Optimización dinámica de portafolios con aprendizaje por refuerzo



Daniel Santiago Forero Garcia

Ciencia de Datos

Directora de carrera: Lina Maria Acosta Avena

Cotutora: Laura Juliana Mora Paez

Tutor: Gabriel Ignacio Penagos Londoño

Evaluador: Juan Carlos Garcia Gomez

Pontificia Universidad Javeriana

2025

Resumen

La construcción de portafolios financieros es un problema secuencial: en cada periodo se observa nueva información, se decide una asignación y se asumen las consecuencias de esa decisión. Este trabajo desarrolla y evalúa un pipeline reproducible para la asignación dinámica de portafolios mediante aprendizaje por refuerzo (RL), comparado contra estrategias clásicas de referencia: *Equal Weight*, Momentum y Markowitz de mínima varianza.

El universo de inversión está compuesto por diez ETFs líquidos: SPY, QQQ, XLK, XLF, XLE, XLI, XLV, TLT, IEF y GLD. El problema se formula como un Proceso de Decisión de Markov con acciones discretas: el agente selecciona entre ocho plantillas de portafolio *long-only*, con decisiones semanales y una función de recompensa que incluye costos de transacción y una penalización por riesgo. Se implementaron dos agentes, DQN y PPO, y se evaluaron bajo validación *walk-forward* con tres ventanas temporales.

Los resultados indican que el agente DQN v2 con penalización de riesgo obtuvo el mayor CAGR promedio en los periodos de prueba, con un valor de 24,81 %. Sin embargo, este resultado se alcanzó a costa de mayor rotación, mayores costos y un *drawdown* promedio más alto que el de las estrategias clásicas. En términos ajustados por riesgo, Markowitz de mínima varianza y *Equal Weight* mantuvieron mejores resultados en Sharpe y control de caídas.

La evidencia sugiere que el aprendizaje por refuerzo puede generar valor en la asignación dinámica de portafolios cuando se diseña una recompensa adecuada y se valida rigurosamente fuera de muestra. Sin embargo, no reemplaza de manera concluyente a las estrategias tradicionales. Su aporte principal está en la adaptación dinámica, no en la eficiencia ajustada por riesgo.

Palabras clave: aprendizaje por refuerzo, portafolios, DQN, PPO, ETFs, *walk-forward*, costos de transacción, riesgo.

Índice general

Resumen **1**

1. Introducción **4**
1.1. Introducción del problema 4
1.2. Planteamiento del problema 4
1.3. Justificación 5
1.4. Objetivo general 6
1.5. Objetivos específicos 6
1.6. Alcance y delimitaciones 6

2. Marco teórico **8**
2.1. Contexto y nociones básicas de portafolios 8
2.2. Teoría Moderna de Portafolio como referencia 9
2.3. Decisiones intertemporales y motivación dinámica 9
2.4. Formulación como Proceso de Decisión de Markov 10
2.4.1. Validación de la propiedad markoviana en el proceso de portafolios 10
2.5. Aprendizaje por Refuerzo: conceptos base 11
2.6. Costos de transacción y rotación de portafolio 12
2.7. Medidas de riesgo y desempeño 12
2.8. Validación temporal y prevención de sesgos 13
2.9. DQN y PPO con acciones discretas 14

3. Metodología **15**
3.1. Metodología general 15
3.1.1. Fase 1: Consolidación del marco teórico y alcance 15
3.1.2. Fase 2: Exploración y preparación de datos 15
3.1.3. Fase 3: Estrategias base y sistema de backtesting 16
3.1.4. Fase 4: Formulación del entorno de aprendizaje por refuerzo . . . 16
3.1.5. Fase 5: Entrenamiento y evaluación inicial de modelos RL 16
3.1.6. Fase 6: Robustecimiento metodológico 16
3.2. Desarrollo del proyecto 17

3.2.1. Estructura base del proyecto 17
3.2.2. Preparación y procesamiento de datos 17
3.2.3. Construcción del motor de simulación 17
3.2.4. Implementación de estrategias base 18
3.2.5. Definición del sistema de métricas 19
3.2.6. Construcción de features iniciales y división temporal 19
3.2.7. Construcción de features enriquecidas 19
3.2.8. Normalización de variables y prevención de fuga de información . 20
3.2.9. Formulación computacional del entorno de RL 21
3.2.10. Cadena de Markov asociada al entorno de portafolios 21
3.2.11. Diseño del espacio de acciones discretas 22
3.2.12. Función de recompensa con penalización de riesgo 23
3.2.13. Entrenamiento del agente DQN 23
3.2.14. Entrenamiento del agente PPO 24
3.2.15. Hiperparámetros, sintonización y semillas aleatorias 24
3.2.16. Análisis de sensibilidad a costos de transacción 25
3.2.17. Validación walk-forward 26
3.2.18. Comparación final entre estrategias 26

4. Resultados y análisis **27**
4.1. Resultados de la comparación inicial 27
4.2. Efecto de la penalización de riesgo 27
4.3. Análisis de sensibilidad a costos 28
4.4. Validación walk-forward 28
4.5. Resultados por ventana walk-forward 31
4.6. Análisis de distribución de acciones 32
4.7. Discusión general 33
4.8. Limitaciones 34
4.9. Síntesis de resultados 34

5. Conclusiones y trabajo futuro **36**
5.1. Conclusiones 36
5.2. Trabajo futuro 37

Capítulo 1

Introducción

1.1. Introducción del problema

Asignar pesos a varios activos a lo largo del tiempo, buscando buen rendimiento sin desbordar el riesgo, es un desafío clásico en finanzas. La Teoría Moderna de Portafolio describe bien el equilibrio media-varianza, pero en la práctica la decisión de inversión rara vez ocurre en un único periodo. El mercado cambia, los costos de transacción importan y las restricciones operativas condicionan cuándo conviene mover la cartera. Por eso, construir un portafolio puede entenderse como un problema secuencial: en cada periodo se observa nueva información, se decide una asignación y se asumen las consecuencias.

El aprendizaje por refuerzo, conocido en inglés como *Reinforcement Learning* (RL), ofrece un marco natural para este tipo de decisiones. Un agente observa el estado del mercado y su posición actual, selecciona una asignación y recibe una recompensa según el resultado obtenido, descontando costos y, en este trabajo, también una aproximación sencilla al riesgo.

El objetivo no es construir un modelo complejo ni opaco. Se propone una implementación deliberadamente simple, interpretable y comparable con reglas clásicas. La pregunta central es si un agente de RL, restringido a escoger entre plantillas discretas de portafolio, puede aprender una política de asignación dinámica capaz de competir con estrategias como *Equal Weight*, Momentum y Markowitz de mínima varianza.

1.2. Planteamiento del problema

En cada periodo, el inversionista debe decidir cómo distribuir el capital entre distintos activos sin conocer con certeza los retornos futuros. Las estrategias clásicas ofrecen reglas claras: *Equal Weight* distribuye de forma uniforme, Momentum sigue el desempeño reciente y Markowitz minimiza la varianza histórica. Todas funcionan bajo supuestos que el mercado no siempre cumple.

El problema surge cuando cambia el régimen. Una asignación adecuada en un periodo de crecimiento puede ser mala en uno de alta volatilidad o caídas fuertes. Además, cada rebalanceo tiene un costo, por lo que una estrategia dinámica debe equilibrar retorno, riesgo y rotación. Evaluar una política solo por su retorno acumulado no es suficiente: también importa el desempeño ajustado por riesgo, las caídas máximas y la sensibilidad a costos de transacción.

El aprendizaje por refuerzo permite formular este problema como una decisión secuencial, pero aplicarlo en finanzas trae dificultades concretas: riesgo de sobreajuste, inestabilidad fuera de muestra y sensibilidad al diseño de la función de recompensa.

Por esta razón, este trabajo plantea la siguiente pregunta de investigación:

> ¿Puede un agente de aprendizaje por refuerzo con acciones discretas construir portafolios dinámicos competitivos frente a estrategias clásicas, considerando costos de transacción, riesgo y validación temporal fuera de muestra?

El objetivo no es demostrar que RL domina siempre a los métodos tradicionales, sino evaluar bajo qué condiciones puede competir con ellos y qué limitaciones aparecen cuando se analiza su desempeño de manera rigurosa.

1.3. Justificación

Tomar decisiones de inversión depende de cómo se representa el problema. No basta con estimar retornos esperados: también importa cuándo mover la cartera, cuánto cambiar la exposición, qué costo tiene rebalancear y cómo se comporta la estrategia en periodos de pérdida. El aprendizaje por refuerzo permite formular la asignación de portafolios como una interacción continua entre el agente y el entorno financiero.

Una dificultad frecuente en estas aplicaciones es que los modelos se vuelven demasiado complejos. Cuando el agente produce pesos continuos para cada activo, la política es difícil de interpretar y el riesgo de sobreajuste aumenta. Por eso este trabajo adopta un enfoque más controlado: el agente elige entre un conjunto finito de plantillas de portafolio *long-only*, con pesos que suman 1 y con una interpretación financiera clara.

Comparar contra *baselines* transparentes como *Equal Weight*, Momentum y Markowitz, bajo la misma frecuencia semanal y el mismo esquema de costos, permite responder con evidencia cuándo un esquema de RL logra competir con reglas clásicas y cuándo no. El valor del modelo no se mide solo por el retorno acumulado, sino también por su estabilidad, su rotación y su desempeño ajustado por riesgo.

1.4. Objetivo general

Evaluar si un esquema de optimización dinámica de portafolio con aprendizaje por refuerzo, basado en acciones discretas, decisión semanal y recompensa sensible a costos de transacción y riesgo, puede igualar o superar estrategias clásicas de referencia como *Equal Weight*, Momentum y Markowitz de mínima varianza bajo protocolos de validación temporal.

1.5. Objetivos específicos

1. Formalizar el problema de asignación dinámica de portafolios como un Proceso de Decisión de Markov (*Markov Decision Process*, MDP), definiendo estado, acción, transición y recompensa.

2. Construir un pipeline reproducible de datos financieros que incluya descarga, limpieza, agregación semanal, cálculo de retornos y generación de variables explicativas sin fuga de información futura.

3. Implementar estrategias clásicas de referencia: *Equal Weight*, Momentum y Markowitz de mínima varianza, bajo las mismas condiciones de frecuencia, costos y universo de activos.

4. Definir un espacio de acciones discretas e interpretables, compuesto por plantillas de portafolio *long-only* que suman 1 y representan distintos perfiles de asignación.

5. Implementar modelos de aprendizaje por refuerzo con acciones discretas, incluyendo un agente DQN como aproximación basada en valores y un agente PPO actor–crítico como modelo de política.

6. Robustecer el entorno de aprendizaje mediante features enriquecidas, normalización sin fuga de información y una función de recompensa con penalización de riesgo.

7. Evaluar el desempeño mediante métricas de retorno, riesgo y operación, incluyendo CAGR, volatilidad anualizada, Sharpe, máximo *drawdown*, VaR, CVaR y *turnover*, sobre conjuntos de entrenamiento, validación y prueba.

8. Analizar la estabilidad de los resultados mediante sensibilidad a costos de transacción, distribución de acciones elegidas y validación *walk-forward*.

1.6. Alcance y delimitaciones

El alcance del trabajo se delimita con el fin de mantener una implementación clara, reproducible e interpretable. En primer lugar, el universo de inversión se restringe a diez

ETFs líquidos, lo que permite trabajar con activos ampliamente negociados y representativos de distintas clases de exposición: renta variable amplia, sectores económicos, bonos del Tesoro y oro.

En segundo lugar, la frecuencia de decisión es semanal. Esta elección busca reducir ruido de muy corto plazo, disminuir la rotación excesiva y mantener coherencia con un enfoque de asignación táctica de portafolio. El trabajo no utiliza datos intradía ni modela ejecución real de órdenes.

En tercer lugar, todos los portafolios son *long-only*; no se permiten posiciones cortas ni apalancamiento. Además, los pesos de cada portafolio deben sumar 1. Estas restricciones facilitan la interpretación de las estrategias y reducen la complejidad operativa del entorno.

En cuarto lugar, el agente no decide pesos continuos. En su lugar, selecciona entre un conjunto finito de plantillas discretas de portafolio. Esta decisión es central en el proyecto, ya que permite interpretar cada acción del agente como una postura financiera concreta: agresiva, defensiva, sectorial, de refugio o basada en momentum.

Finalmente, el trabajo no incorpora variables macroeconómicas externas, noticias, sentimiento de mercado ni modelos de predicción supervisada de retornos. El enfoque se concentra en evaluar si, a partir de señales históricas simples y una estructura secuencial de decisión, un agente de RL puede competir con estrategias clásicas bajo costos de transacción y validación temporal.

Capítulo 2

Marco teórico

2.1. Contexto y nociones básicas de portafolios

Un portafolio es una combinación ponderada de activos financieros, como acciones, bonos, ETFs o efectivo. Si w_i representa el peso asignado al activo i, entonces una restricción básica de presupuesto es:

$$\sum_i w_i = 1.$$

En este trabajo se consideran portafolios *long-only*, por lo que no se permiten posiciones cortas:

$$w_i \geq 0 \quad \forall i.$$

El propósito de construir un portafolio no es únicamente maximizar el retorno, sino equilibrar el crecimiento esperado con el riesgo asumido. Una estrategia puede tener un retorno acumulado alto y aun así ser poco atractiva si ese resultado vino con alta volatilidad, caídas profundas o rotación excesiva.

Precio y retorno. Sea $P_{i,t}$ el precio del activo i en el tiempo t. El retorno simple se define como:

$$R_{i,t} = \frac{P_{i,t}}{P_{i,t-1}} - 1.$$

El log–retorno asociado es:

$$r_{i,t} = \ln(1 + R_{i,t}).$$

Para un portafolio con vector de pesos w_{t-1}, el retorno simple del portafolio en el periodo t se calcula como:

$$R_{p,t} = w_{t-1}^{\top} R_t.$$

Riesgo y dependencia. La volatilidad mide la variabilidad de los retornos. La correlación entre activos indica en qué medida estos se mueven conjuntamente. La diversifi-

cación busca combinar activos que no respondan igual ante los mismos escenarios. Por eso, evaluar un portafolio solo por su retorno promedio es insuficiente: también importa la trayectoria del valor acumulado, la volatilidad, las pérdidas extremas y el máximo *drawdown*.

2.2. Teoría Moderna de Portafolio como referencia

La Teoría Moderna de Portafolio, propuesta por Markowitz, plantea que el inversionista puede construir portafolios eficientes a partir de la relación entre retorno esperado y riesgo [1]. En su forma clásica, el problema de mínima varianza se expresa como:

$$\min_{w} \; w^{\top} \Sigma w$$

sujeto a:

$$\sum_{i} w_i = 1,$$

donde Σ es la matriz de covarianzas de los retornos.

En este trabajo, la formulación de Markowitz se usa como una referencia clásica para comparar los modelos de RL. En particular, se implementa una versión *long-only* de mínima varianza. Esta elección es intencional: evita depender de estimaciones inestables de retornos esperados y permite contar con una estrategia orientada principalmente al control del riesgo.

Su limitación principal es que se trata de un problema estático. No aprende una política secuencial a partir de la interacción con el mercado, ni optimiza directamente decisiones futuras. Esa limitación es la que motiva la formulación del problema como un entorno de aprendizaje por refuerzo.

2.3. Decisiones intertemporales y motivación dinámica

La inversión es un proceso intertemporal. La decisión tomada hoy afecta el valor del portafolio en el futuro y condiciona las decisiones siguientes. La literatura de selección de portafolios en tiempo continuo ya reconoce que las decisiones óptimas dependen del estado del mercado y del horizonte de inversión [2].

En la práctica, esto significa que una estrategia de portafolio debería responder a cambios en momentum, volatilidad, comportamiento sectorial, posición actual y costos de transacción. El problema no se reduce a encontrar una asignación óptima fija, sino a aprender una regla de decisión que se adapte a lo largo del tiempo.

2.4. Formulación como Proceso de Decisión de Markov

El aprendizaje por refuerzo se apoya en la formulación de un Proceso de Decisión de Markov (MDP), definido por la tupla $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$ [3].

El **estado** contiene la información disponible para el agente en cada periodo. En este proyecto se construye con señales financieras recientes de cada activo y con los pesos actuales del portafolio. De esta forma, el agente no observa únicamente el mercado, sino también su propia posición.

La **acción** corresponde a la decisión de asignación. Para mantener interpretabilidad y reducir complejidad, se define un conjunto discreto de acciones. Cada una representa una plantilla de pesos *long-only* que suma 1.

La **dinámica** describe cómo evoluciona el entorno después de que el agente actúa. En portafolios, esta transición depende de la evolución de los precios, los retornos realizados y la actualización de los pesos.

La **recompensa** resume el resultado inmediato de una acción. En la implementación inicial se usa el log–retorno neto de costos:

$$r_t = \ln(1 + R_{p,t}^{neto}),$$

donde $R_{p,t}^{neto} = R_{p,t}^{bruto} - \text{costo}_t$. En la versión robustecida se agrega una penalización por riesgo:

$$r_t = \ln(1 + R_{p,t}^{neto}) - \lambda \cdot \text{riesgo}_{p,t}.$$

El **factor de descuento** γ determina la importancia relativa de las recompensas futuras frente a las inmediatas. El objetivo del agente es aprender una política que maximice:

$$\max_{\pi} \ \mathbb{E}_{\pi}\left[\sum_{t=0}^{T} \gamma^t r_t\right].$$

2.4.1. Validación de la propiedad markoviana en el proceso de portafolios

La formulación del problema como Proceso de Decisión de Markov requiere que el estado actual contenga información suficiente para tomar la decisión de inversión sin depender explícitamente de toda la trayectoria histórica. En este trabajo, esta propiedad se valida de manera aproximada mediante la construcción del estado s_t, el cual resume la información relevante del mercado y de la cartera al momento de decidir. El estado incluye retornos rezagados, indicadores de momentum, volatilidades móviles, drawdown reciente, fortaleza relativa frente a SPY y los pesos actuales del portafolio. Estas variables condensan información histórica en medidas observables al cierre del periodo anterior.

Por tanto, el agente decide en t usando únicamente información disponible hasta $t-1$, evitando fuga de información futura. Bajo esta representación, la transición hacia el siguiente estado puede expresarse como:

$$P(s_{t+1} \mid s_t, a_t),$$

lo que significa que el nuevo estado depende del estado actual, de la acción seleccionada y de la evolución posterior de los precios. Aunque los mercados financieros pueden presentar memoria, no estacionariedad y cambios de régimen, el diseño del estado busca capturar la información histórica necesaria para que el supuesto markoviano sea una aproximación razonable dentro del entorno computacional. Así, la propiedad markoviana no se interpreta como una afirmación fuerte sobre la naturaleza real del mercado, sino como una condición operativa del modelo: el agente recibe un estado suficientemente informativo para decidir, actuar y actualizar la cartera de forma secuencial.

2.5. Aprendizaje por Refuerzo: conceptos base

El aprendizaje por refuerzo estudia cómo un agente aprende a actuar mediante interacción con un entorno para maximizar recompensas acumuladas [5]. A diferencia del aprendizaje supervisado, el agente no recibe una etiqueta correcta en cada observación; aprende a partir de las consecuencias de sus acciones.

Política. La política $\pi(a|s)$ define cómo se selecciona una acción a dado un estado s. Puede ser determinística o estocástica.

Función de valor. La función de valor estima qué tan conveniente es estar en un estado:

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{T} \gamma^t r_t \mid s_0 = s\right].$$

Función acción–valor. La función $Q^{\pi}(s,a)$ estima el valor esperado de tomar una acción específica en un estado:

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{T} \gamma^t r_t \mid s_0 = s, a_0 = a\right].$$

Estas funciones permiten diferenciar entre métodos basados en valores, como DQN, y métodos basados en políticas, como PPO.

2.6. Costos de transacción y rotación de portafolio

El rebalanceo de un portafolio no es gratuito. Cada cambio en los pesos implica fricciones, comisiones, *bid–ask spread* o costos operativos. Por ello, una estrategia que cambia demasiado puede mostrar buenos retornos brutos, pero perder atractivo al considerar costos.

En la implementación computacional se utiliza un costo proporcional al nivel de rotación de la cartera. Sea w_{t-1} el vector de pesos anterior y w_t el vector de pesos después de la acción seleccionada. El *turnover* se define como:

$$\text{turnover}_t = \sum_i |w_{i,t} - w_{i,t-1}|.$$

El costo de transacción se calcula como:

$$\text{costo}_t = c \cdot \text{turnover}_t,$$

donde c es el costo proporcional por unidad de rotación.

Finalmente, el retorno neto del portafolio es:

$$R_{p,t}^{neto} = R_{p,t}^{bruto} - \text{costo}_t.$$

Esta aproximación es simple, pero cumple un papel importante: penaliza directamente los cambios frecuentes de portafolio y permite comparar todas las estrategias bajo la misma fricción operativa. Una posible extensión futura consiste en incorporar costos fijos, costos diferenciados por activo o reglas de no operación cuando el cambio de pesos sea muy pequeño.

2.7. Medidas de riesgo y desempeño

Las estrategias se evalúan mediante métricas que capturan crecimiento, riesgo y operación.

CAGR. El crecimiento anual compuesto mide el retorno anual equivalente de una estrategia:

$$\text{CAGR} = \left(\frac{V_T}{V_0}\right)^{1/A} - 1,$$

donde V_T es el valor final, V_0 el valor inicial y A el número de años.

Volatilidad anualizada. Si los retornos son semanales, la volatilidad anualizada se calcula como:

$$\sigma_{ann} = \sqrt{52}\,\sigma_{sem}.$$

Sharpe. El índice de Sharpe mide retorno ajustado por riesgo:

$$\text{Sharpe} = \frac{\mu_p - \mu_f}{\sigma_p}.$$

En esta implementación se asume una tasa libre de riesgo igual a cero para simplificar la comparación [10].

Máximo drawdown. El máximo *drawdown* mide la peor caída desde un máximo acumulado:

$$\text{MDD} = \min_t \left(\frac{V_t}{\max_{s \le t} V_s} - 1 \right).$$

VaR y CVaR. El VaR histórico al 95 % representa el percentil 5 % de los retornos. El CVaR estima la pérdida promedio en la cola más desfavorable:

$$\text{CVaR}_{95} = \mathbb{E}[R_t \mid R_t \le \text{VaR}_{95}].$$

Estas dos métricas son útiles para estrategias activas, donde las colas de la distribución de retornos tienden a ser más pesadas que en estrategias pasivas.

Turnover. El *turnover* promedio mide qué tanto cambia la estrategia sus posiciones. Para los modelos de RL es especialmente relevante: un agente puede obtener buen retorno a costa de rotar demasiado la cartera, lo que eleva los costos reales de operación.

2.8. Validación temporal y prevención de sesgos

En series financieras no es adecuado mezclar aleatoriamente observaciones pasadas y futuras. Hacerlo puede introducir *look-ahead bias*, es decir, permitir que el modelo aprenda indirectamente de información que no habría estado disponible en el momento de decisión.

En la implementación inicial se emplea una división temporal ordenada en tres conjuntos:

$$\text{train} \mid \text{validation} \mid \text{test}.$$

El conjunto de entrenamiento se usa para ajustar los agentes, el conjunto de validación permite observar su comportamiento fuera de muestra durante el desarrollo, y el conjunto de prueba se reserva para la evaluación final.

Como etapa de robustecimiento, se implementa además un esquema *walk-forward*, en el cual el entrenamiento y la evaluación se repiten sobre distintas ventanas temporales:

$$\underbrace{\text{Train}}_{T_{tr}} \mid \underbrace{\text{Validation}}_{T_{val}} \mid \underbrace{\text{Test}}_{T_{te}} \quad \longrightarrow \quad \text{desplazar y repetir.}$$

Esta extensión permite evaluar la estabilidad de los modelos en distintos subperiodos de mercado y reduce la dependencia de una única partición temporal.

2.9. DQN y PPO con acciones discretas

En este trabajo se implementan dos enfoques de aprendizaje por refuerzo compatibles con acciones discretas: DQN y PPO.

DQN. DQN (*Deep Q-Network*) es un método basado en valores que aproxima la función $Q(s, a)$, la cual estima qué tan conveniente es tomar una acción a en un estado s [13]. La red recibe el estado del mercado y devuelve ocho valores, uno por cada acción discreta. La acción seleccionada es la de mayor valor estimado, salvo durante la fase de exploración controlada por una política ϵ-greedy.

Se incorpora DQN por dos razones. El espacio de acciones es discreto, lo que hace natural el uso de una función $Q(s, a)$. Además, sirve como primera validación entrenable del entorno antes de implementar PPO.

PPO. PPO (*Proximal Policy Optimization*) pertenece a la familia de métodos actor–crítico [14]. El actor aprende una política $\pi_\theta(a|s)$, es decir, una distribución de probabilidad sobre las acciones disponibles. El crítico estima el valor del estado $V_\phi(s)$. Su mecanismo de *clipping* evita actualizaciones demasiado grandes, lo que mejora la estabilidad del entrenamiento en entornos donde pequeñas variaciones pueden generar comportamientos inestables.

PPO representa la alternativa más alineada con la idea de aprender directamente una política de decisión. Su desempeño se analiza junto con DQN para observar diferencias entre un agente basado en valores y uno actor–crítico.

Capítulo 3

Metodología

3.1. Metodología general

La metodología del proyecto se organiza en fases orientadas a construir una base experimental clara, reproducible y comparable. El objetivo no es partir directamente de un modelo de RL. Primero se preparan los datos y se implementan estrategias clásicas; luego se formula el entorno de decisión secuencial; finalmente se entrenan agentes de aprendizaje por refuerzo y se comparan bajo las mismas condiciones.

3.1.1. Fase 1: Consolidación del marco teórico y alcance

Se revisaron los fundamentos de teoría de portafolio, decisiones intertemporales, métricas de riesgo y aprendizaje por refuerzo. Esta fase delimitó el problema como una asignación dinámica de portafolio con restricciones simples: portafolios *long-only*, pesos que suman 1, frecuencia semanal y costos de transacción explícitos.

3.1.2. Fase 2: Exploración y preparación de datos

El universo final se compone de diez ETFs líquidos: SPY, QQQ, XLK, XLF, XLE, XLI, XLV, TLT, IEF y GLD. Los datos se obtuvieron desde *Yahoo Finance* en frecuencia diaria y se agregaron a frecuencia semanal para mantener coherencia con el horizonte de decisión del portafolio.

Tabla 3.1: Universo de activos utilizado en el proyecto

Ticker	Tipo de exposición	Interpretación financiera
SPY	Renta variable amplia	ETF asociado al S&P 500
QQQ	Renta variable crecimiento	ETF asociado al Nasdaq 100
XLK	Sectorial	Tecnología
XLF	Sectorial	Financiero
XLE	Sectorial	Energía
XLI	Sectorial	Industrial
XLV	Sectorial	Salud
TLT	Bonos del Tesoro	Bonos de largo plazo
IEF	Bonos del Tesoro	Bonos de mediano plazo
GLD	Materia prima	Oro

3.1.3. Fase 3: Estrategias base y sistema de backtesting

Antes de implementar RL, se construyeron estrategias clásicas de referencia: *Equal Weight*, Momentum y Markowitz de mínima varianza. Este paso fue necesario para validar el flujo de datos, el simulador, las métricas y la lógica de costos antes de introducir cualquier modelo de aprendizaje.

3.1.4. Fase 4: Formulación del entorno de aprendizaje por refuerzo

Con la base experimental consolidada, el problema se formuló como un MDP. El estado incluye señales financieras y pesos actuales; las acciones son plantillas discretas de portafolio; la recompensa corresponde al log–retorno neto de costos; y la transición avanza una semana en el tiempo.

3.1.5. Fase 5: Entrenamiento y evaluación inicial de modelos RL

Se implementaron dos agentes: DQN como primera aproximación basada en valores, y PPO como modelo actor–crítico. Ambos se evaluaron en conjuntos de entrenamiento, validación y prueba, y se compararon contra las estrategias clásicas bajo las mismas condiciones.

3.1.6. Fase 6: Robustecimiento metodológico

A partir de los resultados iniciales, se incorporaron mejoras concretas: features enriquecidas, normalización sin fuga de información, recompensa con penalización de riesgo,

análisis de sensibilidad a costos y validación *walk-forward*. Esta fase permitió observar no solo cuánto rindió el agente, sino también cómo tomó decisiones y qué tan estable fue su comportamiento fuera de muestra.

3.2. Desarrollo del proyecto

El desarrollo se llevó a cabo de forma progresiva. En lugar de iniciar con modelos de RL, se construyó primero una infraestructura base: datos procesados, estrategias clásicas, simulador, costos de transacción y métricas de evaluación. Así, los agentes DQN y PPO se evaluaron bajo las mismas condiciones que los *baselines*: mismo universo de activos, misma frecuencia semanal, mismo esquema de costos y mismas métricas.

3.2.1. Estructura base del proyecto

El repositorio se organizó en una estructura compacta y reproducible. La carpeta `data/` contiene los datos originales, procesados y resultados; `src/` concentra la lógica principal; y `notebooks/` se conserva como apoyo para exploración.

Dentro de `src/` se definieron módulos específicos para configuración, descarga y procesamiento de datos, construcción de features, generación de ventanas temporales, estrategias, simulación, métricas, entorno RL, agentes y comparación final. Esta separación facilita la lectura del proyecto y permite validar cada etapa de forma independiente.

3.2.2. Preparación y procesamiento de datos

Se descargaron precios diarios desde *Yahoo Finance* para los ETFs seleccionados. Las series se limpiaron, se ordenaron temporalmente y se agregaron a frecuencia semanal usando el último precio disponible de cada semana. A partir de los precios semanales se calcularon retornos simples:

$$R_{i,t} = \frac{P_{i,t}}{P_{i,t-1}} - 1.$$

Los archivos principales generados fueron precios diarios, precios semanales y rendimientos semanales. Estos últimos sirvieron como insumo para las estrategias clásicas y para la construcción de variables usadas por los agentes de RL.

3.2.3. Construcción del motor de simulación

El simulador recibe una matriz de rendimientos y un historial de pesos por fecha. Para cada periodo calcula el retorno bruto del portafolio:

$$R_{p,t}^{bruto} = \sum_i w_{i,t} R_{i,t}.$$

Luego calcula el *turnover*:

$$\text{turnover}_t = \sum_i |w_{i,t} - w_{i,t-1}|,$$

el costo de transacción:

$$\text{costo}_t = c \cdot \text{turnover}_t,$$

y el retorno neto:

$$R_{p,t}^{neto} = R_{p,t}^{bruto} - \text{costo}_t.$$

Finalmente, el valor acumulado se obtiene como:

$$V_t = V_{t-1}(1 + R_{p,t}^{neto}).$$

Este motor es central porque permite evaluar estrategias distintas bajo una misma lógica económica.

3.2.4. Implementación de estrategias base

Se implementaron tres estrategias de referencia: *Equal Weight*, Momentum y Markowitz de mínima varianza. Cubren tres enfoques complementarios: diversificación simple, selección por desempeño reciente y optimización clásica de riesgo.

Estrategia Equal Weight

La estrategia *Equal Weight* asigna el mismo peso a todos los activos:

$$w_i = \frac{1}{N}.$$

Es un benchmark simple, transparente y difícil de superar en muchos contextos empíricos.

Estrategia Momentum

La estrategia Momentum calcula el rendimiento acumulado de cada activo en una ventana histórica de 12 semanas y selecciona los tres con mejor desempeño reciente. El capital se distribuye en partes iguales entre ellos. Introduce adaptación temporal sin requerir optimización numérica.

Estrategia Markowitz de mínima varianza

La estrategia Markowitz calcula la matriz de covarianzas usando una ventana histórica y resuelve:

$$\min_w \; w^\top \Sigma w$$

sujeto a:

$$\sum_i w_i = 1, \qquad w_i \geq 0.$$

Se eligió mínima varianza porque ofrece una referencia orientada al control del riesgo y evita depender de estimaciones inestables de retornos esperados.

3.2.5. Definición del sistema de métricas

Para evaluar las estrategias se calcularon métricas de crecimiento, riesgo y operación: CAGR, volatilidad anualizada, Sharpe, máximo *drawdown*, VaR histórico al 95 %, CVaR histórico al 95 %, *turnover* promedio y *turnover* total.

Estas métricas permiten analizar no solo cuánto gana una estrategia, sino cómo obtiene ese resultado. Una estrategia con alto retorno pero excesiva rotación puede ser menos atractiva cuando los costos de transacción son relevantes.

3.2.6. Construcción de features iniciales y división temporal

Para construir el estado inicial del agente se generaron features semanales por activo:

$$ret_t,\ ret_1,\ mom_4,\ mom_{12},\ vol_4,\ vol_{12}.$$

La variable ret_t representa el retorno realizado en la semana t y se usa exclusivamente para calcular la recompensa. Las variables rezagadas funcionan como información observable por el agente.

Para evitar fuga de información futura, todas las variables de estado se rezagaron una semana con `shift(1)`. El agente decide en la semana t usando información disponible hasta $t-1$, y solo después recibe el retorno realizado en t.

Las features se dividieron respetando el orden temporal en:

$$\text{train}, \quad \text{validation}, \quad \text{test}.$$

3.2.7. Construcción de features enriquecidas

Para la versión robustecida del entorno se amplió el conjunto de variables observadas por el agente. El conjunto final de features por activo fue:

$$ret_1,\ mom_4,\ mom_{12},\ mom_{26},\ vol_4,\ vol_{12},\ vol_{26},\ dd_{12},\ rel_spy_{12}.$$

Tabla 3.2: Variables utilizadas para construir el estado del agente

Variable	**Ventana**	**Uso**	**Interpretación**
ret_t	1 semana	Reward	Retorno realizado; se usa para calcular la recompensa, no como entrada del estado.
ret_1	1 semana	Estado	Retorno semanal rezagado.
mom_4	4 semanas	Estado	Momentum de corto plazo.
mom_{12}	12 semanas	Estado	Momentum de mediano plazo.
mom_{26}	26 semanas	Estado	Momentum extendido.
vol_4	4 semanas	Estado	Volatilidad reciente.
vol_{12}	12 semanas	Estado / Reward	Señal de riesgo usada en el estado y en la penalización de la recompensa.
vol_{26}	26 semanas	Estado	Volatilidad extendida.
dd_{12}	12 semanas	Estado	Caída frente al máximo móvil de doce semanas.
rel_spy_{12}	12 semanas	Estado	Fortaleza relativa frente a SPY.

Cada variable cumple un papel específico. ret_1 captura el comportamiento más reciente del activo. Los indicadores mom_4, mom_{12} y mom_{26} registran tendencias de corto, mediano y mayor plazo. Las volatilidades vol_4, vol_{12} y vol_{26} aproximan distintos horizontes de riesgo. El *drawdown* dd_{12} mide la caída reciente frente al máximo de las últimas doce semanas. rel_spy_{12} compara el momentum del activo frente al de SPY para identificar fortaleza relativa respecto al mercado amplio.

La variable ret_t se conserva en la base de datos pero no entra al estado; se usa únicamente para calcular el retorno y la recompensa después de que el agente selecciona una acción.

Con nueve features por activo y diez activos en el universo, el componente de mercado del estado tiene 90 variables. Sumando los 10 pesos actuales del portafolio, el estado total tiene:

$$9 \times 10 + 10 = 100 \text{ variables.}$$

3.2.8. Normalización de variables y prevención de fuga de información

Las variables del estado no están naturalmente en la misma escala: los retornos semanales, los indicadores de *momentum*, las volatilidades móviles y los *drawdowns* tienen magnitudes distintas. Para mejorar la estabilidad del entrenamiento se aplicó una normalización tipo *z-score*:

$$x_t^{norm} = \frac{x_t - \mu_{train}}{\sigma_{train}},$$

donde μ_{train} y σ_{train} se calcularon exclusivamente sobre el conjunto de entrenamiento.

Este punto es metodológicamente importante. Si la media y la desviación se calcularan sobre el conjunto completo de datos, la información de validación y prueba influiría indirectamente en el entrenamiento. Por eso los parámetros de normalización se estimaron solo en *train* y se aplicaron sin modificación a *validation* y *test*.

Las variables originales en escala real se conservaron para el cálculo de retornos y recompensas. El agente recibe un estado normalizado, pero las métricas financieras se calculan con retornos reales.

3.2.9. Formulación computacional del entorno de RL

El entorno conecta los elementos del problema secuencial: estado, acción, recompensa y transición.

El **estado** está compuesto por las features de los diez activos y por los pesos actuales del portafolio. En la versión robustecida tiene 100 variables.

La **acción** es un entero entre 0 y 7. Cada número representa una plantilla de portafolio definida previamente, lo que evita que el agente produzca pesos continuos difíciles de interpretar.

La **recompensa** inicial se calcula como:

$$r_t = \ln(1 + R_{p,t}^{neto}).$$

En la versión robustecida se agrega una penalización por riesgo:

$$r_t = \ln(1 + R_{p,t}^{neto}) - \lambda \cdot \text{riesgo}_{p,t}.$$

La **transición** avanza una semana: el entorno convierte la acción en pesos, calcula el retorno neto, actualiza el valor acumulado y pasa al siguiente periodo.

Antes de entrenar los agentes, el entorno se validó con una política aleatoria. El objetivo no fue competir con las estrategias, sino verificar que el entorno recorriera correctamente las fechas, transformara acciones en pesos y calculara costos y recompensas de forma consistente.

3.2.10. Cadena de Markov asociada al entorno de portafolios

La cadena de Markov que sustenta el MDP está dada por la sucesión de estados observados por el agente:

$$s_0, s_1, s_2, \ldots, s_T.$$

Cada estado s_t representa una fotografía del mercado y del portafolio en una semana específica. En cada paso temporal, el agente observa s_t, selecciona una acción discreta

a_t, el entorno transforma esa acción en una plantilla de pesos w_t, calcula el retorno neto de costos y genera el siguiente estado s_{t+1}. La transición del sistema combina dos componentes. El primero es exógeno y corresponde a la evolución de los precios de los ETFs. El segundo es endógeno y corresponde a la actualización de la cartera después del rebalanceo elegido por el agente. Por esta razón, la cadena de Markov no describe únicamente la evolución del mercado, sino la evolución conjunta entre mercado, posición del portafolio y decisiones de inversión. Formalmente, la dinámica se resume como:

$$s_t \xrightarrow{a_t} (r_t, s_{t+1}),$$

donde r_t es la recompensa obtenida después de aplicar la acción. Esta estructura permite representar el problema de inversión como una secuencia de decisiones en la que cada acción afecta los pesos futuros, los costos de transacción, el riesgo asumido y las recompensas acumuladas.

3.2.11. Diseño del espacio de acciones discretas

El espacio de acciones se diseñó para mantener interpretabilidad financiera. Cada acción representa una plantilla de portafolio *long-only* cuyos pesos suman 1.

Tabla 3.3: Acciones discretas definidas para el agente de RL

Acción	Nombre	Intuición financiera
0	Equal Weight	Diversificación uniforme entre todos los activos.
1	Equity agresivo	Alta exposición a renta variable estadounidense.
2	Tecnología / crecimiento	Concentración en QQQ y XLK, asociados a crecimiento y tecnología.
3	Sectores diversificados	Exposición balanceada a sectores accionarios.
4	Defensivo bonos	Mayor peso en TLT e IEF, reduciendo exposición accionaria.
5	Refugio bonos / oro	Combinación de bonos y oro como activos defensivos.
6	Barbell riesgo / refugio	Mezcla de acciones, bonos y oro.
7	Momentum top 3	Selección dinámica de los tres activos con mejor momentum reciente.

Esta estructura hace que la política aprendida sea interpretable. Si el agente elige repetidamente la acción 2, está favoreciendo un perfil de crecimiento tecnológico. Si elige las acciones 4 o 5, adopta una postura defensiva. Esa legibilidad es una ventaja frente a modelos que producen pesos continuos sin una interpretación directa.

3.2.12. Función de recompensa con penalización de riesgo

En la primera versión del entorno, la recompensa se definió como el log-retorno neto de costos:

$$r_t = \ln(1 + R_{p,t}^{neto}).$$

Esta formulación premia el crecimiento del portafolio después de costos, pero no penaliza la exposición al riesgo.

La segunda versión incorpora una penalización sencilla:

$$r_t = \ln(1 + R_{p,t}^{neto}) - \lambda \cdot \text{riesgo}_{p,t}.$$

El riesgo del portafolio se aproximó como la volatilidad promedio ponderada de los activos seleccionados:

$$\text{riesgo}_{p,t} = \sum_i w_{i,t} \cdot vol_{12,i,t}.$$

En lugar de estimar una matriz de covarianzas dentro del entorno, se usa la volatilidad móvil de doce semanas, que ya está disponible en las features. Así el agente recibe una señal que premia el retorno neto y desincentiva asignaciones concentradas en activos con alta volatilidad reciente.

Se usó $\lambda = 0{,}05$ como primera aproximación. Este valor no pretende ser óptimo, sino servir para analizar si la penalización de riesgo mejora la estabilidad de la política aprendida.

3.2.13. Entrenamiento del agente DQN

El primer agente entrenado fue DQN. Recibe el estado y devuelve ocho valores Q, uno por cada acción discreta. Durante el entrenamiento se usó una política ϵ-greedy: al inicio el agente explora con alta probabilidad y, a medida que avanza, reduce la exploración para explotar las acciones que estima más convenientes.

DQN cumplió dos funciones en el proyecto. Primero, validó que el entorno era entrenable con acciones discretas. Segundo, ofreció una comparación directa frente a PPO. Para estabilizar el aprendizaje se incorporaron una memoria de experiencias y una red objetivo. La versión DQN v2 añadió features enriquecidas, normalización y penalización de riesgo.

3.2.14. Entrenamiento del agente PPO

PPO se implementó como agente actor–crítico. El actor aprende una distribución de probabilidad sobre las ocho acciones disponibles; el crítico estima el valor esperado del estado. En evaluación se selecciona la acción con mayor probabilidad estimada.

Su mecanismo de *clipping* evita cambios excesivos de política durante el entrenamiento, lo que resulta útil en problemas financieros donde pequeñas variaciones pueden generar comportamientos inestables.

3.2.15. Hiperparámetros, sintonización y semillas aleatorias

Para mantener la reproducibilidad del experimento, se definieron hiperparámetros fijos para los modelos de aprendizaje por refuerzo. La selección no se hizo de forma arbitraria. Primero se tomaron valores de referencia comunes en implementaciones estándar de DQN y PPO. Luego se realizaron pruebas controladas sobre el conjunto de validación, observando estabilidad del entrenamiento, retorno acumulado, Sharpe, drawdown y turnover. En DQN se evaluaron principalmente el número de episodios, el factor de descuento γ, la tasa de aprendizaje, el tamaño del batch, el tamaño de la memoria de repetición y el esquema de exploración ϵ-greedy. En PPO se consideraron el factor de descuento, la tasa de aprendizaje, el parámetro de clipping, el coeficiente de valor y el coeficiente de entropía. El criterio de selección priorizó configuraciones que no solo obtuvieran mayor retorno, sino que mostraran mejor desempeño fuera de muestra y menor inestabilidad entre entrenamiento, validación y prueba. Por esta razón, se descartaron configuraciones con alto retorno en entrenamiento pero fuerte deterioro en validación, ya que eran señales de sobreajuste. También se controló la aleatoriedad mediante semillas fijas. Las semillas afectan la inicialización de las redes neuronales, la exploración del agente, el muestreo de experiencias en DQN y la selección estocástica de acciones durante el entrenamiento. Para reducir la dependencia de una única corrida favorable, las configuraciones principales se probaron con distintas semillas y se conservaron aquellas con comportamiento más estable en validación. La Tabla 3.4 resume los valores utilizados en los experimentos finales.

Tabla 3.4: Hiperparámetros principales de los modelos RL

Componente	Parámetro	Valor
DQN / DQN v2	Episodios	200
DQN / DQN v2	Factor de descuento γ	0.99
DQN / DQN v2	Learning rate	0.001
DQN / DQN v2	Batch size	32
DQN / DQN v2	Tamaño memoria replay	5000
DQN / DQN v2	Epsilon inicial	1.00
DQN / DQN v2	Epsilon final	0.05
DQN / DQN v2	Decaimiento epsilon	0.98
DQN v2	Arquitectura	$100 \to 64 \to 32 \to 8$
PPO	Episodios	200
PPO	Factor de descuento γ	0.99
PPO	Learning rate	0.0003
PPO	Clip epsilon	0.20
PPO	Coeficiente de valor	0.50
PPO	Coeficiente de entropía	0.01
Reward v2	λ de riesgo	0.05
Costos	Costo base	0.0010
Sensibilidad	Costos evaluados	0.0005, 0.0010, 0.0020
Semillas	Semillas aleatorias	42, 123, 2025

Nota: La arquitectura $100 \to 64 \to 32 \to 8$ describe las capas de la red neuronal del agente DQN v2. La primera capa recibe las 100 variables del estado, las capas intermedias reducen progresivamente la dimensión y la última produce los ocho valores Q correspondientes a las ocho acciones discretas.

3.2.16. Análisis de sensibilidad a costos de transacción

Además del costo base, se evaluaron tres niveles de fricción operativa:

$$c \in \{0{,}0005,\ 0{,}0010,\ 0{,}0020\}.$$

El análisis se realizó de forma *ex-post*: se mantuvieron las acciones y pesos generados por cada política y se recalcularon los costos, retornos netos y métricas bajo cada nivel. Esto permite observar qué tan sensible es cada estrategia a las fricciones operativas sin necesidad de reentrenar los modelos.

Los agentes de RL son más vulnerables a este análisis porque tienden a aprender políticas más activas. Una política con alto *turnover* puede verse afectada de forma significativa cuando los costos aumentan.

3.2.17. Validación walk-forward

Se implementó una validación tipo *walk-forward* para evaluar la estabilidad del modelo en diferentes subperiodos de mercado. Se construyeron tres ventanas, cada una con:

$$\text{90 semanas de entrenamiento}, \quad \text{20 semanas de validación}, \quad \text{20 semanas de prueba}.$$

Después de cada ventana, el bloque temporal avanza. La estructura es:

$$\text{Train}_1|\text{Val}_1|\text{Test}_1 \quad \rightarrow \quad \text{Train}_2|\text{Val}_2|\text{Test}_2 \quad \rightarrow \quad \text{Train}_3|\text{Val}_3|\text{Test}_3.$$

En cada ventana los parámetros de normalización se calculan solo con el conjunto de entrenamiento correspondiente y se aplican sin modificación a validación y prueba. Se entrenó el agente DQN v2 con penalización de riesgo y se comparó su desempeño contra *Equal Weight*, Momentum y Markowitz de mínima varianza en las mismas fechas.

3.2.18. Comparación final entre estrategias

Una vez entrenados los modelos de RL, se construyó una comparación final entre:

$$\text{Equal Weight}, \quad \text{Momentum}, \quad \text{Markowitz Min Var}, \quad \text{DQN}, \quad \text{PPO}.$$

Para la etapa robustecida, se comparó DQN v2 con penalización de riesgo contra los *baselines* clásicos bajo el esquema *walk-forward*. Esta comparación permite observar no solo qué estrategia obtiene mayor retorno, sino también cuál presenta mejor desempeño ajustado por riesgo, menor *drawdown* y menor rotación.

Capítulo 4

Resultados y análisis

4.1. Resultados de la comparación inicial

La primera etapa de resultados comparó las estrategias clásicas con los modelos de aprendizaje por refuerzo usando una partición temporal simple en entrenamiento, validación y prueba. Se evaluaron *Equal Weight*, Momentum, Markowitz de mínima varianza, DQN y PPO.

DQN alcanzó un desempeño competitivo en el conjunto de prueba, especialmente en crecimiento acumulado. Sin embargo, su desempeño en entrenamiento fue considerablemente superior al observado fuera de muestra, lo que indica sobreajuste. PPO logró una política estable y de baja rotación, pero tendió a concentrarse en una única acción dominante, lo cual limitó su capacidad adaptativa.

Estos resultados señalaron dos ajustes necesarios: enriquecer el estado del agente con nuevas variables financieras y normalización sin fuga de información, y modificar la función de recompensa para incluir una penalización explícita por riesgo. A partir de esas observaciones se construyó la versión DQN v2 con penalización de riesgo.

4.2. Efecto de la penalización de riesgo

La versión DQN v2 sin penalización incorporó features enriquecidas y normalizadas, pero no mejoró de forma consistente la generalización. En validación el modelo mostró un desempeño débil, lo que indica que aumentar la información disponible para el agente no es suficiente si la señal de recompensa sigue premiando únicamente el retorno neto.

Al incorporar la penalización de riesgo el comportamiento cambió. En el conjunto de prueba, DQN v2 con riesgo obtuvo mayor CAGR, mejor Sharpe, menor drawdown y menor *turnover* que la versión sin riesgo. La recompensa ajustada orientó al agente hacia decisiones más estables.

El resultado confirma que el diseño de la recompensa es determinante. Premiar solo

el retorno neto lleva al agente a favorecer asignaciones más agresivas. Descontar una medida de riesgo, aunque sea simple, produce políticas con mejor comportamiento fuera de muestra.

4.3. Análisis de sensibilidad a costos

El análisis evaluó el comportamiento de las estrategias bajo tres niveles de costo de transacción: bajo (0.05 %), base (0.10 %) y alto (0.20 %). La pregunta central es si las estrategias más activas mantienen su atractivo cuando aumentan las fricciones operativas.

Equal Weight no se ve afectada porque mantiene pesos constantes. Markowitz presenta baja sensibilidad por su *turnover* reducido. Momentum tiene sensibilidad intermedia. Las versiones de DQN son las más afectadas: su mayor rotación implica que cada aumento en costos erosiona más el retorno neto.

DQN v2 con penalización de riesgo mostró mejor desempeño ajustado que DQN v2 sin penalización en todos los niveles evaluados. Incorporar riesgo en la recompensa no solo estabiliza la política, sino que también reduce parcialmente la sensibilidad a costos al incentivar menor rotación.

4.4. Validación walk-forward

Para evaluar la robustez temporal del modelo se implementó una validación *walk-forward* con tres ventanas. En cada una, el agente DQN v2 con penalización de riesgo se entrenó sobre el conjunto de entrenamiento y se evaluó en validación y prueba, comparado contra *Equal Weight*, Momentum y Markowitz bajo las mismas fechas y costos.

La Tabla 4.1 presenta el promedio de resultados en los conjuntos de prueba de las tres ventanas.

Tabla 4.1: Resumen promedio en test bajo validación walk-forward

Estrategia	Valor final prom.	CAGR prom.	Sharpe prom.	Max Drawdown prom.	Turnover prom.	Costo total prom.
DQN v2 riesgo WF	1.0835	0.2481	1.7599	-0.0924	0.6389	0.0128
Equal Weight	1.0540	0.1507	1.8564	-0.0507	0.0000	0.0000
Momentum	1.0432	0.1402	1.1748	-0.0583	0.3222	0.0064
Markowitz Min Var	1.0332	0.0959	1.9121	-0.0481	0.0174	0.0003

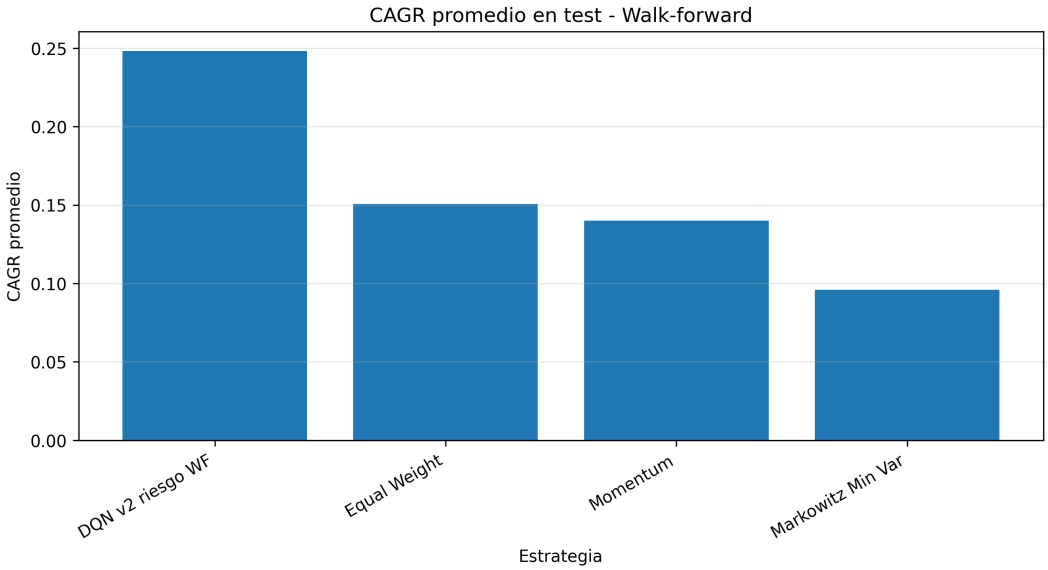


Figura 4.1: CAGR promedio en test bajo validación walk-forward

DQN v2 con penalización de riesgo alcanzó el mayor CAGR promedio en los periodos de prueba (24.81 %), superando a *Equal Weight* (15.07 %), Momentum (14.02 %) y Markowitz (9.59 %). El agente capturó oportunidades dinámicas que las estrategias clásicas no aprovecharon con la misma intensidad. Este resultado, sin embargo, debe leerse junto con las métricas de riesgo.

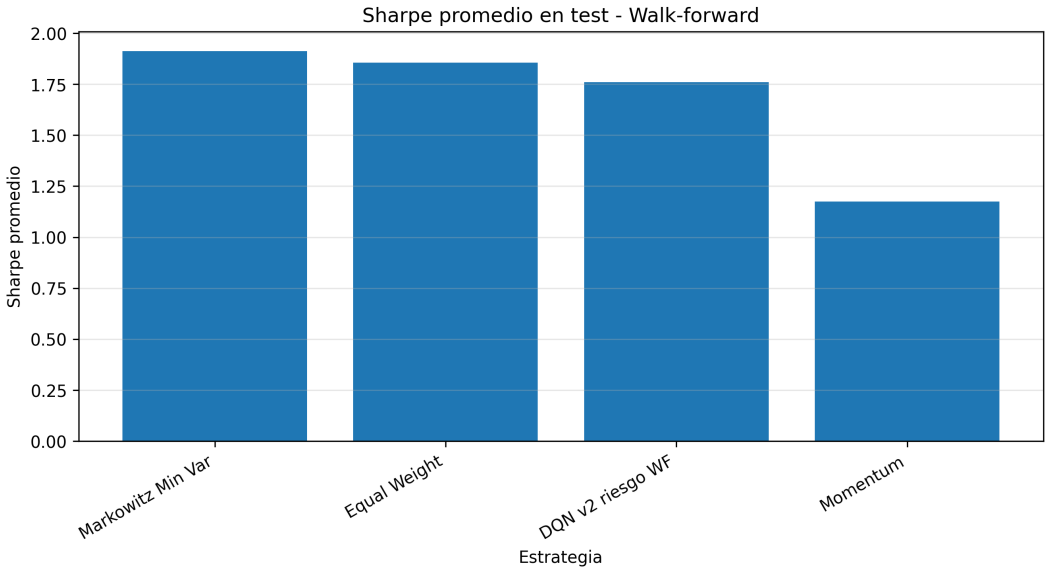


Figura 4.2: Sharpe promedio en test bajo validación walk-forward

En términos ajustados por riesgo el orden se invierte. Markowitz obtuvo el Sharpe más alto (1.91), seguido de *Equal Weight* (1.86). DQN v2 quedó en tercer lugar (1.76), por encima de Momentum (1.17). Mayor crecimiento no implicó mejor relación retorno-volatilidad.

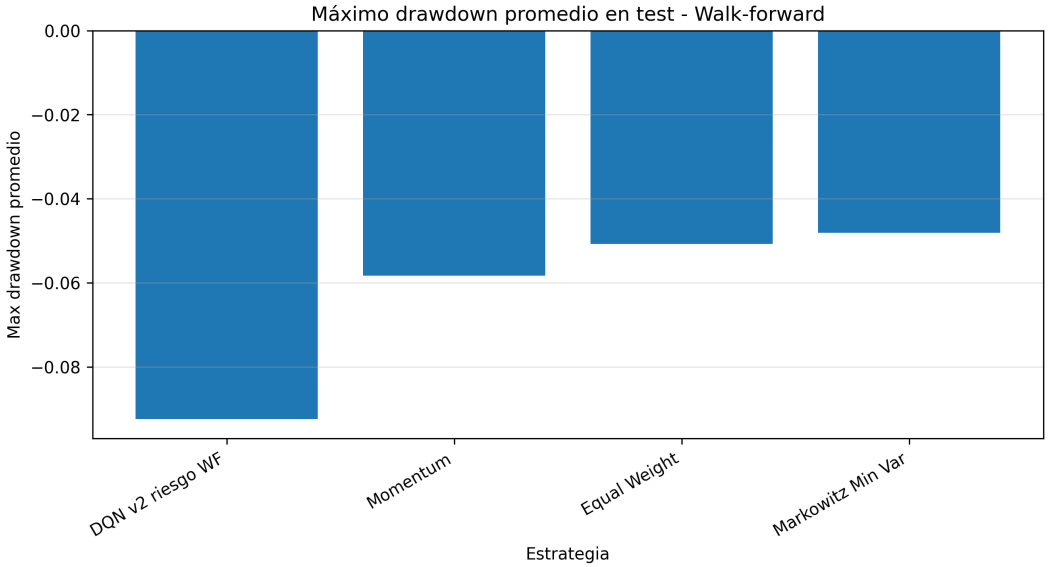


Figura 4.3: Máximo drawdown promedio en test bajo validación walk-forward

El drawdown promedio de DQN v2 fue de -9.24 %, el más alto entre las cuatro estrategias. Markowitz registró el menor (-4.81 %), seguido de *Equal Weight* (-5.07 %). El mayor CAGR del agente vino acompañado de una exposición más alta a caídas. Su aporte se concentra en crecimiento, no en estabilidad del capital.

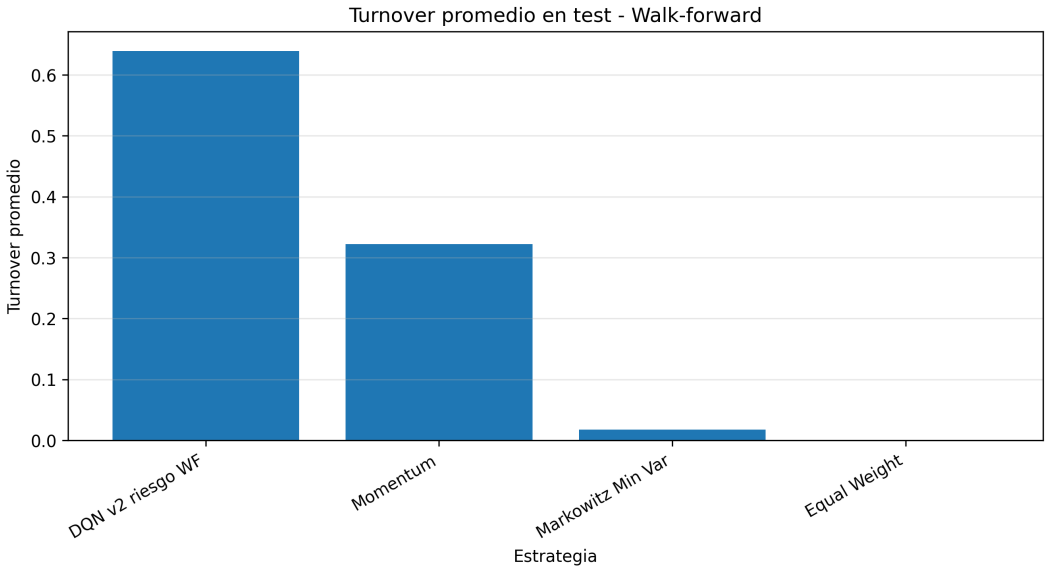


Figura 4.4: Turnover promedio en test bajo validación walk-forward

DQN v2 tuvo el mayor *turnover* promedio (0.64), muy por encima de Momentum (0.32), Markowitz (0.02) y *Equal Weight* (0.00). Esta rotación explica tanto su sensibilidad a costos como el costo total promedio de 1.28 %, frente a 0.64 % de Momentum y prácticamente cero de las otras dos estrategias.

El balance general es claro: DQN v2 lidera en crecimiento promedio pero no en eficiencia ajustada por riesgo. Markowitz y *Equal Weight* son más estables, menos costosas

y más predecibles. El agente de RL muestra potencial, pero asume más rotación y mayor exposición a caídas para obtener ese crecimiento adicional.

4.5. Resultados por ventana walk-forward

La Tabla 4.2 desglosa los resultados por ventana. El desempeño del agente no fue homogéneo: ganó en algunas ventanas y quedó por debajo en otras.

Tabla 4.2: Resultados en test por ventana walk-forward

Ventana	Estrategia	Valor final	CAGR	Sharpe	Max Drawdown	Turnover
window_01	DQN v2 riesgo WF	1.0127	0.0334	0.2648	-0.0658	0.9500
window_01	Equal Weight	1.0245	0.0650	0.5879	-0.0389	0.0000
window_01	Momentum	1.0327	0.0873	0.8289	-0.0306	0.3667
window_01	Markowitz Min Var	1.0245	0.0650	0.5879	-0.0389	0.0000
window_02	DQN v2 riesgo WF	1.0339	0.0906	0.4396	-0.1860	0.4600
window_02	Equal Weight	1.0225	0.0595	0.4331	-0.0978	0.0000
window_02	Momentum	0.9193	-0.1966	-1.2214	-0.1193	0.4000
window_02	Markowitz Min Var	0.9662	-0.0856	-0.5808	-0.0978	0.0388
window_03	DQN v2 riesgo WF	1.2040	0.6203	4.5755	-0.0255	0.5067
window_03	Equal Weight	1.1151	0.3276	4.5482	-0.0155	0.0000
window_03	Momentum	1.1776	0.5297	3.9169	-0.0249	0.2000
window_03	Markowitz Min Var	1.1090	0.3085	5.7292	-0.0077	0.0135

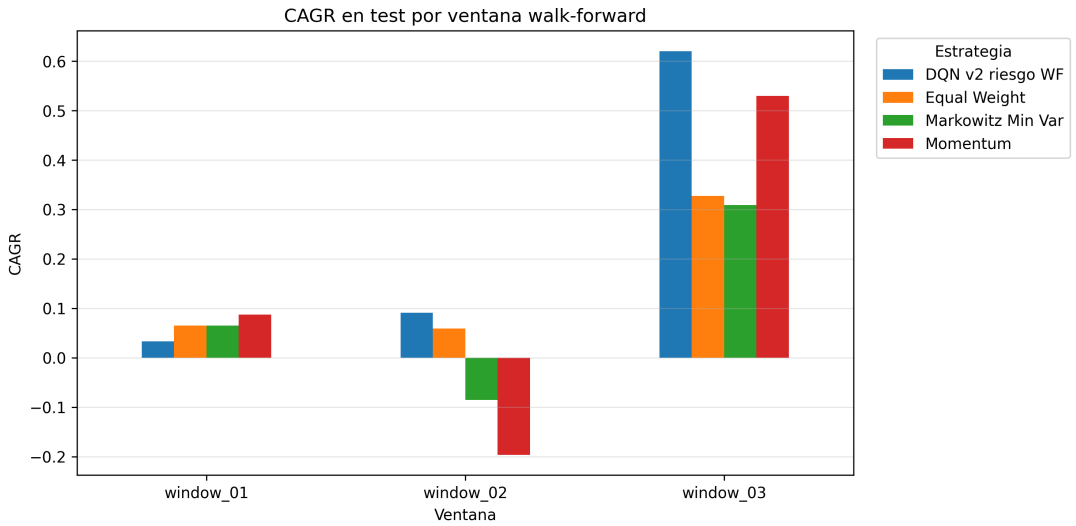


Figura 4.5: CAGR en test por ventana walk-forward

En la primera ventana DQN v2 obtuvo el menor CAGR (3.34 %) y el mayor drawdown (-6.58 %), mientras Momentum lideró con 8.73 %. En la segunda ventana el resultado

se invirtió: DQN fue el único modelo dinámico con retorno positivo (9.06 %), mientras Momentum cayó -19.66 % y Markowitz -8.56 %. En la tercera ventana DQN alcanzó su mejor desempeño (62.03 % CAGR), aunque Markowitz mantuvo el Sharpe más alto (5.73 frente a 4.58 de DQN).

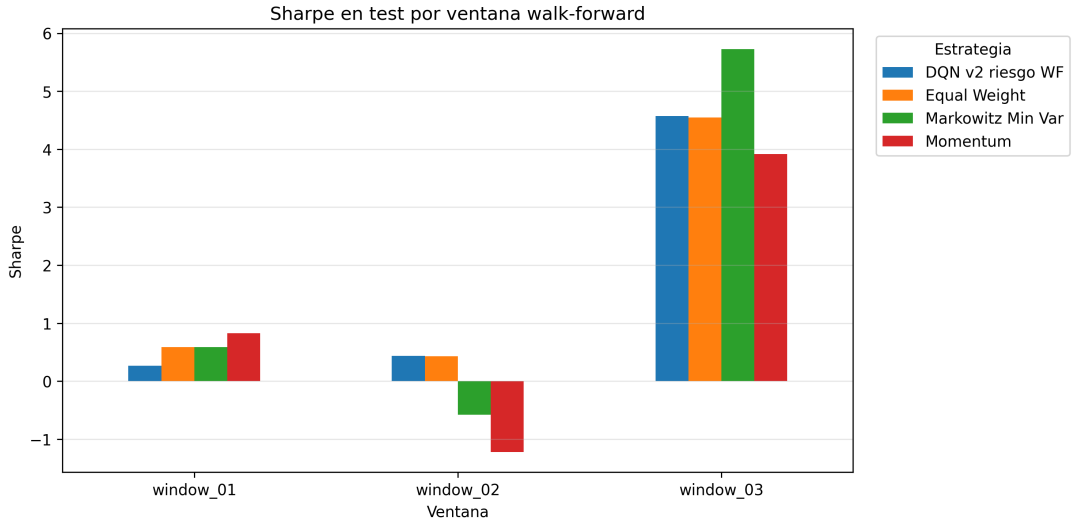


Figura 4.6: Sharpe en test por ventana walk-forward

El Sharpe por ventana confirma que ninguna estrategia domina en todos los periodos. DQN v2 tuvo el peor Sharpe en la primera ventana (0.26) pero fue el único positivo relevante en la segunda (0.44). En la tercera, todas las estrategias mostraron Sharpe alto, con Markowitz a la cabeza. Esta variabilidad es la razón principal por la que la validación walk-forward es más informativa que una única partición temporal.

4.6. Análisis de distribución de acciones

Se analizó qué acciones eligió el agente DQN v2 con penalización de riesgo en cada ventana de prueba. El objetivo es entender si la política aprendida tiene estructura o si el agente simplemente repite una misma plantilla.

La Tabla 4.3 resume la acción más utilizada en cada ventana.

Tabla 4.3: Acción más usada por DQN v2 con riesgo en test walk-forward

Ventana	Acción más usada	Porcentaje	Acciones usadas	HHI
window_01	Tecnología / crecimiento	0.45	4	0.375
window_02	Tecnología / crecimiento	0.60	3	0.440
window_03	Momentum top 3	0.60	3	0.485

Nota: El HHI (índice de Herfindahl-Hirschman) mide la concentración de las decisiones del agente. Se calcula como HHI $= \sum_a p_a^2$, donde p_a es la proporción de veces que se eligió la acción a. Un valor de 1 indica que el agente usó una única acción en todo el periodo; valores cercanos a $1/8 = 0{,}125$ indican distribución uniforme entre las ocho acciones disponibles.

En la primera ventana el agente combinó tecnología/crecimiento (45 %) con refugio bonos/oro. En la segunda se concentró más en tecnología/crecimiento (60 %), apoyado por sectores diversificados y bonos defensivos. En la tercera dominó Momentum top 3 (60 %), lo que coincide con el fuerte desempeño de activos de crecimiento en ese periodo.

El agente usó entre 3 y 4 acciones distintas en cada ventana, nunca colapso hacia una sola plantilla. El HHI de concentración fue moderado (entre 0.375 y 0.485), lo que indica cierta diversidad en las decisiones. Que la acción dominante cambie entre ventanas sugiere que el modelo responde al régimen de mercado, aunque todavía con margen para usar de forma más equilibrada todo el espacio de acciones disponible.

4.7. Discusión general

El resultado más claro del experimento es el intercambio entre crecimiento y estabilidad. DQN v2 con penalización de riesgo lideró en CAGR promedio (24.81 %) pero quedó tercero en Sharpe y tuvo el mayor drawdown y *turnover*. Markowitz y *Equal Weight* fueron más estables, menos costosas y más consistentes entre ventanas.

Esto no significa que el RL sea inferior. Significa que su ventaja depende del criterio de evaluación y del régimen de mercado. En la segunda ventana, cuando Momentum y Markowitz tuvieron retornos negativos, DQN fue el único modelo dinámico con resultado positivo. Esa capacidad de adaptación tiene valor real, pero no aparece si se mira solo el promedio.

El diseño de la recompensa fue el factor que más influyó en la calidad del modelo. Sin penalización de riesgo, DQN v2 no generalizó bien. Con penalización, el agente tomó decisiones más conservadoras y obtuvo mejores resultados fuera de muestra. En finanzas, la función de recompensa no es un detalle técnico: define qué aprende el agente.

La sensibilidad a costos es otra limitación concreta del enfoque. Con un *turnover* promedio de 0.64, DQN v2 genera costos que erosionan parte de su ventaja en retorno bruto. A medida que los costos aumentan, su atractivo relativo frente a estrategias pasivas

disminuye. Cualquier aplicación práctica de este tipo de agente debe considerar ese trade-off de forma explícita.

4.8. Limitaciones

El periodo de datos disponible es relativamente corto para entrenar modelos de aprendizaje profundo. Con tres ventanas de 90 semanas de entrenamiento cada una, existe riesgo de que el agente aprenda patrones específicos de ciertos subperiodos que no se repiten en otros contextos.

El sobreajuste fue visible en varias ventanas: el desempeño en entrenamiento superó al de validación y prueba de forma consistente. El modelo se adapta bien al periodo in-sample pero no siempre traslada esa capacidad fuera de muestra.

La función de recompensa, aunque mejorada, sigue siendo una aproximación. La penalización por volatilidad ponderada no captura la estructura de covarianzas entre activos ni los riesgos de cola. Incorporar CVaR o drawdown directamente en la recompensa podría producir políticas más robustas.

El análisis de sensibilidad a costos se realizó de forma *ex-post*: se evaluaron las políticas ya entrenadas bajo distintos niveles de costo, sin reentrenar los modelos. Una extensión más rigurosa consistiría en entrenar los agentes bajo cada escenario de costos para observar si la política aprendida cambia.

PPO no fue robustecido en la misma medida que DQN. Su política inicial tendió a concentrarse en una sola acción, lo que limitó su utilidad comparativa. Ajustar el coeficiente de entropía, la arquitectura o la función de recompensa podría mejorar su diversidad.

El espacio de acciones discretas, aunque interpretable, restringe la capacidad del agente para encontrar asignaciones intermedias. Es una decisión metodológica justificada, pero implica renunciar a la flexibilidad que ofrecen los pesos continuos.

4.9. Síntesis de resultados

El agente DQN v2 con penalización de riesgo fue la estrategia con mayor crecimiento promedio en los periodos de prueba bajo validación walk-forward. Ese resultado no se explica por suerte en una sola partición: en dos de las tres ventanas el agente superó en CAGR a al menos dos de los tres *baselines* clásicos.

Las estrategias clásicas, sin embargo, conservaron ventajas importantes. Markowitz de mínima varianza y *Equal Weight* tuvieron mejor Sharpe, menor drawdown y menor rotación. Son más predecibles y menos sensibles a costos. Para un inversionista que prioriza estabilidad sobre crecimiento, siguen siendo opciones difíciles de superar.

El aporte del trabajo no está solo en los resultados del agente. Está en el pipeline: datos reproducibles, features sin fuga de información, costos explícitos, acciones interpretables y validación temporal rigurosa. Ese marco permite evaluar estrategias dinámicas de portafolio con criterios claros, y es la base sobre la cual se pueden construir mejoras futuras.

Capítulo 5

Conclusiones y trabajo futuro

5.1. Conclusiones

Este trabajo construyó y evaluó un pipeline reproducible para la asignación dinámica de portafolios mediante aprendizaje por refuerzo. El sistema integra descarga y procesamiento de datos, construcción de variables financieras sin fuga de información, simulación de portafolios, estrategias clásicas de referencia, entorno RL con acciones discretas, entrenamiento de agentes y evaluación bajo validación temporal. Cada etapa fue diseñada para ser independiente, verificable y comparable.

Formular el problema con acciones discretas resultó útil para mantener interpretabilidad. En lugar de permitir que el agente genere pesos continuos, se le restringe a seleccionar entre plantillas de portafolio con significado financiero claro: agresiva, defensiva, sectorial, de refugio o basada en momentum. Esa restricción no elimina capacidad de adaptación, pero sí hace que cada decisión del agente sea legible.

El aprendizaje por refuerzo puede ser competitivo frente a estrategias clásicas, pero su ventaja depende del criterio de evaluación. Bajo validación walk-forward, DQN v2 con penalización de riesgo alcanzó el mayor CAGR promedio en prueba (24.81 %), superando a *Equal Weight* (15.07 %), Momentum (14.02 %) y Markowitz (9.59 %). Sin embargo, no obtuvo el mejor Sharpe ni el menor drawdown. Las estrategias clásicas mantuvieron ventajas en eficiencia ajustada por riesgo y estabilidad operativa.

El diseño de la recompensa fue el componente más determinante del proyecto. DQN v2 sin penalización de riesgo no generalizó bien fuera de muestra. Al incorporar una penalización basada en volatilidad ponderada, el agente tomó decisiones más conservadoras y obtuvo mejores resultados en validación y prueba. En finanzas, premiar solo el retorno neto lleva al agente hacia políticas agresivas que no necesariamente funcionan fuera del periodo de entrenamiento.

La validación walk-forward aportó una lectura más realista que una única partición temporal. El agente no se comportó igual en todas las ventanas: en la primera quedó por

debajo de los *baselines*, en la segunda fue el único modelo dinámico con retorno positivo cuando Momentum y Markowitz cayeron, y en la tercera obtuvo el mayor CAGR. Esa variabilidad confirma que evaluar un modelo en un solo conjunto de prueba puede llevar a conclusiones equivocadas.

Los costos de transacción y la rotación son determinantes para evaluar el atractivo real del agente. DQN v2 tuvo un *turnover* promedio de 0.64, frente a 0.32 de Momentum y prácticamente cero de *Equal Weight* y Markowitz. Ese nivel de rotación genera costos que erosionan parte de la ventaja en retorno bruto. A medida que los costos operativos aumentan, el margen del agente sobre las estrategias pasivas se reduce.

El análisis de distribución de acciones mostró que el agente no colapsó hacia una única plantilla. Usó entre 3 y 4 acciones distintas por ventana, con una acción dominante que cambió entre periodos: tecnología/crecimiento en las dos primeras ventanas y Momentum top 3 en la tercera. Ese cambio coincide con el régimen de mercado de cada periodo, lo que sugiere que la política tiene cierta capacidad de adaptación. El HHI de concentración, sin embargo, indica que todavía usa solo una fracción del espacio de acciones disponible.

En síntesis, el aprendizaje por refuerzo no reemplaza de manera concluyente a las estrategias tradicionales en este experimento. Pero sí muestra potencial como herramienta complementaria, especialmente en periodos donde la adaptación dinámica genera oportunidades que las reglas estáticas no capturan.

5.2. Trabajo futuro

La función de recompensa es el componente con más margen de mejora. La penalización por volatilidad ponderada es una aproximación útil pero limitada: no captura la estructura de covarianzas entre activos ni los riesgos de cola. Incorporar la volatilidad completa del portafolio mediante matrices de covarianza, penalizaciones por drawdown o CVaR directamente en la recompensa podría producir políticas más robustas y con menor exposición a pérdidas extremas.

Reducir la rotación del agente es otra dirección concreta. Aunque los costos de transacción ya entran en el entorno, DQN sigue presentando mayor *turnover* que las estrategias clásicas. Penalizaciones adicionales por cambio de acción, reglas de no operación o umbrales mínimos de rebalanceo podrían hacer que el agente aprenda políticas más parsimoniosas sin sacrificar capacidad de adaptación.

Ampliar el esquema walk-forward con más ventanas o periodos históricos más largos permitiría evaluar si el agente mantiene su desempeño en distintos regímenes de mercado: crisis, recuperaciones, periodos laterales, alta inflación o cambios en tasas de interés. Con solo tres ventanas los resultados son indicativos, no concluyentes.

PPO quedó sin robustecimiento en este trabajo. Su política inicial tendió a concentrarse en una sola acción, lo que limitó su utilidad comparativa. Ajustar el coeficiente

de entropía, modificar la arquitectura de red o cambiar la función de recompensa podría mejorar la diversidad de su política y hacer la comparación con DQN más informativa.

Entrenar los agentes bajo distintos niveles de costos de transacción, en lugar de evaluar las políticas ya entrenadas de forma *ex-post*, permitiría observar si el agente aprende a ser más conservador cuando enfrenta mayores fricciones desde el inicio del entrenamiento. Es probable que un agente entrenado con costos altos desarrolle una política con menor rotación que uno entrenado con costos bajos y evaluado después bajo costos altos.

Finalmente, el espacio de acciones discretas podría ampliarse con plantillas adicionales: baja volatilidad, paridad de riesgo simplificada o combinaciones específicas entre renta variable y renta fija. Cada nueva acción añade expresividad a la política pero también aumenta la complejidad del aprendizaje. Cualquier ampliación debería evaluarse con el mismo esquema walk-forward para verificar que el beneficio en flexibilidad no se traduce en peor generalización fuera de muestra.

Bibliografía

[1] Markowitz, H. (1952). Portfolio selection. *The Journal of Finance*, 7(1), 77–91.

[2] Merton, R. C. (1969). Lifetime portfolio selection under uncertainty: The continuous-time case. *The Review of Economics and Statistics*, 51(3), 247–257.

[3] Puterman, M. L. (2014). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley.

[4] Moody, J., & Saffell, M. (2001). Learning to trade via direct reinforcement. *IEEE Transactions on Neural Networks*, 12(4), 875–889.

[5] Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press.

[6] Jiang, Z., Xu, D., & Liang, J. (2017). A deep reinforcement learning framework for the financial portfolio management problem. *arXiv:1706.10059*.

[7] Liu, X.-Y., Yang, Y., Wang, J., et al. (2020). FinRL: A deep reinforcement learning library for automated stock trading in quantitative finance. *NeurIPS 2020 Workshop on Robust AI in Financial Services*.

[8] Liu, X.-Y., Yang, Y., Wang, J., Wang, J., Lai, K. K., & Zhang, R. (2022). FinRL-Meta: Market Environments, Benchmarks and DataHub. *NeurIPS 2022 Datasets and Benchmarks*.

[9] Li, Y., Jiang, Y., Zhang, Z., & Chen, Y. (2023). Deep reinforcement learning for portfolio management: A survey. *Expert Systems with Applications*, 211, 118529.

[10] Sharpe, W. F. (1966). Mutual fund performance. *Journal of Business*, 39(1), 119–138.

[11] Sortino, F. A., & Price, L. N. (1994). Performance measurement in a downside risk framework. *Journal of Investing*, 3(3), 59–64.

[12] Young, T. W. (1991). Calmar ratio: A smoother tool. *Futures*, 20(1), 40–41.

[13] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518, 529–533.

[14] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal Policy Optimization Algorithms. *arXiv:1707.06347*.

[15] OpenAI. (2018). *Spinning Up in Deep Reinforcement Learning*. Recuperado de https://spinningup.openai.com/.

[16] Schulman, J., Moritz, P., Levine, S., Jordan, M. I., & Abbeel, P. (2016). High-dimensional continuous control using generalized advantage estimation. *ICLR*.