

**UNIVERSIDAD NACIONAL DEL SANTA
ESCUELA DE POSGRADO
PROGRAMA DE DOCTORADO EN INGENIERÍA DE
SISTEMAS E INFORMÁTICA**



UNS
ESCUELA DE
POSGRADO

**Sistema inteligente de gestión de
semáforos en tiempo real**

**Tesis para optar el grado académico de
Doctor en Ingeniería de Sistemas e Informática**

Autor:

**Ms. Apestegui Florentino, Yim Isaias
Código ORCID: 0000-0003-2873-1748**

Asesor:

**Dr. Caselli Gismondi, Hugo Esteban
DNI N° : 32819296
Código ORCID: 0000-0002-2812-6727**

**Línea de investigación
Sistemas inteligentes**

NUEVO CHIMBOTE - PERÚ

2024



UNS
ESCUELA DE
POSGRADO

CONSTANCIA DE ASESORAMIENTO

Yo, **Hugo Esteban Caselli Gismondi**, mediante la presente doy conformidad de mi asesoramiento de la tesis doctoral titulada "**Sistema inteligente de gestión de semáforos en tiempo real**" que tiene como autor al **Ms. Yim Isaias Apestegui Florentino**, que ha sido elaborada de acuerdo con el Reglamento General de Grados y Títulos de la Universidad Nacional del Santa para obtener el grado de Doctor en Ingeniería de Sistemas e Informática de la Escuela de Posgrado de la Universidad Nacional del Santa.

Nuevo Chimbote, julio de 2024

Dr. Caselli Gismondi, Hugo Esteban

Asesor

DNI N°: 32819296

Código ORCID: 0000-0002-2812-6727



UNS
ESCUELA DE
POSGRADO

AVAL DEL JURADO EVALUADOR

Sistema Inteligente de Gestión de Semáforos en tiempo Real

**Tesis para optar el grado de Doctor en Ingeniería de Sistemas e
Informática**

Revisado y Aprobado por el Jurado Evaluador:

Dr. Vega Moreno, Carlos Eugenio

PRESIDENTE

Código ORCID: 0000-0003-2955-0674

DNI N°.:32937583

Dra. Briones Pereyra, Lizbeth Dora

SECRETARIA

Código ORCID: 0000-0003-0626-7227

DNI N°: 32960646

Dr. Caselli Gismondi, Hugo Esteban

VOCAL

Código ORCID: 0000-0002-2812-6727

DNI N°: 32819296



UNS
ESCUELA DE
POSGRADO

ACTA DE EVALUACIÓN DE SUSTENTACIÓN DE TESIS

A los seis días del mes de julio del año 2024, siendo las 11:00 horas, en el Laboratorio de Aplicaciones Informáticas III de la EP de Ingeniería de Sistemas de la Universidad Nacional del Santa, se reunieron los miembros del Jurado Evaluador, designados mediante Resolución Directoral N° 105-2024-EPG-UNS de fecha 05.03.2024, conformado por los docentes: Dr. Carlos Eugenio Vega Moreno (Presidente), Dra. Lizbeth Dora Briones Pereyra (Secretaria) y Dr. Hugo Esteban Caselli Gismondi (Vocal); con la finalidad de evaluar la tesis titulada **SISTEMA INTELIGENTE DE GESTIÓN DE SEMÁFOROS EN TIEMPO REAL**; presentado por el tesista **Yim Isaias Apestegui Florentino**, egresado del programa de **Doctorado en Ingeniería de Sistemas e Informática**.

Sustentación autorizada mediante Resolución Directoral N° 332-2024-EPG-UNS de fecha 26 de junio de 2024.

El presidente del jurado autorizó el inicio del acto académico; producido y concluido el acto de sustentación de tesis, los miembros del jurado procedieron a la evaluación respectiva, haciendo una serie de preguntas y recomendaciones al tesista, quien dio respuestas a las interrogantes y observaciones.

El jurado después de deliberar sobre aspectos relacionados con el trabajo, contenido y sustentación del mismo y con las sugerencias pertinentes, declara la sustentación como Aprobado, asignándole la calificación de 19.

Siendo las 12:23 horas del mismo día se da por finalizado el acto académico, firmando la presente acta en señal de conformidad.

Dr. Carlos Eugenio Vega Moreno
Presidente

Dra. Lizbeth Dora Briones Pereyra
Secretaria

Dr. Hugo Esteban Caselli Gismondi
Vocal



Recibo digital

Este recibo confirma que su trabajo ha sido recibido por **Turnitin**. A continuación podrá ver la información del recibo con respecto a su entrega.

La primera página de tus entregas se muestra abajo.

Autor de la entrega:	Yim Isaias Apestegui Florentino
Título del ejercicio:	Revisión de Similitud
Título de la entrega:	INFORME POSTERIOR SUSTENTACION
Nombre del archivo:	2024-InformeTesis-YIM-APESTEGUI-PostSustentacion.docx
Tamaño del archivo:	9.08M
Total páginas:	159
Total de palabras:	31,425
Total de caracteres:	160,108
Fecha de entrega:	10-jul.-2024 07:27p. m. (UTC-0500)
Identificador de la entrega...	2414963123



Turnitin Informe de Originalidad

Procesado el: 10-jul-2024 7:28 p. m. -05

Identificador: 2414963123

Número de palabras: 31425

Entregado: 1

INFORME POSTERIOR SUSTENTACION Por Yim
Isaias Apestequi Florentino

Índice de similitud

14%

Similitud según fuente

Internet Sources: 13%
Publicaciones: 5%
Trabajos del estudiante: 9%

1% match ()

[Calderon Rojas, Milagros, "Seguridad ciudadana y comercio informal ambulatorio en los comerciantes, del distrito del San Juan del Lunigancho - 2023", "Universidad Cesar Vallejo", 2024](#)

< 1% match ()

[Arriola Pareja, Henry, Gamarra Sánchez, Jehzeel, "Comparación de intersecciones convencionales versus intersecciones tipo tokio usando micro simulación vehicular y peatonal en VISSIM en intersecciones de la Avenida de la Cultura y Av. El Sol en la ciudad del Cusco, año 2021", "Baishideng Publishing Group Inc.", 2022](#)

< 1% match ()

[Giménez Campos, Javier, "APLICACIONES DE GPUs EN VISIÓN E INTELIGENCIA ARTIFICIAL PARA EL RECONOCIMIENTO DE FORMAS", "Universitat Politècnica de Valencia", 2021](#)

< 1% match ()

[Chancasanampa Carmona, Yari, Cuba Paucar, Julio Cesar, "Implementación de la metodología 6s para mejorar la productividad en el área de distribución de Softys Perú SAC, Cañete - 2022", "Universidad Cesar Vallejo", 2023](#)

< 1% match ()

[Delgado Marti, Alejandro, "Desarrollo de un sistema de detección y predicción de la pose 3D de objetos en la escena mediante técnicas de deep learning", 2019](#)

< 1% match ()

[Arzapalo Camarena, Richard Ronald, "Estudio del tráfico vehicular y solución integral para el mejoramiento del transporte en el centro histórico del distrito de Ayacucho, 2021", "Universidad Cesar Vallejo", 2021](#)

< 1% match ()

[Aguinaga Landa, Marco Antonio, Salvador Vásquez, Joseph Smith, "Sistema inteligente basado en Machine Learning para la predicción de inundaciones en zonas agrícolas del Perú", "Universidad Cesar Vallejo", 2023](#)

< 1% match ()

[Llave Condori, Jhonatan Marcos, Sánchez Anicama, Fiama Yadira, "Modificación e implementación de una infraestructura vial para el mejoramiento de la transitabilidad vehicular y peatonal, Imperial - Cañete", "Universidad Cesar Vallejo", 2021](#)

< 1% match ()

[Maquera Cruz, Pedro Valerio, "Evaluación del Nivel de Servicio de Flujos Vehiculares, en Dos Intersecciones SemafORIZADAS](#)

DEDICATORIA

A Dios, fuente inagotable de energía,
luz y vida, quien nos sostiene cada día
y permite las circunstancias de la vida
para nuestro aprendizaje y para
nuestro bien.

A mi esposa, quien siempre está
motivándome a seguir
esforzándome, estudiando
rumbo a nuevos retos, y a desarrollar
la vocación de servicio.

A mis hijos, quienes son la
principal razón de seguir
luchando en esta vida, a fin
de verlos realizados en su
vida académica y personal.

AGRADECIMIENTOS

Al **Dr. Hugo Esteban Caselli Gismondi**, docente principal de la Universidad Nacional del Santa (UNS), quien me ha asesorado en la ejecución, seguimiento y culminación de la investigación.

Al **Dr. Jorge Luis Gutiérrez**, docente de la Universidad Nacional de Trujillo (UNT) quien me asesoró en la idea inicial de investigación y planteamiento del proyecto de investigación

Al **Dr. Luis Vladimir Urrelo Huiman**, docente de la Universidad Privada Antenor Orrego (UPAO), quien con sus experiencias, aportes y conocimientos nos encaminó a iniciar el proceso de investigación

Finalmente, mi agradecimiento a la **Escuela de Posgrado de la UNS** por ser el siguiente peldaño en la formación científica, académica y humanista a nivel doctoral y nos ha permitido llegar al más alto grado académico que confiere la universidad.

ÍNDICE GENERAL

DEDICATORIA	vii
AGRADECIMIENTOS.....	viii
RESUMEN	xix
ABSTRACT	xx
CAPÍTULO I	1
INTRODUCCIÓN	1
1.1 Planteamiento y fundamento del problema de investigación	1
1.1.1 Crecimiento del parque automotor desordenado	2
1.1.2 El problema de la congestión vehicular	2
1.1.3 Pico y placa una solución temporal.....	3
1.1.4 Causas principales de la congestión vehicular.....	4
1.1.5 El problema de la semaforización	4
1.1.6 Observación del problema de la asignación de tiempos	5
1.1.7 El problema de los semáforos inteligentes	6
1.1.8 El sistema inteligente como componente	7
1.2 Formulación del problema de investigación	8
1.3 Delimitación del estudio	8
1.4 Objetivos de la investigación.....	9
1.4.1 Objetivo general	9
1.4.2 Objetivos específicos.....	9
1.5 Formulación de la hipótesis central de la investigación	9
1.5.1 Variables e indicadores de la investigación	9
1.5.2 Indicadores	10
1.6 Justificación e importancia de la investigación	11

1.6.1	Justificación social	11
1.6.2	Justificación teórica	11
1.6.3	Justificación metodológica	11
1.6.4	Justificación práctica	11
CAPÍTULO II		12
MARCO TEÓRICO		12
2.1	Antecedentes de la investigación.....	12
2.2	Fundamentos teóricos de la investigación.....	14
2.2.1	Sistema inteligente de gestión del tráfico	14
2.2.2	Sistema de control de señales de tráfico metaheurísticos	14
2.2.3	Optimización de colonias de hormiga	15
2.2.4	Sistema de control de señales de tráfico con Lógica Difusa	16
2.2.5	Algoritmo de control difuso autoadaptativo de dos etapas.....	17
2.2.6	Sistema de control de señales de tráfico con RNA	17
2.2.7	Redes Neuronales Convolucionales	17
2.2.8	Redes Neuronales Recurrentes	17
2.2.9	Redes Neuronales LSTM	18
2.2.10	Diseño de sistemas en tiempo real.....	19
2.2.11	Detección de objetos y seguimiento de objetos	19
2.2.12	Algoritmo K-means	20
2.2.13	Método de Desplazamiento Meanshift.....	22
2.2.14	Método de flujo óptico	24
2.2.15	Método de flujo óptico de Lucas_Kanade.....	26
2.2.16	El filtro de Kalman	27
2.2.17	Redes de regresión profunda.....	28
2.2.18	Método ROLO	29

2.2.19	Método YOLO.....	30
2.2.20	Los semáforos	34
2.2.21	Precisión en la clasificación de objetos	35
2.2.22	Cálculo del MAPE, MAE Y RMSE	36
2.3	Marco conceptual	37
2.3.1	Glosario de términos por categoría.....	37
2.3.2	Determinación de la densidad del tráfico.....	39
2.3.3	Determinación de la intensidad del tráfico	40
2.3.4	Determinación de la velocidad del tráfico	41
CAPÍTULO III		42
MATERIALES Y MÉTODOS		42
3.1	Enfoque epistemológico	42
3.1.1	Matriz de consistencia	44
3.1.2	Operacionalización de variables	45
3.2	Métodos de la investigación	46
3.3	Diseño o esquema de la investigación	46
3.4	Población y muestra	46
3.5	Actividades del proceso investigativo	47
3.6	Técnicas e instrumentos de recolección de datos.....	48
3.7	Procedimiento de recolección de datos	49
3.7.1	Adquisición de imágenes	49
3.7.2	Recolección de datos de velocidad media.....	50
3.7.3	Recolección de datos para conteo tiempo de fase.....	52
3.8	Técnicas de procesamiento y análisis de los datos	53
3.8.1	Conteo de vehículos.....	53
3.8.2	Determinación de la medida de la velocidad	56

3.8.3	Determinación de características de tráfico.....	57
3.9	Algoritmo para flujos no conflictivos.....	73
3.10	Algoritmo de inicio y configuración.....	74
3.11	Algoritmo respecto al número de vehículos y densidad.....	75
3.12	Algoritmo gestión de operaciones con la velocidad media	76
3.13	Algoritmo para obtener el tipo de congestión	78
3.14	Algoritmo de conteo regresivo	79
3.15	Algoritmo del semáforo	79
3.16	Algoritmo de predicción de estado con Cadena de Markov.....	81
3.17	Algoritmo de Cadena de Markov para los tiempos de Espera.....	83
3.18	Ejecución de aplicación Semáforo.....	85
3.19	Despliegue de componentes.....	87
3.20	Plataforma de gestión de Confluent Kafka	89
3.21	Aplicación cliente Web responsivo.....	90
CAPÍTULO IV.....		91
RESULTADOS Y DISCUSIÓN.....		91
4.1	Desarrollo del sistema inteligente en tiempo real.....	91
4.2	Determinación de características de tráfico en intersección vial	92
4.2.1	Respecto al factor de índice.....	94
4.2.2	Respecto a la predicción de estado basado en densidad.....	96
4.2.3	Respecto a la predicción de estados hipotéticos.....	98
4.2.4	Respecto al tipo de flujo	100
4.3	Verificación de un modelo de reconocimiento y conteo	101
4.3.1	Supuesto de normalidad de la variable precisión.....	101
4.3.2	Respecto a la velocidad	103
4.3.3	Supuesto de normalidad de la variable DIFERENCIA	103

4.3.4	Supuesto de normalidad de la variable PRECISIÓN	105
4.3.5	Prueba de una muestra: precisión	106
4.4	Desarrollo del algoritmo de obtención de características.....	107
4.4.1	Determinación del ICT del tramo actual.....	108
4.5	Desarrollo del algoritmo de asignación de intervalos de tiempos	109
4.6	Simulación del sistema de control de semáforo	110
4.6.1	Supuesto de normalidad de tiempo de espera.....	111
4.6.2	Prueba no paramétrica de una muestra	112
4.6.3	Respecto a los tiempos de espera con cadena de Markov.....	115
4.7	Respecto a la hipótesis central de investigación	117
CAPÍTULO V.....		119
CONCLUSIONES Y RECOMENDACIONES.....		119
5.1	Conclusiones	119
5.2	Recomendaciones.....	122
VI. REFERENCIAS BIBLIOGRÁFICAS Y VIRTUALES		123
VII. ANEXOS.....		129

ÍNDICE DE TABLAS

Tabla 1 Matriz de consistencia problema, objetivo, hipótesis y variables	44
Tabla 2 Matriz de operacionalización de variables	45
Tabla 3 Técnicas e instrumentos de recolección de datos	48
Tabla 4 Tabla de asignación de tiempos de fase en una intersección	52
Tabla 5 Dimensiones de vehículos ligeros	62
Tabla 6 Prueba de normalidad de los datos de vehículos ligeros	63
Tabla 7 Tramos vehiculares con longitud y ancho respectivo	66
Tabla 8 Tipo de congestión de acuerdo al ICT.....	67
Tabla 9 Escenario 1: Densidad en 2 carriles menor que densidad de 3 carriles	67
Tabla 10 Escenario 2: Densidad en 2 carriles mayor que densidad de 3 carriles.	68
Tabla 11 Escenario 3: Congestión alta en 2 y 3 carriles	68
Tabla 12 <i>Matriz de transición M1 de la cadena de Markov</i>	82
Tabla 13 <i>Matriz de transición para predicción de tiempos de espera</i>	83
Tabla 14 Comparación de tiempos de espera de vehículos	92
Tabla 15 Matriz de transición estabilizada	96
Tabla 16 Matriz de transición para estados hipotéticos	98
Tabla 17 Matriz de transición estabilizada para estados hipotéticos	99
Tabla 18 Tabla de flujos de tráfico en una intersección	100
Tabla 19 Estadístico descriptivo de la variable precisión	102
Tabla 20 Pruebas de normalidad de la variable precisión	102
Tabla 21 Prueba de hipótesis del nivel de precisión de conteo de vehículos	102
Tabla 22 Estadísticos descriptivos de la variable diferencia	104
Tabla 23 Pruebas de normalidad de la variable diferencia	104
Tabla 24 Estadístico descriptivo de la variable precisión en la velocidad	105

Tabla 25	Pruebas de normalidad de la variable precisión en la velocidad	105
Tabla 26	Prueba de hipótesis del nivel de precisión de conteo de vehículos	106
Tabla 27	Tabla de pruebas T para una muestra, precisión en la velocidad.....	107
Tabla 28	Resultado de procesamiento y consumo de RAM del algoritmo.....	108
Tabla 29	Resultado de procesamiento de asignación de tiempos de color	109
Tabla 30	Criterios para el algoritmo de reajuste de tiempos de color	110
Tabla 31	Prueba de normalidad de tiempo de espera por ciclo	111
Tabla 32	Estadístico descriptivo para la variable tiempo de espera por ciclo	112
Tabla 33	Prueba de hipótesis de la variable tiempo de espera por ciclo	112

ÍNDICE DE FIGURAS

Figura 1	Red de prueba de 12 nodos con 4 intersecciones	15
Figura 2	Sistema de tiempo real embebido típico.....	19
Figura 3	Peores valores de error cuadrático en 10 ejecuciones	21
Figura 4	Secuencia de tráfico con el algoritmo MeanShift	23
Figura 5	Arquitectura de una red de seguimiento con regresión profunda	28
Figura 6	Arquitectura del método ROLO.....	29
Figura 7	Sistema de detección YOLO.....	33
Figura 8	Arquitectura de red YOLO.....	33
Figura 9	Comparación de medidas de precisión media de R-CNNs.....	36
Figura 10	Diagrama de bloques del procedimiento de recolección de datos	48
Figura 11	Ensamble de quipos para la captura de imágenes	50
Figura 12	Dimensiones de línea discontinua y espaciado de segmento	51
Figura 13	Detección en tiempo real	54
Figura 14	Solapamiento de vehículos	55
Figura 15	Flujos de tráfico en una intersección vial.....	57
Figura 16	Fases no conflictivas	58
Figura 17	Elementos de un tramo vehicular	59
Figura 18	Demarcación de línea de cruce peatonal	60
Figura 19	Datos de vehículos MTC.....	61
Figura 20	Esquema para el tiempo de arribo.....	70
Figura 21	Pares de flujos no conflictivos.....	73
Figura 22	Diagrama de estados de Markov para intercambios de color.....	81
Figura 23	Ejecución del programa de Control de Semáforo	86
Figura 24	Diagrama de despliegue	88
Figura 25	Infraestructura tecnológica.....	88
Figura 26	Plataforma Confluent Kafka	89

Figura 27 Aplicación web de datos de intersección.....	90
Figura 28 Arquitectura del modelo de Sistema Inteligente.....	91
Figura 29 Gráfico de la relación Velocidad Densidad.....	93
Figura 30 Factor de índice de vehículos livianos.....	95
Figura 31 Factor de índice de vehículos pesados.....	95
Figura 32 Secuencia de predicción de estados según la cadena de Markov.....	97
Figura 33 Secuencia de predicción de estados según estados hipotéticos.....	99
Figura 34 Gráfico de la mediana de tiempo de espera por ciclo.....	113
Figura 35 Gráfico de frecuencia de tiempo de espera por ciclo.....	113
Figura 36 Histograma simple del número de tramos vs. TIPO_CONGESTION.....	114
Figura 37 Secuencia de tiempos de espera con cadenas de Markov, caso 1.....	115
Figura 38 Secuencia de tiempos de espera con cadenas de Markov, caso 2.....	116
Figura 39 Secuencia de tiempos de espera con cadenas de Markov, caso 3.....	117

ÍNDICE DE ANEXOS

Anexo A: Principales intersecciones viales urbanas de la ciudad de Nuevo Chimbote, Perú	129
Anexo B: Mapa de la intersección vial Av. Country - Av. Pacífico, Nuevo Chimbote, Perú.....	130
Anexo C: Tabla de datos de conteo de vehículos	131
Anexo D: Velocidades obtenidas por conteo manual (observador) y por conteo de algoritmo	135
Anexo E: Tabla de datos de factor de índice de vehículos ligeros y pesados	136
Anexo F: Tabla de equivalencia de km/h, a otras unidades de medida	137
Anexo G: Velocidades permitidas en zonas urbanas y centros poblados	138
Anexo H: Extracto de ejecución de iteraciones de la matriz de transición de Markov.....	139
Anexo I: Extracto de la tabla de datos de trafico	140

RESUMEN

La congestión vehicular es un gran problema en las ciudades que van creciendo demográficamente, debido a sus efectos como son: contaminación ambiental por CO₂, afectación a la salud mental, estrés, malhumor y tiempos de retraso para llegar al centro de labores o estudios. Uno de los principales problemas se encuentra en la gestión del tráfico, específicamente en la gestión de semáforos que se configura con tiempos pre establecidos o fijos, haciendo asignaciones de tiempo de color verde aun cuando el tramo no necesita dicho tiempo o peor cuando el tramo no tiene vehículos en espera. El objetivo de esta tesis es reducir los tiempos de espera de vehículos en una intersección vial a través de un sistema inteligente basado en un modelo heurístico, de acuerdo a las características principales del tráfico como son velocidad y densidad vehicular. Para la fase de detección y seguimiento de objetos se ha utilizado librerías de OpenCV y Yolo y se ha diseñado modelos de algoritmos heurísticos que contemplen las características del tráfico para la toma decisiones en un entorno de tiempo real. Luego, se ha procedido a realizar las pruebas experimentales correspondientes. El modelo heurístico generó 15000 tramos a fin de contrastar el tiempo de espera por ciclo semafórico fijo vs. el tiempo de espera de ciclo semafórico variable. Se observó que, el modelo propuesto logró reducir en un 18,3% el tiempo de ciclo de semáforo y por ende mayor cantidad de ciclos generados por hora. En promedio, el modelo genera 47 ciclo/h con tiempos de ciclo variable frente a 39 ciclos/h de un semáforo con tiempos de ciclo fijo.

Palabras clave: algoritmo heurístico, congestión vehicular, seguimiento de objetos

ABSTRACT

Vehicle congestion is a big problem in cities that are growing demographically, due to its effects such as: environmental pollution by CO₂, impact on mental health, stress, bad mood and delay times to get to work or study. One of the main problems is found in traffic management, specifically in the management of traffic lights that are configured with pre-established or fixed times, making green time assignments even when the section does not need said time or worse when the section does not. has vehicles waiting. The objective of this thesis is to reduce vehicle waiting times at a road intersection through an intelligent system based on a heuristic model, according to the main characteristics of traffic such as speed and vehicle density. For the object detection and tracking phase, OpenCV and Yolo libraries have been used and heuristic algorithm models have been designed that take into account the traffic characteristics for decision making in a real-time environment. Then, the corresponding experimental tests were carried out. The heuristic model generated 15,000 sections in order to contrast the waiting time per fixed traffic light cycle vs. the variable traffic light cycle waiting time. It was observed that the proposed model managed to reduce the traffic light cycle time by 18,3% and therefore a greater number of cycles generated per hour. On average, the model generates 47 cycles/h with variable cycle times compared to 39 cycles/h for a traffic light with fixed cycle times.

Keywords: heuristic algorithm, vehicular congestion, object tracking.

CAPÍTULO I

INTRODUCCIÓN

1.1 Planteamiento y fundamento del problema de investigación

El crecimiento demográfico en las ciudades, procesos de urbanización, procesos de migración, crecimiento y necesidad de desarrollo económico requieren de un mayor consumo de energía, para los siguientes años. Según la Organización para la Cooperación y el Desarrollo Económicos (OCDE), la tasa de crecimiento de la demanda de energía primaria en el mundo se incrementará en 1,46% anual de 2009 a 2035. Esta demanda impactará en el mayor consumo de combustibles fósiles y el aumento anual de 1,6% de emisiones de CO₂ (OCDE, 2011) . En esta demanda de consumo se encuentra el transporte. Trasladarse al lugar de trabajo, a los centros de estudios como colegios, institutos, universidades y otras actividades crean la necesidad que debe ser atendida por los servicios de transporte urbano público y particular, quienes consumen en su mayoría combustibles fósiles, por tanto, la contaminación por CO₂ está en crecimiento.

El transporte urbano debería obedecer a un plan de transporte debidamente gestionado a corto, mediano y largo plazo, por las autoridades competentes como gestión de las emisiones, gestión del parque automotor, gestión del tráfico, pero no es así. Aun cuando la OCDE en un estudio denominado Crecimiento Verde Sostenible, propone el desplazamiento de vehículos que consumen combustibles fósiles por vehículos eléctricos (OCDE, 2011), el Perú aún no plantea estrategias integrales, para dicha fase de migración hacia vehículos eléctricos. En consecuencia, no aplicar un plan de gestión para el transporte hace que éste se comporte de manera caótica en diversos puntos de las ciudades del Perú, generando entre los aspectos mencionados congestión de tráfico vehicular. Gestionar la congestión del tráfico vehicular debido al enorme crecimiento vehicular en las ciudades del Perú y del mundo, es una gran preocupación y un gran problema. Básicamente, la congestión de tráfico genera los siguientes grandes problemas.

Primero, la contaminación atmosférica que se genera debido al consumo de combustible en tiempo de encendido de los motores, causa enfermedades al sistema respiratorio y la piel. Segundo, la congestión de tráfico conlleva a problemas de la salud física como son los accidentes de tránsito, personas lesionadas con discapacidad temporal o permanente, inclusive hasta muertes, agregado a ello, problemas a la salud mental pues crea conflicto entre conductores imprudentes, estrés y malhumor (Salinas, 2018).

Por ende, la congestión de tráfico ha producido, aparte de pérdidas de vidas, pérdidas del patrimonio vehicular, pérdida de tiempo para todos los usuarios de la vía y en consecuencia pérdida de productividad.

1.1.1 Crecimiento del parque automotor desordenado

El crecimiento demográfico urbano y económico del Perú, en los últimos años reflejado en el ingreso promedio de los habitantes de las principales ciudades del país, ha permitido la compra y venta de vehículos, incrementando el parque automotor a nivel nacional. Evidentemente, el caso más notorio es en la ciudad de Lima, Perú, quien ha experimentado el crecimiento poblacional más desordenado y el crecimiento del parque automotor más pronunciado y desordenado de las últimas décadas (Bonifaz & Aparicio, 2013).

El caso Lima, se presenta por ser el más emblemático; situación similar ocurre en las ciudades más importantes del Perú, como Chimbote, Trujillo, Arequipa, Piura, Chiclayo entre otras ciudades, donde dicho crecimiento incide en problemas específicos del transporte como es la congestión vehicular.

1.1.2 El problema de la congestión vehicular

En el Perú, día a día los pasajeros tienen que soportar la mala calidad del servicio de transporte público local, la mala atención de los choferes y cobradores, la imprudencia de los mototaxistas, colectiveros y los conductores particulares. Una de las mayores incomodidades y por ende el mayor problema de los pasajeros y de los conductores en las vías de transporte es la congestión vehicular, o denominada también congestión vial.

Según un estudio realizado por la encuestadora Marketwin, el 92% de los habitantes de la ciudad de Lima afirmó que la congestión vehicular les genera estrés, mientras que el 82% señaló que este problema les quita calidad de vida (Manrique W. , 2017).

Al año siguiente la misma encuestadora reporta, que el 86% de los limeños afirmó que la congestión vehicular les genera estrés, mientras que el 74% señaló que este problema les quita calidad de vida y a un 57% el tráfico le deprime (Manrique W. , 2018).

1.1.3 Pico y placa una solución temporal

En la ciudad de Lima, Perú el 6 de agosto del 2019, se ejecutó el plan “pico y placa”, que sostiene que si se reduce el número de vehículos de acuerdo al número de placa por horas y días de circulación en la vía, obviamente el tráfico debe fluir más y mejor. La medida solo se aplicaba a determinadas vías de alto tráfico de lunes a jueves y en dos horarios: 6:30 a.m. a 10:00 a.m. y de 5:00 p.m. a 9:00 p.m., de acuerdo al número de placa o matrícula de los vehículos. Así, los vehículos con placas o matrículas terminadas en número par (incluido el cero) no circulaban lunes y miércoles. Y los vehículos cuyas placas o matrículas terminen en número impar no circulaban los martes y jueves.

Este modelo fue una imitación de la medida de “pico y placa” de la ciudad de Bogotá, Colombia. Pero, la realidad demuestra que es un plan que no cumple con el objetivo de la medida, sino que, por el contrario, ha generado un aumento del parque automotor, pues en el año 2002 había 500,000 automóviles registrados; al 2011 se registraron 1,400,000 automóviles particulares, generando inclusive más congestión de tráfico y contaminación que el transporte público y el transporte oficial (Camargo, 2018).

Mencionado lo anterior, la experiencia demuestra que hoy la ciudad de Bogotá ha vuelto a ser una ciudad con más congestión de tráfico y más contaminada que antes; respecto a la ciudad de Lima, al mes de abril del 2023 la medida se encuentra suspendida.

1.1.4 Causas principales de la congestión vehicular

De acuerdo al estudio realizado por Manrique (2018), presenta que las principales causas de congestión de tráfico vehicular son:

- a) Saturación de vehículos y exceso de carros:
- b) Señalización, semaforización y mala infraestructura.
- c) Falta de respeto a las normas y conductores imprudentes.:
- d) Escasos policías de tránsito.
- e) Incumplimiento de buenas normas y peatones imprudentes.

Esta realidad descrita como causas principales de la congestión vehicular aplica y se puede observar en las ciudades del Perú. Por lo que vemos que son una serie de causas muy recurrentes y comunes que a diario se presenta, sobre todo en ciertas horas de mayor tráfico de vehículos que son las denominadas “horas pico”. También existe otro tipo de causas, que se debe mencionar como: el transporte informal, pistas en mantenimiento o en mal estado, transporte pesado, colegios sin zona de parqueos y escasez de ciclovías.

1.1.5 El problema de la semaforización

Respecto al problema para analizar y resolver las causas principales de congestión vehicular, relacionado a saturación de vehículos y exceso de carros, son las autoridades del Perú quienes deberán abordarlo desde un punto de vista más holístico a través de un sistema integral del transporte público local, regional y nacional. Pues son causas relacionadas con políticas de gestión del transporte como la compra venta del parque automotor y gestión de la educación como la formación y capacitación de peatones, conductores. Por tanto, son las políticas de gestión gubernamentales cuya aplicación efectiva podrían regular este tipo de causas y por ende contribuir a disminuir la congestión vehicular.

En esta sección nos enfocaremos respecto al problema de la semaforización en las intersecciones viales de una ciudad, donde se presentan semáforos con luces malogradas, con poca visibilidad al tramo vehicular y tiempos asignados no de acuerdo a la realidad del tráfico.

Durante mucho tiempo los municipios a través de sus unidades correspondientes han convocado a licitaciones para la implementación de semaforización inteligente. Sin embargo, luego de la implementación se han presentado varios problemas siendo el más notorio el de asignación de tiempos inadecuados al tramo vehicular por lo que el término “inteligente”, queda sin efecto. Inclusive se hace necesario y recurrente la intervención de personal de la Policía de Tránsito del Perú, a fin de que intervenga ordenando el tráfico generado debido a bloqueos mutuos en la intersección.

1.1.6 Observación del problema de la asignación de tiempos

Se ha podido apreciar en el distrito de Nuevo Chimbote, Perú que la asignación de tiempos de luz verde, ámbar y rojo del semáforo no es la más justa y son predefinidas. Esto aplica a la mayoría de los semáforos del casco urbano (observar Tabla 4) donde los tiempos de rojo y verde son predefinidos. Por ejemplo en la intersección 7(Av. Pacífico y Av. Country) Plaza Mayor el semáforo está configurado para tiempo verde de 55s y tiempo rojo de 34s.

De esta manera, se presentan casos donde:

- a) Se tiene intersecciones en las que el semáforo de luz verde se pasa directo a luz roja, esto genera accidentes y congestión de tráfico en medio de la intersección.
- b) Se ha observado que el tiempo asignado a luz roja en vía principal es mayor que a su luz verde, lo que hace que se otorgue mayor tiempo en verde a la vía alimentadora donde el tráfico es menor que la vía principal.
- c) El uso de software incompatible entre dispositivos hace imposible un intento de sincronizar tiempos en dos tramos de

semaforización, a fin de crear “ola verde” y dar fluidez y continuidad en una misma vía.

- d) Los contratos no estipulan el uso de un protocolo único y universal en semaforización, a fin de regular los tiempos de los semáforos de las vías alimentadoras transversales con las vías principales.
- e) Los semáforos funcionando a tiempo predefinido hacen perder tiempo de 1,5 a 2 minutos en promedio por cada intersección, pues se asigna tiempos de luz verde a tramos vehiculares vacíos; entre tanto hay una larga cola en rojo que espera que termine el tiempo de una luz verde que no se utiliza en la vía trasversal.

1.1.7 El problema de los semáforos inteligentes

En el Perú se implementaron proyectos bajo la denominación “semáforos inteligentes”. Uno de ellos se implementó en el año 2009 donde algunas autoridades ediles realizaron una compra valorizada en 8 millones 309 mil 710 soles con 81 céntimos (Seace, 2019), situación que entre otros aspectos legales llevó a los involucrados a procesos penales por el aprovechamiento indebido del Decreto de Urgencia N° 041-2009, por sobrevalorar el proyecto en 3 millones 562 mil 692 soles adicionales.

Lo más relevante es que el proyecto de “Semáforos inteligentes” consideraba el mejoramiento de 22 intersecciones con ampliación a 26 nuevas intersecciones, mediante semáforos inteligentes con transmisión inalámbrica y centro de comando computarizado que se encargara del control y monitoreo, el proyecto incluía contadores vehiculares con 48 paneles, 48 sistemas de sincronismo inalámbrico con controladores de tráfico, pozos a tierra y como sustento del proyecto se plantean obtener una señalización adecuada de las intersecciones, brindar seguridad, fluidez al tránsito vehicular y peatonal. Se debe indicar que el proyecto ejecutado no presentó un estudio del tráfico real en cuanto a cantidad de automóviles o transitabilidad y tampoco pudo cumplir con las especificaciones requeridas.

Diez años después, en el 2018 se aprueba el proyecto de “Mejoramiento del sistema de seguridad vial en el casco urbano, centro

poblado de Buenos Aires, distrito de Nuevo Chimbote, provincia de Santa, región Ancash” cuyo objetivo es la seguridad en el casco urbano.

En dicho proyecto se establece el indicador de transitabilidad que se refiere a la cantidad de población atendida; iniciando el primer año con 33 mil 326 habitantes atendidos y proyectados a 10 años con 38 mil 105 habitantes atendidos. Además, se propuso un adecuado sistema de seguridad vial para el tránsito vehicular y peatonal, con un equipamiento de 19 semáforos valorizados en 819 mil 793 soles con 20 céntimos (Seace, 2019).

Como sustento del proyecto se menciona que de cada 2000 vehículos que circulan por las avenidas que deberían estar semaforizadas, un rango de 2 a 8 han sufrido choques involuntarios por colectivos y mototaxis. Indican además que, los beneficios cualitativos que genera el proyecto serían:

Aumento en la seguridad en el transporte de peatones, ahorro en tiempo de los peatones, facilidad de tránsito de los peatones y facilidad de acceso a sus inmuebles, mayor orden en el tránsito vehicular, reducción de la frecuencia de cierto tipo de accidentes y un ahorro considerable debido a un mejor control del tránsito con respecto a la utilización de otras formas de control, como por ejemplo policías o inspectores de tránsito. Se hace alusión nuevamente, que son semáforos inteligentes con controladores y cuentas regresivas.

Lo más relevante, es que las cuentas regresivas sí funcionan, pero todo indica que el concepto de semáforo inteligente aún se debe mejorar en su aplicación. Asimismo, se debe mencionar que se maneja el concepto de transitabilidad a nivel habitantes, mas no a nivel de densidad vehicular o conteo de vehículos.

1.1.8 El sistema inteligente como componente

Luego de lo mencionado en los párrafos anteriores, se deduce que tenemos el problema del conteo real de vehículos para la determinación de la densidad y el problema de la asignación de tiempos a los colores de semáforo, para mejorar la fluidez.

De manera que la propuesta es diseñar un modelo que tenga las características de un sistema inteligente que determine la situación de su entorno vehicular. Luego de percibido la situación del entorno pueda tomar decisiones inteligentes para asignar los tiempos más justo posibles de los colores verde, ámbar y rojo a los tramos correspondientes. Como se puede apreciar el diseño del sistema inteligente debe contemplar un aspecto relacionado al método eficiente para determinar la densidad vehicular y otro aspecto relacionado al método para la gestión de asignación de tiempos; la investigación se centró en ambos aspectos.

1.2 Formulación del problema de investigación

A continuación, se procede a formular el siguiente problema de investigación:

¿En qué medida un sistema inteligente basado en un modelo heurístico, mediante la gestión en tiempo real de un semáforo, reducirá los tiempos de espera de vehículos en una intersección vial?

1.3 Delimitación del estudio

- La investigación se desarrolló considerando la realidad de la asignación de tiempos de color de semáforo en una intersección vial, no considera semáforos en óvalo o rotonda. Siendo el tráfico vehicular un tema tan complejo por los innumerables tipos de escenarios que se presentan en cada tramo vehicular por cada intervalo de tiempo; la investigación ha recurrido a mecanismos de simulación para la generación de escenarios diversos.
- Esta investigación no considera procesos de sincronización de la red de semáforos. Aunque se ha tomado como datos de referencia a 9 intersecciones del distrito de Nuevo Chimbote, Perú (ver Tabla 4), el modelo aplica de manera general a cualquier semáforo de la localidad en una intersección vial de manera independiente
- El modelo tiene un enfoque de componentes de sistema inteligente en tiempo real como servicio mas no como sistema inteligente asíncrono o de sistema de información.

1.4 Objetivos de la investigación

1.4.1 Objetivo general

Reducir los tiempos de espera de vehículos en una intersección vial a través de un sistema inteligente basado en un modelo heurístico.

1.4.2 Objetivos específicos

- Determinar las características de tráfico en una intersección vial para un entorno de tiempo real
- Verificar un modelo de reconocimiento y conteo vehicular en tiempo real
- Desarrollar el algoritmo de obtención de características de tramo vehicular
- Desarrollar el algoritmo que permita asignar intervalos de tiempos de color al semáforo en el tramo correspondiente
- Simular el funcionamiento del sistema de control de semáforo

1.5 Formulación de la hipótesis central de la investigación

Un sistema inteligente basado en un modelo heurístico, mediante la gestión en tiempo real de un semáforo, reduce significativamente los tiempos de espera de los vehículos en una intersección vial.

1.5.1 Variables e indicadores de la investigación

Se propuso la siguiente definición y operacionalización de variables

1.5.1.1 Definición conceptual

Variable Independiente:

Sistema inteligente basado en un modelo heurístico con gestión en tiempo real de un semáforo: Sistema que puede percibir una realidad del tráfico vehicular, procesa datos y toma decisiones.

Variable Dependiente:

Tiempos de espera de los vehículos en una intersección vial: Tiempo promedio en que un grupo de vehículos está en espera en una intersección vial.

1.5.1.2 Definición operacional

Variable Independiente:

Sistema inteligente basado en un modelo heurístico con gestión en tiempo real de un semáforo: Ejecución de componentes de software que ejecutan algoritmos, procesan datos y emiten un resultado.

Variable Dependiente:

Tiempos de espera de los vehículos en una intersección vial: Ejecución del cálculo de los tiempos de fase, registra el tiempo de espera e indica el nivel de tráfico.

1.5.2 Indicadores

Indicadores de Variable independiente

- Número de vehículos
- Densidad vehicular
- Tiempo de color de semáforo

Indicadores de Variable dependiente

- Tiempo de espera
- Tipo de congestión

1.6 Justificación e importancia de la investigación

1.6.1 Justificación social

La necesidad de mejorar la calidad de vida de peatones y transeúntes en términos de mayor tiempo de productividad, menor tiempo de espera en intersección, menor consumo de combustible y contribuir a la conservación del medio ambiente a través del uso óptimo de las vías de tránsito justifica socialmente la presente investigación.

1.6.2 Justificación teórica

Esta investigación se realizó con el propósito de aportar al conocimiento teórico existente de los sistemas inteligentes, específicamente a los métodos de detección y reconocimiento de imágenes aplicados al tráfico vehicular y a los métodos de control de asignación de tiempos. Se incide fuertemente en el aspecto teórico, por cuanto constituye la base científica de la presente investigación.

1.6.3 Justificación metodológica

El sistema inteligente que se propuso, requiere de un proceso metodológico riguroso durante el diseño y la implementación del mismo, a fin de que el sistema pueda ser replicado en otros contextos. Una vez que se haya demostrado su validez y confiabilidad el modelo propuesto se podrá replicar con motivos de investigación o con motivos de implementación y/o producción en diferentes contextos de intersecciones viales.

1.6.4 Justificación práctica

La necesidad de mejorar la gestión del tráfico justifica el planteamiento de una solución como lo es un Sistema inteligente para la gestión de tráfico en intersecciones viales, de manera que esta investigación contribuye con un aporte a la solución a corto plazo al problema del tráfico en la ciudad y del transporte en general.

CAPÍTULO II

MARCO TEÓRICO

2.1 Antecedentes de la investigación

Se ha estudiado el problema del tráfico en Intelligent Traffic Analytics: From Monitoring to Controlling [Análítica de Tráfico Inteligente: Desde el Monitoreo al Control], donde se menciona que se puede monitorear tramos de carreteras a fin de predecir el tráfico. Para ello los autores proponen un sistema de tráfico inteligente denominado T4, el cual les permite un análisis inteligente sobre trayectorias históricas y en tiempo real de vehículos. Para ello entrenaron un modelo profundo para la predicción de flujo de tráfico y la luz de control de tráfico reducir la congestión del tráfico. En dicho trabajo se concluye que el resultado de esas consultas puede ser utilizado para fines adicionales de monitoreo y análisis. (RNN) (Wang et al., 2019).

Desde otro enfoque Inglés-Romero, Cano-Baños, Doménech-Asens, & Morales-Estera (2018) en Traffic Metrics at Urban Intersections using a Low-Cost Image Processing Approach [Métricas de tráfico en intersecciones urbanas utilizando un enfoque de procesamiento de imágenes de bajo costo] presentan un sistema para obtener métricas respecto al problema del tráfico en intersecciones urbanas. El sistema combina técnicas de sustracción de fondo con métodos de análisis de contraste para obtener métricas de tráfico basadas en varias regiones de interés. Para reducir los costos de instalación, el sistema ha sido diseñado para trabajar con cámaras ya instaladas. Las métricas utilizadas en este trabajo fueron: (a) la cantidad de vehículos que dejaron de esperar para que un semáforo rojo se vuelva verde y (b) el número vehículos que fluyen hacia la intersección justo antes que el semáforo cambie de verde a rojo.

Dicho trabajo también sugiere que, aparte de estas dos métricas, otras podrían proporcionarse, por ejemplo, (c) la densidad de tráfico que se puede calcular como el número de vehículos por unidad de tiempo. El objetivo es enviar estas métricas a un semáforo gestor para optimizar automáticamente las fases de tiempo verde para reducir la congestión.

Por tanto: el enfoque está en un sistema de visión por computadora de bajo costo para obtener métricas de tráfico en intersecciones urbanas basadas en Procesamiento de Textura de un conjunto de ROI (región de interés).

Respecto al problema de la congestión del tráfico Ashifuddin Mondal & Rehena (2019) en Intelligent Traffic Congestion Classification System using Artificial Neural Network [Sistema inteligente de clasificación de congestión de tráfico mediante red neuronal artificial] presentan una solución denominada Sistema de Transporte Inteligente (ITS) basado en Internet de las Cosas (IoT) que puede ayudar a administrar la congestión del tráfico de manera eficiente. Se considera que la Estimación y Clasificación del del estado de congestión del tráfico en diferentes segmentos de carreteras es uno de los aspectos importantes de la gestión inteligente del tráfico.

El sistema propuesto por Ashifuddin Mondal & Rehena usa sensores estacionarios y está basado en una red neuronal artificial (ANN) para clasificar los estados de congestión del tráfico. Según el estado de congestión del tráfico, el ITS actualizará automáticamente las regulaciones de tráfico como, cambiando la longitud de la cola en la señal de tráfico, lo que sugiere rutas alternativas. Está compuesto de 3 unidades: a) Unidad de recopilación de datos donde el sistema propuesto utiliza sensores estacionarios en carretera para monitorear en tiempo real el tráfico y capturar la velocidad de los vehículos; b) Unidad de transmisión de datos donde los datos de tráfico detectados son enviados de manera periódica del campo a la unidad de análisis de datos remotos, esto a través de comunicación inalámbrica o por cable. Y c) Unidad de procesamiento de información donde los datos de tráfico sin procesar recopilados del campo se utilizan para calcular la velocidad promedio de los vehículos y densidad de tráfico de la carretera.

Finalmente, los parámetros de tráfico vial considerados fueron: densidad de tráfico y velocidad como medidas para clasificar los estados de congestión del tráfico y tres niveles de congestión: congestión alta, congestión media y flujo libre (Ashifuddin Mondal & Rehena, 2019).

2.2 Fundamentos teóricos de la investigación

En este apartado, se aborda el tema de estudio desde diversas perspectivas teóricas en base a la revisión de la literatura actualizada y nos permite desarrollar una perspectiva teórica de investigación.

2.2.1 Sistema inteligente de gestión del tráfico

Un sistema inteligente para la gestión del tráfico tiene un conjunto de componentes como:

a) **Adquisición de imágenes:** que consiste en extraer una escena de la realidad a fin de ser convertido en píxeles, el objetivo de esta fase es tan importante, porque de la calidad y precisión que se obtenga de esta adquisición los algoritmos encontrarán características relevantes.

b) **Extracción de características:** en este aspecto el algoritmo es capaz de identificar características del escenario (vehículos, peatones, velocidad, dirección, matrículas, color, etc.), dependiendo de la necesidad características requeridas y también dependiendo de la velocidad de extracción de dichas características.

c) **Software para el Sistema inteligente de gestión de tráfico (ITMS):** actualmente las aplicaciones de software usan sensores de tráfico en tiempo real y el sistema de posicionamiento global (GPS) para obtener información en tiempo real del tráfico en la ciudad o carreteras (Nigam et al., 2023).

2.2.2 Sistema de control de señales de tráfico metaheurísticos

Hay una variedad de métodos aplicados a sistemas de control de señales de tráfico (TSCS) como los de optimización metaheurística como los algoritmos genéticos (GA), optimización de colonias de hormigas (ACO), optimización de enjambre de partículas (PSO) y búsqueda de tabú (TS) entre otros como los sistemas de control metaheurísticos híbridos que se propone en el “Modelo evolutivo Híbrido Anidado para la optimización de señales de tránsito” (Srivastava & Sahana, 2017).

2.2.3 Optimización de colonias de hormiga

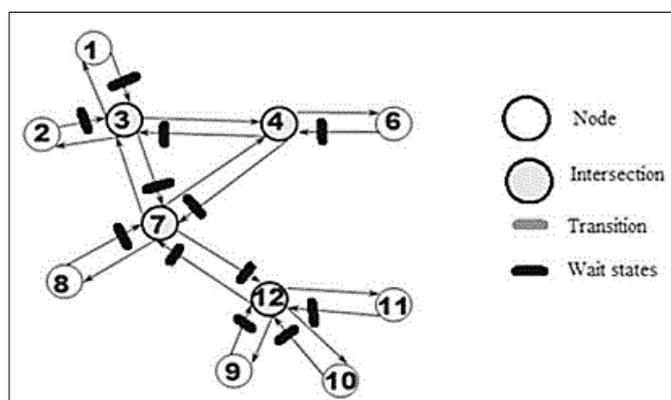
La optimización de colonias de hormigas (Ant Colony Optimization, ACO) está inspirada en el comportamiento de envejecimiento de las colonias de hormigas y la “evaporación” de sus feromonas. Las hormigas dejan sus feromonas cuando se mueven por el suelo, lo que hace que otras hormigas dejen de andar al azar. El algoritmo genético de Srivastava & Sahana (2017) utiliza el algoritmo 2.2, basado en el algoritmo 2.1, para la optimización de colonias de hormigas (Figura 1) y acelerar el proceso de convergencia (ir a glosario para ver los tipos de convergencia).

Algoritmo 2.1: Optimización de colonia de hormigas

```
Función ACO(grafo, num_hormigas, num_iteraciones):  
  Inicializar feromonas en el grafo  
  Mejor_solución_global = Vacío  
  Para cada iteración de 1 a num_iteraciones:  
    Construir solución de la hormiga basada  
    en reglas de probabilidad y feromonas  
    Evaluar la calidad de la solución construida  
    por la hormiga  
    Si la solución es mejor que Mejor_solución_global:  
      Actualizar Mejor_solución_global con la nueva solución  
      Actualizar feromonas en el camino de la hormiga  
      basado en la calidad de la solución  
      Evaporar feromonas en todos los caminos  
  Devolver Mejor_solución_global  
Fin de la función ACO
```

Figura 1

Red de prueba de 12 nodos con 4 intersecciones



Nota: las intersecciones 3,4,7 y 12 tienen flechas con sentido que confluyen y salen del mismo. Imagen tomada de Srivastava y Sahana (2017).

Algoritmo 2.2: Modelo evolutivo híbrido anidado

```
Procedimiento Híbrido
  Inicializar parámetros de GA
  Inicializar los parámetros de ACO
  Repetir
    Iniciar una nueva generación de hormigas
    ConstruirSolucionesHormiga
    Actualizar feromonas
  Fin
  Aplicar cruce y mutación sobre las n soluciones
  principales para generar p soluciones
  Actualizar feromonas en todas las rutas de solución p
  Hasta la condición de terminación
Fin del procedimiento
```

A partir de los resultados de la investigación con la demostración de la mejora del algoritmo híbrido ACO-GA frente a algoritmos individuales e independientes de ACO y GA para el problema de optimización de la sincronización de las señales de tránsito se evidencia que hay reducción del tiempo de espera en cada intersección y mejora la movilidad dentro de la carretera.

2.2.4 Sistema de control de señales de tráfico con Lógica Difusa

El estado del arte en el control de señales con lógica difusa se encuentra en el uso de controladores lógicos difusos. Primero, con la tasa de llegada y los vehículos de cola, se considera una regla específica de orden de fase para evitar la selección de fase desordenada en el control difuso. Segundo, el módulo de detección de tiempo verde decide si extender el tiempo verde actual o cambiar las fases por algunos segundos, por lo que aquí la tasa de llegada de vehículos no es necesaria como entrada al controlador en tiempo real (Taiping et al., 2021).

Para evitar la incertidumbre artificial se aplica un algoritmo de evolución diferencial con baja complejidad espacial y rápida convergencia.

Taiping et al. (2021) menciona que el mecanismo principal es la detección de tiempo verde y determinar si se debe cambiar la fase actual. Las variables de entrada son: el número de vehículos de cola de la fase actual y el número medio de vehículos de cola de otras fases.

2.2.5 Algoritmo de control difuso autoadaptativo de dos etapas

En este algoritmo la primera etapa calcula el grado de urgencia del tráfico para todas las fases rojas y selecciona la fase roja con el grado máximo de urgencia del tráfico como la siguiente fase verde. La segunda etapa determina si extender o terminar la fase de señal actual. El modelo considera un algoritmo genético híbrido mejorado para resolverlo y permitir que el controlador aprenda por sí mismo (Wang et al., 2022).

Bajo una plataforma de microsimulación basada en el lenguaje VISSIM y Python se evalúa la eficiencia y confiabilidad del controlador en condiciones de tráfico complejas. Los resultados mostraron que el tiempo promedio de retraso por vehículo se reduce en un 14,59%, mientras que el número promedio de paradas por vehículo se reduce en un 0,71% en comparación con el método de control tradicional.

2.2.6 Sistema de control de señales de tráfico con RNA

Una red neuronal artificial (RNA) es un modelo computacional, cuya estructura se basa en las redes neuronales biológicas, y está compuesta de muchas unidades interconectadas denominada neuronas artificiales (Gómez S, 2016).

2.2.7 Redes Neuronales Convolucionales

Las redes neuronales convolucionales (CNN o ConvNet) son un tipo de redes neuronales profundas aplicados en visión artificial e imágenes visuales. La CNN a diferencia de una red neuronal simple, constan de múltiples capas que requiere de un pre procesamiento relativamente pequeño si se quisiera comparar frente a otros algoritmos de clasificación de imágenes (Bendemra, 2018).

2.2.8 Redes Neuronales Recurrentes

Una red neuronal recurrente (RNN) representa un tipo de red neuronal artificial que se especializa en el procesamiento de datos secuenciales y series temporales.

Son usados para resolver problemas que involucran orden o dependencia temporal, como la traducción de idiomas, el reconocimiento de voz y la generación de subtítulos para imágenes. Por ejemplo, en el reconocimiento del habla es de mucha importancia comprender las palabras de una frase a fin de predecir la palabra siguiente (Shiri F. et al., 2023).

En una arquitectura RNN, basada en el modelo codificador-decodificador el codificador lee y codifica la oración fuente en un vector de longitud fija mientras que el decodificador produce una salida de traducción. La salida de traducción es un del vector codificado (Bahdanau et al., 2014). En consecuencia, las RNN permiten utilizar los recursos humanos y de datos de manera eficiente y tal ha sido el éxito de estos modelos que las principales empresas de traducción automática (MT) como Google, Systran, Microsoft, etc. los han integrado en las tecnologías de sus traductores automáticos (Rivera-Trigueros, 2022).

2.2.9 Redes Neuronales LSTM

Long Short-Term Memory (LSTM) es una variante de las RNN, que se caracteriza por captar las dependencias a largo plazo. En comparación con las RNN estándar, los modelos LSTM han demostrado ser más eficaces a la hora de retener y utilizar información en secuencias largas o largos periodos de tiempo (Shiri F. M. et al., 2023). La LSTM consta de tres puertas: puerta de entrada, puerta de olvido y puerta de salida. Cada puerta realiza una función específica en el control del flujo de información.

En concreto, la **puerta de entrada** determina la entrada del paso de tiempo actual y la actualización del estado interno del estado del sistema del paso de tiempo anterior; la **puerta de olvido** determina la actualización del estado interno del paso de tiempo anterior al estado interno del paso de tiempo actual; la **puerta de salida** determina el efecto del estado interno del paso de tiempo actual sobre el estado interno del paso de tiempo anterior (Fang et al., 2021).

2.2.10 Diseño de sistemas en tiempo real

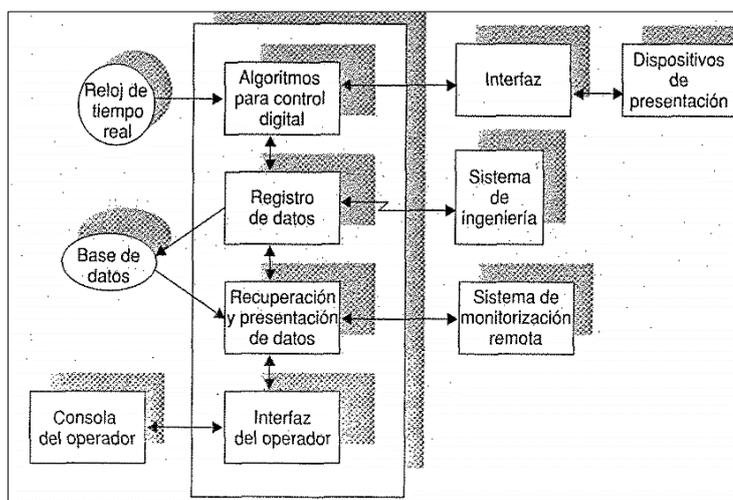
Para diseñar un sistema en tiempo real (STR), debe cumplir tres condiciones básicas:

1) Interactúa con el mundo real, 2) emite respuestas correctas y 3) cumple restricciones temporales (Burns & Wellings, 2003). Un sistema real comprende algoritmos que funcionan controlando en función de reloj de tiempo real, así como el registro, recuperación y presentación de datos asociados a una base de datos.

Además un STR, puede requerir de una interfaz de operador, consola de operación, sistemas de monitorización, etc. tal como se puede apreciar en la Figura 2, un sistema embebido típico.

Figura 2

Sistema de tiempo real embebido típico



Nota: Cuando se menciona tiempo real se debe considerar que todo el sistema está funcionando en tiempo real. Figura tomada de Burns y Wellings (2003).

2.2.11 Detección de objetos y seguimiento de objetos

A estas alturas de la revisión de la literatura se puede definir que no es lo mismo detección de objetos y seguimiento de objetos (Asmara et al., 2020).

Detección de objetos: En términos simples, en la detección de objetos detectamos un objeto ubicado en un fotograma.

La idea es que ello permita clasificar al objeto y el algoritmo detector de objeto culminará su trabajo cuando encuentra un margen de error mínimo aceptable o un porcentaje de similitud de acuerdo a su entrenamiento y/o aprendizaje.

Seguimiento de objetos: si se necesita hacer un seguimiento a los objetos, como sería el caso de los automóviles que se acercan a una intersección de semáforo a velocidades muy variadas, entonces ya estamos hablando de un seguidor de objetos o rastreador de objetos. La idea es que se mantenga reconocido al mismo objeto en los diversos fotogramas en el tiempo.

2.2.12 Algoritmo K-means

K-means es un algoritmo de clasificación no supervisada que agrupa objetos en k grupos (clúster) basándose en sus características. El agrupamiento se realiza minimizando la suma de distancias entre cada objeto con el centroide de su grupo o clúster.

Peña et al. (1999) realizan una comparación de 4 métodos de inicialización para el algoritmo K-Means y asevera que de acuerdo a la revisión de la literatura el rendimiento depende de dos puntos clave: la agrupación inicial y el orden de las instancias. Los resultados obtenidos ilustran que los métodos de inicialización aleatorios y de Kaufman superan al resto de los métodos comparados, ya que hacen que las K-Means sean más efectivas y más independientes en la agrupación inicial y en el orden de las instancias.

Criterio de Error Cuadrático. La ecuación 1, representa a la función F y la ecuación 2 a la definición de centroide.

$$F(\{C_1, \dots, C_k\}) = \sum_{i=1}^k \sum_{j=1}^{k_i} \|w_{ij} - \bar{w}_i\|, \quad (1)$$

Donde:

K : es el número de clúster

K_i : es el número de objetos del clúster i

w_{ij} : es el j -ésimo objeto del i -ésimo clúster

\bar{w}_i : es el centroide del i-ésimo clúster el cual está definido como:

$$\bar{w}_i = \frac{1}{K_i} \sum_{j=1}^{k_i} w_{ij}, i = 1, \dots, k \quad (2)$$

Algoritmo 2.3: Algoritmo K-Means

Inicio

Paso 1: Inicialización: una vez escogido el número de grupos, k, establecer k centroides en el espacio de los datos.
//escogiéndolos de manera aleatoria.

Hacer:

Paso 2: Asignación de objetos a los centroides:

Cada objeto de los datos es asignado a su centroide más cercano.

Paso 3: Actualización centroides:

Actualizar la posición del centroide de cada grupo.

//Nuevo centroide es la posición del promedio de los objetos pertenecientes a dicho grupo.

Hasta:

Hasta que los centroides ya no se muevan, o se muevan por debajo de una distancia umbral en cada paso.

Fin

Figura 3

Peores valores de error cuadrático en 10 ejecuciones

	Iris		Ruspini		Glass		
	K = 3	K = 4	K = 4	K = 5	K = 2	K = 7	K = 10
RANDOM	78.95	71.66	36,160.5	35,242.69	840.19	571.6	321.31
FA	145.23	144.1	36,160.6	35,311.39	1243.83	590.91	504.41
MA	145.23	144.1	36,160.19	35,311.38	1243.83	590.38	503.73
KA	78.94	67.47	36,160.21	8032.4	840.19	375.78	287.02

Nota: RANDOM del método de Kaufman (KA), son los valores que menor error cuadrático presentan, para la Base de datos Iris, Ruspini y Glass donde k es el número de clústeres. Figura tomada de Peña et al. (1999).

De acuerdo a la Figura 3 se llega a la conclusión que los métodos RANDOM y KA son métodos eficaces y robustos para el algoritmo K-Means. Además, KA muestra un comportamiento más deseable, ya que hizo que el

algoritmo K-Means no pudiera alcanzar particiones tan malas como el resto de métodos comparados. Respecto a la velocidad KA tiene mejor capacidad para inducir al algoritmo K-Means un comportamiento más deseable que RANDOM.

2.2.13 Método de Desplazamiento Meanshift

Basado en el Algoritmo de K-means de Fukunaga y Hosteller. se tiene una función $N(x)$ para determinar cuáles son los vecinos de un punto $x \in X$. Los puntos vecinos son los puntos dentro de una cierta distancia. La métrica de la distancia suele ser la Distancia Euclidiana. Se tiene una función kernel $K(d)$ para usar en Meanshift. K suele ser un núcleo gaussiano y d es la distancia entre dos puntos de datos.

La ecuación 3 permite hallar el desplazamiento medio $\mathbf{m}(x)$ y se usará en el algoritmo Meanshift para un conjunto de puntos de datos X .

$$\overrightarrow{\mathbf{m}(x)} = \frac{\sum_{x_i} K(x_i - x) \overrightarrow{x_i}}{\sum_{x_i} K(x_i - x)}, x_i \in N(x) \quad (3)$$

Algoritmo 2.4: Algoritmo K-Means

Inicio

Para cada punto de datos $x \in X$, encuentre los puntos vecinos $N(x)$ de x .

Para cada punto de datos $x \in X$, calcule el desplazamiento medio $\mathbf{m}(x)$ con la ecuación (3).

(1) Para cada punto de datos $x \in X$, actualice $x \leftarrow \mathbf{m}(x)$.

(2) Repita (1). para $n_{iteraciones}$ o hasta que los puntos casi no se muevan o no se muevan.

Fin

Lo importante es calcular el desplazamiento medio $\mathbf{m}(x)$. Luego hacemos $\sum_{x_i \in N(x)} K(x_i - x)$ que es la sumatoria de W_i , por lo que la fórmula se convierte en la ecuación (4)

$$m(x) = \frac{\sum_i w_i x_i}{\sum_i w_i} \quad (4)$$

Por otro lado, formalmente la **media aritmética ponderada**, tiene la expresión de la ecuación (5):

$$\bar{x} = \frac{w_1 x_1 + w_2 x_2 + \dots + w_n x_n}{w_1 + w_2 + \dots + w_n} \quad (5)$$

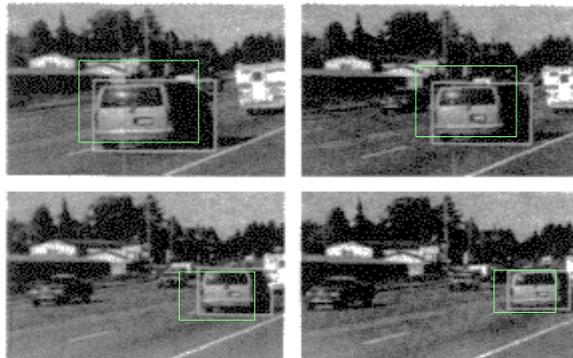
Donde $\{x_1, x_2, \dots, x_n\}$ es un conjunto múltiple de datos no vacíos y $\{w_1, w_2, \dots, w_n\}$ sus correspondientes pesos no negativos, de manera que tenemos la ecuación (6).

$$\bar{x} = \frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i} \quad (6)$$

Un ejemplo del uso de este algoritmo se tiene en la Figura 4 donde se hace un seguimiento a un objeto durante marcos diferentes y realiza el seguimiento basado en el método de MeanShift mediante el uso de filtros de Kalman para filtrar el objeto y realizarlo de manera continua (Ningson et al., 2005).

Figura 4

Secuencia de tráfico con el algoritmo MeanShift



Nota: Seguimiento de marcos. Figura tomada de Ningson et al. (2005)

2.2.14 Método de flujo óptico

Este método, a diferencia de extraer características de desplazamiento de puntos como el de MeanShift, utiliza una técnica basada en las variaciones de brillo de la imagen en el espacio y tiempo. La idea es hacer el seguimiento a través de los fotogramas considerando lo siguiente:

- 1.- Si se tiene una región pequeña el brillo debe permanecer constante, aunque la ubicación de la región cambie.
- 2.- Los puntos vecinos en escena típicamente llegan a pertenecer a la misma superficie, de manera que los movimientos típicamente llegan a ser los mismos.
- 3.- El movimiento de un parche realiza cambios graduales.
- 4.- Los puntos no se mueven muy lejos ni al azar.

Considere un píxel de intensidad $I(x, y, t)$ en el primer fotograma: Notar que aquí se tiene una nueva dimensión, el tiempo t . Evidentemente en los métodos anteriores que hasta ahora se ha mostrado (en este marco teórico) se ha trabajado solo con imágenes, por lo que no era necesario el tiempo.

Este píxel con $I(x, y, t)$ se mueve por distancia (dx, dy) en el siguiente fotograma tomado después del tiempo dt . Entonces, dado que esos píxeles son los mismos y la intensidad no cambia, se puede expresar la ecuación (7).

$$I(x, y, t) = I(x + dx, y + dy + t + dt) \quad (7)$$

Resolviendo la Serie de Taylor y sus derivadas respectivas, se tiene:

$$\frac{\partial f}{\partial x} \frac{dx}{dt} + \frac{\partial f}{\partial y} \frac{dy}{dt} + \frac{\partial f}{\partial t} \frac{dt}{dt} = 0$$

La ecuación tomaría la forma:

$$f_x u + f_y v + f_t = 0 \quad (8)$$

Interpretando la ecuación (8), el flujo óptico es el patrón de movimiento de los objetos de la imagen entre dos fotogramas consecutivos. Obviamente, causado por el movimiento del objeto o el movimiento de una cámara. Matemáticamente es un campo vectorial 2D donde cada vector es un vector de desplazamiento que muestra el movimiento de puntos desde el primer fotograma al segundo fotograma en un intervalo de tiempo.

Luego, se realiza un ajuste ponderado por mínimos cuadrados de la ecuación de flujo óptico a un modelo constante para (V_x, V_y) en cada sección.

El método consigue este ajuste minimizando la ecuación 9, en la que w es una función ventana que enfatiza el flujo óptico en el centro de cada sección (Xing et al., 2021)

$$\sum w^2 I_x v_x + I_y v_y + I_t^2 = 0 \quad (9)$$

La solución al problema de minimización se muestra en la ecuación 10

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} \sum w I_x^2 & \sum w I_x I_y \\ \sum w I_x I_y & \sum w I_y^2 \end{bmatrix}^{-1} \begin{bmatrix} -w I_x I_t \\ -w I_y I_t \end{bmatrix} \quad (10)$$

Donde I_x, I_y e I_t corresponden a los valores de gradiente de píxeles de la imagen en las direcciones x, y, t . Los detalles de la derivación se han desarrollado en la ecuación 7.

2.2.15 Método de flujo óptico de Lucas_Kanade

Un tipo de iteración que se utiliza en la técnica de registro de imágenes con gradiente de intensidad espacial y que permite encontrar una buena coincidencia es la iteración numérica de Newton-Raphson (Lucas & Kanade, 1981).

El método de Lucas-Kanade aplica bajo la consideración o supuesto que todos los píxeles vecinos tendrán un movimiento similar. Es un método diferencial para el cálculo del movimiento. Si se cumple el supuesto de que la vecindad local tiene un movimiento similar, entonces el flujo será constante. Por tanto, para el seguimiento de objetos aplica la ecuación de flujo óptico (ecuación 8) mejorado con la técnica de Lucas-Kanade en los criterios de vecindad de píxeles y el criterio de mínimos cuadrados que hace que el método sea menos sensible a ruido (Yedjour, 2021).

El método propuesto por Lucas-Kanade ofrece una solución al resolver un sistema 2x2 de ecuaciones lineales (ecuación 8) usando el principio de mínimos cuadrados. Una solución obtenida con el método de ajuste de mínimos cuadrados da los vectores de velocidad u y v donde f_x y f_y son gradientes de imagen y f_t es el gradiente a lo largo del tiempo. La información de movimiento obtenida en los vectores de flujo óptico resultantes u y v de la ecuación 8 permite medir las velocidades y las direcciones de movimiento percibido en cada nivel de píxel entre dos imágenes sucesivas (Yedjour, 2021)

Respecto a la estimación de movimiento: para obtener una mejor visualización, se procede a la codificación por colores del campo de flujo basada en el flujo máximo presente. El flujo óptico calculado se convierte en coordenadas polares h y q , donde h representa la coordenada angular y q representa la coordenada radial. Luego, se utiliza el color RGB normalizado a fin de obtener una imagen de flujo óptico codificada con colores RGB.

Las coordenadas polares angular y radial se calculan mediante las ecuaciones 11 y 12 respectivamente

$$\theta = \arctan2(u, v) \quad (11)$$

$$\rho = \sqrt{u^2 + v^2} \quad (12)$$

Respecto a la segmentación de flujo: el resultado de estimación del movimiento puede contener movimientos erróneos debido a factores como la iluminación y el ruido de fondo. Para resaltar los objetos de interés, es necesario segmentar el resultado de estimación de movimiento utilizando un umbral **th**. El cálculo del umbral **th** para cada píxel en la posición **(i, j)** debe ser un parámetro dinámico y viene dado por la ecuación 13.

$$th = \sqrt{u(i,j)^2 + v(i,j)^2} \quad (13)$$

El valor de intensidad de cada píxel del resultado de movimiento estimado se compara con **th** y la separación de los objetos en movimiento se consigue mediante la siguiente condición de segmentación:

2.2.16 El filtro de Kalman

Este filtro es muy usado en visión por computadora porque involucra predicción. La idea central de un filtro de Kalman es utilizar las detecciones disponibles y las predicciones previas para llegar a la mejor suposición del estado final o actual mientras se mantiene la posibilidad de errores en el proceso.

Por ejemplo, podemos utilizar un método de flujo óptico, pero en ocasiones se pierde precisión o detecciones de un porcentaje de los fotogramas. Si se desea predecir el próximo estado de un objeto, teniendo un "modelo de velocidad constante", aplicando ecuaciones de la mecánica clásica podríamos predecir dónde estará el objeto en el siguiente fotograma. Todo ello suena ideal, pero en el mundo real hay otros componentes a considerar como el ruido y si se tiene ruidos de proceso o medición ya no se puede esperar velocidad constante. Entonces, básicamente se reduce a inferir una nueva distribución (las predicciones) a partir de la distribución de estado anterior y la distribución de medición. El filtro de Kalman, tiene una fuerte aplicación a procesos de desplazamiento lineales, el desarrollo matemático se puede localizar en (Thacker & Lacey, 1998).

2.2.17 Redes de regresión profunda

Uno de los mayores avances en el tema de seguimiento es el uso de CNN con regresión. El proceso es que una red profunda aplica un rastreador que aprende el movimiento de objetos genéricos y se puede usar para rastrear objetos nuevos que no aparecen en el conjunto de entrenamiento. Lo interesante es que aprende a rastrear objetos genéricos en tiempo real mientras se mueven.

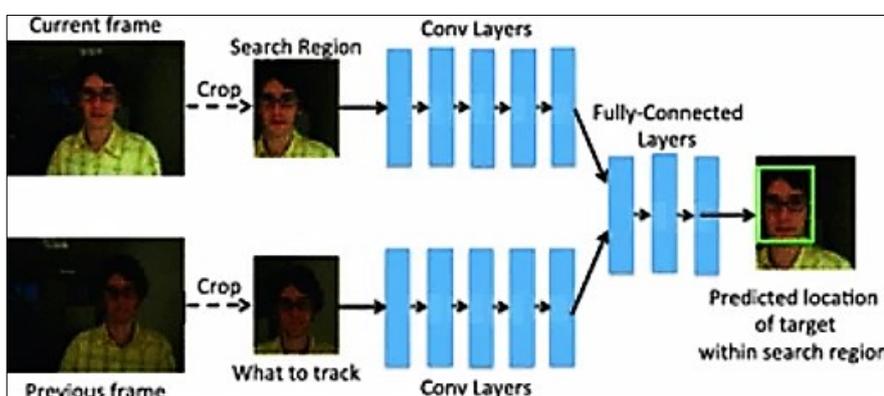
Held et al. (2016) proponen una arquitectura de red simple con CNN seguida de capas completamente conectadas que dan directamente las coordenadas del cuadro delimitador. El método sugiere introducir fotogramas de un vídeo en una red neuronal, y la red emite sucesivamente la ubicación del objeto rastreado en cada fotograma.

En caso de que haya varios objetos en el vídeo, la red debe recibir información sobre ¿qué objeto del vídeo se está siguiendo? Para ello, se introduce en la red una imagen objetivo. Luego se recorta y se escala el fotograma anterior para centrarlo en el objetivo.

Como se muestra en la Figura 5. esta entrada permite a nuestra red seguir objetos nuevos que no ha visto antes; la red seguirá cualquier objeto que se introduzca en este recorte.

Figura 5

Arquitectura de una red de seguimiento con regresión profunda



Nota: En la red hay una región de búsqueda del fotograma actual y un objetivo del fotograma anterior. Figura tomada de (Held et al., 2016).

La función de las capas completamente conectadas es comparar las características del objeto de destino con las características del marco actual para encontrar dónde se ha movido el objeto de destino (Held et al., 2016).

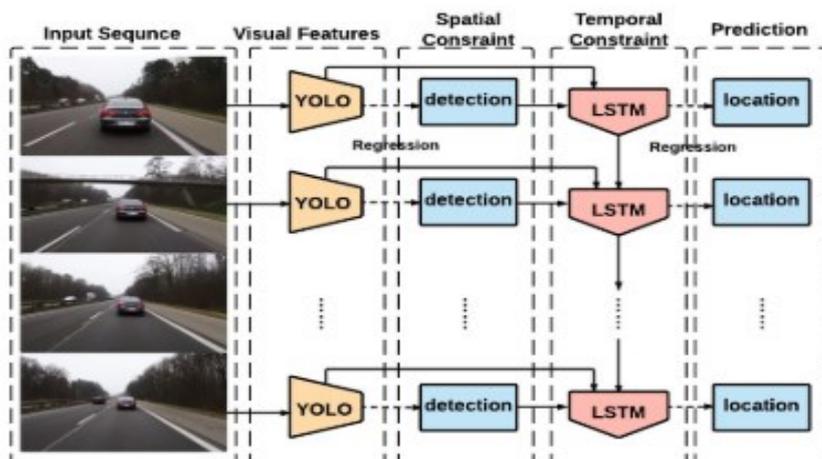
Por otro lado, es necesario recalcar que, entre estos marcos, el objeto puede haber sufrido una traslación, rotación, cambio de iluminación, oclusión o deformación. La función aprendida por las capas completamente conectadas hace una comparación de características complejas que aprende a través de muchos ejemplos (Ning, 2017).

2.2.18 Método ROLO

Una forma de hacer seguimiento de objetos es utilizando CNN profundo. Se diría que ROLO es un YOLO ligeramente modificado con una unidad Long Short Term Memory (LSTM) recurrente adjunta al final, que permite rastrear objetos al capturar las características espacio temporales

Figura 6

Arquitectura del método ROLO



Nota: Visión general simplificada del sistema y del procedimiento de seguimiento ROLO. Figura tomada de Ning (2017).

El sistema detector de objetos está basado en regresión, en el que presenta un sistema de redes neuronales para procesar información espacio-temporal bajo la ecuación 14 y luego inferir ubicaciones de regiones.

$$p(B_1, B_2, \dots, B_T | X_1, X_2, \dots, X_T) = \prod_{t=1}^T p(B_t | B_{<t}, X_{\leq t}) \quad (14)$$

En síntesis, ROLO extiende YOLO (una CNN profunda) en el dominio espacio-temporal utilizando redes neuronales recurrentes. Específicamente, usa YOLO para recopilar características visuales ricas y robustas, así como inferencias de ubicación preliminares; y usa LSTM en la siguiente etapa, ya que es espacialmente profundo y apropiado para el procesamiento de secuencias.

Hay tres fases para el entrenamiento de un extremo a otro del modelo ROLO: la **fase de entrenamiento previo** de capas convolucionales para el aprendizaje de características, la **fase de entrenamiento tradicional de YOLO** para la propuesta de objetos y la **fase de entrenamiento LSTM** para el seguimiento de objetos.

Las RNN convencionales no pueden acceder al contexto de largo alcance debido al error propagado hacia atrás, lo que se denomina problema del gradiente de fuga. Por el contrario, LSTM RNN son capaces de modelar información de contexto auto aprendida. LSTM tiene una celda de memoria c_t que esencialmente actúa como un acumulador de la información de estado. Cada vez que llega una nueva entrada, su información se acumulará en la celda si la puerta de entrada i_t está activada. El estado anterior de la celda c_{t-1} podría “olvidarse” en este proceso si la puerta de olvido f_t está activada. Si la última salida de celda c_t se propaga al estado final h_t se controla mediante la puerta de salida O_t . En el sistema, se utiliza la unidad LSTM como módulo de seguimiento (Ning, 2017).

2.2.19 Método YOLO

Los métodos que anteceden a You Only Look Once (YOLO), para detección de objetos usan R-CNN o Fast R-CNN con la propuesta de región para generar posibles recuadros delimitadores en una imagen (ver Figura 7), luego ejecutan un clasificador en dichos recuadros.

Después de la clasificación se usa el posprocesamiento para definir los recuadros delimitadores, eliminar detecciones duplicadas y establecer un score a los recuadros en función de otros objetos de la imagen.

Estos procesos, aunque son robustos y precisos son complejos procesos caracterizados por ser lentos y difícilmente optimizables porque cada componente debe entrenarse por separado (Redmon et al., 2016). YOLO plantea la detección de objetos como un único problema de regresión, directamente de los píxeles de la imagen a las coordenadas de la caja delimitadora y las probabilidades de clase, para ello usa solo una CNN que predice recuadros delimitadores múltiples y probabilidades de clase para esos recuadros.

Ventajas de YOLO

Rapidez: dado que no requiere de procesos complejos solo usa regresión con la red neuronal y predice las detecciones. Funciona de 45 fps en una GPU Titan X, hasta 150 fps, lo cual tranquilamente puede procesar videos en tiempo real a una latencia menor a 25 ms. A diferencia de los métodos que usan ventanas deslizantes y en propuestas de regiones, YOLO capta la imagen como un todo durante entrenamiento y prueba, por lo que codifica contexto de las clases implícitamente.

Si se compara con Fast R-CNN (método que no puede ver el contexto ampliamente) YOLO arroja menos del 50% de errores de fondo que Fast R-CNN. Supera ampliamente a R-CNN cuando aprende representaciones generalizables de objetos, por ello si se aplica a nuevos dominios o entradas inesperadas YOLO responderá mejor.

Desventajas

YOLO en cuanto a precisión está detrás de los sistemas más avanzados. Identifica rápidamente objetos en las imágenes, pero presenta dificultades para localizar con precisión objetos pequeños. El sistema Yolo divide a la imagen de entrada en cuadrículas de tamaño $S \times S$. Si el centro de un objeto cae en una celda de la cuadrícula. La celda se encargará de

detectar el objeto. Luego cada celda predice B recuadros delimitadores y puntuaciones de confianza para esos recuadros.

Formalmente se define a la confianza como el producto de la probabilidad de que un objeto sea detectado en una región o celda específica de una imagen multiplicado por la intersección sobre unión (IOU) real y de predicción (Redmon et al., 2016).

$$Confidence = Pr(Object) * IOU_{\substack{truth \\ pred}} \quad (15)$$

Si existe un objeto en la celda se desea que la puntuación de confianza sea igual a la intersección sobre la unión (IOU) entre el cuadro predicho y la verdad fundamental.

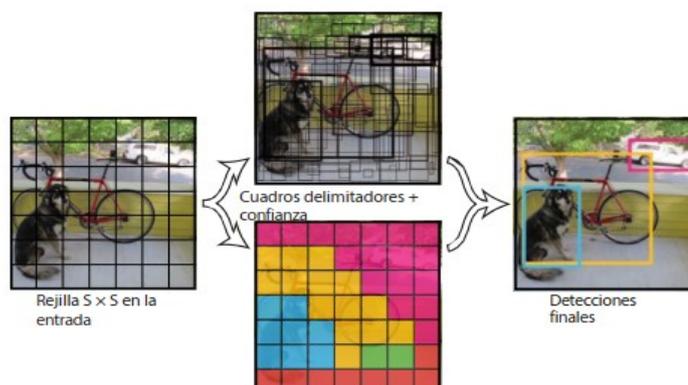
Si no existe objeto en la celda, las puntuaciones de confianza deben ser cero.

Cada cuadro delimitador consta de 5 predicciones: x , y , w , h y **confianza**. Las coordenadas (x, y) representan el centro del cuadro en relación con los límites de la celda de la cuadrícula. El ancho w y el alto h se predicen en relación con toda la imagen.

Finalmente, la predicción de confianza representa el IOU entre el recuadro predicho y cualquier recuadro real. La celda de la cuadrícula también predice C probabilidades de clase condicionales $Pr(Clase_i, Objeto)$

Figura 7

Sistema de detección YOLO



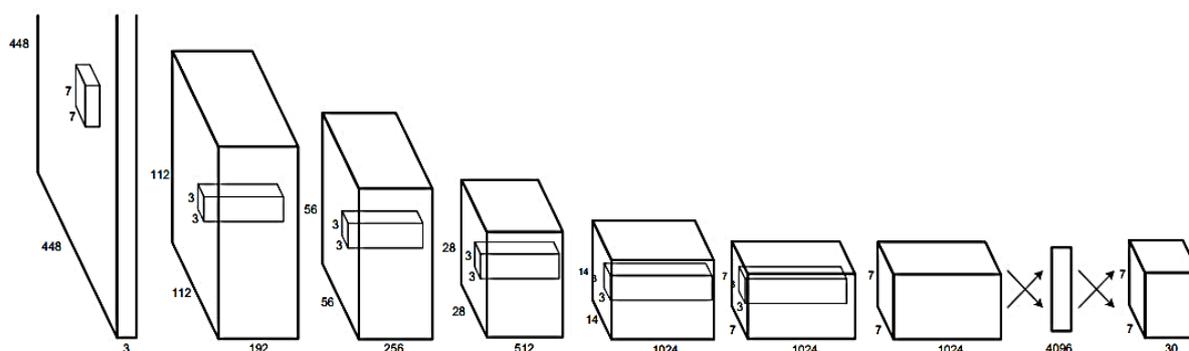
Nota: el sistema divide la imagen en una cuadrícula $S \times S$ y para cada celda de la cuadrícula predice cuadros delimitadores B , confianza para dichos cuadros y probabilidades de clase C . Estas predicciones están codificadas como un tensor $S \times S \times (B * 5 + C)$.

Arquitectura de red:

respecto a la arquitectura de red YOLO usa 24 capas convolucionales seguidas de 2 capas totalmente conectadas, usa capas de reducción de 1x1 seguidas de capas convolucionales de 3x3 (ver Figura 8).

Figura 8

Arquitectura de red YOLO



Nota: La red de detección tiene 24 capas convolucionales seguidas de 2 capas de conexión (Redmon et al., 2016).

Observar que el resultado final es el tensor de 7x7x30 imagen de entrada de 224 × 224 y luego se duplica la resolución para la detección

Finalmente, YOLO como modelo unificado para la detección de objetos es simple de construir y puede entrenarse directamente en imágenes completas. A diferencia de los enfoques basados en clasificadores. Por otro lado, YOLO impulsa el estado del arte en la detección de objetos en tiempo real y se generaliza bien a nuevos dominios, lo que lo hace ideal para aplicaciones de detección de objetos rápida y robusta.

2.2.20 Los semáforos

Según el Ministerio de Transportes y Comunicaciones (2016) define a los semáforos como dispositivos de control del tránsito cuya finalidad es regular y controlar el tránsito vehicular motorizado y no motorizado, y claro está también el peatonal, a través de las indicaciones de luces de color que oficialmente son: rojo, verde y amarillo o ámbar.

Estación Central o Control Maestro

El Ministerio de Transportes y Comunicaciones (MTC) menciona que son centros de maniobras que tienen por finalidad optimizar los movimientos del tránsito vehicular y además contribuyen en la solución de los problemas de congestión.

Condiciones para verificar la necesidad de implementación semafórica

1) Volumen vehicular para 8 horas, 2) Volumen vehicular para 4 horas, 3) Volumen vehicular para horas punta, 4) Volumen peatonal, 5) Movimiento o circulación progresiva, 6) Accidentes frecuentes, 7) Red vial, 8) Intersecciones cercanas a pasos de ferrovías 9) Combinación de las condiciones anteriores (Ministerio de Transportes y Comunicaciones, 2018).

Clasificación de los semáforos

Semáforos para vehículos: que pueden ser fijos o pre sincronizados, sincronizados por el tránsito y semáforos adaptados al tránsito (semáforos inteligentes).

Según el (Ministerio de Transportes y Comunicaciones, 2018), respecto a Semáforos Inteligentes define:

Son aquellos cuyo funcionamiento es ajustado continua y automáticamente en todos los accesos a una intersección, de acuerdo a la información sobre el flujo vehicular que coleccionan los detectores de tráfico y envían la información sobre la secuencia de fases, intervalos de fases, ciclos y/o desfases, a una Estación Central o Control Maestro (pág. 385)

Semáforos para peatones, que se complementan con ondas sonoras para personas con limitación visual.

Semáforos Especiales: con destellos intermitentes, para advertir situaciones especiales, obstrucciones, instrucciones como “CEDA EL PASO”, “PARE”, semáforos para regular el uso de carriles, semáforos para paso de vehículos de emergencia, semáforos para indicar la aproximación de trenes y semáforos para regular el uso de carriles de peaje.

2.2.21 Precisión en la clasificación de objetos

En un estudio en el que se compara la máquina de soporte de vectores (SVM) y redes neuronales convolucionales (CNN) encontró que cuando se utiliza un gran conjunto de datos *mnist* de muestra, la precisión de SVM es 0,88 y la precisión de CNN es 0,98; cuando se utiliza una muestra pequeña *COREL1000* conjunto de datos, la precisión de SVM es 0,86 y la precisión de CNN es 0,83. De manera que los resultados experimentales en dicho estudio muestran que el aprendizaje automático tradicional tiene un mejor efecto de solución en conjuntos de datos de muestra pequeños, y el marco de aprendizaje profundo tiene una mayor precisión de reconocimiento en grandes conjuntos de datos de muestra (Wang et al., 2021).

Empresas como Amazon AWS Rekognition, Google Vision, IBM Watson y Microsoft Azure Computer Vision, están a la vanguardia en el tema de investigación en reconocimiento de imágenes y manejan niveles de precisión de 77,7%, 81,7%, 55,6%, 75,8% (Perficient, 2019).

En ese sentido tenemos que la Precisión Media, entre los frameworks de reconocimiento mencionado es de 72,7%. En la Figura 9, tenemos que

las mAP (medida de la precisión media) en investigaciones de alto desempeño como es el caso de Microsoft Reserach, se encuentran desde 76% a 79%, pues claro está que eso también dependiendo del GPU a utilizar (Dai et al., 2016).

Figura 9

Comparación de medidas de precisión media de R-CNNs

Table 3: Comparisons between Faster R-CNN and R-FCN using ResNet-101. Timing is evaluated on a single Nvidia K40 GPU. With OHEM, N RoIs per image are computed in the forward pass, and 128 samples are selected for backpropagation. 300 RoIs are used for testing following [18].

	depth of per-RoI subnetwork	training w/ OHEM?	train time (sec/img)	test time (sec/img)	mAP (%) on VOC07
Faster R-CNN	10		1.2	0.42	76.4
R-FCN	0		0.45	0.17	76.6
Faster R-CNN	10	✓ (300 RoIs)	1.5	0.42	79.3
R-FCN	0	✓ (300 RoIs)	0.45	0.17	79.5
Faster R-CNN	10	✓ (2000 RoIs)	2.9	0.42	N/A
R-FCN	0	✓ (2000 RoIs)	0.46	0.17	79.3

Nota: Imagen tomada de (Dai et al., 2016)

2.2.22 Cálculo del MAPE, MAE Y RMSE

El error medio porcentual absoluto (MAPE), el error cuadrático medio (RMSE) y el error absoluto medio (MAE) son tres medidas usadas para evaluar el rendimiento de predicción multipaso de distintos modelos, se utilizan MAE, MAPE cuyas formulas son las expresadas en la ecuación 16, 17 y 18 respectivamente (Tang et al., 2017).

$$MAE = \frac{\sum_{i=1}^N |\hat{S}_i - S_i|}{N} \quad (16)$$

$$MAPE = \frac{1}{N} \sum_{i=1}^N \frac{|\hat{S}_i - S_i|}{S_i} \times 100\% \quad (17)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (\hat{S}_i - S_i)^2}{N}} \quad (18)$$

Donde:

N : es el número de observaciones,

S_i : es el valor real en el momento i

\hat{S}_i : es el valor previsto.

En general el error cuadrático medio mide la cantidad de error que hay entre dos clústeres o conjuntos de datos, comparando un valor de predicción y un valor real, observado o conocido. Si RMSE es pequeño, más cercanos son los valores de predicción y los valores observados.

2.3 Marco conceptual

2.3.1 Glosario de términos por categoría

Categoría tráfico vehicular:

Intersecciones: son lugares donde confluyen vehículos y son de vital importancia para la eficiencia funcional del sistema vial urbano, ya sea por la jerarquía de las vías que las conforman, y/o por los flujos vehiculares que transitan por ellas.

Carril: es el ancho de división de una vía, generalmente paralelo a su división vecino. Una vía puede dividirse en varios carriles, cada uno de los cuales puede ser utilizado por una línea de vehículos separados entre sí. Por lo que el ancho de la carretera siempre determinará el número de carriles.

Señal de movimiento: este es un semáforo que indica el flujo de tráfico.

Fase: es el orden en que se configuran los semáforos para permitir el flujo de tráfico por la intersección durante un tiempo específico. La fase también limita el volumen de vehículos que se mueven por la intersección a la vez.

Duración del ciclo: tiempo en que todas las fases de una red están en verde.

Flujo vehicular: Es el número de vehículos que pasan por una posición en un periodo de tiempo determinado. Se expresa en número de vehículos por hora (Veh/h).

Factor de hora pico: es el volumen total de vehículos entre la tasa de flujo máxima (pico) obtenida en la hora pico.

Categoría algoritmos:

Algoritmo de evolución diferencial: (DE, por sus siglas en inglés, Differential Evolution) es una técnica de optimización estocástica utilizada para encontrar soluciones aproximadas a problemas de optimización continua y combinatoria. Ha demostrado ser eficaz en una variedad de aplicaciones (Storn & Price, 1997).g

Convergencia: se refiere a la propiedad de que el algoritmo tiende a acercarse o converger hacia una solución óptima o estable a medida que avanza en su proceso de búsqueda o iteración. Si el algoritmo se ejecuta y se acerca a una solución de alta calidad, se dice que ha convergido. La velocidad y la calidad de la convergencia pueden variar según el algoritmo y el problema en cuestión.

Convergencia global: significa que el algoritmo tiende a acercarse a la mejor solución posible en todo el espacio de búsqueda sin importar dónde se encuentre en el espacio de búsqueda inicial.

Convergencia local: significa que el algoritmo tiende a acercarse a una solución óptima dentro de un área específica del espacio de búsqueda. Se puede dar el caso que encontró una solución de alta calidad, pero no necesariamente la mejor solución global.

Intersection Over Union (IOU) es una métrica comúnmente utilizada en tareas de detección de objetos y segmentación semántica en visión por computadora. La métrica IOU se calcula mediante la superposición de dos regiones: la región predicha por un algoritmo de detección (por ejemplo, una caja delimitadora o un área segmentada) y la región real o verdadera (anotada por un humano). La fórmula para calcular el IOU es la siguiente:

$$IOU = \frac{\text{área de intersección}}{\text{área de unión}} \quad (19)$$

Donde:

Área de Intersección: Es la cantidad de píxeles o la superficie que se superponen entre la región predicha y la región real.

Área de Unión: Es la cantidad total de píxeles o la superficie combinada de ambas regiones.

El resultado de esta división es un valor entre 0 y 1, donde 0 indica una falta de superposición (ninguna coincidencia) entre la predicción y la anotación, y 1 indica una superposición completa y perfecta.

Categoría lógica difusa

Conjuntos difusos: se intenta modelar la ambigüedad con la que se percibe una variable. Son la base para la lógica difusa, así como la teoría clásica de conjuntos es la base para la lógica Booleana. En este campo se realizan afirmaciones lógicas del tipo si-entonces. Los conjuntos difusos definen se caracterizan por manejar ambigüedades.

Función de membresía: Las funciones de membresía representan el grado de pertenencia de un elemento a un subconjunto difuso. Las funciones de membresía más comunes son del tipo trapezoidal, S, triangular y singleton.

Incertidumbre artificial: se refiere a la falta de certeza o precisión en los datos, modelos o decisiones generadas por sistemas de inteligencia artificial. Esta incertidumbre puede surgir por varias razones, incluyendo:

Incertidumbre en modelo: modelos como las redes neuronales, pueden producir predicciones con cierto grado de incertidumbre. Esto puede manifestarse como una distribución de probabilidad en lugar de una predicción única.

Incertidumbre aleatoria: es la incertidumbre inherente a los procesos estocásticos o aleatorios en los datos o en el proceso de modelado.

Datos imprecisos: cuando los datos de entrada son incompletos, ruidosos o ambiguos, lo que hace que sea difícil para un sistema de inteligencia artificial tomar decisiones definitivas.

2.3.2 Determinación de la densidad del tráfico

La densidad de tráfico es el número de vehículos que hay en un tramo de carretera en un momento específico en función de la velocidad e intensidad. Por tanto, una densidad máxima (D_{max}) se presenta cuando los vehículos en el tramo o sección están detenidos.

$$D_{max} = \frac{\bar{ql}}{c} \quad (20)$$

Donde:

D_{max} : densidad máxima

ql : media de longitud de vehículos

c : número de carriles

En realidad, hay variables más complejas de medir como: la inusitada reacción de los conductores, la velocidad de cada vehículo, número de maniobras realizadas, número de paradas, número de arranques, dimensión de separación entre vehículos delantero y posterior y laterales. Por ahora, consideraremos el aspecto teórico descrito en la ecuación 20, para hallar la densidad máxima.

2.3.3 Determinación de la intensidad del tráfico

La intensidad del tráfico I es el número de vehículos que pasa en un tramo específico en un tiempo específico.

$$I = \frac{n}{t} \quad (21)$$

Donde:

I : intensidad del tráfico

n : total de vehículos que pasa por un tramo

t : unidad de tiempo

Por ejemplo, tendríamos vehículos / hora.

Otro aspecto a considerar en cuanto a la intensidad es que esto da origen a la denominación de intensidad de hora punta, donde ocurre el mayor tráfico. Las fiestas de la localidad, eventos, manifestaciones de la población entre otros aspectos también inciden en la intensidad, pero con menos frecuencia de ocurrencia.

Medina, G. y Ulloa, N. (2015), nos presentan unos resultados del volumen vehicular en las horas punta en la intersección Av., Pacífico Plaza Mayor, al cual lo denominaron Estación Nro. 1 cuando la dirección de la ruta es de sur a norte y Estación Nro. 2, cuando la dirección del transporte es de norte a sur.

Así se tiene: **Estación 1 (sur a norte)**: de 7:00h a 8:00h, 1102 vehículos /hora; de 13:00h a 14:00 h, 799 vehículos /hora; de 19:00h a 20:00 h; 1019 vehículos /hora, con un índice medio diario de 11824 vehículos/día. **Estación 2 de norte a sur**: de 7:00h a 8:00h, 934 vehículos /hora; de 13:00h a 14:00h, 1036 vehículos /hora; de 19:00h a 20:00h, 1101 vehículos /hora, con un índice medio diario de 11942 vehículos/día. Estos datos fueron obtenidos con conteo manual y son de importancia para la implementación de proyectos de semaforización, ver el Anexo A para ubicar geográficamente la intersección Av. Pacífico Plaza Mayor.

2.3.4 Determinación de la velocidad del tráfico

La velocidad es el espacio recorrido y el tiempo empleado en recorrerlo. Se expresa en km/h, o en metros/s.

$$v = \frac{d}{t} \quad (22)$$

Donde;

v : velocidad del vehículo

d : es el tramo de recorrido en metros

t : es la unidad de tiempo en segundos

CAPÍTULO III

MATERIALES Y MÉTODOS

3.1 Enfoque epistemológico

El enfoque epistemológico de la investigación cuantitativa se centra en la recopilación y el análisis de datos numéricos para obtener conocimiento objetivo y generalizable sobre fenómenos particulares, lo que permite establecer relaciones causales, hacer predicciones y tomar decisiones informadas.

A partir de lo observado, se logra obtener datos los cuales, a través de la inteligencia, la razón natural, la comprensión y el entendimiento humano, logran tener significado. Pero los datos no tienen un significado intrínseco, sino que es el ser humano quien le otorga significado y validez científica.

Para el caso de la presente investigación la observación y experiencia empírica directa del tráfico vehicular en diferentes contextos y condiciones proporciona una base fundamental para comprender la dinámica del tráfico y sienta las bases del inicio del proceso experimental. Luego, se hace necesario el análisis de datos de tráfico en múltiples escenarios, por lo que se debe recurrir al modelado de posibles escenarios.

La epistemología de la investigación cuantitativa requiere de métodos numéricos y un análisis estadístico que permita en base a los resultados tomar una postura teórica y el supuesto teórico sea válido (Hernández et al., 2014).

Por tanto, un enfoque epistemológico de aplicación del conocimiento en ingeniería relacionado con la investigación del tráfico vehicular nos conlleva a reforzar la postura de la investigación aplicada, pues de observación empírica, pasamos a un modelado teórico, luego se realiza la experimentación, análisis estadístico y síntesis para comprender y abordar la realidad del tráfico vehicular.

Además, en esta sección se describe a los materiales, procedimientos realizados en la investigación a fin de que sea posible ser replicado. En ese sentido el presente marco metodológico comprende dos partes importantes: una primera parte correspondiente al diseño de la investigación (hipótesis, ámbito del estudio, variables, diseño, población y muestra); y los métodos utilizados tanto para el trabajo observacional de campo como de laboratorio en correspondencia con la primera parte.

3.1.1 Matriz de consistencia

Tabla 1

Matriz de consistencia problema, objetivo, hipótesis y variables

Formulación del problema	Objetivos	Hipótesis	Variables
<p>Problema general: ¿En qué medida un sistema inteligente basado en un modelo heurístico, mediante la gestión en tiempo real de un semáforo, reducirá los tiempos de espera de vehículos en una intersección vial?</p>	<p>Objetivo general Reducir los tiempos de espera de vehículos en una intersección vial a través de un sistema inteligente basado en un modelo heurístico.</p>		<p>Variable Independiente Variable X: Sistema inteligente basado en un modelo heurístico con gestión en tiempo real de un semáforo</p>
<p>Problemas específicos (PE)</p> <p>PE1: ¿Cuáles son las características de tráfico en una intersección vial?</p> <p>PE2: ¿Existe algún modelo reconocimiento y conteo vehicular eficiente en Tiempo real?</p> <p>PE3: ¿Cómo se obtendrá automáticamente las características del tramo vehicular ?</p> <p>PE4: ¿Cómo se realizará la asignación de tiempos de colores de semáforo?</p> <p>PE5. ¿Cómo se va a comparar los resultados del funcionamiento del sistema ?</p>	<p>Objetivos específicos (OE)</p> <p>OE1: Determinar las características de tráfico en una intersección vial para un entorno de tiempo real</p> <p>OE2: Verificar un modelo de reconocimiento y conteo vehicular en tiempo real</p> <p>OE3: Desarrollar el algoritmo de obtención de características de tramo vehicular</p> <p>OE4: Desarrollar el algoritmo que permita asignar intervalos de tiempos de color al semáforo en el tramo correspondiente</p> <p>OE5: Simular el funcionamiento del sistema de control de semáforo</p>	<p>Un sistema inteligente basado en un modelo heurístico, mediante la gestión en tiempo real de un semáforo, reduce significativamente los tiempos de espera de los vehículos en una intersección vial</p>	<p>Variable dependiente Variable Y: Tiempos de espera de los vehículos en una intersección vial</p>

Nota: X e Y representan a las variables independiente y dependiente respectivamente.

En la Tabla 1 se muestra la matriz de consistencia, en la que se establece la correspondencia entre el problema de investigación y el objetivo del mismo.

3.1.2 Operacionalización de variables

Tabla 2

Matriz de operacionalización de variables

Hipótesis	Variables	Dimensión	Definición conceptual	Definición operacional	Indicadores	Instrumento	Medida					
Un sistema inteligente basado en un modelo heurístico, mediante la gestión en tiempo real de un semáforo, reduce significativamente los tiempos de espera de los vehículos en una intersección vial	Variable Independiente Variable X:	X:	Sistema que puede percibir una realidad del tráfico vehicular, procesa datos y toma decisiones.	Ejecución de Componentes de software que ejecutan algoritmos, procesan datos y emiten un resultado	Número de vehículos	Algoritmo	Unidades					
	Sistema inteligente basado en un modelo heurístico con gestión en tiempo real de un semáforo	Sistema inteligente			Sistema inteligente	Velocidad de vehículos	Algoritmo	Km/h				
						Número de carriles	Algoritmo	Unidad				
						Densidad vehicular	Algoritmo	Unidades/ carriles				
						Tiempo de color de fase	Algoritmo	Segundos (S)				
	Tiempos de espera de los vehículos en una intersección vial	Variable dependiente Variable Y:			Y:	Tiempo promedio en que un grupo de vehículos está en espera en una intersección vial	Ejecución del cálculo de los tiempos de fase, registra el tiempo de espera e indica el nivel de tráfico	Tiempo de espera	Reporte	Segundos (s)		
								Tiempos de espera de los vehículos en una intersección vial	Tiempos de espera	Nivel de tráfico en intersección	Reporte	Nivel de congestión vehicular

Nota: Km/h significa la medida de la velocidad en kilómetros por hora y S es el tiempo en segundos.

En la Tabla 2 se muestra la matriz de operacionalización de variables, donde se muestra los instrumentos e indicadores de acuerdo a la definición operacional.

3.2 Métodos de la investigación

Inicialmente la investigación usó el método de observación en segmentos de vías vehiculares (tramos) antes de las de las intersecciones, a fin de captar de manera manual la realidad y tomar datos. Estos datos se pueden verificar en la Tabla Nro. 4, respecto a los tiempos de semáforo de 9 intersecciones del distrito de Nuevo Chimbote, Perú.

Se procedió luego al método experimental, a través de la simulación, dado que la simulación permitió investigar y evaluar el impacto de las variables en el sistema sin necesidad de realizar experimentos físicos, costosos o peligrosos, como es el caso del tránsito vehicular. Por ende permitió verificar los algoritmos del sistema y los resultados obtenidos de aplicación de manera general un semáforo de una intersección.

3.3 Diseño o esquema de la investigación

O1-----X-----O2

O1: Observación inicial, sin sistema inteligente

O2: Observación posterior, luego de la aplicación del sistema inteligente.

3.4 Población y muestra

- **Población:**

- Un escenario representa un instante en que se tiene una cantidad de vehículos de norte a sur, de oeste a este y el tipo de tráfico en dicho instante. En el Anexo I, se puede ver características de diversos escenarios con valores. Así mismo, en un escenario se presentan tiempos de rojo, amarillo y verde.
- Por cuanto los escenarios de la realidad en cuanto al número de vehículos que llegan a una intersección son aleatorios y de acuerdo a las limitaciones del estudio, se ha procedido a la simulación, en este caso infinitos escenarios de simulación de la realidad por el orden de $N = 100000$ escenarios o infinitos.

- Muestra:

Cuando se tiene muestras poblacionales grandes, entonces se debe aplicar la ecuación 23, para obtener el número de muestras

$$n = \frac{N * Z^2 * p^2}{E^2 * (N - 1) + Z^2 * p^2} \quad (23)$$

Donde:

n=	tamaño de la muestra
Z = 2,58	nivel de confianza a 99%
p = 0,5	probabilidad de ocurrencia
E= 0,01	error muestral

$$n = 14267$$

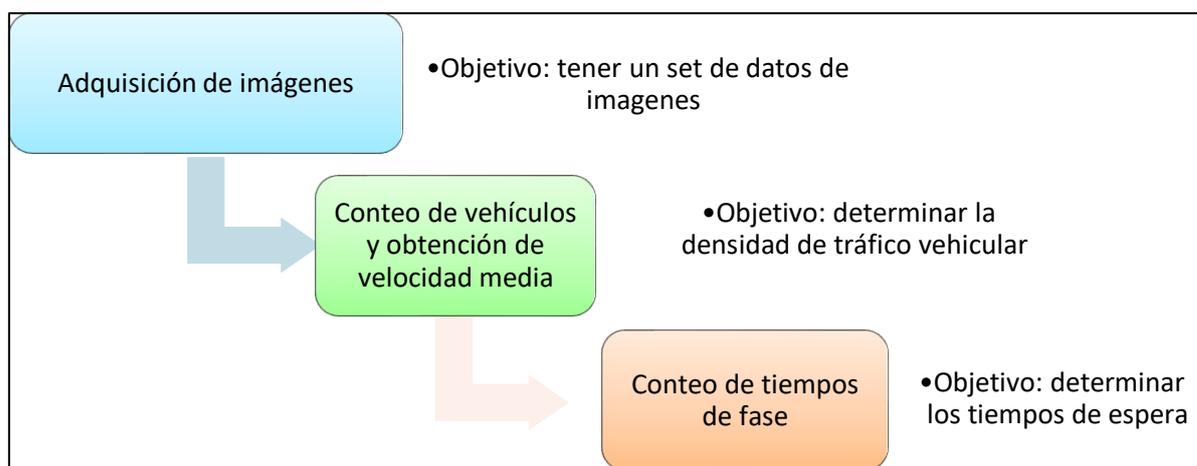
Para tener una muestra representativa se requiere de 14267 escenarios en nuestro caso se aproximó a los 15000 escenarios.

3.5 Actividades del proceso investigativo

Para comprender las actividades del proceso investigativo, luego de haber efectuado las fases de revisión bibliográfica, establecer el marco teórico respectivo y tener definido nuestra hipótesis, variables, población y muestra, se procedió a la recolección de datos cuyos componentes principales son: el proceso de adquisición de Imágenes, el conteo manual de vehículos, la determinación de la velocidad media de tramo y el conteo de tiempos de fase del semáforo en intersección (ver Figura 10).

Figura 10

Diagrama de bloques del procedimiento de recolección de datos



Nota: Todos los procesos del diagrama se efectuaron de manera manual.

3.6 Técnicas e instrumentos de recolección de datos

A continuación, en la Tabla 3 se lista las técnicas e instrumentos de recolección de datos que se usaron en la investigación:

Tabla 3

Técnicas e instrumentos de recolección de datos

Técnica	Instrumentos
Captura de imágenes	Cámara digital
Detección y reconocimiento de imágenes	Algoritmo pre entrenado
Seguimiento de objetos en tiempo real	Algoritmo de seguimiento
Conteo manual de vehículos	Lista de observación
Medición manual de espacio recorrido	Cronómetro, wincha, lista de observación.
Simulación de escenarios	Algoritmo heurístico, del autor y aplicación de cadenas de Markov,

3.7 Procedimiento de recolección de datos

El procedimiento de recolección de datos se inició identificando las intersecciones viales que tienen semáforos en la localidad de Nuevo Chimbote, a fin de recolectar datos de tráfico vehicular de dichas intersecciones. De 13 intersecciones con semáforo que cuenta la localidad de Nuevo Chimbote, se ha elegido 2 intersecciones que en las horas punta generan mayor congestión de tráfico, y en periodo escolar, que son: la intersección de Av. Pacífico y Av. Anchoqueta y la intersección de Av. Pacífico Av. Country.

3.7.1 Adquisición de imágenes

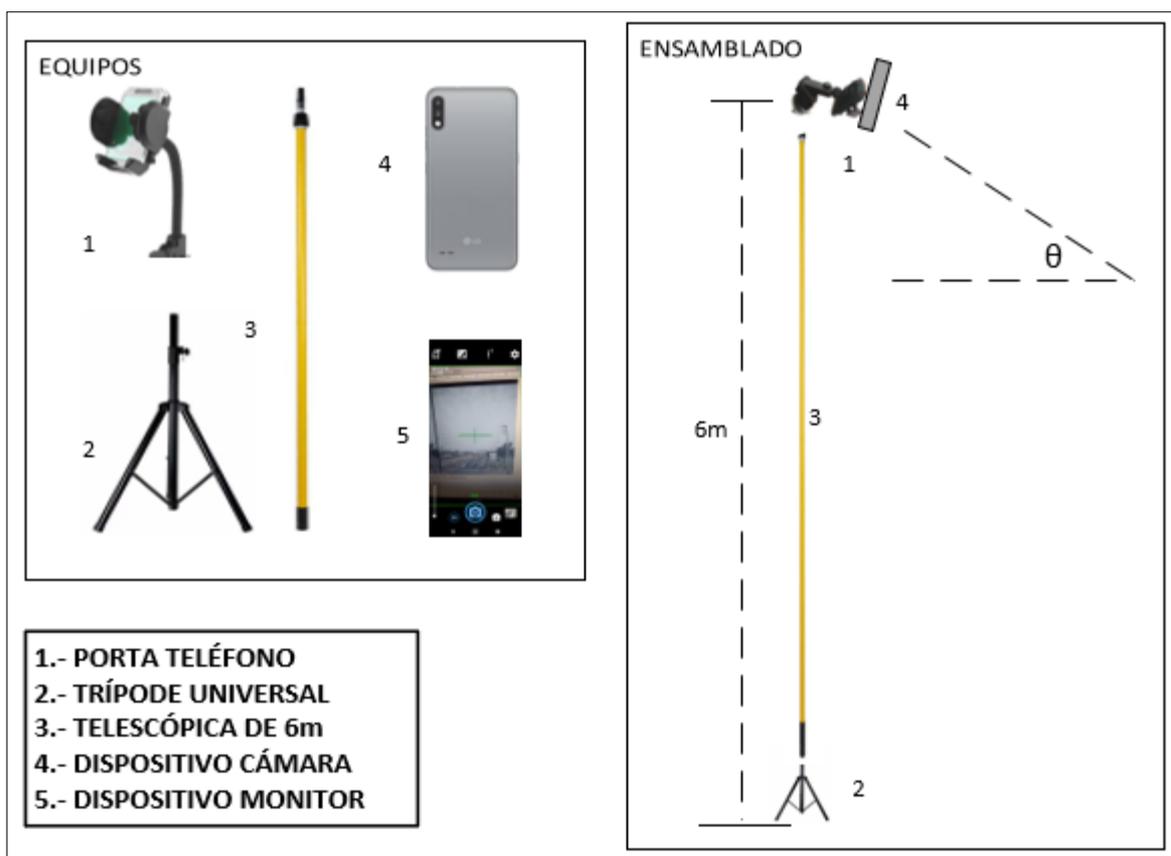
Especificaciones de los equipo y software que se utilizó:

- Porta teléfono
- Trípode universal
- Extensión telescópica de aluminio hasta 6m.
- Dispositivo smartphome de cámara principal de 2 mega pixel, 800x600, aspecto 4:3
- Dispositivo monitor smartphome de 2 megapíxel mínimo
- Open Camera y TeamViewer

Se ensambló el equipo completo tal como muestra la Figura 11, se instala open Camera en el equipo (4), se instala TeamViewer en el equipo (4) y (5) , de manera que con el equipo (5) se pudo visualizar el tramo vehicular y realizar las grabaciones correspondientes en imágenes y video de 7 a.m. a 8:30 a.m.

Figura 11

Ensamble de quipos para la captura de imágenes



Nota: El ángulo de inclinación dependerá de la ubicación del equipo ensamblado y de la distancia al pase peatonal.

3.7.2 Recolección de datos de velocidad media

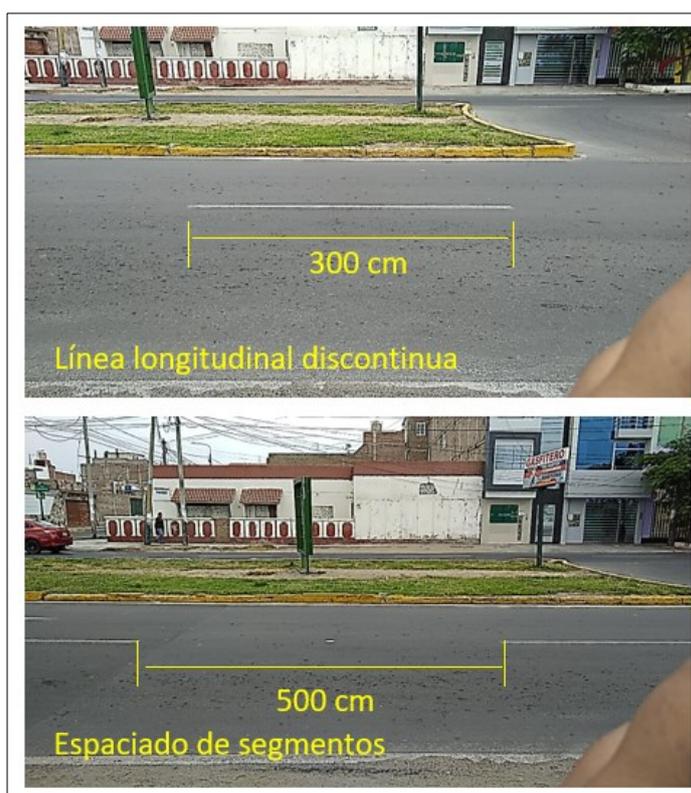
Para la recolección de datos de la velocidad media de los vehículos, se procedió de la siguiente manera:

- Se midió la línea longitudinal discontinuas de la pista, luego se midió la distancia entre las líneas longitudinales, expresadas en metros
- Luego se elige un vehículo antes de la intersección y se mide el tiempo en segundos que demora en atravesar cierto número de líneas discontinuas y segmentos entre líneas discontinuas.
- Se usó un cronómetro digital que permita dar inicio y parada al conteo y con redondeo hasta las centésimas de segundo.

- Para el caso de los vehículos que están al inicio del tramo vehicular, se mide la distancia de la intersección y se toma el tiempo que demora en atravesar la intersección.
- Asimismo se registró los videos para la contrastación con las recolección manual.

Figura 12

Dimensiones de línea discontinua y espaciado de segmento



Nota: las medidas de la figura corresponden a pistas urbanas, 3m para la línea discontinua y 5m para el espaciado de segmentos.

Además de los mostrado en la Figura 12, según el Ministerio de Transportes y Comunicaciones del Perú, en carreteras y autopistas la distancia de cada línea discontinua es de 4,5 m. y el espaciado de segmentos es 7,5 m.

3.7.3 Recolección de datos para conteo tiempo de fase

Para recolectar datos de los tiempos de duración de fase de los semáforos, se tuvo que observar los tiempos de VERDE, AMARILLO, Y ROJO de cada semáforo y se tuvo que anotar los tiempos de fase.

Todos estos datos se obtuvieron mediante observación presencial ubicándonos en cada intersección descrita en la Tabla 4. A la fecha de elaboración del presente informe aún mantienen dicha configuración.

Se ha observado que los tiempos que asigna el semáforo a cada color son de manera predefinida.

Tabla 4

Tabla de asignación de tiempos de fase en una intersección

Intersección	Nombre del lugar de referencia	Sentido	Sentido	Sentido	Sentido	AMARILLO	Tramo Mayor Tráfico
		N-S VERDE	N-S ROJO	E-O VERDE	E-O ROJO		
1	Mercado Belén	45	39	39	45	3	N-S/S-N
2	Ovalo Bruces 1	36	49	49	36	3	N-S/S-N
3	Ovalo Bruces 2	34	45	45	34	3	N-S/S-N
4	Mercado Bs. As.	36	34	34	36	3	N-S/S-N
5	Colegio Sr. de la vida	51	34	34	51	3	N-S/S-N
6	Of. Rectorado UNS	35	26	26	35	3	N-S/S-N
7	Plaza Mayor	55	34	34	55	3	N-S/S-N
8	SENATI-UNS	36	41	41	36	3	E-O/OE
9	Anchoveta-Argentina	34	48	48	34	3	E-O/OE

Nota: Observar que el tiempo de fase VERDE de un sentido coincide con el tiempo de fase ROJO del sentido transversal

3.8 Técnicas de procesamiento y análisis de los datos

3.8.1 Conteo de vehículos

Especificaciones de Software y hardware

Software

- Sistema Operativo Linux 22.04 LTS
 - Anaconda 3.x, Python 3.8.x, OpenCV 4.5.1 (para Linux) , Yolo V4
- Sistema Operativo Windows 10pro
 - Software para el análisis estadístico de datos SPSS 25

Hardware

- Memoria RAM de 8Gb
- Procesador Intel Core I7 1,9 GHz
- GPU NVIDIA GeForce MX130,VRAM 2Gb,

Descripción del proceso

En esta etapa se utilizó un algoritmo en Python que se encarga de detectar objetos y en la medida que detecta objetos va contabilizando en tiempo real cada uno de los vehículos.

Se utilizó 96 escenarios (imágenes) de manera aleatoria a fin de contabilizar de manera manual y mediante algoritmo.

Al momento de aplicar las librerías de clasificación de objetos, se tuvo que modificar el código de las librerías para poder obtener la clasificación de vehículos y el algoritmo, discrimine objetos que no son de interés de la captura como postes de luz, ventanas, arboles etc. (ver Figura 13).

El retorno de la ejecución del código devolvió los parámetros número de vehículos (ver Anexo C) quien genera la variable ContadorAlgoritmo que es la muestra 1.

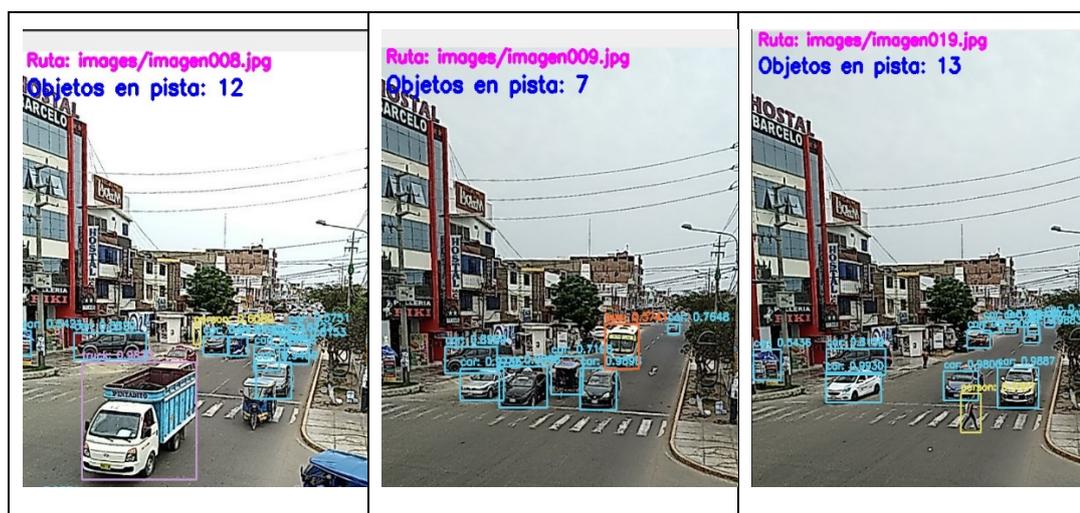
Por otro lado, los datos del conteo manual se expresan en la variable ContadorManual que es la muestra 2

El valor absoluto de la diferencia entre ContadorAlgoritmo y ContadorManual es igual a la variable Variación, lo cual permite hallar nivel de precisión.

Finalmente, para cerrar este proceso se usó SPSS software estadístico para verificar diferencias significativas de las dos muestras y dar validez a los resultados.

Figura 13

Detección en tiempo real



Nota: En la figura se presenta la detección de vehículos de un tramo de intersección. Av. Pacifico. Nvo. Chimbote, Perú.

El solapamiento de vehículo

A medida que los vehículos aparecen de manera ordenada en el tramo, el detector acierta en el conteo de vehículos con una mejor precisión.

Como se puede visualizar en la figura 14A no existe problemas de solapamiento, pues al haber espacios separados entre vehículos quedan a vista de la cámara y esta puede obtener una vista de todos los objetos.

El caso de solapamientos es el estado del arte a la fecha, que se está estudiando con el tema seguimientos de objetos.

Figura 14

Solapamiento de vehículos



Nota: Los vehículos (figura 14B) a medida que se acercan a la calzada se van solapando, para la vista de la cámara, es decir; un vehículo empieza a ocultar a otros.

3.8.2 Determinación de la medida de la velocidad

En este caso se realizó una comparación de los datos de velocidad de vehículos obtenidos mediante conteo y se contrastó con los datos obtenidos mediante la aplicación de un algoritmo traducido a Python con la técnica de seguimiento de objetos en tiempo real.

Descripción del proceso

En el Anexo D, se observa que se utilizó 16 casos (vehículos) de manera aleatoria en un lapso de tiempo a fin de contabilizar de manera manual, se debe indicar que se grabó la ocurrencia en un archivo de video en cada toma de datos.

Por otro lado, para medir el tiempo que el vehículo atraviesa cierta distancia, en nuestro caso se usó la distancia $d = 9,5$ m que es la distancia del segmento entre líneas discontinuas + 2 metros de tolerancia.

Con un cronometro digital, se mide el instante que el vehículo inicia el recorrido de la distancia d y se detiene al término de la distancia d . En este anexo D, una variable se denomina *ConteoManual* (s). El tiempo (t) en segundos obtenido de manera manual se usa para obtener la velocidad en km/h de la siguiente manera:

$Velocidad = (9,5m / t)(3600 s / 1h)(1km/1000m)$, la velocidad en km/h es la muestra 1.

En el Anexo F, se puede observar las equivalencias correspondientes de transformación de km/h a m/s.

Asimismo, el retorno de la ejecución del algoritmo devolvió los parámetros velocidad de vehículos quien genera la otra variable *ConteoAlgoritmo* que es la muestra 2.

El valor absoluto de la diferencia entre *ConteoAlgoritmo* y *Velocidad* es igual a la variable *Variación*, lo cual permitió hallar nivel de precisión.

Finalmente, para este proceso se usó SPSS software estadístico para realizar la Diferencia de Medias, y verificar diferencias significativas de las dos muestras y dar validez a los resultados.

3.8.3 Determinación de características de tráfico

3.8.3.1 El flujo de Tráfico

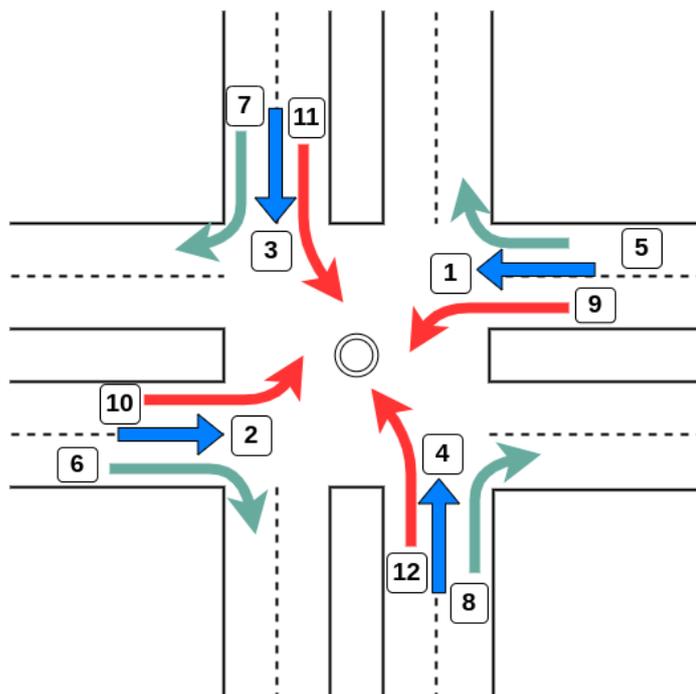
En cuanto al flujo de tráfico en una intersección separada por tiempo o controlada por semáforo, se pueden presentar múltiples escenarios de comportamiento de flujos.

Descripción de flujos

Se ha representado gráficamente los escenarios de los flujos que tratan de competir para cruzar la intersección para determinar los flujos conflictivo y no conflictivos, ver figura 15.

Figura 15

Flujos de tráfico en una intersección vial



Nota: El orden de la enumeración de los flujos de tráfico está basado en la figura de (Younes & Boukerche, 2018) y (Younes et al., 2023), está permitida voltear a la izquierda y presenta 12 flujos de tráfico.

Los flujos de tráfico 1,2 y luego los flujos 3 y 4 se identifican como flujo de tráfico no conflictivos.

Los flujos de tráfico 5,6,7 y 8 se identifican como flujos de tráfico no conflictivos, por cuanto en el momento en que se asigna el color verde a dichos flujos, éstos podrán girar a la derecha sin problemas y con seguridad.

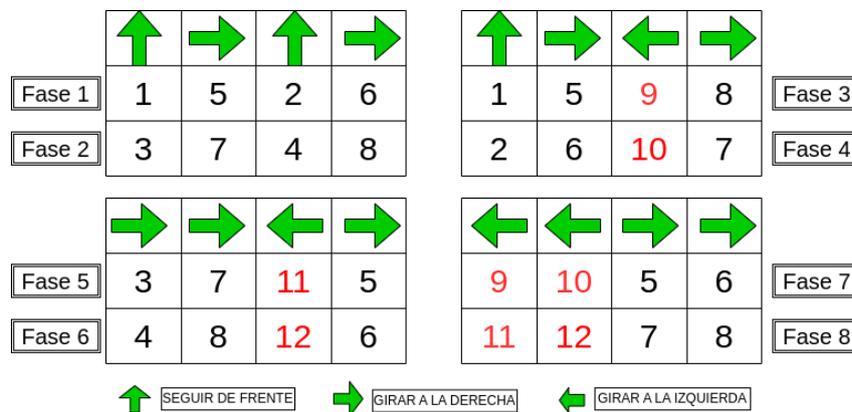
Los flujos de tráfico 9,10,11 y 12, tienen la posibilidad de girar a la izquierda; situación que entra en conflicto con los flujos 2,1, 4 y 3 respectivamente, cuando éstos entran en fase.

3.8.3.2 Fases de flujo conflictivo y no conflictivo

Las fases con flujos de tráfico no conflictivos, son las fases que brindan cierto nivel de seguridad y equidad en la asignación de tiempos de fase. Pero, al permitirse el giro a la izquierda en la intersección, se forman otros flujos de tráfico con flujos conflictivos.

Figura 16

Fases no conflictivas



Nota: En la Figura 16, se observa que las fases 1 y 2 se presentan flujos de tráfico no conflictivos, pero en las demás fases sí se presentan al menos un flujo conflictivo.

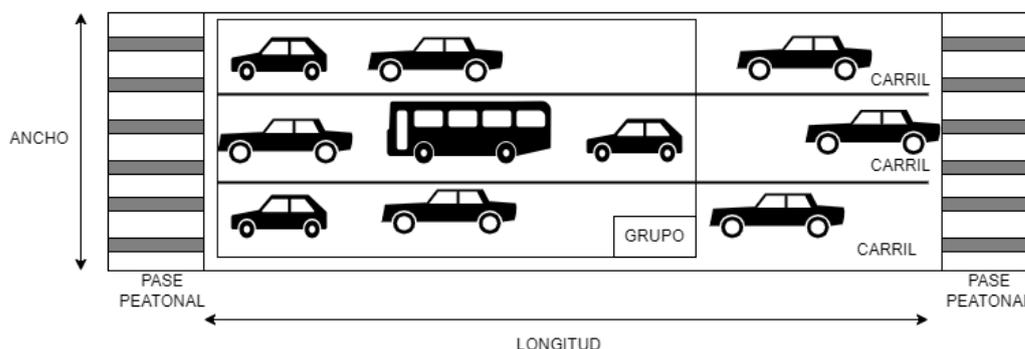
Observar que aun cuando en una fase interviene un flujo de tráfico conflictivo, en su comportamiento de fase no genera demasiado conflicto o al menos hay cierto nivel de seguridad; esto se puede apreciar en las fases 3,4,5 y 6 donde 3 flujos de tráfico salen del mismo tramo y el cuarto flujo es el flujo más distante. Por otro lado, en las fases 7 y 8 intervienen 2 flujos que giran a la izquierda y 2 flujos que giran a la derecha, los cuales en su comportamiento de fase no entrarán en conflicto y brindan cierto nivel de seguridad.

3.8.3.3 Análisis de la densidad vehicular

Como ya se ha visto en el marco teórico la densidad expresa la cantidad de vehículos por unidad de tiempo o espacio. Para comprender este aspecto de la densidad de grupo se ha diseñado la Figura 17 a fin de comprender los elementos que intervinieron en el cálculo matemático.

Figura 17

Elementos de un tramo vehicular



Un **grupo (G)** es el conjunto de vehículos que llegan a la intersección dentro del área de un tramo.

La **longitud** es la distancia entre pasos peatonales del tramo.

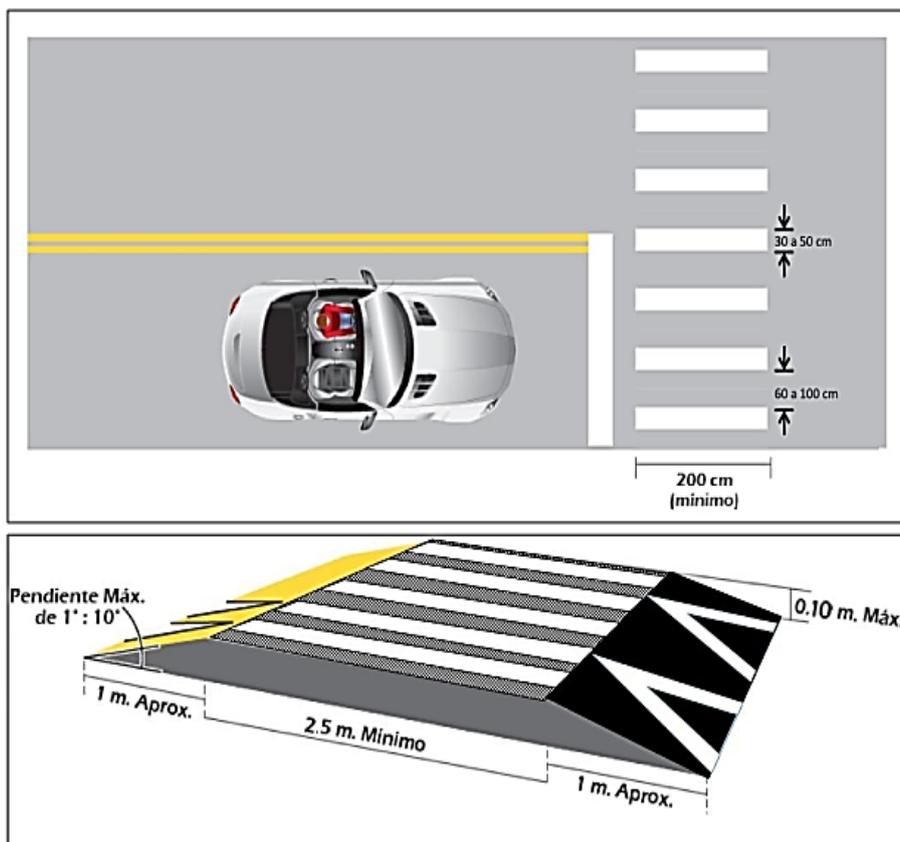
EL **ancho** es la distancia que cubre el ancho de la pista únicamente por donde circulan los vehículos.

Carril es la sección del tramo longitudinal en el mismo sentido. Entre carriles, se presenta una línea de carril es de color blanco, discontinua o segmentada. Se usan en las aproximaciones de las intersecciones.

Paso peatonal o cruce peatonal : es la sección del tramo establecido para el pase de peatones, mínimo de 2,00 metros de ancho para pases peatonales sin reductores o mínimo de 2,50 metros de ancho para pases peatonales con reductores de ancho de acuerdo al MTC como se puede ver en la Figura 17 y 18.

Figura 18

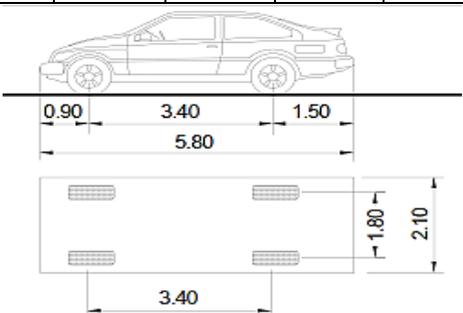
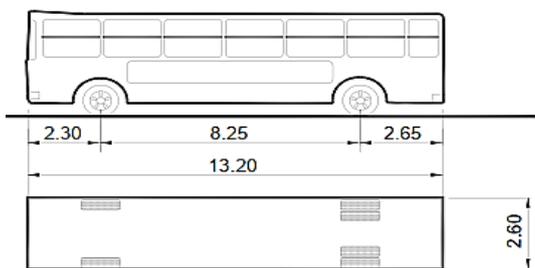
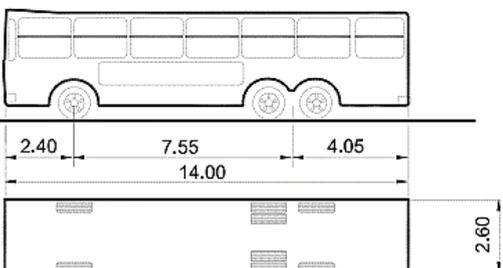
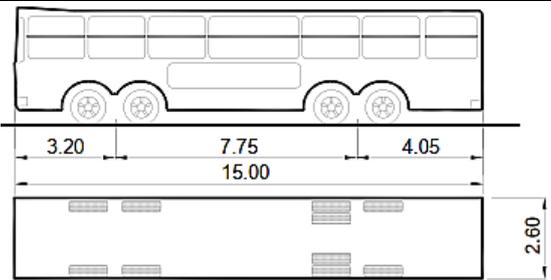
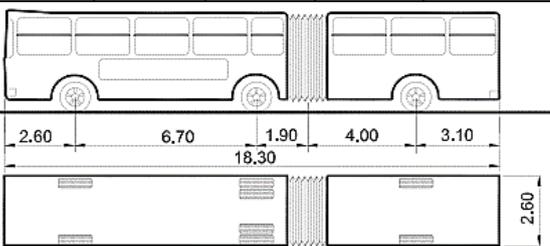
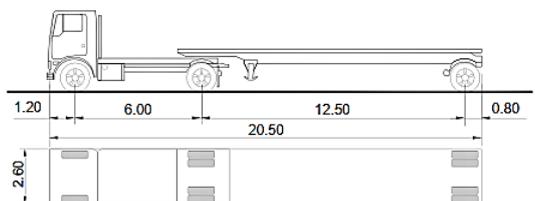
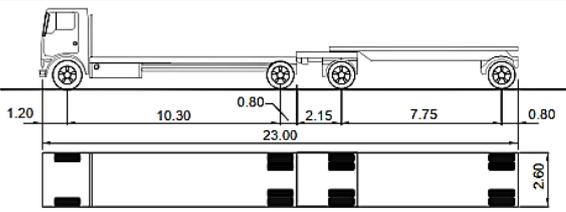
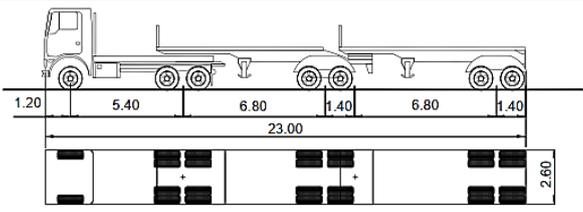
Demarcación de línea de cruce peatonal



Nota: Cuando el cruce peatonal tiene reductores de velocidad el ancho del mismo debe ser de 2,5 m en otro caso es de 2m (Ministerio de Transportes y Comunicaciones, 2018).

Figura 19

Datos de vehículos MTC

Tipo de vehículo	Código MTC	Alto	Ancho	Largo	Tipo de vehículo	Código MTC	Alto	Ancho	Largo
Vehículo ligero	VL	1.30	2.10	15.80	Ómnibus de dos ejes	B2	4.10	2.60	13.20
									
Ómnibus de tres ejes	B3-1	4.10	2.60	14.00	Ómnibus de cuatro ejes	B4-1	4.10	2.60	15.00
									
Ómnibus articulado	BA-1	4.10	2.60	18.30	Semirremolque simple	T2S1	4.10	2.60	20,50
									
Remolque simple	C2R1	4.10	2.60	23.00	Semirremolque doble	T3S2S2	4.10	2.60	23.00
									

Nota: El MTC, usa los datos de este grafico para el dimensionamiento de carreteras (MTC-DG, 2018).

Tabla 5*Dimensiones de vehículos ligeros*

Marca y modelo	Largo	Ancho	Alto
MINIVAN HYUNDAI STARIA	5,52	2,00	1,99
NISSAN NP300 NAVARA	5,33	1,85	1,84
TOYOTA HILUX	5,33	1,86	1,82
MAZDA 6	4,87	1,84	1,45
HYUNDAI IONIQ 6 2023	4,86	1,88	1,50
LAND CRUISER PRADO	4,84	1,89	1,89
TOYOTA 4RUNNER	4,83	1,93	1,78
KIA STINGER	4,83	1,87	1,40
KIA SORENTO	4,81	1,90	1,70
TOYOTAFORTUNER	4,80	1,86	1,84
HYUNDAI SANTA FE	4,79	1,90	1,69
VOLKSWAGEN TIGUAN ALLSPACE	4,73	1,84	1,67
CHERY TIGGO 8	4,70	1,86	1,71
MAZDA CX-30	4,70	1,80	1,54
NISSAN X-TRAIL 2023	4,68	1,84	1,73
KIA EV6	4,68	1,88	1,55
HYUNDAI NEXO	4,67	1,86	1,64
MAZDA 3 SEDAN	4,66	1,80	1,44
CHEVROLET CAPTIVA	4,66	1,84	1,76
NISSAN SENTRA	4,64	1,82	1,46
HYUNDAI IONIQ 5	4,64	1,89	1,61
TOYOTA RAV4	4,60	1,86	1,69
NISSAN ARIYA	4,60	1,85	1,66
HYUNDAI I30 CW	4,59	1,80	1,48
NISSAN E-NV200 EVALIA (2018-2021)	4,56	1,76	1,86
VOLKSWAGEN TIGUAN	4,54	1,84	1,67
CHERY ARRIZO 5	4,53	1,81	1,49
KIA SPORTAGE	4,52	1,87	1,65
HYUNDAI TUCSON	4,50	1,87	1,65
NISSAN TOWNSTAR	4,49	1,86	1,89
AUTO NISSAN LEAF	4,48	1,79	1,54
HYUNDAI IONIQ	4,47	1,82	1,45
TOYOTA COROLLA CROSS	4,46	1,83	1,62
MAZDA 3	4,46	1,80	1,44
HYUNDAI I30 FASTBACK	4,46	1,80	1,43
CHERY ARRIZO 3	4,45	1,75	1,49
NISSAN QASHQAI	4,43	1,84	1,64
HYUNDAI ACCENT 2022	4,40	1,73	1,46
TOYOTA ETIOS	4,36	1,69	1,51
HYUNDAI KONA 2023	4,36	1,83	1,58
AUTO HYUNDAI I30	4,34	1,80	1,46
JAC JS3	4,33	1,77	1,66
CHERY TIGGO 3	4,32	1,83	1,66
CHEVROLET AVEO	4,31	1,71	1,51
HYUNDAI CRETA	4,30	1,79	1,64
VOLKSWAGEN GOLF	4,26	1,79	1,45
CHEVROLET SAIL	4,25	1,69	1,50
CHERY TIGGO 2 PRO	4,20	1,66	1,57
KIA E-SOUL	4,20	1,80	1,61
AUTO HYUNDAI BAYON	4,18	1,78	1,49
SUZUKI VITARA	4,18	1,78	1,61
VOLKSWAGEN T-CROSS	4,11	1,76	1,58
MAZDA 2 2023	4,08	1,70	1,50
KIA RIO SEDAN	4,07	1,73	1,45
AUDI A1 STREET	4,05	1,76	1,48
HYUNDAI ACCENT	4,05	1,70	1,47
AUTO HYUNDAI I20	4,04	1,75	1,45
TOYOTA RAIZE	4,03	1,71	1,61
CHEVROLET BEAT NOTHCBACK	4,00	1,60	1,55
AUTO NISSAN MICRA	3,99	1,74	1,60
SUZUKI SWIFT	3,84	1,74	1,50
AUTO HYUNDAI I10	3,67	1,68	1,48
SUZUKI JIMNY	3,65	1,65	1,72
KIA PICANTO	3,60	1,60	1,49
Media	4,46	1,80	1,60

3.8.3.4 Obtención del tamaño promedio de vehículos

Se tomó una muestra de 64 vehículos que corresponden a los vehículos ligeros más usados en el transporte de la localidad y se procedió a obtener las dimensiones correspondientes de acuerdo a la marca y modelo, datos que se pueden obtener de sus respectivas especificaciones publicadas.

Resultados del proceso

Se necesitó saber si los datos mostrados en la Tabla 5., en cuanto a largo, ancho y alto pertenecen a una distribución normal, con un nivel de confianza de 95% y una significancia alfa de 0,05, bajos los considerandos siguientes:

- Si la muestra es mayor igual que 50 se emplea Kolmogórov-Smirnov
- Si la muestra es menor que 50 se emplea Shapiro-Wilk
- Si p-valor < 0,05 la hipótesis de que los datos tiene distribución normal **se rechaza**, caso contrario **se acepta**

Tabla 6

Prueba de normalidad de los datos de vehículos ligeros

Variable	Min	Max	Media	Error porcentual	Kolmogorov - Smirnov gl	Sig.	Shapiro-Wilk gl	Sig.
Largo	3,60	5,52	4,4553	0,04655	64	0,200	64	0,164
Ancho	1,60	2,00	1,7989	0,00978	64	0,191	64	0,160
Alto	1,40	1,99	1,5977	0,01711	64	0,002	64	<0,001

Nota: gl son los grados de libertad, Sig. es el p-valor

Tras observar los datos y dado que la muestra es mayor de 50, se considera la prueba de Kolmogórov-Smirnov donde se observa que la variable Largo y Ancho siguen una distribución normal por cuanto p-valor > α (0,05), en cambio la variable Alto no sigue una distribución normal debido a p-valor < α (0,05).

Análisis de los datos del tamaño promedio de vehículos

Por cuanto nuestra prioridad es determinar el valor de Largo y Ancho promedio de los vehículos ligeros, a fin de obtener el área en promedio que ocupan en un tramo, la variable alto por ahora no será considerado.

De acuerdo a la muestra de vehículos ligeros se tiene que los valores de las medias obtenidas para Largo y Ancho son valores representativos por cuanto siguen una distribución normal.

Según el diseño de los vehículos ligeros del MTC, estos tienen un largo de 5,80m, un ancho de 2,10m y un alto de 1,30m (ver Figura 19). De acuerdo a nuestra muestra de dimensiones de vehículos ligeros tenemos un Largo de 4,46m, un ancho de 1,80m y un alto de 1,60m (ver Tabla 5 y 6). Sin embargo, solamente los vehículos ligeros de tipo camioneta con tolva y minivan bordean o superan el tamaño de largo de 5,80m propuesto por el MTC.

Por otro lado, el MTC no especifica exactamente la distancia prudencial entre vehículos que se van a detener, dado que es muy variable, sin embargo, sugiere y responsabiliza al conductor que va detrás de otro vehículo mantener su distancia prudencial. Para fines experimentales se ha considerado que la distancia de separación prudencial sea mínimamente de 1m. y para la distancia de ancho lateral sea de 0,7 m. mínimamente.

Cálculos para obtener el área de tramo ocupada por vehículo

$$\text{Espaciamiento} = \text{Media obtenida} + \text{separación prudencial}$$

$$\text{Espaciamiento} = 4,46\text{m} + 1,00\text{m} = 5,46 \text{ m}$$

$$\text{Ancho} = 1,80\text{m} + 0,70\text{m} = 2,50 \text{ m}$$

$$\text{Área de tramo ocupado por Vehículo (AV)} = \text{Largo} \times \text{Ancho}$$

$$\text{AV} = 5,46\text{m} \times 2,50 \text{ m} = \mathbf{13,65 \text{ m}^2}$$

De acuerdo a la ecuación (20) la densidad máxima es igual al número de vehículos dividido entre el número de carriles (veh/carril) o el número de vehículos por kilómetro de carretera (veh/km).

Pero los vehículos en una intersección se aproximan en Grupos (G), por lo que usaremos la densidad como el número de vehículos dividido entre al Área del tramo.

3.8.3.5 Fórmula para hallar la densidad del grupo

$$D_G = \frac{N_{VG}}{A_G} \quad (24)$$

Donde:

D_G : es la densidad del grupo que se acerca a la intersección

N_{VG} : es el número de vehículos que llegan a la intersección

A_G : es el área expresada en metros cuadrados.

3.8.3.6 Índice de congestión de tráfico (ICT)

Para los fines de esta investigación a la relación de la Densidad de grupo entre la densidad máxima se denominó índice de congestión de tráfico (ICT). Si D_G/D_{max} es cercano a 1 indica un tramo más congestionado, mientras que una densidad cercana a cero sugiere menos congestión.

$$ICT = \frac{D_G}{D_{max}} \quad (25)$$

Donde:

ICT : es el índice de congestión de tráfico

D_G : es la densidad del grupo que se acerca a la intersección

D_{max} : es la densidad máxima cuando los vehículos están detenidos

EL rango de valores de ICT permite definir el tipo de congestión vehicular, bajo, medio alto, o de libre flujo, etc.

Se debe indicar que para medir el ancho y largo de un tramo se usó las herramientas de OpenStreetMaps y Google Earth (ver Anexo B). El cual se puede usar en servicios de posicionamiento de alta precisión, como reconocimiento de campo, diseño de redes topográficas y descripción de estaciones (Wang & Wang, 2020).

Tabla 7

Tramos vehiculares con longitud y ancho respectivo

Tramo	Flujo Tráfico	Nombre de tramo	Longitud	Ancho	Área (m2)	Carriles	AV	MaxVeh
1	1	Pacifico-Anchoveta-Sur a Norte	101,45	9,51	964,79	3	13,65	56
2	2	Pacifico-Anchoveta-Norte a Sur	108,98	9,87	1075,63	3	13,65	60
3	3	Anchoveta-Pacifico Este a Oeste	116,85	6,19	723,30	2	13,65	43
4	4	Anchoveta-Pacifico Oeste a Este	70,36	10,50	738,78	3	13,65	39
Total								215
5	1	Pacifico-Country- Sur a Norte	99,24	10,70	1061,87	3	13,65	55
6	2	Pacifico-Country-Norte a Sur	136,40	10,38	1415,83	3	13,65	75
7	3	Country-Pacifico-Este a Oeste	110,23	4,85	534,62	2	13,65	40
8	4	Country-Pacifico-Oeste a Este	92,32	4,87	449,60	2	13,65	34
Total								204
9	1	Pardo-Gálvez-Sur a Norte	97,92	7,50	734,40	3	13,65	54
10	2	Pardo-Gálvez-Norte a Sur	85,09	8,26	702,84	3	13,65	47
11	3	Gálvez-Pardo Este a Oeste	98,03	9,39	920,50	3	13,65	54
12	4	Gálvez-Pardo Oeste a Este	92,74	9,63	893,09	3	13,65	51
Total								205

Nota: los tramos vehiculares fueron elegidos aleatoriamente y corresponden a tres intersecciones del área geográfica de la ciudad de Chimbote y Nuevo Chimbote, Perú. AV es el área promedio para un vehículo.

3.8.3.7 Análisis de la densidad en tramos de 2 y 3 carriles

Basándonos en la propuesta de propuesta de (Ashifuddin Mondal & Rehena, 2019). Donde propone 3 tipos de congestión: alta, congestión media y de flujo libre asignamos valores a los rangos respectivos.

Tabla 8

Tipo de congestión de acuerdo al ICT

Tipo de Congestión	ICT
FLUJO LIBRE	[0 ; 0,5>
MEDIA	[0,5; 0,8>
ALTA	[0,8; 1]

Nota: El índice de congestión de tráfico (ICT) se encuentra en el rango de 0 a 1.

Se hizo pruebas en base a los datos de la Tabla 7 y Tabla 8 y se obtuvo algunos escenarios relevantes, como se muestran a continuación:

Tabla 9

Escenario1:Densidad en 2 carriles menor que densidad de 3 carriles

Tramo	FlujoTraf	Longitud	Ancho	Área (m2)	Carriles	AV	MaxVeh	Dmax	Rand(Veh)	D _G	ICT
1	1	101,45	9,51	964,79	3	13,65	56	0,0578	54	0,0560	0,9688
2	2	108,98	9,87	1075,63	3	13,65	60	0,0557	35	0,0325	0,5845
3	3	116,85	6,19	723,30	2	13,65	43	0,0592	13	0,0180	0,3037
4	4	70,36	10,50	738,78	3	13,65	39	0,0523	19	0,0257	0,4915
Total							197		121		

En el escenario1, Tabla 9, se tiene al flujo 1 en un sentido y al flujo 2 en sentido contrario cuyos índices de congestión de tráfico son 0,9688 (congestión alta) y 0,5845 (congestión media) respectivamente. En este caso las vías de 3 carriles tienen mayor ICT, lo cual indica que un 97% de área del tramo 1 está ocupada por vehículos y se encuentra al tope o muy cerca de la capacidad de automóviles en el tramo.

Tabla 10*Escenario 2: Densidad en 2 carriles mayor que densidad de 3 carriles*

Tramo	FlujoTraf	Longitud	Ancho	Área (m2)	Carriles	AV	MaxVeh	Dmax	Rand(Veh)	D _G	ICT
5	1	99,24	10,70	1061,87	3	13,65	55	0,0514	37	0,0348	0,6786
6	2	136,40	10,38	1415,83	3	13,65	75	0,0529	58	0,0410	0,7739
7	3	110,23	4,85	534,62	2	13,65	40	0,0755	28	0,0524	0,6935
8	4	92,32	4,87	449,60	2	13,65	34	0,0752	30	0,0667	0,8871
Total							204	153			

En el escenario 2 , Tabla 10, se puede ver el flujo 4 cuyo ICT es de 0,8871 (congestión alta) el cual presenta un 89% del área del tramo 8 ocupada. El tramo 8 es de 2 carriles con una densidad mayor al tramo 6, sin embargo, el tramo 6 tiene una mayor cantidad de vehículos, situación que se debe considerar para establecer prioridades.

Tabla 11*Escenario 3: Congestión alta en 2 y 3 carriles*

Tramo	FlujoTraf	Longitud	Ancho	Area (m2)	Carriles	AV	MaxVeh	D _{max}	Rand(Veh)	D _G	ICT
1	1	101,45	9,51	964,79	3	13,65	56	0,0578	48	0,0498	0,8611
2	2	108,98	9,87	1075,63	3	13,65	60	0,0557	54	0,0502	0,9018
3	3	116,85	6,19	723,30	2	13,65	43	0,0592	36	0,0498	0,8411
4	4	70,36	10,50	738,78	3	13,65	39	0,0523	29	0,0393	0,7501
Total							197	167			

En el escenario 3, Tabla 11, se puede ver que los flujos 1,2 y 3 presentan congestión alta. El flujo 3 es transversal a los flujos 1 y 2. Este escenario es un buen punto a considerar dado que, se tiene 2 vías transversales con congestión alta.

3.8.3.8 Análisis de la velocidad de tráfico

La importancia de la determinación de la velocidad del tráfico radica en que es un parámetro que nos indica con qué velocidad arriba el grupo (G_i) al inicio de intersección o con qué velocidad inicia el recorrido del tramo el siguiente grupo (G_j). Cada vehículo que arriba a una intersección tiene su propia velocidad (V_i), por tanto, lo más razonable es trabajar con la velocidad media del grupo de vehículos.

Otro de los principales usos es que al saber la velocidad media del grupo se evitaría las paradas bruscas que suele suceder cuando hay cambio de fase a ámbar y rojo, por ejemplo, otorgar unos segundos adicionales de fase si la velocidad media cae.

3.8.3.9 Fórmula para hallar la Velocidad Media de Grupo

$$VG_i = \frac{\sum_{i=1}^{N_{VG}} V_i}{N_{VG}} \quad (26)$$

Donde:

VG_i : es la velocidad media del grupo

N_{VG} : es el número de vehículos que llegan a la intersección

V_i : velocidad de cada vehículo individual del grupo

3.8.3.10 Fórmula para hallar el tiempo de arribo del Grupo

El tiempo de arribo se entiende como el tiempo en que el primer vehículo del grupo Veh1 (ver Figura 20), llega al inicio de la intersección (Inicio-Inter). Se obtiene dividiendo la distancia del vehículo1 al inicio de intersección entre la velocidad media del grupo VG_i .

$$TAG_i = \frac{d_{(Veh1, InicioInter)}}{VG_i} \quad (27)$$

Donde:

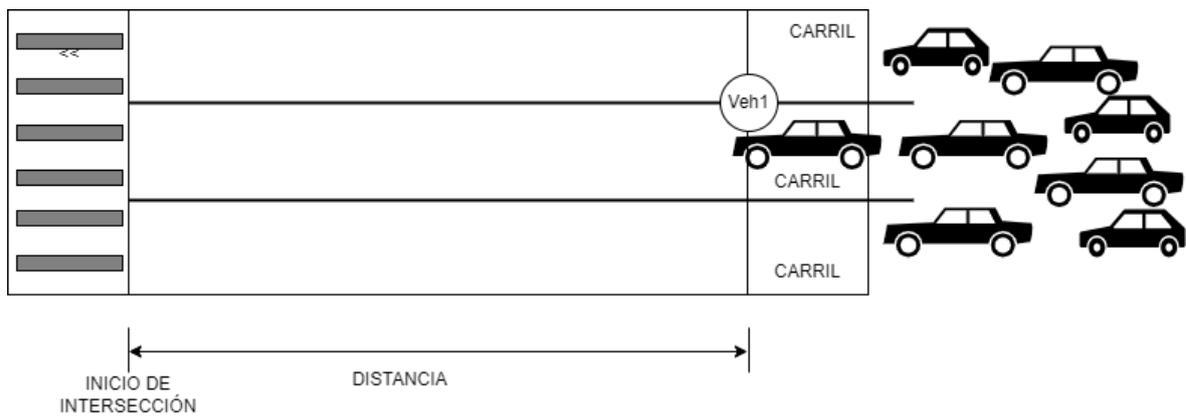
TAG_i : es el tiempo de arribo del grupo G_i

$Veh1$: es el primer vehículo del grupo G_i

d : es la distancia del primer vehículo del grupo G_i al inicio de intersección

Figura 20

Esquema para el tiempo de arribo



3.8.3.11 Establecimiento de prioridades

La expresión lógica es la siguiente: A mayor densidad de grupo y mayor velocidad media de grupo entonces mayor prioridad de flujo

$$(D_G \wedge V_G) \rightarrow P_{fi} \quad (28)$$

Donde:

D_G : es la densidad de grupo

V_G : es la velocidad media de grupo

P_{fi} : es la prioridad del flujo de tráfico

3.8.3.12 Tiempo de espera promedio

Es el tiempo de espera promedio de un vehículo (TEPV) se refiere al tiempo a que un vehículo (Veh_i) está detenido en intersección. Cuando el flujo de tráfico no está en fase cada vehículo tiene que detenerse (Veh_{stop}) en la intersección esperando que el flujo entre a fase para poder cruzarla.

El tiempo de espera promedio será la sumatoria de los tiempos de espera de cada vehículo dividido entre el total de vehículos. Estos valores se calculan para cada paso de selección de la fase siguiente en el algoritmo de programación del semáforo.

La fórmula del factor de índice es la siguiente:

$$TEPV_{stop} = \frac{\sum_{i=1}^{N_{Veh}} TEV_{stop\ i}}{N_{Veh}} \quad (29)$$

Donde:

$TEPV_{stop}$: Tiempo de espera promedio de vehículos detenidos

$TEV_{stop\ i}$: Tiempo de espera de un vehículo detenido en intersección

N_{Veh} : Número de vehículos

Para fines algorítmicos el tiempo de espera en el tramo será el tiempo de fase del tramo transversal.

3.8.3.13 Análisis de datos del factor de índice (Ei)

Es la sumatoria de las velocidades de los vehículos en movimiento (V_{mov}) multiplicados por sus respectivos pesos (W_{mov} en kg o libras). Younes et al. (2023), proponen este parámetro en primer lugar porque el peso y el tamaño del vehículo inciden en el consumo de combustible y la emisión de gases.

El criterio de la aplicación del E_i cobra sentido cuando el flujo de tráfico está caracterizado fuertemente por el tránsito de vehículos pesados. Porque una asignación de fase de pocos segundos haría que los vehículos pesados continúen en cola por más tiempo.

Cuando el factor de índice (E_i) de dos o más flujos de tráfico es el mismo, y compiten por obtener la fase, el tiempo de espera promedio (TEPV) determinará qué flujo debe programarse en primer lugar. Por tanto, el flujo de tráfico que consiga el mayor TEPV debe ser programado en primer lugar (Younes et al., 2023).

3.8.3.14 Fórmula del factor de índice

La fórmula del factor de índice es la siguiente:

$$E = \sum_{i=1}^{N_{VG}} V_{mov\ i} \times W_{mov\ i} \quad (30)$$

Donde:

E : Factor de índice

$V_{mov\ i}$: Velocidad de vehículo en movimiento

$W_{mov\ i}$: Peso del vehículo en movimiento

N_{VG} : Número de vehículos del grupo

3.9 Algoritmo para flujos no conflictivos

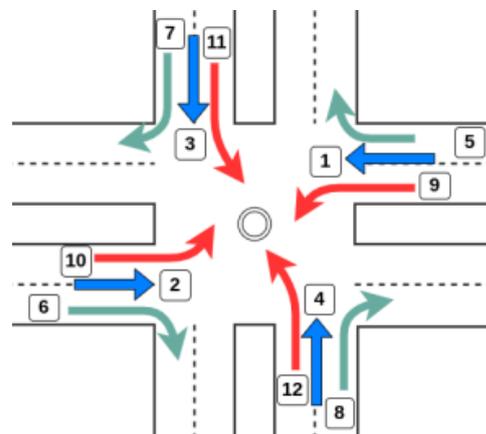
Para elaborar el algoritmo de flujos no conflictivos se ha elaborado el cuadro de flujos pares no conflictivos para que se pueda aplicar en el momento de tomar decisiones.

De acuerdo a la Figura 21, Pares de flujos no conflictivos (FNC), se tiene el siguiente cuadro, considerando que los flujos que va a la derecha no generan conflicto en dicho esquema.

Figura 21

Pares de flujos no conflictivos

PAR	Par1	Par2	Par3	Par4	Par5	Par6	Par7	Par8
Flujos	(9,10)	(9,1)	(2,10)	(2,1)	(12,11)	(12,4)	(3,11)	(3,4)
MODO	3	1	1	2	3	1	1	2
PAR	NO	Par3	Par2	NO	NO	Par7	Par6	NO



MODO 1: se emplea en situaciones de cualquier tipo de congestión

MODO 2: se emplea cuando se detecta congestión extrema y la velocidad media ha caído cercano a cero (Prohibir Verde a la Izquierda)

MODO 3: sólo si es imposible que los flujos lineales atraviesen la intersección.

Por ejemplo:

El flujo 9 y el flujo 1 están de MODO PAR = 1 cuyo flujo par simultáneo es el Par3, o sea los flujos 2 y 10.

El flujo 9 y flujo 10 son flujo par no conflictivos de MODO PAR = 3 que no tiene PAR SIMULTÁNEO

Descripción del algoritmo 3.1:

- Se define espacios para el número de flujos de la intersección
- Se asigna a una estructura de datos del siguiente tipo
Estructura [Flujo A, Flujo B, MODO]
- Mediante un recorrido se puede listar la estructura

Algoritmo Nro. 3.1: Gestión FNC

```
1 //Nombre:Algoritmo de flujos no conflictivos FNC
2 Funcion flujosNoConflictivos()
3   FNC = [[0,0,0],[0,0,0],[0,0,0],[0,0,0],[0,0,0],[0,0,0],[0,0,0],[0,0,0]]
4   Funcion flujo(k)
5     Si(k <= 4):
6       Si ( k == 1)
7         FNC[k-1][0], FNC[k-1][1], FNC[k-1][2] = 9, 10, 3
8         FNC[k][0], FNC[k][1], FNC[k][2] = 9, 1, 1
9       Sino k == 2
10        FNC[k][0], FNC[k][1], FNC[k][2] = 2, 10, 1
11        FNC[k+1][0], FNC[k+1][1], FNC[k+1][2] = 2, 1, 2
12      Sino k == 3
13        FNC[k+1][0], FNC[k+1][1], FNC[k+1][2] = 12, 11, 3
14        FNC[k+2][0], FNC[k+2][1], FNC[k+2][2] = 12, 4, 1
15      Sino k == 4
16        FNC[k+2][0], FNC[k+2][1], FNC[k+2][2] = 3, 11, 1
17        FNC[k+3][0], FNC[k+3][1], FNC[k+3][2] = 3, 4, 2
18      retornar FNC
19   FinFuncion
20   Para x=1 hasta 4
21     //Crear los FNC
22     flujo(x)
23     Imprimir ("k=",x," ",flujo(x))
24     Imprimir (FNC)
25   FinPara
26 FinFuncion
27 Llamar a flujosNoConflictivos()
```

3.10 Algoritmo de inicio y configuración

Descripción del algoritmo 3.2:

Este algoritmo es iterativo secuencial, y se utiliza para:

- Leer el largo y ancho de cada tramo de intersección así como el número de carriles (lane).

- Leer el espaciamiento longitudinal y lateral de vehículos.
- En base a largo y ancho determina el área de cada tramos de intersección.
- En base al número máximo de vehículos por tramo se determina la densidad máxima

Algoritmo Nro. 3.2: Gestión del Inicio y configuración

```

1  funcion iniciar()
2      //l:largo,a:ancho:a de cada tramo de intersección
3      //lane: nro de carril de tramo,At: área de tramo
4      n = 4
5      Dimensionar largo[n],ancho[n],lane[n],At[n],MaxVeh[n],Dmax[n]
6      Para i<-1 Hasta n Con Paso 1 Hacer
7          Leer largo[i]
8          Leer ancho[i]
9          Leer lane[i]
10     Fin Para
11     //espaciamientos entre vehículos
12         Leer EspaLong      //espaciamiento longitudinal
13         Leer EspaLate      //espaciamiento lateral
14         //
15     //Determinar el área de tramo
16     Para i<-1 Hasta n Con Paso 1 Hacer
17         At[i]=largo[i]*ancho[i]
18     Fin Para
19     AV = EspaLong * EspaLate    // Area de tramo ocupado por vehículo
20     //Obtener el maximo Número de Vehiculos y la Densidad Maxima
21     Para i<-1 Hasta n Con Paso 1 Hacer
22         MaxVeh[i] = largo[i]/EspaLong*lane[i]//Max numero de vehiculos
23         Dmax[i] = MaxVeh[i]/At[i]    //Densidad Máxima
24     Fin Para
25 Fin

```

3.11 Algoritmo respecto al número de vehículos y densidad

Descripción del algoritmo 3.3:

- Este algoritmo se encarga de almacenar el número de vehículos que viene de la funciones de lectura de cámara.
- Permite hallar la densidad de grupo. DG
- Permite hallar el índice de congestión de tráfico ICT

- Se debe mencionar que estas funciones deben estar disponibles para sus uso de manera permanente bajo un enfoque de servicio.

Algoritmo Nro. 3.3: Operaciones con el número de vehículos y densidad

```

1 //Funciones respecto a Numero de vehiculos y densidad
2 Funcion ContarVehiculos()
3     Dimensionar numVeh[n]
4     Para i<-1 Hasta n Con Paso 1 Hacer
5         //obtener del proceso de camara i
6         numVeh[i] = procesosLecturaCamara(i)
7     Fin Para
8     retornar numVeh
9 Fin Funcion
10
11 Funcion DensidadGrupo()
12     Dimensionar dg[n]
13     nroVeh = ContarVehiculos()
14     Para i<-1 Hasta n Con Paso 1 Hacer
15         dg[i] = nroVeh[i]/At[i]
16     Fin Para
17     retornar dg
18 Fin Funcion
19
20 Funcion IndiceCongestionTRafico()
21     Dimensionar ICT[n]
22     DG = DensidadGrupo()
23     Para i<-1 Hasta n Con Paso 1 Hacer
24         ICT[i] = DG[i]/Dmax[i]
25     Fin Para
26     retornar ICT
27 FinFuncion
28 Fin

```

3.12 Algoritmo gestión de operaciones con la velocidad media

Descripción del algoritmo 3.4:

- Este algoritmo realiza la suma de velocidades de Grupo.
- Con las velocidades de grupo puede obtener la Velocidad media del grupo.
- Dependiendo del tipo de Velocidad media determina si la velocidad media es baja, moderada, alta o muy alta.

- La velocidad máxima actual permitida es de 50 Km/h, tal como se aprecia en el Anexo G (Ministerio de Transportes y Comunicaciones, 2021).
- Velocidades superiores a 50 km/h en un contexto urbano, se ha catalogado como muy alta en esta tesis.

Algoritmo Nro. 3.4: Gestión de operaciones con la velocidad media

```

1  Funcion SumVelocidadesGrupo //Funciones Numero de vehiculos y densidad
2      Dimensionar svg[n]
3      Para i<-1 Hasta n Con Paso 1 Hacer
4          //obtener del proceso de camara i
5          svg[i] = procesosVelocidadCamara(i)
6      Fin Para
7      retornar svg
8  FinFuncion
9  Funcion VelocidadMediaGrupo()
10     sumv = SumVelocidadesGrupo()
11     Para i<-1 Hasta n Con Paso 1 Hacer
12         Si numVeh[i] == 0:
13             vmg[i] = 0
14         SiNo
15             vmg[i] = sumv[i]/numVeh[i]
16     FinPara
17     retornar vmg
18 FinFuncion
19 Funcion TipoVelocidadMedia()
20     tvn = [[1,None],[1,None],[1,None],[1,None]]
21     j=0
22     Mientras j<=3:
23         Si vmg[j] >= 0 and vmg[j] < 10
24             tvn[j][0] = 1
25             tvn[j][1] = "Vel. Baja"
26         Sino vmg[j] >= 10 and vmg[j] < 30
27             tvn[j][0] = 2
28             tvn[j][1] = "Vel. Moderada"
29         Sino vmg[j] >= 30 and vmg[j] < 50
30             tvn[j][0] = 3
31             tvn[j][1] = "Vel. Alta"
32         Sino
33             tvn[j][0] = 4
34             tvn[j][1] = "Vel. muy Alta"
35         j += 1
36     retornar tvn
37 FinFuncion

```

3.13 Algoritmo para obtener el tipo de congestión

Descripción del algoritmo 3.5: este algoritmo hace uso del índice control de tráfico (ICT), obtenido del algoritmo 3.3 y cubre 5 posibilidades de congestión más específicas a diferencia de lo mencionado en la literatura en que solo considera 3 tipos (Ashifuddin Mondal & Rehena, 2019).

La estructura de datos de tipo de congestión es de la siguiente forma:

Estructura [[ID TIPO, NOMBRE DE CONGESTIÓN, TIEMPO MAX], ...]

Algoritmo Nro. 3.5: Tipo de congestión

```
1 Funcion TipoCongestion()
2   tc = [[1,None,0],[1,None,0],[1,None,0],[1,None,0]]
3   i=0
4   Mientras i<=3
5       Si ICT[i]>=0 and ICT[i]< 0,25
6           tc[i][0] = 1 //Tipo 1- Congestión Muy Bajo
7           tc[i][1]= "Congestión muy Bajo"
8           tc[i][2]= 15
9
10          SiNo ICT[i]>=0,25 and ICT[i]< 0,5
11              tc[i][0]= 2 //Tipo 2- Congestión Bajo
12              tc[i][1] = "Congestión Bajo"
13              tc[i][2]= 25
14
15          SiNo ICT[i]>=0,5 and ICT[i]< 0,75
16              tc[i][0]= 3 //Tipo 3- Congestión Medio
17              tc[i][1] = "Congestión Medio"
18              tc[i][2]= 35
19
20          SiNo ICT[i]>=0,75 and ICT[i]< 0,90
21              tc[i][0]= 4 //Tipo 4- Congestión Alta
22              tc[i][1] = "Congestión Alto"
23              tc[i][2]= 45
24
25          SiNo
26              tc[i][0]= 5 //Tipo5 -Congestión muy Alta
27              tc[i][1] = "Congestión muy Alto"
28              tc[i][2]= 55
29          i = i+1
30   return tc
31 FinFuncion
```

3.14 Algoritmo de conteo regresivo

Descripción del algoritmo 3.6: este algoritmo permite visualizar el tiempo de verde, amarillo o rojo de acuerdo a cómo la función sea invocada. Cada lapso de tiempo es de 1 segundo y va desde el valor n , $n-1$, $n-2$ hasta $n=1$.

Algoritmo Nro. 3.6: Conteo de tiempo regresivo

```
1  Función TiempoRegresivoVerde(n)
2      Mientras n <> 0:
3          imprimir("COLOR VERDE")
4          time.sleep(1segundo)
5          n -= 1
6  FinFuncion
7  Funcion TiempoRegresivoAmbar(n)
8      Mientras n <> 0:
9          imprimir("COLOR ÁMBAR")
10         time.sleep(1segundo)
11         n -= 1
12  FinFuncion
13  Funcion TiempoRegresivoRojor(n)
14      Mientras n <> 0:
15         imprimir("COLOR ROJO")
16         time.sleep(1segundo)
17         n -= 1
18  FinFuncion
```

3.15 Algoritmo del semáforo

Descripción del algoritmo 3.7:

- El algoritmo del semáforo permanentemente está recibiendo datos del entorno a fin de ajustar el mejor tiempo de programación de fase de semáforo al tramo actual.
- El algoritmo conoce su Fase Actual, o sea a qué tramo se le ha asignado FASE VERDE.
- El Algoritmo conoce cuál es el índice de congestión de tráfico actual y cuál es la velocidad media del tramo de cada flujo de tráfico.
- Es capaz de reajustarse aumentando o disminuyendo segundos adicionales en caso la velocidad media se dispare o caiga y en caso la densidad también se dispare a caiga.

- De manera paralela y asíncrona, se debe monitorizar, la ocurrencia de cambios de estado.

Algoritmo Nro. 3.7: Gestión del semáforo

```

1 Funcion semaforo_inteligente()
2   f=0
3   FASE = 1 //El sistema inicia su funcionamiento
4
5   Mientras True:
6       //Monitorizar entorno
7       IndConTraf      = IndiceCongestionTRafico()
8       TipoCon  = TipoCongestion()
9       VelMedGr      = VelocidadMediaGrupo()
10      TipoVelMed      = TipoVelocidadMedia()
11      //Monitorizar tramo Sur-Norte-Norte-Sur
12      faseActual = FASE
13      Si faseActual == 1
14          if IndConTraf[0] > IndConTraf[1]
15              posMayorTraficFase = 0
16              tramo="Sur-Norte"
17          Sino
18              posMayorTraficFase = 1
19              tramo="Norte-Sur"
20      //Monitorizar tramo Este-Oeste-Oeste-Este
21      Si faseActual == 2
22          Si IndConTraf[2] > IndConTraf[3]
23              posMayorTraficFase = 2
24              tramo="Este-Oeste"
25          SiNO
26              posMayorTraficFase = 3
27              tramo="Oeste-Este"
28      TipoTraf = TipoCon[posMayorTraficFase]
29      tinicio = time.time()
30
31      //Algoritmo de Reajuste: Incrementar tiempo de verde para Tipos de congestión
32      //Si densidad es alta y velocidad promedio es muy baja, aumentar tiempo verde 5s
33      Si TipoCon[posMayorTraficFase][0] == 4 and TipoVelMed[posMayorTraficFase][0] == 1
34          tiempoAdicional = 3
35      //Si densidad es muy alta y velocidad promedio es muy baja, aumentar tiempo verde 10s
36      SiNo TipoCon[posMayorTraficFase][0] == 5 and TipoVelMed[posMayorTraficFase][0] == 1
37          tiempoAdicional = 5
38      //Si densidad es muy baja y velocidad promedio es muy alta, reducir tiempo verde -5s
39
40      SiNo TipoCon[posMayorTraficFase][0] == 1 and TipoVelMed[posMayorTraficFase][0] == 5
41          tiempoAdicional = -5
42      //Si densidad es baja y la velocidad promedio es muy alta, reducir el tiempo verde -5s
43      SiNo TipoCon[posMayorTraficFase][0] == 2 and TipoVelMed[posMayorTraficFase][0] == 5:
44          tiempoAdicional = -5
45      //Se mantiene la duración de verde y no se agrega tiempoAdicional
46      SiNo
47          tiempoAdicional = 0
48
49      TiempoRegresivoVerde(TipoCon[posMayorTraficFase][2] + tiempoAdicional)
50      tfinal = time.time()

```

```

51 tiempoTranscurridoVerde(tfinal-tinicio)
52 TiempoRegresivoAmbar(3)
53 tfinal = time.time()
54 tiempoTranscurridoAmbar(tfinal-tinicio)
55 Si FASE == 1
56     FASE = 2
57 SiNo
58     FASE = 1
59 Imprimir ("Fin de Fase")

```

FinFuncion

3.16 Algoritmo de predicción de estado con Cadena de Markov

Descripción: Primero se define las probabilidades de estado, ver Figura 22, del semáforo para ello se realizó lo siguiente:

- Se elaboró la Matriz de transición de Markov
- Se consideró que las densidades se miden en términos muyBajo, bajo, medio, alto y muy alto
- La probabilidad P de pasar de pasar de verde a amarillo es:
 - $P(\text{Verde a Amarillo}) = 0,1$
- La probabilidad P de pasar de amarillo a rojo es:
 - $P(\text{Amarillo a Rojo}) = 0,2$
- La probabilidad P de pasar de rojo a verde es:
 - $P(\text{Rojo a Verde}) = 0,2$
- Las demás probabilidades se completan, de manera que la suma sea 1

Figura 22

Diagrama de estados de Markov para intercambios de color

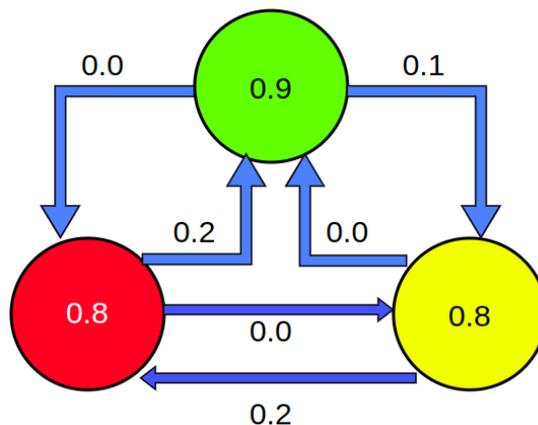


Tabla 12*Matriz de transición M1 de la cadena de Markov*

Color	Verde	Amarillo	Rojo
Verde	0,9	0,1	0,0
Amarillo	0,0	0,8	0,2
Rojo	0,2	0,0	0,8

Nota: Los valores numéricos en la matriz representan la probabilidad de transición entre colores

En la Tabla 12 verificamos que la matriz de transición de probabilidades cumpla:

1. Que debe de ser cuadrada (en este caso es de 3x3)
2. Los valores tienen que estar entre cero y uno
3. La suma de cada fila debe de ser uno

Algoritmo Nro. 3.8: Secuencia de estado con la cadena de Markov

```

1 estados = ['VERDE', 'AMARILLO', 'ROJO'] // Definir los estados del semáforo inteligente
2 Imprimir ('Matriz de transición basada en la densidad vehicular')
3 Dimensionar matriz_transicion[3,3]
4 matriz_transicion = ([
5     [0,9, 0,1, 0,0], // Probabilidad de pasar de Verde a Amarillo: 0,1
6     [0,0, 0,8, 0,2], // Probabilidad de pasar de Amarillo a Rojo: 0,2
7     [0,2, 0,0, 0,8] // Probabilidad de pasar de Rojo a Verde: 0,2
8 ])
9 estado_actual = 0 //Estado inicial (comienza en verde)
10 num_iteraciones = 30//Número de iteraciones
11 // Ver Algoritmo 5 para densidad: muyBajo,bajos, medio, alto y muy alto
12 densidades = ['MuyBajo','Bajo', 'Medio', 'Alto','MuyAlto']
13 //Simular la cadena de Markov basada en la densidad vehicular
14 historial_estados = [estado_actual]
15 historial_densidad = [0]
16 Para i<-1 Hasta num_iteraciones Con Paso 1 Hacer
17     // Seleccionar la densidad vehicular actual de manera aleatoria
18     densidad_actual = np.random.choice(range(len(densidades)))
19     // Ajustar la matriz de transición según la densidad vehicular
20     matriz_transicion_ajustada = matriz_transicion * (0,5 + densidad_actual * 0,5)
21     // Normalizar para asegurar que las probabilidades sumen 1
22     matriz_transicion_ajustada = matriz_transicion_ajustada / suma_prob_de_filas
23     //Elegir el próximo estado
24     estado_actual = Aleatorio(estados, p=matriz_transicion_ajustada[estado_actual])
25
26     historial_estados = agregar(estado_actual)//Corresponde al eje Y
27 FinPara

```

En el Anexo H, se puede observar parte del proceso iterativo correspondiente, donde a partir de la iteración 40 la matriz se estabiliza.

3.17 Algoritmo de Cadena de Markov para los tiempos de Espera

Descripción:

Dada la matriz de transición, formada por los estados:

VERDE_1 = 0 # Verde para dirección Norte-Sur, Sur-Norte

VERDE_2 = 1 # Verde para dirección Este-Oeste, Oeste-Este

AMARILLO_1 = 2 # Amarillo para dirección Norte-Sur, Sur-Norte

AMARILLO_2 = 3 # Amarillo para dirección Este-Oeste, Oeste-Este

ROJO_1 = 4 # Rojo para dirección Norte-Sur, Sur-Norte

ROJO_2 = 5 # Rojo para dirección Este-Oeste, Oeste-Este

En este caso para la predicción de tiempos de espera, se requiere la matriz de transición de la Tabla 13.

Tabla 13

Matriz de transición para predicción de tiempos de espera

Color	Verde1	Verde2	Amarillo1	Amarillo2	Rojo1	Rojo2
Verde1	0,50	0,00	0,20	0,00	0,30	0,00
Verde2	0,00	0,50	0,00	0,20	0,00	0,30
Amarillo1	0,20	0,00	0,00	0,00	0,80	0,00
Amarillo2	0,00	0,20	0,00	0,00	0,00	0,80
Rojo1	0,10	0,70	0,00	0,00	0,20	0,00
Rojo2	0,70	0,10	0,00	0,00	0,00	0,20

Consideraciones:

- Definir los estados del semáforo para dos direcciones (norte-sur y viceversa, este-oeste y viceversa).

- Definir la matriz de transición con las probabilidades de cambio de estado.
- Establecer la duración de cada estado.
- Elaborar la función para generar la secuencia de estados del semáforo durante un ciclo.
- Establecer parámetros de (duración del ciclo, número de ciclos y número de iteraciones).
- Calcular los tiempos de espera promedio para cada tramo

Algoritmo Nro. 3.9: Tiempo de espera con cadena de Markov

```

1 //Definición de los estados
2 VERDE_1 = 0 // Verde para dirección Norte-Sur, Sur-Norte
3 VERDE_2 = 1 // Verde para dirección Este-Oeste, Oeste-Este
4 AMARILLO_1 = 2 // Amarillo para dirección Norte-Sur, Sur-Norte
5 AMARILLO_2 = 3 // Amarillo para dirección Este-Oeste, Oeste-Este
6 ROJO_1 = 4 // Rojo para dirección Norte-Sur, Sur-Norte
7 ROJO_2 = 5 // Rojo para dirección Este-Oeste, Oeste-Este
8 //Definir la Matriz de transición
9 matriz_transicion = [[],[],[],[],[],[]]
10 // Establecer la duración de cada estado (segundos)
11 duracion_estados = [t0, t1, t2, t3, t4, t5]
12 //Función para generar la secuencia de estados siguientes según Markov del semáforo
13 Funcion generar_secuencia(nroestados)
14     secuencia = []
15     estado_actual = VERDE_1
16     Para i=1 Hasta nroestados
17         // Generar siguiente estado
18         siguiente_estado = random.choices([VERDE_1, VERDE_2, AMARILLO_1, AMARILLO_2,
19     ROJO_1, ROJO_2],
20     matriz_transicion[estado_actual])[0]
21     secuencia.agregar(siguiente_estado)
22     estado_actual = siguiente_estado
23     FinPara
24     retornar secuencia
25 //Parámetros de iteración
26 duracion_ciclo = sumatoria(t0 hasta t5) // Tiempo en Segundos
27 numero_ciclos = 3600/duración_ciclo //ciclos aproximadamente en una 1 hora
28 nro_estados = 200 #recreación de estados bajo la Matriz de Markov
29 tiempos_espera_1 = []
30 tiempos_espera_2 = []
31 secuencia = generar_secuencia(nro_estados)
32 //Escenario para ejecución de 1 hora
33 tiempo_1 = 0

```

```

33 tiempo_2 = 0
34 Para i=1 hasta numero_ciclos
35     tiempo_1 = 0
36     tiempo_2 = 0
37     estado = secuencia[i]
38     Si estado == VERDE_1
39         tiempo_2 += duracion_estados[estado]
40     SiNo estado == VERDE_2
41         tiempo_1 += duracion_estados[estado]
42     SiNo estado in (AMARILLO_2, AMARILLO_1):
43         tiempo_1 += duracion_estados[estado] / 2
44         tiempo_2 += duracion_estados[estado] / 2
45     tiempos_espera_1.append(tiempo_1)
46     tiempos_espera_2.append(tiempo_2)
47 FinPara
48 tiempoTotal = tiempo_1 + tiempo_2
49 Mostrar secuencia, tiempoTotal
50 Mostrar tiempos_espera_1 / numero_ciclos //Tiempo promedio en segundos
51 Mostrar tiempos_espera_2 / numero_ciclos //Tiempo promedio en segundos

```

3.18 Ejecución de aplicación Semáforo

Los algoritmos para la determinación de las características de tráfico se programaron con el lenguaje Python 3.8.x, en un entorno de Anaconda, a fin de obtener la salida que se muestra en la Figura 23.

La sección características de tráfico muestran lo siguiente:

- Cuatro tramos de intersección de Sur a Norte, de Norte a Sur, de Este a Oeste y de Oeste a Este.
- El número de vehículos de cada tramo.
- El índice de congestión de tráfico expresado numéricamente
- Los tipos de congestión.
- Suma de velocidades
- Velocidad Media de grupo
- Los tipos de velocidad media

La sección datos actuales muestran lo siguiente:

Fase, tramo, tipo de congestión y velocidad media actual

El tiempo en ROJO es el tiempo de fase del tramo transversal

La sección datos enviados a la nube muestran lo siguiente:

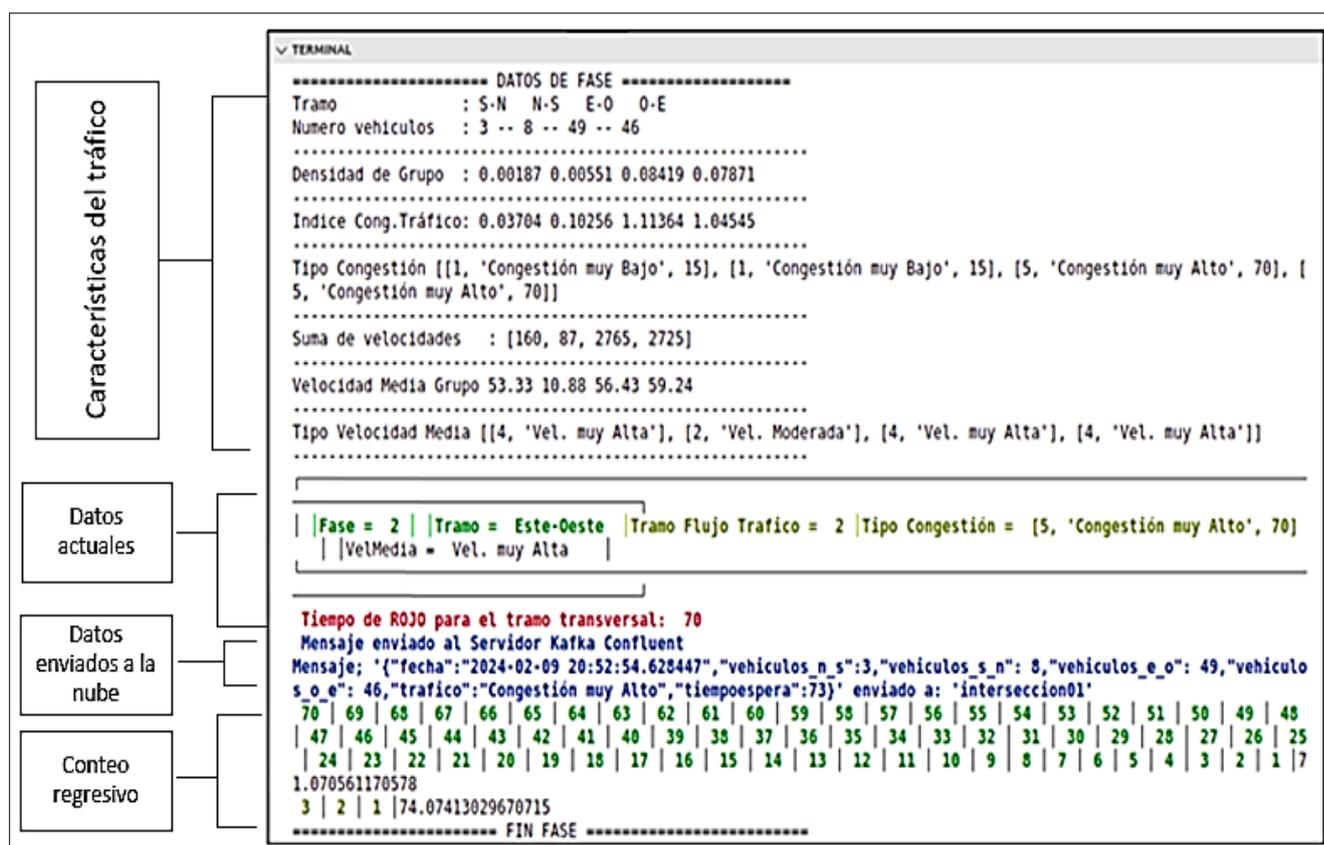
Se construye un archivo en formato json, donde va la fecha y hora del instante del envío, el número de vehículos de los cuatro tramos de intersección el tipo de tráfico, y el tiempo de espera. Se debe mencionar que el tópicos del Servidor de Kafka usado se denomina: interseccion01

La sección conteo regresivo muestran lo siguiente:

- El tiempo regresivo de fase
- EL tiempo regresivo de amarillo
- El tiempo total transcurrido de verde y amarillo

Figura 23

Ejecución del programa de Control de Semáforo



Nota: Hay en promedio un segundo de latencia de respuesta en la ejecución total del algoritmo.

3.19 Despliegue de componentes

Para poder desplegar la arquitectura propuesta (ver Figura 24 y 25) se consideró lo siguiente:

Sensores de cámaras: contiene a los artefactos que hacen posible traer los datos del entorno (de la intersección) como son número de vehículos y velocidad de vehículos, dichos artefactos están contruidos en Python y estrictamente deben ejecutarse con GPU para ejecutarse en Tiempo Real.

Servidor semáforo: contiene los artefactos que aplican los algoritmos relaciona a la gestión de semáforo y el artefacto encargado de enviar los datos al servidor Kafka.

Para que se realice el envío de datos al Servidor Kafka, se debe disponer las credenciales de Kafka respectivas como:

- La dirección del servidor y el número de puerto del servidor
- El protocolo de seguridad
- Id del cliente
- Nombre de usuario
- Contraseña con SSL
- El Nombre del Tópico creado en dicho servidor

El Servidor Web: tiene dos componentes: Un servidor web en el puerto 3000 y un servidor websocket en el puerto 3002 (ambos se ejecutaron en puertos diferente).

- El servidor web es ejecutado con NODEJS y el componente Websocket también debe ser compatible con NODEJS.

Nodo cliente: Todos los artefactos Web se construyeron en NodeJS y los artefactos de Cliente se construyeron Html5, Script.js y hoja de estilo css. De manera que los usuarios puedan visualizar los datos en tabletas, celulares y pc.

Figura 24

Diagrama de despliegue

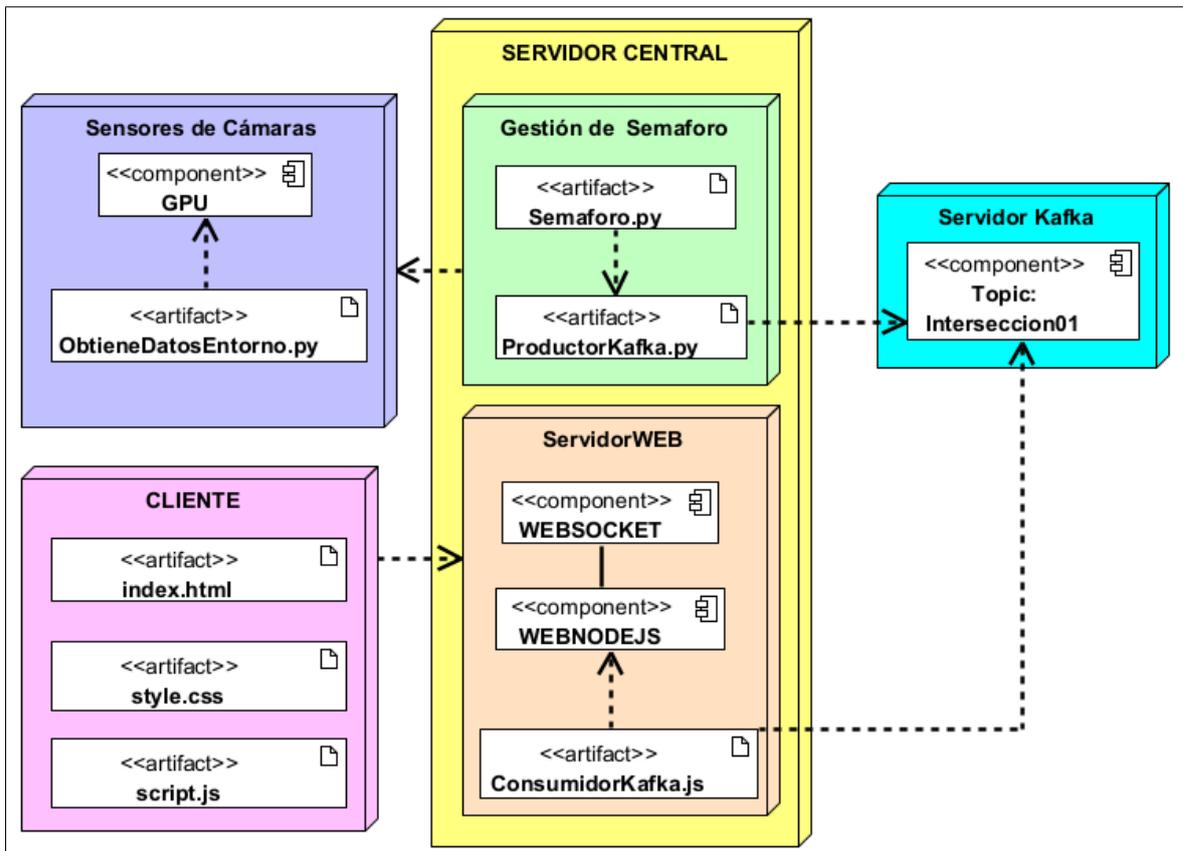
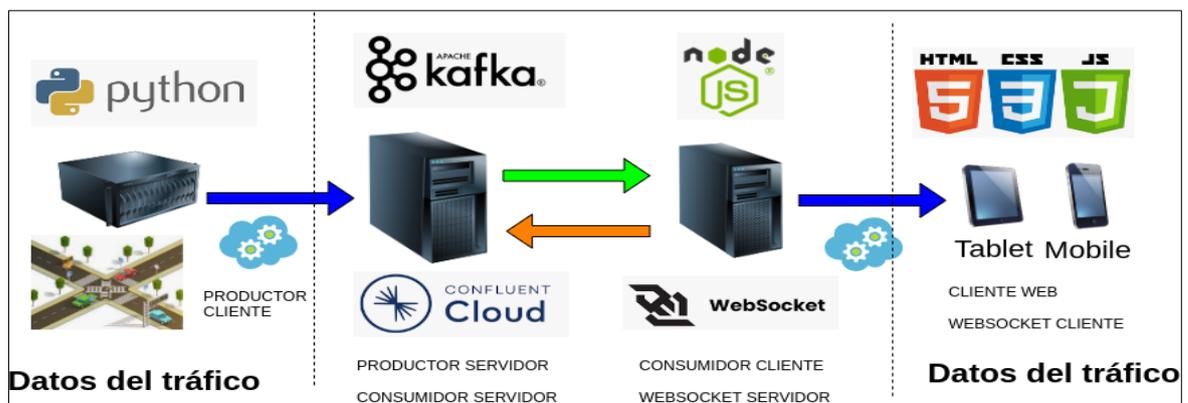


Figura 25

Infraestructura tecnológica



Nota: Los datos de tráfico llegan al cliente en tiempo real

Acciones de los componentes:

- Sistema NodeJS: iniciar el servicio con “npm start”, Inicio del Consumidor con WebSocket Server.js, iniciar cliente con Script de WS.Cliente y navegación de clientes HTML

Sistema Productor Python: Confluent Envio.py y Confluent Produce.py

- Iniciar el Sistema Gestión de Semáforos Python
- Iniciar los sensores de datos de entorno

3.20 Plataforma de gestión de Confluent Kafka

La plataforma permite configurar, mejorar el plan, realizar ajustes a diversos tipos de clientes (java, C++, Ruby) y visualizar los mensajes que llegan en tiempo real como muestra la figura 26

Se configuró Confluent Kafka con:

- 01 Tópico
- 02 particiones
- max.message.bytes = 2097164
- Retención de mensajes = una semana (versión gratuita)

Figura 26

Plataforma Confluent Kafka

The screenshot displays the Confluent Kafka Message details interface. On the left, a list of messages is shown with their values. On the right, a detailed view of a selected message is displayed, including its timestamp, offset, and partition. The message value is a JSON object containing traffic-related data.

Timestamp	Offset	Partition
2/9/2024, 8:54:08 PM (1707530048703)	428	0

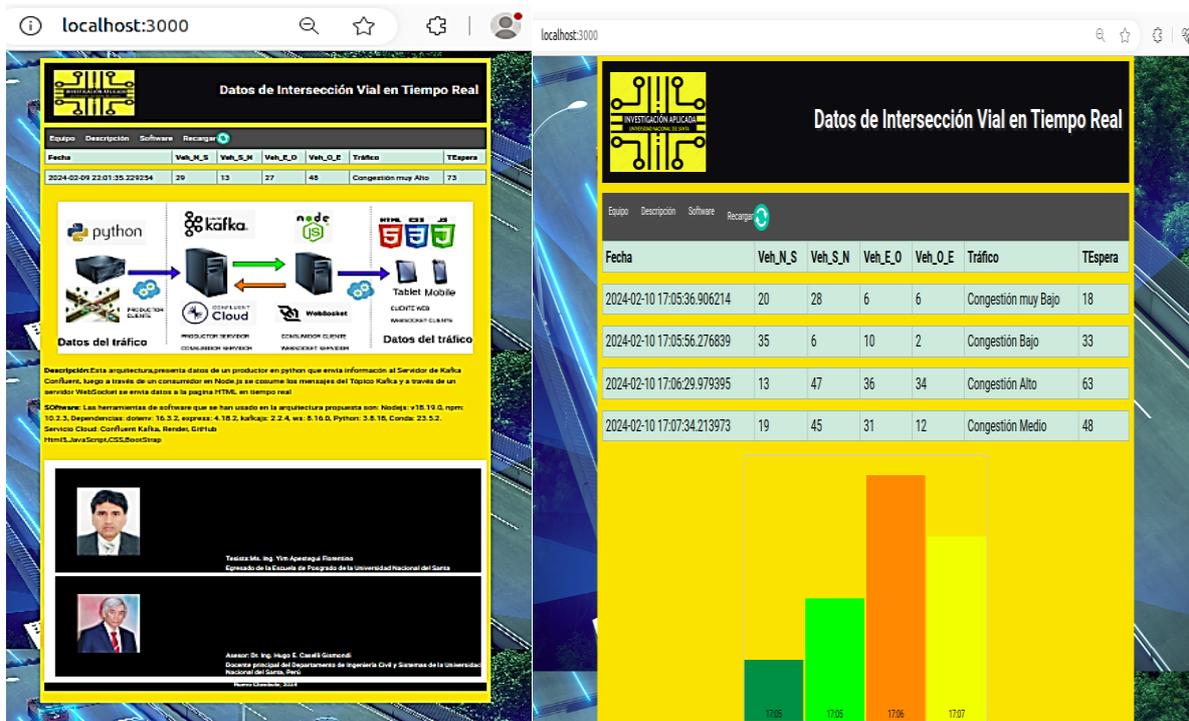
Key	Value	Headers
	<pre>{ "fecha": "2024-02-09 20:54:08.703007", "vehiculos_n_s": 9, "vehiculos_s_n": 39, "vehiculos_e_o": 9, "vehiculos_o_e": 26, "trafico": "Congestión Medio", "tiempoespera": 48 }</pre>	

3.21 Aplicación cliente Web responsivo

Finalmente se construyó el componente cliente con html5, Jscript y CSS para que los usuarios puedan visualizar información en tiempo real

Figura 27

Aplicación web de datos de intersección



CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

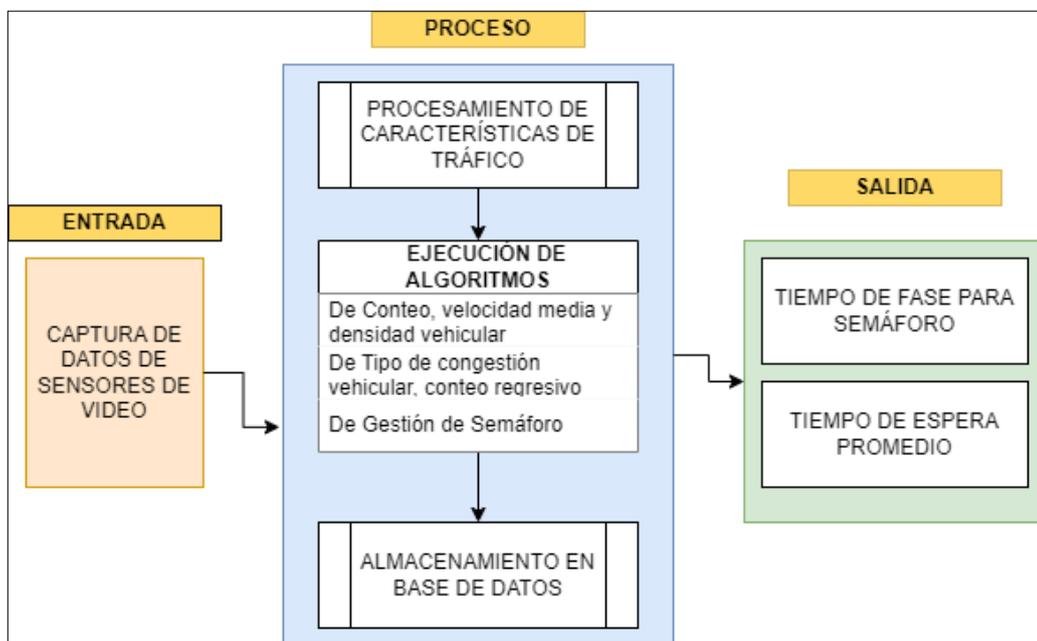
A continuación, se presenta los principales hallazgos de la investigación, para los cuales se empieza describiendo el resultado del sistema inteligente y en las siguientes secciones los resultados de cada uno de las actividades específicas que contribuyeron a lograr el resultado principal del estudio.

4.1 Desarrollo del sistema inteligente en tiempo real

El desarrollo del sistema inteligente en el contexto del objetivo general que es reducir los tiempos de espera de vehículos en una intersección vial a través de un sistema inteligente basado en un modelo heurístico, se inició con la captura de datos del entorno, luego los mismos se procesaron mediante algoritmos heurísticos y fueron enviados a una base de datos en la nube en tiempo real. Como salida del modelo se presenta los tiempos de fase y asimismo el tiempo espera promedio vehicular, tal como se puede observar el modelo arquitectónico en la siguiente Figura 28.

Figura 28

Arquitectura del modelo de Sistema Inteligente



El resultado final del sistema inteligente se observa en la reducción de los tiempos de espera de vehículos en una intersección vial.

El sistema inteligente, haciendo uso de algoritmos heurísticos para la determinación de los tiempos de colores de fase y basados en los datos del entorno, proporciona un tiempo de espera promedio de 76 segundos, lo que representa una reducción del 18,3%, tal como se puede observar en la siguiente tabla.

Tabla 14

Comparación de tiempos de espera de vehículos

Medias estadísticas no paramétricas	Sin el sistema inteligente	Con el sistema inteligente	Diferencia	Resultado
Tiempo de espera de ciclo semafórico	93 s	76 s	17 s	-18,3%
Número de ciclos / hora	38,7	47,3	9	+22,2%

Nota: El ciclo semafórico está formado por el tiempo de rojo, verde y amarillo.

La Tabla 14 nos indica que hay una diferencia de 17 s para el tiempo de espera de ciclo semafórico con el sistema inteligente, lo que indica que hay una reducción de 18,3 % respecto al enfoque tradicional, sin el sistema inteligente. Por otro lado, un incremento de 9 ciclos/h implica un mayor flujo de tráfico. Por tanto, se hace notorio la reducción significativa de los tiempos de espera de vehículos en una intersección vial, con el sistema inteligente.

4.2 Determinación de características de tráfico en intersección vial

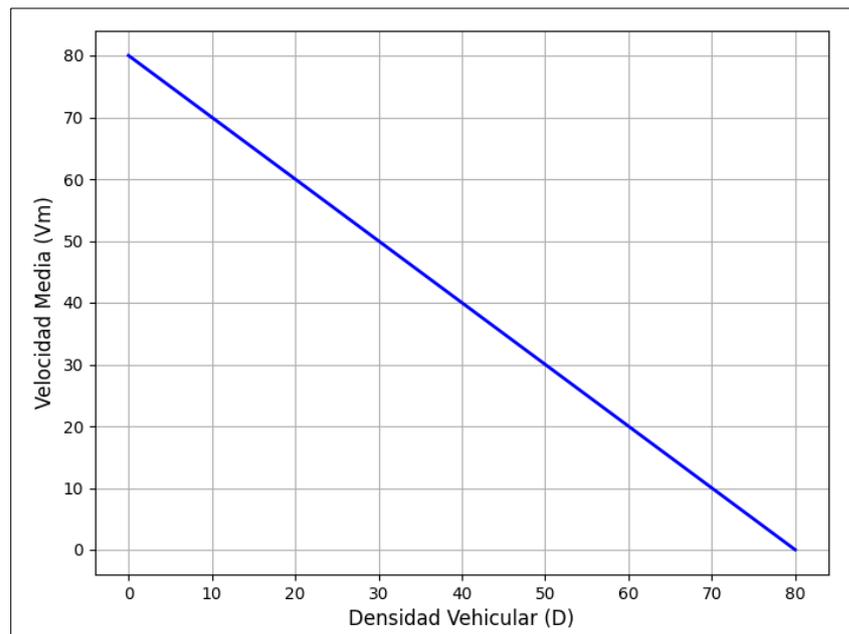
Si bien es cierto en la parte metodológica se ha descrito todo un conjunto de variables del entorno necesarios para la toma de decisiones del algoritmo, finalmente todas las ecuaciones que caracterizan al tráfico en una intersección vial nos conducen a la determinación de la densidad vehicular y la velocidad vehicular, como los parámetros de vital importancia del modelo.

La ecuación $Q = Nv / t$, que a su vez es $Q=Vm \times D$

Donde, Q es el flujo de tráfico, V_m es la velocidad media de los vehículos en la carretera y D densidad vehicular, implica que el flujo de tráfico es el producto de la velocidad media y la densidad. O sea, si la velocidad media disminuye o la densidad vehicular aumenta, el flujo de tráfico tiende a disminuir como se puede observar en la Figura 29.

Figura 29

Gráfico de la relación Velocidad Densidad



Nota: A medida que D crece V_m decrece

La relación lineal entre la densidad de tráfico y la velocidad media implica que un aumento en la densidad de tráfico llevará a un decremento en la velocidad media, tal como se observa en la Figura 29. Por lo que se puede observar que **a mayor D entonces menor V_m y a menor D entonces mayor V_m .**

Por tanto, las características resultantes fundamentales en este estudio son los parámetros son Densidad y Velocidad.

4.2.1 Respecto al factor de índice

De acuerdo a la ecuación (30), se puede observar que la función E crece en la medida que la velocidad de movimiento crece o el peso del vehículo crece.

Se realizó las pruebas con simulación para la velocidad de movimiento y peso de vehículo ligero y pesado (ver datos en Anexo E).

Donde:

Velocidad de movimiento : entre 30 y 60 km/h

Peso de vehículos ligeros : entre 1000 y 5000 kg.

Peso de vehículos pesados : entre 18000 y 52000 kg.

$E = \text{Vel} \times \text{Peso}$

Se puede observar en la Figura 30 y Figura 31 que:

Los valores de los pesos de vehículos, tanto ligeros como pesados, mantienen una tendencia lo cual no presenta demasiada variación y tiene un comportamiento no muy disperso.

A diferencia de la velocidad cuya variación hace que el factor de índice E sea creciente o decreciente.

De lo observado en el gráfico, para un entorno de tráfico urbano mayor relevancia tiene la velocidad de movimiento del vehículo, aunque dicho dato (el peso) podría considerarse a efectos de evaluar otras variables como el consumo de combustible, emisión de gases entre otros.

Figura 30

Factor de índice de vehículos livianos

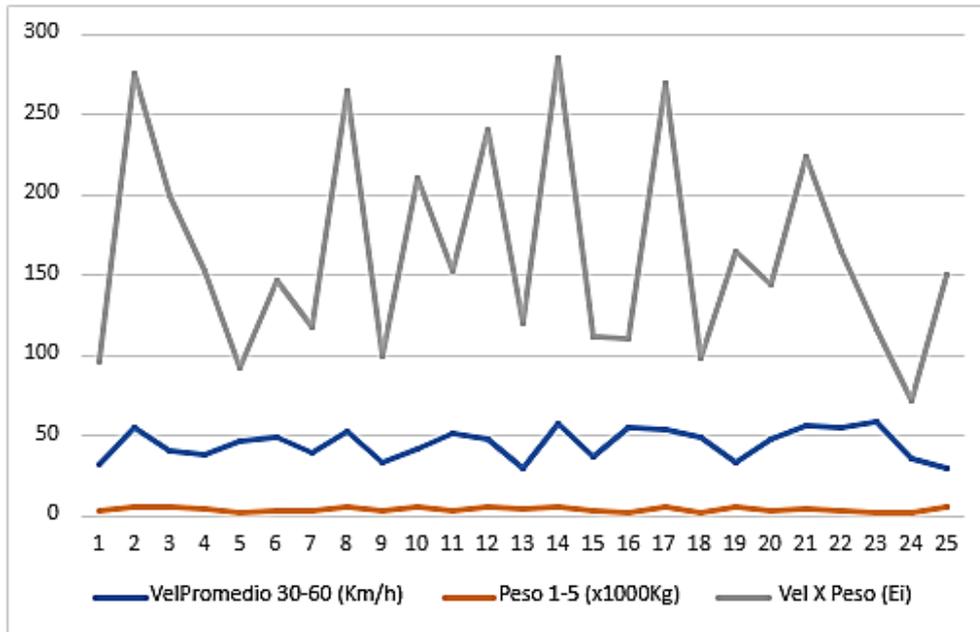
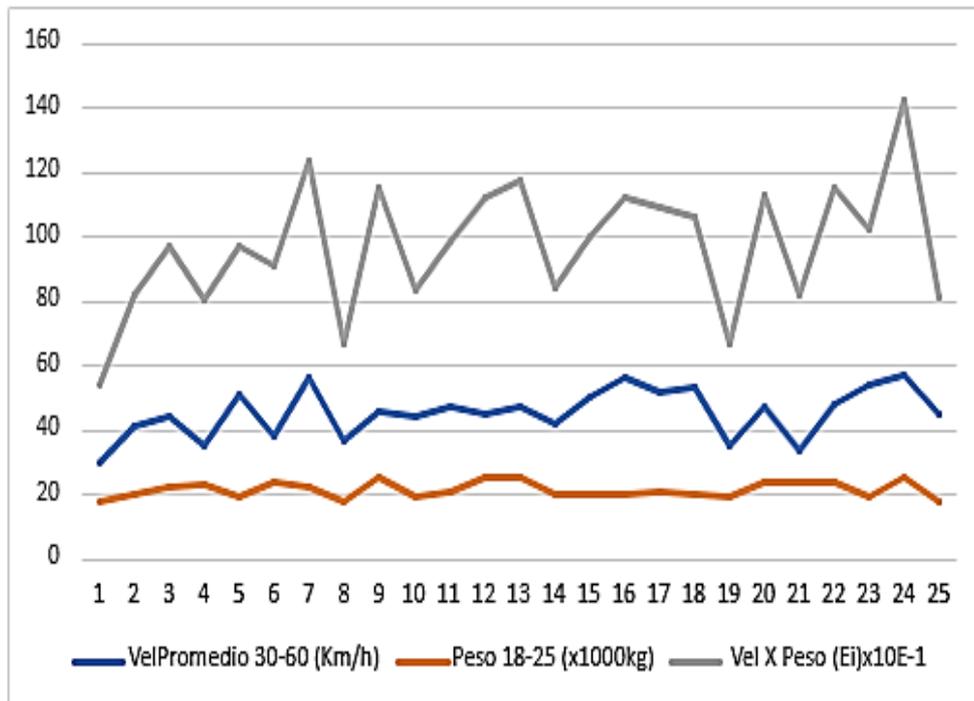


Figura 31

Factor de índice de vehículos pesados



4.2.2 Respecto a la predicción de estado basado en densidad

El tipo de densidad es un factor determinante, para la asignación de tiempos del semáforo, por lo que se hace necesario evaluar dichos cambios de estado a un nivel predictivo, en este caso con cadenas de Markov

Al realizar los procesos iterativos a partir de la Matriz de transición M1, en cuanto a la predicción de estado se observa que en la iteración nro. M40, la matriz se estabiliza, lo cual indica que los estados VERDE, AMARILLO, Y ROJO, tienen valores de probabilidad que ya no cambiarán de manera significativa:

En M1 $P(\text{Verde a Amarillo}) = 0,1$, en M40 $P(\text{Verde a Amarillo}) = 0,25$

En M1 $P(\text{Amarillo a Rojo}) = 0,2$, en M40 $P(\text{Amarillo a Rojo}) = 0,25$

En M1 $P(\text{Rojo a Verde}) = 0,2$, en M40 $P(\text{Rojo a Verde}) = 0,5$

Tabla 15

Matriz de transición estabilizada

Iteración	Color	Verde	Amarillo	Rojo
M40	VERDE	0,500009356	0,249999843	0,249990801
	AMARILLO	0,499981603	0,250009513	0,250008884
	ROJO	0,499999686	0,249990801	0,250009513
M41	VERDE	0,500006581	0,25000081	0,24999261
	AMARILLO	0,499985219	0,250005771	0,25000901
	ROJO	0,50000162	0,24999261	0,250005771

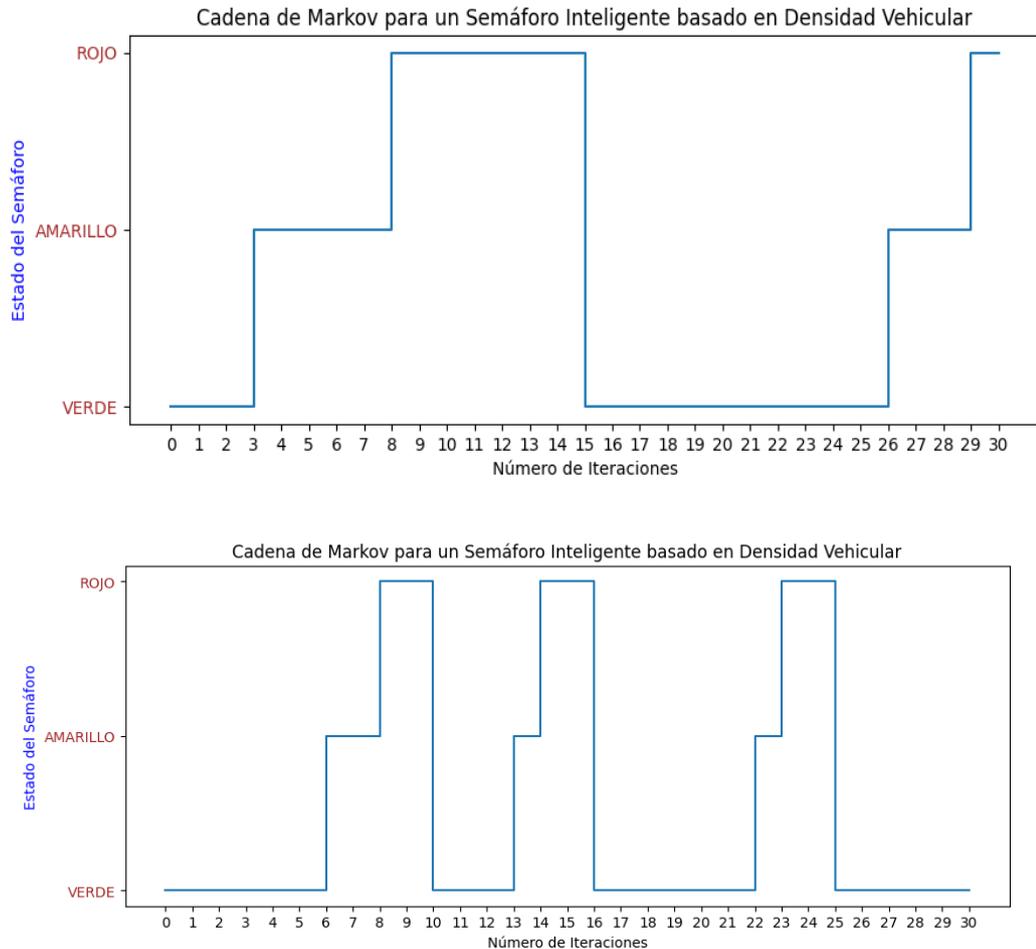
Al observar la iteración M41, de la Tabla 15, nos encontramos frente a una matriz estabilizada por lo que el algoritmo de cambios de estados de Markov debe detenerse pues las siguientes iteraciones ya no aportarán cambios significativos.

Por tanto, de Verde a Amarillo hay un 25% de probabilidad de cambio de estado, de Amarillo a Rojo un 25% y de Rojo a Verde un 50%. Y esos valores permanecerán a menos que ocurra un nuevo escenario en el que las

densidades en los tramos vehiculares hagan que los valores de la matriz de transición cambien (ver Figura 32).

Figura 32

Secuencia de predicción de estados según la cadena de Markov



Como se puede observar, en la simulación de predicciones para 30 iteraciones aplicando la Cadena de Márkov refleja que la secuencia de estados es tal cual se había definido. Inicia en un estado VERDE, luego el estado siguiente debe ser AMARILLO y luego el estado ROJO, luego nuevamente al estado VERDE, y así sucesivamente.

No se observa, alteraciones como por ejemplo que de un Rojo pase a un Amarillo o que de un Amarillo regrese a un Verde, que probabilísticamente podría suceder, pero eso dependerá del modelo de matriz de transición reajustado.

Por lo que el modelo de Markov nos presenta una probabilidad de cambio de estado coherente en función de la densidad.

4.2.3 Respecto a la predicción de estados hipotéticos

En un entorno de decisiones de semáforo inteligente se puede presentar variados casos hipotéticos como:

- En condiciones de poco tráfico, el semáforo tiende a permanecer más tiempo en verde.
- En horas pico, el semáforo para un tramo puede permanecer más tiempo en rojo por el tiempo adicional asignado (ver Algoritmo 3.7).
- Por otro lado, el tiempo en amarillo, aunque es breve, tiene una baja probabilidad de regresar al estado verde, pero finalmente es probable.
- Por ejemplo, si el tiempo de amarillo en un semáforo es 3 segundos y en ese tiempo la mayoría de los semáforos deben cambiar a rojo entonces se debe establecer una probabilidad alta de cambio, como de 0,8 o más.
- Por ello es necesario verificar qué resultados tendríamos frente a estos casos probabilísticos.

Partiendo de la siguiente matriz de transición, que se encuentra en la Tabla 16, en la cual la transición de amarillo a rojo es 0,8, se puede modelar el comportamiento de este caso hipotético.

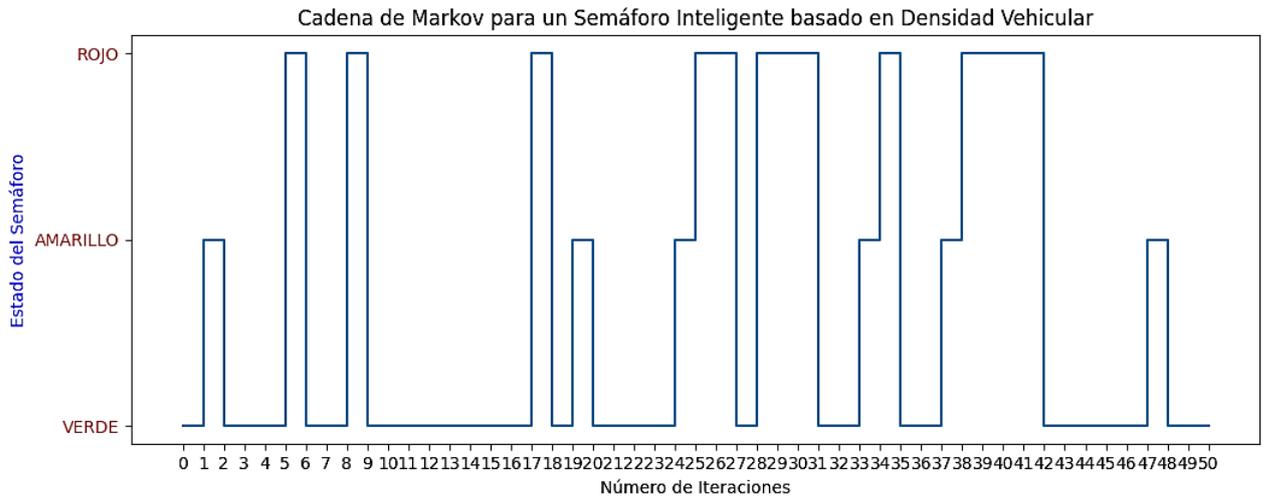
Tabla 16

Matriz de transición para estados hipotéticos

Color	Verde	Amarillo	Rojo
Verde	0,70	0,20	0,10
Amarillo	0,20	0,00	0,80
Rojo	0,25	0,00	0,75

Figura 33

Secuencia de predicción de estados según estados hipotéticos



Dada la matriz de transición, se puede verificar en la ejecución del Algoritmo 3.8, cómo en las iteraciones 1 y 2 el color de amarillo pasa a verde, situación de baja probabilidad, lo mismo sucede en las iteraciones 19-20 y 47 – 48. Ver figura 33.

Tabla 17

Matriz de transición estabilizada para estados hipotéticos

Iteración	Color	Verde	Amarillo	Rojo
M21	VERDE	0,446428582	0,089285719	0,464285699
	AMARILLO	0,446428561	0,08928571	0,464285729
	ROJO	0,446428564	0,089285711	0,464285726
M22	VERDE	0,446428576	0,089285716	0,464285708
	AMARILLO	0,446428567	0,089285712	0,464285721
	ROJO	0,446428568	0,089285713	0,464285719

Por otro lado, en la Tabla 17, se observa que al realizar los procesos iterativos a partir de la Matriz de transición M1, en la iteración nro. M21, M22, se observa que la matriz comienza a estabilizarse, lo cual nos indica que a

partir de dichas matrices estados VERDE, AMARILLO, Y ROJO, tienen valores de probabilidad que ya no cambiarán de manera significativa.

En M1 $P(\text{Verde a Amarillo}) = 0,2$, en M22 $P(\text{Verde a Amarillo}) = 0,1$

En M1 $P(\text{Amarillo a Rojo}) = 0,8$, en M22 $P(\text{Amarillo a Rojo}) = 0,46$

En M1 $P(\text{Rojo a Verde}) = 0,25$, en M22 $P(\text{Rojo a Verde}) = 0,45$

Por tanto, estos resultados son de vital importancia porque refuerza el concepto que:

- La cadena de Markov predice en función al último estado generado.
- Y que en algún momento las matrices de transición se estabilizan .
- Y que la densidad vehicular como dato que viene del entorno, genera cambios en los estados de semáforo.

4.2.4 Respecto al tipo de flujo

En la Tabla 18, podemos observar que los 12 flujos de tráfico son 12 flujos de entrada, pero que comparten algunos el mismo flujo de salida, por ejemplo: en el caso 1 los flujos 1,7 y 12 comparten el mismo flujo de salida 1; los flujos 2,8 y 11 de manera similar comparten el flujo de salida 2 y así los demás casos. Por lo que se tiene como resumen, 12 flujos de entrada, 4 flujos de salida, 8 flujos sin conflicto y 4 flujos con conflicto.

Tabla 18

Tabla de flujos de tráfico en una intersección

Caso	Flujo de entrada	Flujo de salida	Flujo sin Conflicto	Flujo con conflicto
1	1,7 y 12	1	1 y 7	12
2	2,8 y 11	2	2 y 8	11
3	3,6 y 9	3	3 y 6	9
4	4,5, y 10	4	4 y 5	10
Total	12	4	8	4

Nota: En esta tabla cada flujo de salida corresponde a cada vía en línea recta de la intersección.

Discusión: aun cuando los flujos que giran a la derecha son denominados flujos no conflictivos, desde nuestra postura no es conveniente que un flujo a la derecha con un flujo de salida simultáneamente concorra en la intersección a fin de prevenir accidentes. Por tanto la Tabla 18, solo determina los flujos de la intersección, mas no para ponerlos en fase.

4.3 Verificación de un modelo de reconocimiento y conteo

En esta sección se presenta los resultados de verificar el conteo de número de vehículos mediante un modelo de reconocimiento y conteo basado en detección de objetos Yolov4 frente a la muestra obtenida por el conteo de un observador real.

Variables:

Contador 1: Algoritmo

Contador 2: Observador

Precisión: $(\text{Algoritmo}/\text{Observador}) * 100\%$

4.3.1 Supuesto de normalidad de la variable precisión

Paso 1: Hipótesis de normalidad

H_0 : Los datos siguen una distribución normal

H_1 : Los datos no siguen una distribución normal

Paso 2: Nivel de significancia

$NC = 0,95$

$\alpha = 0,05$

Paso 3: Test de Normalidad

Si $n > 50$, se aplica Kolmogorov - Smirnov

Si $n \leq 50$, se aplica Shapiro - Wilk

Paso 4: Criterio de decisión

Si **p-valor < 0,05** se rechaza H_0

Si $p\text{-valor} \geq 0,05$, se acepta H_0 y se rechaza H_1

Tabla 19

Estadístico descriptivo de la variable precisión

Variable	Estadístico	Valor
PRECISIÓN	Media	0,9011
	95% de intervalo de confianza para la media	Límite inferior 0,8753
		Límite superior 0,9269
	Media recortada al 5%	0,9127
	Mediana	1,0000
	Varianza	0,0160
	Desviación	0,1272

Tabla 20

Pruebas de normalidad de la variable precisión

Variable	Kolmogorov-Smirnov			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
PRECISIÓN	0,292	96	0	0,782	96	0

Según los datos de la Tabla 18 y la prueba de normalidad de Kolmogorov, Tabla 20, se obtiene un $p\text{-valor} < 0,05$, por tanto se rechaza H_0 , y acepta la H_1 . O sea, los datos de la variable PRECISIÓN para el contador de vehículos no siguen una distribución normal.

Por tanto, debemos recurrir a una prueba **no paramétrica** a fin de verificar si la precisión deseada está dentro de la mediana

Tabla 21

Prueba de hipótesis del nivel de precisión de conteo de vehículos

Hipótesis nula	Prueba	Significancia	Decisión
La mediana de PRECISIÓN es igual a 0,95	Prueba de rangos con signo de Wilcoxon para una muestra	0,083	Retener hipótesis nula

Nota: El nivel de significancia es usado en la prueba es de 0,05

Según la prueba no paramétrica de WILCOXON, Tabla 21, se obtiene $p\text{-valor} \geq 0,05$, por tanto se acepta H_0 y se rechaza H_1 . O sea, la precisión de 95% se acepta para la muestra de datos.

En cuanto al nivel de precisión del contador de vehículos, se ha podido notar que la media de precisión se ubica en un 91,27% y la mediana al 100%, pero con la prueba no paramétrica de Wilcoxon, la mediana cubre un 95% de nivel de precisión.

Así, tenemos evidencia que el algoritmo de Opencv + Yolo usado para el conteo de vehículos funciona a un 95 % de nivel de precisión, muy cercanos o superiores a lo establecido en la tabla de (Dai et al., 2016).

4.3.2 Respecto a la velocidad

En esta sección se presenta los resultados de comparar los datos velocidad de vehículo mediante un algoritmo basado en seguimiento de objetos frente a la muestra obtenida por un observador real.

Variables:

Contador 1: Algoritmo

Contador 2: Observador

Diferencia = $ABS(\text{Algoritmo} - \text{Observador})$

Precisión: $(\text{Algoritmo}/\text{Observador}) * 100\%$

4.3.3 Supuesto de normalidad de la variable DIFERENCIA

Paso 1: Hipótesis de normalidad

H_0 : Los datos siguen una distribución normal

H_1 : Los datos no siguen una distribución normal

Paso 2: Nivel de significancia

$NC = 0,95$

$\alpha = 0,05$

Paso 3: Test de Normalidad

Si $n > 50$, se aplica Kolmogorov - Smirnov

Si $n \leq 50$, se aplica Shapiro - Wilk

Paso 4: Criterio de decisión

Si p-valor < 0,05, se rechaza H_0

Si p-valor $\geq 0,05$, se acepta H_0 y se rechaza H_1

Los estadísticos se muestran en la Tabla 22 a un nivel de confianza de 95%.

Tabla 22

Estadísticos descriptivos de la variable diferencia

Variable	Estadístico	Valor
DIFERENCIA	Media	1,3850
	95% de intervalo de confianza para la media	Límite inferior 0,8809
		Límite superior 1,8891
	Media recortada al 5%	1,3633
	Mediana	1,3500
	Varianza	0,8950
	Desviación	0,9460

Tabla 23

Pruebas de normalidad de la variable diferencia

Variable	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
DIFERENCIA	0,110	16	0,200*	0,952	16	0,521

Nota: *. Esto es un límite inferior de la significación verdadera, a. Corrección de significación de Lilliefors.

Según la prueba de normalidad de Shapiro-Wilk, Tabla 23, se obtiene un p-valor $\geq 0,05$, por tanto se acepta la H_0 , y se rechaza la H_1 . O sea, los datos de la variable DIFERENCIA siguen una distribución normal.

Lo que implica que en su conjunto los datos de la variable Algoritmo y Observador respecto a velocidad de vehículos siguen una distribución normal.

4.3.4 Supuesto de normalidad de la variable PRECISIÓN

Paso 1: Hipótesis de normalidad

H₀: Los datos siguen una distribución normal

H₁: Los datos no siguen una distribución normal

Paso 2: Nivel de significancia

NC = 0,95

α = 0,05

Paso 3: Test de Normalidad

Si $n > 50$, se aplica Kolmogorov - Smirnov

Si $n \leq 50$, se aplica Shapiro - Wilk

Paso 4: Criterio de decisión

Si p-valor < 0,05, se rechaza H₀

Si p-valor \geq 0,05, se acepta H₀ y se rechaza H₁

Tabla 24

Estadístico descriptivo de la variable precisión en la velocidad

Variable	Estadístico		Valor
PRECISIÓN	Media		0,9631
	95% de intervalo de confianza para la media	Límite inferior	0,9506
		Límite superior	0,9755
	Media recortada al 5%		0,9638
	Mediana		0,9585
	Varianza		0,0010
	Desviación		0,02336

Tabla 25

Pruebas de normalidad de la variable precisión en la velocidad

Variable	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadístico	gl	Sig.	Estadístico	gl	Sig.
PRECISIÓN	0,131	16	,200*	0,953	16	0,540

Nota: *. Esto es un límite inferior de la significación verdadera, a. Corrección de significación de Lilliefors.

Según los datos de la Tabla 24 y la prueba de normalidad de Shapiro-Wilk, Tabla 25, se obtiene un p-valor $\geq 0,05$, por tanto se acepta la H_0 , y se rechaza la H_1 . O sea, los datos de la variable PRECISIÓN siguen una distribución normal.

De manera que podemos realizar una prueba paramétrica T-Student denominada “Prueba T de una muestra” a fin de Verificar la media de la variable PRECISIÓN con el valor de prueba esperado igual o superior a 0,95, 0,96, 0,97, etc.

4.3.5 Prueba de una muestra: precisión

Paso 1: Hipótesis de la variable PRECISIÓN

H_0 : se acepta el valor de prueba

H_1 : no se acepta el valor de prueba

Paso 2: Nivel de significancia

NC = 0,95

$\alpha = 0,05$

Paso 3: Prueba estadística

T-Student

Paso 3: Criterio de decisión

Si p-valor < 0,05, se rechaza H_0

Si p-valor $\geq 0,05$, se acepta H_0 y se rechaza H_1

Tabla 26

Prueba de hipótesis del nivel de precisión de conteo de vehículos

Variable	N	Media	Desviación	Desviación error promedio
PRECISIÓN	16	0,9631	0,02336	0,00584

La estadística para una muestra, Tabla 26, nos indica que la media tiene un valor de 0,9631 de nivel de PRECISIÓN, lo cual es relativamente alto. Sin embargo, será necesario realizar la prueba de media para una muestra.

Tabla 27

Tabla de pruebas T para una muestra, precisión en la velocidad

VARIABLE	PRUEBA	t	gl	Sig. (bilateral)	Diferencia de medias	95% de intervalo de confianza de la diferencia	
						Inferior	Superior
PRECISIÓN	0,95	2,238	15	0,041	0,01307	0,0006	0,0255
	0,96	0,526	15	0,607	0,00307	-0,0094	0,0155
	0,97	-1,187	15	0,254	-0,00693	-0,0194	0,0055
	0,98	-2,899	15	0,011	-0,01693	-0,0294	-0,0045
	0,99	-4,611	15	0,000	-0,02693	-0,0394	-0,0145

Nota: gl, grados de libertad, observar el sig. bilateral de la prueba 0,96 y 0,97.

Según la prueba T para una muestra, Tabla 27, se puede observar que para una media de prueba de 0,95, 0,98 y 0,99 dicho valor de prueba no se acepta por cuanto su **p_valor < 0,05**. Por lo que únicamente los valores de prueba de 0,96 y 0,97 se acepta como valores de prueba por cuanto su **p_valor ≥ 0,05**.

Por tanto, se tiene evidencia que la variable PRECISIÓN tiene una media aceptable entre 0,96 y 0,97, lo cual significa que el algoritmo de seguimiento de objetos para detectar la velocidad de vehículos tiene una precisión que puede ir en una media de 96% y 97%, fuera de dicho rango no se tiene evidencia.

4.4 Desarrollo del algoritmo de obtención de características

El desarrollo del algoritmo de obtención de características del tramo vehicular está conformado por una secuencia de algoritmos, descritos en la parte metodológica que son los algoritmos 3.2, 3.3, 3.4, 3.5. En la línea de código nro. 6 del algoritmo 3.3, se menciona el uso del método proceso de lectura de cámara para contar vehículos, y en la línea de código nro. 5 del algoritmo 3.4 se menciona al método del proceso de lectura de la cámara para obtener la velocidad.

Estos métodos son los métodos que requieren mayor consumo de recursos de procesamiento como son CPU y GPU.

De las pruebas iniciales se obtuvo los resultados que se muestran en la Tabla 28, con fotogramas de entre 24 fps y 30 fps.

Tabla 28

Resultado de procesamiento y consumo de RAM del algoritmo.

Parámetro	GPU	CPU
Tiempo de procesamiento (ms)	5	250
Consumo de RAM (Gb)	2	4,5
Complejidad temporal	O(Log N)	O(N ^x)
Cualificación	Baja latencia	Alta latencia

Nota: Estos datos se obtuvieron en un equipo con memoria RAM de 8Gb, procesador Intel Core I7 1,9 GHz, GPU NVIDIA GeForce MX130, VRAM 2Gb.

De los datos mostrados en la Tabla 28, se observa que el algoritmo de obtención de características de tramo vehicular será viable en un entorno real en tanto se use GPU, debido a su baja latencia y una mejor complejidad temporal.

4.4.1 Determinación del ICT del tramo actual

Otro de los resultados importantes obtenidos en el desarrollo de esta secuencia de algoritmos es la obtención de una ecuación para la determinación del Índice de Congestión del Tráfico (ICT), donde el algoritmo se encarga de verificar en qué rango se encuentra el ICT del tramo actual y así establecer el tiempo de fase al tipo de congestión. El tiempo de fase T_{fase} se determina con la ecuación $T_{fase} = L/V_{mg}$, donde L es la longitud del tramo y V_{mg} es la velocidad media del grupo. Lo cual indica que a menor velocidad media, mayor tiempo de fase y a mayor velocidad media, menor tiempo de fase.

Considerando $L=100$ m y $V_{mg}=V_{min}=7,2$ km/h, según observación empírica, se ha obtenido la representación matemática que se muestra en la siguiente ecuación.

$$f(x) = \begin{cases} T_{max} - 5(2x + 1), Si x = 3, ICT \in [0, 0,25) \\ T_{max} - 5(2x + 1), Si x = 2, ICT \in [0,25, 0,5) \\ T_{max} - 5(2x + 1), Si x = 1, ICT \in [0,5, 0,75) \\ T_{max} - 5(2x + 1), Si x = 0, ICT \in [0,75, 0,9) \\ T_{max} - 5(2x + 1), Si x = -1, ICT \in [0,9, 1] \end{cases} \quad (31)$$

Donde:

$f(x)$ es el tiempo de fase asignado para cada tipo de congestión dado x

ICT es el Índice de Congestión de Tráfico (ICT).

T_{max} es el tiempo de fase máximo, permitido para la intersección.

Observar que, a medida que ICT crece, la función $f(x)$ incrementa su valor.

4.5 Desarrollo del algoritmo de asignación de intervalos de tiempos

El desarrollo del algoritmo de asignación de intervalos de tiempos de color está principalmente desarrollado en la secuencia de algoritmos 3.6 y 3.7 descritos en la parte metodológica, donde se monitoriza el entorno y los tramos. En cuanto al procesamiento se tiene los siguientes datos:

Tabla 29

Resultado de procesamiento de asignación de tiempos de color

Parámetro	CPU
Tiempo de procesamiento (ms)	0,05
Consumo de RAM (Mb)	aprox100
Complejidad temporal	O(Log N)
Cualificación	Latencia muy baja

Nota. Estos datos se obtuvieron en un equipo con memoria RAM de 8Gb, procesador Intel Core I7 1,9 GHz

Po lo que se entiende, de acuerdo a la Tabla 29, el comportamiento de su complejidad logarítmica es favorable, por otro lado el consumo de memoria RAM de aprox100 Mb en esta fase es aceptable y el tiempo de procesamiento es muy

veloz, bajo las consideraciones del equipo en que se hizo las ejecuciones del algoritmo. Además, para este algoritmo, no es necesario el uso de GPU, sólo se usó recursos de CPU y RAM.

El aporte sustancial de este algoritmo está en el algoritmo de reajuste, que básicamente tiene la función de balancear la relación entre tipo de densidad y la velocidad promedio, ver Tabla 30. De acuerdo al tipo de congestión se realiza el algoritmo de reajuste y por ende la asignación de intervalos de tiempos de color.

Tabla 30

Criterios para el algoritmo de reajuste de tiempos de color

Densidad (NumVeh/Tramo)	Velocidad promedio (km/h)	Reajuste tiempo de fase (s)
Alta	Muy baja	+3
Muy alta	Muy baja	+5
Muy baja	Muy alta	-5
Baja	Muy alta	-5
Otros casos	Otros casos	0

Nota: Otros casos se refieren a tipos de velocidad moderada o congestión media.

Los tiempos de reajuste han sido obtenidos en base a pruebas de simulación y múltiples procesos iterativos.

4.6 Simulación del sistema de control de semáforo

En la tabla de datos de tráfico del Anexo I se puede observar que:

- El tiempo de espera del ciclo será la suma del tiempo de espera de un tramo más el tiempo de espera del tramo transversal.
- Por tanto, se debe evaluar el tiempo de espera por ciclos frente al tiempo de espera predefinido.

4.6.1 Supuesto de normalidad de tiempo de espera

Variable: TIEMPO_ESPERA_POR_CICLO

Paso 1: Hipótesis de normalidad

H₀: Los datos siguen una distribución normal

H₁: Los datos no siguen una distribución normal

Paso 2: Nivel de significancia

NC = 0,95

$\alpha = 0,05$

Paso 3: Test de Normalidad

Si $n > 50$, se aplica Kolmogorov - Smirnov

Si $n \leq 50$, se aplica Shapiro - Wilk

Paso 4: Criterio de decisión

Si p-valor < 0,05, se rechaza H₀

Si p-valor \geq 0,05, se acepta H₀ y se rechaza H₁

Tabla 31

Prueba de normalidad de tiempo de espera por ciclo

Variable	Kolmogorov-Smirnov ^a		
	Estadístico	gl	Sig.
TIEMPO_ESPERA_POR_CICLO	0,151	7500	0,000

Nota: a. Corrección de significación de Lilliefors

Según la prueba de normalidad de Kolmogorov-Smirnov, Tabla 31, se obtiene un **p-valor < 0,05**, debido a ello se rechaza H₀, y acepta la H₁. O sea, los datos de la variable **TIEMPO_ESPERA_POR_CICLO** no siguen una distribución normal. Por lo que debemos recurrir a una prueba **no paramétrica** a fin de verificar si el tiempo esperado o deseada está dentro de la mediana

4.6.2 Prueba no paramétrica de una muestra

Tabla 32

Estadístico descriptivo para la variable tiempo de espera por ciclo

Variable	Estadístico	Valor
TIEMPO_ESPERA_POR_CICLO	Media	76,0661
	95% de límite inferior de intervalo de confianza para la media	75,7368
	Límite superior	76,3954
	Media recortada al 5%	76,5713
	Mediana	76,0000
	Varianza	211,6620
	Desviación	14,54862
	Mínimo	36,0000
	Máximo	101,0000

Prueba de Hipótesis

H₀: Los datos tienen una media igual a 76

H₁: Los datos difieren de la media de 76

Tabla 33

Prueba de hipótesis de la variable tiempo de espera por ciclo

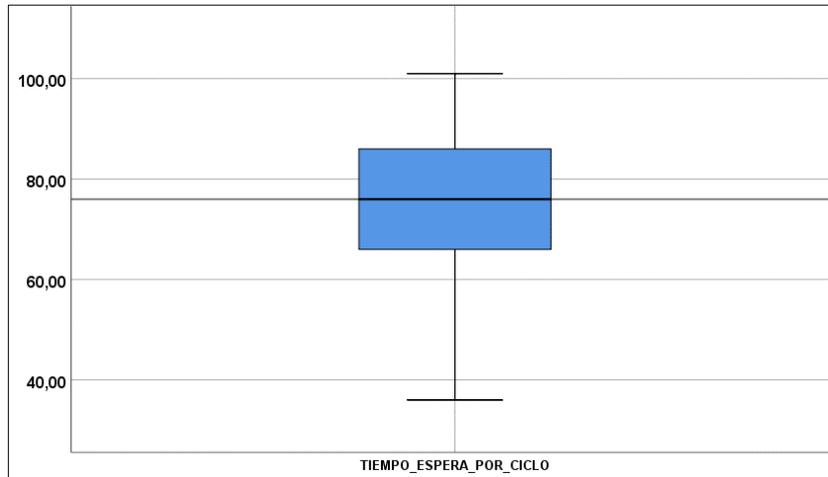
Hipótesis nula	Prueba	Sign.	Decisión
La mediana de TIEMPO_ESPERA_POR_CICLO es 76,00	Prueba de rangos con signo de Wilcoxon para una muestra	0,103	Retener hipótesis nula

Nota: Nivel de significación es 0,05

Según la prueba no paramétrica de Wilcoxon, Tabla 33, para una muestra la mediana para la variable TIEMPO_ESPERA_POR_CICLO tiene el valor de 76, reafirmando la media de 76,0661 del estadístico de la Tabla 32. Por cuanto **p-valor ≥ 0,05**, se acepta la hipótesis nula.

Figura 34

Gráfico de la mediana de tiempo de espera por ciclo

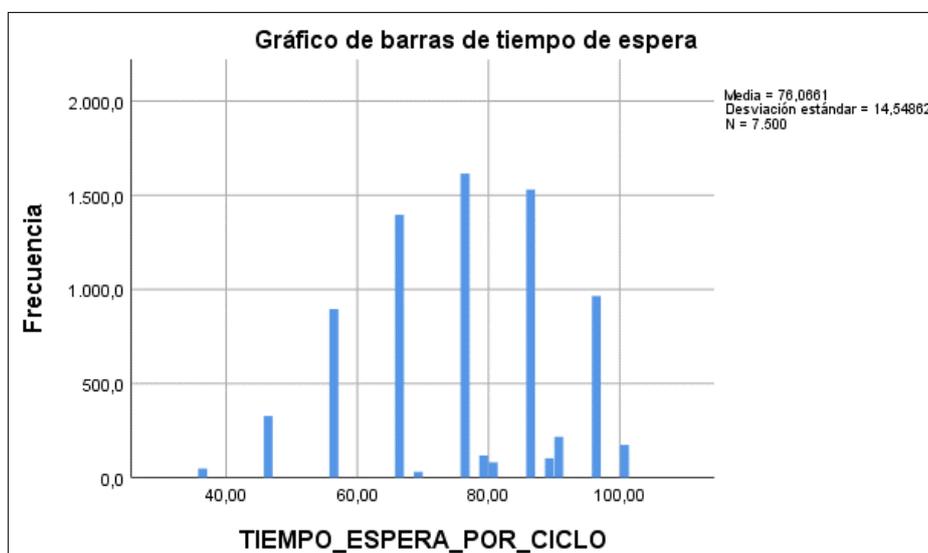


Nota: Según el estadístico descriptivo el valor de la mediana es igual a 76 muy cerca a la media 76,0661

Como se puede observar en el gráfico de frecuencia, Figura 35, los valores de tiempo de espera se acumulan cerca al valor de 76 segundos.

Figura 35

Gráfico de frecuencia de tiempo de espera por ciclo



Concluimos que, el valor esperado con el algoritmo propuesto reduce de manera significativa respecto al valor predefinido en un escenario predefinido de 55s para fase, 35 para rojo y 3s para amarillo.

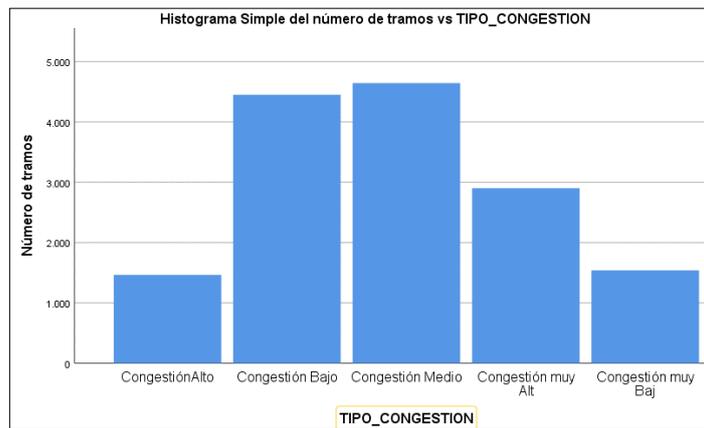
Porcentaje de ciclos de semáforo = $76/93 \times 100 = 81,7\%$

Reducción de tiempo: 18,3%

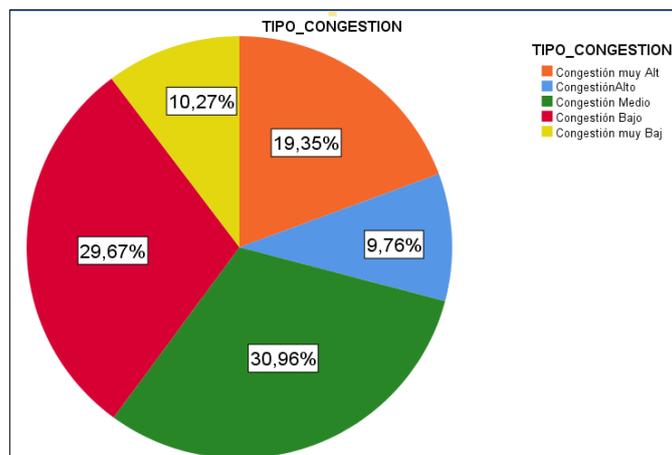
Figura 36

Histograma del número de tramos vs. TIPO_CONGESTION

36a



36b



Cómo podemos observar en la Figura 36a y 36b, de los escenarios evaluados, congestión bajo y congestión medio son los que más se presentan, pues representan un 29,67% y 30,96% respectivamente de todos los 15000 escenarios.

4.6.3 Respecto a los tiempos de espera con cadena de Markov

Para analizar los resultados respecto a los tiempos de espera basado en la predicción de cadena de Markov, se recurrió al gráfico generado por operaciones iterativas del algoritmo 3.9

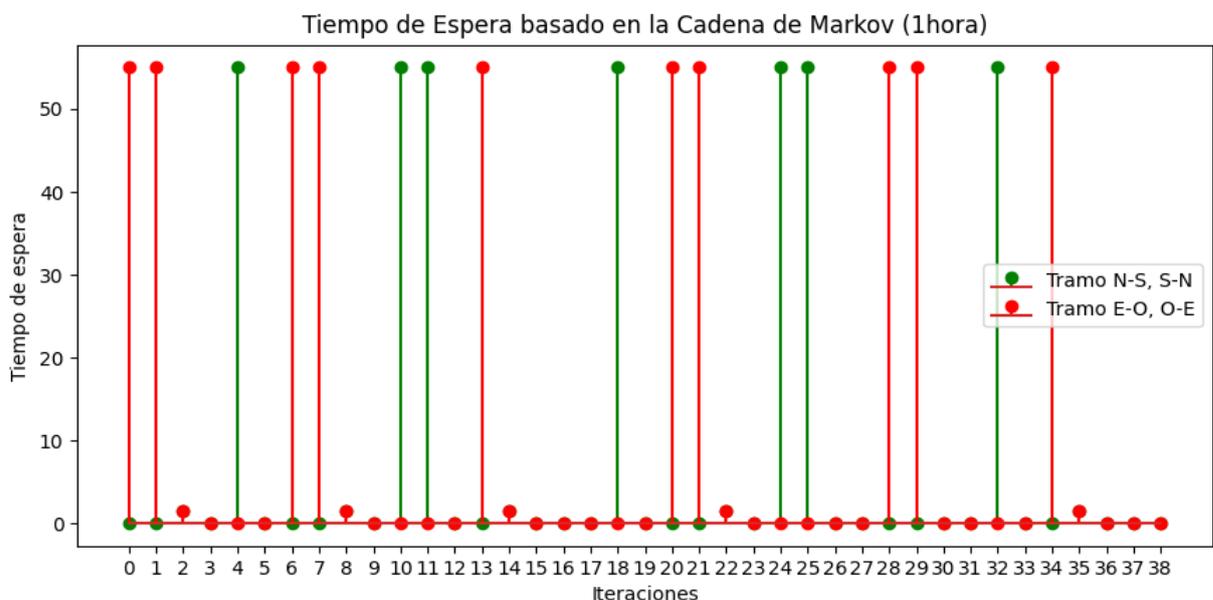
Cómo se observa en la Figura 37, cuando a ambos tramos se le asigna el mismo tiempo de verde, dicho tiempo verde se constituye en el tiempo de espera del otro tramo, esto se ha plasmado en las líneas 38, 39 y 40 del algoritmo 3.9.

El algoritmo permitió generar escenarios cuando se itera la matriz de transición de Markov y hay intercambio de estados (colores) de semáforo, pero en este caso se cubre la secuencia de estados ya no desde un solo tramo de la intersección, sino considerando el tramo transversal.

El proceso iterativo, considera un escenario con una duración de 1 hora, con tiempos de 55 segundos para verde para ambos tramos y 35 segundos de verde para el tramo transversal.

Figura 37

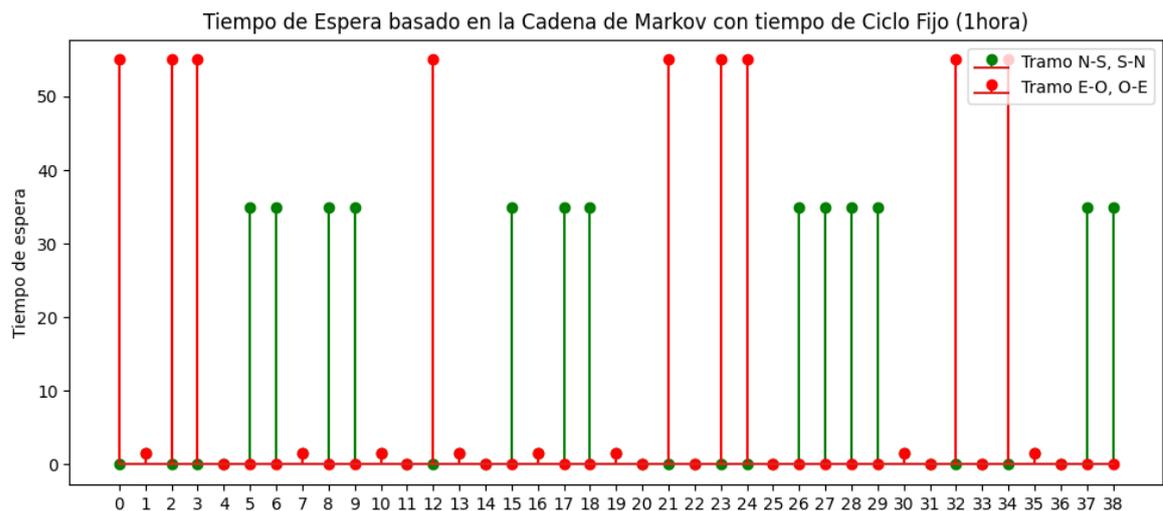
Secuencia de tiempos de espera con cadenas de Markov, caso 1



En la Figura 37 se observa cuando un tramo se encuentra en verde el otro tramo de manera simultánea está en rojo con la misma duración de 55 segundos y 3 segundos para el color amarillo.

Figura 38

Secuencia de tiempos de espera con cadenas de Markov, caso 2



En la Figura 38 se observa cuando un tramo se encuentra en rojo el otro tramo de manera simultánea está en verde con la misma duración de 55 segundos y 3 segundos para el color amarillo. Además, si un tramo se encuentra en fase verde de 35 segundos el tramo transversal estará en rojo de 35 segundos.

Ejemplo de una secuencia de generación fases 55 -35

---Secuencia--- [0, 2, 0, 0, 4, 1, 1, 3, 1, 1, 3, 5, 0, 2, 4, 1, 3, 1, 1, 3, 5, 0, 4, 0, 0, 4, 1, 1, 1, 1, 3, 5, 0, 4, 0, 2, 4, 1, 1, 5, 0, 0, 0, 4, 4, 1, 1, 1, 5, 0, 4, 1, 5, 0, 4, 0, 4, 1, 1, 3, 5, 5, 0, 4, 1, 5, 0, 0, 0, 4, 1, 3, 5, 5, 0, 0, 4, 4, 1, 1, 3, 5, 5, 0, 2, 4, 1, 1, 5, 0, 0, 4, 4, 0, 0, 0, 4, 1, 1, 5, 0, 0, 4, 4, 1, 1, 5, 1, 3, 5, 0, 2, 4, 1, 3, 5, 0, 4, 4, 0, 4, 4, 4, 1, 5, 0, 0, 4, 1, 1, 1, 1, 1, 5, 0, 4, 1, 3, 5, 1, 3, 1, 5, 5, 5, 0, 0, 4, 1, 1, 3, 5, 0, 0, 0, 2, 4, 1, 5, 5, 0, 4, 4, 4, 4, 1, 1, 3, 5, 0, 0, 4, 4, 1]

---TiempoTotal- 35, Tiempo promedio de espera por ciclo: Norte--Sur--Sur-Norte: 11,974358974358974 segundos

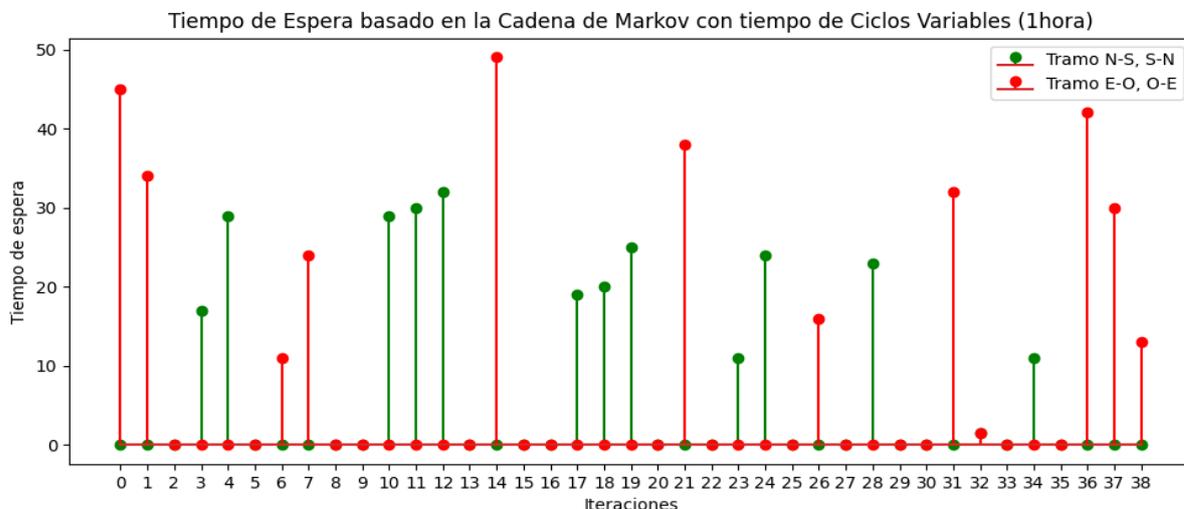
Este-Oeste-Oeste-Este: 13.0 segundos

[0, 1.5, 0, 0, 0, 35, 35, 1.5, 35, 35, 1.5, 0, 0, 1.5, 0, 35, 1.5, 35, 35, 1.5, 0, 0, 0, 0, 0, 0, 35, 35, 35, 35, 1.5, 0, 0, 0, 0, 1.5, 0, 35, 35]

[55, 1.5, 55, 55, 0, 0, 0, 1.5, 0, 0, 1.5, 0, 55, 1.5, 0, 0, 1.5, 0, 0, 1.5, 0, 55, 0, 55, 55, 0, 0, 0, 0, 0, 1.5, 0, 55, 0, 55, 1.5, 0, 0, 0]

Figura 39

Secuencia de tiempos de espera con cadenas de Markov, caso 3



En la Figura 39, se observa que cuando los ciclos de fase son variables, o sea el tiempo es variable para cada tramo, entonces en su respectivo tramo transversal también se notará el tiempo de espera variable, según sea el caso.

Ejemplo de una secuencia de generación fases de tiempo variable

---Secuencia--- [0, 0, 4, 1, 1, 5, 0, 0, 4, 4, 1, 1, 1, 5, 0, 4, 4, 1, 1, 1, 5, 0, 4, 1, 1, 5, 0, 4, 1, 5, 0, 4, 1, 5, 5, 0, 2, 4, 1, 5, 0, 0, 0, 0, 4, 1, 1, 5, 0, 2, 4, 1, 5, 0, 2, 4, 4, 4, 1, 5, 0, 0, 4, 4, 1, 3, 5, 0, 0, 4, 1, 5, 0, 0, 4, 1, 5, 5, 0, 0, 2, 4, 1, 1, 5, 0, 4, 1, 3, 5, 5, 0, 4, 1, 1, 5, 0, 0, 2, 0, 0, 0, 4, 1, 1, 5, 1, 1, 1, 3, 5, 0, 0, 0, 4, 1, 5, 5, 5, 0, 0, 0, 0, 0, 0, 2, 4, 1, 1, 1, 3, 1, 3, 5, 5, 0, 4, 4, 1, 3, 5, 1, 1, 5, 0, 0, 0, 4, 4, 1, 5, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 4, 1, 5, 0, 2, 4, 4, 1, 1, 1, 5, 0, 0, 4, 0, 0, 0, 0, 0, 2, 4, 1, 1, 1, 1, 5, 0, 4, 1, 1, 3, 5, 5, 1, 1, 3]

---TiempoTotal- 13 Tiempo promedio de espera por ciclo: Norte--Sur--Sur-Norte: 6.961538461538462 segundos

Este-Oeste-Oeste-Este: 8.602564102564102 segundos

[0, 0, 0, 17, 29, 0, 0, 0, 0, 0, 29, 30, 32, 0, 0, 0, 0, 19, 20, 25, 0, 0, 0, 11, 24, 0, 0, 0, 23, 0, 0, 0, 1.5, 0, 11, 0, 0, 0, 0]

[45, 34, 0, 0, 0, 0, 11, 24, 0, 0, 0, 0, 0, 0, 49, 0, 0, 0, 0, 0, 38, 0, 0, 0, 0, 16, 0, 0, 0, 0, 32, 1.5, 0, 0, 0, 42, 30, 13]

4.7 Respecto a la hipótesis central de investigación

Al respecto, la acumulación de evidencia relevante a partir de los resultados de la investigación obtenidos, respaldan la afirmación realizada en la hipótesis que

un sistema inteligente basado en un modelo heurístico, mediante la gestión en tiempo real de un semáforo, sí reduce de manera significativa los tiempos de espera de los vehículos en una intersección vial.

Por cuanto todos los resultados, confluyen a que el resultado del valor esperado con el sistema inteligente propuesto reduce el tiempo de espera de los vehículos, dado que los tiempos de espera no son fijos o predefinidos frente a los escenarios fijos o predefinidos donde el tiempo promedio por ciclo es de 93s

Una diferencia de 17 s para el tiempo de espera de ciclo semafórico con el sistema inteligente, indica que hay una reducción de 18,3 % respecto al enfoque tradicional, sin el sistema inteligente. Además, el incremento de 9 ciclos/h aumenta el flujo de tráfico. En ese sentido, se verifica la reducción significativa de los tiempos de espera de vehículos en una intersección vial, con el sistema inteligente, que fue el objetivo de nuestra investigación.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

A continuación, se presenta las conclusiones más resaltantes del proceso investigativo y que han permitido responder al problema de investigación.

La investigación realizada demuestra que se ha logrado reducir los tiempos de espera de vehículos en una intersección vial a través de un sistema inteligente basado en un modelo heurístico, en un 18,3%. Esta reducción de tiempo de espera de vehículos constituye una reducción significativa respecto al tiempo de espera predefinido en la misma intersección, sin la aplicación de un sistema inteligente (ver Tabla 14).

Se determinó las características de tráfico en una intersección vial para un entorno de tiempo real, mediante un desarrollo metódico detallado de los elementos que intervienen en el tráfico vehicular. Se ha logrado determinar que las características más resaltantes de tráfico vehicular son la Densidad Vehicular y la Velocidad de los vehículos (ver Figura 29), específicamente la velocidad media. Esto debido a que hay un relación lineal entre la densidad y la velocidad, por cuanto a mayor densidad se presenta menor velocidad del grupo de vehículos y a menor densidad se presenta mayor velocidad de los vehículos.

Según los análisis de la densidad de carriles, ha quedado claramente demostrado que la densidad vehicular es un parámetro muy robusto de flujo de tráfico e imprescindible para una programación de fases pues nos permite obtener el índice de congestión de tráfico. Sin embargo, también se ha demostrado que la variable pesos de vehículos no presenta demasiada variación y tiene un comportamiento no muy disperso. A diferencia de la velocidad cuya variación hace que el factor de índice E sea creciente o decreciente (ver Figura 30 y 31).

Se verificó un modelo de reconocimiento y conteo vehicular en tiempo real basado en los algoritmos de OpenCV y Yolo el cual mostró un nivel de precisión de 0,95 de 1 (ver Tabla 21). Además, se encontró que la principal razón de usar librerías OpenCV y Yolo es por la ventaja de la rapidez frente al nivel de precisión. Otros algoritmos como Fast RCNN con Tensor Flow pueden llegar a niveles más elevados de precisión pero su contraparte es que se toman mayor tiempo de procesamiento en la detección, y en tiempo real no es muy conveniente la latencia. El principal problema encontrado ha sido el problema del solapamiento que puede ocultar algunas imágenes de la realidad. Según las pruebas estadísticas que se realizó, el solapamiento presentado, no ha generado problemas significativos en el nivel de precisión. Sin embargo, una situación de super solapamiento debería considerarse en otros estudios.

Además del conteo de vehículos, en esta investigación se verificó el modelo de seguimiento de objetos con fines de obtener la velocidad de los vehículos, usando librerías de OpenCV. Se obtuvo una precisión de 0,96 y 0,97 de 1, lo cual supera al 95% esperado (ver Tabla 27). La mayor dificultad presentada fue la aplicación del método de medida del observador real, para cronometrar de manera manual el paso de vehículo, sin embargo, definiendo bien el inicio de tránsito del vehículo del segmento entre líneas discontinuas se logró superar esta dificultad.

En esta tesis se ha desarrollado el algoritmo de obtención de características de tramo vehicular como está descrito en el algoritmo para flujos no conflictivos para el cual como aporte de esta investigación se propuso 3 modos de uso de pares de flujo de tráfico que en la estructura de datos se denomina MODO (Ver Figura 21). La importancia de MODO, radica en la usabilidad en distintos escenarios de congestión vehicular. Esta secuencia de rutinas del algoritmo necesariamente requiere del uso de GPU, debido al fuerte consumo de procesamiento (ver Tabla 28).

Un aporte importante de este algoritmo en la tesis se encuentra en la ecuación (31), ecuación que se ha podido generalizar, para los intervalos del

índice de congestión de tráfico (ICT), donde si el ICT crece entonces el tiempo de fase también crece.

En cuanto al algoritmo que permite asignar intervalos de tiempos de color al semáforo en el tramo correspondiente, su desarrollo nos permitió concluir que es la parte medular del sistema de control de semáforo, y su aporte más resaltante es el algoritmo de reajuste de tiempo de fase para las diferentes densidades y velocidades de acuerdo al tipo de congestión (Ver Tabla 30). Añadido al desarrollo del algoritmo en mención, las decisiones de permiten asignar o disminuir segundos adicionales a la fase, según corresponda.

Finalmente, como cierre del proceso investigativo se simuló el funcionamiento del sistema de control de semáforo, mediante la generación de 15000 escenarios de fase, los cuales representan 7500 ciclos de semáforo obteniendo una mediana de 76 segundos de espera por ciclo de semáforo, frente a un escenario predeterminado de tiempos de 55s, 3s y 35s de fase. O sea, el escenario actual se toma 93 segundos de ciclo, frente a 76 segundos que el modelo heurístico propuesto arroja. El hecho de tomar menos tiempo de ciclo significa, que el modelo propuesto estaría generando 47 ciclos/hora frente a los 39 ciclos/hora que actualmente se tiene preconfigurado en los semáforos. En contraste con el modelo de control difuso de Wang et al. (2022) quien logró reducir un 14,59% el tiempo de espera promedio; con el modelo heurístico propuesto en esta tesis se obtuvo una mejora reduciendo un 18,3% el tiempo de espera promedio en cada ciclo de semáforo (ver Tabla 14).

Otros aportes no menos importantes se encuentran en la parte metodológica como: la propuesta de una mayor segmentación de los tipos de congestión; en contraste con Ashifuddin Mondal & Rehena, (2019) quien presenta tres tipos de congestión, en esta investigación se modeló 5 tipos de congestión, a fin de cubrir la gama de congestión muy baja y congestión muy alta, los cuales en las pruebas experimentales componen un 10,27% y 19,35% del total de 15000 escenarios generados por el algoritmo de semáforo inteligente (ver Figura 36a y 36b).

5.2 Recomendaciones

El algoritmo del sistema inteligente propuesto gestiona los tiempos de fase de un semáforo, pero no considera tiempos de fase igual a cero, donde no habría automóviles en el tramo; pero pueda que haya peatones quienes necesitan un tiempo mínimo para el cruce peatonal. Un estudio posterior puede involucrar detección de peatones para el cruce de peatonal, y así obtener una mejor aproximación del modelo para aplicaciones del mundo real.

Se está usando las cadenas de Markov, para garantizar la transición de estados, pero las cadenas de Markov no tienen memoria, por cuanto solo verifican cual fue el último estado anterior al actual. Se podría mejorar esto con un aprendizaje por refuerzo, porque puede que haya habido una buena experiencia de reajuste en algún momento de la realidad y la cadena por su naturaleza ya lo olvidó.

Por cuanto el modelo heurístico que se propone en esta tesis genera un Dataset muy rico en datos, un estudio basado en redes LSTM podría mostrar resultados que retroalimenten y mejoren el modelo actual.

Finalmente, para un arquitectura de tiempo real, se recomienda contemplar los tiempos de latencia, tiempo de respuesta de los servidores y servicios del modelo de despliegue de componentes.

VI. REFERENCIAS BIBLIOGRÁFICAS Y VIRTUALES

- Ashifuddin Mondal, M., & Rehena, Z. (2019). Intelligent Traffic Congestion Classification System using. *ACM Library*, 110-116.
<https://doi.org/https://doi.org/10.1145/3308560.3317053>
- Asmara, R. A., Syahputro, B., & Handayani, A. N. (2020). Prediction of Traffic Density Using YOLO Object Detection and Implemented in Raspberry Pi 3b + and Intel NCS 2. *4th International Conference on Vocational Education and Training (ICOVET)*, 391–395.
- Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint, arXiv:1409.0473*.
- Bendemra, H. (Mayo de 2018). *Ciencia y Datos*. Ciencia y datos:
<https://medium.com/datos-y-ciencia/construye-tu-primer-clasificador-de-deep-learning-con-tensorflow-ejemplo-de-razas-de-perros-ed218bb4df89>
- Bonifaz, J., & Aparicio, C. (2013). *La Gestión del Sistema de Transporte Público*. Lima, Perú: CEPLAN.
- Burns, A., & Wellings, A. (2003). *Sistemas de Tiempo Real y Lenguajes de Programación*. España: ADDISON-WESLEY Iberoamericana España.
- Camargo, E. (2018). El “Pico y Placa” en Bogotá: ¿peor el remedio que la enfermedad? *Universidad Externado de Colombia*, 47.
<https://doi.org/https://doi.org/10.18601/01236458.n47.07>.
- Dai, J., Li, Y., He, K., & Sun, J. (21 de junio de 2016). R-FCN: Object Detection via Region-based Fully Convolutional Networks. *arXiv:1605.06409 [cs.CV]*.
<https://arxiv.org/format/1605.06409v2>
- Deshpande, A. (12 de Diciembre de 2018). <https://adeshpande3.github.io/>.
<https://adeshpande3.github.io/>:
<https://adeshpande3.github.io/adeshpande3.github.io/>

- Fang, W., Chen, Y., & Xue, Q. (2021). Survey on research of RNN-based spatio-temporal sequence prediction algorithms. *Journal on Big Dat*, 3(3), 97-109. <https://doi.org/10.32604/jbd.2021.016993>
- Gómez S, G. (2016). *Generación de voz usando aprendizaje profundo*. Barcelona: Escuela Técnica Superior de Ingeniería de Telecomunicaciones.
- Google. (2021). *Google Maps*. Google Maps: <https://www.google.com/maps/@-9.1234917,-78.5252392,14.74z?entry=ttu>
- Held, D., Thrun, S., & Savarese, S. (2016). Learning to Track at 100 FPS with Deep Regression Networks. *In European Conference Computer Vision (ECCV)*.
- Hernández, R., Fernández, C., & Baptista, M. (2014). *Metodología de la investigación*. México: Mc Graw Hill.
- Inglés-Romero, J., Cano-Baños, M., Doménech-Asens, G., & Morales-Estera, V. (2018). Traffic Metrics at Urban Intersections using a Low-Cost Image Processing Approach. *Association for Computing Machinery*, Article No. 9. <https://doi.org/https://doi.org/10.1145/3243394.3243693>
- Lucas, B., & Kanade, T. (1981). An iterative image registration technique with an application to stereo. *Proceedings of the International Joint Conference on Artificial Intelligence*, hal-03697340, 674– 679.
- Manrique, W. (2017). *Estudio del tráfico y tendencias de movilidad urbana en Lima - 2017*. Lima: Marketwin. <http://willardmanrique.pe/wp-content/uploads/2018/08/Presentacion-final-Estudio-del-trafico-2017.pdf>
- Manrique, W. (2018). *Estudio del tráfico y tendencias de movilidad urbana en Lima - 2018*. Lima: Marketwin. <http://willardmanrique.pe/wp-content/uploads/2018/09/Estudio-de-Trafico-y-Tendencias-de-Movilidad-Urbana-2018.pdf>
- Medina R., G., & Ulloa Q., N. (2015). *Estudio para la implementación de una tercera vía como alternativa de solución al problema del transporte interurbano entre Chimbote y Nuevo Chimbote*. Nuevo Chimbote: Universidad Nacional del Santa.

- Ministerio de Transportes y Comunicaciones. (2018). *Manual de dispositivos de control de tránsito automotor para calles y carreteras*. Peru: MTC.
- Ministerio de Transportes y Comunicaciones. (2021). *Reglamento del Sistema de Control de Licencias de Conducir por Puntos*. Lima: El Peruano.
<https://busquedas.elperuano.pe/dispositivo/NL/1974058-2>
- MTC-DG. (2018). *Manual de carreteraS: diseño geométrico DG-2018*.
- Nigam, N., Pratap Singh, D., & Choudhary, J. (2023). A Review of Different Components of the Intelligent Traffic Management System (ITMS). *Simetría*, 583(3), 15. <https://doi.org/https://doi.org/10.3390/sym15030583>
- Ning, G. (2017). Spatially supervised recurrent convolutional neural networks for visual object tracking. *in Proceedings - IEEE International Symposium on Circuits and Systems*, 1-4.
<https://doi.org/https://doi.org/10.1109/ISCAS.2017.8050867>
- Ningson, P., Jie, Y., & Erqui, L. (2005). Model update mechanism for mean-shift tracking. *Journal of Systems Engineering and Electronics*, 16(1), 52–57.
- OCDE. (2011). OCDE. OCDE: <http://www.oecd.org/greengrowth/greening-energy/49157219.pdf>
- Peña, J., Lozano, J., & Larrañaga, P. (1999). An empirical comparison of four initialization methods for the K-Means algorithm. *Pattern Recognition Letters*, 20(10), 1027-1040. [https://doi.org/https://doi.org/10.1016/S0167-8655\(99\)00069-0](https://doi.org/https://doi.org/10.1016/S0167-8655(99)00069-0)
- Perficient. (30 de julio de 2019). *Image Recognition Accuracy Study*. Image Recognition Accuracy Study: <https://www.perficient.com/insights/research-hub/image-recognition-accuracy-study>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 779-788.
- Rivera-Trigueros, I. (2022). Machine translation systems and quality assessment: a systematic review. *Lang Resources & Evaluation*, 56(2), 593–619.
<https://doi.org/https://doi.org/10.1007/s10579-021-09537-5>

- Salinas, N. E. (2018). El transporte y el estrés en la ciudad de Lima. *Universidad Ricardo palma*, 1-18.
- Seace. (17 de Octubre de 2019). Seace:
<https://prodapp2.seace.gob.pe/seacebus-uiwd-pub/fichaSeleccion/fichaSeleccion.xhtml>
- Shiri, F. M., Perumal, T., Mustapha, N., Mohamed, R., Ahmadon, M., & Yamaguchi, S. (2023). A Survey on Multi-Resident Activity Recognition in Smart Environments. *arXiv preprint, arXiv:2304.12304*.
<https://doi.org/https://doi.org/10.48550/arXiv.2304.12304>
- Shiri, F., Perumal, T., Mustapha, N., & Mohamed, R. (2023). A Comprehensive Overview and Comparative Analysis on Deep Learning Models: CNN, RNN, LSTM, GRU. *arXiv e-prints, arXiv-2305*.
- Srivastava, S., & Sahana, S. (2017). Nested Hybrid Evolutionary Model for Traffic Signal Optimization. *Appl. Intell.*, 46(1), 113-123.
<https://doi.org/https://doi.org/10.1007/s10489-016-0827-6>
- Storn, R., & Price, K. (1997). Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*(11), 341–359.
<https://doi.org/https://doi.org/10.1023/A:1008202821328>
- Taiping, J., Wan, Z., & Chen, F. (2021). Urban traffic signals timing at four-phase signalized intersection based on optimized two-stage fuzzy control scheme. *Mathematical Problems in Engineering*, 2021, 6693562.
<https://doi.org/https://doi.org/10.1155/2021/6693562>
- Tang, J., Llu, F., Zou, Y., Zhang, W., & Wang, Y. (2017). An Improved Fuzzy Neural Network for Traffic Speed Prediction Considering Periodic Characteristic. *IEEE Transactions on Intelligent Transportation Systems*, 18(9), 2340-2350. <https://doi.org/10.1109/TITS.2016.2643005>
- Thacker, N., & Lacey. (1998). Tutorial: The kalman filter. *Imaging Science and Biomedical*, 133–140.

- Wang, M., Wu, X., Tian, H., Lin, J., He, M., He, M., & Ding, L. (2022). Efficiency and Reliability Analysis of Self-Adaptive Two-Stage Fuzzy Control System in Complex Traffic Environment. *Journal of Advanced Transportation*, 2022, 6007485. <https://doi.org/https://doi.org/10.1155/2022/6007485>
- Wang, P., Fan, E., & Wang, P. (2021). Comparative analysis of image classification algorithms based on traditional machine learning and deep learning. *Pattern Recognition Letters*, 141, 6167. <https://doi.org/https://doi.org/10.1016/j.patrec.2020.07.042>
- Wang, S., Shen, Y., Bao, Z., & Qin, X. (2019). Intelligent Traffic Analytics: From Monitoring to Controlling. *Association for Computing Machinery*, 778-781.
- Wang, X., & Wang, F. (2020). The precision og Google Earth Map Analysis with the coordinates of lgs Stations. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-3/W10, 1053--1056. <https://doi.org/10.5194/isprs-archives-XLII-3-W10-1053-2020>
- Wojke, N., Bewley, A., & Paulus, D. (September de 2018). Simple online and realtime tracking with a deep association metric. *in Proceedings - International Conference on Image Processing, ICIP, 2017*, 3645–3649.
- Xing, X., Yonglie, Y., & Huang, X. (2021). Real-Time Object Tracking Based on Optical Flow. *2021 International Conference on Computer, Control and Robotics (ICCCR)(20447503)*, 315-318. <https://doi.org/https://doi.org/10.1109/ICCCR49711.2021.9349376>
- Yedjour, H. (2021). Optical Flow Based on Lucas-Kanade Method for Motion Estimation. *International Conference in Artificial Intelligence in Renewable Energetic Systems*, 174, 937-945. https://doi.org/https://doi.org/10.1007/978-3-030-63846-7_92
- Younes, M. B., & Boukerche, A. (2018). An efficient dynamic traffic light scheduling algorithm considering emergency vehicles for intelligent transportation systems. *Wireless Netw*, 24, 2451–2463. <https://doi.org/https://doi.org/10.1007/s11276-017-1482-5>

Younes, M. B., Boukerche, A., & DeRango, F. (2023). SmartLight A smart efficient traffic light scheduling algorithm for green road intersections. *Ad Hoc Networks*, 140, 12-13.

<https://doi.org/https://doi.org/10.1016/j.adhoc.2022.103061>

VII. ANEXOS

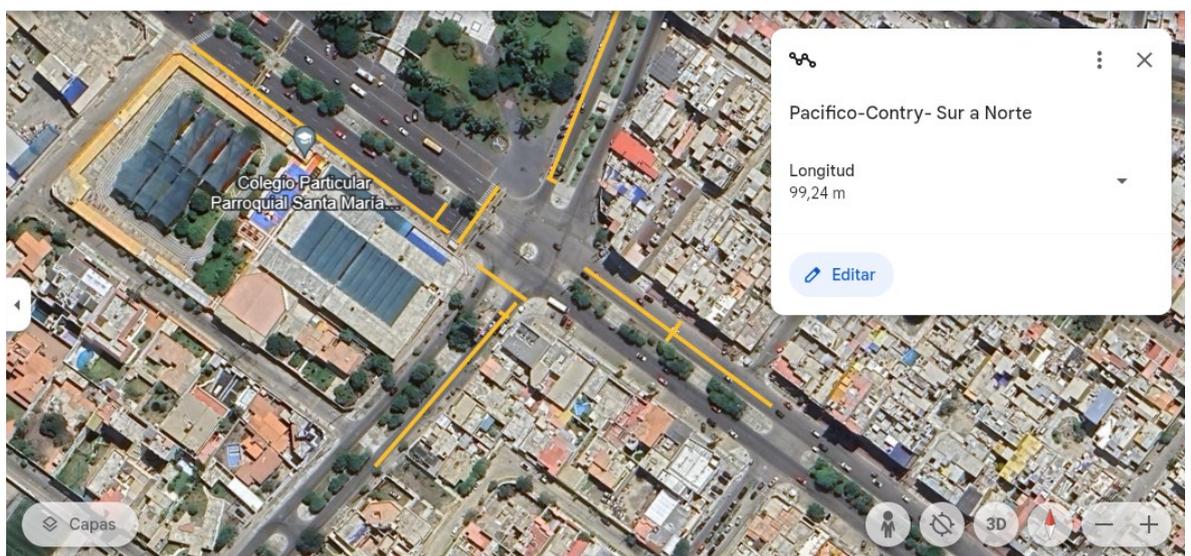
Anexo A: Principales intersecciones viales urbanas de la ciudad de Nuevo Chimbote, Perú



Nota: Esta figura has sido elaborada en base a la captura de imagen satelital de (Google, 2021). Donde se ha enumerado la intersección que tenga un semáforo. No se ha considerado intersecciones de carretera Panamericana de la ciudad

Nro.	Tramo	Referencia
1	Av. Pacífico	Centro comercial Metro
2	Av. Pacífico Av. Country	Plaza Mayo de Nvo. Chimbote
3	Av. Pacífico - Av. Chimbote	Oficinas rectorado UNS
4	Av. Pacífico - Av. Anchoveta	Colegio Señor de la Vida
5	Av. Pacífico – Mcdo. Bss. Ass	CIP,- colegio de Arquitectos
6	Ovalo de la Familia	Ovalo de la Familia
7	Av. Pacífico - Av. Central	Mercado Belén
8	Av. Argentina - Calle 101	Iglesia Tabernáculo de la Fe
9	Av. Argentina - Av. Anchoveta	Es salud Nvo. Chimbote
10	Av. Brasil - Av. Anchoveta	Hospital Regional- Rico Chimbote
11	Av. Brasil – Emergencia HEGB	Emergencia Hospital Regional
12	Av. Universitaria – Av. Brasil	SENATI -Campus UNS

Anexo B: Mapa de la intersección vial Av. Country - Av. Pacífico, Nuevo Chimbote, Perú



Nota: OpenStreetMaps y Google Earth se utilizaron para medir el largo y ancho de los tramos vehiculares, estos valores sirven como datos de entrada para determinar el área de un tramo vehicular.

Anexo C: Tabla de datos de conteo de vehículos

VARIABLE	Contador	Contador	Variación	Variación %	Precisión
Imagen	Algoritmo	manual			
1	12	13	1	0,077	0,92
2	8	13	5	0,385	0,62
3	10	15	5	0,333	0,67
4	9	15	6	0,400	0,60
5	11	15	4	0,267	0,73
6	10	15	5	0,333	0,67
7	10	15	5	0,333	0,67
8	12	13	1	0,077	0,92
9	7	7	0	0,000	1,00
10	9	9	0	0,000	1,00
11	10	10	0	0,000	1,00
12	6	6	0	0,000	1,00
13	11	11	0	0,000	1,00
14	11	11	0	0,000	1,00
15	9	9	0	0,000	1,00
16	9	9	0	0,000	1,00
17	11	11	0	0,000	1,00
18	9	10	1	0,100	0,90
19	12	12	0	0,000	1,00
20	11	11	0	0,000	1,00
21	11	11	0	0,000	1,00
22	11	11	0	0,000	1,00
23	11	12	1	0,083	0,92

24	11	12	1	0,083	0,92
25	11	13	2	0,154	0,85
26	7	7	0	0,000	1,00
27	8	8	0	0,000	1,00
28	8	8	0	0,000	1,00
29	10	10	0	0,000	1,00
30	9	12	3	0,250	0,75
31	8	10	2	0,200	0,80
32	10	11	1	0,091	0,91
33	7	7	0	0,000	1,00
34	9	9	0	0,000	1,00
35	10	11	1	0,091	0,91
36	10	12	2	0,167	0,83
37	9	9	0	0,000	1,00
38	10	10	0	0,000	1,00
39	10	12	2	0,167	0,83
40	10	13	3	0,231	0,77
41	13	15	2	0,133	0,87
42	13	15	2	0,133	0,87
43	13	15	2	0,133	0,87
44	14	15	1	0,067	0,93
45	6	8	2	0,250	0,75
46	7	10	3	0,300	0,70
47	13	13	0	0,000	1,00
48	8	13	5	0,385	0,62
49	9	13	4	0,308	0,69

50	12	13	1	0,077	0,92
51	10	13	3	0,231	0,77
52	7	13	6	0,462	0,54
53	11	13	2	0,154	0,85
54	9	13	4	0,308	0,69
55	8	8	0	0,000	1,00
56	9	9	0	0,000	1,00
57	10	10	0	0,000	1,00
58	8	10	2	0,200	0,80
59	6	10	4	0,400	0,60
60	8	10	2	0,200	0,80
61	7	10	3	0,300	0,70
62	9	9	0	0,000	1,00
63	9	9	0	0,000	1,00
64	9	11	2	0,182	0,82
65	9	9	0	0,000	1,00
66	8	9	1	0,111	0,89
67	9	9	0	0,000	1,00
68	11	11	0	0,000	1,00
69	11	11	0	0,000	1,00
70	12	12	0	0,000	1,00
71	12	12	0	0,000	1,00
72	14	14	0	0,000	1,00
73	13	14	1	0,071	0,93
74	12	13	1	0,077	0,92
75	12	16	4	0,250	0,75

76	9	9	0	0,000	1,00
77	9	9	0	0,000	1,00
78	11	11	0	0,000	1,00
79	9	10	1	0,100	0,90
80	12	12	0	0,000	1,00
81	11	11	0	0,000	1,00
82	11	11	0	0,000	1,00
83	11	11	0	0,000	1,00
84	11	12	1	0,083	0,92
85	9	9	0	0,000	1,00
86	10	11	1	0,091	0,91
87	10	12	2	0,167	0,83
88	9	9	0	0,000	1,00
89	10	10	0	0,000	1,00
90	8	10	2	0,200	0,80
91	7	10	3	0,300	0,70
92	9	9	0	0,000	1,00
93	9	9	0	0,000	1,00
94	12	12	0	0,000	1,00
95	12	12	0	0,000	1,00
96	11	11	0	0,000	1,00

Anexo D: Velocidades obtenidas por conteo manual (observador) y por conteo de algoritmo

VARIABLE	ConteoAlgoritmo	Conteo manual (s)	distancia	Velocidad	Velocidad	Variación	
ID_VEHICULO	km/h	(s)	d(m)	m/s	km/h	Variación	
1	4	20	1,71	9,5	5,555555556	20,00	0,000
2	6	38	0,86	9,5	11,04651163	39,77	1,767
3	5	34	0,86	9,5	11,04651163	39,77	5,767
4	7	26	1,14	9,5	8,333333333	30,00	4,000
5	10	57	0,57	9,5	16,66666667	60,00	3,000
6	13	36	0,86	9,5	11,04651163	39,77	3,767
7	17	41	0,86	9,5	11,04651163	39,77	1,233
8	19	46	0,68	9,5	13,97058824	50,29	4,294
9	25	88	0,43	9,5	22,09302326	79,53	8,465
10	27	27	1,14	9,5	8,333333333	30,00	3,000
11	29	46	0,68	9,5	13,97058824	50,29	4,294
12	30	22	1,14	9,5	8,333333333	30,00	8,000
13	50	19	1,37	9,5	6,934306569	24,96	5,964
14	51	19	1,7	9,5	5,588235294	20,12	1,118
15	55	24	1,14	9,5	8,333333333	30,00	6,000
16	82	52	0,57	9,5	16,66666667	60,00	8,000

km/h : kilómetros por hora

(s) : segundos

d(m) : distancia en metros

m/s : metros por segundo

Anexo E: Tabla de datos de factor de índice de vehículos ligeros y pesados

VelPromedio 30-60 (Km/h)	Peso 1-5 (x1000Kg)	Vel X Peso (Ei)	VelPromedio 30-60 (Km/h)	Peso 18-25 (x1000kg)	Vel X Peso (Ei)x10E-1
32	3	96,00	30	18	54,00
55	5	275,00	41	20	82,00
40	5	200,00	44	22	96,80
38	4	152,00	35	23	80,50
46	2	92,00	51	19	96,90
49	3	147,00	38	24	91,20
39	3	117,00	56	22	123,20
53	5	265,00	37	18	66,60
33	3	99,00	46	25	115,00
42	5	210,00	44	19	83,60
51	3	153,00	47	21	98,70
48	5	240,00	45	25	112,50
30	4	120,00	47	25	117,50
57	5	285,00	42	20	84,00
37	3	111,00	50	20	100,00
55	2	110,00	56	20	112,00
54	5	270,00	52	21	109,20
49	2	98,00	53	20	106,00
33	5	165,00	35	19	66,50
48	3	144,00	47	24	112,80
56	4	224,00	34	24	81,60
55	3	165,00	48	24	115,20
58	2	116,00	54	19	102,60
36	2	72,00	57	25	142,50
30	5	150,00	45	18	81,00

km/h : kilómetros por hora

kg : kilogramos

VelPromedio : velocidad promedio

Vel X Peso : velocidad por peso

Anexo F: Tabla de equivalencia de km/h, a otras unidades de medida

km/h	cuad/min	m/min	m/s
10	2	167	3
20	3	333	6
30	5	500	8
40	7	667	11
50	8	833	14
60	10	1000	17
70	12	1167	19
80	13	1333	22
90	15	1500	25
100	17	1667	28
110	18	1833	31
120	20	2000	33

km/h : kilómetros por hora
cuad/min : cuadras por minuto
m/s : metros por segundo
m/min : metros por minuto

Nota: Para fines prácticos se considera una cuadra equivalente a 100 metros aprox.

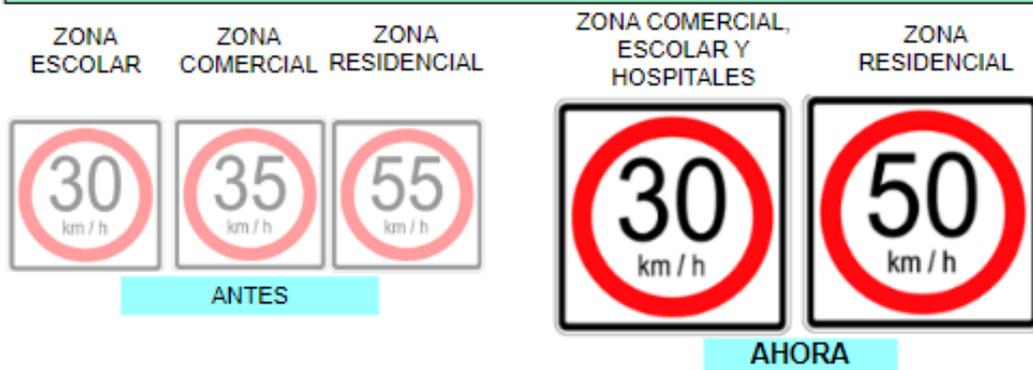
Anexo G: Velocidades permitidas en zonas urbanas y centros poblados

LÍMITES MÁXIMOS DE VELOCIDAD

EN ZONAS URBANAS



CARRETERAS QUE CRUZAN CENTROS POBLADOS



Nota: Actualmente la velocidad máxima permitida en avenidas es de 50 km/h, de acuerdo al Reglamento del Sistema de Control de Licencias de Conducir por Puntos y aprobado por Decreto Supremo Nro. 025-2021-MTC, del 17 de julio de 2021 en el diario oficial El Peruano.

Anexo H: Extracto de ejecución de iteraciones de la matriz de transición de Markov

Matriz M1		VERDE	AMARILLO	ROJO
	VERDE	0,9	0,1	0
	AMARILLO	0	0,8	0,2
	ROJO	0,2	0	0,8

M2		VERDE	AMARILLO	ROJO
	VERDE	0,81	0,17	0,02
	AMARILLO	0,04	0,64	0,32
	ROJO	0,34	0,02	0,64

M3		VERDE	AMARILLO	ROJO
	VERDE	0,733	0,217	0,05
	AMARILLO	0,1	0,516	0,384
	ROJO	0,434	0,05	0,516

M4		VERDE	AMARILLO	ROJO
	VERDE	0,66970	0,24690	0,08340
	AMARILLO	0,16680	0,42280	0,41040
	ROJO	0,49380	0,08340	0,42280

M5		VERDE	AMARILLO	ROJO
	VERDE	0,61941	0,26449	0,11610
	AMARILLO	0,23220	0,35492	0,41288
	ROJO	0,52898	0,11610	0,35492

M6		VERDE	AMARILLO	ROJO
	VERDE	0,58068900	0,27353300	0,14577800
	AMARILLO	0,29155600	0,30715600	0,40128800
	ROJO	0,54706600	0,14577800	0,30715600

M7		VERDE	AMARILLO	ROJO
	VERDE	0,5517757	0,2768953	0,171329
	AMARILLO	0,342658	0,2748804	0,3824616
	ROJO	0,5537906	0,171329	0,2748804

...

M40		VERDE	AMARILLO	ROJO
	VERDE	0,500009356	0,249999843	0,249990801
	AMARILLO	0,499981603	0,250009513	0,250008884
	ROJO	0,499999686	0,249990801	0,250009513

M41		VERDE	AMARILLO	ROJO
	VERDE	0,500006581	0,25000081	0,24999261
	AMARILLO	0,499985219	0,250005771	0,25000901
	ROJO	0,50000162	0,24999261	0,250005771

Anexo I: Extracto de la tabla de datos de trafico

1	Fase	Tramo	Veh_NS	Veh_SN	Veh_EO	Veh_OE	pmtf	TipoCongest	NombCongestion	maxVeh	TipoVelocidad	TiempoEspera
2	1	Norte-Sur	23	47	14	19	1	3	Congestión Medio	35	Vel. Alta	38
3	2	Este-Oeste	28	48	50	29	2	5	Congestión muy Alto	55	Vel. Moderada	58
4	1	Sur-Norte	43	25	26	27	0	3	Congestión Medio	35	Vel. Alta	38
5	2	Oeste-Este	48	10	11	20	3	2	Congestión Bajo	25	Vel. Moderada	28
6	1	Sur-Norte	41	11	37	47	0	3	Congestión Medio	35	Vel. muy Alta	38
7	2	Oeste-Este	5	35	34	44	3	5	Congestión muy Alto	55	Vel. Alta	58
8	1	Sur-Norte	18	15	37	41	0	1	Congestión muy Bajo	15	Vel. Alta	18
9	2	Oeste-Este	12	4	46	50	3	5	Congestión muy Alto	55	Vel. muy Alta	58
10	1	Sur-Norte	49	4	44	13	0	3	Congestión Medio	35	Vel. muy Alta	38
11	2	Este-Oeste	45	32	43	26	2	5	Congestión muy Alto	55	Vel. Alta	58
12	1	Sur-Norte	46	35	23	41	0	3	Congestión Medio	35	Vel. Moderada	38
13	2	Este-Oeste	26	8	48	9	2	5	Congestión muy Alto	55	Vel. Moderada	58
14	1	Sur-Norte	42	14	21	12	0	3	Congestión Medio	35	Vel. muy Alta	38
15	2	Este-Oeste	30	28	48	37	2	5	Congestión muy Alto	55	Vel. Alta	58
16	1	Sur-Norte	8	0	34	5	0	1	Congestión muy Bajo	15	Vel. Alta	18
17	2	Oeste-Este	20	27	12	36	3	4	CongestiónAlto	45	Vel. muy Alta	48
18	1	Norte-Sur	25	28	48	49	1	2	Congestión Bajo	25	Vel. Baja	28
19	2	Oeste-Este	17	35	0	20	3	2	Congestión Bajo	25	Vel. Moderada	28
20	1	Sur-Norte	44	3	2	33	0	3	Congestión Medio	35	Vel. muy Alta	38

...

14981	2	Oeste-Este	38	28	7	24	3	3	Congestión Medio	35	Vel. Alta	38
14982	1	Sur-Norte	30	19	27	29	0	2	Congestión Bajo	25	Vel. Alta	28
14983	2	Este-Oeste	12	11	32	1	2	3	Congestión Medio	35	Vel. Moderada	38
14984	1	Sur-Norte	9	8	32	15	0	1	Congestión muy Bajo	15	Vel. Alta	18
14985	2	Este-Oeste	0	20	40	16	2	5	Congestión muy Alto	55	Vel. Moderada	58
14986	1	Sur-Norte	22	4	45	48	0	2	Congestión Bajo	25	Vel. Alta	28
14987	2	Oeste-Este	16	43	1	17	3	2	Congestión Bajo	25	Vel. Baja	28
14988	1	Sur-Norte	27	8	46	5	0	2	Congestión Bajo	25	Vel. Moderada	28
14989	2	Este-Oeste	14	17	41	5	2	5	Congestión muy Alto	55	Vel. Baja	63
14990	1	Norte-Sur	20	24	29	47	1	2	Congestión Bajo	25	Vel. Baja	28
14991	2	Este-Oeste	49	47	29	17	2	3	Congestión Medio	35	Vel. muy Alta	38
14992	1	Norte-Sur	33	49	39	16	1	3	Congestión Medio	35	Vel. Baja	38
14993	2	Este-Oeste	38	19	50	31	2	5	Congestión muy Alto	55	Vel. Alta	58
14994	1	Sur-Norte	29	5	16	1	0	2	Congestión Bajo	25	Vel. Alta	28
14995	2	Oeste-Este	37	2	10	50	3	5	Congestión muy Alto	55	Vel. Moderada	58
14996	1	Norte-Sur	1	40	19	14	1	3	Congestión Medio	35	Vel. Baja	38
14997	2	Este-Oeste	32	37	27	2	2	3	Congestión Medio	35	Vel. Alta	38
14998	1	Norte-Sur	11	36	14	17	1	2	Congestión Bajo	25	Vel. Alta	28
14999	2	Este-Oeste	26	40	43	38	2	5	Congestión muy Alto	55	Vel. Moderada	58
15000	1	Sur-Norte	45	40	26	49	0	3	Congestión Medio	35	Vel. Baja	38
15001	2	Oeste-Este	15	23	12	40	3	5	Congestión muy Alto	55	Vel. Baja	63

Se presenta un extracto de las características de tráfico vehicular. Con 15000 filas generadas por algoritmo.

	TIEMPO_ESPERA_POR_CICLO	TIEMPO_ESPERA_FIJO
7493	56,00	93,00
7494	91,00	93,00
7495	66,00	93,00
7496	96,00	93,00
7497	86,00	93,00
7498	76,00	93,00
7499	86,00	93,00
7500	101,00	93,00

El tiempo de 2 tramos conforman el tiempo de un ciclo, por eso se presenta 7500 ciclos