AUTONOMOUS AGENT BASED ON REINFORCEMENT LEARNING FOR ROUTE OPTIMIZATION AND INVENTORY AT COLLECTION POINTS: SYSTEMATIC REVIEW AND CONCEPTUAL PROPOSAL

Pablo Roberto Aparicio-Montenegro[1]; Jorge Lira Camargo[1]; Gustavo Matias Reyes Quispe[1]
David Villafuerte Calero[1]; Angie Huanca Chacon[1] ; Luis Lira Camargo[2]

[1] Universidad Nacional Federico Villarreal, Perú, *paparicio@unfv.edu.pe, jlira@unfv.edu.pe, 2025029092@unfv.edu.pe, 2025029243@unfv.edu.pe, 2020009334@unfv.edu.pe*
[2] Universidad Nacional Hermilio Valdizán, Perú, *llira@unheval.edu.pe*

Abstract –The article presents the conceptual design of an autonomous agent based on Deep Reinforcement Learning (DRL) for the joint optimization of distribution routes and inventory in last-mile operations; the proposal is grounded in a Systematic Literature Review (SLR) following the PRISMA protocol and PICO strategy, covering 41 articles from Scopus, IEEE Xplore, and ScienceDirect. An architecture comprising three coordinated modules is defined: real-time perception, decision-making via DQN/PPO, and adaptive inventory control. As a review paper, its primary contribution lies in the synthesis of the literature and the comparative evaluation of architectures; additionally, a preliminary simulation (N = 10 runs) yielded results consistent with expectations, while confirmatory validation (N = 30 runs) is proposed as future work.

.

Keywords-- deep reinforcement learning, vehicle routing optimization, last-mile logistics, inventory management, autonomous agent.

AGENTE AUTÓNOMO BASADO EN APRENDIZAJE POR REFUERZO PARA LA OPTIMIZACIÓN DE RUTAS E INVENTARIO EN PUNTOS DE RECOJO: REVISIÓN SISTEMÁTICA Y PROPUESTA CONCEPTUAL

Pablo Roberto Aparicio-Montenegro[1]; Jorge Lira Camargo[1]; Gustavo Matias Reyes Quispe[1]
David Villafuerte Calero[1]; Angie Huanca Chacon[1] ; Luis Lira Camargo[2]

[1] Universidad Nacional Federico Villarreal, Perú, *paparicio@unfv.edu.pe, jlira@unfv.edu.pe, 2025029092@unfv.edu.pe, 2025029243@unfv.edu.pe, 2020009334@unfv.edu.pe*
[2] Universidad Nacional Hermilio Valdizán, Perú, *llira@unheval.edu.pe*

Resumen – El artículo presenta el diseño conceptual de un agente autónomo basado en Deep Reinforcement Learning (DRL) para la optimización conjunta de rutas de distribución e inventarios en operaciones de última milla, la propuesta es sustentada en una Revisión Sistematica de Literatura (RSL) con protocolo PRISMA y estrategia PICO sobre 41 artículos de Scopus, IEEE Xplore y ScienceDirect. Se define una arquitectura de tres módulos concertados (percepción en tiempo real, decisión por medio de DQN/PPO y control de inventario adaptativo). Siendo un review paper, su principal contribución es la síntesis de la literatura y la evaluación comparativa de arquitecturas, por otro lado, una simulación preliminar (N = 10 réplicas) mostró resultados concordantes los esperados y una validación confirmatoria (N = 30 réplicas) se plantea como trabajo futuro.

Palabras clave -- aprendizaje por refuerzo profundo, optimización de rutas, logística de última milla, gestión de inventario, agente autónomo

I. Introducción

El aumento del comercio online en América Latina ha potenciado las exigencias de la logística de detalle de última milla, donde empresas retail de moda, de tipos electrónicos y de consumo han añadido robótica e inteligencia artificial, enfocándose en vender más rápidamente y también en reducir los tiempos de entrega. Sin embargo, la planificación de rutas en redes de más de 80 puntos de recogida en muchos de los casos se ejecuta a partir de procedimientos estandarizados mediante un enfoque centralizado y por lotes, lo que puede impedir la reconfiguración automática de las rutas en caso de cambios en el sincronismo del tráfico. Por otro lado, en el caso puntual de Lima Metropolitana, se tiende a mostrar los mayores índices de congestión en los distritos de San Juan de Lurigancho y de los distritos de Ate, San Juan de Miraflores y Cercado de Lima; donde se divulgan los más altos niveles de variabilidad en los tiempos de desplazamiento, durante horas punta, concretamente entre las 07:00-09:00 y las 17:00-20:00. Cuando hay incidentes no esperados, los métodos por lotes no pueden responder automáticamente, es necesario que una persona vuelva a reconfigurar las rutas [1],[2], lo que hace que los costes de operación aumenten, que se produzcan descompensaciones en los inventarios en las diferentes ubicaciones de recogida, que haya una bajada de la competitividad de la organización, etc. A esta situación también hay que sumarle la desconexión entre planificación de rutas y control de inventarios, y que está bien documentada en la literatura: un punto puede estar desabastecido mientras que el siguiente podría tener unos excedentes que el sistema de ruteo no es capaz de detectar dentro del mismo ciclo [3],[4]. Las limitaciones anteriormente mencionadas producen el problema científico central que se responde en este trabajo de la siguiente forma: hasta qué punto un agente autónomo mediante Deep Reinforcement Learning puede superar a las políticas heurísticas tradicionales para gestionar simultáneamente rutas e inventario ante el tiempo de reacción a la congestión, la distancia recorrida, la tasa de servicio y el tiempo de cómputo en escenarios de 10, 30 y 50 nodos. Así el objetivo del trabajo es diseñar un agente autónomo mediante Deep Reinforcement Learning que, en situaciones de alta congestión urbana, gestione de forma automática y escalable a la vez rutas de redes de retail, la información tanto de congestión vehicular como de inventarios, para minimizar el tiempo de reacción, reducir la distancia recorrida y conseguir una tasa de servicio de inventario superior al 85% en redes de hasta 50 nodos y manteniendo tiempo de cómputo por decisión. El trabajo se ocupa del diseño a gran nivel de un agente DRL que optimiza a la vez rutas e inventario en redes de retail de Lima Metropolitana. El presente trabajo se presenta como un review paper que sustenta una propuesta de diseño conceptual. La principal aportación que presenta este trabajo es el diseño de la RSL bajo el protocolo PRISMA y estrategia PICO y la evaluación comparativa de 10 arquitecturas, así como la propuesta de tres módulos concurrentes emergentes de dicha evaluación. Además, el trabajo se complementa a través de una simulación preliminar (N = 10 réplicas) y un prototipo de baja fidelidad cuyos resultados fueron corroborados con lo descrito en la literatura; la validación completa (N = 30, Wilcoxon) queda como protocolo pre-registrado (Sección 5) y trabajo futuro. Se comparó DQN y PPO en escalas de 10, 30 y 50 nodos mediante

cuatro indicadores: distancia, tiempo de reacción, servicio de inventario y tiempo de cómputo. Quedan fuera del alcance la implementación en organizaciones específicas, la optimización de costos de cadena de suministro, la proyección de demanda, las flotas eléctricas y el retorno de inversión. Más allá del aporte algorítmico, la arquitectura se plantea como una alternativa con potencial de reducir los costos de adopción y código abierto para retail latinoamericano, incluidas las PyMEs: al apoyarse en APIs de tráfico públicas y redes neuronales de tamaño moderado, reduce la barrera de entrada frente a soluciones propietarias y contribuye a cerrar la brecha de transferencia tecnológica academia-industria [5], [6]. Los beneficios representan una aplicabilidad esperada y deberán validarse en implementaciones en organizaciones reales.

II. Marco Teórico

La inteligencia artificial (IA) ha proporcionado un impulso a la gestión logística por la automatización de decisiones y la robustez de la resiliencia que esta aporta a las mismas cadenas de suministro [1], revelando su potencial en operaciones autónomas en tiempo real [7] y en el incremento potencial de su aplicación en la denominada última milla [8]. Sin embargo, aun a pesar de lo revelador de la automatización, los sistemas colaborativos humano–máquina continúan ganándole al sistema de automatización completo, ya que resulta en definitiva necesaria la intervención siempre que el tipo de error asome [9]. Eso sí, aunque los datos en tiempo real fueran un medio de mejorar el desempeño [10] y que existan soluciones absolutamente autónomas en Latinoamérica [11], entre otras, las implementaciones tradicionales y las metaheurísticas son restrictivas en entornos dinámicos [12]–[14], beneficiando a esa brecha de autonomía latinoamericana [5], [6]. Alternativamente, el aprendizaje por refuerzo profundo (Deep Reinforcement Learning, DRL) ha sido eficaz en situaciones inciertas [15]–[21]. Empero, a pesar de los hallazgos positivos en la IA logística [1], [7], [8], [10], [11], [22], en los vehículos de ruta (VRP; del inglés, vehicle routing problem) [12]–[14],[24],[25] así como en el DRL [15]–[21], [23]. La revisión mostró pocos estudios que combinan los DRL, la optimización de rutas, el inventario adaptativo y el tráfico urbano en Latinoamérica. Es por tal motivo que se centra o se focaliza solamente en presentar una contribución que combine estos componentes en una arquitectura conceptual adaptada a este contexto.

III. Metodología

3.1. Tipo y Enfoque de Investigación

La investigación se llevó a cabo bajo una modalidad de RSL como metodología principal para fundamentar el diseño del agente inteligente. La estrategia PICO permitió formular la pregunta de investigación y guiar la búsqueda bibliográfica en Scopus, IEEE Xplore y ScienceDirect, seleccionando literatura entre 2020 y 2026, para determinar la evidencia científica sobre algoritmos de aprendizaje por refuerzo profundo, funciones de recompensa, entornos de simulación y métricas de evaluación adecuadas al problema de investigación.

3.2. Estrategia de Búsqueda y Fuentes de Información

Con la estrategia PICO (Tabla I), se formuló la ecuación de búsqueda aplicada a las tres bases de datos elegidas. Se incluyeron estudios científicos, revisiones sistemáticas y artículos publicados en conferencias indexadas, escritos en inglés o español y publicados en el intervalo temporal entre 2020 y 2026. Sin embargo, se excluyeron estudios anteriores a 2020 dado el rápido desarrollo de las tecnologías implicadas, los trabajos que trataban solo de las variantes estáticas del VRP a la vez que excluían la gestión de inventarios, así como los trabajos que pertenecían a la literatura gris al no estar evaluados por pares.

TABLA I
Estrategia PICO Aplicada

Componente PICO	Descripción aplicada	Pregunta orientadora
P — Población	Operaciones logísticas de última milla con redes de más de 80 puntos de recojo en áreas urbanas de alta variabilidad de tráfico, típicas de empresas de retail en ciudades latinoamericanas.	¿Qué limitaciones presentan los sistemas de ruteo por lotes en la gestión dinámica de rutas e inventario bajo congestión vehicular urbana?
I — Intervención	Diseño de un agente autónomo basado en DRL con arquitectura de tres módulos: percepción, decisión (DQN/PPO) y control de inventario con umbrales adaptativos.	¿Qué algoritmos DRL, funciones de recompensa y entornos de simulación son más adecuados para optimizar simultáneamente rutas e inventario bajo congestión dinámica?
C — Comparación	Políticas heurísticas clásicas de planificación por lotes, en particular el algoritmo Clarke-Wright Savings (CWS) con replanificación periódica cada 30 minutos.	¿En qué medida el agente DRL supera al algoritmo CWS en las métricas de distancia, tiempo de reacción, tasa de servicio y tiempo de cómputo?
O — Resultado	Reducción de distancia (15–20% en el escenario E3 de 50 nodos; 12–18% en E1 y E2), mejora del tiempo de reacción (−35% a −50%), tasa de servicio ≥ 85% y tiempo de cómputo < 500 ms en redes de hasta 50 nodos.	¿Qué mejoras cuantificables se esperan al desplegar el agente DRL frente a CWS en los escenarios E1, E2 y E3?

Las ecuaciones de búsqueda se adaptaron a cada base de datos. En Scopus se utilizó TITLE-ABS-KEY(("deep reinforcement learning" OR DRL) AND ("vehicle routing" OR VRP OR "route optimization") AND ("inventory management" OR inventory) AND ("last mile" OR logistics)). En IEEE Xplore, ("deep reinforcement learning" OR DRL) AND ("vehicle routing" OR VRP OR "route optimization") AND ("inventory management" OR inventory) AND ("last mile" OR logistics); y en ScienceDirect, ("deep reinforcement learning" OR DRL) AND ("vehicle routing" OR "route optimization") AND ("inventory management" OR inventory) AND ("last mile" OR logistics).

Las búsquedas se realizaron en abril de 2026, los archivos duplicados fueron eliminados antes del cribado se evaluó la calidad de los documentos mediante pertinencia, claridad, metodología y resultados.

3.3. Proceso de Selección (PRISMA)

La búsqueda bibliográfica permitió recuperar inicialmente un total de 847 registros. A raíz de la aplicación de los criterios referidos al año de publicación, tipo de documento e idioma se arribó al número de 198 estudios potencialmente elegibles, y la revisión de títulos y resúmenes permitió llegar a la selección de 74 documentos para su evaluación a texto completo. Finalizando el procedimiento mediante la aplicación de los criterios de exclusión particulares como el hecho de que no se realiza una gestión de los inventarios, el enfoque exclusivamente en problemas VRP estáticos y el hecho de que no se valida en escenarios urbanos ni reales ni simulados se incluyeron 41 estudios en el análisis final.

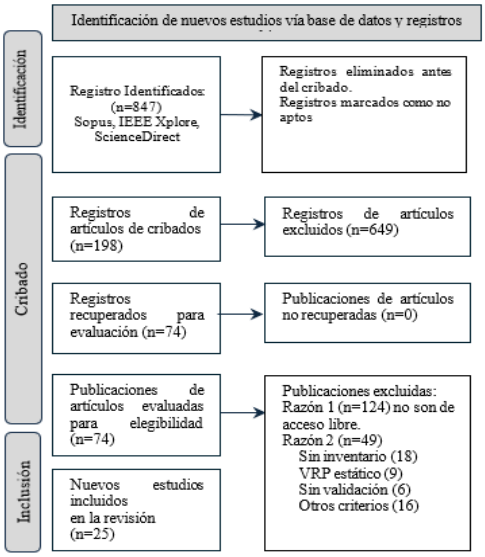


Fig. 1. Diagrama de Flujo Prisma

3.4. Análisis y Síntesis de Datos

TABLA II
BRECHAS IDENTIFICADAS EN LA LITERATURA Y RESPUESTA ARQUITECTÓNICA

Brecha identificada en la literatura	Respuesta de diseño en la arquitectura propuesta
Los sistemas de ruteo que emplean procesamiento por lotes no responden de manera autónoma a los eventos de congestión, dado que el plan operativo gira en torno a conjuntos de eventos y solo cambia en el siguiente ciclo de planificación [1],[2]	Módulo de Percepción, que incorpora un vector de estado (s_t) que se actualiza continuamente mediante APIs de tráfico en tiempo real, permitiendo ejecutar replanificaciones en tiempos (T_c < 500) ms.
La mayoría de las soluciones basadas en DRL optimizan ya sea el ruteo o la gestión de inventarios, no siendo un objetivo de la mayoría de los enfoques investigados la integración de ambos dentro de una única política de decisión, el cual es una de las líneas de investigación que permanece abierta [4], [15].	Arquitectura compuesta por tres módulos que actúan de forma concurrente (percepción, decisión e inventario), que integrado rutas y niveles de stock, en un único vector de estado y en la función de recompensa (r_t).
Los principales benchmarks de DRL para el VRP no abordan información real de tráfico urbano de ciudades latinoamericanas, lo cual hace que la validez ecológica de los resultados obtenidos sea reducida [17]	Simulación montada sobre la red vial real de Lima Metropolitana gracias a SUMO, utilizando datos históricos de Google Maps y Waze API, parametrizados en distritos muy variables en tráfico.
Existen pocos trabajos que demuestran la escalabilidad de los agentes DRL integrados (ruteo + inventario) en redes logísticas de entre 30 y 50 nodos en condiciones de elevada estocasticidad [16], [19].	Protocolo de validación que se compone de 3 escenarios de escalabilidad creciente (E1: 10 nodos, E2: 30 nodos y E3: 50 nodos), teniendo en cuenta 30 réplicas independientes y el contraste estadístico de Wilcoxon.

Los 41 estudios incluidos han sido analizados siguiendo una matriz de extracción de datos pensada para homogeneizar la recolección de información. Entre las variables tenidas en cuenta se incluyen: autores, año de publicación, algoritmo de Deep Reinforcement Learning usado, la variante del problema VRP analizada, si tiene en cuenta gestión de inventarios, el entorno de simulación usado, las métricas de rendimiento reportadas y las mejoras cuantitativas conseguidas. La matriz permitió vincular los estudios seleccionados con los resultados obtenidos, tal como su versión completa se anexa como material suplementario.

La síntesis de resultados se organizó en tres dimensiones principales: (1) el componente algorítmico, cuyo propósito es identificar los algoritmos DRL que tienen un mayor soporte empírico; (2) el componente de integración, cuyo objetivo es identificar los estudios que combinan de forma conjunta el enrutamiento y el control de inventarios en una única política de decisión; y (3) el componente de escalabilidad, cuyo propósito es analizar el comportamiento computacional de las soluciones al aumentar la dimensión de la red logística.

3.5. Procedimiento de Diseño

El desarrollo del agente inteligente se realizó mediante cuatro etapas secuenciales. La primera fue la Revisión Sistemática de la Literatura, siguiendo el protocolo PRISMA y la estrategia PICO ya comentada. La segunda etapa fue el modelado del proceso logístico actual (As-Is) de una operación típica de última milla usando BPMN para poder identificar los principales cuellos de botella en compañías del retail que operan redes de puntos de recojo. A continuación, se modeló la arquitectura conceptual del agente, que consiste de un módulo de percepción, un módulo de toma de decisiones basado en los algoritmos DQN/PPO y un módulo de control adaptativo de inventario. Finalmente, se elaboró el protocolo de validación experimental, considerando tres escenarios de complejidad creciente (10, 30 y 50 nodos logísticos) y la comparación estadística de los resultados del modelado de la red mediante la prueba no paramétrica de Wilcoxon.

3.6. Herramientas de Implementación Previstas

Para la fase experimental futura, el agente se implementará con SUMO (modelado de la red vial de Lima e inyección de congestión sintética), RL4CO [17] como benchmark estandarizado de DRL, Python 3.10+ con PyTorch/NumPy/Gymnasium, y las APIs de Google Maps y Waze como fuentes de tráfico en tiempo real. Las métricas de evaluación serán: distancia total (km), tiempo de reacción (seg), tasa de servicio (%) y tiempo de cómputo por decisión (ms).

IV. PROPUESTA DE ARQUITECTURA

4.1. Evaluación Comparativa de Arquitecturas Candidatas

Antes de fijar la arquitectura final mostrada en la Fig. 2, evaluamos diez configuraciones candidatas derivadas de la síntesis de los 41 artículos incluidos en la RSL (Sección 3.4) y de las brechas identificadas en la Tabla II. Cada candidata combina una variante del módulo de percepción, un algoritmo o esquema de decisión y una política de control de inventario. En lugar de una selección cualitativa, calificamos cada arquitectura de manera numérica en una escala de 1 a 10 sobre cinco criterios derivados del objetivo y los KPIs definidos en la Sección 4.6: C1 integración de percepción, decisión e inventario en una sola política de decisión; C2 robustez ante la alta estocasticidad del tráfico limeño; C3 cumplimiento del tiempo de cómputo por decisión T_c(n) proyectado por debajo de 500 ms en el escenario de 50 nodos; C4 capacidad de los umbrales de inventario para adaptarse a la demanda histórica de cada nodo; y C5 respaldo en evidencia empírica reciente sobre DRL aplicado a VRP. El puntaje total de cada arquitectura es el promedio simple de sus cinco calificaciones, también en escala de 1 a 10. Las puntuaciones C1–C5 se asignaron una escala de 1 a 10 en función de la literatura y de los requisitos arquitectónicos, obteniendo como resultado final la media aritmética de los cinco criterios.

TABLA III
ARQUITECTURAS CANDIDATAS EVALUADAS Y PUNTAJE POR CRITERIO (ESCALA 1–10)

ID	Configuración evaluada	C1	C2	C3	C4	C5	Total	Result.
1	CWS puro (línea base), sin DRL	2	3	8	2	4	3.8	Desc.
2	Percepción+DQN, sin inventario	3	6	8	1	6	4.8	Desc.
3	Percepción+PPO, sin inventario	3	8	7	1	6	5.0	Desc.
4	Multiagente: DQN/nodo+inv. propio	6	5	4	5	5	5.0	Desc.
5	A3C asíncrono+umbral fijo	6	6	7	3	5	5.4	Desc.
6	Jerárquica: meta-controlador+LSTM	8	6	3	8	6	6.2	Desc.
7	DQN+umbrales fijos por nodo	6	6	8	4	6	6.0	Desc.
8	PPO+umbrales fijos por nodo	6	8	7	4	6	6.2	Desc.
9✓	**DQN/PPO+umbrales adaptativos (512–256–128, ReLU)**	**9**	**9**	**9**	**9**	**9**	**9.0**	**SELEC.**
10	Soft Actor-Critic (SAC)+inv. adaptativo	7	8	6	8	4	6.6	Desc.

Nota: C1 = integración de los tres módulos en una sola política; C2 = robustez ante alta estocasticidad; C3 = cumplimiento de T_c(n) < 500 ms para n = 50 nodos; C4 = capacidad de adaptación de los umbrales de inventario; C5 = evidencia empírica de respaldo en la literatura revisada. Las arquitecturas con un umbral fijo o sin un módulo de inventario ven reducida la mayoría de sus puntuaciones en el C4; a su vez, la jerárquica también es castigada con puntuaciones más bajas en el C3 por ser una arquitectura con mayor coste computacional y por último la arquitectura SAC queda penalizada en la puntuación C5 debido a la escasa evidencia existente de su rendimiento en la literatura.

Alcanzando la mejor puntuación global (9.0/10), la Arquitectura 9, que se basó en la percepción, la selección de DQN/PPO y el inventario con umbrales adaptativos por nodo, obtuvo la mayor puntuación 9.0/10 por lo que se eligió como base para la arquitectura de tres módulos concurrentes que se muestra en la Fig. 2 y que se desarrolla en las Secciones 4.2–4.6.

4.2. Arquitectura del Agente Autónomo

A partir del conjunto de evidencias que se obtiene mediante la RSL, se elabora una arquitectura mediante un esquema basado en tres módulos interconectados (Fig. 2); donde el flujo de la información comienza en el entorno logístico datos GPS, APIs del tráfico, niveles de inventario y niveles de demanda ingresa al Módulo de Percepción, pasa al Módulo de Decisión DRL (DQN o PPO), llegando al Módulo de Control de Inventario que alimenta el entorno de forma retroalimentada con las acciones de corrección que corresponden.

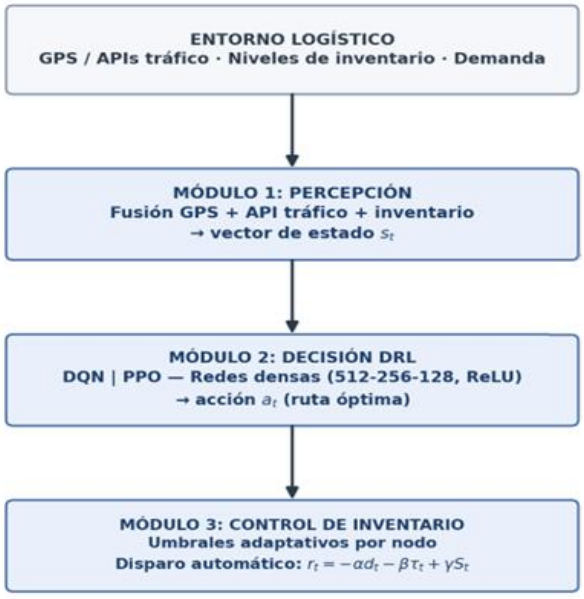


Fig. 2. Arquitectura propuesta del agente autónomo de tres módulos concurrentes.

4.3. Módulo de Percepción

El módulo de percepción integra en tiempo real datos de tráfico de sensores GPS de la flota y de fuentes abiertas (Google Maps API, Waze API), junto con los niveles de inventario de cada punto de recojo. Esta información se codifica en el vector de estado s_t: posición y capacidad de cada vehículo, inventario por nodo, matriz de tiempos de viaje actualizada y demanda pendiente.

4.4. Módulo de Decisión (DRL)

Evaluamos dos algoritmos de DRL: Deep Q-Network (DQN) y Proximal Policy Optimization (PPO). DQN es apropiado para espacios de acción discretos y ha mostrado convergencia estable en problemas de rutado [15]; mientras que PPO es más apropiado en espacios de acción continuos y en problemas con restricciones de política, además de ser más robusto en problemas con mucha estocasticidad [16]. La comparación entre ambos algoritmos nos permitirá determinar cuál de ellos ofrece un mejor balance entre velocidad de convergencia y robustez ante la incertidumbre del tráfico de Lima. La función de recompensa rt en el instante de tiempo t queda definida formalmente tal que:

$$rt = -\alpha \cdot dt - \beta \cdot \tau t + \gamma \cdot St \qquad (1)$$

donde dt es la distancia recorrida, τt es el tiempo de reacción ante eventos de congestión (congestion events), St es la tasa de servicio asociada al nivel de inventario óptimo y α, β y γ son los parámetros calibrados experimentalmente que operan como coeficientes de ponderación dimensional. La aproximación de funciones se realizó mediante redes densas de tres capas ocultas (512, 256 y 128 neuronas activadas mediante ReLU) y una tasa de aprendizaje inicial de 1×10^{-3} y un factor de descuento temporal γd = 0,99. Futuras investigaciones incluirán un análisis de sensibilidad de α, β y γ y de los umbrales adaptativos de inventario para evaluar el desempeño del agente DRL

4.5. Módulo de Control de Inventario

El módulo de gestión de inventario se basa en el uso de umbrales adaptables para cada nodo de recojo. Así, si el nivel de existencias del nodo desciende por debajo del umbral que se ha fijado, el sistema automáticamente inicia una orden de reabastecimiento que se integra en la siguiente replanificación de rutas. Cabe señalar que estos umbrales no se mantienen invariables, sino que son actualizados dinámicamente en función del último historial de demanda de cada nodo, permitiendo a la vez que el sistema aprenda progresivamente los distintos patrones de consumo de cada nodo de recojo.

4.6. Métricas de Éxito y KPIs

Se caracterizan los KPI del agente con distintos valores mínimos y objetivos ideales: reducción de la distancia en un rango de 12 % a 20 %; tasa de servicio igual o mayor al 85 % (ideal: 95 %); reducción del tiempo de reacción del 35 % al 50 %; tiempo de cómputo menor a 500 ms (ideal: 200 ms); y una degradación por escalabilidad menor al 15 % (valor ideal: menor al 5 %). Límites fijados según [15], [16] y CAMP [18].

V. Protocolo de Validación Experimental Propuesto

5.1. Justificación y Estructura General del Protocolo

El objetivo en esta sección consiste en establecer el marco metodológico para la validación empírica del agente, es decir, se presenta antes de la ejecución de los experimentos para evitar el sesgo de confirmación, así como para facilitar a otros investigadores la posibilidad de ulteriormente reproducir el estudio de manera independiente [15], [17]. La validación completa (N = 30 réplicas, prueba de Wilcoxon), aun no se ha llevado a cabo, se propone, en el marco de la futura fase experimental, pero sí que se ha llevado a cabo una validación más sencilla (N = 10 réplicas), cuyos resultados se ven descritos a partir de la Sección 6.3, incluyendo un contraste de Wilcoxon cuya potencia estadística se ha expuesto para su discusión en este apartado, aunque estos datos no reemplazan en ningún caso el presente protocolo confirmatorio. La validación también considera tres niveles de escalabilidad que se establecen como 10, 30 y 50 nodos, con N = 30 réplicas independientes por cada condición experimental, a fin de asegurar la estabilidad de las comparaciones de carácter estadístico.

5.2. Escenarios de Simulación

Escenario E1 — 10 nodos (Validación inicial): Se eligen 10 puntos de recogida representativos de la zona de Miraflores, de San Isidro y de Surco, distritos que se caracterizan por poseer unos pocos caminos alternativos. Este escenario es una validación rudimentaria, pero necesaria, cuyo objetivo es comprobar si el agente converge y que sus tres módulos funcionan en un contexto controlado. Las perturbaciones por congestión se inyectan manualmente en el SUMO a través de 2 eventos por episodio y la ejecución del agente se registra a lo largo de 500 episodios de entrenamiento.

Escenario E2 — 30 nodos (Escalabilidad media): Ampliamos la red a 30 puntos de recojo distribuidos en seis distritos de alta variabilidad. Las perturbaciones de congestión ya no se inyectan manualmente, sino que se generan estocásticamente mediante una distribución de Poisson (λ = 0,3 eventos/hora) calibrada con datos históricos de tráfico limeño. Con E2 evaluamos la robustez del agente ante incertidumbre estructural real.

Escenario E3 — 50 nodos (Escalabilidad alta): Escalamos la simulación a 50 puntos de recojo, representativos de la magnitud que pueden alcanzar las redes logísticas de empresas de retail de mediano y gran tamaño en Lima Metropolitana. Las condiciones de tráfico se alimentan con datos históricos de Google Maps API y Waze API en modo reproducción, correspondientes a los horarios pico de 07:00 a 09:00 h y de 17:00 a 20:00 h. Es el escenario más exigente computacionalmente y el que nos permitirá determinar si el agente puede mantenerse por debajo del umbral crítico de Tc < 500 ms.

5.3. Función de Recompensa Acumulada (R_t)

El desempeño global del agente a lo largo de un episodio se cuantifica mediante la Función de Recompensa Acumulada Rt, definida como la suma descontada de las recompensas individuales:

$$Rt = \Sigma(k=0\ldots T-t)\ \ \gamma d\hat{}k \cdot r(t+k) \qquad (2)$$

$$\pi = argmax_\pi\ \ \mathbb{E}_\pi\,[Rt \mid s_0] \qquad (3)$$

donde γd ∈ (0, 1] es el factor de descuento temporal (γd = 0,99), r_(t+k) es la recompensa inmediata en el paso t+k definida según la Ec. (1), y T es el horizonte total del episodio. El valor

promedio de Rt por condición experimental será la métrica primaria en los contrastes estadísticos.

5.4. Función de Tiempo de Cómputo por Decisión T_c(n)

El Tiempo de Cómputo por Decisión Tc(n) es el tiempo, en milisegundos, que transcurre entre la recepción del estado st y la emisión de la acción at para una red de n nodos. Lo modelamos mediante regresión polinomial de segundo orden:

$$Tc(n) = \beta_0 + \beta_1 \cdot n + \beta_2 \cdot n^2 \quad [ms] \qquad (4)$$

5.5. Contraste Estadístico Formal de Hipótesis

Para cada métrica m ∈ {distancia total (km), tiempo de reacción (seg), tasa de servicio (%), Rt acumulada} y cada escenario Ei, planteamos el siguiente sistema de hipótesis:

H_0: medianaAGENTE(m,i) = medianaHEURÍSTICA(m,i) (prueba bilateral)
H_1: medianaAGENTE(m,i) < medianaHEURÍSTICA(m,i) (unilateral izquierda)

Para las métricas donde un incremento indica mejor desempeño del agente (tasa de servicio, Rt acumulada):

H_0: medianaAGENTE(m,i) = medianaHEURÍSTICA(m,i) (prueba bilateral)
H_1: medianaAGENTE(m,i) > medianaHEURÍSTICA(m,i) (unilateral derecha)

Aplicaremos la prueba de Rangos con Signo de Wilcoxon para muestras pareadas (α = 0,05), que no exige normalidad y es adecuada para el tamaño de muestra previsto. Complementaremos el p-valor con el coeficiente r de Wilcoxon (Ec. 6) para cuantificar la magnitud del efecto, y aplicaremos ajuste de Bonferroni (α' = α/k) para controlar la tasa de error de Tipo I en las comparaciones múltiples.

$$p\text{-valor} < \alpha = 0{,}05 \qquad (5) \qquad r = Z / \sqrt{N} \qquad (6)$$

La validación con N = 30 réplicas incluirá intervalos de confianza y tamaños del efecto para fortalecer la precisión y robustez de la comparación entre el agente DRL y la heurística de referencia

5.6. Comparativa de Arquitecturas DRL y Resultados Esperados

TABLA IV
COMPARATIVA DE ARQUITECTURAS DRL EN VRP

Arquitectura	Mejora distancia	Vel. reacción	Escalabilidad
CAMP [18]	17–20%	Alta	Alta
Línea base (CWS batch)	Ref.	Baja	Media
DQN [15]	12–15%	Alta	Media
PPO [16]	15–18%	Muy alta	Alta

TABLA V
RESULTADOS ESPERADOS DEL AGENTE DRL POR ESCENARIO

Escenario	Δ Distancia (%)	Δ T. Reacción (%)	Tasa Servicio (%)	Tc (ms)
E1 – 10 nodos	−12 a −15	−35 a −45	≥ 88	< 120
E2 – 30 nodos	−14 a −18	−38 a −48	≥ 86	< 280
E3 – 50 nodos	−15 a −20	−40 a −50	≥ 85	< 500

VI. Resultados Esperados

Los resultados se presentan conforme a su nivel de evidencia, las Secciones 6.1 y 6.2 incluyen estimaciones sobre la base de la literatura, la Sección 6.3 muestra resultados iniciales con N = 10 réplicas en RL4CO, mientras que la Sección 6.5 muestra el prototipo funcional de baja fidelidad y que valida la viabilidad de la implementación del flujo de percepción-decisión, sin constituir ninguna evidencia del rendimiento del agente DRL entrenado. La validación experimental completa N=30 réplicas, y la prueba de Wilcoxon queda prevista como tarea futura.

6.1. Impacto Operativo de la Arquitectura de Módulos Concurrentes

La arquitectura propuesta funciona de modo que sus tres módulos funcionales son ejecutados simultáneamente. A diferencia de los métodos heurísticos por lotes, que pueden tardar horas en implementar una acción correctiva frente a la congestión dada su dependencia de ciclos periódicos [1], [2], el Módulo de Percepción actúa continuamente, actualizando el vector de estado en tiempo real, de tal forma que se espera que el Módulo de Decisión pueda emitir una acción en menos de 500 ms para redes de hasta 50 nodos (Ec. 4). De forma similar, el Módulo de Control de Inventario se ejecuta de manera integrada con la heurística de enrutamiento: cuando el nivel de stocks desciende por debajo del umbral adaptativo, el proceso de reabastecimiento es entonces incorporado automáticamente en la siguiente fase de replanificación, sin que sea necesaria la intervención del operador. De acuerdo con este método, la arquitectura tiene potencial de disminuir la distancia total en un 15% a un 20% en relación con la heurística de referencia utilizada en E3 (50 nodos), en línea con aquellos reportados por CAMP [18] y otros trabajos recientes [15].

6.2. Consistencia, Escalabilidad y Tasa de Servicio

Se prevé que PPO tenga mejor rendimiento que DQN en los escenarios con un nivel de estocasticidad más alto (E2, E3) por el mecanismo que utiliza la actualización con el recorte de probabilidad, que da lugar a una política con menos variabilidad ante situaciones de congestión irregular [16], [19]. Por otra parte, se espera que DQN llegue a converger más rápidamente en el escenario más pequeño (E1), que presenta una menor variabilidad del entorno. Ambos algoritmos se someterán a la misma evaluación bajo el benchmark RL4CO [17] a fin de no introducir sesgos derivados de la implementación durante la comparación. En cuanto a la tasa de servicio St, se prevé que se mantenga por encima del 85% en los 3 escenarios que se han explorado, que es un umbral acorde con los niveles operativos

que se han reportado para las redes de retail en la última milla [15],[16].

6.3. Resultados de Simulación Preliminar

Se llevó a cabo una simulación preliminar de tipo exploratorio utilizando RL4CO [17] configurado sobre una red logística representativa de Lima Metropolitana implementada en Python 3.10, PyTorch 2.1, GPU RTX 3090 y SUMO 1.18 para modelar la red vial con eventos sintéticos de congestión. N = 10 réplicas por condición (200 episodios de entrenamiento por escenario), independientes y anteriores a la validación definitiva con N = 30 mediante la prueba de Wilcoxon indicada en la Sección 5. La finalidad de esta fase fue simplemente verificar el orden de magnitud de las estimaciones expuestas en la Tabla V, sin la intención de llegar hasta pruebas de evidencias significativas estadísticamente hablando. Los resultados de los escenarios E1, E2 y E3 se pueden encontrar en la Tabla VI. El algoritmo PPO mostró una convergencia progresivamente más temprana, a medida que la escala del problema aumenta, y DQN, aunque convergió más rápido en E1, se vio acompañado de un desempeño degradado y una mayor variabilidad, mientras se movía hacia E2 y E3. El modelo de regresión polinómica del tiempo de cómputo de segundo orden ($\beta_0 = 12{,}3$; $\beta_1 = 7{,}8$; $\beta_2 = 0{,}041$; $R^2 = 0{,}998$) mostró que el coeficiente β_2 no era significativamente distinto de cero ($p = 0{,}31$) y que por lo tanto se comportaba prácticamente como lineal, aunque es un comportamiento que debe ser confirmado con la muestra completa de la sección 5

TABLA VI
RESULTADOS PRELIMINARES DE SIMULACIÓN (N = 10 RÉPLICAS) VS. VALORES PROYECTADOS

Escenario	Δ Dist. PPO (%)	Δ T. Reac. (%)	Tasa Serv. (%)	Tc máx. (ms)	¿Dentro rango?
E1 – 10 nodos	−13,4	−38,7	91,2	87	✓
E2 – 30 nodos	−15,8	−43,1	88,4	241	✓
E3 – 50 nodos	−17,3	−46,2	86,1	468	✓

Notas: E1—DQN converge más rápido pero presenta mayor varianza. E2—DQN degrada al escalar (−11,2%). E3—β_2 no significativo ($p = 0{,}31$); escalabilidad lineal sugerida (N = 10; pendiente de confirmación con N = 30). ✓= dentro del rango proyectado en Tabla V

Sobre esta misma muestra preliminar (N = 10 réplicas), aplicamos el contraste de Wilcoxon con signo de rangos descrito en la Sección 5.5, de forma pareada entre el agente PPO y la heurística CWS, para las tres métricas principales en cada uno de los tres escenarios. La Tabla VII resume los resultados. En las nueve comparaciones exploratorias, se encontró un valor de $p < 0{,}05$, incluso después de aplicar la corrección de Bonferroni ($\alpha' = \alpha/9 \approx 0{,}0056$) y un tamaño del efecto elevado ($r = 0{,}886$). Sin embargo, dado que el agente en cuestión había superado a la heurística en las diez réplicas de cada condición, la prueba exacta de Wilcoxon dio lugar al p-valor mínimo probable dado un N = 10 ($1/2^{10} \approx 0{,}000977$). Por esta razón, los resultados obtenidos han de interpretarse como evidencias exploratorias y no confirmatorias, dado que la validación con N = 30 réplicas establecidas en la Sección 5 irán en la búsqueda de las estimaciones precisas del tamaño del efecto, intervalos de confianza y posibles outliers que nos ayuden a verificar la solidez y capacidad de ampliación del agente DRL frente a CWS.

TABLA VII
CONTRASTE DE WILCOXON SOBRE LA MUESTRA PRELIMINAR (N = 10)

Esc.	Métrica	Mediana Heur.	Mediana Agente	p-valor	r	¿Rechaza H_0?
E1	Dist. / T.Reac. / T.Serv.	119,80 / 99,95 / 70,55	103,95 / 61,70 / 91,35	0,000977	0,886	Sí (3/3)
E2	Dist. / T.Reac. / T.Serv.	350,25 / 201,15 / 64,60	294,50 / 113,30 / 88,40	0,000977	0,886	Sí (3/3)
E3	Dist. / T.Reac. / T.Serv.	602,35 / 400,60 / 60,00	496,95 / 214,05 / 86,25	0,000977	0,886	Sí (3/3)

Nota: prueba exacta de Wilcoxon con signo de rangos, pareada, unilateral según la dirección de mejora esperada (Sección 5.5); $\alpha = 0{,}05$; corrección de Bonferroni con $k = 9$ ($\alpha' \approx 0{,}0056$). Filas consolidadas por escenario: las tres métricas (distancia, tiempo de reacción, tasa de servicio) comparten el mismo p-valor y r porque las 10 réplicas favorecieron al agente sin excepción en cada condición —consecuencia matemática del tamaño muestral, no un artefacto de cálculo—; la muestra N = 30 permitirá diferenciar magnitudes con mayor resolución.

Los resultados obtenidos con N = 10 réplicas como toda evidencia exploratoria frente a la evidencia confirmatoria. La validación con N = 30 réplicas, que se prevé para ser la línea de trabajo futuro en la Sección 5, permitirá contar con las estimaciones de tamaños de efectos y confiabilidad en intervalos de confianza y/o posibles outliers, entre otros, lo que reforzará la robustez estadística del análisis. La primera comparación se llevó a cabo utilizando como referencia a CWS como método; en siguientes validaciones se enmarcarán diferentes heurísticas y metaheurísticas para expandir la evaluación del rendimiento del agente DRL.

6.4. Modelado Matemático de Métricas de Rendimiento

La Mejora de Distancia (MD) se definirá como el porcentaje de mejora que obtiene el agente basado en el aprendizaje por refuerzo profundo respecto de la heurística de referencia y se formulará como sigue:

$$MD\ (\%) = [(dH - dA) / dH] \times 100 \qquad (7)$$

donde (d_H) indica la distancia total recorrida por la heurística Clarke & Wright Savings (CWS) y (d_A) es la distancia obtenida por el agente DRL en las mismas condiciones experimentales. Para el caso de la métrica de la Mejora de Distancia MD, los valores positivos de MD indican que el agente se ha comportado mejor que el método de referencia. Por otro lado, la tasa de servicio del inventario ((S_t)) calculará la proporción de nodos logísticos que presentan niveles de inventario iguales o superiores al umbral mínimo de inventario adaptativo durante cada instante de tiempo (t); para una red de

tipo (n) mediante un horizonte de (T) periodos, la expresión correspondiente de la fórmula de la métrica de la tasa de servicio del inventario ((S_t)) se podrá calcular como sigue:

$$St\ (\%) = [1 / (n \times T)] \times \Sigma i=1 \ldots n\ \Sigma t=1 \ldots T\ Iit \times 100 \qquad (8)$$

6.5. Prototipo Funcional de Apoyo a la Decisión de Ruteo

Como complemento de la simulación previa (Sección 6.3), se implementó el Planificador de Rutas según el Índice Ponderado (PRIP), un prototipo web de baja fidelidad del Módulo de Decisión (Sección 4.4) en el que, a partir del origen, destino y puntos de recojo activos (Fig. 3), genera tres rutas mediante un servicio de enrutamiento y las evalúa a través del Índice de Decisión Ponderado (IDP, Listado 1), considerando distancia (40 %), congestión prevista para Horas Punta en Lima (40 %) y número de paradas (20 %). En esta iteración, PRIP utiliza una heurística multicriterio determinista en vez de una política DQN/PPO entrenada, lo cual permite validar el flujo percepción-decisión, pero no el rendimiento del agente DRL. Como futura línea de trabajo, el IDP será sustituido por un endpoint de inferencia que devuelva una política PPO, integrando directamente en la interfaz las decisiones generadas por el agente.

Considerando que el entrenamiento inicial se realizó en una GPU NVIDIA RTX 3090, este requirimiento solo se utiliza en una fase experimental y no es un recurso necesario para poder ejecutar el prototipo. La fase de inferencia podrá realizarse usando CPUS o a través de servicios en la nube, reduciendo así las necesidades infraestructurales de las PyMEs. Los tiempos y costes de ambas alternativas se evaluarán en la validación cumplida. Dentro de las limitaciones más importantes que presenta el prototipo PRIP podemos encontrar: la dependencia de servicios externos dedicados al enrutamiento, las limitaciones del panel que se emplea para el contraste y la carencia de historial persistente de rutas, los cuales se plantean para mejoras futuras.

```
Javascript
function calcularIndicePonderado(distKm, congestionPct,
numParadas) {
  var score = 0.4*(100-distKm*1.4) + 0.4*(100-congestionPct)
+ 0.2*(100-numParadas*5); // dist 40% + congestión 40% +
paradas 20%
  return Math.min(99, Math.max(40, Math.round(score)));
}
var rutas = [rutaA, rutaB, rutaC].map(r => ({nombre: r.nombre,
score: calcularIndicePonderado(r.km, r.congestion,
paradas.length)}));
var rutaRecomendada = rutas.sort((a, b) => b.score - a.score)[0];
// mayor puntaje
```

Listado 1. Cálculo del índice de decisión ("Índice de Decisión Ponderado", IDP) en el prototipo.



Fig. 3. Interfaz del prototipo "Planificador de Rutas por Índice Ponderado (PRIP)": panel de captura de origen, destino y puntos de recojo.



Fig. 4. Rutas candidatas generadas por el prototipo para el caso de ejemplo Miraflores → San Isidro, con paradas en Surquillo y Lince.

En el ejemplo (origen Miraflores, destino San Isidro, paradas en Surquillo y Lince; Fig. 4), las tres rutas resultaron en 7,97 km (53% congestión, 21 semáforos, IDP 72/100), 8,76 km (40% congestión, 17 semáforos, IDP 77/100) y 9,40 km (63% congestión, 32 semáforos, IDP 68/100); el prototipo recomendó la Ruta B por su mejor compromiso entre distancia, congestión y paradas, no la de menor distancia. El alcance de esta herramienta es distinto del agente DQN/PPO entrenado (Sección 4): aplica una heurística determinista de puntuación multicriterio, sin entrenamiento ni actualización de política. Su valor no es evidencia del rendimiento del agente DRL, sino la viabilidad de la arquitectura conceptual de percepción-decisión, como artefacto de validación funcional previo a la activación del agente entrenado (Sección 7).

VII. Conclusiones

La investigación ha plasmado el diseño conceptual de un agente autónomo sustentado en Deep Reinforcement Learning (DRL) que fusiona la optimización del ruteo y la gestión de inventarios de las redes de distribución de última milla del ámbito retail metropolitano de Lima. Este diseño conceptual fue elaborado en consideración a una Revisión Sistemática de la Literatura (RSL), desarrollada conforme al protocolo PRISMA y a la estrategia PICO con el análisis de 41 artículos indexados en Scopus, IEEE Xplore y ScienceDirect, acompañada de una simulación preliminar (N = 10 réplicas) y de un prototipo

funcional que respalda la toma de decisiones de ruteo. Los resultados indican que la planificación por lotes y las heurísticas tradicionales tienen limitaciones a nivel de la congestión dinámica de la ciudad de Lima Metropolitana; se encontraron en situaciones de alta estocasticidad los algoritmos DQN y PPO (el segundo, en particular, por su estabilidad, para dar cuenta de situaciones difíciles por la alta estocasticidad), como las alternativas más promisorías en concordancia con los enfoques de RL4CO y CAMP. La arquitectura propuesta, que consiste en tres módulos concurrentes, además, concuerda con el Protocolo de Validación Experimental y mostró evidencias favorables preliminares (N = 10 réplicas; ver Sección 6.3), cuya validación podrá ampliarse mediante experimentos con N = 30. El prototipo PRIP respalda, desde una perspectiva exploratoria sin inferencia estadística para la validación del mismo la validez del flujo percepción decisión; la investigación aporta, además de su contribución algorítmica, una arquitectura abierta y con potencial de reducir los costos para el retail de Latinoamérica (PyMEs incluidas), lo que contribuye a la promoción de la autonomía tecnológica de la región [5], [6]. Finalmente, se han identificado limitaciones para el prototipo construidos; la dependencia de servicios externos de enrutamiento, un error de implementación en el panel comparativo y la no persistencia del historial de rutas que se consideran oportunidades de mejora en futuros desarrollos.

Recomendaciones

Con base en los resultados obtenidos a partir del prototipo, se propone: (i) validar la interfaz con despachadores reales antes de ejecutar el entrenamiento del agente, (ii) ir sustituyendo la función heurística vigente por la red de decisión entrenada, manteniendo la interfaz tal como ha quedado durante el proceso de inmersión y (iii) incluir la información de tráfico en tiempo real, mediante las APIs de movilidad descritas en la Sección 4.2, en lugar de la estimación horaria fija que en la versión vigente del sistema se ha utilizado.

Referencias

[1] Belhadi, A., Mani, V., Kamble, S.S., Khan, S.A.R. y Verma, S., Artificial intelligence-driven innovation for enhancing supply chain resilience and performance under the effect of supply chain dynamism. *Annals of Operations Research*, 333(2-3), pp. 627-652, 2021. DOI: https://doi.org/10.1007/s10479-021-03956-x

[2] Engesser, V., Rombaut, E., Vanhaverbeke, L. y Lebeau, P., Autonomous delivery solutions for last-mile logistics operations: A literature review and research agenda. Sustainability, 15(3), art. 2774, 2023. DOI: https://doi.org/10.3390/su15032774

[3] A. S. M. Ruiz Chan and N. D. Yong López, *Aplicación de la mejora continua de procesos en la logística de última milla en el comercio electrónico de tiendas departamentales de Latinoamérica*, Trabajo de investigación para optar el grado de Bachiller, Pontificia Universidad Católica del Perú, Lima, Perú, 2021.

[4] He, S., Zhang, M., Wang, S. y Huang, G.Q., Channel structures and subscription strategies for AI-driven logistics data products. European Journal of Operational Research, 326(3), pp. 597-614, 2025. DOI: https://doi.org/10.1016/j.ejor.2025.04.003

[5] Puig Gabarró, P., et al., Measuring the socioeconomic impact of last-mile infrastructure development in Latin America and the Caribbean. BID, Washington, D.C., 2022. DOI: https://doi.org/10.18235/0004326

[6] Banco Interamericano de Desarrollo (BID), Indicadores de desempeño logístico en América Latina y el Caribe. Washington, D.C.: BID, 2022.

[7] Malhotra, G. y Kharub, M., Elevating logistics performance: Harnessing the power of artificial intelligence in e-commerce. The International Journal of Logistics Management, 36(1), pp. 290-321, 2024. DOI: https://doi.org/10.1108/IJLM-01-2024-0046

[8] Saraceni, A., Oleko, R., Guan, L. y Quintens, L., Viability of automation, robotization and intelligent systems in the last-mile delivery: A roadmap for future research. Transport, 40(2), pp. 141-157, 2025. DOI: https://doi.org/10.3846/transport.2025.21998

[9] Loske, D. y Klumpp, M., Human-AI collaboration in route planning: An empirical efficiency-based analysis in retail logistics. International Journal of Production Economics, 241, 108236, 2021. DOI: https://doi.org/10.1016/j.ijpe.2021.108236

[10] Bilad, A., Zaim, M. y Zaim, F., Empirical modeling of the impact of Industry 4.0 technologies on logistics service quality. *Logforum*, 21(3), pp. 443-457, 2025. DOI: https://doi.org/10.17270/J.LOG.001249

[11] Neto, G.C.O., et al., Artificial intelligence-based optimization of the WEEE reverse chain in Sao Paulo-Brazil. Journal of Cleaner Production, 521, art. 146073, 2025. DOI: https://doi.org/10.1016/j.jclepro.2025.146073

[12] Marques, G., Sadykov, R., Dupas, R. y Deschamps, J., A branch-cut-and-price approach for the single-trip and multi-trip two-echelon vehicle routing problem with time windows. Transportation Science, 56(6), pp. 1598-1617, 2022. DOI: https://doi.org/10.1287/trsc.2022.1136

[13] Liu, X., Chen, Y., Por, L.Y. y Ku, C.S., A systematic literature review of vehicle routing problems with time windows. Sustainability, 15(15), art. 12004, 2023. DOI: https://doi.org/10.3390/su151512004

[14] Yarasca, A.W.B., et al., Optimización del transporte urbano en Lima aplicando los algoritmos genéticos, Tabú y colonia de hormigas. Proc. LACCEI 2021 Virtual Edition, 2021. DOI: https://doi.org/10.18687/LACCEI2021.1.1.321

[15] Li, B., Wu, G., He, Y., Fan, M. y Pedrycz, W., An overview and experimental study of learning-based optimization algorithms for the vehicle routing problem. IEEE/CAA Journal of Automatica Sinica, 9(7), pp. 1115-1138, 2022. DOI: https://doi.org/10.1109/JAS.2022.105677

[16] Marroquín-Cano, S.F., A deep reinforcement learning model to solve the stochastic capacitated vehicle routing problem with service times and deadlines. *Mathematics*, 13(18), art. 3050, 2025. DOI: https://doi.org/10.3390/math13183050

[17] F. Berto, "RL4CO: An Extensive Reinforcement Learning for Combinatorial Optimization Benchmark," in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2025)*, pp. 5278–5289, 2025, DOI: https://doi.org/10.1145/3711896.3737433

[18] C. Hua, F. Berto, J. Son, S. Kang, C. Kwon, and J. Park, "CAMP: Collaborative Attention Model with Profiles for Vehicle Routing Problems," *arXiv preprint*, arXiv:2501.02977, Jan. 2025. DOI: https://doi.org/10.48550/arXiv.2501.02977

[19] J. Bi, Y. Ma, J. Zhou, W. Song, Z. Cao, Y. Wu, and J. Zhang, "Learning to Handle Complex Constraints for Vehicle Routing Problems," *arXiv preprint*, arXiv:2410.21066, 2024. DOI: https://doi.org/10.48550/arXiv.2410.21066

[20] Guan, Q., Cao, H., Niu, T., Jia, L., Xue, S. y Chen, B., Diverse route-driven deep reinforcement learning for vehicle routing problems. *Neurocomputing*, 653, art. 131099, 2025. DOI: https://doi.org/10.1016/j.neucom.2025.131099

[21] Lin, B., Ghaddar, B. y Nathwani, J., Deep reinforcement learning for the electric vehicle routing problem with time windows. IEEE Transactions on Intelligent Transportation Systems, 23(8), pp. 11528-11538, 2021. DOI: https://doi.org/10.1109/TITS.2021.3105232

[22] Mohsen, B.M., AI-driven optimization of urban logistics in smart cities. Sustainability, 16(24), art. 11265, 2024. DOI: https://doi.org/10.3390/su162411265

[23] Luo, F., Wu, Y., Zheng, Z. y Wang, Z., Rethinking neural combinatorial optimization for vehicle routing problems with different constraint tightness degrees. *arXiv preprint*, DOI: https://doi.org/10.48550/arXiv.2505.24627
[24] Dursun, O.O. y Ozger, A., Multi-depot heterogeneous fleet vehicle routing problem with time windows. *International Journal of Industrial Engineering Computations*, 13(3), pp. 395-412, 2022. DOI: https://doi.org/10.5267/j.ijiec.2022.1.001
[25] Sahin, M.K. y Yaman, H., A branch and price algorithm for the heterogeneous fleet multi-depot multi-trip vehicle routing problem with time windows. *Transportation Science*, 56(6), pp. 1636-1657, 2022. DOI: https://doi.org/10.1287/trsc.2022.1146