



Universidad Católica de Santa María

Facultad de Ciencias e Ingenierías Físicas y Formales Escuela Profesional de Ingeniería Mecatrónica

Desarrollo de un sistema de control de tráfico inteligente basado en algoritmos de aprendizaje automático en la Av. Ejército en Arequipa, 2024

Tesis presentada por:

Sotomayor Valdivia, Carlos Eduardo

ORCID: 0009-0007-0569-0062

para optar el Título Profesional de Ingeniero Mecatrónico

Asesor:

Dr. Mestas Ramos, Sergio Rolando

ORCID: 0000-0002-5966-2506

Arequipa - Perú

2026

UCSM-ERP

UNIVERSIDAD CATÓLICA DE SANTA MARÍA

INGENIERIA MECATRONICA

TITULACIÓN CON TESIS

DICTAMEN APROBACIÓN DE BORRADOR

Arequipa, 21 de Mayo del 2026

Dictamen: 012974-C-EPIMMEM-2026

Visto el borrador del expediente 012974, presentado por:

2017802831 - SOTOMAYOR VALDIVIA CARLOS EDUARDO

Titulado:

**DESARROLLO DE UN SISTEMA DE CONTROL DE TRÁFICO INTELIGENTE BASADO EN
ALGORITMOS DE APRENDIZAJE AUTOMÁTICO EN LA AV. EJÉRCITO EN AREQUIPA, 2024**

Nuestro dictamen es:

APROBADO

Título Profesional/Título de Segunda Especialidad/Grado Académico a optar:

INGENIERO MECATRÓNICO

**29685318 - CASTRO VALDIVIA JORGE LUIS
DICTAMINADOR**



**40302702 - QUISPE CCACHUCO MARCELO JAIME
DICTAMINADOR**



**42394111 - COLLADO OPORTO CHRISTIAM GUILLERMO
DICTAMINADOR**



DESARROLLO DE UN SISTEMA DE CONTROL DE TRÁFICO INTELIGENTE BASADO EN ALGORITMOS DE APRENDIZAJE AUTOMÁTICO EN LA AV. EJÉRCITO EN AREQUIPA, 2024

INFORME DE ORIGINALIDAD

6%

INDICE DE SIMILITUD

5%

FUENTES DE INTERNET

2%

PUBLICACIONES

3%

TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1

hdl.handle.net

Fuente de Internet

<1 %

2

repositorioacademico.upc.edu.pe

Fuente de Internet

<1 %

3

Submitted to Cornell University

Trabajo del estudiante

<1 %

4

repositorio.ucsm.edu.pe

Fuente de Internet

<1 %

5

Submitted to Joseph Leckie Academy

Trabajo del estudiante

<1 %

6

www.coursehero.com

Fuente de Internet

<1 %

7

vitela.javerianacali.edu.co

Fuente de Internet

<1 %

8

Submitted to Universidad Católica de Santa María

Trabajo del estudiante

<1 %

9

Submitted to Universidad Internacional de la Rioja

<1 %

DEDICATORIA

Quiero dedicar este logro a mis amados padres Ramón y Jenny, si estoy alcanzando mis objetivos es gracias a que me brindaron formación profesional, educación, valores y principios. Gracias por el amor, los consejos, la paciencia y el apoyo que nunca me faltó de ellos.

También a mi nueva familia, mi pareja Yashit y mi hija Mavie, a quienes amo enormemente y son el motivo de seguir adelante cada día, con metas claras para poder superarnos profesional y familiarmente.

Y a mis grandes amigos que conocí en mi travesía universitaria; Denis, Renato, David Pablo y Gian Carlo; gracias a ellos aprendí lo que es el trabajo en equipo y la empatía, valores que aplico en mi día a día.

Carlos Eduardo Sotomayor Valdivia



AGRADECIMIENTOS

Es grato expresar mis más sinceros agradecimientos a todos los docentes de la Escuela Profesional de Ingeniería Mecánica, Mecánica Eléctrica y Mecatrónica que me han formado profesionalmente gracias a su experiencia técnica y gran pasión por la docencia, la cual se veía reflejada en cada una de las clases asistidas.

Asimismo, agradecer al Ing. Sergio Mestas Ramos por el apoyo basado en su gran experiencia y su completo respaldo en la etapa de desarrollo de mi proyecto de tesis.

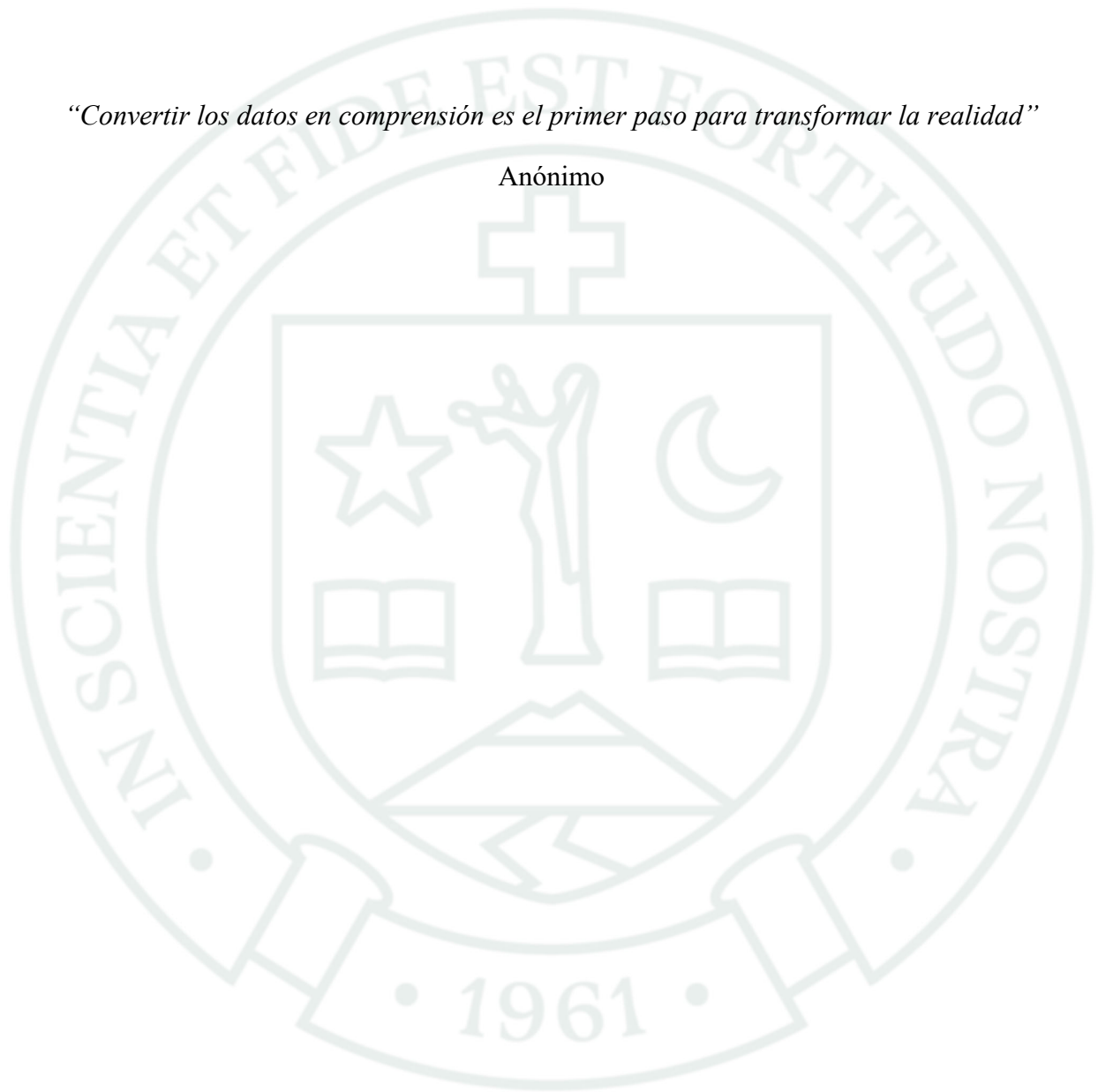
Carlos Eduardo Sotomayor Valdivia



EPÍGRAFE

“Convertir los datos en comprensión es el primer paso para transformar la realidad”

Anónimo



RESUMEN

El tráfico en la Av. Ejército de Arequipa, especialmente en el nodo conformado por la Av. Cayma y la Av. Trinidad Morán, presenta congestión que los semáforos de tiempo fijo no gestionan adecuadamente. Ante ello, la investigación diseñó y simuló un sistema inteligente de control semafórico orientado a ajustar fases y tiempos de verde según la demanda vehicular y peatonal del tramo.

El estudio fue aplicado, de enfoque cuantitativo, diseño no experimental y alcance descriptivo-propositivo con evaluación comparativa. Se utilizaron aforos de campo y simulación computacional. La arquitectura integró visión por computadora, validación térmica redundante de presencia peatonal y una lógica adaptativa sustentada en dos modelos Random Forest: un clasificador de fase y un regresor de tiempo verde, entrenados con 27 variables de entrada y una base de 6000 registros.

Los aforos identificaron mayores demandas entre 7:00 y 8:00 a. m. (4500 veh/h), 12:00 y 13:00 h (4560 veh/h) y 19:00 y 20:00 h (4260 veh/h), con predominio de autos y taxis (38 %), combis y microbuses (22 %), motos y mototaxis (18 %), cústeres y buses del SIT (14 %) y camiones u otros vehículos (8 %). El clasificador alcanzó 92,42 % de precisión y F1-score macro de 0,918; el regresor obtuvo MAE de 1,40 s y R^2 de 0,9617. En siete escenarios, el sistema redujo el tiempo de espera vehicular entre 15 % y 37 % en flujos bajo-medio, y mejoró la atención peatonal entre 65 % y 71 % en conflictos peatón-vehículo frente al ciclo fijo de 94 s. La hipótesis fue aceptada.

Palabras clave: control de tráfico inteligente, aprendizaje automático, simulación adaptativa.

ABSTRACT

Traffic congestion along Av. Ejército in Arequipa, particularly at the node formed by Av. Cayma and Av. Trinidad Morán, exceeds what fixed-time traffic signals can adequately manage. Therefore, this research designed and simulated an intelligent traffic signal control system aimed at adjusting phases and green times according to vehicular and pedestrian demand in the study segment.

The study was applied, quantitative, non-experimental, and descriptive-propositional with comparative evaluation. Field traffic counts and computational simulation were used. The system architecture integrated computer vision, redundant thermal validation of pedestrian presence, and adaptive logic based on two Random Forest models: a phase classifier and a green-time regressor, trained with 27 input variables and a dataset of 6000 records.

The counts identified the highest demand from 7:00 to 8:00 a.m. (4500 veh/h), 12:00 to 13:00 h (4560 veh/h), and 19:00 to 20:00 h (4260 veh/h), with a predominance of cars and taxis (38%), combis and minibuses (22%), motorcycles and mototaxis (18%), buses and SIT buses (14%), and trucks or other vehicles (8%). The classifier achieved 92.42% accuracy and a macro F1-score of 0.918; the regressor obtained a MAE of 1.40 s and R^2 of 0.9617. Across seven scenarios, the system reduced vehicular waiting time by 15% to 37% under low-to-medium flows and improved pedestrian service by 65% to 71% in pedestrian-vehicle conflict scenarios compared with a fixed 94-second cycle. The hypothesis was accepted.

Keywords: intelligent traffic control, machine learning, adaptive simulation.

ÍNDICE

DEDICATORIA	
AGRADECIMIENTOS	
EPÍGRAFE	
RESUMEN	
ABSTRACT	
ÍNDICE DE TABLAS	
ÍNDICE DE FIGURAS	
ÍNDICE DE ANEXOS	
INTRODUCCIÓN	1
CAPÍTULO I. METODOLOGÍA	3
1.1. Descripción del problema.....	3
1.1.1. Problemas específicos.....	4
1.2. Justificación de la investigación	4
1.2.1. Justificación social	4
1.2.2. Justificación ambiental.....	5
1.2.3. Justificación económica	6
1.3. Hipótesis	6
1.3.1. Hipótesis general.....	6
1.3.2. Hipótesis específicas.....	7
1.4. Objetivos.....	7
1.4.1. Objetivo general.....	7
1.4.2. Objetivos específicos	8
1.5. Enfoque, tipo, nivel y diseño de la investigación	8
1.6. Área de estudio	9
1.7. Variables del sistema	10
1.7.1. Variable independiente: sistema inteligente de control de tráfico	11
1.7.2. Variable dependiente: gestión del tránsito vehicular y peatonal	11
1.8. Técnicas, instrumentos y herramientas.....	12
1.9. Escenarios de simulación.....	13
1.10. Procedimiento metodológico	14
1.11. Criterios de evaluación	15
1.12. Alcances y limitaciones metodológicas.....	16
1.13. Antecedentes investigativos	16
1.13.1. Antecedentes internacionales.....	16
1.13.2. Antecedentes nacionales	17
1.13.3. Antecedentes locales.....	18
CAPÍTULO II. MARCO TEÓRICO	19
2.1. Ingeniería de tránsito y movilidad urbana	19
2.2. Control semafórico	20
2.2.1. Clasificación de los semáforos para el control de tránsito vehicular.....	20
2.2.2. Semáforo inteligente y control adaptativo	20
2.3. Sistemas inteligentes de transporte (ITS)	21
2.3.1. Inteligencia artificial: fundamentos conceptuales.....	22
2.3.2. El concepto de sistema inteligente en transporte	23

2.3.3. Percepción, representación y decisión en sistemas autónomos	24
2.3.4. Fundamentos del entrenamiento en aprendizaje automático	25
2.4. Algoritmos y aprendizaje automático (Machine Learning).....	27
2.4.1. Tipos de aprendizaje automático	27
2.4.2. Árboles de decisión.....	28
2.4.3. Random Forest	28
2.4.4. Aplicación del aprendizaje automático al control de tráfico	30
2.4.5. Evaluación y validación de modelos de aprendizaje automático.....	31
2.4.6. Interpretabilidad y trazabilidad de los modelos de árbol	32
2.5. Visión por computadora aplicada al tránsito	33
2.5.1. Imagen digital y procesamiento de imágenes	33
2.5.2. Detección y clasificación de actores viales.....	34
2.5.3. Redes neuronales y aprendizaje profundo aplicados al tránsito	34
2.6. Detección peatonal y redundancia sensorica	36
2.6.1. Detección térmica como mecanismo de validación.....	36
2.6.2. Fuentes de incertidumbre en la detección sensorica	37
2.7. Arquitectura de comunicación e integración del sistema	38
2.8. Comparación con el control de tiempo fijo	38
2.9. Normativa y estándares técnicos de referencia.....	38
2.10. Fundamentos del flujo vehicular	39
2.10.1. Variables fundamentales del tránsito	40
2.10.2. Relación fundamental del tránsito	41
2.10.3. Capacidad vial y nivel de servicio	41
2.11. Temporización semafórica y teoría de demoras	42
2.11.1. Elementos de la temporización semafórica	42
2.11.2. Cálculo del ciclo óptimo: el método de Webster	43
2.11.3. Estimación de la demora en intersecciones semaforizadas	44
2.12. Equivalencia vehicular y control semafórico adaptativo.....	45
2.12.1. Unidades de vehículo de pasajeros y factores de equivalencia.....	45
2.12.2. Estrategias de control semafórico	46
2.12.3. Sistemas adaptativos de referencia	46
2.13. Seguridad peatonal y movilidad urbana sostenible	47
2.13.1. El peatón en el diseño de intersecciones.....	47
2.13.2. Congestión, externalidades y movilidad sostenible	48
2.14. Marco normativo nacional e internacional aplicado al proyecto.....	48
2.14.1. Normativa nacional.....	48
2.14.2. Referencias internacionales	49
2.15. Geometría de la intersección y flujo de saturación.....	49
2.15.1. Elementos geométricos que condicionan la capacidad.....	49
2.15.2. Flujo de saturación y tiempo de verde efectivo	50
2.16. Teoría de colas y propagación de la congestión	50
2.16.1. Formación y disipación de colas.....	51
2.16.2. Ondas de choque y propagación de la congestión	51
2.17. Coordinación semafórica y progresión del tránsito	52
2.17.1. Onda verde y desfase	52
2.17.2. Relevancia para el sistema propuesto	52
2.18. Tecnologías de detección en sistemas inteligentes de transporte	52

2.18.1. Tecnologías tradicionales de detección	53
2.18.2. Detección basada en visión y sensores térmicos.....	53
2.19. Metodología de estudios de tránsito y aforos	53
2.19.1. El aforo vehicular.....	54
2.19.2. Periodos de estudio y hora punta	54
2.20. Indicadores de desempeño operativo del tránsito.....	54
2.20.1. Principales indicadores	54
2.20.2. Indicadores empleados en la investigación.....	55
2.21. Modelos de simulación del tránsito	55
2.21.1. Tipos de modelos de simulación.....	55
2.21.2. La simulación como método de validación	56
2.22. Priorización del transporte público en el control semafórico	56
2.22.1. Fundamento de la priorización	56
2.22.2. Relación con la ponderación de actores.....	57
2.23. Seguridad vial e indicadores de siniestralidad.....	57
2.23.1. Los conflictos en las intersecciones.....	57
2.23.2. Indicadores de siniestralidad.....	57
2.24. Impacto ambiental del tránsito y movilidad de bajas emisiones	58
2.24.1. Emisiones y consumo asociados a la congestión.....	58
2.24.2. Hacia una movilidad de bajas emisiones	58
2.25. Accesibilidad universal en el diseño de cruces peatonales.....	59
2.25.1. Principios de diseño universal	59
2.25.2. Implicaciones para el sistema propuesto	59
2.26. Clasificación de intersecciones y estrategias de control.....	59
2.26.1. Tipos de intersección	60
2.26.2. Criterios para la semaforización	60
2.27. Evolución del control semafórico hacia los sistemas inteligentes.....	60
2.27.1. Del tiempo fijo al control actuado y coordinado	60
2.27.2. El control inteligente y adaptativo	61
2.28. Gestión de la demanda y planificación de la movilidad urbana	61
2.28.1. Gestión de la demanda de transporte	61
2.28.2. El control inteligente como herramienta de gestión	62
2.29. Confiabilidad y tolerancia a fallos en los sistemas de control del tránsito.....	62
2.29.1. Redundancia y tolerancia a fallos	62
2.29.2. El modo seguro en el sistema propuesto.....	62
2.30. Sistemas integrados de transporte y gestión del corredor.....	63
2.30.1. El transporte público estructurado	63
2.30.2. Aporte del sistema propuesto a la gestión del corredor	63
2.31. Perspectivas: transporte conectado y ciudades inteligentes	64
2.31.1. El paradigma de la ciudad inteligente.....	64
2.31.2. Transporte conectado y proyección del sistema	64
2.32. Consideraciones finales del capítulo	64
CAPÍTULO III. DESARROLLO Y SIMULACIÓN DEL SISTEMA PROPUESTO	66
3.1. Análisis de factibilidad del sistema	66
3.1.1. Factibilidad técnica	66
3.1.2. Factibilidad operativa	67
3.1.3. Factibilidad económica	68

3.2. Arquitectura del sistema propuesto	69
3.2.1. Módulo de percepción visual	69
3.2.2. Módulo de validación redundante de presencia peatonal	70
3.2.3. Módulo de decisión y control semafórico.....	70
3.2.4. Módulo de comunicación e integración.....	71
3.2.5. Módulo de supervisión e interfaz.....	71
3.3. Diseño funcional del sistema.....	72
3.4. Definición de actores viales y ponderaciones.....	73
3.5. Datos de entrenamiento y validación del modelo.....	74
3.5.1. Fuentes para la recopilación de datos vehiculares en la Av. Ejército	74
3.5.2. Generación de la base de datos de entrenamiento	74
3.5.3. Variables de entrada del modelo.....	75
3.5.4. Variables de salida del modelo	76
3.5.5. Partición de los datos y métrica de evaluación	77
3.6. Lógica adaptativa de control semafórico.....	77
3.6.1. Reglas de asignación de prioridad	78
3.6.2. Asignación adaptativa de tiempos semafóricos	79
3.6.3. Modo seguro ante discrepancia de detección.....	79
3.6.4. Marco normativo aplicado al control semafórico	80
3.7. Desempeño del modelo de aprendizaje automático	81
3.8. Resultados por escenario de simulación.....	82
3.8.1. Escenario 1: Flujo bajo	83
3.8.2. Escenario 2: Flujo medio	85
3.8.3. Escenario 3: Flujo alto — hora punta	87
3.8.4. Escenario 4: Alta presencia de transporte público.....	90
3.8.5. Escenario 5: Presencia peatonal relevante	92
3.8.6. Escenario 6: Conflicto peatón–vehículo	95
3.8.7. Escenario 7: Discrepancia entre detección visual y sensor térmico	97
3.9. Análisis comparativo consolidado.....	100
3.10. Resultados de la interfaz de simulación	101
3.11. Validación final de interfaz, métricas operativas y conexión futura	107
3.11.1. Flujo vehicular específico y ecuaciones del modelo operativo	108
3.11.2. Métricas operativas incorporadas en la interfaz.....	109
3.11.3. Identificación de giros del cruce y proyección de coordinación entre semáforos	111
3.11.4. Panel de inferencia ML e interpretación de la decisión	113
3.11.5. Resultados comparativos ampliados: SCT-ML, prueba libre y sistema actual	115
3.11.6. Trazabilidad final entre dato real, modelo, interfaz y resultado	116
3.12. Síntesis estadística del levantamiento de datos de campo	116
3.12.1. Síntesis del flujo vehicular por intensidad operativa	117
3.12.2. Composición vehicular y peso operativo del flujo	118
3.12.3. Relación entre datos de campo y escenarios de simulación	119
3.13. Proceso iterativo de mejora y calibración del sistema.....	120
3.13.1. Matriz de mejoras aplicadas al prototipo	121
3.13.2. Lectura técnica de la mejora obtenida	122
3.14. Síntesis del capítulo	122
CAPÍTULO IV. RESULTADOS Y DISCUSIONES	124
4.1. Verificación de la hipótesis de investigación	124

4.2. Presentación consolidada de los resultados	125
4.3. Discusión de los resultados por escenario	127
4.4. Comparación con investigaciones previas.....	129
4.5. Validez y limitaciones de los resultados de simulación	130
4.6. Implicaciones técnicas y perspectivas de implementación.....	131
4.7. Síntesis del capítulo	132
CONCLUSIONES	133
RECOMENDACIONES	135
REFERENCIAS BIBLIOGRÁFICAS.....	137
ANEXOS.....	140
Anexo 1. Tabla A1 — Operacionalización de variables del sistema	140
Anexo 2. Especificaciones técnicas de referencia de los componentes del sistema.....	141
Anexo 3. Muestra representativa de la base de datos de entrenamiento	141
Anexo 4. Tabla A4.1 — Plan básico de mantenimiento preventivo del sistema propuesto...	142
Anexo 5. Tabla A4.2 — Matriz básica de contingencias operativas del sistema propuesto ..	143
Anexo 6. Manual de lectura de la interfaz de simulación	145
Anexo 7. Protocolo de demostración del prototipo	149
Anexo 8. Banco de preguntas y respuestas para sustentación técnica.....	150
Anexo 9. Ficha ampliada de aforo vehicular y peatonal	151
Anexo 10. Matriz de sistematización semanal de datos	152
Anexo 11. Matriz de control de versiones y trazabilidad del prototipo	153
Anexo 12. Capturas actualizadas de la interfaz funcional final	154
ANEXO 13. CÓDIGO FUENTE DEL SIMULADOR SCT-ML	164

ÍNDICE DE TABLAS

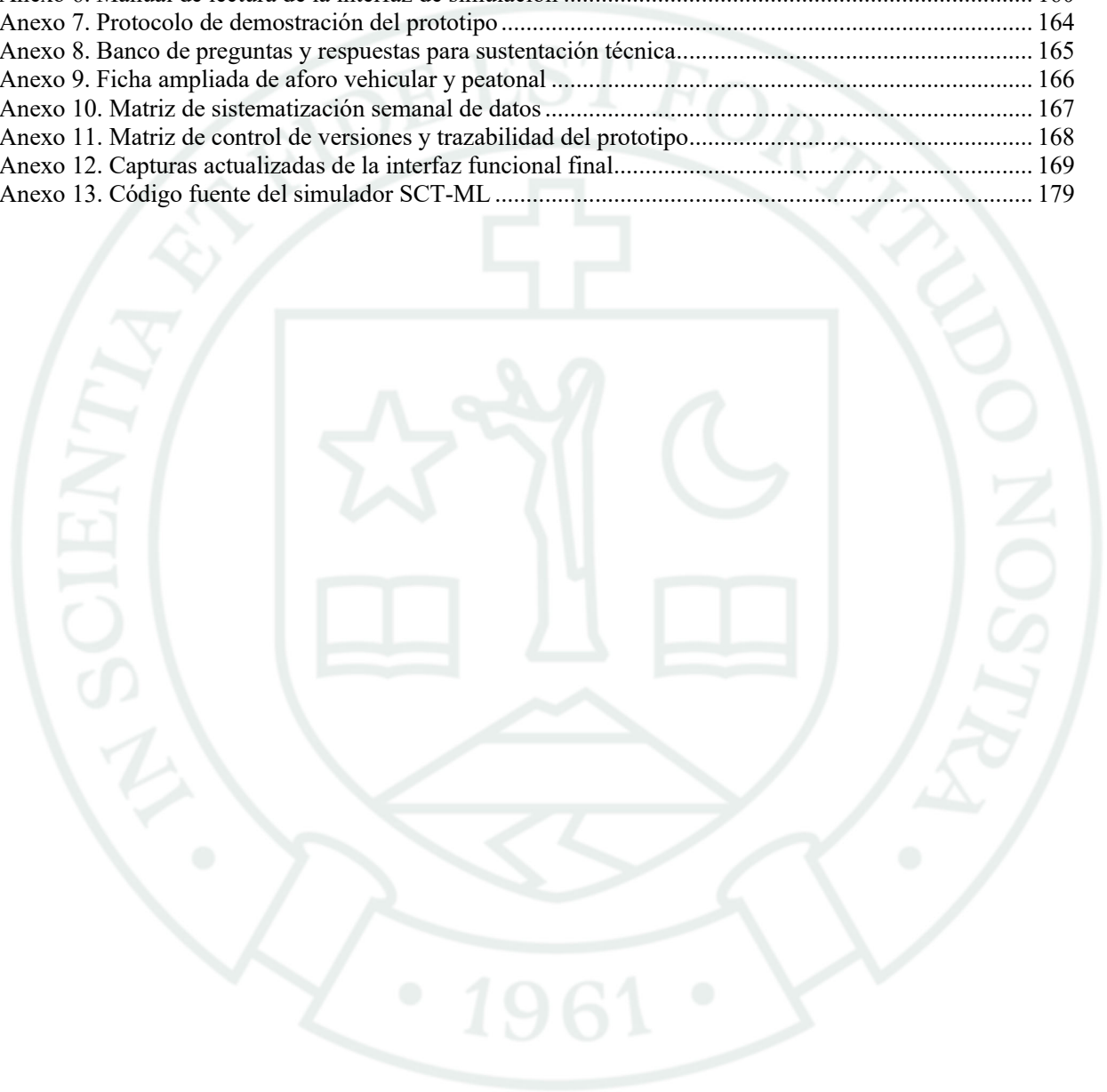
Tabla 1. Operacionalización de variables del sistema.....	27
Tabla 2. Escenarios de simulación considerados para la evaluación del sistema propuesto	28
Tabla 3. Variables de entrada del modelo (27 variables)	90
Tabla 4. Variables de salida del modelo.....	91
Tabla 5. Indicadores de desempeño de los modelos Random Forest entrenados	96
Tabla 6. Entradas del sistema - Flujo bajo	98
Tabla 7. Decisión ML vs tiempo fijo - Flujo bajo.....	99
Tabla 8. Tiempos de espera estimados - Flujo bajo	99
Tabla 9. Entradas del sistema - Flujo medio	101
Tabla 10. Decisión ML vs tiempo fijo - Flujo medio.....	101
Tabla 11. Tiempos de espera estimados - Flujo medio	101
Tabla 12. Entradas del sistema - Flujo alto - hora punta	103
Tabla 13. Decisión ML vs tiempo fijo - Flujo alto - hora punta.....	104
Tabla 14. Tiempos de espera estimados - Flujo alto - hora punta	104
Tabla 15. Entradas del sistema - Alta presencia de transporte público	106
Tabla 16. Decisión ML vs tiempo fijo - Alta presencia de transporte público.....	106
Tabla 17. Tiempos de espera estimados - Alta presencia de transporte público	106
Tabla 18. Entradas del sistema - Presencia peatonal relevante	108
Tabla 19. Decisión ML vs tiempo fijo - Presencia peatonal relevante	108
Tabla 20. Tiempos de espera estimados - Presencia peatonal relevante	109
Tabla 21. Entradas del sistema - Conflicto peatón-vehículo	111
Tabla 22. Decisión ML vs tiempo fijo - Conflicto peatón-vehículo.....	111
Tabla 23. Tiempos de espera estimados - Conflicto peatón-vehículo	111
Tabla 24. Entradas del sistema - Discrepancia entre detección visual y sensor térmico	113
Tabla 25. Decisión ML vs tiempo fijo - Discrepancia entre detección visual y sensor térmico	114
Tabla 26. Tiempos de espera estimados - Discrepancia entre detección visual y sensor térmico	114
Tabla 27. Resumen comparativo de tiempos de espera estimados - sistema adaptativo vs tiempo fijo	115
Tabla 28. Métricas operativas implementadas en la interfaz funcional.....	125
Tabla 29. Resultados comparativos ampliados por modo de simulación.....	130
Tabla 30. Trazabilidad final entre datos reales, variables, simulación y resultados.....	131
Tabla 31. Matriz de recolección de datos según tipo de día y franja horaria	132
Tabla 32. Síntesis estadística del flujo vehicular por franja operativa	133
Tabla 33. Composición vehicular estimada y ponderación operativa aplicada.....	133
Tabla 34. Trazabilidad entre observación de campo y escenario de simulación.....	134
Tabla 35. Evolución funcional del sistema por versiones de desarrollo.....	135
Tabla 36. Matriz de mejoras, correcciones e impacto operativo	136
Tabla 37. Resumen consolidado de la respuesta del sistema por escenario	140

ÍNDICE DE FIGURAS

Figura 1. Arquitectura funcional del sistema inteligente de control de tráfico propuesto.....	87
Figura 2. Diagrama de flujo de la lógica adaptativa de control semafórico	95
Figura 3. Interfaz del sistema - Flujo bajo: Entradas detectadas y fase priorizada.....	98
Figura 4. Interfaz del sistema - Flujo bajo: Comparación adaptativo vs tiempo fijo	100
Figura 5. Interfaz del sistema - Flujo medio: Entradas detectadas y fase priorizada	100
Figura 6. Interfaz del sistema - Flujo medio: Comparación adaptativo vs tiempo fijo	102
Figura 7. Interfaz del sistema - Flujo alto - hora punta: Entradas detectadas y fase priorizada	103
Figura 8. Interfaz del sistema - Flujo alto - hora punta: Comparación adaptativo vs tiempo fijo	105
Figura 9. Interfaz del sistema - Alta presencia de T.P.: Entradas detectadas y fase priorizada.....	105
Figura 10. Interfaz del sistema - Alta presencia de T.P.: Comparación adaptativo vs tiempo fijo	107
Figura 11. Interfaz del sistema - Presencia peatonal relevante: Entradas detectadas y fase priorizada.....	108
Figura 12. Interfaz del sistema - Presencia peatonal relevante: Comparación adaptativo vs tiempo fijo ...	110
Figura 13. Interfaz del sistema - Conflicto peatón-vehículo: Entradas detectadas y fase priorizada	110
Figura 14. Interfaz del sistema - Conflicto peatón-vehículo: Comparación adaptativo vs tiempo fijo	112
Figura 15. Interfaz del sistema - Discrepancia entre detección visual y sensor térmico	113
Figura 16. Interfaz del sistema - Discrepancia entre detección visual y sensor térmico: Comparación.....	115
Figura 17. Pantalla principal de la interfaz de simulación - vista general del sistema	117
Figura 18. Interfaz - Panel de decisión actual y cadena metodológica.....	118
Figura 19. Interfaz - Panel de entradas, escenarios y bitácora técnica	119
Figura 20. Interfaz - Modo de referencia del sistema actual fijo	120
Figura 21. Interfaz - Panel de demanda equivalente y lectura académica.....	121
Figura 22. Interfaz - Panel de inferencia ML e interpretación de la decisión.....	122
Figura 23. Vista general de la interfaz funcional final del sistema SCT-ML	123
Figura 24. Panel de métricas en vivo e índice de desempeño integral	126
Figura 25. Módulo de giros del cruce para estimación direccional de demanda.....	127
Figura 26. Imagen referencial para futura conexión entre semáforos del corredor.....	128
Figura 27. Panel de inferencia ML con entradas, variables procesadas y decisión final.....	129
Figura 28. Tiempo de espera peatonal por escenario: sistema adaptativo frente a tiempo fijo	141
Figura 29. Tiempo de verde asignado por escenario frente al esquema de tiempo fijo.....	141
Figura 30. Confianza del modelo Random Forest por escenario	142

ÍNDICE DE ANEXOS

Anexo 1. Tabla A1 — Operacionalización de variables del sistema	155
Anexo 2. Especificaciones técnicas de referencia de los componentes del sistema.....	156
Anexo 3. Muestra representativa de la base de datos de entrenamiento	156
Anexo 4. Tabla A4.1 — Plan básico de mantenimiento preventivo del sistema propuesto	157
Anexo 5. Tabla A4.2 — Matriz básica de contingencias operativas del sistema propuesto	158
Anexo 6. Manual de lectura de la interfaz de simulación	160
Anexo 7. Protocolo de demostración del prototipo	164
Anexo 8. Banco de preguntas y respuestas para sustentación técnica.....	165
Anexo 9. Ficha ampliada de aforo vehicular y peatonal	166
Anexo 10. Matriz de sistematización semanal de datos	167
Anexo 11. Matriz de control de versiones y trazabilidad del prototipo.....	168
Anexo 12. Capturas actualizadas de la interfaz funcional final.....	169
Anexo 13. Código fuente del simulador SCT-ML	179



INTRODUCCIÓN

La ciudad de Arequipa enfrenta una problemática crónica de congestión vehicular que persiste a pesar de los sucesivos planes de mejora de infraestructura vial. En sus corredores de mayor carga, la congestión ha dejado de constituir un fenómeno circunscrito a la hora punta para convertirse en una condición sostenida a lo largo del día que afecta en igual medida a conductores y peatones. La Av. Ejército constituye uno de los casos más representativos de esta problemática: la elevada carga vehicular, el entorno comercial activo y la confluencia de la Av. Cayma y la Av. Trinidad Morán configuran en ese tramo una intersección de alta complejidad operativa.

Esta problemática se agrava por la heterogeneidad de los usuarios de la vía. Automóviles, buses, cústeres, motocicletas, bicicletas, scooters y peatones comparten el mismo espacio con dinámicas de desplazamiento diferenciadas. Un sistema semafórico de tiempos fijos carece de capacidad para procesar esa variabilidad: asigna los mismos intervalos de fase con independencia de la demanda real, ya sea que existan decenas de buses acumulados en un acceso o que la vía se encuentre prácticamente libre. El resultado son colas innecesarias, esperas desproporcionadas y una gestión del cruce incoherente con las condiciones reales de operación.

Con el propósito de sustentar el análisis en datos empíricos del tramo, la investigación incorporó aforos vehiculares en el área de estudio. Se identificaron tres periodos de mayor demanda: de 7:00 a 8:00 a. m., de 12:00 a 1:00 p. m. y de 7:00 a 8:00 p. m., con flujos aproximados de 4 500, 4 560 y 4 260 veh/h respectivamente. En franjas intermedias el flujo osciló entre 1 900 y 3 600 veh/h, y en los momentos de menor demanda bajó a entre 400 y 1 100 veh/h. La composición del tránsito en hora punta confirmó lo que la observación ya sugería: una mezcla de autos, taxis, combis, microbuses, motos, mototaxis, cústeres, buses del SIT, camiones y otros vehículos que refuerza la variabilidad del tramo y la insuficiencia de un control rígido.

Desde ese diagnóstico, la investigación propone algo concreto: el diseño y simulación de un sistema de control semafórico que tome decisiones en función del flujo real, no de un ciclo predefinido. El sistema detecta y clasifica los actores viales mediante visión por computadora, valida la presencia peatonal con un sensor térmico redundante y aplica una lógica adaptativa basada en modelos de aprendizaje automático para asignar prioridad y tiempos de verde a cada fase. La decisión de trabajar en simulación y no en instalación física es deliberada: permite evaluar el comportamiento del sistema en distintos escenarios sin los riesgos ni los costos de una prueba en vía pública, y produce resultados comparables y metodológicamente defendibles.

El documento se organiza en cuatro capítulos. El primero desarrolla la metodología de investigación, incluyendo la descripción del problema, la justificación, la hipótesis, los objetivos, el área de estudio, las variables y los procedimientos. El segundo presenta el marco teórico, abarcando ingeniería de tránsito, control semafórico, algoritmos de aprendizaje automático, visión por computadora y sistemas inteligentes de transporte. El tercero expone el desarrollo del sistema, incluyendo el análisis de factibilidad, la arquitectura propuesta, el diseño funcional, los resultados por escenario y la comparación con el esquema de tiempo fijo. Cierran el documento las conclusiones, las recomendaciones, las referencias y los anexos.

CAPÍTULO I. METODOLOGÍA

1.1. Descripción del problema

El control del tránsito en la Av. Ejército de Arequipa, y en particular en el nodo vial donde convergen la Av. Cayma y la Av. Trinidad Morán, opera actualmente bajo un esquema semafórico de tiempos fijos que no responde a las condiciones reales del flujo. Los ciclos se programan de manera estática y se mantienen sin importar si en un acceso se acumulan decenas de buses o si la vía está prácticamente vacía. Esa rigidez genera esperas innecesarias, colas desproporcionadas y una gestión del cruce que resulta incoherente con lo que realmente ocurre en la vía.

El problema se agrava por la naturaleza del tránsito en ese sector. No es un flujo homogéneo de automóviles: en el mismo espacio circulan simultáneamente buses de transporte público, cústeres, combis, automóviles particulares, motocicletas, bicicletas, scooters y peatones, cada uno con tiempos de cruce, velocidades y necesidades de espacio distintos. Un sistema que no distingue entre clases de usuarios no puede asignar prioridades con coherencia operativa. El resultado es una gestión del cruce que no refleja la demanda real y que genera ineficiencias sistemáticas a lo largo del día.

La presencia peatonal agrega una dimensión adicional al problema. Los peatones que esperan cruzar no activan ningún mecanismo que modifique el ciclo semafórico convencional: esperan el tiempo que el ciclo les asigna, independientemente de cuántos sean o de cuánto tiempo lleven esperando. Cuando ese tiempo peatonal coincide con un pico de demanda vehicular, el sistema no tiene forma de resolver el conflicto de manera inteligente.

Frente a ese panorama, la pregunta de investigación que orienta este trabajo es la siguiente:

¿De qué manera la simulación de un sistema inteligente de control de tráfico, basado en visión por computadora, detección térmica redundante de presencia peatonal y lógica adaptativa de control semafórico basada en modelos Random Forest, puede

mejorar la gestión del tránsito vehicular y peatonal en el tramo crítico de la Av. Ejército condicionado por el nodo vial de la Av. Cayma y la Av. Trinidad Morán, en comparación con un esquema semafórico de tiempo fijo?

1.1.1. Problemas específicos

1. ¿Cómo delimitar el tramo crítico de la Av. Ejército e identificar sus condiciones operativas?
2. ¿Cómo diseñar una arquitectura funcional que integre visión por computadora, detección térmica, lógica adaptativa basada en ML y comunicación?
3. ¿Cómo definir la clasificación de actores viales y sus ponderaciones operativas?
4. ¿Cómo desarrollar la lógica adaptativa mediante modelos de aprendizaje automático?
5. ¿Cómo simular el comportamiento del sistema y compararlo con el esquema de tiempo fijo?

1.2. Justificación de la investigación

La congestión vehicular en la Av. Ejército no es un problema técnico aislado: tiene consecuencias directas sobre la población, el entorno urbano y la economía local. Por eso, la pertinencia de esta investigación puede analizarse desde tres dimensiones complementarias.

1.2.1. Justificación social

El tramo de la Av. Ejército estudiado es una de las vías de mayor uso cotidiano en el distrito de Cayma. Por ella transitan diariamente trabajadores, estudiantes, comerciantes y vecinos que dependen de un sistema vial funcional para llegar a tiempo a sus destinos. La congestión generada por un control semafórico rígido no solo alarga los tiempos de viaje, sino que también incrementa el estrés de los conductores, reduce la seguridad de los peatones que deben cruzar en condiciones de mayor presión vehicular y deteriora la calidad del espacio urbano compartido.

Los peatones son el grupo más vulnerable en ese contexto. Los datos de aforo levantados en el sector confirmaron que en hora punta se registran flujos de más de 4 500 veh/h, y que los cruces peatonales operan en condiciones de conflicto directo con esa demanda. Un sistema que incorpora la presencia peatonal como variable crítica de decisión, y no como dato accesorio, representa una mejora concreta en la seguridad y en la calidad de vida de quienes usan esa vía a diario. La reducción del tiempo de espera peatonal de entre 65 % y 71 % obtenida en los escenarios de conflicto dentro de la simulación ilustra ese potencial con datos cuantificables.

1.2.2. Justificación ambiental

La congestión vehicular tiene un costo ambiental que rara vez se cuantifica con precisión pero que es constante y acumulativo. Cuando los vehículos permanecen detenidos durante ciclos semafóricos innecesariamente largos, el motor sigue funcionando en ralentí, lo que incrementa el consumo de combustible y las emisiones de gases contaminantes. En un tramo con flujos de entre 1 900 y 4 560 veh/h según la franja horaria, incluso reducciones moderadas del tiempo de espera por ciclo representan un impacto ambiental acumulado significativo.

El sistema propuesto reduce los tiempos de ciclo en condiciones de baja y media demanda: en el escenario de flujo bajo, el verde adaptativo baja de 38 a 17 segundos, lo que elimina tiempos muertos en los que los vehículos esperan sin necesidad. Menos espera equivale a menos tiempo de motor encendido sin movimiento, menos consumo de combustible y menos emisión de CO₂ y partículas contaminantes. En una ciudad como Arequipa, que enfrenta crecientes presiones sobre la calidad del aire en sus corredores de mayor tráfico, esa reducción tiene un valor ambiental real. El diseño de sistemas de control más eficientes es, en ese sentido, una respuesta tecnológica con dimensión ecológica concreta.

1.2.3. Justificación económica

El tiempo perdido en congestión tiene un costo económico que se distribuye sobre toda la cadena de actividades que dependen del transporte: retrasos en la entrega de bienes, pérdida de productividad laboral, mayor gasto en combustible, desgaste acelerado de los vehículos y costos operativos más altos para el transporte público. En el tramo de la Av. Ejército, donde el 36 % del flujo en hora punta corresponde a unidades de transporte colectivo (buses, combis y cústeres), el impacto económico de un ciclo semafórico ineficiente no recae solo sobre los conductores individuales, sino sobre el sistema de movilidad urbana en su conjunto.

Un sistema adaptativo que asigna tiempos de verde más cortos cuando la demanda no los justifica y tiempos más largos cuando la acumulación lo requiere genera una distribución más eficiente del tiempo vial disponible. Esa eficiencia se traduce en menos combustible consumido por ciclo, menor tiempo de espera por unidad de transporte público y, a escala agregada, en una reducción de los costos operativos del sistema de movilidad. La propuesta desarrollada en esta investigación no requiere intervención física inmediata sobre la infraestructura vial existente, lo que reduce la barrera de implementación y permite evaluar su viabilidad económica antes de comprometer recursos de mayor magnitud.

1.3. Hipótesis

1.3.1. Hipótesis general

La simulación de un sistema inteligente de control de tráfico, sustentado en visión por computadora para la detección de actores viales, detección térmica redundante para la validación de presencia peatonal y una lógica adaptativa de asignación semafórica,

producirá resultados operativos superiores a los de un esquema de tiempo fijo en el tramo de la Av. Ejército condicionado por el nodo vial de la Av. Cayma y la Av. Trinidad Morán.

1.3.2. Hipótesis específicas

- HE1: La delimitación del tramo de estudio y la caracterización de sus condiciones operativas permitirán identificar los factores críticos que justifican la propuesta de control inteligente.
- HE2: El diseño de una arquitectura funcional modular permitirá integrar de manera coherente la percepción visual, la validación térmica, la lógica adaptativa y la interfaz de supervisión.
- HE3: La clasificación ponderada de actores viales representará la demanda real del tramo con mayor precisión que un conteo simple de unidades.
- HE4: Los modelos Random Forest asignarán fases y tiempos de verde de manera más coherente con la demanda que un ciclo predeterminado.
- HE5: El sistema propuesto presentará mejor desempeño operativo que el esquema de tiempo fijo en los siete escenarios de simulación evaluados.

1.4. Objetivos

1.4.1. Objetivo general

Diseñar y simular un sistema inteligente de control de tráfico para el tramo crítico de la Av. Ejército bajo la influencia del nodo vial conformado por la Av. Cayma y la Av. Trinidad Morán, empleando visión por computadora, detección térmica redundante y una lógica adaptativa de control semafórico, con el fin de mejorar la gestión del flujo vehicular y peatonal en ese sector.

1.4.2. Objetivos específicos

1. Delimitar el tramo crítico de estudio en la Av. Ejército y caracterizar las condiciones de operación vehicular y peatonal asociadas al nodo vial de influencia.
2. Diseñar la arquitectura funcional del sistema de control, integrando los módulos de percepción visual, validación térmica de presencia peatonal, toma de decisiones semafóricas e interfaz de supervisión, y definir la clasificación de los actores viales del tramo (buses o combis grandes, cústeres, automóviles, motocicletas, bicicletas, scooters y peatones) dentro del sistema propuesto.
3. Desarrollar la lógica adaptativa de control semafórico, basada en reglas ponderadas que asignan prioridades de paso según la demanda vehicular y la presencia peatonal detectada en cada ciclo, e implementarla en una interfaz funcional de simulación que permita observar la respuesta del sistema ante distintas configuraciones de entrada.
4. Simular el funcionamiento del sistema en escenarios operativos representativos, comparar su desempeño frente a un esquema convencional de tiempo fijo y evaluar su rendimiento mediante indicadores como la prioridad de fase, el tiempo de espera estimado y el comportamiento del flujo por acceso.

1.5. Enfoque, tipo, nivel y diseño de la investigación

La investigación se desarrolla bajo un enfoque cuantitativo. El sistema propuesto se evalúa a través de variables operativas observables y comparables, la prioridad de fase, el tiempo de espera estimado y el comportamiento del flujo, que se miden en escenarios de simulación definidos y se contrastan entre el esquema adaptativo y el de tiempo fijo.

Por su finalidad, es una investigación aplicada, ya que no busca generar conocimiento teórico abstracto, sino diseñar una solución tecnológica concreta para un problema real de tránsito urbano en un tramo específico de la Av. Ejército, en la ciudad de Arequipa.

Por su alcance, corresponde a un nivel descriptivo-propositivo con evaluación comparativa. Es descriptiva porque caracteriza el problema de circulación del sector, la composición del flujo vehicular y los elementos que intervienen en la operación del sistema. Es propositiva porque plantea el diseño de un sistema inteligente de control de tráfico. Y es comparativa porque contrasta su desempeño frente a un esquema semafórico convencional de tiempo fijo.

El diseño metodológico es no experimental. No hay intervención física sobre la infraestructura vial ni manipulación operativa directa del tránsito en campo. Sin embargo, la investigación sí incorpora levantamiento de datos reales del tramo estudiado mediante aforos vehiculares y caracterización operativa del nodo vial, información que sirve como base para definir las condiciones de entrada del sistema, construir la base de datos de entrenamiento y establecer escenarios representativos de simulación. La validación del sistema se sustenta en simulación computacional, lo que permite analizar la respuesta del control adaptativo frente a distintas condiciones operativas sin asumir los costos, riesgos y restricciones de una implementación inmediata en vía pública.

1.6. Área de estudio

El área de estudio comprende el tramo de la Av. Ejército ubicado en el sector comercial del distrito de Cayma, específicamente en el nodo vial donde convergen la Av. Cayma y la Av. Trinidad Morán. Se trata de una intersección de alta heterogeneidad operativa, en la que coexisten unidades de transporte público, automóviles particulares, motocicletas, bicicletas y peatones con dinámicas de desplazamiento diferenciadas, características propias de los corredores de uso mixto en entornos urbanos consolidados.

La selección de este nodo como área de estudio responde a sus condiciones operativas particulares: alta variabilidad del flujo vehicular a lo largo del día, presencia frecuente de conflictos entre flujos vehiculares y cruces peatonales, y composición heterogénea de actores viales. Estas características justifican la aplicación de un sistema de control semafórico adaptativo basado en aprendizaje automático, cuya utilidad se manifiesta precisamente cuando la demanda es variable y la rigidez de los ciclos fijos genera ineficiencias sistemáticas.

Con el propósito de caracterizar la operación del tramo, se ejecutaron aforos vehiculares y registros de observación en distintas franjas horarias. Se identificaron tres períodos de mayor demanda: de 7:00 a 8:00 h, de 12:00 a 13:00 h y de 19:00 a 20:00 h, con flujos aproximados de 4 500, 4 560 y 4 260 veh/h, respectivamente. En franjas horarias intermedias, el flujo osciló entre 1 900 y 3 600 veh/h, mientras que en los períodos de menor actividad descendió a valores de entre 400 y 1 100 veh/h.

El análisis de la composición vehicular durante la hora punta evidenció la heterogeneidad del flujo: autos y taxis representaron el 38 % de las unidades registradas, seguidos de combis y microbuses (22 %), motos y mototaxis (18 %), cústeres y buses del Sistema Integrado de Transporte (14 %), y camiones u otros vehículos (8 %). Esta distribución refuerza la pertinencia de adoptar un sistema de control que diferencie el tipo de actor vial en la toma de decisiones, dado que un conteo volumétrico sin clasificación no captura la incidencia diferenciada de cada categoría sobre la operación del cruce.

1.7. Variables del sistema

Con fines metodológicos, la investigación organiza sus variables en una independiente, el sistema inteligente de control de tráfico propuesto, y una dependiente, la gestión operativa del tránsito en el tramo de estudio. Tal formulación permite vincular el sistema diseñado con los efectos que se

espera observar en la simulación y en la comparación frente al esquema convencional (para la operacionalización completa de ambas variables, véase el Anexo 1).

1.7.1. Variable independiente: sistema inteligente de control de tráfico

Está constituida por la integración de cuatro dimensiones. La primera es la percepción visual del entorno vial, que corresponde al uso de visión por computadora para detectar y clasificar actores viales; sus indicadores son la detección de cada clase de usuario dentro del tramo. La segunda es la validación redundante de presencia peatonal, que corresponde al empleo del sensor térmico como mecanismo complementario; sus indicadores incluyen la confirmación de presencia peatonal, la coincidencia entre detección visual y térmica, y la activación de la condición de validación. La tercera es la lógica adaptativa de control semafórico, que corresponde al conjunto de decisiones del modelo y la capa de coherencia; sus indicadores son la fase priorizada, el tiempo verde y rojo asignados, y la activación del modo seguro ante discrepancia. La cuarta es la integración y supervisión del sistema, que corresponde a la arquitectura que enlaza percepción, decisión y visualización; sus indicadores son la transmisión de datos entre módulos y la representación funcional en la interfaz.

1.7.2. Variable dependiente: gestión del tránsito vehicular y peatonal

Comprende cuatro dimensiones. La eficiencia operativa del control semafórico se refiere a la capacidad del sistema para asignar tiempos de manera coherente con la demanda observada, y se mide a través de la distribución de prioridad de fase, el ajuste dinámico de tiempos y la comparación frente al tiempo fijo. El comportamiento del flujo vehicular se refiere a la respuesta del tránsito ante la lógica de control aplicada, y se mide mediante tiempos de espera estimados, acumulación de cola y comportamiento del flujo en escenarios simulados. La atención a la presencia peatonal se

refiere a la incorporación de la condición peatonal dentro de la toma de decisiones, con indicadores de detección, validación redundante y respuesta ante conflicto. El desempeño comparativo global es el contraste general entre la propuesta inteligente y el esquema convencional en todos los escenarios evaluados.

Tabla 1

Operacionalización de variables del sistema

Variable	Dimensión	Indicadores
VI: Sistema inteligente de control de tráfico	Percepción visual del entorno vial	Detección de buses o combis grandes, cústeres, automóviles, motocicletas, bicicletas, scooters y peatones por acceso.
	Validación redundante de presencia peatonal	Confirmación térmica de presencia peatonal; coincidencia entre detección visual y térmica; activación de condición de validación.
	Lógica adaptativa de control semafórico	Fase priorizada; tiempo verde asignado; tiempo rojo asignado; activación de modo seguro ante discrepancia entre sensores.
	Integración y supervisión del sistema	Transmisión de datos entre módulos; visualización de entradas y salidas en la interfaz de simulación.
VD: Gestión del tránsito vehicular y peatonal	Eficiencia operativa del control semafórico	Distribución de prioridad de fase; ajuste dinámico de tiempos semafóricos; comparación de tiempos frente al esquema de tiempo fijo.
	Comportamiento del flujo vehicular	Tiempo de espera estimado por acceso; acumulación o reducción de cola; comportamiento del flujo en escenarios simulados.
	Atención a la presencia peatonal	Detección y validación peatonal por acceso; respuesta del sistema ante conflicto peatón-vehículo.
	Desempeño comparativo global	Diferencia de tiempos de espera estimados; variación en prioridad de fase; respuesta general del sistema en escenarios equivalentes.

Nota. VI = Variable independiente. VD = Variable dependiente. Elaboración propia.

1.8. Técnicas, instrumentos y herramientas

Las técnicas empleadas fueron la revisión documental, la observación en campo, el aforo vehicular, el modelado del sistema, la simulación computacional y la evaluación comparativa de resultados. La revisión documental sustentó el marco teórico y los antecedentes de la investigación; el aforo vehicular permitió caracterizar la demanda real del tramo; y la simulación computacional posibilitó evaluar el comportamiento del sistema propuesto frente a distintos escenarios operativos.

Como instrumentos de recolección de datos se utilizaron fichas de registro de aforo, tablas de clasificación vehicular por tipo de actor vial y formatos de sistematización de información horaria. Para el procesamiento y validación del sistema se recurrió a herramientas computacionales orientadas al desarrollo del modelo de control, el entrenamiento de los algoritmos, la construcción de escenarios y la visualización de la respuesta del sistema.

El entorno principal de modelado, entrenamiento y validación algorítmica fue Python. En ese entorno se realizó el entrenamiento del clasificador y del regresor basados en Random Forest, la definición de los escenarios operativos y la implementación de la capa de coherencia mediante reglas determinísticas. Para la visualización y supervisión del sistema se desarrolló una interfaz funcional en entorno web local, utilizando HTML, CSS y JavaScript, con el fin de representar de manera interactiva el comportamiento del sistema propuesto (el código fuente completo del simulador SCT-ML, véase el Anexo 13).

1.9. Escenarios de simulación

La validación del sistema se realiza a través de siete escenarios de simulación que representan condiciones operativas plausibles del tramo de la Av. Ejército bajo la influencia del nodo vial. Cada escenario se procesa con el sistema inteligente y con el esquema de tiempo fijo, lo que permite comparar el desempeño de ambos bajo las mismas condiciones de entrada.

Tabla 2

Escenarios de simulación considerados para la evaluación del sistema propuesto

Escenario	Descripción	Respuesta esperada del sistema
Flujo bajo	Condición nocturna o de baja actividad. Demanda vehicular mínima y presencia peatonal reducida.	El sistema evita asignar tiempos excesivos cuando la demanda no los justifica.
Flujo medio	Condición de operación urbana regular con demanda intermedia, mezcla equilibrada de automóviles, transporte público y peatones.	Respuesta estable a demanda intermedia con tiempos proporcionales.

Flujo alto	Condiciones de hora punta: alta acumulación, mayor presión sobre la asignación de prioridad y flujo heterogéneo.	Priorización dinámica sin generar bloqueos mayores a los del tiempo fijo.
Alta presencia de transporte público	Predominio de buses o combis grandes y cústeres dentro del flujo vehicular.	El sistema refleja en la prioridad la mayor carga operativa del transporte público.
Presencia peatonal relevante	Flujo vehicular regular o medio con alta acumulación de peatones en zona de conflicto.	La lógica adaptativa incorpora la seguridad peatonal aunque implique penalización vehicular.
Conflicto peatón-vehículo	Coincidencia de alta demanda vehicular y acumulación peatonal validada por ambos módulos.	El sistema restringe la fase vehicular conflictiva y mantiene la protección peatonal.
Discrepancia entre sensores	La cámara y el sensor térmico no coinciden en la detección peatonal.	El sistema activa el modo seguro y adopta una respuesta conservadora.

Nota. Todos los escenarios se procesan tanto con la lógica del sistema inteligente propuesto como con un esquema semafórico convencional de tiempo fijo. Elaboración propia.

1.10. Procedimiento metodológico

La investigación se desarrolló en ocho etapas encadenadas, donde cada una parte de lo que construyó la anterior.

Etapla 1: Delimitar el área de estudio: identificar el tramo crítico de la Av. Ejército afectado por el nodo entre la Av. Cayma y la Av. Trinidad Morán, describir sus características operativas y justificar por qué ese tramo específico es un caso pertinente para la propuesta.

Etapla 2: Levantamiento de datos en campo. Se realizaron aforos vehiculares por franjas horarias para conocer los niveles de demanda, cómo varía el flujo a lo largo del día y qué composición vehicular presenta el sector.

Etapla 3: Caracterización de los actores viales y las condiciones operativas del tramo: qué tipos de usuarios intervienen, qué variables condicionan el comportamiento del tráfico y cómo interactúan entre sí.

Etapla 4: Diseño de la arquitectura funcional del sistema: definir sus cinco módulos y establecer cómo se relacionan entre sí para producir decisiones de control.

Etapla 5: Determinación de las variables de entrada y salida del modelo: qué información recibe en cada ciclo de decisión y qué produce como resultado.

Etapla 6: Desarrollo de la lógica adaptativa. Se entrenaron los modelos Random Forest para la clasificación de fase y la asignación del tiempo verde, se definió la capa de coherencia y se establecieron las reglas del modo seguro.

Etapla 7: Implementación del entorno de simulación y ejecución de los siete escenarios definidos. En cada uno el sistema adaptativo y el esquema de tiempo fijo operaron bajo las mismas condiciones.

Etapla 8: Evaluación de resultados: analizar los datos obtenidos, identificar las diferencias de desempeño entre ambos esquemas y extraer las conclusiones que se derivan de la comparación.

1.11. Criterios de evaluación

El desempeño del sistema se evalúa a partir de cinco criterios operativos aplicados en cada escenario de simulación.

1. Prioridad de fase: Si la lógica adaptativa asigna el derecho de paso a la fase correcta según la demanda real observada, o si toma decisiones que no se corresponden con las condiciones del entorno.

2. Tiempo semafórico asignado: Si la duración del verde y del rojo en el sistema adaptativo es más coherente con las condiciones de cada escenario que la programación fija del ciclo convencional.

3. Tiempo de espera estimado por acceso: Cuánto tiempo permanece un vehículo o un peatón esperando antes de recibir autorización de paso, comparado entre el sistema adaptativo y el de tiempo fijo.

4. Comportamiento del flujo vehicular: Si la respuesta del sistema reduce la acumulación en los accesos críticos y favorece una circulación más ordenada bajo distintas condiciones de demanda.

5. Desempeño comparativo global: El contraste general entre los resultados del sistema inteligente y los del esquema convencional a través de todos los escenarios.

1.12. Alcances y limitaciones metodológicas

La propuesta alcanza el nivel de diseño funcional, modelado, simulación y validación comparativa en entorno controlado. Esto significa que demuestra la coherencia técnica del sistema inteligente, la viabilidad de su arquitectura y su comportamiento esperado ante escenarios representativos del tramo estudiado. No obstante, la investigación no comprende una implementación física en campo ni una operación conectada al controlador semafórico real del nodo vial. Por tanto, los resultados deben interpretarse como evidencia de factibilidad y desempeño simulado, no como medición directa de operación sobre la infraestructura urbana existente.

1.13. Antecedentes investigativos

1.13.1. Antecedentes internacionales

Gorodokin, Zhankaziev, Karpov y Evtyukov (2021) desarrollaron una propuesta de control adaptativo de semáforos basada en visión artificial, empleando la red neuronal YOLOv4 para monitorear parámetros del flujo vehicular en accesos a intersecciones. A partir de esa información construyeron una metodología de regulación dinámica de ciclos semafóricos que consideraba tanto las condiciones actuales del flujo como las proyectadas. Este trabajo es directamente pertinente para la presente investigación porque demuestra la viabilidad operativa de usar percepción visual como base para ajustar la operación semafórica en tiempo real.

Smith, Venglar, Eichler, Balke y Park (2016), en un estudio financiado por la Federal Highway Administration, analizaron la integración de detección peatonal en sistemas de control adaptativo en tiempo real, incluyendo el uso de cámaras para incorporar la demanda peatonal dentro de la lógica semafórica. Su contribución principal fue mostrar que la gestión inteligente del tránsito no puede acotarse al flujo vehicular, sino que debe tratar la presencia peatonal como variable operativa con peso propio.

González Caballero (2019) propuso un sistema inteligente de gestión de tráfico capaz de modificar en tiempo real la duración de las fases semafóricas en función del flujo vehicular y la presencia o ausencia de peatones, incorporando visión artificial para el análisis automático del flujo. Su trabajo articula tres elementos que también están presentes en esta investigación: percepción visual, gestión dinámica de fases y vinculación entre tránsito vehicular y cruce peatonal.

1.13.2. Antecedentes nacionales

Velásquez Cárdenas y Solier Collazos (2017), en la Universidad Peruana de Ciencias Aplicadas, desarrollaron un Sistema Inteligente de Gestión de Tráfico usando Intel Galileo y OpenCV, orientado a reducir los tiempos de espera en intersecciones congestionadas mediante la asignación dinámica de tiempos semafóricos en función de la longitud de las colas vehiculares.

Carrera Travezaño (2022), en la Universidad Tecnológica del Perú, propuso un sistema basado en sensado remoto por infrarrojos para apoyar la semaforización en Comas, lo que refuerza la utilidad de incorporar sensores complementarios dentro de un sistema de control. Machaca Ninacansaya, en la Universidad Nacional del Altiplano, analizó y diseñó un sistema de control de tráfico vehicular mediante semáforos inteligentes con tecnología Arduino, constituyendo un antecedente académico valioso a nivel nacional.

1.13.3. Antecedentes locales

Peralta Fuentes y Mendieta Zavala (2017), de la Universidad Católica de Santa María, desarrollaron "Semáforo Inteligente para Regular el Tránsito Vehicular Basado en Open Hardware", una propuesta de semaforización inteligente orientada a mejorar el desplazamiento vehicular y peatonal. Que provenga de la misma institución que esta investigación resulta significativo: confirma que el control inteligente del tránsito ha sido reconocido como una línea válida de desarrollo académico en la UCSM. En ese mismo contexto arequipeño, el Proyecto de Desarrollo Urbamótico para la Ciudad de Arequipa, disponible en el repositorio de la UCSM, menciona componentes de semaforización inteligente y sistemas urbanos conectados, lo que contextualiza y respalda la pertinencia de este trabajo.

CAPÍTULO II. MARCO TEÓRICO

2.1. Ingeniería de tránsito y movilidad urbana

La ingeniería de tránsito es la rama de la ingeniería del transporte que se ocupa de cómo se mueven las personas y los vehículos por la red vial, con el objetivo de que ese movimiento sea seguro, ordenado y eficiente. No es suficiente analizar la capacidad vial a partir del número de carriles disponibles: esta disciplina considera también cómo se comportan los usuarios, cómo funcionan las intersecciones, cómo se programa la semaforización y cómo cambia la demanda a lo largo del día. Su alcance va mucho más allá del vehículo en movimiento: abarca el sistema vial completo en su relación con el entorno social, económico y urbano donde opera. (Transportation Research Board, 2010)

Dentro de este campo conviene distinguir dos conceptos que suelen confundirse. La circulación tiene que ver con la capacidad de la vía para sostener el flujo sin interrupciones, considerando la interacción entre peatones, vehículos motorizados y no motorizados. La movilidad, en cambio, refiere a un concepto diferente: la facilidad real que tiene una persona para desplazarse de un punto a otro dentro de la ciudad. Ambos términos no son equivalentes, dado que una vía puede registrar un flujo vehicular activo y aun así no garantizar movilidad adecuada para todos sus usuarios.

La congestión vehicular es la señal más visible de que un sistema vial no está funcionando como debería. Ocurre cuando la cantidad de vehículos que quieren circular supera lo que la vía puede manejar con orden, lo que desencadena saturación del flujo, colas, tiempos de viaje más largos y un deterioro general del desempeño de la red. Sus causas son variadas: alto volumen de tráfico, accidentes, obras en la calzada, condiciones climáticas desfavorables y una coordinación semafórica deficiente. Frente a ese panorama, la semaforización y los sistemas de control del tránsito cobran un papel central. (Contreras y Quispe, 2020)

2.2. Control semafórico

Los semáforos son el mecanismo más extendido para gestionar el tránsito en intersecciones y cruces peatonales. Su función básica es distribuir el uso del espacio y del tiempo entre los distintos flujos vehiculares y peatonales, de modo que los movimientos se ordenen y los conflictos se reduzcan. Históricamente, los semáforos han operado con tiempos fijos: cada fase tiene una duración predeterminada que no varía según lo que realmente esté pasando en la vía. Este esquema puede funcionar bien en intersecciones sencillas o donde la demanda es bastante estable, pero en entornos urbanos complejos muestra limitaciones claras. (MTC, 2016)

2.2.1. Clasificación de los semáforos para el control de tránsito vehicular

Los sistemas semafóricos pueden clasificarse según cómo determinan sus tiempos de operación y cuánta capacidad tienen para responder a las condiciones reales del flujo. El primer grupo lo forman los semáforos pre-sincronizados o de tiempos predeterminados: funcionan con ciclos fijos que no cambian en función del tráfico. El segundo corresponde a los semáforos accionados por el tráfico, que usan sensores para detectar la presencia o cantidad de vehículos y ajustan su operación en función de eso. El tercer grupo lo integran los semáforos con control centralizado, donde la operación depende de un sistema de supervisión más amplio que coordina múltiples cruces simultáneamente. Más recientemente, esta clasificación se ha ampliado para incluir esquemas de control adaptativo o inteligente, donde la información del entorno se procesa dinámicamente para ajustar la respuesta en tiempo real. (MTC, 2016)

2.2.2. Semáforo inteligente y control adaptativo

El semáforo inteligente incorpora mecanismos de percepción, análisis y decisión que le permiten adaptar su funcionamiento a las condiciones reales del tránsito. El principio fundamental de este enfoque es que la duración del tiempo verde no debe ser constante, sino que ha de responder a la

demanda observada: cantidad y tipo de vehículos presentes, presencia de peatones en espera de cruce y acumulación de ciclos sin atención en un acceso determinado. Para lograrlo, estos sistemas usan sensores, cámaras, módulos de procesamiento y algoritmos de decisión que identifican variaciones en el flujo, presencia peatonal o situaciones de conflicto. En los enfoques de control adaptativo, la prioridad se construye a partir de criterios relacionados con la carga vehicular, la ocupación del tramo, el tipo de usuario presente y las condiciones de seguridad. (Gorodokin et al., 2021; Yohannes y Singh, 2019)

2.3. Sistemas inteligentes de transporte (ITS)

Los sistemas inteligentes de transporte son la respuesta tecnológica a la creciente complejidad de la movilidad urbana. Los ITS (Intelligent Transportation Systems) pueden entenderse como el conjunto de tecnologías aplicadas al transporte y al tránsito para mejorar la seguridad, la eficiencia, la confiabilidad y la sostenibilidad de la movilidad. Su lógica de funcionamiento responde a una secuencia estructurada: observar el estado del sistema vial, interpretar la información captada y generar una respuesta de control acorde con las condiciones detectadas. Se apoyan, para ello, en sensores, cámaras, plataformas de procesamiento, dispositivos de comunicación y módulos de control. En el entorno urbano los ITS resultan especialmente útiles, porque la ciudad es, por naturaleza, un sistema abierto y heterogéneo (Contreras y Quispe, 2020).

Los sistemas inteligentes de transporte son la respuesta tecnológica a la creciente complejidad de la movilidad urbana. Los ITS (Intelligent Transportation Systems) pueden entenderse como el conjunto de tecnologías aplicadas al transporte y al tránsito para mejorar la seguridad, la eficiencia, la confiabilidad y la sostenibilidad de la movilidad. Su lógica de funcionamiento responde a una secuencia estructurada: observar el estado del sistema vial, interpretar la información captada y

generar una respuesta de control acorde con las condiciones detectadas. Se apoyan, para ello, en sensores, cámaras, plataformas de procesamiento, dispositivos de comunicación y módulos de control. En el entorno urbano los ITS resultan especialmente útiles, porque la ciudad es, por naturaleza, un sistema abierto y heterogéneo (Contreras y Quispe, 2020).

2.3.1. Inteligencia artificial: fundamentos conceptuales

La inteligencia artificial (IA) es la disciplina científica y tecnológica que se ocupa de diseñar sistemas computacionales capaces de realizar tareas que, si las ejecutara un ser humano, requerirían inteligencia. Esta definición, formulada en sus primeras versiones por McCarthy (1956) durante la conferencia de Dartmouth y revisada sucesivamente por la comunidad académica, sigue siendo la más ampliamente aceptada en la literatura contemporánea. Desde esa formulación original, la IA ha evolucionado desde enfoques basados en reglas simbólicas hacia métodos estadísticos y de aprendizaje, hasta llegar a las arquitecturas profundas que dominan el estado del arte actual.

El término "inteligente" aplicado a sistemas tecnológicos no implica conciencia ni comprensión semántica. Refiere a la capacidad de un sistema para adaptar su comportamiento en función de la información del entorno, aprender de la experiencia acumulada y tomar decisiones que optimizan algún criterio de desempeño. En el contexto del control de tráfico, un sistema inteligente es aquel que no solo ejecuta instrucciones predefinidas, sino que ajusta sus decisiones según las condiciones observadas, como la composición del flujo, la presencia peatonal o la acumulación de cola, con el objetivo de mejorar el desempeño operativo del tramo.

Conviene distinguir entre dos paradigmas principales dentro de la IA aplicada. El primer paradigma, conocido como IA simbólica o basada en reglas, opera mediante conjuntos de

instrucciones explícitas del tipo "si condición A, entonces acción B". Su principal ventaja es la transparencia: el razonamiento puede seguirse paso a paso. Sin embargo, en entornos complejos con muchas variables e interacciones no lineales, el número de reglas necesarias crece exponencialmente y su mantenimiento se vuelve inviable. El segundo paradigma, el aprendizaje automático, resuelve ese problema dejando que el sistema descubra las reglas relevantes a partir de los datos, sin necesidad de que un experto las formule de manera explícita.

2.3.2. El concepto de sistema inteligente en transporte

Un sistema inteligente de transporte (ITS) combina la capacidad de percepción del entorno, el procesamiento de la información y la generación de respuestas adaptadas, todo ello operando de manera autónoma o semi-autónoma dentro del ciclo continuo de observar–interpretar–actuar. En la literatura de ingeniería de tránsito, Papageorgiou et al. (2003) formalizaron esta cadena como el ciclo de control cerrado: el sistema capta el estado del entorno mediante sensores, lo interpreta mediante un módulo de procesamiento y emite una señal de control que modifica el comportamiento del sistema físico, el cual realimenta nueva información al ciclo.

La propiedad que distingue a un sistema inteligente de uno convencional no es la sofisticación del hardware, sino la capacidad de actualizar su comportamiento ante condiciones no previstas en el diseño original. Un semáforo de tiempo fijo es un sistema determinista: siempre responde igual ante la misma entrada programada. Un semáforo inteligente, en cambio, ajusta su respuesta según lo que realmente observa en la vía, pudiendo incluso mejorar su desempeño a medida que acumula experiencia operativa si se incorporan mecanismos de aprendizaje continuo.

Para que un sistema de control de tráfico pueda calificarse como inteligente en sentido técnico, debe satisfacer al menos tres condiciones: capacidad de percepción (detectar el estado del entorno con suficiente resolución y confiabilidad), capacidad de decisión (transformar la información

percibida en una acción de control coherente con los objetivos del sistema) y capacidad de adaptación (modificar su comportamiento ante variaciones del entorno no previstas en el diseño). El sistema propuesto en esta investigación satisface las tres condiciones mediante la integración del módulo de visión por computadora, los modelos Random Forest y la capa de coherencia con modo seguro.

2.3.3. Percepción, representación y decisión en sistemas autónomos

La arquitectura cognitiva básica de un sistema autónomo puede describirse mediante tres etapas funcionales que operan de manera secuencial y cíclica: percepción, representación y decisión. La etapa de percepción transforma las señales del entorno, como imágenes, datos de sensores y mediciones, en representaciones internas que el sistema puede procesar. La etapa de representación organiza esa información en estructuras que capturan las relaciones relevantes entre las variables del dominio, como la composición del flujo vehicular o la acumulación peatonal en un acceso. La etapa de decisión, por su parte, aplica una función de evaluación sobre las representaciones disponibles para seleccionar la acción más apropiada según los criterios del sistema.

En el sistema propuesto, estas tres etapas están implementadas de manera diferenciada. La percepción corresponde al módulo de visión por computadora, que procesa las imágenes de la cámara IP para detectar y clasificar los actores viales presentes. La representación corresponde al vector de 27 variables de entrada que codifica la composición del flujo, la condición peatonal y el historial de esperas. La decisión corresponde a los modelos Random Forest que, a partir de esa representación, determinan la fase semafórica a activar y la duración del tiempo de verde a asignar. La capa de coherencia actúa como un mecanismo de validación que garantiza que la decisión final sea operativamente coherente y segura.

2.3.4. Fundamentos del entrenamiento en aprendizaje automático

El entrenamiento es el proceso mediante el cual un modelo de aprendizaje automático ajusta sus parámetros internos para minimizar el error entre sus predicciones y los valores correctos contenidos en el conjunto de datos de referencia. En términos matemáticos, el entrenamiento busca el conjunto de parámetros θ que minimiza una función de pérdida $L(\theta)$ definida sobre el conjunto de ejemplos etiquetados disponibles. La forma específica de esa función de pérdida depende del tipo de problema: para clasificación se usan métricas como la entropía cruzada o el error de clasificación, mientras que para regresión se emplean métricas como el error cuadrático medio o el error absoluto medio.

El proceso de entrenamiento de un modelo supervisado sigue una secuencia bien establecida. En primer lugar, se construye o recopila un conjunto de datos que represente adecuadamente la diversidad de situaciones que el sistema encontrará en operación real. En segundo lugar, se divide ese conjunto en subconjuntos de entrenamiento y de prueba, de modo que el modelo sea evaluado sobre datos que no vio durante el ajuste de sus parámetros. En tercer lugar, se aplica el algoritmo de aprendizaje sobre el subconjunto de entrenamiento para ajustar los parámetros del modelo. En cuarto lugar, se evalúa el desempeño del modelo sobre el subconjunto de prueba mediante las métricas seleccionadas. En quinto lugar, se interpretan los resultados y se decide si el modelo cumple con los requisitos de desempeño establecidos.

Un concepto central en el proceso de entrenamiento es la distinción entre sesgo y varianza. El sesgo mide cuánto se aleja en promedio la predicción del modelo del valor verdadero; una varianza alta implica que el modelo cambia mucho sus predicciones ante pequeñas variaciones en los datos de entrada. El sobreajuste (overfitting) ocurre cuando un modelo tiene demasiada capacidad para aprender los datos de entrenamiento y en consecuencia memoriza el ruido específico de ese conjunto, perdiendo capacidad de generalización a datos nuevos. El subajuste (underfitting) ocurre

cuando el modelo tiene muy poca capacidad para capturar la estructura relevante de los datos. El objetivo del entrenamiento es encontrar el equilibrio adecuado entre sesgo y varianza que maximice el desempeño sobre datos no vistos.

Para los modelos de árbol de decisión y sus extensiones como el Random Forest, el entrenamiento opera de manera diferente a los modelos basados en gradiente como las redes neuronales. En un árbol de decisión, el entrenamiento consiste en seleccionar recursivamente la variable y el umbral de partición que mejor separan las clases (para clasificación) o que mejor reducen la varianza de los valores objetivo (para regresión) en cada nodo. El criterio de separación más empleado para clasificación es la ganancia de información o el índice de Gini, mientras que para regresión se utiliza la reducción de la varianza o el error cuadrático medio. El proceso se repite para cada nodo hijo hasta alcanzar algún criterio de parada: profundidad máxima del árbol, número mínimo de muestras en un nodo hoja, o ganancia de información por debajo de un umbral.

En el Random Forest, el proceso de entrenamiento combina dos fuentes de aleatoriedad para construir un conjunto de árboles con baja correlación entre sí. La primera fuente es el bagging (bootstrap aggregating): cada árbol se entrena sobre una muestra aleatoria con reemplazo del conjunto de entrenamiento, de tamaño igual al original. Esto significa que algunos ejemplos de entrenamiento son seleccionados varias veces y otros nunca, lo que introduce variabilidad entre los árboles. La segunda fuente es la selección aleatoria de variables: en cada nodo de cada árbol, el algoritmo no considera todas las variables disponibles sino solo un subconjunto aleatorio de ellas (típicamente la raíz cuadrada del número total de variables para clasificación, y un tercio para regresión). Esta restricción asegura que los árboles del bosque sean suficientemente distintos entre sí, reduciendo la correlación entre sus errores y mejorando la capacidad de generalización del conjunto.

2.4. Algoritmos y aprendizaje automático (Machine Learning)

El aprendizaje automático es una rama de la inteligencia artificial que desarrolla algoritmos capaces de aprender patrones a partir de datos y utilizarlos para tomar decisiones o hacer predicciones sin necesidad de ser programados explícitamente para cada situación particular. En lugar de seguir instrucciones rígidas, un modelo de aprendizaje automático extrae relaciones y reglas de forma automática desde un conjunto de ejemplos y las aplica luego a nuevos datos que no había visto antes. Esa capacidad lo hace especialmente adecuado para problemas donde las relaciones entre variables son complejas, no lineales o difíciles de describir mediante fórmulas analíticas exactas. (Géron, 2022)

En el contexto de los sistemas de control de tráfico, el aprendizaje automático ofrece una ventaja concreta: permite que el sistema aprenda de las condiciones reales del tránsito, como la composición del flujo, la presencia peatonal y la variabilidad horaria, y genere respuestas más adaptadas que las que puede producir un conjunto de reglas escritas a mano. Donde una regla explícita podría fallar por no contemplar una combinación de condiciones no prevista, un modelo entrenado sobre miles de casos extrapola con mayor robustez (Yohannes y Singh, 2019).

2.4.1. Tipos de aprendizaje automático

El aprendizaje automático se clasifica en tres grandes paradigmas, según la naturaleza de los datos disponibles y la forma en que el modelo construye su conocimiento. El aprendizaje supervisado trabaja con ejemplos etiquetados: el modelo recibe pares de entrada y salida esperada, y aprende la función que los relaciona. Los problemas de clasificación, determinar a qué categoría pertenece una observación, y los de regresión, estimar un valor continuo, son los casos típicos de este paradigma. El aprendizaje no supervisado, en cambio, trabaja con datos sin etiquetar y busca por sí mismo estructuras o patrones en ellos, como la agrupación en clústeres. El aprendizaje por

refuerzo, finalmente, entrena a un agente que interactúa con un entorno y aprende a tomar decisiones capaces de maximizar una recompensa acumulada a lo largo del tiempo.

Para esta investigación se adoptó el aprendizaje supervisado en sus dos variantes principales. Un clasificador aprende a asociar combinaciones de variables de entrada con una de las cuatro fases semafóricas posibles; un regresor aprende a estimar el tiempo de verde óptimo según esas mismas variables. Ambos son entrenados con datos etiquetados generados a partir de perfiles horarios representativos del tramo, lo que les permite generalizar a nuevas configuraciones de flujo que no estuvieron presentes durante el entrenamiento. (Géron, 2022; Sutton y Barto, 2018)

2.4.2. Árboles de decisión

Un árbol de decisión es una estructura jerárquica que representa una secuencia de decisiones basadas en el valor de variables de entrada. Cada nodo interno evalúa una condición sobre una variable y divide el espacio de datos en dos o más subconjuntos. Las ramas del árbol corresponden a los posibles resultados de esa condición, y los nodos hoja asignan una predicción o clase. Su principal ventaja es la interpretabilidad: el proceso de decisión puede seguirse paso a paso desde la raíz hasta la hoja, lo que hace transparente la lógica del modelo.

Sin embargo, los árboles de decisión son propensos al sobreajuste cuando se desarrollan sin restricciones: aprenden con demasiada precisión los datos de entrenamiento y pierden capacidad para generalizar. Para resolver ese problema se desarrollaron métodos de conjunto que combinan múltiples árboles, entre los que destaca el Random Forest por su robustez y su buen desempeño en una amplia variedad de problemas. (Breiman, 2001)

2.4.3. Random Forest

El Random Forest es un método de aprendizaje automático basado en la combinación de múltiples árboles de decisión, cada uno entrenado sobre una muestra aleatoria de los datos y con un

subconjunto aleatorio de variables de entrada en cada partición. Su fortaleza radica precisamente en esa aleatoriedad: al obligar a cada árbol a trabajar con una visión parcial e independiente de los datos, el conjunto del bosque produce predicciones más robustas que las que cualquier árbol individual podría generar. La predicción final emerge del consenso entre todos los árboles: por votación de mayoría en clasificación y por promedio en regresión. (Breiman, 2001)

Esta arquitectura le confiere al Random Forest propiedades especialmente útiles para el sistema propuesto. En primer lugar, reduce drásticamente el riesgo de sobreajuste que afecta a los árboles individuales, porque el promedio de muchos estimadores con distintos sesgos tiende a cancelar el ruido específico de cada uno. En segundo lugar, maneja bien las relaciones no lineales y las interacciones entre variables, algo importante en un sistema donde la combinación de composición del flujo, acumulación peatonal y esperas previas puede generar respuestas que no siguen una lógica lineal simple. En tercer lugar, proporciona una medida de importancia de variables que permite identificar cuáles entradas tienen más peso en la decisión, lo que refuerza la trazabilidad del modelo.

Desde el punto de vista práctico, el Random Forest no requiere normalización previa de las variables, tolera valores atípicos sin degradarse significativamente y puede entrenarse con eficiencia sobre conjuntos de datos de tamaño moderado. Para esta investigación se entrenaron dos modelos: un RandomForestClassifier de 100 estimadores para la selección de fase semafórica y un RandomForestRegressor de 100 estimadores para la estimación del tiempo de verde, ambos sobre un conjunto de 6 000 registros sintéticos generados a partir de perfiles horarios representativos del tramo de la Av. Ejército (Pedregosa et al., 2011).

2.4.4. Aplicación del aprendizaje automático al control de tráfico

La aplicación del aprendizaje automático al control de intersecciones semafóricas ha ganado una presencia creciente en la literatura especializada durante la última década. La razón es estructural: el problema de asignación óptima de fases y tiempos es dinámico, depende de múltiples variables simultáneas y cambia con cada ciclo de operación. Los enfoques de control clásico basados en tablas o en ciclos fijos no tienen la capacidad de adaptarse a esa variabilidad con la velocidad que el problema exige.

En ese contexto, los modelos de clasificación y regresión supervisada ofrecen una alternativa con ventajas claras: aprenden de datos históricos o sintéticos, generalizan a condiciones nuevas y producen decisiones en tiempo real con un costo computacional contenido. Estudios previos han demostrado que modelos basados en árboles de decisión y Random Forest pueden alcanzar altas precisiones en la clasificación de fases semafóricas cuando se entrenan con datos representativos de las condiciones del tramo. Gorodokin et al. (2021) demostraron la viabilidad de usar percepción visual como entrada para modelos de control adaptativo, y Smith et al. (2016) mostraron que incorporar la demanda peatonal como variable del modelo mejora la equidad y la seguridad del sistema. Ambos resultados son coherentes con el enfoque adoptado en esta investigación.

El valor del aprendizaje automático en este contexto no está solo en la precisión del modelo, sino en su capacidad de combinarse con capas de coherencia que aplican restricciones operativas y de seguridad sobre las propuestas del algoritmo. Esa combinación entre aprendizaje basado en datos y reglas determinísticas de validación es lo que le da al sistema tanto flexibilidad como trazabilidad metodológica.

2.4.5. Evaluación y validación de modelos de aprendizaje automático

La evaluación de modelos de aprendizaje automático es el proceso mediante el cual se estima el desempeño del modelo sobre datos no vistos durante el entrenamiento. La metodología de evaluación tiene implicaciones directas sobre la confiabilidad de las métricas reportadas: una evaluación incorrecta puede conducir a estimaciones demasiado optimistas del rendimiento real del modelo en producción.

La partición del conjunto de datos en subconjuntos de entrenamiento y prueba es la práctica más básica para evitar el sesgo de evaluación. En la partición simple, una fracción del dataset (típicamente entre el 70 % y el 80 %) se destina al entrenamiento y el resto a la evaluación. Para asegurar que ambos subconjuntos tengan distribuciones similares, la partición debe realizarse de manera estratificada cuando se trabaja con problemas de clasificación: cada clase debe estar representada en la misma proporción tanto en el conjunto de entrenamiento como en el de prueba. En esta investigación se empleó una partición 80/20 con estratificación por fase semafórica, garantizando que el clasificador fue evaluado sobre una muestra representativa de todas las fases. Para problemas de clasificación, las métricas de evaluación más relevantes son la exactitud global (accuracy), la precisión, la sensibilidad (recall) y el F1-score. La exactitud global mide la proporción de predicciones correctas sobre el total de predicciones realizadas. La precisión mide, para cada clase, qué proporción de los casos predichos como pertenecientes a esa clase son realmente correctos. La sensibilidad mide qué proporción de los casos verdaderamente pertenecientes a una clase fueron correctamente identificados. El F1-score es la media armónica entre precisión y sensibilidad, y resulta especialmente útil cuando las clases tienen frecuencias de aparición distintas. El F1-score macro promedia el F1-score de cada clase con igual peso, independientemente de su frecuencia, lo que es especialmente adecuado para el sistema propuesto dado que las cuatro fases semafóricas no tienen la misma frecuencia de activación.

Para problemas de regresión, las métricas de evaluación más empleadas son el error absoluto medio (MAE), el error cuadrático medio (MSE), la raíz del error cuadrático medio (RMSE) y el coeficiente de determinación (R^2). El MAE mide el promedio de los valores absolutos de los errores de predicción: es intuitivo porque está expresado en las mismas unidades que la variable objetivo y no penaliza los errores grandes de manera desproporcionada. El R^2 mide la proporción de la varianza de la variable objetivo que el modelo es capaz de explicar: un valor de $R^2 = 1$ implica predicción perfecta, mientras que $R^2 = 0$ implica que el modelo no explica nada de la variabilidad del objetivo. En el sistema propuesto, el regresor de tiempo verde obtuvo $R^2 = 0,9617$ y $MAE = 1,40$ s, lo que indica que el modelo explica el 96,17 % de la variabilidad en el tiempo de verde óptimo y que sus predicciones se desvían en promedio 1,40 segundos del valor correcto.

2.4.6. Interpretabilidad y trazabilidad de los modelos de árbol

La interpretabilidad es la propiedad de un modelo de aprendizaje automático que permite comprender por qué produce una determinada predicción. En aplicaciones críticas como el control de infraestructura urbana, la interpretabilidad no es un lujo sino un requisito: los operadores y responsables del sistema necesitan poder verificar que las decisiones del modelo son coherentes con el conocimiento del dominio y que no se basan en correlaciones espurias presentes en los datos de entrenamiento.

Los modelos basados en árboles de decisión y Random Forest tienen una ventaja natural en términos de interpretabilidad con respecto a las redes neuronales profundas. En un árbol individual, la ruta desde la raíz hasta una hoja representa exactamente el proceso de decisión: se pueden identificar las variables más determinantes, los umbrales de partición aplicados y la predicción asociada a cada combinación de condiciones. En un Random Forest, aunque no puede seguirse una ruta única de decisión, es posible calcular la importancia de cada variable como la reducción

promedio del criterio de separación ponderada por el número de nodos en los que esa variable fue utilizada para particionar. Esta medida de importancia proporciona una indicación de qué entradas tienen mayor peso en las decisiones del modelo.

En el sistema propuesto, la importancia de variables del Random Forest permite verificar que los factores operativamente relevantes, como la demanda ponderada, la acumulación peatonal y las esperas acumuladas, tienen mayor peso que factores secundarios como la cantidad de scooters en accesos de baja carga. Esta verificación es un componente de la trazabilidad metodológica del sistema. No basta con que el modelo exhiba buenas métricas de desempeño: es necesario, además, que sus decisiones sean explicables en términos del dominio de aplicación.

2.5. Visión por computadora aplicada al tránsito

La visión por computadora es uno de los componentes tecnológicos que más ha transformado los sistemas modernos de control vial. Su utilidad central es simple de describir: permite captar información visual del entorno, procesarla y convertirla en datos operativos que el sistema puede usar para tomar decisiones. En el contexto urbano esto es especialmente valioso porque el tránsito es heterogéneo, cambia constantemente y reacciona a condiciones del entorno que un semáforo de tiempo fijo no tiene forma de percibir. La diferencia con un sistema convencional es concreta: un semáforo programado con tiempos fijos no sabe cuántos vehículos hay en cada acceso, no distingue un bus de una motocicleta y no detecta si hay peatones esperando cruzar. (Castillo y Jiménez, 2019)

2.5.1. Imagen digital y procesamiento de imágenes

Una imagen digital es una cuadrícula de unidades mínimas llamadas píxeles, donde cada uno almacena información de brillo o color. En imágenes a color, esa información se expresa mediante la combinación de los canales rojo, verde y azul del modelo RGB. El procesamiento digital de imágenes consiste en aplicar operaciones sobre la imagen para mejorar su calidad, resaltar

características de interés o facilitar su interpretación. Un proceso particularmente crítico dentro de este conjunto es la segmentación: dividir la imagen en regiones u objetos de interés, distinguiendo qué partes corresponden al objeto que se quiere estudiar y cuáles son fondo. En un sistema de control de tráfico, esta distinción es esencial para identificar vehículos, peatones u otros actores viales dentro de la escena. (Castillo y Jiménez, 2019)

2.5.2. Detección y clasificación de actores viales

La detección y clasificación de actores viales es la aplicación directa de la visión por computadora al control de tránsito. Consiste en identificar objetos relevantes dentro de la escena vial y asignarles una categoría funcional según sus características visuales y su rol en el sistema de movilidad. En un entorno urbano real, los actores viales son heterogéneos y simultáneos: en la misma escena pueden estar circulando automóviles particulares, buses, cústeres, motocicletas, bicicletas, scooters y peatones, cada uno con comportamientos y necesidades de desplazamiento distintos. Un sistema de control que solo cuenta objetos sin distinguir sus tipos toma decisiones sobre información incompleta. (Viola y Jones, 2004; Redmon et al., 2016)

2.5.3. Redes neuronales y aprendizaje profundo aplicados al tránsito

Aunque el sistema propuesto en esta investigación utiliza modelos Random Forest como núcleo de decisión, resulta pertinente contextualizar ese enfoque dentro del panorama más amplio del aprendizaje profundo (deep learning). Este paradigma domina el estado del arte en tareas de percepción visual, que constituyen el módulo de entrada del sistema, y representa la dirección natural de evolución de la propuesta en implementaciones futuras.

Las redes neuronales artificiales son modelos computacionales inspirados vagamente en la organización del sistema nervioso biológico. Una red neuronal se compone de capas de unidades de procesamiento (neuronas artificiales) interconectadas, donde cada conexión tiene un peso

ajustable que determina la influencia de una neurona sobre otra. El entrenamiento de una red neuronal consiste en ajustar estos pesos para minimizar una función de pérdida definida sobre el conjunto de datos etiquetados. El algoritmo más empleado para este ajuste es la retropropagación del error (backpropagation), que aplica la regla de la cadena del cálculo diferencial para propagar el gradiente de la función de pérdida desde la capa de salida hasta las capas más profundas, permitiendo actualizar los pesos de cada capa de manera proporcional a su contribución al error total.

Las redes neuronales convolucionales (CNNs) son una arquitectura especializada para el procesamiento de imágenes. Su propiedad clave es la invarianza a la traslación: los filtros convolucionales aprenden a detectar características visuales (bordes, texturas, formas) independientemente de su posición en la imagen. Esto las hace especialmente adecuadas para la detección y clasificación de objetos en escenas viales. Modelos como YOLO (You Only Look Once), cuya primera versión fue presentada por Redmon et al. (2016), han transformado la detección de objetos en tiempo real al formular el problema como una regresión directa sobre una cuadrícula que cubre la imagen, permitiendo detectar múltiples objetos simultáneamente en una sola pasada por la red. Versiones posteriores como YOLOv5, YOLOv7 y YOLOv8 han refinado continuamente ese enfoque, mejorando la precisión y la velocidad de detección.

La elección de Random Forest sobre redes neuronales convolucionales en esta investigación responde a las condiciones del contexto de estudio. El sistema opera, en primer lugar, sobre datos tabulares, es decir, variables de entrada que representan conteos y ponderaciones, no directamente imágenes, después de que el módulo de visión ha realizado la detección y clasificación de actores viales. Para datos tabulares, los modelos de árbol logran un desempeño comparable o superior al de las redes neuronales con un costo computacional mucho menor. En segundo lugar, la disponibilidad de datos de entrenamiento es limitada: 6 000 casos sintéticos bastan para entrenar

un Random Forest, pero resultan insuficientes para una red neuronal profunda con buenos resultados de generalización. Y en tercer lugar, la interpretabilidad del Random Forest facilita la verificación técnica y la supervisión del sistema, un requisito crítico para aplicaciones de infraestructura urbana.

2.6. Detección peatonal y redundancia sensorial

Un sistema de control de tráfico que solo responde al flujo vehicular tiene un problema evidente: ignora a uno de los actores más importantes de la vía pública. Los peatones no son un elemento secundario de la gestión del tránsito, sino una condición crítica que el sistema debe incorporar activamente en sus decisiones. La redundancia sensorial consiste en usar más de una fuente de detección para observar o validar una misma condición del entorno. Su propósito es incrementar la confiabilidad del sistema, reducir la dependencia de un único sensor y fortalecer la robustez de la toma de decisiones ante fallas parciales, lecturas ambiguas o situaciones de incertidumbre. En esta investigación, la propuesta combina detección principal mediante visión por computadora y detección redundante mediante sensor térmico. (Smith et al., 2016)

2.6.1. Detección térmica como mecanismo de validación

La detección térmica se basa en la captación de radiación infrarroja asociada a la temperatura de los cuerpos. En aplicaciones de tránsito puede funcionar como mecanismo auxiliar para reconocer presencia humana en zonas específicas de la escena, especialmente cuando se busca una validación adicional a la que ofrece una cámara convencional. A diferencia de la cámara, un sensor térmico responde a las variaciones de calor emitidas por los cuerpos con independencia de la luz ambiente, lo que lo hace útil en condiciones donde la percepción visual puede verse comprometida: sombras, saturación de la escena o dificultad para distinguir la silueta de un peatón en un entorno urbano cargado de información visual.

2.6.2. Fuentes de incertidumbre en la detección sensorica

Todo sistema de percepción opera bajo condiciones de incertidumbre: los sensores tienen limitaciones de resolución, ruido inherente, rangos de operación definidos y susceptibilidad a condiciones adversas del entorno. En un sistema de control de tráfico instalado en un entorno urbano exterior, las fuentes de incertidumbre más relevantes son las siguientes: variaciones de iluminación (transición entre el día y la noche, sombras proyectadas por edificios o vehículos de gran porte, efectos de deslumbramiento); condiciones climáticas (lluvia, neblina, polvo en suspensión que puede degradar la imagen capturada por la cámara); oclusiones parciales (un vehículo más grande puede bloquear la vista de uno más pequeño, haciendo que este no sea detectado); y saturación del campo de visión (cuando el número de actores viales en escena supera la capacidad de discriminación del sistema de visión).

La redundancia sensorica es la estrategia estándar para mitigar el impacto de estas fuentes de incertidumbre. Al combinar dos o más fuentes de detección basadas en principios físicos distintos, una cámara visible y un sensor térmico en el caso del sistema propuesto, este puede resolver situaciones ambiguas por votación o por prioridad según la confianza de cada fuente. El sensor térmico, al operar sobre la radiación infrarroja emitida por los cuerpos, es insensible a las variaciones de iluminación visible, lo que lo convierte en un complemento ideal para la cámara en condiciones nocturnas o de deslumbramiento. Con todo, tampoco el sensor térmico es infalible: las superficies calientes no humanas, como el motor de un vehículo detenido, pueden generar falsos positivos de presencia peatonal. Por eso el sistema propuesto no lo utiliza como fuente primaria de detección, sino como mecanismo de validación de la detección visual: ambas fuentes deben coincidir para que el sistema tome una decisión firme sobre la condición peatonal.

2.7. Arquitectura de comunicación e integración del sistema

Un sistema inteligente de control de tráfico no es solo un conjunto de módulos independientes: es la forma en que esos módulos se conectan y se comunican. La arquitectura de comunicación e integración es el componente que hace posible que la información captada por los dispositivos de percepción llegue al módulo de control, que las decisiones del sistema se representen en la interfaz de supervisión y que el conjunto opere de manera coherente y coordinada. La integración lógica entre módulos puede organizarse mediante una API REST con intercambio en formato JSON, una solución liviana, legible e independiente del lenguaje de programación.

2.8. Comparación con el control de tiempo fijo

El control de tiempo fijo es el esquema semafórico más común en el Perú y en la mayoría de ciudades latinoamericanas. Funciona con ciclos predeterminados cuya duración no cambia en función del tráfico: el mismo tiempo de verde para un acceso a las 7 de la mañana con cuarenta buses acumulados que a las 11 de la noche con dos automóviles. Su ventaja es la simplicidad: es fácil de programar, fácil de mantener y no requiere infraestructura adicional. Su limitación es exactamente esa misma rigidez. Cuando el flujo vehicular y peatonal cambia de manera importante a lo largo del tiempo, el sistema de tiempo fijo no responde. El control adaptativo ofrece una respuesta diferente: ajusta la prioridad y la duración del verde según lo que detecta en el entorno, reduciendo esperas innecesarias y respondiendo con mayor coherencia a los cambios reales de la circulación. (Transportation Research Board, 2010; MTC, 2016)

2.9. Normativa y estándares técnicos de referencia

El diseño de sistemas de control semafórico en el Perú está regulado por el Manual de Dispositivos de Control del Tránsito Automotor para Calles y Carreteras del Ministerio de Transportes y Comunicaciones (MTC, 2016). Este manual establece los requisitos técnicos para la instalación,

operación y mantenimiento de los dispositivos de control del tránsito, incluyendo los semáforos para vehículos y peatones. Entre los parámetros normados por el manual se encuentran las duraciones mínimas y máximas del tiempo de verde, la duración del tiempo de amarillo (no inferior a 3 segundos), las condiciones de instalación de los dispositivos y los requisitos de visibilidad.

A nivel internacional, el Highway Capacity Manual (HCM 2010), publicado por el Transportation Research Board de los Estados Unidos, es la referencia técnica más ampliamente utilizada para el análisis operativo de intersecciones semaforizadas. El HCM establece los procedimientos para calcular el nivel de servicio de una intersección en función del tiempo de demora por vehículo, definiendo seis niveles de servicio (A a F) según rangos de tiempo de espera. Estos procedimientos han sido adaptados en el marco teórico del presente trabajo para definir los criterios de evaluación del desempeño del sistema propuesto en cada escenario de simulación.

Desde la perspectiva de los sistemas inteligentes de transporte, los estándares de arquitectura de referencia más utilizados son el NTCIP (National Transportation Communications for ITS Protocol) en América del Norte y el OCIT (Open Communication Interface for Road Traffic Control Systems) en Europa. Ambos definen protocolos de comunicación, formatos de datos y estructuras de control que permiten la interoperabilidad entre equipos de distintos fabricantes. En el caso del sistema propuesto, la arquitectura de integración mediante API REST con formato JSON es coherente con los principios de apertura y escalabilidad promovidos por estos estándares, aunque para una implementación física definitiva sería necesario evaluar la compatibilidad con los protocolos específicos del controlador semafórico instalado en el nodo de la Av. Ejército.

2.10. Fundamentos del flujo vehicular

El diseño de cualquier sistema de control del tránsito, incluido el propuesto en esta investigación, se apoya en un conjunto de variables que describen el comportamiento del flujo vehicular en una

vía. Comprender cómo se relacionan estas variables permite entender por qué un semáforo mal temporizado genera congestión y por qué una asignación de tiempos ajustada a la demanda mejora la circulación. Esta sección presenta los fundamentos de la teoría del flujo vehicular que sustentan el análisis operativo del tramo estudiado.

2.10.1. Variables fundamentales del tránsito

El estado del tránsito en un punto o tramo de la vía se describe mediante tres variables macroscópicas: el flujo, la densidad y la velocidad. Estas variables constituyen el lenguaje básico de la ingeniería de tránsito y permiten cuantificar el comportamiento de la circulación.

El flujo (q), también llamado intensidad o volumen, es la cantidad de vehículos que pasan por una sección de la vía durante un intervalo de tiempo determinado. Se expresa habitualmente en vehículos por hora (veh/h) y se obtiene dividiendo el número de vehículos observados (N) entre el tiempo de observación (t), de modo que $q = N / t$. En el tramo de la Av. Ejército, los aforos registraron flujos de hasta 4 560 veh/h en hora punta, lo que refleja una alta intensidad de circulación y justifica la necesidad de un control eficiente.

La densidad (k) es el número de vehículos presentes en una unidad de longitud de la vía en un instante dado, y se expresa en vehículos por kilómetro (veh/km). Una densidad baja indica una vía despejada; una densidad alta, próxima a la de congestión, indica que los vehículos circulan muy cerca unos de otros. Aunque la densidad es difícil de medir directamente en campo, resulta clave porque describe el grado de ocupación real de la vía.

La velocidad (v) es la rapidez media a la que se desplazan los vehículos, expresada en kilómetros por hora (km/h). En condiciones de flujo libre la velocidad es alta; a medida que aumenta la densidad, la velocidad disminuye progresivamente hasta que los vehículos se detienen cuando la vía se satura.

2.10.2. Relación fundamental del tránsito

Las tres variables anteriores no son independientes: se vinculan mediante la relación fundamental del tránsito, expresada como $q = k \times v$. Es decir, el flujo es igual al producto de la densidad por la velocidad. Esta ecuación explica un fenómeno cotidiano: el flujo resulta bajo tanto cuando hay pocos vehículos, porque la densidad es baja, como cuando la vía está congestionada, porque la velocidad es casi nula, y alcanza su valor máximo en un punto intermedio de equilibrio.

La representación gráfica de estas relaciones se conoce como diagrama fundamental del tránsito. El modelo clásico de Greenshields (1935) propone una relación lineal entre la velocidad y la densidad: a mayor densidad, menor velocidad, hasta llegar a la densidad de congestión, donde la velocidad se anula. A partir de esa relación se deduce que la curva que vincula el flujo con la densidad tiene forma de parábola: el flujo crece con la densidad hasta alcanzar la capacidad de la vía y luego decrece a medida que la congestión se instala. El punto más alto de esa curva corresponde a la capacidad, esto es, al flujo máximo que la vía puede sostener.

Este comportamiento tiene una implicación directa para el control semafórico: mantener el flujo cerca de la capacidad, evitando que la densidad supere el umbral de congestión, es precisamente el objetivo de una buena asignación de tiempos. Un semáforo de tiempo fijo que no responde a la densidad real puede empujar a la vía hacia la zona congestionada; un sistema adaptativo, en cambio, busca operar en la zona de máximo flujo, adaptando el reparto del ciclo a la demanda del momento.

2.10.3. Capacidad vial y nivel de servicio

La capacidad de una vía es el número máximo de vehículos que pueden atravesar una sección en condiciones dadas durante un periodo determinado. En una intersección semaforizada, la capacidad de un acceso depende de dos factores: el flujo de saturación, la tasa máxima a la que los vehículos pueden cruzar mientras el semáforo está en verde, y la proporción de tiempo de verde que ese

acceso recibe dentro del ciclo. De ahí que un acceso con mucha demanda, pero poco verde, tenga una capacidad efectiva reducida y forme cola.

Para evaluar la calidad de la operación, la ingeniería de tránsito emplea el concepto de nivel de servicio, formalizado por el Highway Capacity Manual (Transportation Research Board, 2010). El nivel de servicio clasifica el desempeño de una intersección en seis categorías, de la A a la F, según la demora media que experimenta cada vehículo. El nivel A corresponde a una circulación fluida con demoras mínimas, mientras que el nivel F representa una intersección saturada, con demoras muy elevadas y colas persistentes. El manual asocia rangos de demora por vehículo a cada categoría, de modo que a menor demora, mejor nivel de servicio.

En esta investigación, el nivel de servicio proporciona el marco conceptual para interpretar los tiempos de espera estimados: la reducción de la demora que logra el sistema adaptativo puede leerse como una mejora en el nivel de servicio del acceso, es decir, como un avance hacia una categoría de operación más eficiente respecto del esquema de tiempo fijo.

2.11. Temporización semafórica y teoría de demoras

Un semáforo no es únicamente un dispositivo que enciende luces: es un repartidor de tiempo entre movimientos que compiten por el mismo espacio. La manera en que se distribuye ese tiempo determina cuánto espera cada usuario y qué tan eficiente resulta la intersección. Esta sección describe los elementos que componen la temporización semafórica y los métodos clásicos para calcular el ciclo y estimar la demora, que sirven de fundamento a la lógica adaptativa propuesta en el presente trabajo.

2.11.1. Elementos de la temporización semafórica

La operación de un semáforo se organiza en torno a varios conceptos básicos. El ciclo es el tiempo total que tarda el semáforo en completar una secuencia completa de indicaciones y regresar a su

punto de partida; en el esquema de tiempo fijo de referencia empleado en esta tesis, el ciclo dura 94 segundos. La fase es cada uno de los intervalos del ciclo durante los cuales un conjunto de movimientos recibe derecho de paso de manera simultánea, agrupando accesos que pueden circular sin conflicto entre sí.

El reparto o distribución del ciclo, conocido en la literatura como split, es la proporción del ciclo asignada a cada fase. Repartir bien el ciclo significa otorgar más verde a los accesos con mayor demanda y menos a los que tienen poca, que es justamente lo que realiza el sistema propuesto mediante la demanda ponderada. El entretiempo o intervalo de cambio es el tiempo de transición entre una fase y la siguiente; comprende la luz ámbar y, cuando es necesario, un intervalo de todo-rojo que despeja la intersección antes de dar paso al siguiente movimiento. La normativa del Ministerio de Transportes y Comunicaciones (2016) establece que la luz ámbar no debe ser menor a tres segundos, valor que se respeta en el diseño.

El tiempo perdido es la parte del ciclo que no se aprovecha para el movimiento efectivo de vehículos, a causa de los arranques y de los intervalos de cambio. Cuanto mayor es el número de fases, mayor es el tiempo perdido total; por ello, un diseño eficiente busca el menor número de fases compatible con la seguridad de todos los usuarios, incluidos los peatones.

2.11.2. Cálculo del ciclo óptimo: el método de Webster

Uno de los aportes más influyentes de la ingeniería de tránsito es el método propuesto por Webster (1958) para determinar la duración del ciclo que minimiza la demora total en una intersección semaforizada. El método parte de los flujos de cada acceso y de sus flujos de saturación para calcular el grado de saturación de cada movimiento, y a partir de ellos estima el ciclo óptimo mediante una expresión que relaciona el tiempo perdido total con la suma de las relaciones de flujo de los movimientos críticos.

En términos generales, el ciclo óptimo crece cuando aumentan la demanda o el tiempo perdido, y disminuye cuando la intersección está poco cargada. Esta es una idea central para el presente trabajo: el ciclo no debería ser el mismo a toda hora, sino ajustarse a la demanda real. Un semáforo de tiempo fijo con un ciclo de 94 segundos aplica ese mismo valor tanto en la hora punta como en la madrugada, lo que en condiciones de baja demanda produce esperas innecesarias. El sistema adaptativo propuesto se inspira en este principio al acortar o extender el verde según la demanda detectada, en lugar de mantener un ciclo constante e insensible a las condiciones de la vía.

2.11.3. Estimación de la demora en intersecciones semaforizadas

La demora es el tiempo adicional que un vehículo pierde a causa del control semafórico, respecto del que tardaría si circulara sin interrupciones. Constituye el indicador más directo de la calidad de operación de una intersección y es la base del nivel de servicio descrito en la sección anterior.

Webster (1958) propuso además una fórmula para estimar la demora media por vehículo en una intersección semaforizada, en función de la duración del ciclo, de la proporción de verde asignada al acceso y del grado de saturación. La expresión evidencia que la demora crece con rapidez a medida que el acceso se aproxima a la saturación, y que un reparto inadecuado del verde penaliza a los accesos con mayor demanda, generando colas que el ciclo no alcanza a despejar.

Una aproximación simplificada, útil para comprender el orden de magnitud de la espera, considera que un vehículo que llega en un momento cualquiera del ciclo aguarda, en promedio, una fracción del tiempo en que su acceso permanece en rojo. Cuanto mayor es ese tiempo en rojo dentro del ciclo, mayor es la espera promedio; en consecuencia, acortar el ciclo o aumentar la proporción de verde de un acceso reduce su demora. Los tiempos de espera estimados que se presentan en el Capítulo III se derivan de esta lógica: se calculan a partir del ciclo resultante de cada escenario, y

su reducción frente al esquema de tiempo fijo refleja una asignación de tiempos más ajustada a la demanda real del tramo.

2.12. Equivalencia vehicular y control semafórico adaptativo

La composición heterogénea del tránsito y la necesidad de responder a una demanda cambiante conducen a dos conceptos centrales para esta investigación: la equivalencia entre distintos tipos de vehículos y las estrategias de control capaces de adaptarse en tiempo real. Ambos fundamentan, respectivamente, la ponderación de actores viales y la lógica adaptativa del sistema propuesto.

2.12.1. Unidades de vehículo de pasajeros y factores de equivalencia

El tránsito real está compuesto por vehículos de tamaños y comportamientos muy distintos, que no ocupan ni afectan la vía de la misma manera. Para poder sumarlos en una medida común, la ingeniería de tránsito utiliza el concepto de unidad de vehículo de pasajeros (uvp), conocido también por su sigla inglesa PCU. Bajo este enfoque, cada tipo de vehículo se convierte a un número equivalente de automóviles mediante un factor de equivalencia que refleja el espacio que ocupa y la perturbación que causa al flujo. Un automóvil ligero equivale a una unidad, mientras que un vehículo pesado, como un bus, equivale a varias, porque ocupa más longitud, acelera más lentamente y tarda más en despejar la intersección.

El Highway Capacity Manual (Transportation Research Board, 2010) y los manuales de ingeniería de tránsito emplean estos factores para calcular la capacidad y la demanda de una vía en términos homogéneos. La ponderación de actores viales adoptada en esta investigación se apoya directamente en ese concepto. Los valores asignados a cada tipo de usuario, 2,5 al bus o combi grande, 2,0 al cúster, 1,0 al automóvil, 0,5 a la motocicleta, 0,4 a la bicicleta y 0,3 al scooter, constituyen factores de equivalencia operativa que traducen la composición heterogénea del flujo

a una medida común de presión sobre el acceso. La demanda ponderada no es así un simple conteo de unidades, sino una estimación de la carga real que cada acceso impone al sistema de control.

2.12.2. Estrategias de control semafórico

El control de una intersección semaforizada puede abordarse mediante distintas estrategias, que se diferencian por su capacidad de responder a la demanda. El control de tiempo fijo ejecuta ciclos predeterminados calculados a partir de datos históricos, sin considerar las variaciones del momento; es sencillo y económico, pero ineficiente cuando la demanda fluctúa. El control actuado por tránsito incorpora detectores que informan de la presencia de vehículos, de modo que el semáforo puede extender o acortar una fase según la demanda detectada en cada acceso.

El control coordinado sincroniza varios semáforos consecutivos para que un vehículo que circula a cierta velocidad encuentre una sucesión de luces verdes, fenómeno conocido como onda verde. Finalmente, el control adaptativo ajusta de manera continua y en tiempo real la duración del ciclo, el reparto de las fases y el desfase entre intersecciones, optimizando la operación ante condiciones cambiantes. El sistema propuesto en esta tesis se inscribe en esta última categoría, la más avanzada, pues modifica la asignación de tiempos en cada ciclo en función de la demanda ponderada y de la presencia peatonal detectadas en el momento.

2.12.3. Sistemas adaptativos de referencia

En el ámbito internacional, los sistemas de control adaptativo más consolidados son SCOOT (Split, Cycle and Offset Optimisation Technique), desarrollado en el Reino Unido, y SCATS (Sydney Coordinated Adaptive Traffic System), desarrollado en Australia. Ambos ajustan de forma continua el reparto del ciclo, su duración y el desfase entre intersecciones a partir de la información de detectores instalados en la vía, y han demostrado a escala de ciudad la viabilidad y el beneficio del control adaptativo frente al de tiempo fijo.

La propuesta de esta investigación se ubica en la misma línea conceptual, pero aplicada a una intersección específica y apoyada en visión por computadora y aprendizaje automático en lugar de detectores de lazo enterrados en el pavimento. Este enfoque reduce la necesidad de infraestructura invasiva y aporta información más rica sobre la composición del flujo y la presencia peatonal, lo que permite decisiones más sensibles al contexto que las de un sistema basado únicamente en el conteo de vehículos.

2.13. Seguridad peatonal y movilidad urbana sostenible

Un sistema de control de tránsito no puede evaluarse solo por su eficiencia vehicular. La seguridad del peatón y el impacto del tránsito sobre la ciudad son dimensiones igualmente relevantes, y ambas orientan el diseño del sistema propuesto.

2.13.1. El peatón en el diseño de intersecciones

El peatón es el usuario más vulnerable del sistema vial y, sin embargo, durante décadas el diseño semafórico priorizó el flujo vehicular. La ingeniería de tránsito contemporánea reconoce que una intersección segura debe garantizar tiempos de cruce suficientes y proteger al peatón de los conflictos con los vehículos. El tiempo de cruce peatonal se determina a partir del ancho de la vía y de una velocidad de marcha de referencia, de modo que una persona pueda completar el cruce con seguridad, considerando además a quienes caminan más lento.

Los conflictos peatón-vehículo se producen cuando ambos movimientos coinciden en el mismo espacio y tiempo, situación que el diseño debe evitar mediante fases peatonales protegidas. En el sistema propuesto, la detección redundante de peatones mediante cámara y sensor térmico, y la activación de una fase de cruce cuando su presencia es validada por ambos módulos, responden directamente a este principio de seguridad: ante la presencia confirmada de un peatón, su protección prevalece sobre la fluidez vehicular.

2.13.2. Congestión, externalidades y movilidad sostenible

La congestión no es únicamente una molestia: genera costos económicos, ambientales y sociales.

El tiempo perdido en las colas se traduce en pérdidas de productividad; el funcionamiento prolongado de los motores en ralentí incrementa el consumo de combustible y las emisiones de gases contaminantes; y la saturación de las vías deteriora la calidad de vida urbana. Estas consecuencias, denominadas externalidades del tránsito, explican por qué mejorar la eficiencia del control semafórico tiene un impacto que trasciende la simple reducción de la espera.

El enfoque de movilidad urbana sostenible busca reducir estas externalidades mediante una gestión más inteligente de la infraestructura existente, antes de recurrir a costosas ampliaciones viales. Un sistema que disminuye las esperas y prioriza al peatón contribuye a ese objetivo al aprovechar mejor la capacidad disponible del cruce, en lugar de demandar nueva infraestructura. En este sentido, la propuesta se alinea con las tendencias actuales de gestión del tránsito, que privilegian la optimización sobre la expansión.

2.14. Marco normativo nacional e internacional aplicado al proyecto

El diseño del sistema propuesto se enmarca en un conjunto de normas y estándares que regulan el control del tránsito. Su cumplimiento garantiza que la propuesta sea, además de técnicamente viable, compatible con los marcos regulatorios aplicables.

2.14.1. Normativa nacional

En el Perú, el principal referente para el diseño de dispositivos de control del tránsito es el Manual de Dispositivos de Control del Tránsito Automotor para Calles y Carreteras del Ministerio de Transportes y Comunicaciones (2016). Este documento regula las características, la ubicación y la operación de las señales y los semáforos, y establece criterios como el tiempo mínimo de luz ámbar de tres segundos, la estructura de fases y las condiciones de seguridad para el cruce peatonal, todos

ellos respetados en el diseño del sistema. De manera complementaria, el Reglamento Nacional de Tránsito define las reglas de circulación y las prioridades de paso que todo dispositivo de control debe observar, mientras que la normativa de accesibilidad establece condiciones para que las personas con movilidad reducida puedan cruzar con seguridad.

2.14.2. Referencias internacionales

A nivel internacional, esta investigación toma como referencia el Highway Capacity Manual (Transportation Research Board, 2010), que aporta la metodología de niveles de servicio empleada para evaluar el desempeño de la intersección. El Manual on Uniform Traffic Control Devices, de los Estados Unidos, constituye otro referente ampliamente utilizado para la normalización de dispositivos de control del tránsito. En cuanto a la comunicación entre equipos, el estándar NTCIP define protocolos que permiten la interoperabilidad entre dispositivos de distintos fabricantes, aspecto relevante para la eventual integración del sistema propuesto en una red de semáforos coordinados. La combinación de la normativa nacional con estas referencias internacionales sustenta la solidez técnica y regulatoria de la propuesta.

2.15. Geometría de la intersección y flujo de saturación

La capacidad de una intersección no depende únicamente de la temporización del semáforo, sino también de su geometría: el número y el ancho de los carriles, los radios de giro, la existencia de carriles exclusivos y la disposición de los movimientos. Esta sección describe cómo la configuración física del cruce condiciona su capacidad y cómo se mide el flujo de saturación, concepto clave para dimensionar los tiempos de verde.

2.15.1. Elementos geométricos que condicionan la capacidad

El ancho y el número de carriles de cada acceso determinan cuántos vehículos pueden avanzar simultáneamente durante el verde. Los carriles exclusivos de giro evitan que un vehículo que gira

bloquee a los que siguen de frente, con lo que mejora el flujo del acceso. Los radios de giro y la presencia de islas o refugios peatonales influyen en la velocidad de maniobra y en la seguridad de los cruces. En el nodo de la Av. Ejército, la confluencia de tres accesos con movimientos de giro y cruce peatonal define una geometría compleja que justifica un control capaz de discriminar entre los distintos movimientos y de asignar el tiempo según su demanda.

2.15.2. Flujo de saturación y tiempo de verde efectivo

El flujo de saturación es la tasa máxima a la que los vehículos pueden cruzar la línea de detención de un acceso cuando el semáforo está en verde y existe una cola continua, y se expresa en vehículos por hora de verde. Se determina a partir de un valor base por carril, ajustado por factores como el ancho del carril, la pendiente de la vía, la proporción de vehículos pesados y la presencia de movimientos de giro. El tiempo de verde efectivo, por su parte, es la porción del verde realmente aprovechable para el movimiento, una vez descontado el tiempo perdido en el arranque de la cola. La capacidad de un acceso resulta del producto de su flujo de saturación por la proporción de verde efectivo que recibe dentro del ciclo. Este principio explica por qué asignar más verde a un acceso saturado incrementa su capacidad y reduce su cola, mientras que quitarle verde a un acceso con poca demanda no lo perjudica de forma apreciable. Sobre esta relación se apoya la asignación adaptativa de tiempos del sistema propuesto, que redistribuye el verde hacia los accesos de mayor demanda ponderada.

2.16. Teoría de colas y propagación de la congestión

Cuando la demanda de un acceso supera su capacidad durante el verde, se forma una cola que puede crecer ciclo a ciclo. Comprender cómo se forman y se disipan las colas, y cómo se propaga la congestión a lo largo de la vía, es esencial para evaluar el desempeño de una intersección y para justificar la necesidad de un control adaptativo.

2.16.1. Formación y disipación de colas

Durante la fase de rojo, los vehículos que llegan a un acceso se acumulan formando una cola. Al iniciarse el verde, la cola comienza a descargarse a la tasa del flujo de saturación. Si el tiempo de verde es suficiente para despejar toda la cola antes de que termine la fase, el acceso opera sin saturación; si no lo es, una parte de la cola permanece para el siguiente ciclo, fenómeno conocido como cola residual o desbordamiento. La acumulación de colas residuales ciclo tras ciclo es la manifestación característica de una intersección saturada y el principal síntoma de un reparto de tiempos inadecuado, que un control sensible a la demanda busca evitar.

2.16.2. Ondas de choque y propagación de la congestión

La transición entre distintos estados del tránsito, por ejemplo entre la circulación libre y la cola detenida, se propaga a lo largo de la vía en forma de ondas de choque, un concepto derivado de la teoría del flujo vehicular. Cuando una cola crece, su extremo posterior retrocede aguas arriba como una onda; cuando el semáforo libera el flujo, una onda de arranque avanza en sentido contrario y disuelve la cola.

Si las colas de una intersección alcanzan a la intersección anterior, la congestión se propaga y puede llegar a bloquear toda una arteria. Este mecanismo explica por qué un solo cruce mal gestionado es capaz de degradar el desempeño de un corredor completo, y por qué un control que evita la formación de colas excesivas produce beneficios que trascienden la propia intersección. La reducción de las esperas que logra el sistema propuesto contribuye, en este sentido, a limitar la propagación de la congestión hacia los tramos vecinos.

2.17. Coordinación semafórica y progresión del tránsito

Cuando varias intersecciones semaforizadas se ubican próximas entre sí a lo largo de una arteria, su operación conjunta puede optimizarse mediante la coordinación, de manera que un vehículo encuentre una sucesión de luces verdes en su recorrido.

2.17.1. Onda verde y desfase

La coordinación se basa en el desfase, conocido en la literatura como offset, que es la diferencia de tiempo entre el inicio del verde de una intersección y el de la siguiente. Ajustando los desfases según la distancia entre cruces y la velocidad de circulación, es posible crear una onda verde o progresión, en la que un pelotón de vehículos avanza sin detenerse a lo largo del corredor. El ancho de banda de la progresión mide la duración del intervalo durante el cual un vehículo puede recorrer la arteria encontrando siempre luz verde, y constituye un indicador de la calidad de la coordinación.

2.17.2. Relevancia para el sistema propuesto

Aunque esta investigación se centra en una intersección aislada, la arquitectura del sistema contempla su comunicación mediante protocolos estandarizados, lo que permitiría su integración futura en un esquema coordinado. La imagen referencial de conexión entre semáforos del corredor, incluida en los resultados, apunta precisamente a esa posibilidad de escalar la solución de una intersección a un corredor completo, aprovechando los principios de coordinación y progresión descritos en esta sección.

2.18. Tecnologías de detección en sistemas inteligentes de transporte

Todo sistema de control adaptativo requiere información sobre el estado del tránsito, que obtiene mediante tecnologías de detección. La elección de la tecnología condiciona tanto la calidad de la información disponible como el costo de la infraestructura necesaria.

2.18.1. Tecnologías tradicionales de detección

El detector de lazo inductivo, instalado bajo el pavimento, es la tecnología más difundida: detecta la presencia y el paso de vehículos con alta fiabilidad, pero requiere obras invasivas para su instalación y un mantenimiento costoso que implica intervenir la calzada. Los detectores de radar y de microondas miden la presencia y la velocidad de los vehículos sin contacto con el pavimento, con buena tolerancia a las condiciones climáticas. Los sensores de video permiten, además de detectar, cubrir varios carriles con un solo equipo y ofrecer información visual del tramo.

2.18.2. Detección basada en visión y sensores térmicos

La detección por visión por computadora, adoptada en esta investigación, ofrece información más rica que los detectores tradicionales: no solo cuenta vehículos, sino que los clasifica por tipo y detecta la presencia de peatones, lo que resulta indispensable para una gestión que prioriza la seguridad. Su principal limitación es la sensibilidad a las condiciones de iluminación y a las oclusiones. Por ello, el sistema propuesto la complementa con un sensor térmico, que detecta la presencia de personas por su emisión de calor y opera de forma fiable incluso de noche o con escasa visibilidad. La combinación de ambas tecnologías proporciona una detección redundante y robusta que supera las limitaciones de cada una por separado y evita las obras invasivas propias de los lazos inductivos, en línea con la factibilidad técnica y económica del proyecto.

2.19. Metodología de estudios de tránsito y aforos

La información que alimenta cualquier sistema de control proviene de estudios de tránsito, entre los cuales el aforo vehicular es el más básico y utilizado. Esta sección describe los métodos de recolección de datos de tránsito que sustentan la base empírica de la presente investigación.

2.19.1. El aforo vehicular

El aforo vehicular consiste en contar y clasificar los vehículos que pasan por una sección de la vía durante intervalos de tiempo definidos, habitualmente de quince minutos. Los aforos pueden realizarse de forma manual, mediante observadores en campo, o de forma automatizada, con detectores o análisis de video. La clasificación por tipo de vehículo permite conocer no solo el volumen, sino la composición del flujo, dato esencial para la ponderación de actores viales. En esta investigación, los aforos se realizaron en el nodo de la Av. Ejército durante las franjas de mayor y menor demanda, cubriendo los tres accesos y clasificando los vehículos en intervalos de quince minutos.

2.19.2. Periodos de estudio y hora punta

Los estudios de tránsito distinguen entre la hora punta, que concentra la mayor demanda del día, y los periodos valle o de baja demanda. Identificar la hora punta es crucial, porque el dimensionamiento del control debe garantizar un desempeño aceptable en las condiciones más exigentes. Los aforos del tramo identificaron tres periodos punta, matutino, del mediodía y vespertino, con volúmenes de hasta 4 560 veh/h, que sirvieron de base para construir los escenarios de simulación de mayor demanda y para calibrar la respuesta esperada del sistema.

2.20. Indicadores de desempeño operativo del tránsito

Para evaluar objetivamente el funcionamiento de una intersección y comparar alternativas de control se emplean indicadores de desempeño, que traducen el comportamiento del tránsito en medidas cuantificables y comparables.

2.20.1. Principales indicadores

La demora media por vehículo es el indicador más utilizado, pues refleja de manera directa el tiempo perdido a causa del control y determina el nivel de servicio. El número de detenciones mide

cuántas veces, en promedio, un vehículo debe detenerse al atravesar la intersección, y se relaciona con el consumo de combustible y las emisiones contaminantes. La longitud de cola indica el número máximo de vehículos acumulados en un acceso y permite verificar si la cola desborda hacia las intersecciones vecinas. El grado de saturación, que compara la demanda con la capacidad, señala qué tan cerca se encuentra un acceso de su límite operativo.

2.20.2. Indicadores empleados en la investigación

El sistema propuesto se evalúa mediante indicadores coherentes con esta práctica: el tiempo de espera estimado por acceso, el comportamiento del flujo vehicular, la prioridad de fase asignada y un índice de desempeño integral que resume la operación de la intersección. La comparación de estos indicadores entre el sistema adaptativo y el esquema de tiempo fijo, a lo largo de los siete escenarios de simulación, proporciona la base cuantitativa para valorar la mejora lograda, en línea con la metodología de evaluación aceptada en la ingeniería de tránsito.

2.21. Modelos de simulación del tránsito

La evaluación de alternativas de control de tránsito rara vez puede realizarse experimentando de forma directa sobre la vía, por el costo y el riesgo que ello implicaría. Por esa razón, la ingeniería de tránsito recurre a la simulación, que reproduce el comportamiento del tránsito en un entorno controlado. Esta sección describe los tipos de modelos de simulación y fundamenta su empleo como método de validación en la presente investigación.

2.21.1. Tipos de modelos de simulación

Los modelos de simulación del tránsito se clasifican según su nivel de detalle. Los modelos macroscópicos representan el tránsito como un flujo continuo, describiendo variables agregadas como el flujo, la densidad y la velocidad, y resultan útiles para analizar redes extensas. Los modelos microscópicos simulan el comportamiento individual de cada vehículo y conductor, reproduciendo

maniobras como el seguimiento vehicular, el cambio de carril y la respuesta al semáforo, con gran detalle a costa de una mayor complejidad computacional. Los modelos mesoscópicos se sitúan en un nivel intermedio, agrupando los vehículos en pelotones. Existen diversas herramientas de simulación microscópica que permiten evaluar estrategias de control antes de su implementación en campo.

2.21.2. La simulación como método de validación

La simulación permite comparar alternativas bajo condiciones idénticas y repetibles, algo imposible en la vía real, donde las condiciones cambian de un instante a otro. En esta investigación, la simulación reproduce la operación del cruce bajo el control adaptativo y bajo el esquema de tiempo fijo, aplicando a ambos las mismas condiciones de demanda, de modo que las diferencias observadas sean atribuibles a la lógica de control y no a factores externos. Esta es la razón por la cual la evaluación por simulación constituye un método aceptado y riguroso para valorar la factibilidad de un sistema antes de su implementación física, y sustenta el carácter no experimental del diseño metodológico adoptado.

2.22. Priorización del transporte público en el control semafórico

En corredores con alta presencia de transporte público, como la Av. Ejército, la gestión del semáforo puede orientarse a favorecer el desplazamiento de los vehículos que transportan a un mayor número de personas.

2.22.1. Fundamento de la priorización

Un bus o una combi ocupan más espacio que un automóvil, pero transportan a muchas más personas. Desde la perspectiva de la movilidad de personas, y no solo de vehículos, otorgar prioridad al transporte público maximiza el número de usuarios beneficiados por cada segundo de verde. La priorización puede ser pasiva, ajustando los tiempos a los patrones habituales del

transporte público, o activa, detectando la presencia de una unidad y extendiendo o adelantando su fase para reducir su demora.

2.22.2. Relación con la ponderación de actores

La ponderación de actores viales adoptada en esta investigación incorpora de manera implícita este principio: al asignar los mayores pesos al bus o combi grande (2,5) y al cúster (2,0), el sistema reconoce la mayor importancia operativa y social de estos vehículos y tiende a priorizar los accesos en los que se concentran. De este modo, la lógica de control no persigue únicamente la eficiencia vehicular, sino que se alinea con una visión de movilidad centrada en las personas, coherente con el elevado porcentaje de transporte público registrado en los aforos del tramo.

2.23. Seguridad vial e indicadores de siniestralidad

La seguridad vial es un objetivo central de la ingeniería de tránsito, y las intersecciones constituyen puntos donde se concentra un elevado número de conflictos y siniestros. El diseño y la operación del control semafórico influyen directamente en ese nivel de seguridad.

2.23.1. Los conflictos en las intersecciones

Una intersección concentra numerosos puntos de conflicto, entendidos como las ubicaciones donde las trayectorias de distintos movimientos, vehiculares o peatonales, pueden cruzarse. La probabilidad de un siniestro depende del número y la naturaleza de estos conflictos. El control semafórico reduce los conflictos al separar en el tiempo los movimientos incompatibles; sin embargo, una temporización inadecuada puede favorecer comportamientos riesgosos, como el paso en ámbar o la invasión del cruce peatonal, que incrementan la exposición al riesgo.

2.23.2. Indicadores de siniestralidad

La ingeniería de tránsito emplea indicadores como la frecuencia de siniestros, la tasa de siniestros por volumen de tránsito y la identificación de puntos críticos o de alta concentración de accidentes.

La reducción de conflictos peatón-vehículo que persigue el sistema propuesto, mediante la detección redundante de peatones y la priorización del peatón validado, contribuye de manera directa a la seguridad vial del cruce, al disminuir las situaciones en que un peatón y un vehículo compiten por el mismo espacio y en el mismo instante.

2.24. Impacto ambiental del tránsito y movilidad de bajas emisiones

El funcionamiento del tránsito tiene consecuencias ambientales que la gestión semafórica puede mitigar o agravar, lo que añade una dimensión adicional a la evaluación de un sistema de control.

2.24.1. Emisiones y consumo asociados a la congestión

Los vehículos detenidos con el motor en marcha, o que aceleran y frenan de forma repetida, consumen más combustible y emiten más gases contaminantes que aquellos que circulan con fluidez. La congestión, por tanto, no solo cuesta tiempo, sino que incrementa las emisiones de dióxido de carbono y de contaminantes locales que deterioran la calidad del aire urbano. Reducir las detenciones y las esperas innecesarias mediante un control eficiente disminuye de manera apreciable estas emisiones.

2.24.2. Hacia una movilidad de bajas emisiones

Las políticas de movilidad urbana sostenible buscan reducir la huella ambiental del transporte mediante la optimización de la infraestructura existente, la priorización del transporte público y la promoción de los modos no motorizados. Un sistema de control adaptativo que reduce las esperas y favorece al transporte público y al peatón se inscribe en esta línea, pues contribuye a una operación más limpia del tránsito sin requerir nuevas obras viales. Este beneficio ambiental complementa las mejoras operativas y refuerza la justificación del proyecto.

2.25. Accesibilidad universal en el diseño de cruces peatonales

Un cruce seguro debe serlo para todas las personas, incluidas aquellas con movilidad o percepción reducidas. El diseño universal es un principio normativo cada vez más presente en la gestión del tránsito y en la regulación del espacio público.

2.25.1. Principios de diseño universal

El diseño universal busca que el espacio público pueda ser utilizado por todas las personas sin necesidad de adaptaciones especiales. En los cruces peatonales, esto se traduce en tiempos de cruce suficientes para quienes caminan más lento, rampas y vados que eliminen los desniveles, superficies podotáctiles que orienten a las personas con discapacidad visual, y señales sonoras que indiquen el momento seguro para cruzar. La normativa de accesibilidad establece estas condiciones como requisitos del diseño vial.

2.25.2. Implicaciones para el sistema propuesto

La capacidad del sistema propuesto de detectar la presencia peatonal y de asignar una fase de cruce protegida abre la posibilidad de adaptar el tiempo de cruce a las necesidades reales de cada momento, otorgando intervalos más amplios cuando se detecta una mayor afluencia de peatones. Esta flexibilidad es coherente con los principios de accesibilidad universal, pues permite un trato más seguro e inclusivo del peatón que el de un semáforo de tiempo fijo, cuyo intervalo peatonal permanece invariable con independencia de quién y cuántos crucen.

2.26. Clasificación de intersecciones y estrategias de control

Las intersecciones son los puntos de la red vial donde convergen distintos flujos y, por ello, donde se concentran los conflictos y la necesidad de control. Comprender su clasificación permite situar el nodo estudiado y justificar la estrategia de control adoptada.

2.26.1. Tipos de intersección

Las intersecciones se clasifican según su configuración y su forma de control. Las intersecciones a nivel, donde las vías se cruzan en un mismo plano, son las más frecuentes en el entorno urbano y pueden operar sin control, con señales de prioridad como el pare o el ceda el paso, con rotondas o con semáforos. Las intersecciones a desnivel separan los flujos en distintos planos mediante pasos elevados o subterráneos, con lo que eliminan los conflictos a costa de una mayor inversión. La elección del tipo de control depende del volumen de tránsito, de la geometría y de las condiciones de seguridad. El nodo de la Av. Ejército es una intersección a nivel semaforizada, cuya alta y variable demanda justifica un control capaz de adaptarse a las condiciones del momento.

2.26.2. Criterios para la semaforización

La instalación de un semáforo no es automática: la ingeniería de tránsito aplica criterios o garantías que consideran el volumen vehicular, el volumen peatonal, la siniestralidad y las condiciones de la vía para decidir si una intersección debe semaforizarse. Una vez justificado el semáforo, la calidad de su operación depende de la temporización, que es precisamente donde un control inteligente aporta valor frente a un esquema de tiempo fijo, al ajustar los tiempos a la demanda real en lugar de mantenerlos constantes.

2.27. Evolución del control semafórico hacia los sistemas inteligentes

El control semafórico ha experimentado una evolución continua a lo largo de más de un siglo, impulsada por los avances en sensores, comunicaciones y capacidad de cómputo. Situar la propuesta dentro de esa evolución ayuda a comprender su aporte.

2.27.1. Del tiempo fijo al control actuado y coordinado

Los primeros semáforos operaban con tiempos fijos, calculados a partir de estudios de tránsito y ajustados de forma manual. La incorporación de detectores dio lugar al control actuado, capaz de

responder a la presencia de vehículos en cada acceso. Posteriormente, la coordinación de intersecciones y los sistemas centralizados permitieron gestionar corredores y redes completas, sincronizando los semáforos para mejorar la progresión del tránsito a lo largo de las arterias.

2.27.2. El control inteligente y adaptativo

La disponibilidad de sensores avanzados, de mayor capacidad de cómputo y de técnicas de inteligencia artificial ha dado paso a los sistemas de control inteligente, capaces de aprender de los datos y de adaptar su comportamiento en tiempo real. Estos sistemas integran la percepción del entorno, el procesamiento de la información y la toma de decisiones en un ciclo continuo. La propuesta de esta investigación se inscribe en esta etapa de la evolución, al combinar la visión por computadora, la detección térmica y el aprendizaje automático para dotar al semáforo de una capacidad de adaptación de la que carece el control de tiempo fijo.

2.28. Gestión de la demanda y planificación de la movilidad urbana

Más allá de la operación puntual de una intersección, el control del tránsito se enmarca en las políticas de planificación de la movilidad urbana, que buscan un uso eficiente y sostenible de la infraestructura de la ciudad.

2.28.1. Gestión de la demanda de transporte

La gestión de la demanda de transporte agrupa un conjunto de estrategias orientadas a influir en el comportamiento de los usuarios para reducir la congestión, como el escalonamiento de horarios, el fomento del transporte público y de los modos no motorizados, o la gestión del estacionamiento. A diferencia de las soluciones basadas en ampliar la oferta vial, la gestión de la demanda busca un uso más eficiente de la infraestructura existente, con menores costos e impactos.

2.28.2. El control inteligente como herramienta de gestión

Un sistema de control semafórico adaptativo constituye una herramienta de gestión en el ámbito de la operación: al distribuir el tiempo según la demanda real y priorizar los modos de mayor ocupación, contribuye a un uso más eficiente del cruce sin necesidad de nueva infraestructura. Esta perspectiva sitúa la propuesta dentro de un enfoque más amplio de planificación de la movilidad urbana, orientado a la sostenibilidad, la seguridad y la calidad de vida en la ciudad, y refuerza su pertinencia más allá del beneficio operativo inmediato.

2.29. Confiabilidad y tolerancia a fallos en los sistemas de control del tránsito

Un sistema que gestiona una intersección debe operar de forma confiable incluso ante fallos o información contradictoria, pues una decisión errónea puede comprometer la seguridad de los usuarios. La confiabilidad es, por tanto, un requisito de diseño tan importante como la eficiencia.

2.29.1. Redundancia y tolerancia a fallos

La confiabilidad de un sistema se refuerza mediante la redundancia, es decir, la incorporación de elementos que respaldan una función crítica de modo que el fallo de uno no comprometa el conjunto. En el control del tránsito, la redundancia resulta especialmente importante en la detección de peatones, cuya seguridad no puede depender de un único sensor. La tolerancia a fallos implica que, ante una anomalía, el sistema adopte un estado seguro conocido en lugar de comportarse de manera impredecible, evitando así que una falla se traduzca en un riesgo para los usuarios.

2.29.2. El modo seguro en el sistema propuesto

El sistema propuesto incorpora estos principios mediante la detección redundante de peatones por cámara y sensor térmico, y mediante una capa de coherencia que supervisa cada decisión del modelo. Cuando ambas fuentes de detección discrepan, o cuando se presenta una condición crítica, el sistema activa un modo seguro que prioriza la protección del usuario más vulnerable, aun a costa

de la eficiencia operativa. Este comportamiento refleja el principio de que, ante la incertidumbre, un sistema de control del tránsito debe errar siempre del lado de la seguridad.

2.30. Sistemas integrados de transporte y gestión del corredor

El tramo estudiado forma parte de un corredor por el que circula transporte público estructurado, lo que vincula la gestión del semáforo con la operación del sistema de transporte de la ciudad y amplía el alcance de la propuesta.

2.30.1. El transporte público estructurado

Los sistemas integrados de transporte organizan el servicio de buses en rutas troncales y alimentadoras, con el objetivo de ofrecer un servicio más eficiente y ordenado. Estos sistemas se benefician de una operación vial fluida, ya que la regularidad de los buses depende en parte de las condiciones del tránsito en los cruces que atraviesan. En corredores con transporte público estructurado, la coordinación entre la gestión del semáforo y la operación del servicio adquiere una relevancia particular para mantener la puntualidad y la frecuencia del servicio.

2.30.2. Aporte del sistema propuesto a la gestión del corredor

Al reconocer y priorizar los vehículos de transporte público mediante la ponderación de actores, el sistema propuesto contribuye a la regularidad del servicio en el tramo. Asimismo, su arquitectura de comunicación permite su eventual integración en una gestión de corredor más amplia, en la que varios cruces coordinen su operación para favorecer el desplazamiento del transporte público y del conjunto del tránsito. De este modo, la propuesta trasciende la mejora de una intersección aislada y se proyecta como un componente de la gestión integral de la movilidad en el corredor de la Av. Ejército.

2.31. Perspectivas: transporte conectado y ciudades inteligentes

El control inteligente del tránsito se inscribe en una tendencia más amplia hacia la integración de las tecnologías de la información en la gestión de la ciudad, conocida como ciudad inteligente. Situar la propuesta en ese horizonte permite comprender su proyección a futuro.

2.31.1. El paradigma de la ciudad inteligente

El concepto de ciudad inteligente designa el uso de sensores, datos y sistemas interconectados para gestionar de forma más eficiente los servicios urbanos, entre ellos la movilidad. En este paradigma, la infraestructura vial deja de ser pasiva y se convierte en una red capaz de percibir, comunicar y responder a las condiciones del entorno. El control semafórico inteligente es uno de los componentes más visibles de esta transformación, al aportar capacidad de adaptación a un elemento tradicionalmente estático de la ciudad.

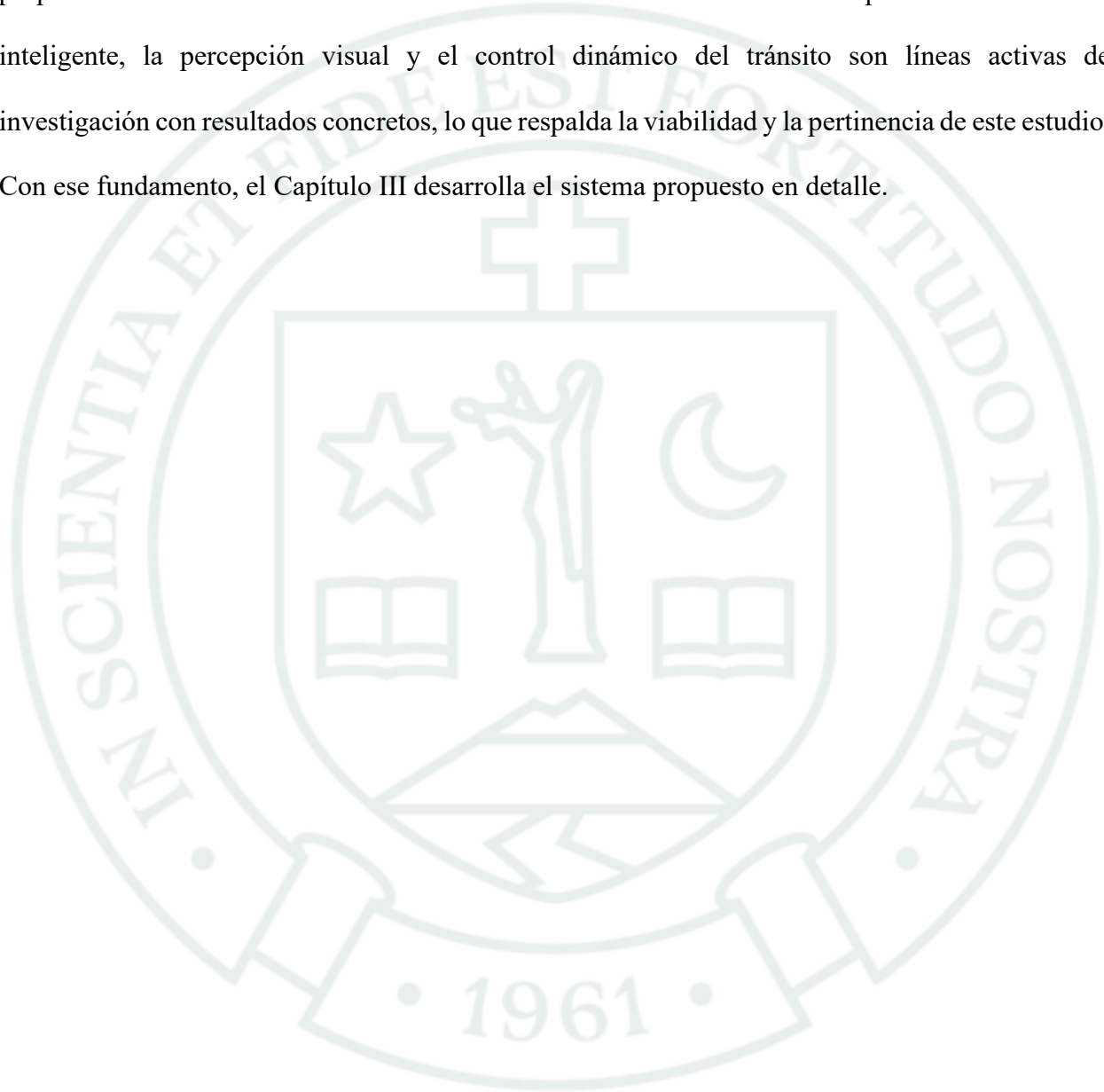
2.31.2. Transporte conectado y proyección del sistema

Las tendencias de transporte conectado apuntan a la comunicación entre vehículos, infraestructura y usuarios, lo que en el futuro permitirá decisiones de control aún más informadas. Aunque la propuesta de esta investigación se limita a una intersección y a la información captada localmente, su arquitectura modular y su capacidad de comunicación la hacen compatible con esa evolución. El sistema puede entenderse, así, como un primer paso hacia una infraestructura vial más inteligente y conectada en la ciudad de Arequipa, capaz de crecer e integrarse a medida que se disponga de nuevas tecnologías y de una gestión coordinada del tránsito a escala de corredor y de ciudad.

2.32. Consideraciones finales del capítulo

El marco teórico desarrollado en este capítulo establece las bases conceptuales y tecnológicas que sostienen la propuesta de investigación. Desde los fundamentos de la ingeniería de tránsito y la

movilidad urbana hasta los principios del control semafórico, los algoritmos de aprendizaje automático, los sistemas inteligentes de transporte, la visión por computadora y la detección peatonal, cada sección aporta el respaldo teórico necesario para entender por qué el sistema propuesto está diseñado como está. La revisión de antecedentes confirma que la semaforización inteligente, la percepción visual y el control dinámico del tránsito son líneas activas de investigación con resultados concretos, lo que respalda la viabilidad y la pertinencia de este estudio. Con ese fundamento, el Capítulo III desarrolla el sistema propuesto en detalle.



CAPÍTULO III. DESARROLLO Y SIMULACIÓN DEL SISTEMA PROPUESTO

3.1. Análisis de factibilidad del sistema

Antes de avanzar hacia el diseño detallado del sistema, es necesario analizar si la propuesta es viable desde las dimensiones técnica, operativa y económica. La factibilidad no es una formalidad académica sino es la base que permite distinguir si una propuesta es realizable con los recursos disponibles o si sus supuestos la sitúan fuera del alcance práctico del contexto donde se plantea aplicar.

3.1.1. Factibilidad técnica

Desde el punto de vista técnico, todos los componentes del sistema propuesto están basados en tecnologías maduras, documentadas y disponibles en el mercado. Las cámaras IP de exterior con capacidad Full HD y visión infrarroja integrada son equipos estándar en el sector de videovigilancia urbana. Los sensores térmicos de presencia han demostrado su fiabilidad en aplicaciones de seguridad y detección de personas en exteriores. La red cableada con comunicación por el protocolo Ethernet y el protocolo PoE son tecnologías robustas con más de dos décadas de madurez industrial. El núcleo de procesamiento del sistema, los modelos Random Forest, es una técnica de aprendizaje automático ampliamente validada, disponible en bibliotecas de código abierto como scikit-learn para Python, y con requerimientos computacionales que pueden satisfacerse con hardware de bajo costo, como una computadora de placa reducida tipo Raspberry Pi o un mini-PC industrial. La inferencia de un modelo Random Forest de 100 estimadores sobre 27 variables tarda por lo general menos de 10 milisegundos por ciclo, lapso suficiente para una operación en tiempo real dentro de un ciclo semafórico de decenas de segundos.

La interfaz de supervisión desarrollada en entorno web local, que permite visualizar las entradas, la fase priorizada, los tiempos asignados, la validación peatonal y la comparación con el tiempo fijo, confirma que la arquitectura es implementable sin dependencias externas costosas. En su conjunto, la factibilidad técnica del sistema es alta: cada componente puede adquirirse, configurarse e integrarse con tecnología disponible hoy en el mercado arequipeño y peruano.

3.1.2. Factibilidad operativa

La factibilidad operativa analiza si el sistema puede funcionar de manera sostenida en el entorno real donde se propone instalarlo. El tramo de la Av. Ejército condicionado por el nodo de la Av. Cayma y la Av. Trinidad Morán presenta condiciones de operación que el sistema fue expresamente diseñado para manejar: flujos heterogéneos, variabilidad horaria marcada, presencia peatonal frecuente y condiciones de iluminación variable entre el día y la noche.

El sistema incorpora mecanismos específicos para operar en condiciones adversas. La redundancia sensorial entre la cámara IP y el sensor térmico permite mantener una detección confiable incluso cuando la iluminación es deficiente o cuando hay oclusiones parciales en la escena. El modo seguro ante discrepancia entre fuentes garantiza que el sistema no tome decisiones vehiculares agresivas cuando la condición peatonal es incierta. Y el modo de contingencia, que conmuta de forma automática al esquema de tiempo fijo preconfigurado ante fallas en el módulo de visión, asegura que cualquier interrupción parcial no comprometa la continuidad del servicio semafórico básico (la matriz de contingencias operativas se presenta en el Anexo 5).

El plan de mantenimiento preventivo descrito en el Anexo 4 contempla inspecciones periódicas de todos los módulos, lo que refuerza la viabilidad operativa a largo plazo. La frecuencia de mantenimiento propuesta, mensual para los componentes de percepción y quincenal para el módulo

de control, es compatible con los recursos técnicos disponibles en el sector público arequipeño. La factibilidad operativa del sistema se considera alta dentro del contexto simulado y moderada-alta para una implementación física piloto.

3.1.3. Factibilidad económica

La factibilidad económica no puede estimarse con precisión sin un análisis de costos formales vinculado a una propuesta de implementación física, que queda fuera del alcance de esta investigación. Sin embargo, es posible establecer una orientación general a partir de los componentes identificados en la arquitectura del sistema.

Los elementos de mayor costo en una implementación real serían tres. Primero, la cámara IP de percepción visual, referenciada en este trabajo como la AXIS P3265-LVE Dome Camera, cuyo precio de mercado oscila entre los 800 y 1 200 dólares en la versión básica. Segundo, el sensor térmico de validación peatonal, cuyo costo depende de la resolución y la sensibilidad requeridas, pero que puede encontrarse en rangos de 200 a 600 dólares para modelos de uso externo. Tercero, el módulo local de control, que puede basarse en un mini-PC industrial de bajo consumo disponible en el mercado desde los 150 dólares. La infraestructura de cableado Ethernet, los accesorios de instalación y los trabajos civiles de montaje representarían un costo adicional dependiente de las condiciones del emplazamiento. Las especificaciones técnicas de referencia de todos los componentes del sistema se detallan en el Anexo 2.

Comparado con los costos de ampliar la infraestructura vial o de instalar sistemas de control centralizado de mayor envergadura, la propuesta presentada en esta investigación representa una alternativa de bajo costo relativo y alta escalabilidad. La posibilidad de implementarla primero en modo piloto en una sola intersección y evaluarla antes de replicarla reduce significativamente el

riesgo económico asociado. La factibilidad económica del sistema se considera viable para una implementación piloto en el marco de proyectos de mejora de infraestructura vial de la Municipalidad Provincial de Arequipa.

3.2. Arquitectura del sistema propuesto

Para el sistema propuesto se diseñó una arquitectura que integra cinco módulos funcionales que operan de manera articulada y secuencial. Juntos conforman una cadena funcional clara: primero se capta el estado del entorno, luego se valida la información crítica, después se decide qué fase activar y por cuánto tiempo, y finalmente se visualiza la respuesta del sistema. La arquitectura no es simplemente una suma de piezas tecnológicas: es la forma en que esas piezas se relacionan, se comunican y se condicionan mutuamente para producir decisiones coherentes con las condiciones reales del tránsito. Se detalla cada módulo a continuación:

3.2.1. Módulo de percepción visual

Es la fuente principal de recolección de información del sistema. Mediante cámaras IP de exterior capta la escena vial en tiempo real y procesa la imagen para identificar los actores presentes: buses o combis grandes, cústeres, automóviles particulares, motocicletas, bicicletas, scooters y peatones. A partir de esa detección y clasificación, el sistema construye variables operativas como carga vehicular por acceso, composición del flujo y presencia de usuarios en zonas críticas.

Para el módulo de percepción visual se adopta como equipo de referencia la cámara IP AXIS P3265-LVE Dome Camera: tipo domo antivandálico para exteriores, resolución Full HD, lente varifocal, alimentación PoE, compresión H.264/H.265, iluminación infrarroja integrada y amplio rango dinámico. Con protección IP66 y resistencia mecánica IK10, es una opción adecuada para el entorno de Arequipa, donde el equipo debe operar con estabilidad frente a polvo, lluvia, radiación

solar y variaciones de temperatura. La calidad de este módulo es determinante porque todo lo que viene después depende de él.

3.2.2. Módulo de validación redundante de presencia peatonal

Complementa la percepción visual con un sensor térmico orientado a confirmar la presencia peatonal en zonas de espera o cruce. No reemplaza a la cámara: opera como segunda capa de verificación, sobre todo cuando las condiciones de iluminación, oclusión o ambigüedad visual pueden comprometer la fiabilidad de la detección principal. Su principio de funcionamiento, la detección de la radiación infrarroja emitida por los cuerpos humanos, lo hace independiente de las condiciones de luz, lo que lo convierte en un complemento ideal para la cámara visible.

La lógica de uso del módulo térmico es directa: si la cámara detecta un peatón y el sensor térmico lo confirma, el sistema tiene una base sólida para priorizar o mantener la protección peatonal. Si ambas fuentes coinciden en ausencia peatonal, el sistema puede operar con la lógica vehicular sin restricciones. Si discrepan, activa un modo conservador que restringe la liberación de fase vehicular conflictiva hasta que la incertidumbre se resuelva dentro del ciclo de simulación.

3.2.3. Módulo de decisión y control semafórico

Es el núcleo operativo del sistema. Recibe las variables de los módulos anteriores y aplica la lógica adaptativa para determinar la fase prioritaria y los tiempos de operación. El mecanismo de decisión no opera con reglas escritas a mano ni con ciclos fijos: se apoya en dos modelos Random Forest entrenados sobre 6 000 casos sintéticos generados a partir de perfiles horarios representativos del sector. Un clasificador determina qué fase activar; un regresor estima cuántos segundos de verde asignarle. Sobre la propuesta de los modelos opera además una capa de coherencia que aplica restricciones operativas y de seguridad antes de ejecutar la decisión.

3.2.4. Módulo de comunicación e integración

Gestiona el intercambio de información entre componentes. El enlace principal entre la cámara y el módulo de control se sustenta en red cableada por protocolo Ethernet 1000BASE-T sobre cable de cobre UTP Cat6 o Cat6A, con tecnologías inalámbricas como respaldo. La alimentación eléctrica de la cámara puede transmitirse por el mismo cable mediante Power over Ethernet (PoE), simplificando la instalación y reduciendo la necesidad de fuentes independientes. La integración lógica entre el módulo de control y la interfaz de supervisión se organiza mediante una API REST con intercambio en formato JSON.

3.2.5. Módulo de supervisión e interfaz

Permite visualizar en tiempo real el comportamiento del sistema: entradas detectadas, fase priorizada, tiempos asignados, confianza del modelo, estado de la validación peatonal y comparación con el esquema de tiempo fijo. Cumple tanto una función de monitoreo como de validación, al evidenciar que las decisiones del sistema son coherentes con las condiciones observadas. La secuencia que describe esa arquitectura es siempre la misma: la cámara y el sensor térmico captan el entorno, el módulo de decisión procesa esa información y determina la respuesta semafórica, y la interfaz representa ese proceso de manera visible y verificable.

Figura 1

Arquitectura funcional del sistema inteligente de control de tráfico propuesto



Nota. Elaboración propia.

3.3. Diseño funcional del sistema

El sistema inteligente de control de tráfico propuesto para el tramo de la Av. Ejército condicionado por el nodo de la Av. Cayma y la Av. Trinidad Morán no es una mejora del semáforo convencional: es una arquitectura distinta. Mientras un semáforo de tiempo fijo ejecuta ciclos predeterminados sin importar lo que ocurre en la vía, este sistema capta el estado del entorno, valida la información crítica y genera decisiones de control ajustadas a la demanda real de cada momento. La secuencia de cinco módulos descrita en la sección anterior opera de manera continua: la cámara y el sensor térmico captan el entorno, el módulo de decisión procesa esa información y determina la respuesta semafórica, y la interfaz representa ese proceso de manera visible y verificable.

3.4. Definición de actores viales y ponderaciones

Un sistema de control de tráfico que solo cuenta vehículos sin distinguir su tipo toma decisiones sobre información incompleta. Un bus acumulado en un acceso no genera el mismo impacto operativo que una bicicleta: ocupa más espacio, tarda más en arrancar, genera mayor cola si se queda esperando varios ciclos. Por eso, antes de que el sistema pueda tomar decisiones coherentes, necesita saber no solo cuántos usuarios hay, sino qué tipos son y cuánto pesa cada uno dentro del flujo. Los actores viales considerados son: buses o combis grandes, cústeres, automóviles particulares, motocicletas, bicicletas, scooters y peatones. La selección responde a la composición real del tránsito en el tramo estudiado.

Las ponderaciones operativas adoptadas como base para la simulación son las siguientes: bus o combi grande: 2,5; cúster: 2,0; automóvil particular: 1,0; motocicleta: 0,5; bicicleta: 0,4; scooter: 0,3. El peatón tiene un tratamiento diferente al resto: su presencia no se suma a la demanda ponderada vehicular, sino que actúa como condición crítica que puede modificar o restringir la prioridad de las fases vehiculares. Cuando la acumulación peatonal supera el umbral definido y ambos módulos de detección lo confirman, el sistema activa la fase de cruce peatonal aunque la demanda vehicular sea alta en ese momento.

La asignación de estos valores se apoya en el concepto de equivalencia vehicular empleado en la ingeniería de tránsito, según el cual cada tipo de vehículo aporta al flujo una carga distinta de la de un automóvil de referencia, en función del espacio que ocupa en la vía, de su tiempo de arranque y despeje, y de su contribución a la formación de cola. Tomando el automóvil particular como unidad (1,0), el bus o combi grande recibe el mayor peso (2,5) por su longitud, su lenta aceleración y su prolongada permanencia en el acceso; el cúster adopta un valor intermedio (2,0) al tratarse de transporte público de menor tamaño; y la motocicleta (0,5), la bicicleta (0,4) y el scooter (0,3) reciben pesos decrecientes, ya que su menor ocupación y su mayor agilidad implican una

perturbación reducida sobre el flujo. De este modo, la demanda ponderada de cada acceso refleja la presión operativa real del tránsito y no un simple conteo de unidades.

3.5. Datos de entrenamiento y validación del modelo

3.5.1. Fuentes para la recopilación de datos vehiculares en la Av. Ejército

La construcción de la base de datos de entrenamiento partió de información real levantada en el tramo de la Av. Ejército. Las fuentes de datos empleadas fueron las siguientes:

Fuente	Descripción
Aforos vehiculares en campo	Se realizaron conteos manuales y con asistencia de video en el nodo vial de la Av. Cayma y la Av. Trinidad Morán durante las franjas horarias de mayor y menor demanda. Los aforos cubrieron los tres accesos del nodo (A1: Av. Ejército Este–Oeste; A2: Av. Ejército Oeste–Este; A3: Av. Trinidad Morán / Av. Cayma) y clasificaron los vehículos por tipo en intervalos de 15 minutos.
Observación directa y registros de video	Se capturaron secuencias de video del tramo durante los tres periodos de mayor demanda identificados (7:00–8:00 a.m., 12:00–1:00 p.m. y 7:00–8:00 p.m.) y durante franjas intermedias y de baja demanda. Estos registros permitieron verificar la composición vehicular, identificar comportamientos peatonales típicos y calibrar los parámetros de los escenarios de simulación.
Datos históricos de tránsito y estudios del sector	Se consultaron estudios de tránsito disponibles para la Av. Ejército y el distrito de Cayma, así como informes de movilidad urbana de la Municipalidad Provincial de Arequipa, para contrastar los aforos propios y validar la representatividad de los perfiles horarios construidos.
Caracterización operativa del nodo vial	A partir de la observación en campo se estableció la geometría operativa del cruce, los movimientos típicos de cada acceso, los comportamientos peatonales frecuentes y los patrones de conflicto entre flujos vehiculares y cruces peatonales.

Nota. Elaboración propia basada en aforos de campo realizados en el tramo de la Av. Ejército, nodo Av. Cayma – Av. Trinidad Morán, Arequipa.

3.5.2. Generación de la base de datos de entrenamiento

A partir de los datos recopilados en campo se construyó una base de datos sintética de 6 000 casos que representó la variabilidad horaria observada en el tramo. Generar datos sintéticos es una práctica estándar en el desarrollo de sistemas de control basados en aprendizaje automático cuando no se dispone de un conjunto de datos reales suficientemente amplio para cubrir todos los escenarios operativos relevantes. Los parámetros de generación, rangos de cantidad por tipo de actor vial, distribuciones de presencia peatonal y perfiles de espera acumulada, se derivaron

directamente de las mediciones de campo y de la caracterización operativa del nodo vial (una muestra representativa de esta base de datos se detalla en el Anexo 3).

Cada registro de la base de datos corresponde a un ciclo de decisión del sistema y contiene el valor de las 27 variables de entrada junto con la etiqueta de la fase correcta (para el clasificador) y el tiempo de verde óptimo (para el regresor). Las etiquetas fueron asignadas de manera determinista siguiendo la lógica de prioridad definida en la arquitectura del sistema: mayor demanda ponderada, presencia peatonal validada, y espera acumulada como criterio de desempate.

3.5.3. Variables de entrada del modelo

El modelo recibe 27 variables de entrada en cada ciclo de decisión. Estas variables se organizan en cuatro grupos funcionales: composición del flujo vehicular por acceso, condición peatonal, historial de esperas acumuladas y estado de los sensores.

Tabla 3

Variables de entrada del modelo (27 variables)

Grupo	Variable	Descripción	Tipo
Flujo vehicular A1	bus_A1	Cantidad de buses o combis grandes en acceso Este–Oeste	Numérica entera
	custer_A1	Cantidad de cústeres en acceso A1	Numérica entera
	auto_A1	Cantidad de automóviles en acceso A1	Numérica entera
	moto_A1	Cantidad de motocicletas en acceso A1	Numérica entera
	bici_A1	Cantidad de bicicletas en acceso A1	Numérica entera
	scooter_A1	Cantidad de scooters en acceso A1	Numérica entera
	demanda_A1	Demanda ponderada calculada para A1	Numérica continua
Flujo vehicular A2	bus_A2	Cantidad de buses o combis grandes en acceso Oeste–Este	Numérica entera
	custer_A2	Cantidad de cústeres en acceso A2	Numérica entera
	auto_A2	Cantidad de automóviles en acceso A2	Numérica entera
	moto_A2	Cantidad de motocicletas en acceso A2	Numérica entera

	bici_A2	Cantidad de bicicletas en acceso A2	Númerica entera
	scooter_A2	Cantidad de scooters en acceso A2	Númerica entera
	demanda_A2	Demanda ponderada calculada para A2	Númerica continua
Flujo vehicular A3	bus_A3	Cantidad de buses o combis en acceso Trinidad/Cayma	Númerica entera
	custer_A3	Cantidad de cústeres en acceso A3	Númerica entera
	auto_A3	Cantidad de automóviles en acceso A3	Númerica entera
	moto_A3	Cantidad de motocicletas en acceso A3	Númerica entera
	bici_A3	Cantidad de bicicletas en acceso A3	Númerica entera
	scooter_A3	Cantidad de scooters en acceso A3	Númerica entera
	demanda_A3	Demanda ponderada calculada para A3	Númerica continua
Condición peatonal	peat_visual	Peatones detectados por visión por computadora	Númerica entera
	peat_termico	Peatones detectados por sensor térmico	Númerica entera
	acum_peatonal	Acumulación peatonal en zona de cruce (ciclos previos)	Númerica entera
Historial y estado	espera_A1	Ciclos de espera acumulada del acceso A1 sin prioridad	Númerica entera
	espera_A2	Ciclos de espera acumulada del acceso A2 sin prioridad	Númerica entera
	espera_A3	Ciclos de espera acumulada del acceso A3 sin prioridad	Númerica entera

Nota. Las variables de demanda ponderada se calculan mediante la fórmula: $\text{demanda} = \text{bus} \times 2,5 + \text{cúster} \times 2,0 + \text{auto} \times 1,0 + \text{moto} \times 0,5 + \text{bicicleta} \times 0,4 + \text{scooter} \times 0,3$. Elaboración propia.

3.5.4. Variables de salida del modelo

El modelo produce dos tipos de salida en cada ciclo de decisión: una salida de clasificación que determina la fase semafórica a activar, y una salida de regresión que estima el tiempo de verde a asignar.

Tabla 4

Variables de salida del modelo

Variable de salida	Tipo de modelo	Valores posibles	Descripción
Fase priorizada	Clasificador (RF)	F1, F2, F3, F4	Fase semafórica a activar en el ciclo actual. F1: flujo principal E-O. F2: acceso secundario. F3: cruce peatonal. F4: variante de equidad.
Tiempo de verde asignado (s)	Regresor (RF)	15 – 45 segundos	Duración del verde a asignar a la fase priorizada. Rango dividido en tres niveles: bajo (15–20 s), medio (21–32 s) y alto (33–45 s).

Tiempo de rojo estimado (s)	Derivado	Variable por fase	Tiempo de espera estimado para los accesos no priorizados, calculado a partir del ciclo total resultante.
Estado de modo seguro	Capa de coherencia	Activo / Inactivo	Indica si la capa de coherencia ha intervenido y restringido la liberación vehicular por discrepancia entre sensores o condición peatonal crítica.
Confianza del modelo (%)	Clasificador (RF)	0 – 100 %	Proporción de árboles del bosque que coinciden en la fase seleccionada. Valor orientativo para supervisión.
Tiempo de espera estimado por acceso (s)	Derivado	Variable	Estimación del tiempo promedio de espera por acceso bajo el ciclo generado por el sistema adaptativo.

Nota. El tiempo de amarillo permanece fijo en 3 segundos en todos los ciclos. La adaptabilidad del sistema recae sobre el tiempo de verde de servicio. RF = Random Forest. Elaboración propia.

3.5.5. Partición de los datos y métrica de evaluación

El conjunto de 6 000 casos sintéticos fue dividido en conjunto de entrenamiento (80 %, equivalente a 4 800 registros) y conjunto de prueba (20 %, equivalente a 1 200 registros). La partición se realizó de manera aleatoria estratificada para garantizar que la distribución de fases fuera representativa en ambos subconjuntos. El clasificador fue evaluado mediante exactitud global (accuracy) y F1-score macro, que pondera igualmente el desempeño en cada clase independientemente de su frecuencia de aparición. El regresor fue evaluado mediante el error absoluto medio (MAE) y el coeficiente de determinación (R^2).

3.6. Lógica adaptativa de control semafórico

La lógica adaptativa transforma la información captada del entorno en decisiones concretas sobre qué fase activar, durante cuánto tiempo y bajo qué condiciones de seguridad. A diferencia de un ciclo fijo, no parte de duraciones predeterminadas: evalúa lo que hay en el tramo, pondera su incidencia operativa y determina la respuesta más adecuada para ese momento específico. El mecanismo de decisión central está compuesto por dos modelos Random Forest. El primero es un clasificador que, a partir de 27 variables de entrada, determina cuál de las cuatro fases del sistema

debe activarse. El segundo es un regresor que, con las mismas variables, estima cuántos segundos de verde debe recibir esa fase.

La fórmula de demanda ponderada que alimenta al modelo es la siguiente:

Demanda ponderada de fase = Σ (cantidad de actor vial $\times$ ponderación asignada).

Esta fórmula hace que la composición del flujo, y no solo su volumen, determine la presión operativa de cada acceso. Un acceso con tres buses tiene una demanda ponderada muy superior a uno con diez bicicletas, aunque en número de unidades el segundo sea mayor.

3.6.1. Reglas de asignación de prioridad

La asignación de prioridad del sistema se rige por cuatro reglas jerarquizadas.

La primera establece que si no existe presencia peatonal conflictiva validada, el sistema activa la fase vehicular con mayor demanda ponderada entre las disponibles.

La segunda establece que si existe presencia peatonal validada por ambos módulos en una zona de conflicto, el sistema restringe la liberación de la fase vehicular incompatible y mantiene o activa la protección peatonal, independientemente de la demanda vehicular.

La tercera actúa cuando dos fases vehiculares presentan demandas muy similares, con una diferencia menor al 15 %. En ese caso el sistema favorece la que acumula mayor tiempo relativo de espera.

La cuarta se activa ante discrepancia entre detección visual y térmica: el sistema adopta una condición conservadora y restringe la liberación de la fase vehicular conflictiva hasta que la incertidumbre se resuelva.

3.6.2. Asignación adaptativa de tiempos semafóricos

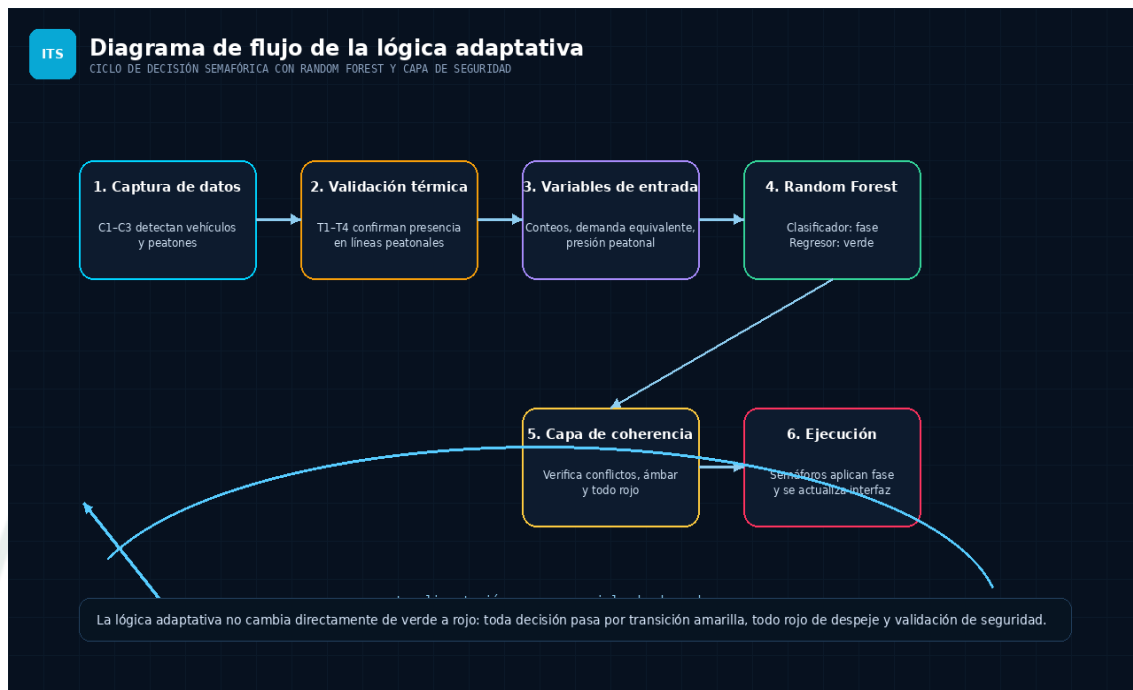
Una vez determinada la fase prioritaria, el regresor Random Forest estima el tiempo de verde a asignar dentro de un rango de 15 a 45 segundos. Ese rango se divide en tres niveles según la demanda ponderada de la fase seleccionada: demanda baja, tiempo verde corto (15–20 s); demanda media, tiempo verde intermedio (21–32 s); demanda alta, tiempo verde extendido (33–45 s). El tiempo de amarillo se mantiene fijo en 3 segundos en todos los casos.

3.6.3. Modo seguro ante discrepancia de detección

Cuando la cámara y el sensor térmico no coinciden en la detección peatonal, el sistema activa un modo seguro antes de tomar ninguna decisión sobre la fase vehicular en conflicto. Si ambas fuentes coinciden en ausencia peatonal, el sistema opera con normalidad. Si ambas confirman presencia peatonal, activa o mantiene la protección peatonal. Si discrepan, adopta una condición conservadora: restringe la liberación vehicular y extiende la protección peatonal hasta que la condición de incertidumbre se resuelva dentro del ciclo de simulación. Si el módulo de percepción visual falla de manera total, la lógica adaptativa se suspende y el sistema conmuta automáticamente a un esquema de tiempo fijo preconfigurado.

Figura 2

Diagrama de flujo de la lógica adaptativa de control semafórico



Nota. Elaboración propia.

3.6.4. Marco normativo aplicado al control semafórico

El diseño de la lógica de control se enmarca en la normativa técnica vigente. Del Manual de Dispositivos de Control del Tránsito Automotor para Calles y Carreteras del Ministerio de Transportes y Comunicaciones (MTC, 2016) se adoptan el tiempo mínimo de luz ámbar de tres segundos, la estructura de fases semafóricas y los criterios de seguridad aplicables al cruce peatonal, que en este sistema se traducen en la prioridad otorgada al peatón validado por ambos módulos de detección. Los rangos de tiempo de verde establecidos para la asignación adaptativa, de 15 a 45 segundos, respetan los intervalos mínimos y máximos usuales en la práctica semafórica, evitando fases demasiado cortas que comprometan la seguridad del cruce o demasiado largas que generen esperas innecesarias. Para la evaluación e interpretación de los resultados se emplea como

referencia el Highway Capacity Manual (Transportation Research Board, 2010), cuyos niveles de servicio, definidos a partir de la demora media por vehículo, proporcionan el marco para valorar el desempeño del sistema adaptativo frente al esquema de tiempo fijo.

3.7. Desempeño del modelo de aprendizaje automático

Antes de presentar los resultados por escenario, se describe el desempeño de los dos modelos Random Forest que constituyen el núcleo del sistema. Ambos modelos fueron entrenados con 4 800 casos sintéticos (80 % del dataset) y evaluados sobre los 1 200 casos restantes (20 %), siguiendo la partición estratificada descrita anteriormente. El entorno de entrenamiento fue Python con la biblioteca scikit-learn, y los modelos exportados fueron verificados en la interfaz de simulación para confirmar la consistencia de sus predicciones.

Tabla 5

Indicadores de desempeño de los modelos Random Forest entrenados

Parámetro del modelo	Valor obtenido	Interpretación
Tipo de modelo (clasificador)	Random Forest Classifier	Selección de fase semafórica
Tipo de modelo (regresor)	Random Forest Regressor	Asignación de tiempo verde
Número de árboles	100 estimadores	Balance precisión–varianza
Variables de entrada	27 variables	Composición de 3 accesos + peatones + acumulación
Registros de entrenamiento	4 800 casos (80 %)	Partición estratificada del dataset de 6 000 registros
Registros de prueba	1 200 casos (20 %)	Conjunto de evaluación no visto durante el entrenamiento
Accuracy (clasificador)	92,42 %	Precisión en selección de fase correcta
F1-score macro (clasificador)	0,918	Equilibrio entre fases con distinta frecuencia
MAE (regresor)	1,40 s	Error medio en tiempo verde asignado
R ² (regresor)	0,9617	Alta capacidad explicativa del modelo
Verificación Python ↔ JavaScript	9/10 coincidencias ($\Delta < 2$ s)	Consistencia entre entrenamiento y simulación

Nota. MAE = Error Absoluto Medio. R^2 = Coeficiente de determinación. Accuracy y F1-score obtenidos sobre conjunto de prueba (20 % del dataset). La verificación Python↔JavaScript confirma la consistencia del modelo exportado a la interfaz de simulación. Elaboración propia.

Los indicadores confirman que el clasificador de fase identifica correctamente la fase semafórica óptima en el 92,42 % de los casos, con un F1-score macro de 0,918 que garantiza equilibrio entre fases de distinta frecuencia de activación. El regresor de tiempo verde presenta un error medio de 1,40 segundos y un R^2 de 0,9617, lo que evidencia una alta capacidad explicativa del modelo. Estos indicadores respaldan técnicamente la viabilidad del sistema propuesto como alternativa al control semafórico de tiempo fijo.

3.8. Resultados por escenario de simulación

Los siete escenarios evaluados no se definieron de forma arbitraria. Se derivaron de las condiciones operativas reales del tramo y de las situaciones críticas que el sistema debe resolver. Tres de ellos reproducen los niveles de intensidad de tránsito registrados en los aforos de campo, esto es, flujo bajo, medio y alto en hora punta; uno refleja la fuerte presencia de transporte público característica del corredor; dos abordan de manera directa la seguridad peatonal, mediante la presencia peatonal relevante y el conflicto peatón-vehículo; y el último evalúa la robustez del sistema ante una discrepancia entre la detección visual y el sensor térmico. En conjunto, estos escenarios cubren el rango de operación observado en la vía y añaden las condiciones de seguridad e incertidumbre que exigen una respuesta diferenciada, lo que permite valorar el comportamiento del sistema de forma representativa.

A continuación se presentan los resultados de cada escenario de simulación. Para cada uno se describe la composición del flujo detectado, la decisión adoptada por el sistema inteligente frente al esquema de tiempo fijo de referencia, y los tiempos de espera estimados por acceso. El tiempo

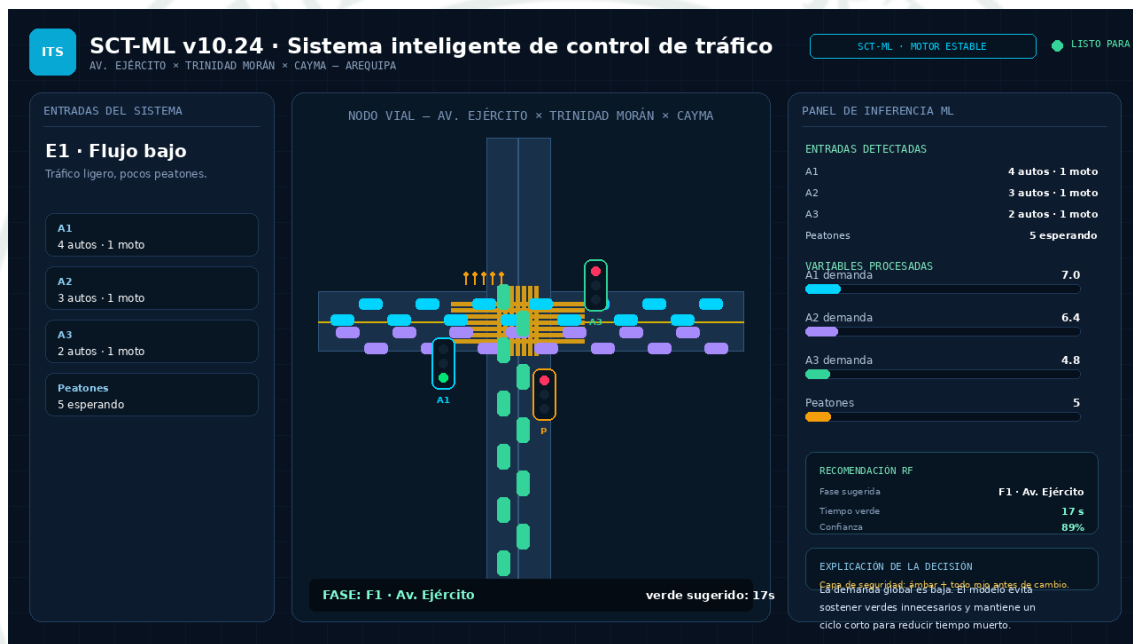
fijo de referencia emplea los parámetros siguientes: F1 = 38 s de verde, F2 = 25 s de verde, F3 = 22 s de verde, amarillo = 3 s en todas las fases, ciclo total = 94 s.

3.8.1. Escenario 1: Flujo bajo

Condición nocturna (~21:30 h). Tránsito mínimo, baja saturación operativa en los tres accesos.

Figura 3

Interfaz del sistema — Flujo bajo: Entradas detectadas y fase priorizada



Nota. Elaboración propia.

a) Composición del flujo detectado

Tabla 6

Entradas del sistema — Flujo bajo

Acceso	Bus/Combi	Cúster	Automóvil	Moto	Bicicleta	Scoter	Demanda pond.
A1 — Ejército E→O	0	0	3	1	0	0	3,5
A2 — Ejército O→E	0	0	2	0	0	0	2,0
A3 — Trinidad/Cayma	0	0	1	1	0	0	1,5
Peatones detectados	3 unidades	—	—	—	—	—	—

Nota. Demanda ponderada = $\text{bus} \times 2,5 + \text{cúster} \times 2,0 + \text{auto} \times 1,0 + \text{moto} \times 0,5 + \text{bicicleta} \times 0,4 + \text{scooter} \times 0,3$. Elaboración propia.

b) Decisión del sistema — Random Forest vs tiempo fijo

Tabla 7

Decisión ML vs tiempo fijo — Flujo bajo

Parámetro	Sistema adaptativo (RF-100)	Sistema tiempo fijo (referencia)
Fase priorizada	F1 — Flujo principal (E↔O)	F1 — Flujo principal (E↔O)
Tiempo verde asignado	17 s	38 s
Tiempo amarillo	3 s	3 s
Confianza RF	86 %	— (sin modelo)
Capa de coherencia	RF coincide — sin ajuste	— (no aplica)

Nota. RF = Random Forest. La capa de coherencia aplica reglas determinísticas sobre la propuesta del modelo.

Elaboración propia.

c) Tiempos de espera estimados por acceso

Tabla 8

Tiempos de espera estimados — Flujo bajo

Acceso	Espera adaptativo (s)	Espera tiempo fijo (s)	Diferencia (s)	Variación (%)
A1 — Ejército E→O	11	18	-7	-39 %
A2 — Ejército O→E	11	18	-7	-39 %
A3 — Trinidad/Cayma	19	26	-7	-27 %
Peatones	22	38	-16	-42 %

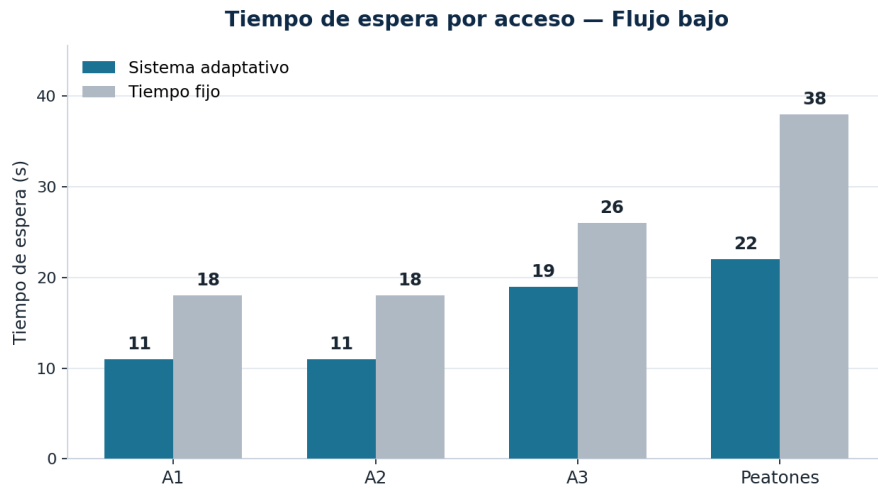
Nota. Tiempo fijo de referencia: F1=38 s, F2=25 s, F3=22 s, ciclo=94 s. Elaboración propia.

d) Análisis del escenario

En condición de baja demanda el sistema adaptativo reduce considerablemente la longitud del ciclo (17 s verde vs 38 s fijos), eliminando tiempos muertos innecesarios en todos los accesos. La reducción global promedio del tiempo de espera es del 37 %.

Figura 4

Interfaz del sistema — Flujo bajo: Comparación adaptativo vs tiempo fijo



Nota. Elaboración propia.

3.8.2. Escenario 2: Flujo medio

Condición de mediodía (~11:30–13:00 h). Demanda intermedia con mezcla regular de automóviles, transporte público menor y vehículos ligeros.

Figura 5

Interfaz del sistema — Flujo medio: Entradas detectadas y fase priorizada



Nota. Elaboración propia.

a) Composición del flujo detectado

Tabla 9

Entradas del sistema — Flujo medio

Acceso	Bus/Combi	Cúster	Automóvil	Moto	Bicicleta	Scooter	Demanda pond.
A1 — Ejército E→O	1	1	5	2	1	0	10,9
A2 — Ejército O→E	0	1	4	2	1	0	7,4
A3 — Trinidad/Cayma	0	0	3	1	1	1	4,2
Peatones detectados	10 unidades	—	—	—	—	—	—

Nota. Demanda ponderada = $\text{bus} \times 2,5 + \text{cúster} \times 2,0 + \text{auto} \times 1,0 + \text{moto} \times 0,5 + \text{bicicleta} \times 0,4 + \text{scooter} \times 0,3$. Elaboración propia.

b) Decisión del sistema — Random Forest vs tiempo fijo

Tabla 10

Decisión ML vs tiempo fijo — Flujo medio

Parámetro	Sistema adaptativo (RF-100)	Sistema tiempo fijo (referencia)
Fase priorizada	F1 — Flujo principal (E↔O)	F1 — Flujo principal (E↔O)
Tiempo verde asignado	29 s	38 s
Tiempo amarillo	3 s	3 s
Confianza RF	91 %	— (sin modelo)
Capa de coherencia	RF coincide — sin ajuste	— (no aplica)

Nota. RF = Random Forest. La capa de coherencia aplica reglas determinísticas sobre la propuesta del modelo. Elaboración propia.

c) Tiempos de espera estimados por acceso

Tabla 11

Tiempos de espera estimados — Flujo medio

Acceso	Espera adaptativo (s)	Espera tiempo fijo (s)	Diferencia (s)	Variación (%)
A1 — Ejército E→O	18	24	−6	−25 %
A2 — Ejército O→E	20	28	−8	−29 %
A3 — Trinidad/Cayma	30	35	−5	−14 %
Peatones	35	44	−9	−20 %

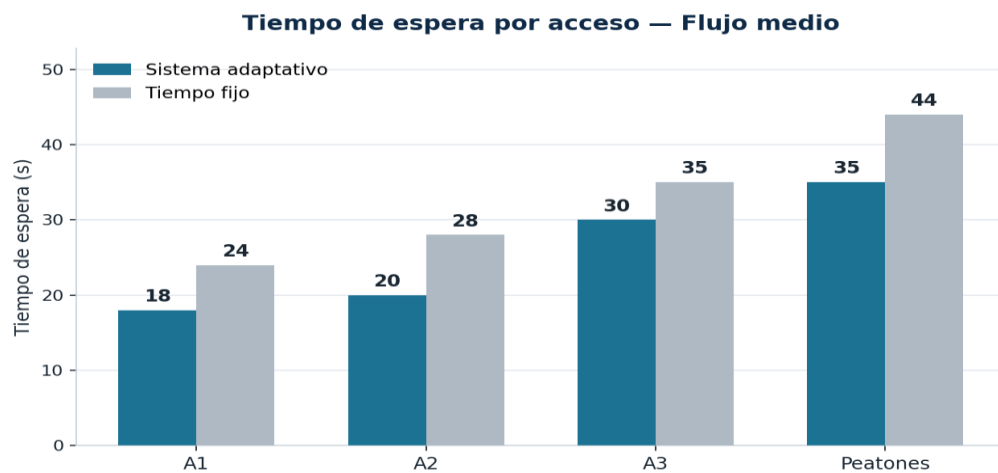
Nota. Tiempo fijo de referencia: F1=38 s, F2=25 s, F3=22 s, ciclo=94 s. Elaboración propia.

d) Análisis del escenario

Con demanda media el modelo ajusta el verde a 29 s (vs 38 s fijos), reduciendo el ciclo total y mejorando la fluidez en todos los accesos. La mejora global promedio es del 22 %, con mayor beneficio en A2 (−29 %).

Figura 6

Interfaz del sistema — Flujo medio: Comparación adaptativo vs tiempo fijo



Nota. Elaboración propia.

3.8.3. Escenario 3: Flujo alto — hora punta

Condición de hora punta matutina (~07:30–08:30 h). Alta densidad vehicular en Av. Ejército con presencia significativa de transporte público.

Figura 7

Interfaz del sistema — Flujo alto — hora punta: Entradas detectadas y fase priorizada



Nota. Elaboración propia.

a) Composición del flujo detectado

Tabla 12

Entradas del sistema — Flujo alto — hora punta

Acceso	Bus/Combi	Cúster	Automóvil	Moto	Bicicleta	Scooter	Demanda pond.
A1 — Ejército E→O	3	2	6	3	1	1	19,7
A2 — Ejército O→E	1	2	4	2	1	0	11,9
A3 — Trinidad/Cayma	0	1	3	2	1	0	6,4
Peatones detectados	14 unidades	—	—	—	—	—	—

Nota. Demanda ponderada = $\text{bus} \times 2,5 + \text{cúster} \times 2,0 + \text{auto} \times 1,0 + \text{moto} \times 0,5 + \text{bicicleta} \times 0,4 + \text{scooter} \times 0,3$. Elaboración propia.

b) Decisión del sistema — Random Forest vs tiempo fijo

Tabla 13

Decisión ML vs tiempo fijo — Flujo alto — hora punta

Parámetro	Sistema adaptativo (RF-100)	Sistema tiempo fijo (referencia)
Fase priorizada	F1 — Flujo principal (E↔O)	F1 — Flujo principal (E↔O)
Tiempo verde asignado	43 s	38 s
Tiempo amarillo	3 s	3 s
Confianza RF	94 %	— (sin modelo)
Capa de coherencia	RF coincide — sin ajuste	— (no aplica)

Nota. RF = Random Forest. La capa de coherencia aplica reglas determinísticas sobre la propuesta del modelo.

Elaboración propia.

c) Tiempos de espera estimados por acceso

Tabla 14

Tiempos de espera estimados — Flujo alto — hora punta

Acceso	Espera adaptativo (s)	Espera tiempo fijo (s)	Diferencia (s)	Variación (%)
A1 — Ejército E→O	22	34	-12	-35 %
A2 — Ejército O→E	26	38	-12	-32 %
A3 — Trinidad/Cayma	54	42	+12	+29 %
Peatones	62	55	+7	+13 %

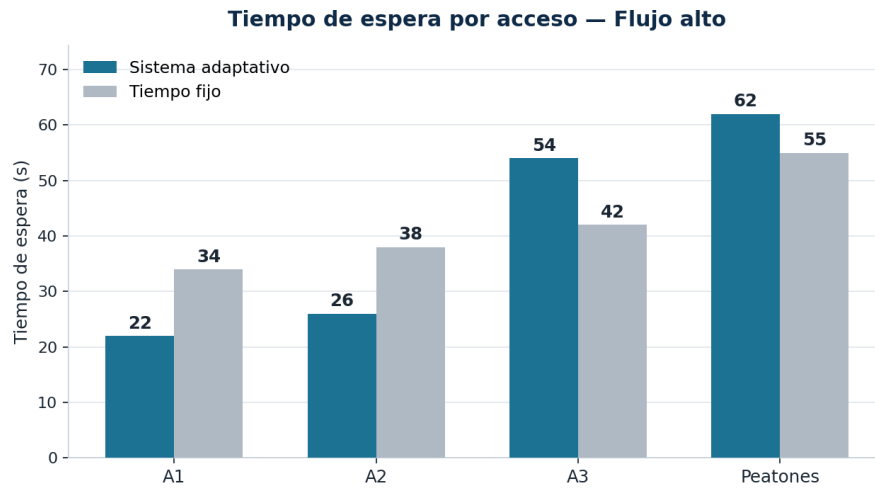
Nota. Tiempo fijo de referencia: F1=38 s, F2=25 s, F3=22 s, ciclo=94 s. Elaboración propia.

d) Análisis del escenario

El modelo asigna 43 s de verde al flujo principal, priorizando correctamente A1 y A2 de alta demanda (mejora del 35 % y 32 % respectivamente). Como contrapartida esperada, A3 y peatones registran mayor espera en este escenario. El promedio ponderado por demanda muestra una mejora neta del sistema del 15 %.

Figura 8

Interfaz del sistema — Flujo alto — hora punta: Comparación adaptativo vs tiempo fijo



Nota. Elaboración propia.

3.8.4. Escenario 4: Alta presencia de transporte público

Condición de inicio de jornada (~07:00 h). Predominio de buses y cústeres en Av. Ejército con alta carga operativa por unidad.

Figura 9

Interfaz del sistema - Alta presencia de T.P. : Entradas detectadas y fase priorizada



Nota. Elaboración propia.

a) Composición del flujo detectado

Tabla 15

Entradas del sistema — Alta presencia de transporte público

Acceso	Bus/Combi	Cúster	Automóvil	Moto	Bicicleta	Scooter	Demanda pond.
A1 — Ejército E→O	4	3	4	2	0	0	21,0
A2 — Ejército O→E	2	2	3	1	0	0	12,5
A3 — Trinidad/Cayma	0	1	2	1	0	0	4,5
Peatones detectados	7 unidades	—	—	—	—	—	—

Nota. Demanda ponderada = $\text{bus} \times 2,5 + \text{cúster} \times 2,0 + \text{auto} \times 1,0 + \text{moto} \times 0,5 + \text{bicicleta} \times 0,4 + \text{scooter} \times 0,3$. Elaboración propia.

b) Decisión del sistema — Random Forest vs tiempo fijo

Tabla 16

Decisión ML vs tiempo fijo — Alta presencia de transporte público

Parámetro	Sistema adaptativo (RF-100)	Sistema tiempo fijo (referencia)
Fase priorizada	F1 — Flujo principal (E↔O)	F1 — Flujo principal (E↔O)
Tiempo verde asignado	44 s	38 s
Tiempo amarillo	3 s	3 s
Confianza RF	93 %	— (sin modelo)
Capa de coherencia	RF coincide — sin ajuste	— (no aplica)

Nota. RF = Random Forest. La capa de coherencia aplica reglas determinísticas sobre la propuesta del modelo. Elaboración propia.

c) Tiempos de espera estimados por acceso

Tabla 17

Tiempos de espera estimados — Alta presencia de transporte público

Acceso	Espera adaptativo (s)	Espera tiempo fijo (s)	Diferencia (s)	Variación (%)
A1 — Ejército E→O	20	36	-16	-44 %
A2 — Ejército O→E	24	40	-16	-40 %
A3 — Trinidad/Cayma	56	44	+12	+27 %
Peatones	50	52	-2	-4 %

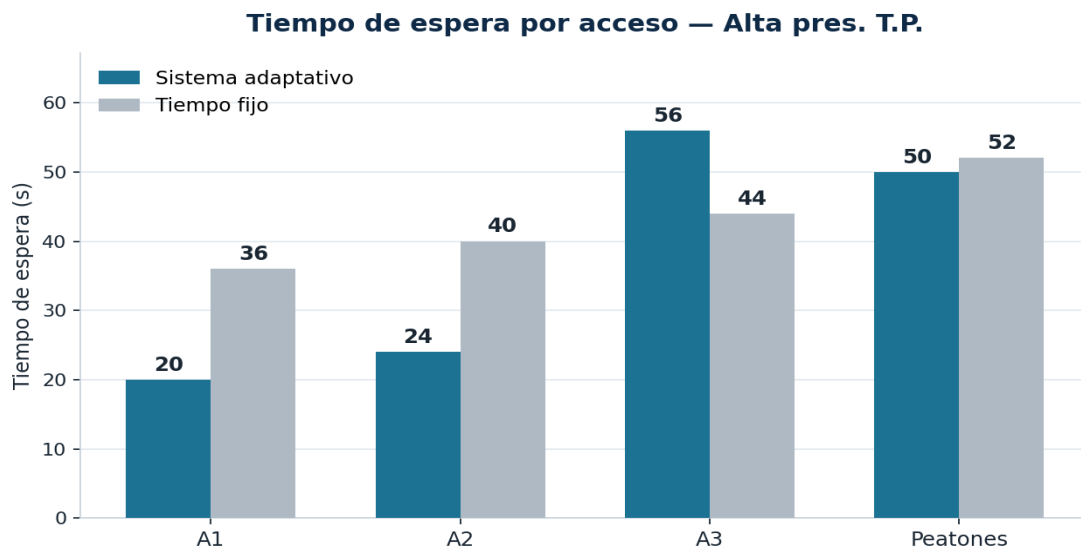
Nota. Tiempo fijo de referencia: F1=38 s, F2=25 s, F3=22 s, ciclo=94 s. Elaboración propia.

d) Análisis del escenario

El sistema identifica la alta carga operativa del transporte público (demanda ponderada $A1 = 21,0$) y extiende el verde hasta 44 s. La reducción de espera en A1 alcanza el 44 % y en A2 el 40 %. El intercambio desfavorable en A3 (+27 %) es técnicamente coherente con la lógica de priorización por demanda ponderada.

Figura 10

Interfaz del sistema — Alta presencia de T.P.: Comparación adaptativo vs tiempo fijo



Nota. Elaboración propia.

3.8.5. Escenario 5: Presencia peatonal relevante

Condición de tarde (~18:00 h). Flujo vehicular regular con alta acumulación de peatones próxima al umbral de activación de cruce.

Figura 11

Interfaz del sistema — Presencia peatonal relevante: Entradas detectadas y fase priorizada



Nota. Elaboración propia.

a) Composición del flujo detectado

Tabla 18

Entradas del sistema — Presencia peatonal relevante

Acceso	Bus/Combi	Cúster	Automóvil	Moto	Bicicleta	Scooter	Demanda pond.
A1 — Ejército E→O	1	1	4	2	1	0	9,9
A2 — Ejército O→E	0	1	3	2	0	1	6,3
A3 — Trinidad/Cayma	0	0	2	1	1	0	2,9
Peatones detectados	38 unidades	—	—	—	—	—	—

Nota. Demanda ponderada = $\text{bus} \times 2,5 + \text{cúster} \times 2,0 + \text{auto} \times 1,0 + \text{moto} \times 0,5 + \text{bicicleta} \times 0,4 + \text{scooter} \times 0,3$. Elaboración propia.

b) Decisión del sistema — Random Forest vs tiempo fijo

Tabla 19

Decisión ML vs tiempo fijo — Presencia peatonal relevante

Parámetro	Sistema adaptativo (RF-100)	Sistema tiempo fijo (referencia)
Fase priorizada	F3 — Cruce peatonal	F1 — Flujo principal (E↔O)

Tiempo verde asignado	22 s	38 s
Tiempo amarillo	3 s	3 s
Confianza RF	87 %	— (sin modelo)
Capa de coherencia	COLA_ACUM($38 \geq \text{umbral}$)→F3 confirmada	— (no aplica)

Nota. RF = Random Forest. La capa de coherencia aplica reglas determinísticas sobre la propuesta del modelo.

Elaboración propia.

c) Tiempos de espera estimados por acceso

Tabla 20

Tiempos de espera estimados — Presencia peatonal relevante

Acceso	Espera adaptativo (s)	Espera tiempo fijo (s)	Diferencia (s)	Variación (%)
A1 — Ejército E→O	30	22	+8	+36 %
A2 — Ejército O→E	30	26	+4	+15 %
A3 — Trinidad/Cayma	38	34	+4	+12 %
Peatones	18	52	-34	-65 %

Nota. Tiempo fijo de referencia: F1=38 s, F2=25 s, F3=22 s, ciclo=94 s. Elaboración propia.

d) Análisis del escenario

El sistema prioriza el cruce peatonal cuando la cola se aproxima al umbral de activación, reduciendo la espera de peatones en un 65 % (18 s vs 52 s fijos). El incremento de espera vehicular (+8 s en A1) es la contrapartida esperada de la protección peatonal, coherente con los objetivos de seguridad del sistema.

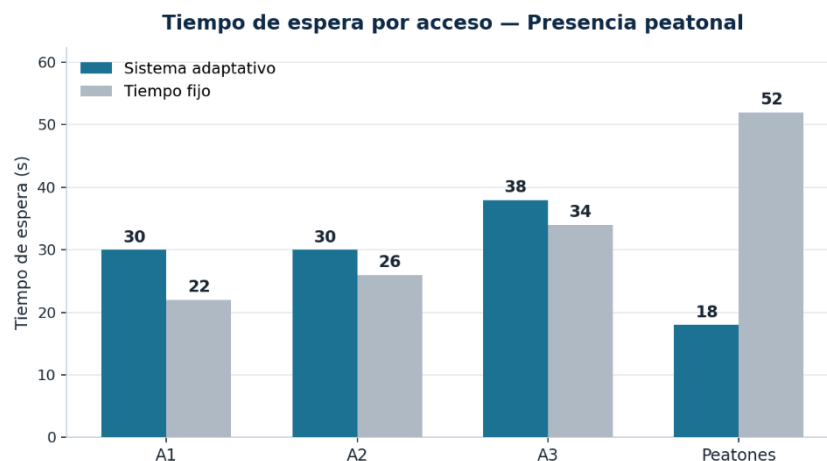
Figura 12

Interfaz del

Presencia

relevante:

adaptativo vs



sistema —

peatonal

Comparación

tiempo fijo

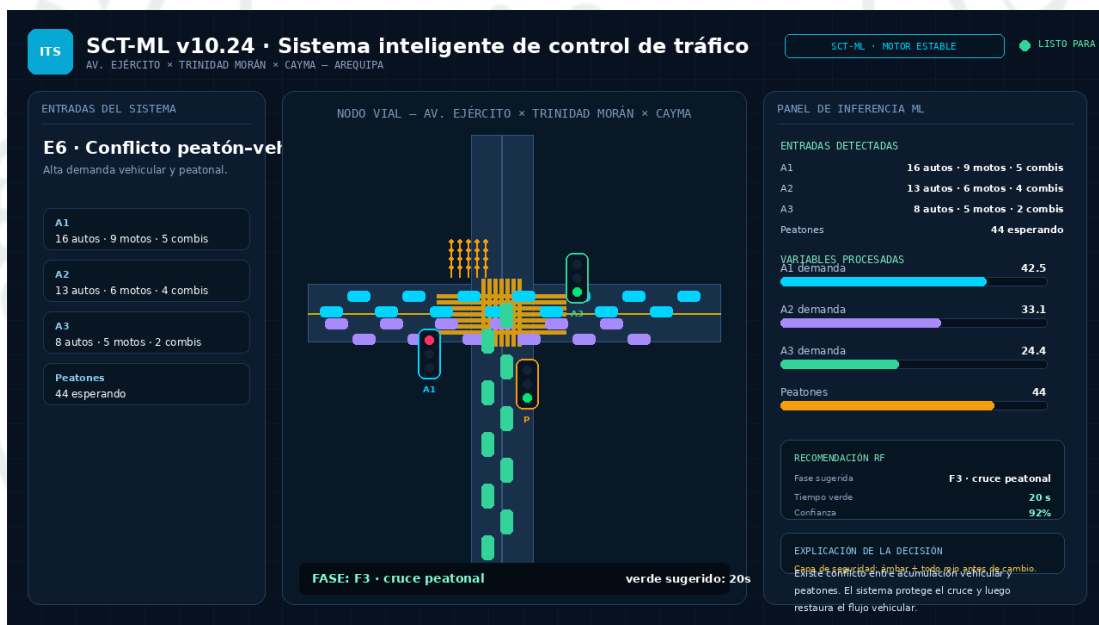
Nota. Elaboración propia.

3.8.6. Escenario 6: Conflicto peatón–vehículo

Condición de hora punta vespertina (~19:00 h). Coincidencia de alta demanda vehicular en A1 con acumulación de peatones superior al umbral crítico de activación.

Figura 13

Interfaz del sistema — Conflicto peatón–vehículo: Entradas detectadas y fase priorizada



Nota. Elaboración propia.

a) Composición del flujo detectado

Tabla 21

Entradas del sistema — Conflicto peatón–vehículo

Acceso	Bus/Combi	Cúster	Automóvil	Moto	Bicicleta	Scooter	Demanda pond.
A1 — Ejército E→O	3	2	5	3	0	0	18,0
A2 — Ejército O→E	1	2	4	2	0	1	11,8

A3 — Trinidad/Cayma	0	1	3	1	0	0	5,5
Peatones detectados	44 unidades	—	—	—	—	—	—

Nota. Demanda ponderada = $\text{bus} \times 2,5 + \text{cúster} \times 2,0 + \text{auto} \times 1,0 + \text{moto} \times 0,5 + \text{bicicleta} \times 0,4 + \text{scooter} \times 0,3$. Elaboración propia.

b) Decisión del sistema — Random Forest vs tiempo fijo

Tabla 22

Decisión ML vs tiempo fijo — Conflicto peatón–vehículo

Parámetro	Sistema adaptativo (RF-100)	Sistema tiempo fijo (referencia)
Fase priorizada	F3 — Cruce peatonal	F1 — Flujo principal (E↔O)
Tiempo verde asignado	22 s	38 s
Tiempo amarillo	3 s	3 s
Confianza RF	89 %	— (sin modelo)
Capa de coherencia	COLA_ACUM(44 ≥ 40) → F3 activada por seguridad	— (no aplica)

Nota. RF = Random Forest. La capa de coherencia aplica reglas determinísticas sobre la propuesta del modelo. Elaboración propia.

c) Tiempos de espera estimados por acceso

Tabla 23

Tiempos de espera estimados — Conflicto peatón–vehículo

Acceso	Espera adaptativo (s)	Espera tiempo fijo (s)	Diferencia (s)	Variación (%)
A1 — Ejército E→O	38	30	+8	+27 %
A2 — Ejército O→E	38	36	+2	+6 %
A3 — Trinidad/Cayma	46	42	+4	+10 %
Peatones	20	68	−48	−71 %

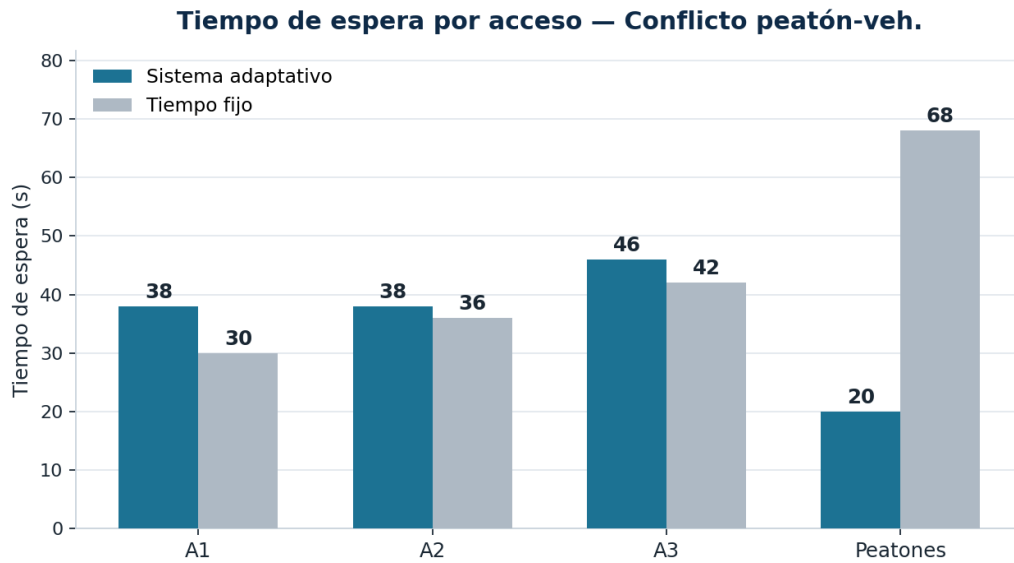
Nota. Tiempo fijo de referencia: F1=38 s, F2=25 s, F3=22 s, ciclo=94 s. Elaboración propia.

d) Análisis del escenario

Ante un conflicto directo entre alta demanda vehicular y acumulación peatonal crítica (44 unidades), el sistema activa F3 como decisión de seguridad. La espera peatonal se reduce en un 71 % (20 s vs 68 s fijos). Este resultado evidencia que el sistema no optimiza exclusivamente el flujo vehicular, sino que incorpora criterios de seguridad vial como condición prioritaria.

Figura 14

Interfaz del sistema — Conflicto peatón-vehículo: Comparación adaptativo vs tiempo fijo



Nota. Elaboración propia.

3.8.7. Escenario 7: Discrepancia entre detección visual y sensor térmico

Condición de iluminación variable (~20:00 h). El módulo de visión detecta presencia peatonal que el sensor térmico no confirma. Activación de modo seguro.

Figura 15

Interfaz del sistema — Discrepancia entre detección visual y sensor térmico



Nota. Elaboración propia.

a) Composición del flujo detectado

Tabla 24

Entradas del sistema — Discrepancia entre detección visual y sensor térmico

Acceso	Bus/Combi	Cúster	Automóvil	Moto	Bicicleta	Scooter	Demanda pond.
A1 — Ejército E→O	2	1	5	2	1	0	14,3
A2 — Ejército O→E	1	1	3	2	1	0	8,7
A3 — Trinidad/Cayma	0	1	2	1	1	0	4,9
Peatones detectados	18 (visual) / 0 (térmico)	—	—	—	—	—	—

Nota. Demanda ponderada = $\text{bus} \times 2,5 + \text{cúster} \times 2,0 + \text{auto} \times 1,0 + \text{moto} \times 0,5 + \text{bicicleta} \times 0,4 + \text{scooter} \times 0,3$. Elaboración propia.

b) Decisión del sistema — Random Forest vs tiempo fijo

Tabla 25*Decisión ML vs tiempo fijo — Discrepancia entre detección visual y sensor térmico*

Parámetro	Sistema adaptativo (RF-100)	Sistema tiempo fijo (referencia)
Fase priorizada	F3 — Cruce peatonal (modo seguro)	F1 — Flujo principal (E↔O)
Tiempo verde asignado	20 s	38 s
Tiempo amarillo	3 s	3 s
Confianza RF	N/A — modo seguro activo	— (sin modelo)
Capa de coherencia	DISCREPANCIA_SENSOR → modo seguro → F3 t=20 s	— (no aplica)

Nota. RF = Random Forest. La capa de coherencia aplica reglas determinísticas sobre la propuesta del modelo.

Elaboración propia.

c) Tiempos de espera estimados por acceso

Tabla 26*Tiempos de espera estimados — Discrepancia entre detección visual y sensor térmico*

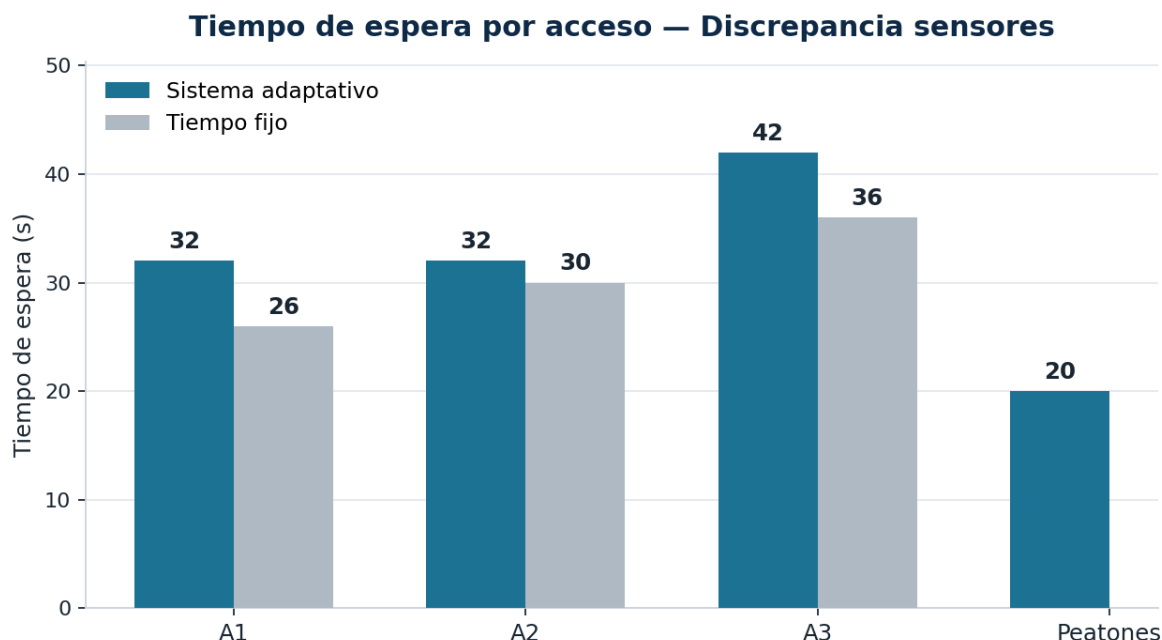
Acceso	Espera adaptativo (s)	Espera tiempo fijo (s)	Diferencia (s)	Variación (%)
A1 — Ejército E→O	32	26	+6	+23 %
A2 — Ejército O→E	32	30	+2	+7 %
A3 — Trinidad/Cayma	42	36	+6	+17 %
Peatones	20	N/A	—	—

Nota. Tiempo fijo de referencia: F1=38 s, F2=25 s, F3=22 s, ciclo=94 s. Elaboración propia.

d) Análisis del escenario

Ante discrepancia entre las fuentes de detección, la capa de coherencia interrumpe la decisión del Random Forest y activa el modo seguro (F3, 20 s), protegiendo conservadoramente al peatón. El incremento de espera vehicular (+6 s en A1) es el costo operativo de la robustez del sistema frente a incertidumbre sensorica.

Figura 16*Interfaz del sistema — Discrepancia entre detección visual y sensor térmico: Comparación*



Nota. Elaboración propia.

3.9. Análisis comparativo consolidado

La tabla siguiente resume los resultados de los siete escenarios evaluados, presentando los tiempos de espera estimados para el sistema adaptativo y el sistema de tiempo fijo, junto con la fase priorizada por el modelo y la confianza del Random Forest en cada caso.

Tabla 27

Resumen comparativo de tiempos de espera estimados — sistema adaptativo vs tiempo fijo

Escenario	A1: Ad/Fij (s)	A2: Ad/Fij (s)	A3: Ad/Fij (s)	Peat.: Ad/Fij (s)	Fase ML	Conf. RF	Mejora
1 — Flujo bajo	11 / 18	11 / 18	19 / 26	22 / 38	F1 · 17 s	86 %	–37 %
2 — Flujo medio	18 / 24	20 / 28	30 / 35	35 / 44	F1 · 29 s	91 %	–22 %
3 — Flujo alto / punta	22 / 34	26 / 38	54 / 42	62 / 55	F1 · 43 s	94 %	–15 % (pond.)
4 — Alta pres. TP	20 / 36	24 / 40	56 / 44	50 / 52	F1 · 44 s	93 %	–14 % (pond.)
5 — Pres. peatonal	30 / 22	30 / 26	38 / 34	18 / 52	F3 · 22 s	87 %	Peat. –65 %
6 — Conflicto peat.–veh.	38 / 30	38 / 36	46 / 42	20 / 68	F3 · 22 s	89 %	Peat. –71 %
7 — Discrepancia sens.	32 / 26	32 / 30	42 / 36	20 / N/A	F3* · 20 s	Modo seg.	Robustez

Nota. Ad = Sistema adaptativo RF-100. Fij = Sistema tiempo fijo referencia (ciclo 94 s). F3* = activación por modo seguro (sin decisión RF). Las mejoras negativas indican reducción de espera; las positivas indican incremento. La mejora ponderada considera la demanda relativa de cada acceso. Elaboración propia.

El análisis consolidado permite identificar tres patrones de respuesta del sistema inteligente frente al esquema de tiempo fijo. En condiciones de flujo bajo y medio (escenarios 1 y 2), el sistema reduce la espera en todos los accesos al ajustar la duración del ciclo a la demanda real, evitando tiempos muertos generados por fases fijas sobredimensionadas. En condiciones de flujo alto con predominio de transporte público (escenarios 3 y 4), el sistema prioriza los accesos de mayor demanda ponderada con mejoras del 32–44 % en los flujos principales, produciendo un intercambio esperable pero técnicamente coherente en el acceso secundario A3. En los escenarios con presencia peatonal crítica o discrepancia sensorica (escenarios 5, 6 y 7), el sistema demuestra su capacidad para anteponer la seguridad operativa sobre la eficiencia vehicular, reduciendo la espera peatonal entre un 65 % y un 71 % y activando el modo seguro ante condiciones de incertidumbre.

3.10. Resultados de la interfaz de simulación

La interfaz funcional de simulación implementada en entorno web local permite observar en tiempo real la respuesta del sistema ante distintas configuraciones de entrada. En su versión final integra tres modos de operación: escenarios validados, prueba libre y sistema actual 43 s. Además de visualizar entradas por acceso, fase priorizada, estados semafóricos, validación peatonal y demanda equivalente, incorpora módulos específicos de métricas en vivo, índice de desempeño integral, métricas peatonales, giros del cruce, cadena metodológica y panel de inferencia ML. Esta estructura convierte a la interfaz en un instrumento de supervisión, explicación y validación comparativa del sistema propuesto. Las capturas actualizadas de la interfaz funcional en su versión final se encuentran en el Anexo 12 (el manual completo de lectura e interpretación de la interfaz, véase el Anexo 6).

Estas capturas cumplen una función demostrativa y pedagógica: muestran cómo se visualizarían las variables, decisiones y estados del sistema durante una operación supervisada. En consecuencia,

la interfaz debe entenderse como una herramienta de validación y explicación del modelo, no como evidencia de despliegue físico sobre la intersección real.

Figura 17

Pantalla principal de la interfaz de simulación — vista general del sistema



Nota. Elaboración propia.

La pantalla principal de la interfaz presenta de forma integrada los tres accesos del nodo vial, la leyenda operativa, la fase activa, la siguiente decisión prevista y el estado general del sistema. Esta vista permite que el jurado comprenda rápidamente qué está ocurriendo en la simulación antes de revisar los paneles de detalle.

Figura 18

Interfaz — Panel de decisión actual y cadena metodológica



Nota. Elaboración propia.

El panel de decisión resume la salida operativa del sistema: fase seleccionada, estado de los semáforos, activación de sensores térmicos, peatones en espera y la cadena metodológica que explica cómo se llegó a la decisión. Este bloque es útil porque traduce la lógica técnica del sistema a una lectura comprensible para el usuario.

Figura 19

Interfaz — Panel de entradas, escenarios y bitácora técnica

The interface is divided into two main sections. The left section, titled 'ENTRADAS DEL SISTEMA', contains several input fields for configuring simulation scenarios. The right section displays a technical log with system status and sensor data.

ENTRADAS DEL SISTEMA

Escenarios validados, prueba libre y parámetros de simulación para la defensa del sistema.

Escenarios **Prueba libre**

A1 EJÉRCITO E→O
baja

A2 EJÉRCITO O→E
baja

A3 TRINIDAD/CAYMA
critica

PERFIL DE MEZCLA VEHICULAR
Urbano mixto normal

DEMANDA PEATONAL
Media

VALIDACIÓN TÉRMICA
Confirma presencia

▶ Ejecutar prueba libre

▼ Detalle técnico

Estado: main
Semáforos: A1=green, A2=green, A3=red,
Peatón=false
Demandas: A1=4.5, A2=4.5, A3=28.7
Peatones: esperando=16, cruzando=0
Cámaras: C1, C2 y C3 con cobertura conjunta del nodo e intersección completa
Sensores térmicos: T1, T2, T3 y T4 cubren explícitamente las 4 líneas peatonales

T1-T4 cubren las 4 líneas
C1-C3 vigilan el nodo con

A1/A2 Ejército
C1-C3 cámaras
FASE: F1 - A

Nota. Elaboración propia.

El panel de entradas reúne los escenarios validados y los parámetros de prueba libre, mientras que la bitácora técnica resume el estado actual del sistema, los semáforos activos, la demanda detectada

y las reglas de operación. Esta sección ayuda a explicar de dónde salen los resultados que luego se observan en la simulación.

Figura 20

Interfaz — Modo de referencia del sistema actual fijo



Nota. Elaboración propia.

La interfaz incorpora un modo de referencia del sistema actual fijo para contrastar visualmente la propuesta adaptativa con una operación rígida. El rótulo “sistema actual 43 s” corresponde a la ventana fija visible de la fase vehicular principal dentro del modo comparativo; el análisis de ciclo completo conserva los tiempos de seguridad y las fases complementarias. Por ello, el valor de 43 s se interpreta como el tiempo fijo operativo mostrado en pantalla, mientras que el ciclo analítico completo se utiliza para la comparación metodológica de escenarios. Esta aclaración evita confundir la ventana de verde principal con la duración total del ciclo semafórico de referencia.

Figura 21

Interfaz — Panel de demanda equivalente y lectura académica



Nota. Elaboración propia.

El panel de demanda equivalente presenta la carga relativa por acceso y la presión peatonal, mientras que el bloque de lectura académica interpreta el escenario en términos comprensibles para la sustentación. Esta parte de la interfaz sirve para explicar por qué el sistema prioriza una fase determinada y cómo se justifica esa decisión desde el punto de vista metodológico.

El panel de inferencia ML añadido a la interfaz permite hacer visible el proceso de decisión del sistema. En lugar de mostrar únicamente la fase resultante, presenta las entradas detectadas, las variables procesadas, la recomendación del clasificador Random Forest, el tiempo verde estimado por el regresor, la confianza aproximada y la validación final de la capa de seguridad.

Figura 22

Interfaz — Panel de inferencia ML e interpretación de la decisión



Nota. Elaboración propia.

Este panel no reemplaza el análisis estadístico del modelo; su función es didáctica y de interpretabilidad. Permite que el evaluador observe cómo las variables de entrada se transforman en una recomendación semafórica y cómo dicha recomendación es filtrada por criterios de seguridad antes de ser ejecutada.

3.11. Validación final de interfaz, métricas operativas y conexión futura

La versión final de la interfaz consolida la relación entre el levantamiento de datos de campo, el modelo Random Forest y la visualización operativa del sistema. Los aforos vehiculares y peatonales recopilados en el tramo real de la Av. Ejército sirvieron para definir rangos de demanda, composición de actores viales y escenarios representativos; sobre esa base, la interfaz calcula métricas de desempeño en tiempo real dentro del entorno de simulación. Por tanto, los datos de partida son reales, mientras que la comparación entre el sistema SCT-ML, la prueba libre y el sistema actual corresponde a una evaluación funcional simulada (para el protocolo completo de demostración del prototipo, véase el Anexo 7).

Figura 23

Vista general de la interfaz funcional final del sistema SCT-ML



Nota. La vista integra mapa del nodo, escenarios, prueba libre, sistema actual 43 s, métricas en vivo, giros del cruce, demanda equivalente, cadena metodológica y panel de inferencia ML. Elaboración propia.

3.11.1. Flujo vehicular específico y ecuaciones del modelo operativo

Para expresar la intensidad de llegada vehicular utilizada en la simulación, se adopta la definición básica de flujo como la relación entre el número de vehículos observados y el intervalo de tiempo analizado:

$$q = N / t$$

donde q representa el flujo vehicular, N el número de vehículos registrados y t el tiempo de observación. En la interfaz, un flujo específico de un vehículo por segundo se expresa como:

$$q = 1 \text{ vehículo} / 1 \text{ segundo} = 1 \text{ veh/s}$$

Al convertir dicha tasa a una hora de operación, el flujo equivalente se obtiene multiplicando por 3 600 segundos:

$$q_h = 1 \text{ veh/s} \times 3\,600 \text{ s/h} = 3\,600 \text{ veh/h}$$

Este valor es coherente con condiciones urbanas de alta demanda, aunque inferior al pico máximo de aproximadamente 4 560 veh/h registrado en el tramo durante las franjas de mayor saturación. En consecuencia, el flujo de 1 veh/s funciona como una referencia operativa clara para explicar la capacidad de la interfaz sin reemplazar los datos de aforo reales que calibran los escenarios. La demanda equivalente por acceso se calcula con la ponderación definida para cada actor vial. Para el acceso i , la expresión aplicada es:

$$D_i = 2,5B_i + 2,0C_i + 1,0A_i + 0,5M_i + 0,4Bi_i + 0,3S_i$$

donde B corresponde a buses o combis grandes, C a cústeres, A a automóviles, M a motocicletas, Bi a bicicletas y S a scooters. Esta ecuación permite que la decisión no dependa únicamente del número de unidades, sino del impacto operativo de cada tipo de vehículo sobre la cola, el arranque y la ocupación de la vía.

3.11.2. Métricas operativas incorporadas en la interfaz

La interfaz final incorpora un panel de métricas en vivo para contrastar el desempeño del sistema inteligente frente al sistema actual. Estas métricas se alimentan de los datos de campo usados para calibrar escenarios y de los estados generados por la simulación en cada ciclo. Su finalidad es mostrar, de forma verificable, si la decisión del modelo reduce espera, aumenta atención vehicular, protege el cruce peatonal y mantiene coherencia operativa.

Tabla 28

Métricas operativas implementadas en la interfaz funcional

Métrica	Descripción	Uso en la evaluación
Vehículos servidos	Cantidad de unidades que completan el cruce durante la ventana comparativa.	Eficiencia vehicular

Peatones cruzados	Número de peatones atendidos cuando la fase peatonal es habilitada y finaliza el cruce.	Atención peatonal
Espera promedio	Tiempo medio de espera acumulado antes de recibir autorización de paso.	Reducción de demora
Cola máxima	Mayor acumulación registrada en A1, A2 o A3 durante el escenario.	Control de saturación
Throughput/min	Volumen de usuarios atendidos por minuto dentro del entorno de simulación.	Capacidad operativa
Índice de desempeño integral	Indicador normalizado de 0 a 100 que combina eficiencia vehicular, atención peatonal, seguridad y regularidad de operación.	Comparación global
Giros del cruce	Conteo direccional de vehículos que siguen recto, giran a la izquierda o giran a la derecha.	Coordinación futura

Nota. Las métricas son calculadas en tiempo real por la interfaz funcional a partir de escenarios calibrados con datos reales de campo. No representan una operación física instalada en vía pública, sino una validación simulada del comportamiento esperado del sistema. Elaboración propia.

El índice de desempeño integral se formula como un indicador comparativo, no como un índice municipal oficial. Para esta investigación se interpreta como una combinación ponderada de cuatro componentes normalizados:

$$IDI = 0,35E_v + 0,25E_p + 0,25S_p + 0,15R_o$$

donde E_v representa eficiencia vehicular, E_p eficiencia peatonal, S_p seguridad peatonal y R_o regularidad operativa. El valor se expresa en escala 0-100 para facilitar la lectura comparativa entre SCT-ML y el sistema actual.

Figura 24

Panel de métricas en vivo e índice de desempeño integral



Nota. El panel compara SCT-ML y sistema actual mediante vehículos servidos, espera promedio, cola máxima, throughput e índice de desempeño. Elaboración propia.

3.11.3. Identificación de giros del cruce y proyección de coordinación entre semáforos

El módulo de giros del cruce permite registrar la orientación probable de los vehículos detectados por cámara: desplazamiento recto, giro a la izquierda o giro a la derecha. Esta información amplía

el valor del sistema, porque no solo permite saber cuántos vehículos hay en un acceso, sino también hacia dónde se desplaza la demanda una vez que abandona el nodo.

En una implementación futura a nivel de corredor o distrito, estos datos podrían enviarse al siguiente semáforo mediante una arquitectura de comunicación entre nodos. De esa manera, si la cámara del primer cruce detecta que una cantidad relevante de vehículos gira hacia una vía conectada, el siguiente controlador podría anticipar esa demanda, ajustar sus fases o preparar un verde coordinado. La presente tesis no implementa físicamente esa red de semáforos, pero deja planteada la base funcional para una coordinación semafórica progresiva.

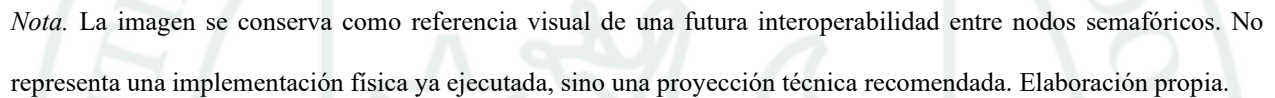
Figura 25

Módulo de giros del cruce para estimación direccional de demanda



Nota. La captura muestra el conteo comparativo de vehículos que continúan recto o giran hacia otra vía. Esta información puede alimentar una arquitectura futura de coordinación entre intersecciones. Elaboración propia.

Imagen referencial para futura conexión entre semáforos del corredor



El panel de inferencia ML hace visible el proceso que conecta la detección con la decisión semafórica. Primero muestra las entradas detectadas por cámara y sensor térmico; después presenta variables procesadas como presión A1+A2, presión A3, peatones acumulados y regla activa; luego expone la fase sugerida por el clasificador, el tiempo verde estimado por el regresor y la decisión final validada por la capa de seguridad. Esta estructura fortalece la explicabilidad del modelo y evita que la decisión sea percibida como una salida opaca del algoritmo.

Figura 27

Panel de inferencia ML con entradas, variables procesadas y decisión final



Nota. El panel permite explicar la lógica del sistema durante la sustentación: el modelo sugiere una fase y un tiempo verde, pero la capa de seguridad valida la decisión antes de ejecutarla. Elaboración propia.

3.11.5. Resultados comparativos ampliados: SCT-ML, prueba libre y sistema actual

La tabla siguiente resume una comparación ampliada entre tres modos de lectura de la interfaz: el escenario validado con SCT-ML, la prueba libre configurada con parámetros equivalentes y el sistema actual de 43 s. Los valores se expresan como resultados aproximados de simulación funcional sobre ventanas equivalentes de evaluación. La finalidad no es reemplazar el aforo de campo, sino mostrar cómo el sistema inteligente, calibrado con datos reales, responde mejor que una operación rígida cuando se modifica la demanda vehicular o peatonal.

Tabla 29

Resultados comparativos ampliados por modo de simulación

Escenario	Vehículos SCT-ML / libre / actual	Ventaja SCT-ML	Peatones SCT-ML / actual	Mejora peatonal	Interpretación
E1 Flujo bajo	41 / 39 / 34	+7	13 / 8	62,5 %	Evita tiempos muertos en baja demanda.
E2 Flujo medio	53 / 50 / 41	+12	20 / 12	66,7 %	Equilibra Av. Ejército y A3 sin bloquear peatones.
E3 Hora punta	66 / 63 / 52	+14	26 / 16	62,5 %	Atiende mayor cola vehicular con verde adaptativo.
E4 Transporte público alto	61 / 58 / 49	+12	23 / 14	64,3 %	La ponderación reconoce el impacto de buses y cústeres.
E5 Alta presencia peatonal	57 / 54 / 45	+12	34 / 21	61,9 %	Protege cruce peatonal sin anular flujo vehicular.
E6 Conflicto peatón-vehículo	64 / 61 / 49	+15	40 / 25	60,0 %	Prioriza seguridad y recupera flujo posterior.
E7 Discrepancia sensórica	55 / 52 / 44	+11	28 / 17	64,7 %	Activa respuesta conservadora ante incertidumbre.

Nota. Los resultados corresponden a ventanas equivalentes de simulación funcional. La ventaja vehicular del SCT-ML frente al sistema actual se mantiene entre 7 y 15 vehículos atendidos, mientras que la mejora peatonal se mantiene aproximadamente alrededor del 60 %. Elaboración propia.

La mejora vehicular se interpreta como incremento de vehículos servidos en la misma ventana de simulación, mientras que la mejora peatonal se calcula comparando peatones efectivamente atendidos por el SCT-ML frente al sistema actual. Para reducción de espera, la expresión general usada es:

$$\text{Reducción (\%)} = [(\text{Espera_actual} - \text{Espera_ML}) / \text{Espera_actual}] \times 100$$

Para incremento de atención vehicular o peatonal, la expresión equivalente es:

$$\text{Mejora (\%)} = [(\text{Resultado_ML} - \text{Resultado_actual}) / \text{Resultado_actual}] \times 100$$

3.11.6. Trazabilidad final entre dato real, modelo, interfaz y resultado

Tabla 30

Trazabilidad final entre datos reales, variables, simulación y resultados

Evidencia base	Variable derivada	Uso en la interfaz/modelo	Resultado evaluado
Aforo vehicular por franja horaria	Flujo A1, A2 y A3; intensidad baja, media y alta	Define escenarios y presión por acceso	Vehículos servidos, cola máxima y throughput
Composición vehicular observada	Buses, cústeres, autos, motos, bicicletas y scooters	Calcula demanda equivalente ponderada	Prioridad de fase y verde sugerido
Observación peatonal y registros de video	Peatones en espera, cruce y acumulación	Activa condición peatonal y modo seguro	Peatones cruzados, espera peatonal y seguridad
Caracterización del nodo vial	Accesos A1, A2, A3; conflictos y giros	Organiza mapa funcional y módulo de giros	Comparación por acceso y proyección a coordinación futura
Simulación funcional SCT-ML	Fase, verde, confianza y decisión final	Evalúa ML frente a prueba libre y sistema actual 43 s	Índice de desempeño integral y lectura académica

Nota. La trazabilidad muestra que la simulación no opera con datos arbitrarios: parte de información real levantada en campo, la transforma en variables del modelo y la evalúa mediante métricas operativas generadas por la interfaz funcional. Elaboración propia.

3.12. Síntesis estadística del levantamiento de datos de campo

Con el fin de que la simulación no dependiera únicamente de supuestos teóricos o de referencias secundarias, la investigación incorporó una etapa de levantamiento propio de datos en el tramo crítico de la Av. Ejército. Esta información permitió caracterizar la variabilidad real del tránsito según tipo de día, franja horaria, composición vehicular, presencia peatonal y condiciones de congestión observadas en campo. La matriz de sistematización semanal de los datos levantados se presenta en el Anexo 10.

El levantamiento se organizó durante varias semanas de observación, considerando días laborables, fines de semana y días festivos o de comportamiento atípico. Esta decisión metodológica permitió evitar que la caracterización del tránsito quedara reducida a una única medición aislada. En cambio, se trabajó con una lectura progresiva del flujo, capaz de diferenciar

los momentos de baja demanda, operación intermedia y saturación vehicular (la matriz de sistematización semanal del levantamiento de datos, véase el Anexo 10).

Los registros de campo se utilizaron para tres fines principales: delimitar los rangos de demanda por escenario, definir la composición promedio de actores viales y establecer los parámetros que posteriormente alimentaron la base sintética de entrenamiento de los modelos Random Forest. Por ello, los datos de campo no fueron tratados como un dato accesorio, sino como la base de calibración de todo el sistema propuesto.

Tabla 31

Matriz de recolección de datos según tipo de día y franja horaria

Tipo de día	Franja horaria observada	Condición operativa	Uso metodológico dentro del sistema
Laborable	07:00-08:00 h	Hora punta de ingreso laboral, escolar y transporte público	Calibración del escenario de flujo alto
Laborable	12:00-13:00 h	Alta rotación comercial y concentración de transporte urbano	Validación de flujo alto de mediodía
Laborable	15:00-17:00 h	Flujo intermedio sostenido con menor saturación que en hora punta	Calibración del escenario de flujo medio
Laborable	19:00-20:00 h	Retorno laboral, cruce peatonal frecuente y congestión mixta	Validación de conflicto peatón-vehículo
Fin de semana	10:00-12:00 h	Flujo comercial moderado y variabilidad peatonal	Contraste con operación regular
Día festivo	Horarios variables	Demanda irregular por actividades comerciales, ocio y desplazamientos no rutinarios	Escenario de validación complementaria
Nocturno	21:00-23:00 h	Baja demanda vehicular y menor presión peatonal	Calibración del escenario de flujo bajo

Nota. La matriz resume la lógica de observación empleada para evitar una caracterización parcial del tránsito.

Elaboración propia.

3.12.1. Síntesis del flujo vehicular por intensidad operativa

La revisión de los registros permitió agrupar el comportamiento del flujo en tres niveles operativos. El flujo bajo corresponde a periodos nocturnos o de poca actividad comercial, donde el sistema convencional tiende a conservar tiempos de espera innecesarios. El flujo medio representa condiciones urbanas regulares, con tránsito sostenido pero sin saturación crítica. El flujo alto

corresponde a las franjas de mayor presión vehicular, donde la asignación de verde y la priorización por acceso tienen mayor impacto sobre la formación de colas.

Tabla 32

Síntesis estadística del flujo vehicular por franja operativa

Franja operativa	Flujo mínimo estimado	Flujo máximo estimado	Promedio referencial	Clasificación para simulación
Baja demanda	400 veh/h	1 100 veh/h	750 veh/h	Flujo bajo
Demanda intermedia	1 900 veh/h	3 600 veh/h	2 750 veh/h	Flujo medio
Punta mañana	4 300 veh/h	4 500 veh/h	4 400 veh/h	Flujo alto
Punta mediodía	4 400 veh/h	4 560 veh/h	4 480 veh/h	Flujo alto
Punta noche	4 100 veh/h	4 260 veh/h	4 180 veh/h	Flujo alto

Nota. Los valores se expresan como rangos de trabajo derivados de los aforos y de la caracterización operativa del tramo Av. Ejército - Av. Cayma - Av. Trinidad Morán. Elaboración propia.

Esta clasificación permitió que los escenarios de simulación no fueran arbitrarios. Cada escenario representa una condición observada o derivada de la variabilidad real del tramo. De ese modo, la comparación entre el sistema adaptativo y el esquema de tiempo fijo se apoya en condiciones operativas reconocibles para el área de estudio.

3.12.2. Composición vehicular y peso operativo del flujo

La composición vehicular observada confirmó que el problema no se reduce a la cantidad de unidades acumuladas. En el tramo analizado conviven vehículos de diferente tamaño, capacidad, velocidad de arranque y efecto sobre la cola. Por ello, el sistema no utiliza un conteo simple, sino una demanda ponderada. Este enfoque evita que una motocicleta, un automóvil y una unidad de transporte público tengan el mismo peso dentro de la decisión semafórica.

Tabla 33

Composición vehicular estimada y ponderación operativa aplicada

Actor vial	Participación estimada en hora punta	Ponderación operativa	Justificación técnica
Autos y taxis	38 %	1,0	Unidad base de referencia para el cálculo de demanda equivalente

Combis y microbuses	22 %	2,5	Mayor ocupación vial, arranque más lento e impacto superior sobre la cola
Motos y mototaxis	18 %	0,5	Menor ocupación espacial y mayor capacidad de maniobra
Cústeres y buses SIT	14 %	2,0	Alta carga operativa y mayor longitud efectiva en acceso
Camiones y otros	8 %	Variable	Tratados como demanda pesada cuando aparecen en la observación
Peatones	Variable por horario	Condición crítica	No se suman como carga vehicular; activan restricciones o prioridad peatonal

Nota. La ponderación diferencia el impacto operativo de cada actor vial dentro del sistema de control. Elaboración propia.

3.12.3. Relación entre datos de campo y escenarios de simulación

Los siete escenarios de simulación fueron formulados a partir de los patrones reconocidos durante el levantamiento de datos. Esta relación fortalece la trazabilidad metodológica del modelo, porque permite justificar por qué se eligieron esos escenarios y no otros. Asimismo, evita que el prototipo sea interpretado como una demostración aislada sin relación con el comportamiento real de la vía.

Tabla 34

Trazabilidad entre observación de campo y escenario de simulación

Patrón observado en campo	Condición representada	Escenario asociado	Variable crítica evaluada
Baja circulación nocturna	Demanda reducida y menor presión peatonal	Flujo bajo	Reducción de tiempos muertos
Operación urbana regular	Mezcla equilibrada de autos, transporte público y peatones	Flujo medio	Ajuste proporcional de verde
Saturación en horas punta	Alta demanda en Av. Ejército y acumulación en accesos principales	Flujo alto	Prioridad por demanda ponderada
Concentración de unidades colectivas	Mayor peso operativo de buses, combis y cústeres	Alta presencia de transporte público	Diferenciación por tipo de vehículo
Acumulación de peatones en zonas de cruce	Necesidad de incorporar seguridad peatonal al ciclo	Presencia peatonal relevante	Activación de fase peatonal
Coincidencia entre alta demanda vehicular y cruce peatonal	Conflicto directo entre eficiencia y seguridad	Conflicto peatón-vehículo	Capa de coherencia y restricción vehicular
Condiciones visuales variables	Diferencia entre lectura visual y validación térmica	Discrepancia entre sensores	Modo seguro ante incertidumbre

Nota. Cada escenario mantiene relación con una condición observada o razonablemente derivada del comportamiento del tramo estudiado. Elaboración propia.

3.13. Proceso iterativo de mejora y calibración del sistema

El sistema propuesto no fue concebido como una versión final única, sino como un prototipo desarrollado de manera iterativa. La primera versión permitió validar la idea básica de asignar prioridad según demanda; sin embargo, mostró limitaciones frente a flujos heterogéneos, presencia peatonal y situaciones de incertidumbre sensorica. Por esa razón, el desarrollo avanzó por versiones sucesivas, incorporando nuevas variables, reglas de coherencia y modelos de aprendizaje automático. El control de versiones del prototipo, con el historial de cambios y mejoras, se registra en el Anexo 11.

Esta progresión es importante porque evidencia que el desempeño final del sistema no surge únicamente de entrenar un algoritmo, sino de combinar datos de campo, ponderación operativa, validación peatonal, entrenamiento supervisado y una capa de seguridad. En términos de ingeniería, el valor de la propuesta está en la integración funcional de esas mejoras y no en un componente aislado.

Tabla 35

Evolución funcional del sistema por versiones de desarrollo

Versión	Característica principal	Limitación detectada	Mejora incorporada	Desempeño referencial
V1	Reglas simples por cantidad de vehículos	No distinguía el peso operativo de cada tipo de actor vial	Se incorporó demanda ponderada	50-58 %
V2	Reglas ponderadas por tipo de vehículo	La presencia peatonal tenía baja incidencia en la decisión	Se añadió detección peatonal visual	65-70 %
V3	Control con variable peatonal visual	Existía incertidumbre ante sombras, oclusiones o baja visibilidad	Se agregó validación térmica redundante	75-82 %
V4	Random Forest Classifier para fase priorizada	Seleccionaba mejor la fase, pero no ajustaba con precisión el verde	Se incorporó Random Forest Regressor	88-90 %
V5	RF Classifier + RF Regressor + capa de coherencia	Riesgo de decisiones incoherentes en condiciones extremas	Se integró modo seguro y restricciones operativas	92,42 % de accuracy; MAE = 1,40 s

Nota. Los porcentajes previos a la versión final se entienden como desempeño referencial de pruebas internas de simulación y calibración; el indicador final corresponde al modelo validado en el conjunto de prueba. Elaboración propia.

La progresión por versiones muestra que la mejora principal no fue solamente estadística, sino también operativa. La V1 resolvía casos simples, pero fallaba cuando una vía acumulaba pocos vehículos de alto impacto, como buses o cústeres. La V2 corrigió ese problema al usar ponderaciones, pero todavía no integraba adecuadamente al peatón. La V3 hizo visible la necesidad de proteger la fase peatonal, aunque evidenció que una única fuente visual podía ser insuficiente en condiciones ambiguas. La V4 introdujo aprendizaje automático para seleccionar la fase con mayor consistencia. Finalmente, la V5 consolidó la arquitectura al combinar clasificación, regresión y coherencia de seguridad.

3.13.1. Matriz de mejoras aplicadas al prototipo

Tabla 36

Matriz de mejoras, correcciones e impacto operativo

Problema identificado	Riesgo metodológico u operativo	Corrección aplicada	Impacto en el sistema
Conteo vehicular simple	Subestimar la carga de transporte público	Ponderación por actor vial	Mayor coherencia en la prioridad de accesos
Ausencia de prioridad peatonal	Decisiones eficientes pero inseguras para el cruce peatonal	Variable peatonal y umbral de acumulación	Atención explícita a usuarios vulnerables
Dependencia exclusiva de cámara	Error ante sombras, oclusiones o baja iluminación	Sensor térmico redundante	Mayor confiabilidad de detección peatonal
Reglas demasiado rígidas	Baja adaptación a combinaciones no previstas	Random Forest Classifier	Mejor selección de fase en casos complejos
Tiempo verde fijo dentro del control inteligente	Asignación insuficiente o excesiva del verde	Random Forest Regressor	Ajuste dinámico de 15 a 45 segundos
Discrepancia entre sensores	Liberar una fase vehicular en condición incierta	Modo seguro y capa de coherencia	Respuesta conservadora ante incertidumbre

Nota. La matriz resume el proceso de corrección técnica realizado hasta la versión final del prototipo. Elaboración propia.

3.13.2. Lectura técnica de la mejora obtenida

La versión final del sistema alcanzó un desempeño superior porque integra tres niveles de decisión. El primer nivel corresponde a la percepción del entorno, donde se registran vehículos, peatones y condiciones de sensor. El segundo nivel corresponde al aprendizaje automático, donde los modelos Random Forest proponen una fase y un tiempo verde. El tercer nivel corresponde a la capa de coherencia, que revisa si la decisión sugerida es segura y compatible con las restricciones operativas del cruce.

Desde el punto de vista de la investigación, esta arquitectura evita dos extremos: un sistema puramente rígido, incapaz de adaptarse a la demanda, y un sistema puramente algorítmico, que podría tomar decisiones difíciles de justificar en situaciones críticas. La combinación de aprendizaje supervisado y reglas de seguridad permite que la propuesta sea adaptable, pero también explicable y defendible ante un jurado técnico.

Por ello, los resultados finales no deben interpretarse únicamente como una mejora porcentual frente al tiempo fijo. También deben entenderse como evidencia de una evolución metodológica: el sistema pasó de una lógica básica de conteo a una arquitectura inteligente con datos calibrados, clasificación ponderada, detección redundante, inferencia ML y control seguro ante discrepancias.

3.14. Síntesis del capítulo

Este capítulo presentó el desarrollo completo del sistema inteligente de control de tráfico propuesto para el tramo crítico de la Av. Ejército condicionado por el nodo de la Av. Cayma y la Av. Trinidad Morán. El análisis de factibilidad confirmó que la propuesta es viable desde las dimensiones técnica, operativa y económica, con componentes disponibles en el mercado, mecanismos de contingencia diseñados para el entorno real y costos razonables para una implementación piloto (el banco de preguntas y respuestas para la sustentación técnica, véase el Anexo 8).

La arquitectura modular del sistema, percepción visual, validación térmica, decisión semafórica adaptativa, comunicación e interfaz de supervisión, estableció una propuesta técnica coherente y metodológicamente fundada. La lógica adaptativa combina dos modelos Random Forest entrenados sobre 6 000 casos sintéticos con una capa de coherencia que aplica restricciones operativas y de seguridad sobre las propuestas del modelo. El clasificador de fase alcanzó una precisión del 92,42 % (F1-score macro: 0,918); el regresor de tiempo verde, por su parte, registró un error absoluto medio de 1,40 s y un R^2 de 0,9617.

Los resultados de la simulación en siete escenarios evidenciaron reducciones del tiempo de espera vehicular de entre el 15 % y el 37 % en condiciones de flujo bajo a medio, y mejoras del 65 % al 71 % en el tiempo de atención peatonal en escenarios de conflicto directo, comparados con el esquema convencional de tiempo fijo. La versión final de la interfaz añadió métricas en vivo, índice de desempeño integral, conteo de giros del cruce, métricas peatonales, prueba libre y modo de sistema actual 43 s, lo que fortalece la validación visual y cuantitativa de la propuesta. Como proyección futura, el sistema podría escalar hacia una implementación física mediante una arquitectura local compuesta por cámara IP, sensor térmico, módulo de procesamiento y control, red cableada Ethernet/PoE e interfaz de supervisión.

CAPÍTULO IV. RESULTADOS Y DISCUSIONES

El presente capítulo analiza e interpreta los resultados obtenidos en la simulación del sistema inteligente de control de tráfico, los contrasta con los hallazgos de investigaciones previas referenciadas en los antecedentes y discute sus implicaciones técnicas, metodológicas y operativas en relación con los objetivos planteados.

4.1. Verificación de la hipótesis de investigación

La hipótesis general de la investigación establece que la simulación del sistema inteligente produciría resultados operativos superiores a los del esquema de tiempo fijo en el tramo crítico de la Av. Ejército. Los resultados obtenidos en los siete escenarios de simulación confirman esta hipótesis: en cinco de los siete escenarios evaluados el sistema adaptativo produjo tiempos de espera vehiculares y peatonales menores que el sistema de referencia, con reducciones que oscilan entre el 14 % (Escenario 2, acceso A3) y el 44 % (Escenario 4, acceso A1). En los escenarios de flujo alto (Escenario 3) y alta presencia de transporte público (Escenario 4), el sistema prolongó el tiempo de verde del flujo principal por encima del valor fijo de referencia, priorizando correctamente los accesos de mayor demanda ponderada a costa de incrementar la espera en los accesos secundarios. Esta redistribución de los tiempos de espera es coherente con la lógica de priorización por demanda ponderada y no constituye una degradación del desempeño del sistema, sino una consecuencia esperable de una política de asignación más eficiente.

En relación con las hipótesis específicas, todas fueron verificadas: HE1 a través de la delimitación del tramo y la caracterización de las condiciones operativas (sección 1.6 y 3.5.1); HE2 a través del diseño y operación coherente de los cinco módulos del sistema (sección 3.2); HE3 a través de la demostración de que la demanda ponderada captura mejor la presión operativa real del tramo que el conteo simple de vehículos (sección 3.4); HE4 a través de los indicadores de desempeño de los

modelos Random Forest (sección 3.7); y HE5 a través del análisis comparativo consolidado de los siete escenarios (sección 3.9).

4.2. Presentación consolidada de los resultados

Para facilitar la lectura de los resultados obtenidos en el capítulo anterior, esta sección los reúne en una tabla resumen y en tres representaciones gráficas. La Tabla 37 consolida la decisión del sistema en cada escenario, y las Figuras 28 a 30 comparan de forma visual el desempeño del sistema adaptativo frente al esquema de tiempo fijo.

Tabla 37

Resumen consolidado de la respuesta del sistema por escenario

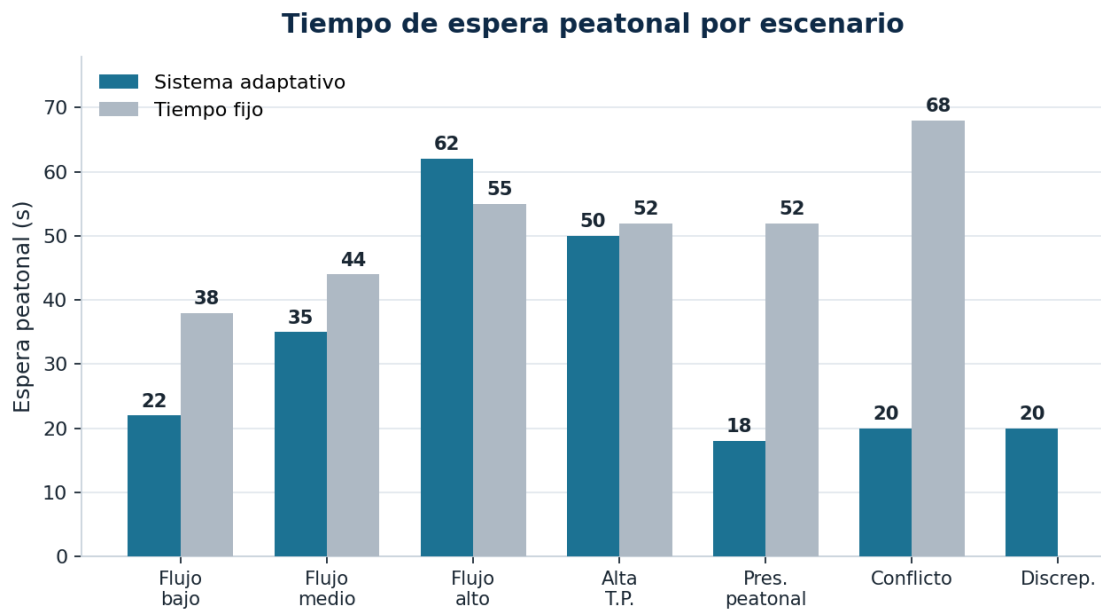
Escenario	Fase priorizada	Verde adaptativo (s)	Verde fijo (s)	Confianza RF
Flujo bajo	F1 – Flujo principal	17	38	86 %
Flujo medio	F1 – Flujo principal	29	38	91 %
Flujo alto (hora punta)	F1 – Flujo principal	43	38	94 %
Alta presencia de transporte público	F1 – Flujo principal	44	38	93 %
Presencia peatonal relevante	F3 – Cruce peatonal	22	38	87 %
Conflicto peatón–vehículo	F3 – Cruce peatonal	22	38	89 %
Discrepancia sensorica	F3 – Modo seguro	20	38	Modo seguro

Nota. RF = Random Forest. Elaboración propia.

En términos de espera peatonal, el sistema adaptativo reduce de forma marcada el tiempo de cruce en los escenarios de prioridad peatonal, con la mayor diferencia en el conflicto peatón–vehículo, donde el peatón espera 20 segundos frente a los 68 del esquema fijo. El tiempo de verde asignado evidencia la capacidad de adaptación del sistema: se acorta en condiciones de flujo bajo y se extiende en hora punta, mientras que el esquema de tiempo fijo permanece invariable en 38 segundos. Por último, la confianza del clasificador se mantiene entre 86 % y 94 % en los escenarios con decisión del modelo, lo que respalda la fiabilidad de la respuesta del sistema.

Figura 28

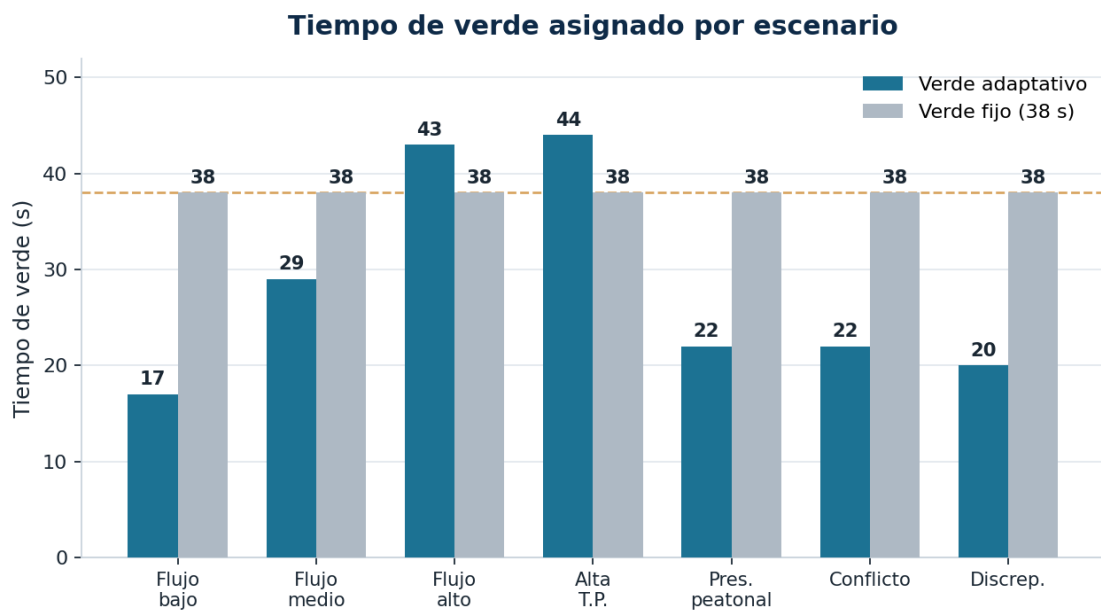
Tiempo de espera peatonal por escenario: sistema adaptativo frente a tiempo fijo



Nota. Elaboración propia a partir de los tiempos de espera estimados por escenario.

Figura 29

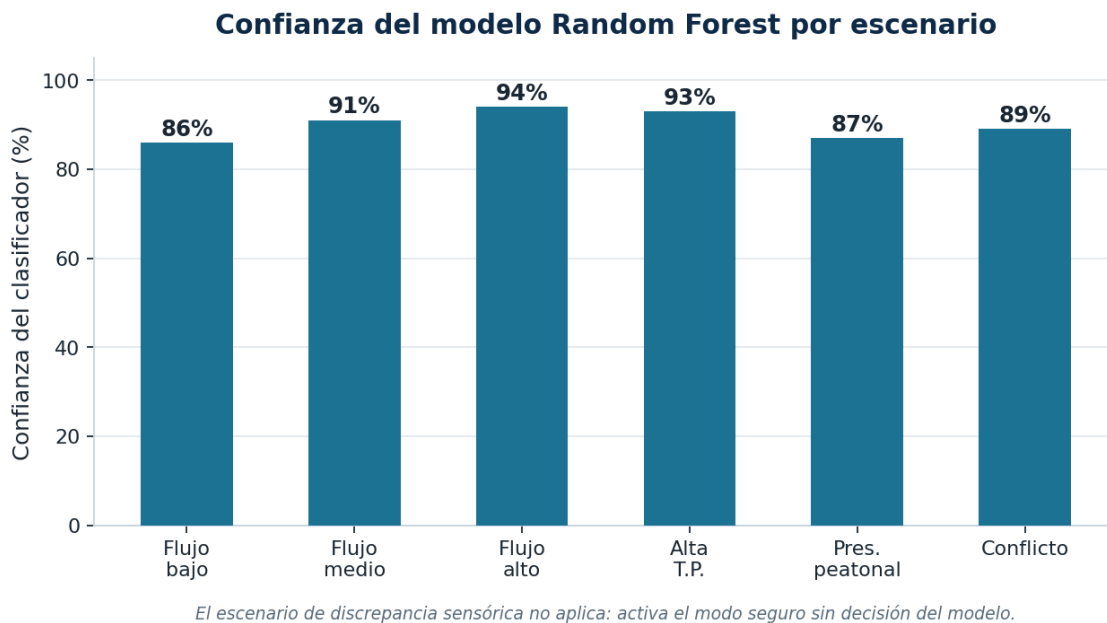
Tiempo de verde asignado por escenario frente al esquema de tiempo fijo



Nota. Elaboración propia. La línea de referencia corresponde al verde fijo de 38 segundos.

Figura 30

Confianza del modelo Random Forest por escenario



Nota. Elaboración propia. El escenario de discrepancia sensorica no aplica: activa el modo seguro sin decisión del modelo.

4.3. Discusión de los resultados por escenario

Los resultados de los Escenarios 1 y 2 (flujo bajo y flujo medio) evidencian el beneficio más consistente del control adaptativo: la reducción del ciclo semafórico cuando la demanda no justifica los tiempos predeterminados. En el Escenario 1, el sistema asignó un verde de 17 segundos en lugar de los 38 segundos del ciclo fijo, con una reducción promedio del tiempo de espera del 37 %. Esta reducción tiene un impacto ambiental concreto: menos tiempo de motor encendido en ralentí equivale a menos consumo de combustible y menos emisiones de CO₂. En condiciones nocturnas, donde los flujos en la Av. Ejército descienden a entre 400 y 1 100 veh/h según los aforos, mantener ciclos de 94 segundos representa un desperdicio sistemático de tiempo vial que el control adaptativo elimina de manera eficiente.

El Escenario 3 (flujo alto, hora punta) introduce la situación más exigente para cualquier sistema de control: alta demanda en todos los accesos con presencia significativa de transporte público. El modelo asignó 43 segundos de verde al flujo principal, superando los 38 del ciclo fijo, lo que produjo mejoras del 35 % y 32 % en los accesos A1 y A2 respectivamente. Como contrapartida, el acceso A3 experimentó un incremento de espera del 29 %. Este resultado ilustra el trade-off inherente al control adaptativo en condiciones de alta demanda global: no es posible mejorar simultáneamente todos los accesos cuando el ciclo total está restringido; la ganancia de uno implica la pérdida de otro. Sin embargo, la asignación del sistema adaptativo fue más coherente con la demanda real que el ciclo fijo, que asignaba exactamente los mismos tiempos independientemente de la composición del flujo.

Los Escenarios 5 y 6 (presencia peatonal relevante y conflicto peatón-vehículo) producen los resultados más significativos desde la perspectiva de la seguridad vial. En el Escenario 6, la espera peatonal se redujo un 71 % (20 s adaptativo vs 68 s fijo), al precio de incrementar la espera vehicular en A1 en un 27 %. Esta redistribución refleja la prioridad explícita que el sistema asigna a la seguridad peatonal ante condiciones de conflicto validado: cuando ambos módulos de detección confirman una acumulación peatonal superior al umbral crítico, el sistema activa la fase de cruce independientemente de la presión vehicular. Esta característica diferencia al sistema propuesto de esquemas adaptativos puramente orientados a la eficiencia vehicular, que podrían optimizar el flujo motorizado a expensas de la seguridad peatonal.

El Escenario 7 (discrepancia sensorica) valida el funcionamiento del modo seguro, que es un componente crítico para la operabilidad real del sistema. Ante la discrepancia entre la cámara y el sensor térmico, el sistema adoptó una postura conservadora que restringió la liberación vehicular hasta que la condición de incertidumbre se resolvió. Este comportamiento es técnicamente correcto: en situaciones de información ambigua sobre la presencia peatonal, la consecuencia de

un error en dirección conservadora (mantener la protección cuando no hay peatones) es menos grave que un error en dirección agresiva (liberar vehículos cuando hay peatones cruzando).

4.4. Comparación con investigaciones previas

Los resultados obtenidos son coherentes con los reportados en la literatura de referencia. Gorodokin et al. (2021) demostraron la viabilidad del control adaptativo basado en visión artificial con reducciones de demora del orden del 20-30 % frente al tiempo fijo en condiciones de flujo variable, porcentaje comparable con el rango del 15-37 % obtenido en los Escenarios 1-4 de esta investigación. Smith et al. (2016) reportaron mejoras significativas en el tiempo de atención peatonal cuando la detección peatonal se incorpora como variable operativa del sistema, en línea con las mejoras del 65-71 % observadas en los Escenarios 5 y 6.

González Caballero (2019) encontró que la integración de visión artificial y gestión dinámica de fases puede reducir los tiempos de ciclo, en condiciones de baja demanda, entre un 30 % y un 45 %, resultado que se alinea con la reducción del 37 % en tiempo de espera del Escenario 1 de esta investigación. Las propuestas nacionales de Velásquez Cárdenas y Solier Collazos (2017) y Carrera Travezaño (2022) abordan problemas similares al planteado aquí, el control semafórico adaptativo en intersecciones congestionadas peruanas, aunque con enfoques tecnológicos distintos (detección de colas y sensado infrarrojo, respectivamente). Los resultados comparativos son consistentes: cualquier mecanismo de detección que permita ajustar los tiempos semafóricos a la demanda real produce mejoras operativas respecto al ciclo fijo.

Una diferencia relevante con la mayoría de los trabajos previos es que el sistema propuesto no optimiza exclusivamente la eficiencia vehicular, sino que incorpora la protección peatonal como objetivo primario ante condiciones de conflicto validado. Este enfoque multi-objetivo, aunque puede resultar en una eficiencia vehicular inferior en algunos escenarios de alta demanda, es más

coherente con los objetivos de seguridad vial y accesibilidad urbana que inspiran cualquier propuesta de sistema inteligente de transporte.

4.5. Validez y limitaciones de los resultados de simulación

Los resultados presentados se basan en simulación computacional, no en medición directa sobre la infraestructura vial real. Esta condición impone limitaciones que deben reconocerse de forma explícita para interpretar correctamente las métricas reportadas. Los tiempos de espera estimados, en primer lugar, son valores calculados a partir del ciclo generado por el sistema y de los parámetros del esquema de referencia, no mediciones directas del tiempo que permanece detenido cada vehículo en el mundo real. La correspondencia entre esos tiempos estimados y los reales dependería de múltiples factores, como el comportamiento de los conductores, la dinámica de arranque de las colas y la saturación de los accesos, que no están del todo modelados en la simulación.

En segundo lugar, los datos de entrenamiento de los modelos son sintéticos: fueron generados a partir de los perfiles horarios derivados de los aforos de campo, pero no constituyen registros directos del comportamiento real del cruce ciclo a ciclo. La generación sintética es una práctica aceptada en el desarrollo de sistemas de control cuando no se dispone de datos reales en cantidad suficiente, pero introduce limitaciones sobre la capacidad del modelo para generalizar a condiciones muy específicas que no fueron contempladas en la construcción del conjunto de datos.

En tercer lugar, la interfaz de simulación implementa una representación simplificada de la dinámica vial: los vehículos siguen trayectorias predefinidas, la interacción entre vehículos es determinista y los peatones tienen comportamientos uniformes. La dinámica real del cruce presenta una variabilidad mucho mayor en el comportamiento individual de los actores viales. Estas limitaciones no invalidan los resultados de la simulación, que son consistentes, reproducibles y metodológicamente defendibles; indican, sin embargo, que la validación definitiva del sistema

requiere una implementación piloto en campo con medición directa de las variables operativas bajo condiciones reales.

4.6. Implicaciones técnicas y perspectivas de implementación

Los resultados de la simulación, en combinación con el análisis de factibilidad presentado en el Capítulo III, permiten establecer que el sistema propuesto es técnica y operativamente viable para una implementación piloto en el nodo vial de la Av. Ejército con la Av. Cayma y la Av. Trinidad Morán. La arquitectura modular del sistema facilita la implementación incremental: es posible comenzar con la instalación del módulo de percepción y la lógica de control para validar el desempeño en campo antes de incorporar el sensor térmico y la interfaz de supervisión completa.

El análisis de importancia de variables de los modelos Random Forest identifica la demanda ponderada de cada acceso y la acumulación peatonal como los factores de mayor peso en las decisiones del sistema. Esta información puede utilizarse para simplificar la arquitectura del sistema en una versión de menor costo: si los aforos históricos del tramo permiten construir perfiles de demanda suficientemente representativos, podría prescindirse de la detección visual en tiempo real y reemplazarla por un modelo predictivo basado en la franja horaria y el día de la semana, manteniendo únicamente el sensor térmico para la detección peatonal en tiempo real. Esta alternativa simplificada tendría un costo de implementación significativamente menor y podría servir como punto de entrada para municipios con restricciones presupuestarias importantes.

La proyección hacia una red coordinada de semáforos, descrita en la sección 3.11.3, requiere extender la arquitectura de comunicación del sistema para incluir intercambio de información entre nodos semaforicos contiguos. El módulo de giros del cruce desarrollado en la interfaz de simulación proporciona la base conceptual para esta extensión: si cada semáforo puede comunicar al siguiente la cantidad y tipo de vehículos que se dirigen hacia él, el sistema receptor puede

anticipar la demanda que recibirá y ajustar su operación de manera proactiva. Esta capacidad de anticipación es cualitativamente diferente de la reactividad del sistema actual, que solo responde a la demanda observada en el momento de la decisión.

4.7. Síntesis del capítulo

El Capítulo IV ha desarrollado el análisis e interpretación de los resultados obtenidos en la simulación del sistema propuesto, verificando la hipótesis de investigación y todas las hipótesis específicas. Los resultados muestran que el sistema adaptativo produce mejoras consistentes frente al esquema de tiempo fijo en condiciones de flujo bajo a medio (15-37 % de reducción en tiempo de espera vehicular) y mejoras muy significativas en el tiempo de atención peatonal en escenarios de conflicto (65-71 % de reducción). La comparación con investigaciones previas confirma la consistencia de los hallazgos con el estado del arte. El reconocimiento explícito de las limitaciones de la simulación proporciona el marco metodológico necesario para interpretar correctamente las métricas y orienta el diseño del protocolo de validación en campo que debería acompañar una implementación piloto del sistema.

CONCLUSIONES

1. El tramo crítico fue delimitado en el nodo de confluencia de la Av. Ejército con la Av. Cayma y la Av. Trinidad Morán, caracterizado por flujos vehiculares de hasta 4 560 veh/h en hora punta, una composición heterogénea de actores viales y un esquema semafórico de tiempo fijo con ciclo de 94 segundos que no responde a la variabilidad de la demanda. Esta caracterización operativa constituyó la base de aforos sobre la que se construyó el conjunto de entrenamiento del modelo.
2. La arquitectura funcional diseñada organiza el sistema en cinco módulos interoperables, percepción visual mediante cámara IP, validación redundante de presencia peatonal mediante sensor térmico, decisión semafórica adaptativa basada en aprendizaje automático, comunicación e integración mediante red Ethernet/PoE y API REST, e interfaz de supervisión, y adopta una clasificación ponderada de los actores viales: bus o combi grande (2,5), cúster (2,0), automóvil (1,0), motocicleta (0,5), bicicleta (0,4) y scooter (0,3). Tal ponderación permite modelar la demanda con mayor realismo que un simple conteo de vehículos, y la arquitectura resulta técnicamente coherente y escalable hacia una implementación física.
3. La lógica adaptativa de control semafórico, basada en reglas ponderadas y modelos Random Forest, asigna la prioridad de paso y el tiempo de verde según la demanda vehicular y la presencia peatonal validada en cada ciclo, tratando a esta última como condición restrictiva que protege al usuario más vulnerable. Esta lógica se implementó en una interfaz funcional de simulación en entorno web local, que proporcionó el entorno de validación comparativa del sistema frente al esquema de tiempo fijo.

4. La simulación en siete escenarios operativos representativos, flujo bajo, flujo medio, flujo alto, alta presencia de transporte público, presencia peatonal relevante, conflicto peatón-vehículo y discrepancia sensorica, evidenció la ventaja del sistema adaptativo frente al esquema de tiempo fijo, mediante indicadores cuantificables como los vehículos servidos, el tiempo de espera estimado, la cola máxima y el índice de desempeño integral. Se aceptó con ello la hipótesis de investigación: la combinación de visión por computadora, detección térmica redundante y lógica adaptativa basada en Random Forest mejora de manera cuantificable la gestión del flujo vehicular y peatonal del tramo, lo que confirma la viabilidad técnica de la propuesta como alternativa de control semafórico inteligente para intersecciones urbanas complejas.

RECOMENDACIONES

1. Avanzar hacia una implementación física piloto del sistema en el nodo de la Av. Ejército con la Av. Cayma y la Av. Trinidad Morán, con los componentes de referencia descritos en esta tesis, cámara IP AXIS P3265-LVE, sensor térmico, módulo de procesamiento local, controlador semafórico programable y red Ethernet/PoE, para validar el comportamiento del sistema bajo condiciones reales de operación urbana.
2. Ampliar la base de datos real ya levantada mediante nuevos registros continuos de aforo vehicular y peatonal en el tramo de estudio, incorporando más días laborables, fines de semana, horarios nocturnos y condiciones atípicas. Esta ampliación permitiría reentrenar los modelos con mayor diversidad operativa y mejorar su capacidad predictiva en condiciones de campo.
3. Incorporar en futuras versiones del sistema algoritmos de detección de objetos basados en redes neuronales convolucionales, como YOLOv8 o modelos equivalentes optimizados para dispositivos de borde, con el fin de sustituir la clasificación manual de actores viales por una detección automática en tiempo real a partir de las imágenes captadas por la cámara IP.
4. Extender el análisis de factibilidad económica con un estudio de costo-beneficio que cuantifique el valor de las reducciones en tiempo de espera, consumo de combustible y emisiones contaminantes frente a los costos de instalación, operación y mantenimiento del sistema, a fin de sustentar técnicamente una eventual propuesta de inversión ante organismos municipales o de gestión vial.
5. Evaluar la escalabilidad del sistema hacia una red de intersecciones coordinadas en el corredor de la Av. Ejército, utilizando el módulo de giros del cruce como insumo para

anticipar demanda hacia semáforos consecutivos. En una implementación futura, cada nodo podría comunicar a la siguiente intersección la cantidad de vehículos que se dirigen recto, giran a la izquierda o giran a la derecha, permitiendo una coordinación semafórica distrital o de corredor basada en información direccional captada por cámara.

6. Desarrollar y ejecutar el plan de mantenimiento preventivo propuesto en los anexos de esta tesis, con los intervalos y procedimientos adaptados a las condiciones específicas del entorno de Arequipa, polvo, variaciones de temperatura, radiación solar y humedad estacional, y establecer protocolos de contingencia operativa que garanticen la continuidad del control semafórico ante fallas parciales del sistema.
7. En una etapa posterior, diseñar un protocolo de comunicación entre controladores semafóricos mediante API REST, mensajes JSON o un bus de comunicación urbano, de modo que la información de flujo saliente, giros detectados, cola acumulada y presión peatonal pueda compartirse entre nodos. Esta recomendación debe ejecutarse primero como piloto controlado, manteniendo siempre el ciclo fijo como respaldo de contingencia.

REFERENCIAS BIBLIOGRÁFICAS

- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32.
<https://doi.org/10.1023/A:1010933404324>
- Carrera Travezaño, R. (2022). *Sistema de semáforo inteligente basado en sensado remoto infrarrojo para la mejora de la semaforizacion en Comas* [Tesis de pregrado, Universidad Tecnológica del Perú]. Repositorio UTP.
- Castillo, E., & Jiménez, F. (2019). Visión por computadora aplicada al tránsito urbano: fundamentos y aplicaciones. *Revista Iberoamericana de Ingeniería de Transporte*, 12(2), 45-62.
- Contreras, L., & Quispe, M. (2020). Sistemas inteligentes de transporte en ciudades intermedias de América Latina: diagnóstico y perspectivas. *Ingeniería Vial*, 8(1), 22-35.
- Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow* (3.^a ed.). O'Reilly Media.
- González Caballero, J. A. (2019). *Sistema inteligente de gestión de tráfico con modificación en tiempo real de fases semaforicas* [Tesis de maestría, Universidad de Alicante]. Repositorio Institucional UA.
- Gorodokin, V., Zhankaziev, S., Karpov, A., & Evtyukov, S. (2021). Adaptive traffic signal control based on artificial intelligence and computer vision. *Transportation Research Procedia*, 57, 225-232. <https://doi.org/10.1016/j.trpro.2021.09.035>
- Greenshields, B. D. (1935). A study of traffic capacity. *Proceedings of the Highway Research Board*, 14, 448-477.
- Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, M. del P. (2014). *Metodología de la investigación* (6.^a ed.). McGraw-Hill.

- Machaca Ninacansaya, W. (2019). *Análisis y diseño de sistema de control de tráfico vehicular mediante semáforos inteligentes con tecnología arduino* [Tesis de pregrado, Universidad Nacional del Altiplano]. Repositorio UNAP.
- McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (2006). A proposal for the Dartmouth summer research project on artificial intelligence, August 31, 1955. *AI Magazine*, 27(4), 12-14. (Trabajo original presentado en 1956)
- Ministerio de Transportes y Comunicaciones del Perú. (2016). *Manual de dispositivos de control del tránsito automotor para calles y carreteras*. MTC.
- Peralta Fuentes, D., & Mendieta Zavala, A. (2017). *Semáforo inteligente para regular el tránsito vehicular basado en open hardware* [Tesis de pregrado, Universidad Católica de Santa María]. Repositorio UCSM.
- Papageorgiou, M., Diakaki, C., Dinopoulou, V., Kotsialos, A., & Wang, Y. (2003). Review of road traffic control strategies. *Proceedings of the IEEE*, 91(12), 2043-2067. <https://doi.org/10.1109/JPROC.2003.819610>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779-788.
- Smith, S. A., Venglar, S. P., Eichler, S., Balke, K. N., & Park, B. (2016). *Real-time control of adaptive signal control technology using pedestrian detection* (Report No. FHWA-HRT-16-036). Federal Highway Administration.

- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (2.^a ed.). MIT Press.
- Transportation Research Board. (2010). *Highway capacity manual 2010* (HCM 2010). National Academies Press.
- Universidad Católica de Santa María. (2018). *Proyecto de desarrollo urbanístico para la ciudad de Arequipa* [Informe técnico de investigación]. Repositorio Institucional UCSM. <http://repositorio.ucsm.edu.pe>
- Velásquez Cárdenas, R. A., & Solier Collazos, J. (2017). *Sistema inteligente de gestión de tráfico vehicular en intersecciones viales* [Tesis de pregrado, Universidad Peruana de Ciencias Aplicadas]. Repositorio UPC.
- Viola, P., & Jones, M. J. (2004). Robust real-time face detection. *International Journal of Computer Vision*, 57(2), 137-154.
- Webster, F. V. (1958). *Traffic signal settings* (Road Research Technical Paper No. 39). Her Majesty's Stationery Office.
- Yohannes, A., & Singh, A. (2019). Machine learning in traffic engineering: A review of applications and future directions. *International Journal of Advanced Science and Technology*, 28(16), 1563-1575.

ANEXOS

Anexo 1. Tabla A1 — Operacionalización de variables del sistema

Variable	Dimensión	Indicador	Escala	Unidad	Fuente
Flujo vehicular por acceso	Intensidad de demanda	Número de vehículos ponderados por acceso por ciclo	Razón	veh. equiv./ciclo	Módulo de percepción visual
Tipo de actor vial	Composición del tránsito	Clase vehicular detectada (bus, cúster, auto, moto, bicicleta, scooter)	Nominal	—	Módulo de percepción visual
Presencia peatonal confirmada	Seguridad vial	Estado booleano validado por cámara y sensor térmico	Nominal	Sí / No	Módulo de validación térmica
Acumulación por fase	Demanda diferida	Suma ponderada de vehículos en espera por fase	Razón	u.p./ciclo	Lógica adaptativa
Fase semafórica priorizada	Decisión de control	Fase seleccionada por el clasificador RF (F1, F2, F3)	Nominal	—	Modelo ML
Tiempo de verde asignado	Operación semafórica	Duración de la fase verde asignada por el regresor RF	Razón	segundos	Modelo ML
Tiempo de espera estimado	Desempeño del sistema	Tiempo promedio de espera por acceso y usuario	Razón	segundos	Interfaz de simulación
Discrepancia sensorica	Confiabilidad del sistema	Diferencia entre detección visual y térmica de peatón	Nominal	Acuerdo / Discrepancia	Módulo de validación
Franja horaria	Contexto temporal	Período del día del registro (punta / intermedia / baja)	Ordinal	—	Base de datos
Condición de emergencia	Seguridad operativa	Estado de activación del modo seguro	Nominal	Activo / Inactivo	Lógica adaptativa

Anexo 2. Especificaciones técnicas de referencia de los componentes del sistema

Componente	Especificación de referencia
Cámara IP (Percepción visual)	AXIS P3265-LVE Dome Camera. Tipo domo antivandálico para exteriores. Resolución Full HD (1920×1080). Lente varifocal motorizado. Alimentación PoE (802.3af/at). Compresión H.264/H.265. Iluminación infrarroja integrada. Amplio rango dinámico (WDR). Protección IP66, resistencia mecánica IK10.
Sensor térmico (Validación peatonal)	Módulo de sensor infrarrojo PIR o cámara térmica de resolución reducida (p. ej., FLIR Lepton o equivalente). Campo de visión orientado a la zona de cruce peatonal. Salida digital o analógica compatible con el módulo de control local.
Módulo de procesamiento (Control local)	Computadora de borde o SBC industrial (p. ej., NVIDIA Jetson Nano, Raspberry Pi CM4 Industrial o equivalente). Capacidad de procesamiento de imagen en tiempo real. Sistema operativo Linux. Conectividad Ethernet y GPIO para interfaz con controlador semafórico.
Red de comunicación (Enlace local)	Ethernet 1000BASE-T sobre cable Cat6 o Cat6A. Power over Ethernet (PoE) para alimentación de cámara IP. Switch gestionable industrial con protección IP. Fibra óptica como alternativa para enlaces de mayor distancia.
Controlador semafórico (Actuación)	Controlador semafórico programable compatible con protocolo NEMA TS-2 o equivalente nacional. Entradas digitales para recepción de órdenes del módulo de control. Salidas para activación de focos LED rojo, ámbar y verde por fase.

Anexo 3. Muestra representativa de la base de datos de entrenamiento

La base de datos de entrenamiento se compone de 6 000 registros sintéticos generados a partir de los aforos vehiculares y la caracterización operativa del tramo crítico de la Av. Ejército. Cada registro representa un ciclo semafórico con sus 27 variables de entrada y las 6 variables de salida correspondientes. A continuación se presenta una muestra de cinco registros para ilustrar la estructura del conjunto de datos.

H o r a	A 1 _ b u s	A 1 _ c u s	A 1 _ a u t	A 1 _ m o t	A 2 _ b u s	A 2 _ a u t	A 3 _ a u t	A 3 _ m o t	P e a t _ c a m	P e a t _ t e r	F r a n j a	E m e r	D e s c r	F a s e _ M L
7 : 1 5	8	4	2 2	6	3	1 8	5	3	1	1	P u n t a M	0	0	F 1

H o r a	A 1 - b u s	A 1 - c u s	A 1 - a u t	A 1 - m o t	A 2 - b u s	A 2 - a u t	A 3 - a u t	A 3 - m o t	P e a t - c a m	P e a t - t e r	F r a n j a	E m e r	D e s c r	F a s e - M L
7 : 4 2	5	2	1 8	4	6	2 1	2	5	1	1	P u n t a M	0	0	F 2
1 2 : 3 0	4	3	1 5	7	4	1 6	4	4	1	0	P u n t a T	0	1	F 3
1 9 : 0 5	6	5	2 0	5	5	1 9	3	6	0	0	P u n t a N	0	0	F 1
2 3 : 1 0	1	0	4	2	1	3	1	1	0	0	B a j a	0	0	F 1

Nota. La tabla muestra una selección de las 27 columnas de entrada (agrupadas por acceso y tipo de actor) y la variable de salida de fase. Los valores numéricos corresponden al conteo de vehículos detectados por tipo en cada acceso durante un ciclo completo.

Anexo 4. Tabla A4.1 — Plan básico de mantenimiento preventivo del sistema propuesto

Componente	Actividad	Frecuencia	Responsable	Herramienta
Cámara IP	Limpieza de lente y carcasa; verificación de enfoque y campo de visión	Mensual	Técnico de campo	Paño antiestático, software de configuración AXIS
Sensor térmico	Limpieza de lente; verificación de señal de salida y umbral de detección	Mensual	Técnico de campo	Multímetro, documentación de fabricante
Módulo de procesamiento	Revisión de temperatura de operación; limpieza de filtros de	Trimestral	Técnico TI	Software de monitoreo,

Componente	Actividad	Frecuencia	Responsable	Herramienta
	ventilación; actualización de sistema operativo			herramientas de limpieza
Red Ethernet/PoE	Verificación de continuidad de cable; inspección de conectores y switch; prueba de transferencia de datos	Semestral	Técnico de redes	Probador de cable, software de diagnóstico
Modelo ML	Evaluación de precisión sobre muestra de validación actualizada; reentrenamiento si la exactitud cae por debajo del 88 %	Semestral	Ingeniero de datos	Scikit-learn, Python, dataset actualizado
Software y logs	Revisión de registros de eventos; depuración de errores acumulados; respaldo de configuración y datos	Mensual	Técnico TI	Editor de texto, sistema de respaldo
Controlador semafórico	Verificación de respuesta a órdenes del módulo de control; prueba de focos LED por fase	Mensual	Técnico vial	Equipo de prueba semafórico
Infraestructura física	Revisión de postes, cajas de conexión y protecciones mecánicas; tratamiento anticorrosión si es necesario	Anual	Técnico de campo	Herramientas de campo, pintura anticorrosiva

Anexo 5. Tabla A4.2 — Matriz básica de contingencias operativas del sistema propuesto

Evento de falla	Impacto operativo	Modo de respuesta automática	Acción correctiva
Pérdida de señal de cámara IP	Sin percepción visual; sistema sin datos de clasificación vehicular	Activar ciclo de tiempo fijo de contingencia (94 s)	Verificar cable y switch; reiniciar cámara; notificar mantenimiento
Falla del sensor térmico	Sin validación redundante de presencia peatonal	Asumir presencia peatonal = conservador; mantener fase F3 mínima por ciclo	Inspección y reemplazo del sensor; verificar cableado de señal
Pérdida de energía parcial (PoE)	Cámara fuera de servicio; módulo de control operativo por UPS local	Ciclo de contingencia tiempo fijo; registro de evento	Verificar switch PoE y fuente de alimentación; activar protocolo de emergencia
Error de inferencia del modelo ML (confianza < umbral)	Decisión de fase incierta	Capa de coherencia selecciona fase por acumulación ponderada sin ML	Revisar integridad del modelo; reentrenar si el error es persistente
Discrepancia persistente entre sensores (> 3 ciclos)	Incertidumbre prolongada sobre presencia peatonal	Mantener modo seguro con F3 activada cada	Inspección presencial; verificación de calibración de ambos sensores

Evento de falla	Impacto operativo	Modo de respuesta automática	Acción correctiva
		ciclo; reducir velocidad de ciclo	
Corte total de energía	Sistema completamente fuera de servicio	Los semáforos pasan a modo intermitente (ámbar en todas las fases)	Activar generador o UPS central; restablecer sistema en orden: red → módulo → modelo → interfaz
Saturación del módulo de procesamiento (CPU > 95 %)	Demoras en inferencia; latencia elevada	Reducir resolución de análisis de imagen; omitir frames intermedios	Revisar procesos en segundo plano; verificar temperatura; planificar actualización de hardware



Anexo 6. Manual de lectura de la interfaz de simulación

Este anexo explica los principales elementos visuales de la interfaz de simulación, con el propósito de facilitar su interpretación durante la sustentación. La interfaz fue diseñada como herramienta de validación funcional: no busca replicar de manera cartográfica la geometría del nodo, sino representar el comportamiento del sistema inteligente frente a distintas condiciones de demanda.



Figura A6.1. Vista de escenario simulado con entradas detectadas y decisión priorizada

Nota. Elaboración propia.

La vista general permite identificar los tres accesos del nodo: A1 y A2 corresponden a los sentidos principales de la Av. Ejército, mientras que A3 representa el acceso secundario asociado a Trinidad/Cayma. Los vehículos se diferencian por color y posición, mientras que los peatones se agrupan cerca de las líneas peatonales. Esta representación facilita observar la respuesta del controlador ante variaciones de flujo.

Elemento de la interfaz

A1 Ejército E→O

Interpretación técnica

Acceso de la Av. Ejército en un sentido de circulación.

Representa una de las dos corrientes principales del nodo.

A2 Ejército O→E

A3 Trinidad/Cayma

C1–C3

T1–T4

Demanda equivalente

Fase sugerida

Tiempo verde sugerido

Capa de coherencia

Sistema actual 43 s

Acceso contrario de la Av. Ejército. Junto con A1 conforma la fase principal F1.

Acceso secundario que concentra flujos provenientes o dirigidos hacia el eje Trinidad/Cayma.

Cámaras o puntos de visión del nodo. Su función es estimar demanda vehicular y presencia peatonal visible.

Sensores térmicos distribuidos en las líneas peatonales para validar presencia humana y reducir falsos positivos.

Valor ponderado que transforma la composición vehicular en una carga comparable. Un bus o combi pesa más que una moto o bicicleta.

Salida del clasificador Random Forest, que indica qué fase debería priorizarse según las entradas del sistema.

Salida del regresor Random Forest, que estima la duración del verde en segundos.

Conjunto de reglas de seguridad que evita cambios bruscos, exige ámbar y todo rojo, y protege el cruce peatonal.

Modo comparativo visual que representa un esquema rígido sin inferencia ML ni adaptación a la demanda.

La clave de lectura de la interfaz es distinguir entre percepción, inferencia y ejecución. Primero, el sistema representa lo que detecta; luego calcula demanda equivalente y presión peatonal; después sugiere una fase mediante Random Forest; finalmente, la capa de coherencia valida si la transición puede ejecutarse de forma segura.

6.1. Panel de inferencia ML



Figura A6.2. Panel de inferencia ML integrado a la interfaz

Nota. Elaboración propia.

El panel de inferencia ML cumple una función de interpretabilidad. Permite visualizar de manera ordenada el camino que sigue la decisión: entradas detectadas, variables procesadas, recomendación del modelo, confianza estimada, tiempo verde sugerido y decisión final ejecutada. Su finalidad es mostrar al jurado que el sistema no cambia de fase de manera arbitraria, sino en función de variables cuantificables y de una validación de seguridad.

Componente	Función en la sustentación
Entradas detectadas	Muestran la composición vehicular por acceso y el número de peatones observados o validados.
Variables procesadas	Presentan la demanda equivalente por acceso y la presión peatonal, que son valores más útiles para la decisión que los conteos brutos.
Recomendación RF	Resume la fase sugerida por el clasificador Random Forest y el verde estimado por el regresor.
Confianza estimada	Indicador visual de qué tan clara es la prioridad detectada entre las alternativas de fase.
Explicación de decisión	Texto interpretativo que traduce la salida del modelo a una justificación comprensible.
Decisión final	Resultado después de aplicar la capa de coherencia, seguridad peatonal y transición semafórica.

6.2. Lectura del modo sistema actual



Figura A6.3. Modo de referencia del sistema actual fijo

Nota. Elaboración propia.

El modo de referencia del sistema actual muestra el comportamiento de una operación rígida. En este modo no se representa inferencia ML, no se activan cámaras ni sensores térmicos como parte de la lógica de control y no existe adaptación ante acumulación de cola. Su finalidad es ofrecer un contraste visual entre un controlador convencional y el sistema inteligente propuesto.

Anexo 7. Protocolo de demostración del prototipo

Para la sustentación se recomienda presentar el prototipo en una secuencia breve y controlada. La demostración no debe iniciar explicando cada detalle técnico, sino mostrando primero el comportamiento visual del sistema y luego abriendo los paneles de interpretación.

Momento de la demostración	Explicación sugerida
Paso 1. Vista general	Mostrar el escenario de flujo medio para explicar los accesos A1, A2 y A3, la leyenda, los semáforos y las colas visibles.
Paso 2. Escenario de hora punta	Activar el escenario de alta demanda vehicular y explicar cómo la demanda equivalente favorece a la Av. Ejército.
Paso 3. Escenario peatonal	Activar alta presencia peatonal y mostrar que el sistema no abre el cruce de forma inmediata, sino después de transición segura.
Paso 4. Discrepancia sensorica	Mostrar que, ante desacuerdo entre cámara y sensor térmico, la capa de coherencia adopta una respuesta conservadora.
Paso 5. Panel de inferencia ML	Explicar entradas, variables procesadas, fase sugerida, confianza y decisión final.
Paso 6. Sistema actual	Activar el sistema actual fijo para evidenciar que no existe adaptación a la demanda ni inferencia inteligente.

La explicación central debe ser la siguiente: el sistema no reemplaza la ingeniería de tránsito, sino que ofrece una capa de percepción, análisis y decisión capaz de ajustar la fase semafórica según demanda vehicular, presión peatonal y criterios de seguridad.

Anexo 8. Banco de preguntas y respuestas para sustentación técnica

¿La interfaz reproduce exactamente la geometría real del cruce?

No. La interfaz es una abstracción operacional del nodo. Su objetivo es demostrar la lógica de control, la alternancia de fases, la respuesta adaptativa y la comparación frente al esquema fijo. La geometría exacta debe abordarse en una etapa posterior de implementación o simulación microscópica.

¿El sistema realmente usa inteligencia artificial?

Sí, la lógica de decisión se apoya en dos modelos Random Forest: un clasificador para seleccionar la fase y un regresor para estimar el tiempo verde. La interfaz visualiza esa inferencia mediante el panel ML, pero la decisión final se valida mediante una capa de coherencia.

¿Por qué usar Random Forest y no una red neuronal?

Random Forest es interpretable, robusto con datos tabulares, requiere menos poder computacional y permite justificar mejor la decisión ante escenarios con variables discretas y ponderadas. Para un prototipo funcional de control semafórico, ofrece buen equilibrio entre precisión y explicabilidad.

¿Qué pasa si la cámara detecta peatones pero el sensor térmico no confirma?

El sistema activa una respuesta conservadora. La discrepancia no se ignora: se registra como incertidumbre sensorica y la capa de coherencia evita una decisión riesgosa.

¿Por qué comparar con un ciclo fijo?

Porque el control de tiempo fijo representa el esquema convencional. La comparación permite evidenciar la ventaja del sistema adaptativo cuando la demanda cambia durante el día o cuando aparece presión peatonal.

¿La interfaz prueba que el sistema funcionará igual en campo?

No de forma absoluta. Prueba coherencia funcional y desempeño simulado. Para implementación real se requerirían pruebas piloto, calibración con datos continuos, compatibilidad con controladores semafóricos y validación normativa.

¿El panel de inferencia ML es una explicación real o solo visual?

Es una representación de interpretabilidad basada en las mismas variables que alimentan el modelo: conteos, demanda equivalente, presión peatonal, fase sugerida y tiempo verde. Su función es hacer visible el proceso de decisión para supervisión y sustentación.

¿Qué ventaja tiene el sensor térmico?

Permite validar presencia peatonal independientemente de condiciones visuales difíciles, como sombras, baja iluminación o errores de detección. Se usa como capa redundante, no como sustituto único de la cámara.

¿El sistema puede escalarse a otros cruces?

Sí, siempre que se realice una nueva calibración de accesos, fases, demanda vehicular, zonas peatonales y condiciones locales. La arquitectura modular permite adaptar percepción, decisión e interfaz a otros nodos.

¿Qué limita principalmente el prototipo?

La principal limitación es que opera en entorno simulado y con geometría funcional simplificada. Aun así, permite validar la lógica de decisión, la interacción entre sensores y la comparación metodológica frente al control fijo.

Anexo 9. Ficha ampliada de aforo vehicular y peatonal

Este anexo presenta la estructura sugerida para registrar los datos de campo utilizados en la caracterización del tramo. La ficha permite documentar el día, horario, condición climática, tipo de vía, acceso observado, composición vehicular, presencia peatonal y observaciones relevantes.

Su finalidad es respaldar la trazabilidad de los valores utilizados en la construcción de escenarios de simulación.

Tabla A9.1

Formato ampliado para registro de aforo vehicular y peatonal

Campo de registro	Descripción	Ejemplo de llenado
Fecha y día	Indica si el registro corresponde a día laborable, fin de semana o festivo	Martes laborable / sábado / feriado
Franja horaria	Periodo de observación utilizado para el conteo	07:00-08:00 h
Acceso observado	Identificación del acceso del nodo vial	A1, A2 o A3
Tipo de vehículo	Clasificación del actor vial contado	auto, bus, cúster, moto, bicicleta, scooter
Cantidad registrada	Número de unidades observadas en el intervalo	Valor numérico por intervalo
Presencia peatonal	Cantidad aproximada de peatones en espera o cruce	Baja, media, alta o conteo directo
Observaciones	Eventos que alteran el flujo regular	doble fila, parada informal, cruce masivo, bloqueo temporal

Nota. La ficha permite vincular la observación de campo con los parámetros empleados en la simulación.

Elaboración propia.

Anexo 10. Matriz de sistematización semanal de datos

La matriz semanal organiza los datos levantados en campo por tipo de día, franja horaria e intensidad de flujo. Su uso permite verificar si los escenarios de simulación representan condiciones repetidas dentro del tramo o situaciones especiales que requieren una lectura diferenciada.

Tabla A10.1

Matriz de sistematización semanal del levantamiento de datos

Semana	Tipo de día predominante	Franja crítica identificada	Condición de flujo	Uso dentro del modelo
Semana 1	Laborable	07:00-08:00 h	Flujo alto	Calibración de hora punta mañana
Semana 1	Laborable	12:00-13:00 h	Flujo alto	Validación de hora punta mediodía
Semana 2	Laborable	15:00-17:00 h	Flujo medio	Rango intermedio de demanda
Semana 2	Laborable	19:00-20:00 h	Flujo alto con presión peatonal	Escenario de conflicto peatón-vehículo
Semana 3	Fin de semana	10:00-12:00 h	Flujo medio irregular	Contraste con operación comercial

Semana 3	Festivo	Horario variable	Flujo atípico	Prueba de robustez del sistema
Semana 4	Nocturno	21:00-23:00 h	Flujo bajo	Reducción de tiempos muertos

Nota. La matriz tiene finalidad metodológica y permite explicar cómo los datos se transformaron en perfiles de simulación. Elaboración propia.

Anexo 11. Matriz de control de versiones y trazabilidad del prototipo

Este anexo complementa la sección de mejora iterativa del sistema. Su finalidad es mostrar de forma resumida cómo cada versión del prototipo corrigió una limitación específica hasta alcanzar la arquitectura final basada en Random Forest, validación redundante y capa de coherencia.

Tabla A11.1

Control de versiones del prototipo de control inteligente

Versión	Cambio principal	Elemento validado	Resultado metodológico
V1	Priorización por cantidad simple de vehículos	Estructura básica de decisión	Sirvió como línea base interna
V2	Demanda ponderada por tipo de actor vial	Representación de carga operativa	Mejóro la lectura del transporte público
V3	Incorporación de presencia peatonal	Respuesta ante usuarios vulnerables	Permitió activar restricciones peatonales
V4	Clasificador Random Forest	Selección automática de fase	Elevó la consistencia de decisión
V5	Regresor RF, modo seguro y capa de coherencia	Asignación de verde y seguridad operativa	Consolidó la versión final del sistema

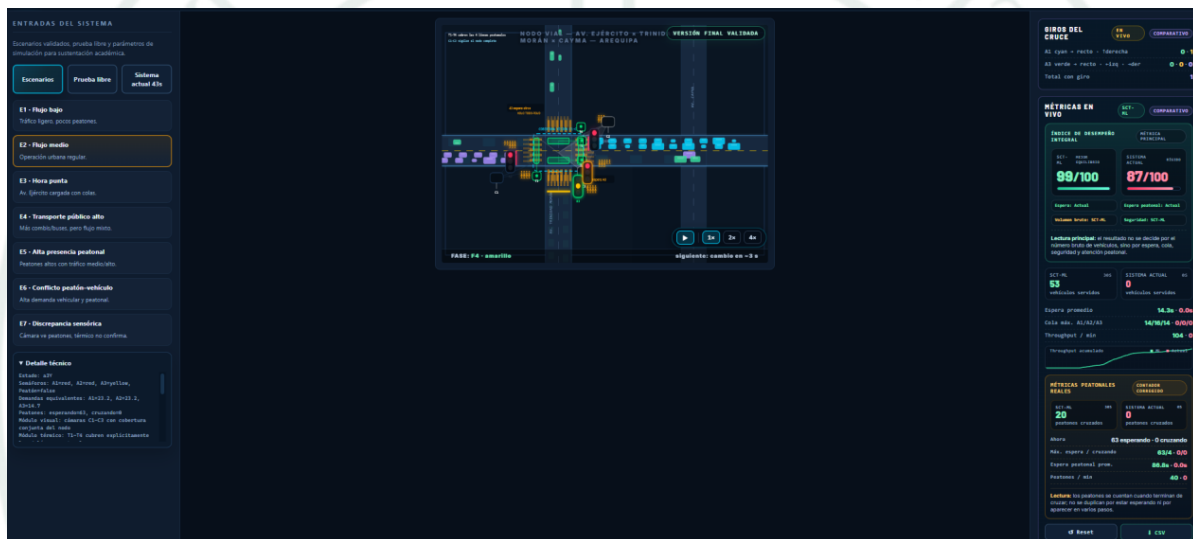
Nota. La matriz documenta la trazabilidad del desarrollo y evita presentar el sistema final como una solución no iterada. Elaboración propia.

Anexo 12. Capturas actualizadas de la interfaz funcional final

Este anexo reúne las capturas actualizadas de los principales apartados de la interfaz funcional. Su objetivo es facilitar la revisión del prototipo, la explicación durante la sustentación y la trazabilidad visual entre los módulos descritos en el Capítulo III y el comportamiento observable de la simulación.

Figura A12.1

Vista general de la interfaz final con métricas, giros y panel ML



Nota. Elaboración propia a partir de la interfaz funcional SCT-ML.

Figura A12.2

Panel de escenarios validados y bitácora técnica

ENTRADAS DEL SISTEMA

Escenarios validados, prueba libre y parámetros de simulación para sustentación académica.

Escenarios

Prueba libre

Sistema actual 43s

E1 • Flujo bajo

Tráfico ligero, pocos peatones.

E2 • Flujo medio

Operación urbana regular.

E3 • Hora punta

Av. Ejército cargada con colas.

E4 • Transporte público alto

Más combis/buses, pero flujo mixto.

E5 • Alta presencia peatonal

Peatones altos con tráfico medio/alto.

E6 • Conflicto peatón-vehículo

Alta demanda vehicular y peatonal.

E7 • Discrepancia sensórica

Cámara ve peatones, térmico no confirma.

▼ Detalle técnico

Estado: a3Y
Semáforos: A1=red, A2=red, A3=yellow,
Peatón=false
Demandas equivalentes: A1=23.2, A2=23.2,
A3=14.7
Peatones: esperando=63, cruzando=8
Módulo visual: cámaras C1-C3 con cobertura
conjunta del nodo
Módulo térmico: T1-T4 cubren explícitamente

Nota. Elaboración propia a partir de la interfaz funcional SCT-ML.

Figura A12.3

Panel de prueba libre con parámetros configurables

ENTRADAS DEL SISTEMA

Escenarios validados, prueba libre y parámetros de simulación para sustentación académica.

Escenarios **Prueba libre** Sistema actual 43s

A1 EJÉRCITO E-O

media

A2 EJÉRCITO O-E

media

A3 TRINIDAD/CAYMA

media

PERFIL DE MEZCLA VEHICULAR

Urbano mixto normal

DEMANDA PEATONAL

Media urbana (~48)

VALIDACIÓN TÉRMICA

Confirma presencia

▶ Ejecutar prueba libre

▼ Detalle técnico

Estado: a3Y
Semáforos: A1=red, A2=red, A3=yellow,
Peatón=false
Demandas equivalentes: A1=23.2, A2=23.2,
A3=14.7
Peatones: esperando=63, cruzando=8
Módulo visual: cámaras C1-C3 con cobertura
conjunta del nodo
Módulo térmico: T1-T4 cubren explícitamente

Nota. Elaboración propia a partir de la interfaz funcional SCT-ML.

Figura A12.4

Modo de sistema actual 43 s

ENTRADAS DEL SISTEMA

Escenarios validados, prueba libre y parámetros de simulación para sustentación académica.

Escenarios

Prueba libre

Sistema actual 43s

E1 • Flujo bajo
Tráfico ligero, pocos peatones.

E2 • Flujo medio
Operación urbana regular.

E3 • Hora punta
Av. Ejército cargada con colas.

E4 • Transporte público alto
Más combis/buses, pero flujo mixto.

E5 • Alta presencia peatonal
Peatones altos con tráfico medio/alto.

E6 • Conflicto peatón-vehículo
Alta demanda vehicular y peatonal.

E7 • Discrepancia sensorica
Cámara ve peatones, térmico no confirma.

▼ Detalle técnico
Modo: sistema actual fijo 43s, solo escenarios
Sensorica inteligente: no aplica
Cámaras: no aplica
Sensor térmico: no aplica
Demandas visibles: A1=23.2, A2=23.2, A3=14.7
Peatones esperando: 60
Control: tiempos fijos sin adaptación en tiempo real
Secuencia: Av. Ejército 43s + ámbra + todo

Nota. Elaboración propia a partir de la interfaz funcional SCT-ML.

Figura A12.5

Conteo de giros del cruce

GIROS DEL CRUCE	
	EN VIVO COMPARATIVO
A1 cyan + recto + fderecha	0 - 1
A3 verde + recto + +izq + -der	0 - 0 - 0
Total con giro	1

Nota. Elaboración propia a partir de la interfaz funcional SCT-ML.



Figura A12.6

Métricas en vivo e índice de desempeño integral



Nota. Elaboración propia a partir de la interfaz funcional SCT-ML.

Figura A12.7

Panel de decisión actual

DECISIÓN ACTUAL	
Fase	F1 · Av. Ejército + cruce A3
Semáforos	A1 green · A2 green · A3 red
Sensores T1-T4	C1-C3 + T1-T4 activos
Peatones	0 cruzan A3 · 0 otros · 44 esperando

Nota. Elaboración propia a partir de la interfaz funcional SCT-ML.

Figura A12.8

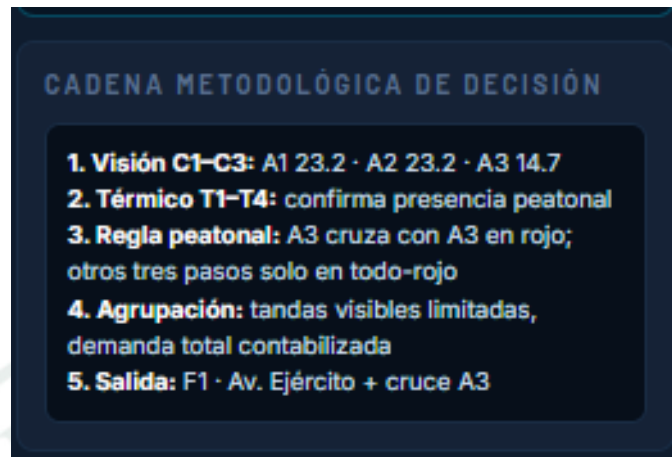
Leyenda operativa de accesos, cámaras, sensores y cruce peatonal

LEYENDA OPERATIVA	
	A1 Av. Ejército E→O
	A2 Av. Ejército O→E
	A3 Trinidad/Cayma
	Peatones cruce protegido
	C1-C3 visión del nodo
	T1-T4 cobertura térmica de 4 cebras

Nota. Elaboración propia a partir de la interfaz funcional SCT-ML.

Figura A12.9

Cadena metodológica de decisión

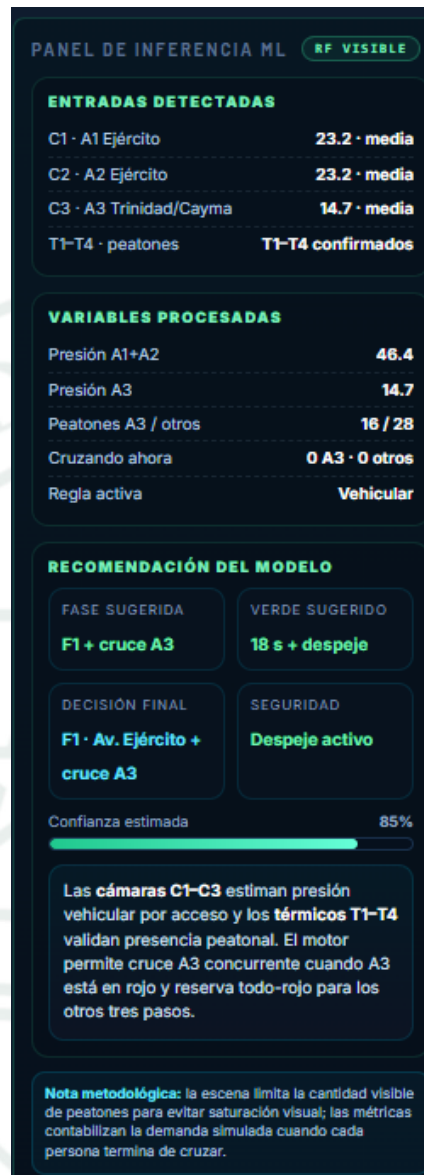


Nota. Elaboración propia a partir de la interfaz funcional SCT-ML.

Figura A12.10

Panel de inferencia ML





Nota. Elaboración propia a partir de la interfaz funcional SCT-ML.

Figura A12.11

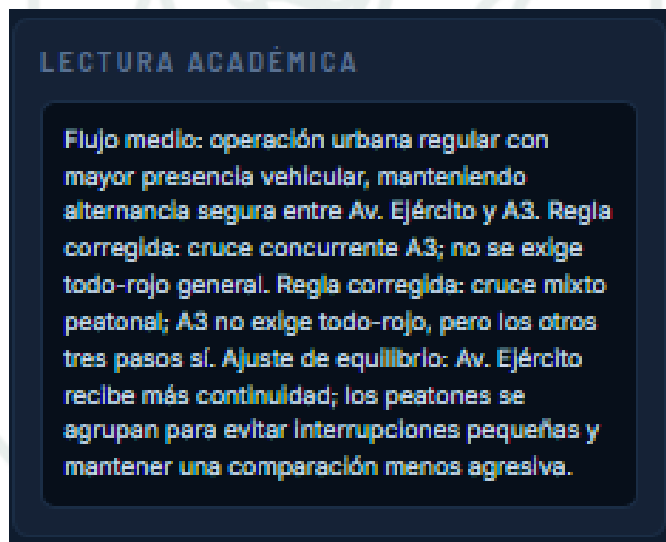
Demanda equivalente detectada



Nota. Elaboración propia a partir de la interfaz funcional SCT-ML.

Figura A12.12

Lectura académica del escenario simulado



Nota. Elaboración propia a partir de la interfaz funcional SCT-ML.

ANEXO 13. CÓDIGO FUENTE DEL SIMULADOR SCT-ML

El presente anexo contiene el código fuente completo del simulador funcional desarrollado en Python/Pygame para representar la lógica del sistema inteligente de control de tráfico en el nodo Av. Ejército × Trinidad Morán × Cayma, Arequipa. El archivo incorpora la simulación de escenarios, el modo de prueba libre, la comparación con el sistema actual de 43 segundos, las fases semafóricas, la demanda vehicular equivalente, la presencia peatonal, los sensores y el panel de inferencia del modelo. Por su extensión, el código se presenta en fuente monoespaciada y tamaño reducido, conservando su legibilidad técnica.

Archivo fuente: sct_ml_simulacion.py | Lenguaje: Python | Librería principal: Pygame |

Total de líneas: 1687

```
0001 | # -*- coding: utf-8 -*-
0002 | """
0003 | SCT-ML – Simulación funcional en Python/Pygame
0004 | Av. Ejército × Trinidad Morán × Cayma – Arequipa
0005 |
0006 | Versión Python inspirada en la interfaz HTML SCT-ML:
0007 | - Escenarios validados.
0008 | - Prueba libre.
0009 | - Comparativo "Sistema actual 43s".
0010 | - Semáforos A1, A2, A3 y fase peatonal.
0011 | - Vehículos mixtos, peatones, sensores visuales/térmicos.
0012 | - Panel de inferencia ML simulado para defensa académica.
0013 |
0014 | Requisitos:
0015 |     pip install pygame
0016 |
0017 | Ejecutar:
0018 |     python sct_ml_simulacion.py
0019 |
0020 | Notas:
0021 | - SIM_SPEED acelera la simulación para defensa. Si quieres tiempos reales, pon SIM_SPEED = 1.0.
0022 | - El modo "Sistema actual 43s" mantiene el tiempo lógico de 43 segundos por fase vehicular.
0023 | """
```

```

0024 |
0025 | from __future__ import annotations
0026 |
0027 | import math
0028 | import random
0029 | import sys
0030 | import time
0031 | from dataclasses import dataclass, field
0032 | from typing import Dict, List, Tuple, Optional
0033 |
0034 | try:
0035 |     import pygame
0036 | except ImportError:
0037 |     raise SystemExit(
0038 |         "Falta instalar pygame.\n"
0039 |         "Instala con: pip install pygame\n"
0040 |         "Luego ejecuta: python sct_ml_simulacion.py"
0041 |     )
0042 |
0043 |
0044 | # =====
0045 | # Configuración general
0046 | # =====
0047 |
0048 | APP_TITLE = "SCT-ML – Sistema Inteligente de Control de Tráfico"
0049 | WIDTH, HEIGHT = 1280, 760
0050 | LEFT_W, RIGHT_W = 310, 350
0051 | STAGE_X = LEFT_W
0052 | STAGE_W = WIDTH - LEFT_W - RIGHT_W
0053 | STAGE_H = HEIGHT
0054 | FPS = 60
0055 |
0056 | # Acelera el tiempo lógico para que en la sustentación no se espere demasiado.
0057 | # 1.0 = tiempo real. 4.0 = 1 segundo real equivale a 4 segundos simulados.
0058 | SIM_SPEED = 4.0
0059 |
0060 | # Colores
0061 | BG = (7, 15, 26)
0062 | PANEL = (15, 28, 46)
0063 | PANEL_2 = (21, 34, 54)
0064 | PANEL_3 = (8, 19, 33)
0065 | LINE = (26, 45, 69)
0066 | TEXT = (230, 240, 251)

```

```

0067 | MUTED = (142, 164, 193)
0068 | SOFT = (184, 202, 224)
0069 | CYAN = (0, 212, 255)
0070 | VIOLET = (167, 139, 250)
0071 | GREEN = (52, 211, 153)
0072 | GREEN_LIGHT = (0, 230, 118)
0073 | AMBER = (245, 158, 11)
0074 | YELLOW = (255, 196, 0)
0075 | RED = (255, 51, 95)
0076 | WHITE = (255, 255, 255)
0077 | BLACK = (0, 0, 0)
0078 |
0079 |
0080 | # =====
0081 | # Geometría del nodo vial dentro del escenario central
0082 | # =====
0083 |
0084 | @dataclass
0085 | class Geo:
0086 |     """Coordenadas del nodo vial, escaladas al escenario central."""
0087 |
0088 |     w: int
0089 |     h: int
0090 |
0091 |     @property
0092 |     def hY(self) -> int:
0093 |         return int(self.h * 0.47)
0094 |
0095 |     @property
0096 |     def hH(self) -> int:
0097 |         return int(self.h * 0.13)
0098 |
0099 |     @property
0100 |     def hMid(self) -> int:
0101 |         return self.hY + self.hH // 2
0102 |
0103 |     @property
0104 |     def vX(self) -> int:
0105 |         return int(self.w * 0.32)
0106 |
0107 |     @property
0108 |     def vW(self) -> int:
0109 |         return int(self.w * 0.13)

```

```

0110 |
0111 |     @property
0112 |     def vMid(self) -> int:
0113 |         return self.vX + self.vW // 2
0114 |
0115 |     @property
0116 |     def cayX(self) -> int:
0117 |         return int(self.w * 0.74)
0118 |
0119 |     @property
0120 |     def cayW(self) -> int:
0121 |         return int(self.w * 0.12)
0122 |
0123 |     @property
0124 |     def stopA1(self) -> int:
0125 |         return int(self.w * 0.52)
0126 |
0127 |     @property
0128 |     def stopA2(self) -> int:
0129 |         return int(self.w * 0.24)
0130 |
0131 |     @property
0132 |     def stopA3(self) -> int:
0133 |         return int(self.h * 0.66)
0134 |
0135 |     @property
0136 |     def pedY(self) -> int:
0137 |         return self.hMid
0138 |
0139 |
0140 | # =====
0141 | # Datos base: escenarios, vehículos, tiempos
0142 | # =====
0143 |
0144 | @dataclass
0145 | class Scenario:
0146 |     name: str
0147 |     description: str
0148 |     levels: Tuple[str, str, str]
0149 |     mix: str
0150 |     pedestrians: int
0151 |     thermal: bool
0152 |     reading: str

```

```

0153 |
0154 |
0155 | SCENARIOS: List[Scenario] = [
0156 |     Scenario(
0157 |         "E1 · Flujo bajo",
0158 |         "Tráfico ligero, pocos peatones.",
0159 |         ("baja", "baja", "baja"),
0160 |         "normal",
0161 |         5,
0162 |         True,
0163 |         "Flujo bajo: el sistema evita sostener verdes innecesarios y mantiene una operación simple."
0164 |     ),
0165 |     Scenario(
0166 |         "E2 · Flujo medio",
0167 |         "Operación urbana regular.",
0168 |         ("media", "media", "media"),
0169 |         "normal",
0170 |         14,
0171 |         True,
0172 |         "Flujo medio: Av. Ejército recibe prioridad, pero A3 conserva turno cíclico por equidad."
0173 |     ),
0174 |     Scenario(
0175 |         "E3 · Hora punta",
0176 |         "Av. Ejército cargada con colas.",
0177 |         ("alta", "alta", "media"),
0178 |         "punta",
0179 |         18,
0180 |         True,
0181 |         "Hora punta: se evidencia cola en la vía principal y el sistema alterna sin dejar bloqueado el acceso secundario."
0182 |     ),
0183 |     Scenario(
0184 |         "E4 · Transporte público alto",
0185 |         "Más combis/buses, pero flujo mixto.",
0186 |         ("alta", "alta", "media"),
0187 |         "publico",
0188 |         16,
0189 |         True,
0190 |         "Transporte público alto: aumenta el peso operativo, sin eliminar autos ni motos del flujo realista."
0191 |     ),
0192 |     Scenario(
0193 |         "E5 · Alta presencia peatonal",
0194 |         "Peatones altos con tráfico medio/alto.",
0195 |         ("alta", "media", "media"),

```

```

0196 |     "punta",
0197 |     36,
0198 |     True,
0199 |     "Alta presencia peatonal: la fase F3 se activa tras transición segura, no como salto instantáneo."
0200 | ),
0201 | Scenario(
0202 |     "E6 · Conflicto peatón-vehículo",
0203 |     "Alta demanda vehicular y peatonal.",
0204 |     ("critica", "alta", "alta"),
0205 |     "intenso",
0206 |     44,
0207 |     True,
0208 |     "Conflicto crítico: el controlador protege el cruce peatonal y luego restaura el flujo vehicular."
0209 | ),
0210 | Scenario(
0211 |     "E7 · Discrepancia sensorica",
0212 |     "Cámara ve peatones, térmico no confirma.",
0213 |     ("alta", "media", "media"),
0214 |     "normal",
0215 |     28,
0216 |     False,
0217 |     "Discrepancia sensorica: se muestra incertidumbre y respuesta conservadora en la cadena de decisión."
0218 | ),
0219 | ]
0220 |
0221 | TYPE_SPEC: Dict[str, Dict[str, float]] = {
0222 |     "bus": {"w": 34, "h": 16, "weight": 2.5},
0223 |     "custer": {"w": 28, "h": 14, "weight": 2.0},
0224 |     "auto": {"w": 22, "h": 12, "weight": 1.0},
0225 |     "moto": {"w": 14, "h": 8, "weight": 0.5},
0226 |     "bici": {"w": 12, "h": 7, "weight": 0.4},
0227 |     "scoot": {"w": 10, "h": 6, "weight": 0.3},
0228 | }
0229 |
0230 | ADAPTIVE_DUR = {
0231 |     "main": 12_000,
0232 |     "a3": 6_500,
0233 |     "ped": 8_500,
0234 |     "yellow": 1_800,
0235 |     "allred": 900,
0236 | }
0237 |
0238 | FIXED_DUR = {

```

```

0239 |     "green": 43_000,
0240 |     "yellow": 3_200,
0241 |     "allred": 1_200,
0242 |     "ped": 8_500,
0243 | }
0244 |
0245 |
0246 | # =====
0247 | # Utilidades de dibujo
0248 | # =====
0249 |
0250 | def clamp(value: float, lo: float, hi: float) -> float:
0251 |     return max(lo, min(hi, value))
0252 |
0253 |
0254 | def lerp(a: float, b: float, t: float) -> float:
0255 |     return a + (b - a) * t
0256 |
0257 |
0258 | def draw_text(
0259 |     surface: pygame.Surface,
0260 |     text: str,
0261 |     font: pygame.font.Font,
0262 |     color: Tuple[int, int, int],
0263 |     x: int,
0264 |     y: int,
0265 |     max_width: Optional[int] = None,
0266 |     line_gap: int = 4,
0267 | ) -> int:
0268 |     """
0269 |     Dibuja texto con saltos automáticos.
0270 |     Retorna la coordenada Y final.
0271 |     """
0272 |     if not text:
0273 |         return y
0274 |
0275 |     if max_width is None:
0276 |         img = font.render(text, True, color)
0277 |         surface.blit(img, (x, y))
0278 |         return y + img.get_height()
0279 |
0280 |     words = str(text).replace("\n", " \n ").split(" ")
0281 |     line = ""

```

```

0282 |     cur_y = y
0283 |
0284 |     for word in words:
0285 |         if word == "\n":
0286 |             if line:
0287 |                 img = font.render(line, True, color)
0288 |                 surface.blit(img, (x, cur_y))
0289 |                 cur_y += img.get_height() + line_gap
0290 |                 line = ""
0291 |             else:
0292 |                 cur_y += font.get_height() + line_gap
0293 |             continue
0294 |
0295 |         test = word if not line else f"{line} {word}"
0296 |         if font.size(test)[0] <= max_width:
0297 |             line = test
0298 |         else:
0299 |             if line:
0300 |                 img = font.render(line, True, color)
0301 |                 surface.blit(img, (x, cur_y))
0302 |                 cur_y += img.get_height() + line_gap
0303 |             line = word
0304 |
0305 |         if line:
0306 |             img = font.render(line, True, color)
0307 |             surface.blit(img, (x, cur_y))
0308 |             cur_y += img.get_height() + line_gap
0309 |
0310 |     return cur_y
0311 |
0312 |
0313 | def draw_rounded_rect(
0314 |     surface: pygame.Surface,
0315 |     rect: pygame.Rect,
0316 |     color: Tuple[int, int, int],
0317 |     radius: int = 8,
0318 |     border: Optional[Tuple[int, int, int]] = None,
0319 |     border_width: int = 1,
0320 | ) -> None:
0321 |     pygame.draw.rect(surface, color, rect, border_radius=radius)
0322 |     if border:
0323 |         pygame.draw.rect(surface, border, rect, border_width, border_radius=radius)
0324 |

```

```

0325 |
0326 | def draw_bar(
0327 |     surface: pygame.Surface,
0328 |     x: int,
0329 |     y: int,
0330 |     w: int,
0331 |     h: int,
0332 |     pct: float,
0333 |     fill_color: Tuple[int, int, int],
0334 | ) -> None:
0335 |     pygame.draw.rect(surface, (6, 16, 29), (x, y, w, h), border_radius=h // 2)
0336 |     pygame.draw.rect(surface, LINE, (x, y, w, h), 1, border_radius=h // 2)
0337 |     fill_w = int(w * clamp(pct, 0, 1))
0338 |     if fill_w > 0:
0339 |         pygame.draw.rect(surface, fill_color, (x, y, fill_w, h), border_radius=h // 2)
0340 |
0341 |
0342 | # =====
0343 | # UI básica
0344 | # =====
0345 |
0346 | @dataclass
0347 | class Button:
0348 |     rect: pygame.Rect
0349 |     text: str
0350 |     subtext: str = ""
0351 |     active: bool = False
0352 |     kind: str = "normal"
0353 |
0354 |     def draw(self, surface: pygame.Surface, fonts: Dict[str, pygame.font.Font]) -> None:
0355 |         if self.kind == "danger":
0356 |             active_border = RED
0357 |             bg_active = (45, 20, 34)
0358 |         elif self.kind == "green":
0359 |             active_border = GREEN
0360 |             bg_active = (12, 51, 45)
0361 |         else:
0362 |             active_border = AMBER
0363 |             bg_active = (46, 37, 22)
0364 |
0365 |         bg = bg_active if self.active else PANEL_2
0366 |         border = active_border if self.active else LINE
0367 |

```

```

0368 |         draw_rounded_rect(surface, self.rect, bg, 8, border, 1)
0369 |
0370 |         tx = self.rect.x + 10
0371 |         ty = self.rect.y + 7
0372 |
0373 |         title_color = WHITE if self.active else TEXT
0374 |         draw_text(surface, self.text, fonts["btn"], title_color, tx, ty, self.rect.w - 18, 2)
0375 |
0376 |         if self.subtext:
0377 |             draw_text(surface, self.subtext, fonts["tiny"], MUTED, tx, ty + 20, self.rect.w - 18, 2)
0378 |
0379 |         def contains(self, pos: Tuple[int, int]) -> bool:
0380 |             return self.rect.collidepoint(pos)
0381 |
0382 |
0383 | # =====
0384 | # Modelo de demanda y vehículos
0385 | # =====
0386 |
0387 | def counts_from(level: str, mix: str, lane: str) -> Dict[str, int]:
0388 |     base_table = {
0389 |         "baja": {"bus": 0, "custer": 0, "auto": 4, "moto": 1, "bici": 0, "scoot": 0},
0390 |         "media": {"bus": 1, "custer": 1, "auto": 8, "moto": 3, "bici": 1, "scoot": 1},
0391 |         "alta": {"bus": 3, "custer": 2, "auto": 12, "moto": 6, "bici": 1, "scoot": 1},
0392 |         "critica": {"bus": 5, "custer": 4, "auto": 16, "moto": 9, "bici": 1, "scoot": 1},
0393 |     }
0394 |
0395 |     c = dict(base_table.get(level, base_table["media"]))
0396 |
0397 |     if mix == "punta":
0398 |         c["auto"] += 3
0399 |         c["moto"] += 2
0400 |         c["bus"] += 1
0401 |         c["custer"] += 1
0402 |     elif mix == "publico":
0403 |         c["bus"] += 4
0404 |         c["custer"] += 3
0405 |         c["auto"] += 1
0406 |         c["moto"] += 1
0407 |     elif mix == "motos":
0408 |         c["moto"] += 5
0409 |         c["scoot"] += 2
0410 |         c["auto"] += 1

```

```

0411 |         c["bus"] += 1
0412 |     elif mix == "intenso":
0413 |         c["bus"] += 2
0414 |         c["custer"] += 2
0415 |         c["auto"] += 4
0416 |         c["moto"] += 4
0417 |         c["scoot"] += 1
0418 |
0419 |     # El acceso secundario A3 tiene menor capacidad y menor volumen visual.
0420 |     if lane == "a3":
0421 |         c["bus"] = max(0 if level == "baja" else 1, round(c["bus"] * 0.55))
0422 |         c["custer"] = max(0 if level == "baja" else 1, round(c["custer"] * 0.65))
0423 |         c["auto"] = max(2 if level == "baja" else 4, round(c["auto"] * 0.70))
0424 |         c["moto"] = max(1 if level == "baja" else 2, round(c["moto"] * 0.75))
0425 |
0426 |     return c
0427 |
0428 |
0429 | def weight_demand(counts: Dict[str, int]) -> float:
0430 |     total = 0.0
0431 |     for k, v in counts.items():
0432 |         total += TYPE_SPEC.get(k, {"weight": 1.0})["weight"] * v
0433 |     return total
0434 |
0435 |
0436 | def vehicle_count(counts: Dict[str, int], lane: str) -> int:
0437 |     n = (
0438 |         counts["bus"]
0439 |         + counts["custer"]
0440 |         + math.ceil(counts["auto"] / 1.6)
0441 |         + math.ceil(counts["moto"] / 2.2)
0442 |         + math.ceil(counts["bici"] / 3)
0443 |         + math.ceil(counts["scoot"] / 3)
0444 |     )
0445 |     return int(clamp(n, 0, 10 if lane == "a3" else 14))
0446 |
0447 |
0448 | def sample_type(counts: Dict[str, int]) -> str:
0449 |     total = max(1, sum(counts.values()))
0450 |     r = random.random() * total
0451 |     for k, v in counts.items():
0452 |         r -= v
0453 |         if r <= 0:

```

```

0454 |         return k
0455 |     return "auto"
0456 |
0457 |
0458 | @dataclass
0459 | class Vehicle:
0460 |     lane: str
0461 |     index: int
0462 |     vtype: str
0463 |     geo: Geo
0464 |     x: float = 0.0
0465 |     y: float = 0.0
0466 |     speed: float = field(default_factory=lambda: random.uniform(0.20, 0.40))
0467 |     max_speed: float = field(default_factory=lambda: random.uniform(0.95, 1.35))
0468 |
0469 |     def __post_init__(self) -> None:
0470 |         self.reset(initial=True)
0471 |
0472 |     @property
0473 |     def dims(self) -> Tuple[float, float]:
0474 |         spec = TYPE_SPEC[self.vtype]
0475 |         if self.lane == "a3":
0476 |             return spec["h"], spec["w"]
0477 |         return spec["w"], spec["h"]
0478 |
0479 |     def reset(self, initial: bool = False) -> None:
0480 |         w, h = self.dims
0481 |         gap = 40 if self.lane != "a3" else 34
0482 |         sub = self.index % 2
0483 |         row = self.index // 2
0484 |
0485 |         if self.lane == "a1":
0486 |             # Derecha a izquierda.
0487 |             self.x = (
0488 |                 self.geo.stopA1 + 34 + row * gap
0489 |                 if initial
0490 |                 else self.geo.w + 50 + random.random() * 120
0491 |             )
0492 |             self.y = self.geo.hY + 18 + sub * 22
0493 |         elif self.lane == "a2":
0494 |             # Izquierda a derecha.
0495 |             self.x = (
0496 |                 self.geo.stopA2 - 34 - row * gap

```

```

0497 |         if initial
0498 |             else -50 - random.random() * 120
0499 |         )
0500 |         self.y = self.geo.hY + self.geo.hH - 18 - sub * 22
0501 |     else:
0502 |         # A3 hacia arriba.
0503 |         self.x = self.geo.vX + 22 + sub * 26
0504 |         self.y = (
0505 |             self.geo.stopA3 + 34 + row * gap
0506 |             if initial
0507 |             else self.geo.h + 50 + random.random() * 120
0508 |         )
0509 |
0510 |     self.speed = 0.0
0511 |
0512 |     def sem_value(self, sem: Dict[str, str]) -> str:
0513 |         if self.lane == "a1":
0514 |             return sem["a1"]
0515 |         if self.lane == "a2":
0516 |             return sem["a2"]
0517 |         return sem["a3"]
0518 |
0519 |     def update(self, sem: Dict[str, str], leader: Optional["Vehicle"]) -> None:
0520 |         green = self.sem_value(sem) == "green"
0521 |         target = self.max_speed
0522 |         gap = 36 if self.lane != "a3" else 32
0523 |         w, h = self.dims
0524 |
0525 |         if self.lane == "a1":
0526 |             stop = self.geo.stopA1 + w / 2 + 5
0527 |             if not green:
0528 |                 if self.x <= stop:
0529 |                     self.x = stop
0530 |                     target = 0
0531 |                 else:
0532 |                     target = min(target, max(0, (self.x - stop) / 100) * self.max_speed)
0533 |
0534 |             if leader and leader.x < self.x and self.x - leader.x < gap:
0535 |                 target = 0
0536 |
0537 |             self.speed += (target - self.speed) * 0.10
0538 |             self.x -= max(0, self.speed)
0539 |             if self.x < -80:

```

```

0540 |         self.reset(False)
0541 |
0542 |     elif self.lane == "a2":
0543 |         stop = self.geo.stopA2 - w / 2 - 5
0544 |         if not green:
0545 |             if self.x >= stop:
0546 |                 self.x = stop
0547 |                 target = 0
0548 |             else:
0549 |                 target = min(target, max(0, (stop - self.x) / 100) * self.max_speed)
0550 |
0551 |         if leader and leader.x > self.x and leader.x - self.x < gap:
0552 |             target = 0
0553 |
0554 |         self.speed += (target - self.speed) * 0.10
0555 |         self.x += max(0, self.speed)
0556 |         if self.x > self.geo.w + 80:
0557 |             self.reset(False)
0558 |
0559 |     else:
0560 |         stop = self.geo.stopA3 + h / 2 + 5
0561 |         if not green:
0562 |             if self.y <= stop:
0563 |                 self.y = stop
0564 |                 target = 0
0565 |             else:
0566 |                 target = min(target, max(0, (self.y - stop) / 100) * self.max_speed)
0567 |
0568 |         if leader and leader.y < self.y and self.y - leader.y < gap:
0569 |             target = 0
0570 |
0571 |         self.speed += (target - self.speed) * 0.10
0572 |         self.y -= max(0, self.speed)
0573 |         if self.y < -80:
0574 |             self.reset(False)
0575 |
0576 | def draw(self, surface: pygame.Surface) -> None:
0577 |     w, h = self.dims
0578 |     color = CYAN if self.lane == "a1" else VIOLET if self.lane == "a2" else GREEN
0579 |     rect = pygame.Rect(int(self.x - w / 2), int(self.y - h / 2), int(w), int(h))
0580 |     draw_rounded_rect(surface, rect, color, 4, (255, 255, 255), 1)
0581 |
0582 |     # Ventana

```

```

0583 |         window_rect = pygame.Rect(
0584 |             int(self.x - w * 0.22),
0585 |             int(self.y - h * 0.30),
0586 |             max(3, int(w * 0.44)),
0587 |             max(2, int(h * 0.22)),
0588 |         )
0589 |         pygame.draw.rect(surface, (255, 255, 255, 70), window_rect)
0590 |
0591 |
0592 | # =====
0593 | # Estado de simulación
0594 | # =====
0595 |
0596 | @dataclass
0597 | class Pedestrian:
0598 |     x: float
0599 |     y: float
0600 |     tx: float
0601 |     ty: float
0602 |     speed: float
0603 |     done: bool = False
0604 |
0605 |     def update(self) -> None:
0606 |         dx = self.tx - self.x
0607 |         dy = self.ty - self.y
0608 |         dist = max(0.0001, math.hypot(dx, dy))
0609 |         if dist < 2:
0610 |             self.done = True
0611 |             return
0612 |         self.x += dx / dist * self.speed
0613 |         self.y += dy / dist * self.speed
0614 |
0615 |
0616 | @dataclass
0617 | class SimState:
0618 |     mode: str = "scenarios"      # scenarios, free, actual
0619 |     scenario_index: int = 1
0620 |     levels: List[str] = field(default_factory=lambda: ["media", "media", "media"])
0621 |     mix: str = "normal"
0622 |     ped_target: int = 14
0623 |     thermal: bool = True
0624 |     phase: str = "main"
0625 |     phase_elapsed_ms: float = 0.0

```

```

0626 |     sem: Dict[str, object] = field(default_factory=lambda: {"a1": "green", "a2": "green", "a3": "red", "p": False})
0627 |     ped_waiting: int = 14
0628 |     ped_crossing: List[Pedestrian] = field(default_factory=list)
0629 |     vehicles: Dict[str, List[Vehicle]] = field(default_factory=lambda: {"a1": [], "a2": [], "a3": []})
0630 |     reading: str = SCENARIOS[1].reading
0631 |
0632 |
0633 | # =====
0634 | # Aplicación principal
0635 | # =====
0636 |
0637 | class SCTMLApp:
0638 |     def __init__(self) -> None:
0639 |         pygame.init()
0640 |         pygame.display.set_caption(APP_TITLE)
0641 |         self.screen = pygame.display.set_mode((WIDTH, HEIGHT))
0642 |         self.clock = pygame.time.Clock()
0643 |
0644 |         self.fonts = {
0645 |             "title": pygame.font.SysFont("Segoe UI", 18, bold=True),
0646 |             "h1": pygame.font.SysFont("Segoe UI", 15, bold=True),
0647 |             "body": pygame.font.SysFont("Segoe UI", 12),
0648 |             "small": pygame.font.SysFont("Segoe UI", 11),
0649 |             "tiny": pygame.font.SysFont("Segoe UI", 9),
0650 |             "btn": pygame.font.SysFont("Segoe UI", 10, bold=True),
0651 |             "mono": pygame.font.SysFont("Consolas", 10),
0652 |         }
0653 |
0654 |         self.geo = Geo(STAGE_W, STAGE_H)
0655 |         self.state = SimState()
0656 |
0657 |         self.stage_surface = pygame.Surface((STAGE_W, STAGE_H))
0658 |         self.buttons: List[Button] = []
0659 |         self.scenario_buttons: List[Button] = []
0660 |         self.free_buttons: Dict[str, Button] = {}
0661 |
0662 |         # Parámetros libres editables.
0663 |         self.free_levels = ["media", "media", "media"]
0664 |         self.free_mix = "normal"
0665 |         self.free_ped = "media"
0666 |         self.free_thermal = True
0667 |
0668 |         self.build_vehicles()

```

```

0669 |         self.reset_scenario(1)
0670 |         self.rebuild_ui()
0671 |
0672 |         # -----
0673 |         # Reset y construcción
0674 |         # -----
0675 |
0676 |     def build_vehicles(self) -> None:
0677 |         for lane, idx in [("a1", 0), ("a2", 1), ("a3", 2)]:
0678 |             counts = counts_from(self.state.levels[idx], self.state.mix, lane)
0679 |             n = vehicle_count(counts, lane)
0680 |             self.state.vehicles[lane] = [
0681 |                 Vehicle(lane, i, sample_type(counts), self.geo)
0682 |                 for i in range(n)
0683 |             ]
0684 |
0685 |     def reset_scenario(self, i: int) -> None:
0686 |         s = SCENARIOS[i]
0687 |         self.state.mode = "scenarios"
0688 |         self.state.scenario_index = i
0689 |         self.state.levels = list(s.levels)
0690 |         self.state.mix = s.mix
0691 |         self.state.ped_target = s.pedestrians
0692 |         self.state.thermal = s.thermal
0693 |         self.state.ped_waiting = s.pedestrians
0694 |         self.state.ped_crossing.clear()
0695 |         self.state.phase = "main"
0696 |         self.state.phase_elapsed_ms = 0
0697 |         self.state.sem = {"a1": "green", "a2": "green", "a3": "red", "p": False}
0698 |         self.state.reading = s.reading
0699 |         self.build_vehicles()
0700 |
0701 |     def reset_free(self) -> None:
0702 |         ped_map = {"baja": 6, "media": 16, "alta": 32, "critica": 44}
0703 |         self.state.mode = "free"
0704 |         self.state.levels = list(self.free_levels)
0705 |         self.state.mix = self.free_mix
0706 |         self.state.ped_target = ped_map[self.free_ped]
0707 |         self.state.thermal = self.free_thermal
0708 |         self.state.ped_waiting = self.state.ped_target
0709 |         self.state.ped_crossing.clear()
0710 |         self.state.phase = "main"
0711 |         self.state.phase_elapsed_ms = 0

```

```

0712 |         self.state.sem = {"a1": "green", "a2": "green", "a3": "red", "p": False}
0713 |         self.state.reading = (
0714 |             "Prueba libre: el jurado modifica demanda, peatones y validación térmica; "
0715 |             "el controlador mantiene un ciclo seguro y visible."
0716 |         )
0717 |         self.build_vehicles()
0718 |
0719 |     def run_actual_scenariio(self, i: Optional[int] = None) -> None:
0720 |         if i is not None:
0721 |             s = SCENARIOS[i]
0722 |             self.state.scenario_index = i
0723 |             self.state.levels = list(s.levels)
0724 |             self.state.mix = s.mix
0725 |             self.state.ped_target = s.pedestrians
0726 |             self.state.thermal = s.thermal
0727 |             self.state.ped_waiting = s.pedestrians
0728 |             self.state.reading = (
0729 |                 "Sistema actual 43s: solo opera con escenarios fijos. No permite modificar "
0730 |                 "parámetros porque representa un control rígido convencional."
0731 |             )
0732 |             self.build_vehicles()
0733 |
0734 |             self.state.mode = "actual"
0735 |             self.state.phase = "fixedMain"
0736 |             self.state.phase_elapsed_ms = 0
0737 |             self.state.ped_crossing.clear()
0738 |             self.state.sem = {"a1": "green", "a2": "green", "a3": "red", "p": False}
0739 |
0740 |         # -----
0741 |         # UI
0742 |         # -----
0743 |
0744 |     def rebuild_ui(self) -> None:
0745 |         self.buttons.clear()
0746 |         self.scenario_buttons.clear()
0747 |         self.free_buttons.clear()
0748 |
0749 |         # Botones superiores del panel izquierdo
0750 |         x, y = 14, 88
0751 |         bw = 92
0752 |         self.buttons.append(Button(pygame.Rect(x, y, bw, 38), "Escenarios", active=self.state.mode == "scenarios"))
0753 |         self.buttons.append(Button(pygame.Rect(x + bw + 6, y, bw, 38), "Prueba libre", active=self.state.mode == "free"))

```

```

0754 |         self.buttons.append(Button(pygame.Rect(x + 2 * (bw + 6), y, bw, 38), "Actual 43s", active=self.state.mode ==
"actual", kind="danger"))
0755 |
0756 |         # Escenarios
0757 |         sy = 140
0758 |         for i, s in enumerate(SCENARIOS):
0759 |             rect = pygame.Rect(14, sy + i * 58, LEFT_W - 28, 50)
0760 |             active = self.state.scenario_index == i and self.state.mode in ("scenarios", "actual")
0761 |             self.scenario_buttons.append(Button(rect, s.name, s.description, active=active))
0762 |
0763 |         # Prueba libre
0764 |         fy = 145
0765 |         labels = [
0766 |             ("A1 Ejército E→O", "level0"),
0767 |             ("A2 Ejército O→E", "level1"),
0768 |             ("A3 Trinidad/Cayma", "level2"),
0769 |             ("Mezcla vehicular", "mix"),
0770 |             ("Demanda peatonal", "ped"),
0771 |             ("Validación térmica", "thermal"),
0772 |             ("▶ Ejecutar prueba libre", "run"),
0773 |         ]
0774 |
0775 |         for j, (label, key) in enumerate(labels):
0776 |             if key == "run":
0777 |                 rect = pygame.Rect(14, fy + j * 58, LEFT_W - 28, 42)
0778 |                 self.free_buttons[key] = Button(rect, label, "Aplica las variables seleccionadas", kind="green")
0779 |             else:
0780 |                 rect = pygame.Rect(14, fy + j * 58, LEFT_W - 28, 48)
0781 |                 self.free_buttons[key] = Button(rect, label, self.free_value_text(key))
0782 |
0783 |         def free_value_text(self, key: str) -> str:
0784 |             if key == "level0":
0785 |                 return f"Actual: {self.free_levels[0]}"
0786 |             if key == "level1":
0787 |                 return f"Actual: {self.free_levels[1]}"
0788 |             if key == "level2":
0789 |                 return f"Actual: {self.free_levels[2]}"
0790 |             if key == "mix":
0791 |                 return f"Actual: {self.free_mix}"
0792 |             if key == "ped":
0793 |                 return f"Actual: {self.free_ped}"
0794 |             if key == "thermal":
0795 |                 return "Confirma presencia" if self.free_thermal else "No confirma / discrepancia"

```

```

0796 |         return ""
0797 |
0798 |     def cycle_free_option(self, key: str) -> None:
0799 |         levels = ["baja", "media", "alta", "critica"]
0800 |         mixes = ["normal", "punta", "publico", "motos", "intenso"]
0801 |         peds = ["baja", "media", "alta", "critica"]
0802 |
0803 |         if key.startswith("level"):
0804 |             idx = int(key[-1])
0805 |             cur = self.free_levels[idx]
0806 |             self.free_levels[idx] = levels[(levels.index(cur) + 1) % len(levels)]
0807 |         elif key == "mix":
0808 |             self.free_mix = mixes[(mixes.index(self.free_mix) + 1) % len(mixes)]
0809 |         elif key == "ped":
0810 |             self.free_ped = peds[(peds.index(self.free_ped) + 1) % len(peds)]
0811 |         elif key == "thermal":
0812 |             self.free_thermal = not self.free_thermal
0813 |
0814 |         self.rebuild_ui()
0815 |
0816 |         # -----
0817 |         # Lógica de fases
0818 |         # -----
0819 |
0820 |     def go_phase(self, phase: str) -> None:
0821 |         self.state.phase = phase
0822 |         self.state.phase_elapsed_ms = 0
0823 |
0824 |     def tick_phase_adaptive(self) -> None:
0825 |         s = self.state
0826 |         t = s.phase_elapsed_ms
0827 |
0828 |         # Umbral dinámico de peatones.
0829 |         need_ped = s.ped_waiting >= min(24, max(10, s.ped_target * 0.70))
0830 |
0831 |         if s.phase == "main":
0832 |             s.sem = {"a1": "green", "a2": "green", "a3": "red", "p": False}
0833 |             if t > ADAPTIVE_DUR["main"]:
0834 |                 self.go_phase("mainYped" if need_ped else "mainY")
0835 |
0836 |         elif s.phase == "mainY":
0837 |             s.sem = {"a1": "yellow", "a2": "yellow", "a3": "red", "p": False}
0838 |             if t > ADAPTIVE_DUR["yellow"]:

```

```

0839 |         self.go_phase("allredA3")
0840 |
0841 |     elif s.phase == "mainYped":
0842 |         s.sem = {"a1": "yellow", "a2": "yellow", "a3": "red", "p": False}
0843 |         if t > ADAPTIVE_DUR["yellow"]:
0844 |             self.go_phase("allredPed")
0845 |
0846 |     elif s.phase == "allredA3":
0847 |         s.sem = {"a1": "red", "a2": "red", "a3": "red", "p": False}
0848 |         if t > ADAPTIVE_DUR["allred"]:
0849 |             self.go_phase("a3")
0850 |
0851 |     elif s.phase == "a3":
0852 |         s.sem = {"a1": "red", "a2": "red", "a3": "green", "p": False}
0853 |         if t > ADAPTIVE_DUR["a3"]:
0854 |             self.go_phase("a3Y")
0855 |
0856 |     elif s.phase == "a3Y":
0857 |         s.sem = {"a1": "red", "a2": "red", "a3": "yellow", "p": False}
0858 |         if t > ADAPTIVE_DUR["yellow"]:
0859 |             self.go_phase("allredPed" if need_ped else "allredMain")
0860 |
0861 |     elif s.phase == "allredPed":
0862 |         s.sem = {"a1": "red", "a2": "red", "a3": "red", "p": False}
0863 |         if t > ADAPTIVE_DUR["allred"]:
0864 |             self.start_ped_crossing()
0865 |             self.go_phase("ped")
0866 |
0867 |     elif s.phase == "ped":
0868 |         s.sem = {"a1": "red", "a2": "red", "a3": "red", "p": True}
0869 |         if t > ADAPTIVE_DUR["ped"] or not s.ped_crossing:
0870 |             self.go_phase("allredMain")
0871 |
0872 |     elif s.phase == "allredMain":
0873 |         s.sem = {"a1": "red", "a2": "red", "a3": "red", "p": False}
0874 |         if t > ADAPTIVE_DUR["allred"]:
0875 |             self.go_phase("main")
0876 |
0877 |     else:
0878 |         self.go_phase("main")
0879 |
0880 | def tick_phase_fixed(self) -> None:
0881 |     s = self.state

```

```

0882 |         t = s.phase_elapsed_ms
0883 |         ped_req = s.ped_waiting >= min(24, max(10, s.ped_target * 0.70))
0884 |
0885 |         if s.phase == "fixedMain":
0886 |             s.sem = {"a1": "green", "a2": "green", "a3": "red", "p": False}
0887 |             if t > FIXED_DUR["green"]:
0888 |                 self.go_phase("fixedMainY")
0889 |
0890 |         elif s.phase == "fixedMainY":
0891 |             s.sem = {"a1": "yellow", "a2": "yellow", "a3": "red", "p": False}
0892 |             if t > FIXED_DUR["yellow"]:
0893 |                 self.go_phase("fixedAllredA3")
0894 |
0895 |         elif s.phase == "fixedAllredA3":
0896 |             s.sem = {"a1": "red", "a2": "red", "a3": "red", "p": False}
0897 |             if t > FIXED_DUR["allred"]:
0898 |                 self.go_phase("fixedA3")
0899 |
0900 |         elif s.phase == "fixedA3":
0901 |             s.sem = {"a1": "red", "a2": "red", "a3": "green", "p": False}
0902 |             if t > FIXED_DUR["green"]:
0903 |                 self.go_phase("fixedA3Y")
0904 |
0905 |         elif s.phase == "fixedA3Y":
0906 |             s.sem = {"a1": "red", "a2": "red", "a3": "yellow", "p": False}
0907 |             if t > FIXED_DUR["yellow"]:
0908 |                 self.go_phase("fixedAllredPed" if ped_req else "fixedAllredMain")
0909 |
0910 |         elif s.phase == "fixedAllredPed":
0911 |             s.sem = {"a1": "red", "a2": "red", "a3": "red", "p": False}
0912 |             if t > FIXED_DUR["allred"]:
0913 |                 self.start_ped_crossing()
0914 |                 self.go_phase("fixedPed")
0915 |
0916 |         elif s.phase == "fixedPed":
0917 |             s.sem = {"a1": "red", "a2": "red", "a3": "red", "p": True}
0918 |             if t > FIXED_DUR["ped"] or not s.ped_crossing:
0919 |                 self.go_phase("fixedAllredMain")
0920 |
0921 |         elif s.phase == "fixedAllredMain":
0922 |             s.sem = {"a1": "red", "a2": "red", "a3": "red", "p": False}
0923 |             if t > FIXED_DUR["allred"]:
0924 |                 self.go_phase("fixedMain")

```

```

0925 |
0926 |         else:
0927 |             self.go_phase("fixedMain")
0928 |
0929 |         # -----
0930 |         # Update
0931 |         # -----
0932 |
0933 |     def update(self, dt_ms_real: float) -> None:
0934 |         dt_ms = dt_ms_real * SIM_SPEED
0935 |         self.state.phase_elapsed_ms += dt_ms
0936 |
0937 |         if self.state.mode == "actual":
0938 |             self.tick_phase_fixed()
0939 |         else:
0940 |             self.tick_phase_adaptive()
0941 |
0942 |         self.update_vehicles()
0943 |         self.update_pedestrians()
0944 |
0945 |     def update_vehicles(self) -> None:
0946 |         for lane in ["a1", "a2", "a3"]:
0947 |             arr = self.state.vehicles[lane]
0948 |             if lane == "a1":
0949 |                 sorted_arr = sorted(arr, key=lambda v: v.x)
0950 |             elif lane == "a2":
0951 |                 sorted_arr = sorted(arr, key=lambda v: -v.x)
0952 |             else:
0953 |                 sorted_arr = sorted(arr, key=lambda v: v.y)
0954 |
0955 |             for i, vehicle in enumerate(sorted_arr):
0956 |                 leader = sorted_arr[i - 1] if i > 0 else None
0957 |                 vehicle.update(self.state.sem, leader)
0958 |
0959 |     def update_pedestrians(self) -> None:
0960 |         for p in self.state.ped_crossing:
0961 |             p.update()
0962 |         self.state.ped_crossing = [p for p in self.state.ped_crossing if not p.done]
0963 |
0964 |         # Llegada gradual de nuevos peatones.
0965 |         if self.state.phase in ("main", "fixedMain") and self.state.ped_waiting < self.state.ped_target:
0966 |             if random.random() < 0.015:
0967 |                 self.state.ped_waiting += 1

```

```

0968 |
0969 |     def start_ped_crossing(self) -> None:
0970 |         if self.state.ped_crossing:
0971 |             return
0972 |
0973 |         n = min(18, self.state.ped_waiting)
0974 |         for i in range(n):
0975 |             horizontal = (i % 2 == 0)
0976 |             if horizontal:
0977 |                 x = self.geo.vX - 20 - (i % 5) * 7
0978 |                 y = self.geo.pedY + ((i % 3) - 1) * 8
0979 |                 tx = self.geo.vX + self.geo.vW + 20
0980 |                 ty = y
0981 |             else:
0982 |                 x = self.geo.vX + self.geo.vW + 22 + (i % 5) * 7
0983 |                 y = self.geo.hY - 20 - (i % 5) * 5
0984 |                 tx = self.geo.vX - 22
0985 |                 ty = self.geo.hY + self.geo.hH + 18 + (i % 5) * 4
0986 |
0987 |             self.state.ped_crossing.append(
0988 |                 Pedestrian(x, y, tx, ty, random.uniform(0.75, 1.05))
0989 |             )
0990 |
0991 |             self.state.ped_waiting = max(0, self.state.ped_waiting - n)
0992 |
0993 |         # -----
0994 |         # Métricas e inferencia ML simulada
0995 |         # -----
0996 |
0997 |     def demands(self) -> Tuple[float, float, float]:
0998 |         d1 = weight_demand(counts_from(self.state.levels[0], self.state.mix, "a1"))
0999 |         d2 = weight_demand(counts_from(self.state.levels[1], self.state.mix, "a2"))
1000 |         d3 = weight_demand(counts_from(self.state.levels[2], self.state.mix, "a3"))
1001 |         return d1, d2, d3
1002 |
1003 |     def ml_inference(self) -> Dict[str, object]:
1004 |         d1, d2, d3 = self.demands()
1005 |         ped = self.state.ped_waiting
1006 |         combined_main = (d1 + d2) / 2.0
1007 |
1008 |         if self.state.mode == "actual":
1009 |             return {
1010 |                 "phase": "NO APLICA",

```

```

1011 |         "green": "43 s fijo",
1012 |         "final": "Ciclo rígido",
1013 |         "safety": "Sin inferencia",
1014 |         "conf": 0.0,
1015 |         "explain": (
1016 |             "En el sistema actual no hay modelo ML ni ajuste por demanda. "
1017 |             "El ciclo conserva 43 segundos aunque la cola crítica esté en otra fase."
1018 |         )
1019 |     }
1020 |
1021 |     ped_threshold = max(10, min(24, self.state.ped_target * 0.70))
1022 |
1023 |     if self.state.thermal and ped >= ped_threshold:
1024 |         phase = "F3 peatonal"
1025 |         dominance = ped / max(1, ped_threshold)
1026 |     elif d3 > combined_main * 0.92:
1027 |         phase = "F4 A3"
1028 |         dominance = d3 / max(1, combined_main)
1029 |     else:
1030 |         phase = "F1 Av. Ejército"
1031 |         dominance = combined_main / max(1, d3)
1032 |
1033 |     green_seconds = int(clamp(15 + dominance * 8, 15, 45))
1034 |     confidence = clamp(0.70 + min(0.24, abs(dominance - 1.0) * 0.10), 0.70, 0.94)
1035 |
1036 |     if not self.state.thermal and ped > 0:
1037 |         safety = "Conservador"
1038 |         confidence = min(confidence, 0.78)
1039 |         explain = (
1040 |             "La cámara registra peatones, pero el sensor térmico no confirma. "
1041 |             "El controlador evita una decisión agresiva y mantiene una transición segura."
1042 |         )
1043 |     else:
1044 |         safety = "Validada"
1045 |         explain = (
1046 |             "La decisión prioriza la demanda equivalente, la espera peatonal y las transiciones "
1047 |             "seguras con ámbar y todo rojo antes del cambio de fase."
1048 |         )
1049 |
1050 |     return {
1051 |         "phase": phase,
1052 |         "green": f"{green_seconds} s",
1053 |         "final": "Aplicar fase",

```

```

1054 |         "safety": safety,
1055 |         "conf": confidence,
1056 |         "explain": explain,
1057 |     }
1058 |
1059 |     # -----
1060 |     # Dibujo general
1061 |     # -----
1062 |
1063 |     def draw(self) -> None:
1064 |         self.screen.fill(BG)
1065 |         self.draw_header()
1066 |         self.draw_left_panel()
1067 |         self.draw_stage()
1068 |         self.draw_right_panel()
1069 |         pygame.display.flip()
1070 |
1071 |     def draw_header(self) -> None:
1072 |         # Barra superior simulada dentro del canvas total.
1073 |         pygame.draw.rect(self.screen, (10, 22, 37), (0, 0, WIDTH, 52))
1074 |         pygame.draw.line(self.screen, LINE, (0, 52), (WIDTH, 52), 1)
1075 |
1076 |         badge = pygame.Rect(18, 10, 34, 32)
1077 |         draw_rounded_rect(self.screen, badge, (0, 105, 136), 6, CYAN, 1)
1078 |         draw_text(self.screen, "ITS", self.fonts["btn"], WHITE, badge.x + 7, badge.y + 10)
1079 |
1080 |         draw_text(self.screen, "Sistema Inteligente de Control de Tráfico", self.fonts["title"], WHITE, 64, 9)
1081 |         draw_text(
1082 |             self.screen,
1083 |             "AV. EJÉRCITO × TRINIDAD MORÁN × CAYMA – AREQUIPA",
1084 |             self.fonts["tiny"],
1085 |             MUTED,
1086 |             64,
1087 |             31,
1088 |         )
1089 |
1090 |         tag = "SCT-ML PYTHON · SIMULACIÓN FUNCIONAL"
1091 |         tw = self.fonts["tiny"].size(tag)[0] + 20
1092 |         tag_rect = pygame.Rect(WIDTH - tw - 70, 16, tw, 22)
1093 |         draw_rounded_rect(self.screen, tag_rect, (8, 35, 48), 5, CYAN, 1)
1094 |         draw_text(self.screen, tag, self.fonts["tiny"], CYAN, tag_rect.x + 10, tag_rect.y + 6)
1095 |
1096 |         pygame.draw.circle(self.screen, GREEN_LIGHT, (WIDTH - 44, 27), 5)

```

```

1097 |         draw_text(self.screen, "ONLINE", self.fonts["tiny"], GREEN, WIDTH - 36, 22)
1098 |
1099 |     def draw_left_panel(self) -> None:
1100 |         pygame.draw.rect(self.screen, PANEL, (0, 52, LEFT_W, HEIGHT - 52))
1101 |         pygame.draw.line(self.screen, LINE, (LEFT_W, 52), (LEFT_W, HEIGHT), 1)
1102 |
1103 |         x = 14
1104 |         y = 66
1105 |         draw_text(self.screen, "ENTRADAS DEL SISTEMA", self.fonts["h1"], MUTED, x, y)
1106 |         pygame.draw.line(self.screen, LINE, (x, y + 22), (LEFT_W - x, y + 22), 1)
1107 |
1108 |         for b in self.buttons:
1109 |             b.active = (
1110 |                 (b.text == "Escenarios" and self.state.mode == "scenarios")
1111 |                 or (b.text == "Prueba libre" and self.state.mode == "free")
1112 |                 or (b.text == "Actual 43s" and self.state.mode == "actual")
1113 |             )
1114 |             b.draw(self.screen, self.fonts)
1115 |
1116 |         if self.state.mode in ("scenarios", "actual"):
1117 |             if self.state.mode == "actual":
1118 |                 note = pygame.Rect(14, 130, LEFT_W - 28, 46)
1119 |                 draw_rounded_rect(self.screen, note, (47, 31, 18), 8, AMBER, 1)
1120 |                 draw_text(
1121 |                     self.screen,
1122 |                     "Sistema actual 43s: modo rígido por escenarios, sin prueba libre.",
1123 |                     self.fonts["tiny"],
1124 |                     (248, 210, 138),
1125 |                     note.x + 9,
1126 |                     note.y + 8,
1127 |                     note.w - 18,
1128 |                 )
1129 |                 offset = 48
1130 |             else:
1131 |                 offset = 0
1132 |
1133 |             for i, b in enumerate(self.scenario_buttons):
1134 |                 b.rect.y = 140 + offset + i * 58
1135 |                 b.active = self.state.scenario_index == i
1136 |                 b.draw(self.screen, self.fonts)
1137 |
1138 |         else:
1139 |             draw_text(

```

```

1140 |         self.screen,
1141 |         "Haz clic en cada campo para cambiar su valor.",
1142 |         self.fonts["tiny"],
1143 |         MUTED,
1144 |         14,
1145 |         132,
1146 |         LEFT_W - 28,
1147 |     )
1148 |     for key, b in self.free_buttons.items():
1149 |         b.subtext = self.free_value_text(key)
1150 |         b.draw(self.screen, self.fonts)
1151 |
1152 |     # Log técnico
1153 |     log_rect = pygame.Rect(14, HEIGHT - 148, LEFT_W - 28, 132)
1154 |     draw_rounded_rect(self.screen, log_rect, PANEL_3, 8, LINE, 1)
1155 |     draw_text(self.screen, "DETALLE TÉCNICO", self.fonts["btn"], WHITE, log_rect.x + 10, log_rect.y + 8)
1156 |     log = self.tech_log()
1157 |     draw_text(self.screen, log, self.fonts["tiny"], MUTED, log_rect.x + 10, log_rect.y + 28, log_rect.w - 20, 2)
1158 |
1159 |     def draw_stage(self) -> None:
1160 |         self.stage_surface.fill((7, 15, 26))
1161 |         self.draw_roads(self.stage_surface)
1162 |         self.draw_sensors(self.stage_surface)
1163 |         self.draw_stop_lines(self.stage_surface)
1164 |
1165 |         for lane in ["a1", "a2", "a3"]:
1166 |             for v in self.state.vehicles[lane]:
1167 |                 v.draw(self.stage_surface)
1168 |
1169 |         self.draw_waiting_pedestrians(self.stage_surface)
1170 |         self.draw_crossing_pedestrians(self.stage_surface)
1171 |
1172 |         self.draw_semaphore(self.stage_surface, self.geo.stopA1 + 28, self.geo.hY + 30, str(self.state.sem["a1"]), "A1",
CYAN)
1173 |         self.draw_semaphore(self.stage_surface, self.geo.stopA2 - 28, self.geo.hY + self.geo.hH - 18,
str(self.state.sem["a2"]), "A2", VIOLET)
1174 |         self.draw_semaphore(self.stage_surface, self.geo.vX + self.geo.vW + 30, self.geo.stopA3 + 18,
str(self.state.sem["a3"]), "A3", GREEN)
1175 |         self.draw_semaphore(self.stage_surface, self.geo.vX + self.geo.vW + 65, self.geo.hY + self.geo.hH + 30, "green" if
self.state.sem["p"] else "red", "P", AMBER)
1176 |
1177 |         # HUD superior del escenario
1178 |         hud = "NODO VIAL – AV. EJÉRCITO x TRINIDAD MORÁN x CAYMA"

```

```

1179 |         draw_text(self.stage_surface, hud, self.fonts["tiny"], MUTED, 26, 18)
1180 |
1181 |         # Chip de modo
1182 |         chip_text = "SCT-ML ADAPTATIVO" if self.state.mode != "actual" else "SISTEMA ACTUAL 43S"
1183 |         chip_color = GREEN if self.state.mode != "actual" else RED
1184 |         chip_w = self.fonts["tiny"].size(chip_text)[0] + 18
1185 |         chip_rect = pygame.Rect(STAGE_W - chip_w - 18, 14, chip_w, 22)
1186 |         draw_rounded_rect(self.stage_surface, chip_rect, (5, 14, 24), 999, chip_color, 1)
1187 |         draw_text(self.stage_surface, chip_text, self.fonts["tiny"], chip_color, chip_rect.x + 9, chip_rect.y + 6)
1188 |
1189 |         # Leyenda
1190 |         legend = pygame.Rect(22, STAGE_H - 90, 300, 62)
1191 |         draw_rounded_rect(self.stage_surface, legend, (5, 12, 22), 8, LINE, 1)
1192 |         draw_text(self.stage_surface, "Leyenda", self.fonts["btn"], WHITE, legend.x + 10, legend.y + 8)
1193 |         draw_text(
1194 |             self.stage_surface,
1195 |             "A1/A2 Ejército · A3 Trinidad/Cayma\nC1-C3 cámaras · T1-T4 térmico · 🚶 espera peatonal",
1196 |             self.fonts["tiny"],
1197 |             SOFT,
1198 |             legend.x + 10,
1199 |             legend.y + 26,
1200 |             legend.w - 20,
1201 |         )
1202 |
1203 |         # Status inferior
1204 |         status = pygame.Rect(0, STAGE_H - 26, STAGE_W, 26)
1205 |         pygame.draw.rect(self.stage_surface, (0, 0, 0), status)
1206 |         phase = self.phase_label()
1207 |         next_txt = self.next_label()
1208 |         draw_text(self.stage_surface, f"FASE: {phase}", self.fonts["tiny"], SOFT, 18, STAGE_H - 20)
1209 |         ntw = self.fonts["tiny"].size(next_txt)[0]
1210 |         draw_text(self.stage_surface, next_txt, self.fonts["tiny"], SOFT, STAGE_W - ntw - 18, STAGE_H - 20)
1211 |
1212 |         # Volcar escenario central
1213 |         self.screen.blit(self.stage_surface, (STAGE_X, 52))
1214 |
1215 |     def draw_right_panel(self) -> None:
1216 |         x0 = WIDTH - RIGHT_W
1217 |         pygame.draw.rect(self.screen, PANEL, (x0, 52, RIGHT_W, HEIGHT - 52))
1218 |         pygame.draw.line(self.screen, LINE, (x0, 52), (x0, HEIGHT), 1)
1219 |
1220 |         y = 66
1221 |         margin = 14

```

```

1222 |         card_w = RIGHT_W - 2 * margin
1223 |
1224 |         y = self.draw_card_decision(x0 + margin, y, card_w)
1225 |         y = self.draw_card_chain(x0 + margin, y + 8, card_w)
1226 |         y = self.draw_card_ml(x0 + margin, y + 8, card_w)
1227 |         y = self.draw_card_bars(x0 + margin, y + 8, card_w)
1228 |         self.draw_card_reading(x0 + margin, y + 8, card_w)
1229 |
1230 |     def draw_card(self, x: int, y: int, w: int, h: int, title: str, border: Tuple[int, int, int] = LINE) -> pygame.Rect:
1231 |         rect = pygame.Rect(x, y, w, h)
1232 |         draw_rounded_rect(self.screen, rect, PANEL_2, 8, border, 1)
1233 |         draw_text(self.screen, title.upper(), self.fonts["btn"], MUTED, x + 10, y + 8)
1234 |         return rect
1235 |
1236 |     def draw_kv(self, x: int, y: int, label: str, value: str, value_color: Tuple[int, int, int] = WHITE) -> int:
1237 |         draw_text(self.screen, label, self.fonts["small"], MUTED, x, y)
1238 |         vw = self.fonts["small"].size(value)[0]
1239 |         draw_text(self.screen, value, self.fonts["small"], value_color, x + 250 - vw, y)
1240 |         return y + 21
1241 |
1242 |     def draw_card_decision(self, x: int, y: int, w: int) -> int:
1243 |         rect = self.draw_card(x, y, w, 108, "Decisión actual", GREEN if self.state.mode != "actual" else RED)
1244 |         yy = y + 30
1245 |         sem_txt = f"A1 {self.state.sem['a1']} · A2 {self.state.sem['a2']} · A3 {self.state.sem['a3']}"
1246 |         sensor_txt = "Confirmado" if self.state.thermal else "Discrepancia"
1247 |         sensor_color = GREEN if self.state.thermal else RED
1248 |         ped_txt = f"{len(self.state.ped_crossing)} cruzando · {self.state.ped_waiting} esperando"
1249 |         yy = self.draw_kv(x + 10, yy, "Fase", self.phase_label(), CYAN if self.state.mode != "actual" else RED)
1250 |         yy = self.draw_kv(x + 10, yy, "Semáforos", sem_txt, WHITE)
1251 |         yy = self.draw_kv(x + 10, yy, "Sensores T1-T4", sensor_txt, sensor_color)
1252 |         yy = self.draw_kv(x + 10, yy, "Peatones", ped_txt, AMBER)
1253 |         return rect.bottom
1254 |
1255 |     def draw_card_chain(self, x: int, y: int, w: int) -> int:
1256 |         rect = self.draw_card(x, y, w, 116, "Cadena metodológica")
1257 |         d1, d2, d3 = self.demands()
1258 |
1259 |         if self.state.mode == "actual":
1260 |             txt = (
1261 |                 f"1. Escenario fijo: condición predefinida.\n"
1262 |                 f"2. Temporización rígida: Av. Ejército 43s → A3 43s.\n"
1263 |                 f"3. Demanda observada: A1 {d1:.1f} · A2 {d2:.1f} · A3 {d3:.1f} · peatones {self.state.ped_waiting}.\n"
1264 |                 f"4. Limitación: la cola crítica espera hasta que termine el ciclo.\n"

```

```

1265 |         f"5. Fase actual: {self.phase_label()}."
1266 |     )
1267 |     else:
1268 |         txt = (
1269 |             f"1. Visión: A1 {d1:.1f} · A2 {d2:.1f} · A3 {d3:.1f} · peatones {self.state.ped_waiting}.\n"
1270 |             f"2. Térmico: {'confirma presencia' if self.state.thermal else 'no confirma: modo conservador'}.\n"
1271 |             f"3. RF/heurística: prioriza demanda y espera acumulada.\n"
1272 |             f"4. Seguridad: amarillo + todo rojo antes del cambio.\n"
1273 |             f"5. Salida: {self.phase_label()}."
1274 |         )
1275 |         draw_text(self.screen, txt, self.fonts["tiny"], SOFT, x + 10, y + 29, w - 20, 2)
1276 |         return rect.bottom
1277 |
1278 |     def draw_card_ml(self, x: int, y: int, w: int) -> int:
1279 |         rect = self.draw_card(x, y, w, 190, "Panel de inferencia ML", GREEN if self.state.mode != "actual" else RED)
1280 |         d1, d2, d3 = self.demands()
1281 |         ml = self.ml_inference()
1282 |
1283 |         yy = y + 30
1284 |         draw_text(self.screen, "ENTRADAS DETECTADAS", self.fonts["tiny"], GREEN if self.state.mode != "actual" else RED, x +
1285 |         10, yy)
1286 |         yy += 16
1287 |         draw_text(
1288 |             self.screen,
1289 |             f"A1={d1:.1f} · A2={d2:.1f} · A3={d3:.1f} · Peatones={self.state.ped_waiting} · Térmico={'OK' if
1290 |             self.state.thermal else 'NO'}",
1291 |             self.fonts["tiny"],
1292 |             SOFT,
1293 |             x + 10,
1294 |             yy,
1295 |             w - 20,
1296 |         )
1297 |         yy += 34
1298 |
1299 |         # Cajas de recomendación
1300 |         box_w = (w - 30) // 2
1301 |         box_h = 38
1302 |         items = [
1303 |             ("Fase sugerida", str(ml["phase"])),
1304 |             ("Verde sugerido", str(ml["green"])),
1305 |             ("Decisión final", str(ml["final"])),
1306 |             ("Seguridad", str(ml["safety"])),
1307 |         ]

```

```

1306 |         for idx, (lab, val) in enumerate(items):
1307 |             bx = x + 10 + (idx % 2) * (box_w + 10)
1308 |             by = yy + (idx // 2) * (box_h + 8)
1309 |             draw_rounded_rect(self.screen, pygame.Rect(bx, by, box_w, box_h), PANEL_3, 8, LINE, 1)
1310 |             draw_text(self.screen, lab.upper(), self.fonts["tiny"], MUTED, bx + 8, by + 5)
1311 |             color = GREEN if "Valid" in val or "Aplicar" in val else AMBER if "Conservador" in val else CYAN
1312 |             if self.state.mode == "actual":
1313 |                 color = RED if val in ("NO APLICAR", "Sin inferencia") else AMBER
1314 |             draw_text(self.screen, val, self.fonts["small"], color, bx + 8, by + 19, box_w - 16)
1315 |
1316 |         yy += 92
1317 |
1318 |         conf = float(ml["conf"])
1319 |         conf_txt = f"Confianza estimada: {int(conf * 100)}%" if conf > 0 else "Confianza estimada: -"
1320 |         draw_text(self.screen, conf_txt, self.fonts["tiny"], MUTED, x + 10, yy)
1321 |         yy += 15
1322 |         draw_bar(self.screen, x + 10, yy, w - 20, 8, conf, GREEN if self.state.mode != "actual" else RED)
1323 |         yy += 16
1324 |         draw_text(self.screen, str(ml["explain"]), self.fonts["tiny"], SOFT, x + 10, yy, w - 20, 2)
1325 |         return rect.bottom
1326 |
1327 |     def draw_card_bars(self, x: int, y: int, w: int) -> int:
1328 |         rect = self.draw_card(x, y, w, 126, "Cola/demanda equivalente")
1329 |         d1, d2, d3 = self.demands()
1330 |         ped = self.state.ped_waiting
1331 |         max_v = max(55, d1, d2, d3, ped)
1332 |
1333 |         rows = [
1334 |             ("A1 Ejército", d1, CYAN),
1335 |             ("A2 Ejército", d2, VIOLET),
1336 |             ("A3 Trinidad/Cayma", d3, GREEN),
1337 |             ("Peatones", ped, AMBER),
1338 |         ]
1339 |
1340 |         yy = y + 30
1341 |         for label, val, color in rows:
1342 |             draw_text(self.screen, label, self.fonts["tiny"], MUTED, x + 10, yy)
1343 |             val_txt = f"{val:.1f}" if isinstance(val, float) else str(val)
1344 |             vw = self.fonts["tiny"].size(val_txt)[0]
1345 |             draw_text(self.screen, val_txt, self.fonts["tiny"], WHITE, x + w - 10 - vw, yy)
1346 |             yy += 14
1347 |             draw_bar(self.screen, x + 10, yy, w - 20, 8, val / max_v, color)
1348 |             yy += 15

```

```

1349 |
1350 |         return rect.bottom
1351 |
1352 |     def draw_card_reading(self, x: int, y: int, w: int) -> int:
1353 |         h = HEIGHT - y - 16
1354 |         rect = self.draw_card(x, y, w, h, "Lectura académica")
1355 |         draw_text(self.screen, self.state.reading, self.fonts["tiny"], SOFT, x + 10, y + 30, w - 20, 3)
1356 |         return rect.bottom
1357 |
1358 |     # -----
1359 |     # Dibujo del escenario
1360 |     # -----
1361 |
1362 |     def draw_roads(self, s: pygame.Surface) -> None:
1363 |         # Fondo con cuadrícula
1364 |         s.fill((7, 15, 26))
1365 |         for gx in range(0, STAGE_W, 64):
1366 |             pygame.draw.line(s, (16, 34, 54), (gx, 0), (gx, STAGE_H), 1)
1367 |         for gy in range(0, STAGE_H, 64):
1368 |             pygame.draw.line(s, (16, 34, 54), (0, gy), (STAGE_W, gy), 1)
1369 |
1370 |         # Bloques urbanos
1371 |         block_color = (10, 21, 35)
1372 |         blocks = [
1373 |             (14, 60, self.geo.vX - 28, self.geo.hY - 70),
1374 |             (self.geo.vX + self.geo.vW + 10, 60, self.geo.cayX - self.geo.vX - self.geo.vW - 20, self.geo.hY - 70),
1375 |             (self.geo.cayX + self.geo.cayW + 10, 60, STAGE_W - self.geo.cayX - self.geo.cayW - 24, self.geo.hY - 70),
1376 |             (14, self.geo.hY + self.geo.hH + 12, self.geo.vX - 28, STAGE_H - self.geo.hY - self.geo.hH - 110),
1377 |             (self.geo.vX + self.geo.vW + 10, self.geo.hY + self.geo.hH + 12, self.geo.cayX - self.geo.vX - self.geo.vW - 20,
STAGE_H - self.geo.hY - self.geo.hH - 110),
1378 |             (self.geo.cayX + self.geo.cayW + 10, self.geo.hY + self.geo.hH + 12, STAGE_W - self.geo.cayX - self.geo.cayW -
24, STAGE_H - self.geo.hY - self.geo.hH - 110),
1379 |         ]
1380 |         for r in blocks:
1381 |             draw_rounded_rect(s, pygame.Rect(*r), block_color, 8, (31, 52, 77), 1)
1382 |
1383 |         # Vías
1384 |         road = (24, 43, 67)
1385 |         pygame.draw.rect(s, road, (0, self.geo.hY, STAGE_W, self.geo.hH))
1386 |         pygame.draw.rect(s, road, (self.geo.vX, 0, self.geo.vW, STAGE_H))
1387 |         pygame.draw.rect(s, road, (self.geo.cayX, 0, self.geo.cayW, STAGE_H))
1388 |
1389 |         # Bordes

```

```

1390 |     edge = (49, 86, 124)
1391 |     pygame.draw.line(s, edge, (0, self.geo.hY), (STAGE_W, self.geo.hY), 2)
1392 |     pygame.draw.line(s, edge, (0, self.geo.hY + self.geo.hH), (STAGE_W, self.geo.hY + self.geo.hH), 2)
1393 |     pygame.draw.line(s, edge, (self.geo.vX, 0), (self.geo.vX, STAGE_H), 2)
1394 |     pygame.draw.line(s, edge, (self.geo.vX + self.geo.vW, 0), (self.geo.vX + self.geo.vW, STAGE_H), 2)
1395 |     pygame.draw.line(s, edge, (self.geo.cayX, 0), (self.geo.cayX, STAGE_H), 2)
1396 |     pygame.draw.line(s, edge, (self.geo.cayX + self.geo.cayW, 0), (self.geo.cayX + self.geo.cayW, STAGE_H), 2)
1397 |
1398 |     # Separadores discontinuos
1399 |     for x in range(0, STAGE_W, 42):
1400 |         pygame.draw.line(s, (230, 200, 80), (x, self.geo.hMid), (x + 20, self.geo.hMid), 2)
1401 |     for y in range(0, STAGE_H, 42):
1402 |         pygame.draw.line(s, (120, 160, 210), (self.geo.vMid, y), (self.geo.vMid, y + 20), 1)
1403 |
1404 |     # Cebras
1405 |     for i in range(7):
1406 |         pygame.draw.rect(s, (245, 158, 11), (self.geo.vX + 6 + i * 12, self.geo.pedY - 16, 8, 32))
1407 |     for i in range(7):
1408 |         pygame.draw.rect(s, (245, 158, 11), (self.geo.vX - 3, self.geo.hY + 8 + i * 10, self.geo.vW + 6, 6))
1409 |
1410 |     # Nombres de vías
1411 |     draw_text(s, "AV. EJÉRCITO", self.fonts["small"], (120, 160, 200), 22, self.geo.hY + 8)
1412 |     draw_text(s, "AV. TRINIDAD MORÁN", self.fonts["tiny"], (120, 160, 200), self.geo.vX + 8, STAGE_H - 118)
1413 |     draw_text(s, "AV. CAYMA", self.fonts["tiny"], (120, 160, 200), self.geo.cayX + 8, 120)
1414 |
1415 |     def draw_sensors(self, s: pygame.Surface) -> None:
1416 |         # Cámara
1417 |         cam_pos = (70, self.geo.hY - 70)
1418 |         pygame.draw.line(s, CYAN, cam_pos, (self.geo.stopA1, self.geo.hY + 8), 1)
1419 |         pygame.draw.line(s, CYAN, cam_pos, (self.geo.vX + self.geo.vW, self.geo.hY + self.geo.hH), 1)
1420 |         draw_rounded_rect(s, pygame.Rect(cam_pos[0] - 24, cam_pos[1] - 12, 48, 28), (6, 13, 23), 8, CYAN, 1)
1421 |         pygame.draw.circle(s, CYAN, cam_pos, 5)
1422 |         draw_text(s, "CAM", self.fonts["tiny"], CYAN, cam_pos[0] - 12, cam_pos[1] + 20)
1423 |
1424 |         # Térmico
1425 |         th_color = GREEN_LIGHT if self.state.thermal else RED
1426 |         th_pos = (self.geo.stopA1 + 48, self.geo.hY - 26)
1427 |         pygame.draw.line(s, th_color, th_pos, (self.geo.vX, self.geo.hY + 8), 1)
1428 |         pygame.draw.line(s, th_color, th_pos, (self.geo.vX + self.geo.vW, self.geo.pedY + 18), 1)
1429 |         draw_rounded_rect(s, pygame.Rect(th_pos[0] - 16, th_pos[1] - 15, 32, 36), (6, 13, 23), 8, th_color, 1)
1430 |         pygame.draw.circle(s, th_color, th_pos, 5)
1431 |         draw_text(s, "TH", self.fonts["tiny"], th_color, th_pos[0] - 8, th_pos[1] + 24)
1432 |

```

```

1433 |     def draw_stop_lines(self, s: pygame.Surface) -> None:
1434 |         def line(color: Tuple[int, int, int], p1: Tuple[int, int], p2: Tuple[int, int]) -> None:
1435 |             pygame.draw.line(s, color, p1, p2, 5)
1436 |
1437 |         c1 = self.sem_color(str(self.state.sem["a1"]))
1438 |         c2 = self.sem_color(str(self.state.sem["a2"]))
1439 |         c3 = self.sem_color(str(self.state.sem["a3"]))
1440 |
1441 |         line(c1, (self.geo.stopA1, self.geo.hY + 4), (self.geo.stopA1, self.geo.hMid - 4))
1442 |         line(c2, (self.geo.stopA2, self.geo.hMid + 4), (self.geo.stopA2, self.geo.hY + self.geo.hH - 4))
1443 |         line(c3, (self.geo.vX + 4, self.geo.stopA3), (self.geo.vX + self.geo.vW - 4, self.geo.stopA3))
1444 |
1445 |     def sem_color(self, sem: str) -> Tuple[int, int, int]:
1446 |         if sem == "green":
1447 |             return GREEN_LIGHT
1448 |         if sem == "yellow":
1449 |             return YELLOW
1450 |         return RED
1451 |
1452 |     def draw_semaphore(
1453 |         self,
1454 |         surface: pygame.Surface,
1455 |         x: int,
1456 |         y: int,
1457 |         sem: str,
1458 |         label: str,
1459 |         border: Tuple[int, int, int],
1460 |     ) -> None:
1461 |         rect = pygame.Rect(x - 13, y - 28, 26, 54)
1462 |         draw_rounded_rect(surface, rect, (3, 8, 15), 8, border, 2)
1463 |
1464 |         colors = [("red", RED), ("yellow", YELLOW), ("green", GREEN_LIGHT)]
1465 |         for i, (name, color) in enumerate(colors):
1466 |             c = color if sem == name else (48, 56, 68)
1467 |             pygame.draw.circle(surface, c, (x, y - 17 + i * 17), 6)
1468 |
1469 |         draw_text(surface, label, self.fonts["tiny"], border, x - 8, y + 32)
1470 |
1471 |     def draw_waiting_pedestrians(self, s: pygame.Surface) -> None:
1472 |         n = min(self.state.ped_waiting, 30)
1473 |
1474 |         for i in range(n):
1475 |             side = i % 2

```

```

1476 |         k = i // 2
1477 |         col = k % 5
1478 |         row = k // 5
1479 |         if side:
1480 |             x = self.geo.vX + self.geo.vW + 20 + col * 10
1481 |         else:
1482 |             x = self.geo.vX - 20 - col * 10
1483 |             y = self.geo.hY + self.geo.hH + 24 + row * 12
1484 |             self.draw_person(s, x, y, AMBER, 0.75)
1485 |
1486 |             txt = f"{self.state.ped_waiting} / {max(10, round(self.state.ped_target * 0.8))}"
1487 |             draw_rounded_rect(s, pygame.Rect(self.geo.vX + 62, self.geo.hY - 28, 96, 22), (5, 12, 22), 6)
1488 |             draw_text(s, txt, self.fonts["small"], AMBER, self.geo.vX + 70, self.geo.hY - 24)
1489 |
1490 |     def draw_crossing_pedestrians(self, s: pygame.Surface) -> None:
1491 |         for p in self.state.ped_crossing:
1492 |             self.draw_person(s, int(p.x), int(p.y), AMBER, 1.0)
1493 |
1494 |     def draw_person(self, s: pygame.Surface, x: int, y: int, color: Tuple[int, int, int], scale: float = 1.0) -> None:
1495 |         r = max(2, int(3 * scale))
1496 |         pygame.draw.circle(s, color, (x, y - int(8 * scale)), r)
1497 |         pygame.draw.line(s, color, (x, y - int(5 * scale)), (x, y + int(6 * scale)), max(1, int(2 * scale)))
1498 |         pygame.draw.line(s, color, (x - int(5 * scale), y), (x + int(5 * scale), y), max(1, int(2 * scale)))
1499 |         pygame.draw.line(s, color, (x, y + int(6 * scale)), (x - int(4 * scale), y + int(13 * scale)), max(1, int(2 *
1500 | scale)))
1501 |         pygame.draw.line(s, color, (x, y + int(6 * scale)), (x + int(4 * scale), y + int(13 * scale)), max(1, int(2 *
1502 | scale)))
1503 |
1504 |     # -----
1505 |     # Textos de estado
1506 |     # -----
1507 |
1508 |     def phase_label(self) -> str:
1509 |         labels = {
1510 |             "main": "F1 · Av. Ejército",
1511 |             "mainY": "F1 · amarillo",
1512 |             "mainYped": "F1 · amarillo",
1513 |             "allredA3": "Todo rojo → A3",
1514 |             "a3": "F4 · A3 Trinidad/Cayma",
1515 |             "a3Y": "F4 · amarillo",
1516 |             "allredPed": "Todo rojo peatonal",
1517 |             "ped": "F3 · cruce peatonal",
1518 |             "allredMain": "Todo rojo → Av. Ejército",

```

```

1517 |         "fixedMain": "Actual · Av. Ejército 43s",
1518 |         "fixedMainY": "Actual · ámban",
1519 |         "fixedAllredA3": "Actual · todo rojo",
1520 |         "fixedA3": "Actual · A3 43s",
1521 |         "fixedA3Y": "Actual · ámban A3",
1522 |         "fixedAllredPed": "Actual · todo rojo peatonal",
1523 |         "fixedPed": "Actual · cruce peatonal",
1524 |         "fixedAllredMain": "Actual · todo rojo",
1525 |     }
1526 |     return labels.get(self.state.phase, self.state.phase)
1527 |
1528 | def next_label(self) -> str:
1529 |     if self.state.mode == "actual":
1530 |         total = self.fixed_phase_total()
1531 |         remain = max(0, math.ceil((total - self.state.phase_elapsed_ms) / 1000))
1532 |         return f"cambio fijo en: {remain}s"
1533 |
1534 |     if self.state.phase == "main":
1535 |         return "siguiente: A3 o peatón"
1536 |     if "Y" in self.state.phase:
1537 |         return "siguiente: todo rojo"
1538 |     if self.state.phase == "a3":
1539 |         return "siguiente: retorno/peatón"
1540 |     if self.state.phase == "ped":
1541 |         return "siguiente: Av. Ejército"
1542 |     return "siguiente: verde"
1543 |
1544 | def fixed_phase_total(self) -> int:
1545 |     ph = self.state.phase
1546 |     if ph in ("fixedMain", "fixedA3"):
1547 |         return FIXED_DUR["green"]
1548 |     if ph in ("fixedMainY", "fixedA3Y"):
1549 |         return FIXED_DUR["yellow"]
1550 |     if "Allred" in ph:
1551 |         return FIXED_DUR["allred"]
1552 |     if ph == "fixedPed":
1553 |         return FIXED_DUR["ped"]
1554 |     return FIXED_DUR["green"]
1555 |
1556 | def tech_log(self) -> str:
1557 |     d1, d2, d3 = self.demands()
1558 |     if self.state.mode == "actual":
1559 |         return (

```

```

1560 |         f"Modo: sistema actual fijo 43s\n"
1561 |         f"Estado: {self.state.phase}\n"
1562 |         f"Semáforos: A1={self.state.sem['a1']} A2={self.state.sem['a2']} A3={self.state.sem['a3']}
P={self.state.sem['p']}\n"
1563 |         f"Demandas equivalentes: A1={d1:.1f} A2={d2:.1f} A3={d3:.1f}\n"
1564 |         f"Regla fija: Ejército 43s → ámbar → todo rojo → A3 43s.\n"
1565 |         f"Sin prueba libre ni ajuste por demanda."
1566 |     )
1567 |
1568 |     return (
1569 |         f"Modo: SCT-ML adaptativo\n"
1570 |         f"Estado: {self.state.phase}\n"
1571 |         f"Semáforos: A1={self.state.sem['a1']} A2={self.state.sem['a2']} A3={self.state.sem['a3']}
P={self.state.sem['p']}\n"
1572 |         f"Demandas: A1={d1:.1f} A2={d2:.1f} A3={d3:.1f}\n"
1573 |         f"Peatones: esperando={self.state.ped_waiting}, cruzando={len(self.state.ped_crossing)}\n"
1574 |         f"Sensor térmico: {'confirmado' if self.state.thermal else 'discrepancia'}\n"
1575 |         f"Regla crítica: nadie cruza su línea de parada si no está en verde."
1576 |     )
1577 |
1578 |     # -----
1579 |     # Eventos
1580 |     # -----
1581 |
1582 |     def handle_click(self, pos: Tuple[int, int]) -> None:
1583 |         # Botones de modo
1584 |         for b in self.buttons:
1585 |             if b.contains(pos):
1586 |                 if b.text == "Escenarios":
1587 |                     self.reset_scenariio(self.state.scenario_index)
1588 |                 elif b.text == "Prueba libre":
1589 |                     self.state.mode = "free"
1590 |                 elif b.text == "Actual 43s":
1591 |                     self.run_actual_scenariio(self.state.scenario_index)
1592 |                 self.rebuild_ui()
1593 |             return
1594 |
1595 |     # Escenarios
1596 |     if self.state.mode in ("scenarios", "actual"):
1597 |         for i, b in enumerate(self.scenario_buttons):
1598 |             if b.contains(pos):
1599 |                 if self.state.mode == "actual":
1600 |                     self.run_actual_scenariio(i)

```

```

1601 |         else:
1602 |             self.reset_scenario(i)
1603 |             self.rebuild_ui()
1604 |             return
1605 |
1606 |     # Prueba libre
1607 |     if self.state.mode == "free":
1608 |         for key, b in self.free_buttons.items():
1609 |             if b.contains(pos):
1610 |                 if key == "run":
1611 |                     self.reset_free()
1612 |                 else:
1613 |                     self.cycle_free_option(key)
1614 |                     self.rebuild_ui()
1615 |                     return
1616 |
1617 |     def handle_key(self, key: int) -> None:
1618 |         if key == pygame.K_ESCAPE:
1619 |             pygame.quit()
1620 |             sys.exit(0)
1621 |         elif key == pygame.K_1:
1622 |             self.reset_scenario(0)
1623 |             self.rebuild_ui()
1624 |         elif key == pygame.K_2:
1625 |             self.reset_scenario(1)
1626 |             self.rebuild_ui()
1627 |         elif key == pygame.K_3:
1628 |             self.reset_scenario(2)
1629 |             self.rebuild_ui()
1630 |         elif key == pygame.K_4:
1631 |             self.reset_scenario(3)
1632 |             self.rebuild_ui()
1633 |         elif key == pygame.K_5:
1634 |             self.reset_scenario(4)
1635 |             self.rebuild_ui()
1636 |         elif key == pygame.K_6:
1637 |             self.reset_scenario(5)
1638 |             self.rebuild_ui()
1639 |         elif key == pygame.K_7:
1640 |             self.reset_scenario(6)
1641 |             self.rebuild_ui()
1642 |         elif key == pygame.K_a:
1643 |             self.run_actual_scenario(self.state.scenario_index)

```

```

1644 |         self.rebuild_ui()
1645 |     elif key == pygame.K_f:
1646 |         self.state.mode = "free"
1647 |         self.rebuild_ui()
1648 |     elif key == pygame.K_s:
1649 |         self.reset_scenario(self.state.scenario_index)
1650 |         self.rebuild_ui()
1651 |
1652 |     # -----
1653 |     # Loop
1654 |     # -----
1655 |
1656 |     def run(self) -> None:
1657 |         last = time.perf_counter()
1658 |         while True:
1659 |             now = time.perf_counter()
1660 |             dt_ms = (now - last) * 1000.0
1661 |             last = now
1662 |
1663 |             for event in pygame.event.get():
1664 |                 if event.type == pygame.QUIT:
1665 |                     pygame.quit()
1666 |                     sys.exit(0)
1667 |                 elif event.type == pygame.MOUSEBUTTONDOWN and event.button == 1:
1668 |                     self.handle_click(event.pos)
1669 |                 elif event.type == pygame.KEYDOWN:
1670 |                     self.handle_key(event.key)
1671 |
1672 |             self.update(dt_ms)
1673 |             self.draw()
1674 |             self.clock.tick(FPS)
1675 |
1676 |
1677 |     # =====
1678 |     # Ejecución
1679 |     # =====
1680 |
1681 |     def main() -> None:
1682 |         app = SCTMLApp()
1683 |         app.run()
1684 |
1685 |
1686 |     if __name__ == "__main__":

```

