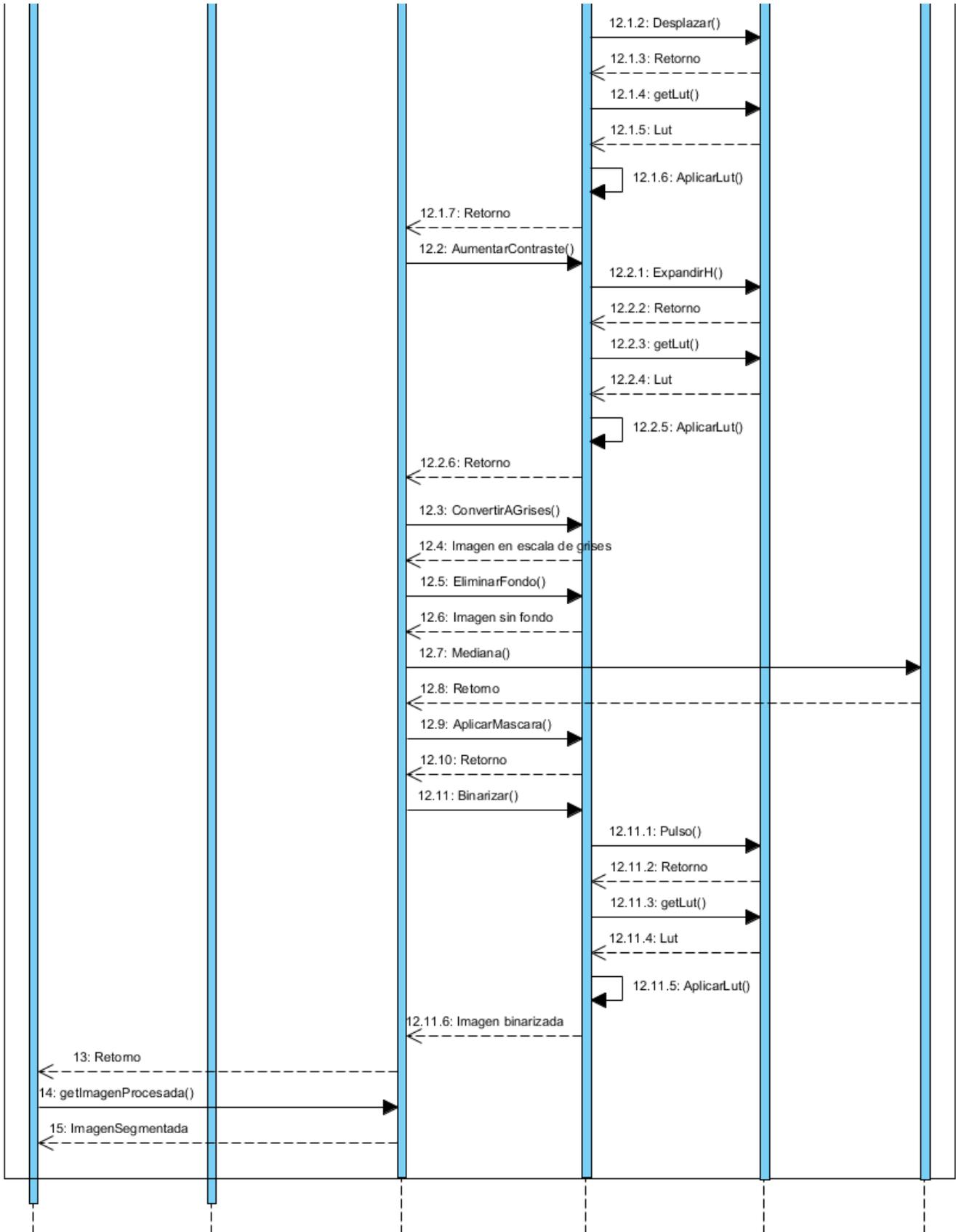


Ilustración 16. Diagrama de secuencia

6.3 Desarrollo

6.3.1 Adquisición del video

El primer paso de este sistema consiste en la adquisición de videos y su posterior separación en los fotogramas que lo conforman, como fue mencionado anteriormente, se propuso el uso de las bibliotecas de OpenCV para esta tarea, ya que brindan portabilidad y compatibilidad, tanto para la carga de archivos de video como con la conexión y adquisición de señales de video desde cámaras de video.

```
#include <cv.h>
#include <highgui.h>

void CapturarVideo(const char* pathVideo)
{
    CvCapture* capturador = cvCreateFileCapture(pathVideo);
    IplImage* fotograma;
    do
    {
        fotograma = cvQueryFrame(capturador);
    }while(0 != fotograma);
}
```

Ilustración 17: Código de ejemplo para la captura de un video y la adquisición de sus fotogramas.

Como puede apreciarse la adquisición del archivo de video resulta bastante simple mediante el uso de OpenCV, solo existen dos factores que deben ser tomados en cuenta para el desarrollo posterior, estos son:

1. Minimización de copia de memoria: El método `cvQueryFrame` se encarga de capturar un nuevo fotograma de una entrada de video, así como de regresar un apuntador a ese fotograma, como es bien sabido la copia excesiva de memoria es un proceso que merma drásticamente el rendimiento de los sistemas. En atención a este punto el método `cvQueryFrame` regresa un apuntador a la memoria interna de la variable `capturador` que contiene los datos del último fotograma cargado. De esta forma se evita copiar la memoria del último fotograma.
2. Representación de canales de color: Para la correcta interpretación de los fotogramas en los procesos posteriores, es necesario conocer el formato u orden en el que se almacenan los datos los canales de una imagen dentro de la variable `fotograma`, se utiliza el modelo de color BGR que asigna un Byte para cada canal de color.

6.3.2 Método análisis del movimiento por diferencia de imágenes

Una vez que se obtienen los fotogramas el siguiente paso es la detección del movimiento en estos fotogramas, para esto se propone el método de análisis del movimiento por diferencia de imágenes planteado por Gonzalo Pajares y Jesús M. de la Cruz [44]. Este método es la base para el algoritmo de detección de movimiento, esta comparación se efectúa entre dos imágenes

diferentes de una secuencia de imágenes, esta comparación se basa en el hecho de que existen dos tipos de objetos dentro de la secuencia de imágenes:

- Objetos estáticos: Como su nombre lo indica son objetos que no se mueven dentro de la secuencia de imágenes, como pueden ser: edificios, arboles, letreros, etc.
- Objetos en movimiento: Son los objetos que pueden cambiar su posición durante la secuencia de imágenes, como pueden ser: automóviles, personas, animales, etc.

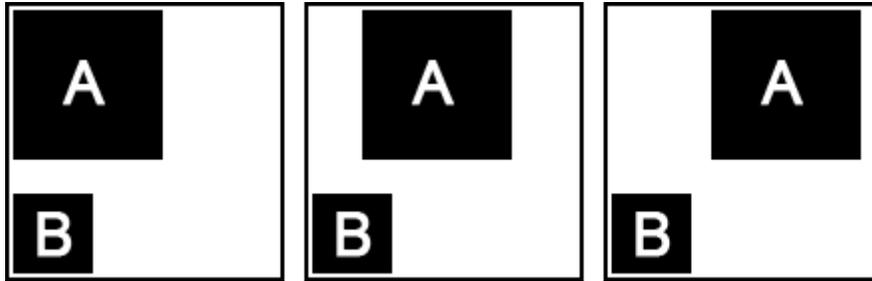


Ilustración 18: Secuencia de imágenes donde el objeto A se desplaza a la derecha, mientras que el objeto B se mantiene estático.

Ya que los objetos estáticos siempre se encuentran en la misma posición, podemos decir que todas las imágenes de una secuencia de imágenes libre de objetos en movimiento siempre serán idénticas. Sin embargo, en una secuencia de imágenes que contenga objetos en movimiento se presentaran cambios, estos cambios en los pixeles pueden ser considerados como los contornos del objeto en movimiento.

$$Img(i,j) = \begin{cases} 0 & \text{Si } Img_1(i,j) \text{ es diferente de } Img_2(i,j) \\ 1 & \text{De otro modo} \end{cases}$$

Ecuación 9: Análisis de movimiento por diferencia de imágenes.

De esta forma, un algoritmo simple para detectar movimientos sería:

1. Tomar dos imágenes de la misma secuencia, la imagen actual y la imagen anterior.
2. Comparar cada pixel de ambas imágenes.
3. Si estos pixeles son diferentes, es un pixel de movimiento, de otra forma es un pixel de fondo.

FLUVER SYSTEM

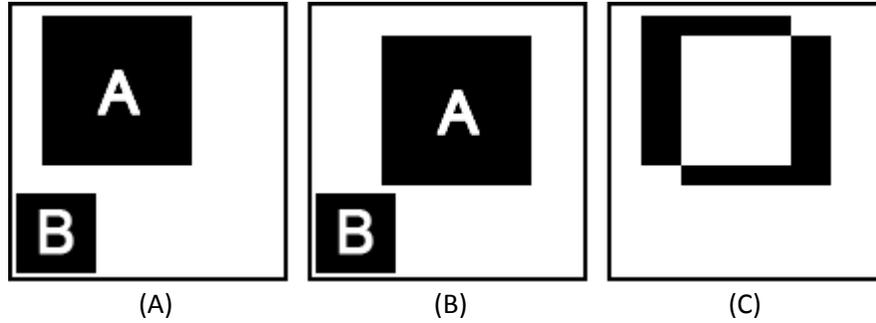


Ilustración 19: (A) Imagen anterior; (B) Imagen actual, con el objeto A en diferente posición; (C) Comparación de las imágenes, las zonas con cambio son marcadas en negro.

Como se nota en la Ilustración 19, las zonas en las que se detectan cambios son los bordes del área estática, de esta forma obtenemos la información sobre que pixeles han cambiado y cuales no, en la Ilustración 20 podemos observar el resultado de la implementación de este algoritmo, el algoritmo fue aplicado al video de una avenida en dos instantes de tiempo diferentes.

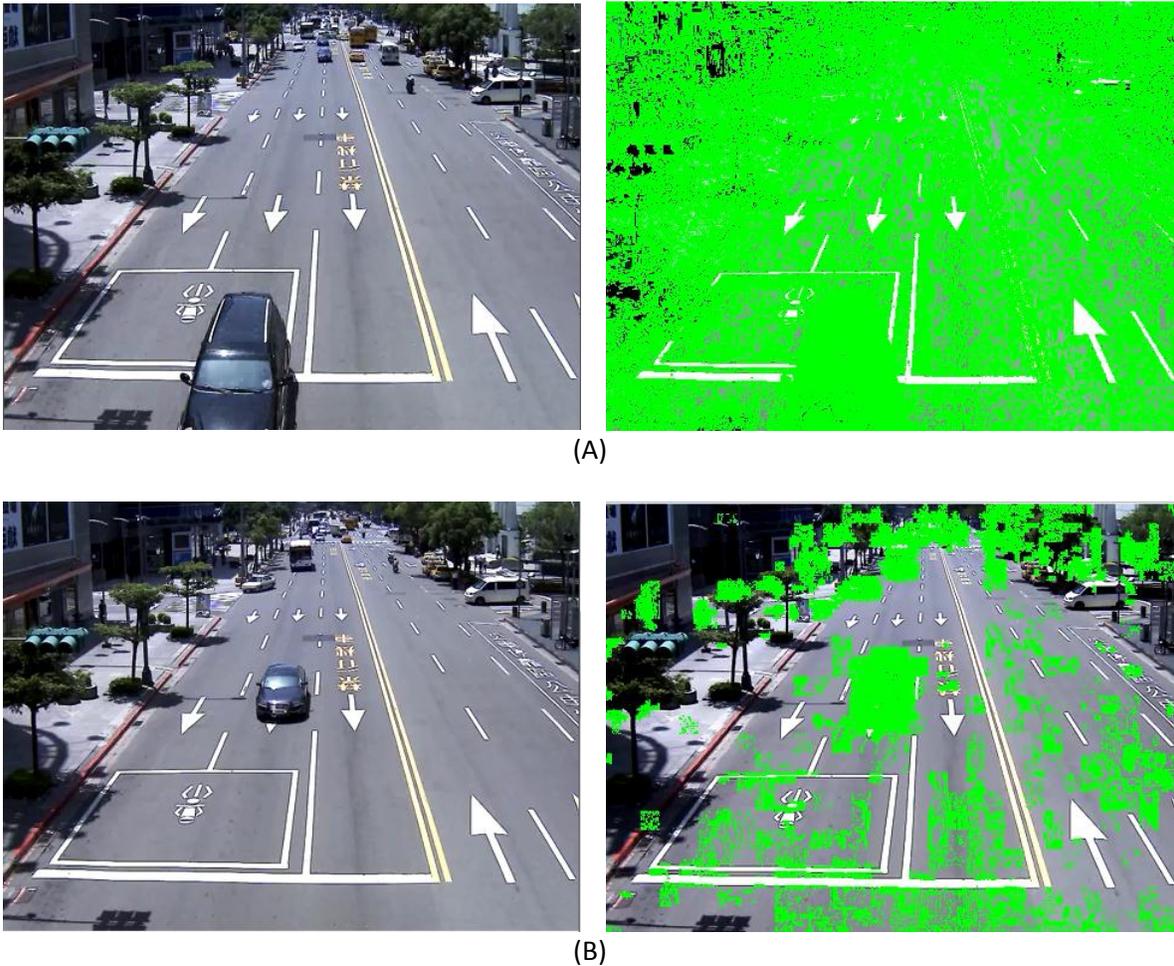


Ilustración 20: Implementación del algoritmo básico, la detección del movimiento es mostrada en verde.

6.3.3 Problemas comunes del algoritmo

La Ilustración 20: Implementación del algoritmo básico, la detección del movimiento es mostrada en verde. muestra los resultados del algoritmo básico de análisis por diferencia de imágenes, como puede notarse, los resultados del algoritmo hasta el momento no son nada buenos, el algoritmo para la detección de movimiento es bastante simple, pero dista mucho de ser completamente fiable por si solo como para implementarlo en una solución final.

El problema de la mala detección del movimiento surge proviene de 2 variables que se asumen constantes, pero en el mundo real nunca se presentan de esta forma. La primera es los cambios de iluminación confunden al algoritmo y la segunda es el movimiento de los objetos estáticos que se suponía nulo.

Cambios de iluminación

Un factor muy importante dentro de un ambiente real es la iluminación, esta cambia constantemente, tanto por el movimiento del sol a lo largo de día como por otros aspectos como nubes, elementos que bloqueen el paso de luz, fuentes de luz adicionales, etc. Esta es una variable muy importante, ya que aunque para una persona es simple reconocer el color de un automóvil tanto en un día soleado como en uno nublado, para una computadora una ligera variación en la iluminación implica colores ligeramente diferentes.

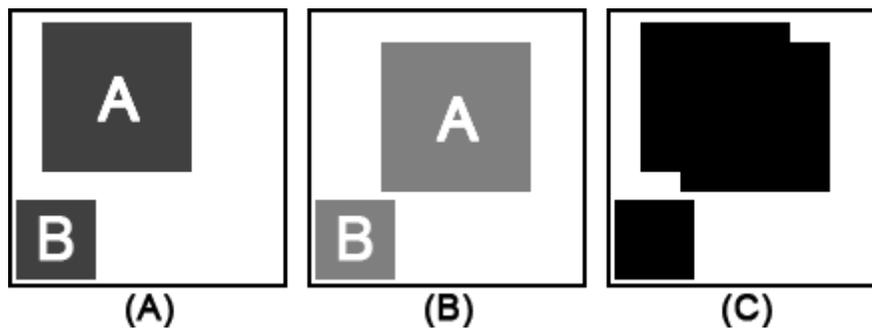


Ilustración 21: (A) Imagen anterior. (B) Imagen actual con cambios en el brillo de los objetos. (C) Comparación de las imágenes las zonas con cambio son marcadas en negro.

Como puede observarse en la Ilustración 21, debido a que el objeto no solo se desplazo, si no que también cambio a un color mas claro, la comparación toma que todo el espacio que ocupaba el objeto en las imágenes anterior y actual ha cambiado, esto es un problema que se presentara dentro de toda la imagen, por lo que incluso los objetos estáticos serian tomados como objetos en movimiento.

Ahora si consideramos que el cambio en la iluminación no suele ser brusco, en intervalos cortos de tiempo, como pasar de la noche a la mañana en un segundo, una solución a este problema es incluir un valor de tolerancia en el momento de la comparación entre pixeles.

De esta forma, para que un pixel sea considerado de movimiento el valor del pixel en la imagen actual debe estar fuera del intervalo comprendido por el valor anterior más o menos la tolerancia.

Ruido en los fotogramas

El ruido en las imágenes es un problema muy común dentro de los medios digitales de captura como lo son las cámaras o videocámaras digitales, este consiste en una alteración de los valores originales de color en los píxeles, debido a la interferencia eléctrica del mismo dispositivo u otros factores externos, el ruido afecta píxeles al azar modificando sus valores de color.



Ilustración 22: Imagen original a la izquierda e imagen contaminada con ruido a la derecha.

Como es visible en la imagen anterior, aunque las formas son aun reconocibles la imagen se ve claramente alterada, esto representa un problema en la detección de movimiento, ya que al afectar a cualquier píxel de forma aleatoria, el ruido puede también darse en los objetos estáticos, lo que tendrá como consecuencia que el ruido mismo sea detectado como un pequeño objeto en movimiento.

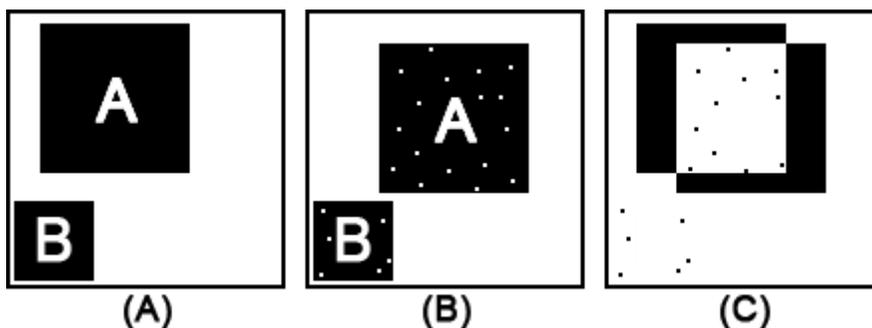


Ilustración 23: (A) Imagen anterior. (B) Imagen actual contaminada con ruido. (C) Resultado de la comparación entre imágenes.

Actualmente existen varias técnicas para reducir el ruido en las imágenes, estas técnicas tienen el objetivo de recuperar, en el mayor grado posible la imagen original, estas técnicas pueden separarse en dos grandes grupos:

FLUVER SYSTEM

- Filtros lineales
- Filtros no lineales

Los filtros lineales con frecuencia suelen dar mejores resultados, sin embargo estos filtros implican convoluciones entre imágenes y el cálculo de la derivada de la imagen, procesos que son costosos computacionalmente. Los filtros no lineales por otro lado suelen ser mas simples, pero a costa de ser menos eficaces en la eliminación del ruido.

Es importante recordar que dentro del contexto de nuestro algoritmo, no es una prioridad recuperar la imagen original, si no, evitar el efecto del ruido en la detección del movimiento, una solución simple para este problema es realizar un muestreo, en lugar de solo ocupar el valor del pixel que estamos verificando, podemos muestrear el valor de los pixeles adyacentes, para calcular el promedio del color para esa posición.

De esta forma, aunque el pixel que estamos comparando haya sido afectado por ruido, el promedio no se vera afectado tan severamente, sumado a la tolerancia que se ocupa para evitar los cambios de iluminación, el efecto del ruido se evita considerablemente.

Como puede observarse en la Ilustración 24 mientras mayor es el valor de tolerancia mejor resulta la detección del movimiento, sin embargo un valor demasiado alto repercute en que los cambios necesarios para la detección del movimiento entre fotogramas deban ser más grandes, es decir, entre mayor sea la tolerancia mayor debe ser el movimiento para detectarlo. Por este motivo se decidió utilizar el valor de tolerancia de 32, ya que aproxima bastante bien los cambios que ocurren en la escena.

FLUVER SYSTEM

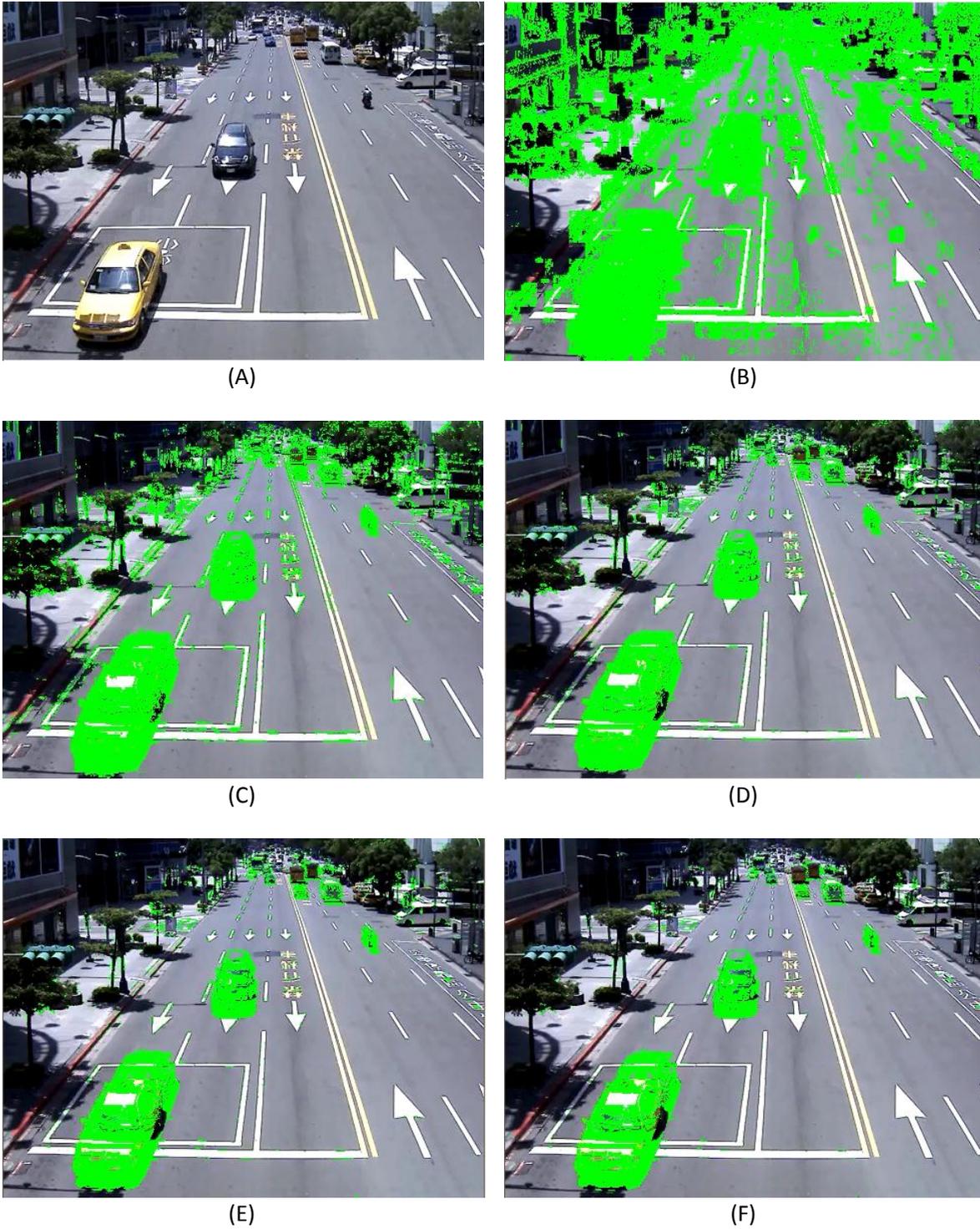


Ilustración 24: (A) Video original; (B) Detección de movimiento básica; (C), (D), (E) y (F), Detección de movimiento considerando una vecindad de 3x3 y con un valor de tolerancia de 8, 16, 24 y 32 respectivamente.

6.3.4 Supresión del ruido residual

A pesar de las técnicas descritas anteriormente para la supresión del ruido, así como los cambios en la iluminación del ambiente, es posible notar zonas en las que aun es posible detectar movimiento pero que no son útiles para nuestros propósitos, elementos como ramas de arboles que se mueven con el viento, reflejos en cristales o personas caminando. Pese a esto es notorio que el movimiento detectado por estos motivos es mínimo, además de mostrarse en secciones pequeñas y en su mayoría de forma similar al ruido, más que como áreas completas. Esto es un punto que nos facilitara considerablemente su supresión.

Ya que las zonas que deseamos eliminar en la detección de ruido resultan pequeñas y aisladas es posible eliminarlas mediante la operación morfológica de la erosión, esta operación esta implementada de la siguiente forma, si pixel ha sido detectado como de movimiento y además es colindante con un pixel que ha sido detectado como de fondo, entonces el pixel es interpretado como un pixel de fondo.

Debido a que las zonas detectadas como de movimiento, y que además son importantes para nosotros se muestran como áreas grandes dentro de la imagen es seguro aplicar esta operación para la supresión del ruido residual, ya que aunque también erosionaremos de forma parcial las zonas que nos interesan, estas no desaparecerán ni se volverán demasiado pequeñas, por lo que es posible recuperarlas posteriormente

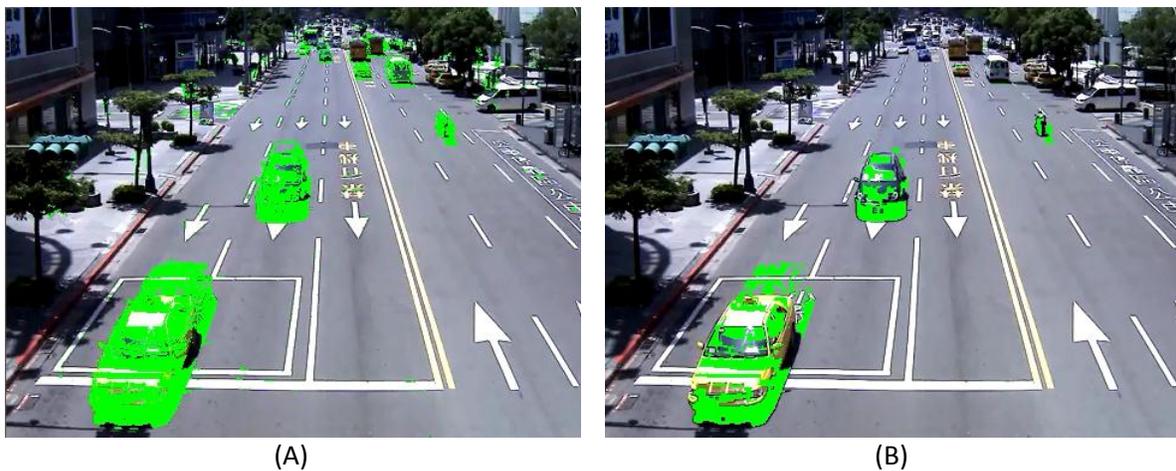


Ilustración 25: (A) Detección de movimiento considerando una vecindad de 3x3 y un valor de tolerancia de 32; (B) Resultado de aplicar la operación de erosión a la imagen (A).

Como es posible observar en la Ilustración 25, el ruido que se encontraba en arboles y algunos contornos ha sido suprimido, esto tras una sola aplicación de la operación de erosión. Aunque la detección de movimiento que nos interesa también ha sido erosionada, aun es visible y detectable.

6.4 Implementación de Pruebas

Iteración	Fecha de ejecución	Analistas de la prueba	Estado de la prueba	Comentarios	
2	9 / 11 / 2011	Morales Báez Jorge Blanco Álvarez Brenda	Concluido	Se implementó un aviso que se muestra mientras el video se está procesando.	
Nombre de la prueba		CP01: Prueba de adquisición de video			
Objetivo		Mostrar que el sistema es capaz de cargar un archivo con extensión avi o mp4 y accede a cada uno de los fotogramas, comprobándolo por medio de una copia del video original.			
Requisitos		<ul style="list-style-type: none"> • Un video en formato avi o mp4. • Sistema operativo Windows 7. • Programa ejecutable "CP01_a.exe" compilado con las .dll para poder ejecutarlo en cualquier máquina: C:\Qt\2010.02.1\mingw\bin libgcc_s_dw2-1.dll mingwm10.dll C:\Qt\2010.02.1\qt\bin QtCored4.dll QtGui4.dll • Instalar el framework Qt 4.7 en la máquina en la que se ejecutarán las pruebas si es que no se cuenta con las dll anteriores. • Framework de OpenCV 2.0. • Se utilizará una memoria USB en la cual se almacenará la carpeta "CP01", la cual contendrá el ejecutable del programa con el que se realizarán las pruebas. 			
Valores a ingresar		Un video en formato avi o mp4.			
Resultados esperados		El programa genera otro video idéntico al original.			
ID	Acciones	Resultados esperados	Correcto	Incorrecto	Comentarios
1	Conectar la memoria a la computadora.	La computadora reconoce al dispositivo y ofrece la opción de ver el contenido de esta.	X		
2	Elegir la opción de ver el contenido.	Se muestra una ventana con el contenido de la memoria y	X		

FLUVER SYSTEM

		aparece la carpeta "CP01".			
3	Acceder a la carpeta con nombre "CP01".	Se muestra una ventana con el contenido de la carpeta.	X		
4	Ejecutar el archivo "CP01\CPO1_a.exe"	Se muestra una ventana de diálogo de windows para abrir un archivo.	X		
5	Seleccionar el archivo de video "CP01\CP01.avi" que se encuentra en la misma carpeta y dar clic en Aceptar.	La ventana se cierra y se muestra una ventana de dialogo donde se informa el momento en el cual se ha procesado el video exitosamente.	X		
6	Abrir el archivo que se ha generado en la ruta "CP01\1.avi" con un doble clic.	Se abre el reproductor Windows Media y se reproduce el archivo de video.	X		
7	Abrir el archivo "CP01\CP01.avi" y compararlo con el que fue generado por el programa.	Los videos deben tener las mismas características en cuanto a duración, peso y lo que muestran en pantalla.	X		

Tabla 20. Prueba de adquisición de video

FLUVER SYSTEM

Iteración	Fecha de ejecución	Analistas de la prueba	Estado de la prueba	Comentarios	
3	18 / 11 / 2011	Morales Báez Jorge Blanco Álvarez Brenda	Concluida	Se cierran las secciones abiertas usando repetidamente filtros de dilatación.	
Nombre de la prueba		CP02: Prueba de adquisición de máscara de movimiento			
Objetivo		Mostrar que el sistema es capaz de detectar los elementos en movimiento de un video.			
Requisitos		*Los mencionados en la prueba CP01.			
Valores a ingresar		Un video en formato avi o mp4.			
Resultados esperados		Un video que muestre en negro el área de los objetos en movimiento y el resto en blanco.			
ID	Acciones	Resultados esperados	Correcto	Incorrecto	Comentarios
1	Conectar la memoria a la computadora.	La computadora reconoce al dispositivo y ofrece la opción de ver el contenido de esta.	X		
2	Elegir la opción de ver el contenido.	Se muestra una ventana con el contenido de la memoria y aparece la carpeta "CP02".	X		
3	Acceder a la carpeta con nombre "CP02".	Se muestra una ventana con el contenido de la carpeta.	X		
4	Ejecutar el archivo "CP02\CP02_a.exe"	Se muestra una ventana de diálogo de Windows para abrir un archivo.	X		
5	Seleccionar el archivo de video "CP02\CP02.avi" que se encuentra en la misma carpeta y dar clic en Aceptar.	La ventana se cierra y se muestra una ventana de dialogo donde se informa el momento en el cual se ha procesado el video exitosamente.	X		
6	Abrir el archivo que se ha generado en la ruta "CP02\1.avi" con un doble clic.	Se abre el reproductor Windows Media y se reproduce el archivo de video. El video "1.avi" debe mostrar en negro los elementos en movimiento.	X		Los segmentos en movimiento se juntan.

Tabla 21. Prueba de adquisición de máscara de movimiento

Cálculo del flujo vehicular mediante segmentación de imágenes.

FLUVER SYSTEM

Iteración	Fecha de ejecución	Analistas de la prueba	Estado de la prueba		Comentarios
3	25 / 11 / 2011	Morales Báez Jorge Blanco Álvarez Brenda	Concluida		Se implementa una serie de dilataciones para cerrar las secciones abiertas en las zonas correspondientes a las sombras de los vehículos.
Nombre de la prueba		CP03: Prueba de detección de automóviles por su sombra.			
Objetivo		Mostrar que el sistema es capaz de detectar las sombras proyectadas por los automóviles.			
Requisitos		*Los mencionados en la prueba CP01.			
Valores a ingresar		Un video en formato avi o mp4.			
Resultados esperados		Un video que muestre en negro la sombra de los automóviles.			
ID	Acciones	Resultados esperados	Correcto	Incorrecto	Comentarios
1	Conectar la memoria a la computadora.	La computadora reconoce al dispositivo y ofrece la opción de ver el contenido de esta.	X		
2	Elegir la opción de ver el contenido.	Se muestra una ventana con el contenido de la memoria y aparece la carpeta "CP03".	X		
3	Acceder a la carpeta con nombre "CP03".	Se muestra una ventana con el contenido de la carpeta.	X		
4	Ejecutar el archivo "CP03\CP03_a.exe"	Se muestra una ventana de diálogo de windows para abrir un archivo.	X		
5	Seleccionar el archivo de video "CP03\CP03.avi" que se encuentra en la misma carpeta y dar clic en Aceptar.	La ventana se cierra y se muestra una ventana de dialogo donde se informa el momento en el cual se ha procesado el video exitosamente.	X		
6	Abrir el archivo que se ha generado en la ruta "CP03\1.avi" con un doble clic.	Se abre el reproductor Windows Media y se reproduce el archivo de video. El video "1.avi" debe mostrar en negro las sombras		X	Se conservan varias secciones del fondo que tenían frecuencias bajas, sin embargo se aíslan satisfactoriamente los vehículos.

Tabla 22. Prueba de detección de automóviles por su sombra

FLUVER SYSTEM

Iteración	Fecha de ejecución	Analistas de la prueba	Estado de la prueba		Comentarios
2	30 / 11 / 2011	Morales Báez Jorge Blanco Álvarez Brenda	Concluida		Se implementa una zona de interés que aisle los automóviles lejanos.
Nombre de la prueba		CP04: Prueba de segmentación de automóviles.			
Objetivo		Mostrar el resultado al juntar la detección de movimiento y la detección de sombras.			
Requisitos		<ul style="list-style-type: none"> Los mencionados en la prueba CP01. 			
Valores a ingresar		Un video en formato avi o mp4.			
Resultados esperados		Un video que muestre en negro únicamente la sombra de los automóviles.			
ID	Acciones	Resultados esperados	Correcto	Incorrecto	Comentarios
1	Conectar la memoria a la computadora.	La computadora reconoce al dispositivo y ofrece la opción de ver el contenido de esta.	X		
2	Elegir la opción de ver el contenido.	Se muestra una ventana con el contenido de la memoria y aparece la carpeta "CP04".	X		
3	Acceder a la carpeta con nombre "CP04".	Se muestra una ventana con el contenido de la carpeta.	X		
4	Ejecutar el archivo "CP04\CPO4_a.exe"	Se muestra una ventana de diálogo de windows para abrir un archivo.	X		
5	Seleccionar el archivo de video "CP04\CP04.avi" que se encuentra en la misma carpeta y dar clic en Aceptar.	La ventana se cierra y se muestra una ventana de dialogo donde se informa el momento en el cual se ha procesado el video exitosamente.	X		
6	Abrir el archivo que se ha generado en la ruta "CP04\1.avi" con un doble clic.	Se abre el reproductor Windows Media y se reproduce el archivo de video. El video "1.avi" debe mostrar en negro los vehículos en movimiento.	X		

Tabla 23. Prueba de segmentación de automóviles

CAPÍTULO 7 Segundo

Prototipo

7.1 Análisis

Una vez realizado el tratamiento y la segmentación del video, se procede a calcular el flujo vehicular y la velocidad promedio.

Objetivo: Calcular el flujo vehicular y la velocidad promedio de la avenida del video segmentado.

7.1.1 Algoritmos

Para realizar el cálculo del flujo vehicular y la velocidad promedio se utilizaron dos algoritmos, los cuales se presentan a continuación.

Algoritmo de Cálculo de Flujo vehicular

El algoritmo utilizado para calcular el flujo vehicular toma una serie de imágenes segmentadas, en estas imágenes se define una sección de interés, en la cual se contarán todos los vehículos que se encuentren en la primera imagen, posteriormente, se calcula una subsección dentro de la sección de interés, dicha subsección servirá para seguir contabilizando los coches que crucen por esa subsección, en el resto de las imágenes segmentadas.

Cálculo de la Subsección

La subsección es una región dentro de la imagen, con un ancho (columnas) del total de la imagen y un largo (filas) calculado de la siguiente forma:

Tomando en cuenta el Artículo 51 de la Ley de Tránsito N° 24.449 [37], el cual establece los límites máximos de velocidad para calles y avenidas de zonas urbanas y rurales; nos dice que la velocidad máxima para las avenidas en zonas urbanas es de $60 \frac{km}{hr}$.

Partiendo de este dato, podemos encontrar cuantos metros avanza por segundo un automóvil, de la siguiente forma:

FLUVER SYSTEM

$$MAS^5 = \frac{60 \text{ km}}{1 \text{ hr}} = 60 \left(\frac{1000 \text{ m}}{3600 \text{ s}} \right) = \frac{50}{3} \text{ m/s} \dots \dots \dots (1) *$$

Ecuación 10. Metros Avanzados por Segundo

Después, calculamos la relación que existe entre 1 pixel y 1 metro en la imagen, para esto se necesita la longitud en metros de la avenida y el tamaño en pixeles de la imagen, ambos dentro de la sección de interés. Calculando esta relación de la siguiente forma:

$$1m = \frac{T_{px}}{T_m} \text{ pixeles} \dots \dots \dots (2)$$

$$1px = \frac{T_m}{T_{px}} \text{ metros} \dots \dots \dots (3)$$

Donde:

T_{px} = Tamaño de la sección de interés en pixeles.

T_m = Longitud de la avenida en metros, dentro de la sección de interés.

Ecuación 11. Relaciones pixel/metro

Teniendo estas ecuaciones, podemos obtener la cantidad de pixeles que avanza un automóvil por segundo a $60 \frac{km}{hr}$, sustituyendo la ecuación 2 en 1, obtenemos:

$$PAS^6 = \left(\frac{50 \text{ m}}{3 \text{ s}} \right) * \left(\frac{T_{px}}{T_m} \right) \text{ px/m} \dots \dots \dots (4)$$

Ecuación 12. Pixeles avanzados por segundo

Debido a que cada frame o imagen que se toma del video, se hace en una fracción de segundo, se tiene que calcular el número de pixeles que un automóvil avanza en dicha fracción de segundo a $60 \frac{km}{hr}$, es decir:

$$PAF^7 = PAS * T_f \dots \dots \dots (5)$$

Ecuación 13. Pixeles avanzados por frame

Donde:

T_f = Tiempo por frame

De esta forma, se obtiene el largo de la subsección que servirá para contabilizar los coches que la atraviesan.

⁵ Metros Avanzados por Segundo

⁶ Pixeles Avanzados por Segundo

⁷ Pixeles Avanzados por Frame

Algoritmo de Cálculo de Velocidad Promedio

El algoritmo de cálculo de velocidad promedio trabaja en tres partes:

1. Define y crea carriles a lo largo de la imagen.
2. Detecta y guarda las coordenadas x, y (Puntos) para cada coche existente en cada carril.
3. Se calcula la velocidad para cada par de puntos, promediándose y obteniendo la velocidad promedio.

La primera parte crea los carriles necesarios, éstos sirven para poder rastrear a los automóviles de acuerdo a su carril.

La segunda parte rastrea los automóviles de acuerdo al carril y guarda un Punto, que son las coordenadas en las que se detecta.

La tercera y última parte, se encarga de calcular la velocidad promedio para cada carril, tomando un par de Puntos, después promedia la velocidad por carril, obteniendo la velocidad promedio existente en la avenida.

Primera Parte: Creación de Carriles.

Objetivo: Detectar, definir y crear carriles para poder rastrear automóviles.

El cálculo de los carriles se hace de la siguiente forma:

Se obtienen dos puntos de dos imágenes, estos dos puntos deben pertenecer al mismo automóvil; con éstos dos puntos se obtiene la pendiente de la recta que define a los carriles, utilizando la siguiente fórmula:

$$m = \frac{\Delta_y}{\Delta_x} = \frac{y_2 - y_1}{x_2 - x_1}$$

Ecuación 14. Pendiente de la recta

Ésta relación nos indica que por cada incremento en y, hay un incremento en x. Para definir la cantidad de carriles que se utilizará:

$$C_c = \frac{V_w}{W_c}$$

Donde:

C_c = Cantidad de carriles

V_w = Ancho del video en pixeles

W_c = Ancho del carril en pixeles

Ecuación 15. Cantidad de Carriles

Con estos datos, solo es necesario posicionarnos en el inicio del primer carril, el cual se puede calcular con el primer punto del automóvil y W_c . De ahí se crean los carriles, y a cada carril se le asocia la información de la pendiente de la recta del carril, así como su límite inferior y superior en x.

Segunda Parte: Detección y creación de puntos

Objetivo: Detectar y crear de puntos, para su posterior procesamiento.

Una vez creados los carriles, se detectarán y guardarán los puntos (automóviles) que aparezcan en cada uno de los carriles, para cada imagen o frame del video.

Para eso se utilizará la pendiente de la recta del carril, el límite inferior y el límite superior, destacando que solo se guardarán los puntos que aparezcan dentro de la sección de interés definida previamente.

Tercera Parte: Cálculo de la velocidad promedio

Objetivo: Calcular la velocidad por cada carril utilizando los puntos guardados y promediar la velocidad por carril para obtener la velocidad promedio.

Una vez calculados los carriles y guardando los puntos rastreados en cada carril para cada frame del video, se calcularán las velocidades para cada par de puntos, utilizando:

$$V = \frac{d}{t} \text{ km/hr}$$

Ecuación 16. Velocidad

Donde:

d: Distancia

t: Tiempo

Es necesario obtener la distancia entre cada par de puntos, cabe mencionar que solo se tomará la distancia positiva para cada par de puntos, debido a que, como se guardan los puntos de cada automóvil desde que entra en la sección de interés del video hasta que sale de la misma, existirán puntos con distancia negativa. Esto solo significará que el rastreo de un automóvil terminó y el rastro de otro automóvil empezó, sobre el mismo carril.

Una vez que se obtiene dicha distancia (medida en pixeles). Se calculará la velocidad en $\frac{px}{s}$, y solo bastará convertirla a $\frac{km}{hr}$, utilizando la relación pixel-metro calculada en el algoritmo de cálculo del flujo vehicular. Esta conversión se realiza a la velocidad promedio final, para evitar un gran número de operaciones innecesarias.

FLUVER SYSTEM

$$\text{Número de Operaciones} = \sum_{n=1}^{C_c} C_n * (PC_n - 1)$$

Donde:
 C_n = Carril n
 PC_n = Puntos del carril n

Ecuación 17. Número de Operaciones

Para convertir la velocidad promedio medida en $\frac{px}{s}$ a $\frac{km}{hr}$, se utilizan las siguientes formulas:

$$1px = \frac{T_m}{T_{px}} \text{ metros}$$

Donde:
 T_{px} = Tamaño de la sección de interés en pixeles.

$$V_{pms} = V_{ppxs} * \frac{T_m}{T_{px}} \text{ m/s}$$

T_m = Longitud de la avenida en metros, dentro de la sección de interés.

V_{pms} = Velocidad promedio en m/s

$$V_{pkmh} = V_{pms} * 3.6 \text{ km/hr}$$

V_{ppxs} = Velocidad promedio en px / s

V_{pkmh} = Velocidad promedio en km/hr

Ecuación 18. Conversión de velocidad de px/s a km/hr

7.2 Diseño

Después de haber analizado lo que se necesita para realizar los algoritmos y su secuencia, se procede a su diseño por medio de un diagrama de clases y de secuencia.

7.2.1 Diagrama de clases

En el siguiente diagrama se muestran las clases empleadas para el desarrollo de los algoritmos:

- **Punto:** Clase que representa un vehículo en forma de punto.
- **ListaPunto:** Clase que se encarga de almacenar puntos, es decir, un arreglo de vehículos.
- **Matriz Velocidad:** Clase que representa la estructura de los algoritmos.
- **Carril:** Clase que calcula la conversión de metros a pixeles y traza los carriles de la imagen.
- **CalculoEstadistica:** Clase principal que ejecuta los algoritmos de flujo vehicular y velo



Ilustración 26. Diagrama de Clases: Cálculo-Estadísticas

7.2.2 Diagrama de secuencia

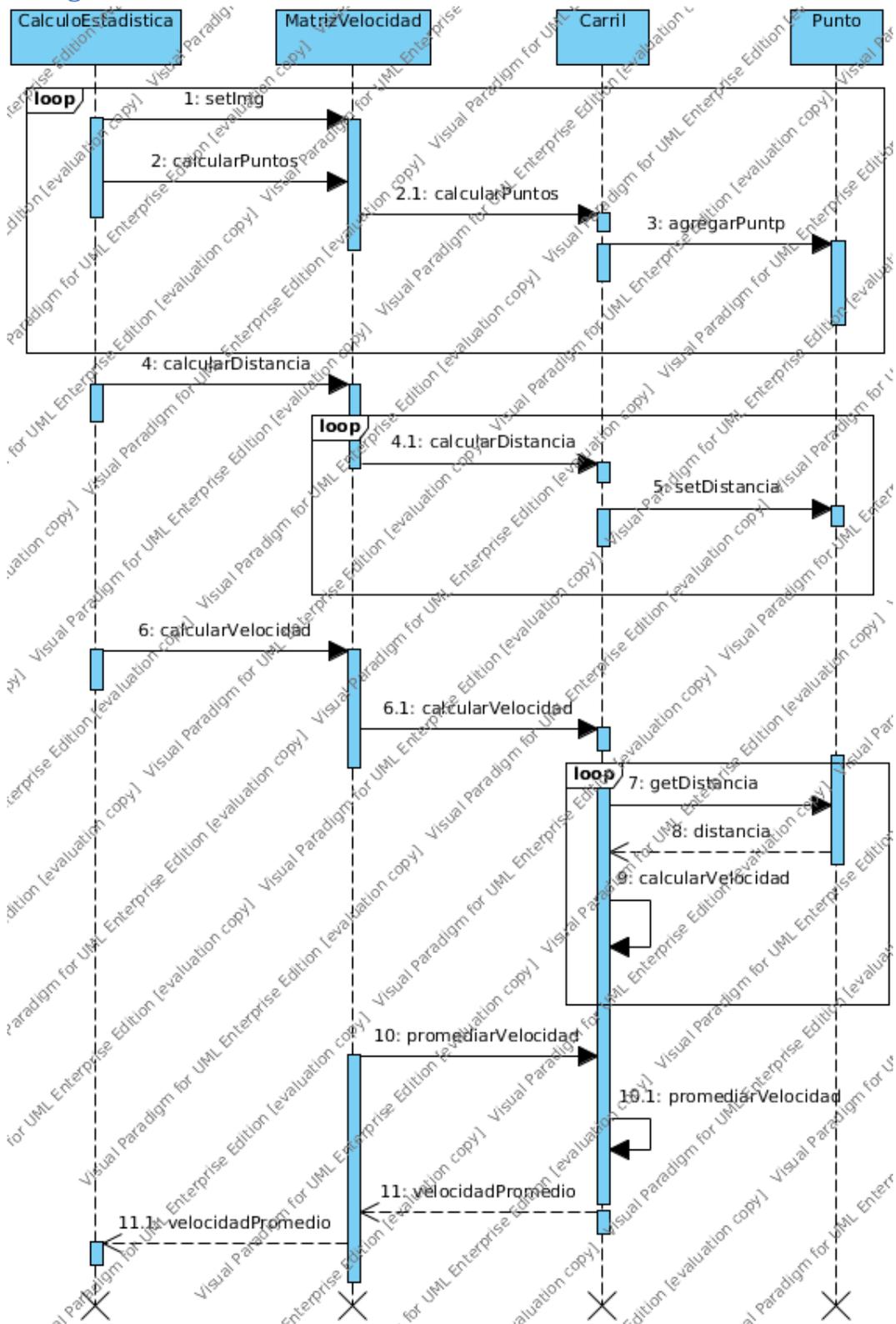


Ilustración 27. Diagrama de secuencia

CAPÍTULO 8 Tercer Prototipo

8.1 Análisis

Como se puede observar en la Tabla 18, el tercer prototipo debe implementar una base de datos con las estadísticas de velocidad y flujo de tránsito correspondientes a cada cámara existente en el sistema y permitir la administración de la información de cada registro.

Objetivo: Desarrollar el módulo del sistema que se encargue de la administración de estadísticas vehiculares obtenidas por medio de los algoritmos de conteo y cálculo de velocidad.

8.1.1 Gestor de Base de Datos



Microsoft SQL Server es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional.

Sus lenguajes para consultas son T-SQL y ANSI SQL. Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle o MySQL.

Microsoft SQL Server 2008 R2 es un conjunto completo de tecnologías y herramientas que ayudan a obtener el máximo valor de la información. Cuenta con excelente rendimiento, disponibilidad y seguridad; se considera una de las herramientas más productivas de administración y desarrollo ya que permite procesamiento de eventos complejos a gran escala y análisis completos con un entorno de Business Intelligence (BI) en modo autoservicio gestionado mediante PowerPivot para Excel y PowerPivot para SharePoint [38].

Características:

- Soporte de transacciones
- Escalabilidad, estabilidad y seguridad
- Soporta procedimientos almacenados
- Incluye un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.

FLUVER SYSTEM

- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.

Microsoft SQL Server 2008 R2 ofrece muchas ventajas y beneficios comparado con IBM DB2, Oracle y MySQL. Incluye las funcionalidades más importantes de bases de datos sin tener que pagar por funcionalidades o productos añadidos, como son alta disponibilidad, recuperación ante desastres en remoto, particionado, compresión de datos, cifrado transparente de datos, datos espaciales, gestión de datos maestros, procesamiento de eventos complejos, ETL, OLAP, minería de datos, generación de informes y herramientas de Business Intelligence (BI) en modo autoservicio [39].

8.1.2 Comparación de MySql y SQL Sever

Ventajas de SQL Server 2008 R2 sobre MySql [39]:

- ✚ Planes de evolución tecnológica estables.
- ✚ Las mejores herramientas de desarrollo del mercado.
- ✚ Escalabilidad alcanzable en la nube.
- ✚ Integración con WordPress, Drupal y PHP.
- ✚ Herramientas para una migración más sencilla.
- ✚ Funcionalidades gratuitas con mejor rendimiento.
- ✚ Herramientas de informes y Business Intelligence en Microsoft Office.
- ✚ Funcionalidades de nivel empresarial disponibles si es preciso.

	MySQL	SQL Server Express	SQL Server 2008 R2
Costo	Gratis	Gratis	De Pago
Límites de DB, CPU y RAM	OS max	Varios Limites	OS max
Soporte para S.O.	Varios	Windows	Windows
Operaciones de BD básicas	✓	✓	✓
Conformidad con ACID	Depende	✓	✓
Vulnerabilidades de seguridad (2004 - 2009)	73 vulnerabilidades	17 vulnerabilidades	17 vulnerabilidades
Backup online		✓	✓
Informes		✓	✓
Versión en nube		✓	✓
Autoajuste del rendimiento		✓	✓

FLUVER SYSTEM

Sincronización		✓	✓
Análisis, OLAP (On-Line Analytical Processing) Procesamiento analítico en línea			✓
Alta disponibilidad			✓

Tabla 24. Comparación SQL Server 2008 R2 vs MySQL

8.1.3 Comparación de SQL Server con Oracle

Ventajas de SQL Server 2008 R2 sobre Oracle 11g [39]:

- ✚ Escalabilidad y rendimiento.
- ✚ Menos vulnerabilidades.
- ✚ Herramientas BI (Business Intelligence)

	SQL Server 2008 R2	Oracle
Rendimiento, escalabilidad y Alta Disponibilidad	✓	✓
Más seguro(menos vulnerabilidades)	✓	
Productividad para desarrolladores	✓	
Business Intelligence	✓	
Integración con Microsoft Office	✓	
Costo Total de Propiedad	✓	

Tabla 25. Comparación SQL Server 2008 R2 vs Oracle

Oracle solo ofrece funcionalidades parciales en estas áreas y requiere inversiones adicionales en licencias.

8.1.4 Comparación de SQL Server con IBM DB2

Ventajas de SQL Server 2008 R2 sobre IBM DB2 9.7 [39]:

- ✚ Rendimiento, escalabilidad y alta disponibilidad.
- ✚ Herramientas BI (Business Intelligence)

	SQL Server 2008 R2	DB2
Rendimiento, escalabilidad y Alta Disponibilidad	✓	✓
Más seguro(menos vulnerabilidades)	✓	
Productividad para	✓	

FLUVER SYSTEM

desarrolladores		
Manejabilidad	✓	
Costo Total de Propiedad	✓	

Tabla 26. Comparación SQL Server 2008 R2 vs DB2

8.1.5 Benchmarks de SQL Server 2008 R2

Rendimiento y Escalabilidad de aplicaciones de proveedores de software independientes.

1. Se comprobó mediante la aplicación de *Camstar* que se pueden tener hasta 205 transacciones por segundo y 60% de reducción de espacio en una compresión de base de datos.
2. Se tiene el registro de 50 000 usuarios concurrentes con una tasa de respuesta inferior al segundo.
3. Microsoft y Unisys anunciaron un record en la carga de datos dentro de la base de datos relacional, usando una herramienta ETL (Extract, Transform and Load). Cerca de 1TB de los datos de TPC-H fueron cargados en menos de 30 minutos. De esta manera se tienen 2.36TB por hora [40].

8.1.5 Elección del Sistema Gestor de Base de Datos



El gestor de Base de Datos que se utilizará para almacenar los datos es Microsoft SQL Server 2008 R2, una de las principales razones es que dicho gestor es utilizado en el laboratorio **C4**⁸, donde a mediano plazo, probablemente se pueda implementar este sistema. No es posible diseñar la base del sistema sin antes tomar en cuenta sus posibilidades de implementación.

Debido a sus ventajas en seguridad, respaldo online, escalabilidad en la nube, manejo de una cantidad grande de registros y tasa de respuesta a usuarios concurrentes se ha tomado la decisión de utilizar este gestor.

⁸ Centro de Comando, Control, Comunicaciones y Cómputo del Proyecto de Video vigilancia del Distrito Federal.

8.2 Diseño

Una base de datos correctamente diseñada permite obtener acceso a información exacta y actualizada. Un diseño correcto es esencial para lograr que la base de datos se adapte a los objetivos fijados y pueda modificarse fácilmente.

8.2.1 Requerimientos de hardware y software para instalar SQL Server 2008 R2

Componente	Requisito
Marco de trabajo	El programa de instalación de SQL Server instala los siguientes componentes de software requeridos por el producto: <ul style="list-style-type: none"> • .NET Framework 3.5 SP1 • SQL Server Native Client • Archivos auxiliares para la instalación de SQL Server
Software	El programa de instalación de SQL Server requiere Microsoft Windows Installer 4.5 o una versión posterior.
Software de red	Los sistemas operativos compatibles tienen el software de red integrado. Las instancias predeterminadas y con nombre independientes admiten los siguientes protocolos de red: <ul style="list-style-type: none"> • Memoria compartida • Canalizaciones con nombre • TCP/IP • VIA
Virtualización	Dentro del rol Hyper-V de Windows Server 2008 SP2, se puede asignar un máximo de cuatro procesadores virtuales a máquinas virtuales que ejecuten las ediciones de 32 ó 64 bits de Windows Server 2008. Se pueden asignar como máximo dos procesadores virtuales a equipos virtuales que ejecuten ediciones de 32 bits de Windows Server 2003. Para equipos virtuales que hospedan otros sistemas operativos, se puede asignar como máximo un procesador virtual a equipos virtuales.
Software de Internet	Se requiere Internet Explorer 6 Service Pack 1 o una versión posterior para Microsoft Management Console (MMC), SQL Server Management Studio, Business Intelligence Development Studio, el componente Diseñador de informes de Reporting Services y la Ayuda HTML.
Pantalla	Las herramientas gráficas de SQL Server 2008 R2 requieren Super VGA o una resolución mayor: resolución mínima de 800 x 600 píxeles.
Procesador	Tipo de procesador: <ul style="list-style-type: none"> • Mínimo: AMD Opteron, AMD Athlon 64, Intel Xeon compatible con Intel EM64T, Intel Pentium IV compatible con EM64T Velocidad del procesador:

FLUVER SYSTEM

	<ul style="list-style-type: none"> • Mínimo: 1,4 GHz • Recomendado: 2,0 GHz o más 												
Sistema operativo	Windows Server 2003 o posterior y Windows XP o posterior.												
Memoria	RAM: <ul style="list-style-type: none"> • Mínimo: 1 GB • Recomendado: 4 GB o más • Máximo: máximo del sistema operativo 												
Disco Duro	<p>Disponibilidad de al menos 3,6 GB de espacio en disco en la unidad del sistema para estos archivos:</p> <table border="1"> <tr> <td>Motor de base de datos y archivos de datos, Replicación y Búsqueda de texto completo</td> <td>711 MB</td> </tr> <tr> <td>Analysis Services y archivos de datos</td> <td>345 MB</td> </tr> <tr> <td>Reporting Services y Administrador de informes</td> <td>304 MB</td> </tr> <tr> <td>Integration Services</td> <td>591 MB</td> </tr> <tr> <td>Componentes de cliente (excepto Libros en pantalla y herramientas de Integration Services)</td> <td>1823 MB</td> </tr> <tr> <td>Libros en pantalla de SQL Server [41]</td> <td>157 MB</td> </tr> </table>	Motor de base de datos y archivos de datos, Replicación y Búsqueda de texto completo	711 MB	Analysis Services y archivos de datos	345 MB	Reporting Services y Administrador de informes	304 MB	Integration Services	591 MB	Componentes de cliente (excepto Libros en pantalla y herramientas de Integration Services)	1823 MB	Libros en pantalla de SQL Server [41]	157 MB
Motor de base de datos y archivos de datos, Replicación y Búsqueda de texto completo	711 MB												
Analysis Services y archivos de datos	345 MB												
Reporting Services y Administrador de informes	304 MB												
Integration Services	591 MB												
Componentes de cliente (excepto Libros en pantalla y herramientas de Integration Services)	1823 MB												
Libros en pantalla de SQL Server [41]	157 MB												

Tabla 27. Requerimientos de hardware y software para instalar SQL Server 2008 R2

8.2.2 Datos a almacenar y Modelo Relacional

Los requisitos del módulo de base de datos exigen crear una relación entre la ubicación de las cámaras y las estadísticas provenientes de cada una de ellas. Por lo tanto los elementos identificables como entidades son la “Cámara” cada dispositivo de vigilancia y las “Estadísticas” obtenidas.

Se determina la razón de cardinalidad a 1:N donde para cada dispositivo (cámara) con su respectiva ubicación (dirección) se tendrá un conjunto de estadísticas.

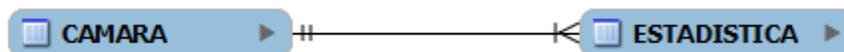


Ilustración 28. Correspondencia de Cardinalidad entre Entidades 1:N

Es necesario evitar la información duplicada y aprovechar los aspectos en común de las entidades para tener un diseño conciso y garantizar la integridad de la información.

FLUVER SYSTEM

Para distinguir unívocamente a los registros pertenecientes a un conjunto de entidades dado, se determina el atributo clave “idCamara” para la tabla “CAMARA” con tipo varchar(), y el atributo clave “fecha” para la tabla “ESTADISTICA” con tipo DateTime y una clave foránea “idCamara” que determine la relación de pertenencia hacia la tabla “CAMARA”.

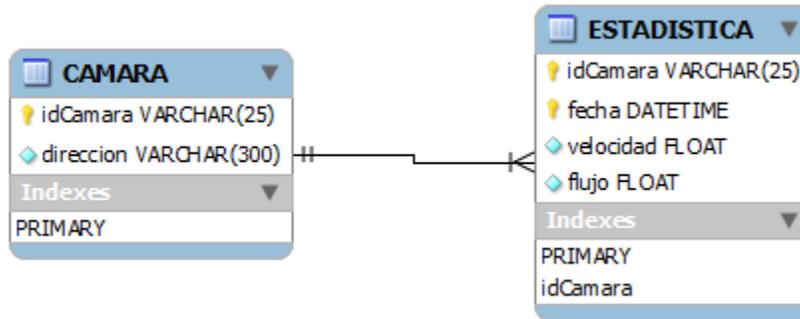


Ilustración 29. Modelo Relacional

Los datos necesarios para estadísticas son la “velocidad” y el “flujo” vehicular, los cuales se definen como tipo float.

Para la tabla Cámara se necesita la ubicación de la cámara “direccion” y el identificador de la cámara “idCamara” el cual en este caso se trata de la dirección MAC⁹ ya que es un identificador de 48 bits, representados mediante dígitos hexadecimales que se agrupan en seis parejas (Ej. FF:FF:FF:FF:FF:FF), que corresponde de forma única a cada dispositivo. Queda abierta la posibilidad de que se utilice la IP de instalación para acceder a la cámara si el usuario así lo requiere (234.23.223.123), realizando los correspondientes cambios de formato únicamente sobre la interfaz de usuario donde se controla el correcto formato del dato. Los campos anteriores se declaran como tipo varchar ().

8.2.3 Diccionario de Datos

Entidad CAMARA

Objetivo: Almacenar una cámara con su correspondiente dirección			
NOMBRE	TIPO DE DATO	DESCRIPCIÓN	EJEMPLO
idCamara	VARCHAR(25) NOT NULL PRIMARY KEY	Identificador de la cámara, ya sea por su IP o su dirección MAC.	00:13:49:00:01:02
direccion	VARCHAR(300) NOT NULL	Descripción de la dirección a la que se hace referencia.	Av. Insurgentes sur

Tabla 28. Diccionario de Datos Entidad Cámara

⁹ Media Access Control, Control de Acceso al Medio físico.

FLUVER SYSTEM

Entidad ESTADISTICA

Objetivo: Almacenar las velocidades y flujos registrados en cierta dirección en un tiempo determinado.

NOMBRE	TIPO DE DATO	DESCRIPCIÓN	EJEMPLO
idCamara	VARCHAR (25) Primary Key Foreign Key	Identificador de la cámara.	00:13:49:00:01:02
fecha	DATETIME Primary Key	Es el campo que tiene el día, mes, año y hora. Se toma como una llave compuesta, ya que tendrá un registro único en una hora dada.	12-02-2012 12:34:12
velocidad	FLOAT NOT NULL	Identifica la velocidad promedio del lugar donde se encuentra la cámara.	60 km/h
flujo	FLOAT NOT NULL	Identifica el flujo aproximado de vehículos en tal dirección	30

Tabla 29. Diccionario de Datos Entidad Estadística

8.2.4 Diagrama de Clases

El siguiente diagrama muestra las clases utilizadas en el módulo para la conexión de la base de datos.

- Clase Database: Es la clase utilizada para realizar la conexión a la base de datos.
- Clase uiDatabase: Clase de control, la cual contiene las operaciones para la administración de la base de datos.
- Clase ui_Database: Clase de interfaz, la cual muestra los controles al usuario y manda la información a la clase control.
- Clase conexionModel1: Clase de frontera, la cual interacciona con las clases del prototipo dos para recibir la información a almacenar.

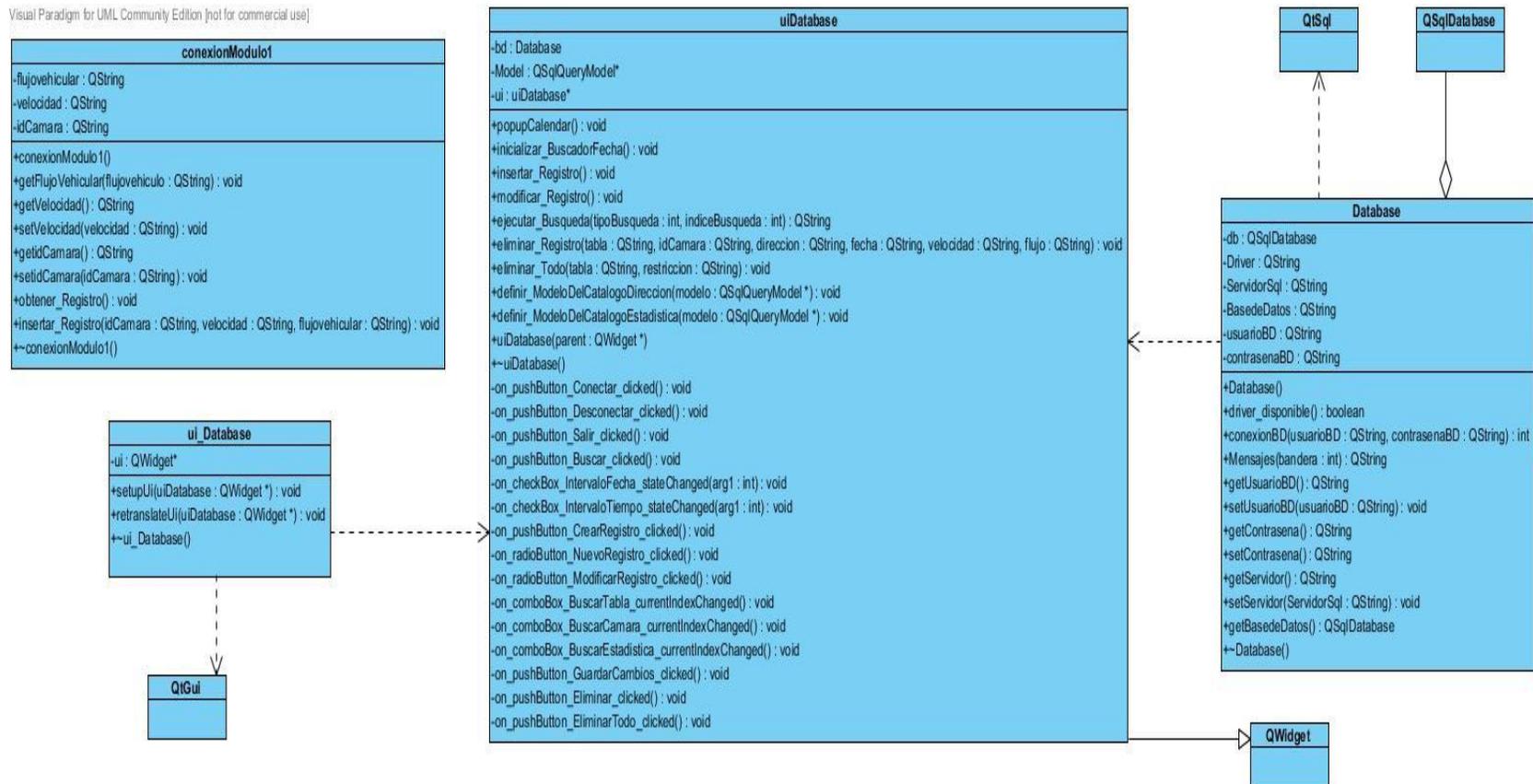


Ilustración 30 Diagrama de Clases de 3er Prototipo

8.2.4 Diagramas de Secuencia

Los diagramas muestran el flujo de datos entre clases, tanto para la conexión como para la administración de los datos en la base de datos.

NOTA: Para la construcción del módulo, a excepción del caso *conexión a base de datos*, se utilizó la clase Actor, la cual será sustituida por la clase

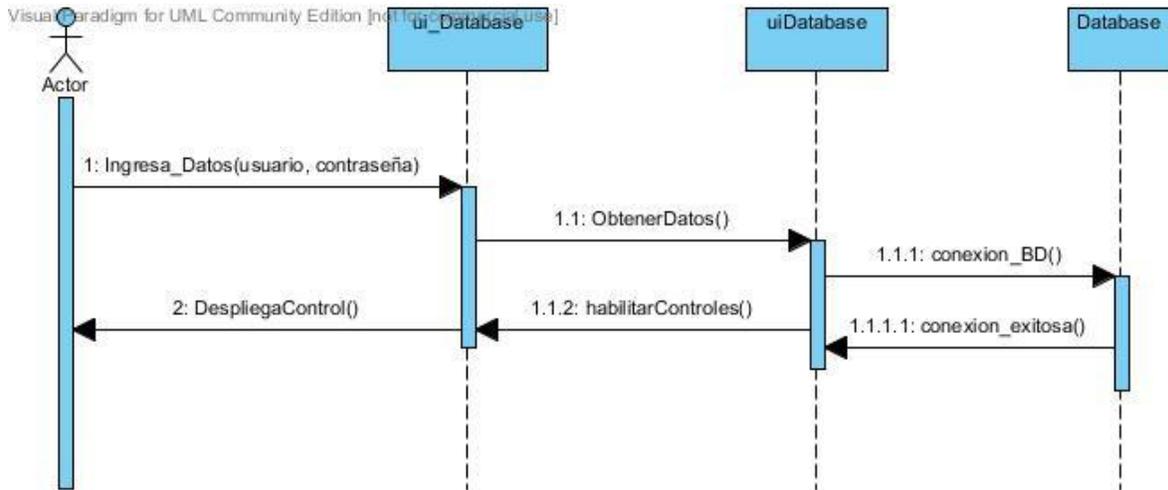


Ilustración 31 Diagrama de secuencia Conexión a la base de datos

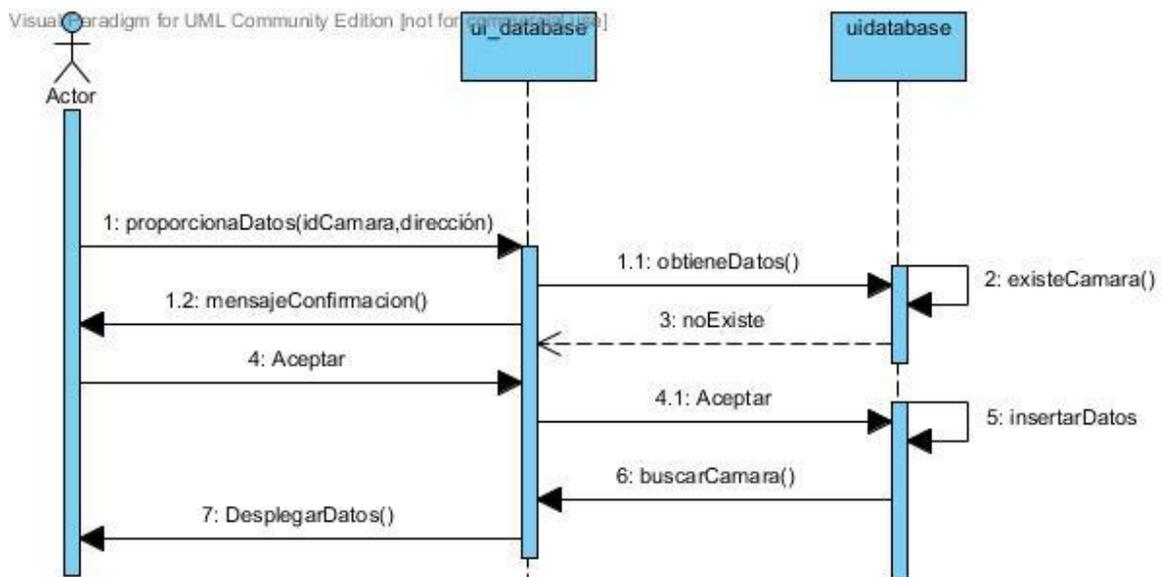


Ilustración 32 Diagrama de secuencia Registrar Cámara

FLUVER SYSTEM

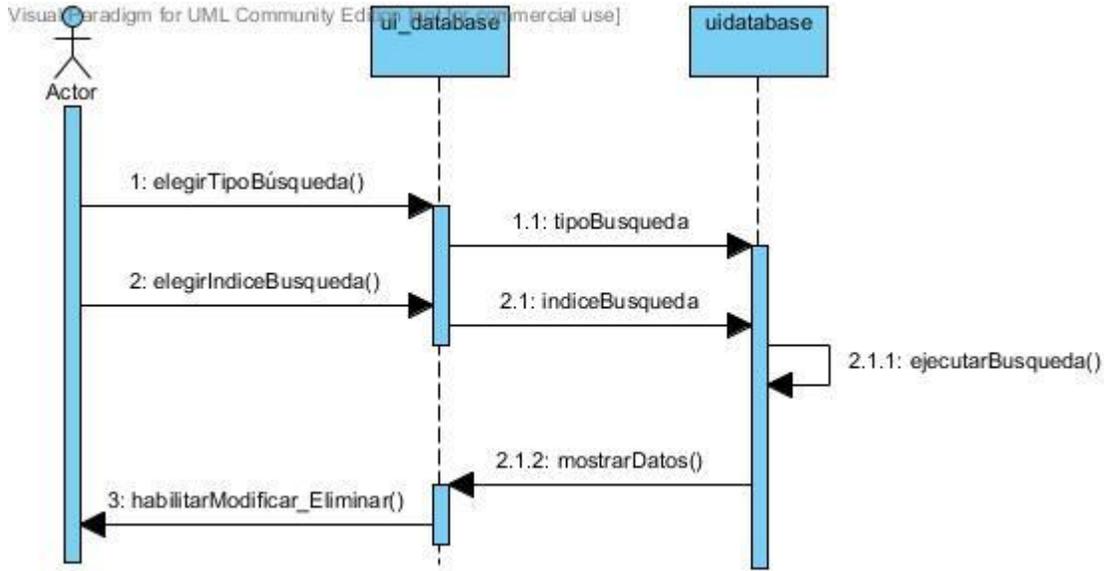


Ilustración 34 Diagrama de secuencia Buscar Registro

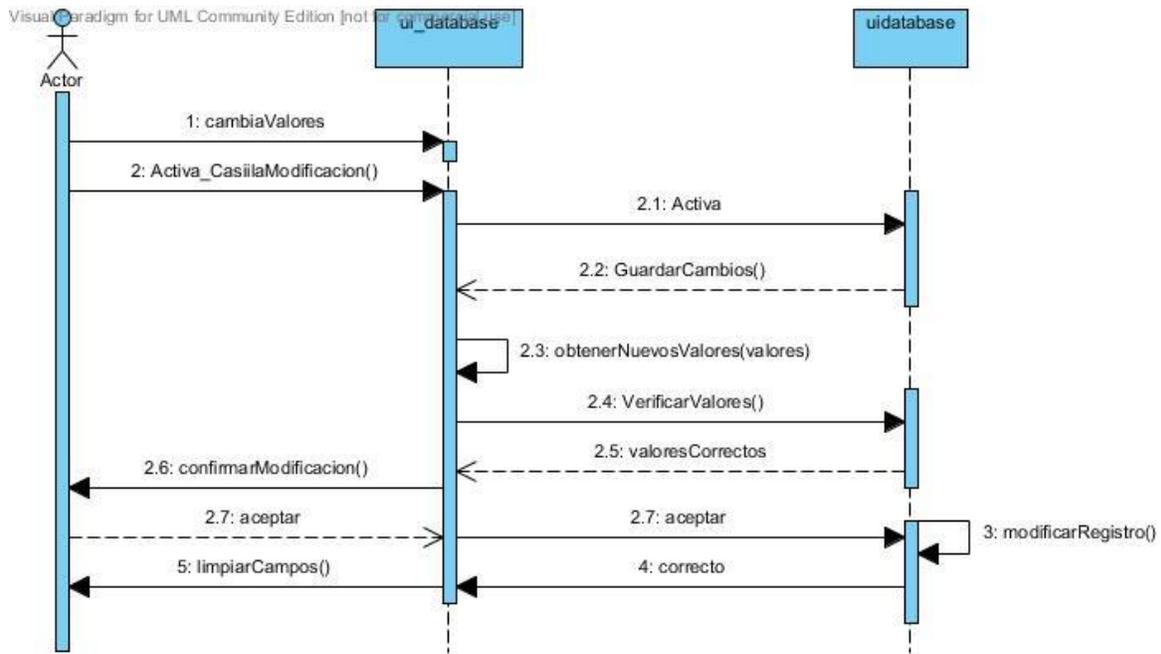


Ilustración 33 Diagrama de secuencia Modificar datos

FLUVER SYSTEM

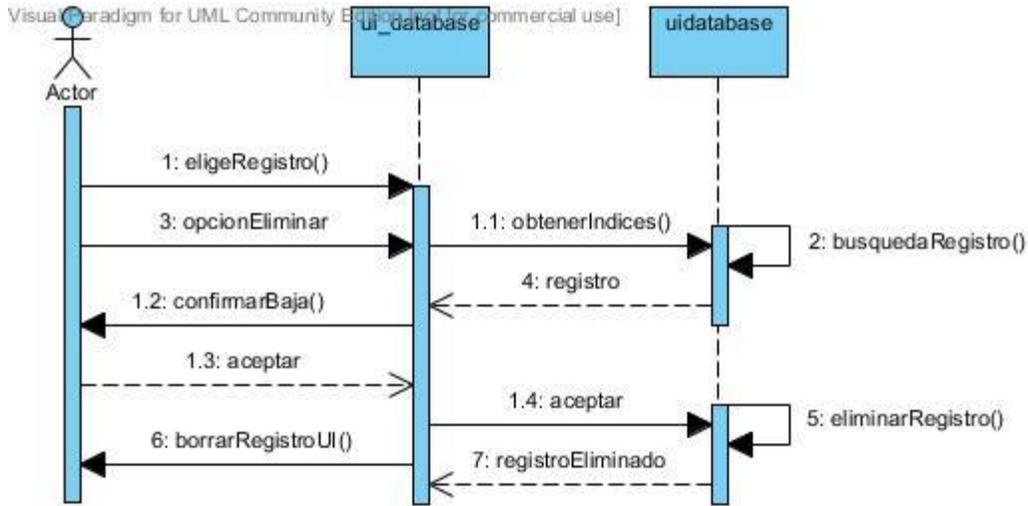


Ilustración 35 Diagrama de Secuencia Eliminar registro

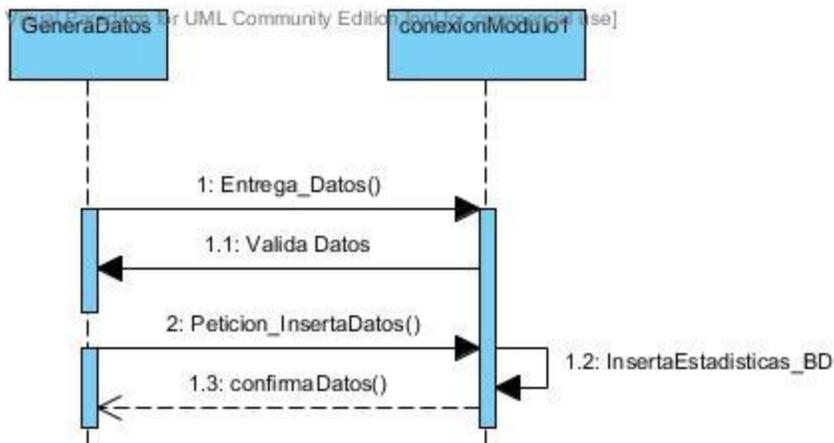


Ilustración 36 Diagrama de Secuencia Obtener estadísticas

8.3 Desarrollo

En esta sección se presenta la creación de la Base de Datos del sistema y conexión y comunicación de la interfaz que permite al usuario la administración de la información.

8.3.1 Creación de la Base de Datos

Para comenzar con la implementación del Módulo de Base de Datos, se definen los Scripts de SQL para crear una Nueva Base de Datos y las tablas que se especificaron en la sección anterior, para que puedan ser ejecutados mediante el Administrador de SQL Server.



Ilustración 37. Scripts para crear una Nueva Base de Datos

Iniciar el Administrador de Acceso a SQL Server 2008 R2:



Ilustración 38. Pantalla de inicio SQL Server 2008 R2

Conectarse con el Motor de Base de Datos por medio del Servidor “TOSHIBA” en este caso, utilizando la autenticación “SQL Server Authentication”, proporcionar nombre de usuario y contraseña.

FLUVER SYSTEM

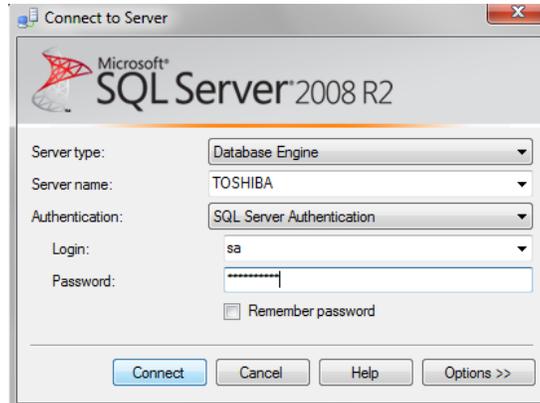


Ilustración 39. Pantalla de conexión SQL Server 2008 R2

En el explorador de objetos aparece una instancia del SQL Server Database Engine, se expande y se observan las Bases de Datos Existentes.

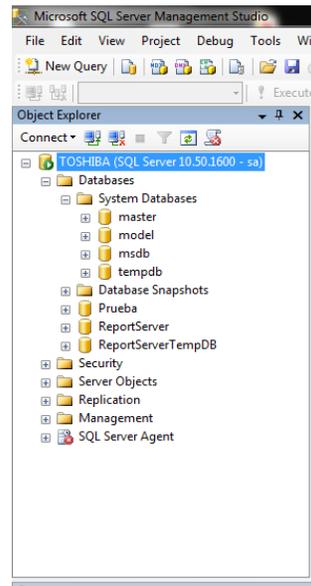


Ilustración 40. Explorador de Objetos SQL Server

Para crear la Base de Datos “FluverSystem” definida en el Script se abre el archivo y se selecciona la opción *Execute*. Se realiza la misma operación para crear las tablas “Camara” y “Estadistica”.

FLUVER SYSTEM

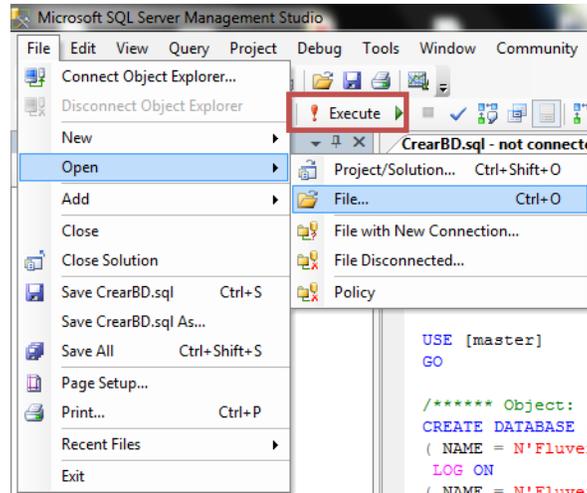


Ilustración 41. Abrir Script SQL Server

Es importante destacar que en el Script de creación de la Base de Datos se definen dos tipos de archivos, ya que como mínimo cada base de datos de SQL Server tiene un archivo de datos (*data file*) y un archivo de registro (*log file*).

Los archivos de datos contienen datos y objetos como tablas, índices, procedimientos almacenados y vistas.

Los archivos de registro contienen la información que se requiere para recuperar todas las transacciones en la base de datos [42].

SQL Server Archivos de datos primarios de la Base de datos

El archivo de datos principal contiene la información de inicio de la base de datos y apunta a los otros archivos en la base de datos. Los datos del usuario y los objetos se pueden almacenar en este archivo o archivos de datos secundarios. Cada base de datos tiene un archivo de datos principal. La extensión de archivo recomendada para los archivos de datos principales es .mdf.

SQL Server Archivos de datos Secundarios de la Base de datos

Los archivos de datos secundarios son opcionales, son definidos por el usuario y almacenan datos del usuario. Pueden ser utilizados para distribuir los datos en varios discos, poniendo cada archivo en una unidad de disco diferente. Además, si una base de datos supera el tamaño máximo para un solo archivo de Windows, puede utilizar archivos de datos secundarios para que la base de datos pueda seguir creciendo.

La extensión de archivo recomendada para los archivos de datos secundarios es .ndf.

SQL Server Archivo de registro de transacciones (Log File) de la Base de datos

Los archivos de registro de transacciones contienen la información de registro que se utiliza para recuperar la base de datos. Debe haber al menos un archivo de registro para cada base de datos. La extensión de archivo recomendada para los registros de transacciones es .ldf.

8.3.2 Comunicación Qt Creator y SQL Server 2008 R2*SQL Controladores de Bases de Datos*

El módulo SQL de Qt utiliza plugins de controladores para comunicarse con las diferentes API¹⁰s de las Bases de Datos. Dado que el API del módulo SQL de Qt es independiente de la base de datos, todo el código específico de la base de datos se encuentra contenido en los controladores. Algunos controladores se incluyen junto con la instalación de Qt y otros tienen que ser agregados. El código fuente de los controladores, se puede utilizar como modelo para escribir controladores propios [43].

Bases de Datos compatibles

La siguiente tabla muestra los controladores incluidos con Qt:

Controlador	DBMS ¹¹
QDB2	IBM DB2 (version 7.1 and above)
QIBASE	Borland InterBase
QMYSQL	MySQL
QOCI	Oracle Call Interface Driver
QODBC	Open Database Connectivity (ODBC) - Microsoft SQL Server and other ODBC-compliant databases
QPSQL	PostgreSQL (versions 7.3 and above)
QSQLITE2	SQLite version 2
QSQLITE	SQLite version 3
QTDS	Sybase Adaptive Server Note: obsolete from Qt 4.7

Tabla 30. Controladores incluidos en Qt

El controlador que se utiliza para comunicar Microsoft SQL Server con Qt Creator es QODBC [43].

QODBC Conectividad Abierta con la Base de Datos (ODBC¹²)

¹⁰ Application Programming Interface, Interfaz de Programación de Aplicaciones.

¹¹ Database Management System, Sistema de Gestión de Bases de Datos

¹² Open Database Connectivity

ODBC es una interfaz general que permite conectarse a varios DBMS utilizando una interfaz común. El controlador QODBC permite conectarse a un gestor de controladores ODBC y acceder a las fuentes de datos disponibles. Es necesario instalar y configurar los controladores ODBC para el gestor de controladores ODBC que se instala en el sistema. Entonces el plug-in QODBC permite utilizar estas fuentes de datos en las aplicaciones desarrolladas en Qt [43].

En Windows un gestor de controladores ODBC debe ser instalado por defecto. Debe tomarse en consideración que el cliente que utilice la aplicación debe tener un gestor de controladores ODBC instalado, de lo contrario el plugin QODBC no funcionará.

Cuando se realiza la conexión a un origen de datos ODBC se debe pasar el nombre del origen de datos ODBC a la función de `setDatabaseName QSqlDatabase :: ()` en lugar del nombre de base de datos real.

El plug-in QODBC por lo tanto, comprueba si el origen de datos puede ser utilizado después de que una conexión se ha establecido y se niega a trabajar si no está disponible.

8.3.3 Creación de un Origen de Datos

ODBC es una tecnología que le permite mover datos de un tipo de base de datos (un origen de datos) a otro. Para ello, necesita el controlador adecuado.

Para agregar un origen de datos al equipo que permita acceder a la información de la Base de Datos creada en SQL Server:

1. Abrir Administrador de origen de datos ODBC, clic en el botón Inicio , en Panel de control, y a continuación, en Orígenes de datos (ODBC).

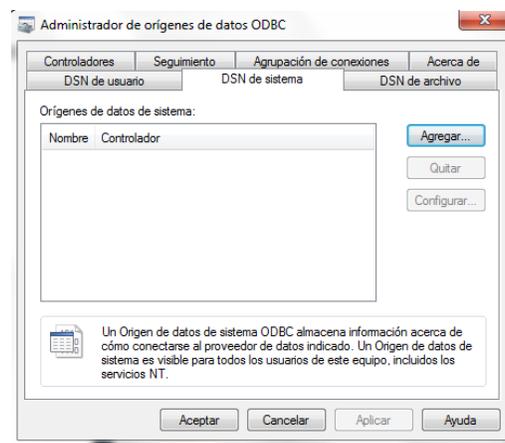


Ilustración 42. Administrador de orígenes de datos ODBC

FLUVER SYSTEM

2. Dar clic en Agregar, elegir el controlador adecuado para el origen de datos que va a agregar en este caso SQL Server Native Client 10.0 (es el controlador correspondiente a la versión SQL Server 2008 R2) y, después, dar clic en Finalizar.

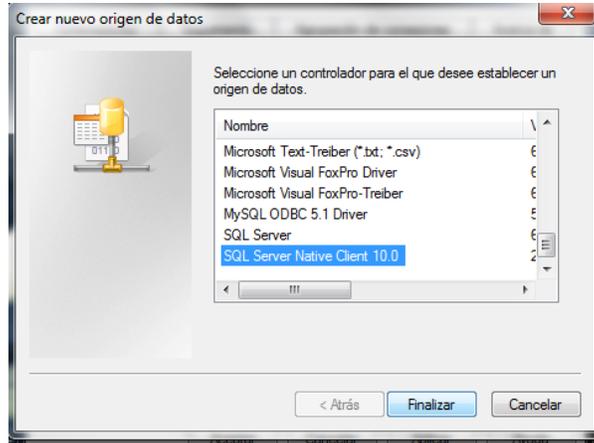


Ilustración 43. Controlador para establecer Origen de Datos

3. En el cuadro Nombre del origen de datos, se escribe un nombre para el origen de datos (en este caso, FluverSystem). Es recomendable escribir una descripción para recordar las razones por las que se utiliza el origen de datos.

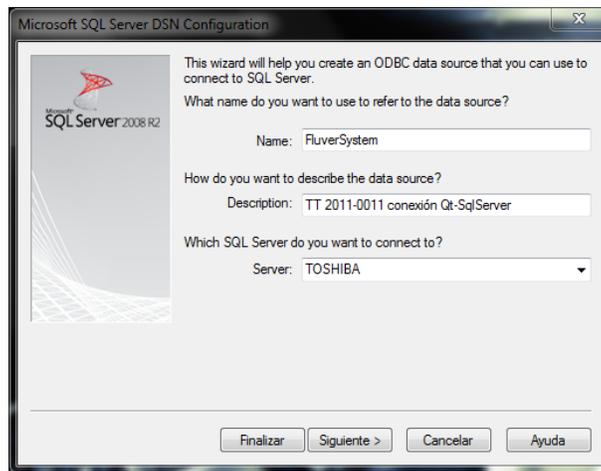


Ilustración 44. Configuración del Origen de Datos parte 1

4. Se elige la forma de autenticación que se desea utilizar para establecer la conexión con SQL Server, en este caso se utiliza un Login y un password.

FLUVER SYSTEM

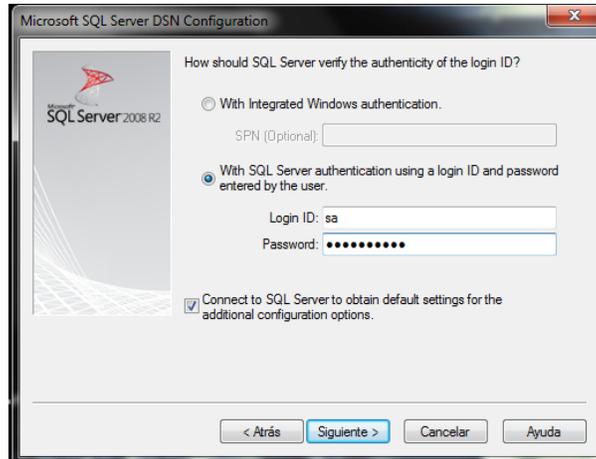


Ilustración 45. Configuración del Origen de Datos parte 2

5. Para finalizar se muestran los ajustes de configuración adicionales:

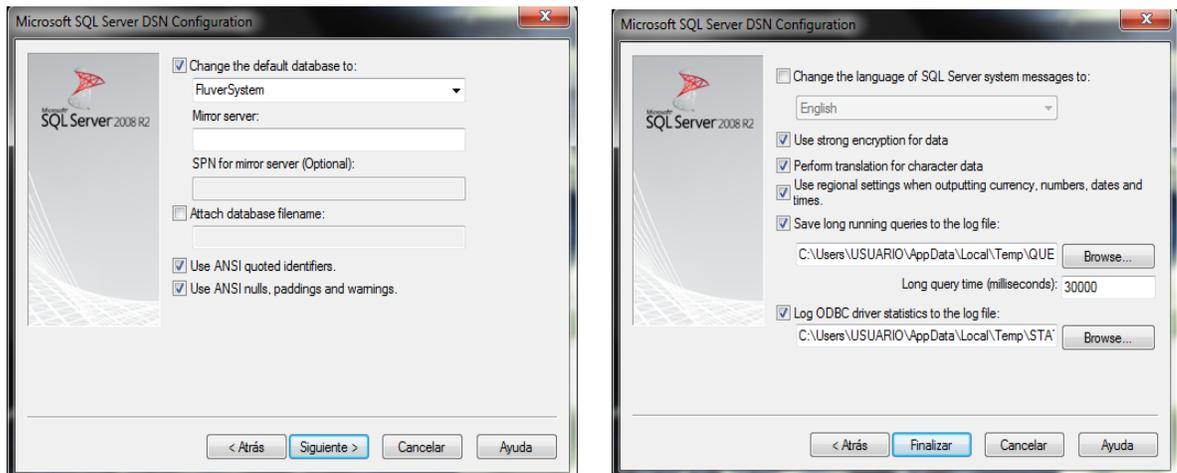


Ilustración 46. Ajustes de Configuración Adicionales

8.3.4 Desarrollo e implementación de la Interfaz

Se ha creado la Base de Datos para administrar las estadísticas de tránsito, y el origen de datos correspondiente que permite la comunicación de SQL Server con Fluver System; por lo tanto a continuación se detalla la estructura de la interfaz para la administración de la Base de Datos.

El módulo de administración de la base de datos se divide en tres partes principales:

- La conexión al Gestor de Base de Datos

En esta parte el usuario se identifica para obtener acceso a la Base de Datos FluverSystem, los datos deben coincidir con los especificados en el origen de datos.

FLUVER SYSTEM



Ilustración 47. Conexión a Base de Datos

- Operaciones para insertar y actualizar registros de la tabla “Cámara” en la Base de Datos FluverSystem.

El usuario tiene la posibilidad de dar de alta nuevos dispositivos de vigilancia (video cámaras) o modificar registros existentes proporcionando su Dirección, es decir donde se encuentran colocados, y un identificador que lo distinga, en este caso su dirección MAC9.



Ilustración 48. Operaciones para insertar y actualizar registros de la tabla Cámara

- Consultar las tablas Cámara y Estadística de la Base de Datos FluverSystem

El usuario tiene la posibilidad de realizar consultas por medio de palabras clave de la dirección, el identificador asignado a una video cámara, por fecha, o intervalo de fechas, se puede especificar una determinada hora o un intervalo de tiempo, o simplemente mostrar todo el contenido de alguna de las dos tablas. Después del obtener el resultado de alguna consulta es posible eliminar uno o varios registros del catálogo.

FLUVER SYSTEM

Consultar Base de Datos

Buscar en: Buscar por: BUSCAR

Intervalo de Búsqueda Fecha Especificar Intervalo de Tiempo

Fecha Inicial: Fecha Final: Tiempo Inicial: Tiempo Final:

Ilustración 49. Consultar Base de Datos

8.4 Implementación de Pruebas

Iteración	Fecha de ejecución	Analistas de la prueba	Estado de la prueba	Comentarios
1	03 / 02/ 2012	Dorantes Reyes Cristina	Concluido	Se implementó un aviso que se muestra si se realizó una conexión exitosa.
Nombre de la prueba		CPBD01: Prueba de conexión entre Qt y SQL Server 2008 R2		
Objetivo		Mostrar que es posible realizar una conexión desde el sistema al Gestor de Bases de Datos SQL Server 2008 por medio de un origen de datos ODBC.		
Requisitos		<ul style="list-style-type: none"> Utilizar el Sistema Operativo Windows XP o posterior. Contar con el Gestor de Bases de Datos SQL Server 2008 R2. Ejecutar los Scripts CrearBD, Crear_tablaCamara y Crear_tablaEstadistica. Contar con el Controlador SQL Server Native Client (incluido en la instalación de SQL Server 2008 R2). Agregar un Origen de Datos ODBC en el equipo para SQL Server 2008 R2 con el controlador SQL Server Native Client y con la información de conexión necesaria para tener acceso a la Base de Datos <i>FluverSystem</i>, es importante especificar un usuario y contraseña. Programa ejecutable "CPBD01.exe" compilado estáticamente, es decir que contenga las .dll para poder ejecutarlo sin incluir los dll: C:\Qt\2010.02.1\mingw\bin libgcc_s_dw2-1.dll mingwm10.dll C:\Qt\2010.02.1\qt\bin QtCored4.dll QtGuid4.dll QtSql4.dll Instalar el framework Qt 4.7 en la máquina en la que se ejecutarán las pruebas si es que no se cuenta con las dll anteriores. Se utilizará una memoria USB en la cual se almacenará la carpeta "CPBD01", la cual contendrá el ejecutable del programa con el que se realizarán las pruebas. 		
Valores a ingresar		Usuario y Contraseña de acceso a SQL Server 2008 R2.		
Resultados esperados		El programa permite la conexión a la base de datos FluverSystem mediante la autenticación de un usuario y		

FLUVER SYSTEM

ID	Acciones	Resultados esperados	Correcto	Incorrecto	Comentarios
1	Conectar la memoria a la computadora.	La computadora reconoce al dispositivo y ofrece la opción de ver el contenido de esta.	X		
2	Elegir la opción de ver el contenido.	Se muestra una ventana con el contenido de la memoria y aparece la carpeta "CPBD01".	X		
3	Acceder a la carpeta con nombre "CPBD01".	Se muestra una ventana con el contenido de la carpeta.	X		
4	Ejecutar el archivo "CPBD01\CPBD01.exe"	Se muestra la ventana de Administrador de Base de datos del Sistema.	X		
5	En el apartado de Conexión a Base de Datos se introduce un Usuario y Contraseña y se finaliza con un clic en el botón Conectar.	Aparece un mensaje de Conexión Exitosa y se activan los apartados Operaciones y Consultar Base de datos, los cuales en un principio estaban desactivados.	X		La primera vez que se realiza la conexión tarda algunos segundos extra.
6	Ir al apartado Consultar Base de Datos, Elegir la tabla Cámara y/o tabla Estadística y Mostrar todo el contenido.	Se despliega el contenido de la tabla Cámara y/o tabla Estadística o se notifica si no existen registros.	X		

Tabla 31. Prueba de conexión entre Qt y SQL Server 2008 R2

FLUVER SYSTEM

Iteración	Fecha de ejecución	Analistas de la prueba	Estado de la prueba	Comentarios		
1	06 / 02/ 2012	Dorantes Reyes Cristina	Concluido	Se insertan nuevos registros (dar de alta dispositivo de vigilancia) en la base de datos FluverSystem.		
Nombre de la prueba		CPBD02: Prueba de inserción de registros a la Base de Datos.				
Objetivo		Mostrar que el sistema es capaz de insertar registros en la Base de Datos.				
Requisitos		*Los mencionados en la prueba CPBD01.				
Valores a ingresar		Usuario y Contraseña de acceso a SQL Server 2008 R2, identificador de la Cámara (dispositivo de vigilancia) y Dirección del dispositivo.				
Resultados esperados		El programa permite visualizar en el catálogo de la Base de Datos el nuevo registro.				
ID	Acciones	Resultados esperados	Correcto	Incorrecto	Comentarios	
1	Conectar la memoria a la computadora.	La computadora reconoce al dispositivo y ofrece la opción de ver el contenido de esta.	X			
2	Elegir la opción de ver el contenido.	Se muestra una ventana con el contenido de la memoria y aparece la carpeta "CPBD02".	X			
3	Acceder a la carpeta con nombre "CPBD02".	Se muestra una ventana con el contenido de la carpeta.	X			
4	Ejecutar el archivo "CPBD02\CPBD02.exe"	Se muestra la ventana de Administrador de Base de datos del Sistema.				
5	En el apartado de Conexión a Base de Datos se introduce un Usuario y Contraseña y se finaliza con un clic en el botón Conectar.	Aparece un mensaje de Conexión Exitosa y se activan los apartados Operaciones y Consultar Base de datos, los cuales en un principio estaban desactivados.	X			
6	Ir al apartado de Operaciones y Seleccionar la opción Nuevo Registro, introducir una Dirección y un Identificador para la Cámara correspondiente, y terminar con un clic en el botón Crear Nuevo.	Se muestra un mensaje de confirmación con los datos del nuevo Registro. Si se acepta la confirmación se muestra el catálogo de la Base de Datos donde se visualiza el nuevo registro.	X			

Tabla 32. Prueba de inserción de registros a la Base de Datos

Cálculo del flujo vehicular mediante segmentación de imágenes.

FLUVER SYSTEM

Iteración	Fecha de ejecución	Analistas de la prueba	Estado de la prueba	Comentarios	
1	06 / 02/ 2012	Dorantes Reyes Cristina	Concluido	Se actualizan registros de la tabla Cámara en la base de datos FluverSystem.	
Nombre de la prueba		CPBD03: Prueba de actualización de registros en la Base de Datos.			
Objetivo		Mostrar que el sistema es capaz de actualizar registros en la Base de Datos.			
Requisitos		*Los mencionados en la prueba CPBD01.			
Valores a ingresar		Usuario y Contraseña de acceso a SQL Server 2008 R2, identificador de la Cámara nuevo (dispositivo de vigilancia) y/o Dirección del dispositivo.			
Resultados esperados		El programa permite visualizar en el catálogo de la Base de Datos el registro modificado.			
ID	Acciones	Resultados esperados	Correcto	Incorrecto	Comentarios
1	Conectar la memoria a la computadora.	La computadora reconoce al dispositivo y ofrece la opción de ver el contenido de esta.	X		
2	Elegir la opción de ver el contenido.	Se muestra una ventana con el contenido de la memoria y aparece la carpeta "CPBD03".	X		
3	Acceder a la carpeta con nombre "CPBD03".	Se muestra una ventana con el contenido de la carpeta.	X		
4	Ejecutar el archivo "CPBD03\CPBDO3.exe"	Se muestra la ventana de Administrador de Base de datos del Sistema.			
5	En el apartado de Conexión a Base de Datos se introduce un Usuario y Contraseña y se finaliza con un clic en el botón Conectar.	Aparece un mensaje de Conexión Exitosa y se activan los apartados Operaciones y Consultar Base de datos, los cuales en un principio estaban desactivados. La opción Modificar registro debe estar deshabilitada mientras no exista una consulta que arroje por lo menos un registro.	X		
6	Ir al apartado de Consultar Base de Datos, elegir una tabla para la búsqueda y consultar. Seleccionar un	La opción Modificar Registro se habilita y coloca en el apartado Dirección y el apartado Identificador de Cámara los	X		

FLUVER SYSTEM

	registro utilizando la numeración que acompaña a cada fila (tupla, registro).	datos del registro seleccionado para que puedan ser actualizados por el usuario.			
7	En el apartado de Operaciones actualizar los datos de Dirección y/o Identificador para la Cámara correspondiente, y terminar con un clic en el botón Guardar Cambios.	Se muestra un mensaje de confirmación con los datos de Actualización del Registro. Si se acepta la confirmación se muestra el catálogo de la Base de Datos donde se visualiza el registro actualizado.	X		

Tabla 33. Prueba de actualización de registros en la Base de Datos

FLUVER SYSTEM

Iteración	Fecha de ejecución	Analistas de la prueba	Estado de la prueba	Comentarios	
1	08 / 02/ 2012	Dorantes Reyes Cristina	Concluido	Se eliminarán uno o más registros de la base de datos FluverSystem.	
Nombre de la prueba		CPBD04: Prueba para eliminar uno o más registros de la Base de Datos.			
Objetivo		Mostrar que el sistema es capaz de eliminar uno o más registros en la Base de Datos.			
Requisitos		*Los mencionados en la prueba CPBD01.			
Valores a ingresar		Usuario y Contraseña de acceso a SQL Server 2008 R2, Consulta.			
Resultados esperados		El programa permite visualizar el catálogo de la Base de Datos sin los registros eliminados.			
ID	Acciones	Resultados esperados	Correcto	Incorrecto	Comentarios
1	Conectar la memoria a la computadora.	La computadora reconoce al dispositivo y ofrece la opción de ver el contenido de esta.	X		
2	Elegir la opción de ver el contenido.	Se muestra una ventana con el contenido de la memoria y aparece la carpeta "CPBD04".	X		
3	Acceder a la carpeta con nombre "CPBD04".	Se muestra una ventana con el contenido de la carpeta.	X		
4	Ejecutar el archivo "CPBD04\CPBDO4.exe"	Se muestra la ventana de Administrador de Base de datos del Sistema.			
5	En el apartado de Conexión a Base de Datos se introduce un Usuario y Contraseña y se finaliza con un clic en el botón Conectar.	Aparece un mensaje de Conexión Exitosa y se activan los apartados Operaciones y Consultar Base de datos, los cuales en un principio estaban desactivados. Las opciones Eliminar Selección y Eliminar Todo deben estar deshabilitadas mientras no exista una consulta que arroje por lo menos un registro.	X		
6	Ir al apartado de Consultar Base de Datos, elegir una tabla para la búsqueda y consultar. Si se desea eliminar un solo registro:	Se muestra un mensaje de confirmación Si se acepta la confirmación se muestra el catálogo de la Base de Datos actualizado.	X		

FLUVER SYSTEM

Seleccionarlo utilizando la numeración que acompaña a cada fila (tupla, registro) y finalizar con un clic en Eliminar Selección. Si se desean eliminar todos los registros arrojados por la consulta dar clic en Eliminar Todo.				
---	--	--	--	--

Tabla 34. Prueba para eliminar uno o más registros en la Base de Datos

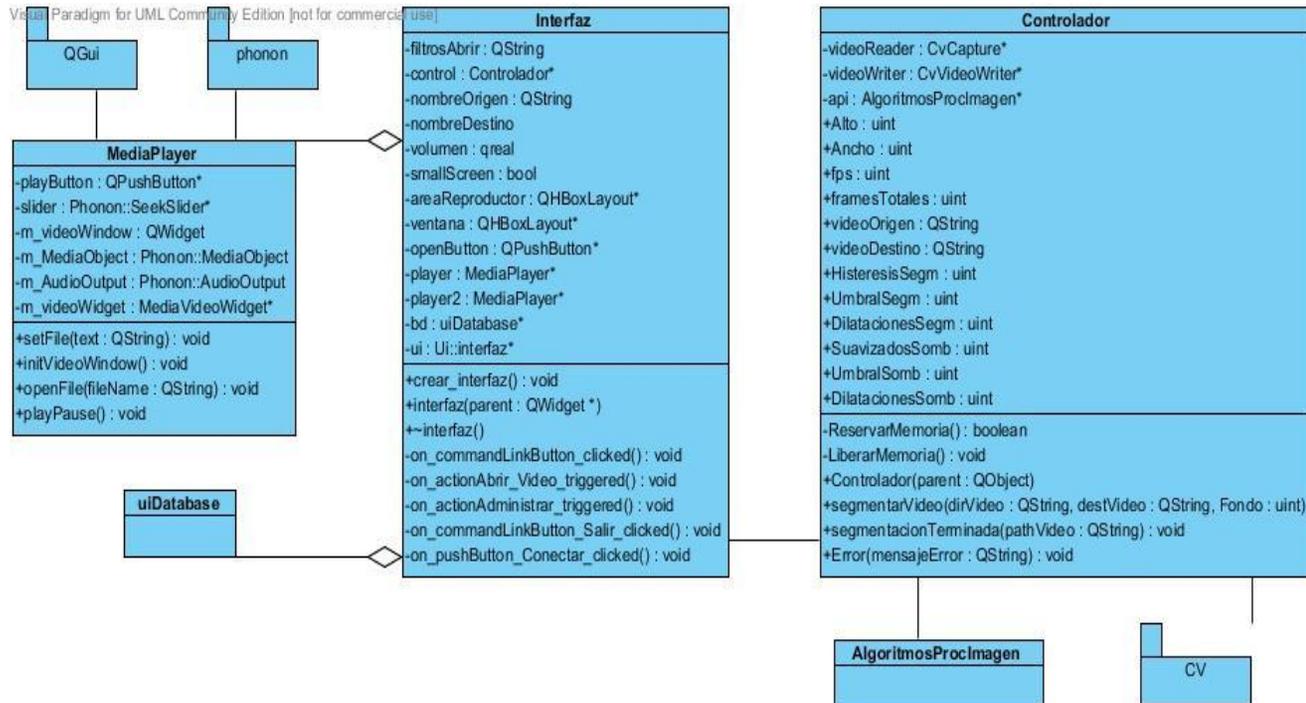
8.5 Integración

Para integrar los tres prototipos anteriores se realizó el diseño de una interfaz, la cual lleva el control de todo el sistema.

8.5.1 Diagrama de Clases

La clase *Interfaz* integra todos los módulos, en el diagrama solo se observa su relación con algunas clases principales que son *Controlador*, *MediaPlayer* y *uiDatabase* ya que estas son las clases principales en cada prototipo. Algunas clases y paquetes no se especifican a detalle puesto que han sido especificados en diagramas anteriores.

- MediaPlayer: Se encarga de la reproducción de los videos, tanto el video original como el segmentado.
- Controlador: Tiene como objetivo obtener el video original, aplicarle los algoritmos que procesan las imágenes para identificación de vehículos y cálculo de velocidad y flujo vehicular.
- uiDatabase: Muestra toda la administración de la base de datos y es la encargada de almacenar tanto el flujo vehicular como la velocidad en cada avenida en un tiempo dado.



Conclusiones

Con la conclusión del trabajo terminal obtenemos un sistema que puede contribuir en gran parte a la disminución de contaminantes, además de solucionar la problemática de no contar con sistemas automatizados de monitoreo de tráfico.

Dicho sistema esta basado en la visión artificial, que es capaz de monitorear el tráfico a través de cámaras colocadas fijamente en avenidas, en las cuales, preferentemente deben contar con un solo sentido y sin intersecciones. De esta manera por medio del tratamiento de imágenes dadas por un video, proporcionar el flujo vehicular y la velocidad promedio de una avenida en particular.

Los algoritmos empleados tanto para la identificación de vehículos como para el cálculo del flujo vehicular y velocidad fueron desarrollados por el equipo de trabajo.

Trabajo Futuro

El trabajo terminal presentado es un sistema funcional y pretende ser base para algunos proyectos a futuro. El sistema cumple con la función de monitoreo automatizado de tráfico por lo que al combinarlo con otros sistemas, se podrían obtener mejoras en cuanto a eficiencia y confiabilidad.

Mejoras:

Las mejoras posibles para ampliar la robustez del sistema son referentes al tratamiento y segmentación del video, para reducir las restricciones en cuanto a cambios de luz durante el día, movimiento de cámara y la dirección de tránsito.

Posibles Proyectos:

El objetivo del sistema es brindar información acerca del flujo vehicular en las avenidas de la ciudad para disminuir la emisión de contaminantes debido al tráfico. Esta información puede ser proporcionada a otros sistemas como:

- **Sistema de Monitoreo de tráfico móvil**

El objetivo de este sistema es brindar la información del tráfico en cualquiera de las avenidas del Distrito Federal a los usuarios por medio de sus dispositivos móviles en tiempo real.

FLUVER SYSTEM

- **Sistema de obtención de rutas alternativas (SORA)**

Este sistema, se encargaría de trazar rutas por la ciudad, tomando en cuenta las rutas más cortas entre dos puntos y consultando la información del tránsito referente a las avenidas y calles. Los objetivos principales de este sistema serían:

- Trazar rutas tomando en cuenta el flujo vehicular y la velocidad promedio de las avenidas y calles en la ruta.
- Reducir el tránsito en las avenidas principales de la ciudad.
- Distribuir de forma eficiente el tránsito por las calles y avenidas de la ciudad.
- Disminuir la cantidad de gases contaminantes generados por un tránsito pesado en las avenidas y calles de la ciudad.

También se puede ampliar la funcionalidad a este sistema, añadiendo módulos para:

- **Reconocimiento de automóviles:**

Este serviría para poder obtener estadísticas útiles para las empresas dedicadas a la venta de coches, más concretamente para poder realizar campañas de publicidad enfocadas a un mercado de coches específicamente.

Referencias

- [1] Proyecto piloto RETROFIT, Contaminación del aire y transporte vehicular en la ZMVM, Recuperado 28 de Noviembre, 2010, de <http://www.sma.df.gob.mx/retrofit/>
- [2] Greenpeace México. Guía de Ahorro de Energía (Marzo 2007), Recuperado 28 de Noviembre, 2010, de www.greenpeace.org
- [3] Martin J. C y Miravete de Marco A., *Mecánica del Automóvil*, Editorial INO Reproducciones, 1997, capítulo 9 Proyectos, pp. 237,238,239,240,241
- [4] Universidad de Illinois-Chicago Centro de Transporte Urbano, *Reporte de Evaluación No.1 del Proyecto ADVANCED*.
- [5] The Diebold Institute for Public Policy Studies, Inc., *The Development of Intelligent Transportation Systems*, Estados Unidos de América, 1995, Representative ITS Implementations, pp. 163, 186,187.
- [6] Proyecto Traffic & Cameras, Departamento de transporte del Estado de Washington, [en línea]. Recuperado el 04 de Enero de 2011. Dirección URL <<http://wsdot.wa.gov/Traffic/Cameras/default.aspx>>
- [7] Granados F. A. y Marín H. J. I., *Detección de flujo vehicular basado en visión artificial*, Universidad Tecnológica de Pereira Colombia, 2007, pp. 163-168.
- [8] Comisión Nacional para el uso eficiente de la energía (CONUEE). Manual de conducción. Técnico-Económica para Operadores de Automóviles del sector Público, [en línea]. Recuperado el 28 de Noviembre de 2010. Dirección URL< www.conuee.gob.mx>
- [9] K. N. Plataniotis, C.S. Regazzoni. "Visual-Centric Surveillance Networks and Services". IEEE signal Processing Magazine, vol. 22, 2005.
- [10] M. Ebbecke, M. B. H. Ali, A. Dengel. "Real Time Object Detection, Tracking and Classification in Monocular Image Sequences of Road Traffic Scenes" In Proc. 1997. International Conference on Image Processing, vol 2, pp. 402-405.
- [11] Michalopoulos, P. G. (1991). Vehicle Detection Video Through Image Processing: The Autoscope System. *IEEE Transactions On Vehicular Technology*, VOL. 40, NO. 1.

- [12] Siyal, M. Y., & Fathy, M. (1995). Real-Time Measurement Of Traffic Queue Parameters By Using Image Processing Techniques. *Image Processing And Its Applications IEEE*.
- [13] Siyal, M. Y., & Fathy, M. (1996). Triple RISC image processor for real-time image processing applications. *Inst. Electr. Eng. Electr. Lett.*, vol 32, pp. 2224–2225.
- [14] Hoose, N. (1991). Computer Image Processing in Traffic Engineering. *U.K.Taunton Research Studies*.
- [15] Dickinson, K. W., & Waterfall, R. C. (1984). Image processing applied to traffic: A general review. *Traffic Eng. Contr.*, vol 1, pp. 6–13.
- [16] Rourke, A., & Bell, M. G. (1988). Traffic analysis using low cost image processing. *Proc. Seminar on Transportation Planning Methods, PTRC, Bath, U.K.*
- [17] Hashimoto, N. (1988). Development of an image processing traffic flow measurement system. *Sumitomo Electronic Tech. Rev.*, no. 25, pp.133–138.
- [18] Takaba, S. (1984). A traffic flow measuring system using a solid state sensor. *Proc. IEE Conf. Road Traffic Data Collection, London, U.K.*
- [19] Fathy, M., & Siyal, M. Y. (1998). A Window-Based Image Processing Technique for Quantitative and Qualitative Analysis of Road Traffic Parameters. *IEEE Transactions On Vehicular Technology*, VOL. 47, NO. 4.
- [20] Dailey, D. J., Cathey, F. W., & Pumrin, S. (2000). An Algorithm to Estimate Mean Traffic Speed Using Uncalibrated Cameras. *IEEE Transactions On Intelligent Transportation Systems*, VOL. 1, NO. 2.
- [21] Yang, J., Wang, Y., Sowmya, A., & Li, Z. (2010). Vehicle Detection and Tracking With Low-Angle Cameras. *IEEE 17th International Conference on Image Processing*, 685-688.
- [22] Gu, Y., Yendo, T., Tehrani, M. P., Fujii, T., & Tanimoto, M. (2011). Traffic Sign Detection in Dual-Focal Active Camera System. *IEEE Intelligent Vehicles Symposium (IV)*.
- [23] Padillo, M. José, M. (2003). “Apuntes de ingeniería de tráfico”. Madrid, España. E.T.S. Ingenieros de Caminos, Canales y Puertos.
- [24] Nicolas J. Garber, Lester A. Hoel, *Ingeniería de Tránsito y Carreteras (2005)*. Thomson, Tercera edición.

- [25] Fathy, M., & Siyal, M. Y. (1998). A Window-Based Image Processing Technique for Quantitative and Qualitative Analysis Of Road Traffic Parameters. IEEE Transactions On Vehicular Technology, VOL. 47, NO. 4.
- [26] Gonzalez, R. C., and Woods, P., Digital Image Processing, Addison Wesley, 2002
- [27] Azriel Rosenfeld and Avinash C. KaK. *Digital Picture Processing*. Academic Press, 1982.
- [28] K. Pratt, William. *Digital Image Processing*. John Wiley & Sons, 1991.
- [29] S. Pressman Roger. *Ingeniería de software un enfoque práctico* (2006). Mc Graw Hill Sexta Edición.
- [30] Sommerville Ian. *Ingeniería de software* (2005). Pearson Educación, Madrid. Pág. 712.
- [31] Spark Systems, The Use Case Model. (2000). Recuperado 13 de septiembre, 2011, de http://www.sparxsystems.com/resources/tutorial/use_case_model.html
- [32] The Computer Language Benchmarks Game. (2004). Recuperado 14 de agosto, 2011, de <http://shootout.alioth.debian.org/>
- [33] Documentación. *Marvin Image Processing Framework*. (2011). Recuperado 23 de agosto, 2011, de <http://marvinproject.sourceforge.net/en/index.html>
- [34] Documentación. *Cinder Library* (2011). Recuperado 23 de agosto, 2011, de <http://libcinder.org/about/>
- [35] Documentación. *Ultimate++ Framework*. (2011). Recuperado 23 de agosto, 2011, de <http://www.ultimatepp.org/>
- [36] Documentación. *Qt SDK Framework*. (2008-2012). Recuperado 23 de agosto, 2011, de <http://qt.nokia.com/products/>
- [37] Consejo Federal de Seguridad Vial. (1995). Ley de Tránsito, Ley N° 24.449. Artículo 51 Velocidad Máxima, Límites máximos de velocidad en zona urbana.
- [38] Microsoft Sql Server Product Information. (2011). Recuperado 04 de febrero, 2012, de <http://www.microsoft.com/es-es/sqlserver/product-info/why-sql-server.aspx>

- [39] Microsoft Sql Server Compare Competitors. (2011). Recuperado 04 de febrero, 2012, de <http://www.microsoft.com/es-es/sqlserver/product-info/competitor-compare.aspx>
- [40] Microsoft Sql Server Benchmarks. (2011). Recuperado 04 de febrero, 2012, de <http://www.microsoft.com/es-es/sqlserver/product-info/benchmarks.aspx>
- [41] MSDN Microsoft Requisitos de hardware y software para instalar SQL Server 2008. (2012). Recuperado 06 de febrero, 2012, de <http://msdn.microsoft.com/es-es/library/ms143506%28v=sql.100%29.aspx>
- [42] MSDN Library Database Files. (2012). Recuperado 11 de febrero, 2012, de [http://msdn.microsoft.com/en-us/library/ms189563\(v=sql.90\).aspx](http://msdn.microsoft.com/en-us/library/ms189563(v=sql.90).aspx)
- [43] Developer Network SQL Database Drivers (2011). Recuperado 20 de febrero, 2012, de <http://qt-project.org/doc/qt-5.0/sql-driver.html#development>
- [44] Pajares, Gonzalo; De la cruz, Jesús, Visión por computador: Imágenes digitales y aplicaciones, Alfaomega, 2008