# Enrutamiento inteligente de tickets de soporte mediante aprendizaje por refuerzo profundo: cuando la generalización importa más que la profundidad

Barboza Gonzales, José Luis
Ruiz Crisóstomo, Paul Vladimir
Salas Vizcarra, Óscar
Tenorio Huarancca, Diego Omar
Yufra Chambilla, Mario
*Sección C — Grupo 3, Curso de Aprendizaje por Refuerzo*

***Resumen*—En este trabajo se formula el enrutamiento de incidentes de una mesa de ayuda tributaria real como un proceso de decisión de Markov (MDP) y se resuelve con métodos tabulares, aproximación lineal y aprendizaje por refuerzo profundo (DQN, Double DQN y actor–crítico A2C). El entorno de simulación se calibra íntegramente con 177 940 incidentes reales (2021–2026): las llegadas se modelan como un proceso de Poisson no homogéneo simulado por *thinning*, los tiempos de servicio se muestrean de las distribuciones empíricas por grupo, el arranque en régimen se fija mediante la ley de Little y los objetivos de SLA se definen como el percentil 75 histórico por tipo. Con un espacio de estados de $8{,}3 \times 10^{15}$ configuraciones, la tabla Q colapsa (cobertura ≈0,01 % en evaluación), lo que motiva la aproximación de funciones. Bajo un protocolo de réplicas múltiples con intervalos de confianza del 95 % y pruebas de significancia cuyas conclusiones sobreviven a la corrección de Holm–Bonferroni, el aproximador lineal (46→15) supera a la heurística histórica de despacho con significancia estadística, mientras que el DQN profundo (46→128→128→15) queda por debajo debido al sesgo de maximización, que se mide directamente como la diferencia entre el valor $Q$ predicho y el retorno realmente obtenido. Double DQN reduce dicho sesgo con significancia ($p = 0{,}016$), lo que confirma el mecanismo, aunque sin revertir el déficit de retorno. Se concluye que, en problemas de despacho con señal moderada y ruido de cola pesada, la ventaja proviene de la capacidad de generalizar y no de la profundidad de la red, y que el protocolo estadístico con réplicas es imprescindible para no extraer conclusiones espurias.**

***Index Terms*—aprendizaje por refuerzo profundo, DQN, Double DQN, enrutamiento de tickets, simulación calibrada, sesgo de maximización**

## I. Introducción

Las mesas de ayuda de gran escala enrutan diariamente cientos de incidentes hacia decenas de grupos resolutores; una asignación incorrecta produce devoluciones, demoras e incumplimientos de acuerdos de nivel de servicio (SLA). En el sistema tributario que se estudia en este trabajo, 78 grupos atendieron 177 940 incidentes entre enero de 2021 y julio de 2026, con tiempos de resolución de cola pesada (mediana global de 46 horas frente a una media de 679) y un enrutamiento real de tipo muchos-a-muchos: la heurística "grupo modal por subcategoría" solo replica el 77,6 % de las asignaciones históricas. Es decir, ni siquiera el sistema humano se comporta como un clasificador determinista, y cualquier método automático debe convivir con esa ambigüedad.

El aprendizaje por refuerzo (RL) [1] ofrece un marco natural para este problema secuencial: cada asignación modifica las cargas de las colas y, con ello, las condiciones de las decisiones futuras. Un clasificador supervisado que aprende "a qué grupo fue este ticket" ignora ese acoplamiento; un agente de RL, en cambio, puede aprender a desviar tickets cuando la cola preferida se congestiona. Sin embargo, la literatura reciente advierte que los métodos profundos de RL son sensibles a semillas e hiperparámetros, y que sin réplicas y pruebas estadísticas los resultados pueden ser irreproducibles [2].

El objetivo de este trabajo es doble: (i) construir un entorno de simulación íntegramente calibrado con datos reales, con todos los supuestos declarados; y (ii) comparar, bajo un protocolo estadístico riguroso, una cadena de métodos donde cada técnica aparece porque la anterior fracasa: métodos tabulares (Q-learning [3], SARSA [4]), aproximación lineal, DQN [5], Double DQN [6] y actor–crítico A2C [7], [8].

El resto del documento se organiza así: la Sección II revisa los trabajos relacionados y su relación con la presente investigación; la Sección III presenta el análisis exploratorio de los datos, define el MDP, el entorno calibrado y los algoritmos; la Sección IV detalla la implementación y la reproducibilidad; la Sección V presenta los resultados con su análisis estadístico; la Sección VI discute por qué la generalización domina sobre la profundidad; y la Sección VII concluye.

## II. Trabajos relacionados

### *II-A. Asignación de tickets como clasificación supervisada*

La línea dominante en la industria trata el despacho de tickets como un problema de clasificación de texto. Mandal *et al.* [9] desarrollaron para una mesa de ayuda corporativa un sistema cognitivo de asignación de correos basado en técnicas de ensamble sobre el asunto y la descripción del ticket, alcanzando una precisión comparable a la de los despachadores humanos e incorporando umbrales de confianza para derivar al humano los casos dudosos. Este resultado apoya a la presente investigación en un punto esencial: demuestra empíricamente

que la categoría y el texto de un ticket contienen señal suficiente para predecir el grupo resolutor correcto, lo que justifica que el tipo del ticket sea la componente principal del estado del agente aquí propuesto. Al mismo tiempo, delimita lo que la clasificación no puede hacer: la decisión es estática, ticket a ticket, y no considera el estado del sistema en el momento de decidir.

En una escala mayor, DeepTriage [10] es el sistema de recomendación de equipos receptores desplegado en Microsoft Azure desde 2017: ante cada incidente de la nube sugiere el equipo que debería atenderlo, lidiando con miles de equipos, clases fuertemente desbalanceadas y una distribución que cambia con el tiempo. Su relevancia para este trabajo es doble. Primero, valida el problema a escala industrial: el enrutamiento erróneo de incidentes es lo bastante costoso como para justificar sistemas automáticos en producción. Segundo, sus autores documentan que los datos históricos de enrutamiento son ruidosos y no estacionarios — exactamente el fenómeno que se cuantifica en la Sección III — lo que respalda la decisión metodológica de evaluar con réplicas y pruebas estadísticas en lugar de corridas únicas. La diferencia de fondo se mantiene: DeepTriage optimiza el acierto de la clasificación respecto del histórico, mientras que aquí se optimiza una recompensa operativa (cumplimiento de SLA y congestión) que el histórico no maximiza necesariamente.

### *II-B. Aprendizaje por refuerzo para despacho y control de colas*

El antecedente más directo de la formulación aquí adoptada es DeepRM, de Mao *et al.* [11], que tradujo la gestión de recursos de un clúster (empaquetar trabajos con demandas múltiples) a un MDP cuyo estado codifica las ocupaciones actuales del sistema, y lo resolvió con una política entrenada por gradiente de política, obteniendo un desempeño comparable al de heurísticas especializadas y mejor adaptación a condiciones cambiantes. De DeepRM se toman dos decisiones de diseño: incluir las cargas de las colas como parte del estado observado — sin ellas el agente no puede reaccionar a la congestión — y comparar siempre contra heurísticas fuertes del dominio, no solo contra políticas triviales. La diferencia es que DeepRM se evalúa sobre cargas de trabajo sintéticas, mientras que en este trabajo cada componente del simulador (llegadas, servicios, compatibilidades, objetivos) se calibra con los datos crudos de la operación real.

En el plano teórico, Walton y Xu [12] revisan el aprendizaje en redes estocásticas y sistemas de colas, donde el estado incluye las ocupaciones y la política optimiza métricas agregadas de largo plazo, y analizan las condiciones de estabilidad y exploración de estos sistemas. Esta literatura sitúa formalmente el problema del despacho de tickets como un problema de control de colas y sustenta la intuición central de la formulación: una buena política de despacho debe ser función de la ocupación del sistema, no solo del trabajo entrante. La característica de "carga normalizada por régimen" del estado propuesto es la materialización directa de esa idea.

### *II-C. Métodos profundos de valor y sus patologías*

La cadena algorítmica evaluada proviene de la línea DQN: la repetición de experiencia se remonta a Lin [13], y Mnih *et al.* [5] la combinaron con una red objetivo para lograr control a nivel humano en Atari, estableciendo la receta estándar que aquí se implementa (replay, red objetivo, pérdida de Huber). Sobre esa base, van Hasselt *et al.* [6] demostraron que el operador $\max$ de la actualización de Q-learning produce sobreestimación sistemática de los valores ($\mathbb{E}[\max_a Q] \geq \max_a \mathbb{E}[Q]$) y propusieron Double DQN, que desacopla la selección de la evaluación de acciones. Esta línea aporta a la presente investigación la hipótesis mecanística central: si el DQN fracasa en un entorno ruidoso, el sospechoso natural es el sesgo de maximización. La contribución de este trabajo al respecto es medir ese sesgo directamente en un problema aplicado — como la diferencia $Q(s_0) - G_0$ entre lo predicho y lo obtenido — en lugar de inferirlo.

Finalmente, Henderson *et al.* [2] documentaron que los resultados publicados de RL profundo pueden depender de la semilla, la implementación y hasta la escala de recompensa, y recomendaron réplicas múltiples, intervalos de confianza y pruebas de significancia. Ese artículo define el protocolo experimental completo de la Sección IV. Su influencia no es cosmética: como se verá en la Sección V, una sola semilla afortunada del A2C habría parecido competitiva, y sin las pruebas de significancia la conclusión principal del trabajo habría sido otra.

## III. METODOLOGÍA

### *III-A. Conjunto de datos y análisis exploratorio*

La fuente de datos es la exportación completa del sistema de gestión de incidentes de la mesa de ayuda tributaria: 177 940 incidentes únicos entre enero de 2021 y julio de 2026, con 15 columnas que incluyen los cuatro *timestamps* del ciclo de vida (inicio, asignación, solución, cierre), el grupo resolutor (78 grupos), el servicio (5), la subcategoría (109), el analista, el estatus, la referencia a un Problema Padre y los textos de asunto y descripción. De ellos, 156 353 cuentan con solución registrada y constituyen la base del análisis descriptivo. Los 15 grupos de mayor volumen concentran el 96,7 % de los tickets (serán las colas del MDP) y las 20 subcategorías más frecuentes cubren el 84 % (serán los tipos, más una clase OTRAS).

La Tabla I muestra la evolución anual. Se aprecia una historia operativa clara: en 2023–2024 el volumen creció un 46 % y el sistema se congestionó (la mediana de resolución se cuadruplicó de 24,6 a 98,2 horas y el SLA cayó a su mínimo de 68,1 %); en 2025–2026 el servicio se recuperó. La columna p90 revela además una trampa estadística que conviene declarar: el p90 de 2026 (431 horas) no indica que los casos difíciles hayan desaparecido, sino que los casos lentos recientes *aún no se resuelven* y por tanto no entran en el cálculo (censura por la derecha). El SLA de 86,5 % de 2026 está inflado por esta razón, y esta no estacionariedad motiva la partición temporal declarada como trabajo futuro.

Tabla I
DISTRIBUCIÓN ANUAL DE LOS INCIDENTES RESUELTOS. SLA MEDIDO CONTRA EL OBJETIVO P75 HISTÓRICO DE CADA TIPO.

| Año | Resueltos | Mediana (h) | p90 (h) | SLA |
|---|---|---|---|---|
| 2021 | 21 799 | 42,3 | 1 729 | 71,5 % |
| 2022 | 17 030 | 24,6 | 1 729 | 75,9 % |
| 2023 | 25 522 | 73,1 | 2 933 | 69,6 % |
| 2024 | 38 296 | 98,2 | 2 422 | 68,1 % |
| 2025 | 36 608 | 24,4 | 1 159 | 82,2 % |
| 2026 † | 17 098 | 21,2 | 431 | 86,5 % |

†Hasta julio; mediana, p90 y SLA inflados por censura por la derecha.

Tabla II
LOS DIEZ GRUPOS RESOLUTORES DE MAYOR VOLUMEN (2021–2026). MEDIANA SOBRE TICKETS RESUELTOS; SLA CONTRA EL OBJETIVO P75 POR TIPO.

| Grupo | Tickets | % | Med. (h) | SLA |
|---|---|---|---|---|
| DAU_TRIBU_CPE_RECAUDACION | 95 867 | 53,9 | 64,7 | 78,0 % |
| DAU_TRIB_RUC_COBRANZA_FISCA | 27 394 | 15,4 | 20,9 | 80,0 % |
| DAU_Mesa_Ofimatica | 17 459 | 9,8 | 5,6 | 85,0 % |
| Desarrollo_TRIB_Mantenimiento | 7 728 | 4,3 | 2 022 | 36,4 % |
| DAU_Mesa_Contact | 7 560 | 4,2 | 30,8 | 84,3 % |
| DAU_Soporte Informático | 4 887 | 2,7 | 23,7 | 71,9 % |
| Desarrollo_TRIB_Registro_Canales | 2 087 | 1,2 | 1 272 | 29,8 % |
| Mesa1-Problemas-SIRE | 1 695 | 1,0 | 1 516 | 21,0 % |
| Desarrollo_TRIB_Reg.Electronicos | 1 613 | 0,9 | 1 737 | 20,9 % |
| Desarrollo_TRIB_CPE | 1 363 | 0,8 | 1 616 | 47,1 % |
| Resto (68 grupos) | 10 287 | 5,8 | — | — |

La Tabla II desglosa los diez grupos de mayor volumen. La estructura es fuertemente asimétrica: una sola cola (DAU_TRIBU_CPE_RECAUDACION) concentra el 53,9 % de todo el sistema y actúa como cola generalista de facto — atiende cuatro de los cinco servicios y sus tiempos son bimodales, con un p25 de 7,7 horas (cuadres y conciliaciones diarias) frente a un p99 de 9 694 horas (≈13 meses). La tabla muestra también dónde vive realmente el incumplimiento: no en el volumen masivo (la cola gigante cumple 78,0 % y las mesas de primera línea superan el 80 %), sino en los grupos de desarrollo y gestión de problemas, con SLA de 20–47 % y medianas de 1 300–2 000 horas, porque reciben los defectos de software y escalamientos que tardan meses. Para el agente esto implica que el mismo tipo de ticket tiene probabilidades de éxito radicalmente distintas según la cola elegida: esa es la señal que puede explotar, y también la fuente del *reward hacking* que obligó a endurecer el entorno durante el desarrollo (Sección III-D).

La Tabla III completa el panorama por servicio de negocio. Aquí la variación es suave (71–77 %), lo que confirma que la dificultad no se explica por la línea de negocio sino por el par tipo–grupo: el servicio CPE, con la mediana más alta (148,7 horas), es el que más depende de los grupos de desarrollo. Nótese que el SLA global ronda el 75 % *por construcción*: al definir el objetivo de cada tipo como el percentil 75 de su propio tiempo histórico (regla declarada en III-D), el sistema histórico cumple aproximadamente el 75 % por definición. Por eso, en todo este trabajo los porcentajes de SLA se leen de forma relativa (¿mejora o empeora respecto de la heurística?) y nunca como una nota absoluta.

Tabla III
DISTRIBUCIÓN POR SERVICIO DE NEGOCIO. SLA CONTRA EL OBJETIVO P75 POR TIPO.

| Servicio | Tickets | % | Med. (h) | SLA |
|---|---|---|---|---|
| MIGE IGV–SIRE | 45 292 | 25,5 | 93,9 | 74,5 % |
| RUC REGISTRO | 42 755 | 24,0 | 23,2 | 76,8 % |
| Comprobantes de Pago Electr. (CPE) | 33 295 | 18,7 | 148,7 | 71,4 % |
| Declaración y Pago Mensual | 28 752 | 16,2 | 39,6 | 75,8 % |
| Software Base | 27 846 | 15,6 | 16,1 | 76,4 % |

### *III-B. Por qué estos datos son ruidosos*

Conviene explicitar por qué este conjunto de datos constituye un entorno de aprendizaje con relación señal–ruido baja, porque esa característica explica el resultado principal del trabajo. Concurren cinco fuentes de ruido:

1. **Colas pesadas.** La mediana global de resolución es de 46 horas pero la media es de 679: una minoría de casos extremadamente lentos arrastra el promedio un orden de magnitud. En la cola gigante la distancia media/mediana es de 10× (638 frente a 64,7 horas). En consecuencia, dos tickets idénticos enviados a la misma cola pueden tardar 8 horas o 8 meses: la recompensa observada para una misma acción en un mismo estado tiene varianza enorme.
2. **Enrutamiento muchos-a-muchos.** No existe una "etiqueta correcta": cada servicio lo atienden decenas de grupos y la heurística del grupo modal solo replica el 77,6 % de las asignaciones históricas. El 22 % restante no es error de los despachadores, sino decisiones legítimas condicionadas por contexto no registrado (carga del momento, especialización del analista, escalamientos).
3. **Censura por la derecha.** Los casos lentos de 2025–2026 aún no se resuelven, de modo que las métricas recientes están sesgadas hacia lo rápido (Tabla I); todo estimador entrenado o validado sobre la cola temporal reciente hereda ese sesgo.
4. **Casos crónicos.** El 8,6 % de los tickets resueltos supera los 90 días; su firma es inequívoca: son defectos de software catalogados (sus asuntos citan literalmente "caso 93", "caso 229" de un catálogo de errores conocidos), el 44 % referencia un Problema Padre y el 71,5 % se creó durante la crisis 2023–2024. Ningún enrutador puede acelerar un ticket que espera una corrección de código; incluirlos en la calibración solo añadiría ruido irreducible.
5. **No estacionariedad.** El régimen de servicio cambia entre épocas (crisis y recuperación de la Tabla I), de modo que el "mismo" estado no tiene la misma dinámica en 2023 que en 2025.

Estas cinco fuentes tienen una consecuencia directa sobre el aprendizaje: el objetivo de la ecuación de Bellman que ajustan las redes es una muestra con ruido intenso, y un aproximador

con capacidad sobrada puede ajustar ese ruido además de la señal. Este es el mecanismo que la Sección VI utilizará para explicar por qué el modelo lineal generaliza mejor que el profundo.

### *III-C. Formulación como MDP*

El episodio simula el despacho de 200 tickets consecutivos. En cada paso $t$, el agente observa el estado $s_t = (\tau_t, h_t, d_t, \mathbf{c}_t, r_t)$, donde $\tau_t$ es el tipo del ticket entrante (las 20 subcategorías más frecuentes más la clase OTRAS, 21 en total), $h_t$ la hora del día, $d_t$ el día de la semana, $\mathbf{c}_t \in \mathbb{R}^{15}$ las cargas de las colas normalizadas por su carga de régimen, y $r_t$ la fracción de episodio restante — incluir el tiempo restante hace markoviana la formulación de horizonte finito [1]. La acción $a_t \in \{1, \ldots, 15\}$ elige el grupo receptor entre los 15 de mayor volumen (96,7 % del total). La recompensa es $R_t = \pm 1/-2$ según se cumpla o no el SLA del tipo, $-0{,}1 \cdot (\text{carga}/L)$ por congestión y $-0{,}5$ si el ticket es devuelto. Para la representación vectorial, el estado se codifica en 46 dimensiones (tipo y día *one-hot*, hora como seno/coseno, cargas continuas, tiempo restante); la codificación tabular discretiza las cargas en 6 niveles, lo que produce $21 \times 24 \times 7 \times 6^{15} \times 5 \approx 8{,}3 \times 10^{15}$ estados.

### *III-D. Entorno calibrado con datos reales*

Todos los parámetros del simulador se derivan con código de los 177 940 incidentes:

- **Llegadas:** proceso de Poisson no homogéneo con la tasa horaria y semanal empírica, simulado de forma exacta mediante el algoritmo de *thinning* de Lewis y Shedler [14]. Validación: distancia de variación total 0,019 y $\chi^2$ con $p = 0{,}29$ frente al histograma real.
- **Servicios:** la duración de cada atención se muestrea de la distribución empírica del grupo receptor (hasta 3 000 muestras reales por grupo), sin asumir forma paramétrica alguna — precisamente por las colas pesadas descritas en III-B.
- **Régimen y congestión:** como el tiempo empírico solución–inicio ya incluye la espera típica, cada cola inicia el episodio con su carga de régimen $L = \lambda W$ (ley de Little [15]) y la congestión solo estira el servicio cuando la carga excede ese régimen. Este diseño corrige un *reward hacking* detectado durante el desarrollo: una versión preliminar contabilizaba la congestión dos veces y un DQN aprendió a explotar esa doble contabilidad en lugar de enrutar mejor.
- **SLA:** el objetivo de cada tipo es el percentil 75 de su tiempo de resolución histórico (regla declarada); la heurística histórica cumple ≈75 % por construcción, como se discutió en III-A.
- **Devoluciones:** un ticket asignado a un grupo históricamente incompatible (participación menor al 5 % en ese tipo) rebota al grupo modal con probabilidad 0,70 y 24 h de retraso; el 3,98 % de incidentes reales con estatus Rechazado/Devuelto es el ancla empírica de este mecanismo.
- **Alcance declarado:** los casos crónicos ($> 90$ días, 8,6 % de los resueltos) se consideran derivados a gestión de problemas en el triaje y se excluyen de la calibración (muestras de servicio, objetivos de SLA e intensidad de llegadas), con la justificación empírica de III-B. La regla se declara explícitamente: el mismo filtro sin declarar sería maquillaje de resultados.

### *III-E. Algoritmos y arquitecturas*

Se comparan siete políticas: (i) aleatoria; (ii) la heurística *cola natural* (grupo modal del tipo); (iii) Q-learning y SARSA tabulares [3], [4]; (iv) un aproximador lineal $46 \to 15$ (691 parámetros), entrenado con la misma tubería de DQN sin capas ocultas; (v) DQN [5]: perceptrón multicapa $46 \to 128 \to 128 \to 15$ (≈25 000 parámetros, salida lineal con un valor $Q$ por acción), *experience replay* [13], red objetivo y pérdida de Huber; (vi) Double DQN [6], que desacopla la selección (red en línea) de la evaluación (red objetivo) para mitigar el sesgo de maximización $\mathbb{E}[\max_a Q] \geq \max_a \mathbb{E}[Q]$; y (vii) A2C [8]: actor softmax $46 \to 64 \to 64 \to 15$ y crítico $V(s)$, entrenados por episodio con ventaja Monte Carlo normalizada y regularización de entropía [7].

Los métodos tabulares actualizan sus valores mediante

$$Q(s,a) \leftarrow Q(s,a) + \alpha\big[r + \gamma \max_b Q(s',b) - Q(s,a)\big], \quad (1)$$

donde SARSA sustituye $\max_b Q(s',b)$ por $Q(s',a')$ con $a'$ la acción realmente ejecutada [3], [4]. El aproximador lineal y las redes minimizan la pérdida de Huber $\ell_\delta$ sobre el error de Bellman muestreado del *buffer*,

$$\mathcal{L}(\theta) = \mathbb{E}_{(s,a,r,s') \sim \mathcal{D}}\big[\ell_\delta\big(Q_\theta(s,a) - y\big)\big], \quad (2)$$

con el objetivo $y$ definido en el Algoritmo 1; en el caso lineal, $Q_\theta(s) = W\phi(s) + \mathbf{b}$ sin capas ocultas, de modo que ambos optimizan exactamente la misma pérdida con distinta clase de hipótesis. El actor–crítico asciende por el gradiente de política con ventaja $\hat{A} = G - V_\phi(s)$ y regularización de entropía,

$$\nabla_\theta J = \mathbb{E}\big[\nabla_\theta \log \pi_\theta(a|s)\, \hat{A}\big] + \beta\, \nabla_\theta H(\pi_\theta), \quad (3)$$

mientras el crítico minimiza $(V_\phi(s) - G)^2$ [7], [8].

La secuencia no es una colección arbitraria de métodos sino una cadena de comparaciones controladas en la que cada eslabón aísla una variable respecto del anterior. El aproximador lineal se entrena con la misma tubería, los mismos hiperparámetros y el mismo esquema de exploración que el DQN, de modo que la única diferencia entre ambos es la presencia de capas ocultas: si sus resultados divergen, la causa es la profundidad y no un detalle de implementación. Double DQN modifica exactamente una línea del algoritmo (el objetivo de Bellman, línea 8 del Algoritmo 1), lo que aísla el mecanismo del sesgo de maximización como única explicación de cualquier diferencia con el DQN. Finalmente, A2C cambia la familia completa (métodos de política en lugar de métodos de valor) manteniendo entorno, protocolo de evaluación y presupuesto de entrenamiento, para descartar que las patologías observadas sean exclusivas del aprendizaje por valores.

**Algorithm 1** DQN / Double DQN sobre el entorno calibrado

Inicializar red $Q_\theta$, objetivo $Q_{\theta^-} \leftarrow Q_\theta$, *buffer* $\mathcal{D}$
**for** episodio $= 1$ **to** $1500$ **do**
  $s \leftarrow$ `env.reset()` {colas en régimen de Little}
  **for** $t = 1$ **to** $200$ **do**
    $a \leftarrow \epsilon$-greedy$(Q_\theta(s))$; ejecutar $a$; observar $r, s'$
    guardar $(s, a, r, s', \text{fin})$ en $\mathcal{D}$
    muestrear *minibatch* de $\mathcal{D}$
    $y \leftarrow r + \gamma(1 - \text{fin}) \cdot \begin{cases} \max_b Q_{\theta^-}(s', b) & \text{DQN} \\ Q_{\theta^-}(s', \arg\max_b Q_\theta(s', b)) & \text{Double} \end{cases}$
    descenso con pérdida de Huber sobre $(Q_\theta(s, a) - y)$
    cada 1 000 pasos: $\theta^- \leftarrow \theta$
  **end for**
**end for**

Parámetros experimentales: $\gamma = 0{,}95$; tabulares con $\alpha = 0{,}1$, 3 000 episodios y $\epsilon$ lineal $1{,}0 \rightarrow 0{,}05$; profundos con Adam ($\eta = 10^{-3}$), 1 500 episodios, *buffer* de 50 000, *minibatch* 64, actualización cada 2 pasos, sincronización de la red objetivo cada 1 000 pasos y $\epsilon$ lineal durante el 70 % del entrenamiento; A2C con $\eta = 7 \times 10^{-4}$ y coeficiente de entropía 0,01.

## IV. Implementación

El proyecto se implementa en Python 3.13 con NumPy (entorno y métodos tabulares), pandas (calibración desde la exportación de incidentes), PyTorch 2.11 (redes), SciPy (pruebas estadísticas) y matplotlib (figuras), en un cuaderno Jupyter de 25 celdas que se ejecuta de principio a fin en ≈57 minutos de CPU. El procedimiento es: (1) calibración y validación del entorno; (2) demostración del colapso tabular; (3) entrenamiento multi-semilla de las políticas profundas; (4) evaluación común y análisis estadístico. El Algoritmo 1 resume el bucle de entrenamiento profundo.

**Protocolo experimental y estadístico.** La unidad de réplica es la semilla de entrenamiento: cada algoritmo se entrena con 8 réplicas independientes (5 para lineal y A2C), y toda política resultante se evalúa en las *mismas* 20 semillas de prueba (200 episodios comunes) — la técnica de *números aleatorios comunes* de la literatura de simulación, que elimina la varianza de escenario de las comparaciones. Sobre las réplicas se reportan medias e intervalos de confianza del 95 % con la $t$ de Student, siguiendo [2]. Las comparaciones contra la heurística — determinista en la evaluación común y, por tanto, sin varianza de réplica — usan la prueba $t$ de una muestra sobre las réplicas del algoritmo; las comparaciones entre pares de algoritmos (retorno y sesgo de DQN frente a Double DQN) usan la prueba de Wilcoxon de rangos con signo pareada por réplica, apropiada para muestras pequeñas sin supuesto de normalidad. Todas las pruebas son bilaterales con $\alpha = 0{,}05$; las tres afirmaciones centrales del trabajo (lineal $>$ heurística, DQN $<$ heurística, sesgo de Double DQN $<$ sesgo de DQN) conservan la significancia tras la corrección de Holm–Bonferroni para comparaciones múltiples ($p$ ajustados: $3{,}9 \times 10^{-6}$, 0,006 y 0,016). Un único generador de números aleatorios por entorno garantiza trayectorias reproducibles; la elección de CPU frente a GPU se decidió con un *benchmark* medido (a este tamaño de red, la transferencia por muestra domina sobre el cómputo).

## V. Resultados

### V-A. Colapso del enfoque tabular

Tras 3 000 episodios (600 000 decisiones), la tabla Q acumula ≈578 000 pares estado–acción y sigue creciendo linealmente: cada episodio nuevo visita estados que ningún episodio anterior había visto, señal de que el espacio de $8{,}3 \times 10^{15}$ configuraciones es esencialmente inagotable. La consecuencia se observa en la evaluación: la cobertura es de 0,01 %, es decir, de cada 10 000 decisiones de prueba solo una encuentra su estado en la tabla. En todas las demás, el agente tabular ejecuta su respaldo (la heurística), de modo que su retorno ($+3{,}62 \pm 0{,}17$) coincide con el de la cola natural hasta el ruido de muestreo. Dicho de otro modo: el agente tabular no es malo, es *inexistente* — casi nunca llega a opinar. La Fig. 1 ilustra el colapso en sus tres facetas (crecimiento sin saturación, cobertura desplomada y retorno indistinguible de la heurística) y contrasta con el problema pequeño del trabajo parcial, donde la misma tabla alcanzaba cobertura total. Esto motiva la aproximación de funciones [1], [16]: en lugar de memorizar valores por estado, aprender una función que interpole entre estados parecidos.

### V-B. Comparación de políticas

La Tabla IV resume la evaluación común sobre las mismas 20 semillas de prueba; conviene leerla fila por fila. La política aleatoria ($-132{,}4$) establece el piso: devuelve el 61,8 % de los tickets porque la mayoría de los pares tipo–grupo son incompatibles, lo que confirma que el mecanismo de devoluciones penaliza de verdad. La heurística de cola natural ($+3{,}62$, SLA 72,0 %) es una línea base exigente: nunca provoca devoluciones por construcción y hereda décadas de conocimiento operativo implícito en el grupo modal. Los dos métodos tabulares la igualan por la razón explicada en V-A.

El aproximador *lineal* la supera con significancia estadística ($+12{,}27 \pm 3{,}72$ frente a $+3{,}62$; prueba $t$ de una muestra sobre las réplicas, $p = 0{,}003$; el intervalo de confianza completo queda por encima del valor de la heurística), siendo el primer método de toda la secuencia del curso que lo consigue. Su mejora no proviene de descubrir asignaciones exóticas: mantiene las devoluciones en 1,0 % (casi tan conservador como la heurística) y gana desviando tickets hacia colas compatibles secundarias cuando la carga normalizada de la cola modal se eleva — exactamente el comportamiento sensible a congestión que un clasificador estático no puede expresar. El SLA sube de 72,0 % a 73,4 %; la ganancia es moderada en puntos porcentuales pero sistemática en las 20 semillas.

El DQN profundo, con la misma tubería de entrenamiento y 36 veces más parámetros, queda *por debajo* de la heurística ($-22{,}46 \pm 4{,}08$, $p = 1{,}3 \times 10^{-6}$). El diagnóstico está en la columna de devoluciones: dedica el 17,2 % de los tickets a

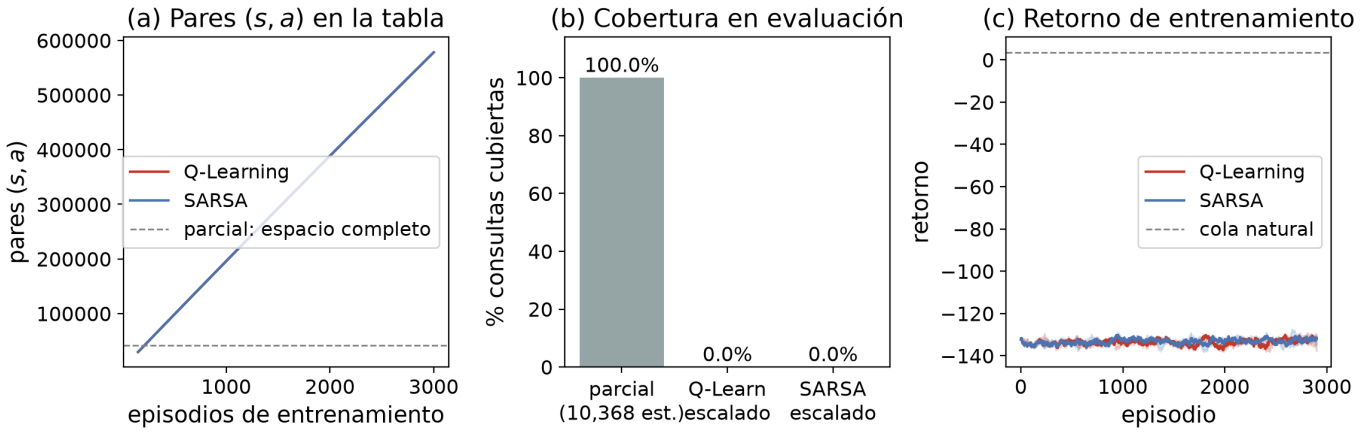


Figura 1. Colapso tabular: (a) la tabla crece sin saturar, (b) la cobertura en evaluación se desploma frente al 100 % del problema pequeño y (c) el retorno no supera a la heurística.

Tabla IV
EVALUACIÓN EN 20 SEMILLAS DE PRUEBA COMUNES (IC 95 % ENTRE RÉPLICAS DE ENTRENAMIENTO)

| Política | Retorno | SLA | Devol. |
|---|---|---|---|
| Aleatoria | $-132{,}4$ | 59,1 % | 61,8 % |
| Cola natural (heurística) | $+3{,}62$ | 72,0 % | 0 % |
| Q-learning tabular | $+3{,}62 \pm 0{,}17$ | 72,0 % | 0 % |
| SARSA tabular | $+3{,}50 \pm 0{,}21$ | 72,0 % | 0 % |
| Lineal 46→15 | $+12{,}27 \pm 3{,}72$ | **73,4 %** | 1,0 % |
| DQN 46→128→128→15 | $-22{,}46 \pm 4{,}08$ | 70,4 % | 17,2 % |
| Double DQN | $-24{,}19 \pm 2{,}71$ | 70,4 % | 18,7 % |
| A2C | $-35{,}98 \pm 12{,}76$ | 67,9 % | 13,6 % |

colas históricamente incompatibles. No se trata de exploración residual (la evaluación es totalmente voraz), sino de convicción errónea: la red asigna valores $Q$ altos a acciones que en la realidad histórica casi nunca atendieron ese tipo de ticket. Double DQN presenta el mismo cuadro ($-24{,}19 \pm 2{,}71$, 18,7 % de devoluciones) y el A2C añade el problema de la varianza: su intervalo de confianza ($-35{,}98 \pm 12{,}76$) es tan ancho que réplicas individuales van de casi competitivas a catastróficas, el comportamiento que la teoría predice para gradientes Monte Carlo sin línea base suficiente [2], [7]. La Fig. 2 muestra las curvas de aprendizaje: el lineal cruza la línea de la heurística de forma estable, mientras el DQN se estanca por debajo desde etapas tempranas.

### *V-C. Medición directa del sesgo de maximización*

Para explicar el fracaso del DQN no basta con constatarlo; hay que medir el mecanismo. La métrica utilizada es la sobreestimación $Q(s_0) - G_0$: al inicio de cada episodio de prueba se registra el valor que la red *predice* para su propia acción ($\max_a Q(s_0, a)$) y, al terminar el episodio, se calcula el retorno descontado $G_0$ que esa política *realmente* obtuvo. Si la red estuviera bien calibrada, la diferencia promediaría cero; un valor positivo significa que la red promete sistemáticamente más de lo que entrega.

El resultado es contundente: el DQN presenta un sesgo de $+9{,}97 \pm 0{,}37$ frente a $+8{,}91 \pm 0{,}44$ de Double DQN (Wilcoxon pareado sobre réplicas, $p = 0{,}016$). Es decir, *la corrección de van Hasselt reduce la sobreestimación con significancia estadística* [6]: el mecanismo teórico queda confirmado en un problema aplicado con datos reales. Sin embargo, el retorno de Double DQN no mejora respecto del DQN ($p = 0{,}64$), y la explicación es instructiva: la corrección reduce la *magnitud* del sesgo (de $+9{,}97$ a $+8{,}91$) pero el sesgo residual sigue favoreciendo las mismas acciones optimistas — las colas incompatibles cuyos valores se inflaron con muestras afortunadas — de modo que el *ranking* de acciones apenas cambia y la política resultante comete los mismos errores (Fig. 3). Corregir el estimador no corrige automáticamente la política.

## VI. DISCUSIÓN: POR QUÉ GANA LA GENERALIZACIÓN Y NO LA PROFUNDIDAD

El orden final — aleatoria $\ll$ profundas $<$ tabular $=$ heurística $<$ lineal — merece una explicación explícita, porque contradice la intuición de que más capacidad es mejor. La explicación tiene tres pasos.

**Primero: la política casi óptima de este problema es aproximadamente lineal en las características.** Una buena política de despacho se reduce a dos reglas: enviar cada tipo a un grupo compatible (una función del *one-hot* del tipo, es decir, una tabla de 21 preferencias aprendible con un peso por par tipo–cola) y desviar hacia la segunda opción cuando la carga normalizada de la primera se eleva (una función monótona — casi lineal — de las 15 cargas). Ambas reglas caben holgadamente en los 691 parámetros del modelo lineal. La capacidad adicional de las capas ocultas del DQN no es necesaria para representar la solución; solo puede usarse para otra cosa.

**Segundo: esa otra cosa es ajustar ruido.** Como se cuantificó en la Sección III-B, la recompensa de este entorno tiene varianza enorme: la misma acción en el mismo estado puede producir $+1$ o $-2$ según qué muestra de servicio toque en la distribución de cola pesada. Un aproximador de 25 000

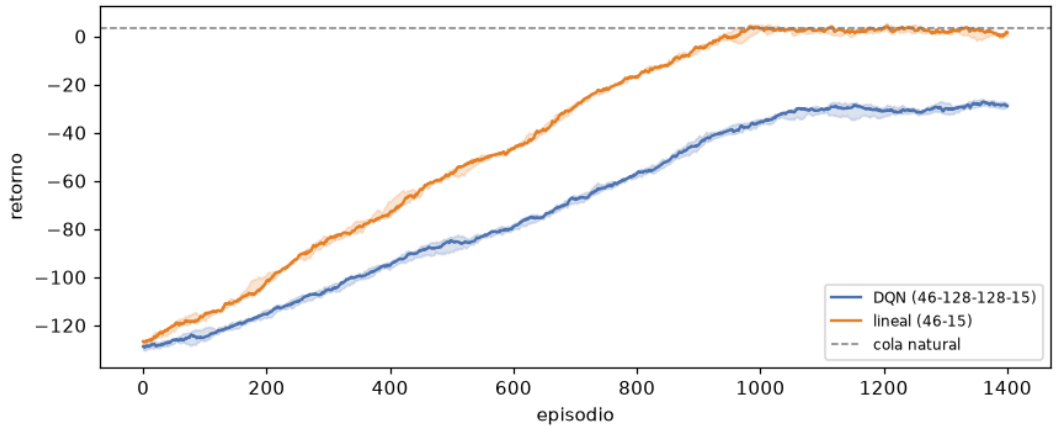


Figura 2. Curvas de aprendizaje (mediana e IQR entre réplicas): el lineal alcanza y supera a la heurística; el DQN se estanca por debajo.

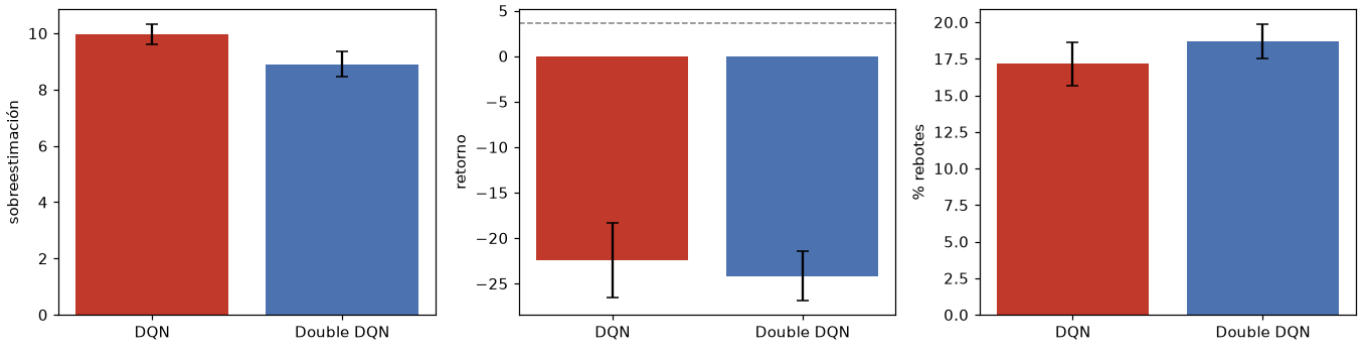


Figura 3. DQN vs. Double DQN: sobreestimación medida $Q(s_0) - G_0$ (izq.), retorno (centro) y devoluciones (der.), IC 95 % sobre 8 réplicas.

parámetros entrenado sobre objetivos de Bellman tan ruidosos tiene capacidad de sobra para memorizar coincidencias — "cuando la carga de la cola 7 vale 1,3 y es martes, la cola 12 dio $+1$" — que no son señal sino azar. El operador máx de Q-learning convierte entonces ese sobreajuste en un sesgo sistemático de sobreestimación (la "maldición del ganador": la acción elegida es precisamente aquella cuyo error de estimación fue más positivo), que en la Sección V-C se midió directamente: la red promete $+9{,}97$ más de lo que entrega, y persigue esas promesas asignando el 17 % de los tickets a colas incompatibles.

**Tercero: la rigidez del modelo lineal actúa como regularización implícita.** El modelo lineal *no puede* representar las coincidencias complejas del párrafo anterior: su clase de hipótesis solo contiene funciones aditivas de las características. Obligado a resumir todas las muestras de un tipo en un único peso por cola, promedia el ruido en lugar de memorizarlo, y lo que sobrevive a ese promedio es precisamente la parte estable de la señal (compatibilidad y congestión). Por eso generaliza a las 20 semillas de prueba nunca vistas: aprendió la estructura, no las anécdotas. Este es el mismo fenómeno que el trabajo parcial del curso observó a escala pequeña — donde un aproximador lineal de 37 pesos superó numéricamente a la tabla exacta — y coincide con el compromiso sesgo–varianza clásico: con datos ruidosos y estructura simple, el estimador restringido vence al flexible.

**Amenazas a la validez.** Cuatro limitaciones acotan el alcance de estas conclusiones. (i) Los hiperparámetros del DQN siguen la receta estándar de la literatura [5], [17] sin búsqueda exhaustiva; no puede descartarse que una red regularizada explícitamente (arquitectura menor, *dropout*, penalización L2) recupere la ventaja del lineal — de hecho, esa es exactamente la hipótesis que este análisis sugiere y que se declara como trabajo futuro. La comparación sigue siendo informativa porque el lineal comparte tubería, hiperparámetros y pérdida con el DQN (la profundidad es la única variable manipulada) y porque el mecanismo del fracaso no se conjetura: se mide (Sección V-C). (ii) Los presupuestos de entrenamiento difieren entre familias (3 000 episodios tabulares frente a 1 500 profundos); esto no compromete la conclusión del colapso tabular, porque la tabla no falla por presupuesto sino por cobertura: su crecimiento lineal sin saturación implica que más episodios agrandan la tabla sin acercarla a cubrir un espacio de $10^{15}$ estados. (iii) Los resultados provienen de un único sistema; la extrapolación a otras mesas de ayuda es plausible — las cinco fuentes de ruido de III-B son genéricas del dominio — pero no está demostrada. (iv) Dos supuestos del simulador (probabilidad de devolución 0,70 y penalización por retención de tickets incompatibles) tienen análisis de sensibilidad pendiente, también declarado como trabajo futuro.

Cabe subrayar, finalmente, la dependencia metodológica del hallazgo: sin el protocolo de réplicas, una semilla afortunada del A2C (cuyo IC abarca desde $-48$ hasta $-23$) habría podido presentarse como competitiva, y una semilla desafortunada del lineal como un empate. Las conclusiones de este trabajo existen *porque* se midieron con réplicas, intervalos, pruebas de significancia y corrección por comparaciones múltiples [2].

## VII. Conclusión

Se presentó una cadena completa de RL sobre un problema de despacho real con simulador íntegramente calibrado: el enfoque tabular colapsa a escala (cobertura 0,01 %), la aproximación de funciones lo reemplaza, y el aproximador lineal supera a la heurística histórica con significancia estadística — la ventaja proviene de la *generalización*, no de la *profundidad*, por el mecanismo de sesgo–varianza desarrollado en la Sección VI. La contribución metodológica principal es la medición directa del sesgo de maximización y la confirmación de que Double DQN lo reduce ($p = 0{,}016$), junto con la evidencia de que corregir el estimador no necesariamente corrige la política.

Como trabajo futuro se declaran: regularización explícita del DQN (arquitecturas menores, *dropout*, penalización L2) para comprobar si una red contenida recupera la ventaja del lineal; validación con partición temporal (entrenar con 2021–2024, validar con 2025–2026) dada la no estacionariedad documentada en la Tabla I; análisis de sensibilidad de los supuestos declarados (probabilidad de devolución, penalización por retención); y una formulación multiagente donde la devolución de tickets emerja del aprendizaje en lugar de suponerse. En lo operativo, los datos señalan que el camino hacia SLA altos no pasa por enrutar mejor: un enrutador *oráculo* calculado del histórico (cada tipo al grupo compatible con mayor probabilidad de cumplir su objetivo, ignorando la congestión) alcanzaría apenas un 83,7 % frente al 79,9 % ponderado de la heurística modal (cifras sobre el histórico completo, no comparables con el SLA simulado de la Tabla IV), y esa cota es optimista porque hereda el sesgo de selección de los datos. El margen explotable por *cualquier* enrutador es estrecho; el SLA incumplido vive en las distribuciones de tiempo — en los defectos catalogados que generan la cola pesada de casos crónicos — y no en la asignación.

## Referencias


[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA: MIT Press, 2018, disponible en: http://incompleteideas.net/book/the-book-2nd.html.

[2] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018, doi: 10.1609/aaai.v32i1.11694.

[3] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, pp. 279–292, 1992, doi: 10.1007/BF00992698.

[4] G. A. Rummery and M. Niranjan, "On-line Q-learning using connectionist systems," Cambridge University Engineering Department, Tech. Rep. CUED/F-INFENG/TR 166, 1994.

[5] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015, doi: 10.1038/nature14236.

[6] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, 2016, doi: 10.1609/aaai.v30i1.10295.

[7] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, pp. 229–256, 1992, doi: 10.1007/BF00992696.

[8] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 2016, pp. 1928–1937, disponible en: https://proceedings.mlr.press/v48/mniha16.html.

[9] A. Mandal, N. Malhotra, S. Agarwal, A. Ray, and G. Sridhara, "Cognitive system to achieve human-level accuracy in automated assignment of helpdesk email tickets," *arXiv preprint arXiv:1808.02636*, 2018, disponible en: https://arxiv.org/abs/1808.02636.

[10] P. Pham, V. Jain, L. Dauterman, J. Ormont, and N. Jain, "DeepTriage: Automated transfer assistance for incidents in cloud services," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 3281–3289, doi: 10.1145/3394486.3403380.

[11] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource management with deep reinforcement learning," in *Proceedings of the 15th ACM Workshop on Hot Topics in Networks (HotNets)*, 2016, pp. 50–56, doi: 10.1145/3005745.3005750.

[12] N. Walton and K. Xu, "Learning and information in stochastic networks and queues," *arXiv preprint arXiv:2105.08769*, 2021, disponible en: https://arxiv.org/abs/2105.08769.

[13] L.-J. Lin, "Self-improving reactive agents based on reinforcement learning, planning and teaching," *Machine Learning*, vol. 8, pp. 293–321, 1992, doi: 10.1007/BF00992699.

[14] P. A. W. Lewis and G. S. Shedler, "Simulation of nonhomogeneous Poisson processes by thinning," *Naval Research Logistics Quarterly*, vol. 26, no. 3, pp. 403–413, 1979, doi: 10.1002/nav.3800260304.

[15] J. D. C. Little, "A proof for the queuing formula: $l = \lambda w$," *Operations Research*, vol. 9, no. 3, pp. 383–387, 1961, doi: 10.1287/opre.9.3.383.

[16] R. Bellman, *Dynamic Programming*. Princeton, NJ: Princeton University Press, 1957.

[17] PyTorch, "Reinforcement learning (DQN) tutorial," https://docs.pytorch.org/tutorials/intermediate/reinforcement_q_learning.html, consultado: agosto de 2026.