# Optimización de la Gestión de Inventarios en cadenas de suministros de alimentos mediante Aprendizaje por Refuerzo

Camilo Aguilar León
Universidad de los Andes

# Agenda

1. **Introducción**
2. **Objetivos**
3. **Antecedentes**
4. **Definición del problema**
5. **Diseño del MDP**
6. **Diseño de agentes y modelos**
    a. **Políticas de manejos de inventarios**
    b. **MIP**
    c. **Q Learning**
    d. **DQN**
7. **Validación y resultados**
8. **Conclusiones**
9. **Trabajo futuro**
10. **Referencias**

# Introducción

Falta un intercambio eficiente de información entre los diferentes agentes de la cadena de suministro, lo que impide un control óptimo y dinámico de las operaciones (Gutiérrez, 2021).

Las cadenas de suministro de alimentos cortas están ganando atención por su capacidad para generar beneficios sociales, económicos y ambientales, marcando un contraste con los enfoques más tradicionales (EU Food Information Council, 2021).

La capacidad de responder rápidamente a cambios en la demanda del cliente, la competencia o interrupciones en el suministro es esencial para mantener la competitividad en el mercado (IBM, 2021).

Los métodos tradicionales de toma de decisiones, particularmente en la optimización y ubicación del inventario, han demostrado ser ineficientes frente a cómo funciona actualmente el comercio (Cozowicz, 2021).

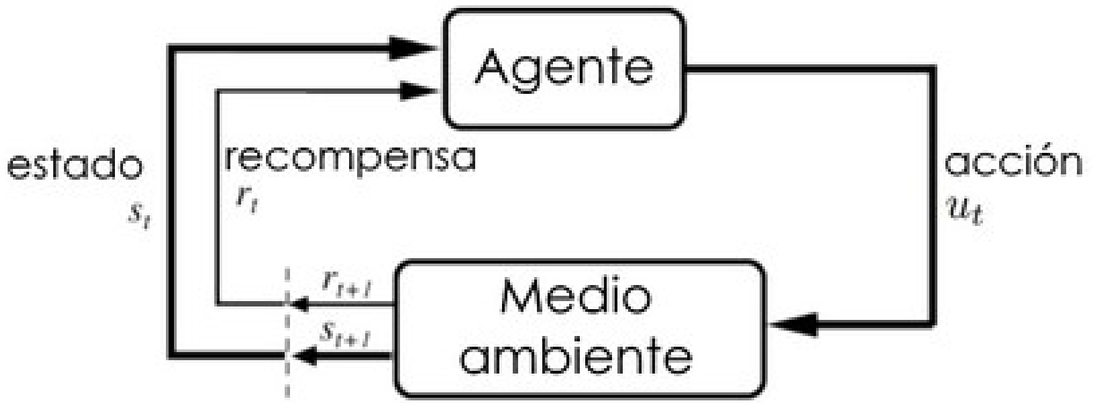
# Introducción

Se propone un modelo diseñado para manejar eficazmente la incertidumbre inherente en las cadenas de suministros de alimentos

Se busca facilitar la toma de decisiones óptimas respecto a la adquisición de productos, considerando una serie de variables críticas como la perecibilidad de los productos, la demanda fluctuante, precios y cantidades inciertas, costos de transporte, y la gestión eficiente del inventario

## Aprendizaje por refuerzo

### Q Learning - DQN




[Figure source: Sutton & Barto, 1998]

Imagen tomada de: https://www.xataka.com/inteligencia-artificial/conceptos-inteligencia-artificial-que-aprendizaje-refuerzo

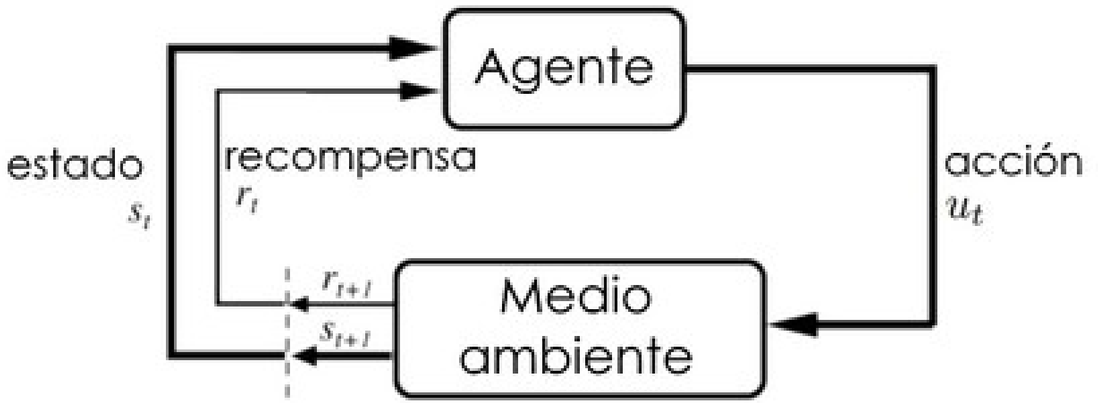
# Introducción

**Hallazgos y puntos claves de los modelos**

- Capacidad de captura de dinámicas complejas
- Mejora sobre políticas tradicionales de manejo de inventarios
- Toma de decisiones multifactorial
- Visión a largo plazo en la toma de decisiones
- Eficacia de la discretización de datos continuos

## Aprendizaje por refuerzo

### Q Learning - DQN




[Figure source: Sutton & Barto, 1998]

Imagen tomada de: https://www.xataka.com/inteligencia-artificial/conceptos-inteligencia-artificial-que-aprendizaje-refuerzo

# Objetivos

**Objetivo Principal**: Desarrollar un modelo de aprendizaje por refuerzo para integrar componentes estocásticos en cadenas de suministros, mejorando la toma de decisiones.

**Objetivos Secundarios**

- Análisis de Elementos Clave
- Diseño de Modelos de Aprendizaje por Refuerzo
- Implementación y Verificación
- Validación de Modelos
- Identificación de Ventajas y Áreas de Mejora

# Antecedentes

- **Investigación de Gómez et al. (2023)**
- **'or-gym' de Hubbs et al. (2019):** Entorno de simulación para problemas de investigación de operaciones, útil para entrenar y validar modelos de RL en cadena de suministros.
- **Contribución de Hutse (2019):** Exploración de técnicas avanzadas de RL con variables como tiempos de entrega y acciones continuas, utilizando la librería GYM de Python.
- **Trabajo de van Helsdingen (2022):** Desarrollo y evaluación de agentes de Q Learning y DQN en cadenas de suministros, con enfoque en ajuste de hiperparámetros y comparación con métodos de simulación comerciales.
- **Double Q Learning de van Hasselt (2010):** Introducción de una metodología robusta para entornos de RL en espacios continuos, minimizando sobreestimación en acciones.

Los estudios e implementaciones existentes se centran principalmente en la formulación básica de agentes de RL para cadenas de suministros sencillas, indicando oportunidades para explorar y maximizar el potencial de estas técnicas avanzadas.

# Definición del problema

Enfocado en la "primera milla" de las cadenas de suministro alimenticias, con énfasis en el rol de un intermediario que adquiere alimentos de agricultores para su reventa.

- **Dinámica de Proveedores:** Cada proveedor tiene un catálogo específico y se recopila información sobre precio y disponibilidad de los productos.
- **Gestión de Inventario por el Intermediario:** Administración de un almacén para satisfacer demandas conocidas de productos, teniendo en cuenta la tasa de perecimiento de los alimentos.
- **Estructura de Precios:** Precio de venta constante y superior al de compra, que puede variar.
- **Costos de Transporte:** Dependientes de la distancia entre proveedores y almacén, y la cantidad de vehículos necesarios, limitada por su capacidad. Viajes de ida y vuelta.
- **Penalizaciones y/o costos adicionales:** Penalizaciones por no cumplir con la demanda o por mantener inventario que afectan los beneficios.
- **Horizonte Temporal y Decisiones:** Enfoque en un horizonte temporal discreto y fijo, buscando maximizar utilidades considerando todos los factores. Decisiones de compra periódicas frente a la incertidumbre de información futura.

# Diseño del MDP

## (S,A,P,R)

**Estados (S):** Inventario, demanda, precios de compra, cantidades disponibles (I,d,p,q)
**Acciones (A):** Cantidad a comprar (z)
**Función de transición (P):** Obtenida a partir del cambio del inventario y los valores aleatorios de las otras componentes del estado.
F**unción de recompensa (R):** Utilidades del periodo contando penalizaciones

**Restricciones:**

$$I_{pt} = I_{pt-1}(1-\phi) + \sum_{s \in S_p} z_{spt} - \min\left(d_{pt}, I_{pt-1}(1-\phi) + \sum_{s \in S_p} z_{spt}\right), \forall p \in P, t \in T | t > 0$$

$$\sum_{p \in P} z_{spt} \leq Q n_{st}, \qquad \forall s \in S, t \in T$$

$$z_{spt} \leq q_{spt}, \qquad \forall s \in S, p \in P, t \in T$$

$$z_{spt} \geq 0, \qquad \forall s \in S, p \in P, t \in T$$

# Diseño del MDP

$$R(s_t, a_t) = \sum_{p \in p} \sum_{s \in S_p} \left( x_p z_{spt} - \max\left( k_p (d_{pt} - z_{spt}), h_p (z_{spt} - d_{pt}) \right) - p_{spt} z_{spt} \right) - \frac{c}{v} \sum_{s \in S} l_s n_{st}$$

**Recompensa obtenida en cada periodo**

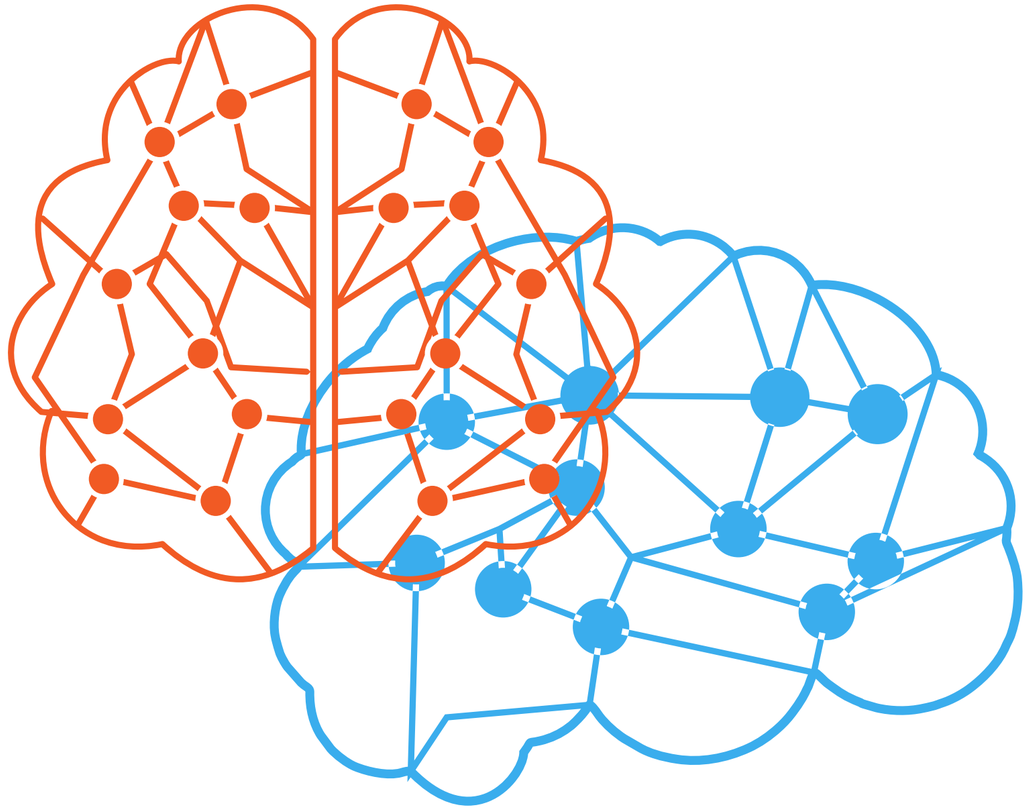
$$\pi^* = argmax_{\pi \in \Pi} \mathbb{E}\left( \sum_{t \in T} R(s_t, a_t^{\pi}) \mid S_0 \right)$$

**Política óptima**

# Diseño de agentes y modelos

**Modelos:**

- R,Q
- s, S
- MIP
- Q Learning s, S
- Q Learning
- DQN



La mayoria de los datos y los métodos para la generación de estos fueron obtenidos de la investigación de  Gómez et al. (2023)

# R,Q y s,S

**Nivel de reorden:** 0.5 veces la demanda máxima estipulada
**Cantidad a ordenar:**

- **R,Q:** 1.5 veces la demanda máxima estipulada
- **s,S:** Lo restante para alcanzar 1.5 veces la demanda máxima estipulada

**Decisión de compra:** En caso de ordenar, se escoge de manera greedy a partir del precio de venta los proveedores a los que compra

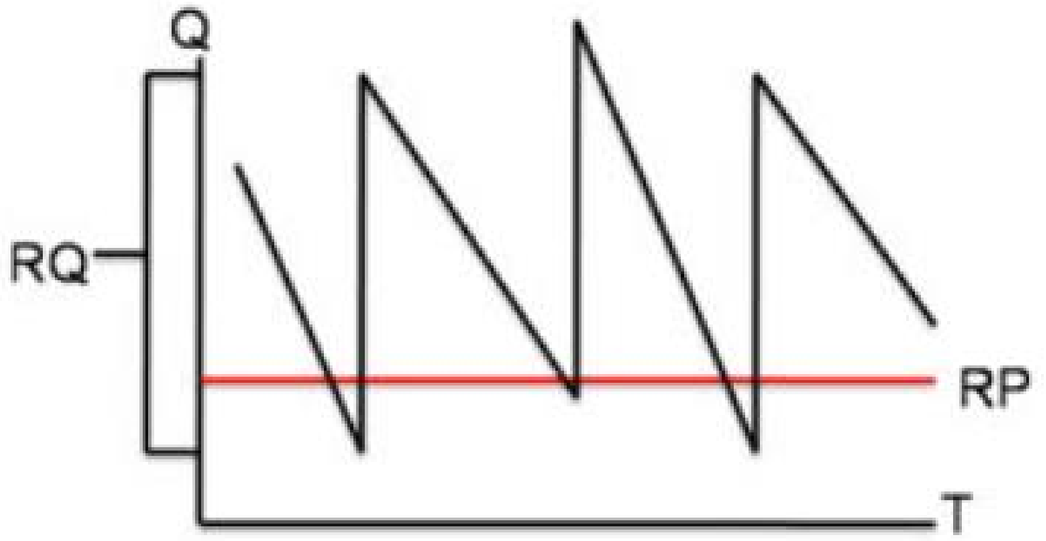



Imagen tomada de: https://help.llama.ai/release/native/modeling/modeling-topics/Modeling_Inventory_Policies.htm

# MIP

$$\max \sum_{t \in T} \sum_{p \in p} \sum_{s \in S_p} \left(x_p z_{spt} - p_{spt} z_{spt}\right) - \frac{c}{v} \sum_{s \in S} l_s n_{st}$$

No se incluyen penalizaciones ya que los escenarios permitiran encontrar soluciones sin recurrir a penalizaciones.

$$I_{pt} = I_{pt-1}(1 - \phi) + \sum_{s \in S_p} z_{spt} - d_{pt}\,, \qquad \forall p \in P, t \in T | t > 0$$

$$\sum_{p \in P} z_{spt} \leq Q n_{st}, \qquad \forall s \in S, t \in T$$

$$z_{spt} \leq q_{spt}, \qquad \forall s \in S, p \in P, t \in T$$

$$z_{spt} \geq 0, \qquad \forall s \in S, p \in P, t \in T$$

$$n_{st} \in \mathbb{Z}^+, \qquad \forall, p \in P$$

$$I_{p0} = 0, \qquad \forall, p \in P$$

# Q Learning (Marco Teórico)

## Ecuación de Bellman

$$Q(S_t, A_t) = (1 - \alpha)Q(S_t, A_t) + \alpha \left[R_t + \gamma \max_{a \in A} Q(S_{t+1}, a)\right]$$

$\alpha \in [0,1]$ **Tasa de aprendizaje**

$\gamma \in [0,1]$ **Tasa de descuento**

$\epsilon \in [0,1]$ **Probabilidad de exploración**

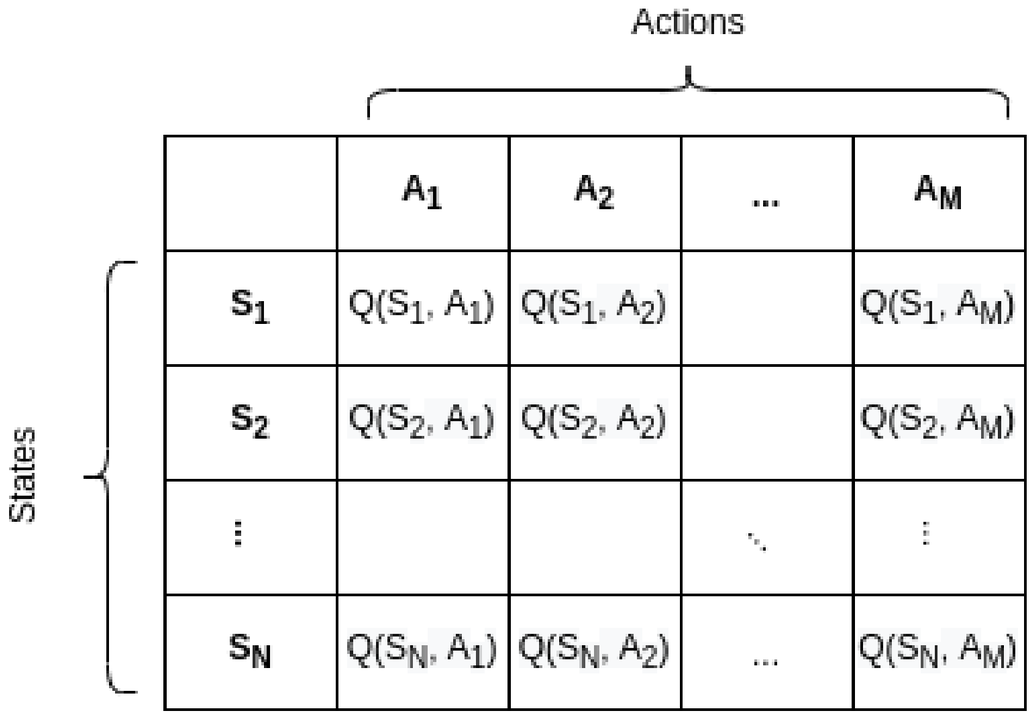



Imagen tomada de: https://www.baeldung.com/cs/epsilon-greedy-q-learning

# Q Learning (Marco Teórico)

**Algorithm 1:** Epsilon-Greedy Q-Learning Algorithm

**Data:** $\alpha$: learning rate, $\gamma$: discount factor, $\epsilon$: a small number
**Result:** A Q-table containing Q(S,A) pairs defining estimated optimal policy $\pi^*$

```
/* Initialization                                          */
```
Initialize Q(s,a) arbitrarily, except Q(terminal,.);
Q(terminal,.) $\leftarrow$ 0;
```
/* For each step in each episode, we calculate the
   Q-value and update the Q-table                          */
```
**for** *each episode* **do**
```
    /* Initialize state S, usually by resetting the
       environment                                          */
```
    Initialize state S;
    **for** *each step in episode* **do**
        **do**
```
            /* Choose action A from S using epsilon-greedy
               policy derived from Q                        */
```
            A $\leftarrow$ SELECT-ACTION(Q, S, $\epsilon$);
            Take action A, then observe reward R and next state S';
            Q(S, A) $\leftarrow$ Q(S, A) + $\alpha$ [ R + $\gamma \max_a$ Q(S', a) - Q(S, A)];
            S $\leftarrow$ S';
        **while** *S is not terminal*;
    **end**
**end**

**Algorithm 2:** Epsilon-Greedy Action Selection

**Data:** Q: Q-table generated so far, : a small number, S: current state
**Result:** Selected action
**Function** *SELECT-ACTION(Q, S, $\epsilon$)* **is**
    n $\leftarrow$ uniform random number between 0 and 1;
    **if** $n < \epsilon$ **then**
        A $\leftarrow$ random action from the action space;
    **else**
        A $\leftarrow$ maxQ(S,.);
    **end**
    return selected action A;
**end**

Imagen tomada de: https://www.baeldung.com/cs/epsilon-greedy-q-learning

# Q Learning

**Estados (S):** Inventario, demanda, tiempo de transporte, precios de compra, cantidades disponibles (I,d,l, p,q)

- El tiempo de transporte no varía dentro de un episodio, pero es importante para tomar decisiones
- Todas las variables requieren ser discretas e idealmente finitas para poder buscar que la tabla converja. Se crea una cantidad k de intervalos dentro de un rango determinado para asociar los valores continuos a estos. Se añaden el intervalo de valores menores al rango y el de valores mayores.
- Es necesario respetar la naturaleza de cada variable de estado y, por lo tanto, existe una para cada proveedor y producto en caso de que aplique. Además, se suman las combinaciones asociadas a la discretización.

El factor del costo computacional empieza a ser un factor a tener en cuenta

# Q Learning

**Acciones (S):**

- **Q Learning:** Porcentaje de la cantidad disponible de un producto de un proveedor a comprar (z)
  - Al igual que para los estados, se debe discretizar este porcentaje, por lo que quedan porcentajes fijos que se puedan comprar.
  - Cantidad de acciones posibles: $k^{SP}$

- **Q Learning s,S:** Si se debe generar una orden para ese producto.
  - Al igual que para la política s,S, la decisión de a que proveedor comprar se hace de manera greedy.
  - Cantidad de acciones posibles: $P$

La diferencia en la cantidad de acciones posibles para ambos casos es demasiado grande. Existe un tradeoff entre la granularidad de las acciones y el costo de entrenar el agente.

# DQN (Marco teórico)

**Aprendizaje por refuerzo a aprendizaje supervisado**

**Replay Buffer**

$$(S, A, R, S')$$

Independencia de los datos

**Target Network**

Correlación entre datos

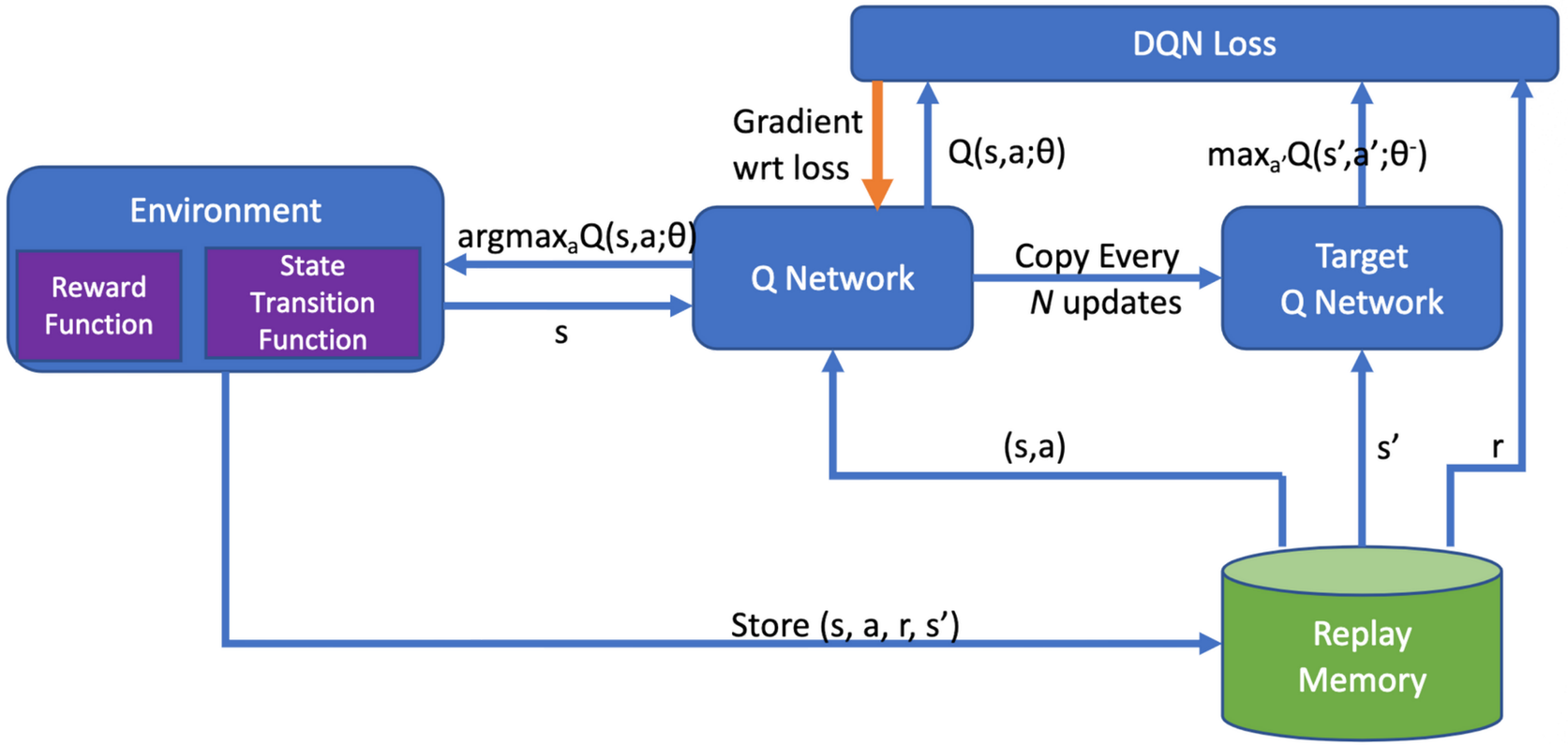



Imagen tomada de: https://rongyi.ai/publication/automap/

# DQN

**Estados (S):** Las variables usadas son las mismas que para el Q Learning solo que no necesitan ser discretizadas, ya que la red neuronal permite tener como entrada valores continuos y no finitos.
**Acciones (A)**: Se mantienen las mismas acciones discretas que en Q Learning

A partir de la red neuronal permite generar una representación no lineal de los valores de Q a partir de los estados y las acciones. Se escoge la acción cuyo valor Q de salida sea el mayor.

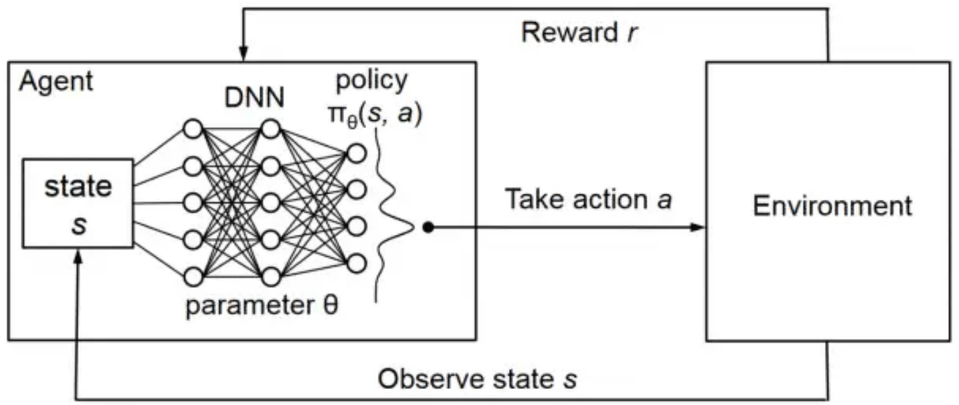
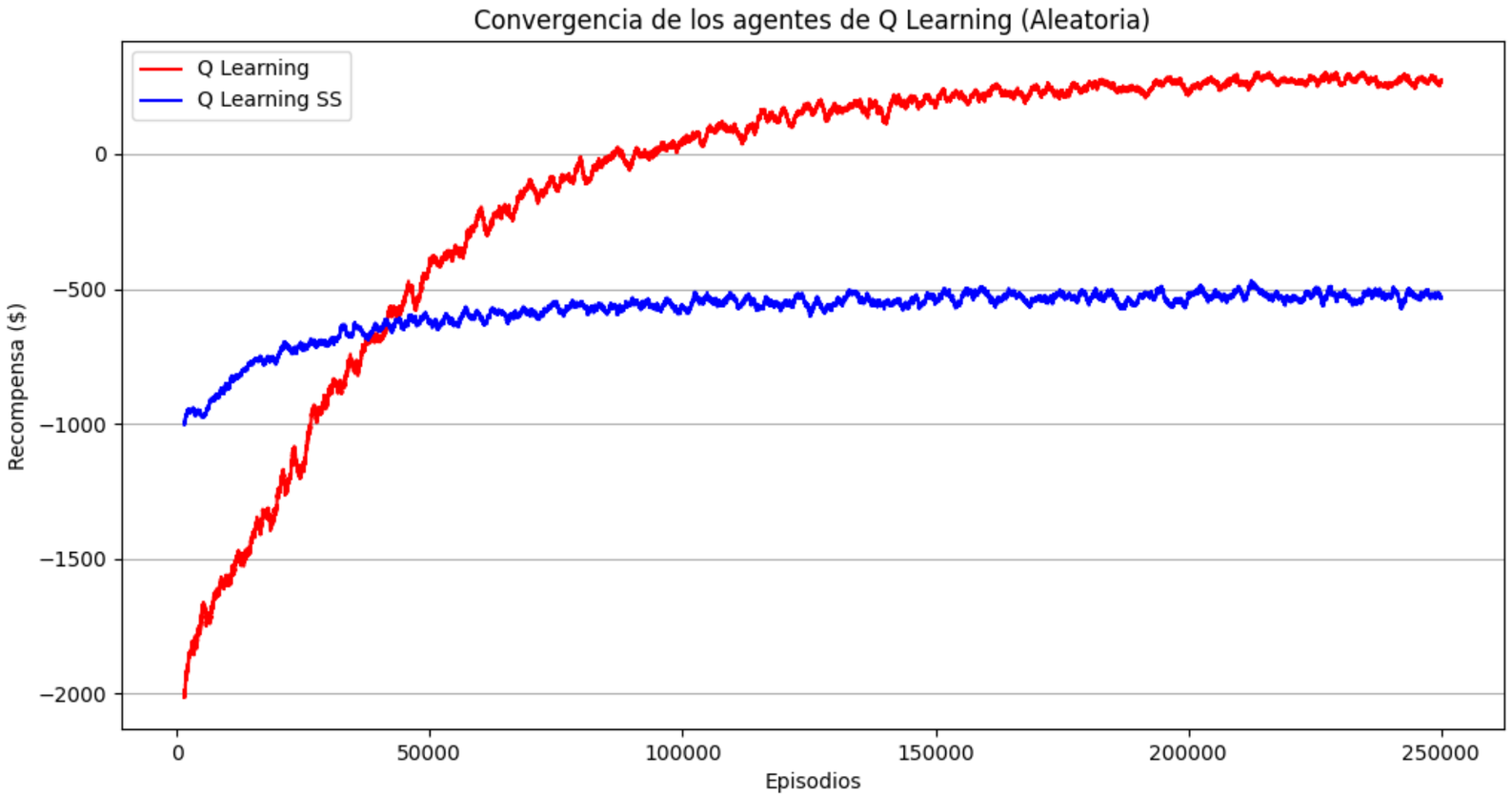


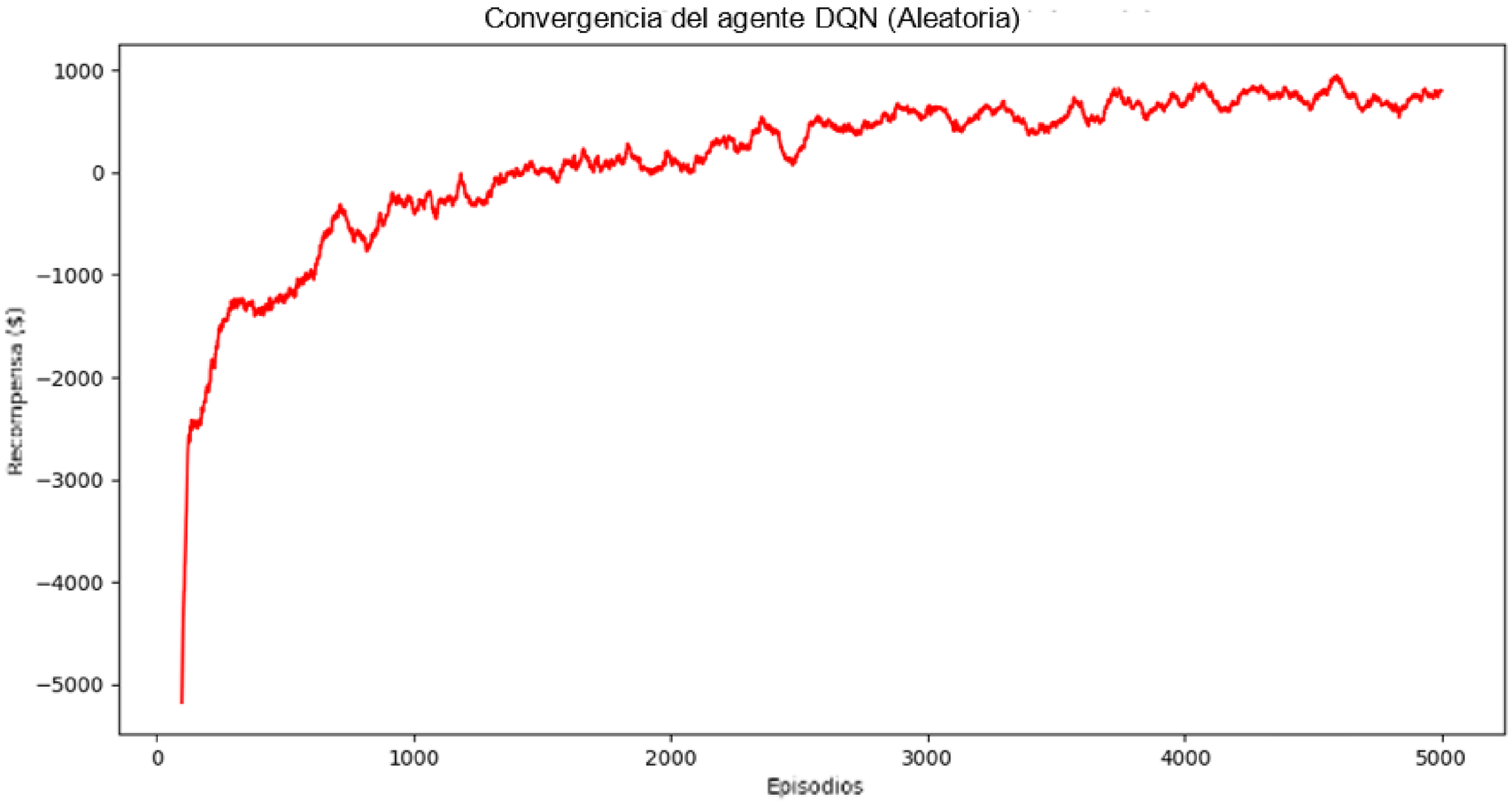

Imagen tomada de: https://dilithjay.com/blog/dqn/

# Validación y resultados



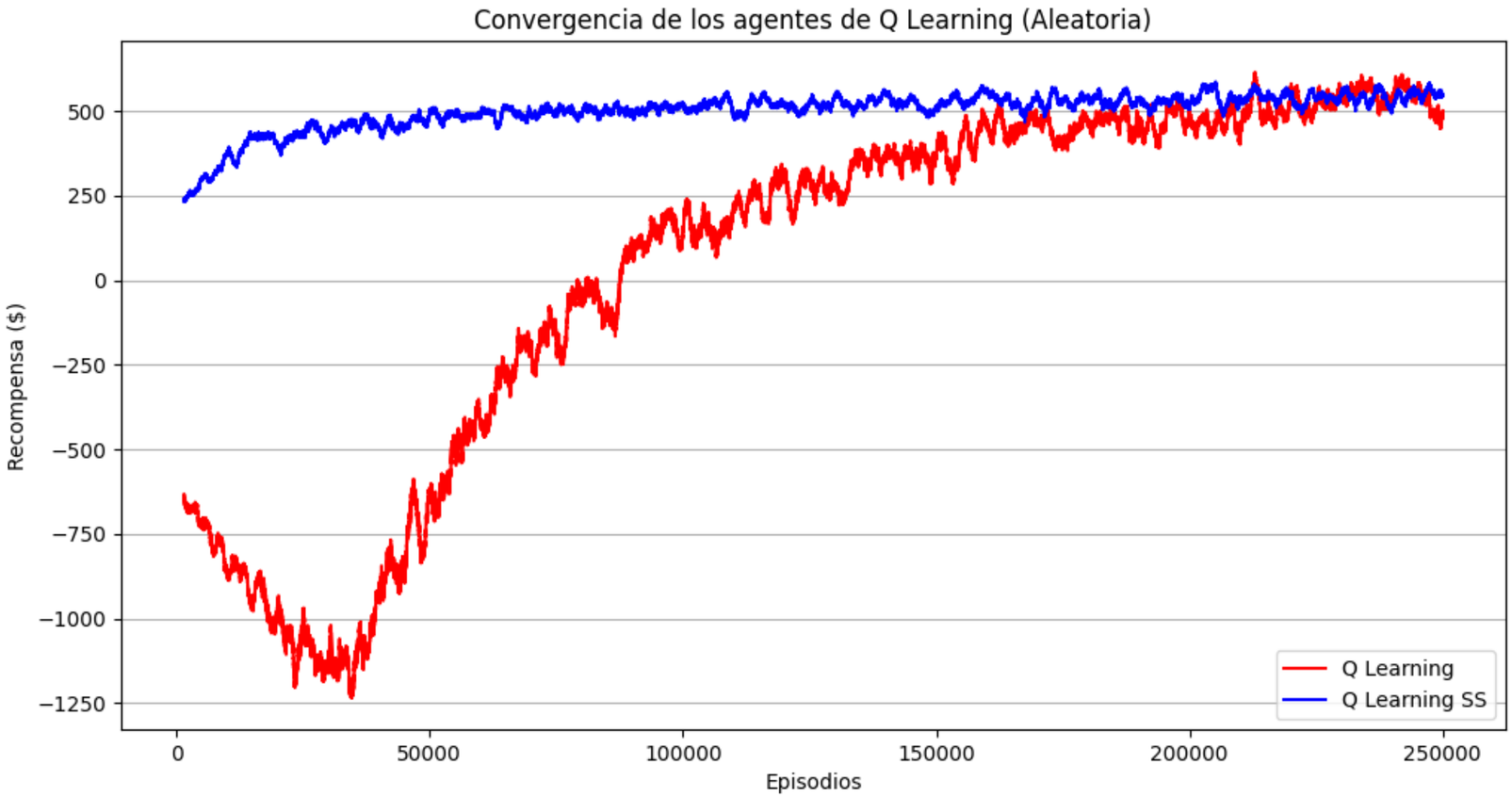

**Entrenamiento de los agentes con datos aleatorios (media móvil 1500 episodios)**

# Validación y resultados



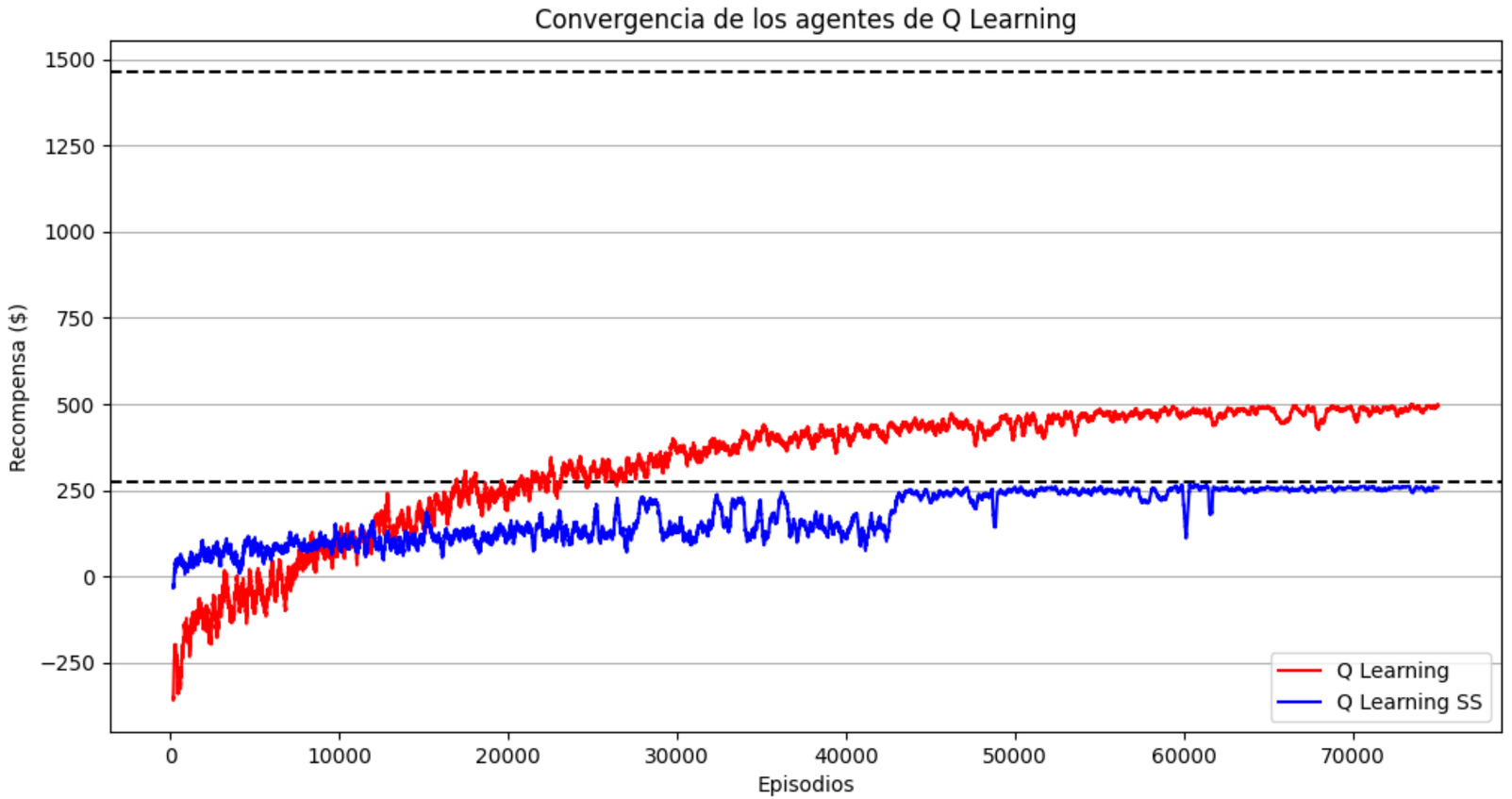

**Entrenamiento de los agentes con datos aleatorios (media móvil 100 episodios)**

# Validación y resultados



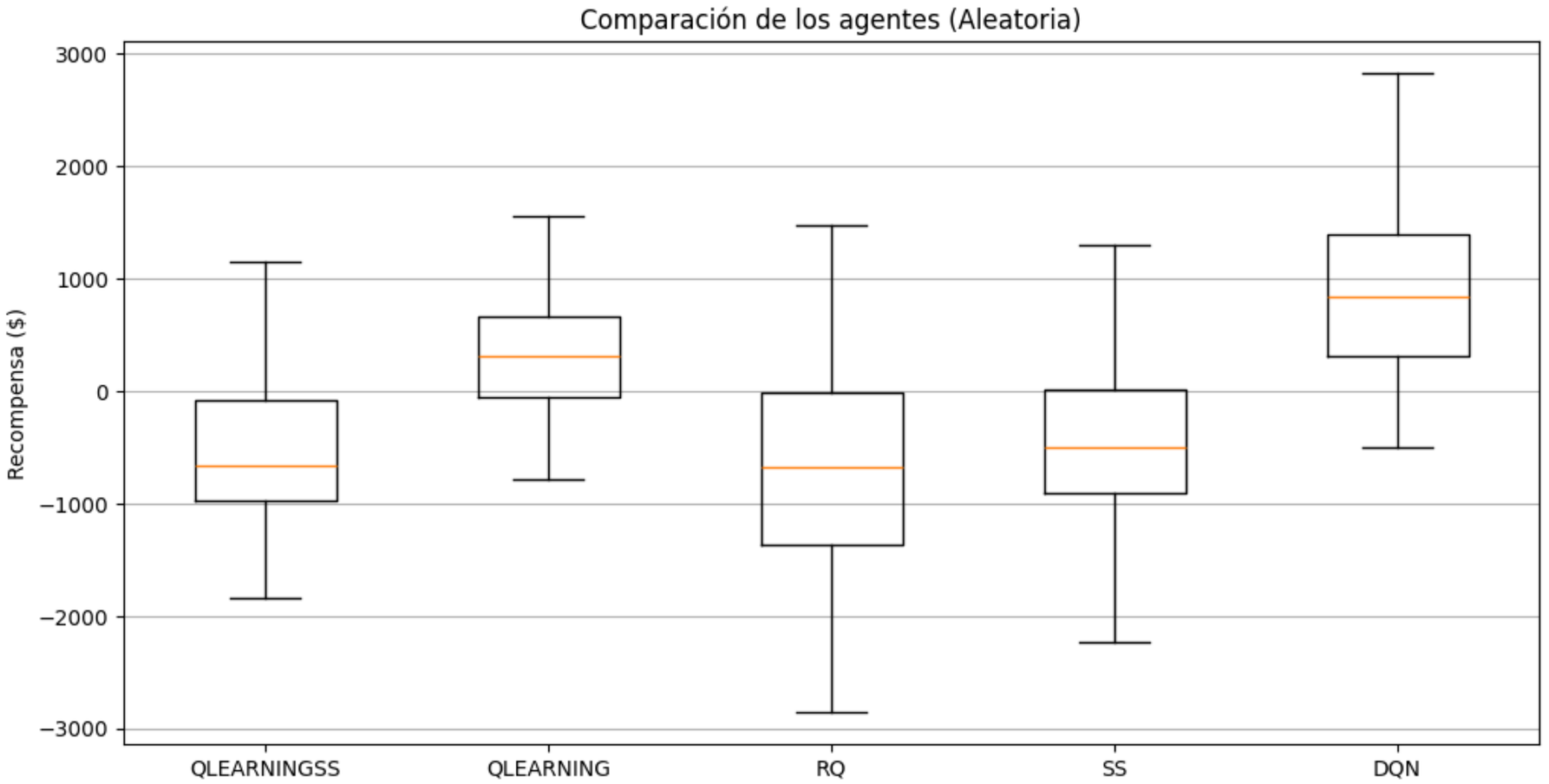

**Entrenamiento de los agentes con datos aleatorios (media móvil 1500 episodios y sin penalizaciones)**

# Validación y resultados

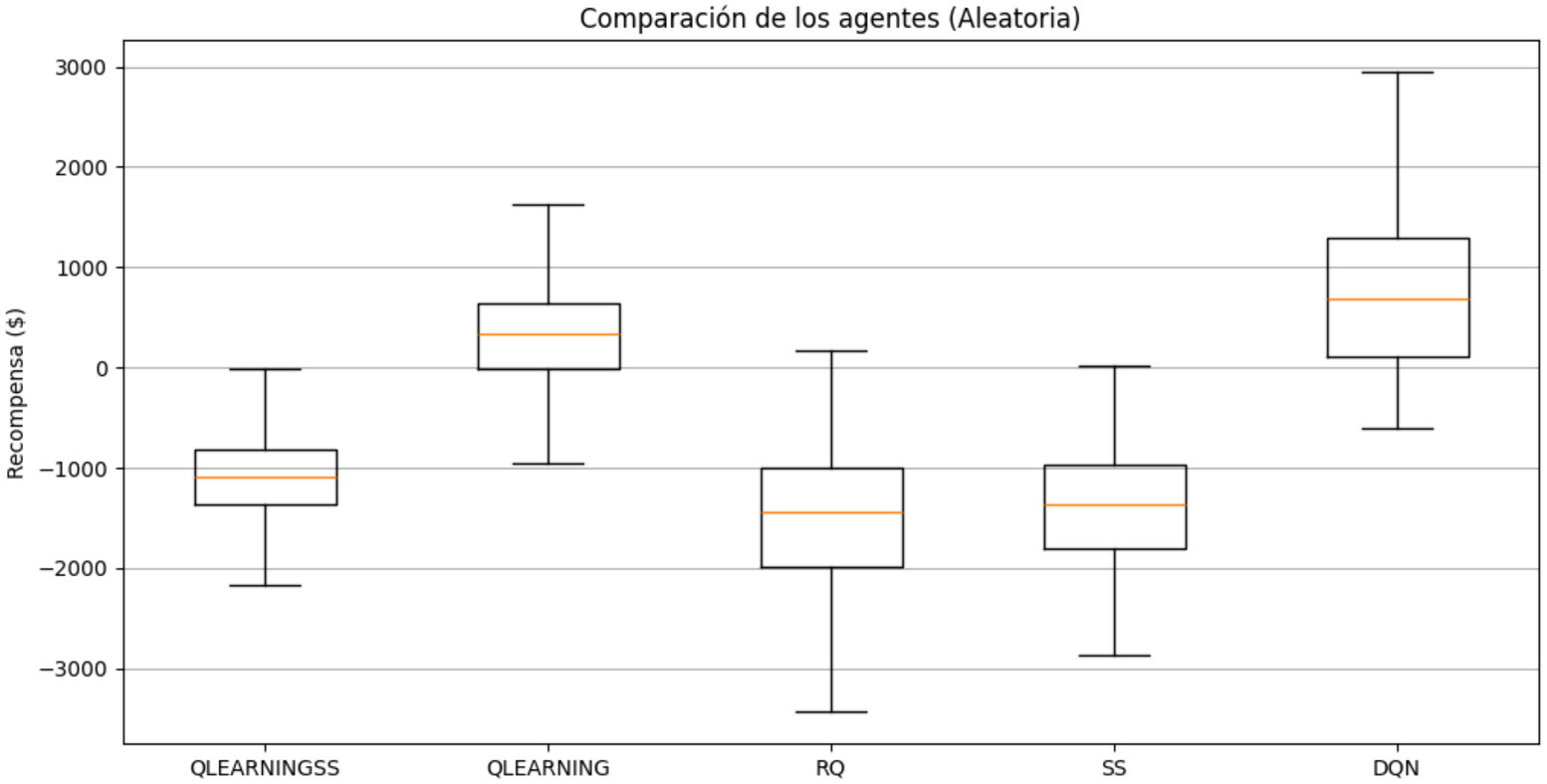



**Entrenamiento de los agentes con semilla (media móvil 1500 episodios y sin penalizaciones)**

# Validación y resultados

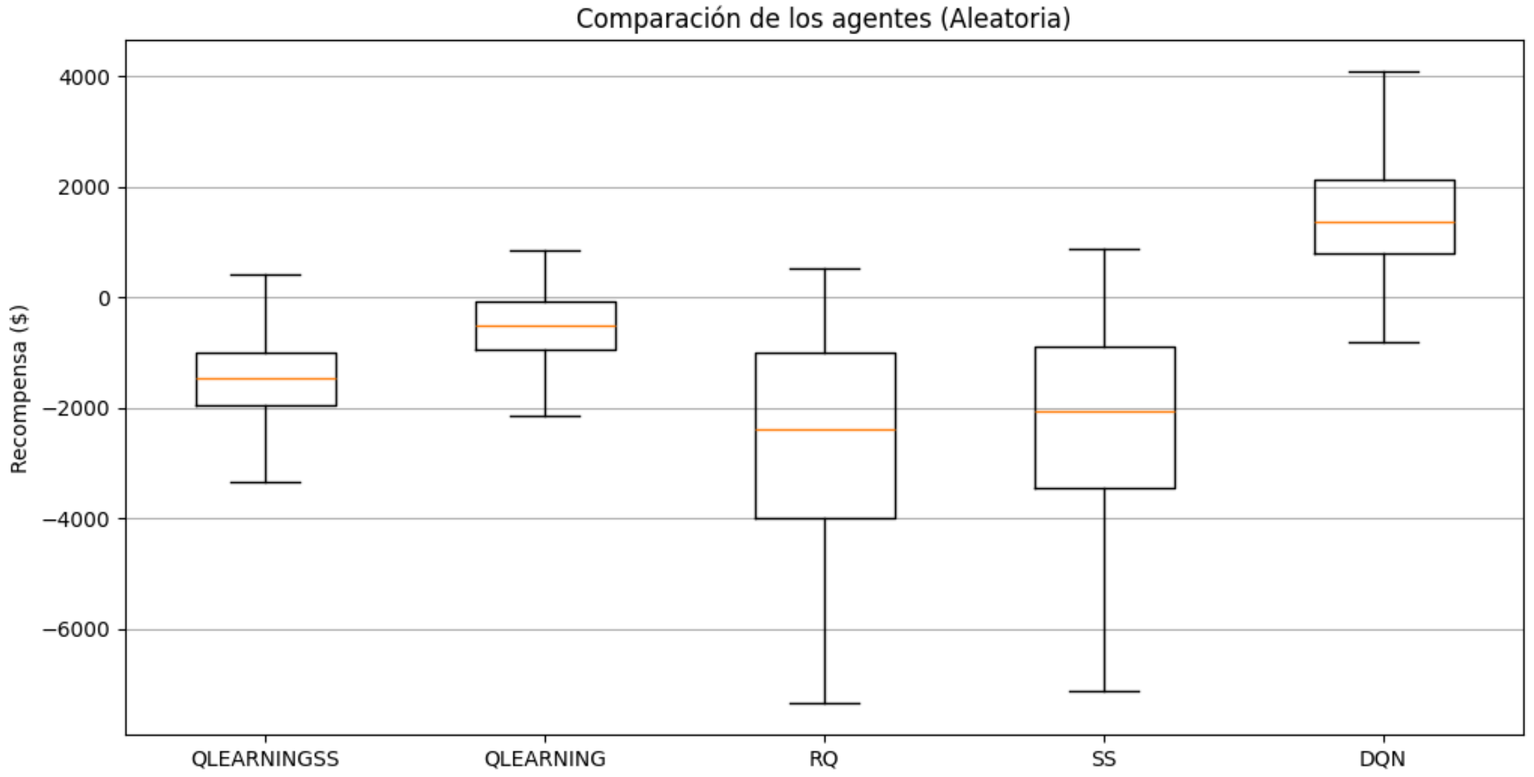


# Validación y resultados



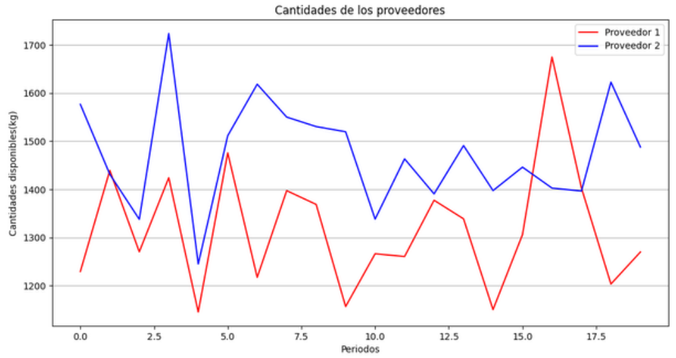

**Comparación de los agentes aumentando perecibilidad a 0.4**

# Validación y resultados



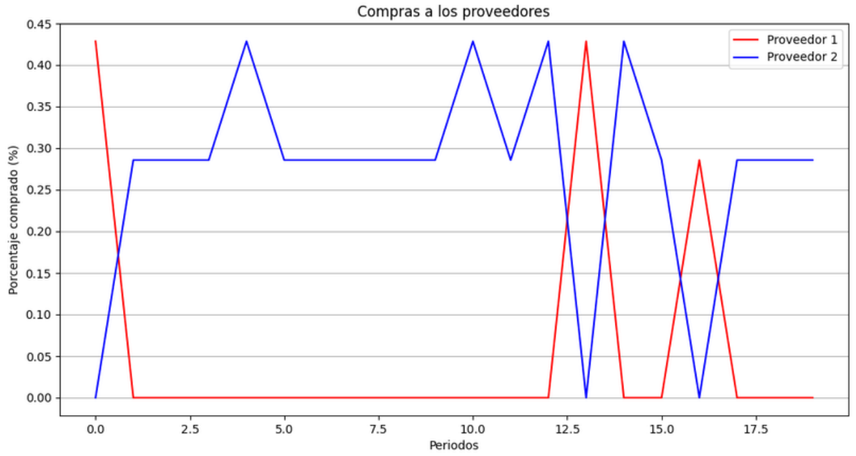

**Comparación de los agentes aumentando máxima demanda a 1800**

# Validación y resultados

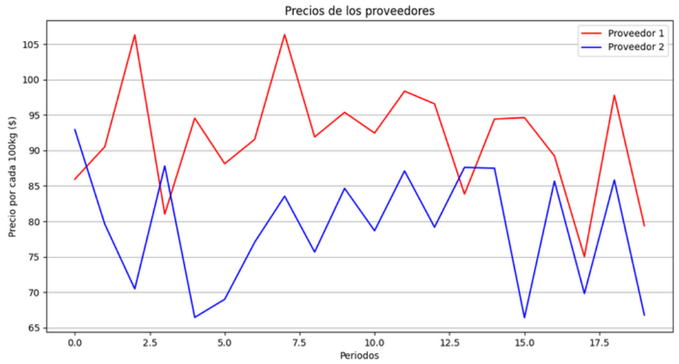



El inventario suele mantenerse en cero por lo que el agente prefiere más que todo penalizar en tanto a mantener inventario

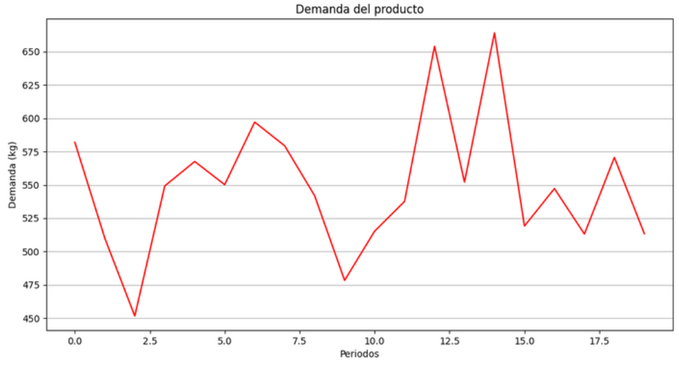

Las únicas excepción donde se compra al proveedor que no tiene el mayor costo se puede explicar porque en ese periodo el otro proveedor tiene bastante más cantidad disponible




Los picos de inventario son bastante pequeño por lo que se pueden explicar a partir de la granularidad de la discretización de las acciones




Existe una relación estrecha entre los precios y a que proveedor se le compra. En la mayoria de los casos se le compra al proveedor con menores precios.



# Conclusiones

- Se ha logrado con éxito la implementación de modelos de aprendizaje por refuerzo que mejoran las políticas de manejo de inventarios existentes y permiten una toma de decisiones más acertada e informada.
- Los agentes desarrollados han demostrado su capacidad para mejorar tanto las recompensas como los resultados operativos, evidenciando la efectividad de ajustar las acciones dentro del modelo de aprendizaje por refuerzo.
- Pese a su elevado costo computacional inicial, los agentes de aprendizaje por refuerzo demostraron que, una vez entrenados, pueden continuar aprendiendo de manera eficiente (online learning) y realizar decisiones operativas de manera casi instantánea. Esto es muy útil en el contexto de manejos de inventarios donde usualmente se deben tomar decisiones rápidas.

# Trabajo Futuro

- Realizar un análisis de sensibilidad más detallado sobre los parámetros de aprendizaje, la discretización, tamaño de la red y los espacios de acción y estado de los agentes. Esto puede permitir reducir el espacio de estados o el de acciones.
- Comparar los resultados obtenidos con modelos más sofisticados.
- Investigar nuevas metodologías para el diseño de agentes que optimicen la toma de decisiones en estos entornos. Buscar maneras de que se complementen las diferentes técnicas.
- Desarrollar un agente más avanzado y realista, que incorpore decisiones adicionales como el ruteo.
- Trabajar en estrategias para disminuir el costo computacional asociado con el entrenamiento de estos agentes.

# Referencias


Cozowicz, M. (2021). Reinforcement Learning in Supply Chain [LinkedIn article]. Recuperado de https://www.linkedin.com/pulse/reinforcement-learning-supply-chain-markus-cozowicz/

EU Food Information Council. (2021). Los beneficios y la sostenibilidad de las cadenas de suministro de alimentos cortas. Recuperado de https://www.eufic.org/es/produccion-de-alimentos/articulo/Los-beneficios-y-la-sostenibilidad-de-las-cadenas-de-suministro-de-alimentos-cortas

Gutiérrez, A. (2021). Estudio de la cadena de suministro. Recuperado de https://repositorio.ulima.edu.pe/bitstream/handle/20.500.12724/13303/Gutierrez_Estudio-cadena-suministro.pdf?sequence=1

IBM. (2021). Optimización de la cadena de suministro. Recuperado de https://www.ibm.com/mx-es/topics/supply-chain-optimization

Hubbs, C., Heitz, G., & Dilkina, B. (2019). Or-gym: A Reinforcement Learning Environment for Operations Research Problems. Recuperado de http://egon.cheme.cmu.edu/Papers/Hubbs_or_gym_9_11.pdf

van Hasselt HP (2010) Double Q-Learning. In: Advances in Neural Information Processing Systems, The MIT Press, vol 23

Wu, G., Servia, M. Á. de C., & Mowbray, M. (2023). Reinforcement Learning for inventory management in multi-echelon supply chains. In A. C. Kokossis, M. C. Georgiadis, & E. Pistikopoulos (Eds.), Computer Aided Chemical Engineering (Vol. 52, pp. 795–800). Elsevier. https://doi.org/10.1016/B978-0-443-15274-0.50127-X

Geevers, K., van Hezewijk, L. & Mes, M.R.K. Multi-echelon inventory optimization using deep reinforcement learning. Cent Eur J Oper Res (2023). https://doi.org/10.1007/s10100-023-00872-2

Cuellar-Usaquén, D., Ulmer, M. W., Gomez, C., & Álvarez-Martínez, D. (2023). Adaptive stochastic lookahead policies for dynamic multi-period purchasing and inventory routing. Working Paper Series. https://doi.org/10.24352/UB.OVGU-2023-097